Complementing Organizational Security in Data Ecosystems with Technical Guarantees

Johannes Lohmöller*, Roman Matzutt†, Joscha Loos*, Eduard Vlad*, Jan Pennekamp*, Klaus Wehrle*
*RWTH Aachen University, Germany · {lohmoeller,loos,vlad,pennekamp,wehrle}@comsys.rwth-aachen.de
†Fraunhofer FIT, Germany · roman.matzutt@fit.fraunhofer.de

***Abstract*—Federated data ecosystems continue to emerge to connect previously isolated data silos across organizational boundaries over the Internet. These platforms aim to facilitate data sharing while maintaining data sovereignty, which is supposed to empower data owners to retain control over their data. However, the employed organizational security measures, such as policy-enforcing middleware besides software certification, processes, and employees are insufficient to provide reliable guarantees against malicious insiders. This paper thus proposes a corresponding technical solution for federated platforms that builds on communication between Trusted Execution Environments (TEEs) and demonstrates the feasibility of technically enforceable data protection. Specifically, we provide *dependable guarantees* for data owners formulated via rich policies while maintaining usability as a *general-purpose data exchange platform*. Further, by evaluating a real-world use case that concerns sharing sensitive genomic data, we demonstrate its real-world suitability. Our findings emphasize the potential of TEEs in establishing trust and increasing data security for federated data scenarios far beyond a single use case.**

I. Introduction

As next-generation information systems, federated Data Ecosystems (DEs) strive to connect formerly isolated data silos across organizational boundaries. For instance, GAIA-X or IDS plan to establish DEs for data exchange between companies and beyond [1] by providing standardized interfaces for the discovery of and collaboration on data. Others plan to share sensitive medical records for treatment and research [2] or replace monopoly-like structures with interconnected networks to facilitate competition [3]. With sensitive trade secrets and shared personal data, data sovereignty is one of the most important reasons for implementing federated DEs [4]. It shall empower data-owning entities (*data owners*) to retain control over their data, e.g., concerning storage, further processing, and sharing with others. To this end, federated data ecosystems enable trustworthy processing without losing control by moving data processing from public cloud providers to dedicated infrastructure under the control of the involved entities [1]. A dedicated infrastructure alone, however, is insufficient to control further data processing [4]. Thus, federated DEs implement usage control principles, including rich policy languages that specify specific permissions and restrictions for data handling and methods to enforce the former [4].

To realize data sovereignty, and more specifically, the enforcement of policies, today's DEs mostly source an organizational approach, e.g., via standardized workflows, certification of IT systems, and employee training [5]. However, such "organizational security" fails to provide reliable guarantees against malicious insiders [4]. Malicious insiders are a credible threat to federated DEs, as participants involved in a data exchange all have powerful insider positions while being incentivized to evade policies to their benefit [4]. Hence, when processing sensitive data, technical guarantees are urgently needed.

To address this gap, we thus propose establishing insider-resistant guarantees by handling data exchanges in federated DEs using *TEEs*. TEEs enable general-purpose, confidential, and integrity-protected computing in hardware within an untrusted environment and thus, introduce fewer restrictions regarding computability than other cryptographic approaches. Moreover, the trust chain of TEEs does not depend on local administrators or the physical owner of a system, rendering them an interesting building block for federated DEs with opaque trust relationships. Consequently, the suitability of TEEs to improve the trust in data exchanges in DEs despite their constrained processing capabilities and required attestation mechanisms remains to be investigated.

Our results indicate that TEEs are indeed suitable to implement *technically enforceable data protection for federated DEs feasibly and conveniently*. Most notably, TEEs not only mitigate the problem of inside attackers but also enable enforcing policies, such as *delete after use*, which are hard to implement otherwise [6]. However, policies with practical deletion obligations are among the most important policies for sovereign data exchange between not fully trusted participants [4]. Jointly, this combination enables the trusted enforcement of policies in federated architectures, which equips data owners with technical guarantees. Compared to most others [7, 8, 9], our approach enables protection throughout the complete lifecycle of data once it has been shared. Via these strong guarantees, we extend the applicability of federated DEs to diverse use cases involving sensitive data. In summary, the main contributions of this work are as follows:

- Our newly proposed TEE-backed federated data platform provides technical data sovereignty guarantees for DEs, even with mutually distrusting entities while easing the sharing of sensitive data.
- We show that relying on TEEs even adds capabilities to express and technically enforce rich usage-control policies, including obligations like delete after use or use n times.
- Our security and performance evaluation, a use case based on exchanging genomic data, underpins our design's practicality in facilitating the sharing of (sensitive) data.

©2024 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. DOI: 10.1109/BuildSEC64048.2024.00016

Structure. The remainder of this paper is structured as follows. In Section II, we provide the preliminaries for our work. Section III then details the assumed scenario and derives requirements for data-sharing platforms with technically enforceable data-usage policies. In Section IV, we present and discuss our proposed TEE-based architecture. Section V demonstrates our approach's feasibility by benchmarking a prototypical implementation on a real-world use case for analyzing sensitive data. Finally, we discuss related work in Section VI before concluding in Section VII.

