Software Impacts 17 (2023) 100554

Contents lists available at ScienceDirect

Software Impacts

journal homepage: www.journals.elsevier.com/software-impacts

Original Software Publication

The Inverse Transparency Toolchain: A Fully Integrated and Quickly Deployable Data Usage Logging Infrastructure

Valentin Zieglmeier

Technical University of Munich, Munich, Germany

ARTICLE INFO

Keywords:
Inverse transparency
Data sovereignty
Accountability
Usage logging

ABSTRACT

Inverse transparency is created by making all usages of employee data visible to them. This requires tools that handle the logging and storage of usage information, and making logged data visible to data owners. For research and teaching contexts that integrate inverse transparency, creating this required infrastructure can be challenging. The Inverse Transparency Toolchain presents a flexible solution for such scenarios. It can be easily deployed and is tightly integrated. With it, we successfully handled use cases covering empirical studies with users, prototyping in university courses, and experimentation with our industry partner.

Current code version	v1.0.0
Permanent link to code/repository used for this code version	https://github.com/SoftwareImpacts/SIMPAC-2023-126
Permanent link to Reproducible Capsule	
Legal Code License	MIT License
Code versioning system used	Git
Software code languages, tools, and services used	Python, Go, Svelte, JavaScript, Shell; HashiCorp Vault, HashiCorp Consul, SQLite, Docker
Compilation requirements, operating environments & dependencies	Python ≥ 3.8, Go ≥ 1.14, Ubuntu ≥ 20.04; dependencies: https://github.com/SoftwareImpacts/SIMPAC-2023-126/blob/master/DEPENDENCIES.md
If available Link to developer documentation/manual	https://github.com/SoftwareImpacts/SIMPAC-2023-126/blob/master/README.md
Support email for questions	valentin.zieglmeier@tum.de

1. Introduction

Employees become increasingly transparent in the digital workplace. Every aspect of work, including the collaboration with colleagues, management of vacation or sick days, but importantly also the status and results of the work itself can be tracked in the tools that are used for these tasks. This transparency can be necessary to run the business or it may simply facilitate work and collaboration, but it is unbalanced. Managers have a much clearer picture of the conducted work, but employees have no tools to oversee how their data are used or uncover potential misusage of their data. Therefore, the idea of creating *inverse transparency* [1] in the workplace was proposed [2,3]. In short, this entails making all usages of employees' data transparent to them. Thereby, misusage of data could be addressed by enabling accountability [4]. To realize inverse transparency, a backbone of auxiliary tools is required. These are necessary to handle the technical steps that enable inverse transparency: logging and storing usage information from various tools and making those logged data visible to data owners. To create a fully integrated setup, a single-sign on system for user authentication and attribution is additionally needed [3]. Especially for research projects and university courses that need to quickly set up this required infrastructure, the overhead that this setup entails can endanger the success of projects. But, also when introducing inverse transparency in practice, having a basic setup for experimentation that allows easy extension and integration is vital. This enables small-scale trials and studies without needing to invest in a fully fledged system up front.

With the Inverse Transparency Toolchain, we present a solution for all these cases. As a fully integrated system, it can be easily deployed in minutes, requiring only basic configuration. This enables research projects, theses, or university courses to be set up quickly. Furthermore, its modular architecture of individual tools means that any component can be replaced or adapted individually, enabling, e.g., experimentation with different variations or a step-by-step rollout of larger setups. In our own work, we have already successfully applied the toolchain in

E-mail address: valentin.zieglmeier@tum.de.

https://doi.org/10.1016/j.simpa.2023.100554
Received 17 March 2023; Received in revised form 9 May 2023; Accepted 23 July 2023

