Rethinking People Analytics With Inverse Transparency by Design

VALENTIN ZIEGLMEIER, Technical University of Munich, Germany
ALEXANDER PRETSCHNER, Technical University of Munich, Germany

Employees work in increasingly digital environments that enable advanced analytics. Yet, they lack oversight over the systems that process their data. That means that potential analysis errors or hidden biases are hard to uncover. Recent data protection legislation tries to tackle these issues, but it is inadequate. It does not prevent data misusage while at the same time stifling sensible use cases for data.

We think the conflict between data protection and increasingly data-driven systems should be solved differently. When access to an employees' data is given, all usages should be made transparent to them, according to the concept of *inverse transparency*. This allows individuals to benefit from sensible data usage while addressing the potentially harmful consequences of data misusage. To accomplish this, we propose a new design approach for workforce analytics software we refer to as *inverse transparency by design*.

To understand the developer and user perspectives on the proposal, we conduct two exploratory studies with students. First, we let small teams of developers implement analytics tools with inverse transparency by design to uncover how they judge the approach and how it materializes in their developed tools. We find that architectural changes are made without inhibiting core functionality. The developers consider our approach valuable and technically feasible. Second, we conduct a user study over three months to let participants experience the provided inverse transparency and reflect on their experience. The study models a software development workplace where most work processes are already digital. Participants perceive the transparency as beneficial and feel empowered by it. They unanimously agree that it would be an improvement for the workplace. We conclude that inverse transparency by design is a promising approach to realize accepted and responsible people analytics.

CCS Concepts: • **Human-centered computing**; • **Security and privacy** → **Social aspects of security and privacy**; • **Software and its engineering** → *Designing software*;

Additional Key Words and Phrases: Data sovereignty, Privacy by design, HR analytics, Qualitative study

ACM Reference Format:
Valentin Zieglmeier and Alexander Pretschner. 2023. Rethinking People Analytics With Inverse Transparency by Design. *Proc. ACM Hum.-Comput. Interact.* 7, CSCW2, Article 292 (October 2023), 29 pages. https://doi.org/10.1145/3610083

1 INTRODUCTION

Workplaces are becoming increasingly digital. Everything from employee communication to the status of tasks and work items is stored and handled digitally. This enables advanced people analytics that process employee data to speed up processes or help with decision-making. But, contrary to the consumer context, employees often have no choice which tools to use in their

Authors' addresses: Valentin Zieglmeier, valentin.zieglmeier@tum.de; Alexander Pretschner, alexander.pretschner@tum.de, Technical University of Munich, TUM School of Computation, Information and Technology, Chair of Software and Systems Engineering, Boltzmannstr. 3, 85748 Garching, Germany.

2573-0142/2023/10-ART292 $15.00
https://doi.org/10.1145/3610083

workplace. Furthermore, the data processing is opaque to those subjected to it. This lack of control and oversight by employees raises concerns [85] and means that there is little recourse in case of data misusage or discriminatory analysis errors. With increasingly automated and automatic decision-making, the risks of data misusage rise further [36, 54]. Data-based insights can play a role in deciding if a person should be invited for a job interview, if they should be assigned to a project, or qualify for a promotion [87]. It is therefore vital that any discrimination or misusage of data can be uncovered and challenged [72].

To protect individuals' privacy and ensure accountability, data protection legislation is employed [29, 73]. Depending on the cultural context and underlying trust model, it takes different shape [13, 63]. A traditional approach is *detective enforcement*, which relies on self-regulation and voluntary codes. It assumes many data usages to be benign and enables them by default. Terms of use or non-disclosure agreements enable *ex post* reaction to misusage [61, 63]. This optimistic solution is common in many parts of the United States of America [73]. Recent privacy legislation such as the 2016 General Data Protection Regulation (GDPR) [34] of the European Union and the 2018 California Consumer Privacy Act (CCPA) [16] goes beyond voluntary codes to implement formal privacy regulation and provide individuals more control over their data [13, 73]. They require the implementation of technical measures that prevent extraneous data usage. This can be seen as a move towards *preventive enforcement*, which only permits data usage for purposes specified in advance. Thereby, it strives to prevent misusage *ex ante* [65, 89].

We think this increased protection is important, but in many cases insufficient to prevent data misusage in the workplace, while at the same time stifling sensible use cases for data. Four factors are, in our view, mainly responsible for this. To start with, (1) opting out of data sharing is not always possible for employees. If the data processing is necessary for core business processes, it does not require consent. And the power asymmetry between employees and management and often forced usage of digital technologies in the workplace mean that, even if employees legally have the right to object, denying consent may effectively remain a theoretical possibility. In case there is a choice, though, (2) use cases for data are becoming more complex, making it harder for individuals to fully understand the impact of giving access to their data. The lengths of typical privacy policies[1] show the complexity of data processing, meaning a full understanding is questionable [see also 57, 84]. This is exacerbated by the fact that (3) software tools are not static products. Software-as-a-service and agile programming mean that software evolves continuously [37, pp. 19–20]. Even if individuals had the capacity to understand how data are processed by their employer, their knowledge can therefore quickly become obsolete. Finally and importantly, (4) a blanket decision for or against data sharing cannot always be made, as the usage context is an important decision factor. Data that are given away can be used in *unexpected* and *unintended* ways [43, 69], and hence be misused from the perspective of the data subject. This can happen intentionally, by trickery or hiding of essential information, or unintentionally, by misreading or misrepresenting those data. Faced with these concerns, overwhelmed by choice and a lack of oversight, and backed by laws such as the GDPR and CCPA, employees might therefore aim for absolute data minimization to lower perceived risks. This ideal certainly reduces the potential for misusage, but opting out could lead to other disadvantages, such as lack of access to data-driven features. In addition, it becomes difficult for companies to utilize data beyond cases in which they are absolutely necessary, restraining legitimately helpful data usages and stifling research and innovation in the big data space [31, 45, 92].

We think that this issue can be solved differently, drawing inspiration from Brin [14]. They describe a dystopia in which citizens are monitored by the police during their every move, making

[1]Google's for example, when viewed as a PDF, is 30 pages long: https://www.gstatic.com/policies/privacy/pdf/20210701/7yn50xee/google_privacy_policy_en_eu.pdf (last accessed 2022-01-20)

their lives transparent. To balance this, they propose to empower individuals by letting them watch over their watchers—providing what they call *inverse transparency*. [14] This idea has been previously proposed as a new digital leadership concept, aiming to solve tensions between managers and employees [35]. Continuing these thoughts, we envision making all usages of employees' data visible (transparent) to them. We think that getting an overview of how their data are used empowers individuals to gain true *data sovereignty* [43], meaning "self-determination [...] with regard to the use of their data" [44, p. 550]. This has the potential to improve their trust to allow data usages that might be beneficial to them. In addition to helping to uncover misusage retroactively which enables *accountability* [91], this transparency could also increase the *felt accountability* of data consumers [39], thereby deterring data misusage even before it occurs.

Some data, such as health or genetic data, will always warrant preventive enforcement due to their high sensitivity. Furthermore, due to the power asymmetry in the workplace, the added transparency alone is not sufficient to protect individuals. Additional safeguards, such as strong workers' councils or appropriate recourse in case of data misusage [see, e.g., 58, p. 36], may therefore be a prerequisite for this idea. Given those, we think it could be a promising solution to the conflict between data protection and data-based use cases.

To concretize our vision, we consider the example of software developers in an IT company. These employees can work remotely, with some companies even adopting "all remote" configurations [20]. This can increase the employer's interest in monitoring employees with people analytics. As software developers are in high demand in the labor market, though, the inherent power asymmetry between them and their employer is reduced. Data about these employees are stored in various systems and accessed through a multitude of tools. In this scenario, employees track their work in issue tracking software and use a workplace messenger. That means that data exist about the specific technologies and problems they work on, as well as whom they collaborate with. The traditional detective enforcement allows utilizing these data for, e.g., managerial decision-making or collaboration between colleagues. However, it makes room for profiling and patronization of employees based on data that might not represent the full picture or be inadequate for these uses. Employees might be fired or discriminated against due to misinterpreted or misused data, and have no recourse against it. With preventive enforcement on the other hand, any data usage beyond those required by core work processes is forbidden. This makes it difficult to implement systems enabling advanced data-based use cases. Yet, as we have deliberated above, misusage of data is not sufficiently prevented. If we now imagine the same example with the envisioned transparency over data usages, those issues are addressed. Employees are free to collaborate without any overhead, and data can be utilized for company-level decision-making. Should data be misused and harmful consequences for an employee arise, they have access to an audit trail. To defend themselves, they can make it available to their workers' council or a lawyer to support their case.

In this paper, we explore the idea of enshrining inverse transparency into people analytics from their conception. We aim to understand how this could change software design and, by extension, foster employees' trust in and acceptance of sensible data usage processes. Our goal is to facilitate data-based use cases that can be beneficial for individuals, while better protecting them from misusage of their data. Evolving the idea of *privacy by design* [18], we describe the software development paradigm of *inverse transparency by design*. As an empirical contribution, we conduct two exploratory studies with students in a controlled environment to understand the developer and user perspectives. In the first, we explore the implications of our approach for software design. To that end, we let small teams of student developers implement various analytics tools based on the principles of inverse transparency by design. We then analyze and discuss the changes they make to their tools to meet transparency requirements, and how they judge the approach in their reflections. In our second study, we consider the perspective of data subjects

on our concept. Therefore, we conduct a controlled laboratory study with students that mirrors the real-world use case of a software development department. Participants worked for three months in a workplace-like setting utilizing transparency-enabled people analytics. We examine and deliberate the user experience and personal perspectives of participants. Note that, for both studies, we worked with university students in a controlled environment. This was an intentional choice to best study a fundamental change in the work and interactions of employees. Thereby, we remove potential confounding factors [see, e.g., 12, 60] to support internal validity [28] and establish causality [47]. The artificial nature of the studies may limit external validity, though. We discuss the impact and our reasoning in more detail in Section 6.

In all, we contribute a comprehensive conceptualization and preliminary evaluation of inverse transparency by design, a new approach for workplace software development. When studying the developer perspective, we find that building with inverse transparency by design does not inhibit core functionality, with developers considering the concept practical. In our study of the user perspective, participants experiencing inverse transparency in practice find it beneficial and feel empowered by it. Given the choice, they would unanimously opt for it in their workplace. We conclude that moving towards incorporating inverse transparency by design is a promising direction for people analytics.

2 RELATED WORK

We begin by describing other ideas besides inverse transparency that aim to find a middle ground between necessary data protection and sensible uses for personal data. Then, we give an overview of related works that specifically propose to provide transparency over data usages to ensure accountability. Finally, we narrow down further to the technical realization. We discuss other works that aim to change personal data processing such that individual data usages are tracked.

2.1 Balancing Data Protection and Sensible Data Usage

The conflict between data protection and sensible data usage has been considered in various works. Cate observed that "many uses of personal information pose no risk of harm to individuals, while creating significant benefits for data subjects and society more broadly" [17, p. 37]. As a solution, they propose *use-based privacy*, which entails defining collective norms that specify acceptable uses [11, 17]. For example, these could explicitly exclude discriminatory uses of data where they are obvious. Then, the norms are enforced automatically by systems that process data [see, e.g., 11]. A similar strategy underlies *distributed usage control*, which instead of collective norms aims to enforce user-defined usage policies [62] with comparable technical challenges [see, e.g., 90].

