Poster: Bridging Trust Gaps: Data Usage Transparency in Federated Data Ecosystems

Johannes Lohmöller
RWTH Aachen University
Aachen, Germany
lohmoeller@comsys.rwth-aachen.de

Eduard Vlad
RWTH Aachen University
Aachen, Germany
vlad@comsys.rwth-aachen.de

Markus Dahlmanns
RWTH Aachen University
Aachen, Germany
dahlmanns@comsys.rwth-aachen.de

Klaus Wehrle
RWTH Aachen University
Aachen, Germany
wehrle@comsys.rwth-aachen.de

ABSTRACT

The evolving landscape of data ecosystems (DEs) increasingly demands integrated and collaborative data-sharing mechanisms that simultaneously ensure data sovereignty. However, recently proposed federated platforms, e.g., Gaia-X, only offer a promising solution to share data among already trusted participants—they still lack features to establish and maintain trust. To address this issue, we propose transparency logs for data usage that retrospectively build trust among participants. Inspired by certificate transparency logs that successfully bridge trust gaps in PKIs, we equip data owners with credible evidence of data usage. We show that our transparency logs for data usage are well scalable to sizable DEs. Thus, they are a promising approach to bridge trust gaps in federated DEs with cryptographic guarantees, fostering more robust data sharing.

CCS CONCEPTS

• **Security and privacy → Information accountability and usage control**; Database activity monitoring; • **Information systems** → *Federated databases.*

KEYWORDS

data usage control, data ecosystems, transparency logs

ACM Reference Format:
Johannes Lohmöller, Eduard Vlad, Markus Dahlmanns, and Klaus Wehrle. 2023. Poster: Bridging Trust Gaps: Data Usage Transparency in Federated Data Ecosystems. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS '23), November 26–30, 2023, Copenhagen, Denmark.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3576915.3624371

1 INTRODUCTION

In today's data-centric era, the challenge is not just about amassing data, but effectively sharing and utilizing it across barriers. Still, data often remains trapped in silos, safeguarded behind company-specific firewalls and intricate protection plans. Emerging federated data ecosystems (DEs), such as Gaia-X, have identified this gap and facilitate data exchange [2]. Looking forward to their benefits, DEs are already picked up by a variety of public and industrial sectors, e.g., the automotive industry [6]. However, to preserve their acceptance, these systems must uphold data sovereignty: While the federated design reduces needed confidence in cloud providers, it requires data owners to trust data users both in upholding their infrastructural safeguards and in abiding by stipulated usage terms—even if it does not align with their self-interest. Consequently, today's DEs require data owners and users to establish trust via organizational means, which creates a questionable basis for fully maintaining data sovereignty. To gain wider adoption, federated DEs need to bridge this gap, i.e., they require mechanisms to build trust among broad, potentially skeptical audiences [5].

With certificate transparency (CT), the web ecosystem has managed to bridge such a trust gap in the recent past: By collectively monitoring certificate authorities (CAs), they can no longer misuse their powerful position without notice [4]. In this paper, we adapt the concept of CT to federated data ecosystems enabling data usage transparency. More specifically, our approach allows to continuously update data owners about their data's activity thus forcing data users to abide by agreed usage terms. Hence, data owners no longer have to trust in data users abiding by stipulated usage terms before exchanging data.

Contributions. Our contributions are as follows:

- We share the idea of applying transparency logs to increase data usage transparency for federated DEs, thereby improving data sovereignty.
- In our evaluation, we provide preliminary evidence of their scalability to envisioned DE sizes.

Although transparent data usage is not entirely a new idea [6], our method distinctively provides data users with tangible proof of adherence to stipulations, thereby lowering the amount of trust, participants need to have in federated DEs.

2 OVERVIEW OF DATA ECOSYSTEMS AND CT

To establish data usage transparency, it is first necessary to recapitulate the concept of federated DEs. Federated DEs, such as Gaia-X, facilitate the collaboration on and exchange of distributed data across organizational boundaries or national borders [2]. Thus,

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

CCS '23, November 26–30, 2023, Copenhagen, Denmark
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0050-7/23/11.
https://doi.org/10.1145/3576915.3624371

they provide common interfaces, architecture, and governance for participants. Participants implement these interfaces on infrastructure under their control, thereby maintaining *data sovereignty*: They never need to share data with an untrusted cloud provider, etc.