2665-9638/© 2023 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

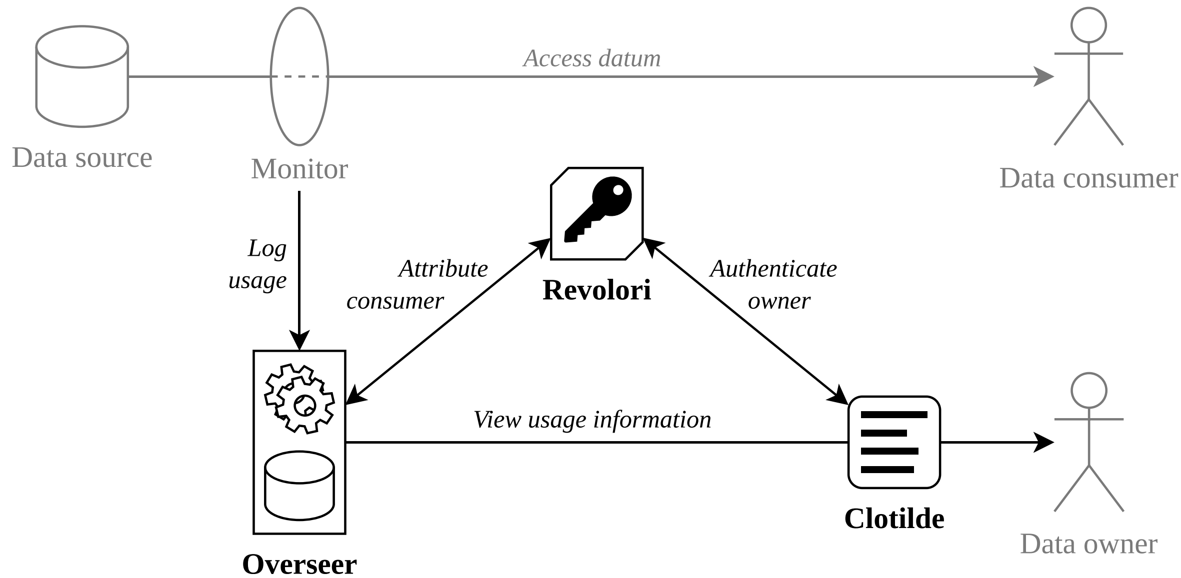

Fig. 1. Overview of how the toolchain components interact. Arrows denote interactions and data flow. Grayed out components are not part of the toolchain, instead representing its operating context.

such scenarios. In the context of research projects and user studies, it allowed us to depend on a robust backbone that ensured a fully inverse transparent environment. For students working on theses or in courses, the toolchain offered a standardized interface that allowed them to integrate their projects with ease, sparing them the effort of creating and setting up their own tools from the ground up. This meant that they could get started on their projects almost immediately. Finally, in our work with our industry partner, employees worked with the toolchain to prototype and experiment with their own solutions. Our choice of creating a toolchain of web apps meant that they did not need to set up a local infrastructure and could focus on developing valuable use cases for their context.

2. Tools and functionalities

Technically realizing inverse transparency, at its core, requires three steps: monitoring and attributing data usages, storing the created logs, and making those data available to view [3]. When considering these tasks, we find that inverse transparency solutions for different contexts vary fundamentally only in one aspect, namely in *how* and *what* is monitored. That is a domain- and context-specific task that cannot easily be abstracted away. To integrate monitoring into data usage processes, we envision building the analytics with *inverse transparency by design* [4]. To support this vision, our toolchain provides tools for exactly those tasks that can be abstracted away: *Overseer* logs usages and stores them, *Clotilde* displays stored usage data, and *Revolori* authenticates and attributes users (see Fig. 1). In the following, we give a brief overview of each tool individually.

2.1. Overseer: logs and stores data usages

As the interface of inverse transparency monitors to the data store, *Overseer* can be thought of as the backend of the toolchain. It enables read and write access to the usage log data via a REST API. It is implemented in Python utilizing FastAPI.[1] That means that an OpenAPI[2] documentation is automatically created and hosted when running it. Data are stored in a local SQLite[3] database file, making the system very portable and easy to set up, using the SQLalchemy[4] ORM. If the database schemas or models are modified, *Overseer* automatically migrates its database with Alembic.[5] Finally, should that be necessary, the implemented *data access object* pattern facilitates replacing the storage backend.

