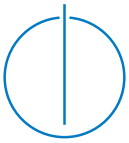
DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Dissertation

Distributed Ledger Technologies in the Automotive Value Chain

Daniel Simon Miehle



TECHNISCHE UNIVERSITÄT MÜNCHEN

DEPARTMENT OF INFORMATICS
CHAIR FOR APPLIED SOFTWARE ENGINEERING

Distributed Ledger Technologies in the Automotive Value Chain

DANIEL SIMON MIEHLE

Vollständiger Abdruck der von der Fakultät fur Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften (Dr. rer. nat.) genehmigten Dissertation.

Vorsitzender:	Prof. Dr. Florian Matthes
Prüfer der Dissertation:	1. Prof. Dr. Bernd Brügge
	2. Prof. Dr. Gilbert Fridgen

Die Dissertation wurde am 29.05.2020 bei der Technischen Universität München eingereicht und durch die Fakultat fur Informatik am 27.09.2020 angenommen.

Abstract

An automotive value chain is a complex network consisting of numerous participants, such as original equipment manufacturers, suppliers, and service providers that perform a set of activities to provide products and services to customers. The ongoing globalization, the growing number and variety of participants and interactions in supply chains, as well as the increasing vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication, require more efficient and secure data exchange between participants to improve collaboration in the automotive value chain.

Distributed ledger technologies (DLTs) have the potential to address these challenges and add new value. By combining peer-to-peer networks and cryptography, DLTs enable participants to exchange data directly and execute transactions securely and without the need for a trusted central party.

In this dissertation, we used a formative research approach to investigate the integration of DLTs into the automotive value chain to increase transparency and efficiency of cross-organizational business processes. The feasibility and value were demonstrated within five DLT applications covering purchasing, logistics, production, sales, and services. For each application, we analyzed the business requirements and designed, implemented, and evaluated prototypes in cooperation with industry partners.

The prototypes demonstrated efficiency-related benefits in the area of production and logistics and opportunities to create new business models in the area of sales and services. One of the applications is already in use in a global supply chain involving two plants of an OEM and three plants of a first-tier supplier. The network is being expanded to include more first-tier and additional second-tier suppliers. The dissertation concludes by summarizing the contributions and discussing future research.

Zusammenfassung

Eine automobile Wertschöpfungskette ist ein komplexes Netzwerk, das aus zahlreichen Teilnehmern besteht, wie z.B. Automobilherstellern, Zulieferern und Dienstleistern, die eine Reihe von Aktivitäten durchführen, um den Kunden Produkte und Dienstleistungen anzubieten. Die fortschreitende Globalisierung, die wachsende Zahl und Vielfalt der Teilnehmer und Interaktionen in den Lieferketten sowie die zunehmende Fahrzeug-zu-Fahrzeug- (V2V) und Fahrzeug-zu-Infrastruktur- (V2I) Kommunikation erfordern einen effizienteren und sichereren Datenaustausch zwischen den Teilnehmern, um die Zusammenarbeit in der automobilen Wertschöpfungskette zu verbessern.
Distributed-Ledger-Technologien (DLTs) haben das Potenzial, diesen Herausforderungen zu begegnen und neuen Nutzen zu schaffen. Durch die Kombination von Peer-to-Peer-Netzwerken und Kryptographie ermöglichen es DLTs den Teilnehmern, Transaktionen sicher auszuführen und Daten direkt auszutauschen, ohne dass eine zentrale Instanz erforderlich ist.
In dieser Dissertation haben wir mit einem formativen Forschungsansatz die Integration von DLTs in die automobile Wertschöpfungskette untersucht, um die Transparenz und Effizienz von Geschäftsprozessen zu erhöhen. Die Durchführbarkeit und der Nutzen wurden anhand von fünf DLT-Anwendungen in den Bereichen Einkauf, Logistik, Produktion, Vertrieb und Dienstleistungen demonstriert. Für jede Anwendung analysierten wir die Geschäftsanforderungen und entwarfen, implementierten und evaluierten Prototypen in Zusammenarbeit mit Industriepartnern.
Die Prototypen demonstrierten Effizienzvorteile im Bereich der Produktion und Logistik und Möglichkeiten zur Schaffung neuer Geschäftsmodelle im Bereich des Vertriebs und der Dienstleistungen. Eine der Anwendungen ist bereits in einer globalen Lieferkette im Einsatz, an der zwei Werke eines Automobilherstellers und drei Werke eines First-Tier-Lieferanten beteiligt sind. Die Dissertation schließt mit einer Zusammenfassung der Beiträge und einer Diskussion der zukünftigen Forschung.

Acknowledgments

First, I would like to thank Prof. Bernd Brügge for the continuous support of my dissertation, for his patience, motivation, and immense knowledge. Besides my advisor, I would like to thank my second supervisor Prof. Gilbert Fridgen for his encouragement and constructive feedback.

My sincere thanks go to my mentors Dr. Andre Luckow and Dr. Michael Ströbel, who offered me the opportunity to work in the BMW Group and supported me through various exciting projects. I also thank my colleagues in the BMW Group and in University for the discussions as well as all the co-authors of the publications that contributed to the research.

I also want to thank my girlfriend Daniela for her love and support. Last but not least, I would like to thank my parents, Andreas and Lydia, and my sister, Michelle, for supporting me throughout my life.

Publication Preface

The dissertation is based on the following five publications:

Publication [A]

Daniel Miehle, Dominic Henze, Andreas Seitz, Andre Luckow and Bernd Bruegge
PartChain: A Decentralized Traceability Application for Multi-Tier Supply Chain Networks in the Automotive Industry
2019 IEEE International Conference on Decentralized Applications and Infrastructures
DOI: 10.1109/DAPPCON.2019.00027

Publication [B]

Daniel Miehle, Matthias M. Meyer, Andre Luckow, Bernd Bruegge and Michael Essig
Toward a Decentralized Marketplace for Self-Maintaining Machines
2019 IEEE International Conference on Blockchain
DOI: 10.1109/Blockchain.2019.00066

Publication [C]

Gonzalo Munilla Garrido, Daniel Miehle, Andre Luckow and Florian Matthes
A Blockchain-based Flexibility Market Platform for EV Fleets
2020 Clemson University Power Systems Conference
DOI: 10.1109/PSC50246.2020.9131332

Publication [D]

Daniel Miehle, Andreas Pfurtscheller and Bernd Bruegge
Starling: A Blockchain-based System for Coordinated Obstacle Mapping in Dynamic Vehicular Environments
2020 Hawaii International Conference on System Sciences
DOI: 10.24251/HICSS.2020.493

Publication [E]

Andreas Seitz, Dominic Henze, Daniel Miehle, Bernd Bruegge, Jochen Nickles and Markus Sauer

Fog Computing as Enabler for Blockchain-Based IIoT App Marketplaces

2018 International Conference on Internet of Things: Systems, Management and Security

DOI: 10.1109/IoTSMS.2018.8554484

Contents

1 Introduction

„I've been at this 35 years, writing about the digital age. I've never seen a technology that I thought had greater potential for humanity." Don Tapscott [1]

1.1 Motivation

Automotive value chains are complex networks consisting of numerous participants, including original equipment manufacturers (OEMs), suppliers, distributors, dealers, service providers, and customers that are spread all over the world. A value chain refers to a set of activities carried out by one or more participants to add value to a product or service offered to manufactures and customers [2]. As shown in Figure 1.1, these activities can be divided into *upstream activities* for selling products or services from suppliers to OEMs and *downstream activities* for selling from OEMs to customers [3]–[5]. According to Porter [6], these activities include both primary activities, such as logistics, production, and sales, as well as supporting activities, such as technology development, procurement, and human resource management. Due to the large number of participants and the heterogeneity of activities, the management and orchestration of such a complex value chain are associated with various challenges.

Trends, such as globalization, sustainability, customization, outsourcing, and flexibility, have a significant impact on the complexity of automotive value chains [8]. Complexity drivers are the increasing number and variety of suppliers, parts, and interactions, non-synchronized decisions and actions, and incompatible IT systems, which lead to a lack of transparency, efficiency, and interoperability, especially in upstream activities [9]–[13]. In downstream activities, autonomous, connected, and electric vehicles increasingly communicate with vehicular and non-vehicular systems, which refers to vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication. Thereby, transactions and data are exchanged with other vehicles or infrastructure, such as charging stations, parking lots, or roads. The main challenges are secure communication (e.g., Sybil attacks) and privacy (e.g., identity disclosure) [14]–[19].

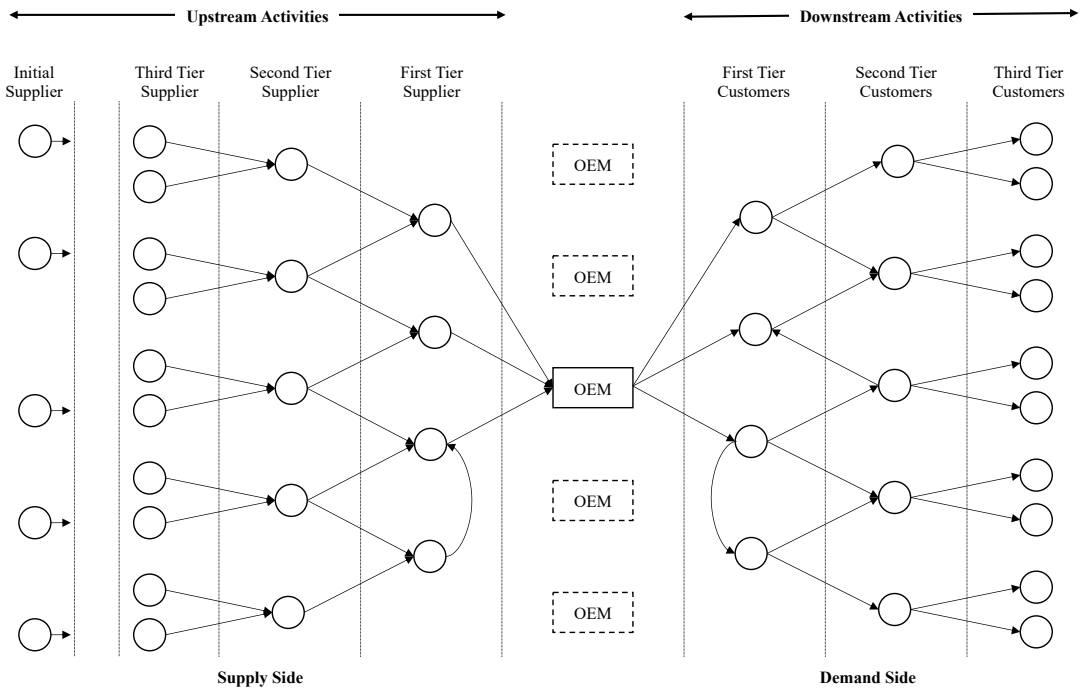


Figure 1.1: **Automotive value chain:** An automotive value chain can be divided into upstream and downstream activities for each OEM. The figure shows the relationships of activities of one specific OEM (represented as a solid rectangle). Adopted from Waters [7].

In this dissertation, we investigated the integration of distributed ledger technologys (DLTs) into the automotive value chain to enhance transparency and efficiency of cross-organizational processes. The feasibility and value are demonstrated within five DLT applications ranging from production and logistics to sales and services. We analyzed the business requirements for each application and designed, implemented, and evaluated prototypes for building cross-organizational platforms. Based on the prototypes, we derived the following core business requirements in the *automotive value chain*:

- **Open platform.** Systems should be designed as open platforms, that are not owned by a single authority. By distributing the control among participants, open platforms reduce barriers to adoption and foster cross-organizational collaboration and innovation (e.g., by sharing developing costs).

- **Traceability.** Participants must be able to record data and transactions concerning a digital or physical asset (e.g., software, vehicle), including the related meta-data across the value chain, to create transparency for all participants involved.
- **Auditability.** Selected data and transactions should be documented in a tamper-proof manner to comply with regulatory requirements and establish trust among the participants. The participants must be able to verify whether data and transactions have been altered or deleted.
- **Process automatization.** Systems should enable process automatization to increase the efficiency and ensure the integrity of cross-organizational processes.
- **Availability.** Systems must be available without any downtime in case they run in critical environments (e.g., traffic), where failures are unacceptable.
- **Financial transactions.** Participants must be able to execute financial transactions in near real-time with low transaction costs, enabling new payment models, such as consumption-based models.

Distributed ledger technologies DLTs allow participants to execute transactions and exchange data directly with each other without the need for a trusted central party [20]–[22]. By combining peer-to-peer networks, cryptography, and consensus mechanisms, DLTs create a transparent and tamper-resistant distributed ledger [22]. One type of DLT is the blockchain technology. Blockchain technology was developed by Satoshi Nakamoto in 2008 for the use in the cryptocurrency Bitcoin [23]. Initial blockchain-based systems focused on cryptocurrencies and financial applications, such as financial trade platforms [24], [25].
The introduction of Ethereum [26] in 2014 enabled the use of executable programs, so-called smart contracts. As a result, the spectrum of DLT applications has expanded from cryptocurrencies to general-purpose applications that can be used in various industries, such as the Internet of Things, supply chain management and purchasing, smart grid, consumer services, healthcare, public administration & governance, and utilities [27], [28]. Depending on the actual implementation, DLTs provide one or more of the following capabilities for applications in the *automotive value chain*.

- **Distributed Ledger.** A distributed ledger is a distributed data structure that stores data and transactions transparently, chronologically, and tamper-resistant

in a distributed network [25], [29]. Each participant of the network maintains a copy of the distributed ledger to ensure transparency and resiliency [30].

- **Cryptography.** Digital signatures, hash functions, and timestamps prevent data and transactions from tampering through encryption. Data and transactions recorded on the distributed ledger cannot be edited, deleted, or copied, creating unique digital records.
- **Consensus Algorithms:** Algorithms ensure the validity of data and transactions through consensus among network participants. Participants in the network collectively validate and approve transactions to the distributed ledger.
- **Smart Contracts:** Smart contracts automatically execute if pre-defined conditions are met enabling the enforcement of shared logic in cross-organizational processes and novel capabilities in products or services.

For each of the investigated applications, we linked the presented business requirements with DLT capabilities in the *automotive value chain*. Our research shows that DLT capabilities provide advantages for the automotive value chain by creating industry-wide platforms through decentralized governance and data management.

1.2 Research Approach

The research field of DLTs and related research topics, such as DLT infrastructures, applications, and ecosystems, are still relatively new. Therefore, we applied a formative research approach to explore the field as it evolves [31]. Formative approaches aim to shape a technology, system, or process through iterative application, evaluation, and improvement. Summative approaches, on the other hand, are not intended to improve a technology system, or process, but rather to evaluate the artifacts they create [32]. The formative approach we used consists of the following three phases: (i) system development, (ii) system validation, and the (iii) deduction of improvements and design recommendations for DLT infrastructures, applications, and ecosystems through observations, interviews, or computer-based simulations [31]. We have employed these three phases to the application areas along the automotive value chain, which we have investigated in cooperation with industry partners.

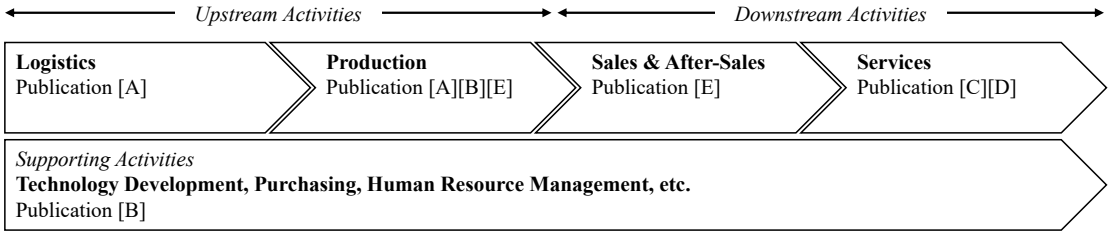


Figure 1.2: **Publication overview:** The publications are ordered along the value chain covering purchasing, logistics, production, sales, and services.

1.3 Outline

Chapter 2 provides an overview about the foundations of the dissertation. Section 2.1 and 2.2 give an introduction to client-server and peer-to-peer systems. Section 2.3 describes the term distributed ledger technology (DLT) more formally as it is used throughout the remainder of the dissertation.

Chapter 3 focuses on the main contribution, mainly the integration of DLTs into applications in the automotive value chain. A forecast summarizes the goal, approach, and results of each publication. The applications are ordered along the value chain of an OEM covering purchasing, logistics, production, sales, and services (see Figure 1.2):

Section 3.1 - Publication [A]: A DLT platform for part traceability in multi-stage supply chain networks in the logistics and production area.

Section 3.2 - Publication [B]: A DLT marketplace for self-maintaining machines enabling automated procurement processes in the purchasing and production area.

Section 3.3 - Publication [C]: A DLT marketplace for trading flexibility of electric vehicles (EVs) in the vehicle and customer service area.

Section 3.4 - Publication [D]: A DLT platform for the secure and verifiable exchange of vehicle and traffic data in the vehicle and customer service area.

Section 4.1 - Publication [E]: A DLT marketplace for industrial mobile applications in the production and after-sales area.

Chapter 5 summarizes the dissertation and the contributions and discusses further ideas for each DLT application.

2 Distributed Systems

In this chapter, we give an introduction to centralized, decentralized, and distributed systems. Furthermore, we present the essential architectures of distributed systems, such as client-server (Section 2.1) and peer-to-peer (Section 2.2) systems that lay the foundation for distributed ledger technologies (Section 2.3).

According to Baran [33], networks and computer systems can be designed centralized or distributed (see types (a) and (c) respectively, in Figure 2.1). They consist of nodes (e.g., clients, servers, peers, or sites) and links that connect the nodes. The difference between centralized and distributed systems is how the configuration of nodes and links is arranged, operations[1] are performed on data, and decisions are reached.

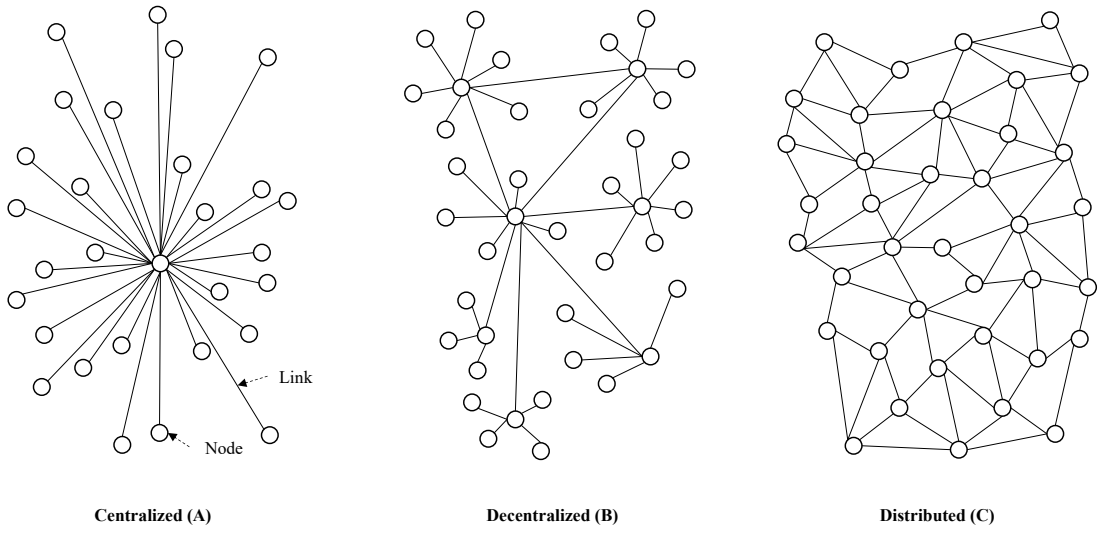


Figure 2.1: **Types of systems:** (a) Centralized. (b) Decentralized. (c) Distributed systems. Adopted from Baran [33].

[1]The four basic operations in computer programming are create, read, update, and delete (CRUD).

In a **central** or **star system,** data is stored in a central database, and operations are performed by a single node owned by a central trusted authority. Access to the database is typically restricted to authorized users. Centralized systems provide low fault tolerance and robustness due to their single point of failure (SPOF) [33]. In the case of a business monopoly, the central authority may charge high transaction costs [25].

In contrast, a **distributed** or **grid** or **mesh system** is a system whose components are distributed among multiple, networked nodes. These nodes communicate with each other and coordinate their actions by sending messages [34]. Distributed systems operate without a central trusted authority. Instead, they use algorithms to maintain data consistency at all nodes. Since data is distributed and replicated across multiple nodes, data is visible to all nodes, and the system continues to function even if one node fails.

In practice, **decentralized systems** are used more frequently and can be considered as a mixture of star and mesh systems [33] and, therefore, as a subset of distributed systems (see type (b), in Figure 2.5).

In addition to the degree of *distribution* (i.e., centralized to full distributed), distributed system architectures differ in terms of *autonomy* (i.e., zero to full autonomy) and *heterogeneity* (i.e., zero to full autonomy) [35]. Autonomy represents the property of being independent of a central trusted authority and having self-control, such as the local execution of operations on data [36]. Heterogeneity describes the use of different components and resources in distributed systems, such as hardware and data models [37]. Reasons for using distributed systems can be grouped into *functional* (e.g., scalability, performance, security, fault-resilience, interoperability), *economic* (e.g., cost of ownership, transparency), and *psychological* (e.g., self-organization, self-control) [36], [38].

Figure 2.2 shows a taxonomy of computer system architectures that include centralized and distributed systems at the first level. In the following sections, we concentrate on distributed systems and give an introduction to client-server (Section 2.1) and peer-to-peer (Section 2.2) systems, and DLTs, which are the focus of this dissertation.

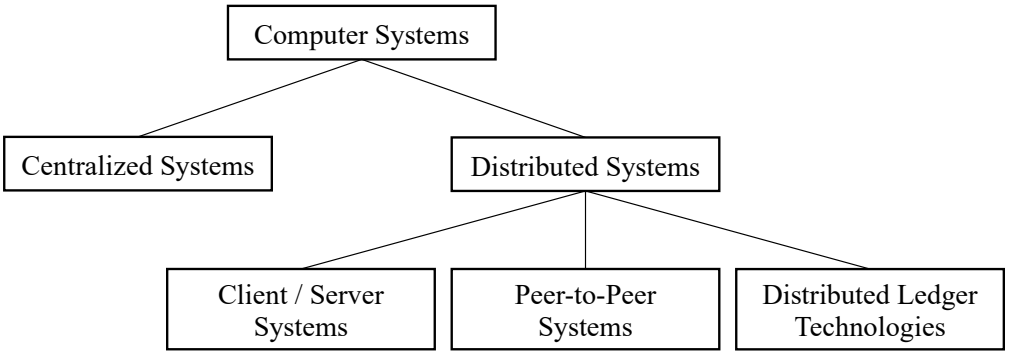


Figure 2.2: **Taxonomy of computer system architectures:** Distributed systems include client-server systems, peer-to-peer systems, and distributed ledger technologies. Adopted and extended from Milojicic et al. [36].

2.1 Client-Server Systems

One of the first mentions of client-server systems described the separation of data from function in distributed networks by dividing the responsibilities between servers and clients [39]. According to Schollmeier [40], a client-server system is a "distributed network which consists of one higher performance system, the Server, and several mostly lower performance systems, the Clients".
Servers share their resources with one or more clients to provide data or services. Clients request data or the execution of services from one or more servers. In a *multiple client/single server* architecture, there is only one server that is accessed by multiple clients. In contrast, in a *multiple client/multiple server* architecture, clients manage their connections to multiple servers themself or are coupled to a dedicated server that interacts with the other servers over the network [41]. The former approach is comparable to a centralized system, whereas the latter represents a distributed system or a distributed database system in case multiple database servers are connected.

2.2 Peer-to-Peer Systems

Peer-to-peer (P2P) systems refer to self-organizing systems of equal, autonomous nodes (i.e., peers) that use heterogeneous distributed resources connected by a network to execute a function without a central trusted party [36], [42]. In comparison to client-

server systems, P2P systems are loosely-coupled, indicating that peers can join and leave the network at any time (cf. open-world assumption). Each peer acts as both client and server [43]. The level of self-organization is defined by the degree of coupling, which differs between *structured* (limited level of self-organization) and *unstructured* (high level of self-organization) P2P systems. While *unstructured* P2P systems are characterized by an ad hoc nature of the links between peers, *structured* P2P systems use a distributed hash table (DHT) to locate peers and resources. Applications of P2P systems include distributed computing, file sharing, collaboration, and platforms [36].

2.2.1 Unstructured P2P Systems

Following Milojicic et al. [36], unstructured P2P systems can be categorized into *centralized* and *pure* P2P systems.

Centralized P2P systems employ a centralized registry to provide metadata about peers and resources. Peers use the metadata to locate resources in the system, while the resources remain with the individual peers. However, these centralized registries represent a SPOF within the system [41], [44]. Napster was one of the first centralized P2P file-sharing systems for music files, serving over 20 million users by the early 2000s [41]. In 2001, a copyright infringement lawsuit against Napster led to the shutdown of the centralized registry server and, thus, to the closing of the file-sharing service [45].

In **pure P2P systems**, peers have equal responsibilities, without using a centralized registry. To locate resources on the network, peers send requests to known peers by maintaining a list of peer addresses or make new connections to unknown or newly joined peers using techniques, such as flooding and random walk [46]. This is a working, but an inefficient approach to locate resources on the system, as a large number of peers could be traversed during discovery, causing delays and additional network traffic. Since there is no SPOF in the system, pure P2P systems provide the described advantages of P2P systems, in particular reliability, robustness, and scalability [44].

2.2.2 Structured P2P Systems

The limitations of *unstructured* P2P systems are addressed by *structured* P2P systems, in which data placement is controlled by a distributed hash table (DHT). A DHT provides a lookup service where resources are indexed using key-value pairs [47], [48]. Any

peer can use a key to retrieve the corresponding value efficiently. This approach solves the low-efficiency problem of unstructured P2P systems and ensures that a lookup is successful if the data is present in the system.

2.3 Distributed Ledger Technology

„We should think about the blockchain as another class of thing like the Internet."
Melanie Swan [49]

A DLT is a distributed database, which is replicated and synchronized across a P2P network, governed by a consensus mechanisms, and secured by cryptography [25], [29]. DLTs extend the concept of P2P systems by incorporating its characteristics, such as data immutability, integrity, fair access, transparency, non-repudiation of transactions, and cryptocurrencies [30], [50]. DLT systems differ in terms of data structures (Section 2.3.2), consensus algorithms (Section 2.3.3), permissions to access and perform operations on the network (Section 2.3.4), and the integration of smart contracts (Section 2.3.5), which are explained in the following sections [27].

2.3.1 Cryptographic Foundations

The section gives a brief overview of the cryptographic foundations, mainly hash functions and asymmetric cryptography, which provide the basis for DLTs.

Hash Functions

A hash function is a cryptographic algorithm that generates a single value with a fixed length, called a hash or hash value, out of arbitrarily large data input. The hash value is a sequence of letters and numbers, creating a unique digital fingerprint for verifying the integrity of transactions [51]. The key properties of a hash function are:

- A hash function is *deterministic*, meaning that it must always generate the same hash value for a given data input.
- A hash function is *collision resistance*, meaning that it is not possible to find a second data input that generates the same hash value with reasonable effort.
- A hash function ensures *information hiding*, meaning that for a given output of a hash function, there is no way to retrieve the input value.

The most commonly used hash function in the context of DLTs is the secure hash algorithm SHA-256. The hash value is a hexadecimal sequence with a length of 64 digits and 256 bits. To demonstrate the functionality of secure hash algorithm SHA-256, we use the two strings `"distributed ledger technology"` and `"distributed ledger technology."` as an example. We vary the two strings by adding a `"."` to the end of the first string.

Input data: "distributed ledger technology"
`"2a96ccbeaeee5fe993dce9c4b6c8922687f8df23f0d57526381353b102c4ccd3"`.

compared to

Input data: "distributed ledger technology."
`"8ba62bfd23e0fe64d50ea0848df3a5c7cfdee907f656eda173ee5ab2d7fbea90"`

As one can see, two utterly different hash values have been created. Thus, the hash value can be considered a unique digital fingerprint for a data input.

Asymmetric Cryptography

Asymmetric cryptography or public key infrastructure technology has become the focus of modern security mechanisms [52]. In contrast to symmetric cryptography that uses the same key to encrypt or decrypt data, asymmetric cryptography uses a pair of *private* and *public* key. The *private* key is only known to the owner to sign or encrypt data, whereas the *public* key may be known by other participants to verify or decrypt data [29]. In the context of DLTs, the signing and verification mechanism is used to authorize and verify transactions in the distributed network.

2.3.2 Data Structures

A data structure describes how transactions are stored in a distributed ledger and can be a blockchain or a directed acyclic graph [25], [29], [30]. A transaction can contain any type of data, ranging from cryptographically signed financial transactions to hashes of physical and digital assets, and executable programs (i.e., smart contracts) [25], [30]. In the following, we introduce the two most common data structures, namely blockchains

and directed acyclic graphs (DAGs).

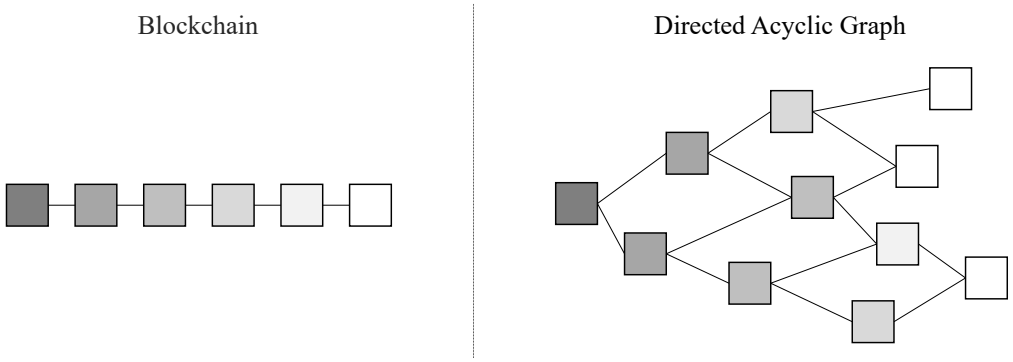


Figure 2.3: **DLT data structures:** Comparison of blockchain and directed acyclic data structures. A blockchain is an append-only linked list of blocks consisting of transactions, while a directed acyclic graph consists of individual transactions connected by the edges of the graph.

The **blockchain** data structure was developed by Satoshi Nakamoto [23] for the use in the cryptocurrency Bitcoin in 2008. As shown in Figure 2.3, a blockchain maintains an append-only linked list of blocks [23], [25]. Blocks are added to the blockchain in a chronological order and are secured using cryptography [29]. The first block of a blockchain is called the genesis block, which has no previous block.
As depicted in Figure 2.4, each block consists of the *block header* and the *block body*. The *block header* includes the version, the Merkle root hash of the current block, the hash value of the previous block, the timestamp, the difficulty target, and the number only used once (NONCE). Blocks are connected by including the root hash of the current block and the hash value of the previous block in the block hash [29]. The *block body* contains the transaction data that is represented as a Merkle tree[2]. To change only one transaction within a block, all subsequent blocks must be changed and re-hashed.
Limitations of the blockchain data structure include scalability, transaction throughput, and high transaction fees. To solve these limitations, DLTs investigate the use of directed acyclic graph (DAG) as the underlying data structure.

[2]A Merkle tree [53] or hash tree is a tree of hash value from data, such as transactions, and allows the efficient verification of large data structures.

