TEADAL

D5.1 TRUSTWORTHY DATA LAKES FEDERATION FIRST RELEASE REPORT

Revision: v.1.0

Work package	WP 5
Task	Task 5.1, 5.2, 5.3
Due date	30/11/2023
Submission date	30/11/2023
Deliverable lead	Technische Universität Berlin (TUB)
Version	1.0
Authors	Sebastian Werner (TUB) Fenando Castillo (TUB) Andre Ostrak (CYB) Eduardo Brito (CYB) Emanuele D'Agostini (ALMAVIVA) Sergio Sestili (ALMAVIVA) Andrea Falconi (Martel)

Grant Agreement No.: 101070186
Call: HORIZON-CL4-2021-DATA-01
Topic: HORIZON-CL4-2021-DATA-01-01
Type of action: HORIZON-RIA

Reviewers	Pierluigi Plebani (POLIMI) Emanuele D'Agostini (ALMAVIVA)
Abstract	This deliverable reports the technical and conceptual summary of the intial approach for the privacy preserving evidence-based trust architecture in federated data lakes in TEADAL. Describing the trust models, deployment models and implementations to collect verifiable evidence in cloud native infrastrcutres used in federated data lakes.
Keywords	TEDAL, Trustworthiness, Evidence Collection, Privacy Preserving Computations

Document Revision History

Version	Date	Description of change	List of contributor(s)
V0.1	03/07/2023	Draft ToC and layout	Sebastian Werner (TUB)
V0.9	19/11/2023	Internal Review Version	Sebastian Werner (TUB) Fenando Castillo (TUB) Andre Ostrak (CYB) Eduardo Brito (CYB) Emanuele D'Agostini (ALMAVIVA) Sergio Sestili (ALMAVIVA) Andrea Falconi (Martel)
V1.0	29/11/2023	Integration of Reviews	Sebastian Werner (TUB) Fenando Castillo (TUB) Sergio Sestili (ALMAVIVA) Emanuele D'Agostini (ALMAVIVA) Pierluigi Plebani (POLIMI)

DISCLAIMER

Funded by
the European Union

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.

COPYRIGHT NOTICE

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*	

* R: Document, report (excluding the periodic and final reports)

DEM: Demonstrator, pilot, prototype, plan designs

DEC: Websites, patents filing, press & media actions, videos, etc.

DATA: Data sets, microdata, etc.

DMP: Data management plan

ETHICS: Deliverables related to ethics issues.

SECURITY: Deliverables related to security issues

OTHER: Software, technical diagram, algorithms, models, etc.

Funded by the European Union

EXECUTIVE SUMMARY

This document describes how the TEADAL project addresses the trust challenge in federated data exchange. When sharing data across organisational boundaries, data providers have to trust data consumers to use data as agreed and that consumers can only access data in the way intended. Similarly, consumers must trust providers to provide data as promised, e.g., based on actual patients and real measurements, even if the shared data is aggregated and anonymised.

However, addressing this challenge solely technically and automatedly will create unaccepted limits in the data-sharing process. Therefore, the TEADAL project explores the concept of verifiable evidence. Specifically, we present a concept and the first architecture to collect independently verifiable provenance records of all steps in a federated data exchange process. This evidence can be used to prove to a consumer that the data is collected from original raw measurements, but it also gives a data provider means to prove that data is exchanged as designed.

Our architecture leverages several layers of technologies, such as distributed ledgers, smart contracts, privacy-preserving computations, such as zero-knowledge proofs, secure multi-party computation, and cloud-native technology platforms, such as Kubernetes and OpenTracing. The presented approach can combine these different technologies to generate this evidence across the entire lifecycle of a federated data product, from its creation and development to its consumption.

Unlike other approaches to create a trustworthy data exchange architecture, we aim to create evidence on an individual data request level. Thus, we will be able to attest which bytes individual users received.

We designed this architecture to fit the needs of the different data spaces explored by the TEADAL project. Depending on the needs of each pilot case, evidence can be collected and verified publicly, only in a network of organisations or internal to an organisation. For each, we discuss trade-offs and deployment models of the architecture and how the different technologies can and should be utilised.

While the core functionality of this architecture can be provided and integrated into the TEADAL infrastructure, continued development is needed to make the evidence collection more viable. This is the first of three deliverables that report on this continued development and the evaluation of this architecture in practice.

Funded by the European Union

TABLE OF CONTENTS

1 INTRODUCTION 9
2 TRUSTWORTHY ARCHITECTURE 11
2.1 Trust in TEADAL 11
2.1.1 Evidence in Roles 11
2.1.2 Evidence in Interactions 12
2.1.3 Evidence in the TEADAL Pilots 13
2.1.4 Trust Models 15
2.2 THE Evidence-BAsed Data Lake Architecture 16
2.2.1 TEADAL Control Plane 17
2.2.2 Federated Data Exchange Plane 17
2.2.3 Data Lake Control Plane 18
2.2.4 Observability Plane 19
2.2.5 Data Lake Trust Plane and Shared TEADAL Evidence Plane 19
2.3 Deployment Models 22
2.3.1 TEADALs Private Blockchain Models 23
3 TRUSTWORTHY FEDERATION MECHANISMS 26
3.1 Core Functionalities 26
3.2 Verifiable EVIDENCE Data 27
3.2.1 Evidence Attestation 27
3.2.2 Evidence Auditing 28
3.2.3 Evidence Collection Example 29
3.3 Verifiable Interactions 33
3.3.1 Tracking API Prototype 34
3.3.2 Integration with the TEADAL Catalogue 36
3.3.3 Evaluation in the context of the first iteration 37
3.4 Privacy-Preserving Computations 37
4 CONCLUSIONS 42

Funded by the European Union

LIST OF FIGURES

FIGURE 1 TEADAL HIGH-LEVEL DATA SHARING PROCESS FROM A TRUSTWORTHINESS PERSPECTIVE 12
FIGURE 2 OVERVIEW OF THE TEADAL ARCHITECTURE WITH A FOCUS ON THE EVIDENCE-BASED DATA LAKE RELEVANT FOR ENHANCING THE TRUSTWORTHINESS OF THE DATA EXCHANGE 16
FIGURE 3 THE FEDERATED DATA EXCHANGE PLANE 17
FIGURE 4 THE DATA LAKE CONTROL PLANE 18
FIGURE 5 THE OBSERVABILITY PLANE 19
FIGURE 6 THE DATA LAKE TRUST PLANE AND SHARED EVIDENCE PLANE 20
FIGURE 7 EVIDENCE LIFE-CYCLE 21
FIGURE 8 EXAMPLE OF AN ENS REGISTRY 21
FIGURE 9 PROTOTYPE PRIVATE BLOCKCHAIN NETWORK DEPLOYED INTO THE POLIMI DATACENTER 24
FIGURE 10 EVIDENCE ATTESTATION PROCESS 27
FIGURE 11 AUDIT PROCESS 28
FIGURE 12 USAGE OF OBSERVABILITY AS AN EVIDENCE PROVIDER 29
FIGURE 13 EXAMPLE OF TRACING INSTRUMENTATION IN FLASK (PYTHON) 30
FIGURE 14 ISTIO AUTHENTICATION CONFIGURATION EXAMPLE 30
FIGURE 15 AUTHORIZATION POLICY EXAMPLE 31
FIGURE 16 JSON EXAMPLE REPRESENTATION OF VERIFIABLE CREDENTIAL 32
FIGURE 17 EXAMPLE OF A FEDERATION CONTRACT IMPLEMENTED IN SOLIDITY 33
FIGURE 18 EXAMPLE OF A CLAIMS REGISTRY CONTRACT IMPLEMENTED IN SOLIDITY 33
FIGURE 19 EXAMPLE JSON PAYLOAD OF A RUNTIME OPERATION 35
FIGURE 20 EXAMPLE JSON PAYLOAD OF A LIFECYCLE OPERATION 35
FIGURE 21 EXAMPLE BPMN FOR SEND EVENT ABOUT FDP LIFECYCLE CHANGES 36
FIGURE 22 MULTI-PARTY COMPUTATION 37
FIGURE 23 ZERO-KNOWLEDGE PROOF 38
FIGURE 24 TEADAL PILOT MPC USE-CASE 39
FIGURE 25 TEADAL PILOT TEES USE-CASE 40

Funded by the European Union

LIST OF TABLES

TABLE 1 TEADAL ROLES RELEVANT FROM A TRUST PERSPECTIVE 11
TABLE 2 PILOT EVIDENCE REQUIREMENTS 14

Funded by
the European Union

ABBREVIATIONS

BPMN	Business Process Model and Notation
CDD	Customer Due Diligence
CWL	Common Workflow Language
FDP	Federated Data Product
DApp	Decentralized Application
DLO	Data Lake Operator
DLT	Distributed Ledger Technology
ENS	Ethereum Name Service
EVM	Ethereum Virtual Machine
GDPR	General Data Protection Regulation
HI	Hardware Isolation
IAMS	Intelligent Asset Management System
IdP	Identity Provider
IDSA	International Data Spaces Association
JWT	JSON Web Token
KYC	Know Your Customer
LDAP	Lightweight Directory Access Protocol
MPC	Multi-Party Computation
OPA	Open Policy Agent
PDP	Policy Decision Point
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

Data lakes[1-3] are a pivotal approach for managing massive amounts of data coming from diverse sources, e.g., IoT devices, social networks, and business applications. However, current solutions still suffer from several limitations and challenges. Among them, the single party ownership — locking valuable data behind organisational silos hindering the creation of data-driven ecosystems; the lack of proper data governance -- preventing the creation of trustworthy data flows across organisations and ensuring data sovereignty, a cloud-centric view — neglecting vast amounts of resources distributed across the cloud computing continuum (i.e., the resources located at the edge, fog, and cloud), and the lack of energy awareness -- as current solutions rarely consider sustainability and energy efficiency aspects. TEADAL is among many projects advancing data spaces by evaluating novel solutions to tackle these limitations and challenges[4].

In particular, TEADAL enables the creation of trusted, mediationless, verifiable, and energy-efficient data flows inside a data lake, across federated data lakes and specifically across stretched data lakes (i.e., deployed in the continuum). TEADAL's architecture is built on top of the concept of data mesh, expanding it into stretchable and federated data products, and leveraging many established tools to ensure easy integration of TEADAL tools in all environments.

Due to this focus, TEADAL must enable trust between data owners and data consumers across different organizations and across multiple network and device boundaries present in the compute continuum. Thus, TEADAL must enable this trustworthiness not only on *an identity level* — i.e., ensuring that the data consumer is who they claim to be — but also on *a data level* — i.e., ensuring that the data are delivered as agreed and expected. Moreover, TEADAL must enable trustworthiness on *a process level* — i.e., ensuring that the data are processed as agreed by all parties and on *a compliance level* — i.e., ensuring that all parties comply with the applicable laws and regulations. For example, taking the Evidence-based Medicine pilot case (see D2.1) into account, TEADAL must ensure that only accredited medical doctors working in hospitals in the EU can query the data space of MARINA, ensure that these doctors can only work with data that the patient has consented to process, aggregated using compliant privacy-preserving mechanisms and that the data are only used for the purpose of the research project.

TEADAL offers technical and process-oriented recommendations to address these different levels through the work packages laid out in the project plan. However, some risks and sources of misuse will remain and cannot, for now, be entirely solved by technical means alone. TEADAL aims to address these challenges first and foremost by providing evidence of every interaction within the data exchange process. Specifically, we build a trail of evidence that tracks how identities are obtained, used and communicated, how data are moved, processed and stored (once they are shared), what processes are used to provide data to consumers and where these processes are executed and how they are controlled and lastly by collecting metadata relevant to evaluate compliance. For example, in the medical pilot, TEADAL aims to provide clear evidence to point to the members of MARINA who installed TEADAL, the users who prepared the study to be shared, and all the software used to transform the data into a shareable format. Moreover, TEADAL can show who requested the data, who allowed a given user to interact with the data and where this user's data lake is located. Lastly, through a provided API the hospital staff can create evidence about consent and purpose of the study that led to the sharing of data.

In the following, we present the part of TEADAL's architecture related to collecting and providing that evidence and, thus, the parts that aim to increase the trust in data sharing between the members of a TEADAL federation. Towards that end, we first present the trust

Funded by the European Union

model and main components of the trust architecture in section 2 and the main mechanisms and approaches in section 3 before concluding this deliverable with an outlook of the next iteration of the TEADAL trust architecture. To see how the trust parts of the architecture are embedded within the entire TEADAL architecture, see D2.2.

Funded by the European Union

2 TRUSTWORTHY ARCHITECTURE

In this section, we present the model of trust we pressure in TEADAL, reviewing the critical roles, interactions, and expectations this model covers. Moreover, we present the first iteration of the evidence-based data lake architecture and deployment model used in TEADAL to build that trust.

2.1 TRUST IN TEADAL

Trust can be defined in many ways and is the subject of a wide variety of research fields [5,6], ranging from political, economic, social, and philosophical to computer science.