Independently of who defines the usage policies, though, it is in our view impossible to know all acceptable usage patterns for data in advance. Therefore, we consider it important to enable more flexibility and instead provide individuals with inverse transparency, allowing them direct oversight over how their data are used. Potential misusage of data can then be handled retroactively, ensuring accountability. Still, the idea to establish collective norms defining appropriate usage of data, enshrined for example in laws or company agreements, is compelling. They could serve as a minimum safeguard for individuals, with inverse transparency helping to protect them for data usage that goes beyond the basic use cases.

2.2 Providing Transparency Over Data Usages to Ensure Accountability

Our work is not the first to introduce the idea of *inverse transparency*, or more broadly aiming to ensure accountability by giving individuals oversight over usages of their data. The general concept of inverse transparency was originally conceived of and presented by Brin [14] (see Section 1). In the software development context specifically, similar concepts have been developed, also aiming to

provide transparency to ensure accountability. An important predecessor to our work is the paper on *hippocratic databases* by Agrawal et al. [2]. They discuss the usefulness of giving individuals access to audit trails of databases holding their information, allowing them to detect misusage [2]. Weitzner et al. also deliberate the potential benefits of making data usages transparent to individuals, achieving what they refer to as *information accountability* [91]. They see two main advantages over the status quo: First, reducing individuals' mental load as they do not have to judge *ex ante* all potential usages of their data. Second, enabling redress in case of harmful misusage of data [91].

These works serve as a motivation and theoretical foundation to our work. We build on their ideas to propose a new approach for people analytics development: inverse transparency by design. As our added contribution, we study its potential effects on software development and users.

2.3 Changing Personal Data Processing to Integrate Usage Logging

Finally, we discuss technical solutions related to our concrete idea how to ensure that employees are provided transparency over all usages of their data: by rethinking people analytics with inverse transparency by design. People analytics today are typically designed the same way as any other business analytics: the data they operate on are collected elsewhere and presumed as given [42]. In a sense, the tools consider these data as a mere resource, with employees under analysis reduced to data sources. Data flow in one direction only: from employees to the tools that analyze them [see, e.g., 25, 56, 76]. This is exemplified by how SAP visualizes the data processing pipeline for their SuccessFactors[2] product, a large people analytics suite: On a high level, they depict a one-way pipeline from various data sources, such as "human resources," into their analytics platform [76, p. 3]. Considering that people analytics analyze humans and not business processes, though, this lacking consideration of individuals' interests has been critically reflected upon [see, e.g., 36, p. 417].

We propose to rethink people analytics design to, conceptually, add a reverse data flow into this process. By integrating data usage tracking into the tools and sending the logs back to the employees under analysis, we give them a view into how their data are analyzed. To our knowledge, we are the first to propose such a rethink of people analytics design. Conceptually comparable ideas have been proposed previously, though. As notable examples, Sundareswaran et al. [83] and Bagdasaryan et al. [7] describe implementations of usage logging that could enable inverse transparency. The *CIA framework* [83] is based on packaging personal data together with a usage logging module in Java JAR files before giving access to them [83]. *Ancile* [7] on the other hand is an online privacy platform. Bagdasaryan et al. identify many of the same challenges that we see. Their *Ancile* server considers Python analysis scripts that are sent to it for execution in a sandbox [7]. Both implementations are relevant to learn from and can be seen as potential instantiations of inverse transparency in software. Yet, they both consider neither the developer nor the user perspective. On the one hand, understanding the developer perspective is important to judge the consequence of transparency requirements on software development. Is it realistic to expect software to be written based on a new paradigm? Could this requirement inhibit development of innovative features? Previous research on privacy by design has recognized the importance of considering the developer perspective to judge the viability of software design principles, as their success directly depends on developers' actions [see, e.g., 38, 79]. On the other hand, considering the user perspective is a vital aspect of privacy and empowerment. While in theory the idea of making visible usages of data can sound obvious and like a clear benefit, we need to deliberate the effect of this transparency on individuals. Does it foster their trust and increase their acceptance of people analytics? Or could it lead to new concerns or other negative consequences for them? To close these gaps, we present results from two empirical studies, which are our novel contribution compared to these works.

[2] https://www.sap.com/products/hcm/workforce-planning-hr-analytics.html

3 CONCEPT

Currently, data usage processes in the workplace happen without oversight of the employees concerned by them. To tackle this, we think that people analytics should be built with inverse transparency by design, a next step after privacy by design. This means they should be built in such a way that data usages can be traced back and attributed. As a first step, we discuss our idea as a theoretical concept, outlining the basic requirements for inverse transparency. Second, we deliberate the potential implications for software design. Finally, we discuss the problems of usage log integrity and system trustworthiness as necessary prerequisites for our concept to function.

From here on, instead of talking specifically about employees or managers, we will refer to the more generic concepts of *data owners* and *data consumers*. This reflects that these roles might be reversed or even inhabited by the same person (accessing their own data). We follow the definition by Pretschner et al., who state that for each datum, there is a *data owner* [64]. They "[possess] the rights to the data" [64, p. 40].[3] They also define the role of the *data consumer* [64]. We personify the data consumer in our concept. In cases of algorithmic data usages, the processing system could be considered the data consumer.

3.1 Requirements for Inverse Transparency

The basis of inverse transparency is to give data owners an overview of all data usages. That requires (1) monitoring every usage of data, (2) verifying the authenticity of these events and storing them, and (3) making this information transparent to data owners. According to the separation of concerns, we can imagine each requirement being fulfilled by individual tools, but the functionality can also be integrated into the operating system or directly into the software that provides data access.

To enable the three steps of our vision, let us therefore consider three (conceptual) components:

(1) *Monitor*: Track data usages
(2) *Safekeeper*: Store monitored usages
(3) *Display*: Make stored usages transparent

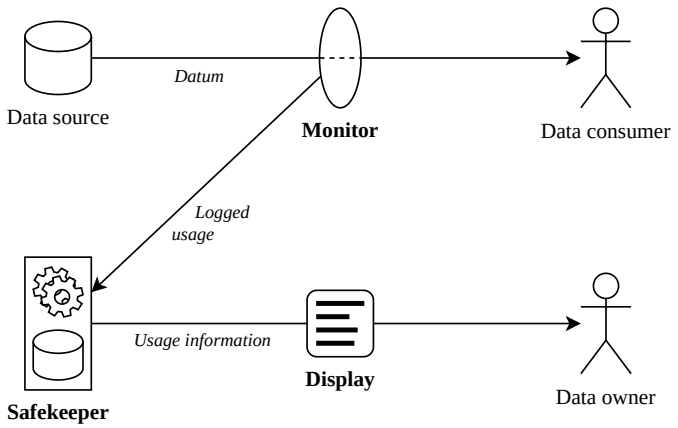

Fig. 1. Providing inverse transparency on a conceptual level. The data consumer accesses a datum from the data source. Their usage is logged and stored. The data owner can now retrieve the logged usage information.

[3]For example, in case the GDPR applies, this would correspond to the "data subject," meaning the individual identifiable through the data [34, Art. 4].

This process is visualized in Figure 1. On this level of abstraction, we can already postulate: Usage tracking requires oversight over all data processing. Therefore, integrating it into the analysis tool itself is ideal. That matches our concept that these tools be built with inverse transparency by design. But it seems as if the tasks of storing monitored usages and making them transparent could be extracted and shared between multiple tools. In that case, we could reduce development effort of tool developers and potentially increase security, as only one database would need to be protected. For data owners, having a single tool to watch over how their data are used has the potential to reduce mental load and may therefore be preferable, too.

3.2 Implications for Software Design

In order to make all data usages transparent, we need to not only track all occurring usages, but also prevent circumvention of the framework logging. Therefore, it is important to ensure that data never leave controlled environments. Accessing them may only be allowed through tools that provide inverse transparency, thereby ensuring that usages are logged [91]. This is why we envision inverse transparency by design: No software tool should be distributed without enabling this transparency. In our concept, this means that tools need to, on a conceptual level, include a *Monitor* that tracks data processing. We can then imagine companies running their own, private log store providing a standardized API for adding and retrieving log entries. Tools they license or deploy are then required to integrate into the company inverse transparency infrastructure, sending the logged data usages to their private *Safekeeper* API.

At first glance, this seems reasonable. Yet, when we look closer, we find that data are usually not "sent" to data consumers, but continuously move between (sub) systems, are aggregated, copied, converted, or moved. A "data usage" may for example just be a software tool starting up—simply to show a data-driven start page, data accesses can be necessary. Therefore, we have to consider what a data usage is and how software developers can ensure that it is logged. For most cases, our concept of developing with inverse transparency by design deals exactly with this problem: Instead of trying to retrofit usage logging to existing software, we expect developers to already recognize the interests of data owners in the design phase. In some cases, though, building with inverse transparency in mind may require a paradigm shift. Specifically, consider a scenario that we refer to as *ambient usage* of data. We illustrate this with a typical home page of an analytics tool. On load,

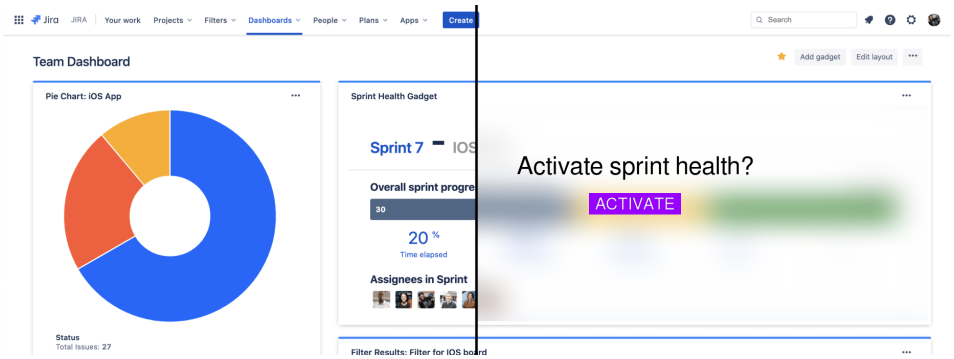

Fig. 2. Example of how interaction paradigms may change as a consequence of inverse transparency, illustrated at the example of Jira Software (image source: Atlassian; modifications: the authors). On the left side, the status quo in Jira: analyses are run and insights presented automatically when the tool is opened, a pattern we refer to as *ambient usage* of data. The right side illustrates how inverse transparency may transform this paradigm: the insights are hidden and need to be explicitly activated, signaling intent by the data consumer.

the tool presents multiple widgets that represent analyses, ready for the data consumer to view (see Figure 2, left side). If we now consider inverse transparency, merely opening the tool already would result in a multitude of recorded data usages, even if none of these widgets are viewed by the data consumer. Data owners may be unnecessarily worried about data usages, and data consumers may need to justify their presumed interest. Conversely, this plausible deniability would make it nigh impossible to differentiate if the data consumer actually made use of the analyses or not. A potential consequence of this may be a fundamental shift in interaction paradigms. Instead of presenting all possible data up-front for data consumers to view, the tool may present blurred windows with a button to explicitly "show" the respective analysis (see Figure 2, right side). Thereby, the consumer specifically expresses their intent to access the data shown, with the corresponding log entry being written in the background. We can conclude that, in the long term, inverse transparency may result in software being implemented with a more mindful approach to data usage, moving away from the currently common arbitrary data processing.

