TEADAL

D2.4 Final general architecture

Work package	WP 2
Task	Task 2.1, 2.2, 2.3
Due date	31/05/2025
Submission date	31/05/2025
Deliverable lead	TUW
Version	1.2
Authors	Alicia Schwabenbauer (TUW), Boris Sedlak (TUW), Cynthia Marcelino (TUW); all partners and pilot cases
Reviewers	POLIMI, TUB
Abstract	This deliverable describes the general architecture of the TEADAL platform at system level, describing its components and their interactions.
Keywords	Architecture, requirements, pilot-cases, deployment, federation

Document Revision History

Version	Date	Description of change	List of contributor(s)
0.1	28.11.2024	First draft	TUW

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

0.2	24.04.2025	All Sections filled	All Partners
0.3	08.05.2025	Added financial pilot again	All Partners
1.0	14.05.2025	Formatting, Submission ready	TUW
1.1	15.05.2025	New Component	Almaviva, TUW
1.2	28.05.2025	Incorporate reviewer comments	TUW

Disclaimer

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

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

Copyright notice

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.

EXECUTIVE SUMMARY

The TEADAL project delivers a comprehensive system for federated data sharing, designed to ensure privacy, trust, and efficiency across organizational boundaries. At the core of the platform lies a versatile and modular architecture that supports our six pilot cases: medical data sharing, access point for public transportation data, smart viticulture monitoring, efficient industry KPIs, financial data governance, and regional planning for environmental sustainability. These pilots were analyzed and documented throughout the series of deliverables D2.1, D2.2, and D2.3, with the current deliverable D2.4 marking the final step in this series. In particular, this deliverable advances on the requirements elicitation of D2.3 and categorizes them into a set of cross-cutting requirements – spanning all pilot cases – and other requirements that are only pilot specific. We then describe the core features and components of the TEADAL platform, as well as the interactions between components. Finally, we use this architectural description and the information in the technical deliverables to argue the degree to which both the cross-cutting and pilot-specific requirements are fulfilled.

Since this deliverable presents the final iteration of the TEADAL architecture, we also revise and summarize the design of the individual TEADAL nodes, including the deployment of individual nodes, as well as how to build a federation of nodes between operators. In composition, this provides a full-fledged and self-contained architectural description of the requirements emerging from the pilot-cases, the intended features of the TEADAL platform, the underlying concepts and mechanisms used for implementing this platform, the components developed in the technical WPs, as well as their interaction, and to summarize, a direct mapping that this architecture is fit for fulfilling the initial requirements.

Funded by the European Union

Table of Contents

Funded by the European Union

LIST OF FIGURES

Funded by the European Union

LIST OF TABLES

Funded by the European Union

ABBREVIATIONS

AI-DPM Artificial Intelligence-Driven Performance Monitoring
AIOps Artificial Intelligence for IT operations
AOI Area of Interest
AVM Automatic Vehicle Monitoring
AVL Automatic Vehicle Location
BPMN Business Process Model and Notation
C2B Customer to business
C2C Customer to customer
CDD Customer Due Diligence
CI/CD Continuous Integration / Continuous Deployment
DAG Directed Acyclic Graph
DLO Data Lake Owner
DSPN Data Sharing Policy Notation
ETL Extract, Transform, Load
FDP Federated Data Product
GA General Assembly
GDPR General Data Protection Regulation
GPS Global Positioning System
GTFS General Transit Feed Specification
GTFS-RT General Transit Feed Specification Realtime
IaC Infrastructure as Code
IAM Identity and Access Management
JWT JSON Web Token
KPI Key Performance Indicator
LLM Large Language Model
MDM Master Data Management
ML Machine Learning
NAP National Access Point
OHDSI Observational Health Data Sciences and Informatics
OLAP Online Analytical Processing
OLTP Online Transaction Processing
OMOP Observational Medical Outcomes Partnership
PDP Policy Decision Point
PII Personally Identifiable Information
RAP Regional Access Point
SaaS Software as a Service
sFDP Shared Federated Data Product
TSP Transport Service Provider

Funded by the European Union

1 INTRODUCTION

The TEADAL project aims to enable trustworthy, privacy-preserving, and energy-efficient data sharing in federated environments. This deliverable, D2.4, represents the final iteration of the TEADAL general architecture and consolidates the requirements, design refinements, and architectural alignment developed throughout the project. This document builds on the previous deliverables (D2.1, D2.2, and D2.3) by incorporating the final updates to both cross-cutting and pilot-specific requirements. Additionally, it expands the technical architecture to reflect the evolving needs of the pilot projects, which include healthcare, mobility, viticulture, industry, and regional environmental development. Each of these pilots has its own unique domain-specific requirements.

The TEADAL architecture introduces a modular and extensible platform designed to support secure, privacy-preserving, and efficient data sharing in federated environments. TEADAL architecture relies on two key components: the Trust Plane and the Control Plane, both of which are indispensable for enabling verifiable, accountable, and policy-compliant operations across all pilots. The Trust Plane is responsible for the systematic creation, collection, and publication of verifiable evidence associated with all data transactions and platform interactions. It ensures that each data-sharing event is auditable and aligned with legal and organisational obligations. Key components of the Trust Plane include the Catalogue, which enables metadata management and federated data discovery, and the Advocate, which records cryptographic evidence and immutably anchors it for later audits. These mechanisms are described in detail in Deliverable D5.2, while Chapter 8.
On the other hand, the Control Plane is central to orchestrating all workloads and runtime behaviors within the TEADAL federation. As described in D2.2 and D4.1, the Control Plane manages the entire lifecycle of TEADAL platform components across multiple nodes. It manages both platform-specific services, such as metrics collection, consent verification, and policy enforcement, and the computational workflows that transform raw datasets into FDPs and their contract-specific sFDPs. Moreover, the Control Plane collaborates closely with the Trust Plane, executing actions based on observed behavior and validated policies. For instance, it may automatically suspend or isolate non-compliant services based on feedback from the Trust Plane, or make intelligent workload placement decisions using performance and energy data provided by the governance components developed in WP3.

This document places significant emphasis on aligning architectural elements with the real-world demands of the pilots. It focuses on capturing and structuring pilot-specific and cross-cutting requirements, and explaining how these requirements are fulfilled by the TEADAL architecture. The interactions between architectural components and between federation nodes are also documented in detail, highlighting how TEADAL enables distributed data sharing. The architecture is positioned to support various deployment topologies and interaction patterns, including decentralised catalogue federation, cross-cluster resource sharing, and federated identity management.

The main chapters of the deliverables outline TEADAL architecture as follows:

Funded by the European Union

- Chapter 2 Requirements: It collects the technical requirements that guide TEADAL's architecture. It includes summaries of each pilot's context, shared cross-cutting needs such as privacy and access control, and domain-specific constraints.
- Chapter 3 TEADAL Features: It highlights TEADAL's fundamental strengths in architecture through three key areas: automation, optimization, and trust.
- Chapter 4 System-Level View: It outlines TEADAL's data sharing lifecycle and xsupporting infrastructure. It introduces foundational concepts such as data mesh and service mesh, the TEADAL cluster runtime, and deployment mechanisms.
- Chapter 5 Component Descriptions: It describes the main components of TEADAL's architecture, including the Catalogue, Control Plane, Trust Plane, Service Mesh, and Data Pipelines. Each component's responsibilities and integration within the overall system are highlighted.
- Chapter 6 Component Interactions: It describes how components collaborate to onboard, register, transform, enforce, and audit data access.
- Chapter 7 Interaction Between TEADAL Nodes: It details how TEADAL nodes interact across federated infrastructures, describing data product sharing, catalogue federation, identity federation, trust establishment, and resource allocation.
- Chapter 8 Deployment and CI/CD Instructions: It summarises how TEADAL nodes are deployed and maintained using GitOps and ArgoCD.
- Chapter 9 Architectural Fit for Purpose: It describes how TEADAL addresses both cross-cutting and pilot-specific requirements. It shows how architectural components support each pilots's goals and demonstrates alignment with federation principles.
- Chapter 10 Conclusion: It summarises the key insights from aligning architecture with real-world requirements and sets the stage for final system convergence.

This deliverable consolidates the final state of the TEADAL architecture, integrating the insights and refinements gained throughout the project's development. By aligning architectural principles with concrete requirements from diverse pilot domains, TEADAL establishes a robust and adaptable framework for federated data sharing. The combined use of the Trust Plane and Control Plane ensures that operations remain verifiable, policy-compliant, and efficient across all environments. Through its modular design and support for various deployment and interaction models, the architecture is well-positioned to meet real-world demands and scale across organisational and technological boundaries.

Funded by the European Union

2 REQUIREMENTS

This section presents the key requirements that guided the development of the TEADAL platform. It includes a summary of the requirements per use case as outlined in D2.3, as well as a general requirements view detailing how these requirements are incorporated into the architecture. In particular, Section 2.1 first refreshes the information and general data flow patterns of the five pilot cases; according to this information, Section 2.2 presents an overview of cross-cutting requirements that span over all five pilot cases. Finally, in Section 2.3, we present pilot-specific requirements that require dedicated attention.
Later in this deliverable, in Section 9, we will then address how the presented architecture addresses the cross-cutting requirements and those specific to pilot cases.

2.1 Overview of Use-Cases

The following section describes the architecture requirements derived from the use cases, as finalized in deliverable D2.3. From the specified pilot cases, we then extract information about cross-cutting and pilot-specific requirements.

2.1.1 Evidence-based medicine

The evidence-based medicine pilot aims to enhance data sharing and analytics in healthcare while addressing privacy restrictions and consent requirements for data processing. RIBERA SALUD will simulate federated data sharing among healthcare organisations, with TEADAL tools facilitating compliance with privacy regulations, establishing trust between organisations, and managing data access based on individual consent.

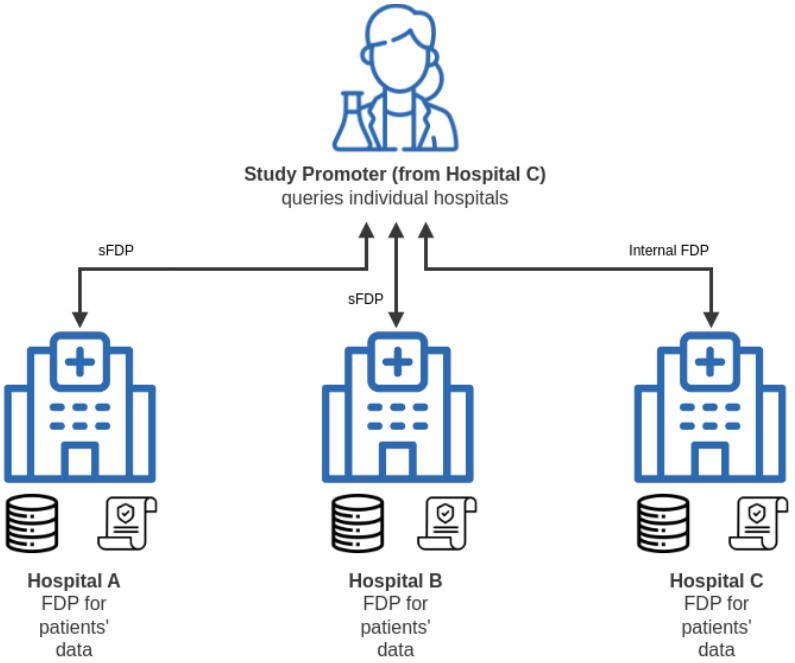


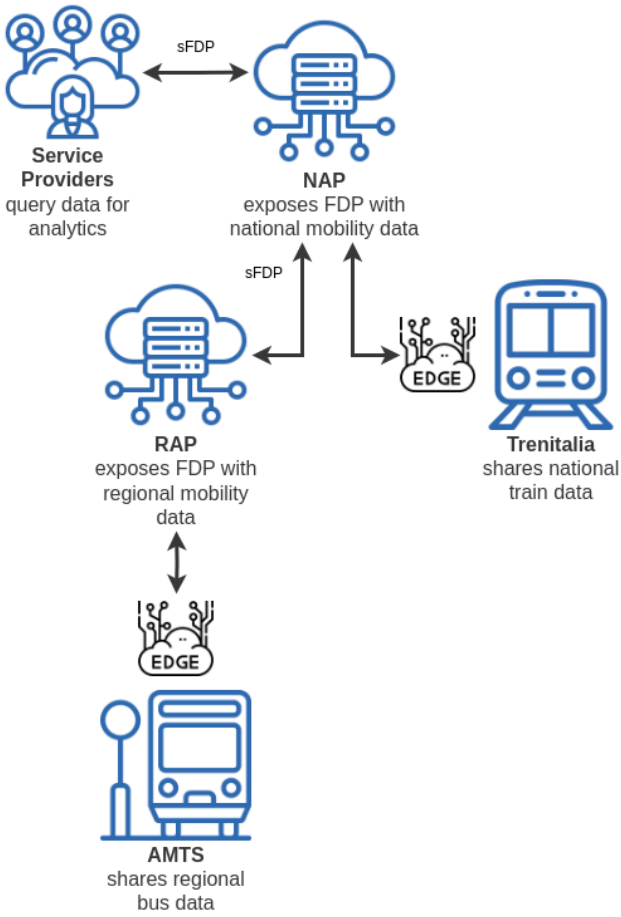
FIGURE 1: OVERVIEW OF THE EVIDENCE-BASED MEDICINE USE CASE PILOT.

Funded by the European Union

The architecture must provide a platform that enables the secure exchange of patient data between healthcare organisations. To this end, It must enable a field-specific ontology in the catalogue and automate study setup processes to reduce manual effort. The data lake must be configurable to meet data-sharing agreements, while offering a variety of computation methods, including privacy-preserving mechanisms, and providing orchestration for these computations. Additionally, the architecture must ensure GDPR-like compliance through tools for anonymization, and privacy checks. To support multi-step studies, it should include mechanisms for persistence, allowing workflows to be maintained and revisited over time. These features must collectively ensure operational efficiency, compliance, and adaptability in a federated environment.