Common among the definitions in these fields is the understanding that trust is an individual opinion about a perceived risk or about a particular system[7,8]. Consequently, in computer science, we mainly focus on trustworthiness as a function of demonstrating a system's or user's ability to act as expected (evidence) and architecting a system or use of a system in such a way that unexpected behavior is minimised [9,10].

Thus, in TEADAL, we focus on trustworthiness by producing verifiable evidence accessible to all parties of the federated data lakes and by architecting TEADAL tools with the means to ensure that interactions adhere to the desired data product workflow. However, we will not be able to create a perfect system. Thus, one objective of TEADAL is to explore how well we can gather evidence of compliant behavior, and we aim to do so not only at the component level, as approaches such as GAIA-X [3] propose, but also consider the usage level, identifying each user that consumes shared data products.

2.1.1 Evidence in Roles

Accordingly, TEADAL distinguishes between multiple roles (see Table 1) that we need to account for when considering the data-sharing process. These roles have different impacts on the data-sharing process, may be identified differently (e.g., being part of a different organization or having different types of access to the data) and thus must be treated differently when considering evidence collection. However, it is crucial always to enable the connection of a role to a natural person to ensure accountability. Towards that end, TEADAL follows a multifaceted approach of combining established identity providers such as LDAP, self-sovereign identity schemas and verifiable identity tokens stored and managed using distributed ledger technologies[11]. Moreover, whenever possible, TEADAL aims to create a provenance record of how a specific human representative was assigned to a given role.

TABLE 1 TEADAL ROLES RELEVANT FROM A TRUST PERSPECTIVE

Name	Description
DLO - Data Lake Operator	Can install TEADAL tools, responsible for security and compliance of the installed environment.
Designer - FDP Designer	Designs policies, specifications and other metadata necessary for creating an FDP, including a description of offered services through this data product.
Developer - FDP Developer	Implements software to deliver data from a dataset as an FDP, defining code, deployment policies, and needed capabilities. Also responsible for ensuring compliance with the TEADAL guidelines and implementing and testing any security policies enforced through TEADAL tools.

Funded by the European Union

Provider - FDP Provider	A generic term indicating the “TEADALisation” whereby the FDP Design, FDP Developer, and Data Lake Operator cooperate to expose an FDP. Each is a natural person who represents an organization of the federation with the right to access and share data.
Consumer - FDP Consumer	Searches and selects FDPs for their organization and negotiates agreements. Also responsible for the development of the client-side code needed to interact with the FDP that is deployed into a TEADAL-compliant Data Lake. Must be at least one natural person that has the right to request data from other organizations.

2.1.2 Evidence in Interactions

At the core of TEADAL is the Federated Data Product (FDP), an extension of the data product approach known from the data mesh, which allows Providers to share data as self-describing, self-service data sets. Thus, a data product can be used as is without the need to ask for approval or schemata. In TEADAL, we have extended this concept to that of a federated data product, which adds several technical and data governance challenges as a data product can now be shared also outside an organization. Thus, reducing the ability of an organization to enforce common data access mechanisms.

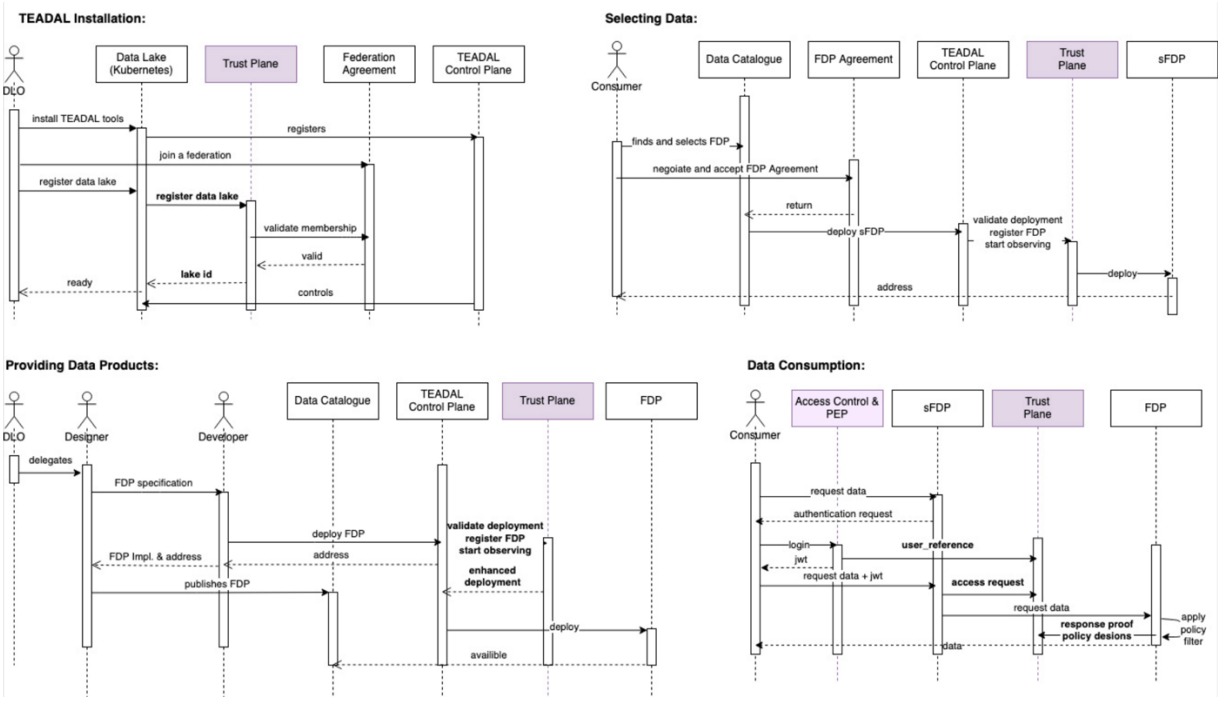

FIGURE 1 TEADAL HIGH-LEVEL DATA SHARING PROCESS FROM A TRUSTWORTHINESS PERSPECTIVE

From a trust perspective, the four phases in Figure 1 represent the most relevant interactions between the roles. The complete process view can be found in D2.2.

First, during the installation of TEADAL, we need to establish an immutable and verifiable identity for the TEADAL data lake being instantiated, including registering and verifying the DLO as a natural person who is ultimately responsible for enabling the sharing of their organisation's data. Authorising a DLO from an organization's perspective will require careful governance, which is out of the scope of the TEADAL project. Ultimately, we envision a federation agreement that contains the identities of the DLOs. Moreover, we assume that

Funded by the European Union

organizations want to join a federation to share or obtain data. Here, we assume for now that federations have procedures in place to grant access to new DLOs and new organizations.

Furthermore, we assume that any legal contracts needed to participate in a federation are referenced and accessible to any member participating in the federation, governing elements such as audit procedures or penalties for misbehaviour. Thus, adding a new DLO should also be accompanied by a link to new or changed legal documents. In this first iteration of the project, it is crucial to understand that we know the list of DLOs that will operate in a federation upfront. Thus, we will not discuss any means to join/leave a federation or change agreements on federation policies.

Secondly, during the provisioning of FDP, we allow for the delegation of rights from a DLO to a Designer. Thus, the Designer will be legally responsible for creating the shared data product and thus also responsible for describing the policies correctly to ensure that only authorised parties can access the data. This act of delegation must be recorded transparently and stored immutably. Moreover, we must ensure that the identity of the Designer and Developer (who is responsible for the technical implementation of these policies) is verifiable to cater to the legal aspects of data sharing.

Thirdly, the selection of an FDP by a Consumer also involves the identification of that Consumer in a verifiable way and the anchoring of the legal and technical contract to an agreement. The concrete implementation of these sharing agreements will be part of the next iteration of the project. However, we consider them to be both a legal framework to handle any contention and a technical description of the data product being shared. Sharing agreements will also include any resources that were provided by the involved parties at the time of agreeing on the data-sharing process. Once such an agreement is reached, TEADAL will instantiate the shared Federated Data Product (sFDP). The sFDP is the specific instantiation of the FDP tailored to the specific agreement between Provider and Consumer. For example, the sFDP can be set up only to share a partial view of the data set provided through the FDP. Here, the creation of the sFDP coincides with the beginning of evidence collection for the purpose of auditing. From this point onward, TEADAL will ensure that both the Consumer and Provider can review and verify any interaction related to this sFDP. This included any transformations that are performed to produce the specific view on the shared data set, e.g., if the sFDP is only set up to share aggregated and filtered data.

Lastly, during the consumption of data, all interactions, including the exact user credentials used to request data, must be recorded to ensure later audits and verification. In short, the Trust plane in TEADAL must be responsible for gathering and storing immutable evidence of the entire FDP life cycle, linking users back to the underlying agreements and natural persons who enabled these users access (either through implemented policies or direct delegation). At the same time, we must ensure that no personal identifiable data are published. Thus, one core principle of the trust plane is to collect evidence in a verifiable but confidential way utilizing practices of data minimization, zero-knowledge proofs and verifiable data aggregation.

2.1.3 Evidence in the TEADAL Pilots

Besides these implicit requirements based on the federated architecture of TEADAL, we must also consider the specifics of the different pilot cases of the project. During the elicitation of requirements of the pilot cases (see D2.1), we asked each pilot to fill out a questionnaire regarding specific requirements. Within this questionnaire, we also provided several questions aimed at evaluating specific needs for trustworthiness in data sharing.

We can broadly categorise the response into three needs: 1) Regulatory compliance, 2) verifiable enforcement and 3) transparency and accountability.

Funded by the European Union

Regarding regulatory compliance, most pilot participants stress the importance of following the due diligence requirements of the GDPR, which includes the need to handle personally identifiable data with great care—specifically, ensuring that the purpose of processing and consent are provided and adhered to. Here, the FDP architecture plays a crucial role, as it allows developers and data owners to integrate their knowledge about the provided data to implement the appropriate means to enforce consent collection and data minimisation [12,13]. TEADAL matching and optimisation process can, if configured, ensure that such critical data will not be processed outside EU data lakes or according to user-provided policies. Moreover, the generated evidence of the trust plane should exhibit evidence that any code required by the Provider was executed before sharing data, thus satisfying compliance requirements. Here, we are also looking into practices such as DevPrivOps [13] to aid Providers in satisfying regulatory compliance.

With regards to verifiable enforcement, each pilot case indicated a large and complex variety of conditions on when data are allowed to be shared. With the data mesh approach, these rules and policies can be integrated at design time. Policies can be bundled with an FDP and, in addition, can be verified during the sFDP execution to ensure compliance and enforcement of computational resource cost targets or other friction rules. From a trustworthiness perspective, we will ensure that any data owner can review these policy decisions at any time.

Lastly, driven by the GDPR but also other regulatory processes, pilot cases require a clear idea of what processes drive data sharing. This is the main objective of the trust plane, to collect comprehensive evidence of what is executed in response to every single data request. This way, a real-time record of any data exchange can be provided. Moreover, we aim to provide publicly verifiable records that can be validated not only by data providers but also by consumers. This way, authenticity requirements and delivery proofs can be used to also monetize data sharing agreements in the future.

TABLE 2 PILOT EVIDENCE REQUIREMENTS

	Pilot #1: Evidence-based Medicine	**Pilot #2:** Mobility	**Pilot #3:** Smart Viticulture	**Pilot #4:** Industry 4.0	**Pilot #5:** Shared Financial Data Governance	**Pilot #6:** Regional Planning for Environmental Sustainability
Federation Type	Open	Open	Bilateral	Internal	Internal	Bilateral
Continuum Variant	Multicloud	Multicloud	Cloud/ Edge	Multicloud	Multicloud	Cloud/ Edge
Personal Data Sharing	Yes	No	No	No	Yes	Yes
Confidential Business Data Sharing	Yes	No	Yes	Yes	Yes	Yes
Data Provenance Requirements	Yes	No	Yes	No	Yes	Yes
Reporting Requirements	Yes	Internal	Internal	No	Yes	Internal
Regulatory Compliance	GDPR	NAP, RTTI, MMTIS	Internal	Internal	GDPR, PSD2	GDPR

Funded by the European Union

Table 2 shows a detailed overview of the requirements (from a trust perspective) of the project pilot cases. Note that we need to support different types of federations. Some are open for many actors to join, e.g., all mobility service providers in a region or country, some are only open to two groups of actors, e.g., between customers and a company and some are internal, connecting different parts of an organization together. Each of these types comes with different inherent trust assumptions and needs for evidence, compliance and enforcement. Moreover, TEADAL's pilot cases vary in the source and direction of shared data, as some operate in multi-cloud environments and others operate in the Cloud/Edge continuum. Consequently, resulting in different requirements on the volume and depth of the evidence data that is stored.

