**Technische Universität München**

TUM School of Computation, Information and Technology

# Platform Emergence in Incumbent Firms – Insights from an Automotive Case Study

Niklas M. Weiß, Master of Science

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften
(Dr. rer. nat.)

genehmigten Dissertation.

| | | |
|---|---|---|
| Vorsitz: | | Prof. Dr. Hans-Joachim Bungartz |
| Prüfer der Dissertation: | 1. | Prof. Dr. Helmut Krcmar |
| | 2. | Prof. Dr. Jens Foerderer |

Die Dissertation wurde am 14.09.2022 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 30.06.2023 angenommen.

# Danksagung

Die vorliegende Arbeit ist im Rahmen des Doktorandenprogramms der BMW Group *ProMotion*, sowie der Betreuung des Lehrstuhls für Wirtschaftsinformatik der Technischen Universität München entstanden. Ich möchte mich an dieser Stelle bei den vielen Beteiligten und Unterstützern bedanken, ohne die diese Arbeit nur schwer vorstellbar gewesen wäre.

Mein erster Dank gilt Herrn Prof. Dr. Helmut Krcmar, der sich nicht nur offen für eine Zusammenarbeit im Rahmen einer Industriepromotion gezeigt hat, sondern durch seine engagierte und persönliche Betreuung maßgeblich zur erfolgreichen Erstellung dieser Arbeit beigetragen hat. Besonders dankbar bin ich außerdem Herrn Dr. Maximilian Schreieck sowie Herrn Prof. Dr. Manuel Wiesche. Als Mentoren konnte ich stets auf ihre Unterstützung bauen und mehr als einmal haben Diskussionen und Anregungen zu maßgeblichen Verbesserungen meiner Arbeit beigetragen. Gleiches gilt für die anderen Promovierende am Lehrstuhl die mich stets herzlichen in ihrer Mitte aufgenommen haben und immer für einen Austausch zu Forschungsthemen und Nicht-Forschungsthemen bereit waren.

Weiterer Dank muss allen meinen Kollegen und Förderern bei der BMW Group gelten. Besonders sind hierbei Philipp Rottgardt, Laura Sophie Brandt sowie James Mallinson zu nennen die mir die Aufnahme in das Doktorandenprogramm der BMW Group erst möglich gemacht haben und im Laufe der Promotion stets unterstützend zur Seite standen. Das gleiche gilt für alle Kollegen im Bereich der Fahrzeugvernetzung. Stellvertretend für sehr viele Menschen sind dabei Max Michel, Andreas Hildisch und Astrid Mahrla zu nennen, die es mir sehr leicht gemacht haben mich in meiner Rolle als Doktorand im Team wohlzufühlen und mir stets als erste Interviewpartner zur Verfügung standen.

Schlussendlich muss ich meiner Familie und meiner Freundin danken. Mir fällt es schwer in Worte zu fassen wie dankbar ich für die Unterstützung, das Interesse und das Vertrauen meiner Eltern in meinen Werdegang bin. Nur dadurch konnte es überhaupt passieren, dass ein mittelmäßiger Schüler zu einem interessierten Studenten und schließlich zu einem Promovierenden werden konnte. Das dieser trotz Berg- und Talfahrten schlussendlich eine Arbeit einreichen konnte ist meiner Freundin Laura zu verdanken. Ohne deine Unterstützung, Geduld und Liebe hätte ich es nicht geschafft.

Pfaffenhofen an der Ilm, 03.09.2022 Niklas Weiß

# Abstract

**Problem Statement:** The rise of digital platforms caused an exceptional transformation of multiple industries. Their generative capabilities facilitate innovation and accelerate the development of digital products. Because of this “platform revolution”, competition no longer revolves around the control of classic value chains but around attracting generative activities associated with a digital platform. Incumbent companies are forced to break away from traditional development practices to compete with the massive innovation power of platform-centric firms. As one important example, the automotive domain is heavily affected by disruptive effects that are caused by digital platforms. Car manufacturers strive for an embracement of digital innovation by implementing digital platforms in their vehicles. However, this change requires the manufacturers to adjust or create new capabilities. Unfortunately, while matured platforms of digital-native players are comprehensively investigated in IS research, the emergence of new digital platforms remained in the dark so far. Therefore, this thesis strives to shed light on the emergence of digital platforms from the perspective of an incumbent firm.

**Research Design:** Exploring and theorizing on digital platform emergence entails the methodological challenges of gaining access to empirical data. The subject of our study reveals a complex and dynamically evolving character. Furthermore, the collection of insights from a digital platform before its actual launch appears challenging. Therefore, we focus on one single case and strive for an in-depth investigation of the observed phenomena. The cooperation with BMW as globally operating car manufacturer allows detailed insights in the emergence of a digital platform for automotive onboard apps. In the course of this collaboration, we conducted 60 interviews with experts from different disciplines that were involved in the design of BMW’s digital platform for automotive onboard apps. Beside that we collected comprehensive secondary data that allowed the triangulation of the insights during our analysis with grounded theory methodology. Moreover, we engaged in an action research approach that provided us further insights in the daily work of BMW’s platform team that was responsible for the design and the governance of the focal digital platform.

**Results:** This thesis provides manifold empirical insights. First, we illustrate that an unexperienced platform owner acquires skills and knowledge on platform governance though interactions with complementors. Second, we identify three learning mechanisms of a platform owner that enhance the design of an emerging digital platform. Third, we prove that the involvement of lead complementors benefits the design of an emerging digital platform. Fourth, we describe three strategical options for an incumbent firm that is confronted with the emergence of platform ecosystems for their product. Eventually, we provide the illustration of challenges and opportunities of the BMW Group as an incumbent firm that is confronted with the platformization of its product in a comprehensive case description that facilitates the teaching of the digital platforms for undergraduate as well as graduate students of Information Systems.

**Contribution**: The results of this thesis contribute to theory as well as practice. The implications for theory focus on contributions to research on digital platforms, platform emergence and incumbent firms. The description of a platform owner’s activities during the emergence of the digital platform extends literature on the platform lifecycle and provides

insights how a platform is designed. From a practical perspective, this thesis provides insights on an incumbent firm that face the platformization of their product.

**Study Limitations:** Among others, this study is subject to three major limitations. First, it is limited to one single case. While this study provides detailed, long-term insights on the platform emergence in one incumbent firm, it is limited to the context of the BMW Group. Second, by nature the period of our investigation is limited and not able to illustrate the full emergence of BMW's digital platform for automotive onboard apps. Third, our qualitative research approach is subject to a generalizability and reporting bias that, even though it was considered, cannot be fully neglected.

**Future Research:** Our results suggest five avenues for future research. First, we propose to extend the results of our single-case study on platform emergence to a broader scope and more cases. Second, we see the application of organizational learning for platform owners in others platform lifecycle phases then the platform emergence as promising field for future scholars. Third, we argue that scholars should investigate the transformation from platform emergence to platform startup in the platform lifecycle. Fourth, we consider the investigation of consortium-owned digital platforms as fruitful avenue for future research. Eventually, our research provides a detailed description of BMW's strategical considerations in the context of platform emergence. We see demand for further investigations on the strategical management of uncertainty in the context of platform emergence.

# Table of Contents

# List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| API | Application Programming Interface |
| BMW | Bayerische Motoren Werke |
| GTM | Grounded Theory Method |
| IS | Information Systems |
| IT | Information Technology |
| OEM | Original Equipment Manufacturer |
| OS | Operating System |
| OTA | Over the Air |
| SDK | Software Development Kit |
| USB | Universal Serial Bus |

# Part A

# 1 Introduction

"*The new ways in which companies are going to survive are different then last century, where it was about mastering the supply and economics of scale and scope. […] We've realized that the old concept that we had that if you are in one industry - all you need to worry about is your rivals – is completely gone.*" ~ Annabelle Gawer, Davos Economic Forum 2019

This quote from Annabelle Gawer, one of the leading scholars on digital platforms and platform ecosystems and advisor of the European Commission on the regulation of digital platforms, indicates the fundamental changes several traditional industries are currently faced with due to the rise of the digital platform era. This doctoral thesis aims to build an empirical understanding of dynamics and mechanism of platform emergence from the perspective of an incumbent firm. Therefore, a comprehensive single case study at the BMW Group, one of the largest, globally acting premium car manufacturers, was conducted. By providing a theoretically grounded understanding, the thesis fills several research gaps on the emergence of digital platforms, platform design and platform strategy. The following first chapter describes motivations and contains detailed problem statements for the three research questions that were investigated in the context of this research project.

## 1.1 Motivation and Goal of the Thesis

In May 2021, Forbes published its latest report on the 2000 largest companies listed on the stock exchange[1]. Like in previous years of the last decade, companies from the information technology (IT) sector dominated the top positions in the ranking. For a large share of people Google's services as Maps or Search, Apple's iPhone and all its apps, Metas's Instagram and Amazon's deliveries became an integral part of their daily live. Despite their comparatively young age these firms achieved that billions of people consume their products and services every day. Even if they differ significantly in aspects such as corporate culture, number of employees or their value proposition, all these companies have one thing in common: Their extraordinary rapid growth and their subsequent dominance is grounded in the concept of digital platforms (Cusumano et al., 2019; Parker, Van Alstyne, & Choudary, 2016; Tiwana, 2014). A digital platform connects different stakeholders in a multi-sided marketplace and enables transactions between them (Rochet & Tirole, 2003). For example, Apple's App Store enables app developers to provide their products on the one side to users of the iPhone on the other side. Thereby, the generative capability of the digital platform enhances innovation. From a technical perspective a digital platform can be considered as "the extensible codebase of a software-based system that provides core functionality shared by apps that interoperate with it, and the interfaces through which they interoperate" (Tiwana et al. (2010), p. 675). The platform design is defined as a blueprint of the partitioning of a "relatively stable platform and a complementary set of apps that are encouraged to vary, and the design rules binding to it" (Tiwana et al. (2010), p. 675). It affects technical as well as organizational aspects and is critical for the evolution of a platform (Asadullah et al., 2018). While external complementors create new apps and services that benefit the user, Apple as platform owner orchestrates all activities in its iOS ecosystem. Considering a technical perspective, platform design covers effects like standardization and

[1] https://www.forbes.com/lists/global2000/#5b81919c5ac0 retrieved 31.07.2021

modularization (Baldwin & Woodard, 2008). Organizational aspects consider the management of network effects beyond different stakeholders of the platform (Eisenmann et al., 2011). The governance of these stakeholders is the "partitioning of decision-making authority between platform owner and app developers, control mechanisms, and pricing and pie-sharing" (Manner et al., 2013; Tiwana, 2014). Thereby, a platform owner can adjust the platform design, for example to leverage scale and/or scope (Eisenmann et al., 2009) or to gain back a desired level of control (Ghazawneh & Henfridsson, 2013). In this way, platforms evolve over time and reveal multiple stages (Staykova & Damsgaard, 2017).

Despite the knowledge that a digital platform changes during its lifecycle, prevalent research has focused on matured platform ecosystems. Beside few exceptions (Hein, Schreieck, et al., 2019a; Amir Mohagheghzadeh & Fredrik Svahn, 2016; Schermuly et al., 2019; Svahn et al., 2017), Information Systems (IS) scholars mainly considered established digital platforms ecosystems that are orchestrated by experienced platform owners (Eaton et al., 2015; Foerderer, Kude, Mithas, et al., 2018; Henfridsson & Ghazawneh, 2013; Karhu et al., 2018). While there is comprehensive knowledge on mechanisms and processes of these successful digital platforms, the path to achieving a matured status remains unclear (de Reuver et al., 2017; McIntyre & Srinivasan, 2017). Within their broadly recognized "research agenda on digital platforms", de Reuver et al. (2017) shift the focus towards platform emergence. They raise the question if a successful digital platform can be consciously designed or if it is just a result of coincidences. One reason for this gap is caused by the difficult access to data. A platform that is not publicly announced is hard to identify for scholars. Hence, the launch of a platform is the first phase in a platform lifecycle that is considered by research (Stummer et al., 2018). Even though the platform may be refined over its whole lifetime (Eaton et al., 2015), its initial design is shaped earlier. Considering the large challenges of a platform launch (Stummer et al., 2018), the platform design is particularly critical in this phase.

The free availability of the iOS Software Development Kit (SDK) for hundreds of thousands app developers, enabled gross digital innovation in the App Store, which enacts an elemental driver of the iPhone's and eventually Apple's remarkable success story of the last decade. By connecting different sides of a market, platform companies have transformed several industries in the last decade. Airbnb is the largest broker of holiday apartments without owning any hotel facility[2], Uber replaces traditional taxis all over the globe without owning a fleet of cars[3] and Facebook can be considered as largest news distributor on planet without running any editorial office[4]. Incumbent companies often required many decades and large investments to be able to demonstrate comparable successes. However, platform companies often outperform these companies within a few years. As a consequence of this "platform revolution" (Parker, Van Alstyne, & Choudary, 2016), in several industries competition no longer revolves around the control of classic value chains that reveal one product that is sold to a customer but around attracting generative activities associated with a digital platform that connect different sides of a market (de Reuver et al., 2017). Therefore, incumbent companies are forced to break away from traditional practices to compete with the massive innovation power of platform-centric firms (Svahn et al., 2017; Yoo et al., 2010). However, while companies as Airbnb or Uber are

---

2 https://www.statista.com/topics/2273/airbnb/#dossierKeyfigures retrieved 17.12.2021

3 https://www.nytimes.com/2020/01/12/business/los-angeles-taxis-uber-lyft.html retrieved 17.12.2021

4 https://harvardpolitics.com/replacing-newsfeeds-for-newspapers-the-detriments-of-using-facebook-for-news ret. 17.12.2021

founded and built around a platform business, traditional firms are optimized to sell products (Porter, 1980). The adaption or extension of these existing capabilities embodies a major challenge for their organization (Bharadwaj et al., 2013; Dyer & Singh, 1998; Teece et al., 1998). Also, the automotive domain is heavily affected by disruptive effects that are caused by the rise of digital platforms. By the introduction of CarPlay and Android Auto, Apple and Google are entering the car's cockpit and thereby penetrate the core territory of traditional car manufacturers. In 2017 Google announced that its Android mobile operating system (OS) should be enabled to become the operating system for cars of the future. The massive innovative power of digital platforms in other domains, as well as the first penetration of foreign platforms into the automotive industry, pushed automotive Original Equipment Manufacturers (OEMs) to think about own digital platforms (Svahn et al., 2015; Svahn et al., 2017; Weiss, Schreieck, Wiesche, et al., 2018). However, this change requires the manufacturers to adjust (Schreieck & Wiesche, 2017b; Tilson et al., 2010; Tiwana et al., 2010) or create new capabilities (Schreieck et al., 2021; Svahn et al., 2017). Considering the lack of knowledge on this earliest phase of the platform lifecycle, **this dissertation project aims for the development of an empirical understanding of platform design in an emerging digital platform ecosystem from the perspective of an incumbent firm**.

## 1.2 Research Questions

The thesis' goal is to understand how digital platforms are designed and how they evolve during emergence. Exploring and theorizing on digital platform emergence entails the methodological challenges of gaining access to empirical data (de Reuver et al., 2017) while subject of our work reveals a complex and dynamically evolving character. Therefore, we focus our endeavors on one single case and strive for an in-depth investigation of the observed phenomena (Yin, 2009). Towards this end, we first evaluate how a platform owner can acquire skills to design a digital platform (RQ1). Second, we analyze how initial complementors can be involved in the process of designing a platform (RQ2). Third we focus on platform maturity and investigate in which way platform design is affected by opening the digital platform (RQ3).

***RQ1:*** *Which governance mechanisms enable a platform owner to learn digital platform design?*

For answering our first research question, we followed an in-depth single case study approach (Yin, 2009). After conducting semi-structured expert interviews (Gläser & Laudel, 2009), we applied a partial portfolio approach of the grounded theory methodology (Wiesche et al., 2017a). Scientific investigations on digital infrastructure evolvement show that alterations may only be observed in the long run (de Reuver et al., 2017; Tilson et al., 2010). For this reason, we extend our investigations by a longitudinal perspective. By conducting 30 semi-structured interviews with platform engineers, app developers and business owners in the period of 2 years, we exploited the actual evolvement of the platform design. Within this time the platform was released to customers and several new applications started their (internal) development process. Based on the analysis of 25 episodes, we identify transfer of perspective, transfer of knowledge and transfer of artifacts as basic mechanisms that enable a platform owner to enhance the design of its digital platform. The inherent improvements of the platform resources entail significant improvements for app developers as automation of processes, more powerful tooling, or more comprehensive documentation. Our work extends the existing theory on platform emergence and provides rare insights into the learning process of an inexperienced platform owner (Weiss, Schreieck, Wiesche, et al., 2018).

***RQ2:*** *How can a platform owner involve complementors to enhance digital platform design?*

While multiple incumbent firms struggle in designing digital platforms (Cusumano et al., 2019; Svahn et al., 2017), literature considers the involvement of users as an essential success factor for designing new information systems (Bano & Zowghi, 2014). The influence of app developers on platform boundary resources in tuning mechanisms (Eaton et al., 2015) is generally acknowledged. However, the conscious exploitation of these mechanisms by the platform owner was not considered so far. Therefore, we propose the approach of active lead user involvement (Füller et al., 2014; von Hippel, 1986) for the transformation of an existing automotive onboard architecture into a digital platform design. Originally, lead user involvement is rooted in marketing research and implies the involvement of users whose present strong needs will become general in the future (von Hippel, 1986). This conforms to the internal app developers that are impeded by insufficient platform design. By considering internal app developers as “lead complementors”, the shift from a customer-centric product design process towards a customer- and developer-centric platform design process should be enhanced. This change enables the creation of generativity that enables app development for programmers without domain specific knowledge (Pühler, 2011).

To prove the efficiency of the approach in the context of digital platform design, we apply an action research method in a first step (Baskerville, 1999; Frank et al., 1998). While the existing onboard systems adjust all functional demands for an up-to-date infotainment system, it lacks generic capabilities (Weiss, Schreieck, Brandt, et al., 2018). The complexity and proprietary character of the exposed APIs are an illustrative example for this shortcoming. By involving developers in the design and development process of platform APIs, we derive challenges and benefits of lead complementor involvement in platform design (Weiss et al., 2019). Thereby, we introduce real change by refining 5 onboard APIs (including 59 methods and 30 properties) and provide detailed insights in the transformation of an incumbent’s technology into a digital platform. We plan to extend our work by analyzing the effects of the applied approach by illustrating the utilization of the developed APIs.

***RQ3:*** *What are strategic options on digital platforms for incumbent firms that are confronted the transformation of their products into a digital infrastructure?*

While a digital platform may exist in an internal stage for the moment, it can be opened towards external partners as the platform owner desires to leverage scale and/or scope by the establishment of a vibrant ecosystem. The evolvement of cars as digital infrastructure (Constantinides et al., 2018; Hanseth & Lyytinen, 2010) reveal the formation of digital platforms for cars. Prior research has already illuminated in which way an incumbent firm adopts platform technology (Hein, Schreieck, et al., 2019a; Rolland et al., 2018) and responds to the entry on multi-sided markets (Seamans & Zhu, 2014). However, platform research also showed that incumbent firms implement different strategies to leverage a platform business (Cusumano et al., 2019; Parker, Van Alstyne, & Choudary, 2016). Also, our data reveals three options for a digital platform for automotive apps: (1) build a proprietary platform, (2) build a collaborative platform, or (3) join an existing platform. Furthermore, a high degree of uncertainty and a subsequent focus on agility and flexibility in the transformation (Eklund & Kapoor, 2019) can be identified. In our next step, we plan to examine these options (Rolland et al., 2018; Svahn et al., 2015) regarding the firms resources (Barney, 1991; Rogan & Greve, 2015) as well as relations to other players (Dyer & Singh, 1998; Koch & WIndsperger, 2017).

In this way we strive to derive new theoretical insights on the diffusion of digital platforms into incumbent industries.

## 1.3 Structure of the Thesis

This thesis is basically structured into three major parts. **Part A** begins with a general introduction (A1) to the topic, including a motivation for research on digital platform and platform emergence in the context of incumbent firms and the subsequently derived research questions. It is followed by an illustration of the conceptual background (A2) that outlines all basic theoretical concepts on which the research that was conducted in the context of this thesis was built on. Subsequently, part A is closed by a section on the research design (A3) that describes the overall mixed method approach of this work and provides an overview on all methodologies that were applied. **Part B** consists of five publications that were conducted to answer the previously raised research questions. While research question one and two and consequently also the inherent publications P1 – P3 focus on dynamics inside an emerging platform ecosystem, research question three with its associated publications P4 and P5 consider the interrelations to other platform ecosystems that emerge simultaneously. The final **Part C** summarizes (C9) and discusses (C10) the collected results, explains implications for theory and practice (C11), and describes limitations of our research approach in the context of this thesis (C12). Eventually, several fruitful avenues for future research that build on insights that were gained in the context of this thesis are illustrated (C13). Figure 1 provides an overview on the described structure of the thesis.

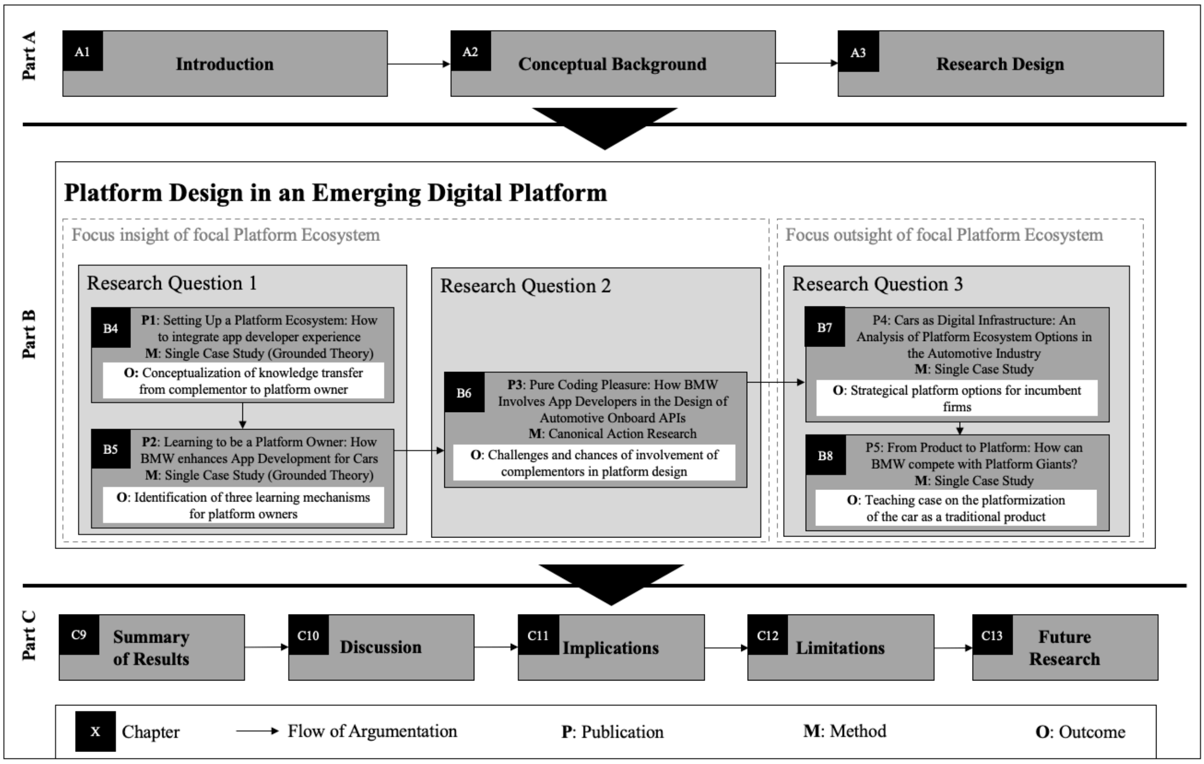



**Figure 1: Overview on thesis structure**

The following paragraphs provide summaries of all five publications that are embedded in this thesis. In doing so, the research problem, the applied methodological approach as well as the main contribution are described.

**P1**: The first paper, *"Setting Up a Platform Ecosystem: How to integrate app developer experience"* provides insights on an unexperienced platform owner who acquires knowledge on the governance of a digital platform through interactions with app developers. Therefore, we conducted a single case with the application of Grounded Theory procedures for the analysis of the collected data set. The study confirms the theory of knowledge boundaries between platform owner and complementors and extends the concept by the aspect of knowledge transfer from complementor to platform owner.

**P2**: The second paper, *"Learning to be a Platform Owner: How BMW enhances App Development for Cars"* extends the research of P1 and investigates different kinds of mechanisms that enable the knowledge transfer from complementor to platform owner. Therefore, we transferred the concept of organizational learning and implemented it in theory on platform governance. Our results are grounded in 30 expert interviews that we conducted during a period of two years and are enriched by extensive secondary data. Eventually, we identified the transfer of perspective, the transfer of knowledge, and the transfer of artifacts as basic mechanisms that enable a platform owner to learn and enhance its digital platform.

**P3**: The third paper, *"Pure Coding Pleasure: How BMW Involves App Developers in the Design of Automotive Onboard APIs"* also builds on the realization of knowledge flows from complementors to the platform owner. However, in contrast to P1 and P2, where the flow of knowledge from complementors to platform owners was passively observed, this study strives for an active engagement of complementors by the platform owner. Inspired by the idea of lead user involvement, the platform owner should actively involve app developers for the design refinement of platform APIs. Therefore, we conducted a canonical action research and accompanied the refactoring of five different platform APIs. Thereby, we identified critical challenges and benefits for the involvement of complementors in the design of digital platforms, especially in the context of an incumbent firm.

**P4**: The fourth paper, *"Cars as Digital Infrastructure: An Analysis of Platform Ecosystem Options in the Automotive Industry"* changes the context from research insight the platform ecosystem (as in P1, P2 and P3) to an investigation of dynamics of platformization on an industry level. The study illustrates the transformation of the car from an analog product to a digital infrastructure. The study is based on 42 expert interviews in the automotive industry and extracts three strategical options that are available established car manufacturers: (1) Build a proprietary platform ecosystem, (2) build a collaborative platform ecosystem or (3) join an existing platform ecosystem.

**P5**: The fifth paper, *"From Product to Platform: How can BMW compete with Platform Giants?"* embodies a teaching case that illustrates the challenges for an incumbent car manufacturer as BMW that experiences the platformization of its product. Thereby, we describe the development of BMW from a traditional car manufacturer to a technology company and the influences of different actors inside as well as outside the automotive industry. Especially, the entrance of Google as provider of a digital platform for cars raises several strategical challenges that are described in the paper.

| No. | Authors | Title | Outlet | Type |
|---|---|---|---|---|
| **P1** | Weiss, Schreieck, Wiesche, Krcmar | Setting Up a Platform Ecosystem: How to integrate app developer experience | ICE/ITMC 2018 (published) | CONF (unranked) |
| **P2** | Weiss, Schreieck, Wiesche, Krcmar | Learning to be a Platform Owner: How BMW enhances App Development for Cars | IEEE TEM 2020 (published) | JNL (VHB: B) |
| **P3** | Weiss, Wiesche, Krcmar | Pure Coding Pleasure: How BMW Involves App Developers in the Design of Automotive Onboard APIs | AMCIS 2019 (published) | CONF (VHB: D) |
| **P4** | Weiss, Schreieck, Wiesche, Krcmar | Cars as Digital Infrastructure: An Analysis of Platform Ecosystem Options in the Automotive Industry | III 2019 (published) | CONF (unranked) |
| **P5** | Weiss, Schreieck, Wiesche, Krcmar | From Product to Platform: How can BMW compete with Platform Giants? | JITTC 2020 (published) | JNL (VHB: B) |
| ICE/ITMC: IEEE International Conference on Engineering, Technology and Innovation; IEEE TEM: IEEE Transactions on Engineering Management; AMCIS: Americas Conference on Information Systems; III: Innovation in information infrastructures workshop; JITTC: Journal of Information Technology Teaching Cases<br>JNL: Journal, CONF: Conference<br>VHB: German Academic Association for Business Research. | | | | |

**Table 1: Overview on embedded publications**

## 1.4 Remarks on Format

The formatting styles of the original publications were different from one another. For consistency, the original research works have been reformatted for this thesis, applying a uniform formatting style. The tables and the figures were redesigned and unified throughout the entire document. Furthermore, the tables and the figures were numbered sequentially across all parts of the thesis. Consequently, the numbering of figures, tables, and in-text-references differ from the original publications. The original section numbering in the publications was substituted by the overarching structure of the present document. Each major part (A, B, and C) comprises a unique numerical structure of sections and subsections. For simplicity, all references to other sections of the same part omit the indication of the part number. Minor editorial revisions to the original publications were also made (e.g., adaptation of reference styles, orthographical and minor grammatical revisions).

# 2 Conceptual Background

The following section illustrates theoretical concepts that that we build on in this thesis. We provide an overview on digital platforms, digital platform ecosystems, platform emergence and incumbent firms. While these topics are relevant for the overarching context of this thesis, theories on platform governance and platform architecture are particularly relevant for research questions one and two, which focus on dynamics insight an emerging platform ecosystem. The concept of platform options, on the other side is especially relevant for the consideration of dynamics outside the focal platform ecosystem, which are investigated in research question three[5]. **Fehler! Verweisquelle konnte nicht gefunden werden.** provides an overview on the illustrated theoretical concepts in the context of the overall structure of this thesis.

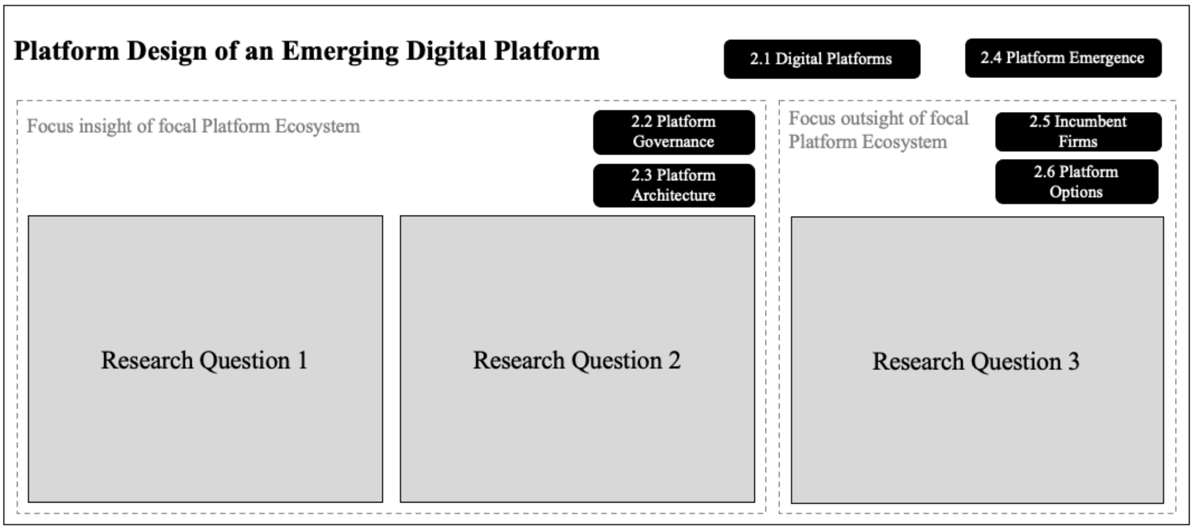



**Figure 2: Overview on theoretical concepts in the context of the overall thesis structure**

## 2.1 Digital Platforms

The tremendous success as well as the omnipresence of digital platforms and digital platform ecosystems has led to a growing interest of academia on the research of the phenomena (Cusumano et al., 2019; de Reuver et al., 2017; Parker, Van Alstyne, & Choudary, 2016; Tiwana, 2014). Its diversity as well as the high level of research activity around it revealed a broad spectrum of topics on digital platforms. The following section provides an overview on the most frequently discussed aspects in IS research and should convey the general understanding of the "digital platform" term that was applied in the context of this thesis.

Hein et al. (2020) consider three typologies of definitions for digital platforms: (1) a market-oriented perspective, (2) a technology-oriented perspective, and (3) a socio-technical perspective. All these perspectives have in common that they view digital platforms embedded in ecosystems, connecting different sides of actors (e.g., users and complementors) and enable interactions between those sides (Hagiu & Wright, 2015; Rochet & Tirole, 2003).

---

[5] The assignment of theoretical concepts to research questions should facilitate the structuring of this thesis and support its articulation. However, it cannot be neglected that for instance theories on platform architecture and platform governance influenced our investigation of platform options in RQ3. However, platform options embody the key concept in this research. Therefore, Figure 2 should provide guidance for the reader but should not be understood as strict classification.

### 2.1.1 The Market-oriented Perspective on Digital Platforms

The market-oriented perspective (Schreieck et al., 2016) compares digital platforms to markets that enable interactions between different sides. Platforms are *"markets, where users' interaction with each other are subject to network effects and are facilitated by a common platform provided by one or more intermediaries"* (Eisenmann et al. (2011), p. 1273). While app developers create new apps on the one side, the integrated iOS App Store promotes these apps to users of the iPhone on the other side. Both sides are part of the *iOS* platform ecosystem (Jacobides et al., 2018; Parker et al., 2017; Sorensen, 2015). Thereby, the platform enables reliable transactions of value between autonomously acting complementors on one side and consumers on the other side (Lusch & Nambisan, 2015). Whenever a user purchases an app, the digital platform as an intermediary manages the transaction and ensures that each side receives the stipulated value, namely that the user can download the app and the app developer receives their money in return. By enabling interactions that are beneficial for both sides, the platform itself creates value (Evans, 2012; Rochet & Tirole, 2003), which can be captured by the platform owner (Rietveld et al., 2016). Finding the ideal degree of value capture in different lifecycle phases and situations embodies a major challenge for a platform owner (Hein et al., 2020). Independent from the degree of value capture, every new participant that enters the ecosystem is beneficial for all others. Every new app in an app store might attract new users, which embody potential customers also for other apps in the store. The more users on the other side are participating, the more app developers are attracted to build high-quality apps that are again available for all users. These self-reinforcing feedback loops (Parker, Van Alstyne, & Choudary, 2016), mostly called network effects (or network externalities (Katz & Shapiro, 1985)) refer to "*the degree to which every additional user of a platform or app makes it more valuable to every other existing user*" (Tiwana (2014), p.40). Network effects embody a central enabler for the tremendous success of digital platforms and superiority of digital platform business model (Parker et al., 2017) in comparison to traditional "pipeline" businesses (Parker, Van Alstyne, & Choudary, 2016) in multiple domains. In this regard, the chicken-and-egg-dilemma constitutes a commonly discussed issue in research (Hagiu & Wright, 2015; Stummer et al., 2018): When a digital platform with latent network effects starts off, generally neither of the platform's stakeholder groups (e.g., users and app developers) has an inherent incentive to join the platform, unless the other side reveals a relevant size with the potential to enable value exchange. On the one hand, an app developer, expect a certain number of potential users that might pay or subscribe to their app. On the other hand, users expect enough available apps for their phone. Even though research identified several strategies to overcome this chicken-egg problem (Stummer et al., 2018), the kick-off of network effect embodies a crucial challenge for emerging platform ecosystems.

### 2.1.2 The Technology-oriented Perspective on Digital Platforms

Another, namely the technology-oriented perspective on digital platforms was defined by Tiwana (2014, p. 5). He describes a digital platform as "*a software-based product or service that serves as a foundation on which outside parties can build complementary products or services*". Tiwana's definition is based on the platform definitions of Baldwin and Woodard (2009) and Eisenmann et al. (2006). Baldwin and Woodard (2009, p.3) provide a more general definition and define a platform as *"a set of stable components that supports variety and evolvability in a system by constraining the linkages among the other components"*. In this sense Apple's iPhone and its iOS operating system embody a digital platform that provide a technological base for developers, who create new digital products, called applications ("apps"). The technology-oriented perspective considers the code-related and expandable nature of digital platforms, their interoperating subsystems (i.e., modules or apps) and their interface (i.e., the technical environment that enables them to interoperate with modules (Tiwana et al., 2010). Via so-called application programming interfaces (APIs), app developers have access to a broad variety of the phone's features like the camera or the audio speakers as well as sensor data like gyroscope or GPS. These characteristics of digital platforms in combination with a platform architecture consisting of both stable and variable components allow for evolvability of the platform (i.e., to adapt to a changing environment) (Baldwin & Woodard, 2009a). Digital platforms are thus versatile enough to react to a changing environment (Baldwin & Woodard, 2009), while providing the stability and the resources for complementors to develop and to integrate apps (Hein et al., 2019; Tiwana et al., 2010).

### 2.1.3 The Socio-technical Perspective on Digital Platforms

Lastly, the socio-technical perspective concentrates on the implementation of an ecosystem of actors and their organization (de Reuver et al., 2017; Hein et al., 2020). As the central construct in a socio-technical perspective, platform governance relates to the "who" and the "what" of platform-based decision-making (Schreieck, Hein, et al., 2017; Tiwana et al., 2010). In that regard, several strategically important decisions must be made by the platform owner. A high degree of openness enables massive potential for innovation on the one end and the potential loss of control on the other hand (Henfridsson & Ghazawneh, 2013; Ondrus et al., 2015). The design and availability of platform boundary resources – as software development kits or development tools for app developers – decides on the capabilities of complementors to create complementary value of the ecosystem (Ghazawneh & Henfridsson, 2010a). By adapting the provided boundary resources (Ghazawneh & Henfridsson, 2010a), a platform owner can control how actors interact within the ecosystem (Hein et al., 2019) and how value is subsequently created and captured (Song et al., 2018). Furthermore, the artefacts serve as tools for communication and collaboration between platform owner and app developers (Foerderer et al., 2019) and enable the platform owner to control the platform ecosystem (Henfridsson & Ghazawneh, 2013). Furthermore, the platform owner might install processes, decide on options for monetization (Boudreau & Jeppesen, 2014) and install input and output control (Tiwana, 2014; Tiwana et al., 2010). Eventually, different modes of platform ownership, either central, consortium-based, or decentral, might influence decisions on platform governance and consequent dynamics within the platform ecosystem (Hein et al., 2020).[6]

## 2.2 Platform Governance

The core of a digital platform is in the center of every digital platform ecosystem (Tiwana, 2014). The exposed capabilities of the platform should attract app developers (Dellermann et al., 2016; Kude et al., 2012) that create complementary apps on the one side, which should be consumed by customers on the other side (Tiwana, 2014). Platform governance is the *"partitioning of decision-making authority between platform owners and app developers, control mechanisms, and pricing and pie-sharing structures"* (Tiwana (2014), p. 25). For instance, in the iOS ecosystem, Apple as platform owner decides, which application

---

[6] This last section of chapter *2.1 Digital Platforms* outlines aspects of platform governance to clarify the socio-technical perspective on digital platforms. However, platform governance itself embodies a central concept for this thesis. Therefore, chapter *2.2 Platform Governance* contains a dedicated, more detailed illustration of the concept.

programming interfaces (APIs) of the iPhone are accessible by third-party app developers (Eaton et al., 2015; Henfridsson & Ghazawneh, 2013). Furthermore, each app needs to be submitted for review by Apple before it can be launched in the App Store. The app developer needs to choose the price of its app in a predefined selection of pricing steps and keeps just seventy percent of its revenue. The rest belongs to Apple as platform owner. However, platform governance should also facilitate the creation of new apps (Kude et al., 2012). By exposing generativity the platform owner unleashes potential for innovation (Tilson et al., 2010). With the annual release of a new iOS version, Apple publishes new APIs that provide new possibilities for app developers. Comprehensive documentation and annual developer conferences inform and attract developers to build new apps. A platform with strictly controlled generativity would counter such endeavors (Ondrus et al., 2015). Hence, the platform owner needs to balance control over the digital platform and its ecosystem on the one side and provide a certain level of autonomy for app developers to foster innovation on the other (Rausch et al., 2012). In this way, platform governance covers multiple tactical decisions that impact interactions between the platform owner and app developers (Kude et al., 2012; Schreieck et al., 2016). The following sections illustrate platform openness and platform boundary resources as two of these platform governance aspects that were especially relevant in the context of this thesis.

### 2.2.1 Platform Openness

Openness in the context of platforms was coined by Eisenmann et al. (2009) as the easing of restrictions for the participation, development, commercialization, and use. Thereby, vertical and horizontal openness can be distinguished. A platform can be opened horizontally by giving up control and interoperating with rivals, licensing the platform to other platform providers or the rather radical broadening of sponsorship. Interoperating with rival platforms is done, if a positive effect on the market size, share and profit margin is expected (Eisenmann et al., 2009; Soto Setzke et al., 2019). By licensing the platform to others, the platform provider can capture additional revenue shares and retain platform technology control (Eisenmann et al., 2009). The degree of openness of a platform lies in the continuum between the extremes completely closed and completely open (West, 2003). On one side closed platforms are owned and controlled by a single platform owner and are proprietary (Boudreau, 2010). The access to the platform is protected by intellectual property rights, secrecy, copyright, and other methods and thereby hinders other stakeholders' usage of the platform in a not by the platform provider intended way (West, 2003; Huang et al., 2009; Boudreau, 2010). On the other side fully open platforms are publicly owned and grant access to all stakeholders (Boudreau, 2010). Such platforms do not have a single platform owner who controls the platform and restricts the access for certain stakeholders. Boudreau (2010) showed that the level of openness is a critical factor for the platform success. Platform owners often use openness to foster positive behaviors of complementors (Benlian et al., 2015). While increased openness yields higher chances of product success and encourages R&D spillovers (Parker et al., 2017), it can also *"destabilize a platform's ecosystem"* (Wessel et al., 2017, p.344). The destabilization of a platform might occur if the risk cycle which favors closeness is neglected and only the positive network effects are considered (Fürstenau and Auschra, 2016).

### 2.2.2 Platform Boundary Resources

Boundary resources were introduced to design and take care of the critical relationship between the platform owner and complementors more efficiently by shifting design capabilities towards the complementors (Ghazawneh and Henfridsson, 2010). Especially Ghazawneh and Henfridsson (2013, p.174) coined the term as *"the software tools and regulations that serve as the interface for the arm's length relationship between the platform owner and the application developer"*. In general, such artifacts or concepts that serve as tools for communication and collaboration of different stakeholder groups are an established unit of analysis (Baldwin & Clark, 2000; Briers & Chua, 2001; Yoo et al., 2009). Thereby, boundary resources are plastic enough to cut across several understandings of different social worlds, and concurrently reveal "enough structure to support several parties and their employed activities within separate social worlds" (Ghazawneh and Henfridsson (2010a), p. 4).

Platform boundary resources can be divided into technical boundary resources which enable to access the platform, and social boundary resources which are used to pass on knowledge and provide support (Ghazawneh and Henfridsson, 2013; Bianco et al., 2014). Technical boundary resources are directly utilized by the third-party developers (Dal Bianco et al., 2014) and play a central role in partitioning the development task between platform and apps (Boudreau, 2012; Yoo et al., 2010). Rudmark and Ghazawneh (2011) claim APIs as one of the most important types of boundary resources in third-party development. Several papers describe the provisioning of APIs to external complementors as way for an opening of the platform (Eaton et al., 2015; Henfridsson & Ghazawneh, 2013; Parker, Van Alstyne, & Jiang, 2016; Svahn et al., 2017). Ghazawneh and Henfridsson (2010b) describe a tremendous increase of third-party iOS developers after the provision of more than 1,000 APIs for Apple's iPhone. Besides the pure provisioning, also the design of APIs is considered as crucial factor for the activation of complementors (Ghazawneh, 2016). Concurrently interfaces serve as tool for control on the development of applications in the ecosystem (Bergvall-Kåreborn & Howcroft, 2014).

Documentations, support, and blogs are examples among the social boundary resources (Soto Setzke et al., 2019). In contrast to open source innovation networks that are characterized by decentralized control of fairly homogeneous knowledge resources (Ghazawneh & Henfridsson, 2010a; Yoo et al., 2009), a digital platform ecosystem is characterized by centralized control by the platform owner and relatively heterogeneous knowledge resources (Ghazawneh & Henfridsson, 2010a). Standardized platform boundary resources enable a broadcasting mechanism that allows a knowledge transfer without any interaction of app developers and platform owners (Foerderer, Kude, Schuetz, et al., 2018). This can be facilitated by centrally provided artefacts a sufficient documentation as well as question and answer forums for mutual support of developers. Yeow et al. (2017) include the management of knowledge in their conceptualization of boundary organization practices as practices enacted as part of the boundary organization to define working routines between multiple stakeholders.

## 2.3 Platform Architecture

Besides governance, architecture is the second fundamental part of platform conceptualization (Tiwana, 2014). Platforms embody modular architectures composed of two sets of modular components. The first is a core module, owned by a platform owner, and offers up functionality through interfaces of low variability to the second module, which are complementary modular components of high variability and built by third parties (Baldwin & Woodard, 2008; Eaton, 2016).

### 2.3.1 Challenges of Platform Architecture

Although, this division serves well as a fundamental conceptualization, platform ecosystems must be understood as complex systems (Tiwana, 2014) comprising of multiple smaller subsystems with interactions and interdependencies (De Weck et al., 2011). The greater the number of these subsystems, the higher the complexity. Complexity is the origin of two challenges platform vendor's face. The first challenge is to be clear and easily understood. Platform ecosystems tend to become increasingly difficult to be intelligible for a single person to grasp (De Weck et al., 2011). Secondly, complexity causes the so-called *gridlock* problem. The management of such a large web of complexity can become so daunting that responsible players lose their ability to develop their subcomponent. The high likelihood of unpredictable side-effects, caused by a poorly implemented adjustment, avoids the courage to alter the "status quo". Platform architecture should prevent such problems and make complexity manageable (Tiwana, 2014). Moreover, diversity of platforms and their players exemplifies a large challenge for platform architects. It demonstrates a platforms biggest potential strength and weakness simultaneously, since diversity may enable the development of multiple innovations or cause massive chaos. Architectural designs, that quash developer's autonomy may avoid such chaos but kill potential for innovation. To have less coordination, on the other hand, may result in an administrative mess. Therefore, platform ecosystems need to handle the balance between coordination and autonomy (Iansiti & Levien, 2004). Another aspect of handling complexity with platform architecture is cost. There are two types of costs within a platform ecosystem that can be identified. First, there are the transaction costs that occur when app developers do business with the platform owner, and secondly, there are the coordination costs incurred to manage interactions between the platform and its auxiliaries. Well-designed architecture can reduce both kinds of costs (Baldwin & Woodard, 2008). A further platform architecture assignment is the partition, or separation, and reintegration of a complex systems. Partition describes the division of the integrated system to increase clarity and manageability. Reintegration outlines the integration of the modularized components into the holistic system. A well-shaped platform architecture should support and enable both mechanisms (Tiwana, 2014).

### 2.3.2 Guidelines for Platform Architectures

Tiwana (2014) concludes that there are four properties from the abovementioned challenges for which software platform architecture should strive. First and foremost, the architecture of a platform should be *simple*, which means it should be conceptually decomposable, and functionalities and interactions should be explicit and well defined. Secondly, the architecture should be *resilient*. One malicious subcomponent should not be able to incapacitate the whole system. The third property is that the architecture should be *maintainable* which means the addition of a new component should be cost-effective and should not break existing apps. Lastly, the architecture should be *evolvable* which means that the architecture is capable to do things in the future for which it was not originally designed. An approach that may endow these properties is *modularization*. In the context of software, modularity describes the degree to which a platform and application can be designed, implemented, operated, and altered independent of each other (De Weck et al., 2011; Tiwana, 2014). Modularization quantifies developers to implement their ideas more independently and creatively and it enables massively distributed innovation that shows increased variety and volume (Tiwana, 2014). Although modularization is capable to enable a healthy platform ecosystem, the grade of modularity must be considered, and since modularity causes substantial initial costs and may reduce the platform's technical performance, it has to be limited (Tiwana, 2014). The measurement of a platform's modularization grade is a further challenge in the conception of a platform architecture.

### 2.3.3 Domain-specific Platform Architectures

Almost all literature previously mentioned, surmise their hypotheses by investigating either digital platforms in common or mobile app store platforms in special (Baldwin & Woodard, 2008; De Weck et al., 2011; Tiwana, 2014). The massive financial success makes the consideration of these platforms seem obvious. Successful patterns and strategies do not seem far off. Nevertheless, there are further industries that require committed investigations. The automotive industry provides unique challenges with regards to the conception of a platform architecture that have yet to be addressed at length in scientific literature. Different security measures are necessary in the deployment of software in cars, that can weigh between one to two tons (or more) and have the potential to move at speeds above two hundred kilometers per hour, than the deployment on a smartphone. Another component that must be considered independently is the distribution of the platform. For example, even the largest car manufacturers do not come close to selling as many products per year as *Apple* sells smartphones. Furthermore, controls within a vehicle differ substantially from the handling of a mobile device. Due to the architecture and construction of a car, the presentation of content will also differ significantly from a smartphone. Factors such as these may complicate the purchase process within the car itself, so that distribution channels outside the car may be required (as opposed to smartphone app stores, which provide a direct purchase on the phone). All these differences may yield varieties in the architecture landscape that has yet to be considered in the design of a digital platform.

## 2.4 Platform Emergence

Prevalent platform research describes several phases in the life cycle of a digital platform (Parker, Van Alstyne, & Choudary, 2016). Usually, the establishment of initial cross-side network effects is considered as birth of the digital platform (Schirrmacher et al., 2017; Stummer et al., 2018), followed by growth and eventual maturity (Parker, Van Alstyne, & Choudary, 2016). However, we argue that this assumption ignores the fact that a digital platform needs to be build and designed before it can facilitate any exchange of value between complementor and user. We refer to platform emergence as phase before the start-up of a digital platform. Thereby, theory on digital infrastructure provides promising concepts to understand the nature of platform emergence.

### 2.4.1 The Digital Platform Life Cycle

Considering the life cycle of a digital platform, Parker, Van Alstyne and Choudary (2016) illustrate three major stages: (1) in the startup phase, the platform owner needs to balance low resources while enabling the core interactions to foster initial network effects. Thereby, different strategies might be applied to resolve the chicken-egg problem (Schirrmacher et al., 2017; Stummer et al., 2018). While several approaches focus on the nurturing of a completely new ecosystem for example by subsidizing one side to attract stakeholders on the other side (Anderson et al., 2014), some platform ecosystems were kicked-started by enveloping other platforms (Eisenmann et al., 2011). Thereby, the new platform owner strives to combine functionality with an established platform to leverage shared user relationships. What all approaches have in common is the fostering of the platform's core interaction (Parker, Van Alstyne, & Choudary, 2016). The metrics to evaluate the strength of core interactions in the startup phase are liquidity (in the ecosystem), matching (of both sides) and trust (of stakeholders in the platform) (Parker, Van Alstyne, & Choudary, 2016).

As soon as a stable level of interactions is established, the platform owner needs adopt metrics to measure the quality and number of interactions in its ecosystem. In the growth phase (2), which describes the – commonly rapid – increase of users and complementors, a balanced producer-to-consumer ratio needs to be ensured (Parker, Van Alstyne, & Choudary, 2016). The platform owner might influence the growth by the degree of openness of its platform and thereby enhance the exploitation of its market potential (Ondrus et al., 2015). In this endeavor they should understand the motivation of its complementors (Boudreau & Jeppesen, 2014) to foster generativity (Um et al., 2013) and facilitate value-creation (Parker et al., 2017). Consequently, Parker, Van Alstyne and Choudary (2016) advise to focus on metrics that influence value creation and value flows in its ecosystem and thereby enable growth.

Eventually, a successful digital platform reaches the maturity phase (3) where the platform owner strives for the preservation and further extensions of the established ecosystem. Therefore, the platform should adopt to changing needs of its user ("drive innovation") and resist potential threads from competitive ecosystems. By "amplify meaningful signals and filter out noise" (Tiwana, 2014, p. 156) the platform owner enables the detection of relevant dynamics and developments in their ecosystems, which can be responded by the flexible allocation of resources to maintain their leadership position (Teece, 2017) as the adoption of control mechanisms (Eaton et al., 2015; Henfridsson & Ghazawneh, 2013; Karhu et al., 2018) or the exploration of new platform functionality (Bender & Gronau, 2017; Saarikko, 2016) perhaps even by enveloping other platform ecosystems (Eisenmann et al., 2011). Altogether, Parker, Van Alstyne and Choudary (2016) describe the degree of innovation, a high signal-to-noise ratio and the facilitation of resource allocation as key metrics during the maturity phase of a platform ecosystem.

The division of the platform lifecycle in startup phase, growth phase and maturity phase is commonly acknowledged in the scientific discourse. While there are small deviations and additions, as Teece (2017) that separates the maturity phase in a "leadership" phase with a usually strong market position of the platform and a "self-renewal" phase that reflects a phase with focus on exploration and adoption, there is no contradiction that a platform owner first needs to solve the chicken-egg problem, then has to facilitate growth, and finally can exploit the platform ecosystem while constantly adopting to new trends. While there is comprehensive research on digital platforms that successfully went through the platform lifecycle, there are little insights on the actual emergence of a digital platform (de Reuver et al., 2017). As already mentioned, IS literature provides comprehensive theoretical considerations on the launch of a digital platform (Stummer et al., 2018). However, before the launch, a new digital platform requires to be designed and build. The platform owner demands knowledge on platform architecture and platform governance to create the basics for a successful digital platform ecosystem. While IS research emphasizes the need for flexibility and constant adoptions, a bad platform design might prevent the nurturing of initial network effects from the beginning without any chance to adjust platform design. While there is diverse and in-depth research on dynamics and mechanisms of digital platforms that successfully went through the platform lifecycle, the emergence of new platform ecosystems remains largely in the dark. This thesis focuses on this earliest phase of digital platforms and strives for its illumination.

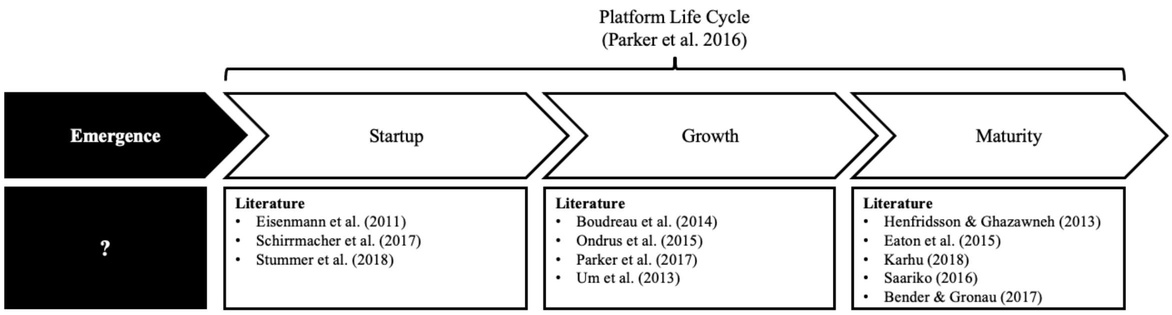



**Figure 3: Platform emergence as underinvestigated phase in the platform life cycle**

### 2.4.2 Digital Infrastructure as a Base for Platform Emergence

While later phases in the platform life cycle have been extensively studied, the emergence of digital platforms has remained in the dark so far. However, there is a common understanding that an underlying digital infrastructure embodies a precondition for the emergence of digital platforms (Constantinides et al., 2018). Digital infrastructures can be understood as "a shared, open (and unbound), heterogeneous and evolving socio-technical system consisting of a set of IT capabilities and their user, operations and design communities" (Hanseth and Lyytinen (2010), p.4). The Internet, smartphones, or the Universal Serial Bus (USB) interface are examples for digital infrastructures. They illustrate that digital infrastructures might be connected, mostly stacked in layers that interact with each other by standardized interfaces (Yoo et al., 2010). The internet is accessible via smartphones, while the USB standards enabled exchange of data between different computing devices that again can upload that data to the internet. Consequently, penetration of computing and networking capabilities in new domains (as cars) nurtures the emergence of new digital infrastructures (as connected cars) as well as the spread of already existing digital infrastructures (as the Internet).

In contrast to analogue infrastructures such as the sewage system or the public power grid, which serve a dedicated purpose, namely the supply of water or electricity to households, digital infrastructures reveal generativity (Henfridsson & Bygstad, 2013). While pipes and cables transport water and electricity exclusively, data connections transmit all kinds of current and future digital resources. As the internet was initially considered for the exchange of scientific knowledge, its generative characteristic nurtured the emergence of technologies as electronic mails, digital music, social networks, electronic commerce and many more. The generative character of digital infrastructures lay the ground for unbound growth and unforeseen innovation by enabling use cases that are not foreseen during their creation (Chesbrough, 2003).

Constantinides et al. (2018) refer to digital infrastructure as "computing and network resources that allow multiple stakeholders to orchestrate their services and content needs" (p. 381). This orchestration however, is not enforced by the infrastructure itself but requires additional control instances (Hanseth & Lyytinen, 2010). Digital platforms embody such kind of artefacts. They are "created and cultivated on top of digital infrastructures" (Constantinides et al., 2018) to facilitate value creation and subsequent value capture. Establishing and retaining control on such "bottlenecks" of digital infrastructure is a critical challenge for firms strive for the exploitation of a platform business model (Hannah & Eisenhardt, 2018).

The penetration of digital technology in formerly analogue environments prepares the foundation for digital infrastructure and the subsequent emergence of digital platforms. This thesis strives to shed light on this evolutionary process by the example of the automotive industry and the "digitalization of the car".

### 2.5 Digital Platforms and Incumbent Firms

Since several decades the automotive industry is dominated by a small number of OEMs (Jacobides et al., 2016). As other incumbent firms, their business model was established prior the begin of the digital transformation in the early 2000s. This is in contrast to digital-native companies that build on business models that exploit the benefits of digitality (Schreieck & Wiesche, 2017a). The spread of digital infrastructure into almost all areas of business allowed digital-native companies to penetrate several domains that were previously dominated by incumbent firms. Svahn et al. (2017) examine eight industry branches, where struggle of incumbent firms with digital innovation have been observed and investigated: the automotive, manufacturing, mining, pharmaceutical, photography, construction, software, and transport sector. In these sectors, many established companies are prone to disruption. New, risk-taking actors challenge incumbent firms by satisfying similar customer needs with the use of new digital technologies that enable entirely different business models (Hein, Schreieck, et al., 2019a). Their superior digital capabilities endanger the market position of several established firms. As a consequence, incumbent firms are forced to react by transforming their organizations (Bharadwaj et al., 2013).

#### 2.5.1 Challenges of the Digital Transformation

The transformation of non-digital to digital businesses reveals several challenges for incumbent firms (Warner & Wäger, 2019). In general, discontinuity of technologies makes existing competences irrelevant and calls for new capabilities which are not yet present in the incumbent firm (R. M. Henderson & K. B. Clark, 1990; Tushman & Anderson, 1986). During the process of transformation, the exploitation of existing capabilities needs be balanced with the building of such new capabilities that, however, still need to be compatible with path dependencies of the past (Svahn et al., 2017). While several established companies successfully transform their businesses to the digital context, a large share of incumbent firms struggle with the digital transformation. The lack of capabilities, actually provokes the *digitization of the cow paths*, meaning that old structures and processes were retained without exploiting the benefits of digitality (Tilson et al., 2010).

#### 2.5.2 The Management of Uncertainty

Considering the tremendous challenges of digital transformation, several incumbent firms hesitate to follow the technological pull (Hein, Schreieck, et al., 2019b). Even though, the digital transformation of their industry is inevitable, irreversible, and fast, the faced uncertainty (Oswald & Krcmar, 2018) impedes appropriate activities. The observation of hesitating incumbents is in line with the established knowledge on uncertainty management in traditional, non-digital businesses. Incumbent business models build on the resource-based view of the firm

(Barney, 1991), which conceptualizes competitive advantage respectively to its position in the industry (Porter, 1980). Firms act proactive and managers try to redeploy excess resources to achieve competitive advantages (Lavie, 2006). Beckman et al. (2004) fundamental work on the management of uncertainty reveals that firms that face market uncertainty tend to exploit existing partnerships. However, the strengthening of existing non-digital business partnerships does not resolve uncertainty regarding digitalization, which again causes further uncertainty. Therefore, recent literature indicates that firms need to adopt their understanding from a resource-based view rather to a relation-based (Dyer & Singh, 1998) or network-based view of a firm (Koch & WIndsperger, 2017) to cope with the challenges of the digital transformation.

### 2.5.3 Platform Emergence from an Incumbent Perspective

As illustrated in *section 1.1*, digital platforms embody a central concept in digital business. Therefore, most of incumbent firms that are confronted with the digital transformation of their business need to embrace the concept of digital platforms (Parker, Van Alstyne and Choudary (2016), p.60 ff.). However, digital platforms reveal fundamental differences to traditional business that need to be internalized. However, there are fundamental differences between traditional businesses and digital platforms that must first be understood before a successful platform business can be established (Cusumano et al. (2019), p.139 ff.). Digital platforms enable and facilitate interactions of different sides. Instead of producing and selling resources (Barney, 1991), the focal firm enables transaction between consumers and complementors (Hagiu & Wright, 2015). The actual value creation takes place outside the direct control of the focal firm (Parker et al., 2017). This implicates new ways of quality management (Cennamo et al., 2018), value-proposition and value-capture (Han & Cho, 2015) or competitive action (Tan et al., 2016). This thesis investigates how an incumbent firm can acquire capabilities to exploit digital platforms and to address these changed conditions for their business.

## 2.6 Platform Options

The emergence of digital platforms requires firms to adapt their strategical endeavors. The following section illustrates why the platform business is different and in which way real option thinking can support firms during the emergence of digital platforms.

### 2.6.1 Strategical Challenges of Digital Platform Business

The environment of emerging digital platform ecosystems is highly volatile and fast-moving. A new platform owner needs to balance interests of multiple stakeholders while protecting its new business from external competitors. Unfortunately, the resource-based view on the firm (Porter, 1980) and its established strategies for defending businesses is not sufficient any longer (see also chapter 2.5). Platform companies need to consider competition inside their ecosystem (e.g. that strives to diminish a dominant position of a platform owner) as well as other platform ecosystems (e.g. that endeavor an envelopment of the ecosystem (Eisenmann et al., 2011). Parker, Van Alstyne and Choudary (2016) refer to the complexity of managing competition in platform ecosystems as “play three-dimensional chess” (p. 210). While the established rules of resource-based markets remain true, platform ecosystems extend these dynamics by two aspects. First, a platform owner can influence network effects which enables shaping markets, not just responding to them. Traditional business strategy implicitly assumes that competition embodies a zero-sum game where a firm needs to gain market share from a competitor to grow the business. However, rather than re-allocation market shares, platform businesses grow the overall market (as Spotify that simplified the publication of music by self-publishing) or create alternative markets (as Airbnb that replaced traditional hotel rooms with renting private facilities). Second, digital platforms move value creation outside the focal firm (Parker et al., 2017). Consequently, business is not driven any longer by seizing each opportunity on its own but by consciously deciding how the co-creation of value with partners in the ecosystem can be optimized.

### 2.6.2 Real Option Thinking for Digital Platforms

The enormous complexity and dynamics of digital platforms implicate that companies must find alternative strategies for dealing with their emergence. The concept of real options thinking is considered as reasonable approach for doing so (Tiwana, 2014). A real option embodies the right or opportunity to do something without the obligation to do it (Trigeorgis, 1993). The embedding of multiple real options allows to react to environmental changes during the trajectory of a project. While also traditional strategies are frequently considered as answer to uncertainty, they rely on an "assumption of predictability in the environment" (Tiwana, 2014). Real option thinking, however, assumes the unpredictability of environmental developments and provides flexibility to deal with unforeseen scenarios (Trigeorgis, 1993). Therefore, the value of real option thinking rises with the level of technical as well as market-related volatility in a project (Tiwana, 2014). Both aspects can be considered as relevant for emerging digital platforms. While the design of a platform architecture reveals a tremendous number of technical design choices, the evolutionary trajectory of e.g., the chosen programming language or the inherent development tools is hard to predict. While matured platform firms established own technologies in the past to reduce such uncertainties (e.g., Apple introduced Swift as programming language for most of their products, Google relies on Kotlin in the Android ecosystem), a new platform owner needs to rely on external technologies. Considering the emergence of new markets, the reception of customers might differ from the expectations. A new platform owner needs to identify to the core transaction that resolves the chicken-egg problem and enables the growth of cross-side network effects (Parker, Van Alstyne, & Choudary, 2016). However, this core-interaction might differ from an initial idea of the new platform owner and requires adaption or needs to be completely replaced. In parallel, control and resourcing mechanism might require adaption to compete with complementors that challenge the dominant position of the platform owner (Eaton et al., 2015) and the platform ecosystem needs to be defended from external rivals (Karhu et al., 2018). This thesis strives to apply the concept of real option thinking to illuminate the management of uncertainties by an incumbent firm that is confronted with the transformation of its products into a digital infrastructure that nurtures the emergence of digital platforms.

# 3 Research Design

This thesis strives to illuminate the design and evolvement of an emerging digital platform in the context of an incumbent firm. Due to the exploratory character of our endeavor, we decided for a qualitative research approach (3.1), which was implemented by a single-case study (3.2) of an incumbent company that is confronted with the “platformization” of its product. Therefore, we selected the BMW Group as case company, which is presented in section 3.3. During our research several (qualitative) research methods were applied, which are introduced in section 3.4.

## 3.1 Qualitative Research Approach

This thesis applies different exploratory analysis to pervade the field of digital platforms for automotive onboard apps. Starting point of an exploration is the examination of different processes and correlations in the field with regards to certain recurring patterns (Deshpande, 1983). There are three different research paradigms that are commonly used in research: The qualitative paradigm, the quantitative paradigm, and the mix of both approaches respectively "across methods triangulation" (Tashakkori & Teddlie, 2010). For this work, the qualitative paradigm is the most appropriate approach since it shows the capability of a process-oriented, phenomenological, and inductive view (Deshpande 1983). Moreover, the qualitative paradigm is more exploratory, grounded, and discovery-oriented than the quantitative paradigm. The approach starts with an extrapolation from "grounded events" and not with predefined hypotheses (Glaser & Strauss, 1967). Since no deeper research on the emergence of digital platforms in an incumbent firm has been developed so far, this thesis should create a fundamental understanding of the topic. Caused by its explorative character which enables the exploitation of underlying respectively hidden factors and motives which are not visible on the first view, the qualitative methodology has been chosen (Holzmüller & Buber, 2009). Furthermore, this work aims to explore an empirically and theoretically less investigated topic. Therefore, it is crucial to select an approach with the ability of an open and variable reaction towards insights that previously have not been assumed (Creswell, 2009). The usage of qualitative methods is less common in scientific literature (Holzmüller & Buber, 2009). Nevertheless, there are diverse reasons why this method is the right choice for this study. Concretely three attributes of qualitative methods justify the suitability for this study - access adequacy, complexity adequacy, and exploration adequacy. With regards to access adequacy, qualitative methods fit well if intuitive associations and preconscious factors are present (Holzmüller & Buber, 2009). This is important since the challenges and impediments to create a digital platform for automotive onboard apps that are seen by car manufacturers for example are not visible at the first sight and strongly depend on secret strategical considerations or complex market stimuli which cannot be clearly defined from the beginning. Second, concerning complexity adequacy, a qualitative approach simplifies the acquisition of insights into complex relations (Holzmüller & Buber, 2009). Through its open and interpretative structure, it is possible to assess multiple point of views within a digital platform ecosystem. For instance, the success factors of a platform with regards to developer engagement as well as end-user attractiveness embody highly complex interrelationships. Knowledge about these relations appears crucial for the conception of a digital platform. Third, relating to exploration adequacy, qualitative methodology is very appropriate for scientific fields that have not been considered yet. The issue of a digital platforms in the context of the automotive domain can be characterized as such a field. A qualitative approach enables a structured and profound way to explore this topic. Moreover, the quantitative methodology has not been chosen since its methods are more suitable to confirm or verify pre-existing theories and hypotheses whereas qualitative research methods serve for discovering or generating theories or concepts.

### 3.2 Case Study Research

With the objective of setting existing theory into the novel and rapidly evolving connected car context and to construct a research model within this new context, an exploratory case study approach with the case company BMW has been chosen as part of the overall multi-method research approach. Besides the fit to the dynamic research field, a case study with BMW was further beneficial as the author was able to utilize company resources and information as part of a collaborative practice-related study program he participated in at the time of this study. Yin (2009) defines a case study as “an empirical inquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident” (p. 18). Considering the emergence of digital platforms as a contemporary phenomenon that requires investigations in practical environments, it appears as an appropriate approach for this thesis. Constituting a qualitative research method, a case study inquiry is focused on gaining an in-depth understanding of a context-specific “case” or research problem (Yin, 2009). For this purpose, a single case or a small number of cases is generally examined in a real-world setting (Gable, 1994; Yin, 2009). A case study therefore provides the possibility to develop a deepened understanding within a specific research context, but often lacks generalizability (Gable, 1994). Considering all IS research, the case study approach enjoys great popularity (Harrison & Wells, 2000). They provide the possibility to craft a detailed understanding of an investigated phenomenon and enables the creation and testing of theory (Darke et al., 1998). This broad versatility presents a well-suited mode of scientific inquiry to IS research. Yin (2009) divides the case study approach into six parts, which are explained below.

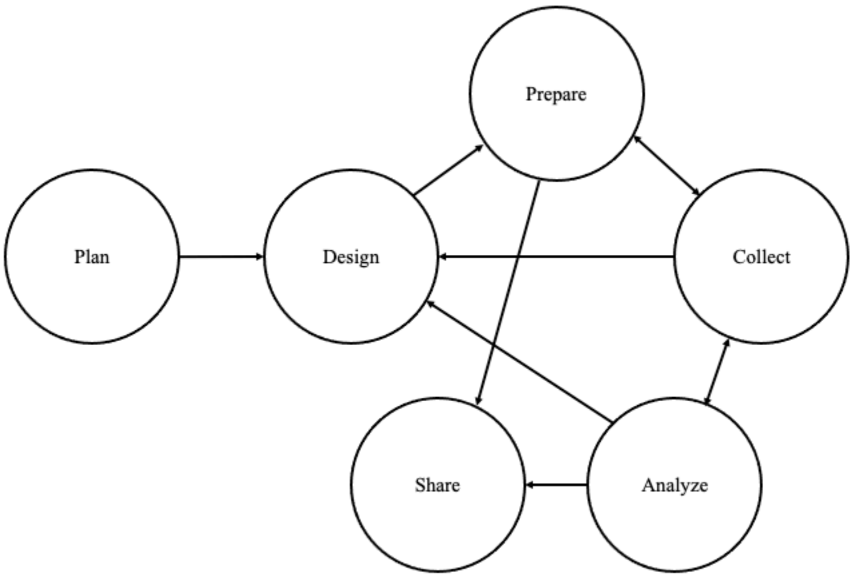



**Figure 4: Case study approach (adapted by Yin 2009)**

**Plan:** In the planning phase the researcher needs to identify research questions or rationales that are used to investigate a selected object. Case studies are especially proper for research on the context (“how?”) or the reason (“why?”) of a considered phenomenon. The extensiveness of a real-live context and the consequent richness of a phenomena requires a well-defined tactic to cope with situation in which more variables than data points exist. Thereby, the triangulation of data sources and their indicators is a frequently applied approach.

**Design:** Yin (2009) defines the research design as "the logic that links the data to be collected (and the conclusions to be drawn) to the initial questions of study (p. 24). It will enable the researcher to answer the defined research questions and requires the definition of (1) the unit of analysis, (2) the boundaries of the research, (3) theoretical foundations and (4) criteria that indicate which aspects of the investigated phenomena should be analyzed. A well-defined study design defines a clear scope that articulates the limits of the planed research endeavor. Thereby, the unit of analysis represents the considered phenomenon and might represent a single person, a company or even a whole industry. A clearly defined unit of analysis sets limit to data collection so that not all available material needs to be considered. While several studies focus on one single unit of analysis others decide to embed multiple units of analysis. Both approaches might be implemented either with a single case design or a multi-case design (see Figure 5). While a multi-case study approach reveals a repetitive character and facilitates the recognition of patterns, a single-case study is especially suited for deep and long-term investigation of one representative case. This thesis embodies a single case study with several units of analysis (that are represented by the studies presented in Part B of this work), which allowed us to collect comprehensive and deep insights on the emergence digital platforms in the context of one incumbent firm (see 3.3).

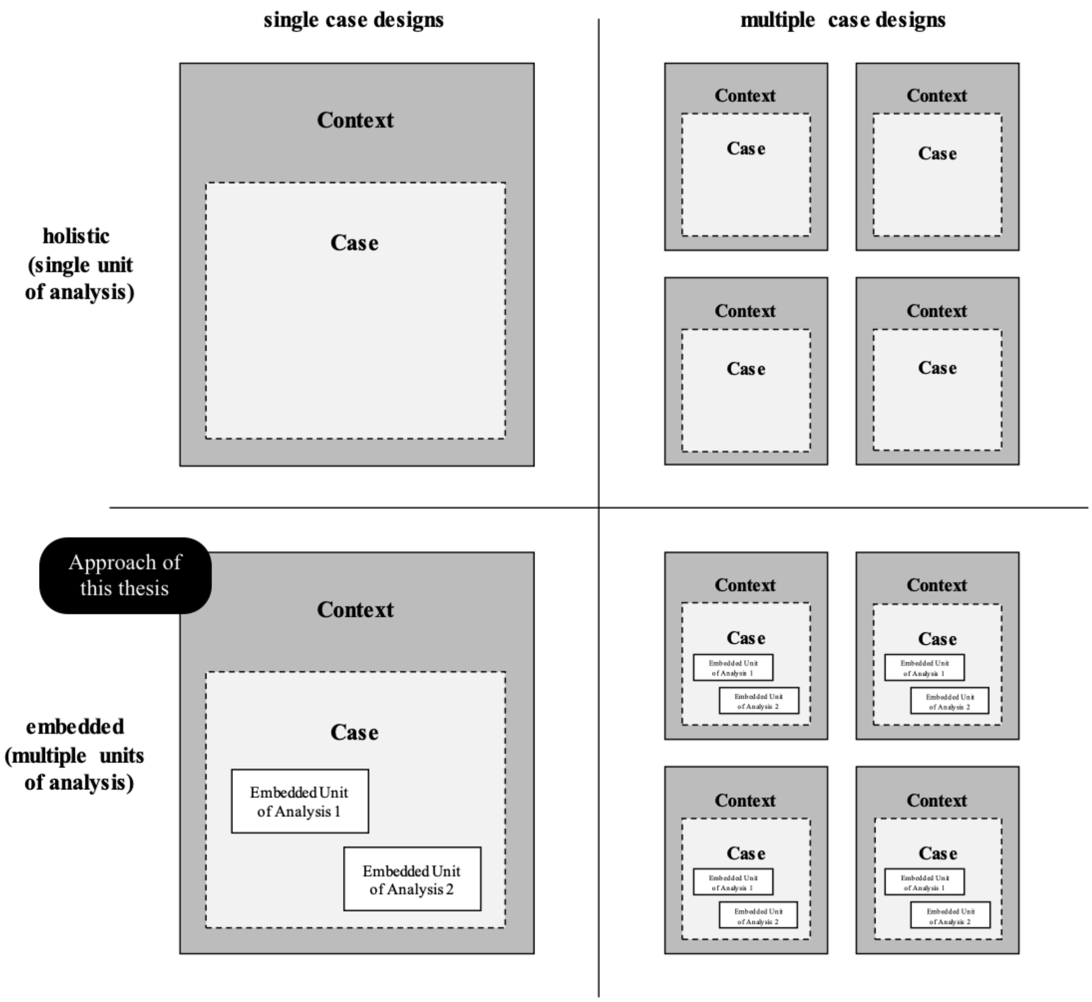



**Figure 5: Case Study Design (adapted by Yin 2009)**

**Prepare:** After the decision on a research design that is suitable to the research questions, the data collection needs to be prepared. That might include the creation of an interview guide (3.4.1 Expert Interviews) or any formalized protocol for data collection. Furthermore, Yin (2009) recommends conducting a "pilot case" to recognize risks and support in "develop relevant lines of questions" (p. 92).

**Collect:** The evidence of a case study might be collected from different sources: interviews, documents, archival records, observations and physical artifacts (Yin, 2009). Data collection does not only involve the collection of data but also the creation of a case data base to persist all collected information as well as the establishment of a chain of evidence. While interviews are the mostly applied approach for data collection in case studies, they are frequently complemented by other sources (Eisenhardt & Graebner, 2007). Independent from data sources, Yin (2009) recommends the triangulation of data to increase validity.

**Analyze:** Data analysis can be considered as a procedure of organizing and understanding the previously collected data. Thereby, Yin (2009) recommends to begin with theoretical assumptions that were extracted from previous research, if available. Even though the collecting and the analyzing of data are described as two distinct steps in a case study design, Yin (2009) emphasizes the iterative character of these two steps (see also Figure 4). The analysis of data might reveal the need for further data collection, which afterwards need to be analyzed again.

This circle is interrupted by theoretical saturation, which indicates that another iteration of data collection won't reveal any new theoretical insights (Glaser, 1992).

**Share:** Eventually, the generated knowledge should be distributed among the respective readerships (Yin, 2009). Therefore, the material needs to be sufficiently composed and presented. Depending on the choice of the outlet and the type of publication the respect audience needs to be identified.

### 3.3 Case Description

To address the central research goal – *development of an empirical understanding of mechanisms and processes of platform design in an emerging digital platform from the perspective of an incumbent firm* – the BMW Group, a globally operating automobile manufacturer, has been selected as a suitable case company. To this date, the German car manufacturer employs over 120,000 people worldwide[7] and is among the 15 largest automotive OEMs, with over 2.5 million passenger vehicles produced and yearly revenues of more than EUR 99 billion in 2020[8].

In addition to its core BMW brand, the BMW Group has further sub-brands such as the BMWi, the MINI or the Rolls-Royce automotive sub-brands[9]. Despite belonging to the most successful companies in the automotive sector, digitalization and rapidly evolving customer expectations concerning digital product- and service-offerings pose a significant challenge to the legacy car manufacturer. While BMW is still rooted deeply in its traditional business model which is focused on the car as a physical entity, newly added digital capabilities are increasingly transforming the automobile into a connected device. Driven by the evolving customer expectations and the newly entering competition from the ICT-industry, BMW is undertaking considerable efforts to successfully manage the transformation from a pure car manufacturing company to a digital service provider.

These efforts include the establishment of a digital platform for automotive onboard applications (i.e., apps) in the vehicle's infotainment system. The operation of such an automotive onboard platform allows the BMW to utilize over-the-air (OTA) software updates to continuously provide new service content in form of apps – even if the hardware has long been manufactured. The possibility to introduce new car-related features in form of onboard apps via OTA software updates, holds, like application stores on smartphones, a great potential for automotive manufacturers. By enhancing existing functionality, car-related digital platforms can not only increase the perceived consumer value for the car. They can also open new forms of revenue streams, by introducing digital service innovations. Like other comparable traditional car manufacturers, BMW has identified the digitalization of their products as a major strategical target. With the BMW Connected Drive-Store, BMW customers can purchase a selection of digital service-packages for their car. Features of such digital service packages include services developed by BMW (e.g., remote services that allow to remotely control the vehicle), or the integration of projected modes. Projected modes allow to mirror external smartphone features to the vehicle's center display. Moreover, with the introduction of BMW's

---

[7] https://www.bmwgroup.com/de/unternehmen.html retrieved 15.02.2022

[8] https://www.bmwgroup.com/content/dam/grpw/websites/bmwgroup_com/ir/downloads/en/2021/bericht/BMW-Group-Bericht-2020-EN.pdf retrieved 15.02.2022

[9] https://www.bmwgroup.com/en/brands-and-services.html retrieved 15.02.2022

operating system OS7 in 2018, the newer generations of BMW vehicles have OTA update capabilities, allowing car owners to continuously update their operating system and to install new digital service content. With an update of BMW's OS7, several digital services, such as connected parking, connected charging or BMW's personal assistant, were wirelessly added to over 750,000 compatible vehicles worldwide[10].


**Figure 6: BMW X5 series with BMW OS 7 with OTA Update Capabilities (Source: BMW Group)**

In 2021 BMW reinforced its endeavors on the digitalization of their cars with the release of BWM OS8 in the new BMW iX. The system enhances known digital functionality as the navigation system or the onboard voice assistant and introduces new capabilities as specific modes that adapt the complete interior of the car according to configurable contexts.[11] Like the technological capabilities of current smartphone variants, BMW is offering cars with increasing connectivity-, data generation- and data processing capabilities that utilize the car's internal sensors, 5G-connectivity, and artificial intelligence to process and transfer vast amounts of data in real-time.

[10] https://www.press.bmwgroup.com/deutschland/article/detail/T0318812DE/bmw-group- retrieved 15.02.2022

[11] https://www.bmw.com/en/events/ces2022/mymodes.html retrieved 15.02.2022

**Figure 7: Cockpit of the BMW iX with BMW OS8 (Source: BMW Group)**

This thesis aims to study the emergence of digital platforms from the perspective of an incumbent firm. BMW as a traditional car manufacturer that is confronted with the digitalization of its products serves as perfect case for this endeavor. Thanks to a close collaboration with the company and an engagement in the company's doctoral program, it was possible to examine activities around the design of the digital platform up close and collect manifold insights.

### 3.4 Applied Research Methods

This thesis investigates the phenomena of platform emergence in the context of one single case – the BMW Group. In the course of this work, several distinctive studies with varying research methods have been conducted. The following sections provide an overview on the applied methods.