3.3 Usage Log Integrity

After considering data flow tracking, we shift our focus to the usage logs and the *Safekeeper*. The integrity (completeness and correctness) of the stored logs is a central requirement for inverse transparency to function. Only then can we achieve accountability of data consumers. Considering completeness, we have deliberated how building software with inverse transparency by design ensures that any data usage can be detected. Yet, even if our approach is adopted immediately, there will be a transitional period, which means we need to contemplate how to enable usage logging for existing systems that have not been built this way yet. Indeed, this exact issue has been worked on extensively in the context of usage control. Concretely, the research on distributed usage control [see, e.g., 48, 63, 64] tackles this problem and can be seen as a complement to ours. For example, Lörscher showed how to implement the necessary data flow tracking for the Thunderbird mail client [53]. Recently, application sandboxing has been proposed to achieve similar goals without requiring changes to the monitored tool [50]. We believe that a combination of distributed usage control and our approach is sufficient to cover all potentially occurring data usages, providing reasonable completeness of the logs.

Therefore, we now consider conceptual attacks on our stored usage log. We find five abstract approaches to attack the log integrity that can be realized through three main attack vectors (see Figure 3).

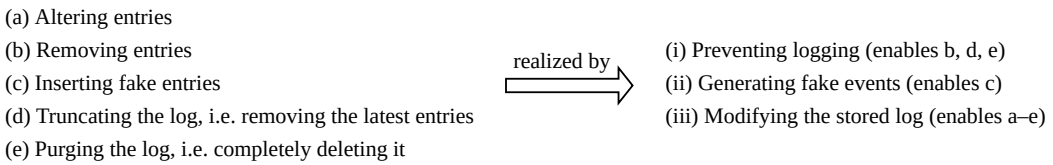

Fig. 3. Conceptual attacks on the usage log [following 3] and their realization.

In this paper, we do not present detailed technical solutions for these attacks. Yet, we should consider if any of these attacks may make our idea conceptually infeasible before we continue. And, while we cannot conclusively judge the applicability of these attacks on this level of abstraction, we do find sensible solutions for them that suggest they might be manageable. For (i), a reasonable expectation would be DDoS attacks. In literature, we find a number of papers and books describing

how to address and prevent such attacks [e.g., 9]. For (ii), we find two attacks reasonable to expect: constructing a realistic-looking fake log event or replaying past events. Again, existing literature presents various ways to approach and prevent such attacks. On the one hand, to prevent attackers from simply faking log events, applications should sign requests to ensure their authenticity [see 23]. Simple strategies to mitigate replay attacks, on the other hand, are also well researched and seem reasonably easy to implement [e.g., 6]. Accordingly, we consider (i) and (ii) to be mainly engineering challenges for the specific instantiation, yet ultimately solvable.

Attack variant (iii) meanwhile is much broader and can encompass various strategies, including hardware-based attacks such as removing a hard disk containing the logs from a server rack. The field of secure logging, which is concerned with mitigating such attacks, is vast. That means we can only roughly sketch potential avenues that exist to protect against various kinds of malicious log modifications. For example, in our research we found existing solutions based on trusted computing modules [see, e.g., 1, 52, 67], cryptographically secured protocols such as forward-secure authentication [see, e.g., 3, 41, 55], or distributed ledger technology, i.e. blockchain [see, e.g., 33, 77, 97]. Recent challenges seem to include heterogeneity of hardware [52] and GDPR compliance, especially considering the right to erasure [96]. Yet, these are robust approaches that claim to guarantee integrity and even confidentiality of the stored logs against reasonably powerful adversaries. That means, while there exist some open research questions, we consider the problem of log integrity to not be in fundamental conflict with our concept.

Of course, this glosses over one important caveat: malicious data consumers. As soon as data are provided to users, even within a monitored environment, our control over it ends. How do we deal with data export functionality or with tools that store data on disk, meaning these data may be accessed by malicious data consumers without utilizing a monitored tool? In short, this may not be necessary at all. While it is still common today to allow users access to the underlying data and files, businesses are quickly moving towards a future of cloud software, provided only as a "service" [37]. There, data are merely an enabler of features, with tools moving away from the idea of "files" as containers for data that are manually handled by users. The corresponding loss of control for users has become so significant that legislation such as the GDPR specifically includes rights to data export and portability for users' own data. While this trend to software-as-a-service arguably reduces users' data sovereignty, it allows us to consider the problems of data exports or files stored outside our control to be solved. And while it naturally is, in any setting, nigh impossible to prevent data consumers from simply taking a picture of the screen [65] or even just memorizing its contents [66], we argue that this does not significantly reduce the value of the provided transparency. On the one hand, measures to control this would be questionable in their effectiveness and highly invasive (e.g., eye tracking), making them unreasonable to consider. On the other hand, large-scale copying or exporting of data is not feasible this way, making it only a theoretical issue.

3.4 Trustworthiness

From a technocratic worldview, it might seem as though the only relevant factor to concern us to enable inverse transparency would be the log integrity discussed above. After all, as long as everything is logged in a tamper-proof way, data owners should be satisfied. This is not the case, though. To enable accountability, we need data owners to accept and trust the provided transparency, and, as a consequence, make use of it. Above, we have deliberated how the conceptual tasks of *Monitor* and *Safekeeper* can enable inverse transparency by protecting log integrity. In the following, we consider the third part, the *Display*, representing the user interface making stored usages transparent to individuals.

Our goal is to enable data owners to make use of the transparency provided to them, enabling their data sovereignty. We find two potential obstacles: individuals not being able to operate the provided transparency interface or understand its contents (usability), and not being able to trust it (trustworthiness). While it is clear that usability is important, user trust in technology is equally relevant, considering that it influences users' intention to use, adoption, and continued use of a tool [71, 82]. We have already covered an important prerequisite for data owners to trust the provided transparency—ensuring the integrity of the log—but we should not underestimate the influence of the design of the user interface on their trust. The way information is presented, framed, and the human factors that surround a system's implementation or rollout are of high relevance for user trust [27, 95]. To illustrate this: If a system is built to be cryptographically secure, but users are not informed of this fact or do not trust the messenger, this technical fact alone will do little to improve their trust in the system.

Following Zieglmeier and Lehene, we therefore consider the three trust dimensions of *purpose* (the intended use of a system), *process* (how it operates), and *performance* (how well it solves its tasks) that are relevant for user interface design [95, pp. 2–3]. As the operation of the system—its process and performance—depend on the concrete instantiation, we therefore focus on its perceived *purpose* here, a facet that could threaten our idea on a conceptual level already. In short, the purpose dimension of trust reflects "the impression of the designer's intentions that users get from interacting with the system" [95, p. 2]. It becomes clear that the intended use of the transparency system is core to its trustworthiness. When introducing an inverse transparency infrastructure, a company needs to ensure that employees trust the system to serve their goals and not those of the company. If this is not addressed, the system may in the worst case not be accepted by employees at all, therefore rendering it meaningless. Accordingly, we need to consider whether there are ways to ensure, on a conceptual level, that the provided transparency is experienced as trustworthy by individuals. When researching approaches to this problem, we find multiple promising solutions. For example, to improve trust, a reputable or well-known (third) party can be made responsible for the development and operation of the transparency-enabling systems or certify their correctness, thereby targeting the trust antecedents of reputation and familiarity [40]. If available, this could be the company-internal workers' council, or alternatively an external workers' rights organization. Should that not be possible, the use of open-source software or code audits can reduce the necessity of interpersonal trust [4, 32]. We find both approaches to be reasonable and realistic to implement. Therefore, we conclude that the trustworthiness of the transparency system is not in fundamental conflict with our concept.

4 STUDY A: SOFTWARE DEVELOPER PERSPECTIVE

Our concept of software being built with inverse transparency by design necessitates a behavioral and mindset change in software developers. Therefore, it is important to consider their perspective to understand potential conceptual issues early. If developers find the concept infeasible to implement in practice, we need to address their issues first before we can continue. Furthermore, we strive to learn about the implications of our concept for software design. As with any restriction on how software should be developed, this may hinder innovative features, or enable completely new solutions to problems not imagined before. We therefore present a preliminary study of the developer perspective. It is designed to answer the following research questions:

(A.1) How does developing people analytics with inverse transparency by design manifest itself in their architecture?
(A.2) How is the approach of inverse transparency by design judged by developers?

4.1 Study Approach

We opted for an artificial setting designed to closely mirror a real-world software development use case. To that end, we created a university practicum spanning over three months with 12 computer science master's students as participants. By fully controlling the setting, we were able to choose the concrete tasks worked on and allow participants appropriate time and resources to implement software according to our principles, establishing causality [81]. Working with students, meanwhile, allowed us to best test our initial hypotheses [86]. Thereby, we strove to exclude entrenched mental models [88] and development culture [5] as confounding factors. For our case of applying a new approach for the first time, students have been shown to perform comparably to professionals [74].

Participants were tasked to develop software in agile development teams. Four groups of developers were formed, considering individuals' skill level and technology preferences. The team members had never worked together before, removing any potential influence of an existing development culture. In the first two months, three teams were tasked to develop people analytics with inverse transparency by design (covering the *Monitor* in our concept), while the fourth team implemented and improved the necessary auxiliary tools for the concept (*Safekeeper* and *Display*) based on feedback from the other teams. In the final month, participants used the developed analysis tools to analyze their own data collected in the months prior. This allowed them to experience the analyses both from the perspective of a developer and a user.

The development tasks for the analytics teams were derived from a set of use cases that we developed with our industry partner. Teams independently chose their concrete task from the set based on technical skills and interests. The provided use cases covered potential data sources as well as relevant insights. Thereby, we made sure that the developed analytics were grounded in practice. For a scientific grounding, meanwhile, teams were tasked to read relevant academic literature as part of their development and link those insights to their implementation decisions.

To foster critical reflection and deliberation among the developers in our study, we instructed them to integrate the ethical deliberation for agile processes (EDAP) [98] into their work process. EDAP is a methodology to interweave ethical deliberation with traditional agile development projects. The goal is to introduce normative deliberation to developers about which technical direction is preferable for their software [98, pp. 11–12]. We found the EDAP schema to be an effective tool to foster reflections and deliberations from participants about their software development projects. The development teams were instructed how to perform an analysis following the EDAP scheme. They were tasked to update their report bi-weekly during the implementation phase. After the last week of development, the deliberation report was finalized and submitted. In addition, after the conclusion of the study, teams submitted their code as well as a longer written analysis reflecting on the impact of inverse transparency on their implementation.

4.2 RQ A.1: How does developing people analytics with inverse transparency by design manifest itself in their architecture?

The feasibility and implications of inverse transparency by design as a development approach can be seen in the architecture of people analytics developed according to the principle. Therefore, we investigate each of the three artifacts implemented by the analytics development teams in our study. To preface, we find that no team encountered fundamental issues with the process and all could implement their envisioned projects.

4.2.1 Team G: Version Control Software Analysis Tool. The first team implemented a history analysis for Git[4] commits in a standalone application. One example for an analysis is the *commit hours*

[4] https://git-scm.com

overview. It summarizes for each developer the number of commits per hour of the day. The analyses are implemented as tabs that the data consumer can switch between. To implement inverse transparency, they include an additional screen in their tool that may not be necessary otherwise: a selection screen for the requested analysis. It manifests a simplistic solution for the issue of *ambient usage* in the context of inverse transparency that we deliberated above (see Section 3.2). Only those analyses that the data consumer explicitly selects are loaded, with a usage log immediately created for every selected one (see Figure 4). After the analysis report has been created, the tool logs no additional accesses even if it is opened again, which is interesting albeit inconsequential. We consider this the most basic form of implementing inverse transparency: the interaction paradigm and provided features do not have to change, instead the tool adds a step to the process before starting the data processing and just records any potential data usage immediately.

