TEADAL

D5.2 TRUSTWORTHY DATA LAKES FEDERATION SECOND RELEASE REPORT

Revision: v.1.0

Work package	WP 5
Task	Task 5.1, 5.2, 5.3
Due date	31/08/2024
Submission date	31/08/2024
Deliverable lead	Technische Universität Berlin (TUB)
Version	1.0
Authors	Sebastian Werner (TUB) Fenando Castillo (TUB) Sepideh Masoudi (TUB) Eduardo Brito (CYB) Patrizio Germani (ALMAVIVA) Andrea Falconi (Martel) Pierluigi Plebani (POLIMI)
Reviewers	Andrea Falconi (Martel)

WWW.TEADAL.EU

Grant Agreement No.: 101070186
Call: HORIZON-CL4-2021-DATA-01

Topic: HORIZON-CL4-2021-DATA-01-01
Type of action: HORIZON-RIA

	Boris Sedlak (TUW) Cynthia Marcelino (TUW) Pierluigi Plebani (POLIMI)
Abstract	This document presents the second iteration of the TEADAL Trust Plane, an evidence-based framework aimed at enhancing trust in federated data exchanges across cloud continuum. Building upon the first iteration of the TEADAL project, this version addresses key challenges in establishing trust by implementing mechanisms that provide verifiable evidence on the identity of actors, services and federated data products and offers ways to make the interaction of these observable and verifiable. In addition, this iteration presents a methodology (TrustOps) to build this evidence not only during execution but already at development time. Thus, with this document we lay the foundation for the final iteration of the TEADAL Trust Plane development and evaluation.
Keywords	TEDAL, Trustworthiness, Evidence Collection, Privacy Preserving Computations

Document Revision History

Version	Date	Description of change	List of contributor(s)
V0.1	24/05/2024	Document initial structure	Fernando Castillo (TUB), Sebastian Werner (TUB)
V0.5	30/06/2024	Contributions from all partners	See Authors
V0.6	06/08/2024	Cleaned version for final internal review	Sebastian Werner (TUB)
V1.0	30/08/2024	Integrated internal review and final cleanup	Fernando Castillo (TUB), Sebastian Werner (TUB)

DISCLAIMER

Funded by the European Union

Project funded by

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

Funded by the European Union (TEADAL, 101070186). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them. This work has received funding from the Swiss State Secretariat for Education, Research and Innovation (SERI).

COPYRIGHT NOTICE

© 2022 - 2025 TEADAL Consortium

Project funded by the European Commission in the Horizon Europe Programme		
Nature of the deliverable:	R	
Dissemination Level		
PU	*Public, fully open, e.g. web (Deliverables flagged as public will be automatically published in CORDIS project's page)*	✓
SEN	*Sensitive, limited under the conditions of the Grant Agreement*	
Classified R-UE/ EU-R	*EU RESTRICTED under the Commission Decision No2015/ 444*	
Classified C-UE/ EU-C	*EU CONFIDENTIAL under the Commission Decision No2015/ 444*	
Classified S-UE/ EU-S	*EU SECRET under the Commission Decision No2015/ 444*	

Funded by the European Union

EXECUTIVE SUMMARY

This document presents the second iteration of the TEADAL Trust Plane, an evidence-based framework aimed at enhancing trust in federated data exchanges across cloud continuum. Building upon the first iteration of the TEADAL project, this version addresses key challenges in establishing trust by implementing mechanisms that provide verifiable evidence on the identity of actors, services and federated data products and offers ways to make the interaction of these observable and verifiable.

The second iteration focuses on several technical advancements that provide additional choices of a TEADAL federation to enhance the trustworthiness. Specifically, we explored decentralized identities, which enable secure and verifiable user authentication, blockchain anchoring for immutable record-keeping, and privacy-preserving technologies that protect sensitive information while maintaining transparency.

Moreover, we also created a central method for archiving a tuneable level of evidence-based trust during the entire life-cylce of a data product through TrustOps, a methodology for continuously collecting verifiable evidence during development and operation. TrustOps leverages existing tools and technologies to ensure that software development and operation align with trust requirements. By incorporating mechanisms such as signed commits in version control systems and trusted execution environments, TrustOps provides a clear provenance for software artifacts, enabling stakeholders to trust the software's integrity and compliance with specified terms.

Both the TrustOps methodology and the technical solutions offered by TEADAL allow Data Lake Operators to a customizable trust-building mechanisms, enabling organizations to tailor their trust strategies according to specific needs, balancing cost, audit-data overhead, and verification complexity. This flexibility is essential for addressing the diverse requirements of federated data lakes in TEADAL.

In summary, the second iteration of the TEADAL Trust Plane significantly advances the framework's capabilities in establishing trust within federated data exchanges, providing a comprehensive and flexible approach to managing trust in decentralized environments.

Funded by the European Union

TABLE OF CONTENTS

Funded by the European Union

LIST OF FIGURES

Funded by the European Union

ABBREVIATION

BPMN	Business Process Model and Notation
FDP	Federated Data Product
DLO	Data Lake Operator
DLT	Distributed Ledger Technology
TNS	TEADAL Name Service
EVM	Ethereum Virtual Machine
GDPR	General Data Protection Regulation
IAMS	Intelligent Asset Management System
IDSA	International Data Spaces Association
JWT	JSON Web Token
MPC	Multi-Party Computation
OPA	Open Policy Agent
PEP	Policy Enforcement Point
sFDP	shared Federated Data Product
SNARK	Zero Knowledge Succinct Non-Interactive Arguments of Knowledge
TEE	Trusted Execution Environment
VC	Verifiable Credential
WP	Work Package
ZKP	Zero-Knowledge Proof

Funded by the European Union

1 INTRODUCTION

This document reports on the second iteration of the TEADAL evidence-based trust framework for federated data exchange known as the TEADAL Trust Plane. Building on the work of the first iteration [1], in the second iteration we further worked on the problem of trust (or lack thereof), which hinders data exchange in distributed, decentralised environments such as federated data lakes that we envision in TEADAL. We have investigated and developed most of the mechanisms necessary for parties wanting to exchange data to gain confidence that data are stored, handled and exchanged according to the terms they dictate. In the following we provide a first overview of these mechanisms and how they fit into the Trust Plane architecture, whereas the remainder of the document delves into the details. Also note that evaluation and validation of the software developed in this second iteration is still ongoing at the time of writing. However, a third and final iteration will follow, at the end of the project, to complete validation tasks and further refine the Trust Plane.

1.1 TRUST IN FEDERATED DATA LAKES

In the ever-evolving landscape of data management, data lakes have emerged as a pivotal solution for handling vast amounts of data originating from diverse sources such as edge devices, medical data networks, and industry 4.0 applications. TEADAL aims to facilitate trusted, mediationless, verifiable, and energy-efficient data exchange both in standalone and federated data lakes, particularly those deployed across the computing continuum. TEADAL leverages established data mesh architectures but augments them with the notion of a federated data product which can be stretched across data lakes, thus ensuring seamless integration of tools and data products across heterogeneous environments in the computing continuum. However, in practice federated data products alone are not enough to effectively enable distributed, decentralised data exchange: the parties involved in the exchange process need to trust the process is carried out according to stipulated terms.

In the context of software systems, trust can be defined in many ways [2] [3]. Notably, most definitions share common traits: trust is a complex interplay of beliefs, expectations, and reliability of software systems and in particular the reliance on the software's ability to meet specified requirements, even amid uncertainty. Moreover, trust is grounded in the expectation of consistent behaviour despite unpredictable conditions [4]. Consequently, TEADAL takes the stance that trust should be built on software behaviour evidence.

Hence, we have devised an evidence-based trust framework: the Trust Plane. Within this framework, all the parties involved in the data exchange process can examine evidence to verify that the software carrying out the process behaves as expected—here "software" refers both to the TEADAL data lake infrastructure and the federated data products which it hosts. A key advancement towards the implementation of our trust framework has been the generation and collection of verifiable evidence throughout the TEADAL infrastructure. The remainder of this document discusses a broad range of evidence types and details the mechanisms that enable verifiability and auditability in a decentralised system. Also detailed is the ongoing development and improvement of Trust Plane features and components such as Advocate, blockchain anchoring and the integration of privacy-preserving technologies and decentralized identity.

The Trust Plane defines several TEADAL data lake roles in relation to trust and evidence. As in the previous iteration, we differentiate between roles depending on who can view or produce evidence. The Data Lake Operator (DLO) is the central role for establishing a TEADAL data lake and enabling other stakeholders to exchange data. A DLO represents an organisation and, acting in its interests, sets up and configures the Trust Plane. All the evidence collected

Funded by the European Union

from the data lake is always linked back to this individual representing an organisation. Hence, a DLO is always able to review all the evidence produced by a data lake. The Trust Plane also includes the FDP Designer and Developer roles which are responsible for making a data product available to consumers. Both roles leverage the Trust Plane observability features to provide detailed evidence about how data products are deployed and consumed. Note that in most cases the Trust Plane can automatically observe FDPs, provided the Developer has suitably configured the FDP deployment. In cases where the automatic observability features are not enough, the Designer and Developer are responsible to interface their FDP with the observability subsystem, typically by equipping the FDP with a compatible tracing library. Unlike the DLO, the Designer and Developer can only review evidence related to their FDP. Finally, FDP Consumers can interact with and consume data from FDPs. The interaction between Consumers and FDPs is crucial to provide evidence-based trust and thus the Trust Plane prioritises FDP and Consumer observations over other kinds of observations. Besides reviewing evidence related to data product consumption, Consumers must also be able to verify that the data which they received from an FDP are in line with the sharing agreement between the data owner and consumer. Thus, once again, the evidence which the Trust Plane collects plays an important role in establishing trust in the system as it can be verified that the observed system behaviour matches expectations.

1.2 THE TEADAL TRUST PLANE

Evidence is, therefore, a fundamental pillar of how TEADAL facilitates trust, hence, the variety, veracity, and ubiquitousness of how and when we collect evidence is critical. Moreover, the authenticity of the collected evidence must be further fortified ensuring authentication and integrity in substantiating the evidence [5]. Moreover, while verifiability of evidence using verifiable credentials, facilitates systematic scrutiny of trust claims [6], authorization governs access rights [7], and distribution, as articulated by Dauterman et al. [8], enhances security and reliability guarantees.

With this in mind, we argue that evidence collection should ideally not only happen during the operation of software, but from the very beginning when software is being created and continuously during the software development cycle. Evidence in this scenario is generated in each phase of the cycle, so they have responsible persons and verifiable processes. Hence, we proposed a new additive approach called TrustOps [9], that can be used both during the development of TEADAL, and more crucially during the development of FDPs, to provide trust also in the way the software to share data is being built.

TrustOps is an approach for continuously collecting verifiable evidence in all phases of the software life cycle, relying on and combining already existing tools and trust-enhancing technologies to do so. For example, we can use already existing facilities to sign commits in git-based version control systems with the ability to run build and test steps in trusted execution environments to produce software that has a clear provenance on who built it and on how it was built and tested.

To enable this, we propose the Trustworthy Infrastructure (see Section 2), with the focus on evidence collection. The infrastructure to enable mechanisms to clearly identify individuals, e.g., using decentralised identities (see Section 2.1.1) as well as mechanisms to run privacy-preserving and verifiable pipelines (see Section 2.2) makes the crucial building blocks to produce this evidence. Moreover, the same systems can also be used to generate verifiable evidence to operate software built with such provenance records. Combined with reproducible

Funded by the European Union

and verifiable infrastructure code, such as the use of NIX[1] for building the TEADAL nodes, and verifiable runtime observations through tools like Advocate (see Section 2.1.3) further allows the additional evidence that can be used to produce a holistic record from creation to data delivery. Hence, this infrastructure can be used to provide users of TEADAL all the necessary mechanisms to build trustworthy data lakes.

Naturally, these mechanisms (see Section 3), and the level of evidence generated must be carefully calibrated with the needs of the respective data-sharing application, e.g., public accessible data may not need a complete record of who, and how data was delivered, whereas in the example ERT's internal data lake may want a partial record but need to keep that record confidential. Hence, the TEADAL trust plane is built as an extensible toolbox that can be used to build up these evidence records, but at the same time not all of it must be used, and other similar tools could be used based on the application needs.

Hence, in Section 3.3, we illustrate how the concepts of trust and evidence collection outlined so far can be applied to the TEADAL use cases. This includes ensuring trust at multiple levels—identity, data, process, and compliance—and providing clear evidence of interactions within the data exchange process. For instance, in the medical pilot case, TEADAL ensures that only accredited medical professionals in the EU can access specific data, adhering to the terms of the patient consent and privacy agreement. Moreover, we discuss how the different roles and stakeholders contribute to the collected evidence in each pilot case and how both external and internal audits can be made to increase the trust in the involved TEADAL data lakes.

[1] https://nixos.org/

Funded by the European Union

2 TRUSTWORTHY INFRASTRUCTURE

In the previous iteration of TEADAL [1], we introduced respective trust models and related evidence that we are collecting to enable all stakeholders to gain trust in the correct exchange of data. In this last report, we mainly focused on the collection of runtime evidence, generated through the execution of FDPs and the storage and distribution of this evidence, e.g., either anchored in a private blockchain or on a public one. In this second iteration, we focus on extending this evidence, by showcasing how we can authenticate different stakeholders through decentralized identifies, by showcasing how this evidence can be aggregated and verified continuously, Evidence Infrastructure as seen in Section 2.1, and how we further can enhance evidence by utilizing privacy-preserving infrastructure, such as TEEs, secure multi-party computation, and zero-knowledge proofs, Trust Enhancing Infrastructure as seen in Section 2.2.