2.1.2 Mobility

The mobility pilot integrates TEADAL technologies for data sharing among four Italian public transport domain stakeholders (AMTS, Trenitalia), and entities in charge of collecting and integrating the data (RAP, NAP) by simulating a system to centralize, manage and provide unified transport data. While the NAP is defined at EU level, RAP acts as an middle layer between public transport stakeholders and the NAP. This initiative addresses challenges in cross-border cooperation and urban data compilation by implementing a three-tiered structure, with regions gathering data from local transport operators (AMTS Catania) through RAP nodes, and sharing it via the NAP as a unified access point for public transport data in Italy.



Funded by the European Union

FIGURE 2: OVERVIEW OF THE MOBILITY USE CASE PILOT.

The architecture must address specific requirements to support efficient and dynamic data access within the TEADAL federation and must enable the exchange of mobility data across different platforms. Therefore it must enable seamless federation access, ensure only updated data is available through dynamic dataset management, and avoid data duplication to prevent outdated or inconsistent information. These measures ensure reliable and efficient data sharing.

2.1.3 Smart Viticulture

The pilot for smart viticulture addresses challenges faced by vineyard operators due to climate change and regulatory demands by enhancing Terraview systems (TerraviewOS and Aquaview) with TEADAL tools to enable data sharing across neighboring vineyards, improving monitoring of changes such as water moisture profiles for proactive decision-making while preserving the confidentiality of the data.

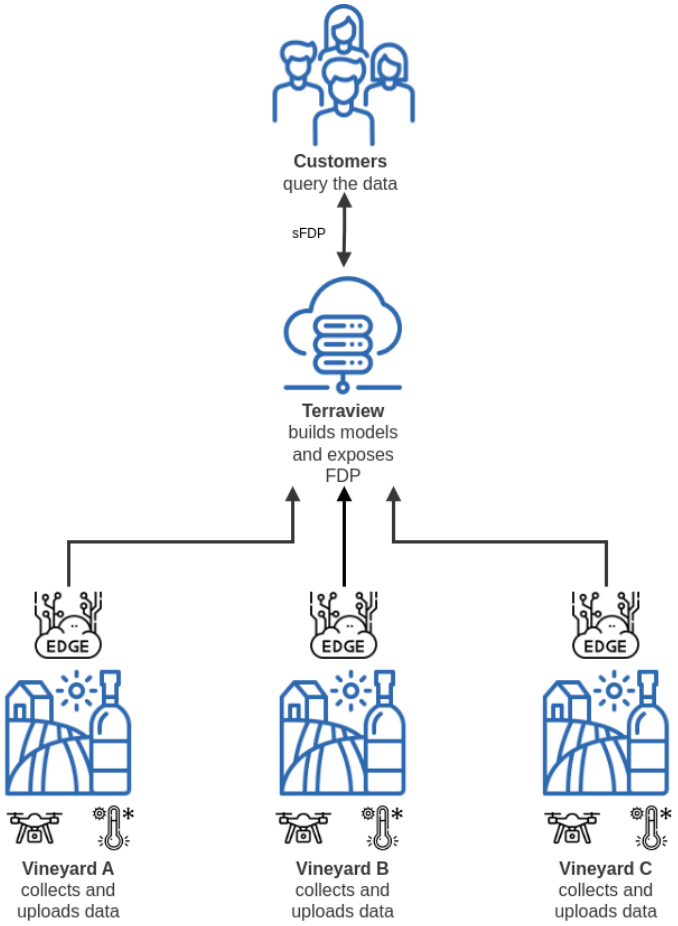


FIGURE 3: OVERVIEW OF THE SMART VITICULTURE USE CASE PILOT.

This use case requires a platform that supports the exchange of agricultural data such as soil moisture analyses. The architecture therefore must support the placement of datasets across

Funded by
the European Union

the computing continuum based on pilot-specific criteria, such as geography. Additionally, it should enable the federation of data lakes at all levels of the continuum, with a particular focus on supporting integration at the edge. These capabilities ensure flexibility and adaptability for diverse use-case requirements.

2.1.4 Industry 4.0

The industry 4.0 pilot focuses on automating the calculation of unified KPIs for ERT Group's facilities in different countries (Portugal and Czech Republic), integrating facility-specific data to align with the group's standardized metrics for operational, commercial, and quality management.

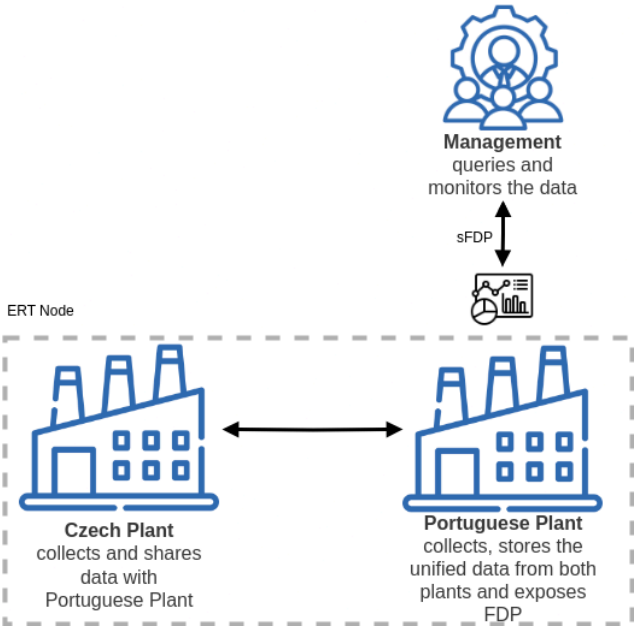


FIGURE 4: OVERVIEW OF THE INDUSTRY 4.0 USE CASE PILOT.

The architecture for the Industry 4.0 use case must support data normalization across distributed databases to ensure consistency. It should provide an easy-access interface, such as APIs, to facilitate data querying for report generation. Additionally, a harmonized protocol is needed for data collection, processing, and sharing across plants, departments, and teams to ensure interoperability.
The system must also support data stored in multiple locations and enable data ingestion directly from SQL databases to streamline integration with existing systems. These features ensure efficient and standardized data management for Industry 4.0 applications.

2.1.5 Financial Data Governance

Funded by
the European Union

The BOX2M pilot case addresses the financial management and optimisation of renewable energies in the context of fluctuating energy prices. Instead of the originally planned ING pilot project, an alternative, financially focussed integration scenario is being pursued. The aim is to use TEADAL components to implement AI-supported forecasting models, a national grid integration system and data-reducing aggregation mechanisms, among other things. Data products (FDP/SFDP) will be used to measure the financial value of photovoltaic systems, optimise production capacities and analyse the costs of decarbonisation in real time. In addition, methods for financial modelling and ROI assessment are integrated into daily operations.

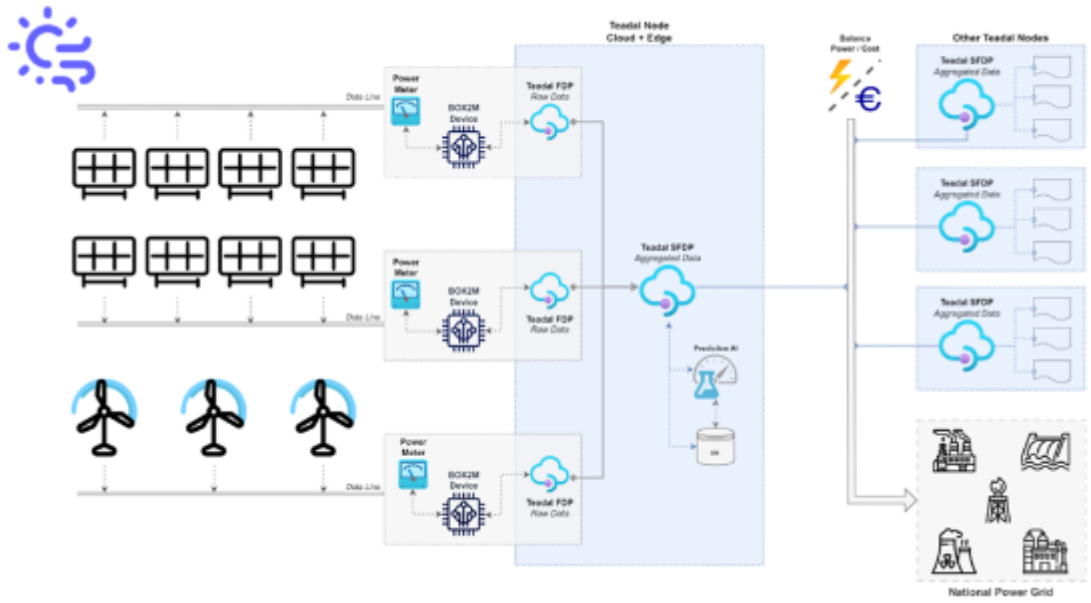


FIGURE 5: OVERVIEW OF THE FINANCIAL USE CASE PILOT.

2.1.6 Regional Planning for Environmental Sustainability

The pilot aims to integrate sensor data on building energy profiles from BOX2M and environmental monitoring data managed by RT, combining public and private data to reconstruct energy profiles, analyze local efficiency, and assess air quality trends using open weather and air quality data.

Funded by the European Union

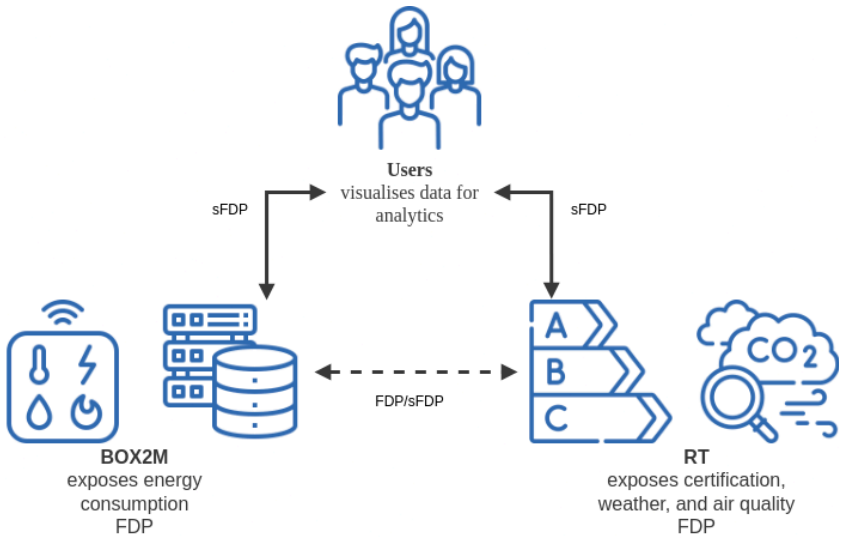


FIGURE 6: OVERVIEW OF THE REGIONAL PLANNING USE CASE PILOT.

The use case for Regional Planning and Environmental Sustainability therefore requires an architecture that defines an ecosystem where RT serves as the central entity and BOX2M acts as a key data provider, with RT being a primary consumer of this data. The solution must include a logical component to federate RT SIERT information system datasets, RT open data, and BOX2M sensor data without replication to ensure efficiency and consistency.

Additionally, the architecture must establish dedicated nodes for RT and BOX2M to support their respective roles. These requirements collectively support a robust and efficient framework for environmental planning.

2.2 Cross-Cutting Architectural Requirements

While each pilot case targets a unique problem, there are certain characteristics they share, which you could also call cross-cutting concerns. In the following, we provide an overview of the base requirements that the pilots have in common, and then provide more specific requirements that are characteristic for the individual pilots.

Policies and privacy

General guidelines for handling data must be defined, in particular for storing, copying, forwarding and deleting data. The guidelines must contain clear specifications on confidentiality and the protection of sensitive information and take into account data protection regulations e.g. GDPR, ISO 27001 and IATF16949 (C1-Pol01). In addition, rules for the control and management of data access must be defined, including access rights (read, edit, use, forward) as well as specifications for the storage and traceability of consents for the use and disclosure of data (C1-Pol02, C1-Pol03). Policies should be established if one of the federated data lakes has stretched components on a public cloud (i.e. Amazon). Table 1 summarizes cross-cutting requirements related to policies and privacy.

Funded by the European Union

ID	Description
C1-Pol01	Definition of clear rules for storing, copying, forwarding and deleting data in compliance with regulations.
C1-Pol02	Ensuring the confidentiality of sensitive information and regulations for access rights (read, edit, use, forward).
C1-Pol03	Requirements for the documentation and traceability of consents to data use and disclosure.

TABLE 1: CROSS-CUTTING REQUIREMENTS: POLICIES & PRIVACY

Data Processing & Sharing

General technical and organisational mechanisms must be provided to enable efficient, secure and standardised processing and sharing of data, ensuring that only authorised data is used (C-Dps01). This includes support for various data storage systems, data types (e.g. XML, JSON etc.) and databases (e.g. SQL) as well as methods for normalisation, cleansing (data sanitisation) and anonymisation of data (C-Dps02). Different processing modes such as real-time and batch processing must also be supported. In addition, a harmonised standard must be implemented for the exchange and processing of data between internal and external stakeholders, including the possibility of secure data transfer with third parties like external service providers or partners (C-Dps03). Table 2 summarizes cross-cutting requirements related to data processing and sharing.

ID	Description
C2-Proc01	Provision of technical mechanisms for efficient, secure and standardised data processing and forwarding (batch and real-time processing).
C2-Proc02	Support for different data storage systems (e.g. SQL) as well as methods for normalisation, data sanitisation and anonymisation of data.
C2-Proc03	Harmonisation of data processing and data exchange between internal and external stakeholders with secure transfer to third parties.