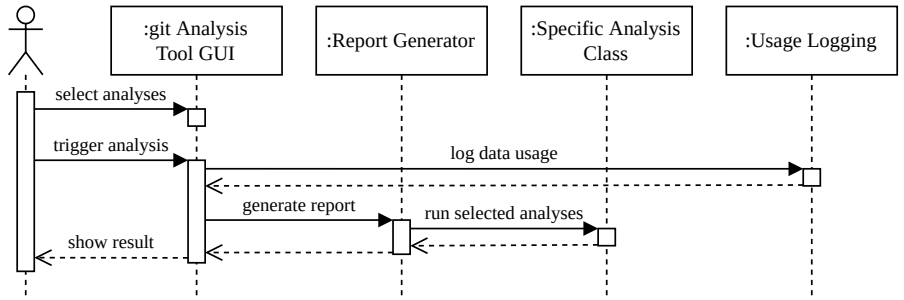

Fig. 4. Sequence diagram of how team G realized inverse transparency. Data consumers have to explicitly select analyses before any data are processed. This data usage is logged once, before the report is generated.

Team G worked together with team S on their implementation of inverse transparency. They implemented a shared library that was then integrated by both into their code. Hereby, we could already see potential synergies when software is implemented with inverse transparency by design: Basic functionality does not have to be reimplemented for every tool and may instead be shared, minimizing the development overhead.

4.2.2 Team J: Issue Tracking Software Analysis Plugins. Team J implemented four analysis gadgets for Jira Software,[5] distributed as one plugin. One example is the *supporter analysis*. It ranks team members by who performed most code reviews. Data consumers can then choose any of the gadgets independently to be added to their main application dashboard, where they have to be explicitly activated to run. Considering its architecture, the artifact implements inverse transparency in a notable fashion—a single module provides both data retrieval and usage logging functionality, allowing the various plugins to share this code. Thereby, developers do not need to consider inverse transparency for every newly created analysis. In the front end code, this manifests as a single function invocation that logs the data usage via the shared back end (see Figure 5). After it completes, the tool continues its operation. On failure, it shows an error message, ensuring that the logging successfully completes before data are presented to the data consumer.

4.2.3 Team S: Chat Analysis Tool. Team S implemented a standalone analysis tool for Slack[6] workspaces. For example, one analysis their tool can perform is the *network analysis*. It creates a social network graph based on who is "mentioned" by whom in their messages. Various analyses such as this can be chosen on the main screen and then independently triggered with a button. For

[5] https://www.atlassian.com/software/jira
[6] https://slack.com/

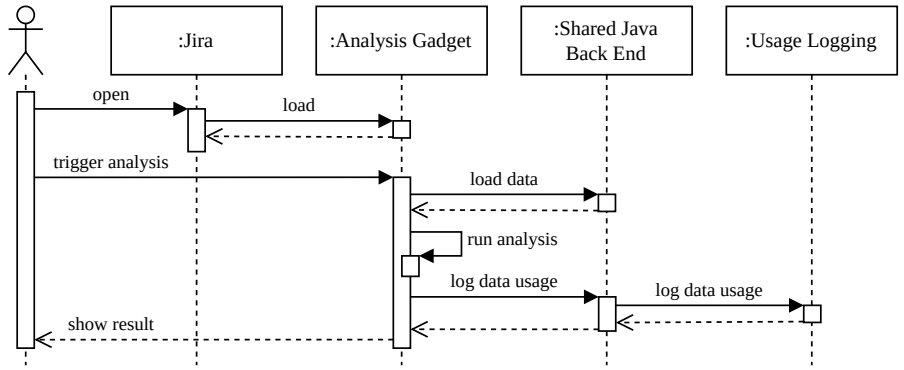

Fig. 5. Sequence diagram of how team J realized inverse transparency. A shared back end component serves as the single point for accessing data and logging usages. Contrary to team S' instantiation, the usage is logged based on the analysis *result*, not the consumer's *request*.

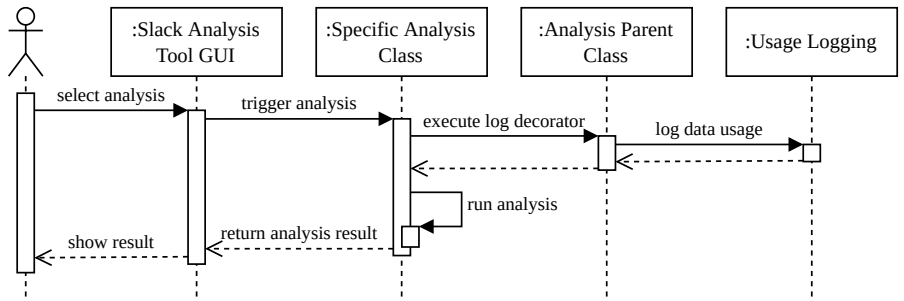

Fig. 6. Sequence diagram of how team S realized inverse transparency. The log decorator logs usages after the analysis is triggered but before it is run. The architecture prevents new analyses to be implemented without usage logging.

team S, thinking with inverse transparency by design meant they built their whole architecture around the concept. First, their program code is structured as such that no new analysis function can be added without inverse transparency. If occurring data usages are not logged before an analysis is run, the code triggers an error that causes the application to close. Second, to implement this logging, they provide a code tag ("decorator") that can be added to new analysis functions (see Figure 6 and Listing 1) This tag then handles the usage logging automatically. That means that

Listing 1. Simplified source code of team S. Their architecture is structured so that, to provide inverse transparency, they only need to decorate the function performing the data analysis with `@Analysis.log_data_usage`. This decorator ensures that the data usage is logged whenever the analysis is requested.

```
class NetworkAnalysis(Analysis):

    @Analysis.log_data_usage
    def perform_analysis(self) -> Result:
        # ...
```

developers adding new analyses only need to add a single line to their program code to integrate inverse transparency.

To run an analysis, data consumers select it from the application's main screen. Only when an analysis is explicitly triggered, the data usage is logged, after which the results are shown.

4.3 RQ A.2: How is the approach of inverse transparency by design judged by developers?

Second, we consider developers' perspectives on the idea of embedding inverse transparency into people analytics. To that end, we analyze the ethical deliberations and reports of the development teams. We find that, in general, the teams show neutral or positive sentiment towards developing with inverse transparency by design.

In the following, we start by investigating the deliberations of all three analytics development teams combined. We reference the source of each quotation by referring to the respective team's letter in parentheses, e.g. (J) for team J. Then, we shift our focus to the fourth team, tasked with developing the auxiliary tools for inverse transparency, notably the user-facing *Display*.

4.3.1 Analytics Developers. To start with, the developers recognize the value of the various implemented analytics, noting that they can be "desirable for both a comfortable and effective work environment" (J). They highlight that the tools can provide "a more objective view" and help "verify [ones] hypotheses about a team" (G). This shows that our participants do not fundamentally oppose the idea of implementing people analytics. Yet, they also point to the risk of such tools, as their insights could be considered "privacy invasive" (J), "flawed", or "biased" (G). Furthermore, they could "become a [...] self-fulfilling truth" if relied on unquestioned (S).

To counter the risks to some degree, "access control and inverse transparency with regards to data access is desirable" (J). Integrating inverse transparency by design, they argue, "increases transparency" and "gives control to the owners of data" (J). One important concern with people analytics are indirect negative effects, such as users adapting their behavior to conform to expectations. This issue is explicitly acknowledged by team S, and they argue that "inverse transparency [...] should help reduce [this] pressure to conform for the users." (S). One potential reason for this is that inverse transparency is judged to be capable to "avoid misuse of [the tool]" (J), which is "likely to reassure the user of the safety of their data" (J) and "[face] the concerns of data owners" (G).

Regarding the technical realization with inverse transparency by design, the developers note no issues. To the contrary, team S argues that they "demonstrated [...] that inverse transparency is a viable concept which can, at least from a technical standpoint, work in practice" (S). This matches our findings in the analysis of the implemented artifacts.

Providing additional protection through inverse transparency is judged to be valuable and technically feasible. Yet, this may also have unexpected side effects on development decisions. Team S worked on analyses that can be considered, in parts, relatively invasive. They acknowledge this in their EDAP report, noting a conflict between "stakeholder evaluation of / knowledge about employees" versus "privacy of workers, freedom of surveillance". As one solution for this conflict though, they note that "inverse transparency [...] ensures that the employee is at all times informed of the extent of the analysis" (S). This hints at an unexpected aspect of developing with inverse transparency by design: It might actually serve as an enabler of features commonly considered privacy-invasive or delicate, as it could to an extent counteract their negative consequences. We deliberate this point further in Section 6.

4.3.2 Inverse Transparency Tool Developers. Second, we analyze the deliberations of the fourth team, tasked with implementing the auxiliary inverse transparency tools. Most interesting is their

work on the user-facing *Display*, which makes data usages visible to data owners. For them, this interface can be seen as the manifestation of inverse transparency.

There is always a manipulative element when designing a user interface. Choosing if and how to display certain information naturally influences how it is perceived. The team recognizes that concern. They note that, on the one hand, data owners of course need guidance "on how [the data] should be interpreted." Yet, providing such guidance by, e.g., coloring specific values has a risk of "arbitrarily highlight[ing] certain usage scenarios and not others." This could be especially critical if data consumers are unaware of being singled out. Therefore, they propose to assist data consumers in understanding how their logged data usages will be displayed. As an example, one could "provide documentation on what presentation [they] can expect as a result of a certain action."

The team discusses such issues under the notion of *fairness* towards the data consumer. They warn that even "facts [...] presented by the system [...] alone may convey the wrong impression of a data consumer which could violate the principle of fairness." Therefore, they argue for allowing data consumers to provide explanations for data usages "to justify their actions." Yet, they also recognize that data consumers may have an interest to manipulate and therefore "caution has to be taken." We find this a very important issue that is critical to deliberate. Providing inverse transparency necessarily also leads to increased transparency over data consumers' actions. Including their perspective in the design of the transparency tools is therefore, in our view, necessary.

4.4 Conclusions

In our study of the developer perspective, we find that developing people analytics with inverse transparency by design is technically feasible. This can be seen from the artifacts as well as developers' reflections. The teams furthermore consider inverse transparency capable to counter some of the risks of people analytics. Considering their implementations of inverse transparency, team J's and team S' architectures are completely built around the usage logging. This prevents accidental circumvention of the logging and facilitates the integration of new analyses with inverse transparency. Refactoring an existing codebase this way retroactively, meanwhile, could mean significant additional effort and technical risks [see, e.g., 80]. Additionally, all implemented artifacts require data consumers to explicitly select and trigger analyses before the results are shown. Thereby, they counteract the issue of *ambient usage* of data that we identify in Section 3.2, as this step prevents unintended data usage.

These findings match our expectation that building with inverse transparency by design has noticeable influence on the architecture and interaction paradigm of people analytics. Accordingly, they reaffirm our approach compared to retrofitting transparency, and show its feasibility.

5 STUDY B: USER PERSPECTIVE

After considering the developer perspective, we shift our focus to the users in a second preliminary study. Inverse transparency can only unfold its full potential if users find it helpful and make use of it to enable accountability. Therefore, our second study is focused on assessing the effect that inverse transparency by design can have on employees and how they might experience it. Our research questions are:

(B.1) How is a unified inverse transparency dashboard judged by participants?
(B.2) Do data owners find inverse transparency beneficial and feel empowered by it?
(B.3) Do participants consider inverse transparency to be capable to influence data consumers' data usage behavior?
(B.4) Is inverse transparency considered valuable in general by participants?