#### 3.4.1 Expert Interviews

Most of the collected data in this study was gained by expert interviews inside the BMW Group. Basically, there are three types of interviews for qualitative research (Bogner et al., 2014). Within this study the theory generating interview type was chosen. In this context, theory generating can be described as the process of developing theories and interrelations through an interpretive analysis of empirical data which is collected by expert interviews data (Gläser & Laudel, 2009). The interview should extract factual knowledge. That demands a certain measure of structuring, which is provided by an interview guide. Besides the narrative interview and the ethnographical interview, the expert interview is generally used for case studies (Myers & Newman, 2007). Due to this work was defined as a grounded-theory-inspired case study, the approach appeared convenient. While initially considering the perspectives of three identified stakeholder groups - platform owner, developers, and end-users - the study aimed on the identification of topic's specifics. Thereby the goal was not touching as many facets as possible but the deep examination of specific aspects which may be distributed across the different stakeholder groups. Instead of considering all possible reasons for this deficiency, a concentrated investigation on the automotive industry related coherences was carried out, since it appeared as the most relevant aspect in the scientific context. The assumption that experts provide exclusive knowledge regarding the topic is a main reason for the conduct of expert interviews and embodies a major part of theory development (Myers & Klein, 2011). The experts can provide binding information based on their individual experience as well as their specialist knowledge (Myers & Newman, 2007). There are three different kinds of knowledge that can be extracted by these interviews (Bogner et al 2014). *Process knowledge* includes insights into interactions and processes within organizations that involve the interviewed expert. Grounding on these experiences the expert may provide special knowledge that might not be scientific knowledge in the beginning. By considering several insights of multiple experts this knowledge will be evaluated. Further, *interpretative knowledge* contains personal perceptions, subjective interpretations, and explanation approaches of the expert. *Technical knowledge* is exclusively accessible to the experts and contains facts which can be codified just like the already mentioned knowledge types (Bogner et al., 2014). Expert interviews differ mainly from other qualitative interviews in terms of targeting the extraction of factual knowledge. An interview guide supports this goal by supporting the researcher in structuring the conducted interviews. During the interview it embodies a guiding theme and provides a selection of possible questions (Kaiser 2014). Eventually, it leverages the comparability of the collected data (Gläser & Laudel, 2009).

### 3.4.2 Grounded Theory

Grounded theory is a research method which was coined by (Glaser & Strauss, 1967) and facilitates the development of theories from qualitative data. This data-oriented approach yields closeness and thereby provides one major advantage compared to other qualitative research methods, namely relevance (Urquhart & Fernández, 2013). The Grounded Theory Method (GTM) enables the development of such theories that are grounded in data itself through *"a logically consistent set of data collection and data analysis procedures"* (Charmaz, 2001, p. 245). Patterns within the qualitative data are identified and formalized in a theory. This bottom-up process ensures that the theory really is bound to the empirical qualitative data (Eisenhardt, 1989; Urquhart, 2013). Grounded theory is especially suitable for novel fields as the theory is developed from the data and no predefined hypothesis or theories are needed (Urquhart & Fernández, 2013; Wiesche et al., 2017a). However, if the researcher starts with an existing theory the goal of grounded theory is never to verify or falsify the theory (Urquhart et al., 2010). In such a case the GTM can be used to extend and widen the scope or improve the existing theory. Essential characteristics of grounded theory are according to Charmaz (2001) (1) the continuous interplay of data collection and analysis (2) the creation of codes from the data and not from predefined hypothesis, (3) the establishment of theories, which explain behavior and processes, (4) the conceptualization of a theory through writing memos, (5) theoretic sampling which is the decision where to sample from next and (6) the postponement of the literature analysis, because the researcher’s prior knowledge should not bias the theory building process. Contrary to (Charmaz, 2001) other researchers such as (Urquhart, 2013; Urquhart & Fernández, 2013; Wiesche et al., 2017a) are convinced that it is nearly impossible to be completely new to a topic. Existing knowledge should simply be hidden by the researcher while using the GTM (Urquhart & Fernández, 2013). However, there is still a more profound conflict when it comes to grounded theory. The founding fathers Glaser and Strauss created over time two different viewpoints on grounded theory (Urquhart & Fernández, 2013; Urquhart et al., 2010). Strauss focused on concretizing the method and introduced a coding paradigm for an increased process guidance, while Glaser’s viewpoint rather concerns the methodology of grounded theory and focuses on flexibility (Urquhart et al., 2010; Wiesche et al., 2017a). Glaser criticized the use of a conditional matrix and the coding paradigm Strauss and Corbin (1990) introduced as ready to go tools for the conceptualization (Urquhart et al., 2010). Strauss and Corbin (1990)'s approach is well known but it is seen by literature as too restrictive on the method and thereby limiting the efficiency of grounded theory (Urquhart, 2013; Wiesche et al., 2017a). The researcher's choice for the Glaserian or Straussian GTM approach depends on the individual preference (Schreieck & Wiesche, 2017a; Urquhart, 2013). There is no right or wrong choice. The absence of a one-size-fit-all GTM makes the selection of an ad- equate GTM procedure combination more problematic for the researchers (Schreieck & Wiesche, 2017a). The main difference between the Glaserian and Straussian GTM is the coding (Wiesche et al., 2017a). While Strauss uses open, axial, selective and coding for the process, Glaser uses open, selective and theoretical coding (Urquhart et al., 2010).

For this thesis the Glaserian approach is applied, because the increased flexibility offers numerous different possibilities to link fundamental different views on the topic with categories. Different views, such as the technical, economical or a view that emerges during the process, influence the researcher's understanding of categories (Urquhart et al., 2010). These different views are called slices of data and ensure a holistic data collection and analysis. Following the Glaserian approach the procedures theoretical sampling, constant comparison, open coding, selective coding, theoretical coding and memoing are utilized (Glaser, 1978). Theoretical sampling means selecting the new data sources based on the insights of previous gathered and already analyzed data (Glaser & Strauss, 1967). The process aims at improving the quality and relevance of the study. The GTM therefore iteratively performs the steps of data collection and data analysis. Glaser and Strauss (1967) mention that the theory development might be hindered when not doing data collection, coding, and analysis together and that grounded theory addresses this problem. Coding performs the data analysis. The first step of the coding process is open coding which is defined by (Glaser, 1978, p. 56) as *"coding the data every way possible"*. Followed by selective coding which allows to group the open codes in higher-level categories that are related to the core category (Glaser, 1978). Naturally, due to its bottom-up nature of coding the categories are often too detailed (Urquhart & Fernández, 2013). The scaling up of categories to higher-level comprehensive categories needs to be done deliberately (Urquhart & Fernández, 2013). The highest-level theories are formal theories which deal with theories on social capital, actor network theory and structural theory (Urquhart, 2013). The third and final step of coding is theoretical coding where relationships between the categories, found during selective coding, are uncovered (Glaser, 1978). Theoretical coding is vitally important for grounded theory as the theory building happens in this step (Urquhart, 2013).

Memos that capture spontaneous ideas are extremely helpful tools when theorizing and reorganizing identified categories (Glaser, 1978; Urquhart, 2013; Wiesche et al., 2017a). Constant comparison ensures that the data within an identified category really belongs together by continuously comparing data labelled with the same code (Glaser & Strauss, 1967). This process advances the researchers understanding and increases the theoretically of the analysis (Urquhart, 2013). This thesis uses *"the GTM in an interpretivist paradigm"* (Urquhart, 2013, p. 61) as the social context and environment need to be understood to interpret the practices and meanings. According to (Glaser, 1978) it is furthermore crucial that the researcher understands the context in which the theory is developed because it influences the theoretical sensitivity (Urquhart et al., 2010). Glaser (1978) defines theoretical sensitivity as the researcher's sensitivity to theories.

### 3.4.3 Action Research

Action research originated in the social and medical science in the middle of the twentieth century where massive social changes caused by the second World War emerged. It was developed by Kurt Lewin, a social psychologist at the University of Michigan, with the goal to apply some theoretical ideas to a practical endeavor (Adelman, 1993). Its “strongly post-positivist assumptions such as idiographic and interpretive research ideals” (Baskerville, 1999, p. 2) made it a widely used methodology in the social sciences in the second half of the 20th century. Since the 1990s, action research has also been gaining increasing popularity in the area of information systems science (Baskerville, 1999).The research approach has a strong focus on the involvement and collaboration between the researcher and the practitioners; it is a qualitative and constructive research method with its foundation in user involvement and the need for action (Avison, 1998). In the action research domain, the researcher is concerned about the current state of an organizational environment. He tries to introduce change while simultaneously researching the process (Babüroglu & Ravn, 1992). This also results in a problem-solving mentality that carefully informs the theory. This setup makes it a strong tool for research in information systems (IS) (Avison, 1998).

In their ground-laying MISQ "Special Issue on Action Research in Information Systems" (2004), Myers and Baskerville define **four essential and pragmatist premises** of action research. Before starting an action research project, it is essential to state and **define a clear purpose (1)**. Here, Myers and Baskerville refer to Charles S. Pierce, an American chemist and philosopher, who is frequently considered as “the father of pragmatism”. Peirce (1905) stated that human concepts are defined by their consequences. Therefore, it is essential for setting a concept in focus to define human purpose and assume human volition. People must have an independent will to decide what actions they plan to do. This means for the action researcher that they must define the underlying theoretical purpose for the intended action. It is also necessary to concretely define the theory before the action is taken. If this order is not followed, the risk arises the action is purposeless and therefore meaningless. Myers and Baskerville’s second premise states that there needs to be **practical action (2)**. Thereby they refer to William James, another founder of the psychological school of pragmatism, who sees truth embodied in practical outcome. In the action research context this means that action must follow the determined purpose. Reflecting this idea, Reason and Torbert (2001) define action research as "a set of practices that responds to people’s desire to act creatively in the face of practical and often pressing issues in their lives in organizations and communities" (Reason & Torbert, 2001, p. 3). The actual practical involvement of the researcher is such a central concept that the “action” became eponymous for the approach. Third, the action needs to **inform the theory (3)**. This premise establishes an essential foundation for rigor research. Thereby, the applied theories need to be adjusted according to the action's practical outcome. The goal is that action research "expands scientific knowledge" (Baskerville, 1999, p.7). It is "research in action, rather than research about action" (Coghlan & Brannick, 2001, p.3). If action would not inform theory, the rigor of the research project could not be guaranteed. In general, the aspect of rigor is a much-debated topic in the scientific discourse on action research. It is argued that the involvement of the scholar will entail the loss of distance between researcher and research subject, which puts the scientific objectivity at risk (Avison, 1998). Even if exactly that low distance provides chances of pursuing both practical and theoretical goals in this area (Susman, 1983), a clearly defined procedure in an action research approach is essential (see next section). Fourth, it is important to **involve respective stakeholders or actors (4)**. One aspect that distinguishes action research from several other research methods is the active involvement of the researcher in the social situation with others. Coghlan and Brannick (2001) call it a "collaborative democratic partnership". Only through a high degree of involvement is it possible for the researcher to introduce and gasp the spirit of emerging change (McNiff et al., 1996). Therefore, all “reasoning must be socially situated" (Baskerville & Myers, 2004).

There are multiple approaches to description of the action research process since it has introduced by Lewin (1946). The aspect that most approaches have in common is an iterative process which means that action research is a cyclic process that introduces learning and improvements in each new iteration. In this way, the researcher can flexibly react to newly introduced change or challenges. It is especially suited for research, where the research goal is not or cannot be specified in detail from the beginning. In this way, action research reveals an explorative character. In the early years of action research, the research process consisted of two basic stages, which were iteratively repeated. The diagnostic stage where social situation was analyzed and studies were introduced in a collaborative way and the therapeutic stage where change was introduced and the corresponding effects were studied (Baskerville, 1999). In 1983, Gerald Susman enhanced this basic approach by three additional stages (Action Planning, Evaluating and Specifying Learning) and embedded it into an environmental setting (Client-System-Infrastructure) (Susman, 1983). Fifteen years later in 1998, the model was transferred to the IS research context by Richard Baskerville and Trevor Wood-Harper (Baskerville & Wood-Harper, 1998). Today this “canonical” action research approach one of the most widely used model for action research in information systems and is also the guideline for the action research that was carried out as part of this work.

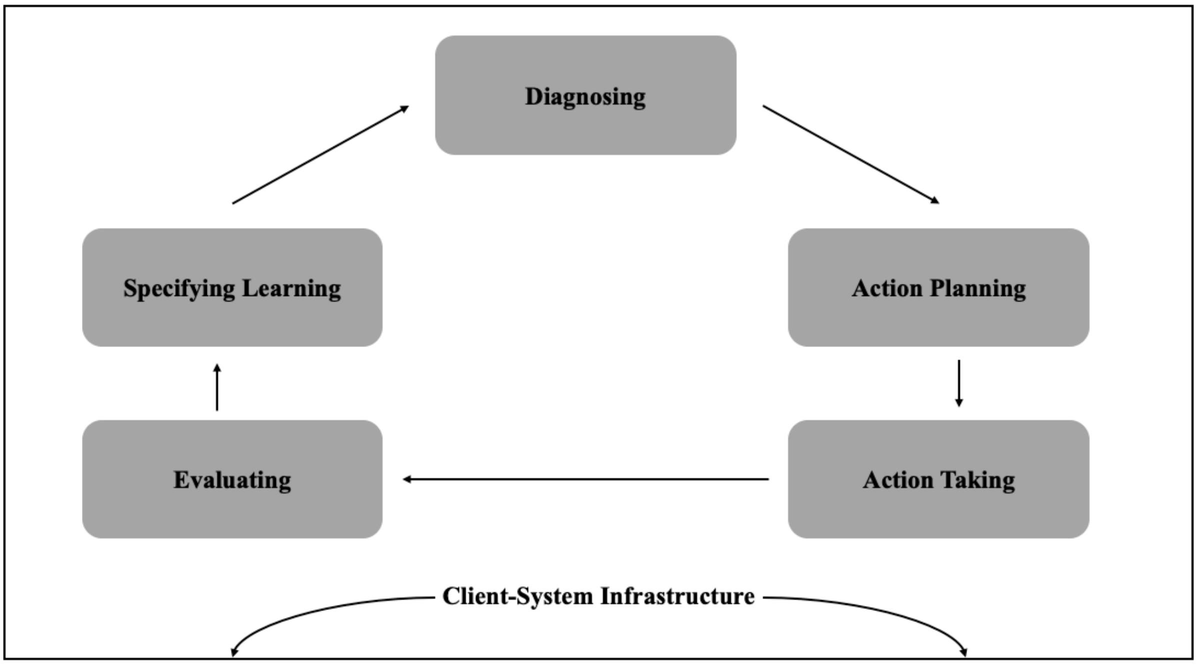



**Figure 8. Action research process by Susman (1983); Picked up by Baskerville & Wood-Harper (1998)**

Before the actual five-step process can be kicked off, the researcher needs to establish an appropriate setup, referred as **Client-System-Infrastructure**. It can be considered as an agreement between researcher and the investigated organization on goals and duties of both sides in the context of the project. Furthermore, it should describe the boundaries of the research and define a clear starting as well as ending point. The main goal of the client-system infrastructure is to enable a collaborative nature of the involvement and to enable a participatory role for the researcher. When the contextual setup was established, the actual process starts with the **diagnosis stage (1)**. Thereby the researcher identifies the primary problems that are anchored in the organizations major drive for introducing change. Furthermore, theoretical assumptions might be developed e.g., working hypothesis about the organization and the problem scope. Thereby, key stakeholders should be identified to enable a better understanding of the existing structure. As soon as the researcher has achieved a sufficient level of understanding for the prevalent situation, the **action planning (2)** should be started. It is the collaborative step of the researcher to define next steps to relieve organizational pain or improve the existing situation. Hereby, the desired future state of the organization should be identified as well as the respective changes that need to be introduced to achieve the specified state. Goal is to have a plan for all actions that are conducted in the subsequent **action taking (3)**, where all planned actions are implemented in the client organization. The action might be implemented directly be the researcher or indirect by the setting up an environment that should influence the organization to move in the desired direction. After the action was taken, the results are collaboratively **evaluated (4)**. If the introduced change revealed positive results, it needs to be identified if the positive outcome was solely established by the change or if other forces influenced the outcome of the evaluation. In case of no positive effect, the reasons need to be identified to be considered in the **specification of learnings (5)**. In this last step of the action research cycle, the researcher collects all insights that were gained during the conducted research iteration, either to take it as base for the next iterative cycle or to derive theoretical insights. Basically, there are three different kinds of learnings that can be extracted from action research. First, it might be revealed that organizational norms need to be adapted to the knowledge gained in the research; Second, in case of an unsuccessful change, the additional knowledge gained can be used as a foundation for further research; And third, the success or failure of an approach can give important insights to the scientific community.

# Part B

# 4 Setting Up a Platform Ecosystem: How to integrate app developer experience (P1)

| | |
|---|---|
| Authors | Weiß, Niklas* (niklas.weiss@tum.de)<br>Schreieck, Maximilian* (maximilian.schreieck@in.tum.de)<br>Wiesche, Manuel* (wiesche@in.tum.de)<br>Krcmar, Helmut* (krcmar@tum.de)<br><br>* Technical University of Munich, Chair for Information Systems, Boltzmannstraße 3, 85748 Garching, Germany |
| Publication | 2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC) |
| Status | Published |
| Contribution of First Author | Introduction, Theoretical Background, Research Design, Results, Discussion |

**Table 2. Fact Sheet Publication P1**

**Abstract**

Platform governance is a fundamental task for operating a platform ecosystem. A platform owner needs to orchestrate the app developers on its platform. Even though platform governance is frequently considered in information systems (IS) research, there is little knowledge on how platform owners can acquire skills and knowledge on platform governance. This is particular relevant in early phases of platforms. This paper strives to shed light on how an automotive manufacturer who has little experience as platform owner creates a platform and improves its platform governance. Based on four episodes from observed in the development department's practices, we identify transfer of knowledge and integration of artifacts as basic mechanisms that enable the platform owner to benefit from interactions with app developers. Our work contributes to the under-investigated field of platform emergence and provides valuable insights for practitioners.

## 4.1 Introduction

Software platforms massively contribute to a disruptive change in multiple traditional industries. Platform companies such as Uber or Airbnb are increasingly successful and attack traditional companies in their industries. In contrast to such platform-native companies, traditional companies need to transform their organization and processes for embracing digital innovation through software platforms (Svahn et al., 2017). As a consequence, such companies need to acquire skills and knowledge on their new role as platform owner and their new task of orchestrating app developers (Rickmann et al., 2014). If they fail to do so, they may endanger not only the platform ecosystem itself, but also traditional value creation processes of the company. Furthermore, suboptimal platform governance may harm the company's public brand image (Keller et al., 2011). Hence, the challenge for a company that is new to the platform business is to embrace their role as platform owner by applying platform governance, without harming current value creation (Stjepandić et al., 2015).

Platform governance refers to mechanisms, which enable the platform owner to exert influence on app developers. It needs to respect developers' autonomy and embrace innovation while also being able to integrate the developers' contributions into the ecosystem as a harmonious whole (Tiwana, 2014). Finding the right balance on platform openness and control in the ecosystem is frequently considered as crucial challenge for platform owners (Eaton et al., 2015; Henfridsson & Ghazawneh, 2013). Further, a platform owner may utilize complementary features, initially provided by app developers, for the evolvement of the platform itself. Bender and Gronau (2017) describe the process of coring as the integration of features provided by applications in the platform core. This may contribute to the platform competitiveness in comparison to competing platform ecosystems (Parker et al., 2017). An emerging platform owner needs to determine all these routines and balances regarding interactions with app developers (Mo & Lorchirachoonkul 2016).

Even though software platforms are frequently considered in current literature, large parts of IS research investigate already established, viral platform ecosystems (de Reuver et al., 2017).The work presented here intends to shed light on the emergence of a new platform owned by a traditional company that has little experience as platform owner. It aims to understand how the company interacts with app developers and in which way it acquires knowledge on governing a platform ecosystem. We state the following research question:

*How can a platform owner utilize interactions with app developers to improve platform governance?*

To answer this question, we investigate a globally operating automotive manufacturer and the establishment of its software platform for automotive onboard applications.

## 4.2 Theoretical Background

A platform ecosystem consists of two basic components. First, the platform core is an extensible code base that serves as foundation for the development of apps by sharing certain features through interfaces. Complementary apps as second component, interoperate with the platform by using the provided interfaces (Tiwana et al., 2010; Verba et al., 2017). The stakeholder that is primarily responsible for the platform core is referred to as platform owner, whereas the development of apps is up to multiple app developers (Tiwana et al., 2010). The relationship between platform owner and app developers has been considered recently in IS research (Eaton et al., 2015; Amir Mohagheghzadeh & Fredrik Svahn, 2016). The platform owner has to orchestrate the platform ecosystem (Tiwana, 2014). Thereby the platform owner manages the tradeoff between an enhancement of external contributions by app developers on the one side and securing the infrastructural control over the platform on the other (Henfridsson & Ghazawneh, 2013). Thereby the platform holds multiple artifacts, called platform boundary resources that serve as interfaces for the relationship between platform owner and app developers (Ghazawneh & Henfridsson, 2010a). Platform boundary resources can be embodied by application programming interfaces (APIs), platform tooling or code documentation. However, in recent IS literature the stream of information and resources has only been discussed in the direction from the platform owner to app developers. Our approach is an investigation of knowledge and artifacts that are transferred in the opposite direction, that is, from app developers to the platform owner.

## 4.3 Research Design

We apply a single case research strategy (Yin, 2009). Our research question considers a specific phase in the evolvement of a platform ecosystem. With the case study method, we meet the requirements for the investigation of such a dynamic, contemporary phenomenon. Furthermore, the large diversity of stakeholders in the context of a digital platform within a traditional company causes a certain level of complexity. Information is heterogeneously distributed across several knowledge providers and multiple projects.

We collected data in the context of a software platform for automotive onboard applications of a globally operating automotive manufacturer. The access to the platform is exclusively limited to employees and subcontractors of the company. Initially, we gathered data by interviews with partners in the platform ecosystem. This includes app developers, platform owners, and employees of the automotive manufacturer that pose the role of customers. All conversations were recorded and subsequently transcribed. Building on an established collaborative practice research initiative at the automotive manufacturer, the first author collected additional data in form of emails, conversation protocols, and entries from internal knowledge boards.

For the coding and analysis of the interview data, we applied grounded theory procedures (Glaser & Strauss, 1967). Within a partial portfolio approach (Wiesche et al., 2017b), 144 codes were identified in an open coding procedure. Further, these findings were condensed to core categories in a selective coding step. The results served as basis for the episodes we describe in the next section.

## 4.4 Results

The following section describes four episodes that describe in which form the platform owner benefits from the involvement of app developers and the respective effects on the investigated platform itself.

| Episode | Description |
|---|---|
| Establishing a Developer Forum | The platform owner established a “question and answer” forum that enables developers to support each other. Furthermore, platform team members support solution finding. The forum was well accepted, and a highly active developer community was established. |

| | |
|---|---|
| Code Review Process | Sample tests of several applications indicated unintended usage of platform mechanisms by applications. This motivated the platform team to conduct code reviews for every app that should be deployed to the production environment. |
| Public Library and SDK | Several app developers claimed missing or insufficient platform features. Even though the implementation or refactoring of the feature was already in planning, the availability for the app developers was too late for their project plans. For this reason, the platform team decided to make platform code repository accessible for the developer community and allow commits from app developers. |
| Provisioning of Start App | Several app developers recognized the challenges and impediments for novice developers to start implementation. For this reason, one team of app developers created a starter app, which contains a feature tour of the platform functionalities, as well as several development tools that were not provided by the platform itself. The platform owner recognized the value of the starter app and decided to officially support it and claim it as the official way for starting a new app development. |

**Table 3. Overview on considered episodes**

### 4.4.1 Establishing a Developer Forum

In the early phase of the platform ecosystem several app developers complained about limited knowledge exchange within the app developer community. Support from the platform team was handled by a front-desk mail address. This caused opacity regarding the provided support since the answer was exclusively visible for the asking app developer:

*"So, the actual how-to knowledge is not transferred. […] This is absolutely a problem, that code is pushed around exclusively and that there is no platform for knowledge management."*

To tackle this insufficient procedure, the platform team established a developer forum that enables knowledge exchange between all involved stakeholders. App developers provide mutual support, and the platform team also engages to provide problem solutions. This form of public knowledge board enabled the platform owner to recognize and understand problems of the app developer community without any direct interaction. In this way the knowledge transfer from developers to the platform owner is standardized and visible for every party of the app development community.

### 4.4.2 Code Review Process

Sample tests of several applications indicated unintended usage of platform mechanisms by applications. This motivated the platform team to conduct code reviews for every app that should be deployed to a production environment. In this way, unintended behavior should be avoided, and platform stability should be improved. By reviewing the apps, the platform team recognized volatile quality of the submitted artifacts:

*"We just recognized that there are apps that perform fine whereas others reveal bad mistakes. We are not sure why this is the case. We have just assumptions."*

Based on the most frequently identified mistakes, the platform team creates several tutorials and coding guidelines. Besides notifying the app development teams regarding the identified mistakes, face-to-face meetings are organized. Within these conversations, the mistakes are discussed and agreements on rectifications on the app side are made. This approach supports the learning process on the platform owner side. Furthermore, by considering the app developer perspective in the face-to-face meetings, the platform team was able to understand why the identified mistakes were made.

### 4.4.3 Public Libraries and SDK

After some weeks of app development in the platform ecosystem, several app developers claimed missing or insufficient platform features. Even though the implementation or refactoring of these features was already in planning, the availability for the app developers was too late for their project plans. For this reason, the platform team decided to make platform code repository accessible for the developer community and allow commits from app developers:

*“And then there are cases, where you catch a new (platform) version with hardly any changes und suddenly an arbitrary generic error message pops up and you have to investigate what the actual concern is. A random feature is not working, and you don’t know if this is a bug or if it is intended, since everything is in movement.”*

By default, the code repository is accessible for every participant in the ecosystem. When an app development team recognizes the demand for an unavailable platform feature, it usually requests the implementation by the platform team. However, the required change may not fit into the working plan of the platform team. Then, the app team has the option to implement the feature by itself. By reviewing the submitted code, the platform team ensures a sufficient quality of the implemented feature.

### 4.4.4 Provisioning of Starter App

Several app developers recognized the challenges and impediments for novice developers to start implementation. It took large efforts to onboard new team members in already ongoing implementations. Furthermore, new teams were engaged in infrastructure and setup related tasks over weeks before they were able to start the actual building of their apps:

*“It is really hard to get running. It takes you weeks – also if you are experienced with ancestor generations of platform development – to start working in a productive way.”*

Motivated by improving this situation, one team of app developers created a starter app containing a feature tour of the platform functionalities as well as several development tools that were not provided by the platform itself. The feature tour contains examples that demonstrate the utilization of the most relevant platform interfaces such as user interface APIs or interfaces for triggering certain activities in the car’s navigation system. Furthermore, the tooling provides features that are not crucially necessary for app development on the platform, though they are mandatory for professional software development as scripts for static and dynamic code analysis or automated deployment to the test environment. The starter app was spread in the developer community and became best practice for every app team that started developing. After recognizing the large advantages of the starter app, the platform team decided to claim it as standard support it officially. The app developers that initially created the starter app still contribute and support the maintenance of the artifact.

### 4.5 Discussion

Before analyzing the actual interactions between platform owner and app developers, we need to emphasize the status of the platform ecosystem we investigated. The establishment of a software platform that is exclusively accessible for app developers that are employees of the platform company enables the involvement of app developers in the process of platform evolution. Internal app developers will not have the intention to harm the ecosystem. Concurrently, they are able to provide unfiltered and veritable insights from the app developer perspective to the platform owner. This enables organizational learning without a potentially harmful platform tuning process by app developers (Eaton et al., 2015).

Considering the interactions between platform owner and app developers, we identify several types of interactions (Table 4).

| Episode | Knowledge Transfer | Integration |
|---|---|---|
| Establishing Developer Forum | • Common problems of app developer community<br>• Best practices and solutions of app developer groups<br>• Specific problems of single app developers | • Solutions suggested in forum are implemented by the platform team |
| Code Review Process | • Common error patterns<br>• Opportunities for unintended app behavior<br>• Best practices of app developer groups<br>• Problems and challenges of single app developers | • Integration of good practices in platform documentation<br>• Derivation of platform guidelines from unintended app behavior |
| Public Libraries and SDK | • Common problems of app developer community | • Direct platform development by app developers<br>• Joint development and maintenance |
| Provisioning of Starter App | • Common pain point for new app developers | • Integration of app artifact in platform<br>• Joint development and maintenance |

**Table 4. Identified interaction types between app developers and platform team**

Involving app developers in a platform ecosystem is frequently considered under the aspects of transparency and accessibility (Benlian et al., 2015). However, these considerations illustrate interactions from an app developer's perspective. We invert this perspective and apply these aspects to the platform owner's view. We consider the observed transfer of knowledge as transparency of app development activities towards the platform. The observation of app developers enables the platform owner to detect omissions and failures in the platform and to improve these aspects. Foerderer et al. (2019) identified three levels of knowledge transfer from platform owner to developer: broadcasting, brokering, and bridging. Following our logic of assuming the platform owner's perspective, we invert this consideration and reveal three ways of knowledge transfer from app developers to platform owner.

(1) *Reverse Broadcasting* is a standardized knowledge transfer in the context of an arm's length governance between platform owner and app developers. In our observations we can find this form of knowledge transfer in the developer forum. The platform owner is able to gain knowledge on common problems of the app developer community by scanning standardized forum posts without any direct interactions. (2) *Reverse Brokering* is an intermediate type of knowledge transfer in which the platform owner may interact with a subset of app developers. It occurs in our data in the episode on the provisioning of the starter app. After recognizing the large distribution of the starter app, the platform owner contacted the app team that created the artifact. The app developers reported the need for more support for app developers that start with their work. Thereby they are able to provide insights from the app developer perspective directly to the platform team and thereby contribute to learning on the platform owner side. (3) *Reverse Bridging* is a direct and problem-specific knowledge exchange of platform owner and app developers. This mechanism can be identified in the code review process. After analyzing the code of a specific app, the review team schedules a meeting with the app developers. Beside the agreement on certain patches in the app, the platform team tries to understand the emergence of the mistakes made when using platform features.

Accessibility is the second facet of involving app developers in platform ecosystems. In our consideration, it embodies the capabilities to integrate solutions of app developers directly into the platform. Similar to the transparency aspects we distinguish three levels of accessibility. (1) *Integration of community activities* describes a process in which the community develops solutions or features that are integrated in the platform without any direct interaction between platform owner and app developers. One example for this can be found in the episode on public libraries and SDK. Similar to an open source contribution work flow, there is no direct interaction required for integrating code of app developers into the platform. The (2) *integration of best practices* covers the platform owner recognizing the evolvement of a common praxis and cooperates with the responsible team to integrate the feature into the platform. This mechanism can be found in the episode on the starter app. The platform owner was convinced by the developed artifact and integrated it into the platform in cooperation with the app developer team. Finally, the (3) *integration of single solutions* is the transferring of a specific problem's solution from an app into a platform feature in a mode of close cooperation. This can be found in the episode on the developer forum. In multiple cases, the platform team adopted the specific solutions posted in the forum.

## 4.6 Conclusion

In this study, we investigate how interactions with app developers enable platform owners to master platform governance. We confirm the forms of knowledge boundaries in platform ecosystems established by Foerderer et al. (2019). However, by inverting the direction of information flow we extend the existing body of knowledge. Furthermore, we transfer the different levels of interactions to the integration of real artifacts into the platform. Thereby, we add the aspect of platform boundary resources as tool for platform evolvement to the existing body of knowledge. Finally, we highlight the importance of artifacts like a developer forum, a starter app or a code review process, providing valuable insights for platform owners in the field.

# 4 Learning to be a Platform Owner: How BMW enhances App Development for Cars (P2)

| | |
|---|---|
| Authors | Weiß, Niklas* (niklas.weiss@tum.de)<br>Schreieck, Maximilian* (maximilian.schreieck@in.tum.de)<br>Wiesche, Manuel** (wiesche@in.tum.de)<br>Krcmar, Helmut* (krcmar@tum.de)<br><br>*Technical University of Munich, Chair for Information Systems, Boltzmannstraße 3, 85748 Garching, Germany<br>** Technical University of Dortmund, Chair for Digital Transformation, Martin-Schmeißer-Weg 12, 44221 Dortmund, Germany |
| Publication | IEEE Transactions on Engineering Management (Early Access) |
| Status | Published |
| Contribution of First Author | Introduction, Theoretical Background, Research Design, Results, Discussion |

**Table 5. Fact Sheet Publication P2**

**Abstract**

Platform owners face multiple challenges such as onboarding and orchestrating app developers as well as providing resources to enable the development of complementary apps. Information systems research considers digital platform governance as key to address these challenges. Thereby, the focus lies on the relationship of a platform owner and app developers. However, while there is evidence how app developers acquire skills through these interactions, there is limited knowledge of how platform owners benefit from interacting with app developers to improve their digital platforms. To address this gap, in this article, we study the emergence of a digital platform for automotive onboard apps within the BMW Group. Our results are grounded in 30 expert interviews that we conducted during a period of two years and are enriched by extensive secondary data. We identify transfer of perspective, transfer of knowledge, and transfer of artifacts as basic mechanisms that enable a platform owner to enhance its digital platform. The inherent improvements of the digital platform facilitate the app development. Our work extends the existing theory on platform emergence and provides insights into the learning process of an inexperienced platform owner. Our findings reveal valuable recommendations for organizations that are struggling to establish digital platforms.

**Managerial Relevance Statement**

In this study, we analyzed in which way a platform owner acquires knowledge from interactions with app developers during the emergence of a digital platform. Our results have important implications for practitioners. First, our findings illustrate challenges of a manufacturing organization that transforms itself into a provider of digital services. This new role requires new skills. Our case illustrates how an organization can learn from app developers when becoming a platform owner. Therefore, managers should promote organizational setups that enable learning in the context of an emerging digital platform. Second, managers should enhance interaction channels for platform owner and app developers. Even though investing in such channels does not create direct value for the platform owner, they foster the emergence of communities-of-practice, which enable learning. Third, our findings help unexperienced platform owners identify the optimal point in time to start app development activities. App development activities that are initiated too early suffer from a lack of standardization by the platform. This causes problems for the platform team which needs to align these early apps to standards that are introduced at a later point/subsequently. Hence, managers should consciously decide which level of maturity is required for opening the platform for app development activities.

## 5.1 Introduction

The phenomenon of digital platforms is a major trend in research as well as practice that disrupts multiple industries. Scholars try to understand in which way Intel could dominate the microprocessor industry for years (Gawer & Cusumano, 2014), Apple has become one of the most valuable firms on the planet (den Hartigh et al., 2015) or game developers decide for specific console platforms (Srinivasan & Venkatraman, 2020). However, while this work illustrates how firms master the orchestration of their digital platform ecosystem (Huber et al., 2017), organizations need to acquire knowledge on platform ownership during the emergence of their digital platform (Hein, Schreieck, et al., 2019b; Svahn et al., 2017). Even though organizational learning has been broadly investigated for new product development (Cui et al., 2014), project management in firms (Williams, 2008) or open source software (Huntley, 2003), the learning process of a platform owner during the emergence of a digital platform has not been considered so far. To address this gap, this paper illustrates platform emergence within an established firm from an organizational learning perspective.

We understand a digital platform as a set of stable components that supports variety and evolvability in a system by constraining the linkages among the other components (Baldwin & Woodard, 2008). It incorporates a central core surrounded by multiple actors in its digital platform ecosystem (Hein et al., 2020; Tiwana, 2014). Its generative capabilities enhance innovation and accelerate development processes of digital services (Chesbrough & Bogers, 2014). Hence, competition no longer revolves around the control of classic value chains but around attracting generative activities associated with a digital platform (de Reuver et al., 2017). One major shortcoming of existing research on digital platforms is its focus on matured digital platform ecosystems (de Reuver et al., 2017): Other than a few exceptions (Amir Mohagheghzadeh & Fredrik Svahn, 2016; Svahn et al., 2017; Tiwana & Bush, 2014), scholars mainly consider established digital platforms from an ex-post perspective (Eaton et al., 2015; Foerderer et al., 2019; Henfridsson & Ghazawneh, 2013; Karhu et al., 2018). While there is comprehensive knowledge on mechanisms and processes of successful orchestration of digital platforms, the path to achieve a matured status remains unclear. In their broadly recognized "research agenda on digital platforms", de Reuver et al. (2017) shift the focus towards platform emergence. They raise the question if a successful digital platform can be consciously designed or whether it is just a result of coincidences. Even though the platform may be refined over its whole lifetime (Eaton et al., 2015), it is initially designed during platform emergence. Considering the large challenges of a platform launch (Schirrmacher et al., 2017; Stummer et al., 2018), the platform design is particularly critical in this phase. Therefore, an inexperienced firm needs to acquire knowledge on the design of digital platforms and its associated ecosystem (Rausch et al., 2013). However, it is unclear in which way these firms can do so. While learning mechanisms on the app developers' side are broadly considered (Eaton et al., 2015; Foerderer et al., 2019; Henfridsson & Ghazawneh, 2013; Srinivasan & Venkatraman, 2020), the platform owners' acquisition of knowledge remains opaque. Organizational learning theory provides a perspective that allows us to describe and understand learning effects of an inexperienced platform owner. Thereby, we focus on interactions, which are a basic condition for organizational learning (Simon, 1991).

In a platform organization the platform owner is responsible for the development of a stable core which is provided to developers that build complementary apps (Baldwin & Woodard, 2008). Both sides embody communities-of-practice that constantly adapt and evolve their abilities by accumulating knowledge by practical experience (Brown & Duduid, 1991). Moreover, the platform owner is responsible for the governance of the interactions between these parties (Tiwana, 2014). For example, a platform owner provides platform boundary resources to enable app developers to exploit platform capabilities (Foerderer, 2017). However, interactions always provoke mutual learning (March, 1991). The exchange between both sides should bring the articulation of knowledge also from app developers to the platform owner. With this consideration, we define our research question: "Which mechanisms enable a platform owner to learn digital platform design from interactions with app developers?" To address our research question, we analytically explored the coherences of organizational learning and platform governance. We argue that transfer of knowledge from platform owner to app developers is a proven pattern in digital platform theory (Foerderer et al., 2019), while the opposite direction has remained understudied so far. To address our research question, we conduct an exploratory case study in the context of a digital platform for automotive onboard apps at the globally operating car manufacturer BMW. By analyzing our qualitative data set with grounded theory procedures (Glaser & Strauss, 1967), we identify 25 representative events within four episodes that richly illustrate how the platform owner improved platform design according to interactions with app developers. These episodes serve as the basis for our theorizing on three basic mechanisms that enable a platform owner to learn: (1) transfer of perspective, (2) transfer of knowledge, and (3) transfer of artifacts.

By addressing this question, we can contribute to a better understanding on coherences in an initial phase of platform design and its meaning for later platform success. For this research, we posit that platform design is a critical factor for the launch of the digital platform and the attracting of app developers. Furthermore, we prove the applicability of organizational learning theory to the new context of digital platforms. By intertwining literature on organizational learning with platform governance, we contribute to prevalent theory on product development and innovation management.

The remainder of this paper is structured as follows. The next section describes the theoretical background of our considerations and clarifies the gap in prevalent theory. Secondly, our methodological approach and the case of BMW as owner of a just emerging digital platform for automotive onboard apps is described. Then we illustrate the four episodes we identified in the context of platform design and describe our findings regarding concrete learning of the platform owner. The subsequent discussion theorizes which general mechanisms enable learning regarding platform design and in which way our findings are embedded in prevalent literature.

## 5.2 Theoretical Background

The following section illustrates in which way our research is embedded in prevalent literature. We therefore explain general coherences of platform governance and subsequently introduce the lens of organizational learning theory. Our considerations reveal the lack of research on platform emergence in general and learning effects within a platform owner that occur through interactions with app developers in particular.

### 5.2.1 Platform Governance

The core of a digital platform is in the center of every digital platform ecosystem (Hein et al., 2020; Tiwana, 2014). The exposed capabilities of the digital platform attract app developers (Dellermann et al., 2016; Kude et al., 2012), who build complementary apps on the one side, which are consumed by customers on the other side (Tiwana, 2014). Platform governance is the "partitioning of decision-making authority between platform owners and app developers, control mechanisms, and pricing and pie-sharing structures" (Tiwana, 2014). For instance, in the iOS ecosystem, Apple as platform owner decides, which application programming interfaces (APIs) of the iPhone are accessible by third-party app developers (Eaton et al., 2015). Furthermore, each app needs to be submitted for review by Apple before it can be launched in the App Store. App developers need to choose the price of their app in a predefined selection of pricing steps and keep just 70% of the revenue. The rest belongs to Apple as platform owner. However, platform governance should also facilitate the creation of new apps (Kude et al., 2012). By exposing generativity, the platform owner unleashes potential for innovation (Tilson et al., 2010). With the annual release of a new iOS version, Apple publishes new APIs that provide new possibilities for app developers. Comprehensive documentation and annual developer conferences inform and attract developers to build new apps. A platform with strictly controlled generativity would counter such endeavors (Ondrus et al., 2015). Hence, the platform owner needs to balance control over the digital platform and its ecosystem on the one side and provide a certain level of autonomy for app developers to foster innovation on the other (Rausch et al., 2012). In this way, platform governance covers multiple tactical decisions that impact interactions between the platform owner and app developers (Kude et al., 2012; Schreieck et al., 2016).

Usually these decisions are transmitted and enforced by resources that enable app developers in their activities (Ghazawneh & Henfridsson, 2010a). While there are technical resources such as software development kits (SDK) and development tools, the platform owner also provides knowledge on app development as boundary resource. Foerderer et al. (2019) describe multiple levels of knowledge transfer through boundary resources. By transferring knowledge from the platform owner to app developers via multiple channels, app developers are able to choose the most appropriate way for their individual requirements. The design of platform boundary resources is constantly shaped and evolved in a common refining process of platform owner and app developers (Eaton et al., 2015; Srinivasan & Venkatraman, 2020). Since platform boundary resources correspond to the platform owner's decisions on platform governance, refinements on platform boundary resources mirror adjustments of platform governance decisions. The adaption of platform governance indicates learning effects within the organization of a platform owner.

The accessibility of the provided platform resources is defined by the configuration of the platform's vertical openness (Boudreau, 2010; Eisenmann et al., 2011; Zaggl et al., 2020). Vertical openness defines the degree of accessibility and transparency of platform boundary resources for external actors. A platform owner can decide to limit access to certain platform assets for a defined group of users or restrict the usage of specific resources. In this way, vertical openness decides on the potential of knowledge transfer from platform owner to app developers. In the other way around, a platform owner can pull external innovation back into the core of its platform (Bender & Gronau, 2017). Even though the potential danger of getting replaced by a platform feature may discourage app developers from entering an digital platform ecosystem, Parker et al. (2017) argue that the overall digital platform ecosystem mostly benefits from improvements of the platform core through coring. In addition to the degree of vertical openness, a platform owner has to decide on the horizontal openness of its platform (Eisenmann et al., 2011). This refers to the interoperability with other platforms and the willingness of sharing platform ownership with others. In specific cases, the platform is not orchestrated by a single owner (as Apple) but driven by a developer community, which is globally distributed (Huntley, 2003). Even though, also these platforms require governance, different members of the community consolidate certain decisions. In the Linux platform, the app developers are able to get involved as platform owner and contribute to platform core by submitting pull requests. Hence, the Linux platform exhibits a high level of horizontal openness. Prior research illustrates that the degree of a platform's openness change over time (Ondrus et al., 2015). In their extensive case study, Karhu et al. (2018) describe in which way the openness of Google's Android decreased over the last decade. Likewise, the adaptions of platform openness indicate that the platform owner learns on the optimal degree of openness for the platform over time.

Boundary resources as well as openness constitute crucial aspects of platform governance that are affected by constant adoptions during the platform's lifecycle. However, we argue that especially the phase of platform emergence requires appropriate platform governance to manage the initial challenges of a new digital platform (Schirrmacher et al., 2017; Stummer et al., 2018). Platform boundary resources that do not enable app developers in a sufficient way may have the potential to discourage app developers and aggravate the chicken-egg problem (Stummer et al., 2018). The same effect may apply to a platform with an insufficient level of openness and inadequate transparency or accessibility to app developers. An excessive level of openness on the side may involve the danger of losing control (Ondrus et al., 2015). Our research illuminates the crucial stage of platform emergence and provides valuable insights that complement knowledge on digital platforms and support inexperienced platform owners in designing their platform.

### 5.2.2 Organizational Learning in Digital Platform Ecosystems

Even though the refinement of platform boundary resources as well as platform openness indicates the existence of learning effects within the organization of a platform owner (Eaton et al., 2015), such mechanisms were exclusively described on the app developers' side so far (Foerderer et al., 2019). To shed light on this research gap we take the lens of organizational learning. It is understood as "a change in the organization that occurs as the organization acquires experience" (Argote & Miron-Spektor, 2011). There are two basic processes that enable organizational learning (Li et al., 2010). First, acquisitive learning, which is the exploitation of access to preexisting knowledge and its subsequent implementation. In the context of digital platforms, acquisitive learning occurs whenever an app developer consumes documentation that was provided by an external source such as the platform owner. Second, experimental learning, describes the extraction of first-hand knowledge through own experience (Dess et al., 2003; Li et al., 2010; Zahra et al., 1999). It describes the process of an app developer that learns best practices and code patterns during the actual programming activity. The ability to learn is critical for the performance of an organization and its long-term success (Argote & Miron-Spektor, 2011; Grant, 1996). Hence, there is a large amount of research that investigates the effectiveness of organizational learning and its influence on the firm's performance (Adler & Clark, 1991; Argote et al., 1990; Cui et al., 2014). However, measuring the effects of a phenomenon requires to understand its characteristics. Though we argue that there are indications of organizational learning within the organization of a platform owner, there is no evidence so far. Therefore, we align our efforts on another stream inside the organizational learning literature to illuminate the actual organizational learning process and its strategical implications (Chuang, 2014; Huber, 1991; Maria et al., 2018; Moellers et al., 2018; Williams, 2008).

In general, learning may cause a change of behavior, a change of cognition or both (Crossan et al., 1995). Furthermore, many researchers assume a link between learning and an improvement of performance (Knight, 2002). Henard and Szymanski (2001) argue that learning within a development team leads to a product advantage, which describes the degree to which a product is superior to its market alternatives. However, incorrect learning or the learning of incorrect coherences may even diminish the subsequent performance of an actor (Huber, 1991). To evaluate learning mechanisms, Crossan et al. (1995) define three dimensions that need to be considered. First, the unit of analysis needs to be clarified. In the context of our study, the digital platform and its governance represent the unit of analysis. Second, the outcome of learning needs to be analyzed. This affects changes in cognition or behavior. We address this aspect by considering changes in the platform or its governance as result of platform owner's learnings. Lastly, the link between learning and performance needs to be clarified—which we understand as improvements of the platform design or its governance.

To understand the characteristics of organizational learning within the organization of a platform owner we need to elucidate the mechanisms that evoke learning. We understand organizational learning in terms of communities that emerge. People work and learn collaboratively, and vital interstitial communities are formed and reformed. Since learning is a commuting process, the interaction of different communities enables mutual insights and an overall increase of knowledge (March, 1991). The higher the amount of interactions, the larger the chance to learn from experience of others (Levitt & March, 1988). Information circulates fast inside of communities. Between communities and its environment information gets lost or is distributed slowly. Transferring this idea to a digital platform, we consider the app developers as one community and the platform owner, with its collaborating team members and different organizational units, as another community. Both communities are embedded in a common network, the digital platform ecosystem (Ceccagnoli, 2012; Hein et al., 2020). In general, different actors in a network are able to generate assets through interactions such as knowledge-sharing routines and complementary endowments (Dyer & Singh, 1998; Mesquita et al., 2008). Considering a digital platform ecosystem as a network of different actors, a platform owner can facilitate such mechanisms through appropriate governance. Smedlund and Faghankhani (2015) emphasize the importance of interactions between participants in a digital platform ecosystem to facilitate reciprocal learning and co-evolution. Hence, an incremental improvement of the platform and the complementary products can be achieved.

Even though organizational learning of focal firms was considered in the past, we argue that digital platform ecosystems are different. While collaboration with suppliers considers organizational learning in traditional value chains (Maria et al., 2018), interactions in the context of digital platforms are affected by network effects (Parker & Van Alstyne, 2005). The organizational learning of latecomer firms and the constitution of technological capabilities (Chuang, 2014) differs also by the creation of values in pipelines and the lack of network effects (Parker, Van Alstyne, & Choudary, 2016). Furthermore, the collaboration of different units inside an organization for new product development activities (Moellers et al., 2018) differ from a platform setup by its incentive. A firm’s employees develop a product that serves customer needs, and receives extrinsic motivation, for example through their wages. However, app developers immediately benefit from platform improvements and therefore, reveal an intrinsic motivation for interactions with the focal firm—in this case the platform owner (Kude et al., 2012). We strive to understand in which way these interactions are perceived by the platform owner in the context of organizational learning. With theories on platform boundary resources, knowledge resources and platform openness, the learning mechanisms of app developers through interactions with a platform owner are well understood. However, the

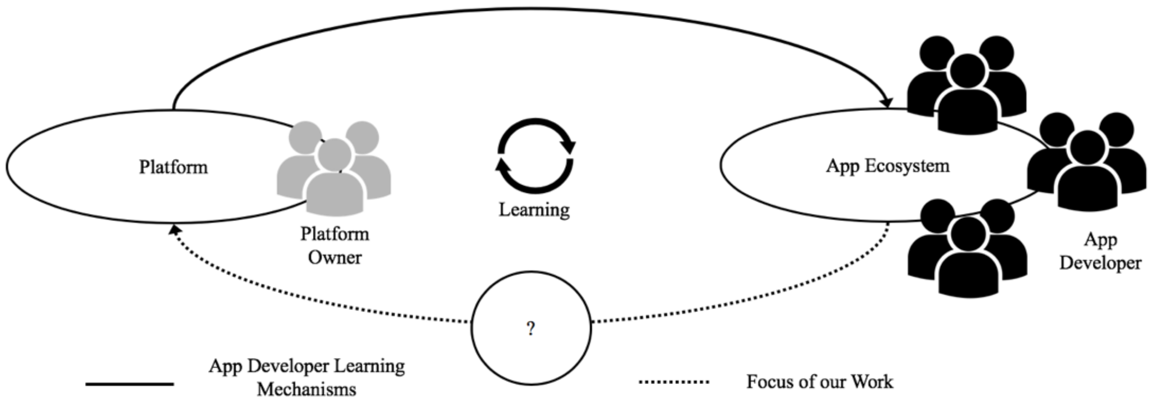



**Figure 9. Platform owner learning mechanisms as focus of our work**

opposite direction remains understudied / opaque (see Figure 9). We argue that understanding platform design is especially crucial during the emergence of a digital platform in which the foundation for success or failure is laid. Furthermore, research on organizational learning indicates that interactions cause mutual learning effects.

## 5.3 Methodology

Our single case study (Eisenhardt, 1989; Yin, 2009) provides longitudinal insights into the emergence and early evolution of a platform for automotive onboard apps at the BMW Group. Driven by evolved customer expectation, the globally operating car company is transforming from a pure car manufacturer into a provider of digital services. This includes the development and operation of a platform for automotive onboard apps, which allows a flexible deployment of new features in the form of apps for the car, even after the vehicle has left the production plant. Following our research question, we strived to identify and understand learning mechanisms at the department that is responsible for the development and operation of the app platform. With the case study method, we satisfy the requirements for investigating such a dynamic and fast evolving phenomenon. Furthermore, observing organizational learning requires collecting "time-series or longitudinal data" (Argote & Miron-Spektor, 2011). Therefore, we our studies covers the period of two years.

The large diversity of stakeholders in the context of a digital platform and its ecosystem causes a tremendous level of complexity (Rausch et al., 2013). Information is distributed heterogeneously to several knowledge providers and multiple projects. In our study, we gathered data with 30 semi-structured interviews (Gläser & Laudel, 2009) with 26 actors in the digital platform ecosystem to synthesize a comprehensive understanding of coherences and dynamics in the interactions of different ecosystem actors. Interview participants include representatives from the platform owner side as well as the app developer side. Due to their central role within the digital platform ecosystem, two app developers as well as two members of the platform team were interviewed twice. All conversations were recorded and transcribed. Based on a collaborative practice research initiative (Mathiassen, 2002), the first author engaged in the daily activities of the platform team which allowed the collection of additional data in form of meeting protocols, code in development repositories and entries from internal knowledge boards. This participation within the two years of our research enabled the collection of rich data which is illustrated in Table 6.

| | **Data Source** | **Amount** | **Use in Analysis** |
|---|---|---|---|
| **Primary Data** | Platform team members | 16 interviews (à 45 - 60 minutes each) | Insights into the beliefs, motivations, and strategies insight the platform team |
| | App developer | 14 interviews (à 45 - 60 minutes each) | Insights into the beliefs, motivations, problems and desires of app developers |
| **Secondary Data** | App review meetings | 49 meetings (à 45 minutes each) | Insights into problems, perceptions, and strategies of the app review team |
| | Community meet-ups | 8 meet-ups (à 90 minutes each) | Insights into the culture and communication between platform team and app developers |
| | Platform team meetings | 23 meetings (à 180 minutes each) | Insights into the decisions, motivations, and strategies insight the platform team; |
| | Code repository of internal developer portal | 146 commits | Insights into the development of content and amount of documentation |
| | Analytics of internal question and answer forum | 735 discussions threads | Insights into engagement of the platform team and app developers as well as concerns and problems of app developers |

**Table 6. Overview of collected data**

For the coding and analysis of the interview data, we applied a partial portfolio approach (Wiesche et al., 2017a) of the Grounded Theory methodology (Glaser & Strauss, 1967). The goal of Grounded Theory is the creation of a theory that accounts for patterns of behavior which is relevant and problematic for those involved (Glaser, 1978). The idea, that empirical facts are the starting point of theory development and research in general and not the reference point of empirical assessment, is the key element of the Grounded Theory. The analysis of the collected data was conducted in three coding steps. The first step was open coding, a detailed, line-by-line analysis of the collected data that helps create a broad understanding of the phenomena by the researcher (Allan, 2006). The open coding of our collected interview data yielded 254 codes. The subsequent selective coding step considers the emergence of a core category which unites all identified issues under one thematic umbrella. The core category needs to be central, occur frequently in the data and should be related to most of the identified categories. In our research, the category that satisfied all of these criteria was "learning of the platform owner". Finally, we performed the step of theoretical coding. It is the property of coding and constant comparative analysis that yields the conceptual relationship between categories and their properties as they emerge (Glaser, 1992). The step of theoretical coding should reveal the generalizable contribution of the research.

Since Grounded Theory does not claim to be a perfect and finished product, but underlies permanent development, theory development should be presented as an ongoing process. We addressed this point by constantly refining our theorizing and collecting additional data within the considered period of two years. Due to the revealed level of theoretical saturation of refinements in our perceptions (Glaser, 1992), we decided to terminate our research after two years and present the results in this study.

### 5.4 The BMW Onboard App Platform

To explore learning mechanisms of a platform owner in the context of an emerging digital platform, we choose the case of BMW and its platform for automotive onboard apps. The automotive industry is heavily affected by digitalization and the inherent change in customer expectations. The car is considered as digital device that receives frequent software updates and provides options for extensions and customization of software features via apps. The considered app platform is one measure that enables BMW to face these new challenges. It enables modular deployment of apps in the car. The platform is part of the BMW OS 7.0, the company's latest infotainment system, released in summer 2018 with the start of production of the latest BMW X5 series. Large numbers of the models that were released since then as well as upcoming BMW models will run this system. By exposing multiple functionalities of the car to apps, the platform provides the base for a broad spectrum of use cases. At the release, more than 20 apps were available, providing services as a parking lot finder[12], music streaming[13], Microsoft Office 365[14] and apps for different BMW service calls[15].

---

[12] https://www.bmw-connecteddrive.de/app/index.html#/portal/store/Base_Connected_Parking retrieved 19.02.2019

[13] https://www.bmw-connecteddrive.de/app/index.html#/portal/store/Base_Connected_Music retrieved 19.02.2019

[14] https://www.bmw-connecteddrive.de/app/index.html#/portal/store/Base_ExchangeOffer retrieved 19.02.2019

[15] https://www.bmw-connecteddrive.de/app/index.html#/portal/store/Base_AssistNoTPEGOffer retrieved 19.02.2019


**Image 1. BMW OS 7.0**

The platform is managed by a development department, which is responsible for the vehicle's connectivity. We refer to this department as the "platform team" in the context of this study. The platform is used by app development teams from multiple, different departments, which are distributed all over the organization. The platform development started in May 2016. After an initial phase of pure platform development activities, app development teams from the infotainment service department as well as telematics department started their activities in December 2016. More and more apps started their development in the following month. Figure 10 presents an overview 25 identified events that revealed learning mechanisms for the platform owner. The events are clustered into four major episodes, which provide the structure for the subsequent illustration of our results.

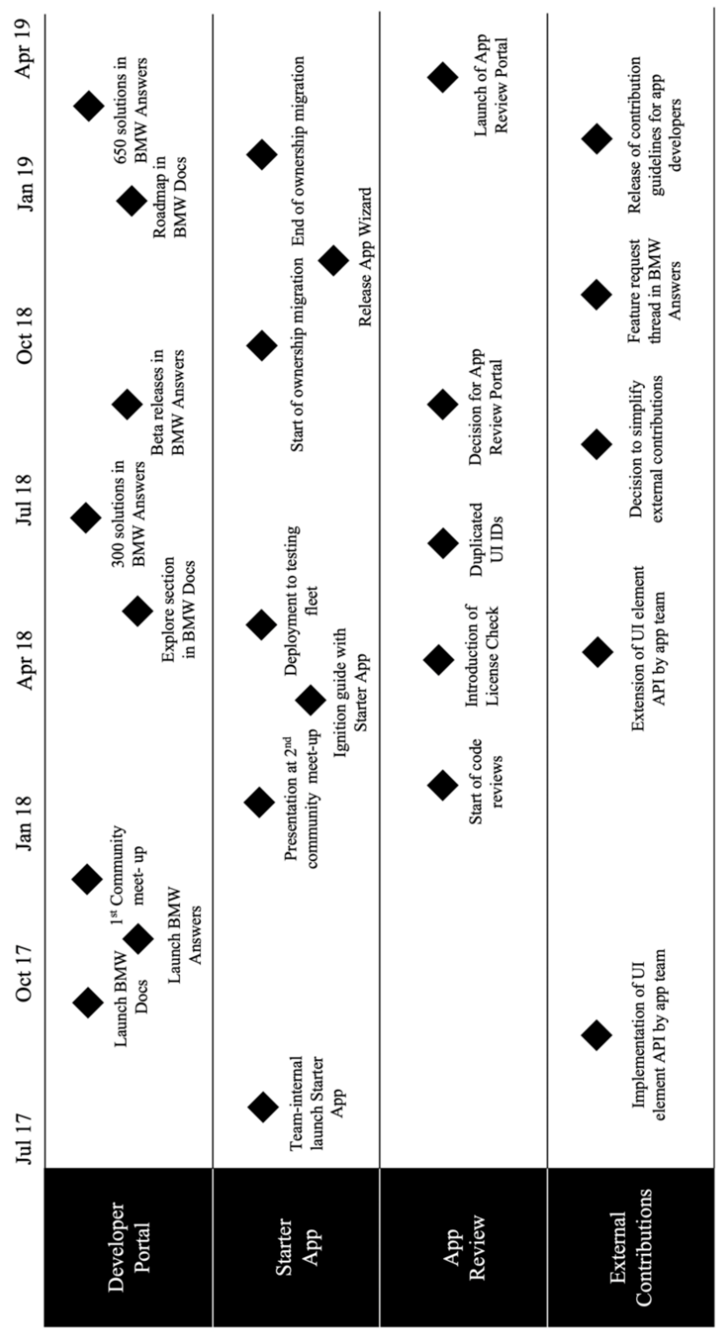



**Figure 10. Identified events in the context of the BMW app platform**

### 5.4.1 The Developer Portal

In the summer of 2017 several app developers complained about documentation that was stored in multiple wiki spaces inside BMW's intranet. Access to these pages was limited and the information these pages contained was fragmented and partially contradictory.

*"So, the actual how-to knowledge is not transferred. [...] This is absolutely a problem, that documentation is pushed around exclusively and that there is no central platform for knowledge management." ~ App developer*

A new app developer was blocked for weeks, waiting for access to all information and resources that were required for app development. In August of 2017 the platform team decided to support the plans of a developer portal that was available to every employee inside of BMW and represented a single point of truth. In October 2017 the "BMW Docs" was launched. It included a basic ignition guide with a description of all required steps for the setup of a developer environment on a programmer's machine and further basic guides for app developers. From this point on, new documentation was pushed to the developer portal. Furthermore, existing documentation, which was distributed across several wiki pages, was transferred. After a while, the platform team recognized a significant decrease of support requests from app developers. However, the team considered the support infrastructure as insufficient. Requests from app developers were handled by a front-desk mail address. Members of the platform team checked the mailbox and answered all requests. By doing so, the answer was only visible to the app developer who asked the question, even though other developers faced the same issues. To tackle this insufficient procedure, the team decided to establish a BMW internal question and answer forum for developers, similar to the public Stack Overflow forum on the web.

*"We recognized that the prevalent support via mail simply did not scale. We had to answer identical requests multiple times while developers waited a long time for a response in certain cases." ~ Platform team member*

"BMW Answers" started in November 2017. From then on, when app developers faced a problem, they were advised to open a thread in the question and answers forum. Similar to "BMW Docs" the forum was open to everybody inside the BMW network by default. Any direct requests to the platform team via the former support mail address or another channel were rejected with direction to go BMW Answers. The forum was quickly adopted by the app developer community. Within the first month, the forum revealed almost 50 daily active users. After a few weeks the team recognized that the number of support requests decreased dramatically. One reason for this was rooted in the searchable content of the forum. Whenever a problem is solved by a reply, the creator of the thread was encouraged to mark the appropriate post as solution. In this way, the solution was easy to find for a later visitor of to the topic. If a solution was provided once, other app developers just needed to search the forum to solve the identical problem. Furthermore, the platform team recognized that experienced app developers started to respond to posts of other developers, so that there was no need for the platform team to engage. In June 2018, the 300th solution was entered into BMW Answers; by March 2019, more than 650 issues had been solved. To support this formation of a community, the platform team decided to organize periodic community meetings, which were to take place every one to two months. These "meet-ups", which were open for everybody interested, should facilitate connections between different app development teams and encourage mutual knowledge exchange. A meet-up usually started with a technical presentation of several dedicated topics such as new platform features or adjustments in platform-related processes. Subsequently, an open discussion was initiated by the platform team to learn about problems and challenges the app developers are currently facing. The first meet-up in December 2017 started with a group of 15 interested people. The number of participants grew event by event and reached a plateau of 40 to 50 people after the fourth meeting.

*"The idea of meet-ups for our app developer community emerged when we recognized the benefits of exchange between different app development teams in Answers. The initial motivation for the meet-ups was to connect people and foster this exchange." ~ Platform team member*

Parallel to the community building meet-ups, the engagement of app developers in BMW Answers grew constantly. While the forum revealed 45 daily active users in January 2018, the number increased to 120 daily active users by July 2018 and eventually to 150 daily active users by February 2019. Thereby, the forum was used for different purposes: (1) The largest percentage of threads addressed a lack of documentation. App developers requested instructions for specific platform features that were available, although there was no documentation available. The platform team usually responded to such requests by creating the required documentation in BMW Docs and posting a link to the created article as solution in the thread. In May 2018 the platform team decided to shift more efforts to documentation tasks. From that time on, the documentation of new platform features in BMW Docs was integrated into the platform's release process. (2) A second kind of posts considered platform bugs that were identified by app developers. They used BMW Answers to report bugs while the platform team notified the app developer in the same thread as soon as the bugfix was released in the platform. (3) Furthermore, app developers created threads to ask for the availability of specific platform features. The uncertainty regarding the platform's capabilities resulted in large efforts for app development teams since the feasibility of every new feature in their app required an evaluation of the platform. The platform team therefore created an "Explore" section in BMW Docs in May 2018, which contains a description of all basic platform capabilities. In January 2019, this section was enhanced by a roadmap for future platform features. In this way, app developers were not only able to estimate feasibility of features in their apps based on the current but also on future platform capabilities. (4) The fourth kind of thread in BMW Answers contained announcements of the platform team to the community. Even though also BMW Docs also contained a news section, the announcement in BMW Answers were used for gathering fast and public feedback regarding certain decisions or deliveries. For example, in August 2018 the platform team started to announce beta releases of the platform in the forum. The thread served as collecting tank for feedbacks, desires and requirements of app developers, that were discussed internally by the platform team and implemented if they were considered to be reasonable.

### 5.4.2 The Starter App

Concurrently to the establishment of the developer portal, several app teams recognized the large efforts that were required for the onboarding of new team members. During that time, not just the new developer was hindered from productive development activities but also the teaching developer was also blocked from his work.

*“You always need to sit together with new team members. You need to explain a lot. Because there isn’t any well prepared learning content. When you want to learn working with another platform, you go to YouTube and watch tutorials. There is Udemy or Udacity which provide multiple learning classes. There was nothing comparable for the BMW app platform.” ~ App developer*

In August 2017 the app development team that is responsible for the development of location-based services decided to build a so-called Starter App. The app contained three things that were required by each new app developer: First, the app entailed all basic mechanisms that are required by the platform as implementations of app lifecycle management, memory management and the configuration that is required for the deployment to a car. Second, several support scripts and tools as a basic unit testing setup were part of the package. The Starter App itself was written in TypeScript, which extends JavaScript by several mechanisms like typing and improved tool support. Third, the Starter App contained basic examples of UI-elements as well as the implementation of basic vehicle APIs such as setting of destinations in the vehicle's navigation system. When a new developer entered the team, an experienced team member explained the basic idea of the Starter App and asked him or her to implement further examples without providing further initial information. That way, the new developer experienced all of the pitfalls and specifics of the platform on his own and learned how to solve a problem from existing examples. Since the Starter App already contained a comprehensive tooling, the developer did not need to spend any efforts on tasks such as unit testing setup or static code checks. When the developer was not able to solve a specific problem, he/she gained support from other team members. After this onboarding process, the Starter App was extended by further example and the new app developer was ready to start his work on the real product.

*"We want to have a simple app that does not require to knowing the long history of BMW onboard software development but were you are able to add new features simply and fast even as beginner. This was the original vision which revealed the Starter App as onboarding tool. You can simply give it to the hands of a beginner and let him start." ~ App developer*

In January 2018 the location-based services team presented its Starter App to other app developers in a platform community meet-up. Subsequently, more and more additional teams used the starter for onboarding new team members or even as starting point for their own developmental activities. All extensions that were created within the external usage of the Starter App were reviewed by the location-based service team and subsequently merged into the app.

*"I started with the topic and after a while two or three people from our team joined me. Then more colleagues from other app teams started participating. This resulted in a lengthy cooperation. We thereby learned how to cooperate in this context and were able to sharpen the scope of the project. I think, we really proved that this can work – and this is something I am really proud of – that multiple developers from different teams and even departments inside BMW and also on the supplier side can collaborate on work with code." ~ App developer*

In March 2018 a member of the platform team extended the ignition guide in the developer portal by a tutorial on setting up a "hello world" app. The guide explained how to modify the Starter App to serve as generic base for any new app project. In the context of this work, the platform developer cooperated tightly with the location-based service team. Furthermore, the platform team decided to prescribe the utilization of the guide as mandatory for all new app projects. From this point on, the Starter App became the basis for every new app project that was started. The team subsequently developed a tool that automatically creates a basic app project, which is based on the original Starter App. This "App Wizard", which was finally released in December 2018, enables new developers to setup their own app project including comprehensive tooling and a proven basic architecture within a few minutes.

*"We recognized that we needed more standardization for apps. The complex process for setting up a new app caused a high variation inside the app's architecture. Furthermore, high efforts were required for every new app project. The App Wizard should simplify and standardize this process." – Platform team member*

In April 2018 the Starter App was deployed in the integration environment of the platform. The app was therefore now available in hundreds of cars in the companies testing fleet. The BMW platform team subsequently recognized the advantage of the Starter App as feature tour for interested stakeholders. The platform's capabilities could also be easily demonstrated to current and potential stakeholders with no technical background. From March until September 2018 more and more contributions to the Starter App were made by the platform team. These updates contained improvements to several tools as well as examples for platform features. During this period the location-based service team retained ownership of the code. In October 2018 both teams agreed that the responsibility of the starter app should be transferred to the platform team.

*"Basically, we always considered the Starter App as a task of the platform team. However, it wasn't given to them at that time. This is why I appreciate that we agreed with the platform team to transfer the Starter App to them." ~ App developer*

After a transfer period from October 2018 until January 2019, the platform team adopted the ownership of the Starter App. Since, November 2018 new platform releases have contained a new version of the Starter App that has implemented all of the new platform features of the release. The new app version was tested comprehensively and has served as gatekeeper for the new platform versions. During the implementation of new platform features, the team recognized bugs and insufficiencies in the design and was able to fix them, before the actual platform release.

### 5.4.3 The App Review

From the beginning of the platform the team installed an app review process that basically included three gates. The process was mandatory for all apps that should be deployed to a customer. In the first gate, the app developers needed to provide a rough description of their app. In a face-to-face meeting, the platform team evaluated whether the concept of the app was feasible with the given platform capabilities. If the meeting revealed that the concept was not realizable with the current platform, the app developers started a change request to the platform, adjusted their concept or declined their project. The second meeting that embodied the second gate usually took place several weeks after the first gate. Now the app developers were asked to provide a concrete technical concept including a proposal for their software architecture and backend communication concept. Furthermore, the app developers were asked in which countries their app should be available at release and whether they have already triggered the inherent processes with the respective business departments. After approval of this second gate, the actual development of the app started. At the end of 2016 some app developers deployed their first running versions on an end-to-end environment, meaning that the sources were uploaded to a repository that provided the apps as download to real cars. Here, the platform team installed a third gate, which includes the indication of certain resource consumption values by the apps as memory performance and the expected occupied space on the hard drive. The review team evaluated the provided values and urged teams with insufficient performance to improve their code. However, a rejection of a new app version was rare in the third gate. Even if the review team identified insufficiencies, it provided support for the app teams to improve their performance instead of blocking the app from a release.

*"We just saw that several apps revealed multiple different problems due to a lack of standardization. The identified issues in the review showed us that a pure rejection of deployment will not bring any efforts. We needed to support them individually." ~ Platform team member*

Moreover, the platform team decided to start detailed code reviews of every single app. However, while the third app review gate assimilated a mandatory step in the release process of a new app version, the code reviews were conducted on a sample base with a current version of the apps. It extended the existing gate process and supported apps in case of wrong or missing implementations of platform mechanisms or conceptual problems. For example, the code review team recognized that just a small number of apps implemented the lifecycle mechanisms in the way that it was intended by the platform team. This deficiency let to inflated booting durations of apps. Furthermore, multiple apps revealed insufficient test coverage in unit tests while others deployed development-specific code as part of their business logic. When the platform team recognized that such issues were not a problem of individual apps but were common practice, it decided to act. First, a comprehensive documentation on the most frequent findings were added to the developer portal. Furthermore, the reviewer scheduled meetings with app development teams to discuss the identified failures and propose potential solutions. At the end of the meeting, both sides agreed on a timeline for fixes and a repeated check according to this schedule.

*“I consider the code review to be helpful. The guys pointed us to issues we hadn’t considered so far and already provided potential solutions already. It really helped to increase the overall quality of our app.” ~ App developer*

Within the next month, the platform extended gate three several times as reaction to inadequate behavior of apps. To avoid an illegal implementation of libraries in their code, app developers needed to verify that all libraries that were implemented in their code were licensed as open source software. A check for the implemented libraries was therefore added to the third gate in April 2018.

*“We realized that app teams weren’t aware of the rules for implementing open source libraries in their code. The license checker tool should raise the attention on the one hand and enforce app teams to check their open source licenses to avoid legal problems with the implementation of unauthorized packages.” ~ Platform team member*

In March 2018, the platform team was confronted with the issue that several apps were not available in cars of the testing fleet. Even though the logging files revealed that the apps were downloaded and installed, no icon appeared in the user interface. After a few days of investigation, the team realized that the cause for the issue was rooted in new app development teams that copied code of existing apps as base for their own project. The copied code contained unique identifier of several user interface (UI) elements. When these new apps were deployed to a system where other apps already implemented the identical unique identifiers one app randomly disappeared. Besides the need for larger robustness of the platform, the team realized the need for further checks of released app versions. In parallel, the limited capacities of the review team on one side and the growing number of apps on the other hampered more detailed checks in the prevalent setup. In September 2018 the team therefore decided to build a web portal for apps that automates large parts of the review process. During the creation of the new portal, the team realized that extensibility embodies a fundamental requirement. Whenever the team identified another unintended behavior by apps that could be identified via automation, it created a new check.

*“We just realized that we are checking the same things each and every time a new version of an app was released. Central automatization enables faster and deeper analysis of the app code.” ~ Platform team member*

The portal was finally released in April 2019 and replaced the manual process that was required for passing the third gate. Moreover, large parts of the code review were included in the automated checks. From then on, an app that failed the standardized checks was automatically rejected from the third review gate.

### 5.4.4 External Platform Contributions

From the beginning of the platform development in summer 2016, the platform team worked with a development repository that was open for every interested party inside BMW. Although new parties needed to request access rights, the platform team never intended to keep the platform's code hidden. Hence, no request for access to the repository was ever denied. Besides platform artifacts such as the SDK and the runtime environment, the platform developers also stored example apps and development tools in the repository. Before the release of the Developer Portal in October 2017, the code repository incorporated a central source for documentation for app developers.

*"In the beginning the platform repository was indeed our most important source for information. All other documentation was hard to find, incomplete or challenging to understand." ~ App developer*

In July 2017, one app development team requested an API for one specific UI element that was not yet available in the SDK so far. The availability of the UI element was mandatory for the implementation of a central feature within the app. Hence, the missing API blocked the progress of the app team. The platform team however perceived extensive pressure due to general stability issues at that time. As a consequence, the team had to prioritize internal optimization issues above feature requests such as the required new UI element. When the app team realized that the platform team was not able to deliver the API in time, it decided to build it by its own. The platform team agreed to the approach and confirmed to support the app team in case of problems. After a few weeks the app team finished its work on the API and provided the code to the platform team for review. The platform team approved the basic logic, though it noted several smaller issues such as the applied coding style or documentation. After fixes of the annotated issues, the new API was finally integrated in the SDK in September 2017. Seven months after this first external contribution to the platform, another app team faced similar issues. While the app required an extension of an API, the platform team was not able to accommodate this demand. Again, the app team reconciled their actions with the platform team and implemented the required extensions on their own. After a comprehensive review by the platform team, the code was integrated into the platform. These two incidents triggered discussions regarding an active opening of the platform in the platform team. Especially the platform team's limited capacities and the growing number of apps and respective requirements appeared as challenge that could be mitigated by external contributions. However, the platform team agreed that the code ownership needed to remain on its side. In August 2018 the team decided to simplify external contributions to the platform.

*"We as platform developer appreciate external input. However, we recognize that we need clear rules for that. Otherwise, we spend too much efforts in discussions and in general communication with app developers that want to contribute to the platform." ~ Platform team member*

In addition to the benefits regarding capacity concerns the platform team hoped for new inputs and refinements of the platform and its features by the app developers. The team therefore created a new discussion thread in BMW Answer to collect input from app developers. Even before this thread, apps were able to request changes to the platform. However, these change requests required complex processes in organization. The thread should enable direct exchange between platform developers and app developers, inspire a common discussion and enhance small refinements and improvements which were too small or too insignificant for an official change request.

*“We consider our platform as product for developers. For this reason, we want to know the needs and requirements of the app developer community. Of course, we need to validate and prioritize every request, although we consider them as valuable extensions of our general platform feature planning.” ~ Platform team member*

In February 2019 the platform team released contribution guidelines for the platform SDK. The guidelines aimed at facilitating external contributions by decreasing efforts for app developers as contributors as well as the platform team as reviewers. These contained detailed descriptions on the process that has to be followed by a contributor as well as a clear definition what has to be done for a contribution. Besides the actual code, the platform team asks for working sample code of the new feature, a comprehensive documentation in the BMW Docs and inherent release notes. Each contribution is still reviewed by the platform team.

"*We just need to stay in control of the code. When an app developer who has contributed to the platform leaves BMW, we need to ensure the maintenance of this function. In this way, we prescribe rules for contributions and we need to certify all changes.” ~ Platform team member*

## 5.5 Learning Mechanisms for Platform Owners

In this study, we identify mechanisms that enable learning of a platform owner by interacting with app developers. We therefore consider the platform owner as organization in the center of a digital platform ecosystem (Tiwana, 2014). Learning in organizations is commonly defined as change in the organization’s knowledge based on the organization’s experience (Argote & Miron-Spektor, 2011). This might cause changes in behavior or cognition and effects an improvement in performance (Crossan et al., 1995). In our context, changes of cognition or behavior of the platform team cause improvements of the platform. The analysis of 25 events within two years in the context of the BMW app platform allowed us to identify three learning mechanisms: *transfer of perspective*, *transfer of knowledge* and *transfer of artifacts*. Table 7 provides an overview of the identified learning mechanisms and how they are related to the illustrated events. The following sections contain descriptions of the three learning mechanisms. We illustrate the explanations with representative events from our case study episodes. Furthermore, we embed our findings in prevalent literature and elaborate the novelty of our findings.

| Episodes | Transfer of Perspective | Transfer of Knowledge | Transfer of Artifacts |
|---|---|---|---|
| **Developer Portal** | • The platform team took the role of an unexperienced app developer for the creation of tutorials and documentation in BMW Docs<br>• Requests in BMW Answers established recognition of app developers' problems within the platform team<br>• Exchange with app developers in community meet-ups enabled platform team to understand app developers' problems from an app developer's perspective | • Reporting of bugs and requirements in BMW Answers<br>• Creation of Explore section in BMW Docs as result of frequent requests on platform capabilities in BMW Answers<br>• Creation of roadmap in BMW Docs as result of frequent requests on status of upcoming platform features in BMW Answers<br>• Surveys regarding most required new platform features in BMW Answers | • App developers wrote own documentation and submitted it in BMW Docs |
| **Starter App** | • Platform owner team recognized insufficient API design during the implementation of a new SDK feature in the Starter App<br>• Platform team recognized lack of sufficient infrastructure for developers | - | • A large amount of example code was transferred in the Starter App<br>• Multiple tools were transferred in context of the Starter App<br>• TypeScript was established as platform standard for app development with making the Starter App the default starting point for app development activities |
| **App Review** | • The platform team needed to understand the code of app developers and recognized flawed lifecycle management by the platform | • App developer reported insufficient memory measurement methodology<br>• The platform architect recognizes misinterpretation of app start concept by app developers<br>• Platform team recognized need to check of open source licenses | - |
| **External Contributions** | - | • Feature requests in BMW Answers trigger discussions in the platform team. Even though, a request is rejected by the platform team, the insufficiency of a certain aspect was communicated from app developers to the platform owner | • App developers contributed two UI-elements to the SDK |

**Table 7. Learning mechanisms of the platform team**

### 5.5.1 Transfer of Perspective

While a digital platform needs to maintain a certain degree of stability in the core, it simultaneously evolves constantly over time (de Reuver et al., 2017; Eaton et al., 2015). For instance, when the platform team recognized flawed implementations of the app's lifecycle, it adjusted the associated boundary resources (Ghazawneh & Henfridsson, 2010a). From a platform owner perspective, the extension of the platform gate process impeded flawed implementations and enhanced the platform's overall stability and performance during runtime. This procedure confirmed the common understanding that the refinement of boundary resources is not exclusively controlled by the platform owner, but is also influenced by app developers' behavior (Eaton et al., 2015). By adjusting the boundary resource, the platform owner secured control over the app developers and improved the overall quality of the platform (Henfridsson & Ghazawneh, 2013). However, since the inspection of the app's source code requires the platform team to actually understand the logic, it had to take the perspective of an app developer. In this way, the interaction of app developer and platform owner affected a further mechanism. The review induced the platform team to consider the task of implementing the app's lifecycle management from an app developer's perspective. While the implementation of the lifecycle management appeared obvious to the platform team, it recognized that this was not the case for most of the app developers. The team thus realized that the documentation that should support app developers in the implementation of lifecycle management was not sufficient. This change in cognition (Argote & Miron-Spektor, 2011) motivated the platform owner to refine the provided platform resources embodied by the documentation on app lifecycle management. Considering the platform as generic toolset that enables the development of complementary features (Tiwana, 2014), an improvement of this enablement needs to be considered as improved performance of the platform (Crossan et al., 1995).

This experience of "eat your own dog food" was even more concise in the Starter App episode. After taking over the responsibility of the app, the platform recognized insufficient platform design from an app developer perspective, every time it implemented new platform features with sample code in the app. In this way the team became part of the app developers' community of practice (Brown & Duduid, 1991). The app developer perspective enabled the platform owner to recognize and refine insufficient platform design even before the feature was officially released to the app developers. One member of the platform team even stated that this change of perspective was one of the major motivation points for assuming responsibility for the Starter App.

Both episodes exemplify learnings of the platform team that caused improvements of the platform. Figure 11 illustrates how the transfer of perspective enables learning for the platform owner. Interactions of platform owner and app developer persuade the platform owner to take the perspective of an app developer and collects experience by own app development activities. These experiences extend the knowledge of the platform owner and causes changes in his cognition. The new cognition enables the platform owner to detect insufficiencies of the platform and improve it appropriately.