2.2. Clotilde: makes logged usage information accessible

Arguably the most important component is *Clotilde*, which can be thought of as the frontend of the toolchain. It is a modern single-page web application implemented in JavaScript with Svelte,[6] using the Rollup[7] module bundler for deployment. The user interface styling is based on Bootstrap.[8] Through *Clotilde*, users can access any usage logs relating to their data. A summary is provided to give an overview over data usages in the past seven days. Furthermore, a full table of all logged data usages is available. The viewed data can be exported in form of a PDF report. To allow users to limit who uses their data, a page for simple usage policies exists. It is relatively elementary, but can be easily extended to explore various options for a usage policy screen. If this functionality is not required, the page can easily be hidden from the navigation bar.

2.3. Revolori: authenticates and attributes users

Both *Overseer* and *Clotilde* depend on *Revolori*, a single sign-on server that offers both user authentication but also attribution services. It is a lightweight implementation in Go, depending on HashiCorp Vault[9] for secret management and encryption, which in turn utilizes the HashiCorp Consul Secrets Engine.[10] The secrets engine can easily be replaced, but even Vault can be supplanted by a different system, if necessary. *Revolori* offers a REST API. We integrate the Swaggo swag library[11] to automatically create and host an OpenAPI documentation when running it. As its main task, the *Revolori* API offers endpoints for

1 https://fastapi.tiangolo.com
2 https://www.openapis.org
3 https://sqlite.org
4 https://www.sqlalchemy.org
5 https://alembic.sqlalchemy.org
6 https://svelte.dev
7 https://rollupjs.org
8 https://getbootstrap.com
9 https://www.vaultproject.io
10 https://developer.hashicorp.com/vault/docs/secrets/consul
11 https://github.com/swaggo/swag

user session management, enabling login and logout functionality. To achieve that, JSON web tokens[12] are issued and tracked in the system. This functionality is used by *Clotilde* to log in users, but can also be integrated into inverse transparency monitors. Its secondary task, used by *Overseer*, is user attribution. To motivate this briefly, individuals use various identifiers depending on the tool they interact with: often their primary email, but they may have multiple emails or services may expect a username. Single sign-on integration can help in most cases, but some tools, such as Git, do not support it. To identify users and attribute data usages correctly, Revolori can store all related identifiers of users and provides an API that matches any given identifier to their main identifier.

3. Usage examples

The flexibility of the toolchain allows for various usage scenarios. In the following, we give a brief overview of some example applications.

3.1. Base case: Using the integrated toolchain

As the primary use case, the toolchain can be used unchanged, with various usage monitors feeding data into Overseer (see Fig. 1), enabling *inverse transparency*. For example, in a study, we let developers implement people analytics with inverse transparency by design [see 4]. To realize inverse transparency, the developed people analytics only needed to call the Overseer API to log usage information. Overseer stored the usage logs in a central database for later access, without the developers needing to implement this functionality for each tool. Then, data owners could use Clotilde to view the logged information, gaining inverse transparency over how their data were used [see 4, Sec. 5.2].

3.2. Integrating a custom dashboard with overseer and revolori

One of the important advantages of the toolchain is its flexibility, as it offers the ability to replace individual components. A recent research field that can benefit from this adaptation is the area of *privacy dashboards* [e.g., 5–8]. These are dashboards that display personal data flows, among other information. Clotilde can be considered a basic instantiation of a privacy dashboard. Researchers working on their own privacy dashboard can simply replace Clotilde with their own dashboard, and benefit from the robust infrastructure of the toolchain for data storage and user authentication. This removes the need to implement their custom solution [see, e.g., 6, Sec. 5] and accelerates the research.

3.3. Adapting clotilde to test user interface modifications

Some research does not require a full privacy dashboard, but is limited to individual features and qualities. For example, researchers consider usability, user experience, or trustworthiness of transparency tools and dashboards [see 9–11]. To that end, the extensibility of Clotilde can be utilized to test user interface modifications. We specifically designed the tool to be modular, meaning individual pages and components can be replaced. We made use of this in our own research, designing two variants of the dashboard: one to elicit trust, one to reduce it. This allowed us to quickly evaluate the effectiveness of our measures in influencing user trust [see 11, Sec. 6].