Lastly, Table 2 also reveals different needs when it comes to protecting shared data, as data may contain personal identifiable information or confidential business information. In these cases, TEADAL tools must enable data owners to verify that implemented measures to ensure regulatory compliance, e.g., anonymization, filtering, or encryption, are executed before data are shared with consumers. Here, we distinguish between two main needs, data provenance and reporting. For data provenance we must show data consumers where their data is coming from and how it was transformed. This also includes evidence of data sharing purpose and the mandate which resulted in data being produced. For reporting, we must ensure that a data owner can always review all the events that led to a data exchange, including all the steps that were taken to send data to a consumer.

Since some of the required regulations also demand additional records, such as data processor, consumption purpose and user consent, TEADAL's evidence collection system must allow for easy extension to enable pilot cases to track regulatory information.

2.1.4 Trust Models

Taking together the requirements from the different data lakes planned for the TEADAL pilot cases and the general requirements of creating verifiable, evidence-based data exchange within the federated data product approach demands a differentiated view on how trustworthiness is implemented. Naturally, the demands for providing evidence in a verifiable, immutable, and permanent way are far less stringent in a scenario where data are shared within an organization or when data are meant to be open and public. Consequently, TEADAL provides several trust models that govern how and where we store evidence which will also significantly impact the cost of operating TEADAL.

Open - a Provider trusts the Consumer to process raw or nearly raw data. This implies that data can be considered open, e.g., as envisioned in Pilot #2. In these federations, there is no strict need to audit or verify all interactions. We can assume that data are provided through a public license or a lightweight registration process, which requires little supervision. Here, all enforcement relies on the FDP implementation, including possible collection of user data.

Centralized - in a centralized federation, the trust lies within a single entity or organization or the data is controlled by a centralized authority. Thus, policy enforcement, user management and other crucial elements of the data-sharing process can be dealt with using internal mechanisms. This also implies that both Provider and Consumer are in an established contractual relationship, i.e., both employed by the same company. In these cases, it is plausible to provide access to all audit data only within the organization and also utilizes permission-based DLT for managing most of the immutable records. This type of federation applies to Pilot #4 and #5.

Decentralized - in fully decentralized federations, the Provider and Consumer might not be known beforehand, implying a need for a consensus when joining a federation or when requesting a federated data product. In these cases, evidence must be accessible and verifiable by all parties in a free and open manner. Moreover, Providers and Consumers must reach specific, legally enforceable agreements before exchanging data. In these cases, the

Funded by the European Union

means to ensure that data are only provided in accordance with all applicable laws and regulations is crucial, requiring additional means for auditing and legal hold of access records.

In the context of the project, we must ensure that any of these cases can be supported, meaning that the collection of evidence can be configured to address these needs. Towards this end, the project also investigates different deployment models that may offer cost and performance advantages in cases where verifiability or availability of evidence can be relaxed.

2.2 THE EVIDENCE-BASED DATA LAKE ARCHITECTURE

In order to fulfill the TEADAL vision, we offer several components and integrate them with the general TEADAL architecture (see D2.2). As explained in the previous chapter, we are mainly concerned with gathering verifiable evidence of any data exchange facilitated by the TEADAL tools. To archive this, we place or augment several of the components within TEADAL's architecture with observation points.

Hence, in the following, we present the current vision of this architecture with a focus on the components we need for evidence collection. An overview of this architecture highlighting the flow of evidence can be seen in Figure 2; the central components and the evidence provided are highlighted in several info boxes throughout this section. This architecture is divided into several logical planes, each playing a different role with respect to what evidence can be observed and which actors we need to observe, e.g., for changes in the FDP we need to observe both consumers as well as FDP Designers. In contrast, we need to observe the DLO for changes to infrastructure services.

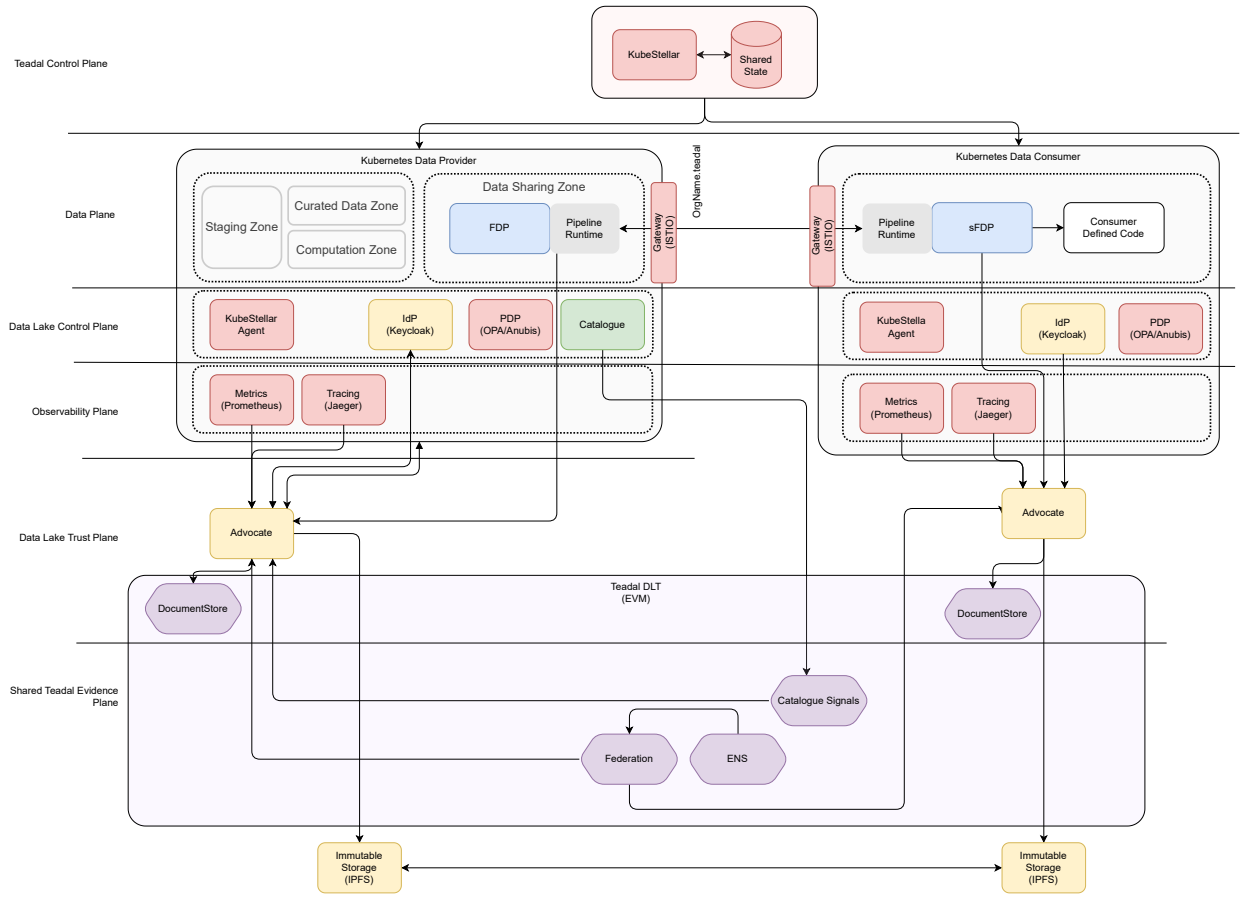

FIGURE 2 OVERVIEW OF THE TEADAL ARCHITECTURE WITH A FOCUS ON THE EVIDENCE-BASED DATA LAKE RELEVANT FOR ENHANCING THE TRUSTWORTHINESS *OF THE DATA EXCHANGE*

Funded by the European Union

2.2.1 TEADAL Control Plane

This plane centralizes the control function of the components deployed in the stretched data lakes, enabling global control of the computing and data resources distributed across the cloud continuum. In the current iteration of the evidence-based architecture, we are mainly collecting control actions.

In the long term, this layer could be instrumented to validate the source of control actions and be used to enforce policies before anything gets deployed, e.g., ensure that all containers come from known sources or that each endpoint is protected using an OPA agent and only apply changes to data lakes if these policies apply. A more detailed description is found in D4.1.

2.2.2 Federated Data Exchange Plane

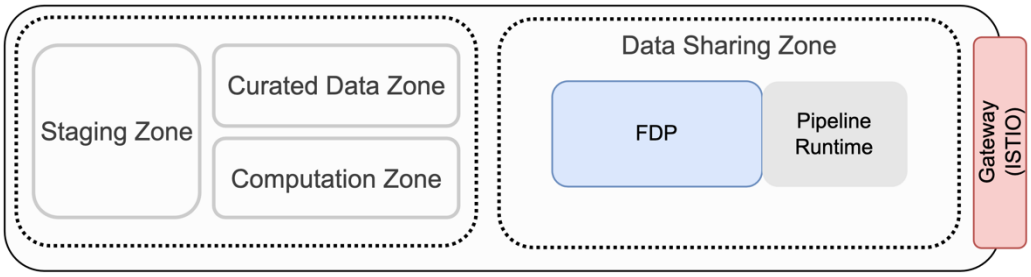

FIGURE 3 THE FEDERATED DATA EXCHANGE PLANE

This plane is composed of all the components that actively enable data exchange. This plane offers the capability to store datasets in S3-like object stores, providing capabilities to operate ETL-oriented applications like Spark, Airflow, Kubeflow that can be utilised to prepare datasets to be shared as well as to ingest the data into the data lake. Most crucially, however, this plane contains the FDP and sFDP components necessary to offer the data products to the data consumers, including all the business logic to evaluate and enact data-related policies.

If the datasets are stored or produced within the cluster, we can further collect evidence from the logical metadata of the datasets (e.g., file type and file size). Also, most modern data stores can provide detailed access logs, operation and change logs, and checksums, which can be integrated into TEADAL. The specificity of what details of datasets can be observed in that layer depends on the data lake implementation. While we will not be able to cover all possible data stores, we provide the procedures needed to integrate potentially relevant evidence information into TEDAL and link it appropriately to all other evidence collected. TEADAL can always observe the interactions within the FDP, as we provide clear guidelines on how an FDP developer should integrate evidence generation. At the minimum, we capture all the queries made to an FDP through its REST interface. Moreover, we collect all enforcement actions and other policy decisions of the Gateway.

For ETL applications, the system gets evidence from the workflows (e.g., start and end times of jobs, current job status, events triggering the workflow) if the workflow engines are implementing OpenTracing. The workflows could extract data from an external source or from inside the federated data lake. For all pipelines that provide transformed views of an FDP, i.e., an sFDP, the pipeline developers can decide how much internal information to provide as evidence. We collect evidence of the execution of each task at a minimum.

Besides these common components, we utilize Kubernetes to inject sidecars alongside any application deployed in a Federated Data Exchange Plane. For example, we use Istio to facilitate routing and act as a gateway, PEP and observability point. But we can also instrument

Funded by the European Union

Kubernetes to only allow specific containers to run within this plane, e.g., only images signed by the DLO or that are recorded in a Federation Agreement or the catalogue.

2.2.3 Data Lake Control Plane

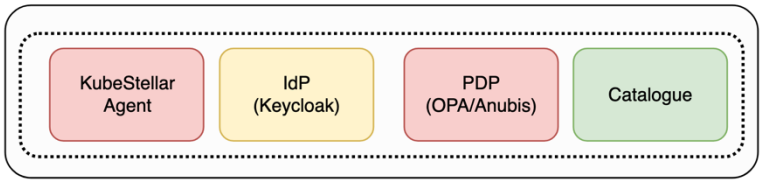

FIGURE 4 THE DATA LAKE CONTROL PLANE

In the data lake control plane, the Kubernetes Agent allows operators to manage the cluster from the central federated control plane (e.g., Kubestellar Agent). The secure management of the data lake is addressed by the Control Plane as specified in D.4.1, while the mechanisms relevant for trustworthiness are described in this document.

The Identity Provider or IdP (e.g., Keycloak) manages the different users of the data lake, including the management of internal data lake roles, e.g., DLO and FDP Designer. Moreover, each IdP can be used to validate and federate identities across the stretched data lake. How identities are mapped to rights within an FDP is always up to the FDP Designer/Developer. In this manner, the IdP connects to the Policy Decision Point (PDP) and the Policy Enforcement Point (PEP) to follow and enforce any policies that the data lake operator or a Federated Data Product (FDP) Designer defines. Here we include the means to observe the PEP, PDP and means to ensure that the linkage of IdPs across the federation will remain transparent. One central component in this architecture is the Catalogue, where the major definitions of the Federation Agreement are collected as well as where all information regarding published FDPs is stored and available to browse. Note that in the first iteration this component does not necessarily need to reside within every data lake but that every data lake will be able to reach a version of the data Catalogue relevant for the TEADAL data exchange.

Catalogue: **Role:** Acts as the main repository that provides metadata and general information about datasets. It also serves as the glue for the different components to get how data is spread across the system. **Evidence Provided:** Offers insights into the data lifecycle, including where the data originated, how it has been transformed, and where it's used. This ensures data traceability and provenance. This includes who is the producer of the data and who is the consumer of the data, as defined in the policies of the data sharing agreement.
Gateway: **Role:** Serves as a point of entry for system access, managing requests and ensuring only authorised access. **Evidence Provided:** Logs and monitors all access requests, granting a clear view of who tried to access what and when. This ensures that only authorised users can access FDPs and provides an audit trail for access patterns.