TABLE 2: CROSS-CUTTING REQUIREMENTS: DATA PROCESSING & SHARING

Data access & cataloguing

A central data platform or catalogue is required to make data easy to find and retrieve (C-Dc01). Each of the use cases requires a clearly defined and transparent structure for data access and a clear assignment of metadata with one or more tags describing the data contained (C-Dc02, C-Dc03). Table 3 summarizes cross-cutting requirements related to data access and cataloguing.

Funded by the European Union

ID	Description
C3-Acc01	Provision of a data catalogue and domain-specific ontologies to make available data clear and efficient to find.
C3-Acc02	Transparency and traceability of data access, e.g. insight into which data was accessed or used by whom.
C3-Acc03	Assignment of Metadata with one or more tags describing the data contained.

TABLE 3: CROSS-CUTTING REQUIREMENTS: DATA ACCESS & CATALOGUING

2.3 Pilot-Specific Architectural Requirements

Ideally, a single solution can be wrapped over a manifold of use cases without additional customization effort. While the cross-cutting concerns were present in all five pilot cases, the reality is that our real-world pilot cases present numerous heterogeneous requirements that are very specific to their domain. In the following, we outline our findings and describe the requirements for each pilot. Together with the cross-cutting requirements, Section 9 will answer how these are addressed by TEADAL.

2.3.1 Evidence-based medicine

The 'Evidence-based Medicine' pilot aims to conduct medical studies in a more efficient, data protection-compliant and sustainable manner. The automation of study preparation (P1-Arch02) and the provision of a comprehensive data catalogue that provides study initiators with targeted information on the availability of suitable patient groups (P1-Mgmt01) play a central role in this. To comply with data protection requirements, various procedures for data anonymisation and sanitisation are supported, whereby feedback with the data holders is made possible (P1-Mgmt02). The architecture also provides technical components for the implementation of data protection-compliant access mechanisms, for example for checking consent or controlling anonymity (P1-Arch06).

The study bureaucracy is relieved by targeted functionalities (P1-Gen05), while at the same time sustainability metrics can be recorded for all actions within the platform (P1-Gen03). Study initiators receive tools for assessing the significance of their study, e.g. via targeted queries on the number of patients affected (P1-Mgmt03), as well as rules for legally compliant access approval (P1-Gen06). The implementation of multi-stage studies is supported by persistent technical structures (P1-Arch09). An overview of the specifically supported requirements and their technical implementation is shown in Table 4.

Req. ID	Description
P1-Arch02	The architecture has to facilitate as much as possible the automation of the process to set up the study.

Funded by the European Union

P1-Mgmt01	Provide a data catalogue that tells the study promoter where he will find the patients that it can study.
P1-Mgmt02	Provide a variety of data sanitization methods with callbacks to data owners (some in TEADAL, others are plugged in by use-cases)
P1-Gen03	Offer means to measure sustainability related metrics for each action taken by TEADAL.
P1-Gen05	Facilitate the bureaucracy of a study proposal.
P1-Mgmt03	Allow to extract information about the amount of patients affected (SELECT COUNT) to know the strength of the study.
P1-Gen06	Allow to approve access to personal data to certain "study promoters" by comparing basic rules.
P1-Arch06	Provide technical components to enable GDPR like compliant data access, e.g., tools to anonymize, tools to check anonymity or tools to check informed consent.
P1-Arch09	Provide some kind of persistence for multiple-step studies.

TABLE 4: PILOT-SPECIFIC REQUIREMENTS: EVIDENCE-BASED MEDICINE

2.3.2 Mobility

The ‘Mobility’ pilot is dedicated to the federated provision and use of mobility data, taking into account transparency, data protection and federal controllability. It is based on access to open and domain-specific data sources such as the road network of OpenStreetMap (P2-Gen01), public transport timetables via national access points (NAP) (P2-Gen02) and live data on arrival times of individual operators (P2-Gen03). Price-sensitive information such as average fares can also be integrated without directly providing a booking option (P2-Gen06).

To ensure data sovereignty, transport service providers (TSPs) are given the option of protecting certain data from access by other TSPs (P2-Privacy01) and deciding independently which data records may be shared (P2-Mgmt03). At the same time, federated access to relevant data from other providers is provided, provided this takes place within the RAP or NAP structure (P2-Mgmt04, P2-Arch01). The technical architecture ensures that only up-to-date data records are available at all times (P2-Arch02), while access and data queries remain traceable at national level (P2-Mgmt02). This enables transparency across the entire data flow (P2-Gen07) as well as mechanisms to prevent unwanted data transfer (P2-Policy03). Table 5 systematically summarises the individual requirements and their respective implementation.

Req. ID	Description

Funded by the European Union

P2-Gen01	To access Open Street Map for the street network of the city/region involved.
P2-Gen02	To be able to take information from the NAP about the schedules of all public transport operators in a specific area.
P2-Gen03	To access data from the operator on the stop arrival times.
P2-Gen06	Include also information about the average price of the trip, without the need to give the opportunity to book the trip.
P2-Gen07	To offer transparency of data access across NAP, RAP, Local, etc. (e.g. allow the NAP to see what happened to the data downstream).
P2-Privacy01	Each Transport Service Provider (TSP) should be able to prevent a subset of his data to be shared with other TSPs (e.g. competitors).
P2-Arch01	To allow for joining the RAP or NAP TEADAL federation to access available data.
P2-Arch02	To provide only updated data, i.e. datasets can be dynamic so that only last versions should be available.
P2-Mgmt02	The TSP should have a way to check who accessed or requested data at the national level.
P2-Mgmt03	The TSP should have a way to decide whether a dataset should be shared in the federation.
P2-Mgmt04	Each TSP should be able to access data belonging to other TSPs through the RAP or NAP.
P2-Policy03	Enable a mechanism to prohibit access or sharing of data that should not be available to the NAP or RAP.

TABLE 5: PILOT-SPECIFIC REQUIREMENTS: MOBILITY

2.3.3 Smart Viticulture

The ‘Smart Viticulture’ pilot focuses on data-supported exchange in the wine industry and aims to support various data flows between consumers, businesses and partner companies. This includes direct exchange between consumers and businesses (C2B) (P3-Gen01) as

Funded by the European Union

well as between agricultural businesses (C2C), for example between two wineries (P3-Gen02). In addition, the transfer of data to external business partners such as insurance companies should be supported (B2B) (P3-Gen03), for example for risk assessment or crop evaluation.
In order to make the data efficiently usable, it could be contextually placed in the Continuum based on pilot-specific factors such as geographical location (P3-Arch01), which enables flexible yet targeted data availability along the real conditions of viticulture. The specific requirements and their implementation are summarised in Table 6.

Req. ID	Description
P3-Gen01	Support C2B.
P3-Gen02	Support C2C (e.g. Vineyard operator-A to Vineyard operator-B).
P3-Gen03	Support B2B sharing (e.g. Terraview to insurance company).
P3-Arch01	The datasets should be placed in the continuum according to pilot-case specific notions, including geography.

TABLE 6: PILOT-SPECIFIC REQUIREMENTS: SMART VITICULTURE

2.3.4 Industry 4.0

The ‘Industrial 4.0’ pilot addresses the efficient, standardised and compliant use of operating data to create meaningful reports and key performance indicators (KPIs). The focus is on the development of an easily accessible interface for queries, for example via an API, which enables access to the underlying data for reporting purposes (P4-Arch02). At the same time, a harmonised protocol for recording, processing and forwarding information should be established that can be used across locations and departments (P4-Arch03).

To ensure consistency and comparability, information from different production sites should be standardised (P4-Gen01), processed and stored via the existing infrastructure of the partner ERT (P4-Gen05). Data protection-compliant handling of visualised key figures can take place via defined rules for aggregation or obfuscation for users with restricted access rights (P4-Policy03). A detailed overview of the requirements and their implementation can be found in Table 7.

Req. ID	Description

Funded by the European Union

P4-Arch02	Providing an easy-access interface for querying the data to generate the reports (e.g., API for accessing data).
P4-Arch03	Create an harmonised/standard protocol for data collecting, processing and information sharing with different plants, departments and teams.
P4-Gen01	Standardising information coming from different plants.
P4-Gen05	Ensure that data is processed and stored using ERTs infrastructure.
P4-Policy03	Define rules (aggregation, obfuscation, etc.) for visualising data, KPI values and/or KPI categories, reports that are accessed by users with restricted rights.

TABLE 7: PILOT-SPECIFIC REQUIREMENTS:INDUSTRY 4.0

2.3.5 Financial Data Governance

The pilot 'Financial Data Governance' aims to enable the processing and evaluation of sensitive financial and production data in compliance with data protection regulations and economic efficiency. The resulting requirements can be seen in Table 8. The aim is to ensure privacy protection, particularly for computationally intensive tasks such as the creation of KYC models, through the use of Trusted Execution Environments (TEEs) (P5-Privacy01). TEEs could be dynamically integrated into privacy-friendly processing pipelines (P5-Arch01), with metadata descriptions serving as the basis for intelligent data flows in the mesh (P5-Mgmt04). To ensure the traceability and auditability of all processing steps, it is planned that these will be cryptographically secured and evidence systematically documented (P5-Mgmt01). National regulations and certifications by authorities should regulate the processing and transfer of sensitive data, whereby defined policies are combined with the characteristics of the data products (P5-Policy01). In addition, the establishment of a coupled FDP-SFDP structure is planned, which should make it possible to link production data with market price information and analysis infrastructure in order to assess profitability and economic potential, for example (P5-Gen01). Market price forecasts at national level are also to be integrated in order to support data-driven optimisation in the energy market (P5-Gen02).

Req. ID	Description
P5-Privacy01	Privacy should be preserved for computation tasks (for KYC model creation at least)
P5-Policy01	Defined policies and data properties should be combined for applying respective the data product policies

Funded by the European Union

	Data transfer and the processing of sensitive data must be controlled and authorised by national guidelines (including certification by authorities).
P5-Mgmt01	Data processing must be traceable and auditable; every transfer and processing must be backed up with evidence. After processing, the outputs must be securely stored and the TEE must be completely deleted (incl. memory footprint).
P5-Mgmt04	Metadata description of the data should be used as an input in the mesh for smart data movements
P5-Arch01	It must be possible to integrate the use of Trusted Execution Environments (TEEs) into the privacy-preserving data pipelines and activate them dynamically.
P5-Gen01	Establishment of a coupled FDP-SFDP structure to evaluate production data, ROI and financial viability by combining edge devices and analytics infrastructure.
P5-Gen02	Integration of national market price data for data-driven optimisation of energy marketing based on forecasts and key financial figures.

TABLE 8: PILOT-SPECIFIC REQUIREMENTS: FINANCIAL DATA GOVERNANCE

2.3.6 Regional Planning

The pilot 'Regional planning' deals with the federated linking and data protection-compliant analysis of administrative data (RT) and dynamic sensor data (BOX2M) to support regional decision-making processes. The aim is to combine the use of both data sources in order to provide users with well-founded analyses and a data-based basis for decision-making (P6-Gen01, P6-Gen02, P6-Gen03). Processing takes place in aggregated form and must take into account data protection thresholds, such as a minimum number of three entries per analysis (P6-Privacy02). In addition, no confidential building data from the RT data pool may be visible (P6-Privacy03), while a three-stage consent model should regulate the processing and forwarding of data by BOX2M (P6-Privacy05). Table 9 summarizes cross-cutting requirements related regional planning pilot use case.

The planned architecture envisages a federated system with clear role allocations, in which RT acts as the central data node and BOX2M acts as the data provider (P6-Arch01). A logical federation component is to connect SIERT data, open data sources and sensor data from BOX2M, without physical data replication (P6-Arch02). In addition, a dedicated node is to be defined for each of the two partners to enable separate administration (P6-Arch03). To ensure data quality, it is planned to mark sensitive data areas as invalid for a certain period of time if, for example, sensors have been manipulated wilfully or negligently (P6-Mgmt01).

Funded by the European Union

Req. ID	Description
P6-Gen01	Data about plants and buildings coming from the Tuscany Region must be enriched with BOX2M dynamic data coming from sensors in an aggregated territorial perspective.
P6-Gen02	The aggregated data must be used for decision support system development.
P6-Gen03	Users must benefit from the combined use of RT and BOX2M data and from the analytics produced on top of these datasets.
P6-Privacy02	The aggregation of data has a minimum threshold of 3 units. No analysis can be performed if less than 3 records are affected.
P6-Privacy03	The system must forbid the users to see confidential data about buildings and plants stored in the RT data lake.
P6-Privacy05	The must be 3 level of consent referring to P6-Privacy04: BOX2M is allowed to collect building's data and move them on cloud. (mandatory) Plant owners give to BOX2M the consent to analyse data, but not to share them (optional) Plant owners give to BOX2M the consent to analyse data and share them (optional)
P6-Arch01	The solution must define a sort of ecosystem where RT is at the centre and BOX2M is one of the actors who is providing data. RT is one of the main consumers of data.
P6-Arch02	The solution must define a logical component which federates SIERT dataset, open data and BOX2M sensor data without replication.
P6-Arch03	The solution must define one node for RT and one node for BOX2M.
P6-Mgmt01	The solution must provide the ability to mark certain data elements as invalid in a particular interval, since some field sensors could be altered by malice, neglect or vandalism

TABLE 9: PILOT-SPECIFIC REQUIREMENTS: REGIONAL PLANNING

Funded by the European Union

The analysis of requirements across the six TEADAL pilot cases demonstrates the diverse and often domain-specific demands placed on the platform architecture. While certain cross-cutting concerns such as privacy, secure data processing, cataloguing, and traceability are shared among all pilots, each use case also presents unique functional and technical requirements tailored to its operational context, regulatory environment, and stakeholder needs. From privacy-preserving medical studies and federated mobility data sharing to smart agricultural practices, industrial reporting, financial forecasting, and regional environmental planning, the TEADAL architecture must offer both flexibility and structural consistency.

By systematically categorizing general and pilot-specific requirements, this section provides the foundation for understanding how architectural design decisions are informed and justified. In Section 9, the document will revisit these requirements and demonstrate in detail how TEADAL's architecture, components, and workflows address them.