4. Impact overview

The relatively large overhead of providing the auxiliary tools required for an inverse transparent environment limits the resources that remain to create truly novel solutions for inverse transparency. This is the core issue that is addressed by the Inverse Transparency Toolchain. Thereby, it supports researchers, lecturers, and practitioners, enabling use cases that would otherwise be infeasible to address.

For researchers, the modular design of the toolchain allows focusing on only those details that are relevant for their research question. As one example, many open research questions still exist with regard to how an effective inverse transparency dashboard should be designed. One needs to balance the ideal of transparency, which would suggest making available all information, and the understandability and usability of the information, which may mean simplifying, aggregating, or interpreting for users. Being able to use the existing toolchain and only modifying the user interface means that realistic exploration and user studies are feasible with minimal adaptations to the code. In our own research, we employed the Inverse Transparency Toolchain in varying ways, which lead to multiple published and in-review papers: [3,4,11–13]

For lecturers, the simple and quick deployment of the toolchain enables them to teach courses that integrate inverse transparency. On the one hand, this can mean letting students experience inverse transparency and empowering them to consider what types of data usages they find acceptable. On the other hand, especially in computer science, students can be guided to build their own inverse transparency solutions, e.g. focusing on developing domain-specific monitors or creating analysis tools with inverse transparency by design. We conducted two university practicums exactly for this purpose. Students built various analysis tools that integrated inverse transparency, showing the different ways that it can impact the software design and architecture.

Finally, practitioners can benefit when bootstrapping inverse transparency pilots or for experimentation. Due to inverse transparency focusing on improving employees' knowledge about how their data are used, such projects may be driven by grassroots efforts from within the company. The out-of-the-box deployment and tight integration of the Inverse Transparency Toolchain allows even single individuals or small teams to set up a proof of concept infrastructure. If resources are available, the web-based approach eases experimentation, as individuals are not required to install tools on their machine to test out inverse transparency tools they develop. In our work with our industry partner, the toolchain enabled employees' experimentation and development of own use cases [see 14, pp. 70–73]. This work was conducted as part of the research project "Inverse Transparenz",[13] which ran from 2018 to 2022. The Inverse Transparency Toolchain is one of our results.

5. Limitations and potential improvements

The toolchain was developed primarily for research and teaching purposes. This focus lead to some limitations, which we identify in the following.

While all *Overseer* API endpoints for reading usage data or manipulating access policies use *Revolori* authentication, we chose to only require basic authentication for the API endpoints used to add new usage logs. This was done to simplify development of new inverse transparency monitors, as implementing a complex authentication flow does not improve the core functionality of such tools but costs time and effort. Yet, this means that *Overseer* trusts the monitors to ensure the correctness of log entries. Furthermore, the data stored in its SQLite database is not encrypted, which allows server admins to read, modify, and delete all entries. This was done to enable the manipulation of log entries for specific use cases, e.g. in studies, but means that the integrity of log entries cannot be guaranteed. Therefore, one potential step for improvement would be to focus on providing integrity guarantees for the data stored by *Overseer*.

Furthermore, we made sure that *Revolori* stores user data securely, utilizing the established HashiCorp Vault system. Yet, our focus was

[12] https://jwt.io

[13] https://www.inversetransparenz.de/

on security, meaning the usability is not optimized for. Specifically, user registration has to be done manually via the API, which may be a hurdle for non-technical users. The OpenAPI website *Revolori* provides does make this simpler and provides some form of user interface, but it is not ideal. Furthermore, to simplify user management, the admin user can currently access and manipulate all user data, except for passwords. To improve this, users could be provided a self-service interface to create accounts, manage their data, or delete it. This would make the admin privileges obsolete and thereby improve information security.

6. Conclusion

Inverse transparency presents a new way to think about data privacy in the workplace, proposing to allow employees to monitor how their data are used. This presents interesting use cases for researchers, lecturers, and practitioners. Yet, the large setup overhead required before new challenges can be addressed has the potential to hinder innovative ideas. Therefore, we provide the Inverse Transparency Toolchain, a flexible solution for setting up an inverse transparent environment in minutes. It offers tools for the basic tasks of inverse transparency, with a modular setup that allows replacing or complementing the provided tools depending on the use case. With it, we hope to support novel research on inverse transparency, innovative courses, but also initiatives that aim to bring inverse transparency into practice.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Supporting developers (listed alphabetically): Yiyu Gu, Johannes Heilmann, Raphael Hohmann, Stefan Knilling, Stefan Madzharov, Patipon Riebpradit, Felix Schorer. Translations contributed by Gabriel Loyola Daiqui.