2.1 EVIDENCE INFRASTRUCTURE

Verifiable evidence is the fundamental building block for the Trust Plane in TEADAL. For this evidence to be verifiable and usable, it must be based on a solid identity. This means that the identity of actors, organizations, and systems must be indisputable, and that actions taken by any of them can be clearly attributed to the responsible parties. Moreover, evidence should ideally be collected automatically and indiscriminately. Thus, in TEADALs evidence infrastructure, we consider multiple ways to add identity through Decentralized Identifiers (DIDs), shared and federated namespaces, and various methods to automatically and indiscriminately collect evidence with Advocate and Catalogue component integration into the Trust Plane.

2.1.1 Decentralized Identifiers

Decentralized Identifiers (DIDs) are a fundamental concept in the field of digital identity technology, mostly within the fields of Web3 and blockchain. They play a crucial role in realizing self-sovereign authentication and identification principles.

In this context, DIDs emerge as a promising solution to further enhance the capabilities of TrustOps. DIDs offer a digital identity model that, instead of relying on centralized entities, uses distributed technologies to ensure data integrity and verifiability. Every action associated with a DID can be recorded on a blockchain, providing an immutable record of activities and thus ensuring security.

Identity verification is distributed across a blockchain network, making the authentication and identification process not only more secure but also interoperable across various platforms. This is critical in an era where trust in traditional intermediaries is increasingly being questioned.

A DID is a Uniform Resource Identifier (URI) that represents a persistent subject, such as an individual, organization, or device. The specifications for DIDs are standardized by the World Wide Web Consortium (W3C).

The DID URI has a standardized structure that includes:

- **Scheme**: Always "did:" to indicate that it is a DID.

Funded by the European Union

- **Method**: The specific method implemented by the blockchain or technology on which the DID is created (e.g., eth for Ethereum, strk for Starknet).
- **Method-specific Identifier**: A unique identifier specific to the chosen method.

The creation and management of DIDs are conducted through DID registries, which can be implemented on various blockchain platforms. These registries ensure the immutability and security of identities due to the distributed nature and censorship resistance of blockchains.

The management and interaction between decentralized identities are realized through specific workflow processes, which include enrollment and authentication/mutual recognition between components. These workflows are essential to ensure the integrity and security of operations in TrustOps environments.

The registration of a new DID is the fundamental first step in the DID lifecycle, establishing a verifiable presence for an entity on the blockchain network. To register a new DID, the DID Issuer, who has the permission to generate DIDs, must perform a registration transaction with the DID Registry.

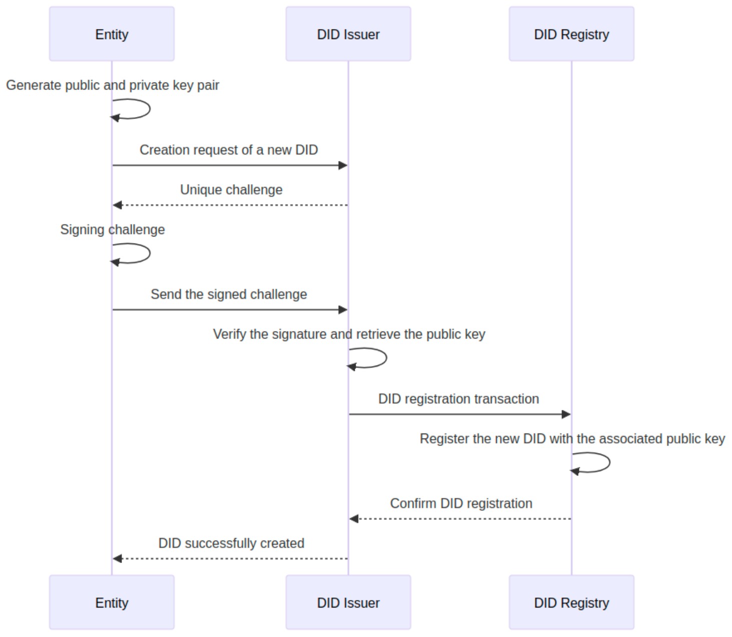
Entity
DID Issuer
DID Registry
Generate public and private key pair
Creation request of a new DID
Unique challenge
Signing challenge
Send the signed challenge
Verify the signature and retrieve the public key
DID registration transaction
Register the new DID with the associated public key
Confirm DID registration
DID successfully created
Entity
DID Issuer
DID Registry

FIGURE 1 THE DIAGRAM SHOWS THE CREATION OF A DID, WHERE AN ENTITY GENERATES KEY PAIRS, REQUESTS A DID FROM THE ISSUER, SIGNS A CHALLENGE, AND THE ISSUER REGISTERS THE DID IN THE REGISTRY.

In the context of TrustOps environments using blockchain technologies and Decentralized Identifiers (DIDs), how interactions among components are managed can have significant implications on system costs, security, and efficiency. We examine two main approaches, and a combination of them: using digital signatures for authorizations and blockchain transactions.

Funded by the European Union

Using Signatures for Authorizations in TrustOps:

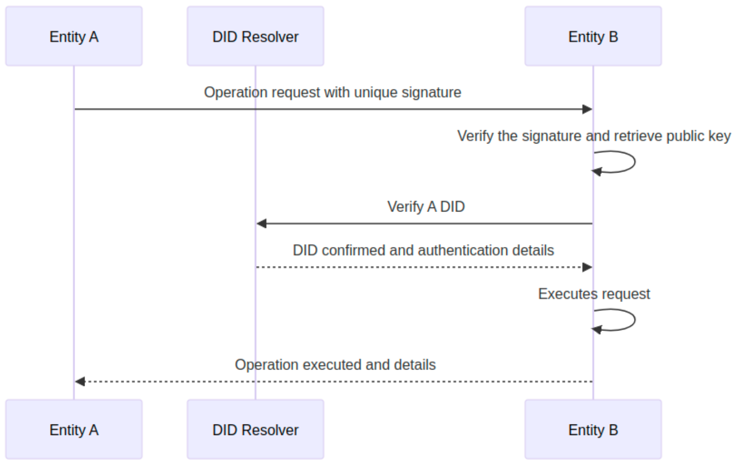
Entity A
DID Resolver
Entity B
Operation request with unique signature
Verify the signature and retrieve public key
Verify A DID
DID confirmed and authentication details
Executes request
Operation executed and details
Entity A
DID Resolver
Entity B

FIGURE 2 THE DIAGRAM SHOWS ENTITY A REQUESTING AN OPERATION FROM ENTITY B, WHICH VERIFIES A'S SIGNATURE AND DID VIA A DID RESOLVER, THEN EXECUTES THE REQUEST AND RETURNS THE RESULTS TO A.

Pros:

- **Low Cost**: Using digital signatures to verify the authenticity of a request is much less expensive than use a transaction on a blockchain. There are no transaction costs (gas fees), which can be significant on public networks like Ethereum.
- **Speed**: Signature-based verifications can be completed quickly as they do not require confirmation from the blockchain network consensus.

Cons:

- **Less Security**: Although digital signatures, between component interactions, provide a high level of security, they do not offer the same immutability and traceability as a transaction recorded on a blockchain.

Funded by the European Union

Using Blockchain Transactions for Authorizations in TrustOps:

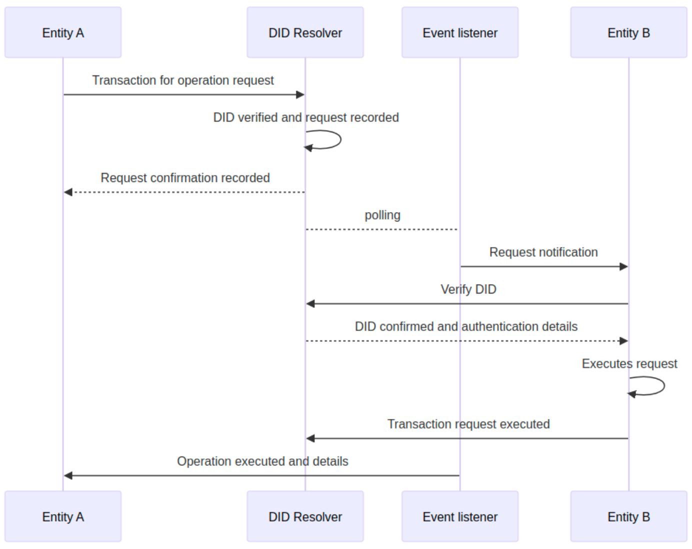
Entity A
DID Resolver
Event listener
Entity B
Transaction for operation request
DID verified and request recorded
Request confirmation recorded
polling
Request notification
Verify DID
DID confirmed and authentication details
Executes request
Transaction request executed
Operation executed and details

FIGURE 3 THE DIAGRAM SHOWS THE PROCESS WHERE ENTITY A REQUESTS AN OPERATION THROUGH A TRANSACTION. THE DID RESOLVER VERIFIES AND CONFIRMS THE REQUEST. THE EVENT LISTENER NOTIFIES ENTITY B, WHICH VERIFIES THE DID, EXECUTES THE REQUEST, AND SENDS THE RESULTS BACK TO ENTITY A.

Pros:

- **Immutability and Traceability**: Every transaction recorded on a blockchain is immutable and traceable. This provides strong auditability, which is particularly beneficial for applications that require a high degree of compliance and documentation.
- **High Security:** Integrity of blockchain transactions is protected by robust cryptographic mechanisms, offering superior security against fraud or manipulation attempts.

Cons:

- **Transaction Costs**: The transactions cost can be expensive, especially on popular and congested networks like Ethereum, where fees can vary significantly.
- **Confirmation Delays**: Transactions may take time to be confirmed by the network, which may not be ideal for operations requiring real-time responses.

Using A Hybrid Approach for Authorizations in TrustOps:

A hybrid approach could be considered where digital signatures are utilized for daily authorizations, maintaining system speed and efficiency, while blockchain transactions are employed only for significant operations requiring a high level of security and auditability.

This method not only meets compliance requirements but also ensures greater traceability and immutability in critical operations, reducing overall operational costs associated with blockchain transactions.

Funded by the European Union

Summary:

The integration of TrustOps with Decentralized Identities (DID) and digital signatures represents a revolution in the field of security and digital identity management. This innovative approach addresses the growing needs for security, privacy, and control in the digital era, overcoming the inefficiencies and vulnerabilities of traditional authentication systems.

DIDs offer a decentralized and interoperable solution that ensures data integrity and traceability. The ability to record every action on a blockchain creates an immutable and reliable trail that develop trust among all involved parties. This technology not only enhances security against external attacks but also ensures that operations are transparent and verifiable.

Moreover, the flexibility offered by the combination of digital signatures and blockchain transactions allows a balance between security and efficiency, enabling adaptable and scalable identity management. The ability to use DIDs across different platforms promotes interoperability and simplifies the integration of heterogeneous systems, consolidating the foundation for a more cohesive and reliable digital ecosystem.

2.1.2 Teadal Name Service

As we've explored, Decentralized Identifiers (DIDs) play a crucial role in establishing verifiable, self-sovereign digital identities within decentralized systems like TEADAL. Building upon this foundation of digital identity, we now turn our attention to two complementary technologies that further enhance our ability to generate and verify evidence: the Teadal Name Service (TNS) and Federation Smart Contracts.

The TEADAL Name Service (TNS) serves as a decentralized naming system within the distributed ledger technology (DLT) where TEADAL operates, with more functionalities analogously to classical Domain Name Service to do records management at a federation members level. This system provides a layer of abstraction, allowing human-readable names to be associated with blockchain addresses and resources. By doing so, TNS enhances the usability and accessibility of the TEADAL Federated Data Lakes for all federation members.

In conjunction with TNS, Federation Smart Contracts play a role in managing the governance and operations of the federation. These smart contracts are initialized with a pre-agreed set of members during the federation's setup phase. Concurrently, federation members register their domains and services in the TNS, creating a cohesive and transparent system of identity and resource management. The smart contracts enable federation members to register their public keys, facilitating secure message verification among participants. Moreover, TNS creates resolvable names for both Federated Data Products (FDPs) and Shared Federated Data Products (SFDPs), streamlining resource discovery and access within the ecosystem.

The Federation Smart Contract defines the initial membership and establishes protocols for accepting new members. Any modifications to the federation, such as registering a new FDP, trigger tamper-proof events in the DLT, creating an immutable record of these important actions. The smart contract emits specific events related to membership management, including MemberAdded, MemberApproved, and NewApprovalCount, providing a transparent and verifiable history of federation governance.

Together, TNS and Federation Smart Contracts contribute to the generation of verifiable evidence within TEADAL Federated Data Lakes. They enhance transparency in federation membership and S/FDP management, create an unchangeable record of actions and

Funded by the European Union

modifications, and simplify the verification of member identities and FDP locations. By ensuring that all interactions with these contracts are logged on the blockchain, they provide a traceable history of events, which is invaluable for auditing, dispute resolution, and maintaining trust among federation members.

This framework for evidence generation and verification is crucial in the TEADAL system. It not only supports the day-to-day operations of the federation but also builds a foundation of trust and accountability.