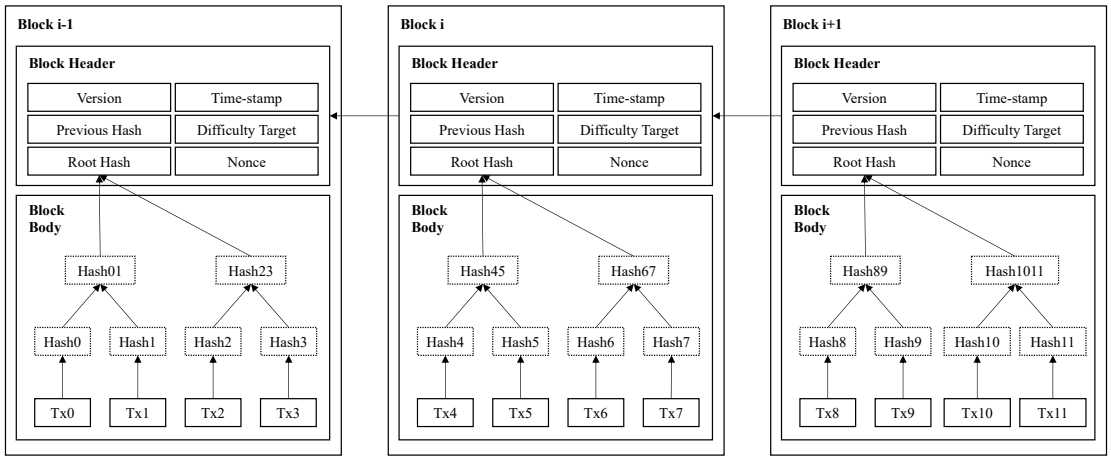


Figure 2.4: **Blockchain data structure:** A blockchain consists of a linear growing list of blocks holding a complete list of transactions in the form of a Merkle tree. Adopted from Zheng et al. [29].

In a **directed acyclic graph (DAG)**, transactions are stored in the nodes, where each node holds a single transaction [54]. A DAG is a graph consisting of nodes and directed edges, in which there are no cycles. Thus, they can depict the sequence of transactions and prevent transactions from referring to themselves. Since nodes are obligated to order and validate their transactions, there is no need to select a leader [55]. DAGs promise to facilitate high scalability of transactions [55], and thus, the combination of DLTs and Internet of Things (IoT) devices, which require high transaction throughput and low power consumption [56]. Examples include Byteball [57], Nano [58], and the cryptocurrency IOTA [59], which uses the pay-it-forward principle, where each new transaction must confirm two existing transactions.

2.3.3 Consensus Mechanisms

A consensus mechanism refers to a mechanism by which a distributed system reaches consensus to make decisions without a central trusted authority [60]. Furthermore, it provides reliable communication by addressing the Byzantine Generals Problem. The Byzantine Generals problem [61] refers to the problem of how nodes in a distributed system can be confident that the information they receive is valid.

Properties	PoW	Pos	PBFT
Validator identity	public	public	private
Energy saving	no	partial	yes
Tolerated power of adversary	< 25% computing power	< 51% stake	< 33.3% faulty peers
Example	Bitcoin [23]	PeerCoin [62]	Hyperledger Fabric [63]

Table 2.1: **Consensus mechanisms:** Comparison of the PoW, PoS, and PBFT consensus mechanisms. Adopted from Zheng et al. [29].

First, the consensus mechanism provides rules for selecting a node as a validator or leader that adds new blocks or transactions to the distributed ledger, thus, ensuring data consistency. Secondly, the consensus mechanism protects the system against manipulation, for example, by preventing double-spending and Sybil attacks. While double-spending assures that a transaction can only be used once, Sybil attacks describe the forgery of multiple peer identities to compromise data on the distributed ledger [30].
Various consensus mechanisms have been proposed to address the different needs of DLT applications. In the following, we introduce and compare the most widely used consensus mechanisms (see Table 2.1), namely Proof of Work (PoW), Proof of Stake (PoS), and Practical Byzantine Fault Tolerance (PBFT).

Proof of Work (PoW) was introduced as the underlying consensus mechanism of the cryptocurrency Bitcoin [23]. In PoW, nodes compete to solve a complex cryptographic puzzle that requires computational power to select a node as a validator. The process of solving this cryptographic puzzle is called *mining*.
In the mining process, pending transactions are bundled into a block and broadcasted to the mining nodes in the network. The mining nodes then try to validate the block by finding a hash value for the block header (see Section 2.3.2) that is less or equal a target value by finding an arbitrary number for the NONCE using the brute force method. The first mining node that validates the block receives both a block reward and the transaction fees in the block [25], [64]. The validated block is added to the blockchain. The rewards provide an incentive system for the mining nodes to act honestly in the distributed network.
PoW implements the rule of the longest chain, where all nodes trust the longest chain because most of the computing power was invested in the creation of this chain [65].

To manipulate the PoW consensus mechanism of Bitcoin [23], it was assumed that at least 50% of the computing power is required, which is known as a 51% attack [66]. However, it has been shown that due to selfish mining strategies, only 25% is actually needed to forge transactions [67]. PoW has some drawbacks, such as low transaction rate and high energy consumption.

Proof of Stake (PoS) uses a weighted random selection process to select a node as a validator. The weight of each node is defined by factors, such as the stake (i.e., number of coins), the average age of these coins (i.e., coinage), and randomness. According to Bach et al. [68], the generated hash value must satisfy the following condition:

$$hashvalue < stake \cdot coingage \cdot target$$

Nodes with a higher *stake* or *coinage* have a broader search corridor and can, therefore, find the hash value easier than nodes with less stake. It is assumed that nodes with higher stake have a higher interest in keeping the distributed ledger free from manipulations. PeerCoin [62] was the first cryptocurrency that used PoS in 2012. While PoW systems use the chain with the highest computing power as the main chain, PoS systems use the chain with the highest coin consumption [62], [68]. Furthermore, PoS does not require any time- and energy-intensive mining. With the PoS consensus mechanisms, an adversary would need to control at least 51% of the stake in the network to forge transactions [68].

In contrast to public networks, where validators are anonymous and driven by an incentive system, private networks assume that validators are known and trusted. Thus, more efficient consensus mechanisms, such as the PBFT [69], can be applied to address challenges, such as high energy consumption, low scalability, and data privacy.

Practical Byzantine Fault Tolerance (PBFT) refers to a state machine that is replicated across different peers [70]. The PBFT consensus mechanism requires that the distributed network consists of `n = 3f+1` peers to be able to tolerate `f` faulty peers [71]–[73]. It uses a round-robin algorithm to select a peer as a leader, which is responsible for validating transactions and broadcasting them to the other peers. Once a client sends a transaction to the leader, a process is executed that consists of the following three phases: pre-prepared, prepared, and commit [29].

First, the leader creates a block containing the pending transactions and broadcasts the candidate block to other peers (pre-prepared phase). Next, the other peers will verify the block, and if it succeeds, they will broadcast prepare messages to the other peers (prepared phase). Peers do nothing if the verification fails. Finally, when peers receive prepare messages from 2/3 of the other peers, each peer then broadcasts a commit message (commit phase). As soon as the leader receives commit messages from 2/3 of the peers, the block is added to the distributed ledger, and the client sees the result. Since PBFT is quite communication-intensive, it can lead to scaling problems if a certain number of nodes are present in the network.

2.3.4 Permissions

In literature, various permission types for DLT infrastructures are proposed (see [25], [30], [66]) and can generally be classified into the following two areas:

- *Permissions for joining the network* addressing the anonymity level of a validator (i.e., public and private networks) and the ownership of the DLT network.
- *Permissions to execute operations* such as writing, reading, or committing transactions on the distributed ledger addressing the trust level to a validator (i.e., permissionless and permissioned networks).

By coupling these two permission types, we obtain the widely used DLT network types, namely public-permissionless, public-permissioned, private-permissionless, and private-permissioned networks as shown in Figure 2.5.

In **public-permissionless** networks, anyone can join the network, read, write, and commit transactions. These networks allow anyone to create a public/private key pair to send transactions to the distributed ledger providing high anonymity and transparency in trustless environments. Examples include cryptocurrencies, such as Bitcoin [23], Ethereum [26], Zerocash [74], and Monero [75], which mainly use the Proof of Work consensus mechanism [76]. In **public-permissioned** networks, anyone can join the network and read transactions, but only authorized users can write and commit transactions. These networks typically use the Proof of Stake consensus mechanism providing high anonymity and moderate scalability. Examples include Ripple [77], EOS [78], Sovrin [79], and Ethereum [80] after the introduction of the Casper protocol.

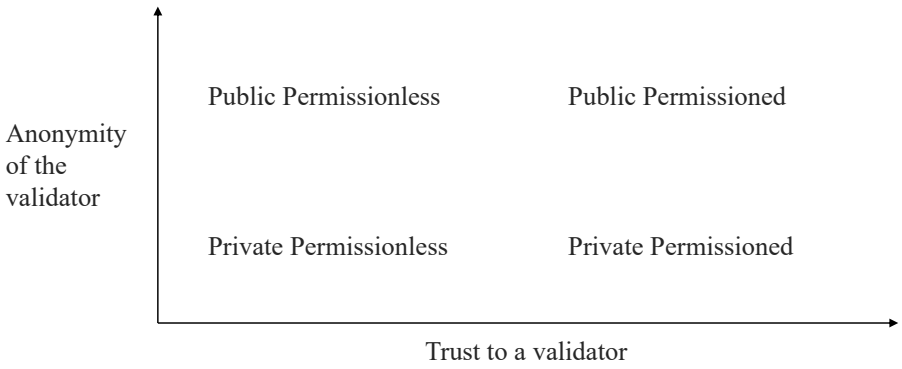


Figure 2.5: **Types of DLT networks:** Classification of DLT networks according to the degree of anonymity of a validator and the trust to a validator.

In **private-permissionless** networks, only authorized users can join the network, read, write, and commit transactions. In **private-permissioned** networks, additional rights are required to write and commit transactions. Examples include Multichain [81] and Hyperledger Fabric [63].

2.3.5 Smart Contracts

A smart contract refers to a computer program that is stored, verified, and executed on a distributed ledger to automatically enforce the terms of an agreement between untrusted parties without the involvement of a trusted third party [82]–[84]. The concept of smart contracts was first described in 1997 by Nick Szabo [85] as a "computerized transaction protocol that executes the terms of a contract." However, the use of smart contracts only became possible with the introduction of Ethereum [26] in 2015.

As shown in Figure 2.6, a smart contract has a *state* and *executable code* that are stored transparently on the distributed ledger [86]. The *state* of a smart contract includes an account balance and a private storage, which are updated each time the smart contract is invoked [82]. To invoke a smart contract, a user must send a transaction to the smart contract's address. Depending on the content of the transaction, smart contracts can read/write data in the private storage, store value in the account balance, or send/receive transactions from users or other contracts [82]. A smart contract is deterministic, which means that the same input will always produce the same output.

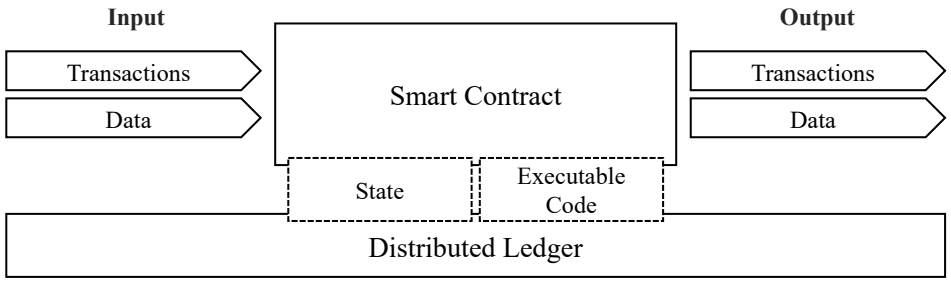


Figure 2.6: **Smart contract:** A smart contract is a computer program that executes the terms of an agreement once the specified conditions are met.

For the execution of business logic, smart contracts may need to access data from external sources (e.g., sensor data), known as oracles. Oracles can be classified depending on the *source of data* (i.e., software, hardware, human), the *direction of data* (i.e., inbound or outbound), and the *degree of trust* (i.e., centralized or decentralized) [87]. In contrast to centralized oracles controlled by a single entity, decentralized or consensus-based oracles request data from multiple oracles to determine the validity and accuracy of data, increasing the reliability, security, and trust of external data sources [87].

Smart contracts can be applied to share data (e.g., distributed file storage), establish agreements (e.g., mortgage payment), manage identities (e.g., IoT devices), or define shared business logic (e.g., supply chain management) [27], [86], [88]. The advantages of smart contracts include the reduction of errors, fraud, and transaction fees by eliminating the need for intermediaries to verify and execute the terms of an agreement. However, smart contracts face various challenges, such as codifying (e.g., writing and testing smart contracts), security (e.g., exploiting vulnerabilities to initiate an attack), privacy (e.g., lack of transactional privacy), and performance (e.g., sequential execution of smart contracts) issues [82], [89].

3 First and lead author publications

In this chapter, we present the first and lead author publications the contribution of this dissertation is based on. Each publication is summarized before the reprint of a publication. Reprints are included in case the paper has already been published, or the reprint was granted. Each summary covers the goal, the approach, and the results of the publication.

Publication [A]
PartChain: A Decentralized Traceability Application for Multi-Tier Supply Chain Networks in the Automotive Industry
Daniel Miehle, Dominic Henze, Andreas Seitz, Andre Luckow and Bernd Bruegge

Publication [B]
Toward a Decentralized Marketplace for Self-Maintaining Machines
Daniel Miehle, Matthias M. Meyer, Andre Luckow, Bernd Bruegge and Michael Essig

Publication [C]
A Blockchain-based Flexibility Market Platform for EV Fleets
Gonzalo Munilla Garrido, Daniel Miehle, Andre Luckow and Florian Matthes

Publication [D]
Starling: A Blockchain-based System for Coordinated Obstacle Mapping in Dynamic Vehicular Environments[1]
Daniel Miehle, Andreas Pfurtscheller and Bernd Bruegge

[1] Best paper nomination

3.1 PartChain: A Decentralized Traceability Application for Multi-Tier Supply Chain Networks in the Automotive Industry

This publication (Publication [A]) describes a novel DLT application for the tracking and tracing of parts in multi-stage supply chains.

Conference: 2019 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON)
Number of Pages: 6
Review: Peer Reviewed (3 Reviewers)

The author of this dissertation contributed substantially to this publication, in particular concerning the development of the idea and approach, the conduction of the literature review, the elicitation of the business requirements, the design of the system architecture, the design and conduction of the evaluations, and writing the publication.

Goal

Due to manual and paper-based processes and incompatible IT systems, only limited or inconsistent information on the origin and transport history of vehicles and parts is available. Thus, multi-stage supply chains lack transparency, efficiency, and responsiveness. The goal of this DLT application was to create an open platform to ensure the authenticity and end-to-end traceability of vehicles and parts in multi-stage supply chain networks in the automotive industry.

Approach

We implemented a DLT application prototype that enables OEMs and suppliers to register and transfer physical parts across organizational boundaries. Each physical part is represented through a unique digital identity on the distributed ledger enabling the verification of the authenticity of parts. Smart contracts automate ownership transfers of parts in logistics processes if the terms of the order or delivery agreements are fulfilled. All events (e.g., origin and location of each part) are recorded on the distributed ledger,

thus, providing traceability and auditability. Besides, we designed the PartChain user interface, which works as the trusted bridge between the physical and digital worlds.

Results

We conducted a case study with our industry partner to demonstrate the feasibility and value of our DLT application. In the case of quality and safety issues, the PartChain system supports OEMs and suppliers to detect defective parts and affected vehicles early, which facilitates timely and efficient recalls. Advantages include reducing the number of blocked or recalled vehicles and the duration of search operations as well as the detection of counterfeit parts, which reduces financial losses and increases vehicle safety. Moreover, the open platform approach provides a low entry barrier, which is essential for small- and medium-sized companies.

IEEE Reuse Requirements

With regard to the IEEE reuse requirements for dissertations, the accepted and not the published version of the publication is used in the following.

PartChain: A Decentralized Traceability Application for Multi-Tier Supply Chain Networks in the Automotive Industry

Daniel Miehle[1], Dominic Henze[2], Andreas Seitz[2], Andre Luckow[1], Bernd Bruegge[2]
[1]BMW Group, Group IT, Munich, Germany
Email: {daniel.miehle, andre.luckow}@bmw.de
[2]Chair for Applied Software Engineering, Technical University of Munich, Munich, Germany
Email: {seitz, henzed, bruegge}@in.tum.de

***Abstract*—Supply chain networks in the automotive industry deal with numerous participants and parts on multiple tiers that are spread across several geographical locations. Due to their inherent complexity and high focus on data protection in a competitive original equipment manufacturer market, supply chains often lack transparency and responsiveness. However, legal authorities such as the Federal Motor Transport Authority of Germany or the National Highway Traffic Safety Administration of USA require end-to-end traceability of parts. This is particularly important for the identification of defective parts in recalls when multiple suppliers are involved. Affected parts must be promptly identified among several million vehicles. Manual investigation of the same results in high costs and risks. Another challenge is the detection of counterfeit parts, especially in after-sales markets. In this paper, we propose a decentralized supply chain traceability application called PartChain that enables the creation, monitoring, and sharing of a unique digital representation of a physical part across a supply chain network using blockchain technology. We conducted a case study with our industry partner to demonstrate the feasibility and potential of our concept. Here, physical parts were tokenized on a permissioned blockchain using our PartChain mobile app, enabling the storage of process data and the ownership transfer of parts in a reliable and immutable manner.**

***Index Terms*—Blockchain Technology, Smart Contract, Supply Chain Management, Traceability, Decentralized Application**

I. Introduction

Car manufacturers, commonly referred to as original equipment manufacturers (OEMs) act as integrators because they do not manufacture a majority of the parts they use in-house. They depend on numerous suppliers providing parts, which are delivered in supply chain networks. Supply chain networks are complex ecosystems consisting of multiple participants such as OEMs, suppliers, and logistics service providers (LSPs) exchanging material, information, and money. According to their rank in the supply chain networks, suppliers are called "tier-n" suppliers. A tier-1 supplier is a first rank supplier, who incorporates parts from a tier-2 supplier and so on. Each supply chain participant uses a local database representation and a company-specific data model leading to unavailable and potentially inconsistent information on the event history of a part. Therefore, in recalls, OEMs can identify which cars are affected. However, the process is cumbersome and often requires significant human intervention due to no common master data, compatibility issues, and missing standards. Thus, detection of defective and counterfeit parts is inefficient, expensive, and error-prone.

Blockchain technology records and shares data using a distributed ledger [1], which provides a single source of truth improving the speed and responsiveness of supply chain networks. A digital twin is a digital representation of a physical object [2], which can be tokenized and used to run business logic in smart contracts on a blockchain. PartChain is a decentralized traceability application for the creation and sharing of unique digital twins of component parts using blockchain technology. Furthermore, we developed the PartChain mobile app, which aims to address the inherent problem of an interface between the digital and physical world. We demonstrate the feasibility and potential of blockchain in a case study that involves tracking four parts across a supply chain consisting of one OEM, three suppliers, and one LSP.

This paper is structured as follows. In Section II we provide an overview of the foundations on which the PartChain concept is based and related works. Section III discusses our PartChain concept including a detailed requirement analysis. Section IV presents a description of the system design, the prototype, and our findings from an evaluation workshop with experts from the industry. The summary of the paper and its contributions are presented in Section V.

II. Foundations

This section discusses the foundations on which our PartChain concept is based and a review of related works.

A. Distributed Ledger Technology

Distributed Ledger Technologies (DLTs) combine technologies such as peer-to-peer networks, cryptography, and consensus algorithms [3]. A DLT is a distributed database, which is replicated and synchronized across a decentralized peer-to-peer network using consensus algorithms. The distributed ledger stores data in an immutable way using cryptography. DLTs differ in terms of data structure, permissions to access the network and to initiate or mine transactions, consensus

algorithms, and smart contract integration [3]. One type of DLT is blockchain technology. Blockchain technology was developed by Satoshi Nakamoto in 2008 for the implementation of the cryptocurrency Bitcoin [4]. Thus, enabling participants to exchange assets in a secure, trusted, and tamper-proof way without the need of a central authority. A blockchain maintains a linear growing chain of blocks that are secured using cryptography [5]. Blocks are added to the blockchain in a chronological order containing a hash of the previous block, a timestamp, and a list of transactions [4]. Data and transactions are processed and synchronized by the network nodes using consensus algorithms. This keeps the nodes of the blockchain consistent by providing a single source of truth [1].

Blockchain networks can either be permissionless or permissioned. Permissions in this context can either be permissions to access the network or to execute operation within the network. In contrast to public blockchains, where anyone can access the network, private blockchains require one or more participants to grant access rights for each new node. In permissionless blockchains, every participant can initiate transactions, mine, and create smart contracts, whereas in permissioned blockchains not all participants are allowed to execute all operations. Moreover, in permissioned blockchains, participants can have permissions to manage the network, e.g., to grant or invoke access rights.

B. Smart Contracts

The concept of self-executing smart contracts was first introduced by Nick Szabo [6] in 1997 as a "computerized transaction protocol that executes the terms of a contract". Following Alharby and van Moorsel [7], a smart contract in the blockchain technology context is an "executable code that runs on top of the blockchain to facilitate, execute, and enforce an agreement between untrusted parties without the involvement of a trusted third party". Ethereum was the first blockchain framework that enabled the integration of smart contracts [8]. Benefits of smart contracts include the reduction of errors, fraud, time and costs of verification, and automatic execution of processes [7], [9], [10].

C. Digital Twin

The concept of the digital twin was proposed by Grieves [11] in 2003 in the context of product life cycle management. A digital twin can be described as a digital representation of a physical object across its lifecycle using real-time data [2], [12]–[14]. A digital twin comprises of the physical object in the physical world, the digital object in the digital world, and the connection of data that ties the two worlds together.

D. Related Works

In literature, other blockchain-based traceability systems focusing on different aspects in supply chain networks have been proposed. For instance, Chen et al. [15] proposed a theoretical framework for a blockchain-based quality management of supply chains addressing the challenge of lack of trust. Nakasumi [16] presented a schema for information sharing in supply chain management based on blockchain technology addressing the problems of double marginalization and information asymmetry. Tian [17] introduced a decentralized traceability system for food safety and demonstrated its capabilities using a demo scenario. Toyoda et al. [18] introduced an anti-counterfeit product ownership management system for anti-counterfeit and implemented a proof-of-concept of the system using a decentralized application based on Ethereum.

Moreover, the use of blockchain technology within the domain of digital twins is becoming popular in recent time. For instance, Kennedy et al. proposed an anti-counterfeit method based on embedded nano materials and blockchain technology for 3D-printed parts [19]. Further, Heber et al. proposed the use of blockchain technology in ensuring the traceability of software configurations in vehicles [20].

However, the proposed solutions often lack concrete implementation, evaluation, and hardly investigated the use of private permissioned blockchain networks. In this paper, we provide a detailed requirement analysis and implementation of the PartChain system using a private permissioned blockchain network. We introduce the PartChain mobile app as an interface for connecting the physical and digital world. Furthermore, we evaluate the system design in conjunction with supply chain experts from the automotive industry.

III. PartChain

In this section, we present the idea of PartChain, a supply chain traceability system, for supply chain networks.

A. Vision

The vision of the PartChain system is to enable participants of supply chain networks to create and transfer a unique digital twin of a physical part using a mobile device. OEMs and suppliers can query parts and trace the complete event history of the parts. In case a delivery contains defective parts, OEMs and suppliers are able to identify affected parts and vehicles promptly via the PartChain system, thus, speeding up the clarification process. Furthermore, participants can validate the authenticity of a part by checking if the part was manufactured and delivered by registered suppliers and LSPs on the PartChain network. In case an LSP tries to insert an unregistered part (e.g., a counterfeit part) into the supply chain network, the handover of the unregistered part is rejected and made visible to all participants. When a LSP picks up or an OEM receives a delivery, the PartChain system checks whether the conditions specified in the order or delivery contract are fulfilled. Suppose the conditions are fulfilled, the ownership transfer of a part is automatically executed and recorded on the PartChain system. This eliminates the need for manual signing. The corresponding inventories of the local enterprise resource planning systems are simultaneously updated.

B. Requirements and Approach

The following functional and non-functional requirements must be met to successfully implement a traceability system for supply chain networks:

(1) Provenance and Authenticity of Parts: OEMs and suppliers must be able to record information about the parts they manufactured on the PartChain system.
(2) Tracking and Tracing of Parts: Participants must be able to record all events concerning a part. Each record must include the current location and a timestamp.
(3) Digitalization of Documents: The handover of a part needs to be validated and recorded in the PartChain system eliminating manual signing and paperwork.
(4) Access Control: The PartChain system should only be accessible to registered participants.
(5) Confidentiality: Participants should only be able to view certain details of transactions according to their permissions.
(6) Interoperability: The PartChain system must support the integration of existing enterprise systems such as enterprise resource systems.
(7) Low Entry Barrier: The PartChain system should provide a minimal economic barrier to enable small and medium-sized companies to join the network.

Additionally, the focus on blockchain requires some other additional constraints. These are as follows:
(8) No Central Authority: The PartChain system should be decentralized without the need of a central authority that maintains part data or the transfer of ownership. Thus, there is no single point of failure and compromise. This brings about trust and auditability in the supply chain network.
(9) Verifiability: Each participant should be able to verify whether data were changed.

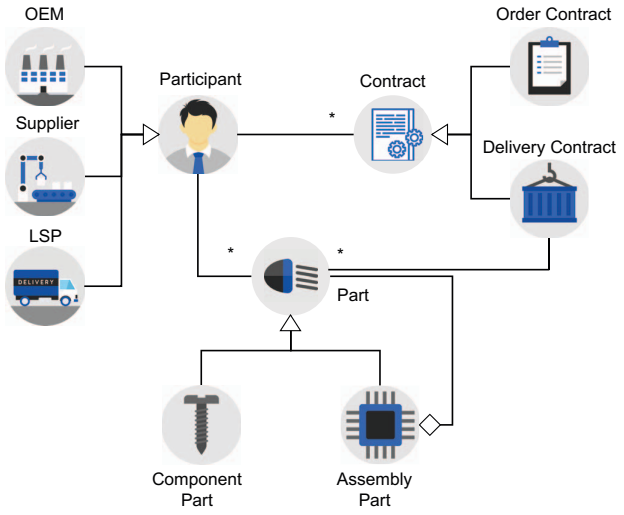


Fig. 1. **PartChain Analysis Object Model:** The model provides a better clarification of the problem domain and an approach for the development of the solution.

C. Analysis

For a better understanding of the problem domain we created an analysis object model (Figure 1). There are three types of *Participants* in the supply chain network: *OEM*, *Supplier*, and *LSP*. *OEMs* and *Supplier* can produce *Parts*, which are either *Component Parts* or *Assembly Parts*. *Assembly Parts* contain a list of *Component Part*. Each *Part* is assigned to a *Participant*. There are two types of *Contracts*: *Order Contract* and *Delivery Contract*. *OEMs* and *Supplier* can create an *Order Contract* to request new *Parts* from a *Supplier*. The selected *Supplier* can create a *Delivery Contract* based on the *Order Contract*, which specifies the delivery conditions. The analysis of the problem domain is kept as generic as possible to represent a multitude of scenarios. The *Participant* and *Contract* taxonomies make it possible to extend and adapt the proposed decentralized traceability application.

D. Top-Level Design

Top-Level Design (c.f. Figure 2) focuses on the main components and technologies of the solution approach. It comprises the following four layers: *Asset*, *Application*, *Middleware*, and *Distributed Ledger*.

(1) Physical Layer: The *Physical Layer* consists of *Parts* and *Contracts*, which are digitalized using the *PartChain Mobile App* on the *Application Layer*.

(2) Application Layer: Before participants can interact with the *Permissioned Blockchain*, they need an invitation that must be validated by one or more participants that manage the network permissions. They can grant and revoke permissions such as access rights via the *PartChain Web App*. Each change of permission is recorded and thus traceable on the *Permissioned Blockchain*. Subsequently, each participant can initiate different transactions using the *PartChain Mobile App* based on their permissions.

(3) Middleware Layer: The *Communication Gateway* is responsible for the communication and works as an adaptor for the *Application Layer*, the *Permissioned Blockchain*, and the *Off-Chain DB*. The *PartChain Mobile App* and *PartChain Web App* can send, write, and read requests to the *Communication Gateway* on the *Middleware Layer*, which transmits it to the *Distributed Ledger Layer*. The *Communication Gateway* can also fetch metadata such as the address of a participant from the *Off-Chain DB*. The *Off-Chain DB* represents the database of other existing systems such as enterprise resource planning systems.

(4) Distributed Ledger Layer: The *Distributed Ledger Layer* is responsible for data storage. The proposed systems uses a *Permissioned Blockchain*, which provides a selective level of privacy and low transaction times and costs. When data is recorded in the *Permissioned Blockchain*, it is verified and synchronized across all *Nodes* in the network.

IV. Case Study

Herein, we discuss the application of the proposed PartChain concept and the status of our prototype. We implemented the prototype using the Hyperledger Fabric framework[1]. A detailed discussion of the PartChain system and its use cases are presented after presenting an overview of the blockchain network definition.

[1]https://www.hyperledger.org/projects/fabric

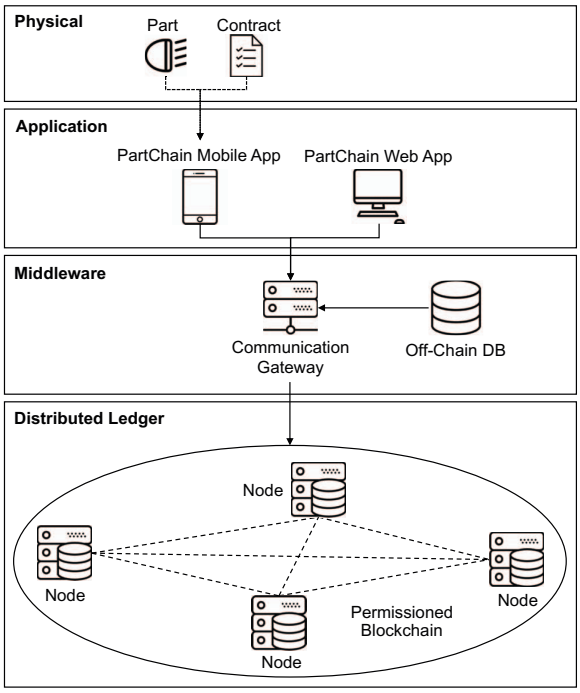


Fig. 2. **Top-Level Design of the PartChain system:** The PartChain system comprises of four layers, which are as follows: Physical, Application, Middleware, and Distributed Ledger. The key component is the Communication Gateway in the middleware layer, which acts as an adaptor between the Application Layer, Distributed Ledger Layer, and the Off-Chain DB.

A. Objectives

A prototype is used to verify the feasibility of the proposed system and to examine feedback by employees of our industry partner. We implemented a prototype of the proposed system to provide further insights into the benefits and challenges of permissioned blockchain networks.

B. Concept and Use Cases

Herein, we discuss the use cases of the prototype, the involved participants, parts, transactions, and smart contracts in detail. In the prototype, only registered participants can interact with the blockchain network. Participants can be OEMs, suppliers, or LSPs. OEMs and suppliers have access to all transactions and operations including network management, whereas LSPs can only initiate the handover of parts and scan them for information. Further, we discuss the use cases of the proposed PartChain concept as shown in Figure 3. The key component of our PartChain concept is the PartChain mobile app. Each time a participant scans a part, the current location and a timestamp is recorded in the event history of that part, enabling comprehensive tracking and tracing of parts.

OEMs and suppliers can *(1) Register a Part* on the *Blockchain* as a *Digital Part* by scanning the quick response (QR) code of a part using the PartChain mobile app. Also, component parts can be added to an assembly part by scanning their QR codes.

We developed two smart contracts that can be used to digitalize order and delivery contracts. The two smart contracts can automatically verify and execute the conditions for the transfer of ownership of a *Digital Part*. The two points where the transfer of ownership is needed are between supplier and LSP and between LSP and an OEM or another supplier. In the case of low inventory, an OEM or supplier can *(2a) Create an Order*. The order sets the conditions for the *Order Contract* such as the quantity of parts required, the supplier and the requested delivery date. The selected supplier is notified and can either *(2b) Create a Delivery* or decline the Order. The supplier scans the QR codes of each part using the PartChain mobile app to add parts to the delivery. Next, the supplier assigns the delivery to a registered LSP. The entered information sets the condition for the *Delivery Contract*.