II. Preliminaries

To underpin the importance of technical guarantees, we briefly recapitulate data sovereignty as a concept, provide essential aspects of policy enforcement, and introduce TEEs.

Data Sovereignty. Data sovereignty focuses explicitly on controlling dataflows and the power to exercise control over one's data across boundaries, such as national or organizational boundaries [10]. Accordingly, different actors, such as nations, organizations, and individuals, can facilitate or demand data sovereignty. For instance, current privacy legislation by the EU [1] forces users to be aware of dataflows and limits what data can be processed where, when, and by whom. The fundamental concept behind the idea is that specific infrastructures, operators, or locations are more trustworthy and shall be preferable by individual choice.

Organizational Security. To establish the necessary trust in the information system as a whole, as well as to improve mutual trust between participants, today's DEs often depend on some form of (external) certification of IT systems and training, e.g., similar to ISO 27001 [4]. Thereby, organizational procedures are validated to ensure compliance with the DEs rules and the data owners' terms.

Policy Enforcement. Policies provide means to formulate specific data sovereignty requirements or constraints in a machine-readable format. Commonly, we distinguish between access control policies (checked once before access) and usage control policies (continual evaluation). Policy languages, such as XACML [11], provide an extensible framework to specify access and usage policies and serve as input to policy enforcement frameworks. Access policies can be evaluated via proxies but also embedded in cryptographic keys, e.g., via attribute-based encryption [12]; usage control policies, on the other hand, depend on additional middleware or tools that control the processing of protected data for their enforcement [13]. TEEs, for instance, provide such enforcement capabilities [9].

TEEs. TEEs are a feature of modern CPUs that provides an isolated environment for running general-purpose code. Specifically, the isolated environment enables integrity protection and confidentiality for code and data *in use*, including protection against adversaries on the same system. Well-known realizations of commercial vendors are Intel SGX/TDX, or AMD SEV-SNP [14, 15], which are supported by large cloud providers. These realizations provide a chain of trust rooted in the CPU vendor and, implement (remote) attestation mechanisms that offer proof to externals concerning code integrity and hardware validity. By trusting the CPU vendor with a secure realization, these features allow running code in an isolated, confidential, and integrity-protected fashion within an otherwise untrusted environment [14]. First generation TEEs impose strict computation limits, including static memory and thread allocations with narrow upper limits [14]; however, newer generations, such as Intel TDX [15] significantly extend these limits. Likewise, the security of specific TEEs realizations has been broadly researched, including the discovery of several weaknesses due to side-channel attacks and implementations flaws [16, 17], besides corresponding mitigation strategies [17, 18], which, e.g., source oblivious RAM strategies to conceal memory access patterns. These are orthogonal to our work. We thus consider its fundamental concept to reliably provide technical security guarantees.

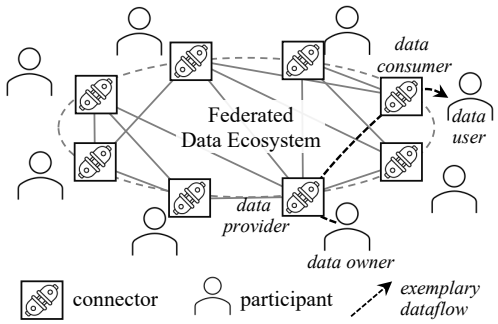

Fig. 1. The technical platform for federated DEs relies on *connectors* to provide interfaces for bilateral data exchange. An exemplary dataflow shows the *roles* of the involved entities.

III. Toward Sovereign Data Sharing

Based on a brief description of the setting federated DEs typically operate in, we illustrate how sharing sensitive data can work in a trusted fashion, state the corresponding threat model, and derive general requirements for federated DEs.

A. Setting: Tearing Down Data Silos

Generally, we assume a one-to-one relationship between potentially many participants: one party, i.e., the *data owner*, offers a data item another party, i.e., the *data user*, is interested in. To this end, both need a technical interface for (a) discovering, (b) requesting, (c) transferring the dataset, optionally with (d) some form of transparency on its intentions. Recently proposed *DEs* form information systems that aim to provide a generic workflow for the above task and implement defined *interfaces* for these tasks [19]. To this end, with the *data provider* and *data consumer*, two other entities participate in the process on behalf of the data owner and data user, providing the necessary features for (a) to (d). So-called *data space connectors* [20] implement standard interfaces for these features, provide a translation layer, e.g., to work with internal databases, and serve as a gateway and audit tool for dataflows.

The described setting generally applies to a diverse set of use cases in which DEs as central, overarching information systems are currently considered. Such use cases include, but are not limited to, digital production, smart cities, as well as healthcare and medical research [4, 21].