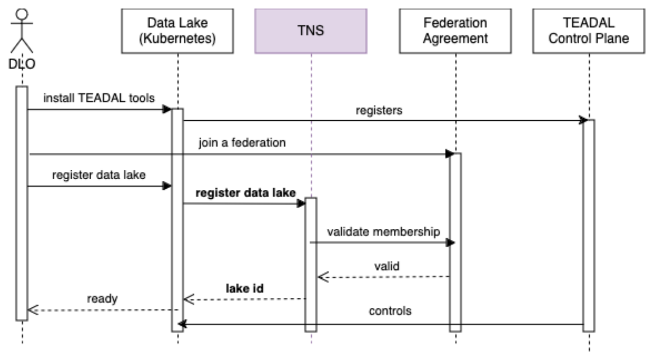

FIGURE 4 EVIDENCE DURING INSTALLATION OF TEADAL TOOLS.

As seen in Figure 4 DLO initializes the setup for the TEADAL tools, registering with the Control Plane and registering at an organizational level the domain in TNS. E.g. registering tub.teadal domain.

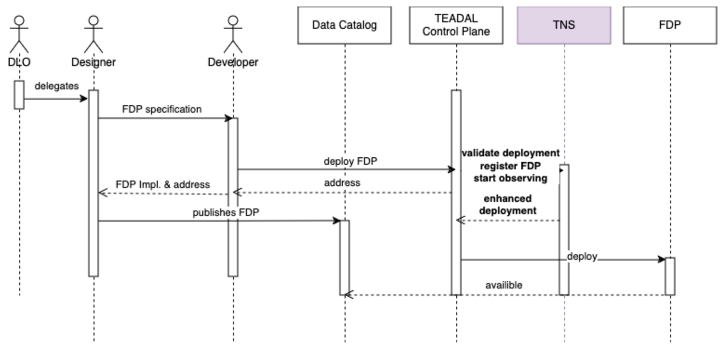

FIGURE 5 EVIDENCE DURING REGISTRATION OF FDP.

As seen in Figure 5 Developer registers a new FDP for consumption. E.g. fdp1.tub.teadal.

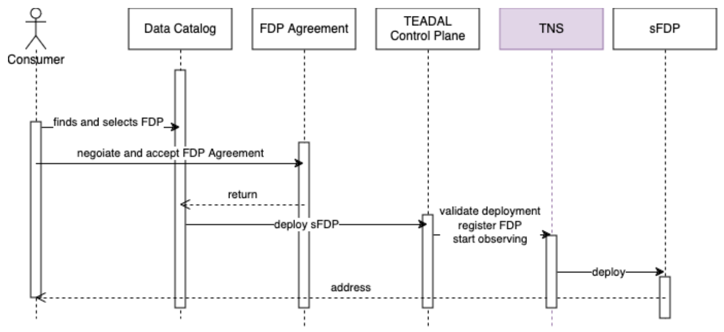

FIGURE 6 EVIDENCE DURING SFDP CREATION.

As seen in Figure 6 the Consumer selects existing FDP for consumption and asks for a new SFDP according to the FDP agreement. E.g. sfdp1.tub.teadal.

Funded by the European Union

2.1.3 Advocate

As outlined in D5.1 [1], Advocate is the core component ensuring trustworthiness and providing evidence for data lakes in TEADAL. The main mission of Advocate is to collect evidence related to FDP/sFDP interactions, store them in shared publicly accessible and immutable storage, and guarantee that the stored claims and evidence are tamper-proof by using a blockchain to anchor all published evidence claims. Each data lake requires one instance of Advocate to gather evidence and each piece of evidence for each data lake is signed with a TEADAL data lake ID, that is created during the bootstrapping process of Advocate.

Claims in Advocate are recorded from different sources in the TEADAL environment. For each data lake, claims can be issued by the Catalogue, through changes in the Kubernetes cluster, based on the use of TNS accesses or interactions with the FDP/sFDP based on tracing and monitoring mechanisms in the federation. To monitor the environment comprehensively, Advocate also offers a policy-based validation framework that can be used by the DLO. This integration involves storing policies related to the expected evidence that should be observed for a given FDP, so that evidence and access can be continuously checked against these expectations, which themselves create evidence claims. In this way, Advocate offers the capability to continuously and comprehensively generate verifiable evidence of the FDP and sFDP interactions. This capability, to collect and verify evidence and also to aggregate multiple observations into new evidence is used to enhance the trust of data sharing between different TEADAL data lakes. With each new expansion of the trust plain we are also expanding the plausibility verification of collected evidence. For example, with the use of DID’s for user identification we can also verify these DIDs and link them to the users that interact with FDPs.

As part of the trustworthy federation mechanisms of TEADAL, Advocate also plays a crucial role in various core functionalities, including identification, observable interaction, verification, and attestation. In continue we are explaining shortly how Advocate contributes in each of the core functionalities.

In the **identification** process, Advocate uses public/private keys to establish the identities of components within TEADAL. Advocate generates these keys and assigns them to components such as FDPs and catalogues in federated environments. For **observable interaction**, Advocate ensures that all interactions within TEADAL are recorded in immutable storage, making the data tamper-proof and traceable to the source using the components’ unique IDs. This is achieved by storing logs and traces securely as evidenced by Advocate. In the **verification and proving** functionality, Advocate is responsible for collecting claims and storing them in a claim repository. By recording application-level and platform-level claims and signing them with IDs specific to each actor, Advocate ensures that actions within the environment are verifiable later. Also as mentioned before, by recording policies and regulations and validating and verifying if activities in the environment were allied with defined policies in a specific time period, Advocate can improve the proving mechanism in TEADAL. Finally, Advocate contributes to the **attestation** functionalities by storing FDP policies and all related evidence in a transparent, shared infrastructure. This approach provides both consumers and providers with the means to facilitate and verify attestation. Moreover, the policy-based validation framework allows the DLO to define easy to understand claims that consumers and data owners can review when monitoring the data exchange process. Furthermore, by using evidence to trace DIDs the attestation is improved.

Advocate Architecture

Funded by the European Union

Advocate's ability to collect, combine, and store evidence from multiple sources makes it a crucial component in the TEADAL ecosystem. By ensuring that all claims are substantiated with telemetry data, Advocate provides a comprehensive view of the data lake's operations. This holistic approach allows for more effective monitoring, auditing, and compliance, helping organizations meet regulatory requirements and maintain trust with consumers of FDPs. In the organization’s services cluster, these claims range from those representing service deployments within that cluster, e.g., FDP creation. In Catalogue-level, claims are a report about consumer interactions, e.g., when creating a contract for consuming an FDP. These claims are stored along with tracing data from observation tools that substantiate them. Observation tools provide tracing logs and metrics, generating Platform-level evidence that Advocate collects, combines, and immutably stores to justify claims. Tracing logs are also used to provide evidence about data exchanges between services. Moreover, we aim to create reliable and trustworthy monitoring sources by enabling mTLS and signed deployment images for all critical infrastructure services.

As shown in **Error! Reference source not found.**7, the architecture of Advocate is designed to handle a diverse range of evidence sources and types, e.g., Jaeger and Kubernetes. By supporting various evidence sources from different providers and integrating seamlessly with existing infrastructure of data lakes, Advocate ensures that the integrity of evidence are maintained across all TEADAL environments and data lakes. The use of public and private keys for signing evidence adds an additional layer of authenticity, making it difficult for unauthorised parties to alter or forge claims alongside with using a blockchain for tamper proof record keeping.

In a federated environment like TEADAL, where every data lake has its own instance of Advocate running, we need to aggregate all the evidence and verifications from different instances across the data lakes. To achieve this, each Advocate instance in each data lake acts as a gateway and provides evidence to other data lakes. This way other Advocate instances can discover, verify and enrich observations and combine evidence to build up a comprehensive picture of FDP, sFDP data exchanges. Moreover, since advocate uses a shared storage (IPFS) and a public (or accessible) blockchain with smart contracts, other third-party implementations can interact and use the evidence created by Advocate across the TEADAL data lakes if needed.

Advocate Deployment Options

During Advocate deployment, both the DLO and the federation can configure Advocate in various ways to ensure compatibility with specific requirements for anchoring evidence. For storing evidence securely and enabling transparent sharing via blockchain, Advocate can connect to either a permissioned blockchain or a public one. During the testing phase, the DLO can utilize a local blockchain for testing purposes. The same flexibility applies to IPFS configuration; Advocate can integrate IPFS (public, private, embedded) using its open standard.

The DLO and data providers can determine which services within the federation can and should generate claims and designate which parts of the data lake Advocate should monitor simply by labelling these services in the data lake. The frequency of evidence generation and the polling interval for tracing data can be configured in Advocate to optimise cost-efficiency for different levels of auditing requirements.

The configurability of Advocate allows it to adapt to varying levels of auditability across different use cases. In a pilot case of evidence-based medicine, where health research institutions utilize data from a private data centre, the sources of evidence are hospitals and research institutions. The frequency of producing evidence is expected to be high to ensure trustworthiness, given the sensitivity and confidentiality of the data. To further enhance

Funded by the European Union

confidentiality for medical data, a private blockchain and private IPFS can be utilized. Similar configurations can be applied to industrial use cases, albeit with a lower frequency of evidence production. However, in regional planning, mobility, and smart viticulture pilot cases, the DLOs may prefer to use a public blockchain with a lower evidence frequency, as the shared data is mostly public and involves actors that are in a commercial or governmental relationship.

In conclusion, Advocate plays a vital role in ensuring the trustworthiness and integrity of evidence in data lakes in the TEADAL ecosystem. Its ability to integrate, store, and secure evidence from multiple sources, combined with its use of blockchain technology, makes it an indispensable tool for managing and verifying claims. By offering a comprehensive and tamper-proof solution, Advocate helps organizations to share their data products with their stakeholders and build trust with them.

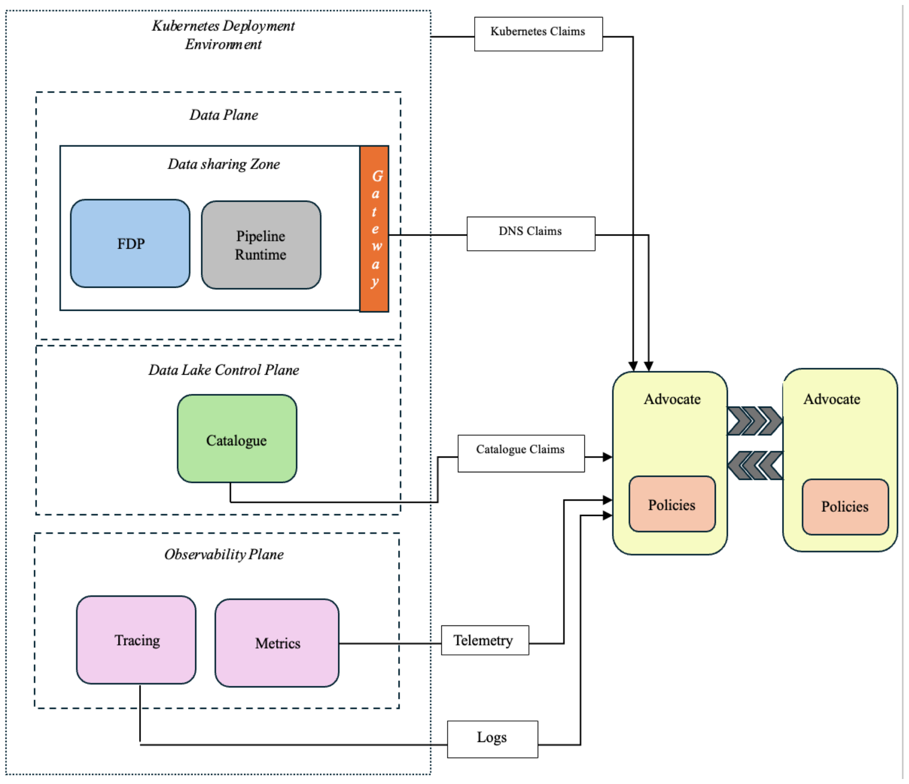

FIGURE 7 INTEGRATION OF ADVOCATE WITH DIFFERENT PLANES AT DIFFERENT LEVELS

2.1.4 Catalogue Transaction Observations

The integration between the Catalogue and Advocate components allows bridging the concept of data governance in a complex scenario such as inter and intra data lake governance with the concept of accountability and tracing of transactions. This integration enables comprehensive tracking of high-level and low-level operations occurring both within the data lake and across different organizations. The usage of BPMN as a way to model governance processes inside the data lake allows integrating the features of Advocate in an easily adaptable way.

The integration is achieved at the API level, leveraging the Service Task concept of BPMN (Business Process Model and Notation) in the processes for publication and access requests. These BPMN processes are orchestrated by the Catalogue component and are designed to interact with Advocate, a tool responsible for monitoring and auditing actions related to data governance. Specifically, the Catalogue component captures high-level events such as:

Funded by the European Union

- **Asset Publication:** events related to the approval or rejection of an asset (Dataset, Federated Data Product, Shared Federated Data Product) publication by the governance process responsible actor.
- **User-related Asset Actions:** events concerning the approval or rejection of access requests for a Dataset or Federated Data Product.

These events contribute ensuring that all actions are properly documented and that the data sharing processes complies with the established governance policies. In Figure 8 is depicted a diagram of the usage of the Service Tasks for sending events to Advocate.