Funded by the European Union

Data Lake Policy Decision Point (PDP):

Role: These components manage authentication and authorization. The identity provider works as an identity and access management solution, the service mesh controls how microservices interact, and PDP (Policy Decision Point) makes decisions based on predefined policies. The use of openID will allow the interactions between multiple IdP instances.

Evidence Provided: They offer logs and records of actor interactions, detailing who did what and when. This ensures that user actions are traceable and can be audited for compliance and security purposes. PDP will also generate the triggers for data transformation depending on user role/level.

2.2.4 Observability Plane

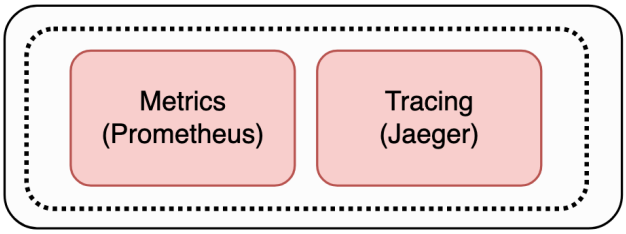

FIGURE 5 THE OBSERVABILITY PLANE

The Tracing and Monitoring components of the observability plane (e.g., Jaeger, Prometheus, Istio) generate the platform-level evidence, gathering information on who accessed the FDP/sFDP, what was accessed, when it was accessed and how it was accessed, among other events. As one of the core pillars of evidence collection, the next iteration of the project aims to improve the verifiability of the data coming from these components. For example, one approach we are investigating is using reviewed versions of these systems and ensuring proper collection by creating verifiable collectors. However, for now, we assume correct and non-malicious installation by the Data Lake Operator, who is also the main consumer of this information and thus is interested in error-free collection of information.

Observability Services and Tracing Services:

Role: They are used for tracing and monitoring all components running in the federated data exchange plane. Collection is enforced and, in combination with interaction of the control plane, ensured, e.g., by comparing requested components from the control plane with running components.

Evidence Provided: They give platform-level evidence, capturing system performance, and interactions. This ensures that the system's operations are transparent and can be monitored for any irregularities. These interactions can be as detailed as every interaction of a user to the bytes the user consumed.

2.2.5 Data Lake Trust Plane and Shared TEADAL Evidence Plane

In this plane, all the evidences provided by the previous planes are combined, verified and stored in an immutable way. This ensures that every interaction and data exchange can later

Funded by the European Union

be audited. The data lake trust plane consists further of the shared data trust plane, accessible to all members of a federation.

The central component of the data lake trust plane is Advocate, which orchestrates the evidence collection, summarizes it and then publishes it. The DLO is responsible for starting Advocate before any FDP can be created. A private/public keypair, issued by the DLO, associated with the DLO is used to then create a TEADAL Datalake ID which is used to sign any evidence collected on this data lake. Consequently, there will always be exactly one Advocate instance per data lake in TEADAL. Once established all collected evidence is then published in an efficient way in the DocumentStore which acts as the Claims Registry, and permanently stored in the Immutable Storage as a Verifiable Claims. We aim to further develop this component to perform verification of the evidence sources and enable compliance proofs instead of the publication of all claims as another avenue of evidence presentation that would have different cost and storage properties in the next iteration.

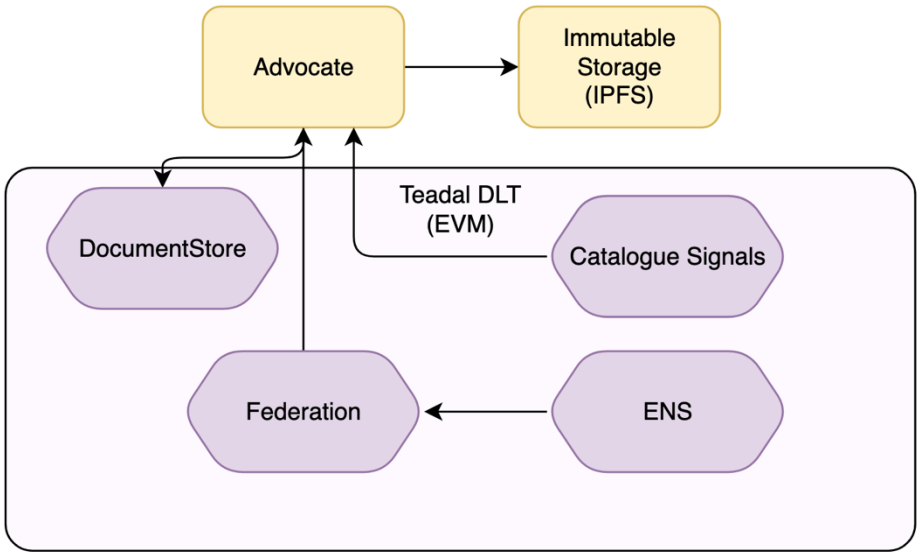

FIGURE 6 THE DATA LAKE TRUST PLANE AND SHARED EVIDENCE PLANE

Advocate: **Role:** Used to summarize and collect the evidence from the many sources in the TEADAL Data Lake and the attesting component by the TEADAL Data Lake that signs the evidence. It stores the results in the Shared Evidence Plane, providing data integrity to the system. **Evidence Provided:** Tamper-proof signed evidence and interactions with the Shared Evidence Plane.
Evidence-API: **Role:** API to interact programmatically with the Shared Evidence Plane trough Advocate. **Evidence Provided:** Provides hashes and elements that can point to the components in the Shared Evidence Plane and be verified with that Plane.

In the shared TEADAL Trust Plane tamper-proof events, FDP lifecycle signals and registries are stored on a blockchain. This plane consists of several components implemented as smart contracts (see Figure 6). One of these components is the Claims Registry, where the hash of claims is stored, allowing independent integrity verification of claims by anyone receiving a claims document. Another component is the Federation Contract, where the members of the

Funded by the European Union

Federation are registered as agreed in the external legal Federation Agreement. The Federation Contract functions as TEADAL anchor to legal entities. Thus, each Data Lake Operator (DLO) is identified with a personal public key used through TEADAL to authorize and enable data sharing (the same key required to start an Advocate instance). Either directly by a DLO or an entity that got these rights delegated by a DLO. In this plane, we also collect the signals from the Data catalogue as events, allowing us to track the life cycle changes of federated data products. This plane's immutable feature provides non-repudiation to the evidence stored and makes audibility and transparency for verification possible. Depending on the deployment and trust model selected for the federation this blockchain and therefore all relevant components of TEDAL are publicly readable thus, allowing easy independent integrity verification.

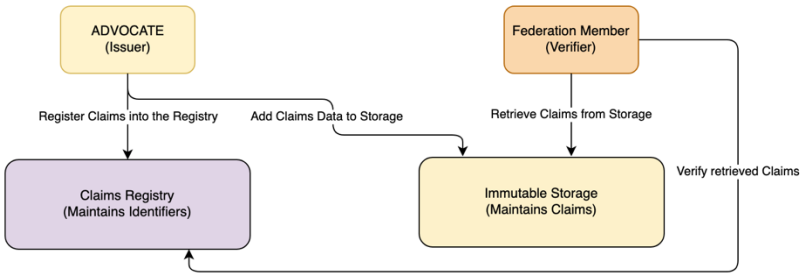

FIGURE 7 EVIDENCE LIFE-CYCLE

The claims of evidence protocol are represented in Figure 7 and are as follows: Advocate pulls the evidence from the observability plane and then proceeds to sign it. Within the collected observability data, we utilize verifiable credentials to associate users, components (FDPs) and environments. Advocate aggregates the pulled data and generates an interaction report document (VC) that is signed. The signed document is stored in the Immutable Storage and a unique hash to retrieve the signed document is obtained. The signature and the hash of the document are then registered in the Claims Registry so they can be verified later.

The verification of claims follows a similar process. First, a claim has to be obtained and/or verified against the Claims Registry; if the claim is not there, then it has not been issued by Advocate. Once the claim is obtained, the contents can be verified against the provided signature. To do so, it can be fully retrieved from the Immutable Storage to get the original signed Document.

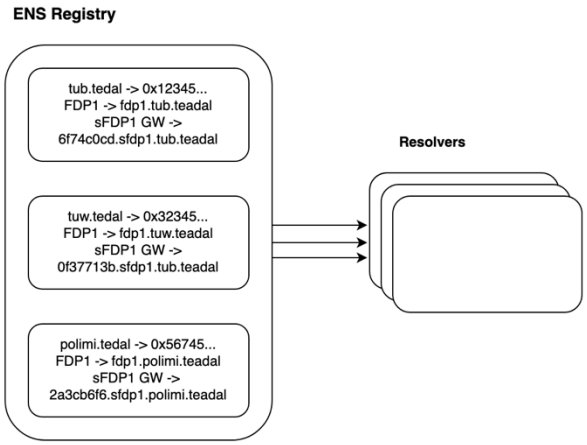

FIGURE 8 EXAMPLE OF AN ENS REGISTRY

Funded by
the European Union

The last component in the shared trust plane is the ENS Contract. This contract works as a decentralized name service in the DLT TEADAL operates; an example of it is shown in Figure 8. A federation member can register its public keys and signatures so other federation members can verify them, like the procedure explained for verifiable credentials. Once created, operators can create resolvable names for relevant objects within TEADAL. For instance, the federation agreement, FDPs, or sFDPs can provide names to respective rest endpoints. The benefits of ENS Contracts are that they are available for all the members of the federation, they can check for any record and manage their own records, while being secured by each organization's signatures. Another benefit of using ENS Contracts is that it reduces the human difficulty of handling 42 characters addresses representing organizations/people/services. Moreover, changes to the ENS and, therefore, changes to addresses of FDPs or sFDPs will always stored as a transaction in the blockchain.

Verifiable Claims and Claims Registry **Role:** Is the summarised evidence that is attested by the TEADAL Data Lake. Provides tamper-proof registry and log for claims. **Evidence Provided:** Cryptographic proofs of evidence, authenticity and origin. Tamper-proof and transparent registry for logs and its URIs to the immutable storage.
Immutable Evidence Storage **Role:** Immutable storage solution for evidence and data. **Evidence Provided:** Immutable storage of logs, ensuring data availability, redundancy, and resilience. Open to audit.
Ethereum Name Service **Role:** Provides tamper-proof DNS-like records of (s)FDPs and human readable link among organizations. **Evidence Provided:** Tamper-proof and transparent records of FDPs URIs.
Federation Contract **Role:** Provides tamper-proof representation of the members of the Federation as agreed in the Federation Agreement. **Evidence Provided:** Tamper-proof and transparent events of participation in a Federation Agreement reachable from an automated service point of view.

2.3 DEPLOYMENT MODELS

TEADAL trust plane relies heavily on the properties of blockchain technology. When federated data lakes are integrated with blockchain technology, it creates a powerful tool for data sharing and collaboration, in terms of

1. **Data Integrity:** The immutability of blockchain records ensures that once data is added to the blockchain, it cannot be altered or deleted. This guarantees the integrity of the data, making it a reliable source of information for all participants as long as the added data to the blockchain is valid.
2. **Transparency:** The distributed nature of blockchain promotes transparency as all participants have access to the same information. This fosters trust among participants as they can verify the data independently.
3. **Security:** Blockchain's decentralized structure and cryptographic techniques provide a high level of security. It's nearly impossible for hackers to alter the data as they would

Funded by the European Union

need to change the information on more than half of the network's nodes, which is computationally impractical.

4. **Efficient Data Sharing:** The combination of federated data lakes and blockchain allows for efficient data sharing. Data can be accessed in real-time, and the use of smart contracts can automate the data-sharing process, reducing the need for manual intervention.

However, using a blockchain to store evidence will also strongly impact the cost and confidentiality of TEADAL. For example, the use of a public permissionless blockchain, such as the Ethereum main net, will mean that anyone in the work can look at the claims contracts deployed by each advocate instance and thus allow the retrieval of the data exchange documents. Moreover, the insertion of claims to these contracts on the main net will always result in transaction fees that will make the use quite costly.

Consequently, in TEADAL, we explore multiple deployment models for the blockchain to enable integrity, transparency and security while balancing cost and public verifiable requirements per the trust models presented in subsection 2.1.4. For example, using a public blockchain is unnecessary when sharing open public data where evidence only needs to be verified by the Provider. Instead, the functionality provided by smart contracts can be reduced to interactions with the immutable storage and advocate itself. Similarly, when sharing data only within organizations or bilaterally between small groups, private blockchains can be explored. Private, often permissioned blockchains do not rely on a consensus algorithm to ensure the integrity of transactions and, therefore, do not require financial incentives in the form of transaction fees to record transactions. However, this also means that access to them requires approval by the operator of each private blockchain, making public verification challenging.