B. Threat Model

We employ the threat model of a *malicious-but-cautious inside attacker* [4, 22], i.e., participants may act maliciously but fear losing their reputation as benign DE participants and therefore cautiously avoid detection by externals. As participants are authenticated, publicly known and often represent some legal entity, we argue that a poor reputation will prohibit future participation in DEs and thus, participants have incentives to create an impression of protocol compliance. Internally, participants aim to maximize their benefit from the data exchange, including direct access to the data for future use or evading policies to run analyses without the data owner's permission; they are only limited by technical or economical means. Likewise, platform operators might be interested in the data for their benefit, e.g., to improve their services or to sell the data to third parties. As DE participants are usually known in advance (e.g., listed in a federated catalog [1]), we do not consider communication metadata, such as who communicates with whom, as sensitive information. Established mechanisms, such as onion routing [23], can protect this information if necessary.

C. Requirements for Widely Accepted DEs

The following general requirements guide the design of information systems that support the setting of DEs, as illustrated in Section III-A. We consider them to be equally relevant when designing practical federated data platforms:

R1: General Purpose Data. One of the goals of federated DEs, such as GAIA-X, is to simplify previously rather tedious bilateral data exchange methods [19]. While having many downsides, such as a lack of traceability, one benefit of the prior workflow is that it imposes few restrictions regarding the processing or volume of such data. To avoid adoption issues, federated data platforms should retain this general computability and processing efficiency concerning larger datasets.

R2: Mutual Distrust. Current federated data platforms employ usage control and fine-grained policies to alleviate this situation and allow for formulating permissions and constraints concerning data handling [24]. In the applied federated setting (cf. Section III-A), participants must trust others to comply with these agreed policies [4]. With incentives to evade policies [4], however, these assumptions hardly work together. Instead, data platforms should be able to *handle mutual distrust* rather than depend on any form of trust between participants.

R3: Federated Architecture. Besides facilitating data exchange, dataflow regulation laws are a central objective of legislators worldwide (cf. Section II). To this end, a federated architecture, i.e., a decentralized system with few central services, such as admission control and identity management, is intentional and desirable. It allows *removing dependencies on cloud providers* or other third parties involved in data processing, a central objective of typical data sovereignty strategies.

R4: Long-term Availability. Today's economy sometimes can be short-lived [3, 25]—a property federated data platforms must account for. This aspect includes access to data provided by a party that no longer exists. While one might argue that enforcement is no longer needed after participants are gone, datasets might involve the rights of others or are a composition of other datasets to which owners still have valid claims. To this end, it should be feasible to enforce policies and *evaluate datasets without the active participation of the data owner*, i.e., maintain the data user's sovereign decision on using the dataset within previously agreed terms. From a different perspective, the involvement of the data owner might even leak access patterns a data user would like to keep private.

R5: Open Platform and Incentives for Participation. The value of federated data platforms critically depends on those entities willing to provide data. As such, platforms should lower the burden of participation as much as possible [19, 26]. To this end, we argue that participants typically have incentives to retrieve datasets to compute something beneficial from them. Especially data owners sometimes see little benefit in sharing data with others if sharing is not to their own benefit [19]. While monetary compensation can provide such an incentive, participation should not be overly complicated. Instead, data users who are interested in the results should also be responsible for providing computing infrastructure and doing calculations.

Overall, the Requirements R1–R5 capture what is needed to ensure the data sovereignty of data owners. However, it is unclear how to technically realize these requirements to facilitate participation while enabling use cases involving sensitive data.

IV. A TEE-Assisted Federated Architecture for Sovereign Data Exchange

We now present a TEE-based approach to implement sovereign data sharing that ensures ongoing enforcement of specific policies in untrusted deployment settings and show how our approach meets the requirements outlined in Section III-C within the efficiency and resource limitations of TEEs (cf. Section II).

A. General Concept and Outline

We complement the concept of data space connectors [20] with TEE-based trusted infrastructure. As the threat model of TEEs considers local hardware and software beyond the inner trusted environment as potentially malicious, TEEs are a promising building block in this context. We also require subsequent processing inside a TEE, orchestrated by the connector implementation, which primarily forwards data to downstream entities [20]. Thereby, our TEE-based architecture allows for building a *chain of trust* from the data owner to an environment provided by data users that can be entrusted with enforcing expressive policies on the remote end and processing sensitive data. The chain of trust forms a transitive and long-term (R4) trust relationship between the data owner, the DE infrastructure, and (future) analysis code to be run by data users. The requirement for mutual trust between participants within federated data platforms can then be lifted. Compared to others [13], our design can enforce sovereignty requirements with technical guarantees in malicious environments (R2),

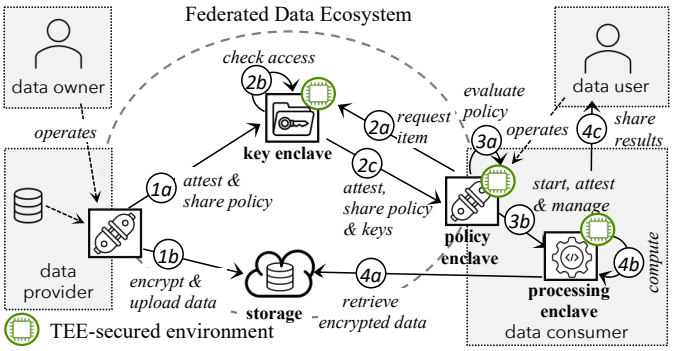

Fig. 2. Via the shown architecture and process, which employs TEEs and attestation, we ensure technical guarantees regarding data sovereignty for a data owner, even in a federated and trustless setting.

including that data will be deleted after usage or may only be used within a specific timeframe (cf. Section IV-C).