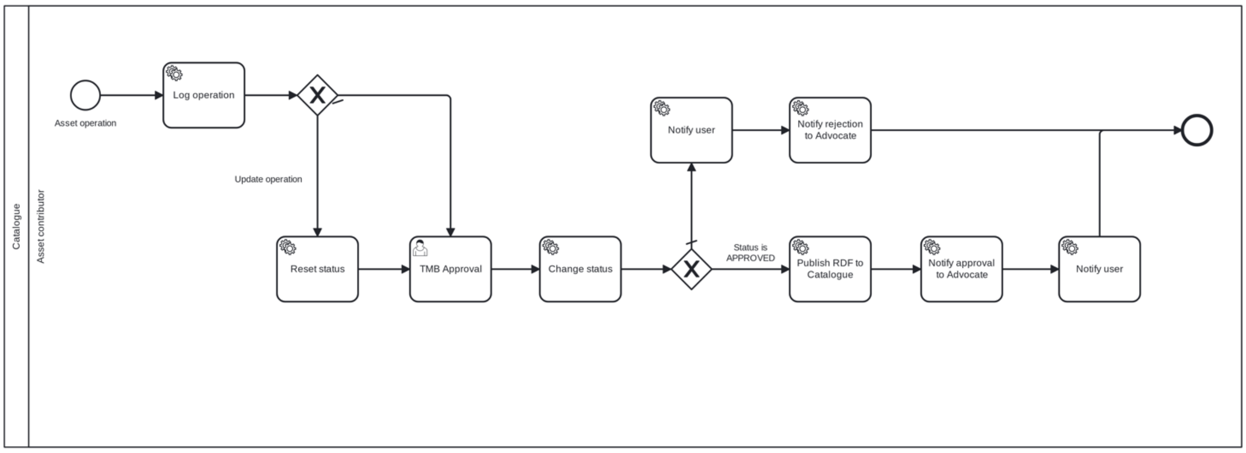

FIGURE 8 HE CATALOGUE EVENTS SENT TO ADVOCATE

Inserting the two Service Tasks block, whenever a digital asset is approved or rejected for publication onto the Catalogue a JSON event like the following is sent as seen in Figure 9, which would be sent to Advocate for anchoring.

```
{
      "claim": "{
                  'user': 'carenini',
                  'time': '2024-08-02 16:35:19.827447',
                  'action': 'publication_approval',
                  'object':
'https://kcong.cefriel.com:8087/assets/Federated%20data%20product/Rib
era%20Salud%20fdp-medicine', 'identifier': 2}",
      "signature":
"12fb6b5153b4e6fe3f4bd822b7e3e91ae9e28336a33a9414e6aaa6ac344f39f413f8
9427b8a372e0fa047cccb217c5cb0a068d78069b5141baac5a2b319b3d06",
      "service_id": "catalouge"
}
```

FIGURE 9 EXAMPLE OF A CATALOGUE EVENT SENT TO ADVOCATE AS A JSON CLAIM.

The current list of actions tracked via this mechanism is:

- publication_approval: the object has been approved for publication onto the Catalogue and will therefore be visible to other users;
- publication_denied: the publication of the object has been denied be the actor who is responsible for enforcing the governance of the data lake;
- access_grant_request: a user requested the right to access the resource on the Catalogue;
- access_grant_approved: a user requested the right to access the resource on the Catalogue and its owner granted it;

Funded by the European Union

- access_grant_denied: a user requested the right to access the resource on the Catalogue and its owner denied it

By implementing this integration, the system ensures that all data transactions are transparent, traceable, and compliant with regulatory and organizational policies. This setup not only fosters trust among participating entities but also enhances the overall security and accountability of data sharing activities.

2.2 TRUST ENHANCING INFRASTRUCTURE

In addition to the use of evidence-generating infrastructure and the necessary means to introduce and manage identity within TEADAL, we must also enable the users of TEADAL to use trust-enhancing infrastructure in situations where data needs to remain confidential while still being shared. In these cases, evidence alone is not sufficient, as mere claims of proper handling of data do not satisfy regulatory or contractual obligations. Therefore, we also consider the use and integration of privacy-preserving technologies into TEADAL's Trust Plane infrastructure, both for providing and processing the data and for sharing and distributing evidence between organisations. For this we rely on privacy preserving pipelines, that utilize different privacy preserving technologies within processing pipelines.

In the age of big data, safeguarding the integrity, confidentiality, and privacy of information is critically important. Traditional data pipelines, which handle the collection, processing, and analysis of vast amounts of data, often struggle with security and privacy issues. These challenges are particularly pronounced when it comes to distributing or scheduling tasks across various computational environments. Privacy-Preserving Data Pipelines aim to overcome these difficulties by incorporating advanced security and privacy measures into typical pipeline workflows. Key Privacy Enhancing Technologies (PETs) driving these solutions include Trusted Execution Environments (TEEs), Secure Multi-Party Computation (MPC), and Zero Knowledge Proofs (ZKP). TEEs provide a secure area within a processor, ensuring that sensitive data can be processed in isolation from the rest of the system, thus protecting it from unauthorised access and tampering. Secure Multi-Party Computation (MPC) allows multiple parties to jointly compute a function over their inputs while keeping those inputs private, enabling collaborative data processing without compromising confidentiality. Zero Knowledge Proofs (ZKP) enable one party to prove to another that a statement is true without revealing any information beyond the validity of the statement itself, ensuring robust verification processes without data exposure.

With the integration of these advanced technologies, Privacy-Preserving Data Pipelines can maintain the highest standards of data security and privacy, even in complex and distributed computational environments. The following sections describe the first investigative efforts of integrating such PETs into pipeline workflow engines, both from an architectural and trust-modelling perspective and from a prototyping stance. The maturity of the work developed here is mainly dependent on the advancements of the baseline technologies and practical frameworks deployed.

2.2.1 Trusted Execution Environments (TEEs)

Confidential Computing (CC) is a security and privacy-enhancing computational technique focused on protecting data in use. It is designed to address software, protocol, cryptographic, and basic physical and supply-chain attacks by performing computations in a hardware-based

Funded by the European Union

Trusted Execution Environment (TEE)[2]. TEEs are segregated areas of memory and CPU that are protected from the rest of the CPU using encryption, ensuring that any data inside the TEE cannot be read or tampered by any code outside that environment [10].

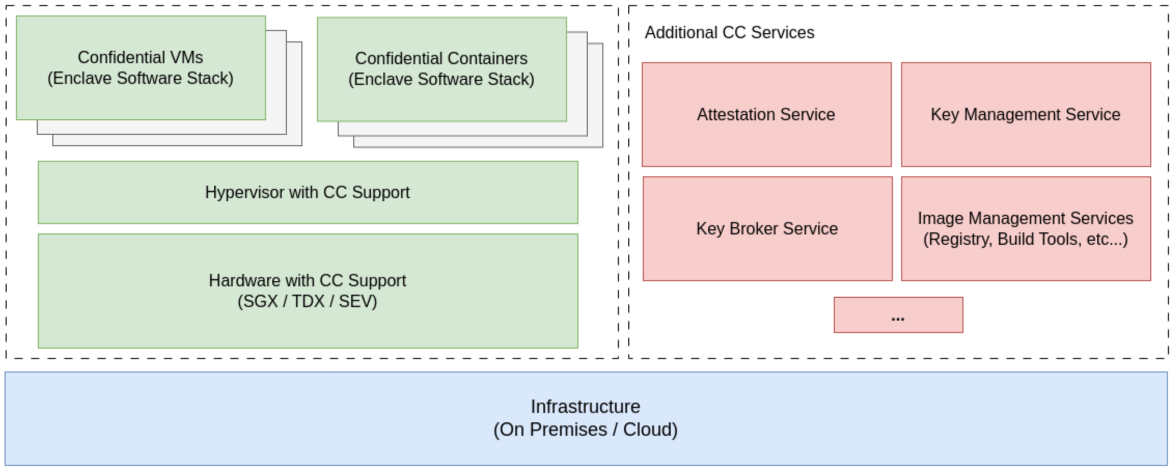

FIGURE 10 THE INFRASTRUCTURE REQUIREMENTS, FEATURES, AND ADDITIONAL SERVICES NEEDED FOR CC[7].

Figure 10 shows the building blocks of CC. The underlying infrastructure shall have compatible hardware and hypervisor support for running CC workloads. A general platform or environment for CC must also enable the integration with additional services to handle or automate the processes of attestation, key and identity management, and other components related to the lifecycle of the software applications.

The shift from process-based TEEs, like Intel Software Guard Extensions (SGX), to virtual machine-based TEEs has been significant, with technologies such as Intel Trust Domain Extensions (TDX), AMD Secure Encrypted Virtualization - Secure Nested Paging (SEV-SNP), and Arm Confidential Computing Architecture (Arm CCA) providing VM-based TEE environments for different computing platforms[3]. Intel Trusted Domain Extensions (Intel TDX) is a cutting-edge technology that provides robust isolation for virtual machines (VMs). It extends the hardware-level security features of Intel SGX (Software Guard Extensions) to virtualized environments, enabling the secure execution of sensitive workloads. Intel TDX creates a TEE within a VM, safeguarding the data and code from the host operating system and hypervisor[4]. This is particularly beneficial in cloud and multi-tenant environments, where trust boundaries are essential for data protection. By leveraging TEE VMs, data pipelines can securely process sensitive data in virtualized environments, maintaining confidentiality and integrity. These TEEs ensure that even if the underlying infrastructure is compromised, the sensitive computations remain protected, making it an ideal choice for integrating privacy-preserving features into data pipelines. In addition to Intel TDX, other TEE technologies such as AMD SEV (Secure Encrypted Virtualization) and Arm TrustZone, are also available across different cloud platforms and hosting services. AMD SEV provides similar memory encryption to protect VMs from unauthorised access[5], and Arm TrustZone, on the other hand, creates a secure area within the processor to execute sensitive code and handle sensitive data separately from the main operating system, providing a versatile solution for embedded systems and IoT devices [11]. These technologies enable the isolation of applications running in VMs, all without requiring refactoring of the software. This shift has made it possible to

[2] https://en.wikipedia.org/wiki/Confidential_computing
[3] https://community.intel.com/t5/Blogs/Products-and-Solutions/Security/Confidential-Computing-the-emerging-paradigm-for-protecting-data/post/1335003
[4] https://www.intel.com/content/www/us/en/content-details/783205/protecting-kubernetes-clusters-in-the-cloud-with-confidential-computing-and-intel-xeon-processors.html
[5] https://docs.edgeless.systems/constellation/overview/performance/

Funded by the European Union

protect data in use across a wide range of scenarios, from legacy VMs to modern containerized and serverless workloads, and has paved the way for the widespread adoption of CC in cloud-native environments[6].

Some key characteristics of CC[7] are [12]:

- **Data Confidentiality:** unauthorised entities are prevented from viewing data while it is in use within the TEE. The data remains encrypted and isolated within the TEE, ensuring that only authorised code can access and process it, safeguarding against unauthorised viewing or interception.
- **Data Integrity:** CC also ensures data integrity by protecting against unauthorised entities attempting to add, remove, or alter data while it is being processed within the TEE. The secure enclave provided by the TEE guarantees that the data remains unchanged and uncorrupted throughout its processing, maintaining the integrity and reliability of the information being operated on.
- **Code Integrity:** Additionally, CC safeguards code integrity by preventing unauthorised entities from tampering with or modifying the code executing within the TEE. The TEE environment ensures that only trusted and verified code can run securely, protecting against unauthorised modifications that could compromise the confidentiality and integrity of the data being processed.

These CC characteristics aid in safeguarding sensitive data throughout its lifecycle. When processing sensitive data, especially in scenarios where multiple actors or entities require access to computing resources, TEEs provide secure enclaves where data can be processed without exposure to external threats, ensuring confidentiality and integrity in shared or untrusted environments. The following is a list of the key events throughout the lifecycle of sensitive data and the requirements around the use of TEEs for handling such data:

- **Data Creation:** data is assumed to be local to an already trusted environment, therefore it's of the responsibility of the data owner to ensure the needed protection.
- **In Transit:** From the data owner to the TEE, the data should be transmitted in a protected, encrypted way, establishing secure communication channels that also allow to exchange identity information and attest to the authenticity of the TEE and its computational content.
- **During Processing:** Has the TEE been attested, data is protected via the underlying (hardware-based) guarantees of the TEE.
- **During Storage:** The application requirements dictate the need for data persistence and storage. If the same protection requirements apply to persisted data, throughout the execution of a given TEE-based workflow, data persisted outside of the TEE should be encrypted, and a key management process should be established to make use of such data within the TEE.
- **Retrieval of Processing Outputs:** Secure authentication and access control mechanisms to retrieve processed data from the TEE may be employed to ensure that only authorised individuals or systems can access the outputs. When designing TEE-based tasks, legal and analytical considerations should be given to the potential for leaking or inference of original data from the processing outputs.