The selection of which deployment model to follow is left up to the federation members. The tools provided in TEADAL all rely on the EVM as the environment for smart contracts. Depending on the needs, members can decide to use no blockchain (open and trusted), a private permissioned blockchain, a private permissioned blockchain with public anchoring or a public blockchain. Similarly, the federation members can also decide if the immutable storage they want to use is public or private. Depending on their choice, they must follow established setup procedures, e.g., key exchanges, before installing the TEADAL tools.

2.3.1 TEADALs Private Blockchain Models

In the following, we describe one of the possible deployment models based on private blockchains, especially relevant for use cases that are between the centralized and decentralized trust models. It's important to underline that this is just a proposal and in contexts where trust among participants is minimal or non-existent, and it's possible to bear higher costs, the use of a public blockchain is more suitable and recommended.

The rise in popularity of private blockchains is attributed to their ability to offer privacy, control, and efficiency while maintaining essential features such as decentralization and security. These networks are customarily designed for a specific group of participants, limiting access to authorised entities, making them suitable for numerous applications across various industries.

Despite numerous advantages, private blockchains encounter challenges regarding data integrity and immutability. Their relatively smaller network sizes make them more prone to collusion or tampering compared to larger, public blockchains.

Funded by the European Union

To improve private blockchain integrity and immutability, we propose to adopt a protocol that periodically anchors their state to public blockchains, taking advantage of the security features of larger, decentralized networks.

The proposed protocol involves periodically creating a snapshot of the private blockchain's state, represented by a cryptographic hash, at specified intervals. This snapshot is then recorded onto a public blockchain, ensuring a tamper-resistant record of the private blockchain's state at various points in time. In case of suspected tampering, participants can verify the private blockchain's integrity by comparing its current state with the previously anchored snapshots on the public blockchain.

Implementing the protocol requires selecting an appropriate public blockchain for anchoring, determining the snapshot frequency, and creating a system for comparing the private blockchain's state with the anchored snapshots when needed.

At regular intervals (e.g., every N blocks), a cryptographic hash of the private blockchain's state is generated. This hash represents a unique fingerprint of the current state and can be used to verify data integrity later, see Figure 9.

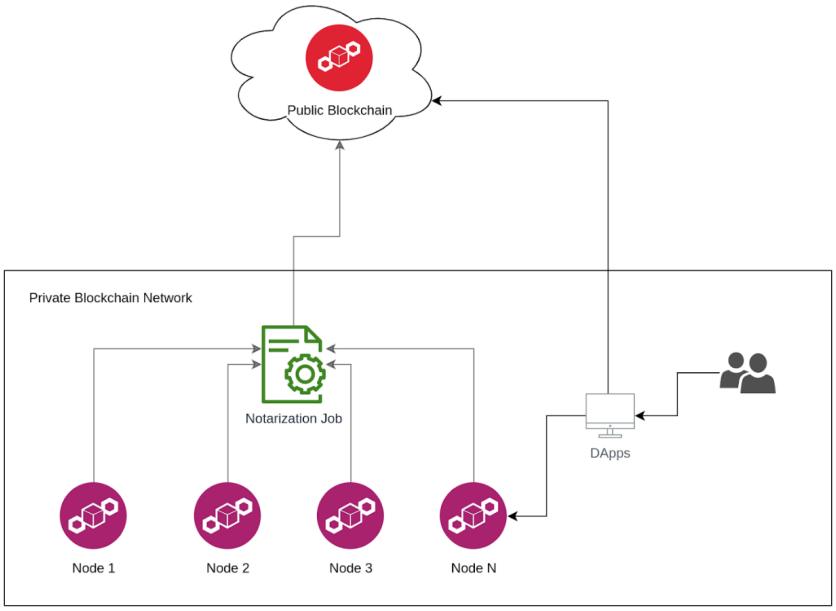

FIGURE 9 PROTOTYPE PRIVATE BLOCKCHAIN NETWORK DEPLOYED INTO THE POLIMI DATACENTER

The generated hash is recorded on a public blockchain, such as Ethereum or Bitcoin, which is considered tamper-resistant due to its large network and robust consensus mechanisms. The recording process can involve creating a transaction that includes the hash as metadata or using a smart contract specifically designed for this purpose.

If any participant suspects tampering within the private network, they can verify the integrity of the private blockchain by comparing its current state with the anchored snapshots stored on the public blockchain. Any discrepancy between the current state and the snapshots would indicate that the private blockchain's data has been altered.

To facilitate the process of recording private blockchain snapshots on the public blockchain, a dedicated smart contract is deployed to the public network. This smart contract is designed to store snapshots for each node in the private network, providing a tamper-resistant and transparent record of the private blockchain's state at different points in time.

Funded by the European Union

For viewing and verifying Private Blockchain Status a decentralized application (DApp) is developed to provide an accessible and user-friendly interface for interacting with the public blockchain and the smart contract storing the snapshots.

The DApp connects directly to the public blockchain, allowing users to view the recorded snapshots and check the status of the private blockchain. In case of any suspected tampering, users can use the DApp to verify the private blockchain's integrity by comparing its current state with the anchored snapshots stored on the public blockchain.

Funded by
the European Union

3 TRUSTWORTHY FEDERATION MECHANISMS

In the following, we discuss the concrete mechanisms available through the TEADAL trust plane to provide trust to data providers and consumers. We provide detailed information about the high-level description of components discussed in Section 2, including the description of developed components and envisioned ongoing research and development.

3.1 CORE FUNCTIONALITIES

As we laid out in the previous chapters, it's crucial to provide evidence of good (compliant) behaviour in order to enable trustworthy interactions between Providers and Consumers. Toward this end, we consider a set of core functionalities that cover the lifecycle of a data lake and federated data products, so we either collect evidence or have points where we can offer verification/auditing of interactions. Within these core functionalities, we differentiate between four categories.

Firstly, **identifying functionalities**: The identification of the different components, actors and data sets within a federated data exchange founds the evidence we provide. Thus, we must be able to identify the different actors (users) by utilizing ideally unique and personal tokens to ensure that each actor is truly the one performing any tasks within TEADAL. In the first iteration, we will rely heavily on DLT-based wallets and the underlying public/private keys to accomplish this. However, other means, such as self-sovereign identity approaches [4] or approaches presented by the IDSA, can also be considered. Moreover, in open or centralized cases, the identity provided already present within the organization will be used instead. Following that, for the identification of components, we will rely on mechanisms such as verifiable credentials generated based on control plane interactions, thus certifying what type of components are deployed and how they are used at runtime. Within this, we also need to identify federated data products as well as any agreements and contracts.

Secondly, **observable interactions**: While identities will mostly be static and have a clear source, the interaction with TEADAL components during runtime requires a different set of functionalities. In this, TEADAL must provide means to track, store and audit any interaction both on the application (FDP/sFDP) and platform (Kubernetes) level as well as on the consumption (FDP lifecycle) and control level. Here, we again rely on verifiable credential approaches that attest the observation of events (logs, traces, breakpoints) as well as attest stage changes in agreements or metadata.

Thirdly, **verification and proving functionalities:** The collection of identities and observable evidence already enable manual auditing of any data exchange process. However, we must further offer functionalities that enable automatic verification and, if possible, even proof of valid behaviour. Thus, enabling envisioned means of mediationless conflict resolution or simplified monetization of data exchanges later. This also includes the improvement of so far used tools on the application and platform side to not only provide the evidence in the form of observable records but also enable a best-effort approach for verifying the completeness of collected data.

Lastly, **attestation functionalities:** In order to give Consumers and Providers means to attest the delivery of data in accordance with agreements, we offer attestation functionalities. This includes means to ensure that data was only shared in a lawful way, e.g., by ensuring that consent from data subjects was collected before sharing data. For these functionalities, we envision the combination of all so far described functionality into a single publicly verifiable credential. This functionality will rely on the careful designs of FDP Developers, where TEDAL

Funded by the European Union

will aim to provide reference implementations or prototypes to demonstrate the possibilities of these features.

In all these functionalities, we are bound by the correct application of the tools and standards we provided to DLO, Developers and Consumers. However, we aim to make the process of using our tools as frictionless as possible by utilizing existing standards and tooling where possible, starting with the development of TEADAL data lakes up to the consumption of data. Moreover, we will aim to provide a set of automatic or almost automatic functionalities as a bare minimum that can be extended by Developers at will to reach the required level of trustworthiness when using TEADAL for sharing data.

3.2 VERIFIABLE EVIDENCE DATA

Auditability is a cornerstone of evidence-based systems. It underscores the principle that for evidence to be trustworthy, it should be open to scrutiny and validation by Federation members. This transparency ensures credibility and reinforces integrity. Especially in use cases concerning sensitive data, such as medical research or business operations, the ability to audit and verify evidence and its compliance is pivotal in maintaining the rigour and reliability of the system.

In the following, we will detail parts of this auditability by discussing in detail the approaches to implementing verifiable data in the first iteration of TEADAL.

3.2.1 Evidence Attestation

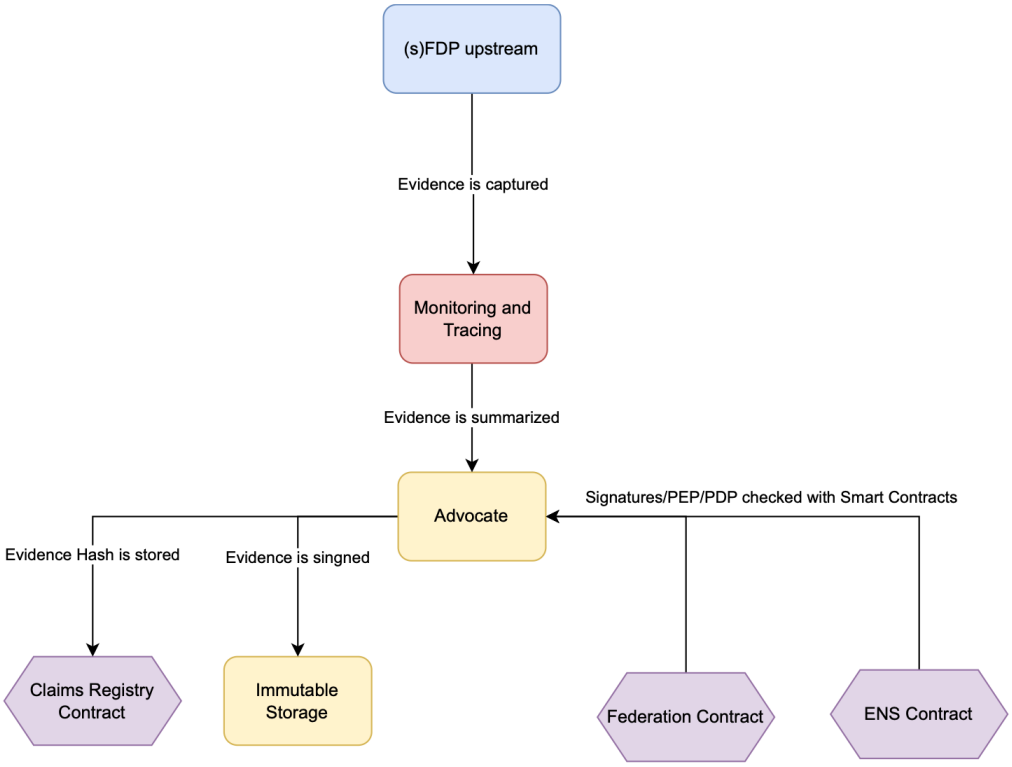

FIGURE 10 EVIDENCE ATTESTATION PROCESS

The first evidence to be collected is during the creation of the FDP by collecting signals from the Data Catalogue. This process not only annotates metadata pertaining to the data's origin and nature but also signifies the capability of an origin data lake to instantiate this FDP. An

Funded by the European Union

FDP's inclusion in the Catalogue inherently denotes that the associated data lake possesses the requisite rights to share the associated data, that policies defined by the FDP designer are sound, and that the data lake has the capacity to serve this data product and necessary transformations. The next observable chain of events is the instantiation of the FDP in the data lake, triggering several Kubernetes events and observability events that are collected as evidence.

During the operation of an (s)FDP, the system routes data access requests through the PEP/Gateway for validation. Utilizing the IdP for authentication and the PDP for authorization verification, the request is either permitted or denied based on the predefined access control parameters. Each of these interactions is persistently logged, thereby creating a traceable record of data access events that will be used as evidence.

To ensure a robust monitoring mechanism, as data traverses the system and as processes get executed, the platform generates granular traces capturing intricate system interactions. These traces are then aggregated and batch-signed, ensuring a consolidated record of evidence. Depending on the implementation provided by the FDP Developer, these traces can reach down to the storage layer of the raw data, revealing the precise bytes returned for each data access request. All traces can always be associated with the specific user accessing the data and the agreement allowing access to the data.

To finalise the evidence chain, each piece of evidence undergoes a cryptographic signature process using verifiable credentials in tandem with crypto wallets. This cryptographic endorsement assures stakeholders of the evidence's non-repudiation, authenticity, and integral preservation and links actions to the humans responsible for creating the FDP or allowing the data to be shared. All evidence is collected in distributed append-only storage and anchored in distributed ledger contracts to ensure tamper resistance.