Funded by the European Union

3 TEADAL FEATURES

The features view outlines the key features that significantly influence the architecture's structure. These features are derived from the requirements identified during the requirements elicitation process for each use case, carried out in the project's initial iteration and finalised in the following iterations (see deliverable D2.1, D2.2, D2.3). Additionally, the design incorporates the project's overarching non-functional requirements. The primary objectives that the architecture, its tools, and the integrated processes must achieve can be categorized into three core aspects: (1) automation of data sharing, (2) optimization of data sharing, and (3) establishing trust among partners.

3.1 Automation of Data Sharing

TEADAL strives to automate data-sharing processes across organisations to the greatest extent possible. This includes a service-oriented design including a data mesh, dynamically preparing shareable data, simplifying infrastructure management, defining policies efficiently, and enabling data discovery, realized by the TEADAL platform (Figure 7). The following key aspects describe how the TEADAL platform automates and streamlines data sharing:

- **Dynamic Shareable Data:** Federated Data Products (FDPs) must adhere to standardized rules within the federated governance framework. FDPs act as pointers to data accessible via REST APIs, including associated policies and computational capabilities. To support tailored sharing agreements, FDPs are not accessed directly by data users but through the Shared Federated Data Products (sFDPs) that TEADAL Platform creates, deploys and manages to instantiate data sharing agreements between the data owners and the data consumers.
- **Simplified Data Lake Management:** TEADAL adopts a service-oriented architecture to abstract complex data lake operations. Rather than relying on a specific serverless stack, the platform allows for declarative service definitions and flexible deployment onto Kubernetes-based federation nodes. This abstraction enables users to define, deploy, and manage data services, in the form of sFDPs, without the need to attend to infrastructure-level concerns.
- **Streamlined Policy Definition:** Policies for data sharing are translated from human-readable formats into machine-executable code, ensuring that both data owners and enforcement systems can seamlessly understand and apply them. TEADAL's architecture supports this process through dedicated software components.
- **Data Discovery:** To ensure privacy and confidentiality, the TEADAL Data Catalogue provides federation members visibility into available FDPs without exposing the underlying data, enabling efficient discovery across organizations.
- **LLM-Assisted Automation:** To further accelerate automation and reduce manual effort, TEADAL explored the possibilities to integrate Large Language Model (LLM) technologies at key points in the data sharing lifecycle. For example, the ASG-tool enables automatic generation of sFDPs from data sharing agreements, using LLMs to interpret the agreement text and produce structured, deployable configurations.

Funded by the European Union

Another example is a custom policy authoring web application that uses LLMs to translate natural-language policy intents into executable Rego policies, supporting transparent review and fine-tuning by data owners. LLMs are also applied in telemetry analysis via the AI-DPM subsystem.

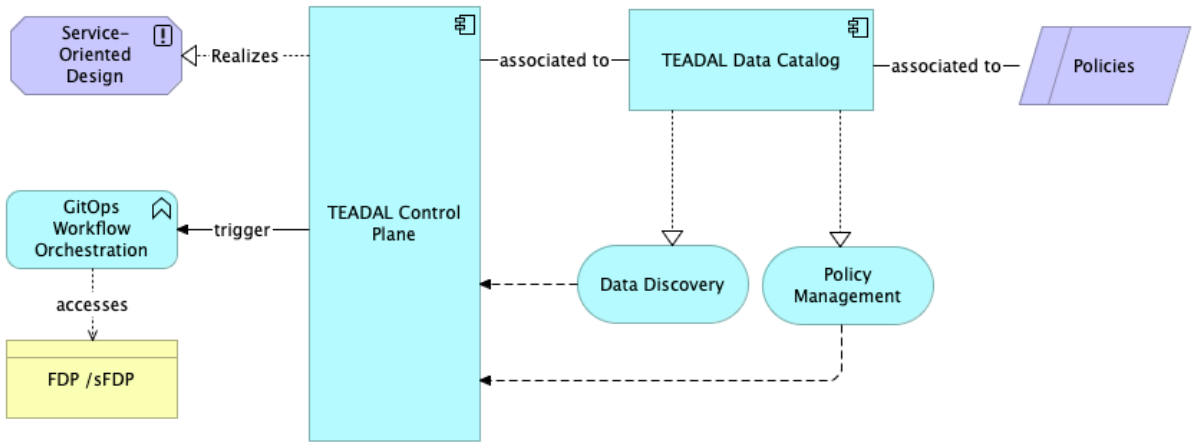


FIGURE 7: SERVICE-ORIENTED ARCHITECTURE MANAGED BY CONTROL PLANE

3.2 Optimization of Data Sharing

Inter-organizational data sharing involves transforming shared data based on attached policies, a process that generates what is known as data friction. To address this, TEADAL's architecture should incorporate performance monitoring tools to reduce this friction and optimize the placement of data and computations across its distributed data lake (Figure 7). Using its control plane, TEADAL can enable efficient processing closer to the data source by:

- **Minimizing Overhead (Friction)**: TEADAL reduces inter-organizational data-sharing friction by managing additional data processing steps transparently. These steps, monitored by TEADAL's control plane, include performance measurement and friction control mechanisms that streamline the process.
- **Optimizing Data and Computation Placement (Gravity)**: By stretching data lakes across the computing continuum, TEADAL leverages its control plane to optimize the placement of data and computation tasks. This ensures efficient use of resources, balancing proximity to data sources with computational constraints.
- **Improving Energy Efficiency**: TEADAL integrates technologies to minimize energy use by deactivating unused services and infrastructure when not needed.

Funded by the European Union

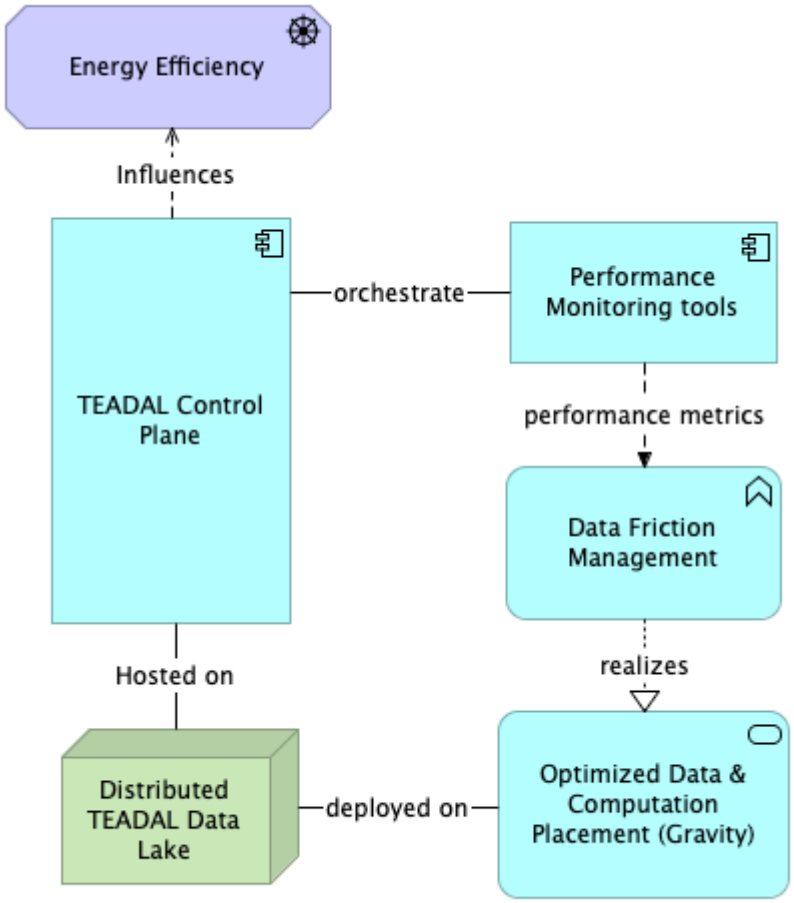


FIGURE 8: DATA FRICTION MANAGEMENT FOR OPTIMAL DATA SHARING.

3.3 Trust in Data Sharing

Trust is a fundamental requirement for data sharing, achieved through strict policy enforcement and verification. To uphold confidentiality and privacy, data visibility is managed and refined using privacy-preserving computations. The trust plane plays a key role by continuously monitoring and verifying these processes. Meanwhile, the architecture leverages its control plane and blockchain technology to ensure data integrity, establish provenance, and enable full traceability and audits throughout the data lifecycle (Figure 9). The following list present key aspects ensured by TEADAL architecture:

- **Policy Enforcement and Verification:** Service Mesh proxies attached to FDPs enforce and verify policies by intercepting requests and validating them against predefined rules. Blockchain technology in the Trust Plane ensures traceability and supports the verification of these processes.
- **Confidentiality and Privacy:** By using FDPs and sFDPs, TEADAL allows data owners to define data visibility for consumers. Confidentiality is further reinforced

Funded by the European Union

through privacy-preserving computation pipelines and Service Mesh capabilities, with the Trust Plane monitoring all activities for compliance.

- **Data Integrity and Provenance:** TEADAL provides end-to-end traceability through its control plane and blockchain technology, capturing evidence of data access and manipulation throughout its lifecycle. This allows federation members to conduct audits and ensure compliance with organizational and regulatory standards.

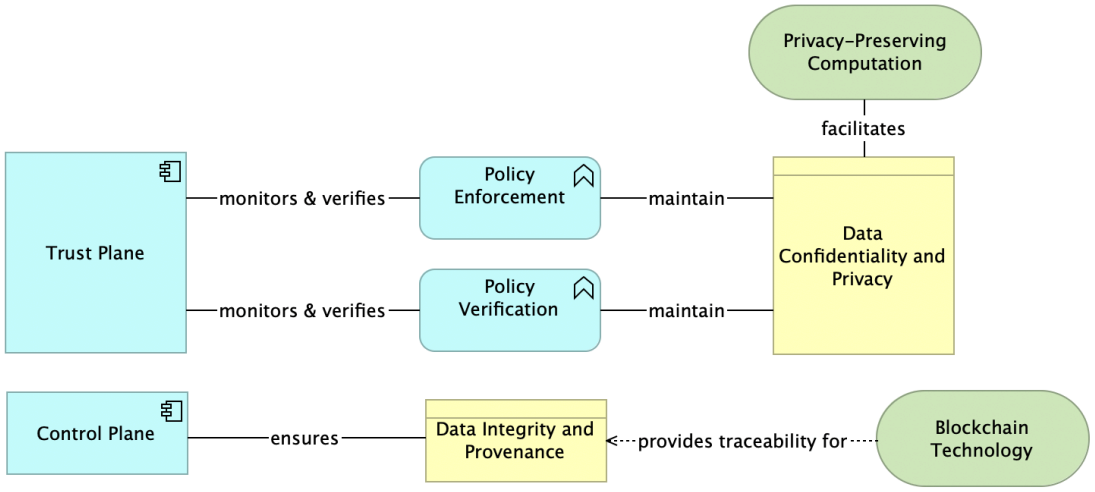


FIGURE 9: ESTABLISHING TRUST THROUGH CONFIDENTIALITY AND PRIVACY

The TEADAL platform offers a comprehensive set of features that support secure, efficient and trusted data exchange in federated environments. The focus on automation reduces manual effort and simplifies the lifecycle of data products through intelligent orchestration, policy translation and LLM-driven workflows. The optimisation features ensure that data and computations are efficiently placed in the infrastructure, reducing latency, energy consumption and operational overhead. Most importantly, TEADAL builds trust through verifiable policy enforcement, privacy mechanisms and cryptographically anchored audit trails. Together, these features enable organisations to collaborate transparently and responsibly while maintaining control over their data.

Funded by the European Union

4 SYSTEM-LEVEL VIEW

This section presents the basic architectural principles that determine the design of TEADAL at system level. It begins with an explanation of the underlying concepts of the data network and the service network, which form the conceptual basis for decentralised data management and secure service communication in the federation. These concepts are then specifically applied to the TEADAL architecture to support privacy-friendly, traceable and policy-driven data sharing. Following this conceptual foundation, this chapter describes the actual implementation of the data sharing lifecycle in TEADAL, including the creation, publication, sharing and eventual decommissioning of data products. The following sections describe the runtime structure of a TEADAL node, explain the deployment model using GitOps and show how services and data products are orchestrated in federated environments.

4.1 Underlying Concepts

This section focuses on two fundamental concepts in data sharing and service provisioning that are applied within TEADAL, namely data mesh and service mesh. For each of them, it is highlighted why they were included and how they are instantiated within TEADAL.

Data Mesh is a design principle that emphasises decentralised data governance and lifecycle management in large-scale organisations. Key concepts include defining the minimal unit of shareable data as a data product by domain experts, ensuring domain ownership of data, managing the data lifecycle through a self-service platform, and implementing federated computational governance with automated policy enactment. TEADAL extends these principles to interactions between organisations, introducing the federated data product (FDP) as the minimal unit of shareable data, addressing associated challenges in cross-organization data sharing.

Service Mesh uses layers of proxies connected to TEADAL to intercept communications, enabling features like security, policy enforcement, and traceability. In a service mesh, proxies handle both inbound and outbound service communications, isolating services from each other and the network. This allows proxies to inspect, route, and possibly alter service requests and responses, enriching functionality without altering service code. TEADAL's service mesh enhances data security, tracks FDP and SFDP life cycles to produce verifiable evidence, and improves observability of complex metrics such as gravity and friction.

4.2 Sequential Data Sharing Processes

This section summarises the data sharing process implemented by TEADAL and the various types of artefacts that are created throughout this process to facilitate the data sharing. In particular, this focuses on what a data product is – inspired by the data mesh, the smallest unit in TEADAL's data sharing process – and what are the phases through which a data product goes.