5.1 Study Approach

We conducted a laboratory study with students, which was specifically designed for our research objectives. Inverse transparency represents a fundamental change in the working environment and interactions of employees. Introducing it to an existing workplace could have triggered confounding change management issues, most notably individual's resistance to change [12], which can limit the success of change initiatives [51]. By working with students and fully controlling the study environment, we could ensure that inverse transparency was actually experienced and used over an extended period of time [47]. Furthermore, having removed external influences of an existing workplace, such as schedule pressure [60], means that we can cleanly attribute any observed effects to inverse transparency instead of potential confounders [28]. Accordingly, we created a university practicum running over three months for our study. To increase the representativeness of the study environment, we emulated the real-world use case of a remote software development workplace as closely as possible. That means having small teams of developers build software in agile teams, each lead by a team lead. Work items and their status were tracked in the issue tracking software Jira Software, messages were exchanged over the business messenger Slack, and code was versioned with the distributed version control software Git. Teams only interacted through digital collaboration tools, representing an all-remote configuration.

In total, 15 master's or final year bachelor's students in computer science were tasked to conduct work in the setting described above. Working with computer science students specifically allowed us to realistically model the software development use case. They were split up in four groups of four students each, one with three (see Figure 7). Each group decided on a team lead to guide the development process and serve as the data consumer. We explicitly gave team leads the task to conduct data-driven management, utilizing analysis tools built with inverse transparency by design to analyze their team members' data and write reports on their performance. Due to the all-remote configuration, team leads had to rely on these analytics to get a full picture of their teams. The other team members worked as developers, representing the data owners in our concept. They were instructed to specifically utilize those data-driven collaboration tools that their team leads' analysis tools were compatible with. That means they worked with Jira, Slack and Git, as described above. The data collected in these tools were then made available to their respective team leads to analyze via people analytics built with inverse transparency by design.

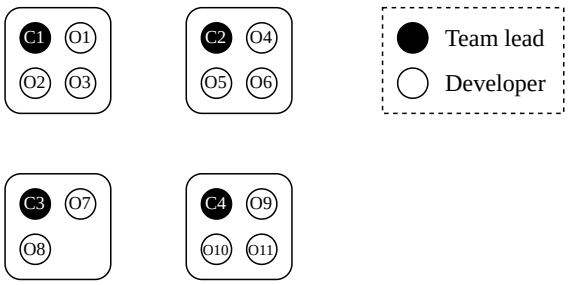

Fig. 7. Participant organization in our study ($n = 15$). Three teams of four and one of three were formed. Each team had one team lead, serving as a data consumer (C1–C4), with the rest of the team working as developers, representing data owners (O1–O11).

We employ a mixed-method evaluation design. During the process, regular written self-reflections were submitted. In addition, participants answered two questionnaires after the study, one of which covers their experience with inverse transparency over the course of the study, the other their

opinion of inverse transparency in general (the questions are listed in Appendix A.2). Our findings summarize the results from all of these evaluations.

5.2 Workflow and Utilized Tools

The teams in study B worked with people analytics that were built by the developers in study A with inverse transparency by design. The employed workflow, visualized in Figure 8, was an instantiation of our concept (see Figure 1 on page 6), with developers representing data owners and team leads representing data consumers. Developers used the tools Jira, Slack, and Git for their work and collaboration. The data collected in these tools were then accessed by the team leads via the people analytics created in study B by team J, team S, and team G. As these analytics were built with inverse transparency by design, any data usage was logged and stored in a database (the *Safekeeper*) to be made available to the data owners. To access usage information relating to their data, developers were given access to a transparency dashboard (the *Display*) that provides a direct view into the tracked usage logs. This dashboard was developed by the fourth team in study A as part of the auxiliary transparency tools, in addition to the *Safekeeper* that stored usage data, and a single sign-on server. Collectively, we refer to these tools as the *inverse transparency toolchain* [93].

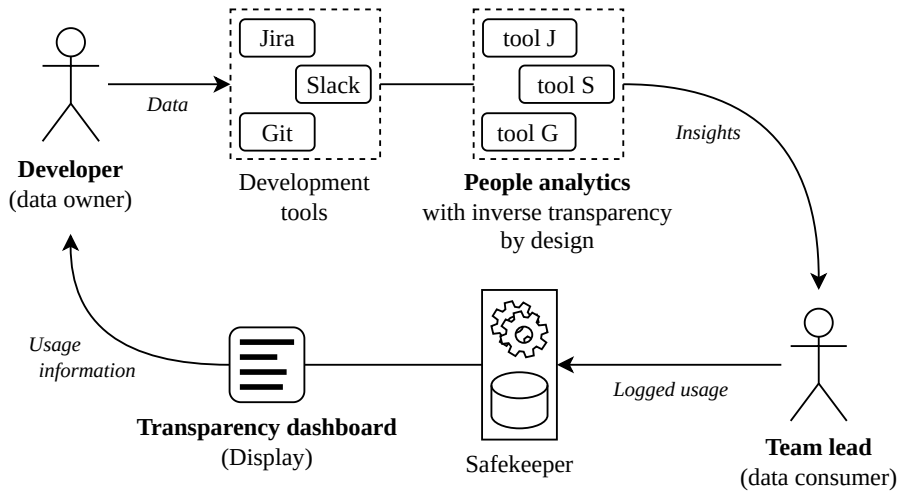

Fig. 8. The workflow and utilized tools in study B. Developers worked with development tools that collect data on their work and collaboration. These data were then analyzed by the team leads with the people analytics created in study A. The tools logged their data usage, storing this data in the *Safekeeper*. The usage information was then made available to developers via the transparency dashboard.

The purpose of the transparency dashboard is to make usage information available to data owners. It is a standalone tool that unifies all logged usage data relating to an individual, independently of the concrete analytics used. All study participants got accounts for the inverse transparency toolchain that they could use to log into the transparency dashboard. To illustrate our following elaborations on the dashboard, find a screenshot of it in Appendix A.1. After logging in, users find two views into the data: an overview area and a detailed table with individual entries. The overview area is meant to give a quick view how one's data were used in the past days. To find out exactly what was accessed, by whom, and when, the table and its filtering functionality can be used.

5.3 RQ B.1: How is a unified inverse transparency dashboard judged by participants?

To begin with, we evaluate how participants judge having one unified inverse transparency dashboard. Alternatively, individual dashboards could be integrated into each tool. We theorize in Section 3.1 that having a single point summarizing all usage information could be preferable.

To exclude potential usability issues that may have impacted participants' experience, we used the commonly applied system usability scale (SUS) [15] (Q1–Q10). The aggregated SUS score of *81.67* indicates that no fundamental usability issues arose during use [see 8], supporting our qualitative results. Then, we asked participants to formulate freely if they enjoyed using the dashboard and what could have been improved (Q11 & Q12). Both questions aim to uncover benefits and issues of our approach of providing a unified dashboard.

Participants unanimously expressed positive sentiment about the unified transparency dashboard, describing it as "useful" (O11, C3), "simple and easy to use" (O2) and noting that it "gets things done fast" (O3). Referring to its concrete value, data owners note that "being able to see [...] data access patterns from managers was very interesting" (O8) and that "it is very useful to have an overview of the data usage" (O11). This ability to detect usage patterns, by getting an overview of data usages, is facilitated by all data being presented in one unified dashboard. For data consumers, meanwhile, this may provide a different benefit. One notes that the unified dashboard was "a useful tool when trying to determine whether my data accesses were logged properly" (C3). As we discuss in our study of the developer perspective (see Section 4.3.2), data consumers may want to verify that their data usages are reported correctly. Having a single dashboard that unifies all collected data can make this easier.

When asked what could be improved, participants mentioned a need for more visual summaries of the recorded accesses (O1, O2, O11, C4) and tool tips or explanations for what is displayed (O4, O7). Providing visual summaries can be especially useful when unifying data from multiple sources into one dashboard, as this may help uncover suspicious usage patterns. Two responses support this point by referring to our approach of a unified dashboard directly, with both stressing its value. In fact, these participants wished for even more of the utilized tools to be integrated into the transparency infrastructure: "Every module [...] should integrated [sic!] into [it]" (O3), as this "would be very interesting" (O8). This confirms our vision; we are convinced that inverse transparency can unfold its full potential only if it is implemented as a foundational paradigm underlying all people analytics, not just a selected few. Our results suggest that a unified dashboard can facilitate this.

5.4 RQ B.2: Do data owners find inverse transparency beneficial and feel empowered by it?

Inverse transparency as a concept mainly targets data owners—those who provide data for others to use. As described above, the data consumers in our study were explicitly instructed to utilize tools built with inverse transparency by design to analyze data collected from Jira Software, Slack, and Git. These data were generated by data owners as side effects of their work. The data usages were then recorded and made available to the data owners on a dashboard to provide them with inverse transparency.

In our questionnaire, we therefore asked the eleven data owners (developers) among our participants to answer on a 5-point Likert scale if they found this transparency helpful and useful (Q13 & Q14). We find that data owners show very positive sentiment towards inverse transparency, judging it as beneficial (see Figure 9).

To understand participants' individual perspectives, we followed up with a free text question about their experience with the provided transparency (Q15). In general, participants expressed

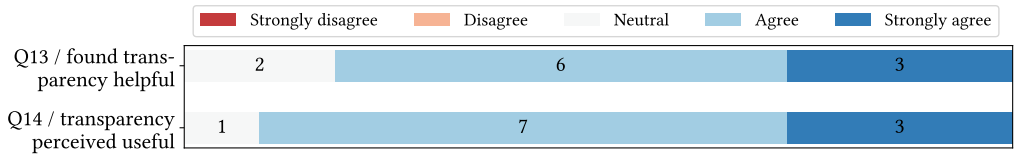

Fig. 9. Results of the questions for data owners ($n = 11$) about how helpful and useful they perceived inverse transparency to be. They were asked to express their agreement on a five-point Likert scale (X-axis), from 1 for "strongly disagree" to 5 for "strongly agree."

positive sentiment, explaining that the transparency "makes me feel safe" (O1) and that "it is a great approach" (O4) or "a good tool" (O10). "I personally like to use it" (O11), wrote one, with another expressing: "I experienced it well" (O5). Notably though, one participant had a very different experience. They write: "This transparancy [sic!] makes me even more aware of the monitoring, so im [sic!] not able to forget it." (O9). Thereby, they touch upon an important point. Many data usages may be benign, but being confronted with them may still have an influence. For most of our participants, the additional transparency induced a feeling of safety, but some may find that it actually creates a sense of worry and a feeling of being monitored. We reflect this thought in Section 6.

5.5 RQ B.3: Do participants consider inverse transparency to be capable to influence data consumers' data usage behavior?

Receiving transparency over how data are used is judged as helpful and useful by data owners. Yet, the question arises if this transparency influences data consumers, as it might elicit a feeling of "being watched." Potential chilling effects could be desired (by preventing data misusage), or problematic (if legitimate usage of data is hindered). Therefore, we asked participants about the potential influence of inverse transparency on data consumers' data usage behavior.

First, we asked the data owners ($n = 11$) if they consider inverse transparency a meaningful deterrent for potential data misusage (Q16). We find just one participant disagreeing with the statement, with all others agreeing (4/11) or agreeing strongly (6/11). In a free text answer, the critical respondent reveals that they consider the "transparency [to be] high but not high enough to be bad", judging the risk of misusage in our concrete setting to be low in general. They add: "I believe if usage of data is within good intention. It does not effect user behavior." This suggests that they might have misunderstood the question to be about our concrete scenario, not the theoretical possibility of misusage.