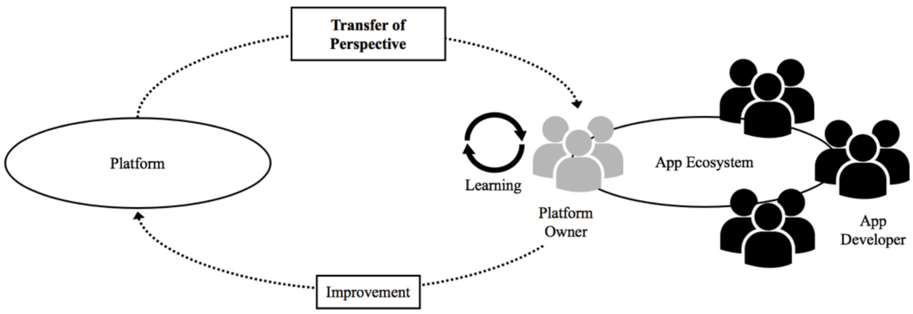



**Figure 11. Transfer of perspective learning mechanism**

### 5.5.2 Transfer of Knowledge

A digital platform is considered as medium for knowledge transfer from platform owner to app developers (Foerderer et al., 2019). Platform boundary resources such as documentation or sample code provide information to app developers and enable app development by partners outside of the platform owner (Tiwana, 2014). However, the episode on the collection of feedback on the latest SDK release in the BMW Answers proves that knowledge is also transferable from app developers to the platform owner via boundary resources. Considering the forum as resource for app developers, it enabled the emergence of a community with the knowledge-sharing routines and complementary endowments (Dyer & Singh, 1998; Mesquita et al., 2008). The mutual exchange and interaction of different app development teams as well as the platform team enabled mutual insights and an overall increase of knowledge (March, 1991). The engagement of the platform team within this community with its viral circulation of information (Levitt & March, 1988) enabled the team to gather knowledge from app developers. When the team asked the community for feedback on the latest SDK version it received spontaneous and comprehensive response. In this way, the platform team benefitted from the experience and the knowledge of the app developers. These learnings enabled the team to improve the SDK, which incorporates an enhancement of the team's performance due to learning (Crossan et al., 1995; Huang et al., 2018).

In a digital platform ecosystem, all apps are implemented on the same technical basis. This commonality enables not only scaling effects in form of value-cocreation (Lusch & Vargo, 2006) and technological benefits as standardization (Tiwana, 2014), but also the potential for scaled learning mechanisms for the platform owner. The large number of responses on the request for feedback on the SDK enabled the platform team to consider multiple perspectives and specific issues of different app developers. The public discussions in the forum effected a reflection and maturing of specific feedback by other developers. In this way the forum enabled not only pure knowledge transfer from app developers to the platform owner but also a maturing of the transferred knowledge, which decreased the likelihood of wrong learning (Crossan et al., 1995) by the platform owner.

Finally, the episode on BMW Answers also illustrates that the transfer of knowledge does not require active initiative by the platform owner. While the request for feedback was actively started by the platform team, it learned from threads that were initiated by app developers. The community used the forum for reporting bugs in the platform SDK, asking for support in cases of insufficient documentation or requesting new platform features. These hints enabled the platform team to gather further knowledge about insufficiencies of platform resources and learn about potentials for improvements without any active involvement. However, the forum enabled the team to inquire about specific information from the requester.

The transfer of knowledge learning mechanism (see Figure 12) starts with a collection of experiences of app developers. Interactions of app developers and platform owner enable a transfer of the gathered knowledge. The resulting change of the platform owner's cognition persuades the platform owner to improve the platform appropriately.

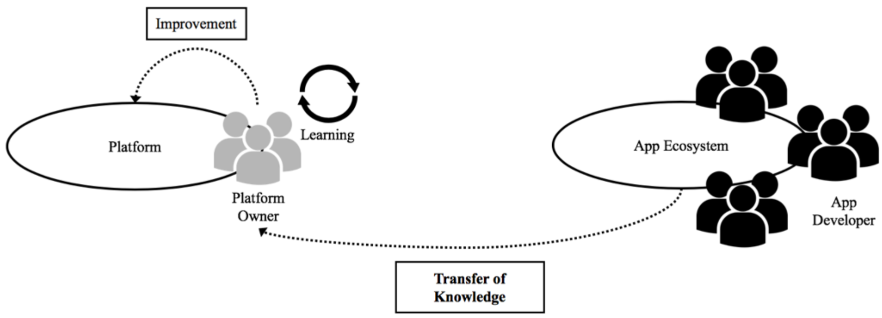



**Figure 12. Transfer of knowledge learning mechanism**

### 5.5.3 Transfer of Artifacts

A prominent characteristic of digital platforms is external value creation, that is, value creation with actors outside the platform owner's organization (Parker & Van Alstyne, 2017). A platform owner needs to foster external value creation and develop effective ways for capturing a share of that value to become successful (Schreieck, Wiesche, et al., 2017). Some platform owners absorb externally created innovation by implementing them directly into the platform core (Schreieck & Wiesche, 2017a). This coring mechanism is commonly considered as an act of competition (Bender & Gronau, 2017; Saarikko, 2016). However, our case illustrates that the transfer of a complementary artifacts from app developers to the platform core may be a result of cooperative interaction.

The Starter App was initiated as project within one app development team. It should enhance the onboarding process of new app developers. After some time, the app development team realized that the Starter App was also of use also for developers outside of the team. Hence, it decided to share the artifact and make it available for everybody in the app community. When the platform team finally realized the need for improved onboarding for app developers, the Starter App had already become common practice among new app developers. The immature condition of the platform led to non-canonical work of the app developers which again triggered potentials that could be exploited by the platform owner. Cooperation with the creators of the Starter App and the subsequent transfer of responsibility to the platform team enhanced the platform with the new resource. The platform owner again learned about potentials for improvement from interactions with app developers, though in this case the platform team integrated a solution that was already prefabricated by the app developers. This transfer of artifacts was also observable in the episode on in external contributions to the platform SDK. An externally crafted API for specific UI-elements were integrated into the platform core two times. The accessibility of the platform development repository enabled app developers to add functionality and merge their change into the platform. However, the platform team realized that rules for the transfer of artifacts from app developers to the platform were needed to avoid integration of malicious or insufficient artifacts. Hence, it decided to create contribution guidelines to define clear rules for external contributions. In addition to protection of the platform, the contribution guidelines increased the transparency to potential contributors, which again facilitated contribution activities. An app developer could inform himself/herself about requirements of a contribution before the actual submission of a contribution to the platform team. This increased the likelihood of sufficient contributions and decreased the demand for clarification meetings between app developers and the platform team.

The episode on external contributions proved the relevance of accessibility as well as transparency for external contributions to a platform (Benlian et al., 2015; Zaggl et al., 2020). However, the aspect of learnings for the platform owner had not been considered yet in the theoretical discussion on platform openness so far (Soto Setzke et al., 2019). The platform's openness enabled a transfer of artifacts that again allowed the platform owner to learn from solutions of app developers. The creators of the Starter App focused on modular design in order that new developers can consider an example of one feature completely distinct from other features. According to the team this design simplifies the understanding and accelerates the onboarding process for new developers, even though it is not recommendable for real apps. When the responsibility for the Starter App was transferred to the platform team, it consciously decided to stick with this kind of design. The team has learned that this kind of design appears optimal for new app developers. In this way the platform team learned through the transfer of the artifact (See Figure 13).

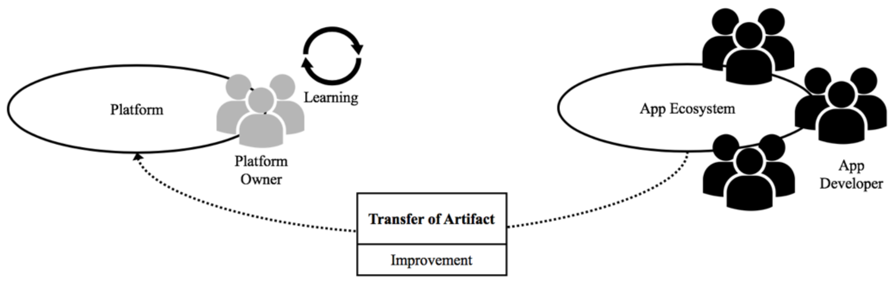



**Figure 13. Transfer of artifact learning mechanism**

## 5.6 Discussion

The emergence of digital platforms constitutes an underrepresented aspect of research on digital platforms so far. While multiple studies describe platform governance in matured digital platform ecosystems, insights on organizations that create a new digital platform is rare (de Reuver et al., 2017). The purpose of this study was to illuminate this gap by taking on the lens of organizational learning theory. Even though it is generally acknowledged that interactions cause mutual potential for organizational learning (March, 1991), prevalent literature has exclusively focused on learning mechanisms on the app developer side (Foerderer et al., 2019; Ghazawneh & Henfridsson, 2010a). By considering the learning mechanisms of a platform owner in an emerging platform phase, we complement prevalent theory. Our findings show in which way a platform owner enhances platform design by exploiting interactions with app developers through platform governance. Thereby, we confirm and extend theory on platform boundary resources and platform openness as central aspects of platform governance (Schreieck et al., 2016). Furthermore, we demonstrate the applicability of organizational learning theory for the context of digital platforms and expose coherent peculiarities. Table 3 provides an overview on our contributions to theory.

| | **Platform Boundary Resources** | **Platform Openness** | **Organizational Learning** |
|---|---|---|---|
| **Transfer of Perspective** | *Confirmation:* Interactions of platform owner and app developers trigger refinements of boundary resources (Eaton et al., 2015; Henfridsson & Ghazawneh, 2013) (Example: Recognition of insufficient infrastructure in starter app episode).<br><br>*Extension:* Interactions of platform owner and app developers trigger refinements that are not limited to boundary resources (Eaton et al., 2015; Henfridsson & Ghazawneh, 2013) but also affect the platform core (Example: Flawed lifecycle in app review episode). | *Confirmation:* Degree of openness in form of transparency is adjusted over time (Karhu et al., 2018; Ondrus et al., 2015) (Example: Community meet-ups in developer portal episode). | *Confirmation:* Experimental learning of the platform owner enabled the acquisition of implicit knowledge that was not articulated by app developers (Dess et al., 2003; Li et al., 2010; Zahra et al., 1999) (Example: Insufficient API design in starter app episode). |
| **Transfer of Knowledge** | *Confirmation:* Platform boundary resources serve as medium for knowledge exchange between the platform owner and app developers (Foerderer et al., 2019) (Example: Establishment of Q&A forum in developer portal episode).<br><br>*Extension:* Platform boundary resources enable knowledge transfer from app developers to the platform owner (Foerderer et al., 2019; Ghazawneh & Henfridsson, 2010a) (Example: Insufficient memory measurement methodology in app review episode). | *Confirmation:* Degree of openness (vertical) in form of transparency is adjusted over time (Karhu et al., 2018; Ondrus et al., 2015) (Example: Creation of roadmap in developer portal episode).<br><br>*Extension:* Adjustment of openness can be caused by learning effects on the platform owner side (Karhu et al., 2018; Ondrus et al., 2015) (Example: Creation of explore section as result of frequent requests in developer portal episode).<br><br>*Extension:* Adjustments of openness as act of platform governance is especially relevant during platform emergence to foster learning of the platform owner (Ondrus et al., 2015; Schirrmacher et al., 2017; Stummer et al., 2018) (Example: Establishment of Q&A forum in developer portal episode). | *Confirmation:* The platform owner gained knowledge, which preexisted on the app developer side and implemented it to improve the platform. This act represents acquisitive learning (Li et al., 2010) (Example: Feature requests in external contributions episode).<br><br>*Extension:* Establishing knowledge-sharing routines (Dyer & Singh, 1998; Mesquita et al., 2008) for organizational learning is relevant in digital platform ecosystems and implemented via platform boundary resources (Example: Creation of Q&A forum and in developer portal episode).<br><br>*Extension:* The platform owner needs to foster learning by incentivizing app developers (Dellermann et al., 2016; Kude et al., 2012) (Example: Participation in community meet-ups). |
| **Transfer of Artifact** | *Confirmation:* Learnings of app developers trigger tuning effects of boundary resources (Example: App | *Confirmation:* Degree of openness (horizontal) is adjusted over time (Ondrus et | *Confirmation:* Platform owner needs to steer interactions to avoid acquisition of wrong knowledge (Huber, 1991) |

| | | | |
|---|---|---|---|
| | developer submitted documentation in der developer portal episode).<br><br>*Extension:* Interaction via boundary resources is not exclusively confrontational (securing vs. resourcing) (Eaton et al., 2015; Henfridsson & Ghazawneh, 2013) but may reveal a cooperative character (Example: Starter app episode). | al., 2015) (Example: External contributions episode).<br><br>*Extension:* Learning on platform owner side is supported selective horizontal openness (Ondrus et al., 2015) (Example: Starter app episode).<br><br>*Extension:* Adjustments of openness as act of platform governance is especially relevant during platform emergence to foster learning of the platform owner (Ondrus et al., 2015; Schirrmacher et al., 2017; Stummer et al., 2018) (Example: External contributions episode). | (Example: External contributions episode).<br><br>*Extension:* Platform owner needs to foster learning by orchestrating app developers (Moellers et al., 2018) (Example: External contributions episode). |

**Table 8. Overview on contributions to theory**

### 5.6.1 Platform Boundary Resources as Medium for Organization Learning

Prevalent platform literature emphasizes the distinction of app developers and platform owner regarding power and control (Benlian et al., 2015; Ghazawneh & Henfridsson, 2010a). The distribution of power is balanced through several governance mechanisms (Schreieck et al., 2016) that are initially designed by the platform owner, influenced by app developers and evolve over time (Srinivasan & Venkatraman, 2020). Eaton et al. (2015) claim that “service systems with digital technology are ripe with political tensions among different actors trying to leverage their resources to influence others” (p. 238). While we acknowledge the significance of research on the balancing of power between app developers and platform owner, we claim that the aspect of cooperative interaction has been ignored so far. Just the platform owner is in charge to implement actual changes at the exposed resources. We argue that these adaptions are caused by learning effects at the platform owner side.

Platform boundary resources serve as medium between platform owner and app developers. The artefacts are provided by the platform owner to enable app developers to build apps (Ghazawneh & Henfridsson, 2010a). Foerderer et al. (2019) illustrate in which ways knowledge is transferred from platform owner to app developers. Our study confirms this transfer of knowledge via platform boundary resources. Furthermore, our findings extend prevalent theory by demonstrating learning effects on the platform owner side. This indicates that transfer of knowledge is not limited to transfer from platform owner to app developers but also occurs vice versa. We go further by stating that these learning effects are not limited to platform boundary resources (Eaton et al., 2015; Henfridsson & Ghazawneh, 2013) but also to other platform parts that are not urgently exposed to app developers. Interactions with app developers cause learnings that also affect the platform core or processes inside the platform owner as illustrated in the flawed lifecycle event in the app review episode.

This rationale represents an important design guideline for platform boundary resources, which should be considered by new platform owner to leverage learning effects. We claim that platform owners, who exploit interactions with app developers through learning will receive an advantage in competition with platform owners who ignore such learning effects (Henard & Szymanski, 2001; Knight, 2002).

### 5.6.2 Platform Openness Enables Organizational Learning

While the adoption of platform openness was considered as a logical consequence of environmental conditions in the digital platform ecosystem so far (Ondrus et al., 2015), we claim that a platform owner may adopt platform openness to leverage learning effects. By conscious and selective adaptions of openness, the platform owner enhances learning and leverages potential for platform improvements. Thereby, the platform owner may increase or decrease the platform's degree of openness to optimize learning effects for vertical openness as well as horizontal openness. Our research illustrates that learning occurs through communities-of-practice that involve app developers and the platform owner without any motivation regarding leveraging respective resources to influence the other side. This differentiates our findings from coring (Bender & Gronau, 2017) or absorption (Schreieck & Wiesche, 2017a) mechanisms that assume a shift of functionality from the app developer side into the platform and incorporates the leveraging of power by the platform owner. Our results indicate that both sides benefit from learnings of the platform owner, due to improvements of the platform. Since, there is no replacement of complementary products in the process of a platform owner's learning, app developers may not fear becoming replaced by platform improvements. Our findings indicate that the degree of openness is not only relevant for control over a digital platform ecosystem but also for the exploitation of learning effects. We argue that this is especially relevant for an unexperienced platform owner. Hence, we appeal that considerations on platform openness should involve the effects of learning mechanisms, especially during the phase of platform emergence.

Even though we claim the importance of learning, we remark that it is not exclusively decisive for the chosen degree of platform openness. It can also follow a long-term strategy, which is not influenced by short-term learning effects. Karhu et al. (2018) describe in which way Google started Android as open as possible to nurture the digital platform ecosystem. However, after attracting complementors and customers, Google started to decrease the degree of openness to increase the amount of control. This approach represents a long-term strategy which might just marginally be influenced by learning effects.

### 5.6.3 Organizational Learning in Digital Platform Ecosystems Requires Orchestration

One goal of our research was to apply the organizational learning lens to the context of digital platforms. Our findings extend organizational learning theory by platform dynamics. Our research confirms the occurrence of acquisitive as well as experimental learning in the context of digital platforms. However, while traditional product development organizations can enforce knowledge-sharing routines (Chuang, 2014; Maria et al., 2018; Moellers et al., 2018), digital platform ecosystems require orchestration of app developers via platform governance (Huber et al., 2017; Kude et al., 2012). Since digital platform ecosystems underly network effects (Parker & Van Alstyne, 2005), a platform owner should facilitate interactions to leverage learning. Thereby, the platform owner needs to consider the incentives for app developers to exploit learning mechanisms. Furthermore, the platform owner needs to avoid wrong learnings by designing appropriate boundary resources. This comes back to the relevance of platform boundary resources as medium for learning in digital platform ecosystems. Here, or findings again indicate the importance conscious boundary resources design to leverage learning effects, especially for unexperienced platform owners.

## 5.7 Conclusion

Our study considers a case of an emerging digital platform for automotive onboard apps. The qualitative character of our study commonly indicates limitations on the generalizability of our findings. Rather than establishing a general theory, our research strives for explorative insights that require further confirmation by additional research (Yin, 2009). Furthermore, our setup reveals certain specifics that need to be clarified to sharpen the scope of this work. First, our studies were embedded in the context of an incumbent organization that reveals legacy of its successful past in the automotive industry. Hence, our findings may differ from a similar study in the context of start-ups or digital native organizations with large experience with the development of digital services. Second, while our research is embedded in an internal setup inside BMW, we consider our results also relevant for digital platform ecosystems that grant access to third-party developers. However, our findings indicate that the required emergence of communities-of-practice demand trust of both sides (Brown & Duduid, 1991), which needs to be established by the platform owner. A comprehensive investigation of learning processes in more open platform settings appears as promising avenue further for research.

Moreover, the context of our study was dictated by the choice of the researched context, which was scoped by the availability of data. Similarly, the selection of interview partners was limited by their willingness to participate. Glaser (1992) emphasizes that the application of Grounded Theory reveals a “mid-term” theory, which is not universally applicable but needs further refinements and constant comparison through data from new contexts (Davison & Martinsons, 2016). The learning mechanisms of a platform owner in the context of an emerging digital platform is an addition to prevalent literature on platform boundary resources, platform openness and organizational learning. However, the specific context needs to be emphasized. Further research in other contexts should help to generate a more generalized theory.

The generative character of digital platforms enables incumbent companies to enrich their products with digital services and satisfy evolved customer expectations. However, a new platform owner needs to gather knowledge on the design and operation of a digital platform. By conducting a Grounded Theory approach our study sheds light on a platform owner’s learning mechanisms in the context of a digital platform for automotive onboard apps at the globally operating car manufacturer BMW.

1. The transfer of perspective describes activities in which the platform owner learns through experience from an app developer perspective and subsequently improves the platform.
2. The transfer of knowledge describes activities in which experiences of app developers are transferred to the platform owner, which causes a change of cognition and subsequent improvements of the platform.
3. Transfer of artifact describes activities in which app developer create own improvements and transfer them to the platform owner, which again causes a change in cognition of the platform owner and an improvement of the platform.

Our results contribute to the current discussion on sufficient platform design. The understanding of the identified learning mechanisms will help new platform owners to foster fruitful interactions with their app developers and enhance digital platform design.

# 6 Pure Coding Pleasure: How BMW Involves App Developers in the Design of Automotive Onboard APIs (P3)

| | |
|---|---|
| Authors | Weiß, Niklas* (niklas.weiss@tum.de)<br>Wiesche, Manuel** (wiesche@in.tum.de)<br>Krcmar, Helmut* (krcmar@tum.de)<br><br>*Technical University of Munich, Chair for Information Systems, Boltzmannstraße 3, 85748 Garching, Germany<br>** Technical University of Dortmund, Chair for Digital Transformation, Martin-Schmeißer-Weg 12, 44221 Dortmund, Germany |
| Publication | 25th Americas Conference on Information Systems (AMCIS) |
| Status | Published |
| Contribution of First Author | Introduction, Theoretical Background, Research Design, Results, Discussion |

**Table 9. Fact Sheet Publication P3**

**Abstract**

Digital transformation requires incumbent companies to accelerate the development of their digital products. The automotive industry is a prominent example of this change. Even though, research frequently considers digital transformation of organizations, there are rare perceptions on the change of the actual technology. This action research study provides deep insights in the endeavors of the global operating car manufacturer BMW towards a generic design of onboard application programming interfaces (APIs) which should enhance accelerated development of digital products. To help the firm embrace generativity, we therefore infused lead user involvement theory and API evaluation criteria into the API design team. As a result, we present 5 majorly refined APIs, which are implemented in the BMW App SDK. Further, we identified critical challenges and benefits for the involvement of lead users in the design of enabling technology within an incumbent company.

## 6.1. Introduction

### 6.1.1 Motivation

Digital transformation affects incumbent enterprises. It challenges firms from multiple domains (Schreieck & Wiesche, 2017a; Schreieck et al., 2018). The automotive industry is one of the most prominent examples of this development. The acceleration of digital product development is one major change that has to be faced by the car manufacturers (Svahn et al., 2017). Several car manufacturers have introduced systems of deploying new versions of the vehicle's software over the internet (AutomotiveWorld, 2019). The system enables regular update of the existing software and the deployment of new digital products to a car even after the car has been sold to a customer. Furthermore, the manufacturers integrate service-oriented software systems in the form of digital platforms into their vehicles to enable flexible deployment of digital products offered in the form of apps (BMW, 2018; Digitaltrends, 2017). In this context, we understand digital platforms as "extensible codebase that provides core functionality shared by apps that interoperate with it, and the interfaces through which they interoperate" (Tiwana, 2014). Thereby, after the initial hand-over to the customer, the codebase in the car can be extended. In this way, the modular extensibility of the system decouples the short-term app development process from the long-term vehicle development process. Already, in a traditional automotive development approach, different software modules interoperate via application programming interfaces (APIs). These APIs expose functionality of one software module to another. However, the utilization of an API is defined in advance. Additionally, the number of an API's consumers is limited to a small number of experts and the primary criterion for assessing an interface is its functionality. The introduction of a modular extensible software architecture that furtherly enables the deployment of unknown and new utilizations of an API in the future, entails different requirements. The capabilities and limitations of an API need to be transparent to all stakeholders including potential app developers without any domain specific expert knowledge (Pühler, 2011; Schlachtbauer et al., 2012). Hence, incumbents need to refine their existing API design. This practical challenge raises this question; how can an incumbent organization evolve its existing system design in the context of digital transformation?

Previous research reveals that multiple traditional companies has failed to transform their technology adequately (R. Henderson & K. Clark, 1990; Tripsas & Gavetti, 2000). Furthermore, literature considers the involvement of users as essential success factor for designing new information systems (IS) (Bano & Zowghi, 2014; von Hippel, 2005). Therefore, we propose the approach of lead user involvement for the refinement of an existing technology, in this case incumbent APIs for an automotive onboard software system. The evolvement of an existing system entails that this system is already in use in practice. A certain amount of lead users may have identified potential improvements for existing solutions (Lüthje & Herstatt, 2004; von Hippel, 1986). In the case of automotive onboard APIs, app developers that already implemented such APIs embody that kind of lead users. This study strives to identify the challenges and benefits that emerge in the involvement of lead users during the analysis and refinement of automotive onboard APIs. For doing so, we applied an action research approach (Baskerville, 1999; Frank et al., 1998; Keng & Rossi, 2011) within the software development department at the global car manufacturer BMW.

In the remainder of the paper, first, we described the theoretical concepts of APIs and lead user involvement. Then, we described BMW's infotainment system, followed by a detailed description of the research project setup. Subsequently, we described the detailed API analysis and design process. Finally, we presented our results, findings, and discussions.

## 6.2 Background

### 6.2.1 Relevance of API Design

Application Programming Interfaces (APIs) exposes a system's core resources as a service to stimulate generativity (Henfridsson & Ghazawneh, 2013). The usage of an APIs does not have to be determined by design but can be utilized in multiple ways. In this way, APIs are able foster innovation by enabling the development of complementary features (Um et al., 2013). Similarly, the utilization by app developers affect the design of the API (Eaton et al., 2015). APIs potentially generic character enables scalability of operations as well as flexibility in acquiring new strategic partners and realizing new business goals (Iyer & Subramaniam, 2015). However, the actual design of APIs is critical to maximizing its potentials. Poor API design results in increased development costs during its implementation by apps (Henning, 2009). If these costs exceed potential benefits of a complementary feature, it will not be created. In this way innovation is blocked and the attractiveness for end users remains constant or even decreases (Tiwana, 2014).

#### 6.2.2 Lead User Involvement

Lead user involvement is a principle often used in system design research. To reduce the risk of failure, the alignment of product development activities with the needs of actual and potential users is crucial (Jaworski & Ajay, 1993). A user-centric focus fosters quality, reliability and uniqueness of a product (Li & Calantone, 1998). The involvement of users already in early phases of an innovation project enhances these potentials (Herstatt & von Hippel, 1992; von Hippel et al., 1999). In their comprehensive literature review, Bano and Zowghi (2014)) identify five relevant perspective of user involvement. The *psychological* perspective considers aspects as the users' motivation or interests to participate. Second, the involvement of users requires appropriate *management*. Moreover, the *political* perspective considers the degree of power that is given to the involved users. The purpose of user involvement can differ for various groups of users. Therefore, *cultural* aspects need to be considered. Finally, different intensities of user involvement require specific *methodological* approaches. The concept of lead user involvement originally is rooted in marketing research, and it considers the involvement of users whose present strong needs will become general in the future (von Hippel, 1986). Lead users are well-qualified and motivated to contribute to an improvement of the status quo (von Hippel, 1986). Their prevalent own need enables them to innovate (von Hippel, 2005). Since lead users embody the leading edge of a market regarding important market trends; their participation in product development activities facilitates innovation, attractive for future users. In our study, we consider lead users as app developers who implemented the exposed interfaces at a large scale or multiple times. Lead user involvement is mainly considered for enhancing innovation. However, von Hippel (2005)) emphasizes that many of the concepts regarding innovations communities "apply to information communities as well." Considering the app developer community as an information community which utilizes APIs, we strive to involve their expertise in the refinement of already existing APIs.

### 6.3 The BMW Case

#### 6.3.1 Initial Situation

This study considers APIs of a digital platform for onboard automotive apps of a global car manufacturing company, BMW. The platform is part of the BMW OS 7.0, the company's latest infotainment system, released in July 2018 with the release of the latest BMW X5 series. Upcoming models from the manufacturer will run the system. The car's central electronic control unit powers the digital platform, and it also enables modular wireless deployment of apps to the digital platform. By exposing multiple functionalities of the car via APIs, the platform provides the base for a broad spectrum of use cases. During the release of the platform in summer 2018, more than 20 apps were available, providing services such as a parking lot finder[16], music streaming[17], Microsoft Office 365[18] and apps for different BMW service calls[19]. The number of available apps is steadily increasing since the initial release. Although the platform was not opened towards third-party developers, multiple stakeholders around the globe are involved in creating apps for the platform. The app developer community is made up of over 120 active members.

The first author of this study is actively involved in the platform's development team as a Ph.D. researcher. In the course of this study, we interviewed eight expert app developers from October until December 2017, and the goal of the interview was to identify challenges at the emergence of the platform. The result of the investigation showed that the platform's APIs has room for significant improvement. The developers pointed out two fundamental issues with the platform's APIs. First, the design of the API was tailored for a single or small amount of use cases.

*"Sometimes it is hard to understand how and why an interface is designed in the way it is. For example, one specific value – which is actual available somewhere in the car - is just missing in the interface while in other cases one value is available in three different ways. However, most probably the missing value was simply not required by the initial use case while in the other case there were three different use cases, which required the value in three different ways."*

Second, the design of the interface was decided by the API provider and the party requesting for the use case. The interest of requesting party is speedy delivery with a minimal budget. The evaluation criteria for the API is the feasibility of the use case while usability is subsidiary. However, abstracting usability was not considered due to the small number of consumers.

*"The implementation of some APIs requires massive efforts. You need to write tons of boilerplate code which don't reveal any functionality towards the application. For example, the implementation of a notification banner in the UI requires more than 350 lines of code."*

### 6.3.2 Project Setup

The platform team reviewed the result of the interview conducted and decided to start a project to create new and high-quality APIs in the platform SDK. The project was conducted in research cooperation, applying action research methodology. This approach allowed the active involvement of the researcher directly in the project and enables deep insights into the actual design process. Hence, the first author of this study took the role of an active architect who was responsible for designing and refining APIs in an iterative approach while an internal BMW engineer was the project lead. This setup embodied the client-system infrastructure, which is required by any action research approach (Baskerville & Wood-Harper, 1998). The project started on March 15, 2018 and ended December 21, 2018.

Besides the establishment of a robust client-system infrastructure, the principles of action research require a sound theoretical foundation of the applied approach (Davison, et al. 2004). Even though, the conducted interviews revealed the demand for action, a more profound

---

[16] https://www.bmw-connecteddrive.de/app/index.html#/portal/store/Base_Connected_Parking retrieved 19.02.2019

[17] https://www.bmw-connecteddrive.de/app/index.html#/portal/store/Base_Connected_Music retrieved 19.02.2019

[18] https://www.bmw-connecteddrive.de/app/index.html#/portal/store/Base_ExchangeOffer retrieved 19.02.2019

[19] https://www.bmw-connecteddrive.de/app/index.html#/portal/store/Base_AssistNoTPEGOffer retrieved 19.02.2019

approach was required for the actual evaluation of the APIs. For this reason, we derived general evaluation criteria for good API design from prevalent literature. Therefore, we conducted a comprehensive review of literature on API design and coded all identified papers. The analysis revealed a large number of different characteristics of good API design. However, four criteria stand out as being named in most of the considered studies: Simplicity, Documentation, Usability, and Tutorial/Sample Code (see Table 1). Hence, our study does not claim to consider all relevant API characteristics but the most relevant. The identified criteria are not exclusive to each other, nevertheless each aspect has unique characteristics which are explained in the *Description* column in Table 10.

| Criterion | Source | Evaluation Grades | Description |
|---|---|---|---|
| Simplicity | Bhaskar et al. (2016)), (Bloch, 2006), (Myers et al., 2016) | Low | It is hard to create basic objects and even to trigger basic operations. There is much unclear overhead that makes no sense from a functional perspective. |
| | | Medium | The usage is generally possible. However, there is still some overhead and tacit knowledge needed. |
| | | High | The API can be used as it is. With some basic knowledge in software development, it is easy to trigger basic functionalities, and the API behaves as "expected." |
| Documentation | (Bhaskar et al., 2016), (Burns et al., 2012), (Lee et al., 2014) | Low | There is no documentation at all, or there is only some documentation that generates no valuable insights for the developer. |
| | | Medium | Some or a big part of the functions are documented. However, use-case oriented usage is unclear. For example, a developer does not know the order of functions calls. |
| | | High | The API is thoroughly documented (each function and property) and contains necessary additional information (e.g., flow charts or sample usages). |
| Usability | (Bhaskar et al., 2016), (Bloch, 2006), (Zghidi et al., 2017) | Low | The naming and usage of types are inconsistent. The API does not conform to the coding guidelines of the respective programming language or environment. |
| | | Medium | There are some inconsistency issues, but generally, the API is consistent and fulfills the usability compliance. |
| | | High | There are no inconsistencies, and the API conform to usability compliance. |
| Tutorial/ Sample Code | (Bhaskar et al., 2016), (Burns et al., 2012), (Zghidi et al., 2017) | Low | There is no tutorial / sample code |
| | | Medium | There is some sample code or tutorials for selected parts. |
| | | High | There are sample code and tutorials available for the API. That means that every complex usage of the API is explained with an example for a better understanding. |

**Table 10. API Evaluation Criteria**

### 6.3.3 The API Design Process

The following section describes the applied API design process which is based on an iterative action research approach (Baskerville & Wood-Harper, 1998). It contains four steps: API Diagnosing, API Action Planning, API Action Taking and API Evaluation. For comprehensibility reasons, we illustrated each step by the representative example of the `startGuidance()` method inside the Navigation API, whose design was refined in the API design process. The method enables an application to change the currently set destination in the vehicles navigation system to given GPS coordinates and activate the guidance.

```
import { MiddlewareModule} from 'sdk/src/system/middlewareModule';
import { RouteIdentifier} from 'sdk/src/navigation/interface/routeIdentifier';
import { DestinationsUnion} from 'sdk/src/navigation/interface/destinationsUnion';
import { NavigationApiProviderModule} from 'sdk/src/navigation/interface/navigationApiProviderModule';

private middlewareModule: MiddlewareModule;
private routeIdentifier: RouteIdentifier;
private destinationsUnion: DestinationsUnion;
private navigationApiProviderModule: NavigationApiProviderModule;

// Actual call which is triggered by tap on an ui-button

button.on('tap', startGuidanceToGeoCoords([48.178326, 11.556802]));

// Helper function to start guidance to a given list of GeoCoords

private startGuidanceToGeoCoords(geoCoords: GeoCoord[]): void {
   const destinations = geoCoords.map(
     (coord: GeoCoord) => new Destination(coord, new FormattedAddressString('', '', ''), ''),
   );
   this.startGuidance(destinations, DestinationValueType.COORDINATE);
}

// Helper function to start guidance to a given list of destinations

startGuidance (destinations: Destination[], type: String) {
  this.middlewareModule
    .getProxy(navigationApiProviderModule, Navigation.getDestinationProxy())
    .then(proxy => {
      const routeIdentifier = Navigation.NaviCommonTypes.getRouteIdentifier();
      const destinationsUnion = destinations.map(dest => StartGuidanceExample.buildDestinationValueUnion(dest, type));
      this.logger.info(`registering for destinationsHandover`);
      return proxy
        .subscribe({
          name: 'destinationsHandover',
          appId: this.app.getAppId(),
          onReceive: (event: any) => {
            this.logger.info(`destinationsHandover event received: ${event}`);
           },
           onError: (error: any) => {
            this.logger.info(`destinationsHandover error received: ${error}`);
           },
          })
          .then((subscriptionId: string) => {
           this.logger.info(`Successfully subscribed to destinationsHandover (subscriptionId ${subscriptionId})`);
           this.logger.info(`Starting guidance`);
           proxy
           .callMethod('startGuidance', {
             appId: this.onlineApp.getAppId(),
             importRouteIdentifier: new routeIdentifier({ sessionID: 0, routeHandle: 0 }),
             destinations: destinationsUnion as any[],
             handoverType: HandoverType.NON_SILENT,
           })
           .then((startGuidanceResult: any) => {
             this.logger.info(`Start guidance result: ${JSON.stringify(startGuidanceResult)}`);
           })
           .catch((error: Error) => {
             this.logger.info(`Error Ocurred while starting guidance ${error}`);
           });
        })
        .catch((error: Error) => {
          this.logger.info(`Error ocurred while subscribing to destinationsHandover broadcast ${error}`);
        });
      })
    .catch(error => {
      this.logger.info(`Start guidance got an error: ${error}`);
    });
}
```

**Image 2. Original Implementation of `startGuidance()` Method**

#### 6.3.3.1 API Diagnosis

During the diagnosis stage, we identified the primary problems that are anchored in the organization's major drive for introducing change. It is required that the researcher identify these problems in a complex organizational structure (Baskerville 1999). In the context of this research, this step required the analysis of the prevalent APIs. To gain deep insights about a specific API, we contacted all the relevant stakeholders. Being part of the project team granted the researchers' access to different experts inside the organization. Overall, 21 app developers were involved in the diagnosis of the APIs. For the navigation API, we enlisted experts from the platform architect team, a software engineer from the navigation module and two app developers frequently using the navigation API. Using a code that implements the navigation functionality in an app, we assessed the API based on API evaluation criteria described in the previous section, and the assessment was via open interviews. The result of the evaluation showed that the navigation API's **simplicity** was low. The experts reported a high internal complexity in the process of creating basic objects and triggering basic features like starting a guidance. In the case of the `startGuidance()` method the implementation of the interface required more than fifty lines of code which was commonly considered as inappropriate by the experts. The **documentation** of the API was rated medium to low. To start the guidance, the instantiation of multiple objects as well as methods calls were necessary. Neither a list of all the required objects and method calls nor the precise sequence of these actions was listed in the official documentation. The API's **usability** was also considered as low. The experts reported inconsistent naming and typing patterns. An inspection of the raw code by an app developer confirmed this assessment. Finally, because there were no tutorial or sample code of the API implementation, on **tutorial** or **sample code** the API was rated low.

#### 6.3.3.2 API Action Planning

Action planning is the collaborative step of the researcher and the practitioners to define next steps to relieve organizational pain or improve the existing situation. In the context of this project, the actual scope of the refined APIs needs to be clarified. Even though large parts of the available APIs were analyzed in the diagnosis phase, the scope of the actual refinements needed to be limited due to the finite capacity of the team. Action research implies the principle of change through action (Baskerville 1999). Following the paradigm, the team decided that the most change for app developers inside BMW could be achieved by covering the most commonly implemented interfaces. Therefore, 26 app development teams were asked for their prioritization. Based on these considerations, the team decided to work on the following APIs: Navigation, User Interface, Vehicle Data, Phone and Speech. Furthermore, a rough estimation of the required time for each APIs' redesign was made. Finally, the order in which the refined APIs should be drafted, programmed, and released was defined. The release of the `startGuidance()` method inside the Navigation API was determined for mid of October 2018. We announced the resulting timeline for releases of the new APIs to the app developer community.

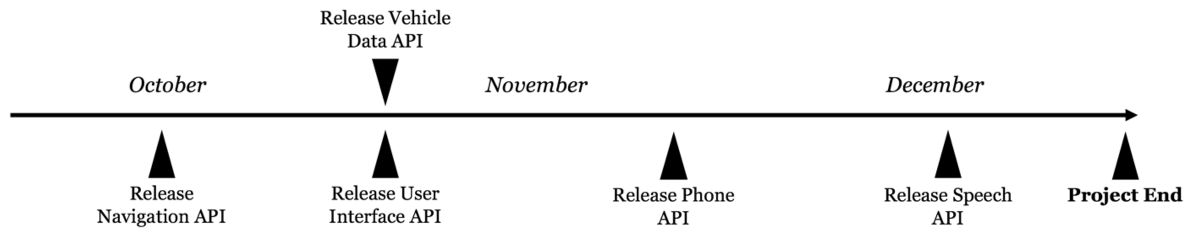



**Figure 14. Announced Timeline for Releaeses of Refined APIs**

#### 6.3.3.4 API Action Taking

Based on the gathered insights in the API diagnosis, a draft for new APIs was created in the action taking phase. Besides the results of the evaluation criteria, we also took into consideration the underlying architecture. The design creation followed an iterative process in which the researchers exposed their design to the experts, collected feedback and refined the API until the design was sufficient for all involved parties. The API action taking involved the same 21 app developers as the initial API diagnosis. For instance inside the `startGuidance()` method, the destination could not exclusively be provided as GPS coordinates but also as address or as free text. One discussion was raised if the type of the destination should be contained in the method's parameters `startGuidance` (destinationType: “GeoCoords”, destination: [48.178325, 11.556802]) or its name startGuidanceToGeoCoords (destination: [48.178325, 11.556802]). The initial proposal was a design that contained this information as property of the method. However, the app developer prompted a shift of the information into the method’s name. The experts argued that the text-completion feature of common code editors will propose all available options (startGuidanceToGeoCoords, startGuidanceToAdress, startGuidanceToFreeText) when the developer starts typing the methods name, while it will not propose all options for the method’s properties at this point. The availability of this text completion feature increases the usability of the API. Hence, the API design was refined. The creation of the complete navigation API design went through three iterations until the experts made no annotations. Furthermore, the implementation of the original interface required multiple lines of so-called boilerplate code (code that does not contain any relevant functionality for app developers but is required in each implementation of the interface). By abstracting this code into the SDK, the required lines of code for an implementation of the startGuidance() method could be reduced from more than fifty lines of code (Image 2) down to three (Image 3). The increased simplicity of the method reduces the efforts for app developers to understand the respective interface and decreases the likelihood of mistakes in the implementation. The design of the actual API in the code affected the evaluation criteria of simplicity and usability. However, also the lack of documentation needed to be fixed. For this reason, as soon as the actual design of the code was finalized a comprehensive documentation was added inline. Further, the team created tutorials for each API which were provided in a web portal for app developers inside BMW. These instructions were based on an implementation of the refined API in a reference app which was also provided to the app developers. In this way the functional principle of the API could be comprehended by the app developers. Moreover, the functionality of the new APIs was validated from an app developer perspective inside the API team, even before it was officially released.

```
import { Guidance} from 'sdk/src/navigation/interface/guidance';

private guidance: Guidance;

button.on('tap', Guidance.startGuidanceToGeoCoords([48.178326, 11.556802]));
```

**Image 3. Refined Implementation of startGuidance() Method**

#### 6.3.3.5 API Evaluation

About three weeks before the release of the API; the team released a beta version of a new SDK, including the refined APIs. Additionally, a thread in an internal app developer forum was started, asking for feedback from the app developer community. Since the release was tagged as beta, the team was allowed to identify the further need for refinements and implement them in the API. Overall, 12 app developers provided feedback via this channel. Based on the response of the app developer community, the team releases three beta versions until a stable version of the SDK was released on the announced date. For the introduction of the navigation API, three beta versions of the SDK were provided to the app developer community until its final version was released. The evaluation of the community increased the usability of the API on the level of methods and properties. App developers pointed out inconsistency in naming patterns and missing properties in specific methods.

## 6.4 Results

The application of action research strives for the achievement of two goals. First, the researcher tries to introduce real change within the organization (Babüroglu & Ravn, 1992). The first part of this section provides an overview of all refined APIs that were created through which change was established in the context of this project. Second, to expand the scientific body of knowledge (Baskerville, 1999). The remaining part of this section describes the observed forms of lead user involvement in the API design process. These results serve as the base for the subsequent discussions on lead user involvement in the design of enabling technology.

### 6.4.1 Created Artefacts

The initial goal of this research project was the analysis and refinement of poorly designed APIs that were exposed to developers of onboard apps inside the car's head-unit. A diagnosis of all prevalent API modules embodied the base for our further proceeding. Considering the relevance of APIs to app developers, we decided to focus on the refinement of the five most used APIs. First, the navigation module exposes the functionality of the onboard navigation system to the apps. Furthermore, the vehicle data API provides information on the current car status as the current vehicle speed or fuel status. The UI and speech APIs provide interfaces that allow the customer to interact with the app via a graphical user interface or voice respectively. Finally, the Phone API provides access to the functionality of a smartphone that is connected to the car's head-unit via Bluetooth. In all, the refined APIs contain 59 methods and 31 properties that were implemented by the app developers. The evaluation of the original (Orig.) as well as the refined (Ref.) API design reveals a clear improvement regarding all defined evaluation criteria through the API design process (Table 11).

| API | #Involved App Developers | #Refined Methods | #Refined Properties | Simplicity (Orig./Ref.) | Documentation (Orig./Ref.) | Usability (Orig./Ref.) | Sample Code (Orig./Ref.) |
|---|---|---|---|---|---|---|---|
| Navigation | 4 | 21 | 6 | Low/High | Medium/High | Low/High | Low/High |
| Vehicle Data | 5 | 2 | 21 | Medium/High | Medium/High | High/High | Low/High |
| UI | 5 | 19 | 2 | Low/High | Medium/High | Low/High | Low/High |
| Phone | 4 | 3 | 1 | Medium/High | Medium/High | Low/High | Low/High |
| Speech | 3 | 14 | 1 | Low/High | Medium/High | High/High | Low/High |

**Table 11. Overview on Refined APIs**

Action research strives for the introduction of change. However, the pure creation of a refined API design does not prove any change within the organization. Only if the created artifact is applied in practice, can it produce an effect (Babüroglu & Ravn, 1992). The implementation of the APIs in the BMW onboard platform SDK should satisfy this requirement. The APIs were released in the platform SDK according to the timeline (Figure 14). All currently developed apps implemented the refined APIs until February 2019. These apps will be to available to BMW customers shortly.

### 6.4.2 Theoretical Learnings

The deep insights we gained through the application of action research allowed us to identify the challenges and benefits of lead user involvement in each step of our applied API design process. Thereby, we classify our findings into psychological, managerial, methodological, cultural and political perspectives (Bano & Zowghi, 2014).

By analyzing the implemented APIs per app, we identified the heaviest users of an API within the app developer community. However, even though all contacted app developers replied positively on our request, just a fraction of them was able to participate in our project. Especially employers of external suppliers were impeded by limited time capacity, since their assignment didn't include such additional activities (*managerial challenge*). Further, the app project leaders were not motivated to shift capacity from their project towards the API design project. High pressure on the app development teams forced them to focus all their efforts on the app development itself (*Psychological challenge*). Finally, the evaluation of the original APIs itself, revealed a high complexity within the system (*managerial challenge*). However, the iterative approach as well as the involvement of the API provider module engineers enabled a comprehensive understanding of all involved parties. The conflation of the API consumer and API provider in common meetings simplified the communication (*managerial benefit*), facilitated knowledge sharing (*cultural benefit*) and created mutual comprehension of the challenges and difficulties of the respective other side (*psychological benefit*). This again motivated all participants to improve the status quo.

The API planning step comprised the prioritization of the refined APIs. Even though the most implemented APIs were refined first, this approach endangers the loss of engaged lead users, since their more specific API wasn't part of the project any longer (*managerial challenge*). Further, the announcement of the API refinements raised expectations from management as well as the app developer community, which needed to be handled by the team (*psychological challenge*). On the other hand, the involvement of the lead users simplified and enhanced the prioritization process (*methodological benefit*). Thus, there were APIs that were implemented by a large number of apps. However, their status quo was more sufficient than the design of other APIs that were implemented by a slightly smaller number of apps. In this way, the involvement of lead users enabled a better understanding of user requirements. Further, the involvement of app developers enabled a better understanding of the required refinements measures and the support the establishment of realistic expectations towards the project within and outside the team (*managerial benefit*).

Next, the API action taking step considered the actual creation of refined API designs. The involvement of multiple lead users enabled the definition of a generic API design that satisfies not only specific but a broad spectrum of use-cases. However, the number of iterations needed for a commitment of all involved parties regarding the created design couldn't be estimated in forward. Hence, the limited amount of time available embody a challenge for the team (*managerial challenge*). Further, it turned out that different app developers had conflicting requirements on specific API methods. Even though, these conflicts could be immediately resolved by the API design team, they embody a challenge (*political challenge*). On the other hand, these conflict resolutions increased the commitment on the refined solution and increased the motivation of all involved parties. All involved parties agreed that these kinds of discussions increase the quality of the final API design (*methodological benefit*). Further, the commitment of all parties increased the likelihood that the final solution would be accepted by a majority of the app developer community (*psychological benefit*).

In the final API evaluation step, the beta versions of the APIs were published to the community. The involvement of further users of the API should increase the maturity of the final API release. However, this required a material amount of app developers to implement and evaluate the APIs. This increased the amount of required time for the overall process (*managerial challenge*). Though, the involvement of even more app developers furtherly increased the likelihood for the identification of insufficiencies or flaws in the final design (*methodological benefit*). Further, the app developer community felt involved and fetched up for the upcoming changes. In this way the overall acceptance for the project could be increased (*psychological benefit*).

| Process Step | Challenges | Benefits |
|---|---|---|
| API Diagnosis | • Time constraints (managerial)<br>• Lack of motivation (psychological)<br>• System complexity (managerial) | • Simplified communication (managerial)<br>• Facilitated knowledge sharing (cultural)<br>• Increased motivation (psychological) |
| API Action Planning | • Time constraints (managerial)<br>• Users and Managers expectations (psychological) | • Better understanding of requirements (methodological)<br>• Development of realistic expectations (managerial) |
| API Action Taking | • Time constraints (managerial)<br>• Conflicts (political) | • Increased quality of final design (methodological)<br>• User acceptance (psychological) |
| API Evaluation | • Time constraints (managerial) | • Increased quality of final design (methodological)<br>• User acceptance (psychological) |

**Table 12. Challenges and Benefits of Lead User Involvement in the API Design Process**

Our analysis reveal that especially managerial challenges affected the API design process. This approves Svahn et al. (2017)) observations from another automotive case at Volvo: The management needs to conceive the need for a shift towards developer-centric software design. Otherwise, real change is hard to achieve in an incumbent context. However, while the Volvo case remains in the observation of these managerial phenomena, our study proposes user involvement as potential approach to address this challenge bottom-up. The interplay of multiple, heterogenous developers is able to achieve real change. Our results prove that this is not just true for matured app platforms (Eaton et al., 2015) but also in the context of a just emerging digital platform in an incumbent context.

## 6.5 Summary and Outlook

In this study we achieved a valid involvement of lead users in the evolvement of an incumbent service system embodied by automotive onboard APIs. The participatory approach for the API design enabled the creation of APIs that does not require domain-specific knowledge to implement (Pühler, 2011; Schlachtbauer et al., 2012). The identified benefits prove the relevance of user involvement for refining existing technology in an incumbent context.

However, BMW as organization is optimized towards the development of traditional products. In this setting, the customer is usually considered as the only user of the developed product. The usability of a feature that is not visible for him is not considered as dispensable. However, APIs are not directly used by the driver of the car, but the developer who is building complementary apps. The identified challenges in the shift towards app-developer centric APIs prove this fact. The elimination of these requires reconfiguration within the organization and its mindset (Svahn et al., 2015). Furthermore, we approve action research as valid method for gathering deep insights in a real organization, its technologies and its processes. The approach enabled a comprehensive understanding of the app developer's initial needs and their positive evaluation of the initiated change. However, these findings reveal a short-range character. The long-term effects on the actual app development activities remain in the dark. Therefore, an investigation on the effects of the new APIs on the app development appears as promising extension of this study. Does the shift to user-centric API design actually foster the development of apps? This question could be addressed by a quantitative analysis of the utilization of the created APIs as well as by further qualitive research.

# 7 Cars as Digital Infrastructure: An Analysis of Platform Ecosystem Options in the Automotive Industry (P4)

| | |
|---|---|
| Authors | Weiß, Niklas* (niklas.weiss@tum.de)<br>Schreieck, Maximilian* (maximilian.schreieck@in.tum.de)<br>Wiesche, Manuel** (wiesche@in.tum.de)<br>Krcmar, Helmut* (krcmar@tum.de)<br><br>*Technical University of Munich, Chair for Information Systems, Boltzmannstraße 3, 85748 Garching, Germany<br>** Technical University of Dortmund, Chair for Digital Transformation, Martin-Schmeißer-Weg 12, 44221 Dortmund, Germany |
| Publication | 6th Innovation in information infrastructures (III) workshop |
| Status | Accepted |
| Contribution of First Author | Introduction, Theoretical Background, Research Design, Results, Discussion |

**Table 13. Fact Sheet Publication P4**

## 7.1 Introduction

Incumbent industries need to adopt platform business models to avoid disruption by foreign players (Parker, Van Alstyne, & Choudary, 2016). The automotive domain is a prominent example for this phenomenon (Svahn et al., 2017). Car manufacturers implement digital technologies in their products to enable the implementation of digital services; examples cover BMW ConnectedDrive, MercedesMe, or AudiConnect.

The widespread distribution of digital technologies into millions of cars around the globe has led to the emergence of a new digital infrastructure with large potential for innovation. However, while a digital infrastructure comprises the pure computing and networking resources and is not owned by anybody (Constantinides et al., 2018) it requires digital platforms and inherent ecosystems on top to facilitate innovation as well as value creation (Gawer, 2014). As opposed to established digital infrastructures as smartphones or computers, the emergence of an app ecosystems for cars is just in the beginning. Car manufacturers need to develop strategies that enable them to create and capture value through digital platforms in their cars to avoid being disrupted by new entrants (Parker, Van Alstyne, & Choudary, 2016).

This study strives for answering the question which strategical options on platform ecosystems occur for incumbent car manufacturers that are confronted the transformation of their products to a digital infrastructure. Therefore, we conduct a case study at a globally operating premium car manufacturer. The collection of data from 42 expert interviews within the period of two years, allowed us to identify three platform ecosystem options: (1) build a proprietary platform ecosystem, (2) build a collaborative platform ecosystem, or (3) join an existing platform ecosystem. Our findings contribute to literature on digital infrastructures as well as theory on platform launch and competitive strategies in platform ecosystem.

## 7.2 Background

Traditionally, a car is a purely physical product. In cohesion with streets, traffic regulations and fuel supply, it plays a central role in an infrastructure that provides transportation for persons and goods. However, the integration of computing and networking capabilities generates a new digital infrastructure (Constantinides et al., 2018). While in traditional infrastructures technology and provided services are tightly coupled – such as cars and transportation - digital infrastructures enable generativity through flexibility, scalability, its recursive nature and the varying substance of data as transported material (Tilson et al., 2010). This means that cars were originally designed for fulfilling one single task, namely the transportation of persons or goods. Subsequently, all services that build on cars as infrastructure relied on the principle that someone or something is transported from A to B, like taxi offers, logistics or food delivery. Even motorsport racing is based on the competition of drivers that try to cover a certain distance from start to finish faster than competitors. However, the distribution of digital technology into cars enables the design of new services that do not implicitly require a relation to transportation. Beside mobility-related services as parking[20] or car sharing[21], car manufacturers started to provide digital services as music streaming[22] or productivity apps[23] to enhance their products.

Even though, these digital services are built on computing and networking capabilities of the car, their development is traditionally conducted in linear value chains. A supplier gets paid for the development of software, which gets integrated by the car manufacturer and eventually sold as feature to the customer. However, a platform ecosystem embodies a network that outperforms such traditional pipelines (Parker, Van Alstyne, & Choudary, 2016). It enables the value creation through third parties without any direct involvement of the platform owner and enables massive scaling effects (Parker et al., 2017). Furthermore, platform ecosystems reveal beneficial network effects that increase with the size of demand side. The more potential customers can be reached with a single app, the more attractive is an ecosystem for app developers and the more attractive it is again for potential customers. In contrast, multihoming of apps in several ecosystems requires larger efforts for app developers (Cennamo et al., 2018). Hence, an ecosystem that enables the development and distribution of one app to cars of multiple brands unleashes massively larger network effects then the manufacturers' currently proprietary systems.

The emergence of a new digital infrastructure without any relevant platform ecosystems attracts experienced players of the platform game to enter the field. Google just announced the release of Android Automotive OS (Google, 2019), which should replace the manufacturer's prevalent proprietary infotainment systems. The convergence to a layered architecture (Yoo et al., 2010) and the coherent decoupling of hardware and software in modern cars allowed the development of an automotive app platform by Google without producing any vehicle. The envelopment of

[20] https://parknowgroup.com retrieved June 3, 2019

[21] https://www.drive-now.com retrieved June 3, 2019

[22] https://www.bmw-connecteddrive.de/app/index.html#/portal/store/Base_Connected_Music retrieved June 3, 2019

[23] https://www.bmw-connecteddrive.de/app/index.html#/portal/store/Base_ExchangeOffer retrieved June 3, 2019

existing systems (Eisenmann et al., 2011) and the consequent large and cross-brand customer base in combination with the unexploited potential for innovation (Henfridsson et al., 2018) of apps in cars enables the nurturing of a vibrant app ecosystem. Now, car manufacturers are confronted with the question if they should join Google's ecosystem or strike out on their own. This leads to a more general question: Which platform ecosystems options emerge for car manufacturers as incumbent firms that are confronted with the transformation of their products to an unexploited digital infrastructure?

To answer this question, we collected 42 expert interviews in the environment of a globally operating car manufacturer who is confronted with the transformation of its products to a digital infrastructure. Our analysis revealed three major options which are illustrated in the remainder of this paper.

### 7.3 Build proprietary platform ecosystem

The traditionally strong competition in the automotive industry motivated manufacturers in the past to build proprietary infotainment systems that differentiated their vehicles from competitional products. Each system reveals specific graphical styles and features that should raise the attention of customers and motive the purchase of a car.

*„We live in a world which is massively fragmented for third-party developers. Mercedes hast its own platform, BMW has its own platform and any other manufacturer has its own platform.“*

To benefit from scaling effects, manufacturers started to design their systems according to a layered platform architecture that allows the modular extension of their software in the car. With Ford[24] and GM[25], two established manufacturers already decided to open their platforms to third-party developers. Our findings also indicate that the opening of an existing system embodies a valid option to endeavor the creation of an automotive platform ecosystem. However, the experts agreed on the massive efforts that are required for launching an own platform ecosystem, which need to be justified by respective benefits.

### 7.4 Build collaborative platform ecosystem

Even though, Ford and GM already decided to open their platforms to third-party developers, the small number of provided apps in their respective app stores indicate that the companies failed in nurturing a vibrant ecosystem. According to our findings, the small number of target devices could be a major reason for this failure. In the mobile world, an app developer needs to build and maintain two versions of an app, to address hundreds of millions of Android and iOS users. In contrast, one single car manufacturer ships a few million vehicles a year – a negligible number of potential customers for the profitable development of a distinctive app.

*"You simply need to increase the user base. Otherwise larger app providers as Spotify, Netflix or whoever are simply not interested in you. The efforts for them are too high in comparison to their potential benefits."*

The tough competition in the automotive industry indicates that one manufacturer will not be able to leverage its sales to achieve a similar magnitude of available target devices as the

24 https://developer.ford.com from June 5, 2019

25 https://developer.gm.com from June 5, 2019

mentioned mobile app ecosystems. Hence, manufacturers could cooperate and provide a cross-brand app platform that enables the deployment one single app to cars of multiple brands.

## 7.5 Join existing platform ecosystem

Beside the building and nurturing an own ecosystem, multiple experts mentioned the option of joining an existing platform ecosystem. Manufacturers would not be required to attract own app developers, since all apps in the ecosystem would be available in their cars as soon as the platform is integrated.

*„If you take an existing ecosystem as Android, you just need to integrate the existing platform into the car and all Android [Auto OS] apps will be available. You don't need to attract any third parties by your own."*

Furthermore, the implementation of an established platform entails the utilization of a matured and proven technology, which should decrease testing efforts in comparison to the creation of an own platform. Moreover, due to its open source approach[26] Android as concrete example provides the option to contribute to the platform. In this way, manufacturers are able to extend the platform's feature set, while Google as platform owner is responsible for long-term maintenance.

## 7.6 Contribution and Next Steps

The results of our work contribute to literature on platform ecosystem in three aspects. First, we illustrate how a traditionally physical product transforms into a digital infrastructure that enables innovation (Henfridsson et al., 2018). While a car was formerly assessed by its driving performance and its design, customer expectations are shifting towards the vehicle's connectivity and the availability of digital services. Driven by this change of expectation, manufacturers implement digital technology into their car and push the emergence of a new digital infrastructure. Second, we consider the process of platform launch from a new perspective by shifting the context from the frequently investigated green field (Stummer et al., 2018) to an environment with established companies that perceive disruptive invasion of digital native players into their domain. This "brown field" perspective reveals new aspects, as the consideration of the incumbent's legacy in culture, skills or technology. Third, our findings embody a theoretical contribution on competitive strategies in platform ecosystem. While prevalent literature focuses on battles between existing ecosystems (Cennamo & Santalo, 2013) or approaches for the launch of new ecosystem against established players (Karhu et al., 2018), we consider the dispersal of platform ecosystems on a new digital infrastructure without any established ecosystem.

Our studies focus on a development in the automotive industry, though we consider our findings as generalizable beyond the specific industry. For instance, the industrial internet of things in which formerly analogue assets like production machines transform to a digital infrastructure reveal multiple similarities to the automotive domain. For this reason, we consider a continuation of our work as worthwhile. Our next steps will focus on a detailed understanding of the three identified platform ecosystem options. We aim to identify actual pros and cons of the options and contribute to the development of specific recommendations for action. Thereby it needs to be considered that different manufacturers reveal different contexts as number of

26 https://source.android.com from June 6, 2019

customers, budget and size of the organization, which may influence the decision for one of the identified options.

# 8 From Product to Platform: How can BMW compete with Platform Giants? (P5)

| | |
|---|---|
| Authors | Weiß, Niklas* (niklas.weiss@tum.de)<br>Schreieck, Maximilian* (maximilian.schreieck@in.tum.de)<br>Wiesche, Manuel** (wiesche@in.tum.de)<br>Krcmar, Helmut* (krcmar@tum.de)<br><br>*Technical University of Munich, Chair for Information Systems, Boltzmannstraße 3, 85748 Garching, Germany<br>** Technical University of Dortmund, Chair for Digital Transformation, Martin-Schmeißer-Weg 12, 44221 Dortmund, Germany |
| Publication | Journal of Information Technology Teaching Cases (JITTC) |
| Status | Accepted |
| Contribution of First Author | Introduction, Theoretical Background, Research Design, Results, Discussion |

**Table 14. Fact Sheet Publication P5**

**Abstract**

Today, it is natural for digital services to be available anytime, anywhere. Thanks to digital platforms, we can use our connected with intelligent assistants, stream movies on our networked televisions, and wirelessly control our homes with smartphone apps. Within the last few years, our cars have also been transformed into computers on wheels. However, the apps with which we are accustomed are often not available in cars. Why are cars different? This teaching case examines how the worldwide automotive manufacturer BMW developed its onboard infotainment system into a digital platform. While the transformation of a manufacturing firm into a technology company reveals multiple challenges, new actors enter the stage. In 2017, Google announced the launch of Android Automotive OS, a complete operating system for cars with access to a vibrant platform ecosystem. Should BMW accept Google's offer to use Android Automotive OS as a digital platform for their cars, or should they strive to create their own proprietary platform ecosystem? This teaching case introduces the dynamics of digital platform ecosystems and illustrates the "platform conundrum" that many traditional companies must confront: is it better to build a new proprietary platform ecosystem or join an existing dominant platform ecosystem? We provide rich insights from BMW's development, sales and strategic divisions, helping students to understand the risks, chances and challenges of various choices that occur in the context of digital platform ecosystems, and why such decisions might be crucial to the future of traditional companies such as BMW.

## 8.1 Introduction

Have you ever been annoyed because you were unable to use your favorite apps while you were driving? Did you ever wonder why you are forced to use the old-school integrated navigation system instead of Google Maps with up-to-date maps and real-time traffic? Or why there is no better way to listen to your favorite Spotify playlist than streaming it via your smartphone, which requires a prior connection to the car? Or why there is no streamlined solution to read and write WhatsApp messages while driving? As smartphone users, we are accustomed to choosing from millions of available apps to enhance our devices at any time. These apps are usually free, frequently updated, and if an alternative with more features and a better look-and-feel pops up, we can simply download the new app and remove the old one. So, why is that possible for smartphones but not for cars?

For almost two decades, the worldwide automotive manufacturer BMW has been trying to integrate their customers' digital world into their vehicles in an attempt to react to changing customer needs. In the past, car buyers emphasized vehicle design, running characteristics, and safety. These qualities are still appreciated by customers today, although they are complemented by the need for connectivity and availability of apps. According to a 2018 McKinsey study, 40% of car owners would change their favorite car brand exclusively based on the availability of enhanced digital services[27]. BMW tried to fulfill these new customer requirements as did the other major car manufacturers. In 2003, the first online services were introduced into a BMW car design. After several advancements, the newest generation of the BMW infotainment system was released in the summer of 2018. The newly designed BMW onboard system is constantly connected to the Internet and provides apps such as a parking lot finder,[28] music streaming,[29] integrated e-mail and calendar features[30] , and apps for different calling services.[31] However, the number of available apps is quite low as compared to the vibrant smartphone platform ecosystems with which most customers are accustomed.

In this teaching case we first introduce BMW ConnectedDrive as the brand representing BMW's infotainment and connectivity features. By describing various milestones in the evolution of ConnectedDrive, the second section illustrates how BMW directly integrated digital services into their cars. The third section examines the emergence of Android Automotive OS, an operating system developed specifically for cars by Google that is independent from any smartphone. The appearance of this new technology presents BMW with three strategic options for the future of their onboard automotive infotainment system, which are described in the final section.

[27] https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/profiling-tomorrows-trendsetting-car-buyers retrieved 16.03.2020

[28] https://www.bmw-connecteddrive.de/app/index.html#/portal/store/Base_Connected_Parking retrieved 10.03.2020

[29] https://www.bmw-connecteddrive.de/app/index.html#/portal/store/Base_Connected_Music retrieved 10.03.2020

[30] https://www.bmw-connecteddrive.de/app/index.html#/portal/store/Base_ExchangeOffer retrieved 10.03.2020

[31] https://www.bmw-connecteddrive.de/app/index.html#/portal/store/Base_AssistNoTPEGOffer retrieved 10.03.2020

## 8.2 BMW ConnectedDrive

BMW encapsulates all infotainment and connectivity features in its vehicles under the BMW ConnectedDrive brand. The onboard infotainment system is called BMW OS and can be considered as the heart of the Connected Drive product offering.

### 8.2.1 Head-Units and Infotainment Systems

The infotainment system is one of many software components that run on miscellaneous electric control units (ECUs) inside a car. All ECUs are connected by means of bus systems that transfer data between the various components. The infotainment system operates on an ECU called the head-unit, which receives data from multiple sensors throughout the entire car that are processed infotainment system applications. For example, the navigation system is one of the applications that receives and processes global positioning system (GPS) data from the vehicle's antenna control unit to display the current position of the car on a map. BMW maintains triennial development cycles for its head-units. When a new head-unit generation development is complete, it is integrated into every car model that will be subsequently produced. Older models, previously produced with older generations of the head-unit are eligible to receive upgrades.

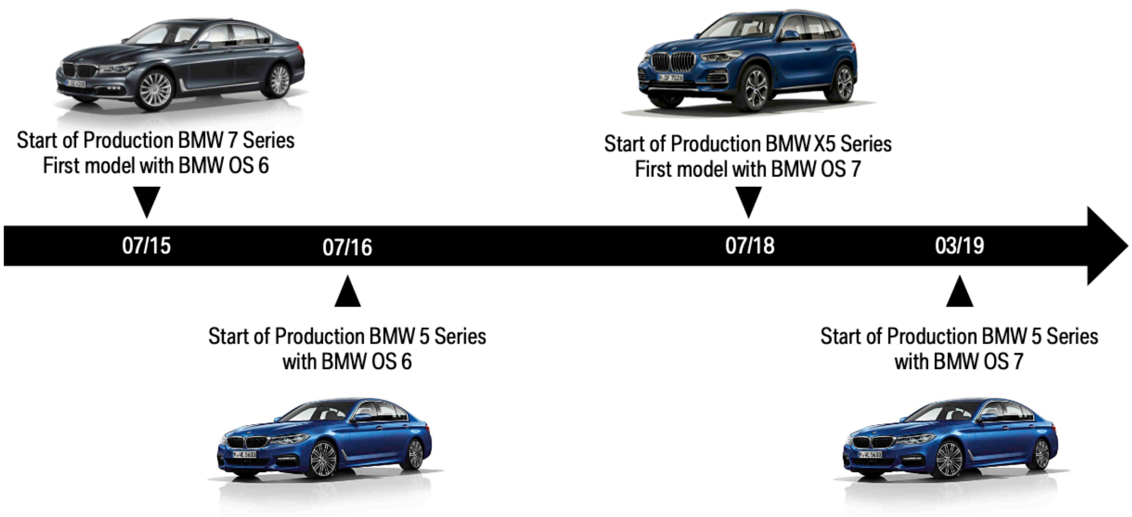



**Image 4. Timeline BMW OS 7 in 5 Series and X5 Series**

A new generation of a head-unit is always expected to provide new features for the lowest possible additional cost. Reducing production costs is a major driver of developmental efforts. There is a fundamental difference between software and hardware. While a head-unit must be purchased for every vehicle produced, one version of the corresponding software can be installed into millions of cars without any sizable additional cost. Therefore, the fewer hardware resources required by the software, the cheaper the per car cost of the hardware. For example, making a decision regarding the memory module capacity of a new generation head-unit illustrates this point. One option would be to use a 16 gigabytes (GB) module that costs 30 Euros per unit. A second option would be to use a 12 GB module costing 25 Euros per unit. The market forecast predicts a production volume of 2.5 million cars. The five-Euro cost difference, multiplied by 2.5 cars translates into a potential savings of 12.5 million Euros, funds that can be allocated to implementing software optimization that will enable the cheaper hardware option.

Software performance increases with the elimination of any abstraction. The more adapted the code is to the inherent hardware, the better its eventual performance. As a result, developed software is inextricably intertwined with its corresponding hardware. Therefore, the release of a new infotainment system is associated with the triennial head-unit development cycle. However, software development does not strictly end with the release of the integrated infotainment system. Refinements such as performance improvements and the incorporation of new features are implemented in subsequent product upgrades. When such an upgrade is completed, the new software is installed into all newly produced cars and distributed to existing customers during routine maintenance checks at BMW servicing partner.

### 8.2.2 Introduction of the ConnectedDrive Store

Customers ordering a new BMW online or through a dealer, can choose from several infotainment options. While the top tier includes all available software features and high-end hardware, the low-budget variants provide less functionality as well as restricted hardware with lower computing power and a smaller sized cockpit screen display. Previously, the chosen configuration was considered to be permanent feature for the lifetime of the car. This setup fundamentally changed in 2012 with the introduction of the BMW ConnectedDrive Store, which enabled customers to extend their car's capabilities by installing new digital services, such as adding real-time traffic information or map updates to the navigation system, months or even years after a car's initial purchase by buying such services through the BMW ConnectedDrive Store.

The ConnectedDrive Store gives BMW several advantages. First, a BMW owner can purchase BMW's digital services at any point in time, which creates the opportunity for generating additional revenue from existing customers. Second, most digital services require online connectivity. In contrast to mobile phones, whose network operation costs are borne by the user, BMW is the carrier for the car's subscriber identity module (SIM) card. Even though utilizing BMW's connectivity features is not billed according to usage, the company can apply value-added subscription models that require additional payments for higher-tier features or to extend subscription services at the end of a predetermined period (see Image 5). Third, BMW was one of the first automotive manufacturers to directly sell digital services in their cars, which provided an opportunity to promote BMW as an innovative and progressive car brand.


**Image 5. BMW ConnectedDrive Store in BMW OS 4**

The introduction of the ConnectedDrive Store enabled BMW to distribute digital services independent from the initial purchase represented the potential for implementing innovative business models. However, BMW's sales strategy was not adapted to this change. The fixed development phase of vehicle equipment features such as brakes or seats usually ends upon the commencement of a car's production cycle - unlike the continuous software development cycle, which persists past a product's sale. Therefore, the idea of rolling out maintenance and feature upgrades in a previously sold car was counterintuitive to most automotive sales managers. Releases of new digital services (or new versions of existing services) were coupled with releases of new vehicles, meaning that digital services were sold as features of a new car and not as standalone products. Since a free testing period was not applicable to features such as brakes or seats, this option was not made available for new digital services.

*"Apps belong to a certain optional equipment line because our organization is just able to work with optional equipment structures. Even if we want to go other ways, we haven't the abilities currently." ~ Sales Operator for Digital Products*

### 8.2.3 The Modular Distribution of Apps

The in-store purchase of a feature did not trigger the deployment of new code to the head-unit. Instead, a functionality was preinstalled in all cars and activated simultaneously. Dynamic content, such as weather or news was displayed by a preinstalled browser. However, this changed with the sixth generation of the infotainment system, which was launched in the BMW 7 series in the summer of 2015. This new system contained a dedicated platform component that was able to remotely download, install and run app packages. While the capability of the preinstalled browser of the previous generation was limited to displaying online content, natively running apps were now able to access interfaces deeply embedded in the car. Hence, it was possible to provide apps that processed information on the current velocity of the car, the destination set in the navigation system or signals of the parking control system. The parking app ParkNow[32] for instance, uses such automotive data to recognize if a driver is searching for a parking lot in the vicinity of the programmed destination and provides relevant information on nearby parking facilities. Even though the new platform's capabilities enabled many new use cases for additional apps, the most prominent features of the infotainment system, such as the navigation system and the radio continued to be deployed as preinstalled software components alongside the platform (see Image 3).

[32] https://www.bmw-connecteddrive.de/app/index.html#/portal/store/Base_Connected_Parking retrieved 10.03.2020

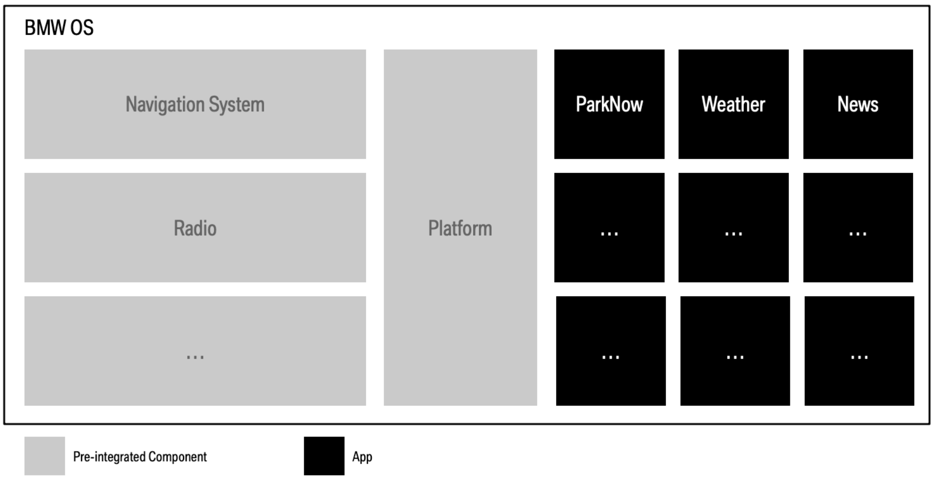



**Figure 15. Infotainment System Architecture**

These enhanced capabilities of the platform engendered more and more departments within BMW, to start own app development efforts. In 2015, software engineers from the research division started the BMW Labs[33] initiative. Customers were able to apply to the program and beta test new onboard services that could be cancelled at any time. This way, BMW engineers could accelerate prototyping efforts and receive faster feedback from actual customers. One of the first features launched by BMW Labs was the integration of the "if this then that" (IFTTT) service that enabled the conjunction of miscellaneous web services. With onboard data, BMW drivers could use their car as a sensor for triggering any kind of action in many digital services; for example, when the fuel level went beyond a predefined level, the IFTTT service could set a refueling reminder in the driver's Google calendar. BMW also began cooperating with established digital service providers such as Spotify and Microsoft to implement Spotify Music and Microsoft Exchange apps in the BMW infotainment system. While these partners provided access to their servers, the onboard apps were developed by BMW.

The rapid emergence of more and more app development projects revealed problems with the platform itself. Although the emergence of innovative features was appreciated, the architecture of the underlying operating software was not designed for modular extensions. As illustrated in the beginning of this chapter, traditional automotive software development is tightly interwoven with the associated hardware development. The memory consumption of every single software component is strictly predefined, which means that a component's budget is estimated and defined before the development efforts begin. Maintaining compliance with the available resources is a central goal of automotive software engineers. Accordingly, the overall memory budget is a sum of all component's memory budgets. However, this approach conflicts with the idea that apps that can be continuously distributed into a system and consume arbitrary amounts of memory. Software architects mitigated this issue by introducing a specific platform component to receive a dedicated memory app's memory budget, which should be allocated dynamically. However, the scarcity of memory and its management became permanent issues for the newly created platform development team. Another problem category concerns inconsistent interfaces provided by foreign onboard modules. The interfaces revealed insufficient documentation or a lack of robustness in the context of generic use by a larger

[33] https://labs.bmw.com retrieved 22.03.2020

number of apps. Consequently, many app developers ended up taking a time-consuming trial-and-error approach to evaluating the functionality and limitations of interfaces.

*“You just don’t know which results in which condition are delivered by the interface. The documentation is either not available or just rather raw. All you can do is implement, deploy your app to a test vehicle and see what happens.” ~ App Developer*

### 8.2.4 App for Automotive

As illustrated in the previous section, directly distributing modular apps into the infotainment system led to several challenges for BMW. The development and operation of a digital platform exposed how fundamentally traditional automotive software development differs from modern software development. Engineers elaborated further approaches to integrating digital services into BMW’s cars. Smartphones, which connect directly to the car appeared to be appeared to be a promising direction for integrating apps into the car’s onboard OS.

In 2011, BMW had already enabled its infotainment system to display applications that were rendered on a connected smartphone. The “Apps for Automotive” (A4A) technology afforded mobile app developers to display user interfaces, by directly embedding them into BMW’s proprietary infotainment system. Therefore, a software development kit (SDK) provided by BMW, had to be integrated into the app’s source code. The SDK exposed access to BMW’s human machine interface (HMI) and allowed mobile app developers to assemble their own user interfaces using standardized building blocks. While the app itself was still processed by the smartphone, it remotely controlled the displayed HMI in the car’s central information display. When the customer connected his or her smartphone to the car, all A4A apps were listed in a dedicated menu.

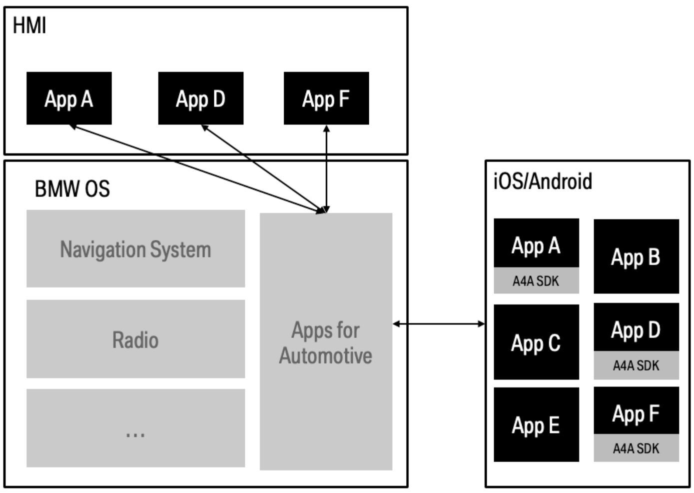



**Figure 16. Apps for Automotive Architecture**

First, BMW used A4A to improve the integration of its own smartphone apps into its cars. For example, the BMW Connected App that enabled customers to control certain automotive functions by means of a smartphone was enhanced by a feature that, at the start of a drive, fetched the phone’s calendar and suggested destinations in the car’s navigation system. However, the focus of A4A technology was integrating third-party apps into BMW’s proprietary infotainment system. Therefore, BMW enabled app developers interested in an integrating of their mobile apps to contact a dedicated partnership management division called the App Center. When the App Center determined that the integration of an app was valuable and met BMW's requirements, the A4A SDK was provided to the app developer. The App Center also provided comprehensive support to third-party developers to facilitate development and ensure the app's smooth integration with BMW's user interface. While many app developers contacted BMW, the reverse was sometimes also true. The music streaming provider Spotify and the online image hosting service Flickr were directly contacted by the App Center. In contrast to partnerships for onboard app development, BMW reached out to partners to build their own integrations of their services into the BMW infotainment system by using A4A technology. All in all, over 50 audio, navigation and entertainment apps that were available on the iOS and Android platforms were made compatible with BMW's proprietary infotainment system.

From a customer perspective, A4A technology was part of the overall BMW infotainment system product offering. When a customer configured his or her new car appropriately, all compatible apps installed on its smartphone were automatically made available in the car. In this way, each app developer who chose to integrate the A4A SDK into its app increased the attractiveness of BMW cars. App developers also increased the attractiveness of their apps by making them available during a car ride to BMW drivers. BMW's reputation and brand power enticed several smaller app development firms to partner with the automaker.

However, while several app development firms appreciated this cooperation, BMW also experienced reluctance from desired partners. Such reluctant firms argued that the number of new users that would specifically be attracted by the integration of their apps with BMW's onboard platform was quite low. The development effort required for such integration was remarkable, yet the number of potential users that did not already have an app installed and who owned an appropriately equipped BMW car was small. While a new Android or iOS feature usually has the potential to reach billions of users, A4A integration could not even reach millions of customers. In 2015, these difficulties increased, when BMW revealed the new generation of its infotainment system. All existing integrations required major updates by app developers to remain functional. While BMW increased its effort in partners, an increasing number of app developers terminated cooperation. Only a small number of A4A apps were available upon the launch of the next generation system. Eventually, the A4A technology was removed from BMW's infotainment system in 2018 and replaced with a lighter-weight technology that enabled the integration of BMW's own smartphone apps, so no further external partners were acquired.

When considering projecting smartphone apps into an external infotainment system, most think of the solutions created by established digital platform companies such as Apple's CarPlay, Google's Android Auto or Baidu's CarLife. BMW has a long history of cooperating with Apple to integrate iOS devices into their cars. In fact, the preceding "iPod Out" technology was result of joint development of BMW and Apple. It enabled the user to control its connected iPod via the vehicle's audio control buttons. Shortly after the release of Apple CarPlay in 2014, it was integrated into BMW OS 6, which was released in 2015. BMW announced the implementation of Android Auto in BMW's OS 7 in 2020. Even though this implementation advanced the integration of the customer's digital world into their cars, this solution was not considered perfect by many experts. First, the functionality of projected modes was limited to the smartphone and did not exploit the vehicle's own capabilities. While there are well-established use cases for the smartphone, such as messaging or audio playback through a third-party OS, automotive-specific application programming interfaces (APIs) were not available to app developers. Second, multiple BMW experts emphasized the importance of a seamless customer experience, meaning that the integration of digital services should appear naturally. The conflict between projected modes through the existing infotainment system and the need to connect a smartphone to use them impeded providing such a seamless experience.

### 8.2.5 Remote System Updates and the Emergence of Platform Governance

Improvements such as memory management optimization were usually implemented in product upgrades distributed to all newly produced cars as well as existing cars during maintenance visits by a service partner. However, very few customer vehicles were regularly serviced by a partner such that they could benefit from every product upgrade. By contrast, apps were distributed remotely and could be easily received by almost all BMWs on the road. The highly dynamic nature of apps, on the one hand, and pre-integrated software components' static development process on the other caused problems. Long-term platform development cycles impeded the team's ability to react to the quickly evolving world of apps. This mismatch was addressed by the introduction of the seventh generation of BMW OS in July 2018 (the most current generation at the time of this article's writing). BMW OS 7 enabled the remote distribution of product upgrades over the air, a highly anticipated feature. First, other premium brands such as Mercedes-Benz and Audi started promoting their wireless updating capability in the beginning of 2018. Tesla implemented remote software updatability in its Model S in 2015. BMW's image as a premium brand and technology leader required a prompt catch-up. Aside from these marketing aspects, engineers appreciated this advancement as well. New features in pre-integrated software components, such as the navigation system or the radio could be distributed to a large number of customer vehicles simultaneously. The platform team also desired the updatability of their component. Remote system updates enabled the iterative evolvement of the BMW platform and therefore more appropriate reactions to current developments by the internal app development teams.