3.2.2 Evidence Auditing

A primary objective of evidence generation is to facilitate its subsequent auditing. For federation members, this auditing is rendered reliably achievable due to the utilization of immutable storage mechanisms coupled with tamper-resistant logging services provided by blockchain technologies. These combined approaches ensure that the evidence remains immutable and trustworthy throughout its lifecycle.

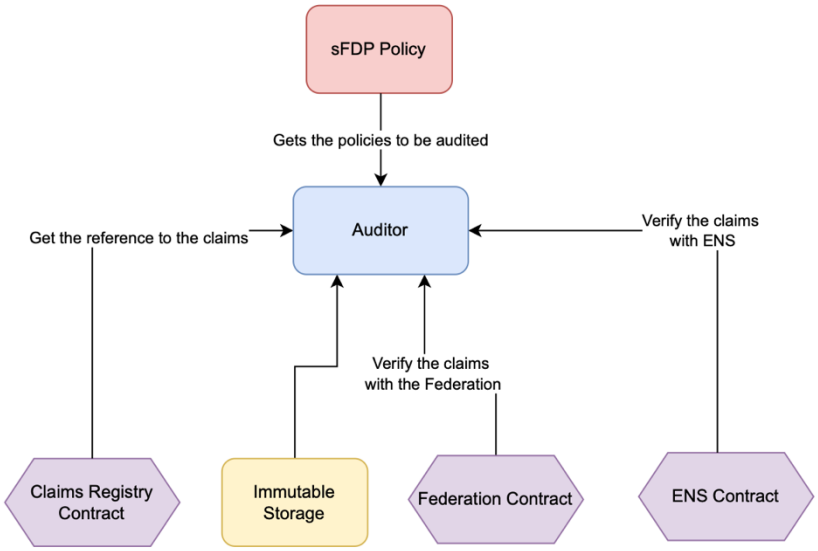

FIGURE 11 AUDIT PROCESS

Funded by the European Union

The process of evidence verification begins with an auditor accessing the data catalogue repository to retrieve the (s)FDP policy. This initial step involves programmatically fetching the pertinent policies, underpinning the audit's foundation, and ensuring the authenticity of the encapsulated claims. Subsequently, the auditor delves into the Claims Registry, a structured digital ledger, to extract the recorded claims. Using the cryptographic hash pointers provided, they can then pinpoint the exact storage locations of these claims within the immutable append-only storage.

The next phase entails engaging with the immutable append-only storage platform. This shared repository, accessible to all federation members, ensures that the retrieved claims are in their original, untampered state, underscoring the fidelity of the data.

With the Federation Contract housed on the blockchain, the auditor confirms the authenticity of the parties involved. By cross-referencing the list of members detailed in the contract, the auditor can verify that each party had the requisite authority to contribute to the evidence generation, rooted in their federation agreement. TEADAL provides the tools to compile this audit record to an auditor to make this process more efficient, but it is architected in such a way that an auditor can also perform an independent review without any TEADAL tools.

Through this procedure, the auditor methodically ensures the veracity, integrity, and legitimacy of the evidence provided by the TEADAL data lake.

3.2.3 Evidence Collection Example

In the remainder of this section, we will present a general example of how this evidence collection can be implemented in practice, following the logical view shown in Figure 12.

Instrumentalisation of (s)FDP

In the context of a Federated Data Product, instrumentation is essential for capturing the nuances of decentralized interactions and data exchanges. It creates a tangible record of data access, distribution, and computation across multiple entities. This data trail serves as evidence of compliance with federated agreements, verifies the integrity of distributed computations, and provides transparency in data usage and sharing. Furthermore, it ensures that each participating entity in the federation adheres to the agreed-upon protocols, respects data sovereignty, and maintains the privacy standards set. In essence, instrumentation in a Federated Data Product offers a foundation of trust and verification.

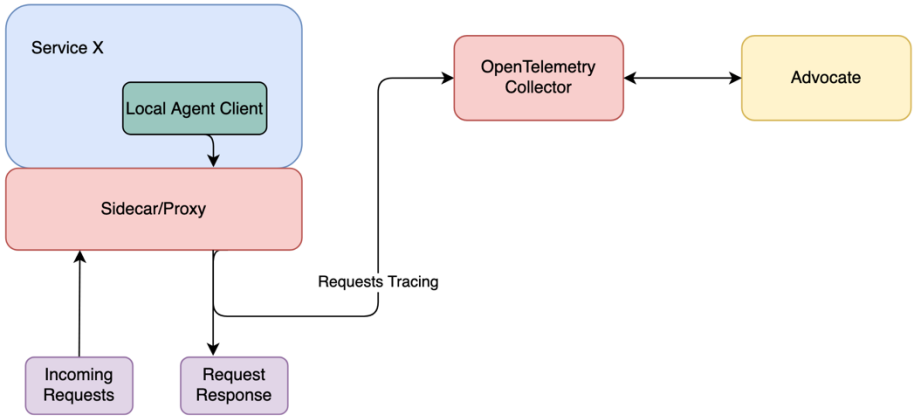

FIGURE 12 USAGE OF OBSERVABILITY AS AN EVIDENCE PROVIDER

Funded by the European Union

In order for TEADAL to observe the interactions within the FDP/sFDP, a Developer must implement an OpenTelemtery-compliant library; this usually can be archived with very little additional code see the example endpoint in Figure 13.

```python
@app.route('/sfdp-endpoint', methods=['GET'])
def sfdp_endpoint():
    with tracer.start_span('sfdp-endpoint') as span:
        id_field = request.json.get('id')
        span.set_tag('sfdp_id', id_field)
    return jsonify(status='success', id=id_field)
```

FIGURE 13 EXAMPLE OF TRACING INSTRUMENTATION IN FLASK (PYTHON)

Interception with Sidecars/Proxies

In FDP, sidecars or proxies, such as Istio or Envoy, act as pivotal intermediaries, intercepting and managing data traffic between federated entities. They ensure and enforce federation-specific access policies and provide crucial observability by logging interactions.

For example, using Istio, we can configure it to ensure that incoming requests to the specified service have valid JWTs issued by the designated issuer, a task that typically would represent a codified policy by the Developer or FDP designer. In practice, this configuration could look like the Istio configuration in Figure 14. However, in the more advanced usage of TEADAL, the policies will rely on an Open Policy Agent to perform such authentication checks.

```yaml
apiVersion: security.istio.io/v1beta1
kind: RequestAuthentication
metadata:
  name: jwt-authn
  namespace: teadal
spec:
  selector:
    matchLabels:
      app: fdp-service-agreement
  jwtRules:
  - issuer: "issuer@tub.teadal"
    jwksUri: "https://tub.teadal/.well-known/jwks.json"
```

FIGURE 14 ISTIO AUTHENTICATION CONFIGURATION EXAMPLE

Other policies can be implemented, such as limiting users with specific claims to only use GET requests, see the following configuration (see Figure 15).

Funded by the European Union

```yaml
apiVersion: security.istio.io/v1beta1
kind: AuthorizationPolicy
metadata:
  name: allow-reader
  namespace: teadal
spec:
  selector:
    matchLabels:
      app: fdp-service-agreement
  rules:
  - to:
    - operation:
        methods: ["GET"]
    when:
    - key: request.auth.claims[role]
      values: ["reader"]
```

FIGURE 15 AUTHORIZATION POLICY EXAMPLE

The specifics are left up to the developer and designer of an FDP. TEADAL is concerned with providing common standards to implement these policies (see D4/D3) and to be able to observe the enactment of the policy on a granular level to provide the necessary evidence. However, since we use CNF-compliant components, basic tracing and logging can always be used to archive basic evidence collection.

Summarisation of Evidence

In a Federated Data Product context, a Verifiable Credential (VC) serves as a digital confirmation of interactions within TEADAL. For example, Figure 16 captures evidence of a data query on a specific product named "energyStats2023" initiated from a node. It provides details like the type of interaction, timestamp, a hash of the response for data integrity, and additional specifics like the query parameters and the accessed federated node. This VC, cryptographically signed by the issuer node, acts as a trustable record, ensuring transparency and authenticity of the interactions that have taken place within the federated environment.

Funded by the European Union

```
{
  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://federated.data.product/credentials/v1"
  ],
  "type": ["VerifiableCredential", "FederatedDataProductEvidence"],
  "issuer": "https://fdp-node.tub.teadal/credentials/issuer",
  "issuanceDate": "2023-10-23T04:24:12Z",
  "credentialSubject": {
    "id": "did:example:12345abcdef",
    "type": "FDPInteraction",
    "interactionType": "dataQuery",
    "dataProduct": "energyStats2023",
    "timestamp": "2023-10-23T03:20:10Z",
    "responseHash": "0x1234567890abcdef",
    "details": {
      "queryParameters": {
        "location": "Berlin",
        "date": "2023-10-22"
      },
      "accessNode": "https://fdp-node2tub.teadal"
    }
  },
  "proof": {
    "type": "RsaSignature2018",
    "created": "2023-10-23T04:24:12Z",
    "proofPurpose": "assertionMethod",
    "verificationMethod": "https://fdp-node.tub.teadal/credentials/issuer/keys/1",
    "jws": "eyJhbG..."
  }
}
```


FIGURE 16 JSON EXAMPLE REPRESENTATION OF VERIFIABLE CREDENTIAL

Smart Contract Events

For the Federation Contract, the events serve as auditable, tamper-proof records of crucial membership actions within the federation. One such contract could look like Figure 17. The `MemberAdded` event confirms the onboarding of new members, ensuring transparency about who has access to the federated data. The `MemberApproved` event showcases peer endorsement, signifying trust and consensus in adding a new member. Lastly, the `NewApprovalCount` event provides the approvals a candidate member has received. Together, these events offer verifiable evidence of the federation's membership dynamics, reinforcing trust and accountability among participants. Note, however, that each trust model and, therefore, each deployment model may require a different implementation of this contract. Figure 17 serves just as an example of a contract applicable in a decentralized use case, with a centralized or notarized authority that can add members to a federation.

Funded by the European Union

```
/**
 * @notice Emitted when a new member is added
 * @param newMember The address of the added member
 */
event MemberAdded(address indexed newMember);

/**
 * @notice Emitted when a member approves a member candidate
 * @param member The address of the member who approved
 * @param newMember The address of the member being approved
 */
event MemberApproved(address indexed member, address indexed newMember);

/**
 * @notice Emitted when an approval count for a candidate adds 1 to its count
 * @param newMember The address of the member whose approval count has increased
 * @param approvalCount The new approval count for the member
 */
event NewApprovalCount(address indexed newMember, uint256 approvalCount);
```

FIGURE 17 EXAMPLE OF A FEDERATION CONTRACT IMPLEMENTED IN SOLIDITY

In the case of the Claim Registry Contract, they have the same properties of tamper-proofness. An example of this can be seen in Figure 18. The DocumentIssued event captures the issuance of a new document, identified by its unique hash, ensuring that every addition to the registry is transparent and traceable. Conversely, the DocumentRevoked event records the removal or invalidation of a document, providing clear evidence of any changes in the validity status of registered claims.

```
/**
 * @notice Emitted when a document is issued
 * @param document The hash of the issued document
 */
event DocumentIssued(bytes32 indexed document);

/**
 * @notice Emitted when a document is revoked
 * @param document The hash of the revoked document
 */
event DocumentRevoked(bytes32 indexed document);
```

FIGURE 18 EXAMPLE OF A CLAIMS REGISTRY CONTRACT IMPLEMENTED IN SOLIDITY

The functionalities provided by the smart contracts are then related to the 4 types of core functionalities. The identification occurs when interactions with the DLT are logged in it with the signature of the transaction executer. Observability comes from the fact the DLT is logging all interactions and is accessible for all the parties. The verification and attestation are part of the signature protocol which allow a party with access to the DLT to verify the signatures in the transaction and from the fact that only the holder of the private key can sign and attest for a specific wallet.

3.3 VERIFIABLE INTERACTIONS

As the Catalogue holds the descriptions of Datasets and Federated Data Product and saves all the versions of an item, it becomes important to ensure that neither the latest nor the previous versions of the metadata of an item are tampered. The objective of the TEADAL Tracking functionality is therefore associated to the tracking and monitoring of (i) the lifecycle of digital artifacts, and (ii) the relevant runtime interactions among digital artifacts. To reach

Funded by the European Union

this goal, it is necessary that, in an immutable manner, lifecycle and runtime operations tracked through the TEADAL Catalogue are written on the blockchain and that they can be validated at a later stage to ensure the authenticity and correctness of the information. However, blockchains are not databases and it is expensive and complicated to manage the writing of large data within them. For this reason, the selected approach is based on storing on the blockchain a concise representation of the content of the tracked operation. Specifically, an irreversible function capable of calculating the hash for the content of an operation is used. The resulting value is written into the blockchain for each operation and the identifier of the transaction returned by the blockchain is stored together with the operation in a separate database.