Fig. 2 visualizes the typical flow for initiating such a data exchange. By completion, the data user has computed (R5) an analysis result on generic data (R1), e.g., a classification or a SQL query result against a dataset he was never granted direct access to. Thereby, the data owner can be sure that her data was never copied or shared for an intention she did not agree to; she thus maintains her data sovereignty (R3).

B. Data Sharing Process

In the following, we detail the process for each entity in the order of involvement when data is shared.

Data Owner and Data Provider. To initiate the chain of trust, the data owner first needs to establish trust in the DE. "Trust" in this setting refers to the assurance that infrastructure and analysis code executed by the entities behaves as required by the protocol. To this end, we rely on the TEE's attestation capabilities, i.e., ensure that the central entity runs a specific binary on a valid platform in an isolated environment. To ensure platform validity, the data provider attests the central entities' enclave (Step 1a). If data owners trust the binary (e.g., after an audit), they can then be sure that the central entity behaves as expected, i.e., enforces their policies and treats shared secrets confidentially. Regarding the sharing of the data item itself, our approach makes few restrictions, i.e., we assume that the data owner will encrypt it efficiently using symmetric encryption and upload it to some storage space (Step 1b). Afterward, data can be obtained authentically with keying material distributed along the chain of trust. Also, further interaction with the data provider is unnecessary, so it can go offline (R4).

Key Enclave. Upon a request for data by a data user (Step 2a), the central enclave evaluates an access control policy (Step 2b) to check whether the requesting entity may request a data item (as specified by the given policy). If access is granted, the central key enclave validates the policy enclave at the consuming data space connector (Step 2c). This step extends the transitive trust relationship from the data owner to the data user's policy enclave. Upon successful attestation, the central key enclave can thus transfer the policy, keying material, and reference to the data item to the data user's policy enclave.

Policy Enclave. With the received policy, keying material and dataset reference, the data user's attested policy enclave can now initiate analysis. Therefore, it first evaluates the received policy for rightful access to the dataset (Step 3a). Here, the enclave can also consider any local state, such as how often a policy was previously evaluated, the current time, or (signed) attributes provided by the data user beforehand. If the policy check succeeds, the enclave bootstraps another enclave with code explicitly allowed by the policy (Step 3b). This code must either be known before granting access (i.e., the data owner does precisely know what will be used to evaluate her shared data), or, for instance, the data space association can offer a registry from which the data owner can allow specific analyses on his data.

Processing Enclave. The processing enclave retrieves the dataset (Step 4a) and runs the user-requested analysis (Step 4b). It marks the tail of the chain of trust; data owners at the head of the chain thus can be sure that their policy will be enforced at runtime. The data user will only see the result he is interested in once the enclave shares computed results (Step 4c). Via the specified policy, limiting the data user's actions, the data owner has implictly agreed to share these insights.

As the policy enclave organizes the processing, further actions, such as the notification of the data owner, transparency, or provenance mechanisms, can be triggered in a trusted fashion. For instance, the trusted environment could be used to sign results, enabling multi-hop data provenance or reporting usage back to the data owner to increase usage transparency. While this work focuses on bilateral exchange, the processing enclave could also be used to publish results as new data items within the DE, thereby enabling multi-hop data processing or aggregation of several source datasets.

We intentionally leave the question of how to match data owners with potential data users and how to agree or negotiate a usage policy open, as these are orthogonal to the security-focused objectives of this work. For instance, one could implement a global catalog for data or handle discovery out-of-band. For non-interactive negotiation, the data user could pick from a set of policies the data owner agreed to.

C. Advanced Policy Enforcement

In Section IV-A, we have established the notion of policy enforcement in enclaves: the central key enclave enforces access control on DE participant level, whereas the policy enclave can also enforce usage control policies at runtime in a scalable fashion. In the following, we discuss that this architecture enables advanced policy enforcement, namely obligations that require data deletion or depend on some local state.

Requiring the deletion of data after use is a common, hard-to-implement policy in data management [4, 6]. We argue that for federated DEs, this notion must be extended to "deleted after use and never been copied before" to maintain the data owner's sovereignty in adversarial settings [4]. Only if no copies of the

data exist deletion is effective regarding data sovereignty. Our employed TEEs can enforce this notion even in the presence of inside attackers (cf. Section III-B) due to their isolation properties, i.e., data can only be copied by the enclave itself, or due to vulnerabilities in the TEE or enclave implementation. In the former case, the data owner has previously agreed to this behavior. In the latter, the data owner or key enclave can at least prevent execution on platforms known to be vulnerable (cf. Section IV-D). This behavior is also a requirement for dependent policies, such as time-based access or restrictions on the number of uses.