The high-level conceptual view of TEADAL's architecture includes multiple components and entities, which are detailed below. Further, it contains their responsibilities, communication interfaces, and interactions to fulfil requirements. Although most of them were introduced in

Funded by the European Union

previous deliverables, we summarise the data exchange process at this point to verify the extent to which the actual implementation matches it.

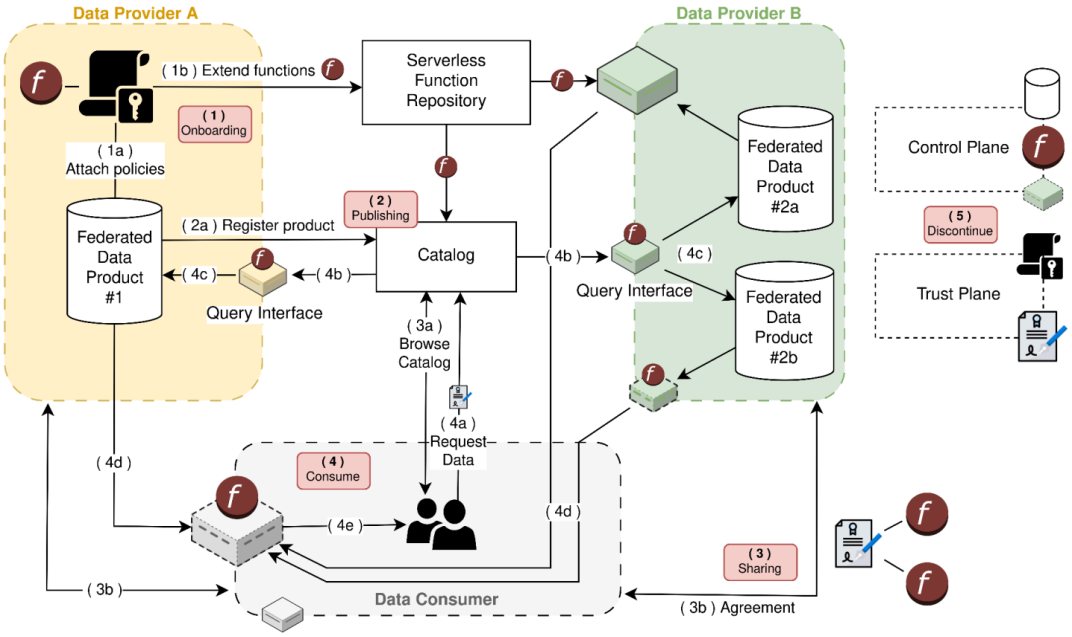


FIGURE 10: CONCEPTUAL VIEW OF TEADAL'S ARCHITECTURE

TEADAL outlines a five-phase lifecycle for a Federated Data Product (FDP) which are also included in Figure 10:

1. **Data Onboarding**: Providers prepare their data products, i..e, extending them with policies and finding a respective storage location in the federation. This transforms the data products into FDPs with policies attached.

2. **Publishing**: The data providers register the FDPs in a federation-wide and decentralised catalogue, which allows potential data consumers within the federation to browse the catalogue and discover FDPs based on their custom requirements.

3. **Sharing**: Data provider and consumer agree on the terms for data sharing through a contract, which defines the data processing pipelines, creating a Shared Federated Data Product (SFDP), a derived instance of the FDP that is provided for the consumer.

4. **Consumption**: Data consumers request access to the SFDP by providing their sharing agreement, which is validated and executed by the trust and the control plane, transforming data as per policies before consumption.

5. **Discontinue**: Agreements end due to various conditions, ceasing access and releasing associated resources using the control plane's data lineage capabilities.

These phases ensure controlled and efficient data sharing between organisations, supported by TEADAL's core components like FDPs, the catalogue, the control plane, and the trust

Funded by the European Union

plane. Below, we now summarise two of the core components in this architecture, the FDP and the SFDP in more detail.

A Federated Data Product (FDP) is a data product according to the data mesh principle that is shared between members of a federation. The FDP allows data to be stored on provider's premises or federation resources, managed by the control plane. After policies are registered in the onboarding phase, a data product is registered in the Data Catalog and federated within TEADAL. Each FDP undergoes a custom transformation process, resulting in a Shared Federated Data Product (SFDP) tailored to a usage agreement. This process is further detailed in Figure 11, where the FDP-SFDP pipeline provides an SFDP for each consumer. On the left, it is visible how the raw data product is combined with data sharing policies to create the FDP. For each particular consumer, the pipeline then creates a derived view into the FDP, which ensures that specific policies and requirements from their data-sharing contract are fulfilled.

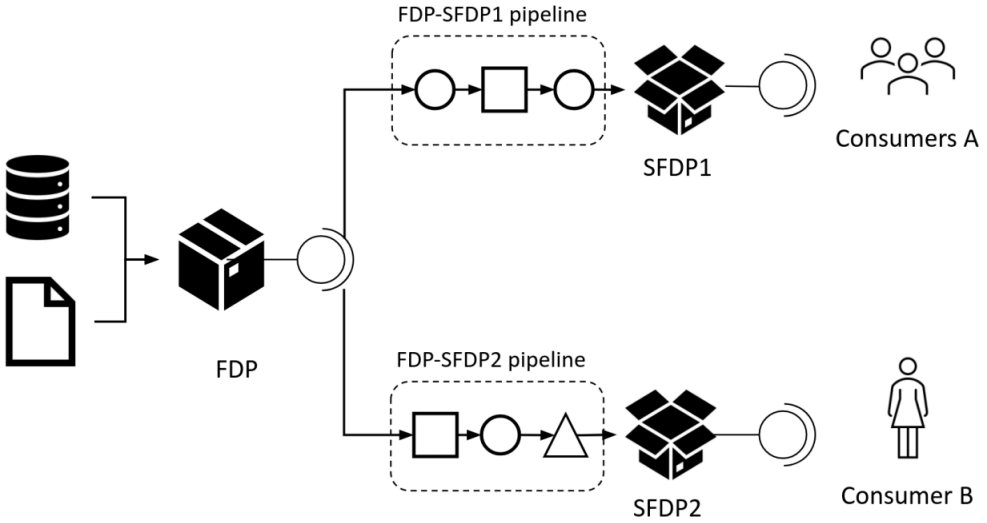


FIGURE 11: FDP TO SFDP PIPELINE BETWEEN DATA OWNER AND CONSUMER

4.3 TEADAL Node Runtime

A TEADAL node includes both the hardware and software deployed to run an instance of a TEADAL data lake. Notably, the software implements the various TEADAL tools and services that allow multiple TEADAL nodes to be joined in a federation where producers and consumers can share data in a trustworthy and secure way, according to agreed-upon governance, privacy and energy-efficiency policies[1]. These tools and services are part of each TEADAL cluster whereas hardware, data products and corresponding data services

[1] Please refer to Deliverable 2.2 "Pilot Cases' Intermediate Description And Initial Architecture Of The Platform", Deliverable 4.1 "Stretched Data Lakes First Release Report" and Deliverable 5.1 "Trustworthy Data Lakes Federation First Release Report" for details

Funded by the European Union

(FDPs and SFDPs[2]) typically differ from cluster to cluster. Each TEADAL cluster is instantiated and then subsequently managed using an Infrastructure-as-Code approach.

From a conceptual standpoint, a TEADAL node is composed of several layers of processes and hardware, arranged hierarchically. Higher layers utilise the functionality provided by lower layers, while lower layers do not depend on higher layers[3]. Referring to Figure 12 we examine the technological specification of each layer in turn, from the bottom up.

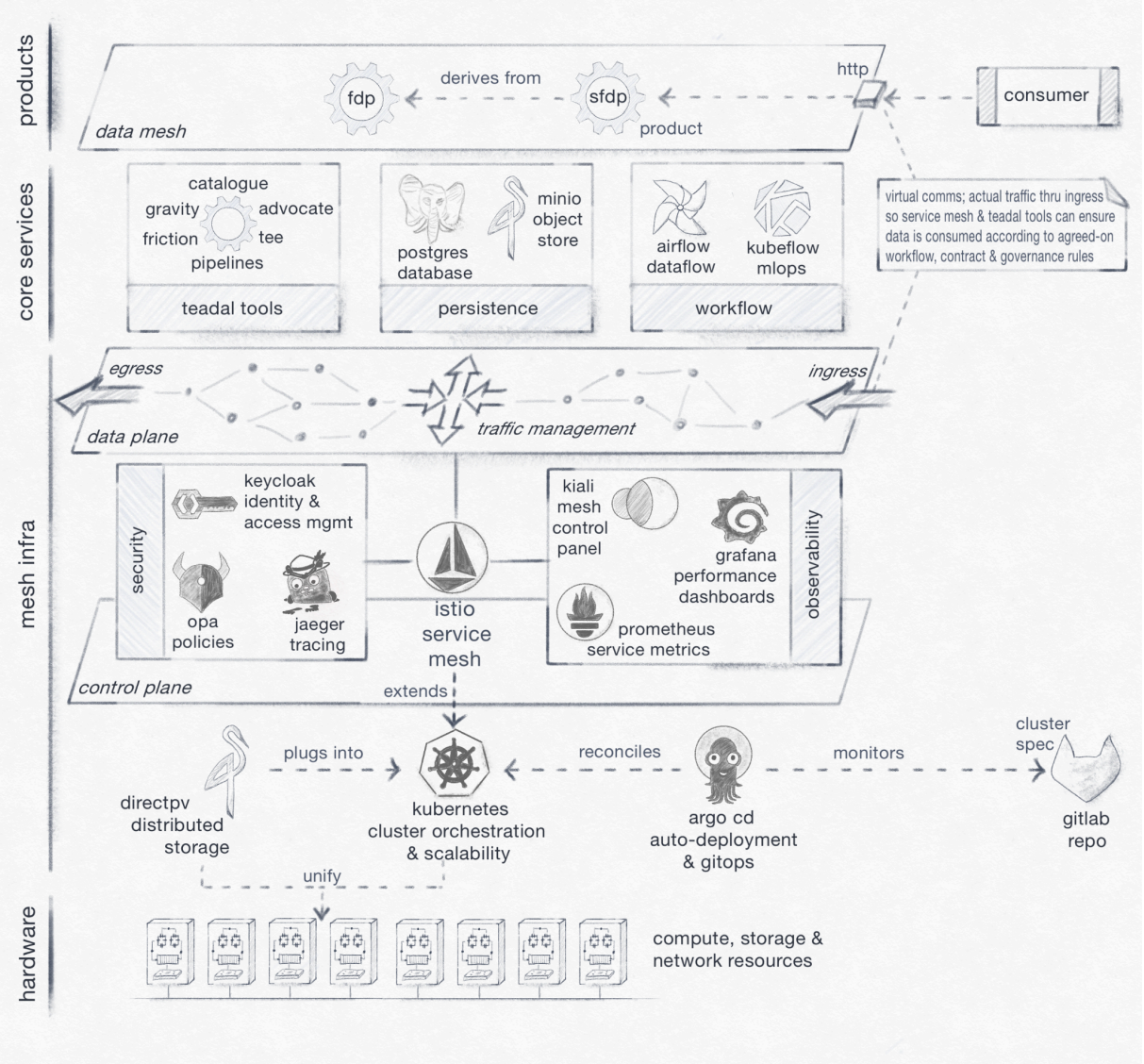


FIGURE 12: TEADAL CLUSTER TECHNOLOGIES

The *hardware layer* is the lowest layer. This is the cluster hardware (computers, network) on which all the TEADAL cluster software runs. In the case of a public cloud deployment, the hardware would typically be virtualized, whereas physical machines would be provisioned for

[2] Please refer to Deliverable 3.1 “Gravity And Friction-Based Data Governance” for details

[3] For a complete description of the data and service mesh runtime architecture, please refer to Deliverable 2.3

Funded by the European Union

an on-premises scenario. In the simplest case, the whole cluster can run on just one machine, whereas more computation-intensive scenarios require several machines.

The *mesh infrastructure* layer interfaces with the hardware layer to provide the service mesh functionality, with Kubernetes at its core for cluster operation, running on a Linux based operating system. This layer manages computational resources and orchestrates the deployment and operation of services by means of containers. Kubernetes interfaces with DirectPV to create a distributed storage facility out of the disks attached to each node whereas Istio complements Kubernetes with mesh control and data planes. The Istio control plane manages a network of proxies that form the Istio data plane, which captures and processes service traffic. The Istio control plane is responsible for overall system management and orchestration, such as scheduling, scaling, and maintaining the desired state of the system. In contrast, the Istio data plane is responsible for the actual movement of packets through the network, handling the flow of application traffic based on the internal policies defined by the Istio control plane. This allows to augment service functionality at runtime without requiring any modifications to the services themselves. The TEADAL cluster exploits this to transparently route and balance service traffic, secure communication and access to service resources (through Keycloak and OPA), and monitor service operation (through Kiali, Prometheus and Grafana). Finally, the mesh infrastructure includes Argo CD, a GitOps continuous delivery tool for Kubernetes, to monitor the cluster Git repository in order to automatically reconcile the desired deployment state declared in the repository with the actual live state of the cluster.

Running on the *mesh infrastructure*, the *core services* layer provides TEADAL baseline functionality which enables federated data products. In this layer, PostgreSQL and MinIO provide database and object storage functionality, respectively. Workflow services are also included: Kubeflow for managing machine learning operations and Airflow for engineering data pipelines. Last but not least, the core services layer hosts the TEADAL-specific tools that allow multiple TEADAL nodes to be joined in a federation where producers and consumers can share data in a trustworthy and secure way, according to agreed-upon governance, privacy and energy-efficiency policies. Catalogue, Advocate, Trusted Execution Environment (TEE), Pipelines, and Policies (security, privacy, and usage) are all examples of TEADAL services and tools in the core services layer.