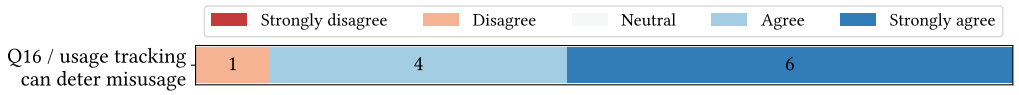

Fig. 10. Results of the question to data owners if they think that inverse transparency can deter misusage. Data owners ($n = 11$) were asked to express their agreement on a five-point Likert scale (X-axis), from 1 for "strongly disagree" to 5 for "strongly agree."

Second, we asked the data consumers ($n = 4$) to elaborate if the monitoring of their data usage behavior influenced their actions (Q17). Two consumers were sure, answering "I believe so" (C3)

and "Definitely" (C2). Both explained that they think the monitoring reduced the number of their accesses. "Without having to worry about potentially having to explain the frequency of my accesses to the data owners, I would likely have conducted the analyses more often" (C3). The third consumer responded that it only influenced them "barely, but I can't be 100% sure" (C1), with the final participant being sure that the answer would be no: "I would have acted the same" (C4). All of our participants could not know, but it is notable that opinions diverge so strongly. With two participants clearly worrying about the impression of their data usage on data owners, while another did not even consider it, other confounding factors may have been responsible. For example, the team culture with regards to data-driven management may change the acceptance of the team lead utilizing analyses. We conclude that inverse transparency is capable, but not guaranteed, to influence data consumers in their data usage behavior.

5.6 RQ B.4: Is inverse transparency considered valuable in general by participants?

Finally, we want to understand if participants would prefer inverse transparency over the status quo, which for them is the GDPR. While data owners considered the provided transparency helpful and useful in our specific scenario (see RQ B.2), they may still judge the value of inverse transparency differently if it was introduced to their workplace. Therefore, we asked them for their agreement to four statements:

- Inverse transparency improves upon the protection of the GDPR. (Q18)
- I would prefer inverse transparency over just having the right to consent to or reject data usages outright. (Q19)
- If my company offered me the choice, I would like to have access to data usage tracking. (Q20)
- I would feel safer knowing how my data are accessed in detail. (Q21)

In addition, we allowed participants to elaborate if they wanted to explain their responses (Q22).

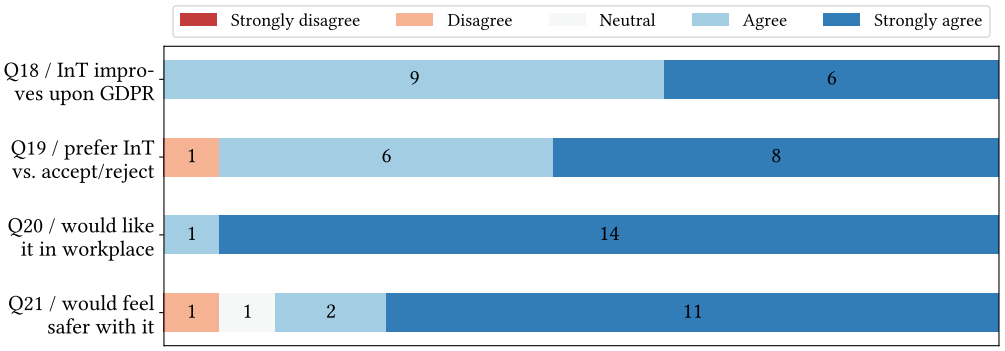

Fig. 11. Results of the questions about inverse transparency generally (abbreviated as "InT" above). All participants (n = 15) were asked to express their agreement on a five-point Likert scale (X-axis), from 1 for "strongly disagree" to 5 for "strongly agree."

Responses are overall very positive, with only one respondent disagreeing for questions Q19 and Q21, respectively. For question Q21, the critical respondent elaborated that they "would not trust that the access data provided to [them] is actually correct" (O9). Possibly, if an integrity guarantee was given, their sense of safety could be improved. Especially striking is the result for question Q20, with all participants agreeing and almost all (14/15) even expressing strong agreement. This

suggests that inverse transparency is considered especially valuable by participants when imagining a workplace setting. This sentiment is best summarized by the following quotes: "I think that if implemented correctly, Inverse Transparency could be a game changer for privacy" (C3), with another respondent adding: "being able to track how my data is used is beneficial to myself, to transparancy [sic!] within the team and ultimately to the relationship between data owners and data consumers" (O6).

5.7 Conclusions

Our study of the user perspective shows that inverse transparency is judged positively and as an improvement over the status quo. All participants agreed that it would be a valuable addition to the workplace. This suggests inverse transparency is capable to empower employees. We furthermore find promising results regarding our goal of creating accountability and deterring misusage of data. Data owners believed that the inverse transparency could influence data consumers' usage behavior, an impression that some data consumers confirmed for their case. Finally, considering the purpose dimension of trust (see Section 3.4), participants judge inverse transparency as beneficial for themselves and their teams. The introduction of inverse transparency is considered valuable for employees, which is reflected in that all participants would like it in the workplace. This shows that the purpose of inverse transparency is perceived as benevolent and supportive of employees.

We conclude that inverse transparency is capable to meet our goals of empowering employees and creating accountability for data usages. It is considered an improvement of the status quo, with the perceived benevolent purpose and felt benefits supporting trust in the concept.

6 DISCUSSION

At first glance, expecting a rethink of people analytics design may seem ambitious at best. Yet, we have seen with the introduction of the GDPR how quickly software firms can adapt and update their tools. We think the key driver for this speed were their customers—the companies buying these tools for their use. To ensure their GDPR compliance, they expect their suppliers to build their software accordingly and integrate the necessary tools for, e.g., anonymization or deletion of user data. Now, companies are faced with potent privacy legislation on the one hand, and privacy concerns of their employees [see 85] on the other hand. Ignoring these concerns can lead to anything from reactance [26] to dissatisfaction with the employer or psychological distress [10]. This means that it is in companies' own interest to address their employees' concerns. Furthermore, where workers' councils are active, they may have a say in if analysis tools are bought, using their power to prevent invasive uses of employee data. Inverse transparency could offer a solution for these scenarios. Not all data analyses are problematic, and some might even be beneficial for data owners. Tools built with inverse transparency by design can unlock this potential by empowering data owners with data sovereignty and holding data consumers accountable in case of data misusage.

The shift to develop with inverse transparency by design may have a more pronounced influence on software design than it first appears. We have deliberated the potential changes that can be envisioned when considering *ambient usage* of data. Developers may need to change fundamental interaction paradigms as they include data owners as stakeholders in their design. On the other hand, as we could see in our preliminary study of the developer perspective, inverse transparency may serve as an enabler for potentially privacy-invasive but useful features. Knowing that data owners will be able to supervise usage of their data may reduce the need to be overly cautious upfront, enabling the ethical development of innovative features.

Considering the users, providing inverse transparency is a valuable step but not sufficient on its own. On the one hand, participants in our preliminary study of the user perspective showed a clear interest in inverse transparency. They felt empowered and considered the provided transparency

helpful and useful. Yet, too much transparency [49] or "wrong" ways to frame this transparency [68] may reduce trust in users. One of our study participants confirmed this, noting that they felt more anxious due to the detailed insight into how their data were used. In addition, if explanations or insights are too technical, this too can render them ineffective or even counterproductive [21, 70]. Making the usage logs comprehensible for individuals is therefore an important part of providing transparency. This can mean following users' mental model to create an intuitive tool or, if necessary, providing explanations and documentation [59]. Moreover, we need to consider the possibility of habituation. While our study spanned three months and therefore provides more than just a snapshot, it cannot predict the very long term effects of having access to relatively repetitive data usage information. Karegar et al. [46] discuss the risk of habituation in the context of privacy notices, as it can reduce user attention. To combat this, they suggest engaging users in different ways [46]. For example, an automated anomaly detection system could be incorporated into the transparency dashboard to highlight unusual usage patterns.

At the same time, the effectiveness of inverse transparency systems in empowering employees depends on the sociotechnical context of the workplace. The insights of people analytics can be important for management [22], which may lead to external pressures for employees to contribute data [94] even if they perceive misusage. Seberger et al. [78] find that, in such cases, technical mechanisms meant to empower users may not be sufficient. Instead, if users perceive no alternative, they accept privacy violations more easily. In fact, seemingly empowering mechanisms may then conversely even lead to resignation due to perceived personal responsibility. [78] This is a critical aspect especially for the workplace context, as the power asymmetry can reduce individual agency further. The fear of being fired or facing other negative consequences precludes empowerment [19]. It is therefore necessary to investigate how to actually empower employees to handle misusage of their data. This should include investigating the trade-offs they face in their choices [75, Table 2]. For example, technical solutions that ensure *plausible deniability* may be essential to remove the pressure to conform [24].

Finally, while our studies show promising results, they are preliminary and therefore potentially limited in their significance. The use of university students as subjects, especially in a controlled environment, can threaten the validity of studies researching the workplace. Students have limited experience with working environments, lacking knowledge of its broader context and influencing factors. Additionally, the inherent complexity of the workplace as well as various confounding factors cannot be fully mirrored in an artificial setting. For our purposes, though, this seeming limitation was instead a sought-after property. We worked closely with our industry partner on a conceptual level to develop realistic use cases for inverse transparency. For our experimental studies, meanwhile, we specifically chose to model a fully controlled workplace-like environment with computer science students instead. Experiments with students can be preferable when testing initial hypotheses [81, p. 739]. Computer science students specifically are judged as sensible stand-ins for professionals [86]. Our proposal to introduce inverse transparency in the workplace means a fundamental change in the work and interactions of employees. Conducting our studies in an existing workplace could have triggered confounding change management issues, most notably individual's resistance to change [12]. This is an important factor limiting the success of change initiatives [51]. Furthermore, external influences such as time pressure from other projects could have influenced our results [60]. Removing these factors was necessary to support internal validity [28] and establish causality [47]. This allows us to cleanly link the study results to our intervention of inverse transparency by design. Additionally, it is infeasible to fully control a real world working environment, including the work tasks and utilized tools, continuously for multiple months. This level of control was essential, though, especially for our study of the user perspective. By completely aligning work tasks and incentives with our study goals, we improve construct

validity compared to a less controllable real world environment [30]. At the same time, we recognize our studies as preliminary, as their artificial nature could reduce external validity. A promising next step may therefore be to explore if confounding factors in a real workplace influence individuals' perceptions. Given those limitations, our results show that inverse transparency by design can be practical from a technical standpoint and is experienced as beneficial by our study participants. We consider these insights promising and a sign that inverse transparency by design has the potential to be an important factor in accepted and responsible people analytics.

ACKNOWLEDGMENTS

This work was supported by the German Federal Ministry of Education and Research (BMBF) under grant no. 5091121. We thank Antonia Maria Lehene for her help in creating and validating our user trust questionnaire. Furthermore, we thank the anonymous reviewers for their helpful comments, which greatly improved the final paper.