The selected LSP is notified to *(2c) Pick-Up the Delivery*. The LSP opens the pending delivery and scans the QR codes of the parts provided by the supplier to collect the delivery. The *Delivery Contract* verifies if the scanned parts match the specified conditions. If the conditions are met, the pick-up of the parts is accepted and the ownership of the digital parts is transferred from the supplier to the LSP on the blockchain.

To *(2d) Accept an Order*, the OEM or Supplier needs to open the respective delivery and scan the QR codes of the received part. The *Order Contract* verifies if the received parts and the requested delivery date match with the specified conditions. If the conditions are met, the receipt of the parts is accepted and the ownership of the digital parts is transferred from the LSP to the respective supplier or OEM on the blockchain. If a participant tries to scan, e.g., the QR code of an unregistered part, the conditions of the smart contract would not be met. In such a scenario, the PartChain mobile app returns an error message and the ownership transfer of the unregistered part is rejected and is recorded on the blockchain.

Each participant can obtain details of a part by *(3) Scanning the Part*. The PartChain mobile app displays basic information and the complete event history of a part either as a list or a map. Furthermore, each participant can query for specific part IDs or part numbers (i.e., the type of part under consideration), and batch numbers, thus enabling a comprehensive tracking and tracing.

C. Hardware/Software Mapping

Figure 4 shows the hardware, software, and communication mechanisms between components that were used to implement the PartChain prototype.

The *:Application* subsystem contains two components: the *PartChain Mobile Client* and the *Web Client*. The *Web Client* component was implemented using Angular 5, running on a local server. Participants with access to the permissions management can login into the *PartChain Web App* using their Hyperledger Fabric credentials to add or edit permissions. Through the *PartChain Mobile App*, participants can initiate

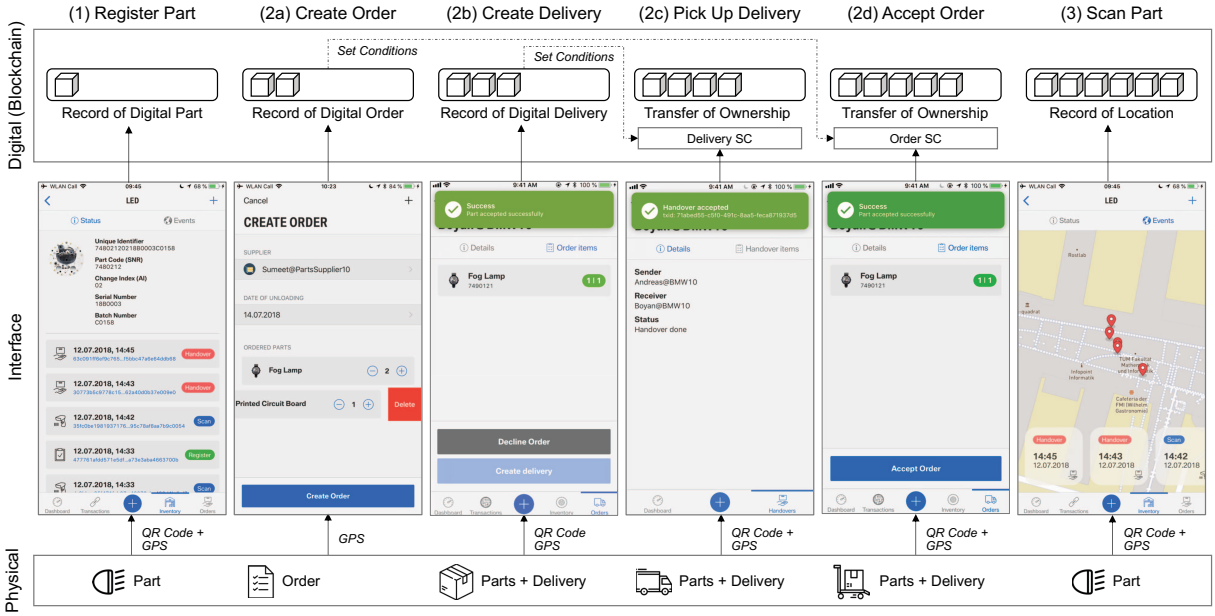


Fig. 3. **PartChain concept and use cases:** Design of the PartChain concept using the PartChain mobile app as a communication interface, which enables the creation and sharing of unique digital representations of physical parts. The Order and Delivery smart contracts (SC) are validated if the set conditions are met to execute the transfer of ownership. All events are recorded on the blockchain in a reliable and tamper-proof manner.

and query transactions such as register part, create order, or create delivery.

The key component of the *:Middleware subsystem* is the *Composer REST Server*, which provides a REST interface to the *PartChain Mobile* and the *PartChain Web Apps*. It works as an adaptor for the *HyperledgerFabric* and for the *Off-Chain DB* by enabling interoperability between the blockchain and existing systems. The *Off-Chain DB* uses a SQL Database. Both, the *PartChain Web* and the *Mobile Apps*, communicate with a *Peer* using the *Composer REST Server* via HTTP.

In the Hyperledger Fabric framework, a *Peer* represents a node of the blockchain network. A *Peer* contains a *Ledger* and the *PartChain Code*. The *Ledger* contains the sequenced, tamper-proof record of all transactions that are synchronized across all *Peers*. The *PartChain Code* defines, among other things, the smart contracts and transaction types. Furthermore, each *Peer* can propose and submit new transactions to the *Ledger*, thus changing the *World State DB* that uses an Apache CouchDB. Transactions are processed by the following *HyperledgerFabric* components: The *Certificate Authority (CA)* provides a number of certificate services to the participants such as the validation of new transactions. The *Orderer* arranges a new transaction, proposed by a *Peer*, into blocks and broadcasts the newly created block to all *Peers* in response.

D. Findings

In this section, we discuss the challenges and limitations of the current development status regarding the PartChain prototype. Although blockchain technology is still in its initial stage, it is obvious that the technology has enormous potential in addressing the challenges lack of transparency and responsiveness in supply chain networks.

We registered 1000 parts on the blockchain with an average transaction time of 0.5 s, enabling information to be shared across the network in a near real-time manner. However, we faced several issues during the development process. The challenges include the immaturity of the Hyperledger Fabric composer tools, which complicated the development process and interfacing with the blockchain component. Additionally, we were not able to view details of transactions. Hence, we had to query individual transaction histories rather than using the intended historian application programming interface.

The PartChain prototype was successfully demonstrated in an evaluation workshop with employees from corporate quality, innovation, information technology, and logistics department of our industrial partner. The participants acknowledged the ability of blockchain technology to improve cross-organization collaboration in supply chain networks. From a business standpoint, however, it is crucial to establish common standards. Hence, the collaboration in business and technology consortia is essential. Our future projects will focus on the most valuable parts, namely A parts according to the ABC inventory analysis. From a technology standpoint, secure and tamper-proof identification of physical parts has to be ensured to verify the authenticity of the parts. Cryptographic anchors could be a promising solution for this purpose.

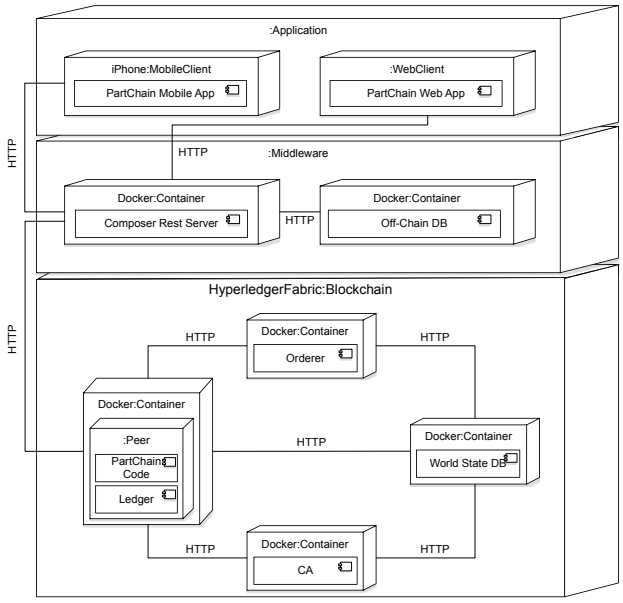


Fig. 4. **PartChain Deployment Diagram:** The figure shows the deployment diagram of the PartChain System. The *:Middleware* and *:Blockchain* subsystems are deployed using docker container and thus can be deployed for any provider.

V. Conclusion

Complex supply chain networks lack transparency and responsiveness. Hence, they hardly provide easy and automated ways of ensuing traceability or proving the authenticity of parts. Here, we presented PartChain and its implementation using the Hyperledger Fabric framework. The idea of PartChain is to create, monitor, and share a unique digital representation of a physical part across a supply chain network in a secure and tamper-proof manner. Hence, the PartChain mobile app bridges the gap between the digital and physical world using a mobile device enabling the digitalization of physical supply chain processes. PartChain provides a cost-efficient possibility to establish a supply chain traceability system with a low barrier to entry, which is particularly important for small- and medium-sized companies.

Here, we addressed the functional and non-functional requirements and constraints described in Section III-B. The requirements of *(1) Provenance and Authenticity of Parts*, *(2) Tracking and Tracing of Parts*, *(3) Digitalization of Documents* and *(7) Low Entry Barrier* were addressed by introducing a mobile application in the supply chain process. This application creates an easy to use transition between the physical object and the digital twin. The *(4) Access Control* and *(5) Confidentiality* requirements were archived using a private permissioned blockchain, which introduced a permission system. Finally, the required *(6) Interoperability* was archived using an adapter to an external Off-Chain database. This database stores additional information about participants and parts.

However, the application and evaluation of the scalability of the PartChain in real-world scenarios were not addressed in this paper. Therefore, our future work would focus on investigating the integration of existing systems, development of reusable micro services, and integration of IoT devices and artificial intelligence to automate data collection and analysis for optimizing supply chains. Moreover, as an increased amount of data is expected to be managed by blockchain-based systems, the adoption of blockchainless frameworks (e.g., IOTA and Hedera Hashgraph) that provide the necessary scalability need to be investigated.

References

[1] D. Tapscott and A. Tapscott, *Blockchain Revolution*. 2016.

[2] R. Bolton, J. McColl-Kennedy, L. Cheung, A. Gallan, C. Orsingher, L. Witell, and M. Zaki, "Customer Experience Challenges: Bringing Together Digital, Physical and Social Realms Customer Experience Challenges," *Journal of Service Management*, 2018.

[3] X. Xu, I. Weber, M. Staples, L. Zhu, J. Bosch, L. Bass, C. Pautasso, and P. Rimba, "A Taxonomy of Blockchain-Based Systems for Architecture Design," *IEEE International Conference on Software Architecture (ICSA)*, 2017.

[4] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.

[5] M. Swan, *Blueprint for a New Economy*. 2015.

[6] N. Szabo, "Formalizing and Securing Relationships on Public Networks," *First Monday*, 1997.

[7] M. Alharby and A. van Moorsel, "Blockchain Based Smart Contracts: A Systematic Mapping Study," *Computer Science & Information Technology (CS & IT)*, 2017.

[8] V. Buterin, "Ethereum White Paper: A Next Generation Smart Contract & Decentralized Application Platform," 2014.

[9] L. W. Cong, Z. He, and J. Zheng, "Blockchain Disruption and Smart Contracts," *SSRN Electronic Journal*, 2017.

[10] M. Kõlvart, M. Poola, and A. Rull, "Smart Contracts," *The Future of Law and eTechnologies*, 2016.

[11] M. Grieves, "Digital Twin: Manufacturing Excellence Through Virtual Factory Replication," 2014.

[12] A. Canedo, "Industrial IoT Lifecycle via Digital Twins," *Proceedings of the Eleventh IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis (CODES)*, 2016.

[13] J. Ríos, J. C. Hernández, M. Oliva, and F. Mas, "Product Avatar as Digital Counterpart of a Physical Individual Product: Literature Review and Implications in an Aircraft," *Advances in Transdisciplinary Engineering*, 2015.

[14] G. N. Schroeder, C. Steinmetz, C. E. Pereira, and D. B. Espindola, "Digital Twin Data Modeling with AutomationML and a Communication Methodology for Data Exchange," *IFAC-PapersOnLine*, 2016.

[15] S. Chen, R. Shi, Z. Ren, J. Yan, Y. Shi, and J. Zhang, "A Blockchain-Based Supply Chain Quality Management Framework," *IEEE 14th International Conference on e-Business Engineering (ICEBE)*, 2017.

[16] M. Nakasumi, "Information Sharing for Supply Chain Management Based on Blockchain Technology," *IEEE 19th Conference on Business Informatics (CBI)*, 2017.

[17] Feng Tian, "A Supply Chain Traceability System for Food Safety based on HACCP, Blockchain & Internet of Things," *International Conference on Service Systems and Service Management*, 2017.

[18] K. Toyoda, P. T. Mathiopoulos, I. Sasase, and T. Ohtsuki, "A Novel Blockchain-Based Product Ownership Management System (POMS) for Anti-Counterfeits in the Post Supply Chain," *IEEE Access*, 2017.

[19] Z. C. Kennedy, D. E. Stephenson, J. F. Christ, T. R. Pope, B. W. Arey, C. A. Barrett, and M. G. Warner, "Enhanced Anti-Counterfeiting Measures for Additive Manufacturing: Coupling Lanthanide Nanomaterial Chemical Signatures with Blockchain Technology," *Journal of Materials Chemistry C*, 2017.

[20] D. Heber and M. Groll, "Towards a Digital Twin: How the Blockchain can Foster E/E-traceability in Consideration of Model-Based Systems Engineering," *21st International Conference on Engineering Design (ICED)*, 2017.

PartChain: A Decentralized Traceability Application for Multi-Tier Supply Chain Networks in the Automotive Industry

Conference Proceedings:
2019 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON)

Author: Daniel Miehle

Publisher: IEEE

Date: Apr 2019

IEEE Requesting permission to reuse content from an IEEE publication

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE

3.2 Toward a Decentralized Marketplace for Self-Maintaining Machines

This publication (Publication [B]) describes a visionary DLT application, that enables self-maintaining machines to act autonomously across an industrial marketplace.

Conference: 2019 IEEE International Conference on Blockchain
Number of Pages: 8
Review: Peer Reviewed (3 Reviewers)

The author of this dissertation contributed substantially to this publication, in particular concerning the development of the idea and approach, the conduction of the literature review, the elicitation of the business requirements, the design of the system architecture, the design and conduction of the evaluations, and writing the publication.

Goal

Industrial procurement processes are characterized by manual processes and offer hardly any technical solutions that allow machines to order and pay for materials and services autonomously. The goal of this DLT application was to create an open marketplace and to enable machines to act as economically, autonomous machines in that marketplace to streamline industrial procurement processes.

Approach

First, we conducted a structured literature review to provide an overview of the current state of research at the intersection of DLTs and industrial procurement. We realized a conceptual implementation of the DLT marketplace, which enables the management of the (i) marketplace, (ii) quotations, and (iii) orders using nine smart contracts. Machines can autonomously create orders, including the search and selection of quotations, and accept orders. For the selection of quotations, we developed a quotation ranking procedure that includes the material price and supplier evaluation. Thereby, the supplier evaluation works as an incentive system. To evaluate our prototype, we conducted a semi-structured interview with thirteen experts from the automotive industry.

Results

We conducted a case study with our industry partner to demonstrate the functionality of our DLT application. On a business level, the results of the semi-structured interview showed that operational procurement tasks would be more likely affected by DLTs than strategic procurement tasks. The respondents confirmed the highest value of the DLTs in the area of *Ordering and Payment* (77%) and *Monitoring of Orders* (69%) improving the traceability and auditability of industrial procurement processes. On a technical level, the case study highlighted that economically, autonomous machines require verifiable digital identities to enable secure and trusted interactions on the marketplace (e.g., by using secure hardware modules).

IEEE Reuse Requirements

With regard to the IEEE reuse requirements for dissertations, the accepted and not the published version of the publication is used in the following.

Toward a Decentralized Marketplace for Self-Maintaining Machines

Daniel Miehle[1], Matthias M. Meyer[2] Andre Luckow[3], Bernd Bruegge[1], Michael Essig[2]
[1]Technical University of Munich, Germany
Email: daniel.miehle@tum.de, bruegge@in.tum.de
[2]Bundeswehr University Munich, Germany
Email: {ma.meyer, michael.essig}@unibw.de
[3]Munich University of Applied Sciences, Germany
Email: luckow@nm.ifi.lmu.de

***Abstract*—Industrial procurement processes are inefficient, expensive, and error-prone as they require the agreement of numerous participants, audit-proof documentation, and are characterized by a high number of manual tasks. Blockchain technology has the potential to counter these challenges by combining peer-to-peer networks, cryptography, and consensus algorithms. A blockchain is a decentralized database that stores transactions and data transparently, chronologically, and tamper-proof in a distributed network. In this paper, we present a blockchain-based industrial marketplace, where machines have an identity and account for selecting, ordering, and paying materials in an automated manner using smart contracts. This allows manufacturers with a low real-net-output-ratio, such as car manufacturers that depend on numerous suppliers, to optimize their purchasing processes. Using our case study, we demonstrate prototypical implementation of the decentralized marketplace and its applications in the area of industrial maintenance. Furthermore, we conducted a semi-structured interview with procurement and information technology experts from the automotive industry. The results indicate that our prototype offers a promising approach streamlining procurement processes within the next two to five years.**

***Index Terms*—Blockchain Technology, Smart Contract, Machine-to-Machine Economy, Smart Maintenance, Smart Procurement, Industry 4.0**

I. Introduction

Procurement describes the process of acquiring materials or services from external partners. Procurement processes are characterized by a high number of manual activities, the coordination of numerous external sources, and high costs (e.g., for escrow services). Due to the growing complexity and individualization of products, manufacturers are facing an increasing number of parts and part variants, which in turn lead to a growing number procurement tasks. As a result, current industrial procurement processes are inefficient, expensive, and error-prone.

In the following, we will concentrate on procurement processes in the automotive sector, where around two thirds of the value added is generated by external partners. Therefore, optimizing existing processes and integrating new processes that increase efficiency and transparency is key to remaining competitive. The goal is to increase automation rates, accelerate international financial transactions, and build a trusted infrastructure to ensure the integrity of procurement processes and the liquidity of companies. Blockchain technology presents promising opportunities for countering these challenges and add new value to industrial procurement processes (see e.g., [1]–[3]). Blockchain is a decentralized database that stores transactions transparently, chronologically, and tamper-proof in a distributed network [4]. It provides the foundation for securely exchanging tangible (e.g., materials) and intangible assets (e.g., digital currencies). Integrating blockchain technology into industrial procurement processes represents the first step toward a machine-to-machine economy, which describes the vision whereby machines become active participants in business processes.

In this paper, we present a decentralized industrial marketplace for materials using an example from industrial maintenance. The blockchain-based marketplace enables machines to have a secure and trustworthy digital identities and their own accounts. Hence, machines can select, order, and pay for materials (e.g., consumables and spare parts) in an autonomous manner using smart contracts. Each step of the procurement process is reliably recorded on the blockchain, creating a tamper-proof audit-trail. We show that blockchain has the potential to streamline and accelerate procurement processes and reduce costs by eliminating the roles of intermediaries (e.g., banks). We demonstrate the feasibility and potential of blockchain in a case study that involves 10 companies, 50 machines, 10 quotations, and over 1000 transactions.

This paper is structured as follows. In Section II, we provide an overview on foundations and a structured review on the investigated literature. Section III presents the vision and conceptual structure of the decentralized industrial marketplace, and describes the vision, requirements, and the proposed system design of the marketplace. In the case study in Section IV, we discuss the prototypical implementation of the Smart Contracts using the Hyperledger Fabric Blockchain framework. We validate the prototypical implementation in Section V by means of a semi-structured interview with experts from the industry. Section VI summarizes the paper and its contributions.

II. Background and Related Work

This section describes relevant foundations on which our decentralized marketplace is based on and reviews the related work.

A. Blockchain Technology

Blockchain technology is a type of distributed ledger technology (DLT) that refers to a decentralized database, which is distributed across multiple nodes and locations. Blockchain was first developed by Satoshi Nakamoto in 2008 as the underlying technology for the cryptocurrency Bitcoin [4]. A blockchain network records and shares data in a secure trusted, and tamper-proof manner using cryptography and consensus algorithms without the need for a trusted third-party [5]. A blockchain is a list of blocks that consists of a timestamp, a nonce, a hash value, and transactions as payload [4]. Each block is linked to its previous block through a cryptographic hash. Consensus algorithms ensure that the decentralized and distributed databases are kept consistent and thus, provide a single source of truth [6]. Several approaches exist for reaching consensus in blockchain networks. In public blockchains such as Bitcoin and Ethereum, anyone can access the network. Data and transactions are visible to all participants and everyone can join the consensus process [7]. Consensus in public blockchains is mainly realized through Proof-of-Work (PoW) and Proof-of-Stake (PoS). In private blockchains such as Corda and Hyperledger, only pre-selected nodes can access the network and participate in the consensus process [7]. Therefore, they can use more efficient consensus algorithms such as Practical Byzantine Fault Tolerance (PBFT) algorithms, which focus on speed and scalability.

B. Smart Contract

Nick Szabo [8] first described the idea of smart contracts as a digital protocol that automatically carries out the terms of an agreement. Ethereum was the first general purpose blockchain framework that provided the ability to integrate smart contracts [9]. Smart contracts provide a programmed algorithm that can be used, e.g., for executing contracts or automating business processes [10]. This makes it possible for agreements and transactions to be carried out among parties or machines in a traceable, transparent, and irreversible manner without the need for a trusted third party [11]. Benefits of smart contracts include reducing errors, time and costs for verifying and executing processes [10], [12], [13].

C. Procurement

Following [14], procurement describes the "management of the external resources of an enterprise." The overall goal of procurement is to secure the availability of all materials and services that are needed for realizing and steering the primary and supporting activities of a company [14]. Purchasing desribes a subset of procurement and refers mainly to buying materials or services. Procurement tasks can be divided into strategic and operational tasks [14], [15]. The focus of operational procurement is to meet daily purchasing needs, such as ordering materials and services, monitoring orders, and evaluating suppliers [14]. Strategic tasks deals with long-term sourcing objectives of companies, such as supplier portfolio management and improving purchasing processes [14], [16].

D. Maintenance

Maintenance is defined as any task performed that keeps technical equipment in or returned to proper working condition [17]. There are two major maintenance tactics, namely corrective and preventive maintenance [17], [18]. Corrective maintenance is performed after detecting the failure (e.g., due to an unexpected breakdown of a machine) [17], [18]. Preventive maintenance includes time-based and condition-based maintenance activities for detecting and correcting incipient failures of machines before failures occur [17], [18]. Condition-based maintenance monitors the condition of machines using sensors to estimate when maintenance services need to be performed [17], [18].

E. Related Work

For the design of our literature review, we rely on methods presented by Bandara et al. [19], vom Brocke et al. [20], and Webster and Watson [21]. Based on this, we derive four steps: Subsequent to a literature search (1), relevant articles are identified (2) and further analyzed (3). Afterward, the results of the analysis are structured (4). With the aim of analyzing a wide range of research literature, we query the databases listed in Table 1. We use search terms for blockchain technologies and related concepts (i.e., distributed ledger technologies, smart contracts) and link them via an `AND` operator to terms that ensure search results within marketplaces and relevant industrial application domains such as procurement and maintenance.

TABLE I
Parameters of Keyword Search

Databases	arXiv, IEEE Xplore, Research Gate
Search Fields	Title, Abstract, Keywords
Source Types	Journales, Conferences
Search Term	("Distributed Ledger" OR "Blockchain" OR "Smart Contract") AND ("Procurement" OR "Purchasing" OR "Maintenance" OR "Marketplace")

The search strategy renders a total of 181 results. Relevant papers were selected by analyzing titles, abstracts and full text. The literature analysis results in 19 papers (doubles removed). All relevant papers were published after the year 2016. When looking at the year distribution of the selected papers, 1 paper (5%) was published in 2016, 3 papers (16%) in 2017, 14 papers (74%) in 2018 and 1 paper (5%) was published in 2019. Regarding the geographic distribution of the selected papers, the top countries of authors are Germany (21%) and the United States of America (16%). Furthermore, we clustered the publications by the type of marketplace and procurement tasks (Table I). Thus, assigning one paper to multiple criteria is possible.

TABLE II
OVERVIEW OF THE INVESTIGATED LITERATURE

Application Areas	Papers
A. Marketplaces	
Data Marketplace	[1], [22]–[31]
Digital Asset Marketplace	[23], [27], [29], [32], [33]
Physical Asset Marketplace	[1], [3], [28], [29], [34]
B. Procurement Use Cases	
Payment Infrastructure	[22]–[27], [29]–[33], [35]–[37]
Contract Creation	[1]–[3], [35]
Order Management	[1]–[3], [34], [35]
Regulatory Transparency	[2], [3], [26], [36]
Demand Planning	[1], [3], [28]
Supplier Evaluation	[1], [2], [29]
Tracking and Tracing	[2], [3], [36]
Price Negotiation	[2], [36]
Supplier Selection	[2], [36]

Blockchain technology has been considered as a possible solution for marketplaces due to ownership neutrality [22], [29], data integrity [29], [32], data privacy [22], [24], [29], and security [22], [29].

In the field of marketplaces, 44% of articles were published on *Data Markets* (e.g., trading of consumer data), 28% percent deal with *Digital Asset Marketplaces* (e.g., purchasing apps for edge devices) or *Physical Assets Marketplaces* (e.g., purchasing consumables in industrial manufacturing), respectively.

The following procurement use cases are mentioned in order of frequency of response (cf. Table I). The majority of publications deals with *Payment Infrastructure*, such as procure-to-pay processes using blockchain technology. Other use cases include automated *Order Management* by machines [1], [3], *Supplier Evaluation* [1], [2], *Tracking and Tracing* of materials in complex supply chain networks [2], [3], [26], [36], and *Price Negotiations* such as e-auctions [2].

In the following, we refer to publications that are closely related to the vision of our decentralized industrial marketplace. For example, Bahga et al. [1] introduce a blockchain-based platform for industrial internet of things in manufacturing enabling machines to execute micropayments for consumables. Nicoletti et al. [2] describe an auction system based on smart contracts to store negotiated contracts and automate settlement improving data integrity, security, and transparency. Seitz et al. [32] introduce a decentralized marketplace for industrial edge applications that relies on blockchain to create transparency for all stakeholders involved and to enable the traceability of app installations on edge devices. Teuteberg et al. [3] propose a blockchain-based order process for increased auditability and automation enabled using smart contracts. However, research at the interface of blockchain technology and industrial procurement processes in terms of theoretical concepts and empirical evaluation (e.g., implementation of prototypes) is yet scarce. In addition, there is no holistic approach that combines the various industrial procurement tasks on a single blockchain-based marketplace.

III. MARKETPLACE

In this section, we describe the vision, functional and non-functional requirements, and the architecture of the decentralized industrial marketplace.

A. Vision

In our vision machines become self-determined market participants in industrial environments. Machines are able to interact with other machines or humans over an industrial decentralized marketplace. Suppliers can initiate quotations into the decentralized marketplace. Buyers of companies serve as quality gate and have to approve initiated quotations as materials have to meet companies standards to make them available to their machines. Smart contracts enable machines to perform various actions such as selecting quotations, ordering materials from other machines, and booking of maintenance services in an automatic and audit-proof manner. Machines are able to confirm the arrival of material, pay for them, and rate the supplier performance based on defined criteria. By running on a peer-to-peer network, participants have control over the marketplace and machines can use digital currencies and escrow services, which only release money once the materials have been delivered. Furthermore, service engineers can document performed services and installation of spare parts in a digital service book increasing the resale value. The digital service book can be shared with authorized third parties.

B. Requirements and Approach

There are different procurement and awarding methods, including the marketplace principle and request for quotation (RfQ) process, which focuses on standard and specific materials and services, respectively. In the marketplace principle, machines are able to select materials based on an algorithm based on quotations approved by buyers of the associated company. Buyers and suppliers are responsible for ensuring that quotations are available in the marketplace and are selectable for machines. In the RfQ process, machines actively ask for specific materials. Requested suppliers participate in a bidding process. In this paper, we consider the marketplace principle as it represents the first step towards an automatic selection process of machines using current technological capabilities.

The following functional and non-functional requirements must be met to successfully implement an industrial marketplace for self-maintaining machines:

(1) Quotation Management: Suppliers must be able to initiate quotations into the marketplace. Buyers must be able to approve quotations to make them available to their machines.
(2) Order Management: Machines must be able to select approved quotations in the marketplace to create an order for materials. Furthermore, machines must be able to confirm the arrival of materials.
(3) Supplier Evaluation: Machines must be able to evaluate the performance of suppliers.

(4) Financial Transactions: Machines must be able to execute payments to other machines or suppliers.
(5) Digital Maintenance Book: Service engineers must be able to record their performed maintenance services and the installation of materials.
(6) Independence: The marketplace should be a decentralized system in which there is no regulating authority.
(7) Traceability: All activities and events must be documented for regulatory reasons in an audit-proof manner. The records must be traceable for the involved participants.
(8) Access Control: The marketplace should only be accessible to registered participants and machines.
(9) Privacy: The marketplace must be able to make private transactions.
(10) Data integrity: The marketplace must assure the accuracy and consistency of data.
(11) Availability: The marketplace must be available at any time.
(12) Scalability of Transactions: The marketplace must be able to process a large number of transactions.
(13) Scalability of Participants: The marketplace must be able to include a large number of participants.

C. Analysis

We created the analysis object model (see Figure 1) to better describe the structure of the problem domain and the relationships among the objects.

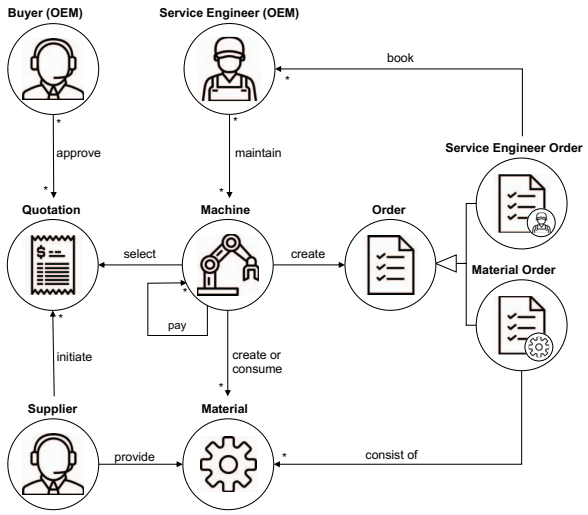


Fig. 1. **Analysis Object Model:** The model supports to better understand the problem domain and to develop the solution approach of our concept.

There are four participants that interact with the marketplace, namely *Supplier*, *Buyer*, *Machine*, and *Service Engineer*. *Suppliers* can initiate *Quotations* into the marketplace and provide *Materials*. *Buyers* can approve *Quotations* that meet corporate standards, to create a purchasing framework for their *Machines*. *Machines* can select approved *Quotations* based on a selection algorithm that takes criteria such as price and supplier rating into account, and can place *Orders* for required *Materials*. In case maintenance services (e.g., the installation of a spare part) are required, *Machines* can also book a *Service Engineer* for the planned arrival time for *Materials*. *Machines* can confirm the arrival of materials, pay for them, and evaluate the *Supplier* based on its performance. Supplier ratings can be used by all participants in the marketplace. After the *Service Engineer* has serviced the *Machine*, the performed services and installed *Materials* are recorded on a digital maintenance book.

The analysis of the problem domain is kept as generic as possible to represent a multitude of scenarios. The *Order* taxonomy allows us to extend and adapt our system.

D. Architecture

Figure 2 shows the proposed architecture for the decentralized industrial marketplace, which focuses on the main components and technologies of the solution approach. It comprises the following four layers: *Physical*, *Application*, *Middleware*, and *Distributed Ledger*.

(1) Physical Layer: The *Physical Layer* contains *Materials*, *Quotations*, *Orders*, and *Machines*.

(2) Application Layer: The *Maintenance Mobile App*, *Marketplace Web App*, and *Machine App* are used to create, monitor, and share digital representations of assets of the *Physical Layer*. Before users and *Machines* can interact with the *Permissioned Blockchain*, they need an invitation that must be validated by one or more participants, who are allowed to manage the marketplace permissions via the *Marketplace Web App*. To interact with the marketplace service engineers use the *Maintenance Mobile App* on the shopfloor, buyers and sellers use the *Marketplace Web App*, and *Machines* use the *Machine App*.