A typical data exchange is bilateral, between a *data owner* providing and a (remote) *data user* consuming a specific dataset or datum. Therefore, *data space connectors* implement the necessary interfaces on both ends. They proxy and translate requests between DE participants and (proprietary) local data stores while performing access and usage control for specific data sets. Here, *usage control policies* allow owners to specify permissions, obligations, and prohibitions on what user can and cannot do with their data [7].

Trust between participants, which is necessary to enforce usage control, nowadays is mostly based on organizational means [5]. In Gaia-X, participants provide an externally-validated self-description of their IT security certifications, such as employee training, regular audits, or their compliance with common standards. Thereby, participants shall be assured that their data exchange partners adhere to common data security practices. However, credible technical guarantees remain absent; for instance, inside attackers currently pose a threat to data sovereignty in these systems [5].

In a similar federated setup of certificate authorities (CAs), the principle of CT addresses the vulnerability inherent in the issuance of misconfigured or maliciously acquired SSL/TLS certificates, which can compromise the integrity and security of the web. At its core, CT ensures the auditability of all certificates issued by CAs by using public, append-only logs of all issued certificates. These logs are audited and monitored by both domain owners and the public, ensuring that any anomalies are detected promptly. The logs function with the support of three main entities: log servers (receive and store certificates), monitors (watch logs for suspicious activity), and auditors (verify the correctness of the log). CT is widely used in the web ecosystem to increase the transparency and accountability of CAs and has also found application in other domains such as messaging [3], or privacy-preserving cloud storage [8].

3 TOWARD DATA USAGE TRANSPARENCY

To make data usage transparent, we equip data users with verifiable proof of their compliance with previously agreed terms. The key idea is that *data users want to utilize the information gained from data usage for some action involving another party*, such as sharing their recently gained knowledge for their benefit. E.g., if the data user is a car manufacturer receiving production plans for customized electronic boards, he will need to share that information with an electric board manufacturer. We denote this additional party the *observer*. Under the assumption of non-collusion between the data user and the observer, the latter can hold the former accountable for logging. Therefore, the observer's mere task is to check that any shared information comes with a matching proof of inclusion in a log. The proof of inclusion assures the observer that the data owner could inspect the log and hence becomes aware of misuse. Figure 1 outlines the corresponding protocol: During each data exchange (initiated via ①), the consuming connector creates a log entry, ② sends it to the log, ③ receives a proof-of-inclusion, and ④ shares that proof with an observer together with insights gained from the data. Finally, ⑤ the data owner can monitor shared items, e.g., by evaluating policies himself against the logged values.

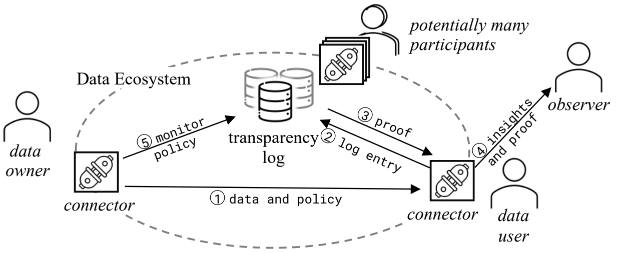

Figure 1: Overview of log integration into data ecosystem architecture.

Achieving these properties requires carefully chosen log entries, which allow verification by the observer and inspection by the data owner; otherwise, they should leak as little as possible information to third parties. Specifically, curious third parties must not learn any metadata of the exchanged data and must be unable to derive statistics, such as who exchanges data with whom. In the following, we thus detail the required information.

To enable verification by the observer, a log must contain some binding to the shared information. Therefore, we include a SHA256 hash of the data user's result, together with a timestamp and nonce, that data users need to share with the observer as part of the proof. Likewise, the inspecting data owner must be able to verify that the inspected transaction does not violate the policy to which the data exchange was subject to. To this end, we include the same claims as made in a policy also in the log entry, together with their assigned value during policy evaluation. Utilizing encryption keys provided by the data owner as part of the policy, we encrypt the latter portion, ensuring that only the data owner can inspect these potentially sensitive claims. The log entry itself can then be published under an identifier known by the data owner, e.g., derived from the encryption key.