Besides the growing complexity of runtime resource management, the platform team recognized an increased demand for the governance of platform processes. The proven procedures of pre-integrated software components did not meet the requirements of the highly dynamic nature of app development. While common product upgrades were released three times a year, several app teams updated their apps many times a month. Furthermore, the platform was not used only by a small group of software engineers but by hundreds of app developers. The established communication routines of cooperating teams in the development division did not scale for one platform team that had to provide support to dozens of app teams.

*"Actually, in traditional software development here at BMW engineers just work for their department. There is no intention for developing things for others outside their own department." ~ Process Manager in the Development Division*

Therefore, the platform team introduced multiple measures. First, documentation for app developers was moved from the internal wikis with restricted access to a developer portal accessible to every BMW employee. Furthermore, several tools were created to enhance app development and large parts of the app release and testing process were automated. The platform team introduced a question and answer forum and fostered the emergence of an app developer community to provide mutual support and reduce platform developer workloads.

With the launch of BMW OS 7, even more teams decided to implement functionality as an updatable app in the infotainment system. In addition, management recognized its rising importance and increased the allocated workforce to the platform. While the previous generation of the platform was developed and maintained by one internal and five external developers, the new platform was operated by a team of 15 internal and more than 30 external workers. However, the general attitude of BMW's developmental divisions still attributed a higher importance to feature development then to the development of a generic platform.

*"Most of the management attention, most of the key performance indicators and most of the budget allocation mechanisms follow the logic of feature implementation. You usually need to provide a business case to receive funding for a development project. However, it is quite harder to provide a business case if your product – in our case the platform – cannot be sold to customers directly." ~ Member of the Platform Development Team*

While several new apps had just launched with the seventh generation of the infotainment system, the apps that existed in the previous generation required porting to the new OS version. Most involved managers and engineers expected all features that were available in the predecessor head-unit to be available in the launch of the new infotainment system, BMW OS 7. However, the platform component itself was bound to the regular software development procedures, which considered platform completion to occur weeks before the beginning of the head-unit's production cycle. Therefore, app developers were forced to follow this strict schedule and start their work before the platform's stability was determined, which caused large efforts on their side and amplified the development of new features.

### 8.3 The Entrance of Google

In May 2017, Google announced the development of Android Automotive OS, an Android operating system familiar to millions of mobile app developers and tailored to run in the head-unit of a car. This system should no longer depend on a connected smartphone as did the previously mentioned Android Auto, but be directly embedded in the car. As its first partners, Audi and Volvo declared that their proprietary infotainment systems would be replaced by Google's Android Automotive. After a two-year development phase, Google invited app developers to submit automotive apps using their new automotive operation system in May 2019. In parallel, Volvo announced the release of the Polestar 2 model as the bearer of several innovative **technologies** including the implementation of Android Automotive. Aside from the basic operation system itself, Android Automotive OS included Google's most prominent and popular apps like Google Maps and Google Assistant as well as the Google Play Store, which was the entry point for third-party apps. While millions of apps were available for Android smartphones, support for Android Automotive OS was not provided by default. First, Google needed to extend and adopt Android's capabilities to the automotive context. Subsequently, developers needed to implement these capabilities to also adopt their apps for the automotive context. For instance, large parts of the system were locked to user input whenever the vehicle's velocity exceeded a certain predetermined limit. This speed-lock feature is a regulatory requirement in several countries so apps must be notified to appropriately manage this system state. The graphical user interface (GUI) also needed to be prepared for interactions that occur during a car ride, such as sizing certain elements appropriately and avoiding unnecessary distractions. In the beginning, the provided GUI libraries of Android Automotive OS focused on specific types of apps. According to Google, media apps that enabled music streaming, the playback of audio books or the streaming of radio broadcasts were the most desired domain of apps for automotive implementation. Therefore, in May 2019 media apps were afforded the first chance to adopt to Android Automotive OS. Later, Google announced that it would support apps from the navigation domain as well as the communication domain.

After the announcement of the Android Automotive OS launch in Volvo's Polestar 2, several other car manufacturers proclaimed their release of an Android-based infotainment system. However, while brands such as General Motors and the Renault-Nissan-Mitsubishi alliance promoted cooperation with Google and the implementation of Android Automotive OS, others, such as the Volkswagen Group and Fiat-Chrysler, declared the development of their own digital platform based on the Android Open Source Project. While both approaches included the implementation of Android as an operation system on automotive head-units with apps pre-installed by the car manufacturer, only Android Automotive OS included Google's most prominent and popular apps as Google Maps, Google calendar and Google Assistant. When users registered their Google accounts in their new cars using Android Automotive OS, they were automatically given access to the already familiar Google, which included their pre-programmed personalized preferences and utilization history. Additionally, Android Automotive OS contained the Google Automotive Services (GAS). GAS is a collection of services that can be implemented by app developers. For example, apps can use location data provided from Google Maps, integrate the Google Assistant to enable speech interactions with the user and manage financial transactions such as subscription fees or in-app purchases with Google Pay. While car manufacturers needed to create their own alternatives to these and other services, Google was able to provides access to already proven solutions from their mobile app ecosystem.

Eventually, Android Automotive OS implicated the availability of Google's Play Store with all third-party apps that were adopted to the automotive context. Developers that decided to submit their apps to the Play Store were assured that their apps would be available in all cars with Android Automotive, independent of the car manufacturer. Google ensured that all specified standards were satisfied by manufacturers and app developers and guaranteed a functional interplay. A car manufacturer with its own marketplace needed to establish own standards, independent from Google's App Store. Furthermore, apps that implemented features based on GAS were dependent on the availability of these services. For example, an app that provided information on electric charging station locations required access to a kind of map, that allowed the implementation of custom points of interest in its user interface rather than of Google Maps. If no appropriate alternative was available, the feature would not be available in the car manufacturer's infotainment system.

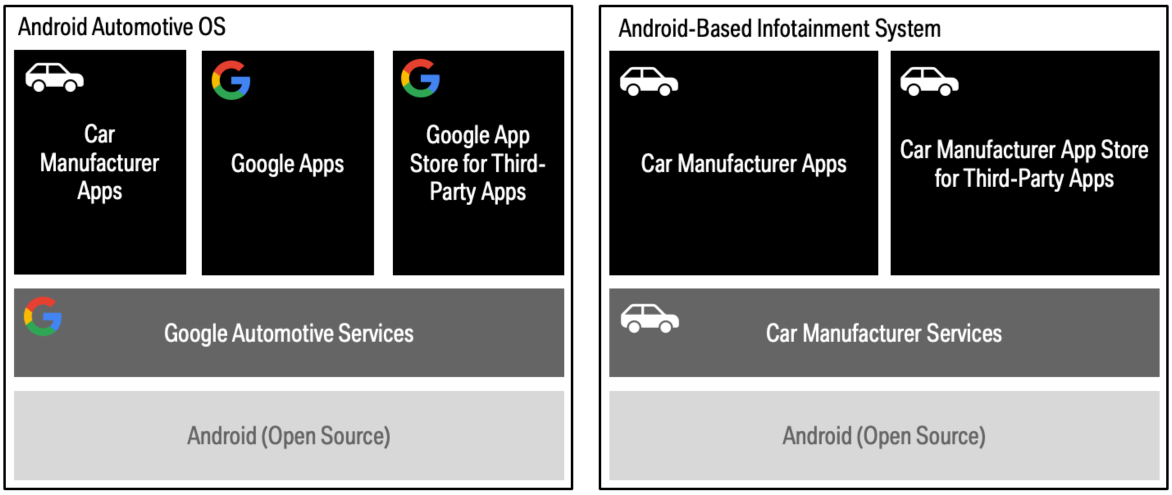



**Figure 17. Android Automotive OS vs. Android-Based Infotainment System**

## 8.4 BMW's Strategic Options

The appearance of Android Automotive OS accelerated the development of digital services for the automotive market and forced car manufacturer to make decisions own automotive app strategies. Should they join Google's ecosystem or strike out on their own? The integration of vibrant digital platform ecosystems appeared to be a critical success factor for the future of the automotive business. While some competitors had already announced their decisions, most manufacturers were still searching for an appropriate answer on Googles offer. In the fall of 2019, BMW was also evaluating a variety of strategic options. Three directions were discussed by the experts involved: creating a proprietary BMW platform ecosystem, establishing a collaborative platform ecosystem, or joining the existing ecosystem.

*"Of course, we are in a competitive situation, but we are also in a partner situation. We will always be in this 'frenemy' situation with a Google, with an Apple and whoever comes next. The question is of course how do I play that?" ~ Digital Product Manager*

### 8.4.1 Creating a Proprietary BMW Platform Ecosystem

The first strategic option was the establishment of a proprietary digital platform developed and operated by BMW. This meant that BMW would follow its prevalent strategy of owning the digital platform but would make it accessible to third-party developers. Several experts inside BMW emphasized that in this scenario BMW would maintain comprehensive control of all decisions regarding its digital platform ecosystem. For example, BMW could independently decide if app developers could utilize the data on the battery status of its electric cars. Consequently, this information would not be exclusively available to BMW so the establishment of profitable business cases would be aggravated. However, the potential for innovative solutions for a car's charging process would be fostered, increasing overall attractiveness of BMWs. The accessibility of autonomous driving capabilities, could be limited to BMW to diminish the danger of misuse and associate the autonomous driving experience with BMW. While such deliberations would also be possible in cooperation with other partners, the existence of an independent BMW platform ecosystem would facilitate such decisions.

*"On the other hand not only profit is an outcome but also other things like data sovereignty and new business models, that I can possibly build if I keep the data. If I have my own system and keep control of it, I can potentially do more than selling cars. [...]" ~ Member of the Platform Development Team*

Considering the technological perspective, the system could be based on the BMW's prevalent infotainment system, on Android Open Source or any other appropriate technology. However, due to its maturity and robustness, Android Open Source appeared to be a promising approach for many BMW software engineers. The system was established and mature, and several other manufacturers had already declared a shift toward Android Open Source. Consequently, the project would receive further maintenance and evolvement. Furthermore, the technology was already popular with developers. Its large amount of documentation, tutorials, and online support communities like Stack overflow facilitated the work of software engineers in comparison to using a proprietary solution. Moreover, Android Open Source facilitates the attraction and integration of new developers.

*"I then I could imagine and this is my personal opinion, then we do it ourselves with a very limited number of apps, but then exactly with those apps we want." ~ Sales Operator for Digital Products*

Even though recruiting new developers to the platform would be facilitated by the implementation of an Android-based platform ecosystem, the challenge of attracting third-party developers would remain. Like the approach to A4A technology, the number of potential target devices for a BMW-specific digital platform ecosystem would be small. Furthermore, multiple BMW experts agree that the massive effort required to launch its own platform ecosystem, would need to be justified by the aforementioned advantages. Some of the most important resources for establishing a well-functioning platform ecosystem are money and employees. Tremendous financial investment is required for to establish and operate a successful platform ecosystem. Considering employees, previous digital platform development efforts bare a certain lack of attention to the development and operation of digital platforms. The required resources, available competencies and skills as well as BMW's ability to attract them impacts the overall platform ecosystem strategy. Most BMW experts agree that the attracting and retaining the most creative minds in the field is important for developing an outstanding platform in an increasingly competitive field. However, this is especially critical because the car manufacturers need to simultaneously engage in various new fields which leads to high efforts and competency requirements. Establishing all of the required competencies seems to be unrealistic for several BMW experts, which prompts them to suggest that the manufacturer should focus on its core competencies and consider what the company is willing to invest in other fields.

### 8.4.2 Establishing a Collaborative Platform Ecosystem

The fierce competition in the automotive industry indicates that one manufacturer will not be able to leverage its sales to achieve a similar magnitude of available target devices for previously mentioned mobile app ecosystems. Several experts at BMW consider a collaborative approach in which various manufacturers develop and run a common digital platform ecosystem as further strategic option. While each partner could maintain its own brand-specific

GUI or other features, third-party apps could be compatible with cars of all manufacturers. Considering the smartphone market, the market share of such an automotive consortium could not exceed 50%. In 2019, just 13% of all smartphones were run on iOS while Android revealed an 87% market share[34]. Although the Android Play Store offers 2.6 million apps compared to the iOS App Store's 1.8 million[35], iOS App Store revenue was twice as high as that of the Play Store[36]. This example indicates that a consortium would need to strive for a relevant rather than dominant market share to attract third-party developers and generate value.

A consortium-based platform ecosystem would distribute all inherent costs and risks beyond the partners involved. Development and operations effort could be equally distributed among the partners and any potential lack in competency could be mutually compensated. Each new partner would increase the number of cars available to third-party developers, decrease the risk to all other consortium members and contribute resources to the development and operation of the platform.

*"[...] The platform ecosystem participation thought is quite interesting. It can find favor with volume manufacturers as well as premium manufacturers. Everything scales much better when I have two-digit millions of new vehicles and they can still be addressed to some extent over lifetime." ~ Project Manager in Platform Development Division*

Although manufacturers could pool their endeavors to develop a digital platform ecosystem, their organizations would need to cooperate with external suppliers. The development of a common digital platform would also require support by delegated contractors. The commission of external software development firms by a larger consortium of car manufacturers with a relevant market share however, implicates massive risk regarding the potential limitations of German and European competition laws. Several BMW experts considered this aspect to be a major challenge to the creation of a collaborative ecosystem.

*"[...] Ideally a platform or an ecosystem already exists and we only participate in it, because that is simply more promising than doing it ourselves. But it is also an effort issue. Somebody has to do all of that. The purchasing department does not have to negotiate with a store provider and conclude a contract, it has to conclude contracts with 50 app providers worldwide, or how many we want to have. Furthermore, we have to validate 50 apps and do all of these things. That has to be done. It is an effort issue." ~ Digital Product Manager*

Furthermore, many experts emphasized that all approaches to the development of common automotive software components have failed in the past. Even though these attempts focused on the development of a common middleware and an operation system for cars, independent of the creation of a digital platform ecosystem with third-party developers, automotive manufacturers still failed. The fear of losing their competitive advantage and their conviction that their individual technical solutions are superior have impeded the establishment of common standards.

---

[34] https://www.statista.com/statistics/272307/market-share-forecast-for-smartphone-operating-systems/ retrieved 03.04.2020

[35] https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/ retrieved 03.04.2020

[36] https://sensortower.com/blog/app-revenue-and-downloads-1h-2018 retrieved 03.04.2020

### 8.4.3 Joining Google's Platform Ecosystem

Finally, experts considered the option that BMW might join an existing and growing platform ecosystem by implementing Android Automotive OS. Like the implementation of Android Open Source, this scenario also entails the utilization of a mature and proven technology, which should decrease testing efforts as compared with the creation of an independent, proprietary platform. With apps like Google Maps, Google Assistant and others, the Play Store and GAS, Android Automotive OS reduced the software development effort required by each manufacturer to a minimum.

*"Considering technical and cost reasons I would only take Android Automotive OS because the platform has already proven itself. [...]" ~ Digital Product Manager*

GAS would facilitate the development of third-party apps and ensure solid payment mechanisms for all platform transactions. Furthermore, Google provides comprehensive support to app developers, including tooling, documentation and collaborative events like developer conferences. BMW would not have to be responsible for the quality of the apps provided in the app store. With its already established control mechanisms, Google would ensure the exclusion of undesired or malicious apps.

Considering the customer perspective, multiple sales experts at BMW emphasized the seamless integration of the customer's digital world into a car afforded by an implementation of Android Automotive OS. Third-party apps that were adopted to the automotive context could be purchased on the customer's smartphone and automatically also be available in the car. Customers would just need to register their Google accounts in the car and all playlists, pictures and videos, as well as all personal data and logins would be synchronized and available without any further effort. Additionally, Google's most popular services such as Google Maps and Google Assistant are exclusively available with Android Automotive.

*"[...] the end user will certainly demand that he not only takes his ecosystem from home, but also that he doesn't have to log in anywhere all the time and then that's it. That speaks more the language that he takes this ecosystem than the manufacturer system. [...]" ~ Sales Operator for Digital Products*

Even though the advantages of Android Automotive OS appear to be magnificent, several BMW experts raised doubts against the option. BMW would lose control over the development and rollout of a central component of its cars and would have to rely on Google's decisions regarding new features, bug fixes and also new version rollouts. Moreover, Google would control which models or markets should first receive any OS version. Beside these strategic decisions, experts mentioned that BMW would lose an important customer contact point. While Android Automotive OS provides options for user interface customization, customers might think of Google, not BMW, as the system's provider. The customer experience during a car ride would no longer be exclusively designed by BMW but a large part would be determined by Google. Strong brand bonding is especially important in the automotive market because cars and driving are especially associated with customer emotions.

Furthermore, experts elaborated that the implementation of Android Automotive OS implicates relinquishing any ambition on digital business model implementation inside BMW's cars. Since Google would manage all transactions on its platform and capture a predefined percentage of each transaction, the car manufacturer would not be involved. Line with the smartphone market, in which manufacturers' value capture is limited to the sale of the device itself, BMW's business model would be limited to the sale of the individual car and several associated maintenance services. The role of a platform owner with scaling platform businesses is obsessed by Google. Experts from BMW's strategic departments link this aspect to the changing nature of the automotive market. While large parts of this business are conducted with sales or lease contracts, an evolution toward a more usage-based business model is expected. Even though, selling cars is expected to remain as a lucrative business in the future, the market share of mobility services will grow. Independent from specific solutions such as car-sharing, ridesharing, and ride-hailing, these businesses are usually used and managed via online apps. As soon as digital platforms are available, the integration of these services with cars appears as logical step. However, the implementation of Android Automotive OS would mandate these transactions to be exclusively managed by Google. BMW would not be involved in such business models.

## 8.5 Questions for Reflection

1. Think about available features and the automotive sensor data that could be used by app developers via API. What apps could be built upon these features and data? Think about current vehicles, but also consider predictable future features such as autonomous driving.
2. What stakeholders are involved in ConnectedDrive? How could a digital platform for automotive apps connect different sides of the market? How could BMW benefit from connecting these sides?
3. Think about the challenges that appear when you start to connect the various sides of a single market. What is the most critical aspect to consider? Which strategies might help a new platform owner to master the inherent challenges?
4. What are the main challenges to BMW in implementing its own digital platform in their cars? For each challenge, consider if it is specific to BMW or if you would expect similar conditions at other firms? Consider technological as well as market-specific aspects.
5. Think about Google's strategic motivation behind the engagement for a digital platform ecosystem for cars. In which way does this strategy differ from BMW's strategy for a digital platform in its cars?
6. How should BMW proceed from here? Consider the chances and risks of the three illustrated options and think about both internal and external factors that might influence its decision.

## 8.6 Teaching Notes

The teaching notes contained in section are not part of the originally published teaching case. However, they are provided on request to support classes in the endeavors understanding the nature of digital platforms. Therefore, they the teaching notes added as additional section in this chapter.

### 8.6.1 Introduction

#### 8.6.1.1 Audience and Focus

This teaching case illustrates the challenges of BMW, a traditional firm that is confronted with the digitalization of its physical product and the inherent implementation of digital platforms. It is intended for undergraduate as well as graduate students of Information Systems and contains rich insights on technological as well as business aspects of digital platforms. The overarching question of the case contemplates BMW's strategical options for digital platform ecosystems and encourage considerations on the "platform conundrum" that several traditional firms are confronted with. After a description of the intended learning objectives, the remainder of these teaching notes contains a proposal for elaborating this case within a student class. The *Teaching Approach* is structured into four major sections: Technical Aspects of Digital Platforms, Market-oriented Aspects of Digital Platforms, Challenges of Traditional Firms, and Digital Platform Ecosystems and Platform Options. Furthermore, the *Additional Material* section contains further exercises that facilitate a deeper understanding of the previously introduced concepts.

#### 8.6.1.2 Learning Objectives

The presented teaching case supports students in:

- Understanding fundamental principles of digital platforms (layered architecture, two-sided markets, network effects, platform governance, platform options) using the example of automotive infotainment systems
- Understanding challenges of a traditional firm that is confronted with the digitalization of its physical product.
- Understanding dynamics of digital platform ecosystems and the inherent danger of disruption and conceive the revealed options ("Make or Join") for traditional firms that are confronted with the "platformization" of their product.

### 8.6.2 Teaching Approach

The following section provides exercises and questions for reflection on the concept of digital platforms and platform ecosystems. The questions address students' analytical and creative thinking. We provide a possible outline for a 120-minute class discussion. While the recommended questions for reflection are attached to the teaching case and are essential for a fundamental understanding of the presented coherences, the optional exercises in section 3 can be applied in a more flexible way and are not part of the estimated 120 minutes. The approach assumes that the teaching case was distributed in advance and the students are familiar with the content. The attached questions for reflection should be prepared by the students to initiate a lively discussion in the classroom.

#### 8.6.2.1 Introduction

**[15 Minutes]** The topic can be introduced by raising the question on examples of digital platforms that students know from their daily life. Thereby, students should recognize that there are different types of platforms. While innovation platforms enable the creation of complementary products, services or technologies, transaction platforms focus on the facilitation of interactions within a multi-sided market. Furthermore, there are hybrid platforms that enable complementary innovation as well as transactions between different sides of a market (Cusumano et al., 2019; Schreieck et al., 2016).

Examples for innovation platforms:

- Desktop operation systems such as Windows or MacOS
- Gaming systems such as Sony's PlayStation, Microsoft's Xbox or Nintendo
- Speech assistants such as Amazon's Alexa or the Google Assistant
- Cloud services such as Amazon AWS, Microsoft Azure, Google Cloud or Alibaba Cloud
- IoT systems such as Siemens' MindSphere
- Business system such as SAP's NetWeaver or the Salesforce App Platform

Examples for transaction platforms:

- Mobility platforms such as Uber, Lyft, Didi
- Booking platforms such as Airbnb, Booking.com, TripAdvisor
- E-commerce platforms such as Google Search, Amazon Marketplace or Alibaba Marketplace

Examples for hybrid platforms:

- Apple's iOS (innovation via apps, transaction via app store)
- Google's Android (innovation via apps, transaction via app store)
- Facebook (innovation via external apps, transaction via advertisement)
- Microsoft's Office 365 (innovation via external extensions, transactions via marketplace for extensions)
- Tencent's WeChat (innovation via mini-programs, transaction via advertisement)

The following box with additional information illustrates the power of digital platforms. The most successful platform companies established hybrid platforms or in Amazon's and Alibaba's case firms operate an innovation as well as a transaction platform.

**Additional Information**

In 2019, the top ten of the largest companies of the world by market value revealed seven firms that rely on platform business: **Apple (961.3$ billion)**, **Microsoft (946.5$ billion)**, **Amazon (916.1$ billion)**, **Alphabet (863.2$ billion)**, Berkshire Hathaway (516.4$ billion), **Facebook (512.0$ billion)**, **Alibaba (480.8$ billion)**, **Tencent (472.1$ billion)**, JPMorgan Chase (368.5$ billion), Johnson & Johnson (366.2$ billion).[37]

#### 8.6.2.2 Technical Aspects of Digital Platforms

Digital platforms reveal technical as well as market-oriented aspects (Cusumano et al., 2019; Hein, Schreieck, Riasanow, et al., 2019; Schreieck et al., 2016). While this teaching approach

[37] https://www.statista.com/statistics/263264/top-companies-in-the-world-by-market-value/ retrieved 28.04.2020

covers both aspects, their considerations are separated in this teaching approach to facilitate the student's comprehension. This first section of the class focuses on the technical perspective of digital platforms, which is especially relevant for innovation platforms.

**[10 Minutes]** Now students should consider the previously developed list of innovation platforms. What do they have in common? What is especially relevant here? The discussion should lead to the subsequently illustrated (technical) definition of digital platforms.

**Definition**

**Digital Platforms**

A digital platform embodies a foundational product or technology upon which complementary products or technologies can be developed (Baldwin & Woodard, 2009b; Tiwana, 2014). Apple's iPhone and its iOS operating system embody a digital platform that provide a technological base for developers, who create new digital products, called applications ("apps"). Via so-called application programming interfaces (APIs), app developers have access to a broad variety of the phone's features like the camera or the audio speakers as well as sensor data like gyroscope or GPS. Generally speaking, apps implement features of the platform core by accessing them via APIs (Henfridsson & Ghazawneh, 2013). While some apps as Instagram or TikTok are globally available and strive for the largest possible user base, others aim at more specific use cases as providing information on the municipal public transportation schedule to residents of one specific city. Independent from their use case, every new app enhances the iPhone's capabilities and attracts new customers (Hagiu & Wright, 2015).

**[15 Minutes]** Now, students should present their prepared solutions for the first question for reflection. Goal of this exercise is to elaborate that the vehicle is not just a new digital platform for the adoption of existing apps but enables completely new and innovative use cases due to the mass number of new capabilities.

### Question for Reflection

Think about available features and the automotive sensor data that could be used by app developers via API. What apps could be built upon these features and data? Think about current vehicles, but also consider predictable future features such as autonomous driving.

### Possible Solution

| Feature | API |
|---|---|
| Parking app that displays parking options as the car approaches the configured destination in the navigation system. As the car is set to park mode, the app notifies cooperating parking services as ParkNow that the billing can be started. As soon as the driver returns to his vehicle, the incurred price is displayed in the car and can be automatically paid. | GPS Data, Parking Mode |
| By introducing so-called environmental zones, cities aim to reduce traffic-related CO2 emissions. It can be assumed that driving bans will not only apply to diesel vehicles but to all types of combustion engines. Hybrid vehicles that have both an electric motor and a combustion engine must be able to react accordingly. A corresponding app could take care of the driving bans, which might even be adjusted daily to the current air values, | GPS Data, Driving Mode |

| | |
|---|---|
| and automatically switch to fully electric operation when driving into an environmental zone. | |
| Smartphones constantly try to learn from our usage behavior to smoothen the user experience. When you open the search function on your iPhone, several apps that are considered as the most appropriate are suggested. These suggestion base on properties as current time, current location or your last actions. A car could do similar things by automatically start the seat heating at a certain temperature. Since different people maintain different preferences, this smart heating will adopt to individual needs. It might be possible that the car is connected to your calendar, aware when you need to leave to work in the morning and preheats in a way that it is perfectly tempered when you enter it on a cold winter morning. Similar features are possible with air conditioning, window lifter or massage seats. | Temperature Sensors, Seat Heating, Air Conditioning, Window Lifter |
| The access to the vehicle interlock could enable app developer to use the car's trunk as public available mailbox for delivery services. It would not require a mailbox at work or at home but every location with connected cars nearby could serve as destination for your orderings. This would require that the owner of the car is aware of it and does not store own possessions in the car. However, he could be engaged by a reward that is earned with every delivery in the car. | Vehicle Interlock, GPS Data |
| The availability of all driving data as well as highly detailed traffic and map data could enable a driving coach for more sufficient driving. The app would notify the driver when he is stepping on the gas while nearing a crossing or traffic circle. Similar to sports apps on the smartphone, the app could visualize improvements, engage user by providing comparisons with other drivers | Driving Data (Accelerometer, Angle of Gas Pedal, Velocity, Breaking Power, GPS Data) and Traffic/Map Data |
| Similar to the previously described driving coach, an app could be a driving school for learner drivers. Modern cars are able to recognize traffic signs, traffic lights, other road users and could detect every mistake you did in your lesson. Furthermore, the app could optimize the next lesson to train your weaknesses and improve your overall skill in an optimal way. And even if worse mistakes were made, the vehicle could intervene if necessary and prevent worse things with driver assistance systems. | Traffic Sign Recognition, Traffic Light Recognition, Driving Assistance, GPS Data |
| Most taxis round the globe track the traveled distance with additional hardware, so-called taximeters. This additional hardware costs hundreds of euros, while it tracks similar data as the already build-in mileage sensors. Consequently, this functionality could be implemented by an app that is automatically managing the billing for customers by interacting with smartphone apps. Furthermore, the owner of a taxi fleet could get insights on centrally collected data of all taxis in the fleet. | Mileage Sensor, GPS Data |
| The introduction of electric cars and the need for charging them at home reveals new requirements for power supply infrastructure. An app that monitors the charging status of multiple cars could orchestrate parallel charging processes in a neighborhood. The charging plan could be optimized according to the user's calendar that the car is always fully charged when it is needed. The user could receive current information on the charging status on his smartphone. Furthermore, the app could integrate cars into existing smart grid solutions. | Battery Charging Management |
| An app could integrate the car in the user's smart home system. The garage door could be automatically opened when the car is nearing home. In the other way around, the car could warm up for an upcoming drive, when the heating inside the home is turned on. | GPS data, Air Conditioning, Seat Heating |

**Table 15: Proposal for automotive APIs with potential for app developers**

| **Additional Information** |
|---|
| A modern BMW car implements more than 1500 interfaces in its infotainment system. Apple's iOS provides access to roughly 200 APIs on an iPhone[38]. |

**[10 Minutes]** The exposing of features as a window opener for app developers is facilitated by the concept of a layered architecture. The concept of a layered architecture should be introduced and discussed in the class. The discussion could be fostered with questions as "Why is layered architecture such an important concept in the context of digital platforms?" or "Which layers can be described in the infotainment system?". An additional exercise on abstraction can be found in section 3.1.

| **Definition** |
|---|
| **Layered Architecture**<br>The idea of abstracting low-level computation processes evolves in the concept of a layered architecture (Yoo et al., 2010). On a hardware level, computers process information as bits that are represented by the presence ("1") or absence ("0") of electricity. However, software developer is not interested in designing electronical circuits on a micro-processor. Therefore, so-called assembler languages transfer binary code ("01001") in human-readable code. However, assembler code still reveals a tight coupling to hardware, meaning that various processor architectures require dedicated assembler code. However, an app developer doesn't want to care about process architecture of the hardware the app is running on neither. Therefore, so-called higher programming languages (as Java, Python, C, C++, Swift, Kotlin…) abstract assembler code and enable the writing of code that runs on different hardware, while being even better readable for humans (You can check Wikipedia for deeper insights in the world of programming languages). An app platform embodies another layer of abstraction that facilitates the development of apps even more. The advantage of layered architecture is not just the simplification (and thereby acceleration) of app development but also the interchangeability of layers. A performance improvement on a lower level brings advantages for all layers above. Layered architecture is a major driver for the rapid advance of computer technology in the last decades. |

### 8.6.2.3 Market-oriented Aspects of Digital Platforms

After elaborating technical aspects of digital platforms, students should understand the market-oriented platform perspective. Especially the idea of two-sided marketplaces and network effects should be imparted. Therefore, we propose to go back to the initially created list of transaction platforms. What do have these digital platforms in common? The discussion should lead to the subsequently illustrated (market-oriented) definition of digital platforms.

| **Definition** |
|---|
| **Two-Sided Markets** |

[38] https://developer.apple.com/documentation/ retrieved 16.03.2020

While Apple provides support and tools for app development, the actual innovation is created by third-party developers that are not employed by Apple. So, why are hundreds of thousands of programmers motivated to develop and maintain apps for the iPhone without receiving any wages by Apple? The answer is the market-oriented perspective on digital platforms. A digital platform enables and coordinates interactions between target groups on two or more sides (Gawer, 2009; Rochet & Tirole, 2003). While app developers create new apps on the one side, the integrated iOS App Store promotes these apps to users of the iPhone on the other side. Whenever a user purchases an app, the digital platform manages the transaction and ensures that each side receives the stipulated value, thus that the user is able to download the app and that the app developer receives his/her the price in return. Thereby, Apple as platform owner captures a previously defined share of the transferred revenue and in this way benefits from each transaction that is conducted on its platform.

**[10 Minutes]** Now, students should present their prepared solutions for the second question for reflection. This exercise aims at the understanding of multi-sided markets and in which way this idea is applicable to the automotive context.

**Question for Reflection**

What stakeholders are involved in ConnectedDrive? How could a digital platform for automotive apps connect different sides of the market? How could BMW benefit from connecting these sides?

**Proposed Solution**

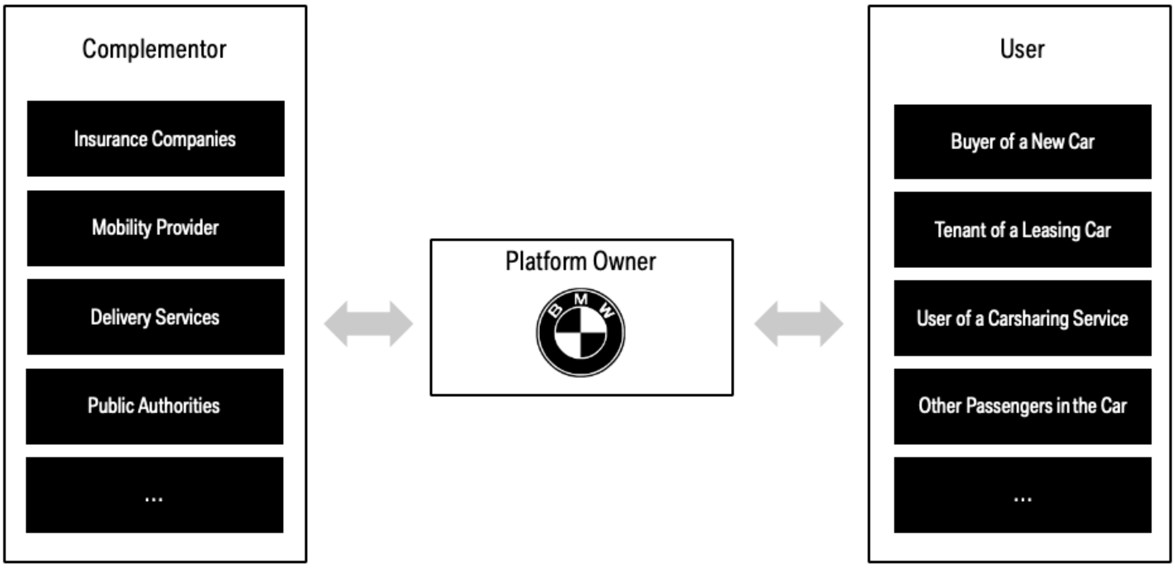



**Figure 18: Stakeholders of a two-sided market for automotive apps**

The platform could connect complementors that are interested in automotive data or providing services directly in the car on the one side and different kind of users on the other side. Thereby, several kinds of transactions could be managed by the platform and connect both sides in this way.

- First and obviously the platform could manage all purchases of new apps by providing a marketplace in shape of an app store. Like a mobile app store, BMW could keep a predefined share of the price that the customer paid for the app.

- The platform could enable that customers allow insurances to collect data from their cars and provide optimized contracts. As reward for enabling this transaction, BMW could receive a predefined share of each contract that relies on the collected data.
- A mobility service app (Lyft, Uber, Didi) that is directly integrated in the car could facilitate the booking of an available car. This is especially relevant when cars are driving autonomously and no driver with a smartphone is available. BMW could receive a share of each mediated drive.
- BMW could support authorities in providing information to customers or enabling smart traffic control systems. The contact to public authorities enables BMW to participate in the process of shaping new mobility solutions.
- As described in the proposed solution for the first question for reflection, delivery services could use the car's trunk as mailbox. BMW could receive a predefined share for each successful delivery.

**[15 Minutes]** The next section focuses on the chicken-egg problem that needs to be solved by every new digital platform and illustrates the central meaning of network effects. Therefore, the students' prepared solutions for the third question for reflection should be discussed.

**Question for Reflection**

Think about the challenges that appear when you start to connect the various sides of a single market. What is the most critical aspect to consider? Which strategies might help a new platform owner to master the inherent challenges?

**Proposed Solution**

Typically, the most critical aspect of connecting two sides of one market is getting both sides on board, while the respective other side is quite small or not existent. This problem is also known as the 'chicken-egg-problem'. BMW's new *Apps for Automotive* technology was just available in a small number of newly produced cars. Why should an app developer spend efforts, while no customers were available? On the other side, customers have no incentive for paying additional money for an app platform, where no apps are available. The chicken-egg problem needs to be solved by every new digital platform. Why should someone register an account for Uber, when no drivers are available? One the other hand, why should someone spend his time in a car and wait for customers, if no users are available? Similar scenarios occurred for Airbnb (hosts and guests), eBay (seller and buyer) or Facebook (users and advertisers). However, there are strategies that helped digital platforms to solve the chicken-egg-problem and become one of today's most successful businesses:

- Subsidize one side: When no side is available so far, the platform owner can subsidize one side to attract it and foster the onboarding of the other side. When Uber starts its business in a new city, it usually has to solve the chicken-egg problem every time. To solve the issue, Uber gives away vouchers that provide a few free rides for new user that register an account within the first weeks. The existence of users again, attracts drivers.
- Exploit available user base: Facebook was not initially designed as digital platform for advertisers but as social network for users around the globe. When enough users were available, Facebook was able to use this huge market-side to attract the side of advertisers.
- Focus on specific target groups: Today, Amazon connects all kind of dealers with their customers. However, at the beginning Amazon started as online shop for books. This

specialization increased the likelihood that new merchants would be connected to customers that are interested in their product—books. When the platform was established, Amazon extended its portfolio and started to become the global marketplace as it is today.

- Envelop existing platforms: By enveloping others, a new platform can leverage the shared relationships with other established platforms and their networks. When BMW developed its *Apps for Automotive* technology for Android as well as iOS, it tried to exploit the developer base of Android as well as iOS. Developers were not required to build completely new apps for BMW, but could integrate an add-on in their existing code.

When the chicken-egg problem is solved, the interactions of both sides provide room for unbound growth. Goal of this exercise is to develop the idea of network effects and its meaning for digital platform. The discussion should lead to the subsequently illustrated definition of network effects. Further exercises to two-sided markets and value creation on digital platform ecosystems can be found in section 8.6.3.2.

**Definition**

**Network Effects**

The extent of users is frequently considered as success factor of a digital platform. The reason for this is the existence of network effects (Rochet & Tirole, 2003). In general network effects refer "to the impact that the number of users on the platform has impact on the value created for each user" (Parker, Van Alstyne, & Choudary, 2016; Schilling, 2002). Imagine the invention of the telephone. The first person that owned a telephone did not receive any value from it, since there is no other person that could be called. However, a second telephone brought value to the owner of this second device as well as the owner of the first telephone. The third telephone again brought value to the first and the second owner and so own. Each new device increased the number of possible connections and created value for every other participant in the network. This effect occurs also for apps. Every new app in an app store might attract new users, which embody potential customers also for other apps in the store. The more users on the other side are participating, the more app developers are attracted to build high-quality apps that are again available for all user. Since these effects occurs between different sides of a market and create value they are called "positive cross-side" network effects. However, also negative and same-side network effects exist and influence a platform ecosystem (see (Parker, Van Alstyne, & Choudary, 2016).

**Additional Information**

The high variety of available apps is considered as one of the main reasons why Apple was able to outperform prevalent phone manufactures after the iPhone's launch in 2007. Back then Nokia boasted a market share of 63 percent in the mobile phone market, while Blackberry was the runaway market leader for mobile devices in professional contexts (Cusumano et al., 2019). However, the in-house development departments of these traditional firms were not able to compete with the rapid velocity of innovation that was created by Apple's third-party app developers. Furthermore, both firms failed with the establishment of own digital platforms. While Nokia sold its mobile phone sector for 5.44 billion dollars to Microsoft in 2014[1] (Apple reported a profit of 39.5 in 2014, mainly originated in iPhone sales[1]), Blackberry's market share needs to be measured in hundredth of percentage points.

#### 8.6.2.4 Challenges of Traditional Firms

**[10 Minutes]** After understanding the fundamental idea of digital platforms, student should now elaborate in which way the concepts align with the traditional organization and technology of a traditional firm as a car manufacturer. Therefore, students should present their prepared solutions for the 4th question for reflection.

**Question for Reflection**
What are the main challenges to BMW in implementing its own digital platform in their cars? For each challenge, consider if it is specific to BMW or if you would expect similar conditions at other firms? Consider technological as well as market-specific aspects.

**Proposed Solution**

| Technological Challenges | Business Challenges |
|---|---|
| **Tight coupling of software and hardware development cycles**<br>As the episode on head-unit development illustrates, the development of automotive software is directly linked to the underlying hardware. When designing a new ECU, all hardware and software requirements are specified. The software is a part of the ECU and, like all other components, is made ready for the market in development processes that sometimes take several years. Once this process is complete, the ECU is secured and released for production. New specifications are then collected and new components developed for the next generation of the ECU. The development and use of a digital platform are difficult to realize with this approach. According to the technology-oriented definition of a platform, applications are based on the technical platform core. Developing the platform according to the established process means completing the software immediately before production begins. The development of applications based on this core is no longer feasible until the start of production. Since a newly specified platform will be developed for the next generation of the ECU in the classic procedure, app development is not possible for this either. For these reasons, the traditional coupling of software and hardware development poses great challenges for manufacturers with regard to the implementation of a digital platform in terms of development and production processes.<br>**Legacy in software architecture** | **Optional equipment as exclusive business logic**<br>According to the market-oriented definition, a platform represents a market that reinforces network effects of different players. In the case of applications, for example, an App Store represents such a market. Users can use the platform to enhance their original product with additional functionality and benefit from the work of external application developers. For an automobile manufacturer, extending the functionality of his product after delivery of the vehicle is an untypical process. After-sales activities are traditionally focused on repair and the sale of spare parts. Apps are therefore currently understood to a large extent as the usual special equipment that can be added to the vehicle during configuration. The sale of applications after the car has been delivered requires various changes and adjustments in processes and organization, but also in the manufacturer's IT systems.<br>**Feature-driven development funding**<br>Traditional companies usually focus on developing new features to differentiate their product from competing products. New or improved functionality are sales arguments to customers and generate revenue. A digital platform, on the other hand, does not generate any direct benefits for customers in the first step. Only the applications made possible by the platform offer the prospect of sales. However, internal evaluations for decision-makers are usually made directly or indirectly via the financial return on a development. Accordingly, function-oriented |

| | |
|---|---|
| The coupling of software and hardware is also visible in the software architecture, which reveals a high amount of proprietary and hardware specific solutions. Even if BMW tries to increase the abstraction and reduce the amount of legacy, a complete refactoring of the software in one big bang is not possible in addition to the usual development efforts for new features. The legacy however, limits the manufacturer's agility and makes it difficult to keep up with the market innovation speed.<br>**Increased Complexity**<br>The introduction of new software layers in form of a digital platform and inherent apps increases the complexity within the software. This increased complexity of software requires the availability of sufficient expertise. However, attracting and retaining the most creative heads out there is important to develop new ways of generating revenues and providing an outstanding platform in an increasingly competitive field. Furthermore, not only additional competences are required due to an increased complexity, but also more financial resources. A vast amount of money is required for the establishment and operation of a successful digital platform. | development projects are often given preferential treatment. The conversion of these mechanisms in order to also motivate the development of basic technologies internally more strongly represents a further challenge for the organization.<br>**Limitation of customer-side**<br>The power of digital platforms is originated in an economy of scale on the demand side, meaning that every new customer attracts further complementors due to cross-side network effects. However, the potential for customers that purchase digital services in a car is currently limited to people that own or rent BMW car. Due to the strong competition in the automotive industry, a change of market share appears not likely. Therefore, BMW has to find other ways to increase the scope of potential participants of its digital platform on the demand side to nurture a vibrant two-sided market. |

**Table 16: Technological and business challenges of BMW becoming a platform owner**

### 8.6.2.5 Digital Platform Ecosystems and Platform Options

In this final section, students should understand in which way digital platforms are embedded in digital platform ecosystems and the inherent power of a platform owner. Therefore, students should understand platform governance. Furthermore, the platform options that are presented in chapter 4 of the teaching case should illustrate the prevalent platform conundrum for BMW as representor of multiple traditional firms that face the platformization of its physical product.

**[10 Minutes]** To understand the relevance on the decision between the presented platform options, the students need to conceive the central role of the platform owner. Therefore, the actual responsibilities of a platform owner should be collected in the classroom.

Responsibilities of a platform owner:

- Maintain relationships to all involved sides in the platform ecosystem (e.g. Promote platform to new customers and new developers, inform on changes, etc.)
- Define pricing and revenue sharing models (e.g. define amount of money that is captured per transaction)
- Provide resources for the creation of complementary products (e.g. Software Development Kits)

- Decide on the platform openness (e.g. who is allowed to develop apps)
- Prevent undesired behaviour (e.g. fraud, low qualitative complements, violent content)
- Develop and maintain platform core (e.g. the operating system)
- Define a competitive strategy towards other digital platform ecosystems (e.g. prevent compatibility with other systems)
- Ensure trust (e.g. customers receive the promoted complement after the purchase)

All these activities consider the management of interactions between different stakeholders in the digital platform ecosystem, also referred as platform governance. The discussion should lead to the subsequently illustrated definition of platform governance.

**Definition**

**Platform Governance**

The core of a digital platform is in the center of every digital platform ecosystem (Tiwana, 2014). The exposed capabilities of the platform should attract app developers (Dellermann et al., 2016; Kude et al., 2012), who build complementary apps on the one side, which should be consumed by customers on the other side (Tiwana, 2014). Platform governance is the "partitioning of decision-making authority between platform owners and app developers, control mechanisms, and pricing and pie-sharing structures" (Tiwana, 2014). For instance, in the iOS ecosystem, Apple as platform owner decides, which application programming interfaces (APIs) of the iPhone are accessible by third-party app developers (Eaton et al., 2015; Henfridsson & Ghazawneh, 2013). Furthermore, each app needs to be submitted for review by Apple before it can be launched in the App Store. The app developer needs to choose the price of its app in a predefined selection of pricing steps and keeps just 70% of its revenue. The rest belongs to Apple as platform owner. However, platform governance should also facilitate the creation of new apps (Kude et al., 2012). By exposing generativity the platform owner unleashes potential for innovation (Tilson et al., 2010). With the annual release of a new iOS version, Apple publishes new APIs that provide new possibilities for app developers. Comprehensive documentation and annual developer conferences inform and attract developers to build new apps. A platform with strictly controlled generativity would counter such endeavors (Ondrus et al., 2015). Hence, the platform owner needs to balance control over the digital platform and its ecosystem on the one side and provide a certain level of autonomy for app developers to foster innovation on the other (Rausch et al., 2012). In this way, platform governance covers multiple tactical decisions that impact interactions between the platform owner and app developers (Kude et al., 2012; Schreieck et al., 2016).

**[10 Minutes]** After the students have understood the central importance of platform governance, the next step is to think about the way Google or BMW would fill the role of a platform owner to leverage their strategical interests. Therefore, the respective motivations need to be considered. The discussion should reveal the difference between a (traditional) resource-based strategy that strives for the exploitation of competitive advantage through the redeployment of resources and a network-based strategy, which aims for an optimal position in previously established value networks.

### Question for Reflection

Think about Google's strategic motivation behind the engagement for a digital platform ecosystem for cars. In which way does this strategy differ from BMW's strategy for a digital platform in its cars?

**Proposed Solution**

- Google is a platform company that is world-leading in connecting different sides of a market to value (co-)creation network but does not reveal expertise in the manufacturing of own physical products. The building up of capabilities for own physical products is expensive and time-intensive. On the other side, Google does have existing digital platform ecosystems, which embody value co-creation networks.
- By implementing its digital platform in cars of several manufacturers, Google enables compatibility for apps in a large number of cars.
- This envelopment of the fragmented automotive industry creates a large network that connects customers of multiple automotive brands with a large amount of Android app developers. The emergent network is structured as platform ecosystem, with Google as platform owner in its center.
- The availability of Google's established and top-edge services as Google Maps and the Google Assistant attracts customers to join the platform ecosystem. Car manufacturers outside Google's platform ecosystem need to compensate these services to remain attractive.
- BMW is a traditional manufacturing company that strives for competitive advantage by the redeployment of resources, meaning that it utilizes available assets and knowledge to create products that appear desirable to customers.
- Considering this resource-based view, the implementation of a digital platform is a measure for making the product even more desirable for customers. The capturing of value by the platform is not covered in this sight.
- Another characteristic of a resource-based strategy is the aiming for a product that is rare, inimitable and non-substitutional. Any kind of cooperation with direct competitors endangers such aspirations. The exchange of information inherits the danger of losing competitive advantages. An implemented digital platform should be as exclusive as possible to create competitive advantage by providing apps and services that are not available in cars of other manufacturers.

Google's strategy is known as platform envelopment. The discussion should illustrate the idea behind that strategy and why it fits Google's strategic interests.

**Definition**

**Platform Envelopment**

Sometimes, a digital platform begins to offer the functionality of another platform to enhance its existing bundle of functionality and leverage shared user relationships (Eisenmann et al., 2006, 2011). The act of envelopment is also frequently referred as 'swallowing'. It occurs when two adjacent platforms have an certain overlap in its functionality or user base (Tiwana, 2014). Furthermore, it is a launch strategy for a new digital platform that enters a market (Stummer et al., 2018). For example, when Apple bundles the news articles of several news providers in its News app, it is no longer a single news provider app that is mediating its content to the users but its Apple's platform. While the user base of Apple's ecosystem might push the popularity of certain news providers or specific articles, the providers are losing control in which way their content is provided to the user.

**[15 Minutes]** Now, students should present their prepared thoughts for the final question for reflection. The discussion should emphasize the prevalent challenge for car manufacturers and illustrate the idea of platform options in general. Furthermore, students should understand the specific advantages and disadvantages of the presented options.

**Question for Reflection**

How should BMW proceed from here? Consider the chances and risks of the three illustrated options and think about both internal and external factors that might influence its decision.

**Proposed Solution**

| Strategical Option | Chances | Risks |
|---|---|---|
| Create own BMW Platform Ecosystem | **Control on portfolio**<br>An own BMW platform ecosystem would leave the full control on BMW's side. While market-rollouts, new platform features or also the guidelines, which kind of apps are allowed and which not would be decided exclusively by BMW while all other options would require agreements with other stakeholders. In this way, BMW's platform could stand out from other platforms.<br>**Governance of available capabilities**<br>BMW would not just decide which kind of apps would be supported but also which capabilities of the car are exposed to these apps. In this way, BMW could surpass other platforms and attract third-party developers with the enablement of new use cases, which are not available in other platform ecosystems.<br>**Value Capture**<br>An own BMW platform ecosystem would bring BMW in the position of a platform owner and enables value capture at every transaction that is managed through the platform.<br>**Acknowledgement**<br>The creation of an own vibrant platform ecosystem embodies a major challenge in several aspects. However, the successful establishment of such an ecosystem would be acknowledged in public perception and contribute to BMW's image as technology leader. | **Attracting of third-party developer side**<br>The lack of an existing developer community requires the attracting of new third-party developers that develop apps for the relatively small customer side, which is limited to BWM drivers. The episode on *Apps for Automotive* illustrates that a failure of this attracting embodies a major risk for BMW.<br>**Financial investments**<br>The standalone development and maintenance of a comprehensive and matured platform ecosystem requires massive funding. However, OEMs as BMW need to manage multiple technological disruptions as the electrification of mobility or the emergence of autonomous driving in parallel. Hence, limited financial resources embody a risk for an own BMW platform ecosystem.<br>**Acquiring of new expertise**<br>Traditionally, software development is not a core expertise of a car manufacturer. Therefore, BMW needs to acquire new expertise at a massive scale. The failure of this recruiting efforts may endanger a successful BMW app platform.<br>**Technical development**<br>Each software development project carries inherent risks of failure due to inappropriate estimations of efforts or unforeseen technical impediments. This is of course also valid for the development of a digital platform. |
| Create Collaborative | **Enlarged customer side** | **Attracting of third-party developer side** |

| | | |
|---|---|---|
| Platform Ecosystem | A cooperation with other OEM's would potentially solve the problem of a too limited customer side. The multiplication of potential users could facilitate the attracting of third-party developers and enhance cross-side network effects.<br>**Collective resources**<br>Multiple experts inside BMW consider the required amount of expertise as well as financial resources as massive challenge for the creation of an own platform ecosystem. However, the cooperation with other OEMs provides the chance for joining forces and reducing the efforts for every single member of the cooperation. Furthermore, the expertise of one OEM could be compensated a potential lack within other OEMs and of course the other way around. | Even though, the cooperation of OEMs increases the potential customer side the non-existent complementor side is not neglectable. Also, a cooperation of OEMs would need to solve the chicken-egg problem and attract new third-party developers to provide an attractive app portfolio for customers.<br>**Cooperation with other OEMs**<br>The creation of a platform ecosystem requires decision-making regarding multiple organizational as well as technological aspects. However, a strong faith in the own brand and superior capabilities of the own organization were reasons for failure of cooperation between different car manufacturers in the past.<br>**Compliance with competition law**<br>Following their traditional development approaches, the creation of a common platform ecosystem would be supported by delegated contractors. However, as described in the case already, the commission of external software development firms by a larger consortium of car manufacturers with a relevant market share however, implicates massive risks regarding potential harm of German and European competition law. |
| Join Google's Platform Ecosystem | **Matured technology**<br>The Android operating systems embodies a matured technology that is currently running on more than one billion devices. The inherent play services respectively automotive services are similarly proved. Furthermore, the transfer into the automotive domain is conducted by Google, a firm that is originated, experienced and leading in the development of digital platforms. The utilization of a solid technological base facilitates the implementation of a valuable product for customers.<br>**Availability of bundled services**<br>Joining Google's platform ecosystem entails access to Google's digital services as Google Maps or the Google Assistant in BMW cars. | **Loss of value capture in platform business**<br>While the car as product strengthened by the implementation of a high qualitative platform and the availability of a large app portfolio, the access to any kind of value capture in this platform business is restricted. This is especially relevant in the context of servitization of mobility, in which customers rent mobility on-demand via digital services instead of owning an own vehicle.<br>**Loss of governance and control**<br>Since Google is the platform owner in its platform ecosystem, Google has the exclusive right to decide on developer guidelines, control mechanisms, available |

| | | |
|---|---|---|
| | The availability of these services is desired by multiple customers and increases the attractiveness of the overall car.<br>**Existing third-party developer side**<br>The existing Android app developer community embodies a major advantage in comparison to options that entail the need for the establishment of a new developer community. BMW would not be required to solve the chicken-egg problem but would benefit from a large and highly qualitative app portfolio. | resources for developers, the revenue sharing and pricing models and more. Consequently, Google will not only decide which kind of apps are allowed or not allowed inside BMW cars but also which kind of apps need to be supported by BMW. However, this platform governance will primarily support Google's interests not BMWs.<br>**Loss of customer touch point**<br>The Android brand is considered as part of Google's corporate identity. Even though, the Android system for smartphones enables device manufacturers to customize the experience, the central design decisions are made by Google. This implicates that BMW would lose the control on the experience within the cockpits of their cars. Furthermore, the differentiation from other OEMs that also implement Google's platform is impeded. |

**Table 17: Chances and risks of BMW's platform options**

Beside the emphasizing of the inherent challenge of the platform conundrum, students should understand the concept of option thinking in the context of digital platforms. The following definition illustrates platform options and its origins in the investment literature.

**Definition**

**Platform Options**

The idea of real options is frequently used in the context of strategies on digital platforms and digital platform ecosystems (Tiwana, 2014). Originally, the term is rooted in investment literature and refers to the possibility of doing something without the obligation to really do it. The idea of real options thinking is typically applied in the context of uncertainty and facilitates the management of a lack of knowledge regarding future events and coherences. While granting multiple options in parallel may cause additional costs, the risk of missing the right option is reduced. The highly volatile characteristics of digital platform ecosystems, with fast changing environments and rapid developments induces the application of real option thinking.

Further exercises for the illustration of the platform conundrum can be found in section 8.6.3.3.

### 8.6.3 Additional Material

#### 8.6.3.1 Optional Exercise - Abstraction

Abstraction is used to make models that facilitate the implementation of complex processes with simple code snippets. Imagine a website that displays a simple black headline on white ground. If no abstraction would be available, the web designer would need to define which specific pixels on a display that shows his website should be black (= the text) while the remaining pixels would have to be defined to be white (= the background). However, when the user starts scrolling and the content should move these assignments need to be realigned. This has to be done for each and every display size and all kinds of devices, where the website should be available. Of course, this is not the case in reality. Modern web development builds on large stack of abstraction that assumes that tasks for web designers. Eventually, the illustration of a black headline on white ground requires one simple line of Hypertext Markup Language (HTML) code.

**Optional Question**

Think about the advantages of abstraction. Consider the lack of abstraction in the context of digital platforms. What problems occur due to the lack of abstraction?

**Proposed Solution**

- Low level code is harder to understand and requires app developers to spend efforts in understanding complex coherences.
- Low level code is harder to write.
- The more complex the implementation the higher danger of mistakes with major impacts. Find and understanding bugs and errors is exacerbated.
- General increased efforts for implementing new features
- Onboarding of new developers is aggravated

**8.6.3.2 Optional Exercises – Value Creation**

**Optional Question**

Think about the value creation process of a car in traditional supply chains and compare it to the value creation process of an app that is developed by a complementor. What are the differences? Try to illustrate both value creation processes to facilitate your analysis.

**Proposed Solution**

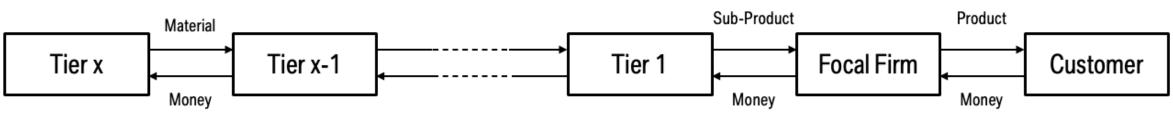



**Figure 19: Value creation in pipeline business**

Traditional value creation in supply chains can be considered as step for step arrangement, where value is transferred from Tier x to Tier x-1 to the focal firm (BMW) and finally to the customer. The value creation in this "pipelines" can also be described as linear value creation. Each transaction on the supplier side is arranged as contracted partnerships, where the amount of value-exchange is predefined. Value creation in pipelines benefits from the economy of scale within this supply chain. The optimization of operations enables the production of more goods with a constant amount of fix costs. Consequently, the costs per produced units decrease.

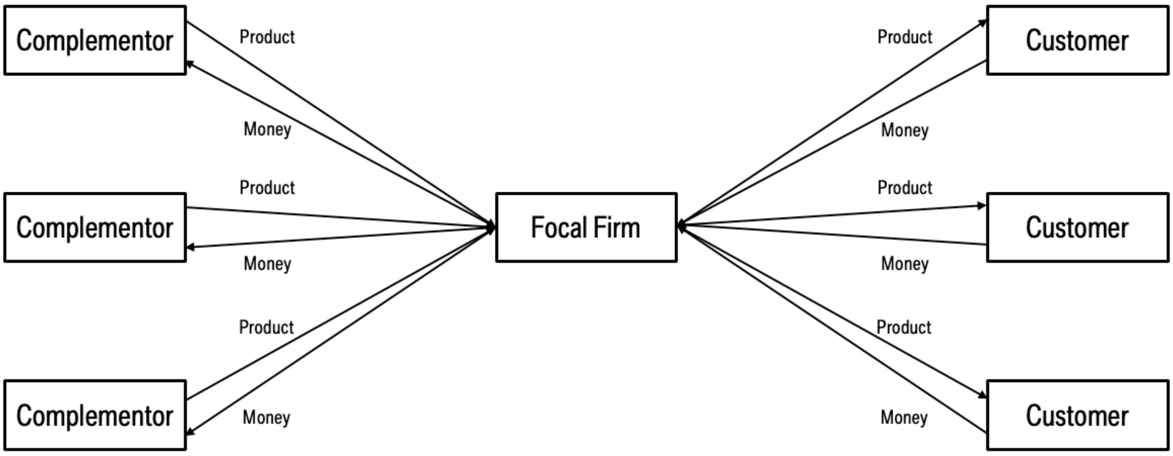



**Figure 20: Value creation in platform business**

Multiple successful businesses still rely on linear value creation. However, when a platform enters the market, the platform "virtually always wins" (Parker, Van Alstyne, & Choudary, 2016). One reason is that platforms create value with resources that they don't own. Consider BMW as large car manufacturer. It requires a globally disturbed set of manufacturing plants to produce round about two million cars a year. Each asset requires high efforts for maintenance, human resources as well as energy costs. Furthermore, a large network of supplier needs to be contracted and managed. All these efforts cause a significant amount of fix costs that are hard to scale. You cannot employ and fire band workers on a monthly or even daily base, just due to changing market demand. Similarly, a new plant itself cannot be build up within one day and supplier contracts need to be fulfilled. Uber, which - similarly to BMW - strives to sell individual mobility to customers, does not own o a single car. It simply connects drivers with customers. The amount of fix costs is reduced to server costs that are required to run their services and can be easily scaled due to cloud technology. In this way, platform business benefit from an economy of scale on the demand side (which is expressed by network effects). Another reason for the dominance of platform businesses is the elimination of gatekeepers that exist in pipelines. A traditional store defines a selection of products that is provided in its shelves and needs to rely on his knowledge and experience that the customers like the selected portfolio. Amazon on the other side can offer every kind of product that was submitted by a registered dealer. Sufficient products will receive good reviews, which again will attract more customers to it. Insufficient products will earn bad reviews and won't be bought by other customers. In this way the market determines by itself, which product will succeed and which will fail. The platform can scale more rapidly since the traditional gatekeepers - the portfolio managers of usual stores - are replaced by automatically provided market mechanisms as Amazon's product reviews.

**Optional Question**

Is BMW Connected Drive a digital platform? Identify at least two pro arguments as well as two reasons why it should not be considered as digital platform.

**Proposed Solution**

| Pro Arguments | Contra Arguments |
|---|---|
| **Modular app disposal and deployment**<br>Considering the technical definition of a digital platform, the platform module in BMW OS 6 and 7 embodies a technical foundation for other complementary products that can be developed on it. Due to the provided update and deployment mechanism the developed complements can be modularly disposed and delivered to the customer side.<br>**Orchestration of app developers**<br>Even though, the platform is not open for third-party developers, the organization within BMW corresponds to a platform owner, embodied by the responsible platform development department and multiple app developers, which are distributed over multiple business lines within BMW. The platform owner orchestrates the app development activities and implements required guidelines control mechanisms as illustrated in chapter 2.4.<br>**Enablement of innovation**<br>The availability of a SDK and modular app development, which is independent from the long-term development cycles of a car, enables the fast and iterative development of innovative products. Especially, the activities of the research departments and the establishment of the BMW Labs initiative illustrates characteristics of an innovation platform.<br>**Projected modes are established digital platforms**<br>The implemented projected modes embody matured digital platforms with a vibrant community of third-party developers, a two-sided market and frequent innovations by new apps that are available in BMW cars. The implementation in BMW cars entails the availability of a digital platform in the product offer of ConnectedDrive. | **Lack of platform architecture**<br>Even though, software engineers are aligning software architecture into layers, a platform architecture is not comprehensively implemented. Consequently, the lack of abstraction reveals frequent challenges for app developers as described in chapter 2.3 of the case. Furthermore, the current technical design impedes scaling business models. For instance, the fact that BMW pays all mobile network traffic impedes the profitable offer for video streaming services.<br>**(Almost) no third-party developers**<br>BMW does not provide any information or documentation that attracts third-party developers. In the latest version of BMW OS, the small number of apps that implement external services as Spotify or Microsoft Office 365 are crafted by BMW software engineers. BMW provides no official partner program or revenue sharing model that could attract third-party developers. Consequently, no two-sided market is established and no cross-side network effects are fostered.<br>**No economy of scale on demand side**<br>While the access for third-party developers is virtually not available, also the customer side is limited. Apps would be exclusively available for drivers of a BMW car. The limitation of access for other customers prevents an economy of scale on the demand side, which is required for a vibrant digital platform.<br>**Strategic focus on pipeline business**<br>Even though, BMW engineers as well as managers emphasize the importance of digital platforms, several business decisions as the removal of apps for automotive technology or the closed character of the BMW Store illustrate a focus on pipeline business. |

**Table 18: Pros and Cons - ConnectedDrive as digital platform**

It is important to mention that BMW ConnectedDrive embodies a marketing term, which describes a portfolio of digital services and products in and around the car. Baseline of this exercise is that there are aspects of ConnectedDrive that reveal platform characteristics, especially from a technical perspective. However, the lack of a comprehensive platform strategy should be revealed.

#### 8.6.3.3 Optional Exercises – Platform Conundrum

**Optional Question**

BMW is generally known as premium brand and technology leader. Think about other car manufacturers that reveal different characteristics. Do their considerations regarding platform options differ? How could their choice affect BMW?

**Proposed Solution**

The development of a comprehensive and high-quality infotainment system with dozens of features occasions a large amount of cost. BMW as a firm that considers itself as technology-leader and premium brand is willing to bear these costs to preserve this image and provide products that satisfy the expectations of its customers. However, the infotainment system is just one of multiple components, which need to meet this requirement and therefore cause large costs in their development. Other OEMs do not strive for such a premium image but aim at the offering of affordable cars for as many customers as possible. Consequently, the reduction of costs is an even larger driver within the organization of such mass producers than it is within BMW. The approach to completely diminish most of the development costs by implementing Google's matured and highly-quality Android Automotive OS should appear promising to these OEMs. While reducing costs on one side, the car itself is upgraded and keeps up or rather overtakes the infotainment features of premium brands as BMW. Google Maps is widely considered as the state-of-the-art solution for navigation apps and the Google Assistant is one of the leading solutions for personal voice assistance. Even though, BMW and other premium OEMs provide similar features, Google's services are not limited to the car but available at many other touchpoints of the user's daily life. Moreover, these services are heavily in use for a long time, which enables Google to learn and improve them since years. Basically, this raises the question why a customer should pay thousands of euros for a BMW infotainment system, when the more desired services are available for free in an even cheaper car.

Furthermore, due to cross-side network effects the implementation of Android Automotive OS by other OEMs does not only affect customers but also developers. Every additional manufacturer that decides for an implementation of Android Automotive OS provides access to more customers, which again attracts more developers for developing appropriate apps. Any other platform ecosystem needs to compete with Android Automotive OS and convince developers for providing an additional app for their platform. Developers need to "multihome" their app (Cennamo et al., 2018). Even if this platform is also based on the Android Open Source Project, the implementation of Google Automotive Services by third-party developers will impede a simply deployment to another platform.

**Additional Information**

The Android Open Source Project provides different shapes of Android, applicable for several domains. Android as OS for smartphones should be the most widely implemented form. While the basic operating system is published under an open-source license and with that free for any kind of utilization, the access to Google's Play Services is limited. The Play Services contain several features for app developers as notification services or transaction management via Google Pay as well as Google's most prominent apps as Search, Maps or the Assistant. They are exclusively available for members of the Open Handset Alliance (OHA), a consortium founded and led by Google (Karhu et al., 2018). All members of OHA agreed to satisfy Google's Mobile Alliance Distribution Agreements in which

Google prescribes guidelines for the utilization of its Play Services. All manufacturers of the most selling Android smartphones as Samsung Electronics, HTC or LG electronics are participating in OHA.

**Optional Question**

How do potential value networks of the three illustrated options look like? Include also other OEMs in your considerations.

**Proposed Solution**

- *Value network for an own BMW platform ecosystem:* Basically, this option illustrates a typical two-sided marketplace. Third-party developers on one side develop apps that are provided to customers on the other side. Transactions between both sides are managed by BMW. Every other OEM with a similar approach requires the developer to create a specific app, which would be available for customers of this other manufacturer. In such a scenario, all OEMs would strive for the establishment of an own platform business.

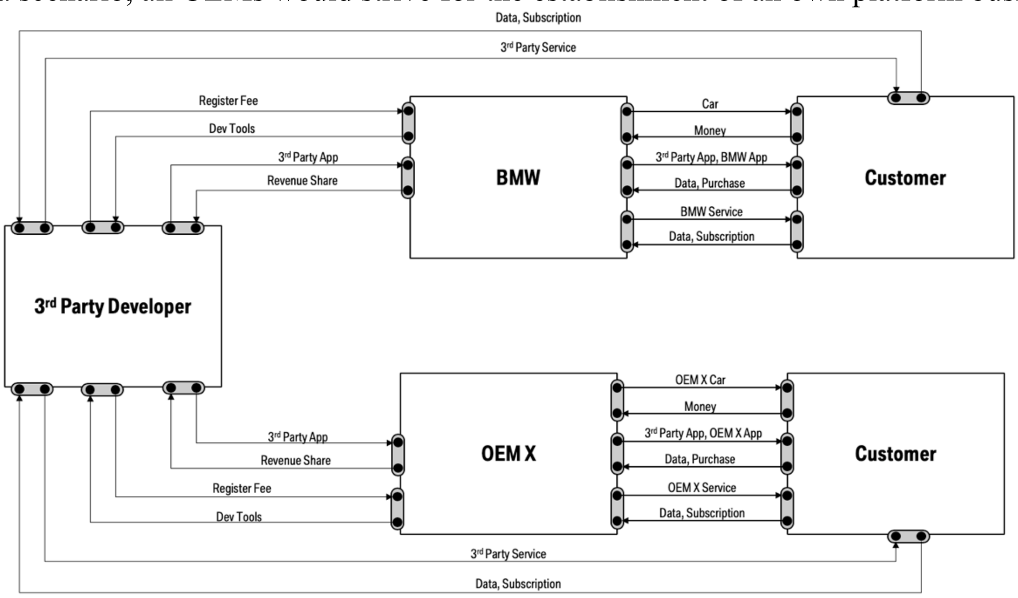



**Figure 21: Value network for an own BMW platform ecosystem**

- *Value network for a collaborative platform ecosystem:* The collaboration with other OEMs for the creation of a common digital platform ecosystem requires the organization in a common institution. Such a platform consortium would consist of all participating OEMs and would take the role of a platform owner. Hence, the consortium would be responsible for the definition of platform governance as well as the guidelines for the technical platform design. Furthermore, all transactions from between developers and customers would be managed by the consortium. The revenue streams would be distributed between all involved OEMs accordingly to previously defined revenue sharing routines. The platform business would be shared by all involved OEMs.

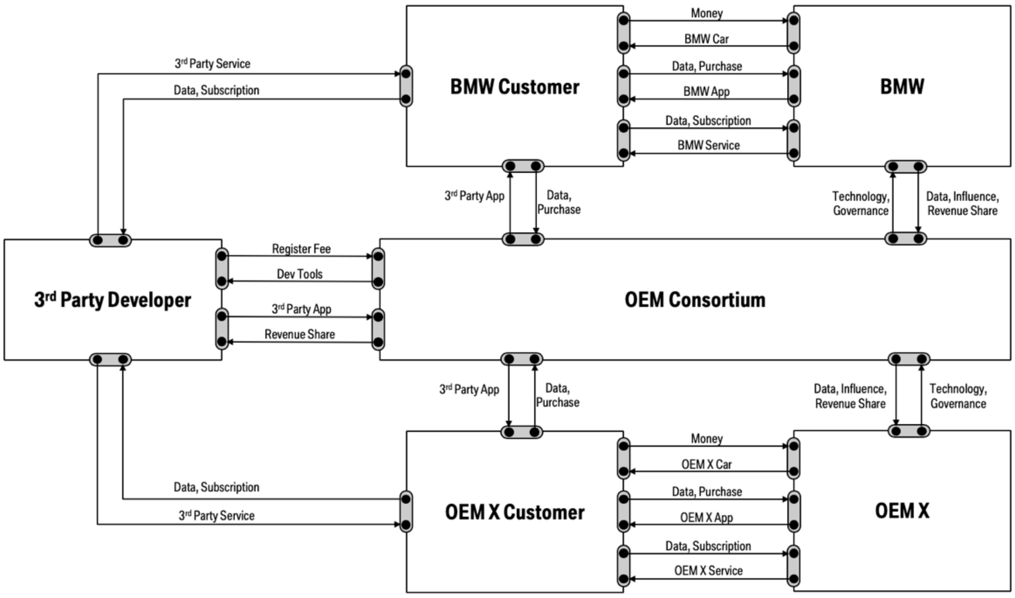



**Figure 22: Value network for a collaborative platform ecosystem**

- *Value network for a platform ecosystem with Google:* The implementation of Android Automotive OS would put Google into the position of a platform owner for automotive apps. All decision on platform governance as well as platform design would be made by Google. Accordingly, Google would manage the complete platform business while the OEMs remain with a traditional pipeline business in form of selling cars.

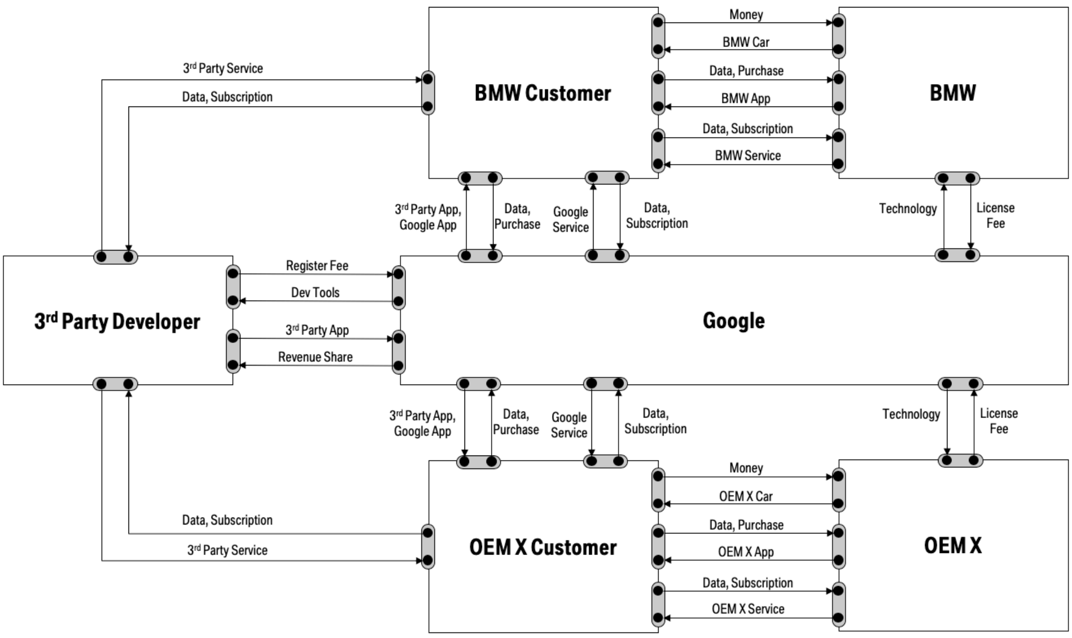



**Figure 23: Value network for a platform ecosystem with Google**

# Part C

# 9 Summary of Results

With the five publications embedded in this thesis, we addressed the three research questions that guided this research endeavor. Subsequently, we summarize the results for each research question before discussing their implications in the next section.

> ***RQ1***: *Which governance mechanisms enable a platform owner to learn digital platform design?*

**Platform owners acquire skills and knowledge through interactions with complementors.** P1 was motivated by a fragmented body of literature on the acquisition of skills and knowledge in digital platform ecosystem. While learning mechanisms of complementors were described, the learning of a platform owner was ignored so far. Therefore, we conducted 15 expert interviews that were subsequently analyzed by the application of grounded theory methods. Our results show that the relationship of complementors and platform owner reveal reciprocal learning effects on both sides (and not just on the complementor side) and point out the importance of learning mechanisms for an unexperienced platform owner. The platform owner's acquisition of knowledge occurs through three different levels of interaction: Reverse broadcasting, reverse brokering, and reverse bridging.

**Three learning mechanisms enhance the design of an emerging digital platform.** Considering our findings that a platform owner learns from interactions with complementors, we conducted a longitudinal study to investigate the characteristics of such learning mechanisms. Based on 30 expert interviews that were conducted during a period of two years in which BMW developed and launched their digital platform for automotive onboard apps, we identified three learning mechanisms that enabled the platform team to enhance the design of the considered digital platform (P2): The transfer of perspective (1), the transfer of knowledge (2) and the transfer of artefacts (3) enabled the platform owner to improve their digital platform and enhance value creation by complementors. Thereby, we identified platform boundary resources as central medium for such learnings and recognized that the learning is enabled by an appropriate level of platform openness, which needs to be orchestrated and adapted by the platform owner.

> ***RQ2***: *How can a platform owner involve complementors to enhance digital platform design?*

**The involvement of lead complementors can enhance the design of a digital platform.** Building on our findings that platform design benefits from interactions with complementors (RQ1), we conducted an action research study that strived for the active involvement of complementors in the design of the considered digital platform (P3). In this way the platform owner wouldn't be reliant on arbitrary interactions with complementors but is able to consciously drive the enhancement of certain platform design aspects. In our case, the platform owner focuses on the refinement of platform APIs, which were evaluated as badly designed by several complementors. The study reveals that just a small fraction of complementors is able and willing to get involved. Comparable to the concept of lead user involvement, just especially engaged complementors with a strong need for improvement are willing to collaborate. However, our findings illustrate that the involvement of such lead complementors can enhance the design of the digital platform. In our iterative action research approach, the design of five different APIs was refined, containing 59 methods and 31 properties that were frequently implemented by several app development teams. All refined APIs were evaluated as improvements by complementors, embedded in the platform SDK and eventually just by dozens of app development teams.

> ***RQ3***: *What are options on digital platforms for incumbent firms that are confronted with the transformation of their products into a digital infrastructure?*

**Three strategical options for incumbent firms that are confronted with the emergence of platform ecosystems for their product.** Based on 42 expert interviews that were conducted within the period of two years during the emergence of different digital platform ecosystems for automotive onboard apps, we identified three strategical options for an incumbent firm as BMW that is confronted with the transformation of their cars from a traditional product to a digital infrastructure (P4). First, BMW could build its own, proprietary digital platform and strive for a digital platform ecosystem that exclusively controlled by the company. Second, BMW could engage in a collaboration of automotive manufacturers to build a collaborative platform ecosystem that is equally controlled by all involved OEMs. Third, BMW could join Google's existing Android ecosystem that was extended to the automotive context.

**Challenges and chances of BMW as an incumbent firm that is confronted with the platformization of its product.** Building on our insights on strategical options for BMW and the discourse within the company on the right strategical decision, we decided to describe the inherent challenges in a teaching case that illustrates the "platform conundrum" that many incumbent firms face in the age of digital platforms (P5). Thereby, we describe the history of ConnectedDrive as BMW's infotainment system, the traditional way of software development for cars in which ConnectedDrive was usually developed and how these procedures are affected by the introduction of digital platforms. Students learn market- oriented as well as technology-oriented perspectives on digital platforms and find these reflected in the described chances and challenges of the three strategical options that BMW needs to evaluate (see P4: Build proprietary platform, build collaborative platform, join existing ecosystem).

Table 19 provides an overview on the key findings of this thesis.

| P | RQ | Findings |
|---|---|---|
| P1 | RQ1 | ▪ Interactions of platform owner and complementors enables the acquisition of knowledge by the platform owner on three levels<br>o Reverse broadcasting enables learning in an arm's length relationship between platform owner and the complementor community<br>o Reverse brokering enables learning by interactions between platform owner and a fragment of complementors<br>o Reverse bridging enables learning by direct communication between one complementor and the platform owner<br>▪ Learnings of the platform owner enables enhanced platform design |
| P2 | RQ1 | ▪ There are three learning mechanisms that enable enhanced platform design: Transfer of perspective, transfer of knowledge and transfer of artefacts<br>▪ Platform boundary resources are the central medium for learnings of the platform owner that are enabled by interactions with complementors<br>▪ Learning by the platform owner needs to be enabled by an appropriate degree of platform openness |

| | | |
|---|---|---|
| | | ▪ The platform owner needs to orchestrate complementors to enable learning |
| P3 | RQ2 | ▪ Lead complementors with strong needs for improvement engage in the design of digital platforms<br>▪ The active involvement enables the platform owner to enhance dedicated aspects of platform design<br>▪ The involvement of complementors can be beneficial for the design of the digital platform |
| P4 | RQ3 | ▪ The penetration of digital technology as connectivity and computational capabilities into traditional products facilitate the emergence of digital platform ecosystems<br>▪ There are three strategical options for incumbent firms that face the platformization of their product: Build a proprietary platform ecosystem, build a collaborative platform ecosystem, or join an existing platform ecosystem |
| P5 | RQ3 | ▪ Illustration of the development of the car's infotainment system from a traditional product to a digital infrastructure<br>o Market-oriented perspective on platformization<br>o Technology-oriented perspective on platformization<br>▪ Illustration of challenges and chances of the available strategical options that are available for incumbent firms that are confronted with the platformization of their product. |

**Table 19: Overview on key results**

# 10 Discussion

The following chapter contains discussion based on the presented results that are of interests with regard to the prevalent body of knowledge on digital platforms. We propose the introduction of platform emergence as additional phase in the lifecycle of a digital platform. Furthermore, we consider the collaboration of complementors and platform owner as valuable addition to the understanding of this relationship, which was previously coined by competition and the struggle for control. Eventually, we emphasize the heterogeneity of complementors and the relevance of this aspect for platform governance.

## 10.1 Platform Emergence as Additional Phase in the Digital Platform Lifecycle

By exploring the nature of a digital platform before its actual launch we expand prevalent knowledge on the platform lifecycle. While prior research focused on the nature of matured digital platforms, there were little insights on the actual emergence of platforms (de Reuver et al., 2017). Parker, Van Alstyne and Choudary (2016) describe the startup phase as phase in the platform lifecycle in which the platform owner strives to establish initial network effects. Therefore, the platform needs to be accessible for complementors and users. However, our research illustrates activities previous the actual launch that affect technical-oriented, the market-oriented as well as the socio-technical aspects of the platform and consider how a digital platform is actually designed (de Reuver et al., 2017).