(3) Middleware Layer: The *Communication Gateway* works as an adaptor for the *Application Layer*, the *Permissioned Blockchain*, and the *Off-Chain DB* and is responsible for the communication. The *Maintenance Mobile* and *Marketplace Web App* can initiate, read, and write requests and transactions to the *Communication Gateway* on the *Middleware Layer*, which transmits it to a company *Node* in the *Distributed Ledger Layer*. In contrast, each *Machine* within its *Machine App* operates its own *Node*, which allows it to have a blockchain identity and account. The *Communication Gateway* can fetch metadata such as the participant's address from the *Off-Chain DB*, which represents the database of other existing systems, such as enterprise resource planning systems.

(4) Distributed Ledger Layer: The *Distributed Ledger* provides data storage and ensures the execution of smart contracts in a decentralized and trusted manner. The marketplace uses a *Permissioned Blockchain*, which provides a selective level of privacy as well as low transaction times and costs. Therefore, it is very feasible to introduce further transactions and participants. Each activity such as the payment of material is recorded and is thus traceable on the *Permissioned Blockchain*.

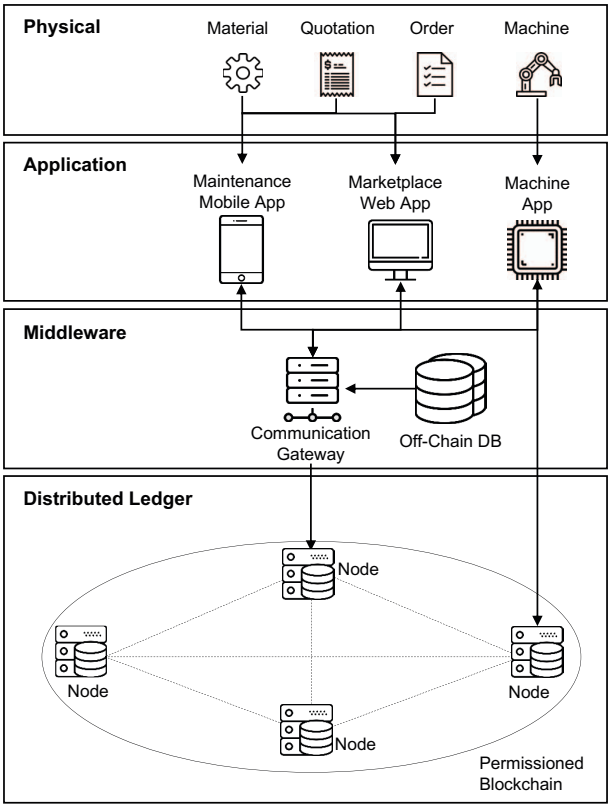


Fig. 2. **Top Level Design:** The marketplace architecture consists of four layers including the Physical, Application, Middleware, and Distributed Ledger. The Maintenance Mobile App, Marketplace Web App, and Machine App act as interfaces and allow the interaction among the Physical Layer, the Off-Chain DB, and the Permissioned Blockchain.

IV. Case Study

In this section, we discuss the application of the proposed industrial marketplace concept and the status of our prototype. We present a discussion of payment implementation options and the required smart contracts.

A. Objectives

The prototype is used to verify the technical feasibility of the industrial marketplace concept and to examine feedback by our industry partner. In the case study, we focus on the implementation of the smart contracts for the decentralized marketplace using Hyperledger Fabric. Therefore, not all parts of the proposed architecture (e.g., user interfaces) were implemented.

B. Payment Implementation Options

There are different options that provide digital currencies in blockchain-based systems:

Digital Currencies: Existing digital currencies / cryptocurrencies such as Bitcoin or Ether can be used that provide easy access and divisibility into fractions enabling micro payments. However, due to the high volatility, low acceptance, and lack of regulation the use of existing digital currencies is not recommended.

Fiat Tokens: Central banks such as the European Central Bank can issue fiat tokens based on blockchain technology offering low volatility. However, users must trust the central bank as a regulating authority. The maturity of this scenario is expected to be reached in more than ten years.

Asset-backed Tokens: A consortium can create a stable digital currency, which is for example pegged to an existing fiat currency that can be validated either on-chain or off-chain (i.e., balances are stored in an off-chain database) In case of off-chain validation, users must trust the gateway as a single point of failure and the operator as a regulating authority. In case of on-chain validation, users must trust the consortium, but there is no single point of failure. This scenario can be implemented immediately, providing lower volatility than pure digital currencies, auditability, and low transaction times and fees. Since one of the requirements of the system is independence, there is no regulatory authority, we choose the on-chain solution for our prototype.

C. Smart Contract Implementation

We implemented a proof-of-concept prototype consisting of nine smart contracts to demonstrate the technical feasibility of the proposed industrial marketplace concept (see Figure 3). The implemented smart contracts can be divided into the three areas: *Marketplace*, *Quotation*, and *Order Management*.

Marketplace Management: Users with access to the marketplace management can *Create and Delete Companies* and associated *Users*. Each user of a company can be assigned specific roles (e.g., buyer, supplier, service engineer, or administrator) with different attributes and permissions. For example, the supplier role has additional attributes (e.g., *OnTimeRate*), which are needed to determine the supplier evaluation. Service engineers can *Register Machines* on the decentralized marketplace for their company. Each machine must be equipped with a unique identifier (e.g., RFID or QR Code), which is used as a machine's identity in the marketplace. In addition, each machine has an array of slots for managing materials (e.g., consumables and tools) and maintains a digital maintenance book on the blockchain network.

Quotation Management: A supplier can *Create a Quotation* for materials. Quotations published in the marketplace are binding and verifiable. The price of a quotation is valid for a certain period of time and can be adjusted. None of the other parameters in the quotation can be changed. The quotation status is initially set to `NOT APPROVED` since it was not checked by buyers from any of the registered companies. Buyers from registered companies are notified about new quotations on the marketplace. If the quotation fulfills the requirements and quality standards of a company, the buyer can *Approve a Quotation*, changing its status to `APPROVED`. Subsequently, the quotation is selectable for all of the company's machines.

Order Management: Machines continuously monitor their state (e.g., stock level, tool condition) and decide if they need

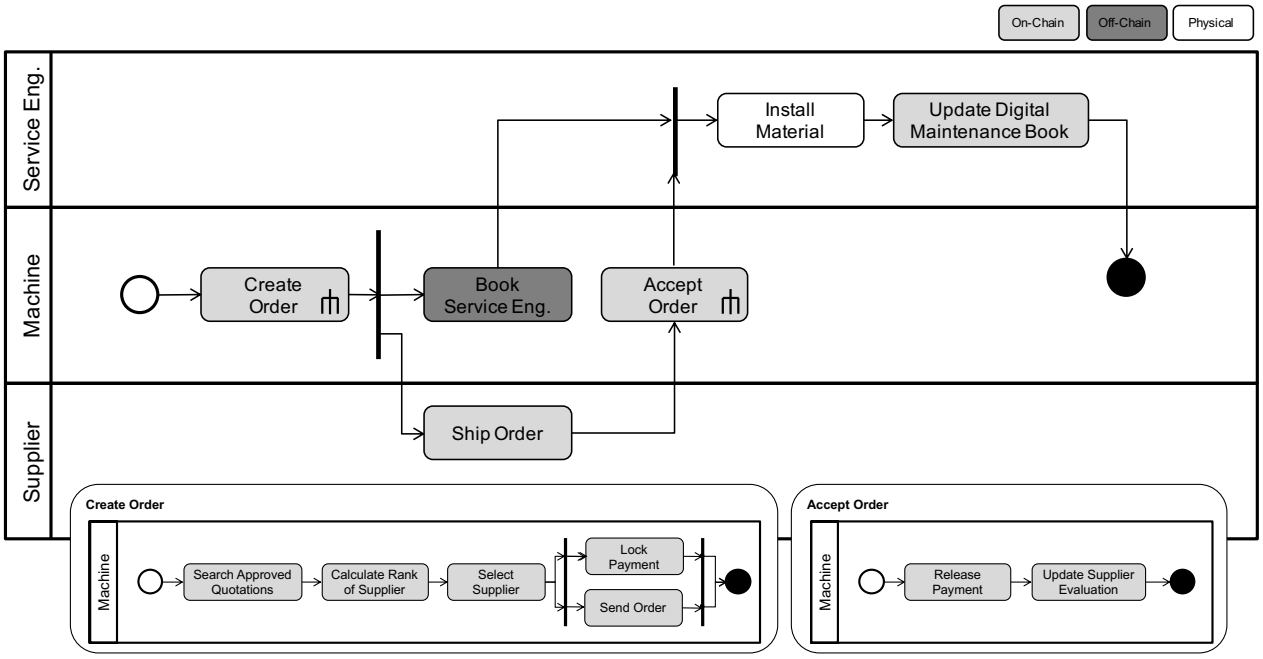


Fig. 3. **Activity Diagram for Order Management:** Visualization of the main process and two sub-processes performed by the machine, supplier, and service engineer. In addition, the on-chain and off-chain data management is shown.

to *Create an Order* for materials or maintenance services. We therefore distinguish between corrective triggers (e.g., low material stock) and preventive triggers (e.g., time-based maintenance services). In general, a machine *Searches the marketplace for Approved Quotations* for the required material number. Based on the price (p) of the required material (i) and the evaluation (e in percent) of a supplier (j), the machine *Calculates the Rank of the Suppliers*.

In the prototype, we use the following quotation for the rank (R) of a supplier (j):

$$R(j) = e(j) * (1 / p(i))$$

Machines *Select the Supplier* with the highest ranking. Thereupon, the smart contract locks the payment from the account for the machine, orders the required material number via the industrial marketplace, and sets the status of the order to *OPEN* This procedure minimizes the counterparty risk. In the event that a maintenance service is also required (e.g., for the installation of a tool), machines can book a service technician off-chain, since this is an internal company process. This sets the status of the service order to `BOOKED`. If the requested and actual delivery times for the materials match, machines *(Accept the Order* by scanning the order identifier (e.g., RFID or QR Code) on the received package. The order status is set to `DELIVERED` and the stock level of the receiving machine is updated. Next, the smart contract *Releases the Payment* to the account of the supplying machine and the order status is set to `CLOSED`. In addition, the machine *Updates the Supplier Evaluation* on the marketplace by comparing the requested and actual delivery times for the order. When carrying out maintenance services, service engineers scan the machine's identifier and *Install the Material*. The service engineer *Updates the Digital Maintenance Book* on the blockchain by recording the performed maintenance service including information about the installed materials (e.g., identifiers and timestamp). Next, the maintenance service order is closed and the status is set to `CLOSED`.

V. Evaluation

In this section, we evaluate the proposed decentralized industrial marketplace design in terms of requirements fulfillment and a semi-structured interview with thirteen procurement and information technology experts from the automotive industry.

A. Requirement Fulfillment

We examined the implemented prototype against the functional and non-functional requirements identified in Section III-B. Requirements for *(1) Quotation Management*, *(2) Order Management*, *(3) Financial Transactions*, *(4) Supplier Evaluation* and *(5) Digital Maintenance Book* were fulfilled by introducing smart contracts that express the business logic for procurement tasks as code. Furthermore, *(8) Access Control* and *(9) Privacy* requirements were archived by using a permissioned blockchain, which provides a permission system. The use of blockchain technology addressed the requirements for *(6) Independence*, *(7) Traceability*, *(10) Data Integrity* and

(11) Availability. Requirements for *(12) Scalability of Transaction* and *(13) Scalability of Participants* were not evaluated with the current prototype and thus, must be investigated in future research.

B. Semi-Structured Interview

We conducted a semi-structured interview with thirteen experts from our industrial partner in order to identify opportunities and challenges of using blockchain technology in procurement and to evaluate the implemented prototype. Of the thirteen experts interviewed, eight experts were from the procurement and seven experts from the IT department, working in the field of blockchain technology. We follow [14]–[16], [38] to identify relevant operational and strategic procurement tasks, which serve as the basis for the designing of the questions for the semi-structured interview. Figure 4 shows the influence of blockchain technology on operative and strategic procurement tasks.

First, we explained the technical basics of blockchain technology as well as the operational and strategic procurement tasks to the experts. Next, the prototype of the decentralized industrial marketplace was presented, after which the experts rated the influence of blockchain technology on the identified procurement tasks on a scale from "0" (no influence) to "5" (very high). Furthermore, they estimated the time required for introducing blockchain technology into operational and strategic procurement tasks on the basis of the categories of "less than 2 years", "2 to 5 years", "5 to 10 years", and "more than 10 years". Finally, we discussed the challenges the experts see for adopting blockchain in procurement and the further potential they see for combining blockchain and AI.

The potential of blockchain technology for operational tasks (mean values: 3.22) was rated higher than for strategic tasks (mean value: 2.88) by the respondents. For **operative procurement tasks**, the respondents see the highest potential in *Ordering and Payment* (77%) and the *Monitoring of Orders* (69%). All other operative procurement tasks indicate a medium potential. For **strategic procurement tasks**, approximately half of the respondents (54%) see the highest potential in *Negotiation Management*, *Process Optimization* and *Purchasing Controlling*, whereas *Strategic Demand Planning* and *New Technology Implementation* using blockchain technology was ranked low. Other strategic tasks are expected to be influenced moderately in the future. The time for adopting blockchain in procurement processes is estimated at two to five years for operational tasks and five to ten years for strategic tasks.

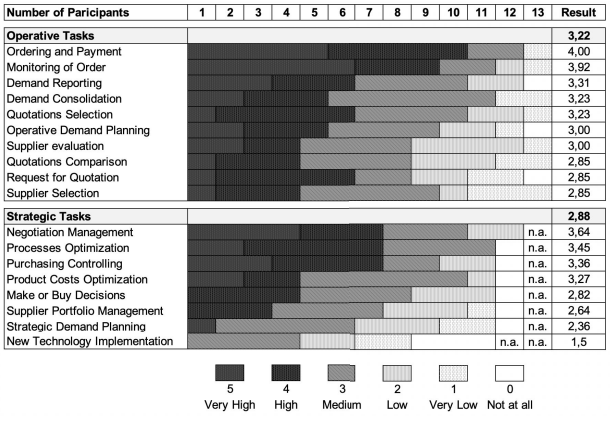


Fig. 4. **Results of the semi-structured interview:** Influence of blockchain technology on operative and strategic procurement tasks.

Overall, the results of the respondents show that IT experts see greater potential for our prototype compared to procurement experts. The highest potential was seen for *Negotiation Management* from an IT standpoint and *Order Management* from a business standpoint, respectively. The results from our procurement experts show a higher standard deviation, which indicates a higher uncertainty.

The main barriers for blockchain adoption are regulatory uncertainty and technical standardization (77%), such as interoperability and data exchange among separate blockchain networks. These are followed by technological readiness and scalability of blockchain-based systems (69%), employee resistance (62%) and the lack of trust among users in algorithms and smart contracts (39%). The interviews also showed the need for regular auditing of industrial procurement in order to maintain high quality standards. Hence, we integrated a supplier preselection in our industrial marketplace concept.

The main potential for combining blockchain with AI was identified for federated learning. Federated learning enables companies to collaboratively train a shared prediction model without having to expose training data. One example for this could be predicting material flows in complex supply chains. In case of a negative event, countermeasures such as hedging risks with derivatives could be initiated in near real-time. Further advantages are seen for adapting procurement processes to current market situations (e.g., customs duties, natural disasters) and the ability to securely store data and decisions during critical decision-making processes in an audit-proof manner.

VI. Conclusion

Current procurement processes are strongly characterized by manual and administrative processes. Hence, they hardly provide easy and automated ways for machines to order and pay for materials and services. In this paper, we present a decentralized industrial marketplace and its implementation in the Hyperledger Fabric framework. The goal of the blockchain-based marketplace is to enable machines to become self-determined market actors, that can order and pay for materials in an automated manner.

First, we conducted a structured literature analysis, which revealed a research gap in decentralized marketplaces in the domain of industrial procurement processes. Based on a detailed requirements analysis, we developed a prototype of a decentralized industrial marketplace using the example of a maintenance process. The prototype was successfully

demonstrated to our industrial partners. The experts confirmed the potential of blockchain technology for creating a trusted infrastructure for operational procurement tasks, such as automated ordering and settlement by machines, which improve auditability and traceability for decisions made by machines. The active participation of machines in industrial processes carry enormous potential and the concept could be implemented in the next two to five years. However, social, legal, and economic standards need to be established to enable an effective collaboration between machines and between machines and humans.

Future work should investigate the scalability of the current prototype (i.e., transaction and participant scalability), integrating existing systems (e.g., enterprise resource planning systems), and off-chain data storage and processing (e.g., using the Inter Planetary File System (IPFS)). Furthermore, adopting Directed Acyclic Graph framework (e.g., IOTA and Hedera Hashgraph) that aim to provide the necessary scalability should be investigated.

REFERENCES

[1] A. Bahga and V. Madisetti, "Blockchain platform for industrial internet of things," *Journal of Software Engineering and Applications*, pp. 533–546, 2016.

[2] B. Nicoletti, "The future: procurement 4.0," *Agile Procurement*, pp. 189–230, 2018.

[3] S. Tönnissen and F. Teuteberg, "Using blockchain technology for business processes in purchasing concept and case study-based evidence," *International Conference on Business Information Systems*, pp. 253–264, 2018.

[4] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[5] X. Xu, I. Weber, M. Staples, L. Zhu, J. Bosch, L. Bass, C. Pautasso, and P. Rimba, "A taxonomy of blockchain-based systems for architecture design," *IEEE International Conference on Software Architecture (ICSA)*, pp. 243–252, 2017.

[6] D. Tapscott and A. Tapscott, "Blockchain revolution," 2016.

[7] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," 06 2017.

[8] N. Szabo, "Formalizing and securing relationships on public networks," *First Monday*, 1997.

[9] B. Vitalik, "A next generation smart contract & decentralized application platform," 2014.

[10] M. Alharby and A. van Moorsel, "Blockchain based smart contracts: A systematic mapping study," *Computer Science & Information Technology*, 2017.

[11] M. Y. Afanasev, Y. V. Fedosov, A. A. Krylova, and S. A. Shorokhov, "An application of blockchain and smart contracts for machine-to-machine communications in cyber-physical production systems," *IEEE Industrial Cyber-Physical Systems (ICPS)*, pp. 13–19, 2018.

[12] L. W. Cong and Z. He, "Blockchain disruption and smart contracts," *The Review of Financial Studies*, pp. 1754–1797, 2019.

[13] M. Kõlvart, M. Poola, and A. Rull, "Smart contracts," *The future of law and eTechnologies*, pp. 133–147, 2016.

[14] M. Essig and A. van Weele, "Strategic Procurement," 2017.

[15] R. M. Monczka, R. B. Handfield, L. C. Giunipero, and J. L. Patterson, "Purchasing and supply chain management," 2015.

[16] S. Parniangtong, "Strategic sourcing," *Supply Management*, 2016.

[17] F. Trojan and R. F M Marçal, "Proposal of maintenance-types classification to clarify maintenance concepts in production and operations management," *Journal of Business Economics*, 2017.

[18] S. Wang, J. Wan, D. Li, and C. Zhang, "Implementing smart factory of industrie 4.0: An outlook," *International Journal of Distributed Sensor Networks*, pp. 1–10, 2016.

[19] W. Bandara, S. Miskon, and E. Fielt, "A systematic, tool-supported method for conducting literature reviews in information systems," *19th European Conference on Information Systems (ECIS)*, 2011.

[20] J. Vom Brocke, A. Simons, B. Niehaves, K. Riemer, R. Plattfaut, A. Cleven, *et al.*, "Reconstructing the giant: On the importance of rigour in documenting the literature search process.," *17th European Conference on Information Systems (ECIS)*, pp. 2206–2217, 2009.

[21] J. Webster and R. T. Watson, "Analyzing the past to prepare for the future: Writing a literature review," *MIS Quarterly*, 2002.

[22] M. Travizano, C. Sarraute, G. Ajzenman, and M. Minnoni, "Wibson: A decentralized data marketplace," *CoRR*, 2018.

[23] P. Banerjee and S. Ruj, "Blockchain enabled data marketplace - design and challenges," 2018.

[24] P. Missier, S. Bajoudah, A. Capossele, A. Gaglione, and M. Nati, "Mind my value: a decentralized infrastructure for fair and trusted IoT data trading," 2018.

[25] A. Panarello, N. Tapas, G. Merlino, F. Longo, and A. Puliafito, "Blockchain and iot integration: A systematic survey," *Sensors*, 2018.

[26] N. Teslya and I. Ryabchikov, "Blockchain-based platform architecture for industrial iot," *21st Conference of Open Innovations Association*, pp. 321–329, 2017.

[27] J. I.-H. Hsiao, "Blockchain for corporate renewable energy procurement-potential for verification of renewable energy certificates," *US-China L. Rev.*, 2018.

[28] H. Subramanian, "Decentralized blockchain-based electronic marketplaces," *Communications of the ACM*, pp. 78–84, 2017.

[29] J.-S. Park, T.-Y. Youn, H.-B. Kim, K.-H. Rhee, and S.-U. Shin, "Smart contract-based review system for an iot data marketplace," *Sensors*, 2018.

[30] F. de la Vega, J. Soriano, M. Jimenez, and D. Lizcano, "A peer-to-peer architecture for distributed data monetization in fog computing scenarios," *Wireless Communications and Mobile Computing*, pp. 1–15, 2018.

[31] S. Delgado-Segura, C. Pérez-Solà, G. Navarro-Arribas, and J. Herrera-Joancomartí, "A fair protocol for data trading based on bitcoin transactions," *Future Generation Computer Systems*, 2017.

[32] A. Seitz, D. Henze, D. Miehle, B. Bruegge, J. Nickles, and M. Sauer, "Fog computing as enabler for blockchain-based IIoT app marketplaces - A case study," *5th International Conference on Internet of Things: Systems, Management and Security*, pp. 182–188, 2018.

[33] M. Klems, J. Eberhardt, S. Tai, S. Hartlein, S. Buchholz, and A. Tidjani, "Trustless intermediation in blockchain-based decentralized service marketplaces," *Service-Oriented Computing*, pp. 731–739, 2017.

[34] M. Albano, P. Sharma, J. Campos, and E. Jantunen, "Energy saving by blockchaining maintenance," *Journal of Industrial Engineering and Management Science*, pp. 63–88, 2019.

[35] T. Weingaertner, R. Rao, J. Ettlin, P. Suter, and P. Dublanc, "Smart contracts using Blockly: Representing a purchase agreement using a graphical programming language," *Crypto Valley Conference on Blockchain Technology (CVCBT)*, pp. 55–64, 2018.

[36] A. Rejeb, J. Keogh, and S. Edit, "Exploring new technologies in procurement," *Transport & Logistics: the International Journal*, pp. 76–86, 2018.

[37] B. Cha, S. Park, B. Shin, and J. Kim, "Draft design of distributed ledger based peer-to-peer business platform," *Complex, Intelligent, and Software Intensive Systems*, pp. 865–872, 2019.

[38] H. Wannenwetsch, "Integrated materials management," *Integrated Materials Management, Logistics and Procurement*, 2014.

IEEE Requesting permission to reuse content from an IEEE publication

Toward a Decentralized Marketplace for Self-Maintaining Machines

Conference Proceedings: 2019 IEEE International Conference on Blockchain (Blockchain)

Author: Daniel Miehle

Publisher: IEEE

Date: Jul 2019

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

3.3 A Blockchain-based Flexibility Market Platform for EV Fleets

This publication (Publication [C]) demonstrates the integration of EVs into flexibility markets based on a DLT platform for offering flexibility in the form of deviations from their cost-optimal charging plan.

Conference: 2020 Power Systems Conference
Number of Pages: 8
Review: Peer Reviewed (2 Reviewers)

The author of this dissertation contributed substantially to this publication, in particular concerning the conduction of the literature review, the elicitation of the business requirements, the design of the system architecture, the design and conduction of the evaluations, and writing the publication.

Goal

With an increasing number of EVs depending on electrical energy, improving power management in energy grids will be a key for future mobility. It is essential that charging stations transform into gateways for EVs to take an active role in the energy market, thus improving the balancing of the network. The goal of this DLT application was the integration and evaluation of EVs in flexibility markets based on a DLT platform.

Approach

Our approach is a conceptual implementation resulting in an application prototype. The application prototype automatically handles the trade of flexibility between EVs, Transmission System Operators (TSO), and Distribution System Operators (DSO) using smart contracts. Furthermore, we designed a flexibility trading mechanism for EVs on the distributed ledger. To evaluate our application, we developed a simulation of EVs participating in the flexibility market platform. In each simulation run, the platform performs a full market sequence of offering and buying.

Results

We conducted a case study with our industry partner to demonstrate the functionality of our DLT marketplace. The prototype demonstrated that DLTs facilitate the integration of EVs into energy markets to offer flexibility in the form of deviations from their cost-optimal charging plan. DLTs enable the cooperation between the different entities by employing smart contracts and their characteristics, such as transparency, tamper-resistance, and security. The results of our simulation showed that the transaction times of the DLT platform are within the limits of the performed market sequence.

IEEE Reuse Requirements

With regard to the IEEE reuse requirements for dissertations, the accepted and not the published version of the publication is used in the following.

A Blockchain-based Flexibility Market Platform for EV Fleets

Gonzalo Munilla Garrido[1,2], Daniel Miehle[2], Andre Luckow[2], and Florian Matthes[1]
[1]Chair for Software Engineering for Business Information Systems, Technical University of Munich, Munich, Germany
[2]BMW Group, Group IT, Munich, Germany
Gonzalo.mg.Munilla-Garrido@bmwgroup.com, Daniel.Miehle@bmw.de, Andre.Luckow@bmwgroup.com, Matthes@in.tum.de

***Abstract*—The increase of renewable energy generated in certain countries has outpaced the expansion of their power grid, causing grid congestion. Currently, grid operators use flexibility measures to counter this challenge. However, these measures struggle to cope with the growth in renewables. There are numerous proposals to improve flexibility measures using distributed energy resources such as electric vehicles (EVs). However, there is a need for a platform whereby EVs can be leveraged directly by grid operators. In answer to the decentralized quality of EVs and the requirements defined by our automotive industry partner, we propose a platform based on a distributed ledger technology (DLT). To achieve this goal, we first designed a concept for a decentralized flexibility market for the stakeholders of the ecosystem. The concept serves as the blueprint for the implementation of the platform. With the design and its implementation and simulation, we validated the use case and technical feasibility of the chosen DLT. We conclude that our prototype has the potential to allow grid operators to leverage idle EVs in aggregation to mitigate congestion.**

***Index Terms*—Blockchain, Charging Stations, Electric Vehicle, Electricity Market, Smart Contracts**

I. Introduction

The expansion of systems such as small wind turbines, photovoltaics, combined heat and power systems, and EVs on the grid distribution level supersedes the expansion rate of the power grid [1]. These distributed energy resources (DERs) [2], in combination with the introduction of renewable energy and its unpredictable nature [3], induce grid congestion [4]. Consequently, grid operators resort to more curtailment and redispatching of power plants, which lead to significant economic losses. This conundrum is commonplace in Germany and the United Kingdom [5]. Looking at Germany in particular, these losses follow an upward trend starting at €368M in 2014 and closing at €1000M in 2017 [4]-[5].

Furthermore, by 2020 the first renewable power plants will stop receiving feed-in tariffs according to the German Renewable Energy Act [6]. Thus, grid payments need to change to incentivize investors to participate as prosumers. Otherwise, renewable energy growth would suffer certain stagnation.

Increasing flexibility measures is one of the strategies to overcome these challenges [1]. Flexibility is the modification of the consumption or generation schedule of an energy resource by an external signal to provide a service [7]. It can be positive or negative depending on the direction of the energy flow and on whether it was scheduled. Flexibility is currently offered in the form of redispatching, curtailment, and demand-side management. These are controlled in existing markets and managed centrally by the same entities that oversee energy transportation, the grid operators [8]. In Germany, depending on the magnitude of the controlled area, the operators take the form of transmission system operators (TSOs) and distribution systems operators (DSOs).

These central markets do not include DERs, whose capacity could be utilized to mitigate congestion. Related work covering different approaches to include and leverage the qualities of DERs in flexibility markets have been proposed. For instance, Zadé et al. [8] provides a method to include the bids of home energy management systems in flexibility markets by introducing a more efficient pricing model. Eid et al. [9] presents an analysis of the limitations of DERs, their sources, and potential trading platforms. Minniti et al. [10] identifies key enablers, stakeholders, and design challenges for DERs to be fully utilized in a local flexibility market. These studies show positive implications on the use of DERs and provide foundational insights on their integration in the system. However, they do not thoroughly cover the inclusion of EVs, which have the potential to be used as balancing assets of renewable energy sources [11]. Additional related work aligns with this focus and presents decentralized approaches in response to the decentralized quality of EVs. For example, Papadaskalopoulos et al. [12] develops a mathematical model for a decentralized pool market mechanism. Gaeda et al. [13] presents an economic assessment resulting from the introduction of EV services at the distribution grid level, while considering grid constraints. Rassaei et al. [14] provides a technique for decentralized demand response whereby the aggregators can alter the demand of their EVs and reduce procurement costs.

However, all these papers do not present a concrete design or implementation of a decentralized flexibility market platform that allows EVs to trade their flexibility. Hence, they do not elaborate either on which technology suits the implementation. In this paper, we bridge this gap by designing a concept for a decentralized flexibility market. In Section II, the market foundations are laid, and in Section III, these foundations are used to design the market. To prove the technical feasibility of our market design, in section IV, we select a distributed ledger technology (DLT) and implement a platform capable of supporting such a flexibility market.

Section V and VI are reserved for the findings and conclusion, respectively.

The use of a DLT is based on the fulfillment of our automotive industry partner's requirements in Section II-A. A central platform would not be able to achieve the requirements of *(1) Independence* and *(5) Immutability*.

The goal of this paper is to (i) tackle the challenge of coordinating numerous EVs, (ii) connect EVs directly with grid operators, (iii) provide a model for EVs to request payment after performing a flexibility service, (iv) propose an incentive system to promote renewable energy growth, (v) enable EVs to mitigate congestion, and (vi) assess the technical and conceptual feasibility of employing a DLT in this use case.

II. Decentralized Flexibility Market Concept Foundations

Here, we describe the foundations to implement our decentralized flexibility marketplace concept.

Our design pivots around the roles of the participants and the information they exchange. Building upon this foundation, the marketplace design, and the sequence diagram are conceived in Section III.

The marketplace consists, on the supply side, of fleet operators (FOs) offering flexibility, and on the demand side, of TSOs and DSOs bidding for it. Once the auction is finished, a binding contract is created between the grid operator and the FO. The payment in exchange for the marketed offer is granted in case the FO can prove the energy transfer was conducted.

A. Requirements

We define, together with our automotive industry partners, the requirements for the successful implementation and acceptance of our decentralized flexibility market place:
(1) Independence: The demand and supply side of the market should trade without the need of an auction house or external platform providers.
(2) Ease of Access: After the verification of a market participant, the barriers of entry to connect to the platform should be low and across borders, thus shortening formal procedures.
(3) Incentive system: The platform should act as an incentive machine to endorse e-mobility. Additionally, penalties should be in place in case of misbehavior of a market participant.
(4) Audit-ability: It is essential for the traceability of energy transfers required by the ISO 15118 [15].
(5) Immutability: The transactions cannot be tampered with once they are published on the ledger.
(6) Transparency: The transactions must be readable by the participants to ensure fair trade. Transparency is enhanced by the requirement of *(5) Immutability*.
(7) Monetary Transactions: A payment system should be integrated into the platform, acting as the backbone of the incentive system.