Finally, the top layer products, hosting cluster-specific data products and services—i.e., federated data products (FDPs), shared federated data products (SFDP), etc. As detailed in D3.1, a federated data product (FDP) extends the notion of data mesh product to cater for sharing data in a data lake federation according to the governance rules of that federation. A shared federated data product (SFDP) encapsulates a consumer-producer agreement (contract) about sharing a part of an FDP and provides the means for the consumer to process the shared data only within the bounds of the agreed-upon contract.

4.4 Deployment Mechanisms

A TEADAL cluster is instantiated and then subsequently updated through a GitOps approach whereby the desired cluster runtime is declared in an online Git repository and a dedicated GitOps cluster service reconciles the desired runtime with the actual cluster state. Thus, there is a Git repository associated with every TEADAL cluster and, as briefly mentioned

Funded by the European Union

earlier, there is an ArgoCD service in that cluster which monitors the Git repository in order to automatically reconcile the desired deployment state with the actual live state of the cluster.

The deployment state in the Git repository is declared through a set of YAML files which Kustomize[4] can process. Each of these files declares a desired instantiation and runtime configuration for some of the components in the TEADAL cluster. Collectively, the files at a given Git revision describe the deployment state of the entire TEADAL cluster at a point in time. Changes to the live system are triggered through an automated workflow which the cluster administrator initiates by creating a new revision of some configuration files (in YAML format) in the Git repository. On detecting a new revision, ArgoCD transitions the cluster to the new desired state. The diagram in Figure 13 exemplifies the GitOps workflow.

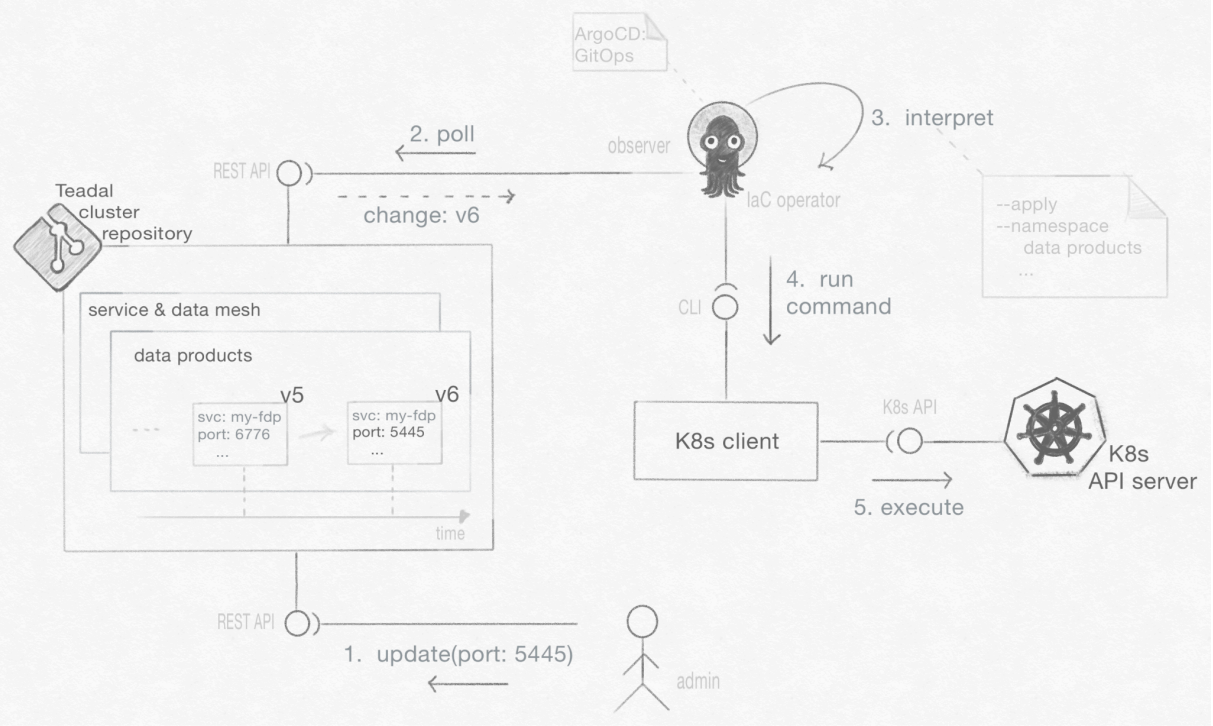


FIGURE 13: TEADAL GitOps WORKFLOW EXAMPLE

Indeed, the diagram depicts a typical scenario where the cluster administrator carries out a change to a data service. As can be seen, the Git repository contains descriptors for an FDP named *my-fdp* as well as other descriptors, not explicitly shown, for the service and data mesh components in the various cluster runtime layers mentioned earlier. The latest Git revision is *v5* where the FDP service port is 6776. The administrator changes the port to 5445, making a new Git revision *v6*. ArgoCD periodically polls the Git repository to detect any new revisions. Thus, shortly after the administrator pushed revision `v6` to the Git repository, ArgoCD realises that the current cluster runtime state refers to a stale revision, *v6*, whereas *v6* is the latest. Hence, ArgoCD proceeds to interpret the stanzas in the YAML file as a command line that the Kubernetes client can understand. After assembling the required

[4] Kustomize (https://kustomize.io/) is a tool to create Kubernetes cluster configuration resources modularly by assembling and extending Kubernetes resource definitions in YAML files.

Funded by the European Union

command, ArgoCD invokes the Kubernetes client with it. In turn, the Kubernetes client calls the Kubernetes API which finally triggers the desired deployment actions on the live cluster, resulting in the deployment state to reflect the YAML configuration at revision *v6*—i.e., *my-fdp*'s port is now 5445.

Funded by
the European Union

5 COMPONENT DESCRIPTIONS

TEADAL is a framework for ensuring trust and transparency in federated data exchange whose architecture consists of several components. The Data Consumer requests access to data, while the Data Provider provides the data, registers it in the Catalogue and applies the appropriate policies to make the data discoverable. The Control Plane manages the onboarding, transformation and enforcement of policies. The Trust Plane validates compliance with these policies, stores evidence of integrity and enables audits, supported by Advocate for capturing and storing evidence of interaction. The Service Mesh ensures that policies are enforced and transparency is maintained. The Data Pipeline handles the actual transformation and processing of the data for the exchange. These components are explained in detail in the following.

5.1 Catalogue

The Catalogue is a cornerstone of TEADAL's architecture, offering a comprehensive and organized overview of all data assets within the federation as it provides metadata and policies for the data to make it discoverable. By integrating with an identity management system, it restricts data asset visibility to authorized users with the appropriate credentials. This functionality ensures that the Catalogue enhances discoverability across the federation, allowing members to explore the available FDPs. Additionally, TEADAL's Catalogue streamlines the creation of both FDPs and SFDPs through its integration with BPMN workflows, automating various steps in the process.

For data providers, the Catalogue serves as a platform to publish metadata for federated data products (FDPs). Once an agreement (contract) between a data provider and a data consumer is established regarding the use of an FDP, the Catalogue facilitates access for the consumer by making the corresponding SFDP discoverable.

5.2 Control Plane

The **TEADAL Control Plane** orchestrates the lifecycle of data products deployed across TEADAL Nodes within the TEADAL Federation. In its final architecture (as detailed in D4.3), the Control Plane is composed of four distinct yet interrelated subsystems:

- **Monitoring Subsystem (AI-DPM):** This subsystem collects and analyzes runtime telemetry from both infrastructure components (e.g., CPU, memory load) and data service execution environments. Leveraging machine learning techniques, it extracts operational insights and generates actionable reports to assist Data Lake Operators in monitoring system health, detecting anomalies, and optimizing performance.

- **Automation Subsystem (ASG):** The ASG subsystem provides tools for the automatic generation and execution of Shared Federated Data Products (sFDPs) as self-contained data services. These services fetch data from source FDPs, apply transformations based on negotiated agreements, and serve the resulting data to consumers. A shared runtime library enables caching, encoding, and efficient delivery, supporting high performance and adaptability.

Funded by the European Union

The automation capabilities are enhanced through LLM-driven tools that assist in generating transformation logic and deployment specifications from human-readable agreements. The transformation library itself is envisioned as a stand-alone component that supports the entire transformation lifecycle—including creation, validation, selection, runtime loading, and execution—aligning with emerging trends of exchanging assets via Model Catalog Platforms (MCPs).

- **Optimization Subsystem:** Significantly simplified compared to its original conception (see D4.1), this subsystem now functions as a lightweight placement engine. Its primary role is to determine the most suitable TEADAL Node for deploying a newly generated sFDP, based on system-wide metadata such as resource availability, data locality, and performance policies. Additionally, it continuously monitors deployed services—using insights from the monitoring subsystem—and can trigger runtime adjustments (e.g., service migration or transformation replacement) to maintain system efficiency and SLA compliance.

- **Deployment Subsystem:** Also simplified in the final architecture, the deployment subsystem is responsible for dispatching sFDPs to the TEADAL Nodes selected by the optimizer. This subsystem is aligned with the technology stack used across the TEADAL Platform, realised as part of WP6. This includes relying on GitOps for service deployment, including the infrastructure-level,the platform-level, and the pilot-level services, ensuring consistency and traceability across the federation, as well as on GitLab as source of truth. In addition, this includes relying on the k8s, with its declarative control plane, as a runtime environment.

The TEADAL Control Plane leverages advanced AI and machine learning techniques to enable dynamic, efficient, and scalable data product lifecycle management. These technologies support not only telemetry analysis but also automate traditionally manual tasks, such as:

- Generating sFDP definitions from data sharing agreements between FDP Consumers and Providers,

- Generate executable Rego rules starting from natural-language policy intents or dedicated diagrams defined with the provided Data Sharing Policy Notation (DSPN),

- Adapting running services to changing workloads and system conditions.

This intelligent, modular approach enables TEADAL to support complex cross-organizational data-sharing workflows with minimal manual intervention and maximum operational agility.

5.3 AI-DPM

The AI-DPM (Artificial Intelligence-Driven Performance Monitoring) component facilitates intelligent monitoring of resource usage, energy consumption, and service mesh telemetry in the TEADAL systems. Using Artificial Intelligence for IT operations (AIOps) approaches, AI-DPM analyses historical, time-stamped metrics to detect anomalies and generate predictive insights. AI-DPM was initially designed to provide insights to the Control Plane

Funded by the European Union

Optimiser to define and establish effective strategies for optimising data flows in TEADAL. In the final version of the architecture, AI-DPM has evolved into a standalone REST API service that is available not only for the Control Plane but also for any TEADAL component that may need AI-DPM prediction and anomaly insights.

AI-DPM system architecture is shown in the diagram below, illustrating the different components and the flow of metadata among them (Figure 14). Architectural components of AI-DPM in TEADAL include the integration of monitoring and observability tools that collect a wide array of timestamped operational metadata, tools that aggregate the collected data, the layer that applies AI algorithms to the data, and finally, a layer that serves the output of the algorithms for insights.

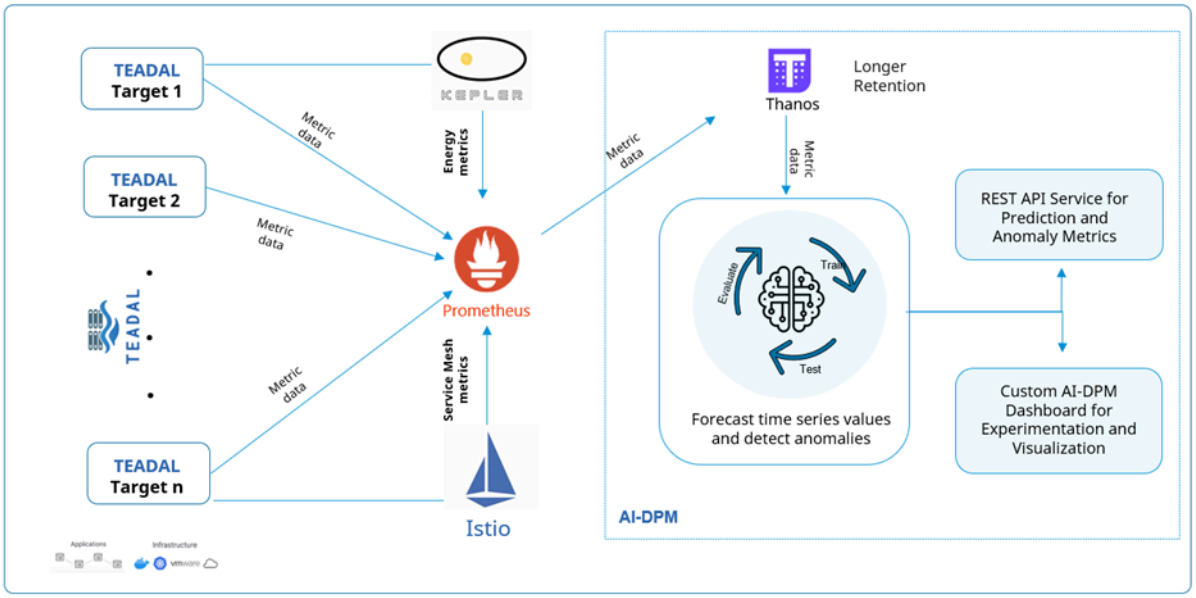


FIGURE 14: AI-DPM APPLICATION ARCHITECTURE (COMPONENTS) DIAGRAM

Starting from the observability and monitoring services of the TEADAL infrastructure and application environment, the metrics for AI-DPM flow through multi-layered architectural components. The **SEQUENCE** diagram illustrating the flow of information between the various layers of AI-DPM components, along with component functionality, is shown below (Figure 15).