This work was supported by the German Federal Ministry of Education and Research (BMBF) under grant no. 5091121.

References

[1] David Brin, The Transparent Society, Basic Books, 1998.
[2] Maren Gierlich-Joas, Thomas Hess, Rahild Neuburger, More self-organization, more control—or even both? Inverse transparency as a digital leadership concept, Bus. Res. 13 (3) (2020) 921–947.
[3] Valentin Zieglmeier, Alexander Pretschner, Trustworthy transparency by design, 2021, arXiv:2103.10769.
[4] Valentin Zieglmeier, Alexander Pretschner, Rethinking people analytics with inverse transparency by design, Proc. ACM Hum.-Comput. Interact. 7 (CSCW2) (2023) 292, http://dx.doi.org/10.1145/3610083, In press.
[5] Julio Angulo, Simone Fischer-Hübner, Tobias Pulls, Erik Wästlund, Usable transparency with the Data Track: a tool for visualizing data disclosures, in: Proceedings of the 33rd ACM Conference Extended Abstracts on Human Factors in Computing Systems, ACM, 2015, pp. 1803–1808.
[6] Christoph Bier, Kay Kühne, Jürgen Beyerer, PrivacyInsight: the next generation privacy dashboard, in: Proceedings of the 4th Annual Privacy Forum, Springer, 2016, pp. 135–152.
[7] Eelco Herder, Olaf van Maaren, Privacy dashboards: The impact of the type of personal data and user control on trust and perceived risk, in: Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization, ACM, 2020, pp. 169–174.
[8] Florian M. Farke, David G. Balash, Maximilian Golla, Markus Dürmuth, Adam J. Aviv, Are privacy dashboards good for end users? Evaluating user perceptions and reactions to Google's My Activity, in: Proceedings of the 30th USENIX Security Symposium, USENIX, 2021, pp. 483–500.
[9] Julia Earp, Jessica Staddon, 'I had no idea this was a thing': On the importance of understanding the user experience of personalized transparency tools, in: Proceedings of the 6th Workshop on Socio-Technical Aspects in Security and Trust, ACM, 2016, pp. 79–86.
[10] Philip Raschke, Axel Küpper, Olha Drozd, Sabrina Kirrane, Designing a GDPR-compliant and usable privacy dashboard, in: Marit Hansen, Eleni KostaIgor, Igor Nai-Fovino, Simone Fischer-Hübner (Eds.), Privacy and Identity Management. the Smart Revolution, Vol. 526, Springer, 2018, pp. 221–236.
[11] Valentin Zieglmeier, Antonia Maria Lehene, Designing trustworthy user interfaces, in: Proceedings of the 33rd Australian Conference on Human-Computer Interaction, ACM, 2021, pp. 182–189, http://dx.doi.org/10.1145/3520495.3520525.
[12] Maren Gierlich-Joas, Valentin Zieglmeier, Rahild Neuburger, Thomas Hess, Leading agents or stewards? Exploring design principles for empowerment through workplace technologies, in: Proceedings of the 42nd International Conference on Information Systems, AIS, 2021.
[13] Valentin Zieglmeier, Gabriel Loyola Daiqui, Alexander Pretschner, Decentralized inverse transparency with blockchain, Distrib. Ledger Technol. (2023) http://dx.doi.org/10.1145/3592624, Pre-published.
[14] Andreas Boes, Thomas Hess, Alexander Pretschner, Tobias Kämpf, Elisabeth Vogl, Daten – Innovation – Privatheit – Mit Inverser Transparenz das Gestaltungsdilemma der digitalen Arbeitswelt lösen, 2022, URL https://www.inversetransparenz.de/neuerscheinung-forschungsreport-daten-innovation-privatheit.