B. Participants

The actors in this concept design have the following motivations and, or behavior:

- The motivation of the **TSO** is to tackle grid congestion events in a more cost-efficient manner. It bids for flexibility offers depending on the needs of the power grid. Additionally, the TSO acts as a verifier of market participants and smart meters (SMs). We assigned the TSO this role due to its broader impact on the ecosystem.
- The motivation of **DSOs** and modus operandi are similar to the ones from the TSO at a smaller scale. However, for this prototype, DSOs are not verifiers.
- The motivation of **FOs** is to generate revenue from idle EVs. FOs predict the behavior of their fleet and create flexibility offers for the market. If EVs comply with the specifications of the TSO, FOs are allowed to join the network. Lastly, EVs are assets with a set of instructions to execute once they are plugged into a charging station.
- The role of **SMs** is to sign an energy transfer record (ETR) after an EV has performed the flexibility offered. SMs need to be verified by the TSO to allow the decentralized market platform to verify, in turn, the validity of the signed ETR.

For our marketplace, the role of charging station operators is limited to enable energy transfers, and host verified SMs in their charging stations. Owing to the need for grid operators to match congested segments of the grid to the locations of charging stations, the latter need to be known to all market participants. Charging stations could, however, offer negative flexibility by locking the station, making them market participants with a revenue stream. This approach would incentivize charging station operators to share their location and include a verified SM. However, for this concept design, it is assumed that these operators would cooperate without being incentivized.

The business logic which runs in a DLT is called smart contract [16]. It can self-execute and allows untrusted parties to establish agreements without any trusted third party. Therefore, a smart contract can substitute the auction house of traditional market solutions. Smart contracts are fundamental for the requirement of *(1) Independence* of Section II-A. Our smart contract has different functionalities:

- The **Whitelist** contains the necessary functions to add or remove the addresses of the participants only by a verifier, which is the TSO in our concept design. Only the registered addresses can interact with the Flexibility Platform.
- The **Token** gathers the functionality necessary for the use of a token as a medium of payment.
- The **Flexibility Platform** keeps track of the offers and updates their stage depending on the information provided. At the latest stage, this contract acts as a factory for the creation of Payment Contracts with their corresponding whitelist restrictions.
- The **Payment Contract** contains the offer to be fulfilled and the functionality to unlock payment if the correct information is input.

C. Verifier

There is a need for a verifier to certify the entities whose intention is to participate in the platform. The certification is necessary to impede entities to join the network without a certified EV or without being a DSO. Additionally, yet another verifier is needed to certify that an energy transfer was successful. This verification ensures that flexibility offers are performed and reflected on the decentralized ledger. To accomplish this in our decentral market, we assign both verifiers to a TSO owned by a trusted party such as a government institution. Such is the case of e.g., Tennet in Germany [17].

The TSO registers the participants in the Whitelist if they fulfill the requirements based on the desired standards chosen by the TSO itself. Moreover, the TSO verifies whether the SMs installed in the charging stations are tamper-proof and perform measurements correctly. If the SM is compliant, the TSO includes its address in the Whitelist. We assume for the concept design that the TSO has already performed the necessary steps to perform both entity certification and energy transfer verification. This assumption is realistic in Germany, as the largest TSO conducts successful pilots in the same field of research [18].

The participation of a verifier does not compromise the requirements in Section II-A. Verifiers are not intermediaries matching supply and demand centrally. Once verified, market participants trade independently from the verifier without an auction house. Additionally, we bootstrap our system with one verifier; however, in following versions, the participants will choose other verifiers in consensus. Furthermore, these verifiers do not control the majority of the nodes forming the network infrastructure. To assure autonomy, additional DSOs and FOs would be included initially as authorities to update the new states of the platform in consensus. Thus, by replacing a single authority with a majority consensus, data immutability can be guaranteed. In comparison, a centralized system cannot satisfy *(1) Independency* and *(5) Immutability*, as the owner has the power to control trading and change the ledger.

III. Decentralized Flexibility Market Concept Design

The following section describes the design and the sequence diagram of our market.

A. Market Design

Every market requires an event sequence in place to describe the stages the offers go through. Additionally, the characteristics of the market need to be defined [19]. This section explores both aspects of our market design.

The auction is divided into three stages:

- In the **Offering** the FOs post offers containing an energy transfer with a duration of multiples of 15-min periods.
- During the **Bidding** the TSO and DSOs clear the market by bidding for the offers provided.
- In the **Contract Creation** the FOs confirm the result of the auction. This confirmation takes the form of a contract with locked payments and penalties. They may be unlocked by presenting a verified ETR after performing the flexibility offer the following day (market settlement).

Here are the characteristics of the designed market: To increase the predictability of fleets and congestion, and looking at other capacity markets [20], we have selected a **day-ahead** market type for the time of delivery. The **market-clearing rules** in the bidding process cannot allow discrimination of players with lower communication speeds, a lack of frequency trading capabilities [19], or significant computation overhead. Furthermore, decentralized technologies usually have latency, which affects the communication speed. Therefore, we set the clearing to discrete-timely to make the market democratic. Thus, the offers and bids are gathered in batches and considered equally irrespective of asynchronicity.

Additionally, there is a clear set of buyers and sellers communicating their offers and bids, which makes the market a **double-sided auction**. With respect to **pricing**, the supply side of the market chooses a price for an offer, from there, the demand side starts bidding. Lastly, any information placed in a decentralized system is visible to the participants. However, some of this information has the potential to be anonymized, which makes the information disseminated **pseudo-sealed**.

As it is a discrete-timely market, we set a time frame for each stage of an auction round. Following similar day-ahead trading markets discussed in [20]-[21], an auction period of 15 min was considered. As three stages have been defined, each is set to last 5 min. The gate closure will take place 1 min in advance, providing additional time for decision making and a buffer for all the offers and bids to be collected by the participants.

Lastly, the FOs need to include a locked penalty in the Payment Contract. The TSO or DSO with which the offer was reserved, is entitled to retrieve this penalty. The retrieval may happen 24 h after the offer should have been delivered. In the opposite case, the penalty is reverted to the FO after successfully performing an energy transfer.

Fig. 1 summarizes this section by depicting the event sequence of the market.

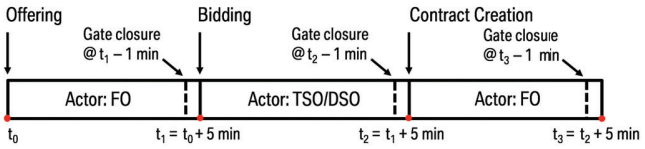


Fig. 1. Event Sequence. The three different stages of the market design and the actors and the duration for each stage may be observed.

B. Sequence Diagram

Herein, we contextualize the marketplace. Additionally, we present two sequence diagrams depicting in detail the auction process and the payment request after the energy transfer. Fig. 2 contains a diagram describing the different events an offer is subjected.

1) The FO predicts the behavior of its fleet of EVs the day ahead. Afterward, the FO calculates the flexibility

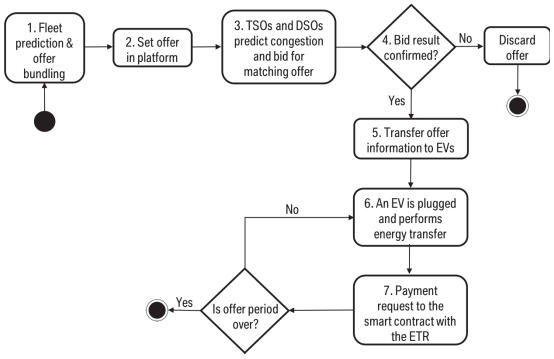


Fig. 2. Diagram of the life cycle of an offer.

offers and bundles them together according to time and location.

2) The offers are set in the platform for the TSO and DSOs to bid for them.
3) The TSO and DSOs simulate the power grid according to the supply and demand of energy reserved for the following day in conventional markets. With the results, they can predict the congestion locations in the grid. Thus, they can match their flexibility needs with the offers posted by the FOs and bid for them.
4) The FO confirms the end of the bidding, which triggers the creation of a Payment Contract.
5) The FO then transfers the necessary information from the offers to its fleet.
6) Some of the EVs (the prediction cannot be individual) are plugged to a charging station in the predicted locations and times of the following day. These EVs then perform an energy transfer according to the information dictated by their respective offers.
7) Afterward, the EV receives an ETR signed by a verified SM. With the signed record, the FO is able to request a partial payment of the total offer from the contract. The payment is partial because the offers are bundled, a group of EVs perform the complete flexibility offer together.

Fig. 3 depicts the sequence diagram modeling the auction process. *PaymentApproval* is the first function the participants trigger. They approve the Flexibility Platform to manage tokens on their behalf before engaging in an auction, which avoids later redundant function calls.

After the FO has calculated the offers, the FO inputs them in the *offering* function. This function sets the offers in the Flexibility Platform. Every time a participant triggers a function from this smart contract, the Whitelist crosschecks its address with *verifyAddress*. If the FO was verified, the offer is registered, and the TSO and DSOs receive the information via events. After the TSO and DSOs have calculated their flexibility needs, they bid for the matching offers using the function *bidding*. Subsequently, the FO finalizes the bidding with the *contractCreation* function. This function automatically triggers, in turn, a set of functions: *transferFrom*, *paymentContractCreation*, and *transfer*. In order, these collect part of the approved tokens, create a Payment Contract, and transfer the payments and penalties to it. Lastly, the Payment Contract address is broadcasted to the corresponding involved parties through events.

In Fig. 4, the sequence diagram describes the process of requesting payment. The ISO 15118 stipulates that the charging station shall verify the existence of the energy transfer contract that will be performed upon plugging in the EV [15]. Once verified, the energy transfer is performed. Subsequently, the verified SM signs the ETR. The FO then uses this ETR to request payment for having fulfilled the offer submitted the previous day in the market. The *paymentRequest* function call contains as input: the energy transferred, offer information, a random number (generated by the SM), and the signed ETR. The latter holds the encrypted components of the signature from the SM. If the EV/FO is trustworthy, the energy transferred and the offer information should be equal to the data encrypted by the signature of the SM on the ETR.

With *verifyOfferOwner*, the Payment Contract first verifies whether the address of the FO is the owner of the offer. Afterward, it verifies if the information contained in the signature of the SM is equal to the data provided by the FO. To accomplish the verification, *extractSignerFromRecord* extracts the SM's address from the encrypted components of its signature, which is only possible if the equality mentioned above exists. If successful, the Payment Contract also verifies with *verifyAddress* whether the TSO whitelisted the SM. Then, the random number is checked with *checkNonce* and compared to the ones already used by the SM (The Payment Contract itself stores these random numbers). This mechanism protects against using the same ETR in a loop to withdraw all the funds.

Lastly, *compareRecordWithOffer* verifies whether the information about the offer stored in the Payment Contract the previous day is equal to the one used as input. Without this step, FOs may withdraw payments from other offers that e.g., they were not able to fulfill. To be precise, not only the information provided by the EV has to contain information unique to the offer, but also the SM can gather equal information at the moment of the energy transfer. Otherwise, the input of the EV and the encrypted values of the ETR would never match. However, the SM is oblivious to the market. To solve this problem, the SM signs together with the energy transfer value: the date, start time, location, type of flexibility offered, and the address of the FO. The EV can provide the latter. Thus, the Payment Contract can verify that the SM signed the appointed offer.

It is only after these verifications that the Payment Contract triggers *calculatePayment* and *recalculatePenalty*. These amounts are then transferred to the FO. Finally, the TSO or DSO, which also owns the offer, can use *enforcePenalty* to withdraw what is left from the payment and penalty only after a time limit has been reached.

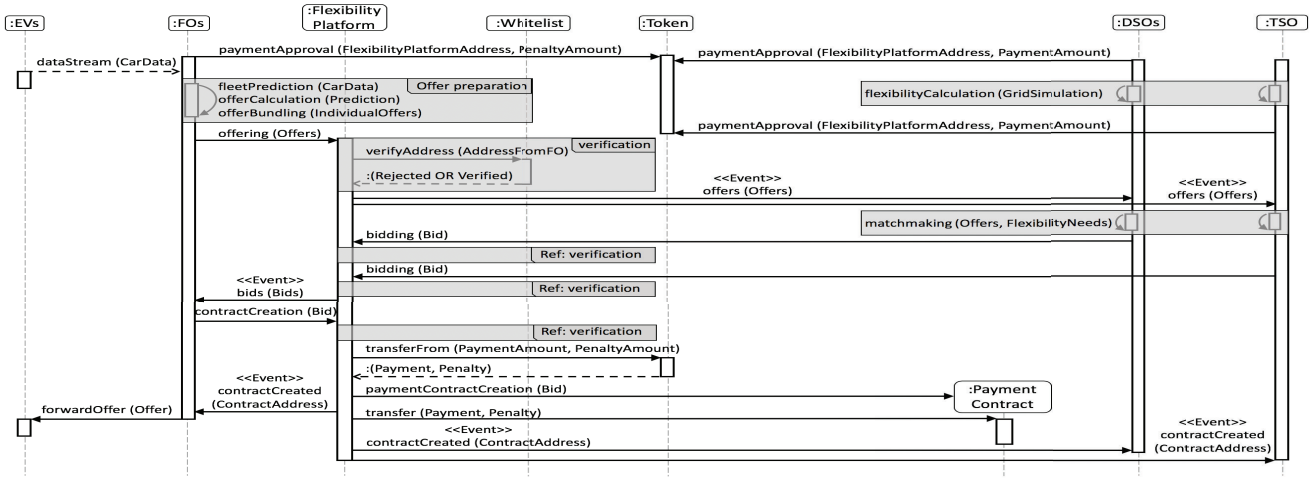


Fig. 3. Sequence diagram depicting the interaction between the participants involved in the auction and the decentralized flexibility market platform.

IV. Technical Implementation

Here, we first select the technology for the implementation. Afterward, we delve into the implementation and simulation of the developed platform.

A. Technology

This section presents the technology selected for the implementation according to the use case and the requirements of Section II-A.

In our system, there is a need for *(1) Independence*, an *(3) Incentive System* [6], *(4) Audit-ability* for tracking energy transactions [15], *(5) Immutability*, and *(6) Transparency* for fair trade. Additionally, there is a reluctance of energy players to yield responsibility to one another. Consequently, we selected a DLT to build a platform to host such a market. A DLT is a distributed database whose state is replicated, synchronized, and agreed upon by a consensus algorithm run by participants of a peer-to-peer network. Among the most prominent ones (Hyperledger Fabric, Ethereum, Iota, and Corda) [22]-[25], we selected the blockchain-based system Ethereum. A decision based on its *(2) Ease of Access*, built-in *(7) Monetary Transactions*, flexible business logic, and disintermediation capabilities. Additionally, we find crucial for implementation its applicability both in a public and private setting, its advanced development in smart contracts, its future projection, large developer community, and development tools [23], [26]-[27],

Nonetheless, before committing to the use of Ethereum, we must assess whether its throughput is enough to handle the transactions from the 1000 EVs of our simulation in Section IV-C. According to Fig. 6, a maximum of 633 transactions can be sent during the offering stage of the market sequence. This upper limit requires a minimum of 2.64 transactions per second (tps) during the 4-min period. For comparison, we selected the throughput of blockchain in [28], averaging to 5.5 tps. Therefore, Ethereum is a feasible technology to implement our prototype.

The last point to be addressed is whether the market runs in the public or a consortia blockchain. Consortia blockchains are free of fixed gas limits, transaction fees, and have less legal implications. A consortia blockchain platform thus increases the chances of the hereby presented solution to be implemented in the real world.

B. Implementation

In this section, we focus on the implementation of the concept design described in Section III.

These are the operational assumptions:

- No malicious entities participate in the market or attack the underlying network. They will be considered in future work.
- Charging stations are considered ubiquitous, and their locations are known.
- Fleet and power grid predictions and the subsequent flexibility calculations are finished.
- The verification of the participants has taken place.
- The TSO and DSOs perform the matchmaking between the offers and their flexibility needs.

Fig. 5 depicts the components and parties the implementation of the concept design would have in actuality. One may observe that the driver and the EV are unaware of the process taking place in the blockchain as they do not host an Ethereum node. The interaction of the driver is confined by the capabilities of the Human Machine Interface. Additionally, he/she does not participate in the market. Furthermore, the EVs are merely an asset used by the FO to perform the dictated flexibility offers. Instead, the FO hosts the node, as the owner of the fleet and participant of the market. Thus, we achieve the same functionality without investing in dedicated hardware in the EVs, such as a wallet or a light node.

The simulations are carried out to imitate the interaction with the platform shown in Fig. 5. The majority of the code has been programmed in Python, leaving JavaScript for event management, and Solidity for smart contract programming.

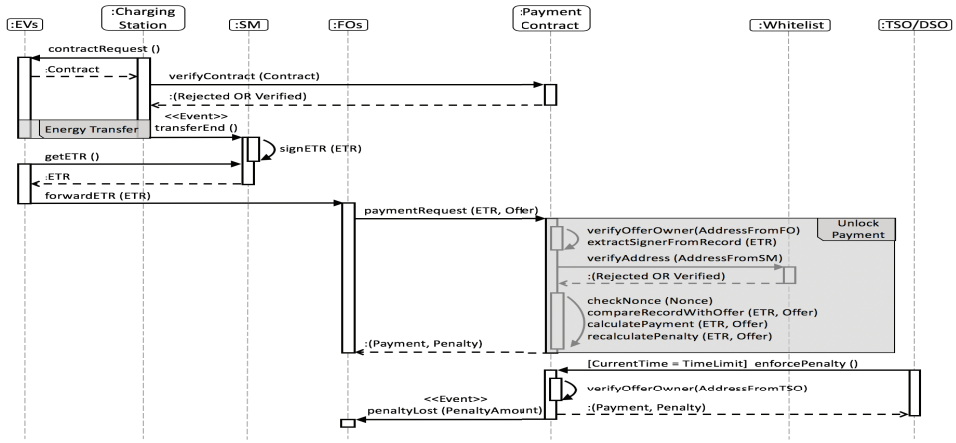


Fig. 4. Sequence diagram depicting the process followed by the FO to unlock its due payment from the Payment Contract.

To simplify the execution, only two trading participants were recreated: a TSO and an FO. Adding more would only increase the number of transactions, which we studied by increasing the number of locations instead.

In order to set up a blockchain, two Geth nodes were installed in the same machine and set to listen in two different ports. The consensus protocol used is Proof of Authority, and the block time is set to 15 s.

Ultimately, the data used to create the bundled offers comprises 1000 EVs and their offers for several 15-min periods of one day. These offers consist of: positive or negative capacity, location, date, time, and price.[1] This data, however, lacks the state of charge of each vehicle for every 15-min period. Therefore, with the information given, an EV can only make one offer per day. If the FO were to commit two offers on the same day without knowing the estimated state of charge of the EV, the FO might be proposing conflicting offers. Conflicts need to be avoided, and it led to choose the most profitable bundled offers amongst the conflicting ones. Using a data set with state of charge information is scheduled for future work.

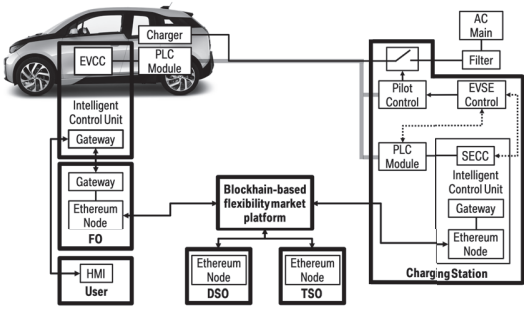


Fig. 5. Implementation architecture adapted from the ISO 15118-1 [15]

[1]These values come from the results achieved in publication [8], which was kindly shared by its authors Zadé et al.

C. Simulation

The simulation scenarios and results are discussed in the following paragraphs. In summary, a simulation consists of running one market sequence cycle, the one presented in Fig. 2. The purpose of these simulations is to prove that the conceptual design described in Section III is technically feasible, given different scenarios. Additionally, different data was measured to draw new insights.

In total, 21 successful simulations where conducted. These scenarios are based on four independent variables:

1) A constant of 1000 EVs
2) Number of locations: 25, 50, 75, 100, and 500.
3) The number of offers rejected by the TSO: 50%, 75%, and 90%.
4) The number of offers the FO ultimately performs: 50%, 80%, and 90%.

It is important to remember there is only one bidder, the TSO, in this simulation. Thus only one bid per offer is registered. Moreover, in reality, the TSO needs a simulation of the power grid to find congested segments. This grid simulation is beyond the scope of this paper. Thus in our system, the TSO selects the offers from the FO randomly.

Part of the results is illustrated in the next three figures. In Fig. 6, one may observe a monotonic relationship between the number of locations and the number of potential offers from a set of 1000 EVs. This illustrates the aggregation of offers. The more locations are considered for the same amount of vehicles, the less single offers can be bundled into one, increasing the number of total offers.

For the next two figures, the number of offers ultimately performed by the FO has been fixed to 80%. In Fig. 7 and Fig. 8, there is a consistent downward trend, as the more offers are rejected by the TSO, the less power is transferred and revenue is generated, respectively. The relationship between power transferred and the number of locations is fuzzy in Fig. 7. However, in Fig. 8, there could be a direct relationship between the number of locations and the revenue generated. Knowing that fewer locations lead to more aggregation, this

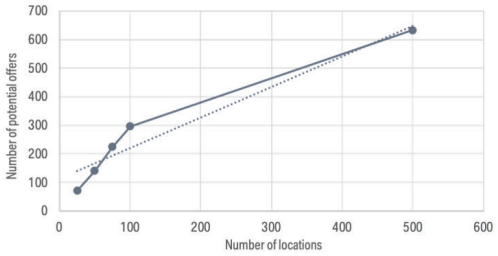


Fig. 6. Number of offers created as a function of the number of locations.

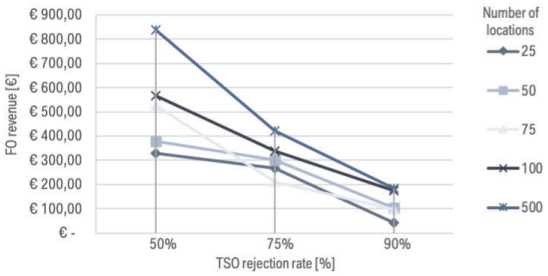


Fig. 8. Revenue generated by the FO. 80% of the offers were performed.

indicates that aggregation is detrimental for wealth generation and thus challenges the idea of bundling offers. While this assertion is correct in this implemented concept, in actuality, a minimum required amount of kW for an offer would exist, forcing bundling to be necessary to reach that minimum.

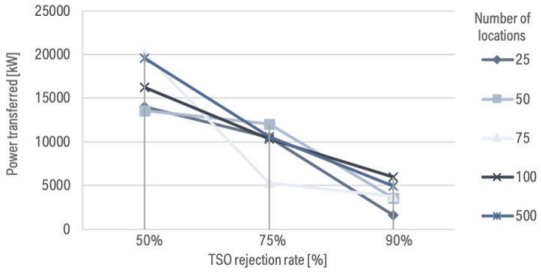


Fig. 7. Power transferred by the FO. 80% of the offers were performed.

Furthermore, the phenomena in Fig. 8 can be explained. The fewer locations there are, the higher is the number of EVs in a bundle. Because one EV can only make one offer per day given the data limitations, it is less likely that its best offer is included in the bundle. Thus, there are fewer vehicles that include their best offer, decreasing total revenue. We can forecast that if the EVs can perform as many offers as their state of charge allows, then the amount of revenue made by the fleet of EVs would be less dependent on the number of locations.

This paragraph revisits the potential issue mentioned in Section IV-A concerning tps. The transaction speed achieved in this implementation with Ethereum ranges between 20 to 25 tps for offering, bidding, and contract creation and from 14 to 16 tps for payment requests. Looking at Fig. 6, the maximum number of transactions sent was 633, which took 24.78 s (25.24 tps). This duration is under the market sequence of 4 min, leaving time for sending another 5424 transactions.

V. Findings

This section discusses the insights and encountered challenges gathered during the implementation of the concept design. The most important one being the limited space Ethereum allows for a string variable. Because the duration of an offer can hold multiple 15-min intervals, the information about positive or negative power and price may become too large. Thus, we hashed the strings and set the hash in the platform as the offer. The readable text was sent by other means to the other participant, who crosschecked the commitment in the blockchain with the hash of the readable version. Another solution is to limit the duration of an offer to a 15-min period. This way, the length of the string would be shortened and fixed (with padding). However, as we aimed to maximize revenue, we aggregated offers spanning longer than 15 min. Thus the latter solution was not chosen, due to the limitation on the number of offers per day.

Furthermore, in case there were malicious participants, the stage closure might be compromised. We rely on the auditability of the blockchain to avert this potential threat. A set of verifiers could penalize participants or grade their reputation if the registered transactions were not within the time limits.

Concerning flexibility, we can forecast that the EVs will not provide the majority of the negative flexibility offers, as charging stations can simply lock their outlets.

Moreover, there are inherent uncertainties in the use of emerging technologies such as Ethereum. Even though we have developed a working prototype, employing a new technology is subject to changes in the future. Furthermore, regulations such as GDPR [29] may hinder the further development of projects like the one hereby presented. These uncertainties and factors external to the core functionality of our blockchain-based flexibility market are reserved for future work.

Lastly, despite proposing a government-owned verifier, centralizing critical processes decreases the trustworthiness of the platform. Incidentally, the institution is a market participant. This conundrum is solved by a vote with several trusted entities to verify participants in consensus, e.g., with additional TSOs or DSOs. Concerning the SM verification, the ones within the EVs may be used, however, only if the flexibility offers come from an entity different from the Original Equipment Manufacturer of the EV, e.g. by an independent FO.

VI. Conclusion and Future Work

With this paper, we define a concept design and a set of requirements for a decentralized flexibility market platform for

EVs. In order to prove its technical feasibility, we implemented a prototype. Here, EVs can offer their flexibility, create a contract with the interested party, and use this contract to unlock payment once the energy transfer was performed. Moreover, a set of successful tests was completed to validate the proper execution of the various functions that conform the system. Additionally, the resulting data provides the foundation for a future economic assessment and a comparison of later iterations of the platform. As a result, we can confirm that the platform is functional, and the time duration of the transactions are within the limits of the designed market sequence.

The research conducted in terms of technology and concept design indicates that a flexibility marketplace is a valid use case for a blockchain application. Furthermore, our automotive industry partners found business potential in our marketplace based on the upcoming trend of e-mobility. Lastly, our platform proves that a blockchain-based decentralized marketplace for flexibility offerings is technically feasible and validates the use of a DLT.

In response to the set of goals of Section I, we conclude that grid operators could directly leverage and coordinate idle EVs in aggregation to improve grid balancing and thus avoid congestion. By extrapolation, this implies that other DERs could also market services to grid operators for a profit using another iteration of our market. This concept design could constitute the incentive system so needed by prosumers in the near future [6].

In future research, the implementation will hold a more significant number of nodes and participants. This arrangement will allow examining if the throughput of the network decreases. Consequently, we will assess whether the duration of an auction cycle should be increased from 15 min to e.g., 30 min or 1 h. Moreover, we will carry out performance and stress analysis. Furthermore, we will implement our concept in a real scenario by utilizing a charging station, a SM and an EV.

Additionally, due to the nature of Ethereum, the platform could be re-worked to be launched in the public blockchain for comparison to the private version. Lastly, with the prospect of dealing with thousands of EVs, future work should investigate other distributed ledger solutions focusing on scalability (e.g., Hyperledger Fabric, Corda, and IOTA). With all the above, we will address the reservations from our industry partners, as they pointed out the need for robustness, scalability, and throughput.

REFERENCES

[1] Agora Energiewende. Energiewende 2030: The big picture. pp. 18, 36, Feb. 2015.

[2] A. Omu, R. Choudhary, and A. Boies. Distributed energy resource system optimisation using mixed integer linear programming. *Energy Policy*, 61:249, Oct. 2013.

[3] E. Bayer. Report on the german power system. Agora Energiewende. pp. 32, Oct. 2015.

[4] Bundesnetzagentur. Monitoring report 2018. pp. 173, May 2019.

[5] M. Joos and I. Staffell. Short-term integration costs of variable renewable energy: Wind curtailment and balancing in britain and germany. *Renewable and Sustainable Energy Reviews*, 86:47, Jan. 2018.

[6] D. Nelson, M. Huxham, S. Muench, and B. O'Connell. Policy and investment in german renewable energy. A CPI report, Climate Policy Initiative, pp. 77, Apr. 2016.

[7] D. Trebolle and R. Otter. Flexibility and aggregation, 2014.

[8] M. Zadé, Y. Incedag, W. El-Baz, P. Tzscheutschler, and U. Wagner. Prosumer integration in flexibility markets: A bid development and pricing model. In *2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2)*, pages 1–2, Oct. 2018.

[9] C. Eid, P. Codani, Y. Perez, J. Reneses, and R. Hakvoor. Managing electric flexibility from distributed energy resources: A review of incentives for market design. volume 64, pages 242 – 246, 2016.

[10] S. Minniti, N. Haque, P. Nguyen, and G. Pemen. Local markets for flexibility trading: Key stages and enablers. *Energies*, 11(11), Nov. 2018.

[11] W. Kempton and J. Tomić. Vehicle-to-grid power implementation: From stabilizing the grid to supporting large-scale renewable energy. *Journal of Power Sources*, 144(1):291, 2005.

[12] D. Papadaskalopoulos and G. Strbac. Participation of electric vehicles in electricity markets through a decentralized mechanism. In *2011 2nd IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies*, pages 2–4, Dec. 2011.

[13] A. Gadea, M. Marinelli, and A. Zecchino. A market framework for enabling electric vehicles flexibility procurement at the distribution level considering grid constraints. In *2018 Power Systems Computation Conference (PSCC)*, pages 3–4, Jun. 2018.

[14] F. Rassaei, W. Soh, and K. Chua. Distributed scalable autonomous market-based demand response via residential plug-in electric vehicles in smart grids. *IEEE Transactions on Smart Grid*, 9(4):3281–3284, Jul. 2018.

[15] ISO 15118-1:2019. Road vehicles – Vehicle to grid communication interface – Part 1: General information and use-case definition. Standard, International Organization for Standardization, Geneva, Switzerland, Apr. 2019.

[16] N. Szabo. Smart contracts: Building blocks for digital markets. *Extropy: Journal of Transhumanist Thought*, (16), 1996.

[17] Tennet. Tennet group legal overview. [Online]. Available: https://www.tennet.eu/fileadmin/user_upload/Company/Profile/Organogram_group_legal_overview_UK_DEC2018_01.pdf. [Accessed on 30 10 2019].

[18] F. Verdeuzeldonk. Continuing with blockchain after successful pilots. Tennet. [Online]. Available: https://www.tennet.eu/news/detail/tennet-continuing-with-blockchain-after-successful-pilots/, Jan. 2019. [Accessed on 30 10 2019].

[19] W. El-Baz, P. Tzscheutschler, and U. Wagner. Integration of energy markets in microgrids: A double-sided auction with device-oriented bidding strategies. *Applied Energy*, 241:628, 2019.

[20] Frontier Economics. Overview of european electricity markets, Feb. 2016.

[21] A. Reuter and S. Breker. C/sells: Netze und märkte verbünden, 2018.

[22] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, and A. De Caro et al. Hyperledger fabric: A distributed operating system for permissioned blockchains, Apr. 2018.

[23] V. Buterin. A next generation smart contract decentralized application platform. Ethereum Project White Paper, Nov. 2013.

[24] S. Popov. The tangle, Apr. 2018.

[25] R. G. Brown. Corda: An introduction, May 2018.

[26] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press, Princeton, NJ, USA, 2016.

[27] C. Dannen. *Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners*. Apress, New York, NY, USA, 2017.

[28] N. El Ioini and C. Pahl. A review of distributed ledger technologies. In *On the Move to Meaningful Internet Systems. OTM 2018 Conferences*, page 287, Cham, 2018. Springer International Publishing.

[29] Regulation (EU) 2016/679. Protection of Natural Persons with Regard to the Processing of Personal Data and on the Free Movement of Such Data, and Repealing Directive 95/46/EC. European Parliament and Council, Apr. 2016. Luxembourg: Office for Official Publications of the European Communities.

IEEE Requesting permission to reuse content from an IEEE publication

A Blockchain-based Flexibility Market Platform for EV Fleets

Conference Proceedings: 2020 Clemson University Power Systems Conference (PSC)

Author: Gonzalo Munilla Garrido

Publisher: IEEE

Date: March 2020

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW

3.4 Starling: A Blockchain-based System for Coordinated Obstacle Mapping in Dynamic Vehicular Environments

This publication (Publication [D]) combines a DLT network with a distributed database to perform coordinated obstacle mapping in an open vehicle-to-vehicle (V2V) platform.