REFERENCES

[1] Rafael Accorsi. 2010. BBox: A distributed secure log architecture. In *Proceedings of the 2010 European Public Key Infrastructure Workshop (Lecture Notes in Computer Science, 6711)*. Springer, 109–124.
[2] Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, and Yirong Xu. 2002. Hippocratic databases. In *Proceedings of the 28th International Conference on Very Large Databases*. Elsevier, 143–154.
[3] Mohsen Ahmadvand, Amjad Ibrahim, and Alexander Pretschner. 2017. A distributed secure logging mechanism with post-compromise security. (2017). Working draft.
[4] Gene M. Alarcon, Anthony M. Gibson, Charles Walter, Rose F. Gamble, Tyler J. Ryan, Sarah A. Jessup, Brian E. Boyd, and August Capiola. 2020. Trust perceptions of metadata in open-source software: the role of performance and reputation. *Systems* 8, 3, Article 28 (2020).
[5] Scott W. Ambler. 2008. Agile software development at scale. In *Proceedings of the 2nd IFIP Central and East European Conference on Software Engineering Techniques (Lecture Notes in Computer Science, 5082)*. Springer, 1–12.
[6] Tuomas Aura. 1997. Strategies against replay attacks. In *Proceedings of the 10th Computer Security Foundations Workshop*. IEEE, 59–68.
[7] Eugene Bagdasaryan, Griffin Berlstein, Jason Waterman, Eleanor Birrell, Nate Foster, Fred B. Schneider, and Deborah Estrin. 2019. Ancile: Enhancing privacy for ubiquitous computing with use-based privacy. In *Proceedings of the 18th ACM Workshop on Privacy in the Electronic Society*. ACM, 111–124.
[8] Aaron Bangor, Philip T. Kortum, and James T. Miller. 2008. An empirical evaluation of the system usability scale. *International Journal of Human-Computer Interaction* 24, 6 (2008), 574–594.
[9] Dhruba Kumar Bhattacharyya and Jugal Kumar Kalita. 2019. *DDoS attacks: evolution, detection, prevention, reaction, and tolerance*. Chapman and Hall/CRC.
[10] Devasheesh P. Bhave, Laurel H. Teo, and Reeshad S. Dalal. 2020. Privacy at work: a review and a research agenda for a contested terrain. *Journal of Management* 46, 1 (2020), 127–164.
[11] Eleanor Birrell, Anders Gjerdrum, Robbert van Renesse, Håvard Johansen, Dag Johansen, and Fred B. Schneider. 2018. SGX enforcement of use-based privacy. In *Proceedings of the 17th Workshop on Privacy in the Electronic Society*. ACM, 155–167.
[12] Wayne H. Bovey and Andrew Hede. 2001. Resistance to organisational change: the role of defence mechanisms. *Journal of Managerial Psychology* 7, 16 (2001), 534–548.
[13] Norman E. Bowie and Karim Jamal. 2006. Privacy rights on the internet: self-regulation or government regulation? *Business Ethics Quarterly* 16, 3 (2006), 323–342.
[14] David Brin. 1998. *The Transparent Society*. Basic Books.
[15] John Brooke. 1986. System usability scale (SUS): a quick-and-dirty method of system evaluation user information. In *Usability Evaluation in Industry*, Patrick W. Jordan, Bruce Thomas, Bernard A. Weerdmeester, and Ian L. McClelland (Eds.). Vol. 43. Taylor & Francis, 189–194.
[16] California Consumer Privacy Act. 2018. An act to add Title 1.81.5 (commencing with Section 1798.100) to Part 4 of Division 3 of the Civil Code, relating to privacy (California Consumer Privacy Act of 2018). *Assembly Bill* 375 (2018), 1–24.
[17] Fred H. Cate. 2002. Principles for protecting privacy. *Cato Journal* 22, 1 (2002), 33–58.
[18] Ann Cavoukian. 2009. Privacy by Design: the 7 foundational principles. Information and Privacy Commissioner of Ontario, Canada.

[19] Tomas Chamorro-Premuzic. 2020. Can surveillance AI make the workplace safe? *MIT Sloan Management Review* 62, 1 (2020), 13–15. https://sloanreview.mit.edu/article/can-surveillance-ai-make-the-workplace-safe/
[20] Prithwiraj Choudhury, Kevin Crowston, Linus Dahlander, Marco S. Minervini, and Sumita Raghuram. 2020. GitLab: work where you want, when you want. *Journal of Organization Design* 9, Article 23 (2020).
[21] Henriette Cramer, Vanessa Evers, Maarten van Someren, Bob Wielinga, Sam Besselink, Lloyd Rutledge, Natalia Stash, and Lora Aroyo. 2007. User interaction with user-adaptive information filters. In *Proceedings of the 2007 International Conference on Usability and Internationalization*. Springer, 324–333.
[22] Thomas H. Davenport, Jeanne Harris, and Jeremy Shapiro. 2010. Competing on talent analytics. *Harvard Business Review* 88, 10 (2010), 52–58. Issue October.
[23] Hans Delfs and Helmut Knebl. 2007. Public-key cryptography. In *Introduction to Cryptography*. Springer, Chapter 3, 33–80.
[24] Mina Deng, Kim Wuyts, Riccardo Scandariato, Bart Preneel, and Wouter Joosen. 2011. A privacy threat analysis framework: Supporting the elicitation and fulfillment of privacy requirements. *Requirements Engineering* 16, 1 (2011), 3–32.
[25] Tommaso Fabbri, Anna Chiara Scapolan, Fabiola Bertolotti, Federica Mandreoli, and Riccardo Martoglia. 2022. Work datafication and digital work behavior analysis as a source of HRM insights. In *Do Machines Dream of Electric Workers? (Lecture Notes in Information Systems and Organisation, 49)*. Springer, 53–65.
[26] Wenting Feng, Rungting Tu, Tim Lu, and Zhimin Zhou. 2019. Understanding forced adoption of self-service technology: the impacts of users' psychological reactance. *Behaviour & Information Technology* 38, 8 (2019), 820–832.
[27] Jolene Fisher and Toby Hopp. 2020. Does the framing of transparency impact trust? Differences between self-benefit and other-benefit message frames. *International Journal of Strategic Communication* 14, 3 (2020), 1–20.
[28] Donald W. Fiske and Susan T. Fiske. 2005. Laboratory studies. In *Encyclopedia of Social Measurement*, Kimberly Kempf-Leonard (Ed.). Elsevier, 435–439.
[29] Nicola Flannery. 2017. GDPR series: A design for life? Designing the future of privacy. *Data Protection Ireland* 10, 2 (2017), 6–8.
[30] Guillaume R. Fréchette. 2015. Laboratory experiments: professionals versus students. In *Handbook of Experimental Economic Methodology*, Guillaume R. Fréchette and Andrew Schotter (Eds.). Oxford University Press, Chapter 17, 360–390.
[31] Michal S. Gal and Oshrit Aviv. 2020. The competitive effects of the GDPR. *Journal of Competition Law & Economics* 16, 3 (2020), 349–391.
[32] Domingo García-Marzá. 2005. Trust and dialogue: theoretical approaches to ethics auditing. *Journal of Business Ethics* 57, 3 (2005), 209–219.
[33] Chunpeng Ge, Siwei Sun, and Pawel Szalachowski. 2019. Permissionless blockchains and secure logging. In *Proceedings of the 2019 IEEE International Conference on Blockchain and Cryptocurrency*. IEEE, 56–60.
[34] General Data Protection Regulation. 2016. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the 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 (General Data Protection Regulation). *Official Journal of the European Union* 59 (2016), 1–88.
[35] Maren Gierlich-Joas, Thomas Hess, and Rahild Neuburger. 2020. More self-organization, more control—or even both? Inverse transparency as a digital leadership concept. *Business Research* 13, 3 (2020), 921–947.
[36] Lisa Marie Giermindl, Franz Strich, Oliver Christ, Ulrich Leicht-Deobald, and Abdullah Redzepi. 2022. The dark sides of people analytics: reviewing the perils for organisations and employees. *European Journal of Information Systems* 31, 3 (2022), 410–435.
[37] Seda Gürses and Joris van Hoboken. 2018. Privacy after the agile turn. In *The Cambridge Handbook of Consumer Privacy*, Jules Polonetsky, Omer Tene, and Evan Selinger (Eds.). Cambridge University Press, Chapter 32, 579–601.
[38] Irit Hadar, Tomer Hasson, Oshrat Ayalon, Eran Toch, Michael Birnhack, Sofia Sherman, and Arod Balissa. 2018. Privacy by designers: software developers' privacy mindset. *Empirical Software Engineering* 23, 1 (2018), 259–289.
[39] Angela T. Hall, Dwight D. Frink, and M. Ronald Buckley. 2017. An accountability account: A review and synthesis of the theoretical and empirical research on felt accountability. *Journal of Organizational Behavior* 38, 2 (2017), 204–224.
[40] Kevin Anthony Hoff and Masooda Bashir. 2015. Trust in automation: Integrating empirical evidence on factors that influence trust. *Human Factors* 57, 3 (2015), 407–434.
[41] Jason E. Holt. 2006. Logcrypt: Forward security and public verification for secure audit logs. In *Proceedings of the 2006 Australasian Workshops on Grid Computing and E-Research*. 203–211.
[42] Joschka A. Hüllmann, Simone Krebber, and Patrick Troglauer. 2021. The IT artifact in people analytics: Reviewing tools to understand a nascent field. In *Proceedings of the 16th International Conference on Wirtschaftsinformatik (Lecture Notes in Information Systems and Organisation, 48)*. Springer, 238–254.

[43] Patrik Hummel, Matthias Braun, Steffen Augsberg, and Peter Dabrock. 2018. Sovereignty and data sharing. *ICT Discoveries* 1, 2 (2018).
[44] Matthias Jarke, Boris Otto, and Sudha Ram. 2019. Data sovereignty and data space ecosystems. *Business & Information Systems Engineering* 61, 5 (2019), 549–550.
[45] Jian Jia, Ginger Zhe Jin, and Liad Wagman. 2018. *The short-run effects of GDPR on technology venture investment.* Technical Report 25248. National Bureau of Economic Research.
[46] Farzaneh Karegar, John Sören Pettersson, and Simone Fischer-Hübner. 2020. The dilemma of user engagement in privacy notices: Effects of interaction modes and habituation on user attention. *ACM Transactions on Privacy and Security* 23, 1 (2020), 1–38.
[47] Elena Katok. 2018. Designing and conducting laboratory experiments. In *The Handbook of Behavioral Operations*, Karen Donohue, Elena Katok, and Stephen Leider (Eds.). John Wiley & Sons, Chapter 1, 3–33.
[48] Florian Kelbert and Alexander Pretschner. 2018. Data usage control for distributed systems. *ACM Transactions on Privacy and Security* 21, 3, Article 12 (2018), 32 pages.
[49] René F. Kizilcec. 2016. How much information? Effects of transparency on trust in an algorithmic interface. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 2390–2395.
[50] Tim Kraska, Michael Stonebraker, Michael Brodie, Sacha Servan-Schreiber, and Daniel Weitzner. 2019. SchengenDB: a data protection database proposal. In *Heterogeneous Data Management, Polystores, and Analytics for Healthcare*. Springer, 24–38.
[51] Chloe N. Kuhlman. 2021. *Will work for change: A correlational study on employee resistance to change and transformational leadership behaviors*. Ph. D. Dissertation. Azusa Pacific University.
[52] Seungho Lee, Wonsuk Choi, Hyo Jin Jo, and Dong Hoon Lee. 2020. Poster: Secure logging infrastructure employing heterogeneous trusted execution environments. In *Proceedings of the 2020 Network and Distributed System Security Symposium*.
[53] Michael Lörscher. 2012. *Data usage control for the Thunderbird mail client.* Master's thesis. University of Kaiserslautern, Germany.
[54] Caitlin Lustig, Katie Pine, Bonnie Nardi, Lilly Irani, Min Kyung Lee, Dawn Nafus, and Christian Sandvig. 2016. Algorithmic authority: The ethics, politics, and economics of algorithms that interpret, decide, and manage. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 1057–1062.
[55] Di Ma and Gene Tsudik. 2009. A new approach to secure logging. *ACM Transactions on Storage* 5, 1 (2009), 1–21.
[56] Akhil Mathur, Marc Van den Broeck, Geert Vanderhulst, Afra Mashhadi, and Fahim Kawsar. 2015. Quantified workplace: opportunities and challenges. In *Proceedings of the 2nd on Workshop on Physical Analytics*. ACM, 37–41.
[57] Aleecia M. McDonald and Lorrie Faith Cranor. 2008. The cost of reading privacy policies. *I/S: A Journal of Law and Policy for the Information Society* 4, 3 (2008), 543–568.
[58] Craig Mundie. 2014. Privacy pragmatism; Focus on data use, not data collection. *Foreign Affairs* 93, 2 (2014), 28–38.
[59] Patrick Murmann and Simone Fischer-Hübner. 2017. Tools for achieving usable ex post transparency: a survey. *IEEE Access* 5 (2017), 22965–22991.
[60] Ning Nan and Donald E. Harter. 2009. Impact of budget and schedule pressure on software development cycle time and effort. *IEEE Transactions on Software Engineering* 35, 5 (2009), 624–637.
[61] Dean Povey. 1999. Optimistic security: a new access control paradigm. In *Proceedings of the 1999 Workshop on New Security Paradigms*. ACM, 40–45.
[62] Alexander Pretschner. 2009. An overview of distributed usage control. In *Proceedings of the 2nd International Conference on Knowledge Engineering, Pinciples and Techniques*. 25–33.
[63] Alexander Pretschner. 2014. Achieving accountability with distributed data usage control technology. In *Proceedings of the 2nd International Workshop on Accountability: Science, Technology, and Policy*. MIT.
[64] Alexander Pretschner, Manuel Hilty, and David Basin. 2006. Distributed usage control. *Commun. ACM* (2006), 39–44.
[65] Alexander Pretschner, Manuel Hilty, Florian Schütz, Christian Schaefer, and Thomas Walter. 2008. Usage control enforcement: present and future. *IEEE Security & Privacy* 6, 4 (2008), 44–53.
[66] Alexander Pretschner, Florian Kelbert, Enrico Kumari, Prachi, and Tobias Wüchner. 2013. A distributed data usage control infrastructure. (2013). Unpublished.
[67] Christian Priebe, Kapil Vaswani, and Manuel Costa. 2018. EnclaveDB: A secure database using SGX. In *Proceedings of the 2018 IEEE Symposium on Security and Privacy*. IEEE, 264–278.
[68] Emilee Rader, Kelley Cotter, and Janghee Cho. 2018. Explanations as mechanisms for supporting algorithmic transparency. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 1–13.
[69] Ashwini Rao, Florian Schaub, Norman Sadeh, Alessandro Acquisti, and Ruogu Kang. 2016. Expecting the unexpected: Understanding mismatched privacy expectations online. In *Proceedings of the 12th Symposium on Usable Privacy and Security*. USENIX, 77–96.