Besides effective deletion, obligations such as "use n times" require stateful policies. Related schemes, such as attribute-based encryption, can hardly fulfill this requirement as attributes are static; they thus depend on further infrastructure, such as distributed ledgers, to maintain authenticated, integrity-protected, and thereby trusted state [27]. Our architecture can instead utilize TEE's sealing capabilities to maintain local state across policy evaluations, which scales well with the number of data items and variables to be maintained. Sealed state can only be accessed by the enclave that sealed it or, optionally, by later instances of the same enclave, enabling its authenticity and integrity.

Finally, our architecture does not depend on a specific policy language but can evaluate any code and thereby handle more advanced obligations. Such an obligation could be, for instance, to require the data user to tell the data owner the result of the analysis or even let the data owner decide whether the result should be shared or not. Overall, these properties increase the data user's sovereignty.

D. Security Discussion

As motivated in Section III-B, insiders pose a significant threat to federated information systems, equipped with incentives to disobey agreed terms of data exchange [4]. We thus now show that our work provides technical guarantees against insiders.

Specifically, our system architecture utilizes TEEs to technically enforce security policies. For one, they provide an integrity-protected environment for evaluating policies: actors cannot simply disobey policies prohibiting data usage. Moreover, their shielding mechanisms hide data from malicious access or unwanted copies. To this end, access to data can be effectively time-bound or revoked after usage. Both of these measures maintain confidentiality and integrity whenever data is in use. For data at rest and in transit, we employ symmetric cryptography and TLS, reflecting best practices in data security.

A simplistic attack in this setting would be presenting the identity function, or similar, as an algorithm, which passes a dataset to the data user as is. Here, our architecture needs some form of algorithm review. For instance, a simple yet practically limited solution would be to include the exact algorithm in the usage policy. A more advanced and practical remedy to this attack could be a repository of generally accepted data processing algorithms, e.g., as already provided for containers on Dockerhub. Then, a policy could grant access to all software from a trusted vendor or allow software to be audited and signed by a trusted entity.

In our scenario, attackers notice the runtime of the TEE and the size of the encrypted data item on disk and when transferred via the network. In addition, they can, at any time, interrupt or stop running enclaves. Hence, we cannot guarantee the availability of the computing results. However, since, on the one hand, the data owner computes in his interest, and on the other hand, the state of policy evaluation cannot be reset, the attacker has no incentive to interrupt enclaves.

Overall, we are confident that our proposed architecture securely increases data sovereignty for federated data platforms well beyond today's primarily organizational security measures [4].

E. Discussion of Design and Desired Requirements

In this section, we argue that our approach is suitable for implementing federated DEs with technical guarantees by revisiting our requirements R1–R5.

First, processing all data within TEEs maintains the capability to perform general-purpose computations (R1). While these computations are subject to the limitations of current TEEs, the key and policy enclave only handle policies, attributes and encryption keys, which are several magnitudes smaller than the 128 MB memory limit of the original SGX version. We further discuss ongoing developments aiming at reducing these limitations in Section V-C.

Moreover, relying on a TEE-driven chain of trust reduces the current need for mutual trust among participants (as required by organizational security [4]) by establishing trust in the underlying infrastructure itself (R2). Thereby, the data sharing requires no bilateral trust on the level of participants or in the platform itself (R2).

We further argue that our design retains a federated architecture (R3). Although the key enclave acts as a centralized proxy between a data owner and a data user for every data exchange, this service is only logically centralized and not bound to a single service provider. Moreover, no respective service provider can observe or modify the policies a data owner manages via their key enclave. Instead, we explicitly designed our approach to satisfy our non-interactivity requirement (R4). Namely, data owners may go offline after sharing a dataset as the key enclave will continue to enforce their policies on their behalves.

Finally, our architecture minimizes the computational efforts the data owner requires, thereby incentivizing participation due to the added value of enforceable data usage policies (R5). The data owner's responsibilities are limited to supplying a symmetrically encrypted dataset along with its usage policy and conducting remote attestation to ensure that policies and decryption keys are only distributed to trustworthy entities. None of these tasks are computationally complex (cf. Section V) or require special hardware on the data owner's end. Hence, our design keeps the barrier to participation by data owners minimal.

In summary, our TEE-based strategy significantly enhances the value proposition of federated DEs by offering technical security guarantees to data owners and enabling sensitive data access for data users with a tolerable performance trade-off, as we will show in the following section.

V. Realization and Evaluation

To show the feasibility of our design, we evaluate the performance of all involved entities based on a sensitive real-world dataset covering genomic disease analysis. Therefore, we briefly discuss relevant implementation considerations, study the performance and scalability of our proposed design, and discuss its applicability within information systems besides security and trust implications.

A. Setup and Prototypical Implementation

We implement the key and policy enclaves, as well as an exemplary analysis algorithm, using Rust due to its memory safety features and good library support.

For our performance evaluation, we utilized the Intel SGX v2 TEE on an Intel Xeon Gold 5411N processor, employing the Fortanix Rust SGX SDK. However, we stress that this SDK does not yet fully exploit SGXv2 capabilities like dynamic memory and thread allocation. Despite this limitation, our findings are conceptually applicable to other TEE platforms.