With log entries consisting of an encrypted list of claims and a hash of the computed result, they do not leak specific content and cannot be referenced to the data owner or data user, other than by the encryption key or the hash with its parameters; both are not public knowledge, but only visible to the data owner and the observer, respectively. The efficacy of our scheme depends on the honesty of the observer; specifically, if the observer and the data user collude, they can avoid logging, and, for instance, exceed agreed terms. A special case of this collusion is that both entities could belong to the same organization. In this case, however, the absence of proof still helps whistleblowers to uncover misuse.

4 A PRELIMINARY EVALUATION

Transparency mechanisms need to keep up with expected ecosystem sizes and expansions, regarding both the number of participants and the number of transactions. Thus, we now show that an integration of cryptographic transparency logs into federated DEs to increase transparency in data usage is viable. To this end, we extended the Eclipse data space connector (EDC), an open-source implementation of the IDS connector specification, with features for transparency logging. We test against the implementation of Merkle^2 [3], a recent transparency log implementation that implements both efficient append and lookup operations by managing a prefix-tree besides a chronological log.

Implementation. We use the EDC together with the published code from [3]. For the latter, we adapted the published Merkle^2 code [3]; other log systems that provide timely proof of inclusion would also work, and we leave their analysis for future work. We instrument the EDC to generate and publish a log entry via gRPC to the log before any transaction completes.

All evaluations were conducted on a single machine (Intel i7-7700 processor, SSD storage, Ubuntu 22.04). For our experiments, we set up a providing and consuming EDC, exchanging a static file together with a policy restricting use to a number of five times and specifying the key under which log entries shall be published. We repeat all our experiments 30 times and provide averages over all runs unless stated otherwise.

Expected DE Sizes. Initiatives employing GAIA-X principles, for instance, in the automotive domain, target thousands of participants while most other initiatives, such as the Mobility Data Space (120 members), currently have fewer members. In addition, the frequency of data exchange and the configuration of policies is unknown. We thus test parameters well beyond current DE sizes (up to $k = 100\,000$ monitored contracts).

Performance Results. For data usage transparency to be viable, two criteria must be fulfilled: (a) the logging should not increase noticeable delays to data sharing, hindering its adoption and (b) it should not hinder scalability to larger scale deployments, as discussed above. We first consider the local overhead (a). After an initial settlement period (cf. Figure 2a (left)), a single exchange without contacting the log consumes 52.2 ms ±16. After integration of the log component, this time increases to 53.3 ms ±16, i.e., logging adds an overhead of 2.1 % in execution time. Synthetic measurements of the logging component only (Figure 2a (right)) reveal that these overheads remain relatively constant also for a higher number of appends. Here, k corresponds to the number of unique identifiers to be appended, e.g., $k = 1000$ (100) corresponds to 1000 (100) participants monitoring one (ten) items each. Also, we did not notice an impact of different log entry sizes.

Concerning the scalability to large DE deployments, we then test how many log entries and lookups a single log deployment can handle. To this end, we simulate clients with a synthetic connector in Go, mimicking log entries produced by the EDC. Figure 2 summarizes the results of this analysis. Here, we observe a runtime of at most 5.48 min to handle 100 000 keys with five log updates each (such as a policy granting multiple uses of data). Hence, we conclude that a transparency log would not limit DEs in their size.

Monitoring, Auditing & Proof Size. For data usage transparency, data owners must also be able to gather relevant log entries. To this end, our chosen log systems allows efficient querying log entries by the data owner's chosen key, i.e., they do not need to retrieve the full log for monitoring. Here, our use-case profits from Merkle^2's low-latency and efficient monitoring approach. We refer to [3] for a detailed analysis of monitoring and auditing costs as their analysis of these aspects already covers these aspects for scenarios of relevant size. A single proof of inclusion for a minimal-sized log entry, as handed over to the observer consumes 1874 bytes.