Conference: 2020 Hawaii International Conference on System Sciences (HICSS)
Number of Pages: 10
Review: Peer Reviewed (3 Reviewers)

The author of this dissertation contributed substantially to this publication, in particular concerning the development of the idea and approach, the conduction of the literature review, the elicitation of the business requirements, the design of the system architecture, the design and conduction of the evaluations, and writing the publication.

Goal

Today's vehicle-to-vehicle (V2V) solutions are not capable of ensuring the authenticity of safety-critical vehicle and traffic data. However, this is becoming necessary as more and more vehicles communicate with each other. The goal of this DLT application was to build an open V2V network that allows vehicles to know of obstacles in their environment without being able to detect them with their sensors.

Approach

We implemented a DLT application prototype that uses a combination of a public blockchain protocol and a distributed database to enable vehicles to share data for coordinated obstacle detection and mapping in a secure and verifiable manner. Each obstacle is represented through a unique digital identity on the distributed ledger. Moreover, we developed reliable services for the detection and mapping of obstacles, the verification of obstacles, and the detection of duplications and faulty obstacle data. To evaluate our application, we implemented a simulation of the platform using a public Ethereum network and the InterPlanetary File System (IPFS).

Results

We realized a conceptual implementation to demonstrate the feasibility and value of our open vehicle-to-vehicle (V2V) platform for coordinated obstacle mapping to our industry partner. We implemented reliable services to detect, verify, and match obstacles preventing duplications and faulty obstacle data. The results of our simulation showed that the use of a public blockchain protocol (i.e., Ethereum) combined with a distributed database (i.e., IPFS) is a feasible approach for creating an open vehicle-to-vehicle platform, that provides high availability, data integrity, security, and traceability. To enhance data privacy, we recommend using one-time addresses per-trip or per-event.

Starling: A Blockchain-based System for Coordinated Obstacle Mapping in Dynamic Vehicular Environments

Daniel Miehle
Technical University of Munich
daniel.miehle@tum.de

Andreas Pfurtscheller
Technical University of Munich
andreas.pfurtscheller@tum.de

Bernd Bruegge
Technical University of Munich
bruegge@in.tum.de

Abstract

Current Vehicle-to-Vehicle solutions cannot ensure the authenticity of safety-critical vehicle and traffic data. Moreover, they do not allow malicious vehicles to be detected and eliminated. However, this is becoming mandatory, as more and more vehicles are on the road and communicating with each other. We propose a system called Starling, which focuses on trusted coordinated obstacle mapping using blockchain technology and a distributed database. Starling enables vehicles to share detected obstacles with other vehicles in a secure and verifiable manner, thus improving road safety. It ensures that data was not manipulated, changed, or deleted and is based on an open protocol so that vehicles can exchange data regardless of their manufacturer. In a case study, we demonstrate how a consensus is reached among vehicles and conduct a comprehensive evaluation of the Starling system using Ethereum and the InterPlanetary File System.

1. Introduction

The world is becoming increasingly interconnected, and so are vehicles. In recent years, science, industry, and governmental institutions have explored approaches to implementing reliable communication between both, vehicular and non-vehicular systems, known as Vehicle-to-Everything (V2X) communication [1]. V2X communication is paving the way for the internet of vehicles, which allows for the gathering, processing, and exchange of information pertaining to vehicles, infrastructure, and their environment. Thereby, Vehicle-to-Vehicle (V2V) communication is one of the most important types of communication with regard to road safety and efficient traffic flow in general.

Research on this topic dates back to the early 1970s, when Rosen et al. [2] proposed a routing system for Vehicle-to-Road infrastructures. In recent years, interest in this subject has increased, driven in particular by the advances in autonomous driving [1]. Autonomous driving systems are designated to make decisions on the basis of the data available to them. This emphasizes the relevance of this topic for the future of connected and autonomous vehicles. Access to more information about the environment enables autonomous vehicles to take early actions against imminent hazards out of sight [3]. Hence, Vehicle-to-Vehicle networks such as the Vehicle Ad-hoc Network (VANET) aim to enhance the visibility of vehicles in situations that cannot be detected by sensors such as cameras. However, current Vehicle-to-Vehicle networks are unable to detect and eliminate malicious vehicles [4] that have the potential to cause accidents, especially in highly dynamic vehicle environments. In addition, the solutions lack openness, security, and data protection and are highly centralized, so that entire systems can fail in the event of a failure. This is unacceptable for safety critical systems of vehicles as the exchange of vehicle and traffic data between them and other services cannot be maintained.

The Starling system presented in this paper attempts to address these challenges by providing trusted vehicle and traffic data for coordinated obstacle mapping using a distributed, peer-to-peer database and blockchain technology for decentralized and verifiable data storage. The goal of Starling is to build an open Vehicle-to-Vehicle network that offers better visibility of obstacles and makes this visibility more secure and tamper-proof for all involved.

This paper is structured as follows. In Section 2, we present the foundations of obstacle detection and mapping, blockchain technology, and distributed databases on which the Starling system is based and related works. We formalize the requirements and architecture of the Starling system in Section 3. In a case study, in Section 4, we describe the the implementation of the Starling prototype. The Starling prototype is evaluated regarding performance and scalability and the results are discussed in Chapter 5 and 6, respectively. Chapter 7 concludes the paper by summarizing the contributions and future work.

2. Foundations and Related Works

In the following section, we describe the foundations of our approach and review related works to embed our approach in the scientific background.

2.1. Obstacle Detection and Mapping

Research in the field of obstacle detection for vehicles was carried out as early as the 1980s and 1990s, long before the advent of autonomous driving. Articles from this period (see [5, 6, 7]) mainly focus on obstacle detection for collision and obstacle avoidance without the exchange of obstacle data between vehicles. With the emergence of high-resolution cameras and enhanced sensor technology such as LIDAR, obstacle detection techniques and algorithms have improved, increasing their accuracy and reliability [8, 9, 10]. Wireless communication makes it possible to share and map information on obstacles detected by individual vehicles so that vehicles cannot only predict the trajectory of moving obstacles [11], but also extend the field of vision of any vehicle [12, 13].

2.2. Distributed Ledger Technology

Distributed Ledger Technology (DLT) extends the concept of distributed, peer-to-peer databases by including features such as data immutability, fair access, transparency, and the verifiability of transactions. Following [14], a distributed ledger is a distributed data structure whose entries are digital records of actions written by the participants of a DLT system after reaching a consensus on the validity of the entries.

One type of DLT is blockchain technology, which stores entries in a linear growing chain of blocks that are secured using cryptography [15]. The data structure of a blockchain is an append-only linked list, which includes a total order of its entities, starting with the so-called Genesis block [16]. Each subsequent block contains a cryptographic hash created using a uniform hash function of the block itself and the previous block, thus linking both blocks. This procedure ensures the immutability of transactions stored in a blockchain, since all subsequent blocks would have to be changed and hashed again to change only one transaction within a block. Verifying hashes is a relatively cheap process, which is why the blockchain can be easily verified by tracing the hashes of each block back to the Genesis block [16]. A transaction can contain any type of data, ranging from cryptographically signed financial transactions, to hashes of digital assets, and Turing-complete executable programs [14].

Blockchain networks can be either permissionless or permissioned. In permissionless blockchains, each participant can initiate transactions, perform mining, and create smart contracts. In contrast, not all participants in permissioned blockchains are allowed to execute all operations. The consensus mechanism ensures, that the nodes of the blockchain network are consistent. It allows the participants of the network to decide on the validity of entries, preventing double-spending and sybil attacks. [17]. Double-spending describes the use of a single asset twice (e.g., one Bitcoin). Sybil attacks address attacks that use fake identities to gain the majority in the system in order to inject faulty information into the network [14]. The consensus mechanism ensures that participants in the consensus process behave honestly and reliably, as it would be more effort in economic terms to do the opposite. [18]. The most common consensus mechanism for public blockchains (e.g., Bitcoin [16], Ethereum [19], is the Proof-of-Work consensus, which utilizes the processing power of computers. Proof-of-Work is the solution to a mathematical puzzle (i.e., mining), which is easy to verify, but solving is both difficult and takes effectively random time. Other consensus mechanisms include Proof-of-Stake (PoS), where evidence is given by providing economic power, and Proof-of-Authority (PoA), where mining is performed by trusted and pre-defined nodes.

2.3. Smart Contract

A smart contract is a computerized transaction protocol that facilitates, executes, and enforces the terms of a contract between untrusted parties without the involvement of a trusted third party [20, 21]. By using business logic implemented in smart contracts, it is possible to access the distributed ledger, the processing power of the system, and its storage [22, 18]. This allows to reduce errors, fraud, and verification time and costs, and to automate process executions [21, 23, 24]. Ethereum has established this concept of distributed computing, which clearly differentiates it from Bitcoin [14, 19, 22].

2.4. InterPlanetary File System

The InterPlanetary File System (IPFS) is a distributed, peer-to-peer file-sharing system that combines a distributed hash-table, an incentivized block exchange, and a self-certifying namespace in order to connect all computing devices with the same system of files [25]. The advantages of IPFS are no single point of failure and nodes do not need to trust each other.

2.5. Related Work

In literature, blockchain-based systems focusing on different aspects of Vehicle-to-Vehicle communication have been proposed.

For instance, Onishi [4] provides a report on the advantages and limitations of Vehicle Ad-hoc Networks and blockchain-based systems for Vehicle-to-Vehicle communication. Rowan et al. [26] propose an inter-vehicle communication by using a blockchain-based public key infrastructure that enhances interoperability between untrusted vehicles, for example, for platooning vehicles. Buzachis et al. [27] introduce an intersection management system to manage negotiated agreement between vehicles crossing through an intersection using blockchain technology and smart contracts. Singh and Kim [28] present an intelligent vehicle data sharing system using a custom consensus mechanism called Proof-of-Driving. Rathore et al. [29] propose the TangleCV, a decentralized solution for secure data sharing and recording for connected vehicles using a directed acyclic graph. Ramachandran et al. [30] introduce MOTIVE, a decentralized framework that allows vehicles to make peer-to-peer micropayments for data, compute and other services obtained from other vehicles or road side infrastructure enabling autonomous operation and trusted interactions between vehicles and nearby entities. Hewlett Packard Enterprise and Continental [31] announced a blockchain-based data monetization platform for sharing vehicle and traffic data. The platform is designed to allow for new digital services that improve security while allowing manufacturers to monetize data. In addition, there are various articles addressing reputation and liability management in vehicular environments (see [32, 33, 34]) and Vehicle-to-Grid applications for peer-to-peer payments (see [35, 36, 37]) based on blockchain technology.

However, the proposed solutions do not investigate the feasibility of blockchain technology and distributed databases for coordinated obstacle mapping.

3. Starling

In this section, we present the idea of establishing a distributed, peer-to-peer system for coordinated obstacle mapping in dynamic vehicular environments.

3.1. Problem

Vehicle-to-Vehicle networks such as VANET aim to enhance the visibility of vehicles in situations that cannot be detected by sensors. It is essential that obstacle mapping data come in the correct order and have not been manipulated or deleted, which could lead to accidents. In addition, current Vehicle-to-Vehicle solutions lack openness, security, and privacy and do not allow malicious vehicles to be detected and eliminated [4]. Only if this is ensured can vehicles improve their systems by using verified obstacle mapping data.

3.2. Vision

The vision for Starling is to provide verified and trusted vehicle and traffic data for coordinated obstacle detection that can be used by vehicles. By using blockchain technology and distributed databases, Starling aims to provide greater visibility of obstacles and make this visibility more secure and tamper-proof for all stakeholders by verifying and matching obstacles, which prevents data redundancy and double entries. Vehicles would know of verified obstacles in their environment without being able to detect them with their own sensors. For example, autonomous driving systems can consider obstacles much earlier in order to avoid accidents and make road traffic more efficient. In addition, Starling introduces the property reputation, which is an indicator of how trustworthy a vehicle is when reporting obstacles and is increased as other vehicles recognize the same obstacles.

3.3. Requirements

In this section, we describe the functional and the non-functional requirements of the Starling system. We start with defining the functional requirements of the Starling system, which aim to describe the interactions between the system and its environment independent of its implementation. The environment includes users and other external systems Starling interacts with.

1. **Provide Obstacle Repository**: The system must enable vehicles to persist mapping data of detected obstacles for coordinated obstacle mapping. This data represents obstacles in the immediate neighborhood of the detecting vehicle.

2. **Avoid Duplication**: The system must ensure that no duplicates of obstacles are persisted in the system, i.e. in case two vehicles detect the same obstacle, there should be a unique entry in the repository.

3. **Detect Faulty Obstacle Data**: The system must be able to detect manipulated or deleted obstacle mapping data. Incorrect data stored in the repository should be detected and not accepted as valid. Moreover, vehicles should

have the property reputation, which indicates the trustworthiness of a vehicle.

4. **Provide Data Access**: The system must provide an interface for clients to retrieve obstacle mapping data from the repository, to retrieve traffic data, or to detect obstacles out of sight.

5. **Provide Authority Access**: The system must ensure that authorities such as police, judiciary, and insurance companies are able to comprehend decisions about vehicles based on the coordinated mapping data.

6. **Provide Traceability**: The system must ensure the complete traceability of all actions in order to reconstruct traffic accidents.

Next, we define the non-functional requirements, which are so-called quality requirements.

7. (*Supportability*) **Open Platform**: The system must be an open platform, which means that the barriers to entry for both, car manufacturers and users must be low. The open platform must create transparency for all stakeholders involved.

8. (*Reliability*) **High Availability**: The system must be available without downtime as it runs in transport environments. A failure is unacceptable.

9. (*Reliability*) **Data Integrity**: The data in the system must be stored so that it cannot be manipulated. For example, traffic incidents must be comprehended when they occur. Therefore, the original data associated with such an incident must be accessible.

10. (*Usability*) **Protect Privacy**: The system must ensure that the tracking of vehicles and their owners should not be possible for anyone other than the vehicle owner and entitled authority actors.

11. (*Performance*) **Low Latency**: The system should be able to store and retrieve obstacles with low latency, as the environment including obstacles changes rapidly due to the high dynamics of traffic. The storing of obstacles should not exceed 500 milliseconds. The verification and retrieving of obstacle data should not exceed 1 second.

12. (*Performance*) **High Throughput**: The system should be able to handle a high data throughput. When scaling the system with many vehicles, the data throughput (i.e., the number of transactions) increases linearly with it.

3.4. Analysis

The analysis object model of the Starling system is derived from the identified requirements as depicted in Figure 1. The main actors of the Starling system are `Vehicles`, the vehicle owners, and authorities such as police officers, which can interact with the system by means of the `VehicleClient` and `AuthorityClient`, respectively. Their abstraction is combined into the `Client` superclass, which provides shared functionality such as retrieving obstacles. The `VehicleClient` allows `Vehicles` to access the `Repository` in order to store and retrieve obstacles. It facilitates the use of the `VehicleIdentifier`, which can be required by the authorities during investigations. The `AuthorityClient` enables authorities to view mapped obstacles and general traffic information.

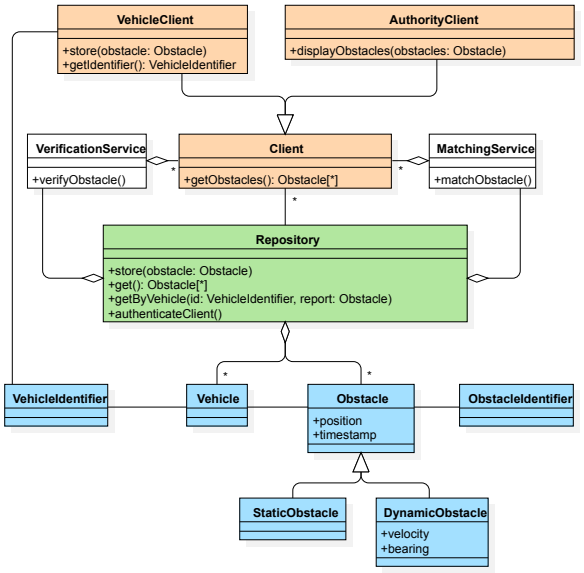


Figure 1. The Analysis object model depicts the classes derived from the requirements. The repository is the key component of the Starling system, containing entity objects like vehicles and obstacle reports. The vehicle and authority clients enable the access to obstacles stored in the repositories.

The core function of the system is to store obstacles in the `Repository`, where the obstacle mapping data is persisted. It allows to read and write obstacles, to register vehicles, and it provides operations such as user authentication.

An `Obstacle` is initially identified and reported by the sensors of a `Vehicle`. An obstacle can be either static or dynamic and consists of a timestamp and a position, which describes the location where

it has been detected by a vehicle. In addition, a dynamic obstacle contains information on the velocity and bearing of the physical obstacle represented. Starling assigns the sensor readings to obstacles already reported by other vehicles so that no duplication of physical obstacles occurs. This association is initiated by the `VehicleClient` and performed by the `MatchingService`. The `MatchingService` is called with information about the detected obstacle and reacts with either a matched or a new obstacle. After obstacles are retrieved from the `Repository`, the `VerificationService` verifies them before further processing.

3.5. Architecture

Starling is decomposed into six independent subsystems. These subsystems are grouped into three hierarchical layers, resulting in an open layered architecture. An additional subsystem indicates services provided by the vehicle's autonomous driving system. We explain the individual layers from bottom to top.

The **Verification Layer** provides `ObstacleVerification` services to the upper layers and does not depend on any other layer. The subsystem `VerificationStore` is derived from the `VehicleRepository`. Since the `Verification Store` does not contain the `ObstacleRepository`, we preserve the relationship between the classes `Vehicle` and `Obstacle` from the analysis object model. Therefore, we introduce the `HashRepository` to our subsystem, which contains permanent links to the `Obstacles` stored in the `ObstacleRepository` subsystem in the layer above.

The **Obstacle Layer** provides services for storing and retrieving obstacle data as well as matching obstacles. It encapsulates the `Obstacle Store` that contains the `ObstacleRepository` with its obstacle mapping data and the `ObstacleMatching` subsystem accessible using the `Obstacle API`. They use the services of the Verification layer below to verify the obstacle data they are working with.

The **Client Layer** contains the client applications `Vehicle Node` and `Authority Node`, which provide system access for the actors `Vehicle` and `Authority`, respectively. They use services provided by subsystems from both, the Obstacle and the Verification Layer.

4. Case Study

Here, we describe the implementation of the Starling prototype using Ethereum and IPFS and present a detailed analysis and discussion of the prototype.

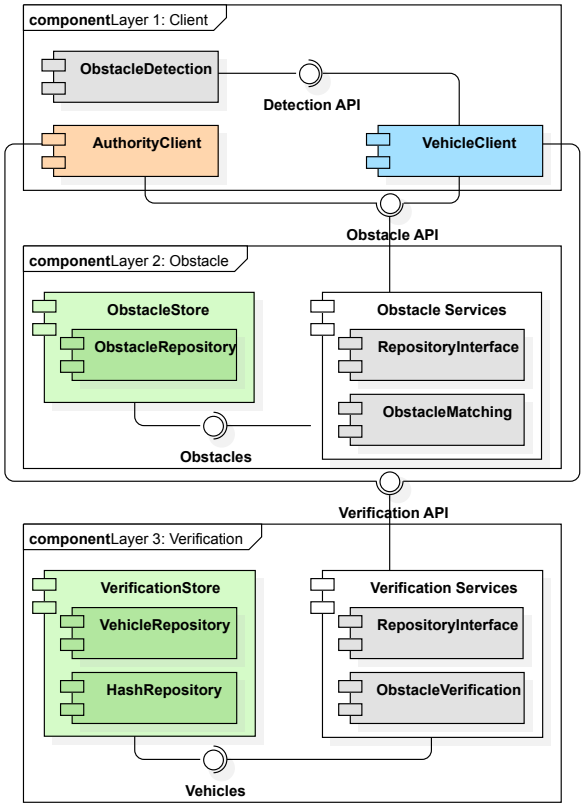


Figure 2. The subsystem decomposition model shows the six subsystems, which are organized into three layers within an open layered architecture.

4.1. Objective

The aim is to prove the feasibility of the decentralized Starling system for coordinated obstacle mapping. We implemented a prototype to gain further insight into the advantages and limitations of combining blockchain technology and distributed databases.

4.2. Hardware / Software Mapping

In the following, the decomposed subsystems are mapped to commercially available software and hardware components with which we implemented the prototype. Considering the requirements and constraints, we obtain the hardware-software mapping as shown in Figure 3.

First, we explain the components that enable our decentralized data storage, which is based on two separate components. For the **distributed databases**, we use the high-throughput, serverless, distributed, and

queryable database OrbitDB[1]. OrbitDB is based on the peer-to-peer protocol IPFS [2] designed to create a distributed database. To achieve consistency across all nodes, OrbitDB uses an immutable, operation-based conflict-free replicated data structure (CRDT) [38], which was proposed by Shapiro et al. [39] in the context of the increasing emergence of distributed systems. CRDTs include, among others, an append-only log that can be used to model a mutable, shared state between peers in peer-to-peer applications meeting the requirements for documentation quality, flexibility, and throughput at the time of system design. We use the official IPFS client implementation, which is written in Go[3] providing the IPFS HTTP API[4] as an interface that allows clients to interact with it and the stored data.

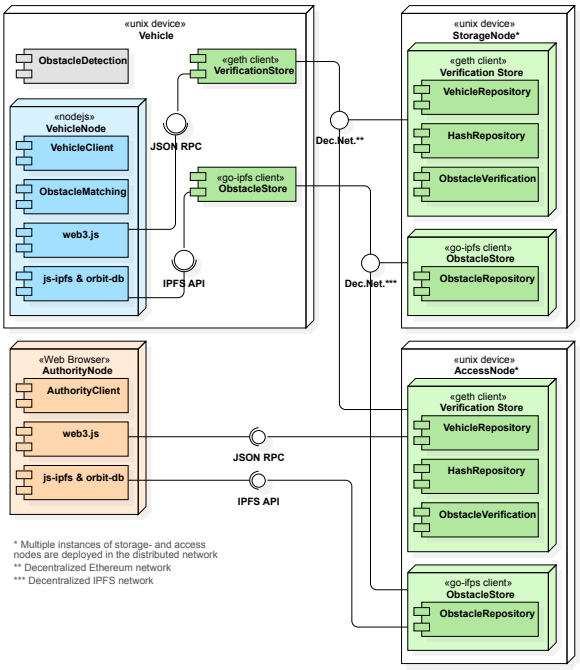


Figure 3. The deployment diagram maps the decomposed subsystems to commercially available software and hardware components.

In order to build the **blockchain** test network, we decided to use Ethereum with the Proof-of-Work consensus mechanism. Ethereum is a general purpose permissionless blockchain, which provides detailed documentation, a high degree of decentralization, and enables the use of smart contacts. To connect to the Ethereum test network, one has the choice between the official Ethereum client named Geth [5] and the independently developed Parity client[6]. We use Geth in Starling, as it is the reference implementation of Ethereum that provides client access via the standardized JSON RPC API[7].

4.2.1. StorageNode and AccessNode The core component of Starling is the distributed data storage in which detected obstacles are mapped and stored. The distributed data storage is separated into two independent components, the distributed file system IPFS, and the Ethereum blockchain.

4.2.2. VehicleNode The `VehicleNode` provides the system with the actual obstacle mapping data. For the case study, a Node.js[8] environment was chosen, in which the `VehicleNode` is executed. It connects the `ObstacleMapping` service with the decentralized `ObstacleStore`. This component expects an interface provided by the `ObstacleDetection` on-board system that allows it to receive notifications of detected obstacles. The `ObstacleDetection` on-board system is not part of the Starling system and is therefore simulated. The subsystem `ObstacleMatching` compares the detected obstacles with those already stored in Starling. These obstacles are then stored in IPFS using the OrbitDB Javascript library, which connects to the local IPFS `Store` via the IPFS Javascript library[9]. Once an obstacle data set is stored in IPFS, a unique hash value of this data set is generated and recorded on the Ethereum blockchain. The hash value serves as the immutable digital identity of the obstacle data and can be verified at any time. The hash value is stored via the local GethClient with web3.js[10], which is a collection of Javascript libraries that allow you to interact with a local or remote Ethereum node over an HTTP or IPC socket connection. `VehicleNodes` can retrieve mapped obstacles via the OrbitDB Javascript library. Based on this information, the autonomous driving systems can, for example, make decisions such as initiating a safety brake. The `ObstacleVerification` subsystem is implemented by a smart contract running on Ethereum, which contains the business logic for obstacle verification. Once the verification process is

[1] `https://github.com/orbitdb/orbit-db`
[2] `https://ipfs.io/`
[3] `https://github.com/ipfs/go-ipfs`
[4] `https://docs.ipfs.io/reference/api/http/`
[5] `https://github.com/ethereum/go-ethereum`
[6] `https://github.com/paritytech/parity-ethereum`
[7] `https://github.com/ethereum/wiki/wiki/JSON-RPC`
[8] `https://nodejs.org/`
[9] `https://github.com/ipfs/js-ipfs`
[10] `https://github.com/ethereum/web3.js/`

completed, the `VerificationStore` service in the `ObstacleVerification` subsystem can be used to perform verification by call.

4.2.3. AuthorityNode The `AuthorityNode` provides authority actors access to the obstacle mapping data by means of a graphical user interface. It is a Javascript application that runs in web browsers with the same libraries that are used for the `VehicleNode`. To access the mapped obstacle data, it connects to the decentralized database over a public interface exposed by a remote `ObstacleRepository`, which is represented by the `AccessNode`.

4.2.4. Network All subsystems and devices of the Starling system can be connected using an arbitrary network. This can be a private network deployed in a local environment or a public one like the internet.

5. Evaluation

We implemented viable subsystems in order to evaluate the proposed Starling system.

5.1. Simulation

As the operation of Starling in a real scenario was not possible, we simulated vehicles that reliably detected obstacles on a selectable route. For the simulation, we developed further subsystems (e.g., simulation manager) that communicate with each other and were integrated into the Starling system. The simulation manager has access to all simulated vehicles as well as their position, speed, and storage data, but is not aware of the extended environment of the vehicles. The data provided by the simulation manager represent the obstacle data that a real vehicle would receive from its sensors. Hence, the simulation manager can be considered as the `ObstacleDetection` subsystem.

The graphical web interface depicted in Figure 4, shows the simulation using a map. Using this interface, any information such as traffic conditions can be displayed and the status of a specific vehicle can be reproduced at a certain point in time.

5.2. Design

For the evaluation, ten equivalent virtual machines were provisioned, each with 8GB RAM, 4 CPU cores and 100GB hard disk space running Ubuntu 16.04 LTS as the operating system. The machines operated in a data center of a cloud hosting provider and were connected

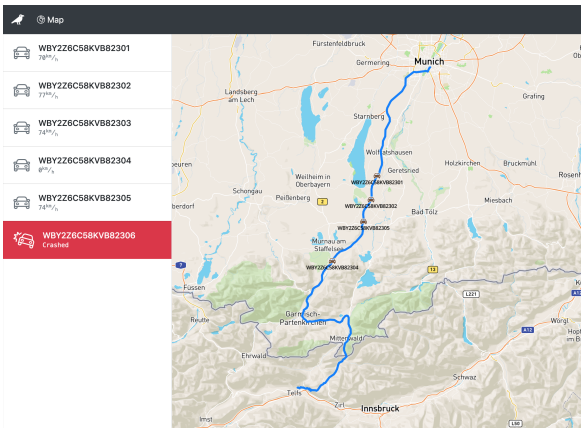


Figure 4. Starling user interface showing the simulation manager and the simulated vehicles.

with each other through a local network. The Docker Engine was installed on all machines, enabling them to be combined into a cluster using Docker's swarm mode. By running the simulation, the performance and scalability of combining the Ethereum blockchain and the decentralized, peer-to-peer data storage IPFS will be quantitatively evaluated for the use of coordinated obstacle mapping. Based on this data, it is verified whether the requirements are fulfilled and if these technologies are ready to be used for the purpose of coordinated obstacle mapping.

5.3. Results

We have performed three simulation runs with different objectives. In the **first simulation run** with six vehicles, the verification process – using the hash values stored in the blockchain – was disabled when matching the detected obstacle with the already mapped ones. The detected obstacles could be matched to the obstacles returned by a query on OrbitDB. In addition, towards the end of the simulation, when all six vehicles were stuck in traffic jams, the exact number of six obstacles were mapped, with the obstacles detected by the vehicles always matching the stored ones. Observing the state of the blockchain using the Ethereum Network Status interface, an average of 13 transactions per block are shown, which is approximately 1.6 transactions per second with an average block time of 8 seconds.

In the **second simulation run**, we continued with six vehicles over a 24-hour period to measure the convergence behavior of the average block time, which settled at 15.6 seconds, while 98% of the blocks propagated in under one second. Another relevant time interval is the average delay that occurs when OrbitDB replicates data between its distributed nodes. Before

measuring this value, the internal clocks of the nodes connected to OrbitDB were synchronized with the same NTP time server, so that the replication delay could be calculated without deviations due to inaccurate clocks. Beyond that, the number of nodes involved was limited to two, so that the actual replication time between these nodes could be calculated. The simulation was started and the timestamps at the time of saving on one node and replication on the other node were extracted. This data set was used to calculate the difference between the two timestamps and their mean value, which is 197 milliseconds.

Finally, we tested the system regarding its scalability in the **third simulation run**. For this purpose, the number of vehicle nodes was increased to 20 while the reaction of the system to this change was observed. It could be seen that with a growing number of vehicles the number of Ethereum and OrbitDB transactions increases. While the vehicles were scattered across the route to the accident site, the blockchain processed an average of 26 transactions per block, which is approximately 1.67 transactions per second with an average block time of 15.6 seconds. This value increases to up to 7 transactions per second as more vehicles get into traffic jams. During this period, however, the replication time of OrbitDB remained constant at about 200 ms.

6. Discussion

The Starling system is a successful proof of concept and is capable of performing coordinated obstacle mapping using Ethereum and IPFS in a reliable way. However, due to the limiting properties of the **Ethereum** virtual machine (e.g., support of floating point numbers), the matching algorithm had to be implemented in the vehicle node running in the distributed database instead in the smart contract. The results of our evaluation show that the average block times using the **Proof-of-Work** consensus mechanism are higher (approx. 15 seconds) than the needed near-real time requirement for obstacle mapping (1 second). This means that the obstacles queried from the distributed database are not validated, which is why incorrect obstacle data could not be identified. However, as soon as transactions are added to the blockchain, obstacle reports can be traced and verified in retrospect.

Concerning **data privacy**, Ethereum provides pseudo-anonymity in the form of addresses. However, once the relation between an address and a vehicle is uncovered, vehicle data can be associated to the vehicle owner. Therefore, vehicles need to generate a one-time address per-trip or per-event while persisting their used addresses locally. In case a one-time address is needed, an authorized actor such as the vehicle owner can access these addresses stored by the vehicle.

Since deleting data from the blockchain is not possible by design, a further challenge is the **increasing demand and costs** for data storage, especially for resource-limited vehicle nodes. As the majority of data stored in vehicles will be obsolete in terms of time or irrelevant based on their position, we propose introducing so-called light nodes. Light nodes do not download or verify the entire chain of blocks, instead they rely on full nodes for sending transactions and querying obstacle data. However, this could not be realized in the course of our simulation, because the still experimental light nodes of Ethereum were unable to discover any full nodes, and OrbitDB was not yet capable of this functionality. To tackle this issue we trade decentralization for storage, which shifts the control of the network towards the operators of full nodes. This compromise could be accepted in the Starling network, as full node operators will be a diverse group of stakeholders including vehicle manufacturers, governments, and insurance companies.

While blockchain technology can guarantee that the obstacle mapping data is not tampered with, it does not guarantee that the data recorded by the sensors is accurate. Additional technical controls such as secure sensor elements with unique private keys to ensure increased data integrity may be required.