Our system architecture involves secure communications over TLS, exchanging JSON messages for necessary operations such as attestation via Intel Attestation Service, authentication through pre-shared secrets, policy, and key exchanges. We defined a simple JSON-based policy language to specify the conditions and restrictions on data access and usage, i.e., who can do what and how often.

Moreover, we distribute algorithms as signed binaries, similar to the approach of shipping containerized applications. This setup exemplifies the standard architecture of proposed data-sharing ecosystems, ensuring both security and operational integrity.

B. Benchmarks

To analyze the performance and applicability of our architecture (addressing R1 and R5), we first analyze the overhead of data processing inside the enclave. Exemplarily, we use a dataset "testbed" containing genomic data, previously published as part of the iDASH genomic data analysis challenge [28]. The dataset has been anonymized and contains no personal identifiers; otherwise, it does not differ from a real-world dataset. We evaluate the original challenge task, i.e., cluster gene sequences. Specifically, we are interested in the overhead and resource consumption occurring during the sharing process, which is generally independent of the data size or analysis complexity. As such, evaluating different datasets or analyses would not provide additional insights such that we refer to prior work for further use cases [4] and a more specific analysis of in-enclave performance measurements [29]. We conduct 30 repetitions of all experiments and present median results with 95 % confidence intervals.

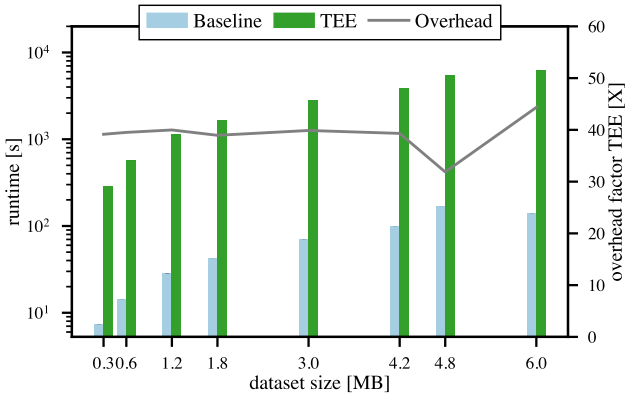

Fig. 3. We compare the runtime of sharing and processing a genomic dataset via our proposed architecture with and without TEE-based protection.

First, we are interested in the performance of running all entities inside a TEE. Fig. 3 compares the overhead of conducting a data exchange and analysis within our architecture inside a TEE to no protection at all, which matches the setting most data platforms apply today [4]. Comparing the baseline and TEE-based variants (see Fig. 3, right y-axis), we find a consistent overhead factor within the TEE of 50.6 ± 9.5 compared to runtime with no protection. This runtime covers the complete process in Fig. 2, i.e., from sharing the dataset at the data owner (Step (1a)) to retrieving the result by the data user (Step (4c)). While this overhead is a substantial penalty, it still shows that processing generic data within a reasonable time is feasible, as the performance penalty factor remains almost constant with increasing dataset size.

The two necessary remote attestations (Step (1a), Step (2c)), together only consume $138.4\,\text{ms} \pm 19.2\,\text{ms}$ and are independent of the dataset size. With the calculation itself accounting for 54.87 ms to 150.26 s for the smallest and largest datasets, respectively, almost all remaining time can be attributed to loading data into or retrieving data from the enclave. As de-/encryption happens inside the enclave, whereas disk access is handled outside, these operations involve frequent and costly context switches. More recent implementations of TEEs lift this requirement [14], as they protect complete operating systems.

Despite the performance drawbacks observed in our initial implementation, we determine that the use of our architecture in real-world applications is already feasible. The primary limitations——restricted memory and limited communication with the enclave——are specific to the TEE design and expected to be resolved in future iterations; they do not fundamentally challenge the viability of our proposed architecture. We emphasize that the benefits of preserving general computability within TEEs (R1), enabling collaboration without mutual trust (R2), and requiring minimal participation effort (R5), outweigh these initial limitations. Future optimizations may include strategies such as multiplexing TLS connections outside the enclave, with only attestation requests and encrypted data handled within the enclave.

C. Applicability and Integratability

Our approach aligns with large-scale data ecosystems like GAIA-X or IDS initiatives, which are challenged by trust issues due to their federated architecture that primarily relies on organizational security [4]. Unlike these initiatives, our design remains unaffected by trust issues among participants due to its robust architectural features (see Section IV-D). We thus propose integrating our design into these ecosystems.

Contrary to the federated architecture requirement, we depend on a (logically) central entity to conduct key management. This requirement, however, does not limit applicability, as data ecosystems usually depend on some central organization operating infrastructure for the discovery and registration of participants or data. We minimize the central data storage to reduce the burden on the central key management enclave.

Moreover, we follow the flow in which data exchange is initiated and conducted. As for our analysis, no interoperability with existing data space connectors, such as the Eclipse Data Space connector [20] is necessary; we thus refrain from an implementation. However, implementing such an integration would not pose conceptual challenges and only minor technical difficulties, mostly related to the TEE-specific memory limitations.

VI. Related Work

Establishing trust, i.e., the necessary prerequisite for sovereign data sharing, has been approached in various contexts with many methods, as briefly introduced in the following.