Considering a technical perspective, a platform owner needs to configure a optimal platform architecture. P3 illustrates that bad API design frustrates and frightens off complementors. A platform owner needs to consciously design boundary resources to be used by complementors. Otherwise, no value creation and no subsequent value exchange with network effects will happen (Stummer et al., 2018). While app developers were part of the platform owner organization and were paid to build their apps, they reported that the platform's bad API design would hold them off to build an app for BMW as an external complementor. P1 and P2 illustrate in which way a platform owner can exploit interactions with internal complementors to acquire the required capabilities to avoid insufficient platform design. While these learnings might also take place in the platform startup phase, when the platform is accessible for external complementors, we argue that it is more effective if the platform owner is not forced to nurture initial network effects in parallel. Potential mistakes in platform design are not punished by complementors leaving the platform ecosystem and the platform owner might iterate on platform design with less pressure. While this proposition requires proving by future research, the prevalent learning effects were observed at a digital platform that was in the platform emergence phase and not available for external complementors.

Focusing the market-oriented perspective, P4 and P5 illustrate that choosing an existing or creating a new digital platform ecosystem embodies a major challenge for a potential future platform owner. The decision needs to be made before the startup of the platform. Therefore, the future platform owner needs to understand which aspects of their platform reveal potential for a "core interaction" that allows the nurturing of a platform ecosystem (Parker, Van Alstyne, & Choudary, 2016). We understand the evaluation of market potentials and for the nurturing of future network effects as part of the platform emergence that requires further investigation. However, P4 and P5 illustrate dynamics and considerations of an organization that endeavors becoming a platform owner.

An inexperienced platform owner needs to acquire capabilities to orchestrate complementors (Schreieck et al., 2021; Svahn et al., 2015). Our research illustrates that this socio-technical perspective is especially relevant for incumbent firms that are used to control suppliers and now needs to learn how to orchestrate third-party complementors that don't maintain a direct relationship to the platform owner. While the acquisition of capabilities that are required to orchestrate a digital platform ecosystem might continue during the whole platform lifecycle, we argue that these are especially relevant during platform startup. A lack of platform owner skills provokes any solving of a chicken-egg problem to nip in the bud. Therefore, we argue that an unexperienced platform owner needs to learn such skills during platform emergence, which again emphasizes the need for a better understanding of this earliest phase in the life of a digital platform.

While there is comprehensive research on characteristics and effects of platform openness (Eisenmann et al., 2009; Ondrus et al., 2015; West, 2003) there is little evidence of what conditions must be met for a digital platform to be open to external complementors. Establishing a digital platform inside an organization appears to be a good practice to elaborate the design of the digital platform before opening it to external complementors. "Opening" means moving from platform emergence to platform startup. While the platform owner needs to focus on sufficient platform design (from a technical-oriented perspective, market-oriented perspective, and socio-technical perspective) in the platform emergence phase, they might ignore the nurturing of network effects. However, the platform design should be trimmed to foster network effects as soon as the digital platform leaves the initial phase of platform emergence and enters the second phase of platform startup.

## 10.2 Collaborative Interactions of Platform Owner and Complementors

By illustrating that the relationship of platform owner and complementors is not exclusively competitive but reveals also collaborative characteristics, we contribute to the socio-technical perspective on digital platforms. While research on platform governance focuses on "partitioning of decision-making authority between platform owners and app developers" (Tiwana, 2014), we argue that the relationship between platform owner and app developers reveals more aspects than the pure fight for control (Song et al., 2018). Especially during platform emergence an unexperienced platform owner needs to understand the needs of complementors (and users) to increase chances for solving the chicken-egg problem (Stummer et al., 2018). While we do not neglect the relevance of control in a platform ecosystem, we argue that research on the relationship between complementors and platform owners needs to be complemented by the aspect of collaboration.

P1 and P2 illustrate that a platform owner benefits from interactions with app developers by different learning mechanisms that benefit both parties. The resulting improvements of platform design contribute to the general value creation inside the platform ecosystem, independent from the distribution of value capture between platform owner and app developers. In this way our research emphasizes the exploitation of platform governance to nurture additional value creation instead of the fight for control on already existing value creation (Henfridsson & Ghazawneh, 2013). Thereby, we build on existing research on interactions between platform owner and app developers. However, while Foerderer et al. (2019) illustrate knowledge transfer from platform owner to app developers, our research sheds light on learning effects at the platform owner side. The results of P2 provide strong incidents that improvements of platform design were caused by learnings in the platform owner organization, which were caused by collaborative interactions with complementors. Furthermore, the results of our action research study in P3 prove that the acquisition of new skills by the platform owner does not just happen by chance through interactions with complementors but can be actively promoted.

In summary, our research implicates that the enablement of collaboration between app developers and platform owner can be influenced by proper platform governance. This observation has several implications. For instance, the degree of platform openness (Ondrus et al., 2015) was recently considered as lever for control of the platform owner. The lower the openness the more control by the platform owner. However, our results show that the level of control is not the only aspect that needs to be considered by the platform owner. The less open the platform design, the less opportunities for collaborative interaction. A platform owner needs to find (and constantly adapt) a platform configuration that allows the collaborative interactions with complementors while avoiding the risk of losing control on the own digital platform. Considering this endeavor, the importance of platform boundary resources' design takes on an even more significant role. While they were mainly considered for resourcing and controlling complementors (Henfridsson & Ghazawneh, 2013), our research expands their importance as a medium that facilitates collaborative interactions. In addition, our insights shed new light on the boundary resource "tuning" phenomenon described by Eaton et al. (2015). P3 describes that the design of platform boundary resources is not just a result of tuning effects that occur during the platform owner's and app developers' striving for control on platform resources but may also a result of collaborative interaction. We believe that both influences on boundary resource design can coexist and be alternately dominant. Thereby, the when and why collaboration dominates over competition or vice versa appears as promising field for future research on the design of platform boundary resources.

### 10.3 Complementor Heterogeneity as Aspect of Platform Governance

By investigating interactions of individual complementors with the platform owner (P1 - P3), we enhance knowledge on platform governance. As we argued before, there is a lack of research on the design of digital platforms with regard to the requirements of complementors. However, an ongoing area of interest within platform research is how platforms interact with complementors in a broader sense (Cennamo & Santaló, 2019; Eaton et al., 2015; Kapoor & Furr, 2015; Rietveld et al., 2020). Often, the underlying assumption is that complementors are homogenous and that they are attracted to digital platforms based on indirect network effects: the larger the potential market size on the side of the users, the more attractive the platform. However, this assumption changed in the recent years. A significant body of research has emerged, for example, showing that "superstar" complements play a disproportionate role in technology adoption (Binken & Stremersch, 2009; Rietveld & Schilling, 2021). Other research looked at the effect of complement quality on technology adoption (Kim et al., 2014), and in which way it can be triggered by impulses of the platform owner (Förderer, 2020). Such research shows the heterogeneity of complementors regarding their capabilities and indicates the need to consider the differences among complementors when designing a platform. There might be complementors that exhibit more expertise, motivation, or experience, while others have little previous experience and might need more guidance.

We add to literature on platform governance that has called for considering complementor heterogeneity (e.g., Huber et al., 2017; Rietveld & Schilling, 2021). Regarding platform design, complementor heterogeneity not only covers differing expectations and requirements for platform design across complementors but also different expertise of complementors in specific areas of the platform. We show that the platform owner can build on that expertise by involving lead complementors in platform design, in particular with regard to the design of the platform's boundary resources. It is thus more promising to develop boundary resources with input from lead complementors than in a top-down manner. This insight adds to previous work on boundary resources that has focused on the role of boundary resources rather than the process how they are designed (Förderer et al., 2019; Ghazawneh & Henfridsson, 2013). In sum, we argue that, as part of platform governance, platform owners can consciously exploit lead complementor involvement to improve platform design. Future research on platform governance could explore further opportunities to leverage the heterogeneity of complementors. For example, surveys among complementors could be used to analyze differences between complementors in a more systematic way and to compare these differences to the different partner categories that many incumbent companies have established to manage interorganizational relationships (e.g., Iansiti & Lakhani, 2009).

# 11 Implications

## 11.1 Implications for Theory

The results of this cumulative dissertation aim to contribute two major streams of IS research, which are illustrated in the following sections. First, we contribute to research on digital platforms and second, we contribute to research on digitalization in incumbent firms.

### 11.1.1 Contribution to Research on Digital Platforms

By conducting an extensive case study in the context of an emerging digital platform, we shed light on the endeavors of a new platform owner that needs to craft an initial platform architecture while establishing platform governance. Thereby, we contribute to the ongoing discussion "how a successful digital platform can be designed" (de Reuver et al., 2017). While the success of the considered platform needs to be proven, our work provides unique insights on the actual birth of a new digital platform and the inherent dynamics. Thereby, we contribute to the existing understanding of the platform lifecycle. While Parker et al.'s model of the platform lifecycle begins with the actual launch of the digital platform, we argue that the initial platform design is done beforehand. Fundamental decisions on platform architecture and platform governance must be made to expose first boundary resources to complementors (Henfridsson & Ghazawneh, 2013). This phase in the platform lifecycle was mostly ignored so far. However, we argue that it reveals major influence on the success of the subsequent platform life. If the initial platform design reveals major flaws that impede the creation of value in the platform ecosystem, the platform will diminish without even beginning to solve the chicken-egg problem (Stummer et al., 2018). Our research illustrates the struggles and efforts of a new platform owner that is faced with the challenge to create the initial design of an emerging digital platform. By the application of organizational learning theory, we introduce three mechanism that enable this platform owner to acquire knowledge on needs of complementors in a digital platform ecosystem (P1 & P2). Thereby, the aspect of collaboration emphasizes another theoretical contribution of our work. While previous research assumed a mostly competitional relationship between platform owner and complementors (Eaton et al., 2015), our research illustrates the power of collaboration between both parties (P1 – P3). By the application of action research, we prove that this dynamic is not arbitrary occurring but can be consciously initiated and exploited by the platform owner. "Lead complementors" might be even involved in the actual design of new platform resources (P3).

### 11.1.2 Contribution to Research on Platformization in Incumbent Firms

The prevalent case study of the BMW Group, a traditional car manufacturer that needs to manage the platformization of its product, embodies an illustrative example of an incumbent firm that is challenged by the platformization of its product. Therefore, the company needs to acquire new capabilities (Selander et al., 2013; Svahn et al., 2015; Svahn et al., 2017) to integrate the value proposition of a digital platform ecosystem into their product (Sandberg et al., 2020; Stonig et al., 2022). Prevalent literature describes the inherent challenges that come along with such an endeavor (Hein, Schreieck, et al., 2019a; Weiss, Schreieck, Brandt, et al., 2018). By the illustration of BMW's activities around the platformization of their cars, P1 - P3 describe different ways of the company dealing with the emergence of a digital platform for their product. These insights add to previous work on platformization that has rather focused on inherent challenges instead of ways of dealing with such. By implementing organizational learning theory in the context of a digital platform governance, P1 and P2 illustrate the acquisition of new capabilities for platform governance (Svahn et al., 2017). P3 further develops the concept of a learning platform owner by actively involving complementors in the design of the platform through the application of lead user involvement methodologies (von Hippel, 1986). Considering the strategical perspective on the emergence of digital platform ecosystems in the automotive industry, P4 illustrates the application of real option thinking within the BMW Group. Even if the approach has been discussed several times in theory (Constantinides et al., 2018; Rolland et al., 2018; Tiwana, 2014), there is still a lack of empirical studies on how companies manage uncertainty with real option thinking. While Svahn et al. (2015) provide an example of an incumbent firm that utilizes option-thinking to shape generativity insight their ecosystem, we focus on the competing ecosystem approaches in the automotive industry. P5 provides detailed insights on BMW's inherent strategical challenges and illustrates pros and cons of different options. Thereby, we contribute to prevalent literature on the platformization in incumbent firms  and provide a starting point for future research on the platformization of cars as a new digital infrastructure (Tilson et al., 2010).

## 11.2 Implications for Practice

The prevalent work was carried out as part of an industrial PhD program at the BMW Group. Subsequently, it is closely intertwined with practical issues of an incumbent firm that faces the platformization. It reveals manifold implications for practitioners, which can be split up into three major directions. First, our work provides implications for software engineers that need to align the design of a digital platform with an existing product. Especially, the concept of a learning platform owner might support inexperienced firms to craft a proper digital platform design. Second, we elaborate business related insights that might support product managers to work with digital platforms. P4 and P5 reveal caveats and risks of misunderstood businesses principles in platform ecosystems. Third, we illustrate option thinking as strategy for the management of uncertainty in the context of the emergence of digital platform ecosystems. In particular, the presented teaching case (P5) explains backgrounds of various platform options that are relevant for BMW as an incumbent firm.

### 11.2.1 Alignment of Product and Digital Platform

The car can be seen as one of the most complex mass-produced products of our time. Dozens of control units must work in concert with various mechanical components such as the engine, the gearbox, or the brakes. Many different systems must always work together reliably under various conditions. The failure of safety-relevant functions such as a brake booster or the engine control can mean great danger for the occupants of the vehicle as well as its environment. Therefore, despite the annual production of several million vehicles in a company like BMW, every single product must meet the highest quality standards. The organization of the automotive OEMs, their development, and production are optimized to meet these requirements on a large scale. Our work shows that some things in the implementation of a digital platform seem to contradict the classic approaches in an automotive company. P5 illustrates various examples of how BMW struggled to get away from traditional approaches in the past 10 years and ignored the basic principles of digital platforms when embedding digital platforms in its product because they contradict the established approaches. We conclude two implications from this observation.

First, it shows the importance of an understanding of digital platforms in traditional companies that are faced with platformization. If the safeguarding of each individual complementary app is carried out using the same methods as the traditional quality management of vehicle components, scaling is prevented. Instead, app developers must be made transparent from the start by means of extensive documentation in which framework they are allowed to create innovations. Instead of the usual specification of a component, complementors need a framework that clearly delimits what is permitted and what is prohibited in a platform ecosystem. Before an app is released, the focus should eventually be on whether the app developer has violated previously defined rules. The rejection of an app which formally fulfills all requirements will destroy trust of the app developer in the platform ecosystem and will eventually lead to renunciation. In addition, the inexperienced platform owner must understand that the quality of the platform boundary resources provided directly influences the quality of the complementary apps. P3 shows that bad API design slows down app developers in their work and can put less focus on quality.

Our second implication relates to the implementation of the digital platform in an existing product. In contrast to a platform that is built on the green field, the nature of the product must be considered when implementing a digital platform in an existing product. On the one hand, the digital platform should enable unbound innovation. To do this, the OEM must give up a certain amount of control, as just described, and apply new approaches to quality management. However, these new practices must never jeopardize the reliable functionality of the driving functions. The more complex the nature of the product, the more attention must be paid to the implementation of the digital platform. The more sensors and actuators are built into a product, the more potential for innovation by third-party app developers who could integrate the components through appropriate APIs. However, the more APIs are available, the greater the potential for willful or involuntary abuse. While the integration of an app store on TVs also required new approaches for manufacturers, for example for quality management, a malfunction of a TV set poses far less danger than a car that no longer works properly while driving. Even supposedly trivial functions such as playing back sound can be potentially dangerous. Spontaneous noise can frighten and distract the driver while driving and cause dangerous situations

### 11.2.2 Exploitation of the Platform Business

Beside the technical expertise that an unexperienced platform owner needs to acquire to build a successful digital platform, the business perspective on digital platforms should be understood as well. Our teaching case P5 illustrates in which way BMW tried (and failed) to implement principles of a digital platform while sticking to traditional product business models.

The traditional business model of premium automakers focuses on the sale of high-quality vehicles. The more functionality is added to the product, the higher the price that can be asked. Therefore, the organization is optimized to build and sell a highly refined product, which delivers profit as it is sold to the customer. In this classic business model, the configuration and ordering of the vehicle is the only opportunity for a transaction between the app developer and the user[39]. As soon as the vehicle is delivered, it remains in the configured state. However, a successful platform ecosystem should strive for maximizing the number of potential transactions to nurture network effects and increase the potential for value capture. Therefore, digital complements should be purchasable at every point in the product lifecycle. The aftersales of digital functionalities as apps requires that a customer can purchase new complements whenever they desire it.

Also, traditional optional equipment as a seat heating or sports suspension can be considered as complements that enhance the basic product. However, these physical complements are bound to the respective car that was initially configured and sold. When the customer decides to resell the vehicle, the price will contain the remaining value of additional equipment and a new car would require paying for the options once again. However, digital complements are not physically bound to the product. They can be detached at any time and be transferred to a new product with any additional costs. In the smartphone world, customers expect that all installed apps are transferred to a new smartphone as part of the initial configuration (as long both phones live the same platform ecosystem). However, if the OEM relies on selling optional equipment every time a new car is sold and the customer is now able to transfer equipment from car to car, alternative business models need to be explored.

While the previously illustrated implications of business aspects are relevant for digital platforms, they ignore if digital complements are built by the OEM or third-party developers. Complementary apps that are sold after the actual handover of the car to the customer could also be exclusively provided by the car manufacturer. However, maximizing the number of transactions in the platform ecosystems involves the opening of the digital platform for external complements. Therefore, in addition to already mentioned technical and security aspects, economic aspects should also be considered. The opening of a previously closed API might disrupt existing businesses of the OEM. For example, APIs that provide information on the battery status of an electric car might enable external and disrupt OEM-exclusive charging services. Our results show that most of the discussions about opening APIs within a traditional company revolve around how the functionality of the product can be most attractively improved by external complements. However, even if the disruption of existing business models is deliberately accepted, the decision that this happens with the publication of an API should be made consciously. Furthermore, the OEM should find business models that benefit from external innovation. As soon as the manufacturer captures a share of the revenue of the external charging provider, replacing its own services might even be more profitable. While these considerations might differ between APIs, we argue that business aspects require major attention while opening interfaces of a traditional product.

### 11.2.3 Management of Platform Uncertainty

The "platform revolution" (Parker, Van Alstyne, & Choudary, 2016) affects several industries that were built on traditional product business models. The emergence of digital platforms causes changes in these industries, which unsettles affected incumbent companies. Difficulties to predict future developments of technology and business reveals uncertainty that requires

---

39 It needs to be acknowledged that the aftersales of services is an established business in the automotive industry. Corporate workshops repair and maintain vehicles and charge the customer for these services. However, this business focuses on the maintenance of already sold, physical products not on selling digital complements. Furthermore, since most of the profit is earned by the dedicated workshops it barely affects the OEMs core business, namely selling new cars.

strategical management. The prevalent case of BMW represents a typical example of an incumbent firm that is confronted with the platformization of its product. It provides manifold learnings for other companies that find their selves in a similar situation.

Especially P5 shows that many participants within BMW were aware of the unused potential regarding the existing digital platform(s). However, it was also revealed that the understanding of a platform differs due to the novelty of such a construct in the organization. The prevailing proprietary systems require different mindsets. Furthermore, the high granularity of the work steps ensures many different perspectives regarding the requirements of a digital platform. After all, the function-driven development culture stands in contrast to the idea of a digital platform. Establishing a common understanding of the platform is a measure that can help overcome the challenges mentioned. Specifically, this could imply the development of a uniform terminology. A common understanding of platform-related terminology facilitates communication and reduces the likelihood of misunderstandings. In the event of a problem on the part of an application developer, a common and precise vocabulary facilitates a concrete description of the problem and thus accelerates support from experts, for example from the platform team. The platform team, on the other hand, can fall back on the established terminology when creating documentation or release notes and thus also reduces the likelihood of misinterpretation by app developers. However, it is not just a common use of language that is conducive to a common understanding of the platform. The definition of a clear and generally understandable platform strategy also appears to be excellent. It enables the derivation of specific goals for individual departments and even employees. Activities for the further development of the platform can thus be orchestrated and prioritized, concurrent activities can be merged, and counterproductive or superfluous activities can be identified more easily. It seems important that the platform strategy and its derivations are the responsibility of a central authority. This enables stringent and persistent handling. In the present case of BMW, specific measures relating to the platform were derived, based on the company-wide digitization strategy. Different entities in the company derive - decentral distributed - different interpretations from this abstract strategy. This approach harbors the risk of inefficient and uncoordinated action regarding the platform.

# 12 Limitations

The studies presented in this thesis and, consequently, the overall thesis reveals several limitations that are presented in the following section.

First, due to the conduction of a **single case study**, the specific context of the BMW Group is determining. A single case study leads to an increased impact of the case company on the results as not as generic results are acquired as when utilizing a multiple case study design. Therefore, the application of the established results for any incumbent company is not recommendable although a profound consideration is possible. Additional evaluations could validate the collected results. Furthermore, multiple manufacturers and suppliers embody the automotive industry. The unfettered appliance of the collected results towards any incumbent enterprise is not recommendable. Further evaluations of the findings could validate the obtained value. However, a reflected consideration is necessary.

Second, the finite **period of our research** limits the knowledge gained. Even if the research stretched over a total period of four years, it must be recognized that the automotive industry operates in long-term cycles. The life cycle of a single vehicle model can be assumed between six and eight years. P5 illustrates that digital platforms have been influencing decisions and actions within the BMW Group for more than a decade. While we were able to observe the establishment of a digital platform for automotive apps and various considerations as to how this platform could also be made available to external complementors were observed, this opening was not applied during our research. Considering Parker et al.'s model of the platform lifecycle (2016), the investigated platform never entered the startup phase, and the success or failure of the platform design could not be proven. You could argue that BMW never transferred its existing product business to a platform model but focused on enhances its existent model of selling cars by digital platforms. Other incumbents might have been forced to really replace their product business by a platform business. We consider the investigation of different platform approaches of incumbent firms as promising opportunity for future research (see 13 Future Research). Nevertheless, we were able to examine challenges and possible solutions in the previous phase of the platform emergence, which was the focus of our research. Furthermore, our collected data is based on interviews with experts who have been dealing with the matter over a longer period.

Third, our research is subject to limitations on **generalizability and reporting bias**, which is inherent to our qualitative research approach (Yin, 2009). While we need to acknowledge the specific conditions of our research, as the engagement in BMW's PhD program, we still strive for generalizable results that contribute to the scientific discourse on digital platforms. Addressing this challenge requires a context-sensitive approach that allows the application of the yielded results, also in other contexts. In our case, an attempt was made to include as wide a range of experts as possible in the data collection to include all perspectives of the examined facts. Furthermore, through the extensive collection of secondary data, we gained the ability to triangulate the collected data. While we cannot, completely free ourselves from the accusation of generalizability and reporting bias, all the studies presented show a comprehensive description of the methods used for data collection and analysis, which should make it possible to carry out the studies in other contexts as well.

Eventually, the basis of all work in this thesis relies on a sound review of prevalent literature. Even though it was conducted in a thorough and diligent way, **we rely on the work of earlier researchers**. Further the access to literature was limited by the provided admission of the Technical University of Munich and the timeframe of the treatment. The rapid advancements of technology and research indicate an early overhauling of the findings.

# 13 Future Research

Throughout our research on the emergence of digital platform in the context of incumbent firms several new research questions emerged, which are out of scope of this thesis and provide fruitful avenues for future research. We identified five areas that appear especially promising for a better understanding of digital platforms.

| Area of research | Exemplary Research Questions |
|---|---|
| Broader Investigation of Platform Emergence | • What different types of platform emergence do exist?<br>• What factors influence the design of a digital platform during platform emergence?<br>• How can a platform owner define platform governance during platform emergence? |
| Learning Mechanism of Platform Owners | • How does learning of platform owners affect the design of matured digital platforms?<br>• What learning mechanisms of platforms owners can be identified in matured platform ecosystems?<br>• How do learning mechanisms evolve during a platform life cycle? |
| Transformation from Platform Emergence to Platform Startup Phase | • What factors influence the decision of a platform owner to start a digital platform?<br>• What capabilities are required for the transformation from platform emergence to the platform startup phase. |
| Consortium-owned Digital Platforms | • What are advantages and disadvantages of consortium-owned digital platforms in comparison to digital platforms that is owned by a single entity?<br>• What are characteristics of consortium-steered platform governance?<br>• How are decisions made in a consortium that owns a digital platform? |
| Strategical Management of Uncertainty in the Context of Platform Emergence | • What capabilities are required to manage uncertainty in the context of digital platforms?<br>• How can an incumbent firm transform existing relationships in the context of platform emergence? |

As already mentioned, the single-case character of our research reveals one of the biggest limitations to our research. Therefore, we argue that our insights on the **emergence of digital platforms needs to be proven in a broader scope**. While our findings are likely to apply to other premium car manufacturers, the situation may be different for other types of automotive firms. For example, vehicle manufacturers whose business model is aimed less at high-priced products but at high sales volumes usually spend way less budget in their development departments. The lack of resources may impede the acquiring of platform owner capabilities and result in different strategical management of platform emergence. Furthermore, it has already been mentioned that the automotive industry is an industry with very long-term cycles. The low velocity of all competitors gives a company like BMW enough time and space to build platform management capabilities. Investigating platform emergence in the context of faster moving domains or startups represents a promising avenue for future research on digital platform. Especially the application of multi-case studies with cross-case comparisons (Yin, 2009) could provide valuable insights in the emergence of digital platforms.

Second, we propose to **apply the idea of a platform owner learning from complementors to other phases in the platform lifecycle**. While our investigations have focused on the learning processes of a new platform owner, we advocate investigations of the cooperation between platform owners and complementors in mature platform ecosystems. This goes for passive learning mechanisms as investigated in P1 and P2, as well as the enhancement of platform design by the active involvement of complementors in P3. While the necessity seems particularly high for incumbent firms that are just starting out with the endeavor of a digital platform, the feedback from complementors regarding new platform functionalities can also be a valuable source of information for experienced platform owners. We believe that knowing how and when and why complementors and platform owners collaborate is key to better understanding good platform design.

Third, we argue that scholars should **investigate the transformation from platform emergence to the platform startup phase**. While there is comprehensive research on platform openness in matured platform ecosystems (Ondrus et al., 2015) the is no knowledge on criteria when to start a new digital platform. During the period of our research, the BMW Group did not manage to open a digital platform for complementors (P5). The reasons for and against such an opening as well as the idea of phased opening approaches provide a promising area for future research.

Fourth, we consider the **investigation of consortium-owned digital platforms** as fruitful avenue for future research. P4 and P5 illustrate that the collaboration with other incumbent firms could provide a solution for the strategical challenge that BMW and other traditional car manufacturers were facing. However, while consortium-owned platform ecosystem are occasionally mentioned in literature (e.g. Hein et al. (2020)) there no scientific description of such a construct is prevalent to our knowledge. Especially investigations of platform governance and control in a consortium-owned digital platform appear as promising fields for new theoretical insights.

Eventually, we see demand on **researching the strategical management of uncertainty in the context of platform emergence.** While P4 and P5 apply real option thinking to illustrate three available options for an incumbent firm that is confronted with the platformization of its product, we argue that the fundamental understanding of uncertainty management needs to be re-considered in the context of digital platform ecosystems. Traditional strategic management literature argues that firms explore existing partnerships during uncertainty (Beckman et al., 2004). However, the emergence of digital platform requires the re-organization of these partnerships (Parker et al., 2017). This paradigm shift offers several opportunities for future research that should help to better understand how companies can develop a product-centric business model towards a business that exploits digital platforms.

# 14 Conclusion

Motivated by the enormous impact of the "Platform Revolution" in many established industries, this thesis strives for a better understanding of the emergence of digital platforms from the perspective of an incumbent firm. By conducting a single-case study at the BMW Group, a well-established, globally operating car manufacturer, it was possible to gain various insights on the endeavors of an unexperienced platform owner that is challenged with the design and management of a new digital platform for automotive onboard apps. Our findings contribute to literature on platform governance, platform emergence and platform design. In particular, (1) we show different learning mechanisms of an inexperienced platform owner, (2) prove the positive contribution of the complementor involvement to the platform design and (3) show different strategic options that a traditional company has available when confronted with the platformization of its product. We hope that this thesis inspires future research on platform design and platform emergence, especially in the context of incumbent firms. In addition to our theoretical contributions, our work also offers valuable insights for practitioners who are confronted with the challenges described. The emergence of digital platforms is causing fundamental shifts of long-dominant structures in established industries. We are confident that the findings of this thesis may support managers as well as engineers in incumbent firms to understand and exploit the characteristics of digital platform ecosystems for their future business.

# References

Adelman, C. (1993). Kurt Lewin and the Origin of Action Reserach. *Educational Action Research, 1*(1), 7-24.

Adler, P. S., & Clark, K. B. (1991). Behind the learning curve: A sketch of the learning process. *Management Science, 37*(3), 267-281.

Allan, G. (2006). The Legitimacy of Grounded Theory. European Conference on Research Methods,

Anderson, E. G., Parker, G., & Tan, B. (2014). Platform performance investments in the presence of network externalities. *Information Systems Research, 25*(1), 152-172.

Argote, L., Beckman, S., & Epple, D. (1990). The persistence and transfer of learning in industrial settings. *Management Science, 36*(2), 140-154.

Argote, L., & Miron-Spektor, E. (2011). Organizational Learning: From Experience to Knowledge. *Organization Science, 22*(5), 1123-1137.

Asadullah, A., Faik, I., & Kankanhalli, A. (2018). *Evolution Mechanisms for Digital Platforms: A Review and Analysis across Platform Types* International Conference on Information Systems, San Francisco.

AutomotiveWorld. (2019). *20 years at the cutting edge: BMW ConnectedDrive.* Retrieved 15.02.2019 from https://www.automotiveworld.com/news-releases/20-years-cutting-edge-bmw-connecteddrive/

Avison, D. (1998). Rigour in Action Research: Some Observations and a Plea. *Scandinavian Journal of Information Systems, 10*(1), 119-124.

Babüroglu, O., & Ravn, I. (1992). Normative Action Research. *Organization Studies, 13*(1), 19-34.

Baldwin, C. Y., & Clark, K. B. (2000). *Design rules: The power of modularity* (Vol. 1). MIT press.

Baldwin, C. Y., & Woodard, C. J. (2008). The Architecture of Platforms: A unified view. *Working Papers - Harvard Business School Division of Research, 1-3*.

Bano, M., & Zowghi, D. (2014). A systematic review on the relationship between user involvement and system success. *Information and Software Technology, 58*, 148-169.

Barney, J. (1991). Firm resources and sustained competitive advantage. *Journal of Management, 17*(1), 99-120.

Baskerville, R. L. (1999). Investigating Information Systems with Action Reserach. *Communications of the AIS, 2*, 3–16.

Baskerville, R. L., & Myers, M. D. (2004). Special Issue on Action Research in Information Systems: Making IS Research revelant to Practive. *MIS Quartely, 28*(3), 329-335.

Baskerville, R. L., & Wood-Harper, A. T. (1998). Diversity in Information Systems Action Research Methods. *European Journal of Information Systems, 7*(2), 1998.

Beckman, C. M., Haunschulid, P. R., & Phillips, D. J. (2004). Friends or Strangers? Firm-Specific Uncertainty, Market Uncertainty, and Network Partner Selection. *Organization Science, 15*(3), 259-275.

Bender, B., & Gronau, N. (2017). *Coring on Digital Platforms – Fundamentals and Examples from the Mobile Device Sector* Thirty Eighth International Conference on Information Systems (ICIS), Seoul, South Corea.

Benlian, A., Hilkert, D., & Hess, T. (2015). How open is this platform? The meaning and measurement of platform openness from the complementors' perspective. *Journal of Information Technology, 30*(3), 209-228.

Bergvall-Kåreborn, B., & Howcroft, D. (2014). Persistent problems and practices in information systems development: a study of mobile applications development and distribution. *Information Systems Journal, 24*(5), 425-444.

Bharadwaj, A., El Sawy, O. A., Pavlou, P. A., & Venkatraman, N. V. (2013). Digital business strategy: toward a next generation of insights. *MIS Quarterly, 37*(2), 471–482.

Bhaskar, R. K., Anslow, C., Brosz, J., & Maurer, F. (2016). Developing usable APIs with XP and cognitive dimensions. In *2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)* (pp. 101–105). IEEE.

Binken, J., & Stremersch, S. (2009). The effect of superstar software on hardware sales in system markets. *Journal of Management, 73*(2), 88-104.

Bloch, J. (2006). How to design a good API and why it matters. In *Companion to the 21st ACM SIGPLAN symposium on Object-oriented programming systems, languages, and applications* (pp. 506-507). ACM.

BMW. (2018). *New BMW Cockpit is digital, intelligent, perfectly tuned to the driver and always up-to-date.* https://www.press.bmwgroup.com/global/article/detail/T0285445EN/new-bmw-cockpit-is-digital-intelligent-perfectly-tuned-to-the-driver-and-always-up-to-date?language=en

Bogner, A., Littig, B., & Menz, W. (2014). *Experteninterviews: Eine Einführung in Theorie und Praxis einer Forschungsmethode.* Springer Fachmedien GmbH.

Boudreau, K. (2010). Open Platform Strategies and Innovation: Granting Access versus Devolving Control. *Management Science, 56*(10), 1849-1872.

Boudreau, K. (2012). Let a thousand flowers bloom? An early look at large numbers of software app developers and patterns of innovation. *Organization Science, 23*(5), 1409-1427.

Boudreau, K., & Jeppesen, L. (2014). Unpaid crowd complementors: The platform network effect mirage. *Strategic Management Journal, 36*(12), 1761-1777.

Briers, M., & Chua, W. F. (2001). The role of actor-networks and boundary objects in management accounting change: a field study of an implementation of activity-based costing. *Accounting, organizations and society, 26*(3), 237-269.

Brown, J. S., & Duduid, P. (1991). Organizational learning and communities-of-practice: Toward a unified view of working, learning, and innovation. *Organization Science, 2*(1), 40-57.

Burns, C., Ferreira, J., Hellmann, T. D., & Maurer, F. (2012). Usable results from the field of API usability: A systematic mapping and further analysis. In *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE.

Ceccagnoli, M. (2012). Co-Creation of Value in a Platform Ecosystem: The Case of Enterprise Software. *MIS Quartely, 36*(1), 263-290.

Cennamo, C., Ozalp, H., & Kretschmer, T. (2018). Platform Architecture and Quality Trade-offs of Multihoming Complements. *Information Systems Research, 29*(2), 461–478.

Cennamo, C., & Santalo, J. (2013). Platform competition: Strategic trade-offs in platform markets. *Strategic Management Journal, 34*(11), 1331–1350.

Cennamo, C., & Santaló, J. (2019). Generativity tension and value creation in platform ecosystems. *Organization Science, 30*(3), 617-641.

Charmaz, K. (2001). Grounded Theory. In L. Y. S. & D. N. K. (Eds.), *The American tradition in qualitative research.* Thousand Oaks: SAGE.

Chesbrough, H. (2003). The Era of Open Innovation. *MIT Sloan Management Review, 44*(3), 34-41.

Chesbrough, H., & Bogers, M. (2014). Explicating Open Innovation: Clarifying an Emerging Paradigm for Understanding Innovation In *New Frontiers in Open Innovation* (pp. 3-28). Oxford University Press.

Chuang, Y.-S. (2014). Learning and International Knowledge Transfer in Latecomer Firms: The Case of Taiwan's Flat Panel Display Industry. *IEEE Transactions on Engineering Management, 61*(2), 261-274.

Coghlan, D., & Brannick, T. (2001). *Doing Action Research In Your Own Organization*. SAGE.

Constantinides, P., Henfridsson, O., & Parker, G. (2018). Introduction—Platforms and Infrastructures in the Digital Age. *Information Systems Research, 29*(2), 381-400.

Creswell, J. W. (2009). *Research Design: Qualitative, Quantitative and Mixed Methods Approaches*. SAGE.

Crossan, M., Lane, H., & White, R. (1995). Organizational learning: Dimensions for a theory. *The international journal of organizational analysis, 3*(4), 337-360.

Cui, A. S., Chan, K., & Calantone, R. (2014). The Learning Zone in New Product Development. *IEEE Transactions on Engineering Management, 61*(4), 690-701.

Cusumano, M., Gawer, A., & Yoffie, D. (2019). *Business of Platforms: Strategy in the Age of Digital Competition, Innovation, and Power*. Harper Business.

Dal Bianco, V., Myllaerniemi, V., Komssi, M., & Raatikainen, M. (2014). *The Role of Platform Boundary Resources in Software Ecosystems: A Case Study* 11th Working IEEE/IFIP Conference on Software Architecture (WICSA), Sydney, Australia.

Darke, P., Shanks, G., & Broadbent, M. (1998). Successfully completing case study research: combining rigour, relevance and pragmatism. *Information Systems Journal, 8*(4), 273-289.

Davison, R. M., & Martinsons, M. G. (2016). Context is king! Considering particularism in research design and reporting. *Journal of Information Technology, 31*(3), 241-249.

de Reuver, M., Sørensen, C., & Basole, R. C. (2017). The digital platform: a research agenda. *Journal of Information Technology, 33*(2), 124-135.

De Weck, O. L., Roos, D., & Magee, C. L. (2011). *Engineering Systems: Meeting Human Needs in a Complex Technological World*. MIT Press.

Dellermann, D., Jud, C., & Popp, K. M. (2016). *Why don´t they join? Analyzing the Nature and Consequences of Complementors´ Costs in Platform Ecosystems* Thirty Seventh International Conference on Information Systems (ICIS), Dublin, Ireland.

den Hartigh, E., Ortt, J. R., van de Kaa, G., & Stolwijk, C. C. M. (2015). Platform control during battles for market dominance: The case of Apple versus IBM in the early personal computer industry. *Technovation, 48*, 4-12.

Deshpande, R. (1983). Paradigms Lost: On Theory and Method in Research in Marketing. *Journal of Marketing, 47*(4), 101-110.

Dess, G. G., Ireland, R. D., Zahra, S. A., Floyd, S. W., Janney, J. J., & Lane, P. J. (2003). Emerging Issues in Corporate Entrepreneurship. *Journal of Management, 29*(3), 351-378.

Digitaltrends. (2017). *Audi shows integrated Android infotainment system at Google I/O*. Retrieved 15.02.2019 from https://www.digitaltrends.com/cars/audi-android-integration-google-io/

Dyer, J. H., & Singh, H. (1998). The relational view: Cooperative strategy and sources of interorganizational competitive advantage. *Academy of Management Review 23*(4), 660-679.

Eaton, B. (2016). *The Dynamics of Digital Platform Innovation: Apple's Strategy to Control Modular and Architectural Innovation in iOS* 49th Hawaii International Conference on System Sciences, Hawaii.

Eaton, B., Elaluf-Calderwood, S., Sørensen, C., & Yoo, Y. (2015). Distributed Tuning of Boundary Resources: The Case of Apple's iOS Service System *MIS Quartely, 39*(1), 217-244.

Eisenhardt, K., & Graebner, M. (2007). Theory building from cases: Opportunities and challenges. *Academy of Management Journal, 50*(1), 25-32.

Eisenhardt, K. M. (1989). Building Theories from Case Study Research. *Academy of Management*

*Review, 14*(4), 532-550.

Eisenmann, T., Parker, G., & Van Alstyne, M. (2006). Strategies for two-sided markets. *Harvard Business Review, 84*(10), 92-101

Eisenmann, T., Parker, G., & Van Alstyne, M. (2009). Opening Platforms: How, When and Why? In A. Gawer (Ed.), *Platforms, Markets and Innovation.* Edward Elgar Publishing Limited.

Eisenmann, T., Parker, G., & Van Alstyne, M. (2011). Platform Envelopment. In A. Gawer (Ed.), *Platforms, Markets and Innovation* (pp. 131-162). Edward Elgar.

Eklund, J., & Kapoor, R. (2019). Pursuing the New While Sustaining the Current: Incumbent Strategies and Firm Value During the Nascent Period of Industry Change. *Organization Science, 30*(2), 383-404.

Evans, D. S. (2012). Governing bad behaviour by users of multi-sided platforms. *Berkeley Technology Law Journal, 2*(27), 1201-1250.

Foerderer, J. (2017). Relational Third Party Governance: Evidence From Apple's World Wide Developer Conference. Platform Strategy Research Symposium, Boston.

Foerderer, J., Kude, T., Mithas, S., & Heinzl, A. (2018). Does platform owner's entry crowd out innovation? Evidence from Google photos. *Information Systems Research, 29*(2), 444-460.

Foerderer, J., Kude, T., Schuetz, S. W., & Heinzl, A. (2018). Knowledge boundaries in enterprise software platform development: Antecedents and consequences for platform governance. *Information Systems Journal*, 1-26.

Foerderer, J., Kude, T., Schuetz, S. W., & Heinzl, A. (2019). Knowledge boundaries in enterprise software platform development: Antecedents and consequences for platform governance. *Information Systems Journal, 29*(1), 119-144.

Foerderer, J. (2020). Interfirm Exchange and Innovation in Platform Ecosystems: Evidence from Apple's Worldwide Developers Conference. *Management Science, 66*(10), 4772-4787.

Frank, U., Klein, S., Krcmar, H., & Teubner, A. (1998). Aktionsforschung in der WI - Einsatzpotentiale und Einsatzprobleme.". *Wirtschaftsinformatik und Wissenschaftstheorie. Grundpositionen und Theoriekerne. Arbeitsberichte des Instituts für Produktion und Industrielles Informationsmanagement, 4*, 71-90.

Füller, K., Engel, T., Benz, M., Goswami, S., & Krcmar, H. (2014). A matrix for selecting appropriate customer integration methods. POMS International Conference, Singapore.

Gable, G. G. (1994). Integrating Case Study and Survey Research Methods: An Example in Information Systems. *European Journal of Information Systems, 3*, 112-126.

Gawer, A. (2009). Platforms, Markets and Innovation: An Introduction. In A. Gawer (Ed.), *Platforms, Markets and Innovation*. Edward Elgar.

Gawer, A. (2014). Bridging differing perspectives on technological platforms: Toward an integrative framework. *Research Policy, 43*(7), 1239-1249. https://doi.org/10.1016/j.respol.2014.03.006

Gawer, A., & Cusumano, M. (2014). Industry Platforms and Ecosystem Innovation. *Journal of Product Innovation Management, 31*(3), 417–433.

Ghazawneh, A. (2016). *The Challenges of Designing Digital Services for Multiple Mobile Platforms* Twenty-Fourth European Conference on Information Systems (ECIS), İstanbul,Turkey.

Ghazawneh, A., & Henfridsson, O. (2010a). *Governing third-party development through platform boundary resources* Thirty First International Conference on Information Systems (ICIS), St. Louis, USA.

Ghazawneh, A., & Henfridsson, O. (2010b). *Governing third-party development through platform boundary resources* Thirty First International Conference on Information Systems,, St. Louis, USA.

Ghazawneh, A., & Henfridsson, O. (2013). Balancing Platform Control and External Contribution in Third-Party Development: The Boundary Resources Model. *Information Systems Journal, 23*(2), 173-192.

Glaser, B. (1978). Theoretical Sensitivity: Advances in the Methodology of Grounded Theory. *Sociology Press*.

Glaser, B. (1992). Basics of Grounded Theory Analysis: Emergence vs. Forcing. *Sociology Press*.

Glaser, B., & Strauss, A. (1967). *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Weidenfeld and Nicolson.

Gläser, J., & Laudel, G. (2009). *Experteninterviews und qualitative Inhaltsanalyse als Instrumente rekonstruierter Untersuchungen*. VS Verlag für Sozialwissenschaften.

Google. (2019). *Developing Apps for Andoird Automotive OS*. https://android-developers.googleblog.com/2019/05/developing-apps-for-android-automotive.html

Grant, R. M. (1996). Toward a Knowledge-Based Theory of the Firm. *Strategic Management Journal, 17, Summer Special Issue*, 109-122.

Hagiu, A., & Wright, J. (2015). Multi-Sided Platforms. *International Journal of Industrial Organization, 43*, 162-174.

Han, J., & Cho, O. (2015). Platform business Eco-model evolution: case study on KakaoTalk in Korea. *Journal of Open Innovation: Technology, Market, and Complexity*.

Hannah, D., & Eisenhardt, K. M. (2018). How firms navigate cooperation and competition in nascent ecosystems. *39, 12*, 3163-3192.

Hanseth, O., & Lyytinen, K. (2010). Design theory for dynamic complexity in information infrastructures: the case of building internet. *Journal of Information Technology, 25*(1), 1-19.

Harrison, R., & Wells, M. (2000). A meta-analysis of multidisciplinary research. . Conference on Empirical Assessment in Software Engineering,

Hein, A., Schreieck, M., Riasanow, T., Setzke, D. S., Wiesche, M., Böhm, M., & Krcmar, H. (2019). Digital platform ecosystems. *Electronic Markets, In Press*.

Hein, A., Schreieck, M., Riasanow, T., Setzke, D. S., Wiesche, M., Böhm, M., & Krcmar, H. (2020). Digital platform ecosystems. *Electronic Markets, 30*, 87-98.

Hein, A., Schreieck, M., Wiesche, M., Böhm, M., & Krcmar, H. (2019b, 2019). The emergence of native multi-sided platforms and their influence on incumbents [journal article]. *Electronic Markets, 29*, 631-647. https://doi.org/10.1007/s12525-019-00350-1

Henard, D. H., & Szymanski, D. M. (2001). Why Some New Products Are More Successful Than Others. *Journal of Marketing Research, 38*(3), 362-375.

Henderson, R., & Clark, K. (1990). Architectural innovation: The reconfiguration of existing product technologies and the failure of established firms. *Administrative science quarterly, 35*(1), 9.

Henderson, R. M., & Clark, K. B. (1990). Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms. *Administrative Science Quarterly, 35*(1), 9.

Henfridsson, O., & Bygstad, B. (2013). The generative mechanisms of digital infrastructure evolution. *MIS Quartely, 37*(3), 907-931.

Henfridsson, O., & Ghazawneh, A. (2013). Balancing platform control and external contribution in third-party development: the boundary resources model. *Information Systems Journal, 23*(2), 173-192.

Henfridsson, O., Nandhakumar, J., & Scarbrough, H. (2018). Recombination in the open-ended value landsacpe of digital innovation. *Information and Organization, 28*(2), 89-100.

Henning, M. (2009). API design matters. *Communications of ACM, 52*(5), 46-56.

Herstatt, C., & von Hippel, E. (1992). From experience: Developing new product concepts via the lead user method: A case study in a "low-tech" field. *Journal of Product Innovation Management, 9*(3), 213-221.

Holzmüller, H. H., & Buber, R. (2009). *Qualitative Marktforschung – Konzepte – Methoden – Analysen*. Gabler.

Huang, P., Tafti, A. R., & Mithas, S. (2018). Platform Sponsor Investments and User Contributions in Knowledge Communities: The Role of Knowledge Seeding. *MIS Quartely, 42*(1), 213–240.

Huber, G. P. (1991). Organizational learning: The contributing processes and the literatures. *Organization Science, 2*(1), 88-115.

Huber, T. L., Kude, T., & Dibbern, J. (2017). Governance Practices in Platform Ecosystems: Navigating Tensions Between Cocreated Value and Governance Costs. *Information Systems Research, 28*(3), 563-584.

Huntley, C. L. (2003). Organizational Learning in Open-Source Software Projects: An Analysis of Debugging Data. *IEEE Transactions on Engineering Management, 50*(4), 485-493.

Iansiti, M., & Lakhani, K. R. (2009). SAP AG: Orchestrating the Ecosystem. *Harvard Business School Teaching Case*.

Iansiti, M., & Levien, R. (2004). *The keystone advantage: what the new dynamics of business ecosystems mean for strategy, innovation, and sustainability.* Harvard Business Press.

Iyer, B., & Subramaniam, M. (2015). Are You Using APIs to Gain Competitive Advantage? *Harvard Business Review*, 2–5.

Jacobides, M. G., Cennamo, C., & Gawer, A. (2018). Towards a theory of ecosystems. *Strategic Management Journal, 39*(8), 2255-2276.

Jacobides, M. G., MacDuffie, J. P., & Tae, J. (2016). Agency, structure, and the dominance of OEMs: Change and stability in the automotive sector. *Strategic Management Journal, 37*, 1942–1967.

Jaworski, B., & Ajay, K. (1993). Market orientation: antecedents and consequences. *Journal of Marketing, 57*(3), 53-70.

Kapoor, R., & Furr, N. R. (2015). Complementarities and competition: Unpacking the drivers of entrants' technology choices in the solar photovoltaic industry. *Strategic Managment Journal, 36*(3), 416-436.

Karhu, K., Gustafsson, R., & Lyytinen, K. (2018). Exploiting and Defending Open Digital Platforms with Boundary Resources: Android's Five Platform Forks. *Information Systems Research, 29*(2), 479–497.

Katz, M. L., & Shapiro, C. (1985). Network Externalities, Competition, and Compatibility. *The American Economic Review, 75*(3), 424-440.

Keller, K. L., MG, P., & I, J. (2011). *Strategic brand management: Building, measuring, and managing brand equity*. Pearson Education India.

Keng, S., & Rossi, M. (2011). Evaluation techniques for systems analysis and design modelling methods - a review and comparative analysis. *Information Systems Journal, 21*(3), 249-268.

Kim, J.-H., Prince, J., & Qiu, C. (2014). Indirect network effects and the quality dimension: A look at the gaming industry. *International Journal of Industrial Organization, 37*, 99-108.

Knight, P. (2002). A systemic approach to professional development: learning as practice. *Teaching and teacher education, 18*(3), 229-241.

Koch, T., & WIndsperger, J. (2017). Seeing through the network: Competitive advantage in the digital economy. *Journal of Organization Design, 6*(1), 6-30.

Kude, T., Dibbern, J., & Heinzl, A. (2012). Why Do Complementors Participate? An Analysis of Partnership Networks in the Enterprise Software Industry. *IEEE Transactions on Engineering Management, 59*(2), 250-265.

Lavie, D. (2006). The Competitive Advantage of Interconnected Firms: An Extension of the Resource-Based View. *Academy of Management Journal, 31*(3), 638-658.

Lee, S., Lee, S., Lim, S., Jung, J., & Choi, S. (2014). An API Design Process in Terms of Usability: A Case Study on Building More Usable APIs for Smart TV Platform. IEEE 38th International Computer Software and Applications Conference Workshops,

Levitt, B., & March, J. G. (1988). Organizational Learning. *Annual Review of Sociology, 14*(1), 319-338. https://doi.org/0360-0572/88/0815-0319502.00

Lewin, K. (1946). Action Research and Minority Problems. *Journal of Social Issues, 2*(4), 34-46.

Li, T., & Calantone, R. (1998). The impact of market knowledge competence on new product advantage: conceptualization and empirical examination. *Journal of Marketing, 62*(4), 13-29.

Li, Y., Zhang, C., Liu, Y., & Li, M. (2010). Organizational Learning, Internal Control Mechanisms, and Indigenous Innovation: The Evidence from China. *IEEE Transactions on Engineering Management, 57*(1), 63-77.

Lusch, R. F., & Nambisan, S. (2015). Service innovation: A service-dominant logic perspective. *MIS Quarterly, 39*(1), 155-175.

Lusch, R. F., & Vargo, S. L. (2006). Service-dominant logic: reactions, reflections and refinements. *Marketing Theory, 6*(3), 281-288.

Lüthje, C., & Herstatt, C. (2004). The Lead User method: an outline of empirical findings and issues for future research. *R&D Management, 24*(5), 553-568.

Manner, J., Nienaber, D., Schermann, M., & Krcmar, H. (2013). Six Principles for Governing Mobile Platforms. Wirtschaftsinformatik, Leipzig, Germany.

March, J. G. (1991). Exploration and Exploitation in Organizational Learning. *Organization Science, 2*(1), 71-87.

Maria, I., Freitas, B., & Fontana, R. (2018). Formalized Problem-Solving Practices and the Effects of Collaboration with Suppliers on a Firm's Product Innovation Performance. *Journal of Product Innovation Management, 35*(4), 565–587.

Mathiassen, L. (2002). Collaborative practice research. *Information Technology & People, 15*(4), 321-345.

McIntyre, D., & Srinivasan, A. (2017). Networks, platforms, and strategy: Emerging views and next steps. *Strategic Managment Journal, 38*(1), 141-160.

McNiff, J., Lomax, P., & Whitehead, J. (1996). *You and your action research project*. Routledge.

Mesquita, L. F., Jaideep Anand, & Brush, T. H. (2008). Comparing the resource-based and relational views: knowledge transfer and spillover in vertical alliances. *Strategic management journal 29*(9), 913-941.

Mo, J. P. T., & Lorchirachoonkul , W. (2016). Lifecycle design and support of intelligent web-based service systems. *International Journal of Agile Systems and Management, 9*(2), 135-153.

Moellers, T., Visini, C., & Haldimann, M. (2018). Complementing open innovation in multi-business firms: practices for promoting knowledge flows across internal units. *R&D Management, 33*(2), 1-20.

Mohagheghzadeh, A., & Svahn, F. (2016). *Shifting Design Capability to Third-Party Developers: An Affordance Perspective on Platform Boundary Resources* Twenty-second Americas Conference on Information Systems, San Diego.

Mohagheghzadeh, A., & Svahn, F. (2016). *Transforming Organizational Resource into Platform boundary Resources* Twenty-Fourth European Conference on Information Systems (ECIS), İstanbul,Turkey.

Myers, B., Ko, A., LaToza, T., & Yoon, Y. (2016). Programmers are users too: Human-centered methods for improving programming tools. *Computer, 49*(7), 44-52.

Myers, M. D., & Klein, H. (2011). A Set of Principles for Conducting Critical Research in Information Systems. *MIS Quarterly, 35*(1), 17-36.

Myers, M. D., & Newman, M. (2007). The qualitative interview in IS research: Examining the craft. *Information and Organization, 17*(1), 2-26.

Ondrus, J., Gannamaneni, A., & Lyytinen, K. (2015). The impact of openness on the market potential of multi-sided platforms: a case study of mobile payment platforms. *Journal of Information Technology, 30*(3), 260-275.

Oswald, & Krcmar, H. (2018). *Digitale Transformation. Fallbeispiele und Branchenanalysen*. Springer Gabler.

Parker, G., & Van Alstyne, M. (2017). Innovation, Openness, and Platform Control. *Management Science, 64*(7), 3015-3032.

Parker, G., Van Alstyne, M., & Choudary, S. P. (2016). *Platform Revolution: How Networked Markets are Transforming the Economy and How to Make Them Work for You*. New York: WW Norton & Co.

Parker, G., Van Alstyne, M., & Jiang, X. (2016). *Platform Ecosystems: How Developers Invert the Firm* (MIS Quartely, Issue.

Parker, G., Van Alstyne, M., & Jiang, X. (2017). Platform Ecosystems: How Developers Invert the Firm. *MIS Quartely, 41*(1), 255-266.

Parker, G. G., & Van Alstyne, M. W. (2005). Two-Sided Network Effects: A Theory of Information Product Design. *Management Science, 51*(10), 1494-1504. https://doi.org/10.1287/mnsc.1050.0400

Peirce, C. S. (1905). The Archetonic Construction of Pragmastism. *Collected Papers of Charles Sanders Peirce, 5*, 3-6.

Porter, M. E. (1980). *Competitive strategy: Techniques for analyzing industries and competitors.* Harper & Row.

Pühler, M. (2011). *Interaktive Gestaltung von Automotive Services durch softwaregestuetzten Einsatz domaenenspezifischer Modellierung.* [Dissertation]. Diss., Technische Universität München, Munich. (Technische Universität München)

Rausch, A., Bartel, C., Herold, S., Klus, H., & Niebuhr, D. (2013). From Software Systems to Complex Software Ecosystems: Model- and Constraint-Based Engineering of Ecosystems. *Perspectives on the Future of Software Engineering*, 61-80.

Rausch, A., Müller, J., Niebuhr, D., Herold, S., & Goltz, U. (2012). IT Ecosystems: A new Paradigm for Engineering Complex Adaptive Software Systems. 6th IEEE International Conference on Digital Ecosystems and Technologies (DEST), Campione d'Italia, Italy.

Reason, P., & Torbert, W. (2001). The action turn: Toward a transformational social science. *Concepts and transformation, 6*(1), 1-37.

Rickmann, T., Wenzel, S., & Fischbach, K. (2014). Software Ecosystem Orchestration: The Perspective of Complementors. Twentieth Americas Conference on Information Systems, Savannah, USA.

Rietveld, J., Ploog, J. N., & Nieborg, D. B. (2020). Coevolution of platform dominance and governance strategies: effects on complementor performance outcomes. *Academy of Management Discoveries, 6*(3), 488-513.

Rietveld, J., & Schilling, M. A. (2021). Platform competition: A systematic and interdisciplinary review of the literature. *Journal of Management, 47*(6), 1528-1563.

Rietveld, J., Schilling, M. A., & Bellavitis, C. (2016). *Reload and Relaunch: Value Creation and Value Capture In Platform-Based Markets.* Proceedings of the Acadamy of Management Annual Meeting, Anaheim, California.

Rochet, J.-C., & Tirole, J. (2003). Platform competition in two-sided markets. *Journal of the european economic association, 1*(4), 990-1029.

Rogan, M., & Greve, H. (2015). Resource dependence dynamics: Partner reactions to mergers. *Organization Science, 26*(1), 239-255.

Rolland, K., Mathiassen, L., & Rai, A. (2018). Managing Digital Platforms in User Organizations: The Interactions Between Digital Options and Digital Debt. *Information Systems Research, 29*(2), 419–440.

Rudmark, D., & Ghazawneh, A. (2011). *Third-Party Development for Multi-Contextual Services: On the Mechanisms of Control* ECIS 2011 Proceedings,

Saarikko, T. (2016). Platform Provider by Accident a Case Study of Digital Platform Coring. *Business & Information Systems Engineering, 58*(3), 177-191.

Sandberg, J., Holmström, J., & Lyytinen, K. (2020). Digitization and Phase Transitions in Platform Organizing Logics: Evidence from the Process Automation Industry. *MIS Quarterly, 44*(1), 129-153.

Schermuly, L., Schreieck, M., Wiesche, M., & Krcmar, H. (2019). Developing an industrial IoT platform–Trade-off between horizontal and vertical approaches. 14th International Conference on Wirtschaftsinformatik, Siegen.

Schilling, M. (2002). Technology Success and Failure in Winner-Take-All Markets: The Impact of Learning Orientation, Timing, and Network Externalities. *Academy of Management Journal, 45*(2), 387-398.

Schirrmacher, N.-B., Ondrus, J., & Kude, T. (2017). Launch Strategies of Digital Platforms: Platforms with switching and non-switching users. European Conference on Information Systems (ECIS), Guimarães, Portugal.

Schlachtbauer, T., Stanzl, G., Pühler, M., Scheermann, M., & Krcmar, H. (2012). Smart Automotive Apps fuer kundenorientierte Dienstleistungen im Fahrzeug. In L.-P. C. e. Verclas S (Ed.), *Smart Mobile Apps* (pp. 27-41). Springer.

Schreieck, M., Hein, A., Wiesche, M., & Krcmar, H. (2017). The Challenge of Governing Digital Platform Ecosystems. In S. R. Linnhoff-Popien C, Zaddach M (Ed.), *Digital Marketplaces Unleashed*. Springer.

Schreieck, M., & Wiesche, M. (2017). How Established Companies Leverage IT Platforms for Value Co-Creation – Insights from Banking. Twenty-Fifth European Conference on Information Systems (ECIS), Guimarães, Portugal.

Schreieck, M., Wiesche, M., & Krcmar, H. (2016). Design and Governance of Platform Ecosystems-Key Concepts and Issues for Future Research. Twenty-Fourth European Conference on Information Systems, Istanbul, Turkey.

Schreieck, M., Wiesche, M., & Krcmar, H. (2017). The Platform Owner's Challange to Capture Value - Insights from a Business-to-Business IT Platform. Thirty Eighth International Conference on Information Systems (ICIS), Seoul, South Corea.

Schreieck, M., Wiesche, M., & Krcmar, H. (2018). Multi-Layer Governance in Platform Ecosystems of Established Companies. Academy of Management Proceedings, Chicago, Illinois.

Schreieck, M., Wiesche, M., & Krcmar, H. (2021). Capabilities for Value Co-Creation and Value Capture in Emergent Platform Ecosystems: A Longitudinal Case Study of SAP's Cloud Platform. *Journal of Information Technology, 36*(4), 365-390.

Seamans, R., & Zhu, F. (2014). Responses to Entry in Multi-Sided Markets: The Impact of Craigslist on Local Newspapers. *Management Science, 60*(2), 476-493.

Selander, L., Henfridsson, O., & Svahn, F. (2013). Capability search and redeem across digital ecosystems. *Journal of Information Technology, 28*, 183-197.

Simon, H. A. (1991). Bounded rationality and organizational learning. *Organization Science, 2*(1), 125-134.

Smedlund, A., & Faghankhani, H. (2015). Platform Orchestration for Efficiency, Development and Innovation. 48th Hawaii International Conference on Information Systems Science, Sorensen, C. (2015). Mobile platforms and ecosystems. *Journal of Information Technology, 30*, 195-197.

Soto Setzke, D., Böhm, M., & Krcmar, H. (2019). Platform Openness: A Systematic Literature Review and Avenues for Future Research. International Conference on Wirtschaftsinformatik, Siegen, Germany.

Srinivasan, A., & Venkatraman, V. N. (2020). Architectural Convergence and Platform Evolution: Empirical Test of Complementor Moves in Videogames. *IEEE Transactions on Engineering Management, 67*(2), 266-282.

Staykova, K. S., & Damsgaard, J. (2017). *Towards an Integrated View of Multi-Sided Platforms Evolution* International Conference on Information Systems, Seoul, South Korea.

Stjepandić, J., Ostrosi, E., Fougères, A.-J., & Kurth, M. (2015). Modularity and supporting tools and methods. In *Concurrent Engineering in the 21st Century: Foundations, Developments and Challenges* (pp. 389-420). Springer.

Stonig, J., Schmid, T., & Müller-Stewens, G. (2022). From product system to ecosystem: How firms adapt to provide an integrated value proposition. *Strategic Managment Journal, Early Access*.

Strauss, A., & Corbin, J. (1990). *Basics of Qualitative Research: Grounded Theory Procedures and Techniques*. Sage.

Stummer, C., Kundisch, D., & Decker, R. (2018). Platform Launch Strategies. *Business & Information Systems Engineering, 60*(2), 167-173.

Susman, G. I. (1983). Action research: a socialtechnical system perspective. In *Beyond Method: Strategies for Social Research* (pp. 95-113). SAGE.

Svahn, F., Lindgren, R., & Mathiassen, L. (2015). Applying Options Thinking to Shape Generativity in Digital Innovation: An Action Research into Connected Cars. International Conference on System Sciences, Hawaii.

Svahn, F., Mathiassen, L., & Lindgren, R. (2017). Embracing Digital Innovation in Incumbent Firms: How Volvo Cars Managed Competing Concerns. *MIS Quartely, 41*(1), 239-255.

Tan, F. T. C., Tan, B., & Pan, S. L. (2016). Developing a Leading Digital Multi-sided Platform: Examining IT Affordances and Competitive Actions in Alibaba.com. *Communications of the AIS, 38*(1), 739–760.

Tashakkori, A., & Teddlie, C. (2010). *Sage handbook of mixed methods in social and behavioral research*. SAGE.

Teece, D., J. (2017). Dynamic capabilities and (digital) platform lifecycles. In A. Gawer, B. S. Silverman, & S. Stern (Eds.), *Entrepreneurship, Innovation, and Platforms* (Vol. 37, pp. 211-225). Emerald Publishing Limited.

Teece, D. J., Pisano, G., & Shuen, A. (1998). Dynamic capabilities and strategic management. *Strategic Managment Journal, 18*(7), 509-533.

Tilson, D., Lyytinen, K., & Sørensen, C. (2010). Research Commentary—Digital Infrastructures: The Missing IS Research Agenda. *Information Systems Research, 21*(4), 748-759.

Tiwana, A. (2014). *Platform ecosystems: Aligning Architecture, Governance, and Strategy*. Elsevier Inc.

Tiwana, A., & Bush, A. A. (2014). Spotting Lemons in Platform Markets: A Conjoint Experiment on Signaling. *IEEE Transactions on Engineering Management, 61*(3), 393-405.

Tiwana, A., Konsynski, B., & Bush, A. A. (2010). Research commentary—Platform evolution: Coevolution of platform architecture, governance, and environmental dynamics. *Information Systems Research, 21*(4), 675-687.

Trigeorgis, L. (1993). The Nature of Option Interactions and the Valuation of Investments with Multiple Real Options. *Journal of Financial and Quantitative Analysis, 28*(1), 1-20.

Tripsas, M., & Gavetti, G. (2000). Capabilities, cognition, and inertia: Evidence from digital imaging. *Strategic management journal 21*(10/11), 1147-1161.

Tushman, M. L., & Anderson, P. (1986). Technological Discontinuities and Organizational Environments. *Administrative Science Quarterly, 31*(3), 439.

Um, S. Y., Yoo, Y., Wattal, S., Kulathinal, R. J., & Zhang, B. (2013). The Architecture of Generativity in a Digital Ecosystem: A Network Biology Perspective. Thirty Fourth International Conference on Information Systems (ICIS), Milan, Italy.

Urquhart, C. (2013). *Grounded theroy for qualitative reserach: a practical guide*. SAGE.

Urquhart, C., & Fernández, W. (2013). Using Grounded Theory Method in Information Systems: The Researcher as Blank Slate and Other Myths. *Journal of Information Technology, 28*(3), 224-236.

Urquhart, C., Lehmann, H., & Myers, M. D. (2010). Putting the theory back into grounded theory: Guidelines for grounded theory studies in information systems. *Information Systems Journal, 20*(4), 357-381.

Verba, N., Chao, K.-M., James , A., Goldsmith, D., Fei, X., & Sergiu-Dan, S. (2017). Platform as a service gateway for the Fog of Things. *Advanced Engineering Informatics, 33*, 243-257.

von Hippel, E. (1986). Lead Users: A Source of Novel Product Concepts. *Management Science, 32*(7), 791-805.

von Hippel, E. (2005). *Democratizing innovation: The evolving phenomenon of user innovation*. The MIT Press, Cambridge, MA.

von Hippel, E., Thomke, S., & Sonnack, M. (1999). Creating breakthroughs at 3M. *Harvard business review 77*, 47-57.

Warner, K. S., & Wäger, M. (2019). Buidling dynamic capabilities for digital transformation: An ongoing process of strategic renewal. *Long Range Planning, 52*(3), 326-349.

Weiss, N., Schreieck, M., Brandt, L. S., Wiesche, M., & Krcmar, H. (2018). Digitale Plattformen in der Automobilbranche – Herausforderungen und Handlungsempfehlungen. *HMD Praxis der Wirtschaftsinformatik, 55*(2), 349-365.

Weiss, N., Schreieck, M., Wiesche, M., & Krcmar, H. (2018). Setting Up a Platform Ecosystem: How to integrate app developer experience. IEEE International Conference on Engineering, Technology and Innovation, Stuttgart, Gemany.

Weiss, N., Wiesche, M., & Krcmar, H. (2019). *Pure Coding Pleasure: How BMW Involves App Developers in the Design of Automotive Onboard APIs* Americas Conference on Information Systems, Cancun, Mexcio.

West, J. (2003). How open is open enough?: Melding proprietary and open source platform strategies. *Research Policy, 32*(7), 1259-1285.

Wiesche, M., Jurisch, M. C., Yetton, P., & Krcmar, H. (2017). Grounded Theory Methodology in Information Systems Research. *MIS Quartely, 31*(3), 685-701.

Williams, T. (2008). How Do Organizations Learn Lessons From Projects—And Do They? *IEEE Transactions on Engineering Management, 55*(2), 248-266.

Yeow, A., Sia, S. K., Soh, C., & Chua, C. (2017). Boundary Organization Practices for Collaboration in Enterprise Integration. *Information Systems Research*.

Yin, R. K. (2009). *Case Study Research Design and Methods*. Sage Inc.

Yoo, Y., Boland Jr., R. J., Lyytinen, K., & Majchrzak, A. (2009). Special Issue: Organizing for Innovation in the Digitized World. *Organization Science, 20*(1), 278-279.

Yoo, Y., Henfridsson, O., & Lyytinen, K. (2010). The New Organizing Logic of Digital Innovation: An Agenda for Information Systems Research. *Information Systems Research, 21*(4), 724-735.

Zaggl, M. A., Schweisfurth, T. G., & Herstatt, C. (2020). The Dynamics of Openness and the Role of User Communities: A Case Study in the Ecosystem of Open Source Gaming Handhelds. *IEEE Transactions on Engineering Management, 67*(3), 712-723.

Zahra, S. A., Nielsen, A. P., & Bogner, W. C. (1999). Corporate Entrepreneurship, Knowledge, and Competence Development. *Entrepreneurship: Theory and Practice, 23*(3), 169-189.

Zghidi, A., Hammouda, I., Hnich, B., & Knauss, E. (2017). On the Role of Fitness Dimensions in API Design Assessment - An Empirical Investigation. IEEE/ACM 1st International Workshop on API Usage and Evolution (WAPI),

# Appendix

## Appendix A: List of Interview Partners

This section provides an overview of all experts that were interviewed for this thesis, a brief job description, the date of the interview, and the respective papers were the collected data was used.

| # | Role | Brief description | Date | Paper |
|---|---|---|---|---|
| I1 | Product owner app platform | ▪ Head of platform team<br>▪ 5 years of experience in app platform development and operations | 27.04.2017 | P1, P2 |
| I2 | Application developer | ▪ Software developer in app development team | 09.05.2017 | P1, P2 |
| I3 | Application developer | ▪ Technical responsible person in app development team<br>▪ Multiple years of experience from multiple app development projects | 16.05.2017 | P1, P2 |
| I4 | Platform developer | ▪ Head of platform programming team<br>▪ Multiple years of software development in platform context | 23.05.2017 | P1, P2 |
| I5 | Application developer | ▪ Team manager of app development team<br>▪ Multiple years of experience from multiple app development projects | 23.05.2017 | P1, P2 |
| I6 | Technical consultant | ▪ Project leader of project for restructuring aftersales of digital products in cars<br>▪ Experience from multiple roles in digital aftersales | 30.05.2017 | P1, P2 |
| I7 | Sales manager | ▪ Business responsible person for one app | 06.06.2017 | P1, P2 |
| I8 | Application developer | ▪ Technical responsible person for development of multiple apps | 13.06.2017 | P1, P2 |
| I9 | Sales manager | ▪ Team member of app store web application team | 14.06.2017 | P1, P2 |
| I10 | Application developer | ▪ Head of development team of onboard store app | 15.06.2017 | P1, P2 |
| I11 | Platform strategist | ▪ Strategist for digital product development | 22.06.2017 | P1, P2 |
| I12 | Platform strategist | ▪ Strategist for digital product development<br>▪ Former team member in platform team | 22.06.2017 | P1, P2 |
| I13 | Sales manager | ▪ Business responsible person for one app<br>▪ Multiple years of experience from digital product sales in automotive context | 22.06.2017 | P1, P2 |
| I14 | Sales manager | ▪ Responsible for web portal as additional touchpoint for in-car apps | 23.06.2017 | P1, P2 |
| I15 | Product owner app platform | ▪ Head of platform team<br>▪ 6 years of experience in app platform development and operations | 19.11.2018 | P2, P3 |
| I16 | Platform architect | ▪ Head of platform architect team | 19.11.2018 | P2, P3 |
| I17 | Product owner app platform | ▪ Technical responsible in platform team | 19.11.2018 | P2, P3 |

| | | | | |
|---|---|---|---|---|
| I18 | Product owner app | ▪ Product owner of app development team | 27.11.2018 | P2, P3 |
| I19 | Application developer | ▪ Software developer in app development team | 29.11.2018 | P2, P3 |
| I20 | Application developer | ▪ Software developer in app development team | 30.11.2018 | P2, P3 |
| I21 | Product owner app | ▪ Software developer in app development team | 11.12.2018 | P2, P3 |
| I23 | Application developer | ▪ Software developer in app development team | 14.12.2018 | P2, P3 |
| I23 | Application developer | ▪ Software developer in app development team | 17.12.2018 | P2, P3 |
| I24 | Technical lead platform development | ▪ Head of platform development team | 18.12.2018 | P2, P3 |
| I25 | Application developer | ▪ Software developer in app development team | 17.01.2019 | P2, P3 |
| I26 | Platform developer | ▪ Member of the platform development team | 23.01.2019 | P2, P3 |
| I27 | Platform developer | ▪ Member of the platform development team | 05.02.2019 | P2, P3 |
| I28 | Platform developer | ▪ Member of the platform development team | 05.02.2019 | P2, P3 |
| I29 | Platform strategist | ▪ Member of the digital strategy department | 05.07.2019 | P4, P5 |
| I30 | Platform manager | ▪ Director of ap platform division | 26.07.2019 | P4, P5 |
| I31 | Product owner app ecosystem | ▪ Product owner third party ecosystem integration in “Car as a platform” project | 30.04.2019 | P4, P5 |
| I32 | Partner management | ▪ Partner manager for digital services | 30.04.2019 | P4, P5 |
| I33 | Product owner platform | ▪ Product owner Tencent platform | 07.05.2019 | P2, P4, P5 |
| I34 | Project manager for digital platforms | ▪ Project manager in “Car as a platform” project | 08.05.2019 | P4, P5 |
| I35 | Portfolio manager | ▪ Digital portfolio manager | 15.05.2019 | P4, P5 |
| I36 | Purchase digital services | ▪ Purchasing manager for digital platforms and apps | 15.05.2019 | P4, P5 |
| I37 | Purchase digital services | ▪ Purchasing manager for digital platforms and apps | 17.05.2019 | P4, P5 |
| I38 | Innovation manager | ▪ Product owner for different app and platform prototypes | 23.05.2019 | P4, P5 |
| I39 | Product owner app | ▪ Product owner of app development team | 05.06.2019 | P2, P4, P5 |
| I40 | Customer analyst | ▪ Customer analyst in strategy department | 22.05.2019 | P4, P5 |
| I41 | Project manager for digital platforms | ▪ Project manager in “Car as a platform” project | 31.05.2019 | P4, P5 |
| I42 | Competition analyst | ▪ Competition analysis for digital services | 06.06.2019 | P4, P5 |
| I43 | Purchase digital services | ▪ Purchasing manager for digital platforms and apps | 11.06.2019 | P4, P5 |
| I44 | Manager digital services | ▪ Manager for digital services & partership management | 12.06.2019 | P4, P5 |
| I45 | Mobility services manager | ▪ Manager for mobility services | 12.06.2019 | P4, P5 |
| I46 | Strategy manager | ▪ Director of digital strategy division | 09.09.2019 | P4, P5 |
| I47 | Partner manager | ▪ Partner manager for potential third-party apps | 16.09.2019 | P4, P5 |
| I48 | Product owner onboard store app | ▪ Product owner of app development team of store app | 03.04.2020 | P4, P5 |
| I49 | Product owner app | ▪ Product owner of app development team | 06.04.2020 | P4, P5 |

| | | | | |
|---|---|---|---|---|
| I50 | Product owner app ecosystem | ▪ Product owner third party ecosystem integration in "Car as a platform" project | 07.04.2020 | P4, P5 |
| I51 | Product owner service portal | ▪ Product owner for automotive service portal | 07.04.2020 | P4, P5 |
| I52 | Analyst | ▪ Manager customer insight center | 13.04.2020 | P4, P5 |
| I53 | Product manager digital services | ▪ Product manager digital services in strategy department | 17.04.2020 | P4, P5 |
| I54 | Platform strategist | ▪ Member of the digital strategy department | 22.04.2020 | P4, P5 |
| I55 | Product manager eCommerce platform | ▪ Product manager eCommerce platform | 29.04.2020 | P4, P5 |
| I56 | Product owner onboard app | ▪ Product owner of app development team | 4.06.2020 | P4, P5 |
| I57 | Platform strategist | ▪ Member of the digital strategy department | 04.05.2020 | P4, P5 |
| I58 | Product mobile app | ▪ Product owner BMW mobile app | 11.05.2020 | P4, P5 |
| I59 | Application developer | ▪ Software developer in app development team | 11.05.2020 | P5 |
| I60 | Application developer | ▪ Software developer in app development team | 13.05.2020 | ISJ |

## Appendix B: Original Publications

# Setting Up a Platform Ecosystem

## How to integrate app developer experience

Niklas Weiß, Maximilian Schreieck, Manuel Wiesche, Helmut Krcmar
Department of Informatics
Technical University of Munich
Munich, Germany
niklas.weiss@tum.de
{maximilian.schreieck, wiesche, krcmar}@in.tum.de

***Abstract*— Platform governance is a fundamental task for operating a platform ecosystem. A platform owner needs to orchestrate the app developers on its platform. Even though platform governance is frequently considered in information systems (IS) research, there is little knowledge on how platform owners can acquire skills and knowledge on platform governance. This is particular relevant in early phases of platforms. This paper strives to shed light on how an automotive manufacturer who has little experience as platform owner creates a platform and improves its platform governance. Based on four episodes from observed in the development department's practices, we identify transfer of knowledge and integration of artifacts as basic mechanisms that enable the platform owner to benefit from interactions with app developers. Our work contributes to the under-investigated field of platform emergence and provides valuable insights for practitioners.**

***Keywords—platform emergence; knowledge boundaries; platform boundary resources; platform orchestration***

### I. Introduction

Software platforms massively contribute to a disruptive change in multiple traditional industries. Platform companies such as Uber or AirBnB are increasingly successful and attack traditional companies in their industries. In contrast to such platform-native companies, traditional companies need to transform their organization and processes for embracing digital innovation through software platforms [1]. As a consequence, such companies need to acquire skills and knowledge on their new role as platform owner and their new task of orchestrating app developers [2]. If they fail to do so, they may endanger not only the platform ecosystem itself, but also traditional value creation processes of the company. Furthermore, suboptimal platform governance may harm the company's public brand image [3]. Hence, the challenge for a company that is new to the platform business is to embrace their role as platform owner by applying platform governance, without harming current value creation [4].

Platform governance refers to mechanisms, which enable the platform owner to exert influence on app developers. It needs to respect developers' autonomy and embrace innovation while also being able to integrate the developers' contributions into the ecosystem as a harmonious whole [5]. Finding the right balance on platform openness and control in the ecosystem is frequently considered as crucial challenge for platform owners [6, 7]. Further, a platform owner may utilize complementary features, initially provided by app developers, for the evolvement of the platform itself. Bender and Gronau [8] describe the process of coring as the integration of features provided by applications in the platform core. This may contribute to the platform competitiveness in comparison to competing platform ecosystems [9]. An emerging platform owner needs to determine all these routines and balances regarding interactions with app developers [10].

Even though software platforms are frequently considered in current literature, large parts of IS research investigate already established, viral platform ecosystems [11].The work presented here intends to shed light on the emergence of a new platform owned by a traditional company that has little experience as platform owner. It aims to understand how the company interacts with app developers and in which way it acquires knowledge on governing a platform ecosystem. We state the following research question:

*How can a platform owner utilize interactions with app developers to improve platform governance?*

To answer this question, we investigate a globally operating automotive manufacturer and the establishment of its software platform for automotive onboard applications.

### II. Theoretical Background

A platform ecosystem consists of two basic components. First, the platform core is an extensible code base that serves as foundation for the development of apps by sharing certain features through interfaces. Complementary apps as second component, interoperate with the platform by using the provided interfaces [12, 13]. The stakeholder that is primarily responsible for the platform core is referred to as platform owner, whereas the development of apps is up to multiple app developers [12]. The relationship between platform owner and app developers has been considered recently in IS research [6, 14]. The platform owner has to orchestrate the platform ecosystem [5]. Thereby the platform owner manages the tradeoff between an enhancement of external contributions by app developers on the one side and securing the infrastructural control over the platform on the other [7]. Thereby the platform holds multiple artifacts, called platform boundary resources that serve as interfaces for the relationship between platform owner and app developers [15]. Platform boundary resources can be embodied by application programming interfaces (APIs), platform tooling or code documentation. However, in recent IS literature the stream of information and resources has only been discussed in the direction from the platform owner to app developers. Our approach is an investigation of knowledge and artifacts that are transferred in the opposite direction, that is, from app developers to the platform owner.

## III. Research Design

We apply a single case research strategy [16]. Our research question considers a specific phase in the evolvement of a platform ecosystem. With the case study method, we meet the requirements for the investigation of such a dynamic, contemporary phenomenon. Furthermore, the large diversity of stakeholders in the context of a digital platform within a traditional company causes a certain level of complexity. Information are heterogeneously distributed across several knowledge providers and multiple projects.

We collected data in the context of a software platform for automotive onboard applications of a globally operating automotive manufacturer. The access to the platform is exclusively limited to employees and subcontractors of the company. Initially, we gathered data by interviews with partners in the platform ecosystem. This includes app developers, platform owners, and employees of the automotive manufacturer that pose the role of customers. All conversations were recorded and subsequently transcribed. Building on an established collaborative practice research initiative at the automotive manufacturer, the first author collected additional data in form of emails, conversation protocols, and entries from internal knowledge boards.

For the coding and analysis of the interview data, we applied grounded theory procedures [17]. Within a partial portfolio approach [18], 144 codes were identified in an open coding procedure. Further, these findings were condensed to core categories in a selective coding step. The results served as basis for the episodes we describe in the next section.

## IV. Results

The following section describes four episodes that describe in which form the platform owner benefits from the involvement of app developers and the respective effects on the investigated platform itself (Table 1).

TABLE I.

| Episode | Description |
|---|---|
| Establishing a Developer Forum | The platform owner established a "question and answer" forum that enables developers to support each other. Furthermore, platform team members support solution finding. The forum was well accepted, and a highly active developer community was established. |
| Code Review Process | Sample tests of several applications indicated unintended usage of platform mechanisms by applications. This motivated the platform team to conduct code reviews for every app that should be deployed to the production environment. |
| Public Library and SDK | Several app developers claimed missing or insufficient platform features. Even though the implementation or refactoring of the feature was already in planning, the availability for the app developers was too late for their project plans. For this reason, the platform team decided to make platform code repository accessible for the developer community and allow commits from app developers. |
| Provisioning of Start App | Several app developers recognized the challenges and impediments for novice developers to start implementation. For this reason, one team of app developers created a starter app, which contains a feature tour of the platform functionalities, as well as several development tools that were not provided by the platform itself. The platform owner recognized the value of the starter app and decided to officially support it and claim it as the official way for starting a new app development. |

### A. *Establishing a Developer Forum*

In the early phase of the platform ecosystem several app developers complained about limited knowledge exchange within the app developer community. Support from the platform team was handled by a front-desk mail address. This caused opacity regarding the provided support since the answer was exclusively visible for the asking app developer:

*"So, the actual how-to knowledge is not transferred. […] This is absolutely a problem, that code is pushed around exclusively and that there is no platform for knowledge management."*

To tackle this insufficient procedure, the platform team established a developer forum that enables knowledge exchange between all involved stakeholders. App developers provide mutual support, and the platform team also engages to provide problem solutions. This form of public knowledge board enabled the platform owner to recognize and understand problems of the app developer community without any direct interaction. In this way the knowledge transfer from developers to the platform owner is standardized and visible for every party of the app development community.