[70] Brad R. Rawlins. 1994. Measuring the relationship between organizational transparency and employee trust. In *CHI'94 Conference Companion on Human Factors in Computing Systems*. ACM, 99–100.

[71] Jens Riegelsberger, M. Angela Sasse, and John D. McCarthy. 2005. The mechanics of trust: A framework for research and design. *International Journal of Human-Computer Studies* 62, 3 (2005), 381–422.

[72] Aaron Rieke, Miranda Bogen, and David G. Robinson. 2018. Public scrutiny of automated decisions: early lessons and emerging methods. An Upturn and Omidyar Network Report. https://apo.org.au/node/210086

[73] Ira S. Rubinstein. 2010. Privacy and regulatory innovation: moving beyond voluntary codes. *I/S: A Journal of Law and Policy for the Information Society* 6, 3 (2010), 355–423.

[74] Iflaah Salman, Ayse Tosun Misirli, and Natalia Juristo. 2015. Are students representatives of professionals in software engineering experiments?. In *Proceedings of the 37th IEEE/ACM International Conference on Software Engineering*, Vol. 1. IEEE, 666–676.

[75] Shruti Sannon, Billie Sun, and Dan Cosley. 2022. Privacy, surveillance, and power in the gig economy. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. ACM, Article 619, 15 pages.

[76] SAP. 2020. SAP SuccessFactors workforce analytics: Optimize performance and results with data-driven insights. https://www.sap.com/products/hcm/workforce-planning-hr-analytics.html?pdf-asset=12e85371-c37c-0010-82c7-eda71af511fa

[77] Christian Schaefer and Christine Edman. 2019. Transparent logging with Hyperledger Fabric. In *Proceedings of the 2019 IEEE International Conference on Blockchain and Cryptocurrency*. IEEE, 65–69.

[78] John S. Seberger, Marissel Llavore, Nicholas Nye Wyant, Irina Shklovski, and Sameer Patil. 2021. Empowering resignation: there's an app for that. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, Article 552, 18 pages.

[79] Awanthika Senarath, Marthie Grobler, and Nalin Asanka Gamagedara Arachchilage. 2019. Will they use it or not? Investigating software developers' intention to follow privacy engineering methodologies. *ACM Transactions on Privacy and Security* 22, 4 (2019), 1–30.

[80] Tushar Sharma, Girish Suryanarayana, and Ganesh Samarthyam. 2015. Challenges to and solutions for refactoring adoption: an industrial perspective. *IEEE Software* 32, 6 (2015), 44–51.

[81] Dag I. K. Sjøberg, Jo Erskine Hannay, Ove Hansen, Vigdis By Kampenes, Amela Karahasanovic, Nils-Kristian Liborg, and Anette C. Rekdal. 2005. A survey of controlled experiments in software engineering. *IEEE Transactions on Software Engineering* 31, 9 (2005), 733–753.

[82] Matthias Söllner, Axel Hoffmann, Holger Hoffmann, Arno Wacker, and Jan Marco Leimeister. 2012. Understanding the formation of trust in IT artifacts. In *Proceedings of the 33rd International Conference on Information Systems*. AIS.

[83] Smitha Sundareswaran, Anna Squicciarini, and Dan Lin. 2012. Ensuring distributed accountability for data sharing in the cloud. *IEEE Transactions on Dependable and Secure Computing* 9, 4 (2012), 556–568.

[84] Jenny Tang, Hannah Shoemaker, Ada Lerner, and Eleanor Birrell. 2021. Defining privacy: How users interpret technical terms in privacy policies. *Proceedings on Privacy Enhancing Technologies* 2021, 3 (2021), 1–25.

[85] Mena Angela Teebken and Thomas Hess. 2021. Privacy in a digitized workplace: Towards an understanding of employee privacy concerns. In *Proceedings of the 54th Hawaii International Conference on System Sciences*. University of Hawaii at Manoa, 6661–6670.

[86] Walter F. Tichy. 2000. Hints for reviewing empirical work in software engineering. *Empirical Software Engineering* 5, 4 (2000), 309–312.

[87] Aizhan Tursunbayeva, Claudia Pagliari, Stefano Di Lauro, and Gilda Antonelli. 2021. The ethics of people analytics: risks, opportunities and recommendations. *Personnel Review* 51, 3 (2021), 900–921.

[88] Sjir Uitdewilligen, Mary J. Waller, and Adrian H. Pitariu. 2013. Mental model updating and team adaptation. *Small Group Research* 44, 2 (2013), 127–158.

[89] Martijn H. Van Beek. 2007. *Comparison of enterprise digital rights management systems*. Master's thesis. Radboud University Nijmegen.

[90] Paul Georg Wagner, Pascal Birnstill, and Jürgen Beyerer. 2018. Distributed usage control enforcement through trusted platform modules and SGX enclaves. In *Proceedings of the 23rd ACM on Symposium on Access Control Models and Technologies*. ACM, 85–91.

[91] Daniel J. Weitzner, Harold Abelson, Tim Berners-Lee, Joan Feigenbaum, James Hendler, and Gerald Jay Sussman. 2008. Information accountability. *Commun. ACM* 51, 6 (2008), 82–87.

[92] Tal Z. Zarsky. 2017. Incompatible: The GDPR in the age of big data. *Seton Hall Law Review* 47, 4 (2017), 995–1020.

[93] Valentin Zieglmeier. 2023. The inverse transparency toolchain: a fully integrated and quickly deployable data usage logging infrastructure. *Software Impacts* (2023). Forthcoming.

[94] Valentin Zieglmeier, Maren Gierlich-Joas, and Alexander Pretschner. 2022. Increasing employees' willingness to share: Introducing appeal strategies for people analytics. In *Proceedings of the 13th International Conference on Software Business (Lecture Notes in Business Information Processing, 463)*. Springer, 213–226. https://doi.org/10.1007/978-3-031-20706-8_15

[95] Valentin Zieglmeier and Antonia Maria Lehene. 2021. Designing trustworthy user interfaces. In *Proceedings of the 33rd Australian Conference on Human-Computer Interaction*. ACM, 182–189. https://doi.org/10.1145/3520495.3520525

[96] Valentin Zieglmeier and Gabriel Loyola Daiqui. 2021. GDPR-compliant use of blockchain for secure usage logs. In *Proceedings of the 25th International Conference on Evaluation and Assessment in Software Engineering*. ACM, 313–320. https://doi.org/10.1145/3463274.3463349

[97] Valentin Zieglmeier, Gabriel Loyola Daiqui, and Alexander Pretschner. 2023. Decentralized inverse transparency with blockchain. *Distributed Ledger Technologies: Research and Practice* (2023), 1–30. https://doi.org/10.1145/3592624

[98] Niina Zuber, Severin Kacianka, Alexander Pretschner, and Julian Nida-Rümelin. 2020. Ethical deliberation for agile software processes: the EDAP manual. In *Hengstschläger, Markus (Hrsg): Digitaler Wandel und Ethik. Österreichischer Rat für Forschung und Technologieentwicklung*. ecowin, 150–175.

A ADDENDUM FOR STUDY B

In the following, find a screenshot of the transparency dashboard (A.1) and a list of the questionnaire questions (A.2).

A.1 The transparency dashboard

Fig. 12. Screenshot of the transparency dashboard used in study B. The top half summarizes the data usages recorded in the last seven days. The bottom presents a detailed list of the logged data usages for data owners to inspect. The email addresses and usage data shown are artificial and for illustration purposes only.

A.2 Questionnaire questions

In the following, find the questions answered by participants in study B.

A.2.1 RQ 1 – Unified dashboard.

- Q1–Q10 (Likert scale): System usability scale (questions in [15])
- Follow-up free text questions
  - Q11: Did you enjoy using Clotilde[7]? Why?
  - Q12: What could be changed to improve your experience? What did you miss?

A.2.2 RQ 2 – Inverse transparency experienced as beneficial.

- Likert questions
  - Q13: I found the additional transparency helpful.
  - Q14: Having more transparency over data usages was useful.
- Q15 (free text): How did you experience the provided transparency? What would have changed your experience?

All questions of this RQ were for data owners.

A.2.3 RQ 3 – Inverse transparency can influence data consumers.

- Data owners – Q16 (Likert scale): I think that the usage tracking could deter data users from misusing my data.
- Data consumers – Q17 (free text): Would you have acted differently in case your accesses weren't monitored?

A.2.4 RQ 4 – Inverse transparency considered valuable.

- Likert questions
  - Q18: Inverse Transparency improves upon the protection of the GDPR (DS-GVO).
  - Q19: I would prefer Inverse Transparency over just having the right to consent to or reject data usages outright.
  - Q20: If my company offered me the choice, I would like to have access to data usage tracking.
  - Q21: I would feel safer knowing how my data are accessed in detail.
- Q22 (free text): Optional: Any further comments?

Received July 2022; revised January 2023; accepted March 2023

[7]The name of the transparency dashboard.