Trustless Distributed Systems. Establishing trust in distributed systems is challenging, exemplified by consensus methods used in Bitcoin and other cryptocurrencies [30]. Both blockchains and distributed ledgers address scenarios involving mutually distrustful parties, similar to ours, and can enhance trust across various distributed environments [31]. Unlike systems that depend on a single remote entity, these technologies rely on an honest majority to facilitate participant trust. They typically focus less on computation-intensive tasks and more on reducing data storage and computational efforts [32]. Additionally, blockchain technology is employed for enhancing data usage transparency [33], access control [9], and provenance tracking [34]. For example, Xiao et al. use Ethereum to log usage, promoting transparency [32]. As such, distributed ledgers have shown to be a helpful building block but, alone, cannot establish mutual trust for federated DEs.

Usage Control. Various methods focus on modeling and enforcing specific data handling protocols before or during data access, referred to as *access* or *usage* control. UCON [35] provides a foundational model for this, supporting different policy languages like XACML [11] to define machine-readable permissions, prohibitions, and obligations [36]. Enforcement centers around cloud infrastructures [7], requiring trust in the provider or through middleware that manages data access and usage [8], necessitating trust in the local environment. However, such middleware does not suit federated DEs due to its reliance on trust in remote systems [4]. Concerning the remedy to this problem that we propose in this work, few others rely on TEEs to enforce usage control policies in this setting. For instance, Lei et al. rely on a TEE to share datasets in a trusted manner via a decentralized ledger [9]. Our proposed architecture supports these models but additionally guarantees reliable enforcement.

Sovereign Data Ecosystems. Research on sovereign data sharing often adopts a broad, non-technical perspective, with less emphasis on usage control [37, 38]. For example, Oliveira and Lóscio discuss general data ecosystems [39], while Ibrahim and Dimitrakos analyze security measures for data sovereignty within this framework [40]. Implementations reveal that reliable usage control mechanisms are critical yet challenging for such platforms. Pampus et al. identify these mechanisms as vital for data ecosystems [20], and Lohmöller et al. highlight that organizational measures are inadequate for protecting against insiders [4]. In the context of the IDS, using "trusted connectors" based on TPMs has been proposed [20] but falls short of the security afforded by TEEs. Consequently, while organizational security measures are commonly proposed, they often fail to deliver robust guarantees in DEs.

Complementary Data Sharing Approaches. In addition to cryptographic, technical, and organizational methods of data sharing, alternative strategies emphasize privacy-preserving dataset publication, such as differential privacy [41] or enhancing privacy through latent representations [42]. These methods complement our infrastructure-focused approach by concentrating on the data itself. A possible integration could be requiring such methods to release analysis results to the user, thereby increasing data privacy.

VII. Conclusion

In this work, we have demonstrated that TEEs provide a viable solution to current problems of federated DEs concerning inside attackers. Inside attackers are indeed a reasonable threat to these information systems, as most of them solely rely on organizational security. Our architecture for a federated data platform accounts for this threat and allows the trusted, reliable enforcement of complex policies, including stateful policies such as *delete after use* or the obligation to communicate results back to the data owner. Our benchmarks show reasonable overall overhead, with most work carried out by the data user, who is incentivized to handle this overhead since data otherwise might not be accessible. Consequently, we consider TEEs a good fit to provide data confidentiality and trust in federated DEs. Future work needs to pursue integration into large-scale initiatives, such as GAIA-X or IDS, to realize data sovereignty on a large scale eventually.

Acknowledgments

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC-2023 Internet of Production – 390621612.

References

[1] A. Braud et al., "The Road to European Digital Sovereignty with Gaia-X and IDSA," *IEEE Netw.*, vol. 35, no. 2, 2021.

[2] T. Berlage et al., "Medical Data Spaces in Healthcare Data Ecosystems," in *Designing Data Spaces*, Springer, 2022.

[3] R. Nicholls, "Interconnection of platforms: A case study in property transfer," in *ITS Online Event*, 2020.

[4] J. Lohmöller et al., "The unresolved need for dependable guarantees on security, sovereignty, and trust in data ecosystems," *Data & Knowl. Eng.*, vol. 151, 2024.

[5] N. Menz et al., "Framework for the IDS Certification Scheme 2.0," IDSA, version 2.0, 2019.

[6] C. Yang et al., "Blockchain-based publicly verifiable data deletion scheme for cloud storage," *Journal of Network and Computer Applications*, vol. 103, 2018.

[7] E. Carniani et al., "Usage Control on Cloud systems," *Future Generation Computer Systems*, vol. 63, 2016.

[8] Q. H. Cao et al., "Policy-based usage control for a trustworthy data sharing platform in smart cities," *Future Gener. Comput. Syst.*, vol. 107, 2020.

[9] H. Lei et al., "SDSBT: A Secure Multi-party Data Sharing Platform Based on Blockchain and TEE," in *Cyberspace Safety and Security*, vol. 12653, Springer, 2021.

[10] P. Hummel et al., "Data sovereignty: A review," *Big Data & Society*, vol. 8, no. 1, 2021.