The second component that was evaluated was the distributed data storage **IPFS**, which met our requirements for storing obstacle data, even replicating in half the time stated in the quality requirements. In the future, the obstacle matching should be executed by the blockchain and not by the vehicle, as it is implemented in the current version of the Starling prototype.

7. Conclusion

In this paper, we successfully designed and developed the Starling system, which allows vehicles to map detected obstacles in a coordinated manner by means of distributed ledger technology. Furthermore, we implemented a reliable algorithm for matching detected obstacles with already mapped ones, so that the validity of the latter can be verified, duplicated reports prevented, and their traceability ensured. The use of a distributed ledger for the coordinated mapping of obstacles in the context of road traffic embodies a promising technology, especially when considering the immutability and traceability of the data stored. Decisions made by autonomous vehicles based on the sensed surroundings can be comprehended in retrospect.

We addressed the functional and non-functional requirements described in Section 3.2.: The requirements of (1) Provide Obstacle Repository, (2) Avoid Duplication, (3) Detect Faulty Obstacle Data, (4) Provide Data Access, (5) Provide Authority Access and (6) Provide Traceability were addressed by introducing IPFS and our User Interface. (7) Open Platform, (8) High Availability, and (9) Data Integrity were archived using a public blockchain protocol based on proof-of-work (i.e., Ethereum) allowing everyone to connect to Starling. Starling is distributed among several nodes fulfilling high availability in a way that hardly any other ordinary database can. Finally, (10) Protect Privacy, (11) Low Latency, and (12) High Throughput could be partially fulfilled.

In order to ensure data privacy in the future, vehicles need to generate a one-time address while persisting its used addresses locally. In addition, as Starling has to deal with several thousand transactions per second, future work should investigate distributed ledger solutions focusing on scalability (e.g., Tendermint, IOTA, and Hedera Hashgraph), second layer solutions (e.g., Raiden and Lightning Network), and the use of light nodes for resource-limited vehicle nodes. Furthermore, it should be investigated how manufacturers and vehicle owners can monetize their data to create new revenue streams.

References

[1] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark, "Connected Vehicles: Solutions and Challenges," *IEEE Internet of Things Journal*, vol. 1, no. 4, pp. 289–299, 2014.

[2] D. A. Rosen, F. J. Mammano, and R. Favout, "An Electronic Route-Guidance System for Highway Vehicles," *IEEE Transactions on Vehicular Technology*, vol. 19, no. 1, pp. 143–152, 1970.

[3] H. Hartenstein and K. Laberteaux, *VANET: Vehicular Applications and Inter-Networking Technologies*, vol. 1. John Wiley & Sons, 2009.

[4] H. Onishi, "A Survey: Engineering Challenges to Implement VANET Security," *IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pp. 1–6, 2018.

[5] H. Endo, "Obstacle Detection System for Use in Vehicles," 1984. US Patent 4,477,184.

[6] K. Storjohann, T. Zielke, H. A. Mallot, and W. von Seelen, "Visual Obstacle Detection for Automatically Guided Vehicles," in *IEEE International Conference on Robotics and Automation*, pp. 761–766 vol.2, 1990.

[7] Y. Asayama, "Obstacle Detecting Device for a Vehicle," 1995. US Patent 5,386,285.

[8] R. Labayrade, D. Aubert, and J.-P. Tarel, "Real Time Obstacle Detection in Stereovision on non Flat Road Geometry Through "v-disparity" Representation," in *Intelligent Vehicle Symposium, 2002. IEEE*, vol. 2, pp. 646–651, 2002.

[9] W. Fuzhong, L. Haibo, and Y. Fashan, "Obstacle Avoiding Strategy of Mobile Robot via Binocular Stereovision," in *27th Chinese Control Conference*, pp. 457–461, IEEE, 2008.

[10] J. Han, D. Kim, M. Lee, and M. Sunwoo, "Enhanced Road Boundary and Obstacle Detection Using a Downward-Looking LIDAR Sensor," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 3, pp. 971–985, 2012.

[11] L. C. McNinch, R. A. Soltan, K. R. Muske, H. Ashrafiuon, and J. C. Peyton Jones, "Application of a Coordinated Trajectory Planning and Real-Time Obstacle Avoidance Algorithm," in *Proceedings of the 2010 American Control Conference*, pp. 3824–3829, 2010.

[12] S. Fujii, A. Fujita, T. Umedu, S. Kaneda, H. Yamaguchi, T. Higashino, and M. Takai, "Cooperative Vehicle Positioning via V2V Communications and Onboard Sensors," in *IEEE Vehicular Technology Conference (VTC Fall)*, pp. 1–5, 2011.

[13] J. Ward, S. Worrall, G. Agamennoni, and E. Nebot, "The Warrigal Dataset: Multi-Vehicle Trajectories and V2V Communications," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 3, pp. 109–117, 2014.

[14] M. C. Ballandies, M. M. Dapp, and E. Pournaras, "Decrypting Distributed Ledger Design - Taxonomy, Classification and Blockchain Community Evaluation," 2018.

[15] M. Swan, *Blockchain: Blueprint for a new Economy*. O'Reilly Media, Inc., 2015.

[16] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.

[17] U. Mukhopadhyay, A. Skjellum, O. Hambolu, J. Oakley, L. Yu, and R. Brooks, "A Brief Survey of Cryptocurrency Systems," in *2016 14th Annual Conference on Privacy, Security and Trust (PST)*, pp. 745–752, IEEE, 2016.

[18] F. M. Benčić and I. Podnar Žarko, "Distributed Ledger Technology: Blockchain Compared to Directed Acyclic Graph," in *IEEE 38th*

International Conference on Distributed Computing Systems (ICDCS), pp. 1569–1570, 2018.

[19] G. Wood, "Ethereum: A Secure Decentralised Generalised Transaction Ledger," *Ethereum Project Yellow Paper*, vol. 151, pp. 1–32, 2014.

[20] N. Szabo, "Formalizing and Securing Relationships on Public Networks," *First Monday*, 1997.

[21] M. Alharby and A. van Moorsel, "Blockchain Based Smart Contracts: A Systematic Mapping Study," *Computer Science & Information Technology (CS & IT)*, 2017.

[22] X. Xu, I. Weber, M. Staples, L. Zhu, J. Bosch, L. Bass, C. Pautasso, and P. Rimba, "A Taxonomy of Blockchain-Based Systems for Architecture Design," in *IEEE International Conference on Software Architecture (ICSA)*, IEEE, 2017.

[23] L. W. Cong, Z. He, and J. Zheng, "Blockchain Disruption and Smart Contracts," *SSRN Electronic Journal*, 2017.

[24] M. Kõlvart, M. Poola, and A. Rull, "Smart Contracts," *The Future of Law and eTechnologies*, 2016.

[25] J. Benet, "IPFS-Content Addressed, Versioned, P2P File System," *arXiv*, 2014.

[26] S. Rowan, M. Clear, M. Gerla, M. Huggard, and C. M. Goldrick, "Securing Vehicle to Vehicle Communications Using Blockchain Through Visible Light and Acoustic Side-Channels," *arXiv*, 2017.

[27] A. Buzachis, A. Celesti, A. Galletta, M. Fazio, and M. Villari, "A Secure and Dependable Multi-Agent Autonomous Intersection Management (MA-AIM) System Leveraging Blockchain Facilities," *IEEE/ACM International Conference on Utility and Cloud Computing Companion*, pp. 226–231, 2018.

[28] M. Singh and S. Kim, "Blockchain Based Intelligent Vehicle Data sharing Framework," *arXiv*.

[29] Rathore, Heena and Samant, Abhay and Jadliwala, Murtuza and Mohamed, Amr, "TangleCV: Decentralized Technique for Secure Message Sharing in Connected Vehicles," *Proceedings of the ACM Workshop on Automotive Cybersecurity*, pp. 45–48, 2019.

[30] G. S. Ramachandran, X. Ji, P. Navaney, L. Zheng, M. Martinez, and B. Krishnamachari, "Micropayments for Trusted Vehicular Services Using MOTIVE," *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, pp. 701–702, 2019.

[31] C. Hewlett Packard Enterprise Company, "Hewlett Packard Enterprise and Continental launch blockchain-based data monetization platform," 2019.

[32] Z. Yang, K. Zheng, K. Yang, and V. C. M. Leung, "A Blockchain-Based Reputation System for Data Credibility Assessment in Vehicular Networks," in *IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 1–5, 2017.

[33] Z. Lu, W. Liu, Q. Wang, G. Qu, and Z. Liu, "A Privacy-Preserving Trust Model Based on Blockchain for VANETs," *IEEE Access*, vol. 6, pp. 45655–45664, 2018.

[34] C. Oham, S. S. Kanhere, R. Jurdak, and S. Jha, "A Blockchain Based Liability Attribution Framework for Autonomous Vehicles," *arXiv*, 2018.

[35] J. Kang, R. Yu, X. Huang, S. Maharjan, Y. Zhang, and E. Hossain, "Enabling Localized Peer-to-Peer Electricity Trading Among Plug-in Hybrid Electric Vehicles Using Consortium Blockchains," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3154–3164, 2017.

[36] F. Gao, L. Zhu, M. Shen, K. Sharif, Z. Wan, and K. Ren, "A Blockchain-Based Privacy-Preserving Payment Mechanism for Vehicle-to-Grid Networks," *IEEE Network*, vol. 32, pp. 184–192, 2018.

[37] Z. Zhou, L. Tan, and G. Xu, "Blockchain and Edge Computing Based Vehicle-to-Grid Energy Trading in Energy Internet," *IEEE 2nd Conference on Energy Internet and Energy System Integration (EI2)*, pp. 1–5, 2018.

[38] OrbitDB Development Team, "OrbitDB: Peer-to-Peer Databases for the Decentralized Web," 2019.

[39] M. Shapiro, N. Preguiça, C. Baquero, and M. Zawirski, "Conflict-Free Replicated Data Types," in *Symposium on Self-Stabilizing Systems*, pp. 386–400, Springer, 2011.

4 Co-authored publication

This chapter contains a co-authored publication that is not part of the contributions of this dissertation.

4.1 Fog Computing as Enabler for Blockchain-Based IIoT App Marketplaces - A Case Study

This publication (Publication [E]) proposes a novel approach for an open app marketplace in industrial environments.

Conference: 2018 International Conference on Internet of Things: Systems, Management and Security (IoTSMS)
Number of Pages: 7
Review: Peer Reviewed (3 Reviewers)

While the focus of the publication is on fog computing, the author of this dissertation contributed to the blockchain-related content of this publication, in particular concerning the conduction of the literature review and technical foundations, writing the paper, and providing valuable feedback.

Goal

With the increasing number of IoT devices in industrial environments, the management of their software is crucial to ensure software security. The goal of this DLT application was to provide an open app marketplace for industrial IoT edge devices.

Approach

We combined a DLT network with the hierarchical, distributed structure of a fog computing architecture. DLT was used to create an open marketplace to provide new delivery channels and flexible payment models (e.g., pay-per-use models) for developers and traceability of software updates for technicians. This allows applications to be

offered and transmitted in a P2P network. In addition, it enables the monitoring of software installations and updates on IoT edge devices.

Results

The DLT marketplace enables developers to offer, control, and monetize their apps for IoT devices in industrial environments. Developers and technicians can obtain information about apps as all interactions with the IoT edge devices are recorded on the distributed ledger, increasing transparency and traceability of the software lifecycle management. Due to the low transaction costs, DLTs enabled consumption-based payment models for the use of the apps.

IEEE Reuse Requirements

With regard to the IEEE reuse requirements for dissertations, the accepted and not the published version of the publication is used in the following.

Fog Computing as Enabler for Blockchain-Based IIoT App Marketplaces - A Case Study

Andreas Seitz[1], Dominic Henze[1], Daniel Miehle[1], Bernd Bruegge[1], Jochen Nickles[2] and Markus Sauer[2]
[1] Chair for Applied Software Engineering, Technische Universitat Munchen,Munich, Germany
Email: {seitz, henzed, miehled, bruegge}@in.tum.de
[2] Siemens AG, Corporate Technology, Munich, Germany
Email: {jochen.nickles, markussauer1}@siemens.com

***Abstract*—Fog Computing, Augmented Reality (AR) and Blockchain are technologies that have been receiving more attention in recent years. Fog Computing for on-site data analysis and Blockchain as a decentralized framework are used to establish security and trust. AR enables new interactive scenarios with devices in the industrial environment and provides user interfaces for devices that had previously lacked one. The combination of these technologies in an industrial environment opens up new possibilities and opportunities. We present the IIoT Bazaar, a decentralized marketplace for industrial edge applications that relies on Blockchain to create transparency for all stakeholders involved and to enable the traceability of app installations on edge devices. With Fog Computing, resource-limited edge devices can be integrated into the IIoT Bazaar ecosystem. Fog Nodes enable the provisioning of applications on edge devices and their integration into a decentralized Blockchain network. AR serves as an interface between the users and the machines, allowing people to interact intuitively with the edge devices. With a case study, we demonstrate the design and prototypical implementation of the IIoT Bazaar and its applications which bring software developers and users together using a combination of Fog Computing, AR, and Blockchain. We share the challenges we faced and our experiences we made during the case study.**

***Index Terms*—Fog Computing, Blockchain, Augmented Reality, IIoT, Smart Contracts, Edge Computing, Industry 4.0, Distributed Ledger**

I. Introduction

Devices and software in industrial environments are distributed and heterogeneous, and so, difficult to administer, as they are often used in many places for different purposes. Moreover, devices and their software are often individual, tailored solutions, which makes them difficult to develop and update. In addition to the maintenance, the distribution and promotion of software and accompanying updates is tedious.

Marketplaces for apps have emerged from the consumer world where they are already well-known and used for smartphones. Examples of consumer marketplaces are the Apple App Store and Google Play Store. Users are accustomed to extending the functionality and possibilities of their smartphones with various apps. These processes are designed to be particularly user-friendly: After authentication and payment, the app is installed on the device and is immediately available to the user. Marketplaces bring advantages not only for the users but also for the developers of apps. If an app reaches a critical mass, it is successfully distributed through a central point of contact.

Contrary to app stores for smartphones, marketplaces in the industrial environment face various challenges. IIoT solutions are often tailor-made individual solutions: both hardware and software side. The reuse of applications is not possible due to the lack of distribution platforms. From the developer's point of view, the appropriate delivery channels are missing. Current solutions also do not offer flexible payment models such as pay per use.

From the perspective of edge device developers and technicians, working with these devices raises further challenges. For example, technicians must visit equipment on site and initiate manual the installation of software and updates. The devices in the field often do not provide a user interface, making feedback to the technician difficult and non-transparent. In addition, there is a lack of standardization in the installation process of updates. Furthermore, it is not traceable who carried out which updates when or why. However, the reliable auditing and traceability of software installations and updates are required.

New payment models, such as purchasing time-based functionalities, are also interesting for companies using IIoT. A lack of technological possibilities currently prevents the establishment of such new payment models. These challenges and problems, as well as the individuality of applications and use cases, make it difficult to develop a platform that meets these heterogeneous requirements.

The IIoT Bazaar presented in this paper, attempts to address these challenges and offers solutions by combining different technologies. Technological advances such as Cloud Computing, Fog Computing, Blockchain and Augmented Reality (AR) have already found their way into the industrial domains [1]. The combination of these technologies with Blockchain opens up new opportunities and can overcome existing limitations [2], [3]. Bottlenecks and single points of failures can be eliminated by decentralization.

The IIoT Bazaar aims to address these challenges by applying a decentralized Blockchain network and the hierarchical, distributed structure of a Fog Computing architecture. Blockchain technology is used to create a marketplace for the exchange of applications in which no third parties need to be trusted [4]. This eliminates the "man-in-the-middle" and

allows apps to be offered and transmitted in a peer-to-peer network. By combining IIoT and Blockchain, new applications as well as advanced payment models can be generated.

This paper builds on [1] and is structured as follows: In Section II, we present the foundations of IIoT, Fog Computing, Augmented Reality and Blockchain, followed by comparable approaches to the IIoT Bazaar. In Section III, we present the conceptual structure and the objectives of the IIoT Bazaar starting from the requirements, to the analysis of the problem domain and an overview of the architecture. In the case study in Section IV we discuss the prototypical implementation using AR, Fog Computing and a private Ethereum Blockchain. We report our experiences and challenges which we have discovered. Section V concludes the paper by summarizing the contributions and offering an outlook on our future work.

II. Foundations & Related Work

The following section describes IIoT, Fog Computing, Augmented Reality (AR) and the Blockchain technology – the foundations of our approach. Additionally, related approaches are reviewed to embed our approach in the corresponding scientific background.

A. Industrial Internet of Things

While IoT devices, such as hozsehold appliances and smart textiles, are already widespread in our daily lives, most of them are considered as Consumer IoT (CIoT) devices. [5]. These devices work very well for private use but do not focus on reliability or connectivity which would be necessary for an industrial setting [5]. IIoT devices on the other hand focus on this aspect [5] and bridge the gap between the physical, industrial world and the digital environment [6] to create cyber-physical settings that use real sensor values, inputs from actuators and interactive devices [7] to collect and process data for pattern recognition, prediction and optimization [8].

B. Fog Computing

Fog Computing was first proposed by Bonomi et al. to extend the possibilities of Cloud Computing all the way down to the edge of the network by introducing an additional layer called Fog [9]. Heavy computational tasks are still executed within the Cloud to harvest the calculation power and power independence of the Cloud servers. IoT devices on the other hand are located in the edge layer with so-called Fog Nodes connecting the devices to the Cloud [9]. Located close to the IoT devices, these Fog Nodes produce and consume data to enable real-time analysis and processing of data, thereby reducing latencies and saving bandwidth [1], [10].

C. Augmented Reality (AR)

Augmented Reality (AR) is a special case of Virtual Reality (VR) [11]. While VR focused on creating an entire virtual environment for the user to interact with, AR introduces virtual objects within the user's physical surrounding. This allows users to interact with the physical surrounding while having additional information and inputs to interact with the virtual environment. Thus, AR can be seen as the 'middle course' between reality and VR [12]. There are many ways of introducing AR using devices, such as head-mounted displays, AR glasses, but also smartphones and tablets.

D. Blockchain Technology

Blockchain technology was first introduced by Satoshi Nakamoto [13] in 2008 for the cryptocurrency Bitcoin and represents a subclass of the Distributed Ledger Technology (DLT). A Blockchain is a distributed database that records transactions in a chronological, ordered list of blocks [14]. Each block consists of transactions, a hash of the preceding block, and a timestamp providing traceability and immutability of transactions. A new block is consistently synchronized across the peer-to-peer network by means of a consensus algorithm. Blockchain technology eliminates the need for a central authority or intermediary to process or validate transactions reducing complexity and cost [15]. The first generation of Blockchain protocols such as Bitcoin mainly focused on the exchange of financial transactions. In contrast, following Blockchain protocols such as Ethereum enable the deployment of executable code, known as Smart Contracts [16], which run on the Blockchain. Smart Contracts are computerized transaction protocols that execute the terms of a contract to enable the automatic exchange of values such as cryptocurrencies in a secure and transparent manner [17], [18].

E. Related Work

The individual technological components have been viewed in many ways by academia and industry. The combination of these topics is still a challenge for research [2]. We refer below to publications that are closely related to our vision and ideas.

In [19], Huckle et al. describe how the IoT and Distributed Ledger Technologies generate opportunities to develop so called distributed applications (DApps) for a shared economy. Their paper describes visionary scenarios made possible by the combination of IoT and Blockchain. To the best of our knowledge, none of the scenarios have been implemented to date. The authors discuss the shared use of apps but do not refer to the marketplace idea. Pan et al. [20] designed and implemented an Edge-IoT framework based on Blockchain and Smart Contracts. The Smart Contracts are for example used to manage the resource allocation of an IoT device in an automated way. Reyna et al. stated that the combination of IoT and Blockchain could "accelerate the creation of an IoT ecosystem of services and data marketplaces" [2]. We agree, but instead of focusing on the data, we focus on the provision of services in the form of apps in an industrial environment. Furthermore, Reyna et al. mentioned that "secure code deployment" through the integration of technologies could be another advantage [2]. Publications [21], [22] discuss how code can be pushed safely and securely into devices using the Blockchain.

III. Bazaar

In this section, we present the idea of establishing a marketplace for IIoT applications based on the technologies of Fog

Computing, AR and Blockchain – the IIoT Bazaar. We chose the name 'Bazaar' to emphasize our wish to bring supply and demand together, thereby creating a marketplace with special features in terms of design, location, and function.

A. Problem

Edge devices serve as enablers for smart factories and establish continuous communication among different data sources that connect machines with sensors, actuators, and Cloud applications.

These systems are business critical to smart factories for ensuring that their machines run uninterrupted. Soon, these edge devices may also enable predictive maintenance, availability and effectiveness.

Technicians are responsible for regularly updating the software and applications on these devices, which is a process that is rather cumbersome and complex. Information about new software or updates is usually published via newsletters or internal portals. The technician must download the software, drag it on an USB drive and then transfer it to the device that requires an update in the first place and possibly waiting up to 15 minutes for the update process to complete. With the increasing number of edge devices, this method of updating poses a scaling problem. Many of the devices do not offer a user interface but give only feedback via LEDs on the installation status. This process is error-prone and in need of improvement. In the consumer world, there is already a simple and elegant way which can also be used for the industrial domain – Marketplaces for apps.

B. Vision

The vision for the IIoT Bazaar is the simplification and acceleration of the installation and updating of apps as easily on edge (IIoT) devices as on smartphones. Using machine learning and QR codes, a technician can detect and identify devices by using a smartphone with AR technology and thus receives a user interface for an edge device that would otherwise lack one. The usually large number of manual steps for installation is reduced to a simple drag & drop operation. When the technician points to the device, the technician's smartphone shows which applications are running on the device, which updates are available, and which new applications can be installed.

The technician takes an app from a list and drags it via AR to the device. The process is started after authentication and visualized in AR. If there is an error, the technician receives feedback via AR.

The IIoT Bazaar app also provides the technician with additional information about the apps: ratings, version number, descriptions, developer information and price.

The metadata about the availability of apps as well as the traceability of installations and updates are stored in the Blockchain, allowing transparency and traceability. The Blockchain as a distributed and open system enables several distributed marketplaces to work in parallel.

C. Requirements

We define the following requirements for the successful implementation and acceptance of the IIoT Bazaar concept:

(1) Open Platform: The IIoT Bazaar must be an open platform, which means that the barriers to entry for both developers and users must be low. The platform must create transparency for all parties involved.
(2) End-to-end Delivery: The IIot Bazaar must support the process of providing an application to installing it on an edge device. End-to-end means from uploading the app to the marketplace through the actual installation of the app on the edge device to payment. Furthermore, there must be a possibility for updates.
(3) User-Centric Design: Usability must be oriented to the same expectations as that from the consumer world, i.e., the rapid notification of the current statuses of the devices and simple initiation of installations and updates.
(4) Independence: The IIoT Bazaar should be a decentralized system in which there is no regulating authority.
(5) Payment Model Flexibility: The IIoT Bazaar must support different payment models. The developers are responsible for offering and providing new and flexible payment models.
(6) On- and Off-Site Remote Update Management: Installation and update management must be available on-site and remotely. The on-site technician can use his smartphone for interaction, but also manage the devices remotely, e.g. from an office.
(7) Flexibility and Extensibility: IIoT must be flexible and expandable. It must support different IIoT technologies and devices and be adaptable for the respective technicians.
(8) Traceability: All installations and updates must be documented for regulatory reasons. The respective processes must be visible and traceable to the different parties.

D. Analysis

For a better understanding of the problem domain we created an analysis object model (cf. Figure 1) to describe the static structure of the IIoT Bazaar.

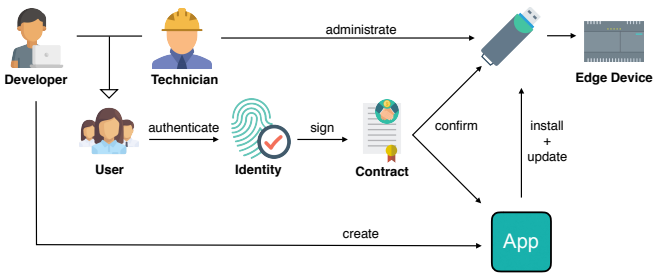


Fig. 1. **IIoT Bazaar Analysis Object Model:** the model helps to better understand the problem and to develop the solution approach of the IIoT Bazaar.

We have two actors that interact with the IIoT Bazaar. IIoT application developers who want to distribute their apps and technicians who are responsible for the installation and

availability of the applications. *Developers* create applications (*Apps*) to run on *Edge Devices*. *Technicians* oversee the installation and updating of those applications onto the *Edge Devices*. Both, the *Technician* and the *Developer*, require authentication to perform these operations. It is essential to keep track of who developed which application and ensure the correct applications/updates are being installed by the technician on the *Edge Device*. The contract exists to validate and confirm these transactions.

E. Architecture

Based on the results of [1] and the hybrid approach as described in [2], we propose the following architecture (c.f. Figure 2) for the IIoT Bazaar, which is based on the Fogxy architectural pattern for Fog Computing as presented in [23]. We discuss the individual components and their roles within the IIoT Bazaars ecosystem.

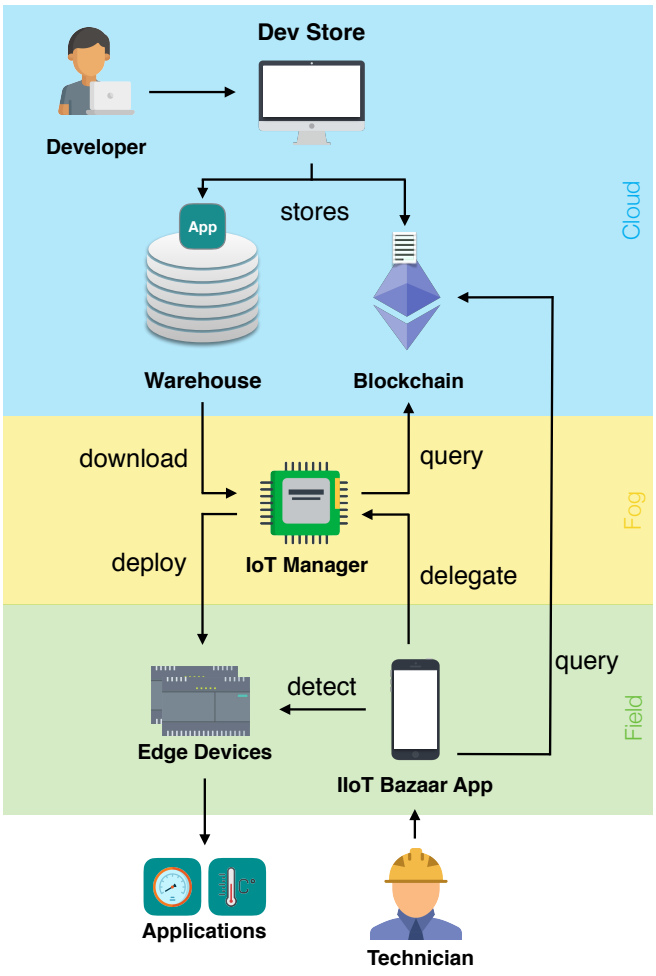


Fig. 2. **IIoT Bazaar Top-Level Design:** IIoT Bazaar's ecosystem is spread over the three layers Cloud, Fog and Field. The *Dev Store*, *Warehouse* and *Blockchain* are used in the cloud layer and are available remotely. The central component in the fog layer is the IoT Manager, which acts as a proxy between the field and cloud layers. In the field layer, we have both edge devices and smartphones that allow interaction with the *Edge Devices*.

IIoT Bazaar App: The *IIoT Bazaar App* is the link between the *Edge Devices* in the field and the *Technician*. The *Edge Devices* rarely have user interfaces. The technicians use their smartphones, which have the *IIoT Bazaar App* installed, to detect, identify and interact with the *Edge Devices* in AR. After successful authentication, the technicians can install or uninstall apps using drag & drop operations. Furthermore, the *IIoT Bazaar App* provides information about the current status of the *Edge Devices* and their interactions with regard to app installations and updates.

IoT Manager: Running on the *Fog Node*, the *IoT Manager* is the heart of the architecture and central point of contact. The *Fog Node* as defined in [24], establishes the connection between the components in the Cloud and those in the Field. The *IoT Manager* compensates for the missing resources on the *Edge Device* to enable on- and off-site maintenance and installation of apps.

Dev Store: The *Dev Store* is the interface for the developers with the IIoT Bazaar ecosystem. *Developers* can log in here to make their applications available for updates and view the data about the distribution.

An application in the IIoT Bazaar consists of a binary file, images such as icons and screenshots and metadata such as name, price and version number. The *Dev Store* passes the metadata with a reference to the warehouse and *Blockchain*.

Blockchain: There are multiple reasons for using a *Blockchain*. First of all, it provides a single source of truth in an environment with a possible lack of transparency. Second, it is open and therefore very feasible to introduce further parties, such as additional marketplaces to the IIoT Bazaar environment. Third, the Blockchain offers Smart Contracts, which deliver data storage and logic layer in a distributed fashion. Fourth, by using cryptography it provides a temper-proof way of recording transactions to enhance trust and traceability.

Warehouse: In the *Warehouse*, the binaries and other information, such as screenshots or app icons, are stored. The *Warehouse* exists parallel to the *Blockchain*. App entries in the Blockchain refer to the data in the *Warehouse*. *Technicians* first refer to the information which apps are available for which devices from the *Blockchain*. If a suitable app is available, it can be downloaded from the *Warehouse* to the selected *Edge Device*.

IV. Case Study

In this section, we describe our implementation of the IIoT Bazaar, which we began together with our industrial partners in the spring of 2018.

A. Objectives

We set ourselves the goal of developing a prototype and testing the implementation of the IIoT Bazaar concept. Furthermore, the prototype serves our industrial partners to validate the acceptance of such a system. Of particular interest to them is the use of the system by technicians in the field. The combination of AR and the deployment of applications using

drag & drop were a vision that can only be validated by trial and error. Since the Blockchain is a new technology from a software engineering point of view, it was also relevant to us how this concept can be combined with other technologies and programming languages. In the following sections, we describe the technical details, the test setup, and the implementation of the user interface.

B. Hardware/Software Mapping

We discuss the different subsystems in three different environments Cloud, Fog, and Field and explain the hardware and software we used to implement the prototype. Figure 3 shows the hardware/software mapping as UML Deployment Diagram explaining the components and communication mechanisms among them.

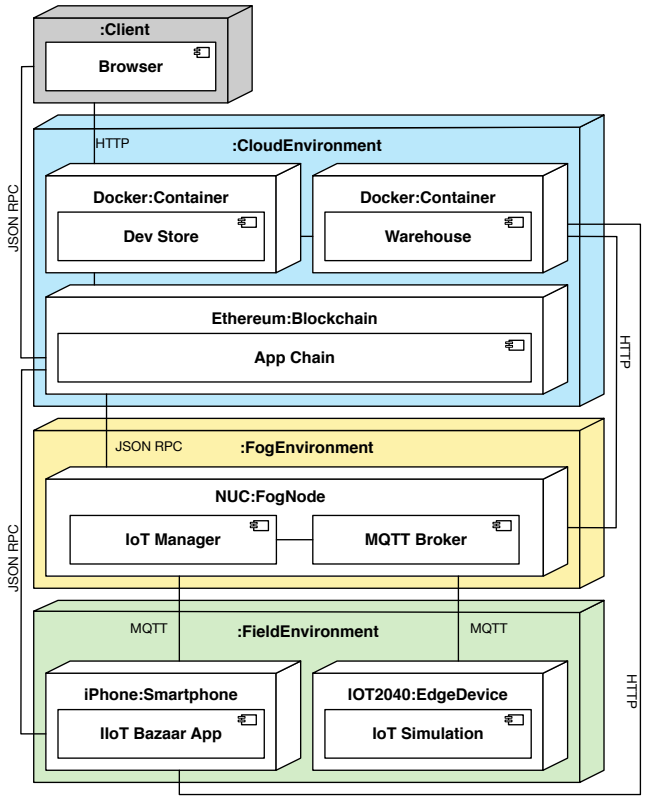


Fig. 3. **IIoT Bazaar Deployment Diagram:** The concept is divided into three layers: *CloudEnvironment*, *FogEnvironment* and *FieldEnvironment*. Different subsystems are deployed on different hardware nodes. Communication among the individual components takes place via JSON RPC, HTTP or MQTT

1) Cloud Environment: The *CloudEnvironment* contains three subsystems: *Dev Store*, *Warehouse* and *App Chain*. The two components *Dev Store* and *Warehouse* were deployed on a self-hosted virtual machine. The *App Chain* runs on AWS[1].