Funded by the European Union

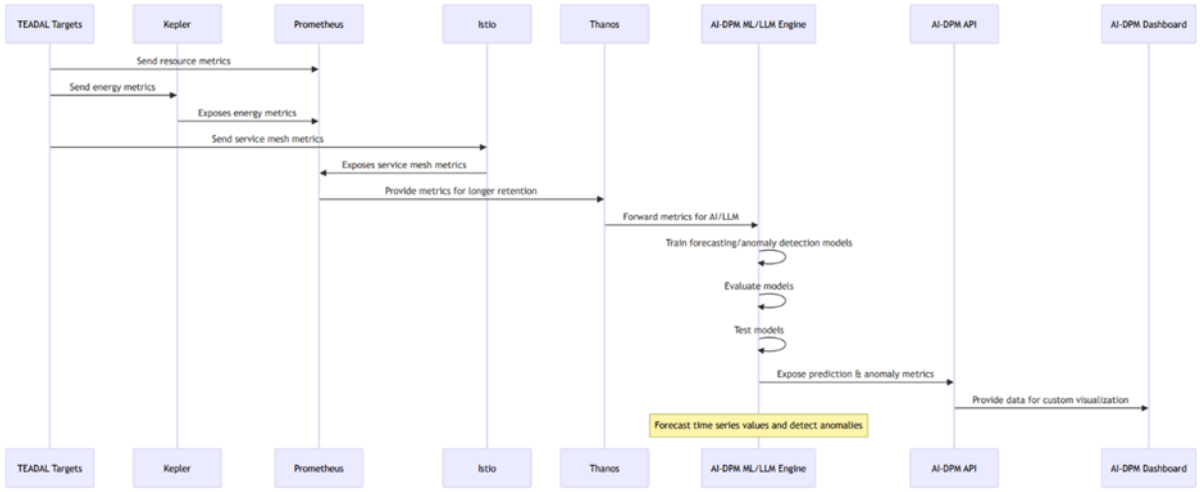


FIGURE 15: AI-DPM FUNCTIONALITY AND INFORMATION FLOW

The AI-DPM is integrated with TEADAL's broader system architecture using the key observability and monitoring services as the central integration hub, particularly through Prometheus. AI-DPM extends the TEADAL's observability and monitoring service triads viz Prometheus[5], Kepler[6], and Istio[7], by integrating with a time series database, Thanos[8]. This database directly interfaces with the AI-DPM's /fetch endpoint to retrieve necessary data for processing. In addition to the /fetch endpoint, the AI-DPM contains three other key endpoints /train, /infer, /anomaly [TO3] and compute_rmse, which are meant for training forecasting models, generating predictions, identifying anomalies, and computing root mean square for model performance evaluation, respectively (Figure 16).

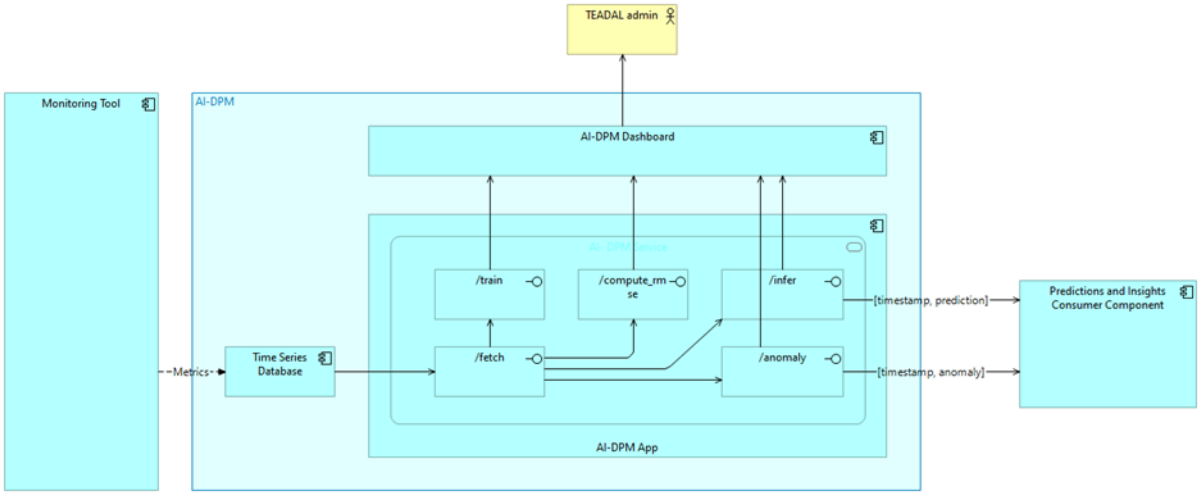


FIGURE 16: AI-DPM INTEGRATION IN TEADAL

Within the AI-DPM ecosystem, data flows from /fetch to /anomaly for anomaly detection, and from /train to /infer for predictive modelling. The AI-DPM Dashboard provides visualization

[5] HTTPS://PROMETHEUS.IO/
[6] HTTPS://GITHUB.COM/SUSTAINABLE-COMPUTING-IO/KEPLER
[7] HTTPS://ISTIO.IO/
[8] HTTPS://THANOS.IO

Funded by the European Union

and management capabilities, feeding information back to TEADAL admins for system oversight. The processed outputs, specifically, timestamps with predictions and anomaly information, are then made available for the insights of the consumer component. This integration pattern creates a comprehensive data processing pipeline that transforms raw monitoring metrics into actionable insights. This setup allows the AI-DPM system to maintain and analyse historical metrics over extended periods, enabling trend analysis, anomaly detection, and predictive modelling that form the core of TEADAL's AI-driven performance monitoring (AI-DPM).

The AI-DPM system has been designed upon the monitoring ecosystem of TEADAL nodes deployed on Kubernetes using open-source toolsets to ensure wide-ranging observability, alerting, and visualisation capabilities. This ecosystem primarily consists of Prometheus, Istio, and Kepler. Building upon this, the AI-DPM component consists of Thanos, AI algorithms, APIs, and GUI for experimentation and visualization. These elements of AI-DPM are part of the general AI-DPM architectural layers, viz. Data Aggregation and Processing Layer (Thanos), AI Analytics Layer (AI algorithms), and the Serving Layers (APIs and UI). The following subsections describe the core architectural aspects of AI-DPM, detailing how monitoring data is collected, processed, and transformed into predictive insights.

5.3.1 Metadata Collection

In TEADAL, Prometheus is an open-source and primary monitoring tool, enabling the gathering and storing of metrics through a process called scraping. It is configured to scrape data from endpoints that expose metrics, such as application performance indicators, resource usage statistics (like CPU, memory, and disk usage), and other system-level data. Prometheus collects real-time data from applications, servers, and network devices, storing it in a time-series database for efficient querying, alerting, and visualization.

Istio is an open platform used in TEADAL for providing a uniform way to integrate microservices, manage traffic flow across microservices, enforce policies, and aggregate telemetry data. Istio's control plane provides an abstraction layer over the underlying cluster management platform, such as Kubernetes. In the AI-DPM context, Istio generates telemetry data such as metrics, logs, and traces via the Envoy proxies, which are configured to expose Prometheus-compatible metrics endpoints. These metrics are scraped by Prometheus, allowing for real-time monitoring and observability of service behavior, traffic patterns, and performance across the mesh.

Kepler has components with specific functionalities. Kepler (Kubernetes-based Efficient Power Level Exporter) is a system-level exporter that collects power consumption metrics from Linux kernel interfaces and attributes them to Kubernetes workloads. It gathers telemetry about CPU, memory, and GPU power usage, along with container-level resource usage. Kepler exposes this telemetry data in a Prometheus-compatible format via a metrics endpoint. Prometheus scrapes this endpoint to collect power and efficiency metrics across the cluster, assisting energy-aware monitoring and visibility into the energy footprint of workloads.

5.3.2 Metadata Aggregation and Storage

Funded by the European Union

Thanos is an open-source project under the Cloud Native Computing Foundation (CNCF). Its primary goals are to provide a global query view, offer unlimited metric retention, and ensure high availability across all components. Thanos is a set of tools designed to enhance an existing Prometheus setup, making it more scalable, highly available, and capable of handling limitless metrics storage. It integrates seamlessly with Prometheus without requiring significant changes to the system. By using the Prometheus storage format, Thanos efficiently stores historical metric data in object storage systems. It ensures fast query performance, even when managing vast amounts of data. Thanos also aggregates metrics from multiple Prometheus servers, including Prometheus high-availability setups, providing a unified view for querying across the entire infrastructure.

Thanos has components with specific functionalities. The Thanos Sidecar is a stateless component that is responsible for establishing a connection between Thanos and Prometheus. It reads metric data from Prometheus and streams it to the object storage for querying. There are also other components in Thanos responsible for aggregating metrics from multiple Prometheus instances, such as the Store Gateway, which simplifies federation by acting as a unified endpoint and implementing the StoreAPI for historical data. The Compactor handles deduplication and downsampling of data in storage buckets, optimizing storage, improving query performance, and enforcing retention policies to manage data lifecycle.

5.3.3 AI models and the machine learning cycle

In AI-DPM predictions and anomaly detection result generation, the machine learning life cycle follows three core stages: training, testing, and evaluation. This process is applied across a diverse stack of six AI models, which include statistical models (ARIMA and Prophet), recurrent neural network (RNN) models (GRU and LSTM), and advanced time-series large language models (Lag-Llama and TimeGPT). Each of these models contributes unique strengths to the overall forecasting and prediction capabilities of the monitoring system.

During the training phase, the models are exposed to historical time-series metadata to learn underlying patterns, trends, and temporal relationships. Statistical models like ARIMA and Prophet identify recurring structures and seasonalities in the data. Deep learning models such as GRU and LSTM are trained to capture complex, long-term dependencies in sequential data. Meanwhile, large language models like Lag-Llama and TimeGPT use pre-trained language modelling capabilities adapted to time-series forecasting.

Following training, the models are subjected to a testing phase, where they are evaluated on previously unseen data. This step is essential to ensure that the models generalize well beyond the data they were trained on and are not simply memorizing patterns. It helps validate the robustness of each model in real-world deployment scenarios. Finally, during the evaluation stage, each model's performance is assessed using standard forecasting metrics such as Root Mean Square Error (RMSE). These metrics enable direct comparison of predictive accuracy across the various model types. The results support a deeper analysis of each model's strengths, highlighting their specific advantages within the overall AI-DPM framework.

AI-DPM outputs – APIs and GUI

Funded by the European Union

The AI-DPM outputs are served as REST APIs, and an interactive experimentation and visualization GUI.

REST APIs: AI-DPM is developed as a REST API service that provides time series forecasting and anomaly detection capabilities using multiple models and integrates with persistently stored Thanos metrics. The service supports multiple models, including a cloud API for TimeGPT LLM (requires tokens) and other local models, ranging from RNNs to classical statistical models and LLMs. In addition, the service offers a configurable training window and flexible data sourcing from Thanos via PromQL queries.

The API service features five main operations:

- Fetching Historical Data (/fetch): This endpoint retrieves time series data based on a specified Prometheus query and time range (in hours). The request requires the query and duration, and optionally accepts a Thanos URL. The response includes the historical data, which is essential for model training and evaluation.
- Model Training (/train): This endpoint trains forecasting models—either local (GRU, LSTM, ARIMA, Prophet) or LLMs (Lag-Llama)—using specified parameters like query, training duration, input/output steps, and model type. Local models are saved under the models/ directory for reuse. The API responds with a message indicating successful training.
- Inference (/infer): This endpoint generates predictions using a previously trained model. It requires the same parameters as training—query, time range, input/output steps, and model name. All the models, including classical as well as local and cloud-based LLMs are supported. The output is a list of future time-stamped predictions.
- Anomaly Detection (/anomaly): This endpoint identifies anomalies in time series data using the provided query and detection method. You can specify the confidence interval and detection duration. It returns a list of time-stamped values flagged as anomalies, helping detect unusual behavior in monitored metrics.
- Model Evaluation (/compute_rmse): This endpoint calculates Root Mean Square Error (RMSE) across multiple predictive models. It returns a number for each model with lower values indicating better model performance. The metric helps identify which models perform best for different scenarios, enables ensemble decision-making, facilitates model comparison, and supports model selection.

These endpoints collectively enable a complete AI-DPM workflow: from data retrieval and model training predictive insights and anomaly monitoring. The detailed parameter examples and schema of endpoints are provided in the Swagger, and with a couple of examples shown in Figure 17 below.