### B. *Code Review Process*

Sample tests of several applications indicated unintended usage of platform mechanisms by applications. This motivated the platform team to conduct code reviews for every app that should be deployed to a production environment. In this way, unintended behavior should be avoided, and platform stability should be improved. By reviewing the apps, the platform team recognized volatile quality of the submitted artifacts:

*"We just recognized that there are apps that perform fine whereas others reveal bad mistakes. We are not sure why this is the case. We have just assumptions."*

Based on the most frequently identified mistakes, the platform team creates several tutorials and coding guidelines. Besides notifying the app development teams regarding the identified mistakes, face-to-face meetings are organized. Within these conversations, the mistakes are discussed and agreements on rectifications on the app side are made. This approach supports the learning process on the platform owner side. Furthermore, by considering the app developer perspective in the face-to-face meetings, the platform team was able to understand why the identified mistakes were made.

### C. Public Libraries and SDK

After some weeks of app development in the platform ecosystem, several app developers claimed missing or insufficient platform features. Even though the implementation or refactoring of these features was already in planning, the availability for the app developers was too late for their project plans. For this reason, the platform team decided to make platform code repository accessible for the developer community and allow commits from app developers:

*"And then there are cases, where you catch a new (platform) version with hardly any changes und suddenly an arbitrary generic error message pops up and you have to investigate what the actual concern is. A random feature is not working, and you don't know if this is a bug or if it is intended, since everything is in movement."*

By default, the code repository is accessible for every participant in the ecosystem. When an app development team recognizes the demand for an unavailable platform feature, it usually requests the implementation by the platform team. However, the required change may not fit into the working plan of the platform team. Then, the app team has the option to implement the feature by itself. By reviewing the submitted code, the platform team ensures a sufficient quality of the implemented feature.

### D. Provisioning of Starter App

Several app developers recognized the challenges and impediments for novice developers to start implementation. It took large efforts to onboard new team members in already ongoing implementations. Furthermore, new teams were engaged in infrastructure and setup related tasks over weeks before they were able to start the actual building of their apps:

*"It is really hard to get running. It takes you weeks – also if you are experienced with ancestor generations of platform development – to start working in a productive way."*

Motivated by improving this situation, one team of app developers created a starter app containing a feature tour of the platform functionalities as well as several development tools that were not provided by the platform itself. The feature tour contains examples that demonstrate the utilization of the most relevant platform interfaces such as user interface APIs or interfaces for triggering certain activities in the car's navigation system. Furthermore, the tooling provides features that are not crucially necessary for app development on the platform, though they are mandatory for professional software development as scripts for static and dynamic code analysis or automated deployment to the test environment. The starter app was spread in the developer community and became best practice for every app team that started developing. After recognizing the large advantages of the starter app, the platform team decided to claim it as standard support it officially. The app developers that initially created the starter app still contribute and support the maintenance of the artifact.

## V. Discussion

Before analyzing the actual interactions between platform owner and app developers, we need to emphasize the status of the platform ecosystem we investigated. The establishment of a software platform that is exclusively accessible for app developers that are employees of the platform company enables the involvement of app developers in the process of platform evolution. Internal app developers will not have the intention to harm the ecosystem. Concurrently, they are able to provide unfiltered and veritable insights from the app developer perspective to the platform owner. This enables organizational learning without a potentially harmful platform tuning process by app developers [6].

Considering the interactions between platform owner and app developers, we identify several types of interactions that (Table 2).

TABLE II.

| Episode | Knowledge Transfer | Integration |
|---|---|---|
| Establishing Developer Forum | ▪ Common problems of app developer community<br>▪ Best practices and solutions of app developer groups<br>▪ Specific problems of single app developers | ▪ Solutions suggested in forum are implemented by the platform team |
| Code Review Process | ▪ Common error patterns<br>▪ Opportunities for unintended app behavior<br>▪ Best practices of app developer groups<br>▪ Problems and challenges of single app developers | ▪ Integration of good practices in platform documentation<br>▪ Derivation of platform guidelines from unintended app behavior |
| Public Libraries and SDK | ▪ Common problems of app developer community | ▪ Direct platform development by app developers<br>▪ Joint development and maintenance |
| Provisioning of Starter App | ▪ Common pain point for new app developers | ▪ Integration of app artifact in platform<br>▪ Joint development and maintenance |

Involving app developers in a platform ecosystem is frequently considered under the aspects of transparency and accessibility [19]. However, these considerations illustrate interactions from an app developer's perspective. We invert this perspective and apply these aspects to the platform owner's view. We consider the observed transfer of knowledge as transparency of app development activities towards the platform. The observation of app developers enables the platform owner to detect omissions and failures in the platform and to improve these aspects. Foerderer, et al. [20] identified three levels of knowledge transfer from platform owner to developer: broadcasting, brokering, and bridging. Following our logic of assuming the platform owner's perspective, we invert this consideration and reveal three ways of knowledge transfer from app developers to platform owner.

(1) *Reverse Broadcasting* is a standardized knowledge transfer in the context of an arm's length governance between platform owner and app developers. In our observations we can find this form of knowledge transfer in the developer forum. The platform owner is able to gain knowledge on common problems of the app developer community by scanning standardized

forum posts without any direct interactions. (2) *Reverse Brokering* is an intermediate type of knowledge transfer in which the platform owner may interact with a subset of app developers. It occurs in our data in the episode on the provisioning of the starter app. After recognizing the large distribution of the starter app, the platform owner contacted the app team that created the artifact. The app developers reported the need for more support for app developers that start with their work. Thereby they are able to provide insights from the app developer perspective directly to the platform team and thereby contribute to learning on the platform owner side. (3) *Reverse Bridging* is a direct and problem-specific knowledge exchange of platform owner and app developers. This mechanism can be identified in the code review process. After analyzing the code of a specific app, the review team schedules a meeting with the app developers. Beside the agreement on certain patches in the app, the platform team tries to understand the emergence of the mistakes made when using platform features.

Accessibility is the second facet of involving app developers in platform ecosystems. In our consideration, it embodies the capabilities to integrate solutions of app developers directly into the platform. Similar to the transparency aspects we distinguish three levels of accessibility. (1) *Integration of community activities* describes a process in which the community develops solutions or features that are integrated in the platform without any direct interaction between platform owner and app developers. One example for this can be found in the episode on public libraries and SDK. Similar to an open source contribution work flow, there is no direct interaction required for integrating code of app developers into the platform. The (2) *integration of best practices* covers the platform owner recognizing the evolvement of a common praxis and cooperates with the responsible team to integrate the feature into the platform. This mechanism can be found in the episode on the starter app. The platform owner was convinced by the developed artifact and integrated it into the platform in cooperation with the app developer team. Finally, the (3) *integration of single solutions* is the transferring of a specific problem's solution from an app into a platform feature in a mode of close cooperation. This can be found in the episode on the developer forum. In multiple cases, the platform team adopted the specific solutions posted in the forum.

## VI. Conclusion

In this study, we investigate how interactions with app developers enable platform owners to master platform governance. We confirm the forms of knowledge boundaries in platform ecosystems established by Foerderer, et al. [20]. However, by inverting the direction of information flow we extend the existing body of knowledge. Furthermore, we transfer the different levels of interactions to the integration of real artifacts into the platform. Thereby, we add the aspect of platform boundary resources as tool for platform evolvement to the existing body of knowledge. Finally, we highlight the importance of artifacts like a developer forum, a starter app or a code review process, providing valuable insights for platform owners in the field.

## VII. References


[1] F. Svahn, L. Mathiassen, and R. Lindgren, "Embracing Digital Innovation in Incumbent Firms: How Volvo Cars Managed Competing Concerns," *MIS Quartely,* vol. 41, no. 1, pp. 239-255, 2017.

[2] T. Rickmann, S. Wenzel, and K. Fischbach, "Software Ecosystem Orchestration: The Perspective of Complementors," in *Twentieth Americas Conference on Information Systems*, Savannah, USA, 2014.

[3] K. L. Keller, P. MG, and J. I, *Strategic brand management: Building, measuring, and managing brand equity*. Pearson Education India, 2011.

[4] J. Stjepandić, E. Ostrosi, A.-J. Fougères, and M. Kurth, "Modularity and supporting tools and methods," in *Concurrent Engineering in the 21st Century: Foundations, Developments and Challenges*Cham: Springer, 2015, pp. 389-420.

[5] A. Tiwana, *Platform ecosystems: Aligning Architecture, Governance, and Strategy*. 225 Wyman Street, Waltham, MA 02451, USA: 2014 Elsevier Inc., 2014.

[6] B. Eaton, S. Elaluf-Calderwood, C. Sørensen, and Y. Yoo, "Distributed Tuning of Boundary Resources: The Case of Apple's iOS Service System " *MIS Quartely,* vol. 39, no. 1, pp. 217-244, 2015.

[7] O. Henfridsson and A. Ghazawneh, "Balancing platform control and external contribution in third-party development: the boundary resources model," *Information Systems Journal,* vol. 23, no. 2, pp. 173-192, 2013.

[8] B. Bender and N. Gronau, "Coring on Digital Platforms – Fundamentals and Examples from the Mobile Device Sector," presented at the Thirty Eighth International Conference on Information Systems (ICIS), Seoul, South Corea, 2017.

[9] G. Parker, M. Van Alstyne, and X. Jiang, "Platform Ecosystems: How Developers Invert the Firm," in "MIS Quartely," 2016, vol. Boston University Questrom School of Business Research Paper No. 2861574.

[10] J. P. T. Mo and W. Lorchirachoonkul "Lifecycle design and support of intelligent web-based service systems," *International Journal of Agile Systems and Management,* vol. 9, no. 2, pp. 135-153, 2016.

[11] M. de Reuver, C. Sørensen, and R. C. Basole, "The digital platform: a research agenda," *Journal of Information Technology,* 2017.

[12] A. Tiwana, B. Konsynski, and A. A. Bush, "Research commentary—Platform evolution: Coevolution of platform architecture, governance, and environmental dynamics," *Information Systems Research,* vol. 21, no. 4, pp. 675-687, 2010.

[13] N. Verba, K.-M. Chao, A. James , D. Goldsmith, X. Fei, and S. Sergiu-Dan, "Platform as a service gateway for the Fog of Things," *Advanced Engineering Informatics,* vol. 33, pp. 243-257, 2017.

[14] A. Mohagheghzadeh and F. Svahn, "Shifting Design Capability to Third-Party Developers: An Affordance Perspective on Platform Boundary Resources," presented at the Twenty-second Americas Conference on Information Systems, San Diego, 2016.

[15] A. Ghazawneh and O. Henfridsson, "Governing third-party development through platform boundary resources," presented at the Thirty First International Conference on Information Systems,, St. Louis, USA, 2010.

[16] R. K. Yin, *Case Study Research Design and Methods*. 2455 Teller Road Thousand Oaks, California, USA: Sage Inc., 2009.

[17] B. Glaser and A. Strauss, *The discovery of grounded theory*. Aldine, Chiacago, 1967.

[18] M. Wiesche, M. C. Jurisch, P. W. Yetton, and H. Krcmar, "Grounded theory methodology in information systems research," *MIS Quartely,* vol. 41, no. 3, pp. 685-701, 2017.

[19] A. Benlian, D. Hilkert, and T. Hess, "How open is this platform? The meaning and measurement of platform openness from the complementors' perspective," *Journal of Information Technology,* vol. 30, no. 3, pp. 209-228, 2015.

[20] J. Foerderer, T. Kude, S. W. Schuetz, and A. Heinzl, "Knowledge boundaries in enterprise software platform development: Antecedents and consequences for platform governance," *Information Systems Journal,* pp. 1-26, 2018.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



# Learning to be a Platform Owner: How BMW Enhances App Development for Cars

Niklas Weiss, Manuel Wiesche, Maximilian Schreieck, and Helmut Krcmar

***Abstract*—Platform owners face multiple challenges such as onboarding and orchestrating app developers as well as providing resources to enable the development of complementary apps. Information systems research considers digital platform governance as key to address these challenges. Thereby, the focus lies on the relationship of a platform owner and app developers. However, while there is evidence how app developers acquire skills through these interactions, there is limited knowledge of how platform owners benefit from interacting with app developers to improve their digital platforms. To address this gap, in this article, we study the emergence of a digital platform for automotive onboard apps within the BMW Group. Our results are grounded in 30 expert interviews that we conducted during a period of two years and are enriched by extensive secondary data. We identify transfer of perspective, transfer of knowledge, and transfer of artifacts as basic mechanisms that enable a platform owner to enhance its digital platform. The inherent improvements of the digital platform facilitate the app development. Our work extends the existing theory on platform emergence and provides insights into the learning process of an inexperienced platform owner. Our findings reveal valuable recommendations for organizations that are struggling to establish digital platforms.**

***Index Terms*—Connected car, digital platforms, grounded theory, organizational learning, platform governance, platform emergence.**

## I. Introduction

THE phenomenon of digital platforms is a major trend in research as well as practice that disrupts multiple industries. Scholars try to understand in which way Intel could dominate the microprocessor industry for years [1], Apple has become one of the most valuable firms on the planet [2], or game developers decide for specific console platforms [3]. However, while this work illustrates how firms master the orchestration of their digital platform ecosystem [4], organizations need to acquire knowledge on platform ownership during the emergence of their digital platform [5], [6]. Even though organizational learning has been broadly investigated for new product development [7], project management in firms [8], or open source software [9], the learning process of a platform owner during the emergence of a digital platform has not been considered so far. To address this gap, this article illustrates platform emergence within an established firm from an organizational learning perspective.

We understand a digital platform as a set of stable components that supports variety and evolvability in a system by constraining the linkages among the other components [10]. It incorporates a central core surrounded by multiple actors in its digital platform ecosystem [11], [12]. Its generative capabilities enhance innovation and accelerate development processes of digital services [13]. Hence, competition no longer revolves around the control of classic value chains but around attracting generative activities associated with a digital platform [14]. One major shortcoming of the existing research on digital platforms is its focus on matured digital platform ecosystems [14]: Other than a few exceptions [5], [15], [16], scholars mainly consider established digital platforms from an *ex post* perspective [17]–[20]. While there is comprehensive knowledge on mechanisms and processes of successful orchestration of digital platforms, the path to achieve a matured status remains unclear. In their broadly recognized "research agenda on digital platforms," de Reuver *et al.* [14] shift the focus toward platform emergence. They raise the question if a successful digital platform can be consciously designed or whether it is just a result of coincidences. Even though the platform may be refined over its whole lifetime [17], it is initially designed during platform emergence. Considering the large challenges of a platform launch [21], [22], the platform design is particularly critical in this phase. Therefore, an inexperienced firm needs to acquire knowledge on the design of digital platforms and its associated ecosystem [23]. However, it is unclear in which way these firms can do so. While learning mechanisms on the app developers' side are broadly considered [3], [17], [18], [20], the platform owners' acquisition of knowledge remains opaque. The organizational learning theory provides a perspective that allows us to describe and understand learning effects of an inexperienced platform owner. Thereby, we focus on interactions, which are a basic condition for organizational learning [24].

In a platform organization, the platform owner is responsible for the development of a stable core, which is provided to developers that build complementary apps [10]. Both sides embody communities-of-practice that constantly adapt and evolve their abilities by accumulating knowledge by practical experience [25]. Moreover, the platform owner is responsible for the

Manuscript received May 3, 2019; revised January 5, 2020; accepted July 14, 2020. Review of this manuscript was arranged by Department Editor C. Bartelt.
*(Corresponding author: Niklas Weiss.)*

Niklas Weiss, Maximilian Schreieck, and Helmut Krcmar are with the Technical University of Munich Boltzmannstr, 85748 Garching, Germany (e-mail: niklas.weiss@tum.de; maximilian.schreieck@tum.de; helmut.krcmar@tum.de).

Manuel Wiesche is with the Technical University of Dortmund, 12 44227 Dortmund, Germany (e-mail: manuel.wiesche@tu-dortmund.de).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEM.2020.3017051

0018-9391 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Technische Universitaet Muenchen. Downloaded on September 15,2020 at 09:57:52 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



governance of the interactions between these parties [11]. For example, a platform owner provides platform boundary resources to enable app developers to exploit platform capabilities [26]. However, interactions always provoke mutual learning [27]. The exchange between both sides should bring the articulation of knowledge also from app developers to the platform owner. With this consideration, we define our research question: "Which mechanisms enable a platform owner to learn digital platform design from interactions with app developers?" To address our research question, we analytically explored the coherences of organizational learning and platform governance. We argue that the transfer of knowledge from the platform owner to app developers is a proven pattern in the digital platform theory [20], while the opposite direction has remained understudied so far. To address our research question, we conduct an exploratory case study in the context of a digital platform for automotive onboard apps at the globally operating car manufacturer BMW. By analyzing our qualitative dataset with grounded theory procedures [28], we identify 25 representative events within four episodes that richly illustrate how the platform owner improved the platform design according to interactions with app developers. These episodes serve as the basis for our theorizing on three basic mechanisms that enable a platform owner to learn the transfer of perspective, transfer of knowledge, and transfer of artifacts.

By addressing this question, we can contribute to a better understanding on coherences in an initial phase of the platform design and its meaning for later platform success. For this research, we posit that the platform design is a critical factor for the launch of the digital platform and the attracting of app developers. Furthermore, we prove the applicability of the organizational learning theory to the new context of digital platforms. By intertwining literature on organizational learning with platform governance, we contribute to the prevalent theory on the product development and innovation management.

The remainder of this article is structured as follows. The next section describes the theoretical background of our considerations and clarifies the gap in the prevalent theory. Second, our methodological approach and the case of BMW as an owner of a just emerging digital platform for automotive onboard apps are described. Then, we illustrate the four episodes that we identified in the context of the platform design and describe our findings regarding concrete learning of the platform owner. The subsequent discussion theorizes which general mechanisms enable learning regarding platform design and in which way our findings are embedded in prevalent literature.

## II. Theoretical Background

The following section illustrates in which way our research is embedded in the prevalent literature. We therefore explain general coherences of platform governance, and subsequently, introduce the lens of the organizational learning theory. Our considerations reveal the lack of research on platform emergence in general and learning effects within a platform owner that occur through interactions with app developers in particular.

### A. *Platform Governance*

The core of a digital platform is in the center of every digital platform ecosystem [11], [12]. The exposed capabilities of the digital platform attract app developers [29], [30], who build complementary apps on the one side, which are consumed by customers on the other side [11]. Platform governance is the "partitioning of the decision-making authority between platform owners and app developers, control mechanisms, and pricing and pie-sharing structures" [11, p. 291]. For instance, in the iOS ecosystem, Apple as platform owner decides, which application programming interfaces (APIs) of the iPhone are accessible by third-party app developers [17]. Furthermore, each app needs to be submitted for review by Apple before it can be launched in the App Store. App developers need to choose the price of their app in a predefined selection of pricing steps and keep just 70% of the revenue. The rest belongs to Apple as platform owner. However, platform governance should also facilitate the creation of new apps [30]. By exposing generativity, the platform owner unleashes potential for innovation [31]. With the annual release of a new iOS version, Apple publishes new APIs that provide new possibilities for app developers. Comprehensive documentation and annual developer conferences inform and attract developers to build new apps. A platform with strictly controlled generativity would counter such endeavors [32]. Hence, the platform owner needs to balance the control over the digital platform and its ecosystem on the one side and provide a certain level of autonomy for app developers to foster innovation on the other [33]. In this way, platform governance covers multiple tactical decisions that impact interactions between the platform owner and app developers [30], [34].

Usually these decisions are transmitted and enforced by resources that enable app developers in their activities [35]. While there are technical resources such as software development kits (SDK) and development tools, the platform owner also provides knowledge on the app development as boundary resource. Foerderer *et al.* [20] describe multiple levels of the knowledge transfer through boundary resources. By transferring knowledge from the platform owner to app developers via multiple channels, app developers are able to choose the most appropriate way for their individual requirements. The design of platform boundary resources is constantly shaped and evolved in a common refining process of platform owner and app developers [3], [17]. Since platform boundary resources correspond to the platform owner's decisions on platform governance, refinements on platform boundary resources mirror adjustments of platform governance decisions. The adaption of platform governance indicates learning effects within the organization of a platform owner.

The accessibility of the provided platform resources is defined by the configuration of the platform's vertical openness [36]–[38]. Vertical openness defines the degree of accessibility and transparency of platform boundary resources for external actors. A platform owner can decide to limit access to certain platform assets for a defined group of users or restrict the usage of specific resources. In this way, vertical openness decides on the potential of knowledge transfer from the platform owner to app developers. In the other way around, a platform owner can pull external innovation back into the core of its platform [39]. Even

Authorized licensed use limited to: Technische Universitaet Muenchen. Downloaded on September 15,2020 at 09:57:52 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



though the potential danger of getting replaced by a platform feature may discourage app developers from entering a digital platform ecosystem, Parker *et al*. [40] argue that the overall digital platform ecosystem mostly benefits from improvements of the platform core through coring. In addition to the degree of vertical openness, a platform owner has to decide on the horizontal openness of its platform [37]. This refers to the interoperability with other platforms and the willingness of sharing the platform ownership with others. In specific cases, the platform is not orchestrated by a single owner (as Apple) but driven by a developer community, which is globally distributed [9]. Even though, also these platforms require governance, different members of the community consolidate certain decisions. In the Linux platform, the app developers are able to get involved as the platform owner and contribute to platform core by submitting pull requests. Hence, the Linux platform exhibits a high level of horizontal openness. Prior research illustrates that the degree of a platform's openness change over time [32]. In their extensive case study, Karhu *et al*. [19] describe in which way the openness of Google's Android decreased over the last decade. Likewise, the adaptions of platform openness indicate that the platform owner learns on the optimal degree of openness for the platform over time.

Boundary resources as well as openness constitute crucial aspects of platform governance that are affected by constant adoptions during the platform's lifecycle. However, we argue that especially the phase of platform emergence requires appropriate platform governance to manage the initial challenges of a new digital platform [21], [22]. Platform boundary resources that do not enable app developers in a sufficient way may have the potential to discourage app developers and aggravate the chicken–egg problem [22]. The same effect may apply to a platform with an insufficient level of openness and inadequate transparency or accessibility to app developers. An excessive level of openness on the side may involve the danger of losing control [32]. Our research illuminates the crucial stage of platform emergence and provides valuable insights that complement knowledge on digital platforms and support inexperienced platform owners in designing their platform.

### B. Organizational Learning in Digital Platform Ecosystems

Even though the refinement of platform boundary resources as well as platform openness indicates the existence of learning effects within the organization of a platform owner [17], such mechanisms were exclusively described on the app developers' side so far [20]. To shed light on this research gap, we take the lens of organizational learning. It is understood as "a change in the organization that occurs as the organization acquires experience" [41, p. 1124]. There are two basic processes that enable organizational learning [42]. First, acquisitive learning, which is the exploitation of access to preexisting knowledge and its subsequent implementation. In the context of digital platforms, acquisitive learning occurs whenever an app developer consumes documentation that was provided by an external source such as the platform owner. Second, experimental learning, describes the extraction of first-hand knowledge through own experience [42]–[44]. It describes the process of an app developer that learns best practices and code patterns during the actual programming activity. The ability to learn is critical for the performance of an organization and its long-term success [41], [45]. Hence, there is a large amount of research that investigates the effectiveness of organizational learning and its influence on the firm's performance [7], [46], [47]. However, measuring the effects of a phenomenon requires to understand its characteristics. Though we argue that there are indications of organizational learning within the organization of a platform owner, there is no evidence so far. Therefore, we align our efforts on another stream inside the organizational learning literature to illuminate the actual organizational learning process and its strategical implications [8], [48]–[51].

In general, learning may cause a change of behavior, a change of cognition, or both [52]. Furthermore, many researchers assume a link between learning and an improvement of performance [53]. Henard and Szymanski [54] argue that learning within a development team leads to a product advantage, which describes the degree to which a product is superior to its market alternatives. However, incorrect learning or the learning of incorrect coherences may even diminish the subsequent performance of an actor [48]. To evaluate learning mechanisms, Crossan *et al*. [52] define three dimensions that need to be considered. First, the unit of analysis needs to be clarified. In the context of our study, the digital platform and its governance represent the unit of analysis. Second, the outcome of learning needs to be analyzed. This affects changes in cognition or behavior. We address this aspect by considering changes in the platform or its governance as result of platform owner's learnings. Finally, the link between learning and performance needs to be clarified—which we understand as improvements of the platform design or its governance.

To understand the characteristics of organizational learning within the organization of a platform owner, we need to elucidate the mechanisms that evoke learning. We understand organizational learning in terms of communities that emerge. People work and learn collaboratively, and vital interstitial communities are formed and reformed. Since learning is a commuting process, the interaction of different communities enables mutual insights and an overall increase of knowledge [27]. The higher the amount of interactions, the larger the chance to learn from the experience of others [55]. Information circulates fast inside of communities. Between communities and its environment, information gets lost or is distributed slowly. Transferring this idea to a digital platform, we consider the app developers as one community and the platform owner, with its collaborating team members and different organizational units, as another community. Both communities are embedded in a common network, the digital platform ecosystem [12], [56]. In general, different actors in a network are able to generate assets through interactions such as knowledge-sharing routines and complementary endowments [57], [58]. Considering a digital platform ecosystem as a network of different actors, a platform owner can facilitate such mechanisms through appropriate governance. Smedlund and Faghankhani [59] emphasize the importance of interactions between participants in a digital platform ecosystem to facilitate reciprocal learning and coevolution. Hence, an incremental improvement of the platform and the complementary products can be achieved.

Authorized licensed use limited to: Technische Universitaet Muenchen. Downloaded on September 15,2020 at 09:57:52 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



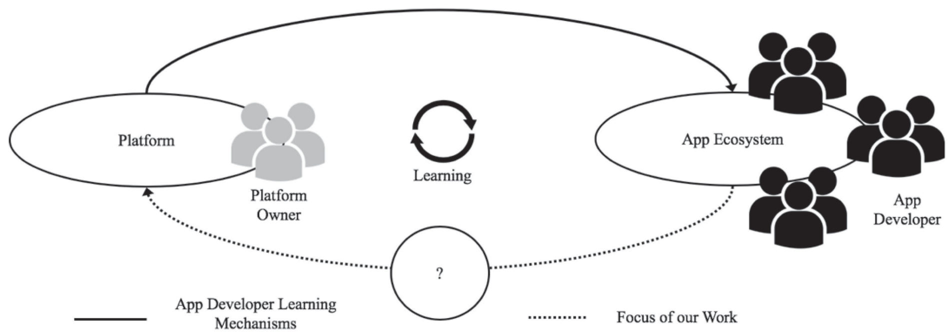



Fig. 1. Platform owner learning mechanisms as focus of our work.

Even though organizational learning of focal firms was considered in the past, we argue that digital platform ecosystems are different. While collaboration with suppliers considers organizational learning in traditional value chains [50], interactions in the context of digital platforms are affected by network effects [60]. The organizational learning of latecomer firms and the constitution of technological capabilities [49] differs also by the creation of values in pipelines and the lack of network effects [61]. Furthermore, the collaboration of different units inside an organization for new product development activities [51] differs from a platform setup by its incentive. A firm's employees develop a product that serves customer needs, and receives extrinsic motivation, for example, through their wages. However, app developers immediately benefit from platform improvements, and therefore, reveal an intrinsic motivation for interactions with the focal firm—in this case, the platform owner [30]. We strive to understand in which way these interactions are perceived by the platform owner in the context of organizational learning. With theories on platform boundary resources, knowledge resources, and platform openness, the learning mechanisms of app developers through interactions with a platform owner are well understood. However, the opposite direction remains understudied / opaque (see Fig. 1). We argue that understanding platform design is especially crucial during the emergence of a digital platform in which the foundation for success or failure is laid. Furthermore, research on organizational learning indicates that interactions cause mutual learning effects.

## III. Methodology

Our single case study [62], [63] provides longitudinal insights into the emergence and early evolution of a platform for automotive onboard apps at the BMW Group. Driven by evolved customer expectation, the globally operating car company is transforming from a pure car manufacturer into a provider of digital services. This includes the development and operation of a platform for automotive onboard apps, which allows a flexible deployment of new features in the form of apps for the car, even after the vehicle has left the production plant. Following our research question, we strived to identify and understand learning mechanisms at the department that is responsible for the development and operation of the app platform. With the case study method, we satisfy the requirements for investigating such a dynamic and fast evolving phenomenon. Furthermore, observing organizational learning requires collecting "time-series or longitudinal data" [41]. Therefore, our studies cover the period of two years.

The large diversity of stakeholders in the context of a digital platform and its ecosystem causes a tremendous level of complexity [23]. Information is distributed heterogeneously to several knowledge providers and multiple projects. In our study, we gathered data with 30 semistructured interviews [64] with 26 actors in the digital platform ecosystem to synthesize a comprehensive understanding of coherences and dynamics in the interactions of different ecosystem actors. Interview participants include representatives from the platform owner side as well as the app developer side. Due to their central role within the digital platform ecosystem, two app developers as well as two members of the platform team were interviewed twice. All conversations were recorded and transcribed. Based on a collaborative practice research initiative [65], the first author engaged in the daily activities of the platform team that allowed the collection of additional data in form of meeting protocols, code in development repositories, and entries from internal knowledge boards. This participation within the two years of our research enabled the collection of rich data, which is illustrated in Table I.

For the coding and analysis of the interview data, we applied a partial portfolio approach [66] of the grounded theory methodology [28]. The goal of the grounded theory is the creation of a theory that accounts for patterns of behavior, which is relevant and problematic for those involved [67]. The idea that empirical facts are the starting point of the theory development and research in general and not the reference point of empirical assessment is the key element of the grounded theory. The analysis of the collected data was conducted in three coding steps. The first step was open coding, a detailed, line-by-line analysis of the collected data that helps create a broad understanding of the phenomena by the researcher [68]. The open coding of our collected interview data yielded 254 codes. The subsequent selective coding step considers the emergence of a core category that unites all identified issues under one thematic umbrella. The core category needs to be central, occur frequently in the data

Authorized licensed use limited to: Technische Universitaet Muenchen. Downloaded on September 15,2020 at 09:57:52 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



TABLE I
OVERVIEW OF COLLECTED DATA

| | Data Source | Amount | Use in Analysis |
|---|---|---|---|
| **Primary Data** | Platform team members | 16 interviews (@45 - 60 minutes each) | Insights into the beliefs, motivations, and strategies insight the platform team |
| | App developer | 14 interviews (@45 - 60 minutes each) | Insights into the beliefs, motivations, problems and desires of app developers |
| **Secondary Data** | App review meetings | 49 meetings (@45 minutes each) | Insights into problems, perceptions, and strategies of the app review team |
| | Community meet-ups | 8 meet-ups (@90 minutes each) | Insights into the culture and communication between platform team and app developers |
| | Platform team meetings | 23 meetings (@180 minutes each) | Insights into the decisions, motivations, and strategies insight the platform team; |
| | Code repository of internal developer portal | 146 commits | Insights into the development of content and amount of documentation |
| | Analytics of internal question and answer forum | 735 discussions threads | Insights into engagement of the platform team and app developers as well as concerns and problems of app developers |

and should be related to most of the identified categories. In our research, the category that satisfied all of these criteria was "learning of the platform owner." Finally, we performed the step of theoretical coding. It is the property of coding and constant comparative analysis that yields the conceptual relationship between categories and their properties as they emerge [69]. The step of theoretical coding should reveal the generalizable contribution of the research.

Since grounded theory does not claim to be a perfect and finished product, but underlies permanent development, theory development should be presented as an ongoing process. We addressed this point by constantly refining our theorizing and collecting additional data within the considered period of two years. Due to the revealed level of theoretical saturation of refinements in our perceptions [69], we decided to terminate our research after two years and present the results in this study.

## IV. BMW ONBOARD APP PLATFORM

To explore learning mechanisms of a platform owner in the context of an emerging digital platform, we choose the case of BMW and its platform for automotive onboard apps. The automotive industry is heavily affected by digitalization and the inherent change in customer expectations. The car is considered as a digital device that receives frequent software updates and provides options for extensions and customization of software features via apps. The considered app platform is one measure that enables BMW to face these new challenges. It enables modular deployment of apps in the car. The platform is part of the BMW OS 7.0, the company's latest infotainment system, released in summer 2018 with the start of production of the latest BMW X5 series (see Fig. 2). Large numbers of the models that

Fig. 2. BMW OS 7.0

were released since then as well as upcoming BMW models will run this system. By exposing multiple functionalities of the car to apps, the platform provides the base for a broad spectrum of use cases. At the release, more than 20 apps were available, providing services as a parking lot finder,[1] music streaming,[2] Microsoft Office 365,[3] and apps for different BMW service calls.[4]

The platform is managed by a development department, which is responsible for the vehicle's connectivity. We refer to this department as the "platform team" in the context of this study. The platform is used by app development teams from multiple, different departments, which are distributed all over the organization.

[1] [Online]. Available: https://www.bmw-connecteddrive.de/app/index.html#/portal/store/Base_Connected_Parking from 19.02.2019

[2] [Online]. Available: https://www.bmw-connecteddrive.de/app/index.html#/portal/store/Base_Connected_Music from 19.02.2019

[3] [Online]. Available: https://www.bmw-connecteddrive.de/app/index.html#/portal/store/Base_ExchangeOffer from 19.02.2019

[4] [Online]. Available: https://www.bmw-connecteddrive.de/app/index.html#/portal/store/Base_AssistNoTPEGOffer from 19.02.2019

Authorized licensed use limited to: Technische Universitaet Muenchen. Downloaded on September 15,2020 at 09:57:52 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



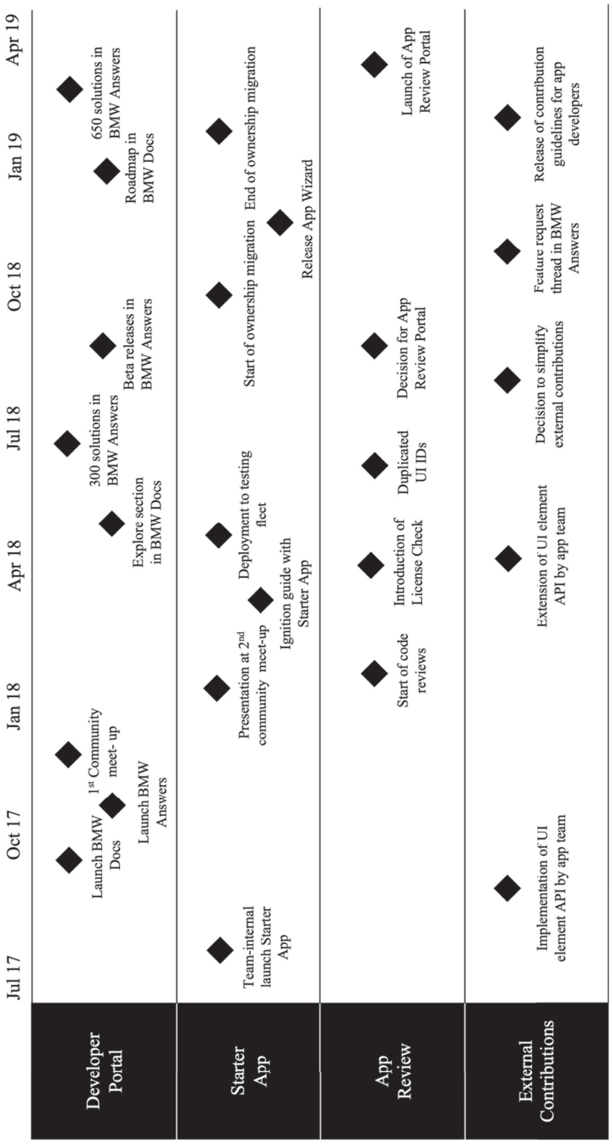



Fig. 3. Identified events in the context of the BMW app platform.

The platform development started in May 2016. After an initial phase of pure platform development activities, app development teams from the infotainment service department as well as telematics department started their activities in December 2016. More and more apps started their development in the following month. Fig. 3 presents an overview of 25 identified events that revealed learning mechanisms for the platform owner. The events are clustered into four major episodes, which provide the structure for the subsequent illustration of our results.

### *A. Developer Portal*

In the summer of 2017, several app developers complained about documentation that was stored in multiple wiki spaces inside BMW's intranet. Access to these pages was limited and the information these pages contained was fragmented and partially contradictory.

> *"So, the actual how-to knowledge is not transferred. [ ... ] This is absolutely a problem, that documentation is pushed around exclusively and that there is no central platform for knowledge management."* ∼ *App developer*

A new app developer was blocked for weeks, waiting for access to all information and resources that were required for app development. In August of 2017, the platform team decided to support the plans of a developer portal that was available to every employee inside of BMW and represented a single point of truth. In October 2017, the "BMW Docs" was launched. It included a basic ignition guide with a description of all required steps for the setup of a developer environment on a programmer's machine and further basic guides for app developers. From this point on, new documentation was pushed to the developer portal. Furthermore, existing documentation, which was distributed across several wiki pages, was transferred. After a while, the platform team recognized a significant decrease of support requests from app developers. However, the team considered the support infrastructure as insufficient. Requests from app developers were handled by a front-desk mail address. Members of the platform team checked the mailbox and answered all requests. By doing so, the answer was only visible to the app developer who asked the question, even though other developers faced the same issues. To tackle this insufficient procedure, the team decided to establish a BMW internal question and answer forum for developers, similar to the public stack overflow forum on the web.

> *"We recognized that the prevalent support via mail simply did not scale. We had to answer identical requests multiple times while developers waited a long time for a response in certain cases."* ∼ *Platform team member*

"BMW Answers" started in November 2017. From then on, when app developers faced a problem, they were advised to open a thread in the question and answers forum. Similar to "BMW Docs," the forum was open to everybody inside the BMW network by default. Any direct requests to the platform team via the former support mail address or another channel were rejected with direction to go BMW Answers. The forum was quickly adopted by the app developer community. Within the first month, the forum revealed almost 50 daily active users. After a few weeks, the team recognized that the number of support requests decreased dramatically. One reason for this was rooted in the searchable content of the forum. Whenever a problem is solved by a reply, the creator of the thread was encouraged to mark the appropriate post as solution. In this way, the solution was easy to find for a later visitor of to the topic. If a solution was provided once, other app developers just needed to search the forum to solve the identical problem. Furthermore, the platform team recognized that experienced app developers started to respond to posts of other developers so that there was no need for the platform team to engage. In June 2018, the 300th solution was entered into BMW Answers; by March 2019, more than 650 issues had been solved. To support this formation of a community, the platform team decided to organize periodic

Authorized licensed use limited to: Technische Universitaet Muenchen. Downloaded on September 15,2020 at 09:57:52 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



community meetings, which were to take place every one to two months. These "meet-ups," which were open for everybody interested, should facilitate connections between different app development teams and encourage mutual knowledge exchange. A meet-up usually started with a technical presentation of several dedicated topics such as new platform features or adjustments in platform-related processes. Subsequently, an open discussion was initiated by the platform team to learn about problems and challenges the app developers are currently facing. The first meet-up in December 2017 started with a group of 15 interested people. The number of participants grew event by event and reached a plateau of 40 to 50 people after the fourth meeting.

> *"The idea of meet-ups for our app developer community emerged when we recognized the benefits of exchange between different app development teams in Answers. The initial motivation for the meet-ups was to connect people and foster this exchange." ~ Platform team member*

Parallel to the community building meet-ups, the engagement of app developers in BMW Answers grew constantly. While the forum revealed 45 daily active users in January 2018, the number increased to 120 daily active users by July 2018 and eventually to 150 daily active users by February 2019. Thereby, the forum was used for different purposes.

1) The largest percentage of threads addressed a lack of documentation. App developers requested instructions for specific platform features that were available, although there was no documentation available. The platform team usually responded to such requests by creating the required documentation in BMW Docs and posting a link to the created article as solution in the thread. In May 2018, the platform team decided to shift more efforts to documentation tasks. From that time on, the documentation of new platform features in BMW Docs was integrated into the platform's release process.
2) A second kind of posts considered platform bugs that were identified by app developers. They used BMW Answers to report bugs, while the platform team notified the app developer in the same thread as soon as the bugfix was released in the platform.
3) Furthermore, app developers created threads to ask for the availability of specific platform features. The uncertainty regarding the platform's capabilities resulted in large efforts for app development teams since the feasibility of every new feature in their app required an evaluation of the platform. The platform team, therefore, created an "Explore" section in BMW Docs in May 2018, which contains a description of all basic platform capabilities. In January 2019, this section was enhanced by a roadmap for future platform features. In this way, app developers were not only able to estimate feasibility of features in their apps based on the current but also on future platform capabilities.

The fourth kind of thread in BMW Answers contained announcements of the platform team to the community. Even though also BMW Docs also contained a news section, the announcements in BMW Answers were used for gathering fast and public feedback regarding certain decisions or deliveries. For example, in August 2018, the platform team started to announce beta releases of the platform in the forum. The thread served as collecting tank for feedbacks, desires, and requirements of app developers that were discussed internally by the platform team and implemented if they were considered to be reasonable.

### B. Starter App

Concurrently to the establishment of the developer portal, several app teams recognized the large efforts that were required for the onboarding of new team members. During that time, not just the new developer was hindered from productive development activities but also the teaching developer was also blocked from his work.

> *"You always need to sit together with new team members. You need to explain a lot. Because there isn't any well prepared learning content. When you want to learn working with another platform, you go to YouTube and watch tutorials. There is Udemy or Udacity, which provide multiple learning classes. There was nothing comparable for the BMW app platform." ~ App developer*

In August 2017, the app development team that is responsible for the development of location-based services decided to build a so-called Starter App. The app contained three things that were required by each new app developer. First, the app entailed all basic mechanisms that are required by the platform as implementations of app lifecycle management, memory management, and the configuration that is required for the deployment to a car. Second, several support scripts and tools as a basic unit testing setup were part of the package. The Starter App itself was written in TypeScript, which extends JavaScript by several mechanisms like typing and improved tool support. Third, the Starter App contained basic examples of user interface (UI) elements as well as the implementation of basic vehicle APIs such as setting of destinations in the vehicle's navigation system. When a new developer entered the team, an experienced team member explained the basic idea of the Starter App and asked him or her to implement further examples without providing further initial information. That way, the new developer experienced all of the pitfalls and specifics of the platform on his own and learned how to solve a problem from existing examples. Since the Starter App already contained a comprehensive tooling, the developer did not need to spend any efforts on tasks such as unit testing setup or static code checks. When the developer was not able to solve a specific problem, he/she gained support from other team members. After this onboarding process, the Starter App was extended by further example and the new app developer was ready to start his work on the real product.

> *"We want to have a simple app that does not require to knowing the long history of BMW onboard software development but were you are able to add new features simply and fast even as beginner. This was the original vision which revealed the Starter App as onboarding tool. You can simply give it to the hands of a beginner and let him start." ~ App developer*

In January 2018, the location-based services team presented its Starter App to other app developers in a platform community meet-up. Subsequently, more and more additional teams used the

Authorized licensed use limited to: Technische Universitaet Muenchen. Downloaded on September 15,2020 at 09:57:52 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



starter for onboarding new team members or even as a starting point for their own developmental activities. All extensions that were created within the external usage of the Starter App were reviewed by the location-based service team, and subsequently, merged into the app.

> *"I started with the topic and after a while two or three people from our team joined me. Then, more colleagues from other app teams started participating. This resulted in a lengthy cooperation. We thereby learned how to cooperate in this context and were able to sharpen the scope of the project. I think, we really proved that this can work—and this is something I am really proud of—that multiple developers from different teams and even departments inside BMW and also on the supplier side can collaborate on work with code."* ∼ *App developer*

In March 2018, a member of the platform team extended the ignition guide in the developer portal by a tutorial on setting up a "hello world" app. The guide explained how to modify the Starter App to serve as generic base for any new app project. In the context of this work, the platform developer cooperated tightly with the location-based service team. Furthermore, the platform team decided to prescribe the utilization of the guide as mandatory for all new app projects. From this point on, the Starter App became the basis for every new app project that was started. The team subsequently developed a tool that automatically creates a basic app project, which is based on the original Starter App. This "App Wizard," which was finally released in December 2018, enables new developers to setup their own app project including comprehensive tooling and a proven basic architecture within a few minutes.

> *"We recognized that we needed more standardization for apps. The complex process for setting up a new app caused a high variation inside the app's architecture. Furthermore, high efforts were required for every new app project. The App Wizard should simplify and standardize this process."* —*Platform team member*

In April 2018, the Starter App was deployed in the integration environment of the platform. The app was, therefore, now available in hundreds of cars in the companies testing fleet. The BMW platform team subsequently recognized the advantage of the Starter App as feature tour for interested stakeholders. The platform's capabilities could also be easily demonstrated to current and potential stakeholders with no technical background. From March until September 2018, more and more contributions to the Starter App were made by the platform team. These updates contained improvements to several tools as well as examples for platform features. During this period, the location-based service team retained ownership of the code. In October 2018, both teams agreed that the responsibility of the starter app should be transferred to the platform team.

> *"Basically, we always considered the Starter App as a task of the platform team. However, it wasn't given to them at that time. This is why I appreciate that we agreed with the platform team to transfer the Starter App to them."* ∼ *App developer*

After a transfer period from October 2018 until January 2019, the platform team adopted the ownership of the Starter App. Since November 2018, new platform releases have contained a new version of the Starter App that has implemented all of the new platform features of the release. The new app version was tested comprehensively and has served as gatekeeper for the new platform versions. During the implementation of new platform features, the team recognized bugs and insufficiencies in the design and was able to fix them, before the actual platform release.

### *C. App Review*

From the beginning of the platform, the team installed an app review process that basically included three gates. The process was mandatory for all apps that should be deployed to a customer. In the first gate, the app developers needed to provide a rough description of their app. In a face-to-face meeting, the platform team evaluated whether the concept of the app was feasible with the given platform capabilities. If the meeting revealed that the concept was not realizable with the current platform, the app developers started a change request to the platform, adjusted their concept or declined their project. The second meeting that embodied the second gate usually took place several weeks after the first gate. Now the app developers were asked to provide a concrete technical concept including a proposal for their software architecture and backend communication concept. Furthermore, the app developers were asked in which countries their app should be available at release and whether they have already triggered the inherent processes with the respective business departments. After approval of this second gate, the actual development of the app started. At the end of 2016, some app developers deployed their first running versions on an end-to-end environment, meaning that the sources were uploaded to a repository that provided the apps as download to real cars. Here, the platform team installed a third gate, which includes the indication of certain resource consumption values by the apps as memory performance and the expected occupied space on the hard drive. The review team evaluated the provided values and urged teams with insufficient performance to improve their code. However, a rejection of a new app version was rare in the third gate. Even if the review team identified insufficiencies, it provided support for the app teams to improve their performance instead of blocking the app from a release.

> *"We just saw that several apps revealed multiple different problems due to a lack of standardization. The identified issues in the review showed us that a pure rejection of deployment will not bring any efforts. We needed to support them individually."* ∼ *Platform team member*

Moreover, the platform team decided to start detailed code reviews of every single app. However, while the third app review gate assimilated a mandatory step in the release process of a new app version, the code reviews were conducted on a sample base with a current version of the apps. It extended the existing gate process and supported apps in case of wrong or missing implementations of platform mechanisms or conceptual problems. For example, the code review team recognized that just a small number of apps implemented the lifecycle mechanisms in the way that it was intended by the platform team. This deficiency let to inflated booting durations of apps. Furthermore, multiple apps revealed insufficient test coverage in unit tests while others deployed development-specific code as part of their

Authorized licensed use limited to: Technische Universitaet Muenchen. Downloaded on September 15,2020 at 09:57:52 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



business logic. When the platform team recognized that such issues were not a problem of individual apps but were common practice, it decided to act. First, a comprehensive documentation on the most frequent findings was added to the developer portal. Furthermore, the reviewer scheduled meetings with app development teams to discuss the identified failures and propose potential solutions. At the end of the meeting, both sides agreed on a timeline for fixes and a repeated check according to this schedule.

> *"I consider the code review to be helpful. The guys pointed us to issues we hadn't considered so far and already provided potential solutions already. It really helped to increase the overall quality of our app."* ∼ *App developer*

Within the next month, the platform extended gate three several times as reaction to inadequate behavior of apps. To avoid an illegal implementation of libraries in their code, app developers needed to verify that all libraries that were implemented in their code were licensed as open source software. A check for the implemented libraries was, therefore, added to the third gate in April 2018.

> *"We realized that app teams weren't aware of the rules for implementing open source libraries in their code. The license checker tool should raise the attention on the one hand and enforce app teams to check their open source licenses to avoid legal problems with the implementation of unauthorized packages."* ∼ *Platform team member*

In March 2018, the platform team was confronted with the issue that several apps were not available in cars of the testing fleet. Even though the logging files revealed that the apps were downloaded and installed, no icon appeared in the user interface. After a few days of investigation, the team realized that the cause for the issue was rooted in new app development teams that copied code of existing apps as base for their own project. The copied code contained unique identifier of several UI elements. When these new apps were deployed to a system where other apps already implemented the identical unique identifiers, one app randomly disappeared. Besides the need for larger robustness of the platform, the team realized the need for further checks of released app versions. In parallel, the limited capacities of the review team on one side and the growing number of apps on the other hampered more detailed checks in the prevalent setup. In September 2018, the team, therefore, decided to build a web portal for apps that automates large parts of the review process. During the creation of the new portal, the team realized that extensibility embodies a fundamental requirement. Whenever the team identified another unintended behavior by apps that could be identified via automation, it created a new check.

> *"We just realized that we are checking the same things each and every time a new version of an app was released. Central automatization enables faster and deeper analysis of the app code."* ∼ *Platform team member*

The portal was finally released in April 2019 and replaced the manual process that was required for passing the third gate. Moreover, large parts of the code review were included in the automated checks. From then on, an app that failed the standardized checks was automatically rejected from the third review gate.

### *C. External Platform Contributions*

From the beginning of the platform development in summer 2016, the platform team worked with a development repository that was open for every interested party inside BMW. Although new parties needed to request access rights, the platform team never intended to keep the platform's code hidden. Hence, no request for access to the repository was ever denied. Besides platform artifacts such as the SDK and the runtime environment, the platform developers also stored example apps and development tools in the repository. Before the release of the Developer Portal in October 2017, the code repository incorporated a central source for documentation for app developers.

> *"In the beginning, the platform repository was indeed our most important source for information. All other documentation was hard to find, incomplete, or challenging to understand."* ∼ *App developer*

In July 2017, one app development team requested an API for one specific UI element that was not yet available in the SDK so far. The availability of the UI element was mandatory for the implementation of a central feature within the app. Hence, the missing API blocked the progress of the app team. The platform team, however, perceived extensive pressure due to general stability issues at that time. As a consequence, the team had to prioritize internal optimization issues above feature requests such as the required new UI element. When the app team realized that the platform team was not able to deliver the API in time, it decided to build it by its own. The platform team agreed to the approach and confirmed to support the app team in case of problems. After a few weeks, the app team finished its work on the API and provided the code to the platform team for review. The platform team approved the basic logic, though it noted several smaller issues such as the applied coding style or documentation. After fixes of the annotated issues, the new API was finally integrated in the SDK in September 2017. Seven months after this first external contribution to the platform, another app team faced similar issues. While the app required an extension of an API, the platform team was not able to accommodate this demand. Again, the app team reconciled their actions with the platform team and implemented the required extensions on their own. After a comprehensive review by the platform team, the code was integrated into the platform. These two incidents triggered discussions regarding an active opening of the platform in the platform team. Especially the platform team's limited capacities and the growing number of apps and respective requirements appeared as challenge that could be mitigated by external contributions. However, the platform team agreed that the code ownership needed to remain on its side. In August 2018, the team decided to simplify external contributions to the platform.

> *"We as platform developer appreciate external input. However, we recognize that we need clear rules for that. Otherwise, we spend too much efforts in discussions and in general communication with app developers that want to contribute to the platform."* ∼ *Platform team member*

Authorized licensed use limited to: Technische Universitaet Muenchen. Downloaded on September 15,2020 at 09:57:52 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



In addition to the benefits regarding capacity concerns, the platform team hoped for new inputs and refinements of the platform and its features by the app developers. The team, therefore, created a new discussion thread in BMW Answer to collect input from app developers. Even before this thread, apps were able to request changes to the platform. However, these change requests required complex processes in organization. The thread should enable direct exchange between platform developers and app developers, inspire a common discussion, and enhance small refinements and improvements that were too small or too insignificant for an official change request.

> *"We consider our platform as product for developers. For this reason, we want to know the needs and requirements of the app developer community. Of course, we need to validate and prioritize every request, although we consider them as valuable extensions of our general platform feature planning."* ∼ *Platform team member*

In February 2019, the platform team released contribution guidelines for the platform SDK. The guidelines aimed at facilitating external contributions by decreasing efforts for app developers as contributors as well as the platform team as reviewers. These contained detailed descriptions on the process that has to be followed by a contributor as well as a clear definition what has to be done for a contribution. Besides the actual code, the platform team asks for working sample code of the new feature, a comprehensive documentation in the BMW Docs and inherent release notes. Each contribution is still reviewed by the platform team.

> *"We just need to stay in control of the code. When an app developer who has contributed to the platform leaves BMW, we need to ensure the maintenance of this function. In this way, we prescribe rules for contributions and we need to certify all changes."* ∼ *Platform team member*

## V. Learning Mechanisms for Platform Owners

In this study, we identify mechanisms that enable learning of a platform owner by interacting with app developers. We therefore consider the platform owner as organization in the center of a digital platform ecosystem [11]. Learning in organizations is commonly defined as change in the organization's knowledge based on the organization's experience [41]. This might cause changes in behavior or cognition and effects an improvement in performance [52]. In our context, changes of cognition or behavior of the platform team cause improvements of the platform. The analysis of 25 events within two years in the context of the BMW app platform allowed us to identify three learning mechanisms: *transfer of perspective*, *transfer of knowledge,* and *transfer of artifacts*. Table II provides an overview of the identified learning mechanisms and how they are related to the illustrated events. The following sections contain descriptions of the three learning mechanisms. We illustrate the explanations with representative events from our case study episodes. Furthermore, we embed our findings in prevalent literature and elaborate the novelty of our findings.

### A. *Transfer of Perspective*

While a digital platform needs to maintain a certain degree of stability in the core, it simultaneously evolves constantly over time [14], [17]. For instance, when the platform team recognized flawed implementations of the app's lifecycle, it adjusted the associated boundary resources [35]. From a platform owner perspective, the extension of the platform gate process impeded flawed implementations and enhanced the platform's overall stability and performance during runtime. This procedure confirmed the common understanding that the refinement of boundary resources is not exclusively controlled by the platform owner, but is also influenced by app developers' behavior [17]. By adjusting the boundary resource, the platform owner secured control over the app developers and improved the overall quality of the platform [18]. However, since the inspection of the app's source code requires the platform team to actually understand the logic, it had to take the perspective of an app developer. In this way, the interaction of the app developer and platform owner affected a further mechanism. The review induced the platform team to consider the task of implementing the app's lifecycle management from an app developer's perspective. While the implementation of the lifecycle management appeared obvious to the platform team, it recognized that this was not the case for most of the app developers. The team thus realized that the documentation that should support app developers in the implementation of the lifecycle management was not sufficient. This change in cognition [41] motivated the platform owner to refine the provided platform resources embodied by the documentation on app lifecycle management. Considering the platform as generic toolset that enables the development of complementary features [11], an improvement of this enablement needs to be considered as improved performance of the platform [52].

This experience of "eat your own dog food" was even more concise in the Starter App episode. After taking over the responsibility of the app, the platform recognized insufficient platform design from an app developer perspective, every time it implemented new platform features with sample code in the app. In this way, the team became part of the app developers' community of practice [25]. The app developer perspective enabled the platform owner to recognize and refine insufficient platform design even before the feature was officially released to the app developers. One member of the platform team even stated that this change of perspective was one of the major motivation points for assuming responsibility for the Starter App.

Both episodes exemplify learnings of the platform team that caused improvements of the platform. Fig. 4 illustrates how the transfer of perspective enables learning for the platform owner. Interactions of the platform owner and app developer persuade the platform owner to take the perspective of an app developer and collects experience by own app development activities. These experiences extend the knowledge of the platform owner and causes changes in his cognition. The new cognition enables the platform owner to detect insufficiencies of the platform and improve it appropriately.

Authorized licensed use limited to: Technische Universitaet Muenchen. Downloaded on September 15,2020 at 09:57:52 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



TABLE II
LEARNING MECHANISMS OF THE PLATFORM TEAM

| **Episodes** | **Transfer of Perspective** | **Transfer of Knowledge** | **Transfer of Artifacts** |
|---|---|---|---|
| **Developer Portal** | • The platform team took the role of an unexperienced app developer for the creation of tutorials and documentation in BMW Docs<br>• Requests in BMW Answers established recognition of app developers' problems within the platform team<br>• Exchange with app developers in community meet-ups enabled platform team to understand app developers' problems from an app developer's perspective | • Reporting of bugs and requirements in BMW Answers<br>• Creation of Explore section in BMW Docs as result of frequent requests on platform capabilities in BMW Answers<br>• Creation of roadmap in BMW Docs as result of frequent requests on status of upcoming platform features in BMW Answers<br>• Surveys regarding most required new platform features in BMW Answers | • App developers wrote own documentation and submitted it in BMW Docs |
| **Starter App** | • Platform owner team recognized insufficient API design during the implementation of a new SDK feature in the Starter App<br>• Platform team recognized lack of sufficient infrastructure for developers | - | • A large amount of example code was transferred in the Starter App<br>• Multiple tools were transferred in context of the Starter App<br>• TypeScript was established as platform standard for app development with making the Starter App the default starting point for app development activities |
| **App Review** | • The platform team needed to understand the code of app developers and recognized flawed lifecycle management by the platform | • App developer reported insufficient memory measurement methodology<br>• The platform architect recognizes misinterpretation of app start concept by app developers<br>• Platform team recognized need to check of open source licenses | - |
| **External Contributions** | - | • Feature requests in BMW Answers trigger discussions in the platform team. Even though, a request is rejected by the platform team, the insufficiency of a certain aspect was communicated from app developers to the platform owner | • App developers contributed two UI-elements to the SDK |

Authorized licensed use limited to: Technische Universitaet Muenchen. Downloaded on September 15,2020 at 09:57:52 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



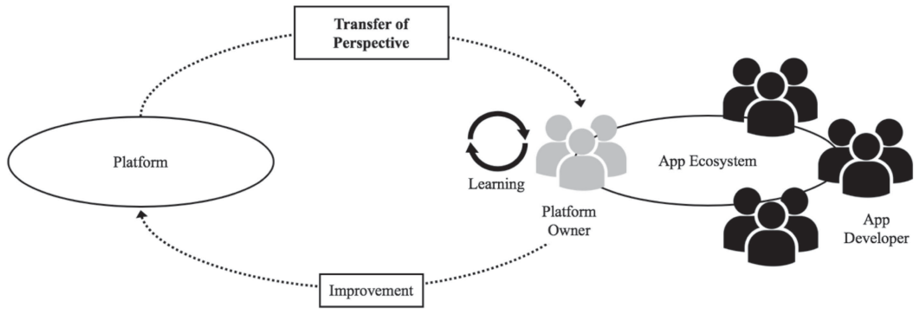



Fig. 4. Transfer of perspective learning mechanism.

### B. Transfer of Knowledge

A digital platform is considered as medium for knowledge transfer from the platform owner to app developers [20]. Platform boundary resources such as documentation or sample code provide information to app developers and enable app development by partners outside of the platform owner [11]. However, the episode on the collection of feedback on the latest SDK release in the BMW Answers proves that knowledge is also transferable from app developers to the platform owner via boundary resources. Considering the forum as resource for app developers, it enabled the emergence of a community with the knowledge-sharing routines and complementary endowments [57], [58]. The mutual exchange and interaction of different app development teams as well as the platform team enabled mutual insights and an overall increase of knowledge [27]. The engagement of the platform team within this community with its viral circulation of information [55] enabled the team to gather knowledge from app developers. When the team asked the community for feedback on the latest SDK version, it received spontaneous and comprehensive response. In this way, the platform team benefitted from the experience and the knowledge of the app developers. These learnings enabled the team to improve the SDK, which incorporates an enhancement of the team's performance due to learning [52], [70].

In a digital platform ecosystem, all apps are implemented on the same technical basis. This commonality enables not only scaling effects in form of value cocreation [71] and technological benefits as standardization [11], but also the potential for scaled learning mechanisms for the platform owner. The large number of responses on the request for feedback on the SDK enabled the platform team to consider multiple perspectives and specific issues of different app developers. The public discussions in the forum effected a reflection and maturing of specific feedback by other developers. In this way, the forum enabled not only pure knowledge transfer from app developers to the platform owner but also a maturing of the transferred knowledge, which decreased the likelihood of wrong learning [52] by the platform owner.

Finally, the episode on BMW Answers also illustrates that the transfer of knowledge does not require active initiative by the platform owner. While the request for feedback was actively started by the platform team, it learned from threads that were initiated by app developers. The community used the forum for reporting bugs in the platform SDK, asking for support in cases of insufficient documentation or requesting new platform features. These hints enabled the platform team to gather further knowledge about insufficiencies of platform resources and learn about potentials for improvements without any active involvement. However, the forum enabled the team to inquire about specific information from the requester.

The transfer of knowledge learning mechanism (see Fig. 5) starts with a collection of experiences of app developers. Interactions of app developers and platform owner enable a transfer of the gathered knowledge. The resulting change of the platform owner's cognition persuades the platform owner to improve the platform appropriately.

### C. Transfer of Artifacts

A prominent characteristic of digital platforms is external value creation, that is, value creation with actors outside the platform owner's organization [72]. A platform owner needs to foster external value creation and develop effective ways for capturing a share of that value to become successful [73]. Some platform owners absorb externally created innovation by implementing them directly into the platform core [74]. This coring mechanism is commonly considered as an act of competition [39], [75]. However, our case illustrates that the transfer of a complementary artifacts from app developers to the platform core may be a result of cooperative interaction.

The Starter App was initiated as project within one app development team. It should enhance the onboarding process of new app developers. After some time, the app development team realized that the Starter App was also of use also for developers outside of the team. Hence, it decided to share the artifact and make it available for everybody in the app community. When the platform team finally realized the need for improved onboarding for app developers, the Starter App had already become common practice among new app developers. The immature condition of the platform led to noncanonical work of the app developers that again triggered potentials that

Authorized licensed use limited to: Technische Universitaet Muenchen. Downloaded on September 15,2020 at 09:57:52 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



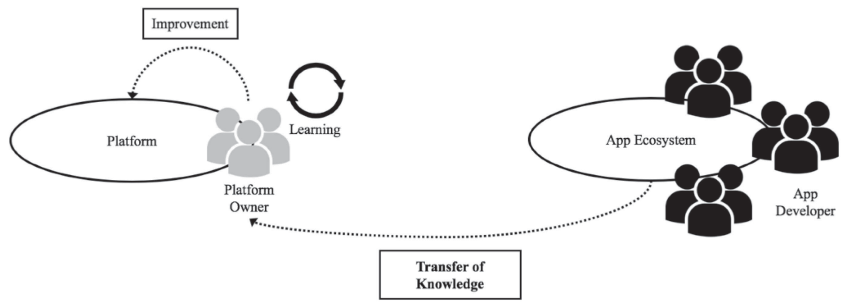



Fig. 5. Transfer of knowledge learning mechanism.

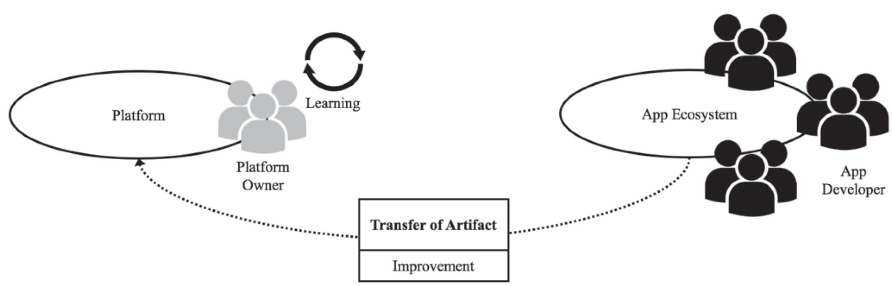



Fig. 6. Transfer of artifact learning mechanism.

could be exploited by the platform owner. Cooperation with the creators of the Starter App and the subsequent transfer of responsibility to the platform team enhanced the platform with the new resource. The platform owner again learned about potentials for improvement from interactions with app developers, though, in this case, the platform team integrated a solution that was already prefabricated by the app developers. This transfer of artifacts was also observable in the episode on in external contributions to the platform SDK. An externally crafted API for specific UI elements was integrated into the platform core two times. The accessibility of the platform development repository enabled app developers to add functionality and merge their change into the platform. However, the platform team realized that rules for the transfer of artifacts from app developers to the platform were needed to avoid integration of malicious or insufficient artifacts. Hence, it decided to create contribution guidelines to define clear rules for external contributions. In addition to protection of the platform, the contribution guidelines increased the transparency to potential contributors, which again facilitated contribution activities. An app developer could inform himself/herself about requirements of a contribution before the actual submission of a contribution to the platform team. This increased the likelihood of sufficient contributions and decreased the demand for clarification meetings between app developers and the platform team.

The episode on external contributions proved the relevance of accessibility as well as transparency for external contributions to a platform [38], [76]. However, the aspect of learnings for the platform owner had not been considered yet in the theoretical discussion on platform openness so far [77]. The platform's openness enabled a transfer of artifacts that again allowed the platform owner to learn from solutions of app developers. The creators of the Starter App focused on modular design in order that new developers can consider an example of one feature completely distinct from other features. According to the team, this design simplifies the understanding and accelerates the onboarding process for new developers, even though it is not recommendable for real apps. When the responsibility for the Starter App was transferred to the platform team, it consciously decided to stick with this kind of design. The team has learned that this kind of design appears optimal for new app developers. In this way, the platform team learned through the transfer of the artifact (see Fig. 6).

## VI. Discussion

The emergence of digital platforms constitutes an underrepresented aspect of research on digital platforms so far. While multiple studies describe platform governance in matured digital platform ecosystems, insights on organizations that create a new digital platform is rare [14]. The purpose of this study was to illuminate this gap by taking on the lens of the organizational learning theory. Even though it is generally acknowledged that interactions cause mutual potential for organizational learning [27], prevalent literature has exclusively focused on learning mechanisms on the app developer side [20], [35]. By considering

Authorized licensed use limited to: Technische Universitaet Muenchen. Downloaded on September 15,2020 at 09:57:52 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



the learning mechanisms of a platform owner in an emerging platform phase, we complement the prevalent theory. Our findings show in which way a platform owner enhances platform design by exploiting interactions with app developers through platform governance. Thereby, we confirm and extend the theory on platform boundary resources and platform openness as central aspects of platform governance [34]. Furthermore, we demonstrate the applicability of the organizational learning theory for the context of digital platforms and expose coherent peculiarities. Table III provides an overview on our contributions to the theory.

### A. Platform Boundary Resources as Medium for Organizational Learning

Prevalent platform literature emphasizes the distinction of app developers and platform owner regarding power and control [35], [76]. The distribution of power is balanced through several governance mechanisms [34] that are initially designed by the platform owner, influenced by app developers, and evolve over time [3]. Eaton *et al*. [17, p. 238] claim that "service systems with digital technology are ripe with political tensions among different actors trying to leverage their resources to influence others." While we acknowledge the significance of research on the balancing of power between app developers and platform owner, we claim that the aspect of cooperative interaction has been ignored so far. Just the platform owner is in charge to implement actual changes at the exposed resources. We argue that these adaptions are caused by learning effects at the platform owner side.

Platform boundary resources serve as medium between the platform owner and app developers. The artifacts are provided by the platform owner to enable app developers to build apps [35]. Foerderer *et al*. [20] illustrate in which ways knowledge is transferred from the platform owner to app developers. Our study confirms this transfer of knowledge via platform boundary resources. Furthermore, our findings extend the prevalent theory by demonstrating learning effects on the platform owner side. This indicates that the transfer of knowledge is not limited to transfer from platform owner to app developers but also occurs vice versa. We go further by stating that these learning effects are not limited to platform boundary resources [17], [18] but also to other platform parts that are not urgently exposed to app developers. Interactions with app developers cause learnings that also affect the platform core or processes inside the platform owner as illustrated in the flawed lifecycle event in the app review episode.

This rationale represents an important design guideline for platform boundary resources, which should be considered by new platform owner to leverage learning effects. We claim that platform owners, who exploit interactions with app developers through learning, will receive an advantage in competition with platform owners who ignore such learning effects [53], [54].

### B. Platform Openness Enables Organizational Learning

While the adoption of platform openness was considered as a logical consequence of environmental conditions in the digital platform ecosystem so far [32], we claim that a platform owner may adopt platform openness to leverage learning effects. By conscious and selective adaptions of openness, the platform owner enhances learning and leverages potential for platform improvements. Thereby, the platform owner may increase or decrease the platform's degree of openness to optimize learning effects for vertical openness as well as horizontal openness. Our research illustrates that learning occurs through communities-of-practice that involve app developers and the platform owner without any motivation regarding leveraging respective resources to influence the other side. This differentiates our findings from coring [39] or absorption [74] mechanisms that assume a shift of functionality from the app developer side into the platform and incorporates the leveraging of power by the platform owner. Our results indicate that both sides benefit from learnings of the platform owner, due to improvements of the platform. Since there is no replacement of complementary products in the process of a platform owner's learning, app developers may not fear becoming replaced by platform improvements. Our findings indicate that the degree of openness is not only relevant for control over a digital platform ecosystem but also for the exploitation of learning effects. We argue that this is especially relevant for an unexperienced platform owner. Hence, we appeal that considerations on platform openness should involve the effects of learning mechanisms, especially during the phase of platform emergence.

Even though we claim the importance of learning, we remark that it is not exclusively decisive for the chosen degree of platform openness. It can also follow a long-term strategy, which is not influenced by short-term learning effects. Karhu *et al.* [19] describe in which way Google started Android as open as possible to nurture the digital platform ecosystem. However, after attracting complementors and customers, Google started to decrease the degree of openness to increase the amount of control. This approach represents a long-term strategy that might just marginally be influenced by learning effects.

### C. Organizational Learning in Digital Platform Ecosystems Requires Orchestration

One goal of our research was to apply the organizational learning lens to the context of digital platforms. Our findings extend the organizational learning theory by platform dynamics. Our research confirms the occurrence of acquisitive as well as experimental learning in the context of digital platforms. However, while traditional product development organizations can enforce knowledge-sharing routines [49]–[51], digital platform ecosystems require orchestration of app developers via platform governance [4], [30]. Since digital platform ecosystems underly network effects [60], a platform owner should facilitate interactions to leverage learning. Thereby, the platform owner needs to consider the incentives for app developers to exploit learning mechanisms. Furthermore, the platform owner needs to avoid wrong learnings by designing appropriate boundary resources. This comes back to the relevance of platform boundary resources as medium for learning in digital platform ecosystems. Here, or findings again indicate the importance conscious boundary

Authorized licensed use limited to: Technische Universitaet Muenchen. Downloaded on September 15,2020 at 09:57:52 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



TABLE III
OVERVIEW ON CONTRIBUTIONS TO THE THEORY

| | **Platform Boundary Resources** | **Platform Openness** | **Organizational Learning** |
|---|---|---|---|
| **Transfer of Perspective** | *Confirmation:* Interactions of platform owner and app developers trigger refinements of boundary resources [17, 18] (Example: Recognition of insufficient infrastructure in starter app episode).<br><br>*Extension:* Interactions of platform owner and app developers trigger refinements that are not limited to boundary resources [17, 18] but also affect the platform core (Example: Flawed lifecycle in app review episode). | *Confirmation:* Degree of openness in form of transparency is adjusted over time [19, 32] (Example: Community meet-ups in developer portal episode). | *Confirmation:* Experimental learning of the platform owner enabled the acquisition of implicit knowledge that was not articulated by app developers [42-44] (Example: Insufficient API design in starter app episode). |
| **Transfer of Knowledge** | *Confirmation:* Platform boundary resources serve as medium for knowledge exchange between the platform owner and app developers [20] (Example: Establishment of Q&A forum in developer portal episode).<br><br>*Extension:* Platform boundary resources enable knowledge transfer from app developers to the platform owner [20, 35] (Example: Insufficient memory measurement methodology in app review episode). | *Confirmation:* Degree of openness (vertical) in form of transparency is adjusted over time [19, 32] (Example: Creation of roadmap in developer portal episode).<br><br>*Extension:* Adjustment of openness can be caused by learning effects on the platform owner side [19, 32] (Example: Creation of explore section as result of frequent requests in developer portal episode).<br><br>*Extension:* Adjustments of openness as act of platform governance is especially relevant during platform emergence to foster learning of the platform owner [21, 22, 32] (Example: Establishment of Q&A forum in developer portal episode). | *Confirmation:* The platform owner gained knowledge, which preexisted on the app developer side and implemented it to improve the platform. This act represents acquisitive learning [42] (Example: Feature requests in external contributions episode).<br><br>*Extension:* Establishing knowledge-sharing routines [57, 58] for organizational learning is relevant in digital platform ecosystems and implemented via platform boundary resources (Example: Creation of Q&A forum and in developer portal episode).<br><br>*Extension:* The platform owner needs to foster learning by incentivizing app developers [29, 30] (Example: Participation in community meet-ups). |
| **Transfer of Artifact** | *Confirmation:* Learnings of app developers trigger tuning effects of boundary resources (Example: App developer submitted documentation in der developer portal episode).<br><br>*Extension:* Interaction via boundary resources is not exclusively confrontational (securing vs. resourcing) [17, 18] but may reveal a cooperative character (Example: Starter app episode). | *Confirmation:* Degree of openness (horizontal) is adjusted over time [32] (Example: External contributions episode).<br><br>*Extension:* Learning on platform owner side is supported selective horizontal openness [32] (Example: Starter app episode).<br><br>*Extension:* Adjustments of openness as act of platform governance is especially relevant during platform emergence to foster learning of the platform owner [21, 22, 32] (Example: External contributions episode). | *Confirmation:* Platform owner needs to steer interactions to avoid acquisition of wrong knowledge [48] (Example: External contributions episode).<br><br>*Extension:* Platform owner needs to foster learning by orchestrating app developers [51] (Example: External contributions episode). |

Authorized licensed use limited to: Technische Universitaet Muenchen. Downloaded on September 15,2020 at 09:57:52 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



resources design to leverage learning effects, especially for unexperienced platform owners.

## VII. Conclusion

This article considered a case of an emerging digital platform for automotive onboard apps. The qualitative character of our study commonly indicated the limitations on the generalizability of our findings. Rather than establishing a general theory, our research strived for explorative insights that require further confirmation by additional research [62]. Furthermore, our setup revealed certain specifics that need to be clarified to sharpen the scope of this work. First, our studies were embedded in the context of an incumbent organization that revealed legacy of its successful past in the automotive industry. Hence, our findings may differ from a similar study in the context of start-ups or digital native organizations with large experience with the development of digital services. Second, while our research was embedded in an internal setup inside BMW, we considered our results also relevant for digital platform ecosystems that grant access to third-party developers. However, our findings indicated that the required emergence of communities-of-practice demand trust of both sides [25], which needs to be established by the platform owner. A comprehensive investigation of learning processes in more open platform settings appeared as promising avenue further for research.

Moreover, the context of our study was dictated by the choice of the researched context, which was scoped by the availability of data. Similarly, the selection of interview partners was limited by their willingness to participate. Glaser [69] emphasized that the application of the grounded theory revealed a “mid-term” theory, which was not universally applicable but needed further refinements and constant comparison through data from new contexts [78]. The learning mechanism of a platform owner in the context of an emerging digital platform was an addition to the prevalent literature on platform boundary resources, platform openness, and organizational learning. However, the specific context needed to be emphasized. Further research in other contexts should help to generate a more generalized theory.

The generative character of digital platforms enabled incumbent companies to enrich their products with digital services and satisfy evolved customer expectations. However, a new platform owner needs to gather knowledge on the design and operation of a digital platform. By conducting a grounded theory approach, our study sheds light on a platform owner’s learning mechanisms in the context of a digital platform for automotive onboard apps at the globally operating car manufacturer BMW.

1) The transfer of perspective described the activities in which the platform owner learned through experience from an app developer perspective, and subsequently, improved the platform.
2) The transfer of knowledge described the activities in which experiences of app developers were transferred to the platform owner, which caused a change of cognition and subsequent improvements of the platform.
3) Transfer of artifact described the activities in which app developer created own improvements and transferred them to the platform owner, which again caused a change in cognition of the platform owner and an improvement of the platform.

Our results contributed to the current discussion on sufficient platform design. The understanding of the identified learning mechanisms will help new platform owners to foster fruitful interactions with their app developers and enhance the digital platform design.

## References


[1] A. Gawer and M. Cusumano, “Industry platforms and ecosystem innovation,” *J. Product Innov. Manag.*, vol. 31, no. 3, pp. 417–433, 2014.

[2] E. den Hartigh, J. R. Ortt, G. van de Kaa, and C. C. M. Stolwijk, “Platform control during battles for market dominance: The case of Apple versus IBM in the early personal computer industry,” *Technovation*, vol. 48, pp. 4–12, 2015.

[3] A. Srinivasan and V. N. Venkatraman, “Architectural convergence and platform evolution: Empirical test of complementor moves in videogames,” *IEEE Trans. Eng. Manag.*, vol. 67, no. 2, pp. 266–282, May 2020.

[4] T. L. Huber, T. Kude, and J. Dibbern, “Governance practices in platform ecosystems: Navigating tensions between cocreated value and governance costs,” *Inf. Syst. Res.*, vol. 28, no. 3, pp. 563–584, 2017.

[5] F. Svahn, L. Mathiassen, and R. Lindgren, “Embracing digital innovation in incumbent firms: How Volvo cars managed competing concerns,” *MIS Quart.*, vol. 41, no. 1, pp. 239–255, 2017.

[6] A. Hein, M. Schreieck, M. Wiesche, M. Böhm, and H. Krcmar, “The emergence of native multi-sided platforms and their influence on incumbents,” *Electron. Markets*, vol. 29, pp. 631–647, 2019.

[7] A. S. Cui, K. Chan, and R. Calantone, “The learning zone in new product development,” *IEEE Trans. Eng. Manag.*, vol. 61, no. 4, pp. 690–701, Nov. 2014.

[8] T. Williams, “How do organizations learn lessons from projects—and do they?” *IEEE Trans. Eng. Manag.*, vol. 55, no. 2, pp. 248–266, Jun. 2008.

[9] C. L. Huntley, “Organizational learning in open-source software projects: An analysis of debugging data,” *IEEE Trans. Eng. Manag.*, vol. 50, no. 4, pp. 485–493, Nov. 2003.

[10] C. Y. Baldwin and C. J. Woodard, “The architecture of platforms: A unified view,” in *Platforms, Markets and Innovation*, A. Gawer, Ed. Cheltenham, U.K.: Edward Elgar, 2009, pp. 19–44.

[11] A. Tiwana, *Platform Ecosystems: Aligning Architecture, Governance, and Strategy*. Waltham, MA, USA: Elsevier, 2014.

[12] A. Hein *et al.*, “Digital platform ecosystems,” *Electron. Markets*, vol. 30, pp. 87–98, 2020.

[13] H. Chesbrough and M. Bogers, “Explicating open innovation: Clarifying an emerging paradigm for understanding innovation ” in *New Frontiers in Open Innovation*. Oxford, U.K.: Oxford Univ. Press, 2014, pp. 3–28.

[14] M. de Reuver, C. Sørensen, and R. C. Basole, “The digital platform: A research agenda,” *J. Inf. Technol.*, vol. 33, no. 2, pp. 124–135, 2017.

[15] A. Mohagheghzadeh and F. Svahn, “Transforming organizational resource into platform boundary resources,” in *Proc. 24th Eur. Conf. Inf. Syst.*, stanbul, Turkey, 2016.

[16] A. Tiwana and A. A. Bush, “Spotting lemons in platform markets: A conjoint experiment on signaling,” *IEEE Trans. Eng. Manag.*, vol. 61, no. 3, pp. 393–405, Aug. 2014.

[17] B. Eaton, S. Elaluf-Calderwood, C. Sørensen, and Y. Yoo, “Distributed tuning of boundary resources: The case of Apple’s iOS service system,” *MIS Quart.*, vol. 39, no. 1, pp. 217–244, 2015.

[18] O. Henfridsson and A. Ghazawneh, “Balancing platform control and external contribution in third-party development: The boundary resources model,” *Inf. Syst. J.*, vol. 23, no. 2, pp. 173–192, 2013.

[19] K. Karhu, R. Gustafsson, and K. Lyytinen, “Exploiting and defending open digital platforms with boundary resources: Android’s five platform forks,” *Inf. Syst. Res.*, vol. 29, no. 2, pp. 479–497, 2018.

[20] J. Foerderer, T. Kude, S. W. Schuetz, and A. Heinzl, “Knowledge boundaries in enterprise software platform development: Antecedents and consequences for platform governance,” *Inf. Syst. J.*, vol. 29, no. 1, pp. 119–144, 2019.

[21] N.-B. Schirrmacher, J. Ondrus, and T. Kude, “Launch strategies of digital platforms: Platforms with switching and non-switching users,” in *Proc. Eur. Conf. Inf. Syst.*, Guimarães, Portugal, 2017.

[22] C. Stummer, D. Kundisch, and R. Decker, “Platform launch strategies,” *Bus. Inf. Syst. Eng.*, vol. 60, no. 2, pp. 167–173, 2018.


Authorized licensed use limited to: Technische Universitaet Muenchen. Downloaded on September 15,2020 at 09:57:52 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.