Dev Store: The *Dev Store* component is implemented using Angular 6.0 and runs on a Nginx web server. The Web application includes two components: User Management and Application Management. The former allows users to log in with their Ethereum key pair via the MetaMask[2] browser extension. No username or password is required, so authentication is done via the user's key. We use the JavaScript implementation of the web3[3] framework to communicate with the *App Chain* via JSON RPC.

Warehouse: The *Warehouse* component is implemented in Kotlin[4] on the Spring[5] framework. The *Warehouse* provides a REST interface that receives and stores the additional app information from the *Dev Store*. Furthermore, it offers an interface for the *IoT Manager* where the binaries files of the app can be obtained to enable the installation on the *EdgeDevices*. The component is packed in a docker container and can be deployed to any Cloud provider.

App Chain: This is the Blockchain component implemented with a private Ethereum Blockchain. It runs on three AWS Ethereum nodes, which find a consensus on the state of the Blockchain using Proof of Authority. Further Ethereum nodes can be used to scale the system.

To realize the IIoT Bazaar we established three Smart Contracts:

- **IIoTAppRecords** - Here, application metadata are stored. These data contain the following characteristics: name, version, owner (developer's Ethereum address), release date, price (token), and a reference to retrieve additional data that may contain the application binaries (*Warehouse* base URL).
- **IIoTInstallationRecords** - This stores the information about the relationship of an app to the *EdgeDevices* on which it is installed. The following features are considered: deviceId, appId, purchaseDate, installDate, and a flag indicating whether the installation of an app is still active or the app has been uninstalled.
- **IIoTToken** - This enables payment and is derived from the EC20 standard token, which is standard implementation for currency management on Ethereum.

We use a private Ethereum Blockchain for performance reasons and to verify the participants of the ecosystem via the nodes as a means of creating transparency and visibility for all stakeholders in the Blockchain.

2) Fog Environment: An *Intel NUC* is the central point of communication and computation in *FogEnvironment*. Ubuntu runs as operating system on the *Intel NUC*. The *AppDevManager* and the *MQTT Broker* component are deployed on it.

IoT Manager: The *IoT Manager* acts as a proxy between the Cloud components and the components in the *FieldEnvironment*. The *IoT Manager* is implemented in Typescript and uses the Node.js JavaScript runtime. In its function as a proxy, the *IoT Manager* makes the applications from the *Warehouse* available to the respective *EdgeDevices*. The binaries from the

[1]https://aws.amazon.com/

[2]https://metamask.io/

[3]https://github.com/ethereum/web3.js

[4]https://kotlinlang.org/

[5]https://spring.io/

Warehouse are loaded to the *EdgeDevice* via the *IoT Manager* after a technician has completed the installation.

MQTT Broker: The purpose of the MQTT Broker is to distribute the data among all the components of the system. We use Mosquitto[6] as MQTT Broker.

3) Field Environment: Two subsystems are used in the *FieldEnvironment*: the *IIoT Bazaar App* which runs on an iPhone and the *IoT Simulation* component which runs on SIMATIC IOT2040[7] devices. iPhones with the *IIoT Bazaar App*, the *EdgeDevices* and the *Fog Nodes* must be available on the same local network to communicate.

IIoT Bazaar App: When the app starts, it first connects to the *MQTT Broker* and subscribes to MQTT topics. The IP address of the broker is resolved using mDNS. The MQTT messages contain information about nearby *EdgeDevices* and applications executed on them. A technician uses a smartphone camera to capture the surrounding environment. We trained a machine learning (ML) model to detect SIMATIC IOT2040 devices with an accuracy of 95 % in the AR View. When the model detects an *IOT2040 EdgeDevice*, it scans the device for a QR for identification. The technician can see which apps are installed on the device or which apps are available for installation via the messages in the *MQTT Broker* and via the query on the *App Chain*. We use Apple's ARKit[8] for the interaction in AR and integrate the ML model using CoreML[9]. We reused the ML model of [1]. The communication with the *App Chain* takes place via JSON RPC and was implemented in the app with the web3swift[10] library.

IoT Simulation: Three *EdgeDevices* with several sensors are available in our test environment. On each of the *EdgeDevices* the *IoT Simulation* is deployed. Like the *IoT Manager*, the *IoT Simulation* component is implemented in Typescript and uses the Node.js JavaScript runtime. The runtime environment allows us to add apps (in our case, sensors). The simulation is compatible with the visualizations in [1]. If the installation of an app works, the sensors and their values are visualized.

C. User Interface Design

In AR, the *IIoT Bazaar App* provides information about the identified *EdgeDevices* and their applications.

The interaction with the *EdgeDevice* in AR on the iPhone is important for the IIoT Bazaar concepts gaining acceptance by technicians.

We tested different user experience concepts in several iterations and evaluated them with our industrial partner. We finally decided on a drag & drop solution. The user interface adapts itself according to the camera angle and which *EdgeDevice* is in the focus of the AR view. If several *EdgeDevices* are available, the lowest common denominator of all the apps installed on these devices is displayed. Figure 4 shows how the Idle Time application is dragged from the static area below into the dynamic AR area. A visual confirmation gives the *EdgeDevice* a green frame around the device indicating that the *Technician* can install the app on the selected device. The installation process is initialized after the drop and visualized in the AR view (loading bar on the app icon). The *IoT Manager* takes care of the error-free installation and the billing of the app's costs.

Fig. 4. **AR User Interface:** A technician drags and drops the app to a detected edge device. The idle time application is about to be dragged onto the edge device *IoT2040 #2*. The technician must authenticate to start the installation process.

V. Conclusion

We extended the AIIoT approach [1] with real-time device interaction to obtain information about the installed applications and the provision to add or uninstall applications on edge devices. To view the historical data on the interactions with the edge device, we applied a Blockchain to store all interactions performed by a technician on the device to increase transparency and traceability. Furthermore, Smart Contracts enabling micropayments, including automation to create efficient and transparent processes were established.

The IIoT Bazaar prototype was successfully demonstrated to our industrial partners, who provided valuable feedback. In the

[6]https://mosquitto.org/

[7]https://w3.siemens.com/mcms/pc-based-automation/en/industrial-iot/Pages/Default.aspx

[8]https://developer.apple.com/arkit/

[9]https://developer.apple.com/documentation/coreml

[10]https://github.com/BANKEX/web3swift

case study, we addressed the requirements described in Section III-C by the various technologies and their combination: *(1) Open Platform*, *(2) End-to-end Delivery*, *(3) User-Centric Design*, *(4) Independence*, *(5) Payment Model Flexibility*, *(6) On- and Off-Site Remote Update Management*, *(7) Flexibility and Extensibility*, and *(8) Traceability* were each addressed and fulfilled by the different technologies.

However, we identified limitations of the current state of the prototype. During the case study, we encountered difficulties that had not been apparent in the design of the system. Due to faulty frameworks and the lack of expert knowledge, development with a Blockchain component proved to be particularly challenging. Even though we had used a private Proof-of-Authority Ethereum network, the confirmation times for the transactions were very high. Additionally, the evaluation of the scalability and the application of the IIoT Bazaar in a real-world scenario with multiple users are open issues.

The IIoT Bazaar concept offers the potential for human machine interactions, as we have shown by the case study. The concept could also be used and extended for machine machine collaborations by using Smart Contracts for autonomous machines that are used as both producers and consumers (ProSumer). For future research, we intend to evaluate the findings of the case study and examine the acceptance of the system. Furthermore, we plan to combine the IIoT Bazaar with the idea of Seamless Computing [25], [26] and use containers on all layers.

ACKNOWLEDGMENTS

We would like to thank those who contributed to our research: Philipp Diller, Moritz Kellermann, Deeksha Sinha, Phuong Mai Mai, Illia Ovchynnikov, Maksym Hrinenko, Tim Lewandowski, Ali Abbas Jaffri and Alina Turbina. Furthermore, we would like to thank Hubertus Hohl for his feedback and technical support.

REFERENCES

[1] A. Seitz, D. Henze, M. Sauer, J. Nickles, and B. Bruegge, "Augmenting the industrial internet of things with emojis," in *International Workshop on Smart Living with IoT, Cloud, and Edge Computing (SLICE)*, 2018.

[2] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, "On blockchain and its integration with iot. challenges and opportunities," *Future Generation Computer Systems*, vol. 88, pp. 173 – 190, 2018.

[3] K. Yeow, A. Gani, R. W. Ahmad, J. J. P. C. Rodrigues, and K. Ko, "Decentralized consensus for edge-centric internet of things: A review, taxonomy, and research issues," *IEEE Access*, vol. 6, pp. 1513–1524, 2018.

[4] H. Subramanian, "Decentralized blockchain-based electronic marketplaces," *Commun. ACM*, vol. 61, no. 1, pp. 78–84, 2017.

[5] S. Schneider, *The Industrial Internet of Things (IIoT)*, pp. 41–81. John Wiley & Sons, Inc., 2007.

[6] E. Fleisch and F. Mattern, *Das Internet der Dinge*. Springer-Verlag GmbH, 2005.

[7] C. Scheuermann, *A Metamodel for Cyber-Physical Systems*. Dissertation, Technische Universität München, München, 2017.

[8] A. V. Dastjerdi, H. Gupta, R. N. Calheiros, S. K. Ghosh, and R. Buyya, "Fog Computing: Principles, Architectures, and Applications," *CoRR*, 2016.

[9] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, MCC, pp. 13–16, ACM, 2012.

[10] A. Seitz, D. Buchinger, and B. Bruegge, "The conjunction of fog computing and the industrial internet of things - an applied approach," in *IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, March 2018.

[11] R. T. Azuma, "A survey of augmented reality," *Presence: Teleoperators and virtual environments*, vol. 6, no. 4, pp. 355–385, 1997.

[12] P. Milgram and F. Kishino, "A taxonomy of mixed reality visual displays," *IEICE TRANSACTIONS on Information and Systems*, vol. 77, no. 12, pp. 1321–1329, 1994.

[13] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," *www.bitcoin.org*, p. 9, 2008.

[14] X. Xu, I. Weber, M. Staples, L. Zhu, J. Bosch, L. Bass, C. Pautasso, and P. Rimba, "A Taxonomy of Blockchain-Based Systems for Architecture Design," *Proceedings - 2017 IEEE International Conference on Software Architecture, ICSA 2017*, pp. 243–252, 2017.

[15] M. Swan, *Blueprint for a new economy*. 2015.

[16] Buterin and Vitalik, "Ethereum White Paper: A Next Generation Smart Contract & Decentralized Application Platform," *Etherum*, 2014.

[17] N. Szabo, "Formalizing and securing relationships on public networks," *First Monday*, 1997.

[18] M. Alharby and A. van Moorsel, "Blockchain Based Smart Contracts : A Systematic Mapping Study," in *Computer Science & Information Technology (CS & IT)*, 2017.

[19] S. Huckle, R. Bhattacharya, M. White, and N. Beloff, "Internet of things, blockchain and shared economy applications," *Procedia computer science*, vol. 98, pp. 461–466, 2016.

[20] J. Pan, J. Wang, A. Hester, I. Alqerm, Y. Liu, and Y. Zhao, "EdgeChain: An Edge-IoT Framework and Prototype Based on Blockchain and Smart Contracts," 2018.

[21] P. Veena, S. Panikkar, S. Nair, and P. Brody, "Empowering the edge-practical insights on a decentralized internet of things," *Empowering the Edge-Practical Insights on a Decentralized Internet of Things. IBM Institute for Business Value*, vol. 17, 2015.

[22] M. Samaniego and R. Deters in *2016 IEEE International Conference on Computer and Information Technology (CIT)*.

[23] A. Seitz, F. Thiele, and B. Bruegge, "Fogxy - An Architecural Pattern for Fog Computing," in *Proceedings of the 23nd European Conference on Pattern Languages of Programs*, EuroPLoP '18, ACM, 2018.

[24] E. Marín-Tordera, X. Masip-Bruin, J. G. Almiñana, A. Jukan, G. Ren, J. Zhu, and J. Farre, "What is a fog node A tutorial on current concepts towards a common definition," *CoRR*, 2016.

[25] H. Mueller, S. V. Gogouvitis, A. Seitz, and B. Bruegge, "Seamless computing for industrial systems spanning cloud and edge," in *2017 International Conference on High Performance Computing Simulation (HPCS)*, pp. 209–216, July 2017.

[26] S. V. Gogouvitis, H. Mueller, S. Premnadh, A. Seitz, and B. Bruegge, "Seamless computing in industrial systems using container orchestration," *Future Generation Computer Systems*, 2018.

Fog Computing as Enabler for Blockchain-Based IIoT App Marketplaces - A Case Study

Conference Proceedings:
2018 Fifth International Conference on Internet of Things: Systems, Management and Security

Author: Andreas Seitz

Publisher: IEEE

Date: October 2018

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE

5 Discussion and Conclusion

This chapter concludes the dissertation, summarizes the contributions of each publication and discusses ideas for future work.

Distributed ledger technologies (DLTs) are a powerful technology for creating cross-organizational platforms that are organized in a decentralized manner.
In this dissertation, we used a formative research approach to investigate the integration of DLTs into the automotive value chain. DLTs address the need to enable multiple parties to exchange data across organizational boundaries, transparently and securely. We demonstrated the feasibility and value of DLT platforms within five application prototypes ranging from production and logistics to sales and services. The case studies were carried out in cooperation with industry partners, namely the BMW Group and Siemens, where we successfully linked their business requirements to the capabilities of DLTs. We have obtained comprehensive results showing that DLTs can make cross-organizational business processes more efficient and secure by using unique digital identities, tracing events along the value chain (e.g., ownership transfers), providing verifiable transactions and data, and removing intermediaries.

5.1 Contributions

Publication [A] realizes a cross-organizational platform for the tracking and tracing of parts and vehicles in multi-stage supply chains. The publication led to a pilot project at the BMW Group in 2019, which confirmed the feasibility and value of our part traceability application in a real-world, global supply chain. The pilot project involved two plants of the BMW Group and three plants of a 1-tier supplier. The pilot project was so successful that the BMW Group is scaling up the project to include more first- and additional second-tier suppliers in 2020.

Publication [B] describes a visionary scenario to create a decentralized marketplace and to enable machines to act as economically autonomous machines in that market-

place. The case study underlined that machines require verifiable digital identities to act as autonomous and trusted actors in a system. The publication stimulated a pilot project at the BMW Group with one first-tier supplier in the area of logistics. The pilot project concentrates on the integration of DLTs and IoT devices using secure hardware modules. The aim is to enable secure and trusted interactions by issuing and verifying digital identities for IoT devices. The results of this pilot project can be incorporated into the part traceability application (see Publication [A]).

Publication [C] describes a prototype that demonstrated that electric vehicles (EVs) can offer their flexibility in the form of deviations from their cost-optimal charging plan using a DLT based flexibility market platform. The prototype was shown to our industry partner and received positive feedback concerning the open flexibility market platform and the flexibility trading mechanism. This publication serves as the foundation for further electric vehicle (EV) projects within the BMW Group. The projects deal with the optimization of charging behavior, such as the stimulation of customers to use sustainable energy offers, and the certification of charging sessions, in particular for plug-in hybrid vehicles.

Publication [D] describes a visionary scenario that handles vehicle and traffic data in a secure and verifiable manner by using a combination of DLTs and distributed databases. This scenario stimulated a research project that investigates the integration of DLTs and differential privacy to ensure the compliant handling of customer and vehicle data.

5.2 Future Work

Successful DLT projects require the collaboration of business and information technology within and across organizations to move from prototypes to ecosystems. Thus, we recommend that further research should be undertaken in the following areas:

Standards. Industry and research institutes need to participate in consortia together to identify common business requirements and incentives, align on common applications, and create standards. To achieve broad adoption, future work should focus on standards that describe identities for organizations, vehicles (cf. [90]), and parts.

Governance. Currently, there are no blueprints with respect to organizational structure, responsibilities, financing models, and release models for DLT networks in the automotive industry. To improve this, organizational and technical processes for the on- and off-boarding of participants, particularly in private networks, need to be investigated. Particularly in DevOps environments, future research should investigate the integration and automation of DLT development in enterprises [89].

Scalability and performance. Future work should focus on DLT infrastructures that enable higher *scalability* (e.g., node scalability) and *performance* (e.g., number of transactions or latency), such as second layer solutions or directed acyclic graph.

Data privacy. The increased awareness of data privacy and ownership requires research of cryptographic techniques for secure communication between two parties. Within the next few years, cryptographic techniques, such as trusted execution environments and zero-knowledge proofs [91] for DLTs, will become essential to improve data privacy and ownership, in particular for customers. Also, guidelines to ensure compliance with the General Data Protection Regulation (GDPR) in DLT applications are required.

Acronyms

DAG directed acyclic graph.

DHT distributed hash table.

DLT distributed ledger technology.

EV electric vehicle.

IoT Internet of Things.

NONCE number only used once.

OEM original equipment manufacturer.

P2P peer-to-peer.

PBFT Practical Byzantine Fault Tolerance.

PoS Proof of Stake.

PoW Proof of Work.

SHA secure hash algorithm.

SPOF single point of failure.

List of Figures

Bibliography

[1] McKinsey. (2016). How blockchains could change the world, [Online]. Available: https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/how-blockchains-could-change-the-world. (accessed: 20.05.2020).

[2] R. Kaplinsky and M. Morris, *A handbook for value chain research*. University of Sussex, Institute of Development Studies Brighton, 2000.

[3] B. M. Beamon, "Supply chain design and analysis: Models and methods," *International journal of production economics*, pp. 281–294, 1998.

[4] D. M. Lambert, M. C. Cooper, and J. D. Pagh, "Supply chain management: Implementation issues and research opportunities," *The international journal of logistics management*, pp. 1–20, 1998.

[5] J. T. Mentzer, W. DeWitt, J. S. Keebler, S. Min, N. W. Nix, C. D. Smith, and Z. G. Zacharia, "Defining supply chain management," *Journal of Business logistics*, pp. 1–25, 2001.

[6] M. E. Porter, *Competitive advantage of nations: Creating and sustaining superior performance*. Simon and Schuster, 2011.

[7] D. Waters, *Logistics An Introduction to supply chain management*. Palgrave Macmillan, 2003.

[8] S. Serdarasan, "A review of supply chain complexity drivers," *Computers & Industrial Engineering*, pp. 533–540, 2013.

[9] D. R. Towill, "Simplicity wins: Twelve rules for designing effective supply chains," *Control, March*, pp. 9–13, 1999.

[10] P. Childerhouse and D. R. Towill, "Reducing uncertainty in european supply chains," *Journal of Manufacturing Technology Management*, pp. 585–598, 2004.

[11] R. Mason-Jones and D. R. Towill, "Shrinking the supply chain uncertainty circle," *IOM control*, pp. 17–22, 1998.

[12] T. Blecker, W. Kersten, and C. M. Meyer, "Development of an approach for analyzing supply chain complexity," in *Mass Customization: Concepts–Tools–Realization. Proceedings of the International Mass Customization Meeting*, 2005, pp. 47–59.

[13] F. Isik, "Complexity in supply chains: A new approach to quantitative measurement of the supply-chain-complexity," *Supply chain management*, pp. 417–432, 2011.

[14] A. Y. Dak, S. Yahya, and M. Kassim, "A literature survey on security challenges in vanets," *International Journal of Computer Theory and Engineering*, pp. 1007–1010, 2012.

[15] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark, "Connected vehicles: Solutions and challenges," *IEEE Internet of Things Journal*, pp. 289–299, 2014.

[16] C. Liu, K. T. Chau, D. Wu, and S. Gao, "Opportunities and challenges of vehicle-to-home, vehicle-to-vehicle, and vehicle-to-grid technologies," *Proceedings of the IEEE*, pp. 2409–2427, 2013.

[17] W. Su, H. Eichi, W. Zeng, and M. Chow, "A survey on the electrification of transportation in a smart grid environment," *IEEE Transactions on Industrial Informatics*, pp. 1–10, 2012.

[18] M. Yilmaz and P. T. Krein, "Review of benefits and challenges of vehicle-to-grid technology," in *2012 IEEE Energy Conversion Congress and Exposition (ECCE)*, 2012, pp. 3082–3089.

[19] E. C. Eze, S. Zhang, and E. Liu, "Vehicular ad hoc networks (VANETs): Current state, challenges, potentials and way forward," in *2014 20th International Conference on Automation and Computing*, 2014, pp. 176–181.

[20] V. Schlatt, A. Schweizer, N. Urbach, and G. Fridgen, "Blockchain: Grundlagen, Anwendungen und Potenziale," 2016.

[21] C. Catalini and J. S. Gans, "Some simple economics of the blockchain," National Bureau of Economic Research, Tech. Rep., 2016.

[22] S. Davidson, P. De Filippi, and J. Potts, "Blockchains and the economic institutions of capitalism," *Journal of Institutional Economics*, pp. 639–658, 2018.

[23] S. Nakamoto, *Bitcoin: A peer-to-peer electronic cash system*, 2008.

[24] J. Yli-Huumo, D. Ko, S. Choi, S. Park, and K. Smolander, "Where is current research on blockchain technology? A systematic review," *PloS One*, 2016.

[25] X. Xu, I. Weber, M. Staples, L. Zhu, J. Bosch, L. Bass, C. Pautasso, and P. Rimba, "A taxonomy of blockchain-based systems for architecture design," in *IEEE International Conference on Software Architecture (ICSA)*, 2017.

[26] V. Buterin, *Ethereum: A next-generation smart contract and decentralized application platform*, 2014.

[27] J. Schütte, G. Fridgen, W. Prinz, T. Rose, N. Urbach, T. Hoeren, N. Guggenberger, C. Welzel, S. Holly, A. Schulte, P. Sprenger, C. Schwede, B. Weimert, B. Otto, M. Dalheimer, M. Wenzel, M. Kreutzer, M. Fritz, U. Leiner, and A. Nouak, "Blockchain und Smart Contracts : Technologien, Forschungsfragen und Anwendungen," 2017.

[28] G. Salviotti, L. M. De Rossi, and N. Abbatemarco, "A structured framework to assess the business application landscape of blockchain technologies.," in *51st Hawaii International Conference on System Sciences*, 2018.

[29] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *2017 IEEE International Congress on Big Data (BigData Congress)*, 2017, pp. 557–564.

[30] M. C. Ballandies, M. M. Dapp, and E. Pournaras, "Decrypting distributed ledger design-taxonomy, classification and blockchain community evaluation," *arXiv preprint arXiv:1811.03419*, 2018.

[31] T. Frick and C. Reigeluth, "Formative research: A methodology for creating and improving design theories," *Instructional-design theories. Hillsdale, NJ: Lawrence Erlbaum Associates*, pp. 633–652, 1999.

[32] R. E. Stake, *Standards-based and responsive evaluation*. Sage Publications, 2003.

[33] P. Baran, "On distributed communications networks," *IEEE Transactions on Communications Systems*, pp. 1–9, 1964.

[34] A. S. Tanenbaum and M. Van Steen, *Distributed systems: Principles and Paradigms*. Prentice-Hall, 2007.

[35] A. Bonifati, P. K. Chrysanthis, A. M. Ouksel, and K.-U. Sattler, "Distributed databases and peer-to-peer databases: Past and present," *ACM SIGMOD Record*, pp. 5–11, 2008.

[36] D. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, and Z. Xu, "Peer-to-peer computing," Tech. Rep., 2002.

[37] A. M. Ouksel and C. F. Naiman, "Coordinating context building in heterogeneous information systems," *Journal of Intelligent Information Systems*, pp. 151–183, 1994.

[38] C. R. d'Oliveira, "An analysis of computer decentralization," Massachusetts Institute of Technology Cambridge Lab For Computer Science, Tech. Rep., 1977.

[39] J. E. Israel, J. G. Mitchell, and H. E. Sturgis, "Separating data from function in a distributed file system," Tech. Rep., 1978.

[40] R. Schollmeier, "A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications," in *Proceedings First International Conference on Peer-to-Peer Computing*, 2001, pp. 101–102.

[41] M. T. Özsu and P. Valduriez, *Principles of Distributed Database Systems, Third Edition*. Springer, 2011.

[42] R. Steinmetz and K. Wehrle, "P2P systems and applications," *Springer Lecture Notes in Computer Science*, 2005.

[43] A. Kemper and A. Eickler, *Datenbanksysteme - Eine Einführung, 7. Auflage*. Oldenbourg, 2009.

[44] Q. H. Vu, M. Lupu, and B. C. Ooi, *Peer-to-Peer Computing - Principles and Applications*. Springer, 2010.

[45] B. Carlsson and R. Gustavsson, "The rise and fall of napster - an evolutionary approach," in *International Computer Science Conference on Active Media Technology*, Springer, 2001, pp. 347–354.

[46] C. Gkantsidis, M. Mihail, and A. Saberi, "Hybrid search schemes for unstructured peer-to-peer networks," in *2005 IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, 2005, pp. 1526–1537.

[47] H. Balakrishnan, M. F. Kaashoek, D. Karger, D. Karger, R. Morris, and I. Stoica, "Looking up data in P2P systems," *Commun. ACM*, pp. 43–48, 2003.

[48] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," *SIGCOMM Comput. Commun. Rev.*, pp. 161–172, 2001.

[49] M. Swan, *Blockchain: Blueprint for a new economy*. O'Reilly Media, Inc., 2015.

[50] M. J. M. Chowdhury, A. Colman, M. A. Kabir, J. Han, and P. Sarda, "Blockchain versus database: A critical analysis," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering*, 2018, pp. 1348–1353.

[51] P. Tasatanattakool and C. Techapanupreeda, "Blockchain: Challenges and applications," in *2018 International Conference on Information Networking (ICOIN)*, 2018, pp. 473–475.

[52] W. Stallings, *Cryptography and network security, 4/E*. Pearson Education India, 2006.

[53] R. C. Merkle, "Protocols for public key cryptosystems," in *1980 IEEE Symposium on Security and Privacy*, 1980, pp. 122–122.

[54] F. M. Beni and I. P. arko, "Distributed ledger technology: Blockchain compared to directed acyclic graph," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, IEEE, 2018, pp. 1569–1570.

[55] H. Pervez, M. Muneeb, M. U. Irfan, and I. U. Haq, "A comparative analysis of dag-based blockchain architectures," in *2018 12th International Conference on Open Source Systems and Technologies (ICOSST)*, 2018, pp. 27–34.

[56] J. Huang, L. Kong, G. Chen, M. Wu, X. Liu, and P. Zeng, "Towards secure industrial iot: Blockchain system with credit-based consensus mechanism," *IEEE Transactions on Industrial Informatics*, pp. 3680–3689, 2019.

[57] A. Churyumov. (2016). Byteball: A decentralized system for storage and transfer of value, [Online]. Available: https://byteball.org/Byteball.pdf. (accessed: 20.05.2020).

[58] C. LeMahieu. (2018). Nano: A feeless distributed cryptocurrency network, [Online]. Available: https://nano.org/en/whitepaper. (accessed: 20.05.2020).

[59] S. Popov, "The tangle," *cit. on*, p. 131, 2016.

[60] L. S. Sankar, M. Sindhu, and M. Sethumadhavan, "Survey of consensus protocols on blockchain applications," in *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2017, pp. 1–5.

[61] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Trans. Program. Lang. Syst.*, pp. 382–401, 1982.

[62] S. King and S. Nadal, "Ppcoin: Peer-to-peer crypto-currency with proof-of-stake," *self-published paper, August*, 2012.

[63] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, *et al.*, "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proceedings of the Thirteenth EuroSys Conference*, 2018, pp. 1–15.

[64] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access*, pp. 22 328–22 370, 2019.

[65] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei, and C. Qijun, "A review on consensus algorithm of blockchain," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, IEEE, 2017, pp. 2567–2572.

[66] I.-C. Lin and T.-C. Liao, "A survey of blockchain security issues and challenges.," *IJ Network Security*, pp. 653–659, 2017.

[67] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in *International conference on financial cryptography and data security*, Springer, 2014, pp. 436–454.

[68] L. Bach, B. Mihaljevic, and M. Zagar, "Comparative analysis of blockchain consensus algorithms," in *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2018, pp. 1545–1550.

[69] M. Castro and B. Liskov, "Practical byzantine fault tolerance and proactive recovery," *ACM Transactions on Computer Systems (TOCS)*, pp. 398–461, 2002.

[70] L. Lamport, "The implementation of reliable distributed multiprocess systems," *Computer Networks (1976)*, pp. 95–114, 1978.

[71] M. Pease, R. Shostak, and L. Lamport, "Reaching agreement in the presence of faults," *J. ACM*, pp. 228–234, 1980.

[72] L. Lamport, "The weak byzantine generals problem," *J. ACM*, pp. 668–676, 1983.

[73] M. Castro and B. Liskov, "Practical byzantine fault tolerance," in *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, 1999, pp. 173–186.

[74] E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin," in *2014 IEEE Symposium on Security and Privacy*, IEEE, 2014, pp. 459–474.

[75] S. Noether, "Ring signature confidential transactions for Monero," *IACR Cryptology ePrint Archive*, p. 1098, 2015.

[76] U. Mukhopadhyay, A. Skjellum, O. Hambolu, J. Oakley, L. Yu, and R. Brooks, "A brief survey of cryptocurrency systems," in *Proc. Security and Trust (PST) 2016 14th Annual Conf. Privacy*, 2016, pp. 745–752.

[77] R. F. A. Britto and D. Schwartz. (2012). Ripple.

[78] IO, EOS. (2017). Eos. io technical white paper, [Online]. Available: https://github.com/EOSIO/Documentation. (accessed: 20.05.2020).

[79] D. Khovratovich and J. Law. (2017). Sovrin: Digital identities in the blockchain era, [Online]. Available: https://sovrin.org/wp-content/uploads/AnonCred-RWC.pdf. (accessed: 20.05.2020).

[80] V. Buterin and V. Griffith, "Casper the friendly finality gadget," *arXiv preprint*, 2017.

[81] G. Greenspan. (2015). Ultichain private blockchain-white paper, [Online]. Available: http://www.multichain.com/download/MultiChain-White-Paper.pdf. (accessed: 20.05.2020).

[82] M. Alharby and A. Van Moorsel, "Blockchain-based smart contracts: A systematic mapping study," *arXiv preprint*, 2017.

[83] V. Morabito, "Business innovation through blockchain," *Springer International Publishing*, 2017.

[84] F. Idelberger, G. Governatori, R. Riveret, and G. Sartor, "Evaluation of logic-based smart contracts for blockchain systems," in *International Symposium on Rules and Rule Markup Languages for the Semantic Web*, Springer, 2016, pp. 167–183.

[85] N. Szabo, "Formalizing and Securing Relationships on Public Networks," *First Monday*, 1997.

[86] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *Ieee Access*, pp. 2292–2303, 2016.

[87] A. Beniiche, "A study of blockchain oracles," *arXiv preprint*, 2020.

[88] K. Korpela, J. Hallikas, and T. Dahlberg, "Digital supply chain transformation toward blockchain integration," in *Proceedings of the 50th Hawaii international conference on system sciences*, 2017.

[89] S. Porru, A. Pinna, M. Marchesi, and R. Tonelli, "Blockchain-oriented software engineering: Challenges and new directions," in *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*, 2017, pp. 169–171.

[90] Mobility Open Blockchain Initiative. (2019). Vehicle identity standard, [Online]. Available: https://dlt.mobi/wp-content/uploads/2020/04/Preview-MOBI-Vehicle-Identity-Standard-v1.0.pdf. (accessed: 20.05.2020).

[91] I. Gudymenko, A. Khalid, H. Siddiqui, M. Idrees, S. Clauss, A. Luckow, M. Bolsinger, and D. Miehle, *Privacy-preserving blockchain-based systems for car sharing leveraging zero-knowledge protocols*, 2020.