[6] https://kubernetes.io/blog/2023/07/06/confidential-kubernetes/

[7] https://github.com/confidential-containers/confidential-containers/blob/main/architecture.md#cncf-confidential-containers

Funded by the European Union

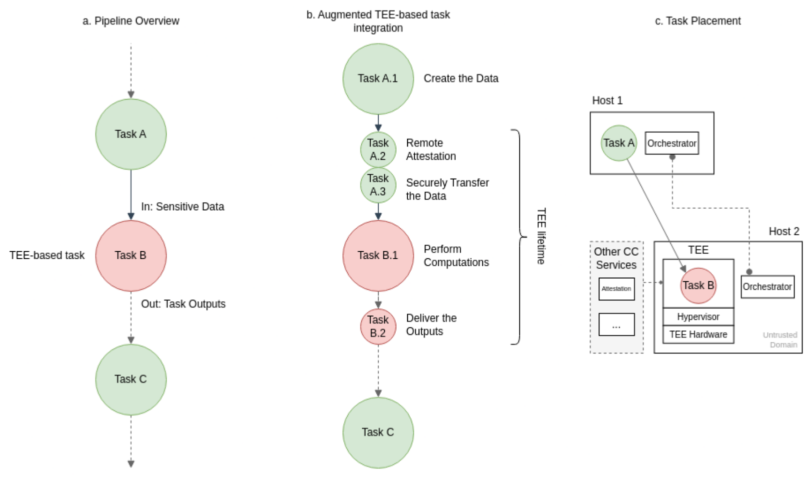

FIGURE 11 PRIVACY-PRESERVING DATA PIPELINES WITH TEE-BASED TASKS: AUGMENTED TASK INTEGRATION AND PLACEMENT.

The lifecycle of sensitive data needs to be mapped and the TEE-based tasks need to be integrated into the pipeline flow. Figure 11 illustrates the considerations over the integration of a TEE-based task into a pipeline. A typical pipeline flow may be augmented to encompass the sub-processes inherent to the lifecycle of a TEE, especially concerning the mandatory security steps prior to the computations. First, sensitive tasks are identified and isolated from non-sensitive tasks. These sensitive tasks are then scheduled to run within TEEs, ensuring that their execution remains confidential and secure. Non-sensitive tasks can continue to run on conventional processing units, optimising the overall workflow. This segregation is crucial for maintaining the privacy and security of sensitive data without totally compromising the performance of the data pipeline. After data is created, a TEE is deployed and attested to verify its authenticity and the integrity of its application code. If successful, data can be securely transferred, triggering the computations. The results of these secure computations are combined with the outcomes of non-sensitive tasks, eventually feeding and continuing the remaining data pipelines workflow and marking the end of the lifetime of the TEE.

The actual translation of these considerations into a concrete pipeline graph may reveal a more complex set of dependencies and coordination between the subtasks. Remains to establish the integration of this augmented flow into a distributed architecture. Given the use-case trust requirements and the hardware constraints of TEE-based tasks, their placement needs to match the environment that possesses compatible TEE hardware resources. Overall cross-host coordination is needed to orchestrate and manage the pipeline flow, ensuring its ordered execution.

To materialise these concepts, several open source projects establish the foundational blocks for TEE deployment, orchestration, and management. The most notorious ones are KubeVirt[8] and Kata Containers + Confidential Containers[9]. At the time of writing, integration with underlying CC hardware is being matured and developed, therefore, the showcase of the entire CC workflow and functionality is limited. We are specifically targeting Intel TDX platform and the orchestration of TEEs via Kubernetes. KubeVirt is a virtual machine management add-on for Kubernetes, extending its capabilities to enable orchestration and management of additional virtualization resource types, such as VMs. KubeVirt doesn't yet have a stable integration with TDX[10]. Kata Containers provide lightweight VMs that integrate seamlessly with Kubernetes, and Confidential Containers build on top of Kata Containers by leveraging its runtime for typical containerized workloads within Kubernetes, running inside protected VMs,

[8] https://github.com/kubevirt/kubevirt
[9] https://github.com/confidential-containers/confidential-containers
[10] https://github.com/intel/kubevirt-tdx

Funded by the European Union

with underlying support for different TEE hardware platforms. In TEADAL, we decided in favour of the Confidential Containers approach, due to its maturity and foreseen ease of the integration with the rest of the project architecture. However, Confidential Containers is also under active development, missing some core functionalities such as remote attestation. Nevertheless, prototypes and demonstrators are possible to execute and will be detailed further in this deliverable.

As detailed in D4.2, Kubernetes-based engines like Argo Workflows and Kubeflow are highlighted for orchestrating data pipelines effectively. Argo Workflows manages complex workflows using declarative YAML specifications, while Kubeflow supports end-to-end ML workflows and integrates with Argo Workflows for optimised pipeline execution. Both can leverage TEEs for secure execution of sensitive tasks, using Kubernetes node affinity and tolerations to manage workload placement on TEE-enabled nodes. Projects like KubeVirt, Kata Containers, and Confidential Containers employ these Kubernetes concepts to ensure sensitive computations run in secure environments, allowing to fulfil data confidentiality and integrity requirements in workflows[9].

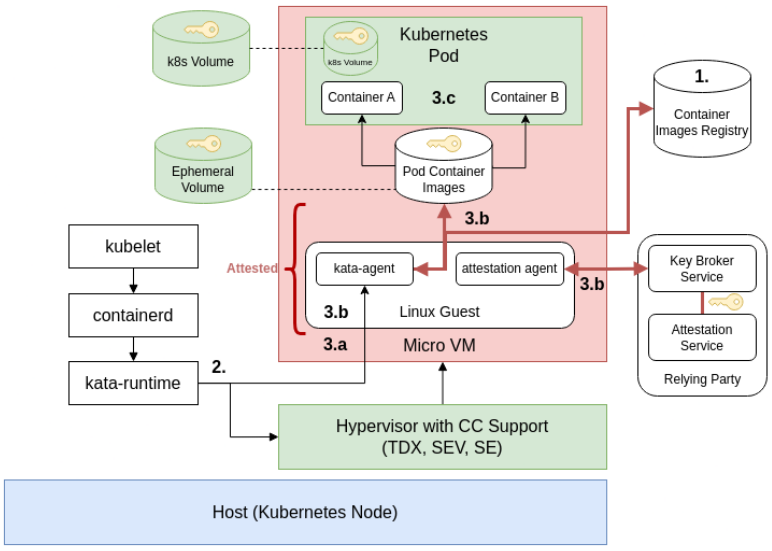

FIGURE 12 ARCHITECTURAL COMPONENTS OF A CC DEPLOYMENT, ACCORDING TO[7]. TEE VMS ARE LOADED WITH A CUSTOM RUNTIME WHICH INTEGRATES WITH THE HOST KUBERNETES ENGINE FOR THE DEPLOYMENT OF WORKLOADS WITHIN THE VM. THIS RUNTIME IS ALSO RESPONSIBLE FOR THE MEASUREMENT, ATTESTATION AND MANAGEMENT OF THE ENCRYPTED WORKLOADS AND RELATED DIGITAL IDENTITIES.

Figure 12 shows the architectural components of Confidential Containers, running a VM-based TEE setup. A typical workflows involves[11]:

1. Integration and Delivery: Application code is built, containerized, the images are signed/encrypted and published in a registry.
2. Deployment: The Kubernetes workloads are deployed in the host holding the CC compatible hardware and virtualization software.
3. Orchestration:
    a. The Confidential Containers runtime starts the VM TEE.
    b. The TEE agent performs remote attestation, during which keys are also exchanged in order to verify or decrypt the container images.

[11] https://github.com/confidential-containers/confidential-containers/blob/main/architecture.md#cncf-confidential-containers

Funded by the European Union

c. The TEE agent downloads the container images, verifies/decrypts them, and starts the workload.

In TEADAL, the data mesh and stretched data lakes architecture requires the extrapolation of this workflow into the federated data lake's domain, integrated with a federated data product computational pipeline. To show its feasibility, we demonstrate, further in this document, the integration of TEEs into a pilot case scenario. Further details on leveraging CC to enforce data contracts and the consequent orchestration of TEEs in data pipeline and workflow engines are present in D4.2.

Nevertheless, apart from enabling privacy-preserving data processing, aligning sensitive task placement and scheduling strategies with energy optimization efforts, such as utilising renewable energy-powered data centres, may also help achieve both data security and enhanced sustainability strategies in data processing operations. D3.2 details how one can leverage TEEs and Privacy-Preserving Data Pipelines to showcase new possibilities for optimising resource utilisation and energy consumption without compromising data privacy.

2.2.2 Secure Multi-Party Computation

Secure multi-party computation (MPC) consists of multiple parties that can jointly compute on private values so that only the computation result is revealed, hence it allows for the processing of data or the combination of multiple unshared data sources to be combined without the need to share them with a single party. There are different techniques for MPC, however, this introduction focuses only on the variant secret sharing. In secret sharing, a secret value x is split into multiple shares, which are used in the computations. The MPC protocols also output some shares and these can be used to reconstruct the final output. Sharemind MPC is a multi-node application server (usually 3 independent computation nodes are needed), that can be deployed in any public/private cloud, to run MPC protocols and perform secure data analysis. For the case of Sharemind MPC, introduced in the previous D5.1, additive secret sharing is used, where each secret x is split into three parts a, b, c so that $x = a + b + c \bmod 2$^k. Each of the shares a, b and c are uniformly random. The computations use interactive protocols between three parties and ensure that none of these parties learn anything besides the protocol output, as long as they correctly follow the protocol. This is suitable for parties with an incentive to cooperate and some mutual trust. Computations, including publishing an output, can only be executed if all computing parties execute them together.

Following a typical MPC architectural deployment, we can distribute the roles of secure computation participants as input parties, computing parties and result parties. The overall flow consists of input parties having some data that they secretly share between the computing parties. The computing parties then store the shares and execute the desired computations. They can send the shares of some value to the result party that can then reconstruct the value and therefore learn the output of the computation. For TEADAL, the data providers are the input parties and the consumers are the natural output parties. The computing parties can either be external service providers or the participants of the federation. The sets of input, computing and result parties can also overlap.

The current refactoring of Sharemind MPC aims to make it easier to orchestrate MPC tasks and integrate with cloud-native technologies such as Kubernetes. The MPC runtime is now extracted into a single independent executable task, a binary with input settings (input data - secret shared data, listening address, peer addresses, etc…), and outputting MPC results. The secret sharing process is done separately by the clients or users of the MPC service, having themselves the requirements for the secure processing of data.

Funded by the European Union

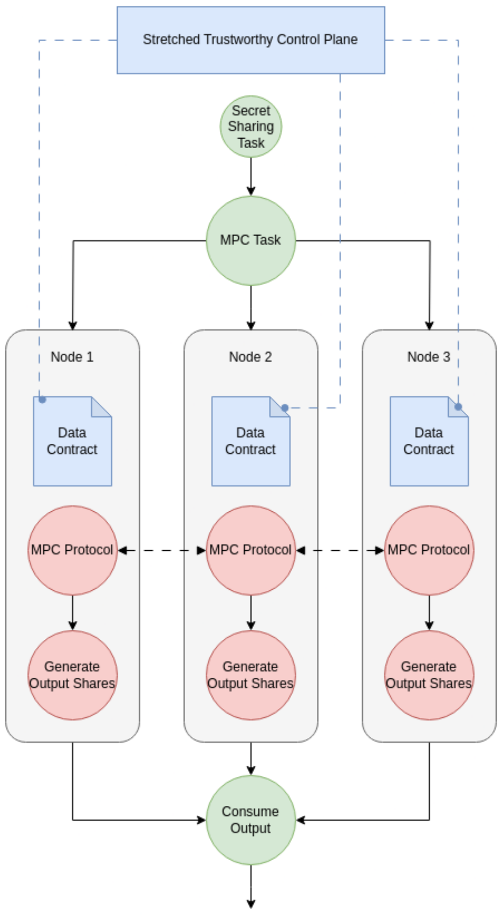

FIGURE 13 OVERVIEW OF MPC IN DATA PIPELINES.

Following Figure 13, the whole MPC process can then be partitioned into two sub tasks, first secret sharing the data, and second executing the actual MPC computations. Secret sharing can be done on the client side, while the MPC computations should be done in three different non-colluding nodes. The flow of execution of these two tasks should be synchronised and orchestrated, hopefully by a higher-level control plane, like Kubernetes, or in the case of TEADAL, by a stretched and trustworthy control plane. Defining the workloads in a stretched control plane should also allow for the placement, triggering, and coordination of these subtasks in different locations (the three non-colluding nodes), to fulfil MPC trust constraints. At the end, an MPC workload, or the data product corresponding to it, should have information about the data providers, the computing nodes, the access policies associated with the data inputs, computations and outputs.

Integrating MPC into data pipelines introduces several innovative possibilities for enhancing data privacy, security, and collaboration, such as secure data aggregations and the simplification in the adherence to strict privacy regulation. One key idea is to leverage MPC to enable secure data aggregation from multiple sources without compromising individual data privacy. In scenarios such as healthcare analytics or financial forecasting, where data from different entities must be analysed collectively while preserving confidentiality, MPC can facilitate collaborative computations. Each party contributes encrypted data shares, ensuring that no single entity has access to the complete dataset while still allowing for meaningful analysis and insights to be derived. Furthermore, MPC can be instrumental in addressing regulatory compliance challenges. Using MPC techniques within data pipelines, organisations can perform computations on sensitive data while adhering to strict data protection regulations. This approach minimises the risk of data breaches and regulatory fines associated with

Funded by the European Union

mishandling sensitive information. Another promising application of integrating MPC into data pipelines is in the realm of machine learning and artificial intelligence. MPC can enable collaborative model training across multiple organisations or jurisdictions without sharing raw training data. Instead, each party contributes encrypted data shares for model training, ensuring privacy while collectively improving model accuracy and robustness. This approach is particularly relevant in sectors where proprietary data and competitive advantage must be safeguarded, such as in pharmaceutical research or autonomous vehicle development. Moreover, integrating MPC into data pipelines can enhance data governance frameworks by providing a verifiable and auditable trail of data access and usage. MPC protocols ensure that computations are performed according to predefined rules and access policies, with results verified and reconciled across participating parties. This transparency helps build trust among data stakeholders and facilitates the adoption of data-driven decision-making processes.