[11] M. Colombo et al., "A Proposal on Enhancing XACML with Continuous Usage Control Features," in *Grids, P2P and Services Computing*, Springer US, 2010.

[12] J. Bethencourt et al., "Ciphertext-Policy Attribute-Based Encryption," in *IEEE SP '07*, 2007.

[13] F. Kelbert and A. Pretschner, "Data usage control enforcement in distributed systems," in *CODASPY*, 2013.

[14] M. Schneider et al. "SoK: Hardware-supported Trusted Execution Environments." arXiv: 2205.12742. 2022.

[15] P.-C. Cheng et al., "Intel TDX Demystified: A Top-Down Approach," *ACM Comput. Surv.*, vol. 56, no. 9, 2024.

[16] A. Nilsson et al., "A Survey of Published Attacks on Intel SGX," 2020. arXiv: 2006.13598.

[17] R. Buhren et al., "Insecure Until Proven Updated: Analyzing AMD SEV's Remote Attestation," in *ACM CCS '19*, 2019.

[18] H. B. Lee et al., "DOVE: A Data-Oblivious Virtual Environment," in *NDSS '21*, 2021.

[19] C. Capiello et al., "Data Ecosystems: Sovereign Data Exchange among Organizations (Dagstuhl Seminar 19391)," Schloss Dagstuhl, Germany, version 1.0, 2020.

[20] J. Pampus et al., "Evolving Data Space Technologies: Lessons Learned from an IDS Connector Reference Implementation," in *Leveraging Applications of Formal Methods, Verification and Validation. Practice*, vol. 13704, Springer Nature Switzerland, 2022.

[21] J. Gelhaar and B. Otto, "Challenges in the Emergence of Data Ecosystems," in *PACIS '20*, 2020.

[22] M. D. Ryan, "Enhanced Certificate Transparency and End-to-End Encrypted Mail," in *NDSS '14*, 2014.

[23] J. Hiller et al., "Tailoring Onion Routing to the Internet of Things: Security and Privacy in Untrusted Environments," in *ICNP '19*, 2019.

[24] J. Zrenner et al., "Usage control architecture options for data sovereignty in business ecosystems," *JEIM*, vol. 32, no. 3, 2019.

[25] J. Pennekamp et al., "An Interdisciplinary Survey on Information Flows in Supply Chains," *ACM Comput. Surv.*, vol. 56, no. 2, 2024.

[26] J. Lohmöller et al., "On the Need for Strong Sovereignty in Data Ecosystems," in *DEco '22*, 2022.

[27] D. Di Francesco Maesa et al., "Blockchain Based Access Control," in *Distributed Applications and Interoperable Systems*, vol. 10320, Springer, 2017.

[28] *iDASH Secure Genome Analysis Competition*, 2016. [Online]. Available: http://www.humangenomeprivacy.org/2016/.

[29] R. Krahn et al., "TEEMon: A continuous performance monitoring framework for TEEs," in *MIDDLEWARE '20*, 2020.

[30] Y. Xiao et al., "A Survey of Distributed Consensus Protocols for Blockchain Networks," *IEEE Commun. Surv. Tutorials*, vol. 22, no. 2, 2020.

[31] K. Gai et al., "Blockchain Meets Cloud Computing: A Survey," *IEEE Commun. Surv. Tutorials*, vol. 22, no. 3, 2020.

[32] Y. Xiao et al., "PrivacyGuard: Enforcing Private Data Usage Control with Blockchain and Attested Off-Chain Contract Execution," in *ESORICS '20*, Springer, 2020.

[33] E. Bertino et al., "Data Transparency with Blockchain and AI Ethics," *J. Data Inf. Qual.*, vol. 11, no. 4, 2019.

[34] W. Dai et al., "SDTE: A Secure Blockchain-Based Data Trading Ecosystem," *IEEE Trans.Inform.Forensic Secur.*, vol. 15, 2020.

[35] J. Park and R. Sandhu, "The UCON $_{\text{ABC}}$ usage control model," *ACM Trans. Inf. Syst. Secur.*, vol. 7, no. 1, 2004.

[36] Y. Sun et al., "Data Security and Privacy in Cloud Computing," *Int. J. Distrib. Sens. Netw.*, vol. 10, no. 7, 2014.

[37] J. Gelhaar et al., "A Taxonomy for Data Ecosystems," in *HICSS '21*, 2021.

[38] M. Huber et al., "Building Trust in Data Spaces," in *Designing Data Spaces*, Springer, 2022.

[39] M. I. S. Oliveira and B. F. Lóscio, "What is a data ecosystem?" In *dg.o '18*, 2018.

[40] A. Ibrahim and T. Dimitrakos, "Towards Collaborative Security Approaches Based on the European Digital Sovereignty Ecosystem," in *Collab. Approaches for Cyber Secur. in CPS*, Springer, 2023.

[41] M. F. S. John et al., "Decision Support for Sharing Data using Differential Privacy," in *IEEE VizSec '21*, 2021.

[42] T. Xiao et al., "Adversarial Learning of Privacy-Preserving and Task-Oriented Representations," *AAAI*, vol. 34, no. 07, 2020.