[23] A. Rausch, C. Bartel, S. Herold, H. Klus, and D. Niebuhr, "From software systems to complex software ecosystems: Model- and constraint-based engineering of ecosystems," in *Perspectives on the Future of Software Engineering*. Berlin, Germany: Springer, 2013, pp. 61–80.

[24] H. A. Simon, "Bounded rationality and organizational learning," *Org. Sci.*, vol. 2, no. 1, pp. 125–134, 1991.

[25] J. S. Brown and P. Duduid, "Organizational learning and communities-of-practice: Toward a unified view of working, learning, and innovation," *Org. Sci.*, vol. 2, no. 1, pp. 40–57, 1991.

[26] J. Foerderer, "Relational third party governance: Evidence from Apple's world wide developer conference," in *Proc. Platform Strategy Res. Symp.*, Boston, MA, USA, 2017.

[27] J. G. March, "Exploration and exploitation in organizational learning," *Org. Sci.*, vol. 2, no. 1, pp. 71–87, 1991.

[28] B. Glaser and A. Strauss, *The Discovery of Grounded Theory*. Chiacago, IL, USA: Aldine, 1967.

[29] D. Dellermann, C. Jud, and K. M. Popp, "Why don't they join? Analyzing the nature and consequences of complementors' costs in platform ecosystems," in *Proc. 37th Int. Conf. Inf. Syst.*, Dublin, Ireland, 2016.

[30] T. Kude, J. Dibbern, and A. Heinzl, "Why do complementors participate? An analysis of partnership networks in the enterprise software industry," *IEEE Trans. Eng. Manag.*, vol. 59, no. 2, pp. 250–265, May 2012.

[31] D. Tilson, K. Lyytinen, and C. Sørensen, "Research commentary—Digital infrastructures: The missing is research agenda," *Inf. Syst. Res.*, vol. 21, no. 4, pp. 748–759, 2010.

[32] J. Ondrus, A. Gannamaneni, and K. Lyytinen, "The impact of openness on the market potential of multi-sided platforms: A case study of mobile payment platforms," *J. Inf. Technol.*, vol. 30, no. 3, pp. 260–275, 2015.

[33] A. Rausch, J. Müller, D. Niebuhr, S. Herold, and U. Goltz, "IT ecosystems: A new paradigm for engineering complex adaptive software systems," in *Proc. 6th IEEE Int. Conf. Digit. Ecosyst. Technol.*, Campione d'Italia, Italy, 2012.

[34] M. Schreieck, M. Wiesche, and H. Krcmar, "Design and governance of platform ecosystems-key concepts and issues for future research," in *Proc. 24th Eur. Conf. Inf. Syst.*, Istanbul, Turkey, 2016.

[35] A. Ghazawneh and O. Henfridsson, "Governing third-party development through platform boundary resources," in *Proc. 31st Int. Conf. Inf. Syst.*, St. Louis, MO, USA, 2010.

[36] K. Boudreau, "Open platform strategies and innovation: Granting access versus devolving control," *Manag. Sci.*, vol. 56, no. 10, pp. 1849–1872, 2010.

[37] T. Eisenmann, G. Parker, and M. Van Alstyne, "Platform envelopment," in *Platforms, Markets and Innovation*, A. Gawer, Ed. Cheltenham, U.K.: Edward Elgar, 2011, pp. 131–162.

[38] M. A. Zaggl, T. G. Schweisfurth, and C. Herstatt, "The dynamics of openness and the role of user communities: A case study in the ecosystem of open source gaming handhelds," *IEEE Trans. Eng. Manag.*, vol. 67, no. 3, pp. 712–723, Aug. 2020.

[39] B. Bender and N. Gronau, "Coring on digital platforms—Fundamentals and examples from the mobile device sector," in *Proc. 38th Int. Conf. Inf. Syst.*, Seoul, South Korea, 2017.

[40] G. Parker, M. Van Alstyne, and X. Jiang, "Platform ecosystems: How developers invert the firm," *MIS Quart.*, vol. 41, no. 1, pp. 255–266, 2017.

[41] L. Argote and E. Miron-Spektor, "Organizational learning: From experience to knowledge," *Org. Sci.*, vol. 22, no. 5, pp. 1123–1137, 2011.

[42] Y. Li, C. Zhang, Y. Liu, and M. Li, "Organizational learning, internal control mechanisms, and indigenous innovation: The evidence from china," *IEEE Trans. Eng. Manag.*, vol. 57, no. 1, pp. 63–77, Feb. 2010.

[43] G. G. Dess, R. D. Ireland, S. A. Zahra, S. W. Floyd, J. J. Janney, and P. J. Lane, "Emerging issues in corporate entrepreneurship," *J. Manag.*, vol. 29, no. 3, pp. 351–378, 2003.

[44] S. A. Zahra, A. P. Nielsen, and W. C. Bogner, "Corporate entrepreneurship, knowledge, and competence development," *Entrepreneurship Theory Practice*, vol. 23, no. 3, pp. 169–189, 1999.

[45] R. M. Grant, "Toward a knowledge-based theory of the firm," *Strategic Manag. J.*, vol. 17, pp. 109–122, 1996.

[46] P. S. Adler and K. B. Clark, "Behind the learning curve: A sketch of the learning process," *Manag. Sci.*, vol. 37, no. 3, pp. 267–281, 1991.

[47] L. Argote, S. Beckman, and D. Epple, "The persistence and transfer of learning in industrial settings," *Manag. Sci.*, vol. 36, no. 2, pp. 140–154, 1990.

[48] G. P. Huber, "Organizational learning: The contributing processes and the literatures," *Org. Sci.*, vol. 2, no. 1, pp. 88–115, 1991.

[49] Y.-S. Chuang, "Learning and international knowledge transfer in latecomer firms: The case of Taiwan's flat panel display industry," *IEEE Trans. Eng. Manag.*, vol. 61, no. 2, pp. 261–274, May 2014.

[50] I. Maria, B. Freitas, and R. Fontana, "Formalized problem-solving practices and the effects of collaboration with suppliers on a firm's product innovation performance," *J. Product Innov. Manag.*, vol. 35, no. 4, pp. 565–587, 2018.

[51] T. Moellers, C. Visini, and M. Haldimann, "Complementing open innovation in multi-business firms: Practices for promoting knowledge flows across internal units," *R&D Manag.*, vol. 33, no. 2, pp. 1–20, 2018.

[52] M. Crossan, H. Lane, and R. White, "Organizational learning: Dimensions for a theory," *Int. J. Org. Anal.*, vol. 3, no. 4, pp. 337–360, 1995.

[53] P. Knight, "A systemic approach to professional development: Learning as practice," *Teaching Teacher Educ.*, vol. 18, no. 3, pp. 229–241, 2002.

[54] D. H. Henard and D. M. Szymanski, "Why some new products are more successful than others," *J. Marketing Res.*, vol. 38, no. 3, pp. 362–375, 2001.

[55] B. Levitt and J. G. March, "Organizational learning," *Annu. Rev. Sociol.*, vol. 14, no. 1, pp. 319–338, 1988.

[56] M. Ceccagnoli, "Co-creation of value in a platform ecosystem: The case of enterprise software," *MIS Quart.*, vol. 36, no. 1, pp. 263–290, 2012.

[57] J. H. Dyer and H. Singh, "The relational view: Cooperative strategy and sources of interorganizational competitive advantage," *Acad. Manage. Rev.* vol. 23, no. 4, pp. 660–679, 1998.

[58] L. F. Mesquita, J. Anand, and T. H. Brush, "Comparing the resource-based and relational views: Knowledge transfer and spillover in vertical alliances," *Strategic Manage. J.*, vol. 29, no. 9, pp. 913–941, 2008.

[59] A. Smedlund and H. Faghankhani, "Platform orchestration for efficiency, development and innovation," in *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, 2015, pp. 1380–1388.

[60] G. G. Parker and M. W. Van Alstyne, "Two-sided network effects: A theory of information product design," *Manage. Sci.*, vol. 51, no. 10, pp. 1494–1504, 2005.

[61] G. Parker, M. Van Alstyne, and S. P. Choudary, *Platform Revolution: How Networked Markets are Transforming the Economy and How to Make Them Work for You*. New York, NY, USA: WW Norton & Co, 2016.

[62] R. K. Yin, *Case Study Research Design and Methods*. Thousand Oaks, CA, USA: Sage, 2009.

[63] K. M. Eisenhardt, "Building theories from case study research," *Acad. Manage. Rev.*, vol. 14, pp. 532–550, 1989.

[64] J. Gläser and G. Laudel, *Experteninterviews und qualitative Inhaltsanalyse als Instrumente rekonstruierter Untersuchungen*. Wiesbaden, Germany: VS Verlag für Sozialwissenschaften, 2009.

[65] L. Mathiassen, "Collaborative practice research," *Inf. Technol. People*, vol. 15, no. 4, pp. 321–345, 2002.

[66] M. Wiesche, M. C. Jurisch, P. Yetton, and H. Krcmar, "Grounded theory methodology in information systems research," *MIS Quart.*, vol. 31, no. 3, pp. 685–701, 2017.

[67] B. Glaser, *Theoretical Sensitivity: Advances in the Methodology of Grounded Theory*. New York, NY, USA: Sociology Press, 1978.

[68] G. Allan, "The Legitimacy of Grounded Theory," in *Proc. Eur. Conf. Res. Methods*, 2006, pp. 1–8.

[69] B. Glaser, *Basics of Grounded Theory Analysis: Emergence vs. Forcing*. New York, NY, USA: Sociology Press, 1992.

[70] P. Huang, A. R. Tafti, and S. Mithas, "Platform sponsor investments and user contributions in knowledge communities: The role of knowledge seeding," *MIS Quart.*, vol. 42, no. 1, pp. 213–240, 2018.

[71] R. F. Lusch and S. L. Vargo, "Service-dominant logic: reactions, reflections and refinements," *Marketing Theory*, vol. 6, no. 3, pp. 281–288, 2006.

[72] G. Parker and M. Van Alstyne, "Innovation, openness, and platform control," *Manage. Sci.*, vol. 64, no. 7, pp. 3015–3032, 2017.

[73] M. Schreieck, M. Wiesche, and H. Krcmar, "The platform owner's challenge to capture value—Insights from a business-to-business IT platform," in *Proc. 38th Int. Conf. Inf. Syst.*, Seoul, South Korea, 2017.

[74] M. Schreieck and M. Wiesche, "How established companies leverage IT platforms for value co-creation—Insights from banking," in *Proc. 25th Eur. Conf. Inf. Syst.*, Guimaraes, Portugal, 2017.

[75] T. Saarikko, "Platform provider by accident a case study of digital platform coring," *Bus. Inf. Syst. Eng.*, vol. 58, no. 3, pp. 177–191, 2016.

[76] A. Benlian, D. Hilkert, and T. Hess, "How open is this platform? The meaning and measurement of platform openness from the complementors' perspective," *J. Inf. Technol.*, vol. 30, no. 3, pp. 209–228, 2015.

[77] D. Soto Setzke, M. Böhm, and H. Krcmar, "Platform openness: A systematic literature review and avenues for future research," in *Proc. Int. Conf. Wirtschaftsinformatik*, Siegen, Germany, 2019.

[78] R. M. Davison and M. G. Martinsons, "Context is king! Considering particularism in research design and reporting," *J. Inf. Technol.*, vol. 31, no. 3, pp. 241–249, 2016.


Authorized licensed use limited to: Technische Universitaet Muenchen. Downloaded on September 15,2020 at 09:57:52 UTC from IEEE Xplore. Restrictions apply.

# Pure Coding Pleasure: How BMW Involves App Developers in the Design of Automotive Onboard APIs

*Completed Research*

**Niklas Weiss**
Technical University of Munich
Boltzmannstraße 3, 85748 Garching
niklas.weiss@tum.de

**Manuel Wiesche**
Technical University of Munich
Boltzmannstraße 3, 85748 Garching
wiesche@in.tum.de

**Helmut Krcmar**
Technical University of Munich
Boltzmannstraße 3, 85748 Garching
krcmar@in.tum.de

## Abstract

Digital transformation requires incumbent companies to accelerate the development of their digital products. The automotive industry is a prominent example of this change. Even though, research frequently considers digital transformation of organizations, there are rare perceptions on the change of the actual technology. This action research study provides deep insights in the endeavors of the global operating car manufacturer BMW towards a generic design of onboard application programming interfaces (APIs) which should enhance accelerated development of digital products. To help the firm embrace generativity, we therefore infused lead user involvement theory and API evaluation criteria into the API design team. As a result, we present 5 majorly refined APIs, which are implemented in the BMW App SDK. Further, we identified critical challenges and benefits for the involvement of lead users in the design of enabling technology within an incumbent company.

**Keywords**

API design, lead user involvement, action research, incumbent firms, digital platforms.

## Introduction

### *Motivation*

Digital transformation affects incumbent enterprises. It challenges firms from multiple domains (Schreieck and Wiesche 2017; Schreieck et al. 2018). The automotive industry is one of the most prominent examples of this development. The acceleration of digital product development is one major change that has to be faced by the car manufacturers (Svahn et al. 2017). Several car manufacturers have introduced systems of deploying new versions of the vehicle's software over the internet (AutomotiveWorld 2019). The system enables regular update of the existing software and the deployment of new digital products to a car even after the car has been sold to a customer. Furthermore, the manufacturers integrate service-oriented software systems in the form of digital platforms into their vehicles to enable flexible deployment of digital products offered in the form of apps (BMW 2018; Digitaltrends 2017). In this context, we understand digital platforms as "extensible codebase that provides core functionality shared by apps that interoperate with it, and the interfaces through which they interoperate" (Tiwana 2014). Thereby, after the initial hand-over to the customer, the codebase in the car can be extended. In this way, the modular extensibility of the system decouples the short-term app development process from the long-term vehicle development process. Already, in a traditional automotive development approach, different software modules interoperate via application programming interfaces (APIs). These APIs expose functionality of one software module to another. However, the utilization of an API is defined in advance. Additionally, the number of an API's consumers is limited to a small number of experts and the primary criterion for assessing an interface is its functionality. The introduction of a modular extensible software architecture that furtherly enables the deployment of unknown and new utilizations of an API in the future, entails different requirements. The capabilities and limitations of an API need to be transparent to all stakeholders including potential app developers without any domain specific expert knowledge (Pühler 2011; Schlachtbauer et al. 2012). Hence, incumbents need to refine their existing API design. This practical challenge raises this question; how can an incumbent organization evolve its existing system design in the context of digital transformation?

Previous research reveals that multiple traditional companies has failed to transform their technology adequately (Henderson and Clark 1990; Tripsas and Gavetti 2000). Furthermore, literature considers the involvement of users as essential success factor for designing new information systems (IS) (Bano and Zowghi 2014; von Hippel 2005). Therefore, we propose the approach of lead user involvement for the refinement of an existing technology, in this case incumbent APIs for an automotive onboard software system. The evolvement of an existing system entails that this system is already in use in practice. A certain amount of lead users may have identified potential improvements for existing solutions (Lüthje and Herstatt 2004; von Hippel 1986). In the case of automotive onboard APIs, app developers that already implemented such APIs embody that kind of lead users. This study strives to identify the challenges and benefits that emerge in the involvement of lead users during the analysis and refinement of automotive onboard APIs. For doing so, we applied an action research approach (Baskerville 1999; Frank et al. 1998; Keng and Rossi 2011) within the software development department at the global car manufacturer BMW.

In the remainder of the paper, first, we described the theoretical concepts of APIs and lead user involvement. Then, we described BMW's infotainment system, followed by a detailed description of the research project setup. Subsequently, we described the detailed API analysis and design process. Finally, we presented our results, findings, and discussions.

## Background

### *Relevance of API Design*

Application Programming Interfaces (APIs) exposes a system's core resources as a service to stimulate generativity (Henfridsson and Ghazawneh 2013). The usage of an APIs does not have to be determined by design but can be utilized in multiple ways. In this way, APIs are able foster innovation by enabling the development of complementary features (Um et al. 2013). Similarly, the utilization by app developers affect the design of the API (Eaton et al. 2015). APIs potentially generic character enables scalability of operations as well as flexibility in acquiring new strategic partners and realizing new business goals (Iyer and Subramaniam 2015). However, the actual design of APIs is critical to maximizing its potentials. Poor API design results in increased development costs during its implementation by apps (Henning 2009). If these costs exceed potential benefits of a complementary feature, it will not be created. In this way innovation is blocked and the attractiveness for end users remains constant or even decreases (Tiwana 2014).

### *Lead User Involvement*

Lead user involvement is a principle often used in system design research. To reduce the risk of failure, the alignment of product development activities with the needs of actual and potential users is crucial (Jaworski and Ajay 1993). A user-centric focus fosters quality, reliability and uniqueness of a product (Li and Calantone 1998). The involvement of users already in early phases of an innovation project enhances these potentials (Herstatt and von Hippel 1992; von Hippel et al. 1999). In their comprehensive literature review, Bano and Zowghi (2014) identify five relevant perspective of user involvement. The *psychological* perspective considers aspects as the users' motivation or interests to participate. Second, the involvement of users requires appropriate *management*. Moreover, the *political* perspective considers the degree of power that is given to the involved users. The purpose of user involvement can differ for various groups of users. Therefore, *cultural* aspects need to be considered. Finally, different intensities of user involvement require specific *methodological* approaches. The concept of lead user involvement originally is rooted in marketing research, and it considers the involvement of users whose present strong needs will become general in the future (von Hippel 1986). Lead users are well-qualified and motivated to contribute to an improvement of the status quo (von Hippel 1986). Their prevalent own need enables them to innovate (von Hippel 2005). Since lead users embody the leading edge of a market regarding important market trends; their participation in product development activities facilitates innovation, attractive for future users. In our study, we consider lead users as app developers who implemented the exposed interfaces at a large scale or multiple times. Lead user involvement is mainly considered for enhancing innovation. However, von Hippel (2005) emphasizes that many of the concepts regarding innovations communities "apply to information communities as well." Considering the app developer community as an information community which utilizes APIs, we strive to involve their expertise in the refinement of already existing APIs.

# The BMW Case

## *Initial Situation*

This study considers APIs of a digital platform for onboard automotive apps of a global car manufacturing company, BMW. The platform is part of the BMW OS 7.0, the company's latest infotainment system, released in July 2018 with the release of the latest BMW X5 series. Upcoming models from the manufacturer will run the system. The car's central electronic control unit powers the digital platform, and it also enables modular wireless deployment of apps to the digital platform. By exposing multiple functionalities of the car via APIs, the platform provides the base for a broad spectrum of use cases. During the release of the platform in summer 2018, more than 20 apps were available, providing services such as a parking lot finder[1], music streaming[2], Microsoft Office 365[3] and apps for different BMW service calls[4]. The number of available apps is steadily increasing since the initial release. Although the platform was not opened towards third-party developers, multiple stakeholders around the globe are involved in creating apps for the platform. The app developer community is made up of over 120 active members.

The first author of this study is actively involved in the platform's development team as a Ph.D. researcher. In the course of this study, we interviewed eight expert app developers from October until December 2017, and the goal of the interview was to identify challenges at the emergence of the platform. The result of the investigation showed that the platform's APIs has room for significant improvement. The developers pointed out two fundamental issues with the platform's APIs. First, the design of the API was tailored for a single or small amount of use cases.

*"Sometimes it is hard to understand how and why an interface is designed in the way it is. For example, one specific value – which is actual available somewhere in the car - is just missing in the interface while in other cases one value is available in three different ways. However, most probably the missing value was simply not required by the initial use case while in the other case there were three different use cases, which required the value in three different ways."*

Second, the design of the interface was decided by the API provider and the party requesting for the use case. The interest of requesting party is speedy delivery with a minimal budget. The evaluation criteria for the API is the feasibility of the use case while usability is subsidiary. However, abstracting usability was not considered due to the small number of consumers.

*"The implementation of some APIs requires massive efforts. You need to write tons of boilerplate code which don't reveal any functionality towards the application. For example, the implementation of a notification banner in the UI requires more than 350 lines of code."*

## *Project Setup*

The platform team reviewed the result of the interview conducted and decided to start a project to create new and high-quality APIs in the platform SDK. The project was conducted in research cooperation,

[1] https://www.bmw-connecteddrive.de/app/index.html#/portal/store/Base_Connected_Parking from 19.02.2019

[2] https://www.bmw-connecteddrive.de/app/index.html#/portal/store/Base_Connected_Music from 19.02.2019

[3] https://www.bmw-connecteddrive.de/app/index.html#/portal/store/Base_ExchangeOffer from 19.02.2019

[4] https://www.bmw-connecteddrive.de/app/index.html#/portal/store/Base_AssistNoTPEGOffer from 19.02.2019

applying action research methodology. This approach allowed the active involvement of the researcher directly in the project and enables deep insights into the actual design process. Hence, the first author of this study took the role of an active architect who was responsible for designing and refining APIs in an iterative approach while an internal BMW engineer was the project lead. This setup embodied the client-system infrastructure, which is required by any action research approach (Baskerville & Wood-Harper, 1998). The project started on March 15, 2018 and ended December 21, 2018.

Besides the establishment of a robust client-system infrastructure, the principles of action research require a sound theoretical foundation of the applied approach (Davison, et al. 2004). Even though, the conducted interviews revealed the demand for action, a more profound approach was required for the actual evaluation of the APIs. For this reason, we derived general evaluation criteria for good API design from prevalent literature. Therefore, we conducted a comprehensive review of literature on API design and coded all identified papers. The analysis revealed a large number of different characteristics of good API design. However, four criteria stand out as being named in most of the considered studies: Simplicity, Documentation, Usability, and Tutorial/Sample Code (see Table 1). Hence, our study does not claim to consider all relevant API characteristics but the most relevant. The identified criteria are not exclusive to each other, nevertheless each aspect has unique characteristics which are explained in the *Description* column in Table 1.

| Criterion | Source | Evaluation Grades | Description |
|---|---|---|---|
| Simplicity | Bhaskar et al. (2016), (Bloch 2006), (Myers et al. 2016) | Low | It is hard to create basic objects and even to trigger basic operations. There is much unclear overhead that makes no sense from a functional perspective. |
| | | Medium | The usage is generally possible. However, there is still some overhead and tacit knowledge needed. |
| | | High | The API can be used as it is. With some basic knowledge in software development, it is easy to trigger basic functionalities, and the API behaves as "expected." |
| Documentation | (Bhaskar et al. 2016), (Burns et al. 2012), (Lee et al. 2014) | Low | There is no documentation at all, or there is only some documentation that generates no valuable insights for the developer. |
| | | Medium | Some or a big part of the functions are documented. However, use-case oriented usage is unclear. For example, a developer does not know the order of functions calls. |
| | | High | The API is thoroughly documented (each function and property) and contains necessary additional information (e.g., flow charts or sample usages). |
| Usability | (Bhaskar et al. 2016), (Bloch 2006), (Zghidi et al. 2017) | Low | The naming and usage of types are inconsistent. The API does not conform to the coding guidelines of the respective programming language or environment. |
| | | Medium | There are some inconsistency issues, but generally, the API is consistent and fulfills the usability compliance. |
| | | High | There are no inconsistencies, and the API conform to usability compliance. |
| Tutorial/ Sample Code | (Bhaskar et al. 2016), (Burns et al. 2012), (Zghidi et al. 2017) | Low | There is no tutorial / sample code |
| | | Medium | There is some sample code or tutorials for selected parts. |
| | | High | There are sample code and tutorials available for the API. That means that every complex usage of the API is explained with an example for a better understanding. |

**Table 1: API Evaluation Criteria**

## *The API Design Process*

The following section describes the applied API design process which is based on an iterative action research approach (Baskerville & Wood-Harper, 1998). It contains four steps: API Diagnosing, API Action

Planning, API Action Taking and API Evaluation. For comprehensibility reasons, we illustrated each step by the representative example of the `startGuidance()` method inside the Navigation API, whose design was refined in the API design process. The method enables an application to change the currently set destination in the vehicles navigation system to given GPS coordinates and activate the guidance.

```
import { MiddlewareModule} from 'sdk/src/system/middlewareModule';
import { RouteIdentifier} from 'sdk/src/navigation/interface/routeIdentifier';
import { DestinationsUnion} from 'sdk/src/navigation/interface/destinationsUnion';
import { NavigationApiProviderModule} from 'sdk/src/navigation/interface/navigationApiProviderModule';

private middlewareModule: MiddlewareModule;
private routeIdentifier: RouteIdentifier;
private destinationsUnion: DestinationsUnion;
private navigationApiProviderModule: NavigationApiProviderModule;

// Actual call which is triggered by tap on an ui-button

button.on('tap', startGuidanceToGeoCoords([48.178326, 11.556802]));

// Helper function to start guidance to a given list of GeoCoords

private startGuidanceToGeoCoords(geoCoords: GeoCoord[]): void {
   const destinations = geoCoords.map(
     (coord: GeoCoord) => new Destination(coord, new FormattedAddressString('', '', ''), ''),
   );
   this.startGuidance(destinations, DestinationValueType.COORDINATE);
}

// Helper function to start guidance to a given list of destinations

startGuidance (destinations: Destination[], type: String) {
  this.middlewareModule
    .getProxy(navigationApiProviderModule, Navigation.getDestinationProxy())
    .then(proxy => {
      const routeIdentifier = Navigation.NaviCommonTypes.getRouteIdentifier();
      const destinationsUnion = destinations.map(dest => StartGuidanceExample.buildDestinationValueUnion(dest, type));
      this.logger.info(`registering for destinationsHandover`);
      return proxy
        .subscribe({
          name: 'destinationsHandover',
          appId: this.app.getAppId(),
          onReceive: (event: any) => {
              this.logger.info(`destinationsHandover event received: ${event}`);
            },
            onError: (error: any) => {
              this.logger.info(`destinationsHandover error received: ${error}`);
            },
          })
          .then((subscriptionId: string) => {
            this.logger.info(`Successfully subscribed to destinationsHandover (subscriptionId ${subscriptionId})`);
            this.logger.info(`Starting guidance`);
            proxy
            .callMethod('startGuidance', {
              appId: this.onlineApp.getAppId(),
              importRouteIdentifier: new routeIdentifier({ sessionID: 0, routeHandle: 0 }),
              destinations: destinationsUnion as any[],
              handoverType: HandoverType.NON_SILENT,
            })
            .then((startGuidanceResult: any) => {
              this.logger.info(`Start guidance result: ${JSON.stringify(startGuidanceResult)}`);
            })
            .catch((error: Error) => {
              this.logger.info(`Error Ocurred while starting guidance ${error}`);
            });
        })
        .catch((error: Error) => {
          this.logger.info(`Error ocurred while subscribing to destinationsHandover broadcast ${error}`);
        });
      })
    .catch(error => {
      this.logger.info(`Start guidance got an error: ${error}`);
    });
}
```

**Image 1: Original Implementation of `startGuidance()` Method**

## API Diagnosis

During the diagnosis stage, we identified the primary problems that are anchored in the organization's major drive for introducing change. It is required that the researcher identify these problems in a complex organizational structure (Baskerville 1999). In the context of this research, this step required the analysis of the prevalent APIs. To gain deep insights about a specific API, we contacted all the relevant stakeholders. Being part of the project team granted the researchers' access to different experts inside the organization. Overall, 21 app developers were involved in the diagnosis of the APIs. For the navigation API, we enlisted experts from the platform architect team, a software engineer from the navigation module and two app developers frequently using the navigation API. Using a code that implements the navigation functionality in an app, we assessed the API based on API evaluation criteria described in the previous section, and the assessment was via open interviews. The result of the evaluation showed that the navigation API's **simplicity** was low. The experts reported a high internal complexity in the process of creating basic objects and triggering basic features like starting a guidance. In the case of the `startGuidance()` method the implementation of the interface required more than fifty lines of code which was commonly considered as inappropriate by the experts. The **documentation** of the API was rated medium to low. To start the guidance, the instantiation of multiple objects as well as methods calls were necessary. Neither a list of all the required objects and method calls nor the precise sequence of these actions was listed in the official documentation. The API's **usability** was also considered as low. The experts reported inconsistent naming and typing patterns. An inspection of the raw code by an app developer confirmed this assessment. Finally, because there were no tutorial or sample code of the API implementation, on **tutorial** or **sample code** the API was rated low.

### API Action Planning

Action planning is the collaborative step of the researcher and the practitioners to define next steps to relieve organizational pain or improve the existing situation. In the context of this project, the actual scope of the refined APIs needs to be clarified. Even though large parts of the available APIs were analyzed in the diagnosis phase, the scope of the actual refinements needed to be limited due to the finite capacity of the team. Action research implies the principle of change through action (Baskerville 1999). Following the paradigm, the team decided that the most change for app developers inside BMW could be achieved by covering the most commonly implemented interfaces. Therefore, 26 app development teams were asked for their prioritization. Based on these considerations, the team decided to work on the following APIs: Navigation, User Interface, Vehicle Data, Phone and Speech. Furthermore, a rough estimation of the required time for each APIs' redesign was made. Finally, the order in which the refined APIs should be drafted, programmed and released was defined. The release of the `startGuidance()` method inside the Navigation API was determined for mid of October 2018. We announced the resulting timeline for releases of the new APIs to the app developer community.

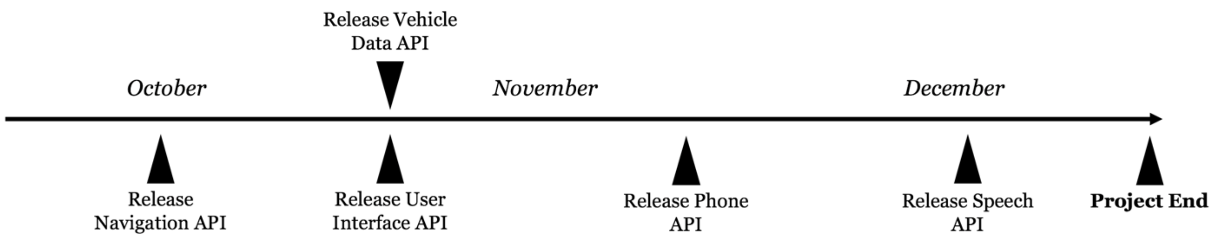



**Image 2: Announced Timeline for Releases of Refined APIs**

### API Action Taking

Based on the gathered insights in the API diagnosis, a draft for new APIs was created in the action taking phase. Besides the results of the evaluation criteria, we also took into consideration the underlying architecture. The design creation followed an iterative process in which the researchers exposed their design to the experts, collected feedback and refined the API until the design was sufficient for all involved parties. The API action taking involved the same 21 app developers as the initial API diagnosis. For instance inside the `startGuidance()` method, the destination could not exclusively be provided as GPS coordinates but also as address or as free text. One discussion was raised if the type of the destination should be contained in the method's parameters `startGuidance (destinationType: "GeoCoords", destination: [48.178325, 11.556802])` or its name `startGuidanceToGeoCoords (destination: [48.178325, 11.556802])`. The initial proposal was a design that contained this information as property of the method. However, the app developer prompted a shift of the information into the method's name. The experts argued that the text-completion feature of common code editors will propose all available options (`startGuidanceToGeoCoords`, `startGuidanceToAdress`, `startGuidanceToFreeText`) when the developer starts typing the methods name, while it will not propose all options for the method's properties at this point. The availability of this text completion feature increases the usability of the API. Hence, the API design was refined. The creation of the complete navigation API design went through three iterations until the experts made no annotations. Furthermore, the implementation of the original interface required multiple lines of so-called boilerplate code (code that does not contain any relevant functionality for app developers but is required in each implementation of the interface). By abstracting this code into the SDK, the required lines of code for an implementation of the `startGuidance()` method could be reduced from more than fifty lines of code (Image 2) down to three (Image 4). The increased simplicity of the method reduces the efforts for app developers to understand the respective interface and decreases the likelihood of mistakes in the implementation. The design of the actual API in the code affected the evaluation criteria of simplicity and usability. However, also the lack of documentation needed to be fixed. For this reason, as soon as the actual design of the code was finalized a comprehensive documentation was added inline. Further, the team created tutorials for each API which were provided in a web portal for app developers inside BMW. These instructions were based on an implementation of the refined API in a reference app which was also provided to the app developers. In this way the functional principle of the API could be comprehended by the app developers. Moreover, the functionality of the new APIs was validated from an app developer perspective inside the API team, even before it was officially released.

```
import { Guidance} from 'sdk/src/navigation/interface/guidance';

private guidance: Guidance;

button.on('tap', Guidance.startGuidanceToGeoCoords([48.178326, 11.556802]));
```

**Image 3: Refined Implementation of `startGuidance()` Method**

### API Evaluation

About three weeks before the release of the API; the team released a beta version of a new SDK, including the refined APIs. Additionally, a thread in an internal app developer forum was started, asking for feedback from the app developer community. Since the release was tagged as beta, the team was allowed to identify the further need for refinements and implement them in the API. Overall 12 app developers provided feedback via this channel. Based on the response of the app developer community, the team releases three beta versions until a stable version of the SDK was released on the announced date. For the introduction of the navigation API, three beta versions of the SDK were provided to the app developer community until its final version was released. The evaluation of the community increased the usability of the API on the level of methods and properties. App developers pointed out inconsistency in naming patterns and missing properties in specific methods.

## Results

The application of action research strives for the achievement of two goals. First, the researcher tries to introduce real change within the organization (Babüroglu and Ravn 1992). The first part of this section provides an overview of all refined APIs that were created through which change was established in the context of this project. Second, to expand the scientific body of knowledge (Baskerville 1999). The remaining part of this section describes the observed forms of lead user involvement in the API design process. These results serve as the base for the subsequent discussions on lead user involvement in the design of enabling technology.

### *Created Artefacts*

The initial goal of this research project was the analysis and refinement of poorly designed APIs that were exposed to developers of onboard apps inside the car's head-unit. A diagnosis of all prevalent API modules embodied the base for our further proceeding. Considering the relevance of APIs to app developers, we decided to focus on the refinement of the five most used APIs. First, the navigation module exposes the functionality of the onboard navigation system to the apps. Furthermore, the vehicle data API provides information on the current car status as the current vehicle speed or fuel status. The UI and speech APIs provide interfaces that allow the customer to interact with the app via a graphical user interface or voice respectively. Finally, the Phone API provides access to the functionality of a smartphone that is connected to the car's head-unit via Bluetooth. In all, the refined APIs contain 59 methods and 31 properties that were implemented by the app developers. The evaluation of the original (Orig.) as well as the refined (Ref.) API design reveals a clear improvement regarding all defined evaluation criteria through the API design process (Table 2).

| API | #Involved App Developers | #Refined Methods | #Refined Properties | Simplicity (Orig./Ref.) | Documentation (Orig./Ref.) | Usability (Orig./Ref.) | Sample Code (Orig./Ref.) |
|---|---|---|---|---|---|---|---|
| Navigation | 4 | 21 | 6 | Low/High | Medium/High | Low/High | Low/High |
| Vehicle Data | 5 | 2 | 21 | Medium/High | Medium/High | High/High | Low/High |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| UI | 5 | 19 | 2 | Low/High | Medium/High | Low/High | Low/High |
| Phone | 4 | 3 | 1 | Medium/High | Medium/High | Low/High | Low/High |
| Speech | 3 | 14 | 1 | Low/High | Medium/High | High/High | Low/High |

**Table 2: Overview on Refined APIs**

Action research strives for the introduction of change. However, the pure creation of a refined API design does not prove any change within the organization. Only if the created artifact is applied in practice, can it produce an effect (Babüroglu and Ravn 1992). The implementation of the APIs in the BMW onboard platform SDK should satisfy this requirement. The APIs were released in the platform SDK according to the timeline (Image 3). All currently developed apps implemented the refined APIs until February 2019. These apps will be to available to BMW customers shortly.

## *Theoretical Learnings*

The deep insights we gained through the application of action research allowed us to identify the challenges and benefits of lead user involvement in each step of our applied API design process. Thereby, we classify our findings into psychological, managerial, methodological, cultural and political perspectives (Bano and Zowghi 2014).

By analyzing the implemented APIs per app, we identified the heaviest users of an API within the app developer community. However, even though all contacted app developers replied positively on our request, just a fraction of them was able to participate in our project. Especially employers of external suppliers were impeded by limited time capacity, since their assignment didn't include such additional activities (*managerial challenge*). Further, the app project leaders were not motivated to shift capacity from their project towards the API design project. High pressure on the app development teams forced them to focus all their efforts on the app development itself (*Psychological challenge*). Finally, the evaluation of the original APIs itself, revealed a high complexity within the system (*managerial challenge*). However, the iterative approach as well as the involvement of the API provider module engineers enabled a comprehensive understanding of all involved parties. The conflation of the API consumer and API provider in common meetings simplified the communication (*managerial benefit*), facilitated knowledge sharing (*cultural benefit*) and created mutual comprehension of the challenges and difficulties of the respective other side (*psychological benefit*). This again motivated all participants to improve the status quo.

The API planning step comprised the prioritization of the refined APIs. Even though the most implemented APIs were refined first, this approach endangers the loss of engaged lead users, since their more specific API wasn't part of the project any longer (*managerial challenge*). Further, the announcement of the API refinements raised expectations from management as well as the app developer community, which needed to be handled by the team (*psychological challenge*). On the other hand, the involvement of the lead users simplified and enhanced the prioritization process (*methodological benefit*). Thus, there were APIs that were implemented by a large number of apps. However, their status quo was more sufficient than the design of other APIs that were implemented by a slightly smaller number of apps. In this way, the involvement of lead users enabled a better understanding of user requirements. Further, the involvement of app developers enabled a better understanding of the required refinements measures and the support the establishment of realistic expectations towards the project within and outside the team (*managerial benefit*).

Next, the API action taking step considered the actual creation of refined API designs. The involvement of multiple lead users enabled the definition of a generic API design that satisfies not only specific but a broad spectrum of use-cases. However, the number of iterations needed for a commitment of all involved parties regarding the created design couldn't be estimated in forward. Hence, the limited amount of time available embody a challenge for the team (*managerial challenge*). Further, it turned out that different app developers had conflicting requirements on specific API methods. Even though, these conflicts could be immediately resolved by the API design team, they embody a challenge (*political challenge*). On the other hand, these conflict resolutions increased the commitment on the refined solution and increased the motivation of all involved parties. All involved parties agreed that these kinds of discussions increase the quality of the final API design (*methodological benefit*). Further, the commitment of all parties increased the likelihood that the final solution would be accepted by a majority of the app developer community (*psychological benefit*).

In the final API evaluation step, the beta versions of the APIs were published to the community. The involvement of further users of the API should increase the maturity of the final API release. However, this required a material amount of app developers to implement and evaluate the APIs. This increased the amount of required time for the overall process (*managerial challenge*). Though, the involvement of even more app developers furtherly increased the likelihood for the identification of insufficiencies or flaws in the final design (*methodological benefit*). Further, the app developer community felt involved and fetched up for the upcoming changes. In this way the overall acceptance for the project could be increased (*psychological benefit*).

| Process Step | Challenges | Benefits |
|---|---|---|
| **API Diagnosis** | • Time constraints (managerial)<br>• Lack of motivation (psychological)<br>• System complexity (managerial) | • Simplified communication (managerial)<br>• Facilitated knowledge sharing (cultural)<br>• Increased motivation (psychological) |
| **API Action Planning** | • Time constraints (managerial)<br>• Users and Managers expectations (psychological) | • Better understanding of requirements (methodological)<br>• Development of realistic expectations (managerial) |
| **API Action Taking** | • Time constraints (managerial)<br>• Conflicts (political) | • Increased quality of final design (methodological)<br>• User acceptance (psychological) |
| **API Evaluation** | • Time constraints (managerial) | • Increased quality of final design (methodological)<br>• User acceptance (psychological) |

**Table 3: Challenges and Benefits of Lead User Involvement in the API Design Process**

Our analysis reveal that especially managerial challenges affected the API design process. This approves Svahn et al. (2017) observations from another automotive case at Volvo: The management needs to conceive the need for a shift towards developer-centric software design. Otherwise, real change is hard to achieve in an incumbent context. However, while the Volvo case remains in the observation of these managerial phenomena, our study proposes user involvement as potential approach to address this challenge bottom-up. The interplay of multiple, heterogenous developers is able to achieve real change. Our results prove that this is not just true for matured app platforms (Eaton et al. 2015) but also in the context of a just emerging digital platform in an incumbent context.

## Summary and Outlook

In this study we achieved a valid involvement of lead users in the evolvement of an incumbent service system embodied by automotive onboard APIs. The participatory approach for the API design enabled the creation of APIs that does not require domain-specific knowledge to implement (Pühler 2011; Schlachtbauer et al. 2012). The identified benefits prove the relevance of user involvement for refining existing technology in an incumbent context. However, BMW as organization is optimized towards the development of traditional products. In this setting, the customer is usually considered as the only user of the developed product. The usability of a feature that is not visible for him is not considered as dispensable. However, APIs are not directly used by the driver of the car, but the developer who is building complementary apps. The identified challenges in the shift towards app-developer centric APIs prove this fact. The elimination of these requires reconfiguration within the organization and its mindset (Svahn et al. 2015). Furthermore, we approve action research as valid method for gathering deep insights in a real organization, its technologies and its processes. The approach enabled a comprehensive understanding of the app developer's initial needs and their positive evaluation of the initiated change. However, these findings reveal a short-range character. The long-term effects on the actual app development activities remain in the dark. Therefore, an investigation on the effects of the new APIs on the app development appears as promising extension of this study. Does the shift to user-centric API design actually foster the development of apps? This question could be addressed by a quantitative analysis of the utilization of the created APIs as well as by further qualitive research.

## REFERENCES


AutomotiveWorld. 2019. "20 Years at the Cutting Edge: Bmw Connecteddrive." Retrieved 15.02.2019, from https://www.automotiveworld.com/news-releases/20-years-cutting-edge-bmw-connecteddrive/

Babüroglu, O., and Ravn, I. 1992. "Normative Action Research," *Organization Studies* (13:1), pp. 19-34.

Bano, M., and Zowghi, D. 2014. "A Systematic Review on the Relationship between User Involvement and System Success," *Information and Software Technology* (58), pp. 148-169.

Baskerville, R. L. 1999. "Investigating Information Systems with Action Reserach," *Communications of the AIS* (2), pp. 3–16.

Bhaskar, R. K., Anslow, C., Brosz, J., and Maurer, F. 2016. "Developing Usable Apis with Xp and Cognitive Dimensions.," in *2016 Ieee Symposium on Visual Languages and Human-Centric Computing (Vl/Hcc)*. IEEE, pp. 101–105.

Bloch, J. 2006. "How to Design a Good Api and Why It Matters.," in *Companion to the 21st Acm Sigplan Symposium on Object-Oriented Programming Systems, Languages, and Applications*. ACM, pp. 506-507.

BMW. 2018. "New Bmw Cockpit Is Digital, Intelligent, Perfectly Tuned to the Driver and Always up-to-Date.", from https://www.press.bmwgroup.com/global/article/detail/T0285445EN/new-bmw-cockpit-is-digital-intelligent-perfectly-tuned-to-the-driver-and-always-up-to-date?language=en

Burns, C., Ferreira, J., Hellmann, T. D., and Maurer, F. 2012. "Usable Results from the Field of Api Usability: A Systematic Mapping and Further Analysis.," in *Ieee Symposium on Visual Languages and Human-Centric Computing (Vl/Hcc)*. IEEE.

Digitaltrends. 2017. "Audi Shows Integrated Android Infotainment System at Google I/O." Retrieved 15.02.2019, from https://www.digitaltrends.com/cars/audi-android-integration-google-io/

Eaton, B., Elaluf-Calderwood, S., Sørensen, C., and Yoo, Y. 2015. "Distributed Tuning of Boundary Resources: The Case of Apple's Ios Service System " *MIS Quartely* (39:1), pp. 217-244.

Frank, U., Klein, S., Krcmar, H., and Teubner, A. 1998. "Aktionsforschung in Der Wi - Einsatzpotentiale Und Einsatzprobleme.",," *Wirtschaftsinformatik und Wissenschaftstheorie. Grundpositionen und Theoriekerne. Arbeitsberichte des Instituts für Produktion und Industrielles Informationsmanagement* (4), pp. 71-90.

Henderson, R., and Clark, K. 1990. "Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms," *Administrative science quarterly* (35:1), p. 9.

Henfridsson, O., and Ghazawneh, A. 2013. "Balancing Platform Control and External Contribution in Third-Party Development: The Boundary Resources Model," *Information Systems Journal* (23:2), pp. 173-192.

Henning, M. 2009. "Api Design Matters," *Communications of ACM* (52:5), pp. 46-56.

Herstatt, C., and von Hippel, E. 1992. "From Experience: Developing New Product Concepts Via the Lead User Method: A Case Study in a "Low-Tech" Field," *Journal of Product Innovation Management* (9:3), pp. 213-221.

Iyer, B., and Subramaniam, M. 2015. "Are You Using Apis to Gain Competitive Advantage?," *Harvard Business Review*), pp. 2–5.

Jaworski, B., and Ajay, K. 1993. "Market Orientation: Antecedents and Consequences.," *Journal of Marketing* (57:3), pp. 53-70.

Keng, S., and Rossi, M. 2011. "Evaluation Techniques for Systems Analysis and Design Modelling Methods - a Review and Comparative Analysis," *Information Systems Journal* (21:3), pp. 249-268.

Lee, S., Lee, S., Lim, S., Jung, J., and Choi, S. 2014. "An Api Design Process in Terms of Usability: A Case Study on Building More Usable Apis for Smart Tv Platform," *IEEE 38th International Computer Software and Applications Conference Workshops*, pp. 567–571.

Li, T., and Calantone, R. 1998. "The Impact of Market Knowledge Competence on New Product Advantage: Conceptualization and Empirical Examination.," *Journal of Marketing* (62:4), pp. 13-29.

Lüthje, C., and Herstatt, C. 2004. "The Lead User Method: An Outline of Empirical Findings and Issues for Future Research," *R&D Management* (24:5), pp. 553-568.

Myers, B., Ko, A., LaToza, T., and Yoon, Y. 2016. "Programmers Are Users Too: Human-Centered Methods for Improving Programming Tools.," *Computer* (49:7), pp. 44-52.

Pühler, M. 2011. *Interaktive Gestaltung Von Automotive Services Durch Softwaregestuetzten Einsatz Domaenenspezifischer Modellierung*. Diss., Technische Universität München, Munich.

Schlachtbauer, T., Stanzl, G., Pühler, M., Scheermann, M., and Krcmar, H. 2012. "Smart Automotive Apps Fuer Kundenorientierte Dienstleistungen Im Fahrzeug," in *Smart Mobile Apps,* L.-P.C.e. Verclas S (ed.). Berlin, Heidelberg: Springer, pp. 27-41.

Schreieck, M., and Wiesche, M. 2017. "How Established Companies Leverage It Platforms for Value Co-Creation – Insights from Banking," *Twenty-Fifth European Conference on Information Systems*, Guimarães, Portugal.

Schreieck, M., Wiesche, M., and Krcmar, H. 2018. "Multi-Layer Governance in Platform Ecosystems of Established Companies," *Academy of Management Proceedings*, Chicago, Illinois.

Svahn, F., Lindgren, R., and Mathiassen, L. 2015. "Applying Options Thinking to Shape Generativity in Digital Innovation: An Action Research into Connected Cars," *International Conference on System Sciences*, Hawaii.

Svahn, F., Mathiassen, L., and Lindgren, R. 2017. "Embracing Digital Innovation in Incumbent Firms: How Volvo Cars Managed Competing Concerns," *MIS Quartely* (41:1), pp. 239-255.

Tiwana, A. 2014. *Platform Ecosystems: Aligning Architecture, Governance, and Strategy*. 225 Wyman Street, Waltham, MA 02451, USA: 2014 Elsevier Inc.

Tripsas, M., and Gavetti, G. 2000. "Capabilities, Cognition, and Inertia: Evidence from Digital Imaging.," *Strategic management journal* (21:10/11), pp. 1147-1161.

Um, S. Y., Yoo, Y., Wattal, S., Kulathinal, R. J., and Zhang, B. 2013. "The Architecture of Generativity in a Digital Ecosystem: A Network Biology Perspective," *Thirty Fourth International Conference on Information Systems (ICIS)*, Milan, Italy.

von Hippel, E. 1986. "Lead Users: A Source of Novel Product Concepts," *Management Science* (32:7), pp. 791-805.

von Hippel, E. 2005. *Democratizing Innovation: The Evolving Phenomenon of User Innovation*. The MIT Press, Cambridge, MA.

von Hippel, E., Thomke, S., and Sonnack, M. 1999. "Creating Breakthroughs at 3m.," *Harvard business review* (77), pp. 47-57.

Zghidi, A., Hammouda, I., Hnich, B., and Knauss, E. 2017. "On the Role of Fitness Dimensions in Api Design Assessment - an Empirical Investigation," *IEEE/ACM 1st International Workshop on API Usage and Evolution (WAPI)*.

# Cars as Digital Infrastructure: An Analysis of Platform Ecosystem Options in the Automotive Industry

Niklas Weiß, Maximilian Schreieck, Manuel Wiesche, Helmut Krcmar
Department of Informatics
Technical University of Munich
Munich, Germany
{ niklas.weiss, maximilian.schreieck, wiesche, krcmar}@tum.de

## Introduction

Incumbent industries need to adopt platform business models to avoid disruption by foreign players (Parker et al. 2016). The automotive domain is a prominent example for this phenomenon (Svahn et al. 2017). Car manufacturers implement digital technologies in their products to enable the implementation of digital services; examples cover BMW ConnectedDrive, MercedesMe, or AudiConnect.

The widespread distribution of digital technologies into millions of cars around the globe has led to the emergence of a new digital infrastructure with large potential for innovation. However, while a digital infrastructure comprises the pure computing and networking resources and is not owned by anybody (Constantinides et al. 2018) it requires digital platforms and inherent ecosystems on top to facilitate innovation as well as value creation (Gawer 2014). As opposed to established digital infrastructures as smartphones or computers, the emergence of an app ecosystems for cars is just in the beginning. Car manufacturers need to develop strategies that enable them to create and capture value through digital platforms in their cars to avoid being disrupted by new entrants (Parker et al. 2016).

This study strives for answering the question which strategical options on platform ecosystems occur for incumbent car manufacturers that are confronted the transformation of their products to a digital infrastructure. Therefore, we conduct a case study at a globally operating premium car manufacturer. The collection of data from 42 expert interviews within the period of two years, allowed us to identify three platform ecosystem options: (1) build a proprietary platform ecosystem, (2) build a collaborative platform ecosystem, or (3) join an existing platform ecosystem. Our findings contribute to literature on digital infrastructures as well as theory on platform launch and competitive strategies in platform ecosystem.

## Background

Traditionally, a car is a purely physical product. In cohesion with streets, traffic regulations and fuel supply, it plays a central role in an infrastructure that provides transportation for persons and goods. However, the integration of computing and networking capabilities generates a new digital infrastructure (Constantinides et al. 2018). While in traditional infrastructures technology and provided services are tightly coupled – such as cars and transportation - digital infrastructures enable generativity through flexibility, scalability, its recursive nature and the varying substance of data as transported material (Tilson et al. 2010). This means that cars were originally designed for fulfilling one single task, namely the transportation of persons or goods. Subsequently, all services that build on cars as infrastructure relied on the principle that someone or something is transported from A to B, like taxi offers, logistics or food delivery. Even motorsport racing is based on the competition of drivers that try to cover a certain distance from start to finish faster than competitors. However, the distribution of digital technology into cars enables the design of new services that do not implicitly require a relation to transportation.

Beside mobility-related services as parking[1] or car sharing[2], car manufacturers started to provide digital services as music streaming[3] or productivity apps[4] to enhance their products.
Even though, these digital services are built on computing and networking capabilities of the car, their development is traditionally conducted in linear value chains. A supplier gets paid for the development of software, which gets integrated by the car manufacturer and eventually sold as feature to the customer. However, a platform ecosystem embodies a network that outperforms such traditional pipelines (Parker et al. 2016). It enables the value creation through third parties without any direct involvement of the platform owner and enables massive scaling effects (Parker et al. 2017). Furthermore, platform ecosystems reveal beneficial network effects that increase with the size of demand side. The more potential customers can be reached with a single app, the more attractive is an ecosystem for app developers and the more attractive it is again for potential customers. In contrast, multihoming of apps in several ecosystems requires larger efforts for app developers (Cennamo et al. 2018). Hence, an ecosystem that enables the development and distribution of one app to cars of multiple brands unleashes massively larger network effects then the manufacturers' currently proprietary systems.
The emergence of a new digital infrastructure without any relevant platform ecosystems attracts experienced players of the platform game to enter the field. Google just announced the release of Android Automotive OS (Google 2019), which should replace the manufacturer's prevalent proprietary infotainment systems. The convergence to a layered architecture (Yoo et al. 2010) and the coherent decoupling of hardware and software in modern cars allowed the development of an automotive app platform by Google without producing any vehicle. The envelopment of existing systems (Eisenmann et al. 2011) and the consequent large and cross-brand customer base in combination with the unexploited potential for innovation (Henfridsson et al. 2018) of apps in cars enables the nurturing of a vibrant app ecosystem. Now, car manufacturers are confronted with the question if they should join Google's ecosystem or strike out on their own. This leads to a more general question: Which platform ecosystems options emerge for car manufacturers as incumbent firms that are confronted with the transformation of their products to an unexploited digital infrastructure?
To answer this question, we collected 42 expert interviews in the environment of a globally operating car manufacturer who is confronted with the transformation of its products to a digital infrastructure. Our analysis revealed three major options which are illustrated in the remainder of this paper.

## Build proprietary platform ecosystem

The traditionally strong competition in the automotive industry motivated manufacturers in the past to build proprietary infotainment systems that differentiated their vehicles from competitional products. Each system reveals specific graphical styles and features that should raise the attention of customers and motive the purchase of a car.

*„We live in a world which is massively fragmented for third-party developers. Mercedes hast its own platform, BMW has its own platform and any other manufacturer has its own platform.“*

To benefit from scaling effects, manufacturers started to design their systems according to a layered platform architecture that allows the modular extension of their software in the car. With Ford[5] and GM[6], two established manufacturers already decided to open their platforms to

[1] https://parknowgroup.com from June 3, 2019
[2] https://www.drive-now.com from June 3, 2019
[3] https://www.bmw-connecteddrive.de/app/index.html#/portal/store/Base_Connected_Music from June 3, 2019
[4] https://www.bmw-connecteddrive.de/app/index.html#/portal/store/Base_ExchangeOffer from June 3, 2019
[5] https://developer.ford.com from June 5, 2019
[6] https://developer.gm.com from June 5, 2019

third-party developers. Our findings also indicate that the opening of an existing system embodies a valid option to endeavor the creation of an automotive platform ecosystem. However, the experts agreed on the massive efforts that are required for launching an own platform ecosystem, which need to be justified by respective benefits.

### Build collaborative platform ecosystem

Even though, Ford and GM already decided to open their platforms to third-party developers, the small number of provided apps in their respective app stores indicate that the companies failed in nurturing a vibrant ecosystem. According to our findings, the small number of target devices could be a major reason for this failure. In the mobile world, an app developer needs to build and maintain two versions of an app, to address hundreds of millions of Android and iOS users. In contrast, one single car manufacturer ships a few million vehicles a year – a negligible number of potential customers for the profitable development of a distinctive app.

*"You simply need to increase the user base. Otherwise larger app providers as Spotify, Netflix or whoever are simply not interested in you. The efforts for them are too high in comparison to their potential benefits."*

The tough competition in the automotive industry indicates that one manufacturer will not be able to leverage its sales to achieve a similar magnitude of available target devices as the mentioned mobile app ecosystems. Hence, manufacturers could cooperate and provide a cross-brand app platform that enables the deployment one single app to cars of multiple brands.

### Join existing platform ecosystem

Beside the building and nurturing an own ecosystem, multiple experts mentioned the option of joining an existing platform ecosystem. Manufacturers would not be required to attract own app developers, since all apps in the ecosystem would be available in their cars as soon as the platform is integrated.

*„If you take an existing ecosystem as Android, you just need to integrate the existing platform into the car and all Android [Auto OS] apps will be available. You don't need to attract any third parties by your own."*

Furthermore, the implementation of an established platform entails the utilization of a matured and proven technology, which should decrease testing efforts in comparison to the creation of an own platform. Moreover, due to its open source approach[7] Android as concrete example provides the option to contribute to the platform. In this way, manufacturers are able to extend the platform's feature set, while Google as platform owner is responsible for long-term maintenance.

## Contribution and next steps

The results of our work contribute to literature on platform ecosystem in three aspects. First, we illustrate how a traditionally physical product transforms into a digital infrastructure that enables innovation (Henfridsson et al. 2018). While a car was formerly assessed by its driving performance and its design, customer expectations are shifting towards the vehicle's connectivity and the availability of digital services. Driven by this change of expectation, manufacturers implement digital technology into their car and push the emergence of a new digital infrastructure. Second, we consider the process of platform launch from a new

[7] https://source.android.com from June 6, 2019

perspective by shifting the context from the frequently investigated green field (Stummer et al. 2018) to an environment with established companies that perceive disruptive invasion of digital native players into their domain. This “brown field” perspective reveals new aspects, as the consideration of the incumbent’s legacy in culture, skills or technology. Third, our findings embody a theoretical contribution on competitive strategies in platform ecosystem. While prevalent literature focuses on battles between existing ecosystems (Cennamo and Santalo 2013) or approaches for the launch of new ecosystem against established players (Karhu et al. 2018), we consider the dispersal of platform ecosystems on a new digital infrastructure without any established ecosystem.

Our studies focus on a development in the automotive industry, though we consider our findings as generalizable beyond the specific industry. For instance, the industrial internet of things in which formerly analogue assets like production machines transform to a digital infrastructure reveal multiple similarities to the automotive domain. For this reason, we consider a continuation of our work as worthwhile. Our next steps will focus on a detailed understanding of the three identified platform ecosystem options. We aim to identify actual pros and cons of the options and contribute to the development of specific recommendations for action. Thereby it needs to be considered that different manufacturers reveal different contexts as number of customers, budget and size of the organization, which may influence the decision for one of the identified options.

## References


Cennamo, C., Ozalp, H., and Kretschmer, T. 2018. "Platform Architecture and Quality Trade-Offs of Multihoming Complements," *Information Systems Research* (29:2), pp. 461–478.

Cennamo, C., and Santalo, J. 2013. "Platform Competition: Strategic Trade-Offs in Platform Markets," *Strategic Management Journal* (34:11), pp. 1331–1350.

Constantinides, P., Henfridsson, O., and Parker, G. 2018. "Introduction—Platforms and Infrastructures in the Digital Age," *Information Systems Research* (29:2), pp. 381-400.

Eisenmann, T., Parker, G., and Van Alstyne, M. 2011. "Platform Envelopment," in *Platforms, Markets and Innovation,* A. Gawer (ed.). Cheltenham, UK: Edward Elgar, pp. 131-162.

Gawer, A. 2014. "Bridging Differing Perspectives on Technological Platforms: Toward an Integrative Framework," *Research Policy* (43:7), pp. 1239-1249.

Google. 2019. "Developing Apps for Andoird Automotive Os." from https://android-developers.googleblog.com/2019/05/developing-apps-for-android-automotive.html

Henfridsson, O., Nandhakumar, J., and Scarbrough, H. 2018. "Recombination in the Open-Ended Value Landsacpe of Digital Innovation," *Information and Organization* (28:2), pp. 89-100.

Karhu, K., Gustafsson, R., and Lyytinen, K. 2018. "Exploiting and Defending Open Digital Platforms with Boundary Resources: Android’s Five Platform Forks," *Information Systems Research* (29:2), pp. 479–497.

Parker, G., Van Alstyne, M., and Choudary, S. P. 2016. *Platform Revolution: How Networked Markets Are Transforming the Economy and How to Make Them Work for You*. New York: WW Norton & Co.

Parker, G., Van Alstyne, M., and Jiang, X. 2017. "Platform Ecosystems: How Developers Invert the Firm," *MIS Quartely* (41:1), pp. 255-266.

Stummer, C., Kundisch, D., and Decker, R. 2018. "Platform Launch Strategies," *Business & Information Systems Engineering* (60:2), pp. 167-173.

Svahn, F., Mathiassen, L., and Lindgren, R. 2017. "Embracing Digital Innovation in Incumbent Firms: How Volvo Cars Managed Competing Concerns," *MIS Quartely* (41:1), pp. 239-255.

Tilson, D., Lyytinen, K., and Sørensen, C. 2010. "Research Commentary—Digital Infrastructures: The Missing Is Research Agenda," *Information Systems Research* (21:4), pp. 748-759.

Yoo, Y., Henfridsson, O., and Lyytinen, K. 2010. "The New Organizing Logic of Digital Innovation: An Agenda for Information Systems Research," *Information Systems Research* (21:4), pp. 724-735.

Check for updates

JITTC

Teaching Case

Journal of Information Technology Teaching Cases
2021, Vol. 11(2) 90–100
© Association for Information Technology Trust 2020

Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/2043886920944185
Journals.sagepub.com/jittc
SAGE

# From Product to Platform: How can BMW compete with Platform Giants?

**Niklas Weiss[1], Maximilian Schreieck[1], Manuel Wiesche[2] and Helmut Krcmar[1]**

**Abstract**

Today, it is natural for digital services to be available anytime, anywhere. Thanks to digital platforms, we can connect with our intelligent assistants, stream movies on our networked televisions, and wirelessly control our homes with smartphone apps. Within the last few years, our cars have also been transformed into computers on wheels. However, the apps with which we are accustomed are often not available in cars. Why are cars different? This teaching case examines how the worldwide automotive manufacturer BMW developed its onboard infotainment system into a digital platform. While the transformation of a manufacturing firm into a technology company reveals multiple challenges, new actors enter the stage. In 2017, Google announced the launch of Android Automotive OS, a complete operating system for cars with access to a vibrant platform ecosystem. Should BMW accept Google's offer to use Android Automotive OS as a digital platform for their cars, or should they strive to create their own proprietary platform ecosystem? This teaching case introduces the dynamics of digital platform ecosystems and illustrates the "platform conundrum" that many traditional companies must confront: Is it better to build a new proprietary platform ecosystem or join an existing dominant platform ecosystem? We provide rich insights from BMW's development, sales, and strategic divisions, helping students to understand the risks, chances, and challenges of various choices that occur in the context of digital platform ecosystems and why such decisions might be crucial to the future of traditional companies such as BMW.

**Keywords**

Digital platforms, digital platform ecosystems, digital platform options, automotive software engineering, automotive apps, automotive app store

## Introduction

Have you ever been annoyed because you were unable to use your favorite apps while you were driving? Did you ever wonder why you are forced to use the old-school integrated navigation system instead of Google Maps with up-to-date maps and real-time traffic? Or why is there no better way to listen to your favorite Spotify playlist than streaming it via your smartphone, which requires a prior connection to the car? Or why is there no streamlined solution to read and write WhatsApp messages while driving? As smartphone users, we are accustomed to choosing from millions of available apps to enhance our devices at any time. These apps are usually free and frequently updated, and if an alternative with more features and a better look-and-feel pops up, we can simply download the new app and remove the old one. So, why is that possible for smartphones but not for cars?

For almost two decades, the worldwide automotive manufacturer BMW has been trying to integrate its customers' digital world into its vehicles in an attempt to react to changing customer needs. In the past, car buyers emphasized

[1]Technical University of Munich, Germany
[2]Technical University of Dortmund, Germany

**Corresponding author:**
Niklas Weiss, Technical University of Munich, Boltzmannstr. 3, 85748 Garching, Germany.
Email: niklas.weiss@tum.de

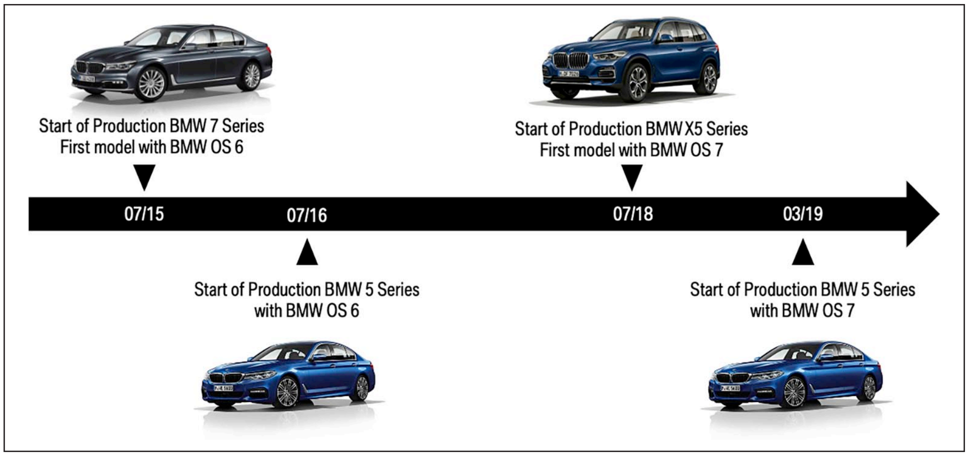



**Image 1.** Timeline BMW OS 7 in 5 series and X5 series.

vehicle design, running characteristics, and safety. These qualities are still appreciated by customers today, although they are complemented by the need for connectivity and availability of apps. According to a 2018 McKinsey study, 40% of car owners would change their favorite car brand exclusively based on the availability of enhanced digital services.[1] BMW tried to fulfill these new customer requirements as did the other major car manufacturers. In 2003, the first online services were introduced into a BMW car design. After several advancements, the newest generation of the BMW infotainment system was released in the summer of 2018. The newly designed BMW onboard system is constantly connected to the Internet and provides apps such as a parking lot finder,[2] music streaming,[3] integrated e-mail and calendar features,[4] and apps for different calling services.[5] However, the number of available apps is quite low compared to the vibrant smartphone platform ecosystems with which most customers are accustomed.

In this teaching case, we first introduce BMW ConnectedDrive as the brand representing BMW's infotainment and connectivity features. By describing various milestones in the evolution of ConnectedDrive, the second section illustrates how BMW directly integrated digital services into its cars. The third section examines the emergence of Android Automotive OS, an operating system developed specifically for cars by Google that is independent of any smartphone. The appearance of this new technology presents BMW with three strategic options for the future of its onboard automotive infotainment system, which are described in the final section.

## BMW ConnectedDrive

BMW encapsulates all infotainment and connectivity features in its vehicles under the BMW ConnectedDrive brand. The onboard infotainment system is called BMW OS and can be considered as the heart of the Connected Drive product offering.

### *Head-units and infotainment systems*

The infotainment system is one of many software components that run on miscellaneous electric control units (ECUs) inside a car. All ECUs are connected by means of bus systems that transfer data between the various components. The infotainment system operates on an ECU called the head-unit, which receives data from multiple sensors throughout the entire car that are processed infotainment system applications. For example, the navigation system is one of the applications that receives and processes global positioning system (GPS) data from the vehicle's antenna control unit to display the current position of the car on a map. BMW maintains triennial development cycles for its head-units. When a new head-unit generation development is complete, it is integrated into every car model that will be subsequently produced. Older models, previously produced with older generations of the head-unit, are eligible to receive upgrades (Image 1).

A new generation of a head-unit is always expected to provide new features for the lowest possible additional cost. Reducing production costs is a major driver of developmental efforts. There is a fundamental difference between software and hardware. While a head-unit must be purchased for every vehicle produced, one version of the corresponding software can be installed into millions of cars without any sizable additional cost. Therefore, the fewer the hardware resources required by the software, the cheaper the per car cost of the hardware. For example, making a decision regarding the memory module capacity of a new generation head-unit illustrates this point. One option would be to use a 16-GB module that costs €30 per unit. A second option would be to use a 12-GB module costing €25 per unit. The market forecast predicts a production volume of 2.5 million cars. The €5 cost difference multiplied by 2.5 cars translates into a potential savings of €12.5 million, funds that can be allocated to implementing software optimization that will enable the cheaper hardware option.

Software performance increases with the elimination of any abstraction. The more adapted the code is to the inherent hardware, the better its eventual performance. As a result, the developed software is inextricably intertwined with its corresponding hardware. Therefore, the release of a new infotainment system is associated with the triennial head-unit development cycle. However, software development does not strictly end with the release of the integrated infotainment system. Refinements such as performance improvements and the incorporation of new features are implemented in subsequent product upgrades. When such an upgrade is completed, the new software is installed into all newly produced cars and distributed to existing customers during routine maintenance checks at a BMW servicing partner.

## *Introduction of the ConnectedDrive Store*

Customers ordering a new BMW online or through a dealer can choose from several infotainment options. While the top tier includes all available software features and high-end hardware, the low-budget variants provide less functionality as well as restricted hardware with lower computing power and a smaller sized cockpit screen display. Previously, the chosen configuration was considered to be a permanent feature for the lifetime of the car. This setup fundamentally changed in 2012 with the introduction of the BMW ConnectedDrive Store, which enabled customers to extend their car's capabilities by installing new digital services, such as adding real-time traffic information or map updates to the navigation system, months or even years after a car's initial purchase by buying such services through the BMW ConnectedDrive Store.

The ConnectedDrive Store gives BMW several advantages. First, a BMW owner can purchase BMW's digital services at any point in time, which creates the opportunity for generating additional revenue from existing customers. Second, most digital services require online connectivity. In contrast to mobile phones, whose network operation costs are borne by the user, BMW is the carrier for the car's subscriber identity module (SIM) card. Even though utilizing BMW's connectivity features is not billed according to usage, the company can apply value-added subscription models that require additional payments for higher tier features or to extend subscription services at the end of a predetermined period (see Image 2). Third, BMW was one of the first automotive manufacturers to directly sell digital services in their cars, which provided an opportunity to promote BMW as an innovative and progressive car brand.

That the introduction of the ConnectedDrive Store enabled BMW to distribute digital services independent of the initial purchase represented the potential for implementing innovative business models. However, BMW's sales strategy was not adapted to this change. The fixed development

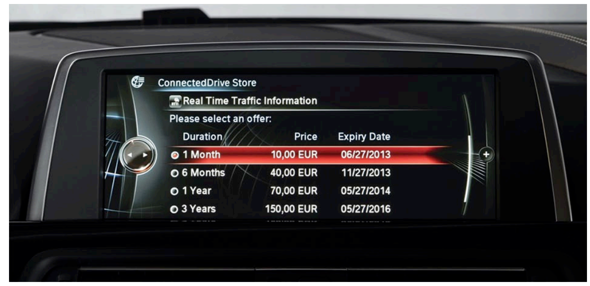



**Image 2.** BMW ConnectedDrive Store in BMW OS 4.

phase of vehicle equipment features such as brakes or seats usually ends upon the commencement of a car's production cycle—unlike the continuous software development cycle, which persists past a product's sale. Therefore, the idea of rolling out maintenance and feature upgrades in a previously sold car was counterintuitive to most automotive sales managers. Releases of new digital services (or new versions of existing services) were coupled with releases of new vehicles, meaning that digital services were sold as features of a new car and not as standalone products. Since a free testing period was not applicable to features such as brakes or seats, this option was not made available for new digital services:

> Apps belong to a certain optional equipment line because our organization is just able to work with optional equipment structures. Even if we want to go other ways, we haven't the abilities currently. (Sales Operator for Digital Products)

## *The modular distribution of apps*

The in-store purchase of a feature did not trigger the deployment of new code to the head-unit. Instead, a functionality was preinstalled in all cars and activated simultaneously. Dynamic content, such as weather or news, was displayed by a preinstalled browser. However, this changed with the sixth generation of the infotainment system, which was launched in the BMW 7 series in the summer of 2015. This new system contained a dedicated platform component that was able to remotely download, install, and run app packages. While the capability of the preinstalled browser of the previous generation was limited to displaying online content, natively running apps were now able to access interfaces deeply embedded in the car. Hence, it was possible to provide apps that processed information on the current velocity of the car, the destination set in the navigation system, or signals of the parking control system. The parking app ParkNow,[6] for instance, uses such automotive data to recognize whether a driver is searching for a parking lot in the vicinity of the programmed destination and provides relevant information on nearby parking facilities. Even though the new platform's capabilities enabled many new

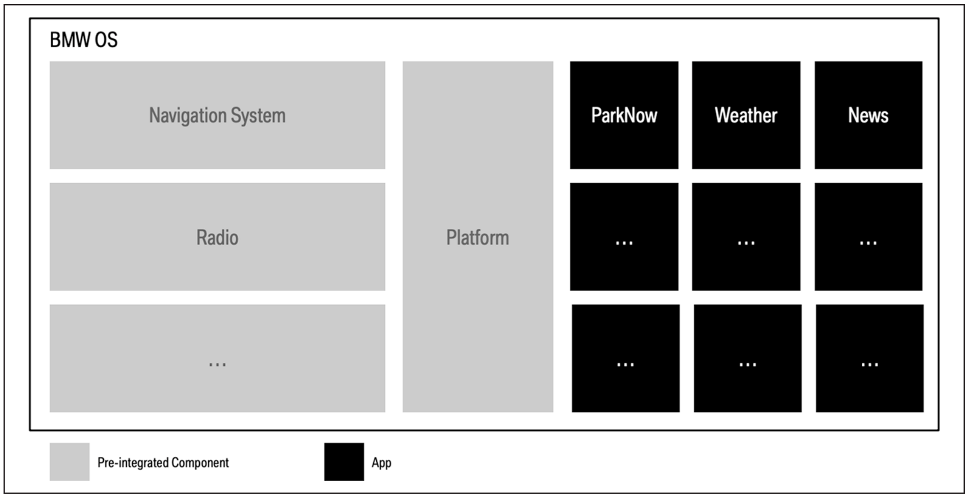



**Image 3.** Infotainment system architecture.

use cases for additional apps, the most prominent features of the infotainment system, such as the navigation system and the radio, continued to be deployed as preinstalled software components alongside the platform (see Image 3).

These enhanced capabilities of the platform engendered increasingly more departments within BMW, to start own app development efforts. In 2015, software engineers from the research division started the BMW Labs[7] initiative. Customers were able to apply to the program and beta test new onboard services that could be canceled at any time. This way, BMW engineers could accelerate prototyping efforts and receive faster feedback from actual customers. One of the first features launched by BMW Labs was the integration of the "if this then that" (IFTTT) service that enabled the conjunction of miscellaneous web services. With onboard data, BMW drivers could use their car as a sensor for triggering any kind of action in many digital services; for example, when the fuel level went beyond a predefined level, the IFTTT service could set a refueling reminder in the driver's Google calendar. BMW also began cooperating with established digital service providers such as Spotify and Microsoft to implement Spotify Music and Microsoft Exchange apps in the BMW infotainment system. While these partners provided access to their servers, the onboard apps were developed by BMW.

The rapid emergence of more and more app development projects revealed problems with the platform itself. Although the emergence of innovative features was appreciated, the architecture of the underlying operating software was not designed for modular extensions. As illustrated at the beginning of this article, traditional automotive software development is tightly interwoven with associated hardware development. The memory consumption of every single software component is strictly predefined, which means that a component's budget is estimated and defined before the development efforts begin. Maintaining compliance with the available resources is a central goal of automotive software engineers. Accordingly, the overall memory budget is a sum of all component's memory budgets. However, this approach conflicts with the idea that apps can be continuously distributed into a system and consume arbitrary amounts of memory. Software architects mitigated this issue by introducing a specific platform component to receive a dedicated memory app's memory budget, which should be allocated dynamically. However, the scarcity of memory and its management became permanent issues for the newly created platform development team. Another problem category concerns inconsistent interfaces provided by foreign onboard modules. The interfaces revealed insufficient documentation or a lack of robustness in the context of generic use by a larger number of apps. Consequently, many app developers ended up taking a time-consuming trial-and-error approach to evaluate the functionality and limitations of interfaces:

> You just don't know which results in which condition are delivered by the interface. The documentation is either not available or just rather raw. All you can do is implement, deploy your app to a test vehicle and see what happens. (App Developer)

## *Apps for Automotive*

As illustrated in the previous section, directly distributing modular apps into the infotainment system led to several challenges for BMW. The development and operation of a digital platform exposed how fundamentally traditional automotive software development differs from modern software development. Engineers elaborated further approaches to integrating digital services into BMW's cars. Smartphones, which connect directly to the car, appeared to be a promising direction for integrating apps into the car's onboard OS.

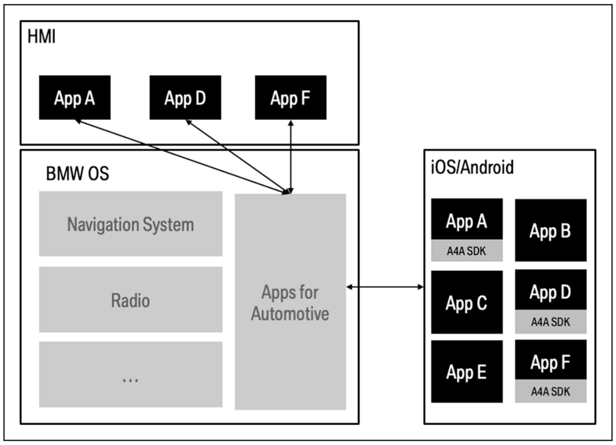



**Image 4.** Apps for Automotive architecture

In 2011, BMW had already enabled its infotainment system to display applications that were rendered on a connected smartphone. The "Apps for Automotive" (A4A) technology afforded mobile app developers to display user interfaces, by directly embedding them into BMW's proprietary infotainment system. Therefore, a software development kit (SDK) provided by BMW had to be integrated into the app's source code. The SDK exposed access to BMW's human–machine interface (HMI) and allowed mobile app developers to assemble their own user interfaces using standardized building blocks. While the app itself was still processed by the smartphone, it remotely controlled the displayed HMI in the car's central information display. When the customer connected his or her smartphone to the car, all A4A apps were listed in a dedicated menu (Image 4).

First, BMW used A4A to improve the integration of its own smartphone apps into its cars. For example, the BMW Connected App that enabled customers to control certain automotive functions by means of a smartphone was enhanced by a feature that, at the start of a drive, fetched the phone's calendar and suggested destinations in the car's navigation system. However, the focus of A4A technology was integrating third-party apps into BMW's proprietary infotainment system. Therefore, BMW enabled app developers interested in integrating their mobile apps to contact a dedicated partnership management division called the App Center. When the App Center determined that the integration of an app was valuable and met BMW's requirements, the A4A SDK was provided to the app developer. The App Center also provided comprehensive support to third-party developers to facilitate development and ensure the app's smooth integration with BMW's user interface. While many app developers contacted BMW, the reverse was sometimes also true. The music streaming provider Spotify and the online image hosting service Flickr were directly contacted by the App Center. In contrast to partnerships for onboard app development, BMW reached out to partners to build their own integrations of their services into the BMW infotainment system using A4A technology. All in all, over 50 audio, navigation, and entertainment apps that were available on the iOS and Android platforms were made compatible with BMW's proprietary infotainment system.

From a customer perspective, A4A technology was part of the overall BMW infotainment system product offering. When a customer configured his or her new car appropriately, all compatible apps installed on its smartphone were automatically made available in the car. In this way, each app developer who chose to integrate the A4A SDK into its app increased the attractiveness of BMW cars. App developers also increased the attractiveness of their apps by making them available during a car ride to BMW drivers. BMW's reputation and brand power enticed several smaller app development firms to partner with the automaker.

However, while several app development firms appreciated this cooperation, BMW also experienced reluctance from desired partners. Such reluctant firms argued that the number of new users that would specifically be attracted by the integration of their apps with BMW's onboard platform was quite low. The development effort required for such integration was remarkable, yet the number of potential users who did not already have an app installed and who owned an appropriately equipped BMW car was small. While a new Android or iOS feature usually has the potential to reach billions of users, A4A integration could not even reach millions of customers. In 2015, these difficulties increased, when BMW revealed the new generation of its infotainment system. All existing integrations required major updates by app developers to remain functional. While BMW increased its effort in partners, an increasing number of app developers terminated cooperation. Only a small number of A4A apps were available upon the launch of the next-generation system. Eventually, the A4A technology was removed from BMW's infotainment system in 2018 and replaced with a lightweight technology that enabled the integration of BMW's own smartphone apps, so no further external partners were acquired.

When considering projecting smartphone apps into an external infotainment system, most think of the solutions created by established digital platform companies such as Apple's CarPlay, Google's Android Auto, and Baidu's CarLife. BMW has a long history of cooperating with Apple to integrate iOS devices into their cars. In fact, the preceding "iPod Out" technology was a result of the joint development of BMW and Apple. It enabled the user to control its connected iPod via the vehicle's audio control buttons. Shortly after the release of Apple CarPlay in 2014, it was integrated into BMW OS 6, which was released in 2015. BMW announced the implementation of Android Auto in its OS 7 in 2020. Even though this implementation advanced the integration of the customers' digital world into their cars, this solution was not considered perfect by many experts. First, the functionality of projected modes was limited to the smartphone and did not exploit the vehicle's own capabilities. While there are well-established use cases for the smartphone, such as messaging or audio playback through a third-party OS, automotive-specific application programming interfaces (APIs) were not available to app developers. Second, multiple BMW experts emphasized the importance of a seamless customer experience, meaning that the integration of digital services should appear naturally. The conflict between projected modes through the existing infotainment system and the need to connect a smartphone to use them impeded providing such a seamless experience.

### Remote system updates and the emergence of platform governance

Improvements such as memory management optimization were usually implemented in product upgrades distributed to all newly produced cars as well as existing cars during maintenance visits by a service partner. However, very few customer vehicles were regularly serviced by a partner such that they could benefit from every product upgrade. By contrast, apps were distributed remotely and could be easily received by almost all BMWs on the road. The highly dynamic nature of apps, on the one hand, and the preintegrated software components' static development process, on the other, caused problems. Long-term platform development cycles impeded the team's ability to react to the quickly evolving world of apps. This mismatch was addressed by the introduction of the seventh generation of BMW OS in July 2018 (the most current generation at the time of this article's writing). BMW OS 7 enabled the remote distribution of product upgrades over the air, a highly anticipated feature. First, other premium brands such as Mercedes-Benz and Audi started promoting their wireless updating capability in the beginning of 2018. Tesla implemented remote software updatability in its Model S in 2015. BMW's image as a premium brand and technology leader required a prompt catch-up. Aside from these marketing aspects, engineers appreciated this advancement as well. New features in preintegrated software components, such as the navigation system or the radio, could be distributed to a large number of customer vehicles simultaneously. The platform team also desired the updatability of their component. Remote system updates enabled the iterative evolvement of the BMW platform and therefore more appropriate reactions to current developments by the internal app development teams.