Integrating an MPC runtime into Kubernetes-based data pipeline workflows like Argo Workflows or Kubeflow requires careful planning to manage computation agreements and synchronisation effectively. Before triggering an MPC pipeline task, it's essential to establish computation agreements among participating parties. These agreements define the scope of computations, data inputs, and expected outputs, ensuring all parties involved are aligned on the objectives and protocols of the MPC process. This step is crucial for maintaining trust and compliance, especially in regulated industries where data privacy and confidentiality are paramount. Once computation agreements are in place, synchronisation steps are implemented to coordinate MPC nodes within Kubernetes. This involves orchestrating the deployment and execution of MPC tasks across multiple computing nodes, typically configured as separate Kubernetes pods. Synchronisation ensures that all nodes begin computations simultaneously and follow the prescribed MPC protocols consistently. Kubernetes provides mechanisms for task scheduling, pod lifecycle management, and inter-pod communication, facilitating synchronisation of MPC nodes. Furthermore, synchronisation steps include monitoring and coordinating the flow of encrypted data shares between MPC nodes to ensure that computations proceed securely and in accordance with predefined policies. Kubernetes role-based access control (RBAC) capabilities can be leveraged to enforce security policies and restrict access to sensitive data shares only to authorised MPC nodes. This approach helps mitigate risks associated with unauthorised data access or tampering during the computation process. Future deliverables will demonstrate the integration of an MPC runtime into data pipeline workflow engines and Kubernetes deployments, similar to the TEE case, showcasing the execution of MPC tasks as part of Privacy-Preserving Data Pipelines.

2.2.3 Zero-Knowledge Proofs

Zero-Knowledge has two main foreseen applications in TEADAL: selective or hierarchical disclosure of private knowledge, and scalable validity of public knowledge. The first implies sensitive data in an adversarial scenario, the second implies publicly available data and cooperation towards scalability. These scenarios can also exist simultaneously.

The scenarios for the first case typically have a Prover who wants to prove a given fact to a Verifier, without disclosing information about why that fact is true. For instance, the Prover has in his possession private data X and wants to prove to the Verifier that a given statement about X is true, without having to share X with the Verifier. Here, trust on the authenticity, integrity, and provenance of X is established by some other means, but eventually also validated during the ZKP protocol (f.e. checking digital signatures). The scenarios for the second case have similar settings, except that the authenticity of the data is pre-established by the fact that data is publicly authenticated and accessible to both the Prover and the Verifier. In this case, the Verifier is motivated to reduce its verification effort and the Prover to eventually reduce its waiting time for verification, so the Prover is tasked to prove that a given statement about X is true, in a correct, succinct and scalable way, without the Verifier having to verify X directly, but

Funded by the European Union

rather a shorter proof. This is often accomplished with non-interactive proofs, that offload the computational efforts to the Prover and reduce the verification efforts for the Verifier.

In TEADAL, we can get inspired by the Regional Planning pilot case, as an example for mapping the first case of ZKP applicability. In this pilot case, we imagine one party holding private information with customer sensitive data. This party represents the Prover, with its willingness to fulfil regulatory constraints, without having to fully disclose its sensitive customer data. Another party, the Verifier, may hold publicly attestable data that can be used within the ZKP protocol, but most importantly, this party defines the proof statement and constraints, with eventual translation to regulatory requirements that it wants to see enforced. This scenario can be mapped to a typical case of selective disclosure or information obfuscation:

- Statement: “Prove that the territorial energy efficiency for the Tuscany region is (above/below) E%” or “Prove that a given building fulfils energy-efficiency certificate requirements R”
- Public data (available to both P and V): Tuscany region territorial mapping and borders, or building identity, and requirement constraints such as E or R.
- Private data (available only to P): Individual authenticated building energy data, for buildings within the Tuscany region.

Such selective disclosure protocols can also be used within Verifiable Credentials, for authentication and authorization purposes. In TEADAL, we may also showcase the integration of ZKPs with Verifiable Credentials, to limit to the minimum the amount of information shared by the VC Holder (Prover) to a potential Verifier. This scenario can also be mapped to a typical case of selective disclosure or information obfuscation:

- Statement: “Prove you queried data product FDP_X before timestamp T1, or by calling method M1, etc..” or “Prove all your interactions with data product FDP_X happened between timestamps T1 and T2”
- Public data (available to both P and V): Trusted Issuer identity, preferably queryable to eventually attest to issued credentials (depending on the statement), and requirement constraints.
- Private data (available only to P): The Verifiable Credential(s) attesting to some prior interaction(s) of the Prover with the system, issued by a trusted Issuer.

Such disclosure protocols can also be integrated with a state anchoring functionality, for external auditability purposes. In TEADAL, we can also have the integration of ZKPs with private blockchain state anchoring, to enable the validation of the private blockchain state, without revealing or at least limiting access to private state information. This scenario can also be mapped to a typical case of selective disclosure or information obfuscation, but eventually goes in line with the second case of ZKP applicability, for scaling blockchain networks. Here, again, the idea is to utilise proofs for off-chain computations (private blockchain) to reduce the on-chain computational efforts (public blockchain) to a verification process rather than an execution effort. Additionally, the zero-knowledge property of these verifiable computation schemes ensures the preservation of the confidentiality of information used in off-chain computations (private blockchain state). In the end, owners of a private blockchain can prove validity and adherence to specific requirements about the state of their private blockchain, without revealing all the information of the state. The authenticity and integrity of the data is ensured by the actual state anchoring in a public blockchain:

Funded by the European Union

- Statement: “Prove event E happened in the private blockchain” or “Prove evidence E exists in the private blockchain”, or more complex or more specific statements…
- Public data (available to both P and V): Data on the public blockchain, specifically the previously anchored states, and requirement constraints.
- Private data (available only to P): The private blockchain data.

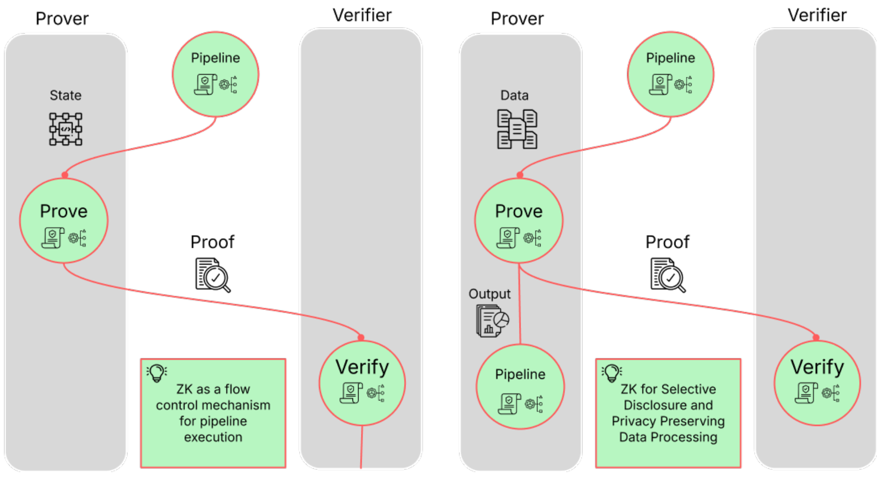
Prover
Verifier
State
Pipeline
Prove
Proof
Verify
ZK as a flow control mechanism for pipeline execution
Prover
Verifier
Data
Pipeline
Prove
Proof
Output
Pipeline
Verify
ZK for Selective Disclosure and Privacy Preserving Data Processing

FIGURE 14 TWO WAYS OF USING ZKPS IN PRIVACY-PRESERVING DATA PIPELINES.

Regarding integrating ZKPs into Privacy-Preserving Data Pipelines, these two ZKPs contexts are illustrated in Figure 14. ZKPs can effectively be integrated into typical pipeline workflows, acknowledging their particularity as both a flow control mechanism for pipeline execution and for selective disclosure and privacy-preserving data processing.

In the first context, ZK as a flow control mechanism, a Prover first generates a proof of the current state, which may be a public or shared state among a federation of pipeline operators, or even an instance of Verifiable Credentials to be authorised by a verification and access control step. This proof is then verified by a Verifier before allowing the pipeline to proceed with its execution. This ensures that each step of the pipeline is executed with validated data, enhancing the overall security and integrity of the workflow. In the second context, ZK for selective disclosure and privacy-preserving data processing, a Prover generates a proof based on the processed data. This proof is verified by a Verifier to ensure the computations were performed correctly without revealing the actual data. This method allows sensitive computations to be securely processed and verified, protecting the confidentiality of the data throughout the pipeline.

To exemplify ZKPs into Privacy-Preserving Data Pipelines, we demonstrate an example prototype, which can be easily adapted and may have wide applications and integrations with several different scenarios, ranging from data processing for the Regional Planning pilot case, to privacy-preserving environmental monitoring and compliance, explored in D3.2, to eventually turn into a TrustOps automation plane, introduced in this document. The prototype of ZK Monitoring Pipelines for continuous SLA monitoring represents an application of ZKPs within telemetry and monitoring processes. This approach aims to address the dual challenges of maintaining data integrity and confidentiality while ensuring compliance with Service Level Agreements (SLAs) in real-time operational environments.

Funded by the European Union

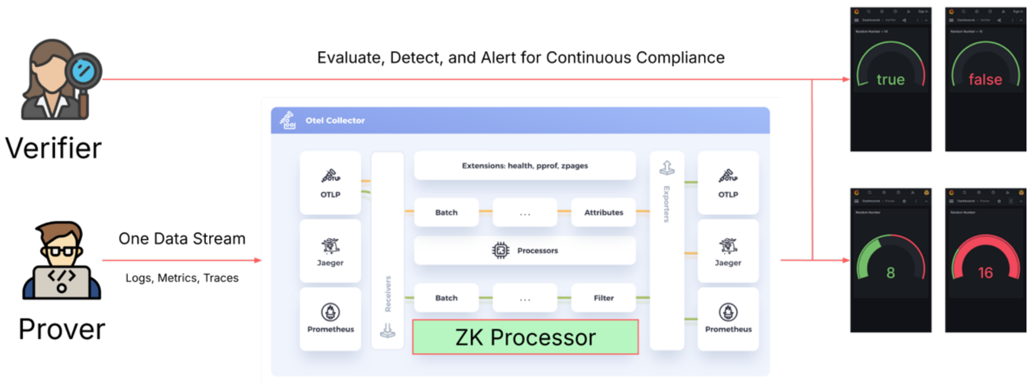

FIGURE 15 ZK MONITORING PIPELINE OVERVIEW.

At its core, the system begins with developers integrating telemetry instrumentation into their codebases to capture essential metrics, logs, and traces critical for ongoing application monitoring. These metrics serve as the foundation for defining SLA monitoring requirements, which are then translated into ZK proof statements. This initial setup ensures that both the prover (data provider) and verifier (SLA enforcer) agree upon the terms and conditions under which compliance will be measured and verified.

Deployment of the monitoring infrastructure involves distinct responsibilities on both sides:

- The prover focuses on continuous telemetry collection using tools like OpenTelemetry, feeding this data into dedicated pipelines. Telemetry processing infrastructure is deployed and custom ZK Coprocessors are integrated, f.e. within the OpenTelemetry Collector service, in order to process telemetry data to generate ZK proofs, continuously exporting them to the verifier domain.
- The verifier side stores and continuously verifies the ZK proofs, which are designed to validate SLA compliance without exposing the prover's sensitive information, a critical feature for preserving data privacy.

Visualisation and verification mechanisms are tailored to each party's needs:

- Developers and internal stakeholders on the prover side can utilise standard and typical monitoring software, like Grafana dashboards integrated with Prometheus and other tools, to visualise telemetry data and monitor application performance.
- Verifiers can leverage custom ZK software, like a tailored Grafana ZK verification plugin connected to a data source storing the ZK proofs, to load, verify and visualise ZK proof results. This setup allows them to independently verify SLA compliance while maintaining confidentiality of the original telemetry data.