This serves to limit the amount of data inserted within the network while simultaneously guaranteeing security, transparency and the validation of data. As a result, it is then possible to compare the hash in the blockchain with the hash of the data of the operation retrieved from the database and check whether the data is correct or whether it is corrupted or altered. Proper measures should be defined for the database selected for storing operation data, such as replicability. Indeed, the hash stored in the blockchain can not be used to recover the operation data but only for the validation of given data.

3.3.1 Tracking API Prototype

The TEADAL Tracking prototype was developed starting from the IAMS (Intelligent Asset Management System) Tracking component developed by CEFRIEL in the context of Shift2Rail IP3 DayDreams project, and some simplifications were introduced. In a real context, the blockchain would have many nodes and define proper governance and management of credentials, mining nodes, and smart contracts. In the first iteration of the TEADAL development cycle, a single node of a private Ethereum blockchain was considered. The node runs in a Docker container and in a dedicated network. The private network is configured defining a specific genesis state. A sufficient number of accounts was set up that possess an adequate amount of ethers to perform operations of any kind, but which are bogus ethers and not connected to the ethers of the Ethereum main-net. One account was associated with the miner node and one account with the TEADAL Catalogue. Furthermore, the network used is a fork of the Ethereum main-net and was created using a random number as chain id. In the implemented prototype, transactions are not governed by smart contracts. To simplify the management of the network the logic for registering and validating transactions was moved within the TEADAL Tracking component. For each operation received, the TEADAL Tracking sends a transaction to the blockchain containing as an arbitrary hexadecimal value the computed hash of the operation data. Once obtained back the identifier of the performed transaction (transaction hash) the TEADAL Tracking component stores in a Redis database the operation data enriched with the computed hash and the transaction hash. Given a stored operation, it is then always possible to retrieve the hash associated with the transaction on the blockchain and validate the operation data.
The TEADAL Tracking component associates all the operations to the same log of operations (associated with the same *artifact_id*) in the database and provides methods for retrieving and validating an operation and/or a log of operations.

The definition of the format for operations leverages the idea of aggregating operations in a log via a specific *logId*. The log of operations is also characterised by a *logType* that can be used to specify the scope of operations in the log. In the prototype, two admissible values for *logType* were defined: runtime and lifecycle. For runtime logs the *logId* corresponds to the *session_id*, and for lifecycle logs the *logId* corresponds to the *artifact_id*. Each operation is associated with the unique identifier of the user of the TEADAL Catalogue

Funded by the European Union

that performed the operation and with a set of operation data partially inspired by the Common Workflow Language (CWL) specification:

- *operationId*: unique identifier of the operation;
- *operationType*: type of operation;
- *operationTimestamp*: timestamp associated with the tracking of the operation;
- *operationInputs*: identifier(s) (e.g., reference to a digital artifact in the TEADAL Catalogue) of input(s) of the operation;
- *operationComponents*: identifier(s) (e.g., reference to a digital artifact in the TEADAL Catalogue) of component(s) used to perform the operation;
- *operationOutputs*: identifier(s) (e.g., reference to a digital artifact in the TEADAL Catalogue) of output(s) of the operation;
- operationParameters: identifier(s) (e.g., reference to a digital artifact in the TEADAL Catalogue) of parameter(s) for the configuration of the operation performed.

```
{
"logId": "42 ms5d",
"logType": "runtime",
"username": "daydreams -user",
"operationId": "runtime -operation -test",
"operationType": "train",
"operationTimestamp ": "16699703458",
"operationInputs": ["Dataset/My -Dataset -1","Dataset/My -Dataset -2"],
" operationComponents": ["Software component/My -Train -Module"],
"operationOutputs": ["Machine learning model/My -Trained -Model"],
"operationParameters": ["--logReg"]
}
```

FIGURE 19 EXAMPLE JSON PAYLOAD OF A RUNTIME OPERATION

```
{
"logId": "My-Dataset",
"logType": "lifecycle",
"username": "admin",
"operationId": "lifecycle -operation -test",
"operationType": "create -artifact",
"operationTimestamp ": "16699703458" ,
"operationInputs": [],
" operationComponents": ["/postAsset"],
"operationOutputs": ["Dataset/My-Dataset"],
" operationParameters": []
}
```

FIGURE 20 EXAMPLE JSON PAYLOAD OF A LIFECYCLE OPERATION

The following paragraphs detail the endpoints exposed by the TEADAL Tracking component.

Funded by the European Union

Log operation

- Interaction description: Track a new operation adding it to a log of operations. The first time a new *logId* is received, a new log of operations is created. The body of the request should contain a JSON representing the operation.
- API endpoint: `tracking/logOperation`

Verify operation

- Interaction description: Verify a tracked operation through the log. The body of the request should contain a JSON representing the operation with the corresponding *transactionHash*.
- API endpoint: `tracking/verifyOperation`

Verify log

- Interaction description: Verify a set of tracked operations through the log. The body of the request should contain a JSON object containing an operations JSON array. Each element of the array should represent a tracked operation with the corresponding *transactionHash*.
- API endpoint: `tracking/verifyLog`

Get log

- Interaction description: Get all the tracked operations associated with the provided logId.
- API endpoint: `tracking/getLog/{logId}`

3.3.2 Integration with the TEADAL Catalogue

The TEADAL Catalogue can be integrated with the Tracking API to ensure that lifecycle operations associated with the digital artifacts are tracked.

The lifecycle process associated with digital artefact types, defined using the BPMN formalism, is responsible for tracking relevant operations for each artefact. The following figure shows an example of how the Tracking API has been exploited to send information about the “publish” operation of an artefact. All the lifecycle operations made by users on a digital artefact can be transparently tracked by the TEADAL Catalogue and forwarded to the Tracking component.

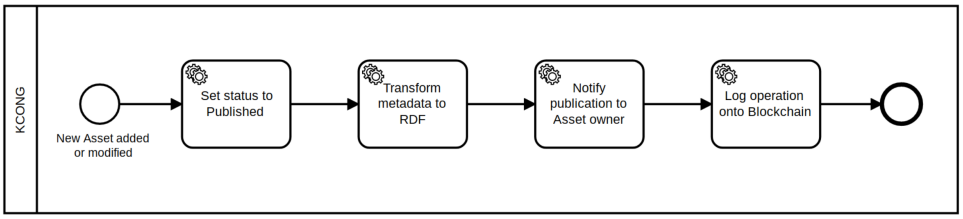

FIGURE 21 EXAMPLE BPMN FOR SEND EVENT ABOUT FDP LIFECYCLE CHANGES

Funded by the European Union

3.3.3 Evaluation in the context of the first iteration

The behaviour of the Tracking component developed in the first iteration is consistent with the initial context of having a single Catalogue for a specific data-sharing ecosystem. In such a scenario, each of the participating entities contributes to the federation by:

- describing Federated Data Products on a nauthorize catalogue
- providing an Ethereum node to support the Tracking component in ensuring the authenticity of the metadata

In the context of the next iterations, we will adapt the implementation to support a fully peer-to-peer federation, listing all the conditions and rules that must be agreed upon by the participants in order to establish a working federation.

3.4 PRIVACY-PRESERVING COMPUTATIONS

Privacy-preserving technologies are indispensable tools in data-driven applications, especially in settings where sensitive data needs to be protected, while maintaining their utility. Three key technologies used in privacy-preserving computations are secure Multi-Party Computation (MPC), Trusted Execution Environments (TEEs), and Zero-Knowledge Proofs (ZKPs). Overall, these technologies make it possible to perform generic computations on sensitive data without compromising it's confidentiality.

MPC is a cryptographic technique that allows multiple parties to gather collective knowledge from their data while keeping their input data private. These parties input their data in a protected form and agree on a function to be computed on that data. The data remains encrypted throughout the computation process and the output of the function gets revealed to a designated output party who learns nothing about the input data besides what can be inferred from the output. Figure 22 shows a general pipeline in an MPC use case.

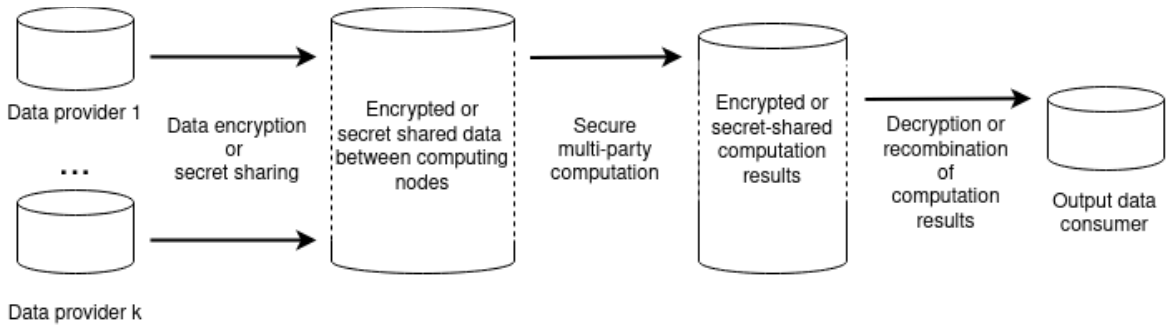

FIGURE 22 MULTI-PARTY COMPUTATION

TEEs are hardware-based security technologies that provide a secure environment for executing sensitive code, or code with sensitive inputs. hey are designed to prevent unauthorized access to the processed data, and to ensure the integrity and confidentiality of the computations. TEEs can also be used for multi-party computation.

ZKPs are cryptographic methods that allow one party (the prover) to prove to another party (the verifier) that some statement is true without conveying any other information. This technology is used in privacy-preserving systems to prove and verify the validity of a statement without revealing any additional information. Figure 23 shows a very general use-case of ZKPs. It is important to note that, while there exist non-interactive ZKPs, they can also be interactive, in which case the prover and verifier need to exchange multiple messages between each other, throughout the proof generation process.

Funded by the European Union

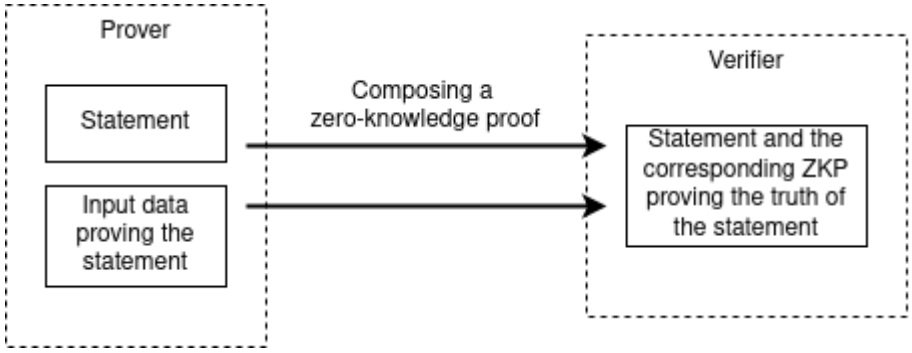

FIGURE 23 ZERO-KNOWLEDGE PROOF

Sharemind[1] is a family of secure computing tools that enable privacy-preserving data processing. Sharemind MPC implements secure computing on encrypted data using cryptographic protocols, and Sharemind HI (hardware isolation) implements secure computing on encrypted data using TEE. Both tools allow users to process and analyze private data without having access to the original values. Sharemind MPC is a multi-node application server (usually 3 independent computation nodes are needed) that can be deployed in any public/private cloud. It may work as a privacy-preserving distributed database, but can also go beyond tha– by performing analytics - ETL, statistical a–alysis, machine learning - on the encrypted data, with a set of analytical tools and available SDKs. Sharemind HI is a single-node application server which uses the Software Guard eXtensions (SGX) technology by Intel to create Trusted Execution Environments (easily found in modern cloud environments, e.g. IBM, or Microsoft Azure clouds). Among the typical scenarios, Sharemind tools can be used when one or more parties are willing to provide encrypted data to be queried by other parties, or two or more (distrusting) parties want to do computations on aggregated data, without disclosing their original values.

Sharemind tools fit into the privacy-focused data processing requirements in a federated data lake architecture of TEADAL, with the data sharing roles matching the FDP lifecycle actors. FDP providers design and define the privacy-preserving FDP, deploy the access policies, and operate the infrastructure. During its runtime, upon request from an FDP consumer, the privacy-preserving data-processing protocols, MPC or TEE, are instantiated and executed to deliver the computational outputs to the consumer. For the consumer, the entire interaction happens without them taking part in the privacy-preserving protocol execution. However, the design of the FDP needs to be compliant with the requirements and trust assumptions of such protocols, taking into account the fact that mutually distrusting or independent parties as data providers wish to compute information together using MPC, or that an independently trusted hardware environment for TEEs doesn't potentiate side-channel attacks. This ensures the meaningful use of privacy-preserving technologies and helps mitigate against different attack vectors. The privacy-preserving protocols should fit into the runtime component of the FDP, fed by the datasets described via the metadata, specified during the FDP design phase. The need for privacy-preserving computations has to be described by access policies, and the process runtime should provide the necessary traceability and metrics for the TEADAL trust plane. As a consequence, FDP development is tied to the protocol constraints, the most notable being the need for multiple independent computing nodes for MPC deployment. Apart from these considerations, which are specific to privacy-preserving technologies, the design of such FDPs also needs to comply with an auditing protocol that independently ensures trust and audits the computations before the FDP is published, with any further changes made in the computations also triggering a new auditing round, before the republishing of the FDP.