Besides the growing complexity of runtime resource management, the platform team recognized an increased demand for the governance of platform processes. The proven procedures of preintegrated software components did not meet the requirements of the highly dynamic nature of app development. While common product upgrades were released three times a year, several app teams updated their apps many times a month. Furthermore, the platform was used not only by a small group of software engineers but also by hundreds of app developers. The established communication routines of cooperating teams in the development division did not scale for one platform team that had to provide support to dozens of app teams:

> Actually, in traditional software development here at BMW engineers just work for their department. There is no intention for developing things for others outside their own department. (Process Manager in the Development Division)

Therefore, the platform team introduced multiple measures. First, documentation for app developers was moved from the internal wikis with restricted access to a developer portal accessible to every BMW employee. Furthermore, several tools were created to enhance app development, and large parts of the app release and testing process were automated. The platform team introduced a question and answer forum and fostered the emergence of an app developer community to provide mutual support and reduce platform developer workloads.

With the launch of BMW OS 7, even more teams decided to implement functionality as an updatable app in the infotainment system. In addition, management recognized its rising importance and increased the allocated workforce to the platform. While the previous generation of the platform was developed and maintained by one internal and five external developers, the new platform was operated by a team of 15 internal and more than 30 external workers. However, the general attitude of BMW's developmental divisions still attributed a higher importance to feature development than to the development of a generic platform:

> Most of the management attention, most of the key performance indicators and most of the budget allocation mechanisms follow the logic of feature implementation. You usually need to provide a business case to receive funding for a development project. However, it is quite harder to provide a business case if your product—in our case the platform—cannot be sold to customers directly. (Member of the Platform Development Team)

While several new apps had just launched with the seventh generation of the infotainment system, the apps that existed in the previous generation required porting to the new OS version. Most involved managers and engineers expected all features that were available in the predecessor head-unit to be available in the launch of the new infotainment system, BMW OS 7. However, the platform component itself was bound to the regular software development procedures, which considered platform completion to occur weeks before the beginning of the head-unit's production cycle. Therefore, app developers were forced to follow this strict schedule and start their work before the platform's stability was determined, which caused large efforts on their side and amplified the development of new features.

## The entrance of Google

In May 2017, Google announced the development of Android Automotive OS, an Android operating system familiar to millions of mobile app developers and tailored to run in the head-unit of a car. This system should no longer depend on a connected smartphone as did the previously mentioned Android Auto, but be directly embedded in the car. As its first partners, Audi and Volvo declared that their proprietary infotainment systems would be replaced by Google's Android Automotive. After a 2-year development phase, Google invited app developers to submit automotive apps using their new automotive operation system in May 2019. In parallel, Volvo announced the release of the Polestar 2 model as the bearer of several innovative technologies including the implementation of Android Automotive. Aside from the basic operation system itself, Android Automotive OS included Google's most prominent and popular apps like Google Maps and Google Assistant as well as the Google Play Store, which was the entry point for third-party apps. While millions of apps were available for Android smartphones, support for Android Automotive OS was not provided by default. First, Google needed to extend and adopt Android's capabilities to the automotive context. Subsequently, developers needed to implement these capabilities to also adopt their apps for the automotive context. For instance, large parts of the system were locked to user input whenever the vehicle's velocity exceeded a certain predetermined limit. This speed-lock feature is a regulatory requirement in several countries, so apps must be notified to appropriately manage this system state. The graphical user interface (GUI) also needed to be prepared for interactions that occur during a car ride, such as sizing certain elements appropriately and avoiding unnecessary distractions. In the beginning, the provided GUI libraries of Android Automotive OS focused on specific types of apps. According to Google, media apps that enabled music streaming, the playback of audio books, or the streaming of radio broadcasts were the most desired domain of apps for automotive implementation. Therefore, in May 2019, media apps were afforded the first chance to adopt to Android Automotive OS. Later, Google announced that it would support apps from the navigation domain as well as the communication domain.

After the announcement of the Android Automotive OS launch in Volvo's Polestar 2, several other car manufacturers proclaimed their release of an Android-based infotainment system. However, while brands such as General Motors and the Renault–Nissan–Mitsubishi alliance promoted cooperation with Google and the implementation of Android Automotive OS, others, such as the Volkswagen Group and Fiat-Chrysler, declared the development of their own digital platform based on the Android Open Source Project. While both approaches included the implementation of Android as an operation system on automotive head-units with apps preinstalled by the car manufacturer, only Android Automotive OS included Google's most prominent and popular apps such as Google Maps, Google calendar, and Google Assistant. When users registered their Google accounts in their new cars using Android Automotive OS, they were automatically given access to the already familiar Google, which included their preprogrammed personalized preferences and utilization history. In addition, Android Automotive OS contained the Google Automotive Services (GAS). GAS is a collection of services that can be implemented by app developers. For example, apps can use location data provided from Google Maps, integrate the Google Assistant to enable speech interactions with the user, and manage financial transactions such as subscription fees or in-app purchases with Google Pay. While car manufacturers needed to create their own alternatives to these and other services, Google was able to provide access to already proven solutions from its mobile app ecosystem.

Eventually, Android Automotive OS implicated the availability of Google's Play Store with all third-party apps that were adopted to the automotive context. Developers that decided to submit their apps to the Play Store were assured that their apps would be available in all cars with Android Automotive, independent of the car manufacturer. Google ensured that all specified standards were satisfied by manufacturers and app developers and guaranteed a functional interplay. A car manufacturer with its own marketplace needed to establish own standards, independent of Google's App Store. Furthermore, apps that implemented features based on GAS were dependent on the availability of these services. For example, an app that provided information on electric charging station locations required access to a kind of map that allowed the implementation of custom points of interest in its user interface rather than of Google Maps. If no appropriate alternative was available, the feature would not be available in the car manufacturer's infotainment system (Image 5).

## BMW's strategic options

The appearance of Android Automotive OS accelerated the development of digital services for the automotive market and forced the car manufacturer to make decisions to own automotive app strategies. Should they join Google's ecosystem or strike out on their own? The integration of vibrant digital platform ecosystems appeared to be a critical success factor for the future of the automotive business. While some competitors had already announced their decisions, most manufacturers were still searching for an appropriate answer on Google's offer. In the fall of 2019, BMW was also evaluating a variety of strategic options. Three directions were discussed by the experts involved: creating a proprietary BMW platform ecosystem, establishing a collaborative platform ecosystem, or joining the existing ecosystem:

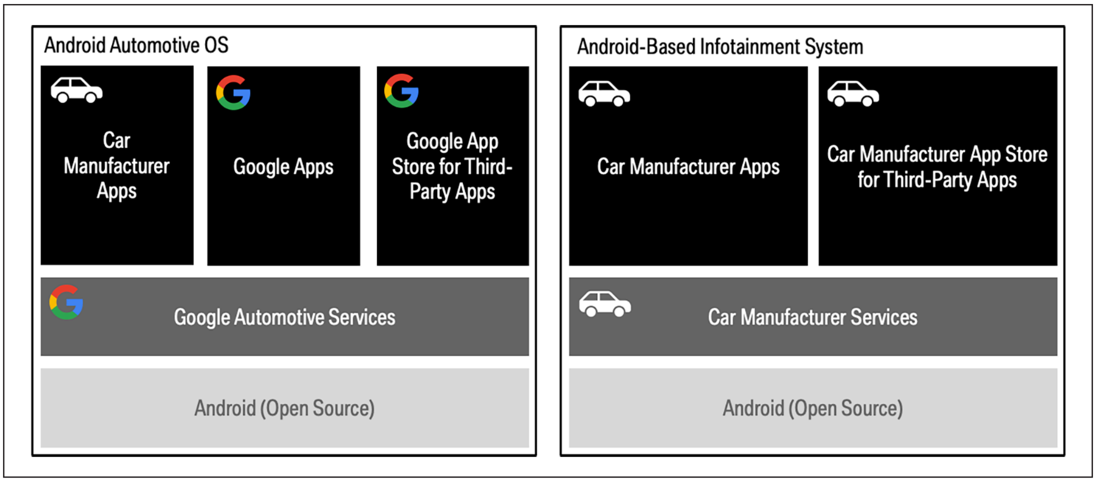



**Image 5.** Android Automotive OS vs Android-based infotainment system.

> Of course, we are in a competitive situation, but we are also in a partner situation. We will always be in this "frenemy" situation with a Google, with an Apple and whoever comes next. The question is of course how do I play that? (Digital Product Manager)

## *Creating a proprietary BMW platform ecosystem*

The first strategic option was the establishment of a proprietary digital platform developed and operated by BMW. This meant that BMW would follow its prevalent strategy of owning the digital platform but would make it accessible to third-party developers. Several experts inside BMW emphasized that in this scenario BMW would maintain comprehensive control of all decisions regarding its digital platform ecosystem. For example, BMW could independently decide whether app developers could utilize the data on the battery status of its electric cars. Consequently, this information would not be exclusively available to BMW, so the establishment of profitable business cases would be aggravated. However, the potential for innovative solutions for a car's charging process would be fostered, increasing the overall attractiveness of BMWs. The accessibility of autonomous driving capabilities could be limited to BMW to diminish the danger of misuse and associate the autonomous driving experience with BMW. While such deliberations would also be possible in cooperation with other partners, the existence of an independent BMW platform ecosystem would facilitate such decisions:

> On the other hand not only profit is an outcome but also other things like data sovereignty and new business models, that I can possibly build if I keep the data. If I have my own system and keep control of it, I can potentially do more than selling cars [. . .]. (Member of the Platform Development Team)

Considering the technological perspective, the system could be based on BMW's prevalent infotainment system, on Android Open Source or any other appropriate technology. However, due to its maturity and robustness, Android Open Source appeared to be a promising approach for many BMW software engineers. The system was established and mature, and several other manufacturers had already declared a shift toward Android Open Source. Consequently, the project would receive further maintenance and evolvement. Furthermore, the technology was already popular with developers. Its large amount of documentation, tutorials, and online support communities like Stack overflow facilitated the work of software engineers in comparison with using a proprietary solution. Moreover, Android Open Source facilitates the attraction and integration of new developers:

> I then I could imagine and this is my personal opinion, then we do it ourselves with a very limited number of apps, but then exactly with those apps we want. (Sales Operator for Digital Products)

Even though recruiting new developers to the platform would be facilitated by the implementation of an Android-based platform ecosystem, the challenge of attracting third-party developers would remain. Like the approach to A4A technology, the number of potential target devices for a BMW-specific digital platform ecosystem would be small. Furthermore, multiple BMW experts agree that the massive effort required to launch its own platform ecosystem would need to be justified by the aforementioned advantages. Some of the most important resources for establishing a well-functioning platform ecosystem are money and employees. Tremendous financial investment is required to establish and operate a successful platform ecosystem. Considering employees, previous digital platform development efforts bare a certain lack of attention to the development and operation of digital platforms. The required resources, available competencies and skills, as well as BMW's ability to attract them impact the overall platform ecosystem strategy. Most BMW experts agree that attracting and retaining the most creative minds in the field is important for developing an outstanding platform in an increasingly competitive field. However, this is especially critical because the car manufacturers need to simultaneously engage in various new fields, which leads to high efforts and competency requirements. Establishing all of the required competencies seems to be unrealistic for several BMW experts, which prompts them to suggest that the manufacturer should focus on its core competencies and consider what the company is willing to invest in other fields.

## *Establishing a collaborative platform ecosystem*

The fierce competition in the automotive industry indicates that one manufacturer will not be able to leverage its sales to achieve a similar magnitude of available target devices for previously mentioned mobile app ecosystems. Several experts at BMW consider a collaborative approach in which various manufacturers develop and run a common digital platform ecosystem as a further strategic option. While each partner could maintain its own brand-specific GUI or other features, third-party apps could be compatible with cars of all manufacturers. Considering the smartphone market, the market share of such an automotive consortium could not exceed 50%. In 2019, just 13% of all smartphones were run on iOS, while Android revealed an 87% market share.[8] Although the Android Play Store offers 2.6 million apps compared to the iOS App Store's 1.8 million,[9] the iOS App Store revenue was twice as high as that of the Play Store.[10] This example indicates that a consortium would need to strive for a relevant rather than dominant market share to attract third-party developers and generate value.

A consortium-based platform ecosystem would distribute all inherent costs and risks beyond the partners involved. Development and operations effort could be equally distributed among the partners, and any potential lack in competency could be mutually compensated. Each new partner would increase the number of cars available to third-party developers, decrease the risk to all other consortium members, and contribute resources to the development and operation of the platform:

> [. . .] The platform ecosystem participation thought is quite interesting. It can find favor with volume manufacturers as well as premium manufacturers. Everything scales much better when I have two-digit millions of new vehicles and they can still be addressed to some extent over lifetime. (Project Manager in Platform Development Division)

Although manufacturers could pool their endeavors to develop a digital platform ecosystem, their organizations would need to cooperate with external suppliers. The development of a common digital platform would also require support from delegated contractors. The commission of external software development firms by a larger consortium of car manufacturers with a relevant market share, however, implicates massive risk regarding the potential limitations of German and European competition laws. Several BMW experts considered this aspect to be a major challenge to the creation of a collaborative ecosystem:

> [. . .] Ideally a platform or an ecosystem already exists and we only participate in it, because that is simply more promising than doing it ourselves. But it is also an effort issue. Somebody has to do all of that. The purchasing department does not have to negotiate with a store provider and conclude a contract, it has to conclude contracts with 50 app providers worldwide, or how many we want to have. Furthermore, we have to validate 50 apps and do all of these things. That has to be done. It is an effort issue. (Digital Product Manager)

Furthermore, many experts emphasized that all approaches to the development of common automotive software components have failed in the past. Even though these attempts focused on the development of a common middleware and an operation system for cars, independent of the creation of a digital platform ecosystem with third-party developers, automotive manufacturers still failed. The fear of losing their competitive advantage and their conviction that their individual technical solutions are superior has impeded the establishment of common standards.

## *Joining Google's platform ecosystem*

Finally, experts considered the option that BMW might join an existing and growing platform ecosystem by implementing Android Automotive OS. Like the implementation of Android Open Source, this scenario also entails the utilization of a mature and proven technology, which should decrease testing efforts as compared with the creation of an independent, proprietary platform. With apps like Google Maps, Google Assistant, and others, the Play Store and GAS, Android Automotive OS reduced the software development effort required by each manufacturer to a minimum:

> Considering technical and cost reasons I would only take Android Automotive OS because the platform has already proven itself [. . .]. (Digital Product Manager)

GAS would facilitate the development of third-party apps and ensure solid payment mechanisms for all platform transactions. Furthermore, Google provides comprehensive support to app developers, including tooling, documentation, and collaborative events like developer conferences. BMW would not have to be responsible for the quality of the apps provided in the app store. With its already established control mechanisms, Google would ensure the exclusion of undesired or malicious apps.

Considering the customer perspective, multiple sales experts at BMW emphasized the seamless integration of the customer's digital world into a car afforded by an implementation of Android Automotive OS. Third-party apps that were adopted to the automotive context could be purchased on the customer's smartphone and automatically also be available in the car. Customers would just need to register their Google accounts in the car, and all playlists, pictures, and videos, as well as all personal data and logins would be synchronized and available without any further effort. In addition, Google's most popular services such as Google Maps and Google Assistant are exclusively available with Android Automotive:

> [. . .] the end user will certainly demand that he not only takes his ecosystem from home, but also that he doesn't have to log in anywhere all the time and then that's it. That speaks more the language that he takes this ecosystem than the manufacturer system [. . .]. (Sales Operator for Digital Products)

Even though the advantages of Android Automotive OS appear to be magnificent, several BMW experts raised doubts against the option. BMW would lose control over the development and rollout of a central component of its cars and would have to rely on Google's decisions regarding new features, bug fixes, and also new version rollouts. Moreover, Google would control which models or markets should first receive any OS version. Besides these strategic decisions, experts mentioned that BMW would lose an important customer contact point. While Android Automotive OS provides options for user interface customization, customers might think of Google, not BMW, as the system's provider. The customer experience during a car ride would no longer be exclusively designed by BMW, but a large part would be determined by Google. Strong brand bonding is especially important in the automotive market because cars and driving are especially associated with customer emotions.

Furthermore, experts elaborated that the implementation of Android Automotive OS implicates relinquishing any ambition on digital business model implementation inside BMW's cars. Since Google would manage all transactions on its platform and capture a predefined percentage of each transaction, the car manufacturer would not be involved. In line with the smartphone market in which manufacturers' value capture is limited to the sale of the device itself, BMW's business model would be limited to the sale of the individual car and several associated maintenance services. The role of a platform owner with scaling platform businesses is obsessed by Google. Experts from BMW's strategic departments link this aspect to the changing nature of the automotive market. While large parts of this business are conducted with sales or lease contracts, an evolution toward a more usage-based business model is expected. Even though selling cars is expected to remain as a lucrative business in the future, the market share of mobility services will grow. Independent of specific solutions such as car-sharing, ridesharing, and ride-hailing, these businesses are usually used and managed via online apps. As soon as digital platforms are available, the integration of these services with cars appears as logical step. However, the implementation of Android Automotive OS would mandate these transactions to be exclusively managed by Google. BMW would not be involved in such business models.

## Questions for reflection (teaching notes to support these questions are available)

1. Think about available features and the automotive sensor data that could be used by app developers via API. What apps could be built upon these features and data? Think about current vehicles, but also consider predictable future features such as autonomous driving.
2. What stakeholders are involved in ConnectedDrive? How could a digital platform for automotive apps connect different sides of the market? How could BMW benefit from connecting these sides?
3. Think about the challenges that appear when you start to connect the various sides of a single market. What is the most critical aspect to consider? Which strategies might help a new platform owner to master the inherent challenges?
4. What are the main challenges to BMW in implementing its own digital platform in its cars? For each challenge, consider whether it is specific to BMW or whether you would expect similar conditions at other firms? Consider technological as well as market-specific aspects.
5. How should BMW proceed from here? Consider the chances and risks of the three illustrated options and think about both internal and external factors that might influence its decision.

### Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

### Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

### ORCID iDs

Niklas Weiss https://orcid.org/0000-0003-0401-287X
Manuel Wiesche https://orcid.org/0000-0003-0401-287X

## Notes

1. https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/profiling-tomorrows-trendsetting-car-buyers (accessed 16 March 2020).
2. https://www.bmw-connecteddrive.de/app/index.html#/portal/store/Base_Connected_Parking (accessed 10 March 2020).
3. https://www.bmw-connecteddrive.de/app/index.html#/portal/store/Base_Connected_Music (accessed 10 March 2020).
4. https://www.bmw-connecteddrive.de/app/index.html#/portal/store/Base_ExchangeOffer (accessed 10 March 2020).
5. https://www.bmw-connecteddrive.de/app/index.html#/portal/store/Base_AssistNoTPEGOffer (accessed 10 March 2020).
6. https://www.bmw-connecteddrive.de/app/index.html#/portal/store/Base_Connected_Parking (accessed 10 March 2020).
7. https://labs.bmw.com (accessed 22 March 2020).
8. https://www.statista.com/statistics/272307/market-share-forecast-for-smartphone-operating-systems/ (accessed 3 April 2020).
9. https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/ (accessed 3 April 2020).
10. https://sensortower.com/blog/app-revenue-and-downloads-1h-2018 (accessed 3 April 2020).

## Author biographies

Niklas Weiss is a participant of the graduation program at the BMW Group and doctoral candidate at the Technische Universität München (TUM) in Germany. His current research comprises on the the emergence of digital platforms and digital platform design in the context of incumbent companies. Niklas Weiss has already published his research in HMD Praxis der Wirtschaftsinformatik and international conferences such as AMCIS and ICE/IEEE ITMC.

Maximilian Schreieck is a postdoctoral researcher at the Chair for Information Systems, Technische Universität München (TUM), Germany and holds a doctoral degree in information systems from TUM. His research interests include platform governance and enterprise software systems. His research has been published in Electronic Markets, Information Technology for Development, and conference proceedings such as ICIS, AOM, ECIS, HICSS, AMCIS, and PACIS.

Manuel Wiesche is full professor and chair of Digital Transformation at TU Dortmund University. He graduated in Information Systems from Westfälische Wilhelms-Universität, Münster, Germany and holds a doctoral degree and a habilitation degree from TUM School of Management, Technische Universität München, Munich, Germany. His current research interests include IT workforce, IT project management, digital platform ecosystems, and IT service innovation. His research has been published in MISQ, EJIS, JMAR, CACM, I&M, EM, and MISQE.

Helmut Krcmar is full professor of Information Systems, Department of Informatics, at Technische Universität München (TUM) with a joint appointment to the School of Management. His work experience includes a PostDoctoral Fellowship, IBM Los Angeles Scientific Center, and Assistant Professor of Information Systems, Leonard Stern School of Business, NYU, and Baruch College, CUNY. From 1987 to 2002 he was Chair for Information Systems, Hohenheim University, Stuttgart, where he served as Dean, Faculty of Business, Economics and Social Sciences. Helmut's research interests include information and knowledge management, engineering, piloting, and management of innovative IT-based services, computer support for collaboration in distributed and mobile work and learning processes. Helmut co-authored a plethora of research papers published in major IS journals including MISQ, JMIS, JIT, JSIS, ISJ, EJIS, I&M, CAIS, TOCHI and BISE. In Germany, his book "Information Management" is now in a 6th edition (2015). Interdisciplinary work incorporates areas such as accounting, mechanical engineering, and health care. Helmut collaborates in research with a wide range of leading global organizations. He is a Fellow of the Association of Information Systems (AIS) and member of acatech-National Academy of Science and Engineering.

# FROM PRODUCT TO PLATFORM: HOW CAN BMW COMPETE WITH PLATFORM GIANTS?

# TEACHING NOTES[1]

**Niklas Weiss**
Technical University of Munich
Boltzmannstr. 3
85748 Garching, Germany
niklas.weiss@tum.de

**Maximilian Schreieck**
Technical University of Munich
Boltzmannstr. 3
85748 Garching, Germany
maximilian.schreieck@tum.de

**Manuel Wiesche**
Technical University of Dortmund
Martin-Schmeißer-Weg 12
44227 Dortmund
Manuel.Wiesche@tu-dortmund.de

**Helmut Krcmar**
Technical University of Munich
Boltzmannstr. 3
85748 Garching, Germany
helmut.krcmar@tum.de

## 1 Introduction

### 1.1 Audience and Focus

This teaching case illustrates the challenges of BMW, a traditional firm that is confronted with the digitalization of its physical product and the inherent implementation of digital platforms. It is intended for undergraduate as well as graduate students of Information Systems and contains rich insights on technological as well as business aspects of digital platforms. The overarching question of the case contemplates BMW's strategical options for digital platform ecosystems and encourage considerations on the "platform conundrum" that several traditional firms are confronted with. After a description of the intended learning objectives, the remainder of these teaching notes contains a proposal for elaborating this case within a student class. The *Teaching Approach* is structured into four major sections: Technical Aspects of Digital Platforms, Market-oriented Aspects of Digital Platforms, Challenges of Traditional Firms, and Digital Platform Ecosystems and Platform Options. Furthermore, the *Additional Material* section contains further exercises that facilitate a deeper understanding of the previously introduced concepts.

[1] The teaching case document is available from the Journal of Information Technology Teaching Cases via open access: https://journals.sagepub.com/doi/full/10.1177/2043886920944185.

### 1.2 Learning Objectives

The presented teaching case supports students in:

- Understanding fundamental principles of digital platforms (layered architecture, two-sided markets, network effects, platform governance, platform options) using the example of automotive infotainment systems
- Understanding challenges of a traditional firm that is confronted with the digitalization of its physical product.
- Understanding dynamics of digital platform ecosystems and the inherent danger of disruption and conceive the revealed options ("Make or Join") for traditional firms that are confronted with the "platformization" of their product.

## 2 Teaching Approach

The following section provides exercises and questions for reflection on the concept of digital platforms and platform ecosystems. The questions address students' analytical and creative thinking. We provide a possible outline for a 120 minutes class discussion. While the recommended questions for reflection are attached to the teaching case and are essential for a fundamental understanding of the presented coherences, the optional exercises in section 3 can be applied in a more flexible way and are not part of the estimated 120 minutes. The approach assumes that the teaching case was distributed in advance and the students are familiar with the content. The attached questions for reflection should be prepared by the students in order to initiate a lively discussion in the classroom.

### 2.1 Introduction

**[15 Minutes]** The topic can be introduced by raising the question on examples of digital platforms that students know from their daily life. Thereby, students should recognize that there are different types of platforms. While innovation platforms enable the creation of complementary products, services or technologies, transaction platforms focus on the facilitation of interactions within a multi-sided market. Furthermore, there are hybrid platforms that enable complementary innovation as well as transactions between different sides of a market (Cusumano et al. 2019; Schreieck et al. 2016).

Examples for innovation platforms:

- Desktop operation systems such as Windows or MacOS
- Gaming systems such as Sony's PlayStation, Microsoft's Xbox or Nintendo
- Speech assistants such as Amazon's Alexa or the Google Assistant
- Cloud services such as Amazon AWS, Microsoft Azure, Google Cloud or Alibaba Cloud
- IoT systems such as Siemens' MindSphere
- Business system such as SAP's NetWeaver or the Salesforce App Platform

Examples for transaction platforms:

- Mobility platforms such as Uber, Lyft, Didi
- Booking platforms such as Airbnb, Booking.com, TripAdvisor
- E-commerce platforms such as Google Search, Amazon Marketplace or Alibaba Marketplace

Examples for hybrid platforms:

- Apple's iOS (innovation via apps, transaction via app store)
- Google's Android (innovation via apps, transaction via app store)
- Facebook (innovation via external apps, transaction via advertisement)
- Microsoft's Office 365 (innovation via external extensions, transactions via marketplace for extensions)
- Tencent's WeChat (innovation via mini-programs, transaction via advertisement)

The following box with additional information illustrates the power of digital platforms. The most successful platform companies established hybrid platforms or in Amazon's and Alibaba's case firms operate an innovation as well as a transaction platform.

**Additional Information**

In 2019, the top ten of the largest companies of the world by market value revealed seven firms that rely on platform business: **Apple (961.3$ billion)**, **Microsoft (946.5$ billion)**, **Amazon (916.1$ billion)**, **Alphabet (863.2$ billion)**, Berkshire Hathaway (516.4$ billion), **Facebook (512.0$ billion)**, **Alibaba (480.8$ billion)**, **Tencent (472.1$ billion)**, JPMorgan Chase (368.5$ billion), Johnson & Johnson (366.2$ billion).[2]

## 2.2 Technical Aspects of Digital Platforms

Digital platforms reveal technical as well as market-oriented aspects (Cusumano et al. 2019; Hein et al. 2019; Schreieck et al. 2016). While this teaching approach covers both aspects, their considerations are separated in this teaching approach to facilitate the student's comprehension. This first section of the class focuses on the technical perspective of digital platforms, which is especially relevant for innovation platforms.

**[10 Minutes]** Now students should consider the previously developed list of innovation platforms. What do they have in common? What is especially relevant here? The discussion should lead to the subsequently illustrated (technical) definition of digital platforms.

**Definition**

**Digital Platforms**

[2] https://www.statista.com/statistics/263264/top-companies-in-the-world-by-market-value/ retrieved 28.04.2020

A digital platform embodies a foundational product or technology upon which complementary products or technologies can be developed (Baldwin and Woodard 2009; Tiwana 2014). Apple's iPhone and its iOS operating system embody a digital platform that provide a technological base for developers, who create new digital products, called applications ("apps"). Via so-called application programming interfaces (APIs), app developers have access to a broad variety of the phone's features like the camera or the audio speakers as well as sensor data like gyroscope or GPS. Generally speaking, apps implement features of the platform core by accessing them via APIs (Henfridsson and Ghazawneh 2013). While some apps as Instagram or TikTok are globally available and strive for the largest possible user base, others aim at more specific use cases as providing information on the municipal public transportation schedule to residents of one specific city. Independent from their use case, every new app enhances the iPhone's capabilities and attracts new customers (Hagiu and Wright 2015).

**[15 Minutes]** Now, students should present their prepared solutions for the first question for reflection. Goal of this exercise is to elaborate that the vehicle is not just a new digital platform for the adoption of existing apps but enables completely new and innovative use cases due to the mass amount of new capabilities.

### 1. Question for Reflection

Think about available features and the automotive sensor data that could be used by app developers via API. What apps could be built upon these features and data? Think about current vehicles, but also consider predictable future features such as autonomous driving.

### Possible Solution

| Feature | API |
|---|---|
| Parking app that displays parking options as the car approaches the configured destination in the navigation system. As the car is set to park mode, the app notifies cooperating parking services as ParkNow that the billing can be started. As soon as the driver returns to his vehicle, the incurred price is displayed in the car and can be automatically paid. | GPS Data, Parking Mode |
| By introducing so-called environmental zones, cities aim to reduce traffic-related $CO_2$ emissions. It can be assumed that driving bans will not only apply to diesel vehicles but to all types of combustion engines. Hybrid vehicles that have both an electric motor and a combustion engine must be able to react accordingly. A corresponding app could take care of the driving bans, which might even be adjusted daily to the current air values, and automatically switch to fully electric operation when driving into an environmental zone. | GPS Data, Driving Mode |
| Smartphones constantly try to learn from our usage behavior to smoothen the user experience. When you open the search function on your iPhone, several apps that are considered as the most appropriate are suggested. These suggestion base on properties as current time, current location or your last actions. A car could do similar things by automatically start the seat heating at a certain temperature. Since different people maintain different preferences, this smart heating will adopt to individual needs. It might be possible that the car is connected to your calendar, aware when you need to leave to work in the morning and preheats in a way that it is perfectly tempered when you enter it on a cold winter morning. Similar features are possible with air conditioning, window lifter or massage seats. | Temperature Sensors, Seat Heating, Air Conditioning, Window Lifter |

| | |
|---|---|
| The access to the vehicle interlock could enable app developer to use the car's trunk as public available mailbox for delivery services. It would not require a mailbox at work or at home but every location with connected cars nearby could serve as destination for your orderings. This would require that the owner of the car is aware of it and does not store own possessions in the car. However, he could be engaged by a reward that is earned with every delivery in the car. | Vehicle Interlock, GPS Data |
| The availability of all driving data as well as highly detailed traffic and map data could enable a driving coach for more sufficient driving. The app would notify the driver when he is stepping on the gas while nearing a crossing or traffic circle. Similar to sports apps on the smartphone, the app could visualize improvements, engage user by providing comparisons with other drivers | Driving Data (Accelerometer, Angle of Gas Pedal, Velocity, Breaking Power, GPS Data) and Traffic/Map Data |
| Similar to the previously described driving coach, an app could be a driving school for learner drivers. Modern cars are able to recognize traffic signs, traffic lights, other road users and could detect every mistake you did in your lesson. Furthermore, the app could optimize the next lesson to train your weaknesses and improve your overall skill in an optimal way. And even if worse mistakes were made, the vehicle could intervene if necessary and prevent worse things with driver assistance systems. | Traffic Sign Recognition, Traffic Light Recognition, Driving Assistance, GPS Data |
| Most taxis round the globe track the traveled distance with additional hardware, so-called taximeters. This additional hardware costs hundreds of euros, while it tracks similar data as the already build-in mileage sensors. Consequently, this functionality could be implemented by an app that is automatically managing the billing for customers by interacting with smartphone apps. Furthermore, the owner of a taxi fleet could get insights on centrally collected data of all taxis in the fleet. | Mileage Sensor, GPS Data |
| The introduction of electric cars and the need for charging them at home reveals new requirements for power supply infrastructure. An app that monitors the charging status of multiple cars could orchestrate parallel charging processes in a neighborhood. The charging plan could be optimized according to the user's calendar that the car is always fully charged when it is needed. The user could receive current information on the charging status on his smartphone. Furthermore, the app could integrate cars into existing smart grid solutions. | Battery Charging Management |
| An app could integrate the car in the user's smart home system. The garage door could be automatically opened when the car is nearing home. In the other way around, the car could warm up for an upcoming drive, when the heating inside the home is turned on. | GPS data, Air Conditioning, Seat Heating |

| **Additional Information** |
|---|
| A modern BMW car implements more than 1500 interfaces in its infotainment system. Apple's iOS provides access to roughly 200 APIs on an iPhone[3]. |

**[10 Minutes]** The exposing of features as a window opener for app developers is facilitated by the concept of a layered architecture. The concept of a layered architecture should be introduced

[3] https://developer.apple.com/documentation/ retrieved 16.03.2020

and discussed in the class. The discussion could be fostered with questions as “Why is layered architecture such an important concept in the context of digital platforms?” or “Which layers can be described in the infotainment system?”. An additional exercise on abstraction can be found in section 3.1.

| **Definition** |
|---|
| **Layered Architecture**<br>The idea of abstracting low-level computation processes evolves in the concept of a layered architecture (Yoo et al. 2010). On a hardware level, computers process information as bits that are represented by the presence (“1”) or absence (“0”) of electricity. However, software developer is not interested in designing electronical circuits on a micro-processor. Therefore, so-called assembler languages transfer binary code (“01001”) in human-readable code. However, assembler code still reveals a tight coupling to hardware, meaning that various processor architectures require dedicated assembler code. However, an app developer doesn’t want to care about process architecture of the hardware the app is running on neither. Therefore, so-called higher programming languages (as Java, Python, C, C++, Swift, Kotlin…) abstract assembler code and enable the writing of code that runs on different hardware, while being even better readable for humans (You can check Wikipedia for deeper insights in the world of programming languages). An app platform embodies another layer of abstraction that facilitates the development of apps even more. The advantage of layered architecture is not just the simplification (and thereby acceleration) of app development but also the interchangeability of layers. A performance improvement on a lower level brings advantages for all layers above. Layered architecture is a major driver for the rapid advance of computer technology in the last decades. |

## 2.3 Market-oriented Aspects of Digital Platforms

After elaborating technical aspects of digital platforms, students should understand the market-oriented platform perspective. Especially the idea of two-sided marketplaces and network effects should be imparted. Therefore, we propose to go back to the initially created list of transaction platforms. What do have these digital platforms in common? The discussion should lead to the subsequently illustrated (market-oriented) definition of digital platforms.

| **Definition** |
|---|
| **Two-Sided Markets**<br>While Apple provides support and tools for app development, the actual innovation is created by third-party developers that are not employed by Apple. So, why are hundreds of thousands of programmers motivated to develop and maintain apps for the iPhone without receiving any wages by Apple? The answer is the market-oriented perspective on digital platforms. A digital platform enables and coordinates interactions between target groups on two or more sides (Gawer 2009; Rochet and Tirole 2003). While app developers create new apps on the one side, the integrated iOS App Store promotes these apps to users of the iPhone on the other side. Whenever a user purchases an app, the digital platform manages the transaction and ensures that each side receives the stipulated value, thus that the user is able to download the app and that the app developer receives his/her the price in return. Thereby, Apple as platform owner captures a previously defined share of the transferred revenue and in this way benefits from each transaction that is conducted on its platform. |

**[10 Minutes]** Now, students should present their prepared solutions for the second question for reflection. This exercise aims at the understanding of multi-sided markets and in which way this idea is applicable to the automotive context.

### 2. Question for Reflection

What stakeholders are involved in ConnectedDrive? How could a digital platform for automotive apps connect different sides of the market? How could BMW benefit from connecting these sides?

### Proposed Solution

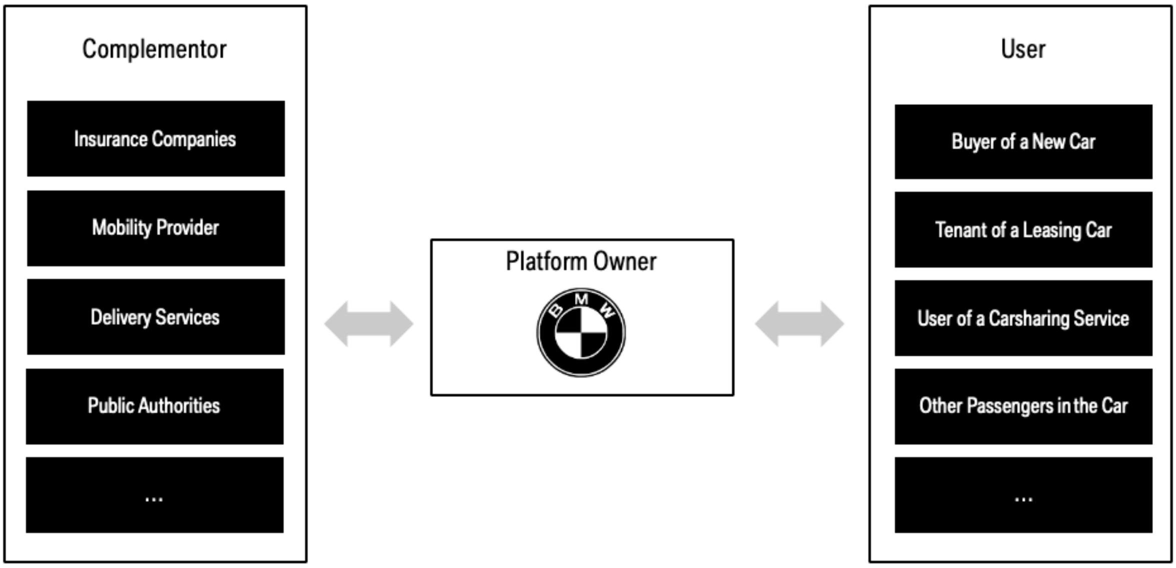



**Figure 1: Stakeholders of a two-sided market for automotive apps**

The platform could connect complementors that are interested in automotive data or providing services directly in the car on the one side and different kind of users on the other side. Thereby, several kinds of transactions could be managed by the platform and connect both sides in this way.

- First and obviously the platform could manage all purchases of new apps by providing a marketplace in shape of an app store. Similar to a mobile app store, BMW could keep a predefined share of the price that the customer paid for the app.
- The platform could enable that customers allow insurances to collect data from their cars and provide optimized contracts. As reward for enabling this transaction, BMW could receive a predefined share of each contract that relies on the collected data.
- A mobility service app (Lyft, Uber, Didi) that is directly integrated in the car could facilitate the booking of an available car. This is especially relevant when cars are driving autonomously and no driver with a smartphone is available. BMW could receive a share of each mediated drive.
- BMW could support authorities in providing information to customers or enabling smart traffic control systems. The contact to public authorities enables BMW to participate in the process of shaping new mobility solutions.

- As described in the proposed solution for the first question for reflection, delivery services could use the car's trunk as mailbox. BMW could receive a predefined share for each successful delivery.

**[15 Minutes]** The next section focuses on the chicken-egg problem that needs to be solved by every new digital platform and illustrates the central meaning of network effects. Therefore, the students' prepared solutions for the third question for reflection should be discussed.

**3. Question for Reflection**

Think about the challenges that appear when you start to connect the various sides of a single market. What is the most critical aspect to consider? Which strategies might help a new platform owner to master the inherent challenges?

**Proposed Solution**

Typically, the most critical aspect of connecting two sides of one market is getting both sides on board, while the respective other side is quite small or not existent. This problem is also known as the 'chicken-egg-problem'. BMW's new *Apps for Automotive* technology was just available in a small number of newly produced cars. Why should an app developer spend efforts, while no customers were available? On the other side, customers have no incentive for paying additional money for an app platform, where no apps are available. The chicken-egg problem needs to be solved by every new digital platform. Why should someone register an account for Uber, when no drivers are available? One the other hand, why should someone spend his time in a car and wait for customers, if no users are available? Similar scenarios occurred for Airbnb (hosts and guests), eBay (seller and buyer) or Facebook (users and advertisers). However, there are strategies that helped digital platforms to solve the chicken-egg-problem and become one of today's most successful businesses:

- Subsidize one side: When no side is available so far, the platform owner can subsidize one side to attract it and foster the onboarding of the other side. When Uber starts its business in a new city, it usually has to solve the chicken-egg problem every time. To solve the issue, Uber gives away vouchers that provide a few free rides for new user that register an account within the first weeks. The existence of users again, attracts drivers.
- Exploit available user base: Facebook was not initially designed as digital platform for advertisers but as social network for users around the globe. When a sufficient number of users was available, Facebook was able to use this huge market-side to attract the side of advertisers.
- Focus on specific target groups: Today, Amazon connects all kind of dealers with their customers. However, at the beginning Amazon started as online shop for books. This specialization increased the likelihood that new merchants would be connected to customers that are interested in their product—books. When the platform was established, Amazon extended its portfolio and started to become the global marketplace as it is today.
- Envelop existing platforms: By enveloping others, a new platform can leverage the shared relationships with other established platforms and their networks. When BMW developed its *Apps for Automotive* technology for Android as well as iOS, it tried to exploit the developer base of Android as well as iOS. Developers were not required to build completely new apps for BMW, but could integrate an add-on in their existing code.

When the chicken-egg problem is solved, the interactions of both sides provide room for unbound growth. Goal of this exercise is to develop the idea of network effects and its meaning for digital platform. The discussion should lead to the subsequently illustrated definition of network effects. Further exercises to two-sided markets and value creation on digital platform ecosystems can be found in section 3.2.

**Definition**

**Network Effects**

The extent of users is frequently considered as success factor of a digital platform. The reason for this is the existence of network effects (Rochet and Tirole 2003). In general network effects refer "to the impact that the number of users on the platform has impact on the value created for each user" (Parker et al. 2016; Schilling 2002). Imagine the invention of the telephone. The first person that owned a telephone did not receive any value from it, since there is no other person that could be called. However, a second telephone brought value to the owner of this second device as well as the owner of the first telephone. The third telephone again brought value to the first and the second owner and so own. Each new device increased the number of possible connections and created value for every other participant in the network. This effect occurs also for apps. Every new app in an app store might attract new users, which embody potential customers also for other apps in the store. The more users on the other side are participating, the more app developers are attracted to build high-quality apps that are again available for all user. Since these effects occurs between different sides of a market and create value they are called "positive cross-side" network effects. However, also negative and same-side network effects exist and influence a platform ecosystem (see (Parker et al. 2016).

**Additional Information**

The high variety of available apps is considered as one of the main reasons why Apple was able to outperform prevalent phone manufactures after the iPhone's launch in 2007. Back then Nokia boasted a market share of 63 percent in the mobile phone market, while Blackberry was the runaway market leader for mobile devices in professional contexts (Cusumano et al. 2019). However, the in-house development departments of these traditional firms were not able to compete with the rapid velocity of innovation that was created by Apple's third-party app developers. Furthermore, both firms failed with the establishment of own digital platforms. While Nokia sold its mobile phone sector for 5.44 billion dollars to Microsoft in 2014[1] (Apple reported a profit of 39.5 in 2014, mainly originated in iPhone sales[1]), Blackberry's market share needs to be measured in hundredth of percentage points[1].

## 2.4 Challenges of Traditional Firms

**[10 Minutes]** After understanding the fundamental idea of digital platforms, student should now elaborate in which way the concepts align with the traditional organization and technology of a traditional firm as a car manufacturer. Therefore, students should present their prepared solutions for the 4th question for reflection.

**4. Question for Reflection**

What are the main challenges to BMW in implementing its own digital platform in their cars? For each challenge, consider if it is specific to BMW or if you would expect similar conditions at other firms? Consider technological as well as market-specific aspects.

**Proposed Solution**

| Technological Challenges | Business Challenges |
|---|---|
| **Tight coupling of software and hardware development cycles**<br>As the episode on head-unit development illustrates, the development of automotive software is directly linked to the underlying hardware. When designing a new ECU, all hardware and software requirements are specified. The software is a part of the ECU and, like all other components, is made ready for the market in development processes that sometimes take several years. Once this process is complete, the ECU is secured and released for production. New specifications are then collected and new components developed for the next generation of the ECU. The development and use of a digital platform are difficult to realize with this approach. According to the technology-oriented definition of a platform, applications are based on the technical platform core. Developing the platform according to the established process means completing the software immediately before production begins. The development of applications based on this core is no longer feasible until the start of production. Since a newly specified platform will be developed for the next generation of the ECU in the classic procedure, app development is not possible for this either. For these reasons, the traditional coupling of software and hardware development poses great challenges for manufacturers with regard to the implementation of a digital platform in terms of development and production processes.<br>**Legacy in software architecture**<br>The coupling of software and hardware is also visible in the software architecture, which reveals a high amount of proprietary and hardware specific solutions. Even if BMW tries to increase the abstraction and reduce the amount of legacy, a complete refactoring of the software in one big bang is not possible in addition to the usual development efforts for new features. The legacy however, limits the manufacturer's agility and makes it difficult to keep up with the market innovation speed.<br>**Increased Complexity**<br>The introduction of new software layers in form of a digital platform and inherent apps increases the complexity within the software. This increased complexity of software requires the availability of sufficient expertise. However, attracting and retaining the most creative heads out there is important to develop new ways of generating revenues and | **Optional equipment as exclusive business logic**<br>According to the market-oriented definition, a platform represents a market that reinforces network effects of different players. In the case of applications, for example, an App Store represents such a market. Users can use the platform to enhance their original product with additional functionality and benefit from the work of external application developers. For an automobile manufacturer, extending the functionality of his product after delivery of the vehicle is an untypical process. After-sales activities are traditionally focused on repair and the sale of spare parts. Apps are therefore currently understood to a large extent as the usual special equipment that can be added to the vehicle during configuration. The sale of applications after the car has been delivered requires various changes and adjustments in processes and organization, but also in the manufacturer's IT systems.<br>**Feature-driven development funding**<br>Traditional companies usually focus on developing new features to differentiate their product from competing products. New or improved functionality are sales arguments to customers and generate revenue. A digital platform, on the other hand, does not generate any direct benefits for customers in the first step. Only the applications made possible by the platform offer the prospect of sales. However, internal evaluations for decision-makers are usually made directly or indirectly via the financial return on a development. Accordingly, function-oriented development projects are often given preferential treatment. The conversion of these mechanisms in order to also motivate the development of basic technologies internally more strongly represents a further challenge for the organization.<br>**Limitation of customer-side**<br>The power of digital platforms is originated in an economy of scale on the demand side, meaning that every new customer attracts further complementors due to cross-side network effects. However, the potential for customers that purchase digital services in a car is currently limited to people that own or rent BMW car. Due to the strong competition in the automotive industry, a change of market share appears not likely. Therefore, BMW has to find other ways to increase the scope of potential participants of its digital platform on the demand side to nurture a vibrant two-sided market. |

| | |
|---|---|
| providing an outstanding platform in an increasingly competitive field. Furthermore, not only additional competences are required due to an increased complexity, but also more financial resources. A vast amount of money is required for the establishment and operation of a successful digital platform. | |

## 2.5 Digital Platform Ecosystems and Platform Options

In this final section, students should understand in which way digital platforms are embedded in digital platform ecosystems and the inherent power of a platform owner. Therefore, students should understand platform governance. Furthermore, the platform options that are presented in chapter 4 of the teaching case should illustrate the prevalent platform conundrum for BMW as representor of multiple traditional firms that face the platformization of its physical product.

**[10 Minutes]** To understand the relevance on the decision between the presented platform options, the students need to conceive the central role of the platform owner. Therefore, the actual responsibilities of a platform owner should be collected in the classroom.

Responsibilities of a platform owner:

- Maintain relationships to all involved sides in the platform ecosystem (e.g. Promote platform to new customers and new developers, inform on changes, etc.)
- Define pricing and revenue sharing models (e.g. define amount of money that is captured per transaction)
- Provide resources for the creation of complementary products (e.g. Software Development Kits)
- Decide on the platform openness (e.g. who is allowed to develop apps)
- Prevent undesired behaviour (e.g. fraud, low qualitative complements, violent content)
- Develop and maintain platform core (e.g. the operating system)
- Define a competitive strategy towards other digital platform ecosystems (e.g. prevent compatibility with other systems)
- Ensure trust (e.g. customers receive the promoted complement after the purchase)

All these activities consider the management of interactions between different stakeholders in the digital platform ecosystem, also referred as platform governance. The discussion should lead to the subsequently illustrated definition of platform governance.

| **Definition** |
|---|
| **Platform Governance**<br><br>The core of a digital platform is in the center of every digital platform ecosystem (Tiwana 2014). The exposed capabilities of the platform should attract app developers (Dellermann et al. 2016; Kude et al. 2012), who build complementary apps on the one side, which should be consumed by customers on the |

other side (Tiwana 2014). Platform governance is the "partitioning of decision-making authority between platform owners and app developers, control mechanisms, and pricing and pie-sharing structures" (Tiwana 2014). For instance, in the iOS ecosystem, Apple as platform owner decides, which application programming interfaces (APIs) of the iPhone are accessible by third-party app developers (Eaton et al. 2015; Henfridsson and Ghazawneh 2013). Furthermore, each app needs to be submitted for review by Apple before it can be launched in the App Store. The app developer needs to choose the price of its app in a predefined selection of pricing steps and keeps just 70% of its revenue. The rest belongs to Apple as platform owner. However, platform governance should also facilitate the creation of new apps (Kude et al. 2012). By exposing generativity the platform owner unleashes potential for innovation (Tilson et al. 2010). With the annual release of a new iOS version, Apple publishes new APIs that provide new possibilities for app developers. Comprehensive documentation and annual developer conferences inform and attract developers to build new apps. A platform with strictly controlled generativity would counter such endeavors (Ondrus et al. 2015). Hence, the platform owner needs to balance control over the digital platform and its ecosystem on the one side and provide a certain level of autonomy for app developers to foster innovation on the other (Rausch et al. 2012). In this way, platform governance covers multiple tactical decisions that impact interactions between the platform owner and app developers (Kude et al. 2012; Schreieck et al. 2016).

**[10 Minutes]** After the students have understood the central importance of platform governance, the next step is to think about the way Google or BMW would fill the role of a platform owner to leverage their strategical interests. Therefore, the respective motivations need to be considered. The discussion should reveal the difference between a (traditional) resource-based strategy that strives for the exploitation of competitive advantage through the redeployment of resources and a network-based strategy, which aims for an optimal position in previously established value networks.

**5. Question for Reflection**

Think about Google's strategic motivation behind the engagement for a digital platform ecosystem for cars. In which way does this strategy differ from BMW's strategy for a digital platform in its cars?

- Google is a platform company that is world-leading in connecting different sides of a market to value (co-)creation network but does not reveal expertise in the manufacturing of own physical products. The building up of capabilities for own physical products is expensive and time-intensive. On the other side, Google does have existing digital platform ecosystems, which embody value co-creation networks.
- By implementing its digital platform in cars of several manufacturers, Google enables compatibility for apps in a large number of cars.
- This envelopment of the fragmented automotive industry creates a large network that connects customers of multiple automotive brands with a large amount of Android app developers. The emergent network is structured as platform ecosystem, with Google as platform owner in its center.
- The availability of Google's established and top-edge services as Google Maps and the Google Assistant attracts customers to join the platform ecosystem. Car manufacturers outside Google's platform ecosystem need to compensate these services to remain attractive.
- BMW is a traditional manufacturing company that strives for competitive advantage by the redeployment of resources, meaning that it utilizes available assets and knowledge to create products that appear desirable to customers.

- Considering this resource-based view, the implementation of a digital platform is a measure for making the product even more desirable for customers. The capturing of value by the platform is not covered in this sight.
- Another characteristic of a resource-based strategy is the aiming for a product that is rare, inimitable and non-substitutional. Any kind of cooperation with direct competitors endangers such aspirations. The exchange of information inherits the danger of losing competitive advantages. An implemented digital platform should be as exclusive as possible to create competitive advantage by providing apps and services that are not available in cars of other manufacturers.

Google's strategy is known as platform envelopment. The discussion should illustrate the idea behind that strategy and why it fits Google's strategic interests.

| Definition |
|---|
| **Platform Envelopment**<br>Sometimes, a digital platform begins to offer the functionality of another platform to enhance its existing bundle of functionality and leverage shared user relationships (Eisenmann et al. 2006; Eisenmann et al. 2011). The act of envelopment is also frequently referred as 'swallowing'. It occurs when two adjacent platforms have an certain overlap in its functionality or user base (Tiwana 2014). Furthermore, it is a launch strategy for a new digital platform that enters a market (Stummer et al. 2018). For example, when Apple bundles the news articles of several news providers in its News app, it is no longer a single news provider app that is mediating its content to the users but its Apple's platform. While the user base of Apple's ecosystem might push the popularity of certain news providers or specific articles, the providers are losing control in which way their content is provided to the user. |

**[15 Minutes]** Now, students should present their prepared thoughts for the final question for reflection. The discussion should emphasize the prevalent challenge for car manufacturers and illustrate the idea of platform options in general. Furthermore, students should understand the specific advantages and disadvantages of the presented options.

### 6. Question for Reflection

How should BMW proceed from here? Consider the chances and risks of the three illustrated options and think about both internal and external factors that might influence its decision.

| Strategical Option | Chances | Risks |
|---|---|---|
| Create own BMW Platform Ecosystem | **Control on portfolio**<br>An own BMW platform ecosystem would leave the full control on BMW's side. While market-rollouts, new platform features or also the guidelines, which kind of apps are allowed and which not would be decided exclusively by BMW while all other options would require agreements with other stakeholders. In this way, BMW's platform could stand out from other platforms.<br>**Governance of available capabilities** | **Attracting of third-party developer side**<br>The lack of an existing developer community requires the attracting of new third-party developers that develop apps for the relatively small customer side, which is limited to BWM drivers. The episode on *Apps for Automotive* illustrates that a failure of this attracting embodies a major risk for BMW.<br>**Financial investments**<br>The standalone development and maintenance of a comprehensive and matured platform ecosystem requires massive |

| | | |
|---|---|---|
| | BMW would not just decide which kind of apps would be supported but also which capabilities of the car are exposed to these apps. In this way, BMW could surpass other platforms and attract third-party developers with the enablement of new use cases, which are not available in other platform ecosystems.<br>**Value Capture**<br>An own BMW platform ecosystem would bring BMW in the position of a platform owner and enables value capture at every transaction that is managed through the platform.<br>**Acknowledgement**<br>The creation of an own vibrant platform ecosystem embodies a major challenge in several aspects. However, the successful establishment of such an ecosystem would be acknowledged in public perception and contribute to BMW's image as technology leader. | funding. However, OEMs as BMW need to manage multiple technological disruptions as the electrification of mobility or the emergence of autonomous driving in parallel. Hence, limited financial resources embody a risk for an own BMW platform ecosystem.<br>**Acquiring of new expertise**<br>Traditionally, software development is not a core expertise of a car manufacturer. Therefore, BMW needs to acquire new expertise at a massive scale. The failure of this recruiting efforts may endanger a successful BMW app platform.<br>**Technical development**<br>Each software development project carries inherent risks of failure due to inappropriate estimations of efforts or unforeseen technical impediments. This is of course also valid for the development of a digital platform. |
| Create Collaborative Platform Ecosystem | **Enlarged customer side**<br>A cooperation with other OEM's would potentially solve the problem of a too limited customer side. The multiplication of potential users could facilitate the attracting of third-party developers and enhance cross-side network effects.<br>**Collective resources**<br>Multiple experts inside BMW consider the required amount of expertise as well as financial resources as massive challenge for the creation of an own platform ecosystem. However, the cooperation with other OEMs provides the chance for joining forces and reducing the efforts for every single member of the cooperation. Furthermore, the expertise of one OEM could be compensated a potential lack within other OEMs and of course the other way around. | **Attracting of third-party developer side**<br>Even though, the cooperation of OEMs increases the potential customer side the non-existent complementor side is not neglectable. Also, a cooperation of OEMs would need to solve the chicken-egg problem and attract new third-party developers to provide an attractive app portfolio for customers.<br>**Cooperation with other OEMs**<br>The creation of a platform ecosystem requires decision-making regarding multiple organizational as well as technological aspects. However, a strong faith in the own brand and superior capabilities of the own organization were reasons for failure of cooperation between different car manufacturers in the past.<br>**Compliance with competition law**<br>Following their traditional development approaches, the creation of a common platform ecosystem would be supported by delegated contractors. However, as described in the case already, the commission of external software development firms by a larger consortium of car manufacturers with a relevant market share however, implicates massive risks regarding potential harm of German and European competition law. |
| Join Google's Platform Ecosystem | **Matured technology**<br>The Android operating systems embodies a matured technology that is currently running | **Loss of value capture in platform business** |

| | | |
|---|---|---|
| | on more than one billion devices. The inherent play services respectively automotive services are similarly proved. Furthermore, the transfer into the automotive domain is conducted by Google, a firm that is originated, experienced and leading in the development of digital platforms. The utilization of a solid technological base facilitates the implementation of a valuable product for customers.<br>**Availability of bundled services**<br>Joining Google's platform ecosystem entails access to Google's digital services as Google Maps or the Google Assistant in BMW cars. The availability of these services is desired by multiple customers and increases the attractiveness of the overall car.<br>**Existing third-party developer side**<br>The existing Android app developer community embodies a major advantage in comparison to options that entail the need for the establishment of a new developer community. BMW would not be required to solve the chicken-egg problem but would benefit from a large and highly qualitative app portfolio. | While the car as product strengthened by the implementation of a high qualitative platform and the availability of a large app portfolio, the access to any kind of value capture in this platform business is restricted. This is especially relevant in the context of servitization of mobility, in which customers rent mobility on-demand via digital services instead of owning an own vehicle.<br>**Loss of governance and control**<br>Since Google is the platform owner in its platform ecosystem, Google has the exclusive right to decide on developer guidelines, control mechanisms, available resources for developers, the revenue sharing and pricing models and more. Consequently, Google will not only decide which kind of apps are allowed or not allowed inside BMW cars but also which kind of apps need to be supported by BMW. However, this platform governance will primarily support Google's interests not BMWs.<br>**Loss of customer touch point**<br>The Android brand is considered as part of Google's corporate identity. Even though, the Android system for smartphones enables device manufacturers to customize the experience, the central design decisions are made by Google. This implicates that BMW would lose the control on the experience within the cockpits of their cars. Furthermore, the differentiation from other OEMs that also implement Google's platform is impeded. |

Beside the emphasizing of the inherent challenge of the platform conundrum, students should understand the concept of option thinking in the context of digital platforms. The following definition illustrates platform options and its origins in the investment literature.

**Definition**

**Platform Options**

The idea of real options is frequently used in the context of strategies on digital platforms and digital platform ecosystems (Tiwana 2014). Originally, the term is rooted in investment literature and refers to the possibility of doing something without the obligation to really do it. The idea of real options thinking is typically applied in the context of uncertainty and facilitates the management of a lack of knowledge regarding future events and coherences. While granting multiple options in parallel may cause additional costs, the risk of missing the right option is reduced. The highly volatile characteristics of digital platform ecosystems, with fast changing environments and rapid developments induces the application of real option thinking.

Further exercises for the illustration of the platform conundrum can be found in section 3.3.

# 3 Additional Material

## 3.1 Optional Exercise - Abstraction

Abstraction is used to make models that facilitate the implementation of complex processes with simple code snippets. Imagine a website that displays a simple black headline on white ground. If no abstraction would be available, the web designer would need to define which specific pixels on a display that shows his website should be black (= the text) while the remaining pixels would have to be defined to be white (= the background). However, when the user starts scrolling and the content should move these assignments need to be realigned. This has to be done for each and every display size and all kinds of devices, where the website should be available. Of course, this is not the case in reality. Modern web development builds on large stack of abstraction that assumes that tasks for web designers. Eventually, the illustration of a black headline on white ground requires one simple line of Hypertext Markup Language (HTML) code.

**Optional Question**

Think about the advantages of abstraction. Consider the lack of abstraction in the context of digital platforms. What problems occur due to the lack of abstraction?

**Proposed Solution**

- Low level code is harder to understand and requires app developers to spend efforts in understanding complex coherences.
- Low level code is harder to write.
- The more complex the implementation the higher danger of mistakes with major impacts. Find and understanding bugs and errors is exacerbated.
- General increased efforts for implementing new features
- Onboarding of new developers is aggravated

## 3.2 Optional Exercises – Value Creation

**Optional Question**

Think about the value creation process of a car in traditional supply chains and compare it to the value creation process of an app that is developed by a complementor. What are the differences? Try to illustrate both value creation processes to facilitate your analysis.

**Proposed Solution**

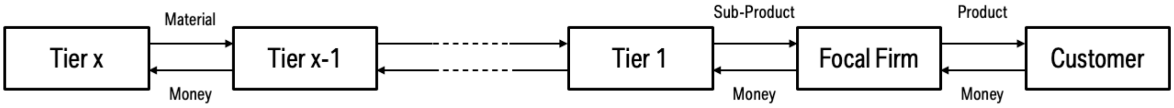



**Figure 2: Value creation in pipeline business**

Traditional value creation in supply chains can be considered as step for step arrangement, where value is transferred from Tier x to Tier x-1 to the focal firm (BMW) and finally to the customer. The value creation in this "pipelines" can also be described as linear value creation. Each transaction on the supplier side is arranged as contracted partnerships, where the amount of value-exchange is predefined. Value creation in pipelines benefits from the economy of scale within this supply chain. The optimization of operations enables the production of more goods with a constant amount of fix costs. Consequently, the costs per produced units decrease.

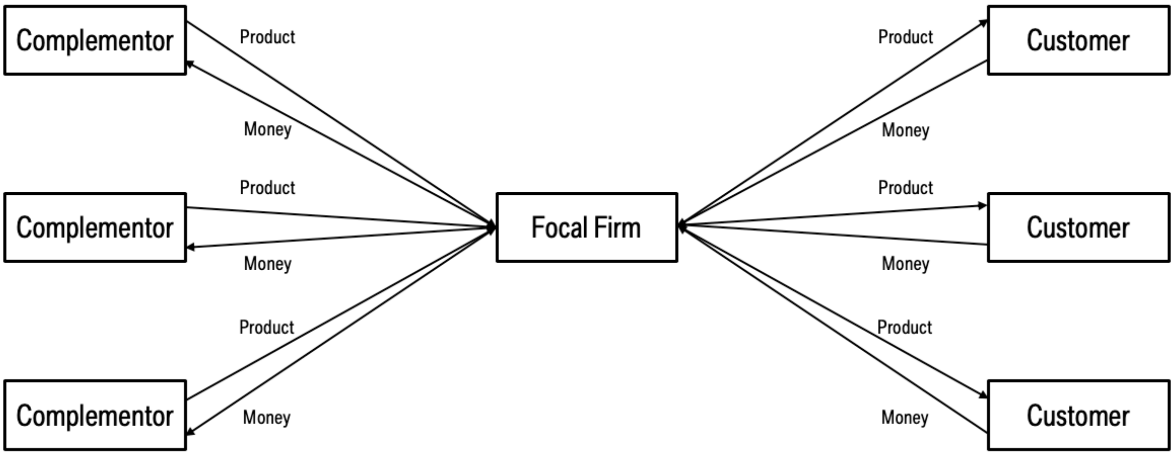



**Figure 3: Value creation in platform business**

Multiple successful businesses still rely on linear value creation. However, when a platform enters the market, the platform "virtually always wins" (Parker et al. 2016). One reason is that platforms create value with resources that they don't own. Consider BMW as large car manufacturer. It requires a globally disturbed set of manufacturing plants to produce round about two million cars a year. Each asset requires high efforts for maintenance, human resources as well as energy costs. Furthermore, a large network of supplier needs to be contracted and managed. All these efforts cause a significant amount of fix costs that are hard to scale. You cannot employ and fire band workers on a monthly or even daily base, just due to changing market demand. Similarly, a new plant itself cannot be build up within one day and supplier contracts need to be fulfilled. Uber, which - similarly to BMW - strives to sell individual mobility to customers, does not own o a single car. It simply connects drivers with customers. The amount of fix costs is reduced to server costs that are required to run their services and can be easily scaled due to cloud technology. In this way, platform business benefit from an economy of scale on the demand side (which is expressed by network effects). Another reason for the dominance of platform businesses is the elimination of gatekeepers that exist in pipelines. A traditional store defines a selection of products that is provided in its shelves and needs to rely on his knowledge and experience that the customers like the selected portfolio. Amazon on the other side can offer every kind of product that was submitted by a registered dealer. Sufficient products will receive good reviews, which again will attract more customers to it. Insufficient products will earn bad reviews and won't be bought by other customers. In this way the market determines by itself, which product will succeed and which will fail. The platform can scale more rapidly since the traditional gatekeepers - the portfolio managers of usual stores - are replaced by automatically provided market mechanisms as Amazon's product reviews.

### Optional Question

Is BMW Connected Drive a digital platform? Identify at least two pro arguments as well as two reasons why it should not be considered as digital platform.

### Proposed Solution

| Pro Arguments | Contra Arguments |
|---|---|
| **Modular app disposal and deployment**<br>Considering the technical definition of a digital platform, the platform module in BMW OS 6 and 7 embodies a technical foundation for other complementary products that can be developed on it. Due to the provided update and deployment mechanism the developed complements can be modularly disposed and delivered to the customer side.<br>**Orchestration of app developers**<br>Even though, the platform is not open for third-party developers, the organization within BMW corresponds to a platform owner, embodied by the responsible platform development department and multiple app developers, which are distributed over multiple business lines within BMW. The platform owner orchestrates the app development activities and implements required guidelines control mechanisms as illustrated in chapter 2.4.<br>**Enablement of innovation**<br>The availability of a SDK and modular app development, which is independent from the long-term development cycles of a car, enables the fast and iterative development of innovative products. Especially, the activities of the research departments and the establishment of the BMW Labs initiative illustrates characteristics of an innovation platform.<br>**Projected modes are established digital platforms**<br>The implemented projected modes embody matured digital platforms with a vibrant community of third-party developers, a two-sided market and frequent innovations by new apps that are available in BMW cars. The implementation in BMW cars entails the availability of a digital platform in the product offer of ConnectedDrive. | **Lack of platform architecture**<br>Even though, software engineers are aligning software architecture into layers, a platform architecture is not comprehensively implemented. Consequently, the lack of abstraction reveals frequent challenges for app developers as described in chapter 2.3 of the case. Furthermore, the current technical design impedes scaling business models. For instance, the fact that BMW pays all mobile network traffic impedes the profitable offer for video streaming services.<br>**(Almost) no third-party developers**<br>BMW does not provide any information or documentation that attracts third-party developers. In the latest version of BMW OS, the small number of apps that implement external services as Spotify or Microsoft Office 365 are crafted by BMW software engineers. BMW provides no official partner program or revenue sharing model that could attract third-party developers. Consequently, no two-sided market is established and no cross-side network effects are fostered.<br>**No economy of scale on demand side**<br>While the access for third-party developers is virtually not available, also the customer side is limited. Apps would be exclusively available for drivers of a BMW car. The limitation of access for other customers prevents an economy of scale on the demand side, which is required for a vibrant digital platform.<br>**Strategic focus on pipeline business**<br>Even though, BMW engineers as well as managers emphasize the importance of digital platforms, several business decisions as the removal of apps for automotive technology or the closed character of the BMW Store illustrate a focus on pipeline business. |

It is important to mention that BMW ConnectedDrive embodies a marketing term, which describes a portfolio of digital services and products in and around the car. Baseline of this exercise is that there are aspects of ConnectedDrive that reveal platform characteristics, especially from a technical perspective. However, the lack of a comprehensive platform strategy should be revealed.

## 3.3 Optional Exercises – Platform Conundrum

**Optional Question**

BMW is generally known as premium brand and technology leader. Think about other car manufacturers that reveal different characteristics. Do their considerations regarding platform options differ? How could their choice affect BMW?

**Proposed Solution**

The development of a comprehensive and high-quality infotainment system with dozens of features occasions a large amount of cost. BMW as a firm that considers itself as technology-leader and premium brand is willing to bear these costs to preserve this image and provide products that satisfy the expectations of its customers. However, the infotainment system is just one of multiple components, which need to meet this requirement and therefore cause large costs in their development. Other OEMs do not strive for such a premium image but aim at the offering of affordable cars for as many customers as possible. Consequently, the reduction of costs is an even larger driver within the organization of such mass producers than it is within BMW. The approach to completely diminish most of the development costs by implementing Google's matured and highly-quality Android Automotive OS should appear promising to these OEMs. While reducing costs on one side, the car itself is upgraded and keeps up or rather overtakes the infotainment features of premium brands as BMW. Google Maps is widely considered as the state-of-the-art solution for navigation apps and the Google Assistant is one of the leading solutions for personal voice assistance. Even though, BMW and other premium OEMs provide similar features, Google's services are not limited to the car but available at many other touchpoints of the user's daily life. Moreover, these services are heavily in use for a long time, which enables Google to learn and improve them since years. Basically, this raises the question why a customer should pay thousands of euros for a BMW infotainment system, when the more desired services are available for free in an even cheaper car.

Furthermore, due to cross-side network effects the implementation of Android Automotive OS by other OEMs does not only affect customers but also developers. Every additional manufacturer that decides for an implementation of Android Automotive OS provides access to more customers, which again attracts more developers for developing appropriate apps. Any other platform ecosystem needs to compete with Android Automotive OS and convince developers for providing an additional app for their platform. Developers need to "multihome" their app (Cennamo et al. 2018). Even if this platform is also based on the Android Open Source Project, the implementation of Google Automotive Services by third-party developers will impede a simply deployment to another platform.

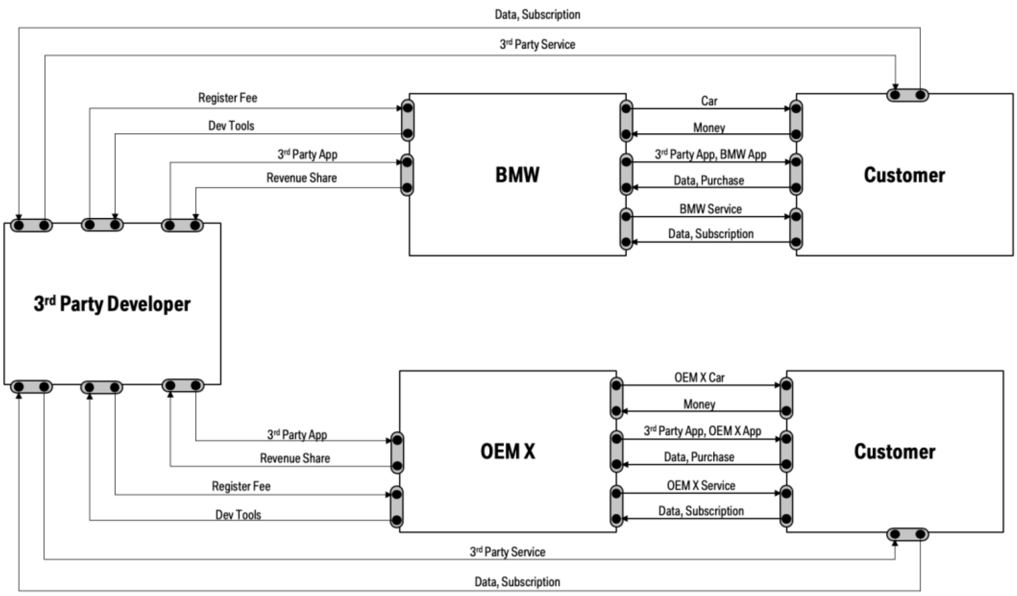
## Additional Information

The Android Open Source Project provides different shapes of Android, applicable for several domains. Android as OS for smartphones should be the most widely implemented form. While the basic operating system is published under an open-source license and with that free for any kind of utilization, the access to Google's Play Services is limited. The Play Services contain several features for app developers as notification services or transaction management via Google Pay as well as Google's most prominent apps as Search, Maps or the Assistant. They are exclusively available for members of the Open Handset Alliance (OHA), a consortium founded and led by Google (Karhu et al. 2018). All members of OHA agreed to satisfy Google's Mobile Alliance Distribution Agreements in which Google prescribes guidelines for the utilization of its Play Services. All manufacturers of the most selling Android smartphones as Samsung Electronics, HTC or LG electronics are participating in OHA.

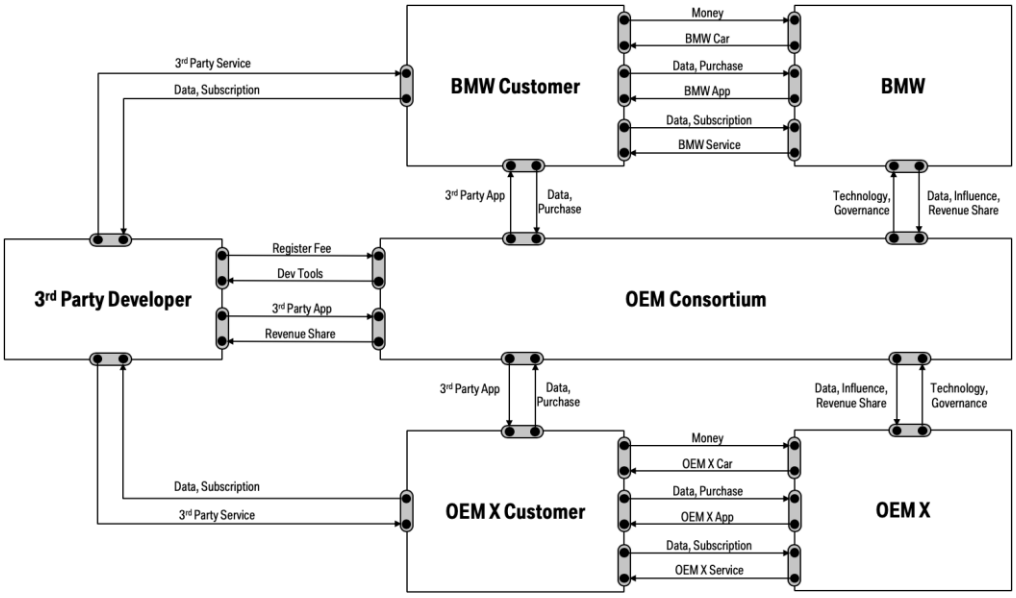
**Optional Question**
How do potential value networks of the three illustrated options look like? Include also other OEMs in your considerations.

**Proposed Solution**

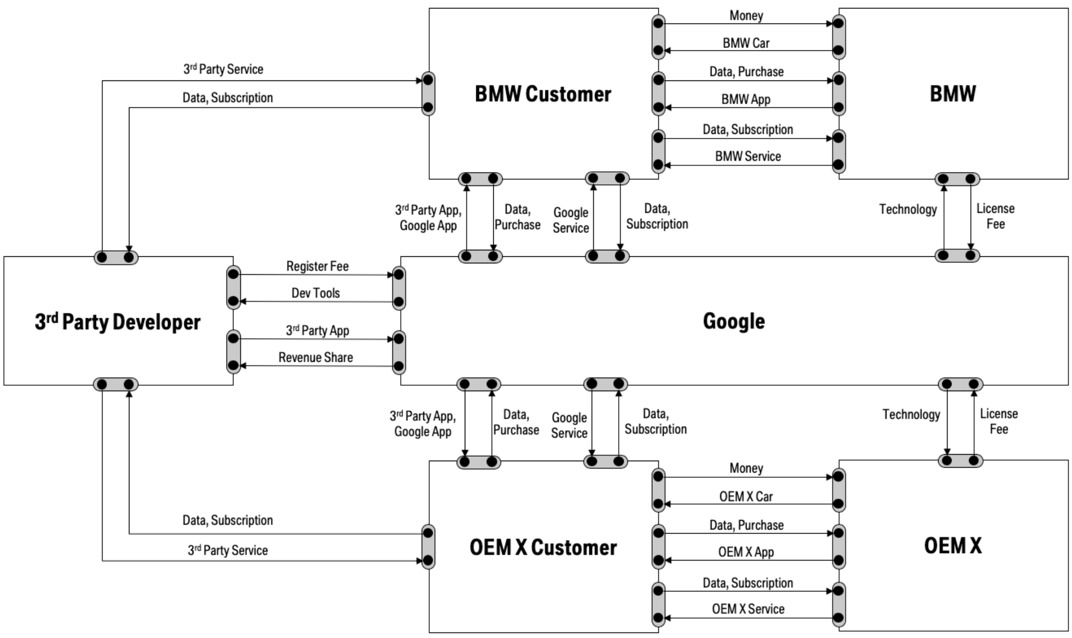
- *Value network for an own BMW platform ecosystem:* Basically, this option illustrates a typical two-sided marketplace. Third-party developers on one side develop apps that are provided to customers on the other side. Transactions between both sides are managed by BMW. Every other OEM with a similar approach requires the developer to create a specific app, which would be available for customers of this other manufacturer. In such a scenario, all OEMs would strive for the establishment of an own platform business.




- *Value network for a collaborative platform ecosystem:* The collaboration with other OEMs for the creation of a common digital platform ecosystem requires the organization in a common institution. Such a platform consortium would consist of all participating OEMs and would take the role of a platform owner. Hence, the consortium would be responsible for the definition of platform governance as well as the guidelines for the technical platform design. Furthermore, all transactions from between developers and customers would be managed by the consortium. The revenue streams would be distributed between all involved OEMs accordingly to previously defined revenue sharing routines. The platform business would be shared by all involved OEMs.




- *Value network for a platform ecosystem with Google:* The implementation of Android Automotive OS would put Google into the position of a platform owner for automotive apps. All decision on platform governance as well as platform design would be made by Google. Accordingly, Google would manage the complete platform business while the OEMs remain with a traditional pipeline business in form of selling cars.



# 4 References


Baldwin, C. Y., and Woodard, C. J. 2009. "The Architecture of Platforms: A Unified View," *Working Papers - Harvard Business School Division of Research, 1-3*).

Cennamo, C., Ozalp, H., and Kretschmer, T. 2018. "Platform Architecture and Quality Trade-Offs of Multihoming Complements," *Information Systems Research* (29:2), pp. 461–478.

Cusumano, M., Gawer, A., and Yoffie, D. 2019. *Business of Platforms: Strategy in the Age of Digital Competition, Innovation, and Power*. New York: HaperBusiness.

Dellermann, D., Jud, C., and Popp, K. M. 2016. "Why Don´T They Join? Analyzing the Nature and Consequences of Complementors´ Costs in Platform Ecosystems," in: *Thirty Seventh International Conference on Information Systems (ICIS)*. Dublin, Ireland.

Eaton, B., Elaluf-Calderwood, S., Sørensen, C., and Yoo, Y. 2015. "Distributed Tuning of Boundary Resources: The Case of Apple's Ios Service System " *MIS Quartely* (39:1), pp. 217-244.

Eisenmann, T., Parker, G., and Van Alstyne, M. 2006. "Strategies for Two-Sided Markets," *Harvard Business Review* (84:10), pp. 92-101

Eisenmann, T., Parker, G., and Van Alstyne, M. 2011. "Platform Envelopment," in *Platforms, Markets and Innovation,* A. Gawer (ed.). Cheltenham, UK: Edward Elgar, pp. 131-162.

Gawer, A. 2009. "Platforms, Markets and Innovation: An Introduction," in *Platforms, Markets and Innovation,* A. Gawer (ed.). Cheltenham, UK and Northampton, MA, US: Edward Elgar.

Hagiu, A., and Wright, J. 2015. "Multi-Sided Platforms," *International Journal of Industrial Organization* (43), pp. 162-174.

Hein, A., Schreieck, M., Riasanow, T., Setzke, D. S., Wiesche, M., Böhm, M., and Krcmar, H. 2019. "Digital Platform Ecosystems," *Electronic Markets* (In Press).

Henfridsson, O., and Ghazawneh, A. 2013. "Balancing Platform Control and External Contribution in Third-Party Development: The Boundary Resources Model," *Information Systems Journal* (23:2), pp. 173-192.

Karhu, K., Gustafsson, R., and Lyytinen, K. 2018. "Exploiting and Defending Open Digital Platforms with Boundary Resources: Android's Five Platform Forks," *Information Systems Research* (29:2), pp. 479–497.

Kude, T., Dibbern, J., and Heinzl, A. 2012. "Why Do Complementors Participate? An Analysis of Partnership Networks in the Enterprise Software Industry," *IEEE Transactions on Engineering Management* (59:2), pp. 250-265.

Ondrus, J., Gannamaneni, A., and Lyytinen, K. 2015. "The Impact of Openness on the Market Potential of Multi-Sided Platforms: A Case Study of Mobile Payment Platforms.," *Journal of Information Technology* (30:3), pp. 260-275.

Parker, G., Van Alstyne, M., and Choudary, S. P. 2016. *Platform Revolution: How Networked Markets Are Transforming the Economy and How to Make Them Work for You*. New York: WW Norton & Co.

Rausch, A., Müller, J., Niebuhr, D., Herold, S., and Goltz, U. 2012. "It Ecosystems: A New Paradigm for Engineering Complex Adaptive Software Systems," *6th IEEE International Conference on Digital Ecosystems and Technologies (DEST)*, Campione d'Italia, Italy.

Rochet, J.-C., and Tirole, J. 2003. "Platform Competition in Two-Sided Markets," *Journal of the european economic association* (1:4), pp. 990-1029.

Schilling, M. 2002. "Technology Success and Failure in Winner-Take-All Markets: The Impact of Learning Orientation, Timing, and Network Externalities," *Academy of Management Journal* (45:2), pp. 387-398.

Schreieck, M., Wiesche, M., and Krcmar, H. 2016. "Design and Governance of Platform Ecosystems-Key Concepts and Issues for Future Research," *Twenty-Fourth European Conference on Information Systems*, Istanbul, Turkey.

Stummer, C., Kundisch, D., and Decker, R. 2018. "Platform Launch Strategies," *Business & Information Systems Engineering* (60:2), pp. 167-173.

Tilson, D., Lyytinen, K., and Sørensen, C. 2010. "Research Commentary—Digital Infrastructures: The Missing Is Research Agenda," *Information Systems Research* (21:4), pp. 748-759.

Tiwana, A. 2014. *Platform Ecosystems: Aligning Architecture, Governance, and Strategy*. 225 Wyman Street, Waltham, MA 02451, USA: Elsevier Inc.

Yoo, Y., Henfridsson, O., and Lyytinen, K. 2010. "The New Organizing Logic of Digital Innovation: An Agenda for Information Systems Research," *Information Systems Research* (21:4), pp. 724-735.