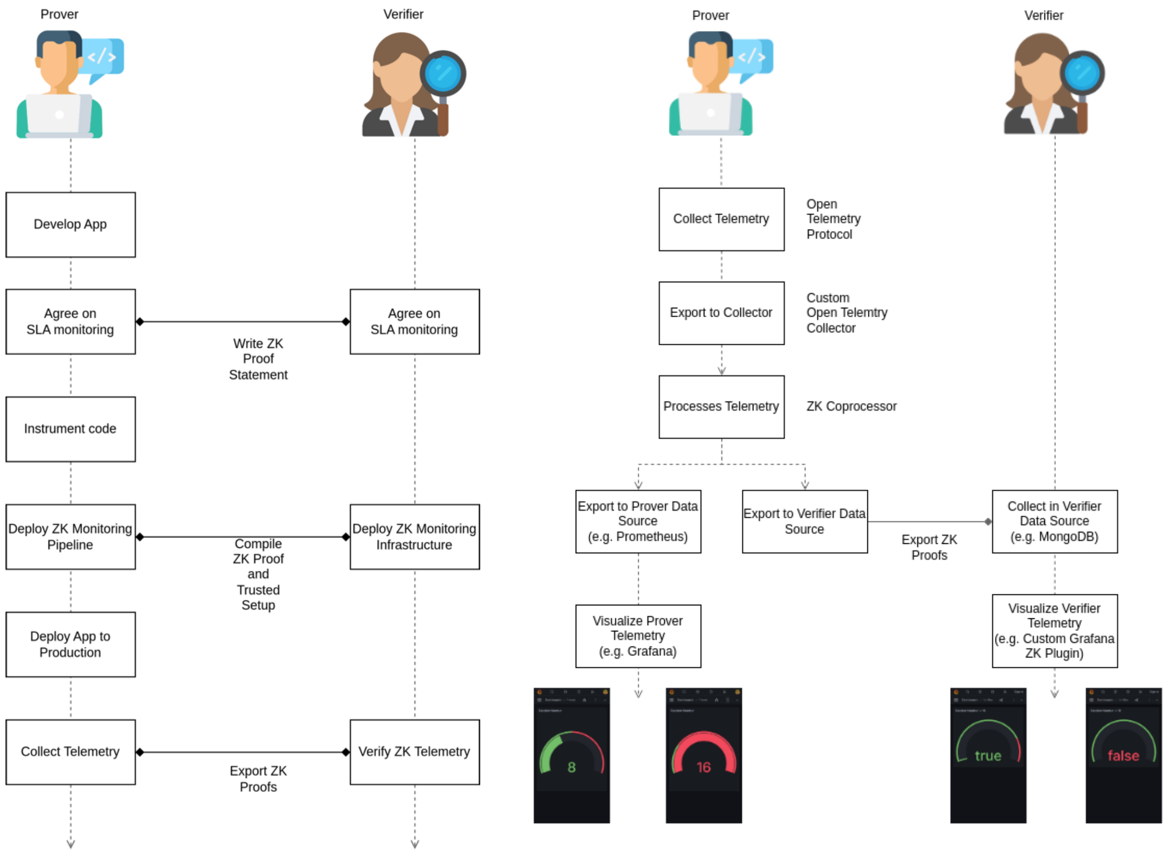
Prover
Verifier
Develop App
Agree on SLA monitoring
Write ZK Proof Statement
Agree on SLA monitoring
Instrument code
Deploy ZK Monitoring Pipeline
Compile ZK Proof and Trusted Setup
Deploy ZK Monitoring Infrastructure
Deploy App to Production
Collect Telemetry
Export ZK Proofs
Verify ZK Telemetry
Prover
Verifier
Collect Telemetry
Open Telemetry Protocol
Export to Collector
Custom Open Telemtry Collector
Processes Telemetry
ZK Coprocessor
Export to Prover Data Source (e.g. Prometheus)
Export to Verifier Data Source
Export ZK Proofs
Collect in Verifier Data Source (e.g. MongoDB)
Visualize Prover Telemetry (e.g. Grafana)
Visualize Verifier Telemetry (e.g. Custom Grafana ZK Plugin)
8
16
true
false

FIGURE 16 ZK MONITORING PIPELINE WORKFLOW AND COMPONENTS.

Illustrated in Figure 15 and Figure 16, the practical prototype to validate this concept demonstrates a method to verify that a Gauge value remains below a specified threshold using ZKPs. The goal was to allow a Prover to view the actual Gauge value while the Verifier only receives a proof that the value is less than a pre-agreed threshold. The demo implementation included a Flask web application integrated with OpenTelemetry for instrumentation. A custom ZK-SNARK OpenTelemetry processor, utilising ZoKrates toolkit for ZKPs[12], processes telemetry data to generate proofs. Prometheus is employed on the Prover side to monitor and store metrics related to the Gauge value. On the Verifier side, MongoDB or MinIO is utilised to store ZK proofs. Additionally, a custom ZK Verifier Grafana Plugin has been developed to query these proofs from the data store and perform verification. This setup ensures that Verifiers can validate compliance with the threshold without accessing sensitive information directly. Metrics are generated with each request made to the Flask application, providing continuous data for monitoring and proof generation. This prototype serves as a foundational demonstration of how ZKPs can be integrated into telemetry and monitoring pipelines to enhance data privacy and ensure compliance with specified thresholds or conditions.

Potential use cases for this technology span industries where stringent data privacy regulations intersect with the need for transparent SLA monitoring. For example, in healthcare, systems can ensure continuous monitoring of patient data while adhering to strict privacy laws. In financial services, it can support continuous compliance with regulations such as GDPR without compromising the confidentiality of transactional information. In the sustainability domain, as explored in D3.2, the technology may facilitate real-time monitoring of environmental metrics and sustainability targets while safeguarding the confidentiality of sensitive operational data. This capability is crucial for organisations and regulators aiming to

[12] https://zokrates.github.io/

Funded by the European Union

track and verify adherence to environmental regulations and commitments, such as carbon emissions reductions or sustainable resource management practices. The integration of ZKPs into monitoring pipelines represents a significant advancement in ensuring secure and auditable SLA compliance monitoring. By enabling verifiers to independently validate compliance without direct access to sensitive data, the systems can promote trust and transparency between stakeholders, essential for maintaining operational integrity in modern data-driven ecosystems.

Funded by the European Union

3 TRUSTWORTHY MECHANISM

In the previous deliverable (5.1) [1], we defined four categories of core functionalities designed to establish trust between the stakeholders in teadal TEADAL. In TEADAL, we specifically, differentiate between identifying, observing, verifying, and attesting mechanisms, that enable the stakeholders to gain trust in TEADAL and the data-sharing process. Within these functionalities, we differentiate between core and advanced mechanisms. The core mechanisms should always be available within the TEADAL data lakes while the advanced mechanisms are use-case dependent. In the following, we will describe these mechanisms and also illustrate different variants on how we can configure them in TEADAL based on the use-case needs. Moreover, we will discuss how these mechanisms provide relevant evidence to build onto the evidence-based trust approach we follow in TEADAL.

3.1 CORE MECHANISMS

Firstly, for identifying mechanisms we need to identify several actors and components to produce accurate, verifiable, and authenticated evidence about the interaction of the two.

The first mechanisms, e.g., to identify actors, is different depending on the role and type of the actors. For Data Consumers, we rely on mechanisms agreed on by the federation, e.g., a common Keycloak or a set of trusted O-Auth providers. Similarly, for accessing and interacting with the Catalogue, and therefore for designing and implementing FDPs, we use identity provides, defined by each organization. For data lake operators (DLO), we require an identity that can be verified by any member of the federation, hence, here we rely on a blockchain-based identity, either provided through the DID approach we outlined in this deliverable or through other compatible means.

The identity of components is also divided, we differentiate between data lake identity, FDP/SFDP identity, and infrastructure-service identities. The data lake identity, vital for building up the layers of evidence, is facilitated through Advocate's bootstrapping process. During startup, Advocate first verifies that the DLO's keypair is valid, and a member of a TEADAL Federation before generating a data lake identity in the form of a node claim, which is referenced as a `lake_id`. During the startup phase, Advocate will also need the right to deploy a smart contract to the respective blockchain used in the federation. The node claim and any other evidence claim generated by Advocate is anchored in this smart contract and stored in a federation-wide accessible IPFS cluster. For FDP/SFDP identity, the Catalogue provides unique names of these artifacts that can be used to later identify them. Infrastructure-Services, e.g., the Catalogue, Keycloak, Jaeger, and any other part of the TEADAL node, vital for its operation can also store a unique keypair and certificate within Advocate, to be later identified.

Unique to FDP/SFDP identity is that we also identify deployed components during runtime. During its runtime, Advocate intercepts the Kubernetes API for new pods and jobs. When a new pod starts, Advocate injects the `lake_id` into the pod's environment variables and generates a new claim for this service, storing it in IPFS. This claim includes the cluster's `lake_id` and information about the created deployment. The resulting hash of the claim is also attached as an environment variable to the pod, referred to as the `entity_id`. All services allowing for API interactions must log these environment variables as part of their output, such as when storing traces in Jaeger.

Besides this identity information, we also provide mechanisms to observe and attest to changes in a TEADAL federation. Specifically, we enable the recording of crucial interactions,

Funded by the European Union

with the Catalogue, deployed FDP pods, and use-case-specific interactions. For this, we provide two approaches, infrastructure observation and application-driven reporting.

For infrastructure observation, Advocate is used. It can be configured to interact and consume different evidence sources, such as Jaeger, Kubernetes, or the open policy agent. From each evidence source, records are periodically collected and enriched with meta-data, such as linking the entity_id to the deployments that caused an observation. For example, relevant Jaeger traces are compiled into interaction claims, stored in IPFS, and, if necessary, in the document store. These collected observations can be consumed through some of the advanced mechanisms offered by TEADAL.

For application-driven reporting, each FDP developer or other TEADAL infrastructure services can choose to also deploy a document store to anchor evidence saved in IPFS as verifiable credential claims. They can use the lake_id and entity_id attached by Advocate, to identify themself thereby creating a valid evidence chain that can be validated later. Alternatively, Advocate also provides a REST API, that can be used in combination with pre-shared keypairs to delegate the evidence generation.

3.2 ADVANCED MECHANISMS

All this so collected evidence, combined with the identity and layers of attestation (i.e., through signatures from the DLO, and signatures from the FDP developer keys) we can enable some advanced mechanisms, notably, the attestation and verification of interactions.

At the most basic level, Advocate provides facilities to periodically process collected evidence and evaluate policy definitions against them. For example, we can check if all JWT used to access an SFDP are from an organization that also requested to access this SFDP originally, from the JWT payload. These periodic checks can themself be published as claims and evidence and provided to data consumers and data providers.

As discussed in Section 2.2, there are use cases for using PETs aiming to have Privacy-Preserving Pipelines in TEADAL. As a core component of providing evidence and trustworthiness for federated data lakes, these PETs can also be used to aggregate and verify evidence, or the verification of the correct usage can be attested to give data consumers and providers additional confidence. An additional feature that we aim to explore further is the use of ZKP and the integration into Advocate to read the evidence and claims data in a privacy-preserved way while still verifying policy adherence. Hence, we can offer ways to attest that the TEADAL data lake was operated correctly and shared data as agreed without the need to share all evidence information publicly. This, we only need to reveal evidence data in the event of disputes, thus offering opportunities to store far less data anchored on the blockchain.

Two additional advanced mechanisms we aim to explore in the last iteration of the trust plane are the announcement of changes to policies, FDPs, or data in a transparent and verifiable way and to enable simplified addressing and resolving of identified entities in TEADAL. For the announcement mechanism, we aim to explore different blockchain-based approaches to communicate state changes as well as interaction between multiple Advocate instances. For the addressing part, we aim to extend the TNS approach and automate the addressing of identified infrastructure services and TEADAL components. This way, changes in access, are also recorded as evidence and can also include additional layers of verification.

3.3 EXEMPLARY SCENARIOS

Funded by the European Union

In this section, we apply the illustrated mechanisms and trustworthy infrastructure components on examples inspired by the use cases of TEADAL. Note that these examples are to illustrate the possibilities, the concrete implementation of the use cases is still ongoing.

3.3.1 Regional Planning

The REGIONAL PLANNING FOR ENVIRONMENTAL SUSTAINABILITY pilot project aims to integrate environmental monitoring and energy consumption sensors of buildings deployed by BOX2M, a private enterprise, with building energy records managed by the public authority of the Tuscany Region, Italy (RT). This so combined seeks to reconstruct static and dynamic energy profiles for both public and private structures and delineate local energy efficiency and air quality trends. For this, the analysis will also incorporate open data concerning weather conditions. One aim of this so combined dataset is to evaluate whether certification documents align with actual energy consumption but could also be used to measure the effectiveness of energy and building policies over time.

A primary challenge of such a combined dataset is ensuring data privacy and confidentiality. Here the data shared from these two TEADAL data lakes (Box2m and RT) must implement strict policies, e.g., ensuring that normal users cannot access raw data. To further protect data subjects, RT wants to ensure that all sensor data is aggregated up to a minimum threshold of three units. Moreover, RT needs to strictly prohibit users from viewing confidential data related to buildings and plants stored in the RT data lake. Such data confidentiality and privacy controls need to be supported by TEADAL.

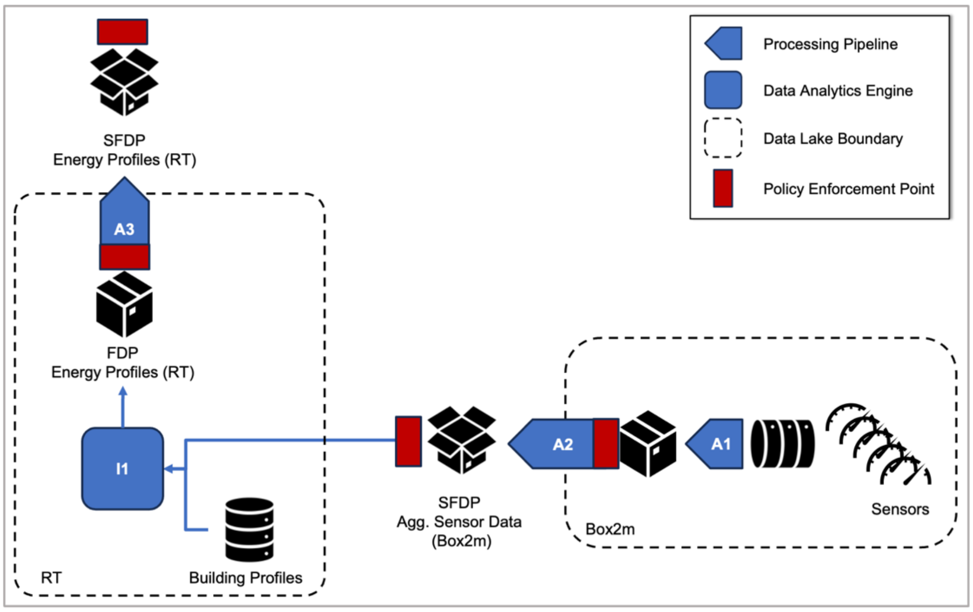

FIGURE 17 SIMPLIFIED VIEW ON THE REGIONAL PLANNING FOR ENVIRONMENTAL SUSTAINABILITY PILOT ARCHITECTURE.