[1] https://sharemind.cyber.ee/

Funded by the European Union

An example use case of MPC, within the TEADAL set of pilot cases, is the healthcare pilot, where a federation of hospitals collaborates to produce a holistic view from their patients datasets, without any hospital compromising the confidentiality of their individual shares. The protocol allows the hospitals to produce a global combined report on the size of the datasets that fits the requirements of a given clinical study, in order for the study promoter to evaluate the potential impact of the study before the individual patients' data is consented to be revealed. Depending on the pilot requirements, MPC would also be useful for performing the actual study calculations, outputting analytics over the data, while maintaining the privacy of the patients' personal data. Figure 24 shows a potential set-up for such use cases. A study promoter is the data consumer who composes a query and asks for data from the federation of medical data providers. The data providers have either previously agreed or will agree on three independent computation nodes, some of whom can be the data providers themselves. These get secret shares of the input data and run the computations on the data without ever revealing the intermediate values. Finally, the output shares are sent to the consumer, who can combine them to get the computation results.

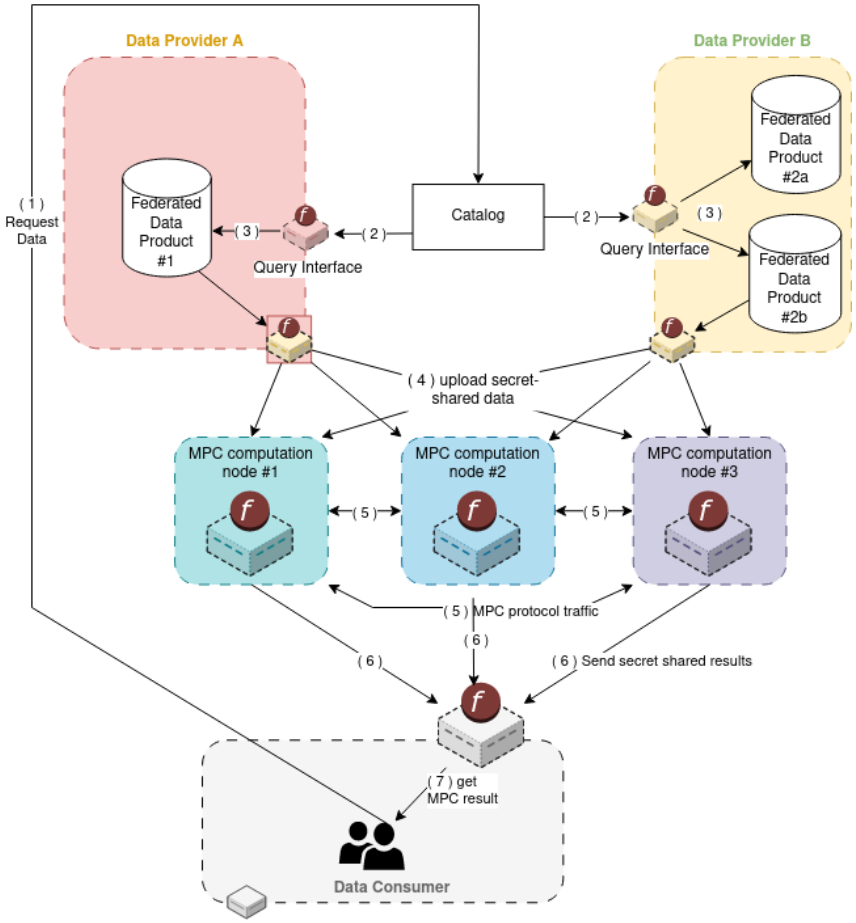

FIGURE 24 TEADAL PILOT MPC USE-CASE

TEEs can also be used for privacy-preserving computation with multiple parties, with the natural differences stemming from the solution's architecture and deployment constraints. A potential use case for TEEs is the financial pilot's need for producing a holistic view of KYC and CDD operational insights across geographies. Legal constraints that affect data movement, provision, or privacy, need to be considered, while still delivering a solution that complies with the analysis and auditability requirements within the financial landscape. For instance (see Figure 25), consider the case where data providers (two different branches of a bank) cannot share their data, but a data consumer wants to gather knowledge from their collective data. One of the data providers can encrypt their data and send it to the other one,

Funded by the European Union

who uses the trusted execution environment to run computations on their collective data. The data of the first provider is only decrypted in a secure enclave, where the second provider cannot access it. The first provider can audit the TEE to make sure the computations are run as agreed. The output of the computation can be exposed to the data consumer, and neither the provider nor the consumer learns more about the data than what can be gleaned from the output.

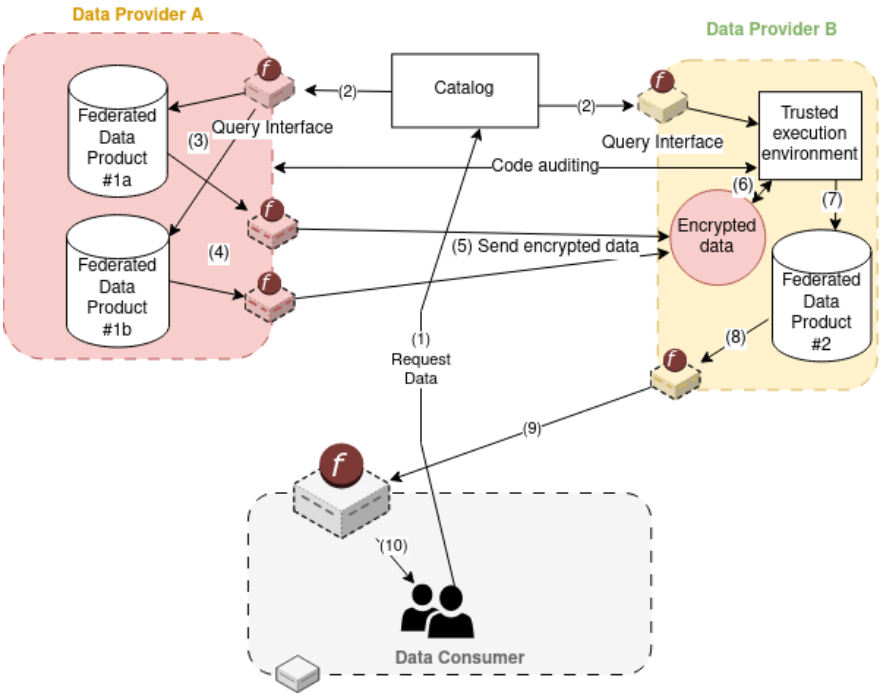

FIGURE 25 TEADAL PILOT TEES USE-CASE

Zero-Knowledge tools potentiate privacy-preserving data sharing by concealing information details while proving their validity. Among these, ZoKrates stands out as a toolbox for zk-SNARKs (Zero Knowledge Succinct Non-Interactive Arguments of Knowledge), especially influential in blockchain applications implementing privacy features [14]. This tool streamlines the integration of off-chain computations, offering a domain-specific language, compiler, and generators for proofs and verification within Smart Contracts. ZoKrates not only simplifies the inherent complexity of zero-knowledge proofs but also provides developers with programming abstractions, enabling circuit integration and fostering the widespread adoption of privacy-enhancing systems in the blockchain landscape. Utilizing non-interactive proofs for off-chain computations reduces the on-chain computational efforts 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.

Other tools like ZK-SecreC[15] offer more general toolkits for simplifying the definition of problem statements, from the business logic to ZK proofs, not necessarily optimised for scaling blockchain applications. The interface with multiple cryptographic back-ends allows for targeting problems with a diverse range of characteristics, for instance, the type of proof interactivity, size of the inputs, high-level requirements, or other architectural and deployment considerations. These are especially beneficial to pilot cases' privacy problem definitions, as eventual real world instantiations of this kind of knowledge proof paradigm, which would take advantage of ZK technology for privacy related matters. One example is the Regional Planning for Environmental Sustainability use case, where both sides collaborate to calculate energy usage statistics, but private data is only inherent to one of the stakeholders. Instead of having to reveal that private data, ZK proofs of correctness over the local computations can be

Funded by the European Union

produced, validating the fulfillment of the pilot's defined policies or other business-level requirements, in a privacy-preserving way.

These are only some of the potential use cases for MPC, TEEs, or ZKPs. The exact details of deployment and development are up to change with added functionalities to the TEADAL platform. In addition, privacy-preserving technologies could be employed in different ways, to deliver different data analysis solutions for other pilots, that conforms to existing constraints for handling the data. D2.1 and D2.2 contain a more detailed business description of potential use cases. With regards to TEADAL's general direction, and in line with the focal points of this deliverable, future plans may also be laid down in order to explore use cases of the technologies mentioned in this section that would go beyond the privacy-preserving pilot cases' data processing requirements. An evidence-based data lake architecture could benefit from such protocols with the aim of increasing trust, integrity, verifiability, or confidentiality, over the multiple data lake operations. Next iterations will dictate the conception and eventual development of these ideas.

Funded by the European Union

4 CONCLUSIONS

This deliverable marks the culmination of the initial phase in crafting TEADAL's trust architecture. Within this report, we delineated the essence of trust within the project, defining the concept of verifiable evidence, and explored the synergy of privacy-preserving technologies, distributed ledgers, and cloud-native platforms in creating trustworthy federated data lakes.

Through the review of the project pilots, we discerned diverse trust and deployment models. These models afford a balance between transparency, confidentiality, and trustworthiness, thereby laying the foundation for an evidence-based architectural design within a large set of data spaces, including inter organizational and publicly shared data exchanges.

Looking ahead, the next iteration involves a continuous refinement and evolution of this evidence-based architecture. Notably, we will integrate zero-knowledge proofs and other privacy-preserving technologies to overcome current limitations of evidence size and confidentiality. The forthcoming iteration will delve deeper into the integration with the data catalogue, enabling comprehensive monitoring of the entire FDP lifecycle. Additionally, a heightened integration with pivotal infrastructure components, such as the OPA agents and Kubernetes in general is crucial.

Building upon the insights gleaned from the upcoming pilot implementations, we aim to not only track consumption evidence at runtime but also throughout the development phase. Our next step focuses on a detailed analysis of specific scenarios, including GDPR compliance and pipeline optimization. This strategic approach will provide more precise and robust evidence for distinct data spaces within TEADAL.

Funded by the European Union

REFERENCES

[1] Alex Gorelik, The Enteprise Big Data Lake, 2019

[2] Council of European Union, on harmonised rules on fair access to and use of data (Data Act), 2022, COM/2022/68

[3] Otto B. et al., GAIA-X and IDS. International Data Spaces. , 2021, https://doi.org/10.5281/zenodo.5675897

[4] Council of European Union, A European strategy for data, 2020, COM/2020/66

[5] Mayer, Roger C., et al. "An Integrative Model of Organizational Trust." The Academy of Management Review, vol. 20, no. 3, 1995, pp. 709–34. JSTOR, https://doi.org/10.2307/258792. Accessed 28 Nov. 2023.

[6] Jonathan Heiss, Trustworthy Data Provisioning in Blockchain-based Decentralized Applications, 2023, Doctoral Thesis, Technische Universität Berlin

[7] Kini and Choobineh, Trust in electronic commerce: definition and theoretical considerations, 1998, https://doi.org/10.1109/HICSS.1998.655251

[8] T. Grandison and M. Sloman, A survey of trust in internet applications, 2000, https://doi.org/10.1109/COMST.2000.5340804

[9] Future Trust Consortium, On Trust and Trust Models, 2017, https://cordis.europa.eu/project/id/700542/results

[10] NIST, Engineering Trustworthy Secure Systems, 2022, https://csrc.nist.gov/pubs/sp/800/160/v1/r1/final

[11] Gaia-X, Gaia-X secure and trustworthy ecosystems with Self Sovereign Identity, 2022, https://www.gxfs.eu/ssi-whitepaper/

[12] Ulbricht, MR., Pallas, F., YaPPL - A Lightweight Privacy Preference Language for Legally Sufficient and Automated Consent Provision in IoT Scenarios, 2018, https://doi.org/10.1007/978-3-030-00305-0_23

[13] Grünewald, Elias, Cloud native privacy engineering through DevPrivOps, 2021, https://doi.org/10.1007/978-3-030-99100-5_10

[14] J. Eberhardt and S. Tai, ZoKrates - Scalable Privacy-Preserving Off-Chain Computations, 2018, https://doi.org/10.1109/Cybermatics_2018.2018.00199

[15] Bogdanov et al., ZK-SecreC: a Domain-Specific Language for Zero Knowledge Proofs, 2022, https://doi.org/10.48550/arXiv.2203.15448

Funded by the European Union