Overall, the preliminary results of our evaluation show that data usage transparency can scale to DE sizes while introducing little overhead for logging and reasonable operation costs.

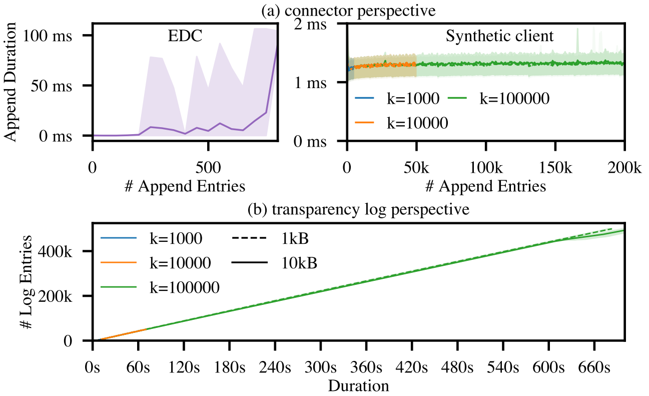

Figure 2: Overhead for logging usage transparency on the connector and the log server.

5 CONCLUSIONS AND FUTURE WORK

Federated data platforms decentralize the processing of data, rendering transparency hard to achieve. Here, our proposed concept of data usage transparency can help alleviate this situation using existing transparency log systems. Specifically, we have equipped data users with proof of their adherence to agreed terms. By having downstream information receivers expect and validate that proof, data owners can be assured that data users log their actions accountably. Our preliminary evaluation using a state-of-the-art transparency log indicates the feasibility of this idea for large-scale DEs. Hence, future research should consider and further investigate the verifiable guarantees such log systems can provide in increasing data usage transparency. First, the dependency on some third party validating the obtained proof limits the efficacy of our concept against inside attackers. Here, other trusted components, such as trusted hardware might help alleviate this situation and thus should be further evaluated. Second, our privacy-preserving log design prohibits, for instance, accountability across multiple hops [1] or aggregation of datasets, which in some cases might be desirable.

Acknowledgements. We thank Philipp Kuhn for his initial exploration of this topic.

REFERENCES

[1] L. Bader et al. 2021. Blockchain-based privacy preservation for supply chains supporting lightweight multi-hop information accountability. *Information Processing & Management*, 58, 3, 102529. DOI: 10.1016/j.ipm.2021.102529.

[2] A. Braud et al. 2021. The Road to European Digital Sovereignty with Gaia-X and IDSA. *IEEE Network*, 35, 2, 4–5. DOI: 10.1109/MNET.2021.9387709.

[3] Y. Hu et al. 2021. Merkle ^{2} : A Low-Latency Transparency Log System. In *2021 IEEE Symposium on Security and Privacy (SP)*. 2021 IEEE Symposium on Security and Privacy (SP). IEEE, San Francisco, CA, USA, 285–303. ISBN: 978-1-72818-934-5. DOI: 10.1109/SP40001.2021.00088.

[4] B. Laurie. 2014. Certificate transparency. *Communications of the ACM*, 57, 10, 40–46. DOI: 10.1145/2659897.

[5] J. Lohmöller et al. 2022. On the Need for Strong Sovereignty in Data Ecosystems. In *Proceedings of the 1st International Workshop on Data Ecosystems (DEco '22)*. Vol. 3306. CEUR-WS, Sydney, Australia, 51–63.

[6] B. Otto and M. Jarke. 2019. Designing a multi-sided data platform: findings from the International Data Spaces case. *Electronic Markets*, 29, 4, 561–580. DOI: 10.1007/s12525-019-00362-x.

[7] J. Park and R. Sandhu. 2004. The UCON $_{ABC}$ usage control model. *ACM Transactions on Information and System Security*, 7, 1, 128–174. DOI: 10.1145/984334.984339.

[8] S. Sundareswaran et al. 2012. Ensuring Distributed Accountability for Data Sharing in the Cloud. *IEEE Transactions on Dependable and Secure Computing*, 9, 4, 556–568. DOI: 10.1109/TDSC.2012.26.