Considering the simplified infrastructure in Figure 17, we can see several crucial processing pipelines and policy enforcement points (PEP) that implement these controls. For example, A2 can already perform the windowed aggregation to ensure that at least 3 sensors from a zip code area are always aggregated together before exposing this data to an FDP, this way the data RT is consuming from Box2m's FDP is always in accordance to the contract policies they negotiated with RT. Moreover, Box2m may enforce further aggregations and data cleaning before exposing their data product in A1. Once the data is ingested into RT's Data Analytics Engine (I1), the data gets combined with housing profiles in the same zip code form the internal database. As RT is in control of this engine, they can ensure that no raw data is ever part of

Funded by the European Union

the result set. Lastly, RT is also able to further limit or aggregate the access to the data set by adding additional processing in A3. Moreover, throughout this data exchange, we see several PEP that are supported by TEADAL. These enforcement points will ensure that only authorized users can access these respective data sets and that only authorized processes can perform requests on FDP/SFDPs to get the data for further processing. Here, we also ensure any contractually required processing.

In this scenario, we have two Advocate instances that collect and later provide evidence of this data exchange. On the Box2m side, Advocate will be observing the containers responsible for performing A1 and A2, as well as the FDP itself. Especially, the enforcement actions on A2 and the SFDP will be part of Box2m's evidence record. But if required, Box2m can also add data provenance claims in the A2 step to later proof that they always combined three distinct data points without revealing these data points themselves, e.g., through hashing or Merkel trees. On the RT side, Advocate, again can collect observations about A3 and the FDP/SFDPs. If RT desires also actions from I1 can be included. Here, specifically, access to the FDP can be monitored to ensure that access is only ever granted to authorized personnel or if processed through A3. Later, the combined evidence from both data lakes can be collected and used to audit that all the privacy and confidentiality needs of the pilot are satisfied regularly.

3.3.2 Finance

To illustrate a TEE scenario for Privacy-Preserving Data Pipelines, we are inspired by a Financial use case[13]. The Financial use case has, among others, the following high-level regulatory constraints: sensitive data from Turkish citizens must not leave Turkey, and must not be stored or processed outside of Turkey, unless protected (which can be mapped into requirements, adapted from D2.2: Req. P5-Privacy01, P5-Policy01, P5-Mgmt03, P5-Mgmt04, P5-Mgmt05). The Financial pilot has computational resources available for the processing of their operational data outside of Turkey (e.g. Netherlands). Given that, sensitive data from Turkish customers should be protected and moved to a TEE hosted in the Netherlands. This environment should first be attested by Turkish authorities responsible for auditing such operations and, if the attestation is successful, the data can be transferred and workloads can be executed within the TEE. The outputs can then be consumed by further tasks in the pipeline, see Figure 18. The properties of TEEs guarantee data protection during processing. After processing, the outputs are moved to a secure storage and the TEE can be safely destroyed along with its memory footprint, leaving no traces of the original data and computations.

[13] This use case was initially developed with a pilot partner who is no longer part of the consortium.

Funded by the European Union

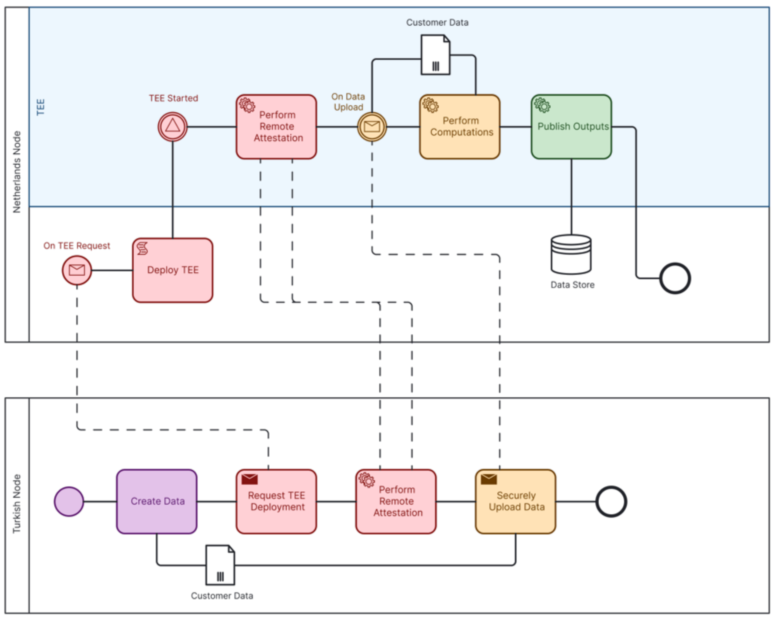

FIGURE 18 BPMN DIAGRAM ILLUSTRATING THE TEE PIPELINE EXECUTION PROCESS.

To also illustrate this in a technical way, we implemented a demonstrator. For the demonstrator, we designed a data analysis task to identify and highlight regions and customer characteristics with higher concentrations of high-risk behaviour, across the pilot jurisdictions or dataset features. The first step is to make data available, and so two datasets of customer activity data were synthesised, containing examples of PII from Turkish and Dutch customers. Following the pilot description, these datasets should feed, or be aggregated into a global KYC model, report, or analysis output. A risk scoring model was used to classify customers into different risk categories (e.g., Low, Medium, High), and customer risk data was hierarchically aggregated by geographical regions, age and other factors, to identify groups with higher concentrations of high-risk customers. These calculations were specified in Python, containerized, and executed inside the TEE, protecting the dataset of Turkish customers throughout the whole process. Figure 18 represents the simplified process flow of the demonstrator in BPMN notation.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: fp-deployment-qemu-tdx
spec:
  selector:
    matchLabels:
      app: geo-risk-score
  replicas: 1
  template:
    metadata:
      labels:
        app: geo-risk-score
    spec:
      runtimeClassName: kata-qemu-tdx
      containers:
      - name: geo-risk-score
        image: geo-risk-score:latest
```

FIGURE 19 KUBERNETES DEPLOYMENT USING KATA-BASED TEE.

Funded by the European Union

Figure 20 and Figure 21show an example of attestation evidence produced by the TEE and a verification report for the running example.

```
{
   "isvEnclaveQuote": "rH1qXy+sM6cdgFbcnWJta73c+jZT9hGrXr3M3PPXyB6F5CkY0TyTYeBkpYvjJb6K",
   "pseManifest": "uT1VmXbMgG2+hYQFbQHmn4KFbFtI5Km5PkvJglLhXwQ9LhD8v12Pv5FpIk1Bg2X5",
   "nonce": "aBcDeFgHiJkLmN"
}
```

FIGURE 20 ATTESTATION EVIDENCE PAYLOAD EXAMPLE, SERIALIZED TO JSON

```
{
   "id": "1234567890abcdef",
   "timestamp": "2024-07-30T12:34:56Z",
   "version": 4,
   "attestationType": 1,
   "isvEnclaveQuoteStatus": "OK",
   "isvEnclaveQuoteBody": "dGhpcyBpcyBhIGR1bW15IGVuY2xhdmUgcXVvdGU=",
   "revocationReason": null,
   "pseManifestStatus": "OK",
   "pseManifestHash": "dGhpcyBpcyBhIGR1bW15IHBoYXNo",
   "platformInfoBlob": "dGhpcyBpcyBhIGR1bW15IHBsYXRmb3JtIGluZm8=",
   "nonce": "abcdef123456",
   "epidPseudonym": "dGhpc2lzYWR1bW15ZXBpZHBzZXVkb255bQ==",
   "advisoryURL": "https://security-center.intel.com",
   "advisoryIDs": "INTEL-SA-00001,INTEL-SA-00002",
   "docIDs": "DOC-001,DOC-002",
   "tcbEvaluationDataNumber": "5"
}
```

FIGURE 21 ATTESTATION VERIFICATION REPORT EXAMPLE, SERIALISED TO A JSON STRING FORMAT

This demonstrator shows potential for the integration of TEEs in typical data pipelines, for executing confidential computations, and maintaining the integrity and privacy of sensitive data. In D4.2, we expand these concepts to explore the enforcement of data contracts between producers and consumers. TEE-based tasks can feature in more complex pipeline flows, playing a key role in task placement, for instance, allowing us to optimise energy consumption without having to trade off privacy for sustainability. By leveraging the secure execution environment provided by TEEs, sensitive operations can be performed securely on less trusted hardware, facilitating more efficient use of resources. This enables the execution of energy-intensive tasks in environments optimised for sustainability, such as renewable energy-powered data centres, without compromising the confidentiality of the data being processed. D3.2 explores these concepts of Privacy-Preserving Data Pipelines for energy-aware placement and scheduling in TEEs. While TEEs offer a robust solution for confidential computation during processing, ensuring responsible data handling post-processing is also crucial. Such frameworks must comply with regulations like GDPR, which grant individuals the right to request deletion of their personal data. Furthermore, solutions based on advanced applied cryptography, such as TEEs, may necessitate new formulations of trust management protocols, especially in the context of decentralized data spaces. Establishing clear accountability for dynamic data access, processing, and deletion becomes increasingly complex in such environments, requiring innovative approaches to ensure transparency and user control.

Funded by
the European Union

4 CONCLUSIONS

This deliverable concludes the second phase in building the TEADAL's trust architecture. Within this report, we solidified the trustworthy mechanism that TEADAL provides to data providers and consumers by expanding on the role of verifiable evidence. Towards this, we expanded on the trustworthy infrastructure, introducing Decentralized Identifies, the Teadal Name Service, Catalogue Transaction, advances in Advocate and the integration of Trust enriching infrastructure components.

We further outlined how these infrastructure components and mechanism apply in two selected exemplary scenarios form the TEADAL pilots. Showcasing, the choices each TEADAL federation can make in fine-tuning the trust-building mechanism to balance cost, audit-data-overhead and verification complexity.

Looking ahead, the next iteration involves a final refinement and evolution of this evidence-based [12] trustworthy architecture. Notably, we will conduct a cost and energy evaluation of different designs of the trustworthy infrastructure to guide adopters of the TEADAL technology. Moreover, we aim to validate the TrustOps concepts within a pilot scenario to show the power of having an end-to-end verifiable FDP.

Funded by the European Union

REFERENCES

[1] TEADAL Consortium, "D5.1 TRUSTWORTHY DATA LAKES FEDERATION FIRST RELEASE REPORT," 2023.

[2] D. Hühnlein, T. Frosch, J. Schwenk, C.-M. Piswanger, M. Sel, T. Hühnlein, T. Wich, D. Nemmert, R. Lottes and J. Somorovsky, "FutureTrust - Future Trust Services for Trustworthy Global Transactions," *Challenges in Cybersecurity and Privacy-the European Research Landscape,* pp. 285-301, 2022.

[3] R. C. Mayer, J. H. Davis and F. D. Schoorman, "An integrative model of organizational trust," *Academy of management review,* vol. 20, no. 3, pp. 709-734, 1995.

[4] F. Hou and S. Jansen, "A systematic literature review on trust in the software ecosystem," *Empirical Software Engineering,* vol. 28, no. 1, p. 8, 2023.

[5] J. Heiss, Trustworthy data provisioning in blockchain-based decentralized applications, TU Berlin, 2023.

[6] A. Delignat-Lavaud, C. Fournet, K. Vaswani, S. Clebsch, M. Riechert, M. Costa and M. Russinovich, "Why Should I Trust Your Code? Confidential computing enables users to authenticate code running in TEEs, but users also need evidence this code is trustworthy.," *Queue,* vol. 21, no. 4, pp. 94-122, 2023.

[7] P. Manuel, "A trust model of cloud computing based on Quality of Service," *Annals of Operations Research,* vol. 233, 2015.

[8] E. Dauterman, V. Fang, N. Crooks and P. R. Ada, "Reflections on trusting distributed trust," in *21st ACM Workshop on Hot Topics in Networks*, 2022.

[9] E. Brito, F. Castillo, P. Pullonen-Raudvere and S. Werner, "TrustOps: Continuously Building Trustworthy Software," in *28th International Conference on Enterprise Design, Operations and Computing*, 2024.

[10] T. Geppert, S. Deml, D. Sturzenegger and N. Ebert, "Trusted Execution Environments: Applications and Organizational Challenges," *Frontiers in Computer Science,* vol. 4, 2022.

[11] Z. Ning, J. Liao, F. Zhang and W. Shi, "Preliminary Study of Trusted Execution Environments on Heterogeneous Edge Platforms," in *IEEE/ACM Symposium on Edge Computing (SEC)*, 2018.

[12] Confidential Computing Consortium, "Confidential Computing: Hardware-Based Trusted Execution for Applications and Data," 2022.

Funded by the European Union