Funded by the European Union

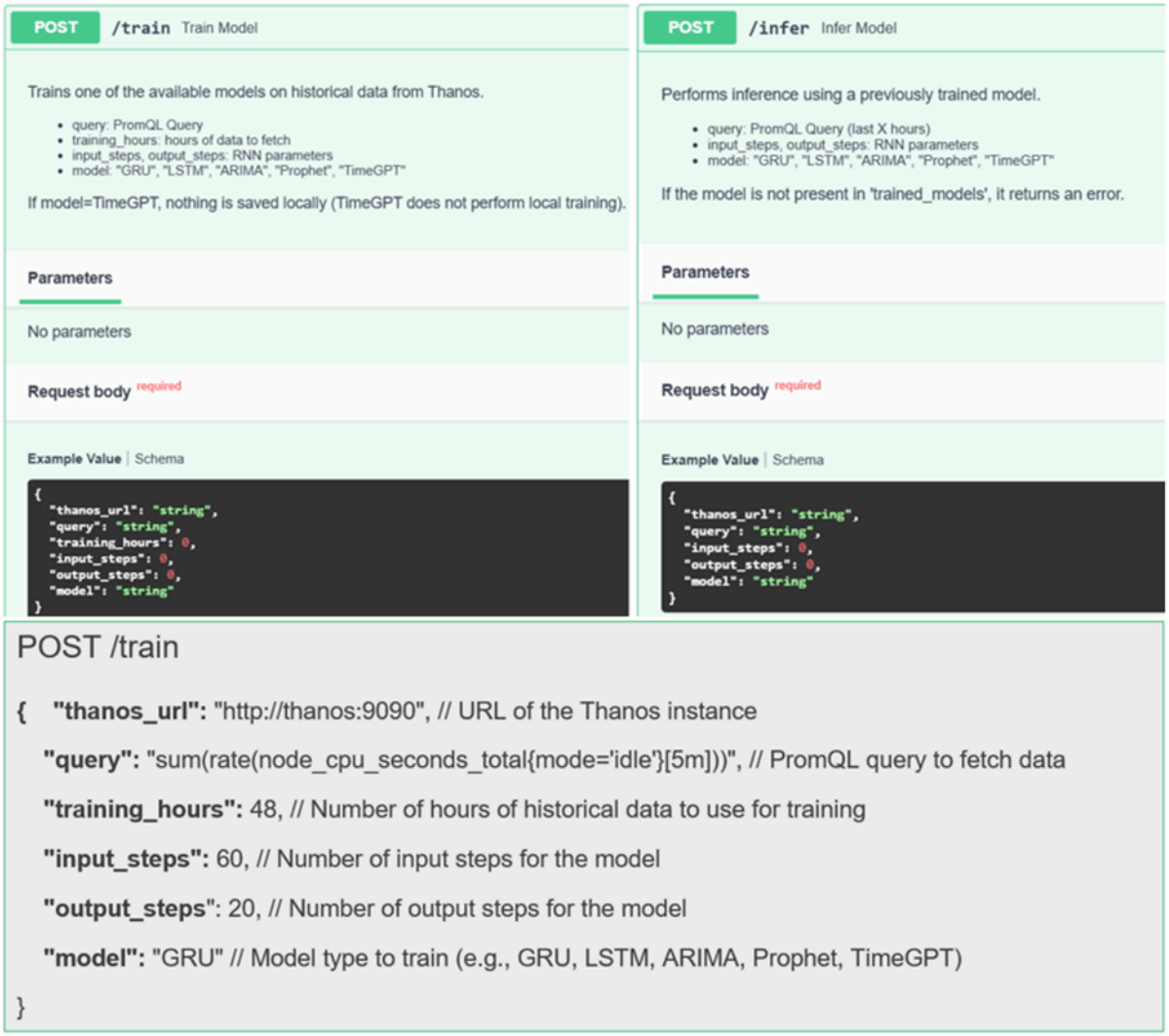


FIGURE 17: AI-DPM's REST API ENDPOINTS; /train (left) and /infer (right)

AI-DPM Dashboard: The AI-DPM Dashboard is a dedicated experimentation interface designed to clearly show the functionalities of the AI-DPM tool that support the basic AI workflow, including training, testing, and evaluation of machine learning models for time-series forecasting and anomaly detection. It integrates the multiple model implementations of the AI-DPM tool with configurable parameters, allowing users to execute training workflows, assess model performance, and compare results across different configurations. It enables quick comparison of AI-DPM multi-model performances and helps users choose the most suitable AI model for their specific context.

The dashboard is organised into two primary sections: the Global Configuration panel on the left for setting global parameters, and the Monitoring & AI/ML Dashboard panel on the right side for executing specific workflows such as data retrieval, model training, and performance evaluation (Figure 18).

Funded by the European Union

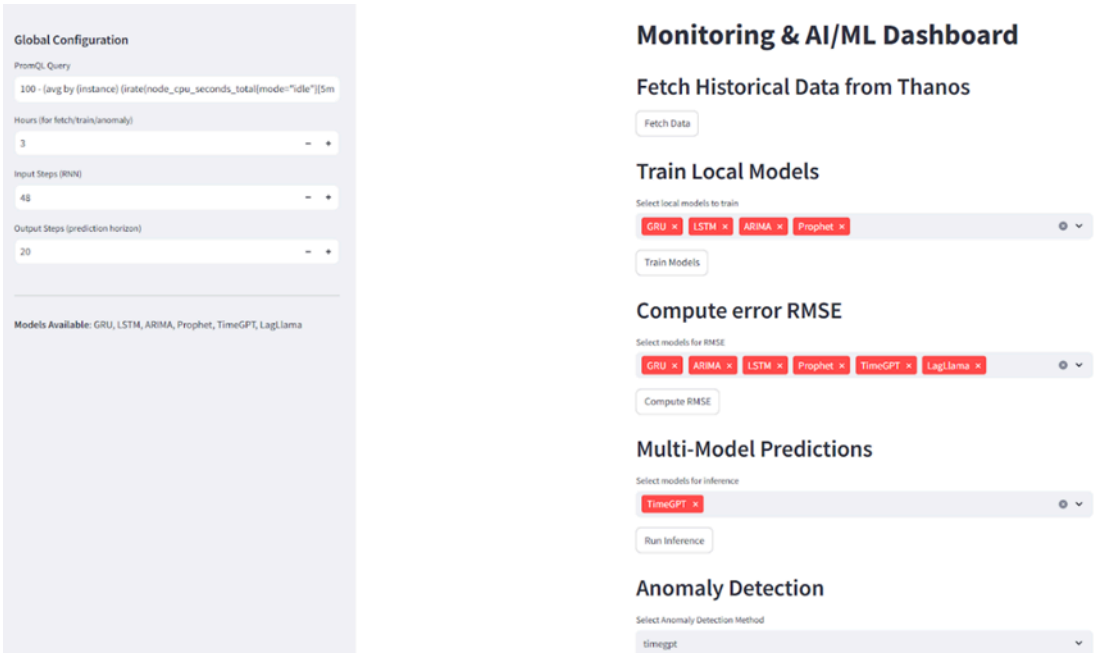


FIGURE 18: AI-DPM SERVICE DASHBOARD

The AI-DMP Dashboard is part of the distribution of the AI-DPM tool. It calls the AI-DPM API to provide a presentation layer that can enable the leveraging of all the functionalities provided by the API. It can be useful mainly for users interested in experimenting with the array of AI models available in AI-DPM. Configure training parameters, practice all the ML cycle to choose the Models that are more suitable for their needs and contexts. Eventually, evaluate the performance of the models and make an informed decision based on the appropriate monitoring metrics.

5.4 Trust Plane

The Trust Plane validates compliance and maintains evidence for integrity and audit. It ensures transparency and accountability within the TEADAL federation by establishing an evidence infrastructure that systematically collects and stores verifiable records of all processes, interactions, and data exchanges. This evidence is securely maintained in an immutable format, enabling robust a posteriori verification. By leveraging a trust-enhancing infrastructure, the Trust Plane fosters confidence among federation members, while the accessibility of this evidence to all participants promotes independent validation and reinforces transparency. The TEADAL trust plane is composed of several components, as introduced in the following.

5.4.1 Advocate

The Advocate is a core component of the Trust Plans and plays a pivotal role within the TEADAL framework by ensuring the integrity and trustworthiness of federated data exchanges. Its primary function is to orchestrate the collection, aggregation, and publication of verifiable evidence related to all significant interactions and transactions involving Federated Data Products (FDPs) and Shared Federated Data Products (sFDPs). By leveraging advanced cryptographic techniques such as public/private key pairs and verifiable

Funded by the European Union

credentials, Advocate generates tamper-proof records of critical events in the data lifecycle. These records are securely stored in an immutable manner, anchored in a blockchain-based claims registry, and made accessible for future audits.

Each data lake in the TEADAL federation hosts an instance of the Advocate, ensuring that all local operations adhere to predefined protocols for data integrity and compliance. The Advocate collects evidence from diverse sources, including tracing logs from observability tools like Jaeger, Kubernetes events, and Catalogue operations. By integrating these inputs, it provides a comprehensive audit trail that traces every action back to its origin, ensuring transparency and accountability across the federation. This evidence is stored in shared infrastructure like IPFS and is enhanced with metadata to link actions to specific deployments.

The Advocate also supports advanced configurations to adapt to the needs of various use cases. For example, it enables policy-based validation of claims, ensuring that observed actions comply with governance policies. By incorporating mechanisms such as smart contracts and decentralized identifiers (DIDs), Advocate strengthens the verifiability of evidence and facilitates interactions across federation members. In addition, it provides APIs for application-driven reporting, allowing services to submit evidence claims directly.

5.4.2. TEADAL Name Service (TNS)

The TEADAL Name Service (TNS) is a decentralized naming system within the TEADAL federation's distributed ledger infrastructure. Its core purpose is simplifying the identification, referencing, and discovery of federation entities such as organizations, services, and data products.

TNS functions similarly to a traditional Domain Name System (DNS), but for a decentralized environment. Instead of dealing with low-level blockchain addresses, TNS allows human-readable names to represent complex resources such as Federated Data Products (FDPs), Shared FDPs (SFDPs), and organizational domains.
TNS achieves this by maintaining a registry on the blockchain where federation members can register their domains and services. These registrations are linked to metadata and cryptographic keys and are coordinated with Federation Smart Contracts, which manage membership and governance rules. When a new resource is registered or updated, TNS records the change immutably in the ledger, producing tamper-proof evidence. By enabling consistent and verifiable naming, TNS enhances interoperability, ensures secure discovery and access, and supports trust-building through transparent logging of events across the federation.

5.5 Data Sharing Policies and Service Mesh

The Data Sharing Policies and the Service Mesh ensure controlled communication between consumers and federated data product (FDP) services by utilizing interception proxies. These proxies enforce access control by intercepting HTTP traffic and validating requests against predefined policies. Data providers create these access control policies, which are stored in a dedicated policy store and evaluated against consumer requests. The component

Funded by the European Union

integrates seamlessly with the TEADAL data mesh, routing requests through proxies for policy checks before they reach the FDP services. It thus ensures that policies are enforced and transparency is maintained.

This component uses open-source technologies such as Istio, Envoy, and Open Policy Agent (OPA). Each data product service is paired with an Istio proxy, which intercepts and routes incoming traffic. The Envoy proxy utilizes an External Authorization Filter to connect with OPA, where policies written in the Rego language are fetched from the policy store and evaluated. This mechanism ensures that only authorized requests are granted access to the data product services, maintaining a high level of security and compliance.

TEADAL includes a Role-Based Access Control (RBAC) framework tailored for RESTful services to simplify access control. This framework not only streamlines the implementation of access control policies but also supports alternative policy decision points. These include the TEADAL Datalog Interpreter and Anubis, providing flexible and robust options for secure access management across the federation. This combination of technologies and frameworks ensures that the Security Policies Component adapts to diverse requirements while maintaining stringent security standards.

5.6 Data Pipelines

In the early architecture presented in D2.3 and D4.1, computational flows applied to data on its way from original datasets to data products and from source FDPs to target sFDPs, were defined as data pipelines, modeled as Kubeflow Pipelines, and planned to be distributed across federation nodes via the Kubestellar framework to be then processed by the Kubeflow engines deployed on each node. The Stretched Data Lake Compiler (SDLC) was created to handle pipeline placement optimization, annotating each task with deployment preferences derived from system metadata, hardware constraints, and transformation policies. Optimization objectives included reducing energy consumption and balancing data gravity (co-locating compute near data) against data friction (minimizing transfer overhead). However, this optimization process was inherently multi-stage and potentially non-convergent, especially for non-trivial workflows. When optimal solutions could not be confidently derived, the SDLC fell back to providing ranked lists of candidate nodes per task, which were later resolved by the Stretched Pipeline Executor into final deployments, fed into local Kubeflow engines.

While this model offered fine-grained control and architectural expressiveness, it introduced substantial operational complexity. The need for heavy optimization logic, tight orchestration loops, and per-task distribution made it difficult to scale, maintain, or adapt pipelines, particularly in dynamic federated environments with evolving data-sharing needs.

To overcome these limitations, TEADAL adopted a simplified, service-oriented approach centered on the ASG subsystem. Rather than generating annotated DAGs for orchestration engines, the ASG tools now synthesize each sFDP as a standalone, self-contained data service. These templated services embed the complete logic required to fetch, transform, and serve data based on human-readable agreements. A shared runtime library ensures support for caching, encoding, transformation execution, and integration of

Funded by the European Union

privacy-enhancing technologies such as TEEs and MPC, thereby eliminating the need for per-task orchestration engines like Kubeflow.

Key advantages of this approach include:

- Coarser-grained, service-level optimization, enabling faster placement decisions for data products without relying on complex (and often premature) task-level planning, which can instead be deferred and adapted as conditions evolve after the data product deployment

- Improved support for dynamic re-optimization, as task-level execution decisions can be revised at the service level without recomputing or re-validating entire DAG-based plans

- Automation, including the use of LLMs to generate transformation logic and deployment policies from natural language contracts

- Alignment with GitOps workflows,enabling consistent, declarative deployment via Kubernetes, with GitLab acting as the unified source of truth for all components—from infrastructure to services

- Reduced complexity and increased maintainability, by eliminating the need for orchestration frameworks like Kubeflow and avoiding dependency on complex, multi-stage multivariate optimization solvers

Overall, this shift from orchestrated pipelines to ASG-generated services represents a fundamental simplification in how TEADAL realizes data products. It enables faster iteration, greater automation, and better adaptability to cross-organizational data-sharing scenarios, without compromising on optimization or privacy guarantees.

The TEADAL architecture consists of interdependent components that work together to enable secure, transparent and efficient data exchange in a federated environment. Each component fulfils a clearly defined role within the system, from discoverability and policy enforcement to monitoring, evidence generation and data processing. The Catalogue, Control Plane and Data Pipelines form the operational backbone of data lifecycle management. The Trust Plane and Advocate ensure integrity and auditability, while the Service Mesh and Policy Engine secure access and enforce governance. AI-DPM provides intelligent monitoring capabilities to support adaptive optimisation. This modular structure enables TEADAL to respond flexibly to domain-specific requirements while ensuring coherence, trust and compliance across the entire network.

Funded by the European Union

6 COMPONENT INTERACTIONS

This chapter presents an end-to-end sequential flow of data sharing within the TEADAL architecture by showing how core components interact throughout the lifecycle of a Federated Data Product in Figure 19. Starting from onboarding and catalogue registration to transformation, policy enforcement, and evidence generation, it shows how responsibilities are distributed across the Control Plane, Trust Plane, Service Mesh, Data Pipelines, and Catalogue. These coordinated interactions ensure that data sharing in the federation remains secure, policy-compliant, and auditable.

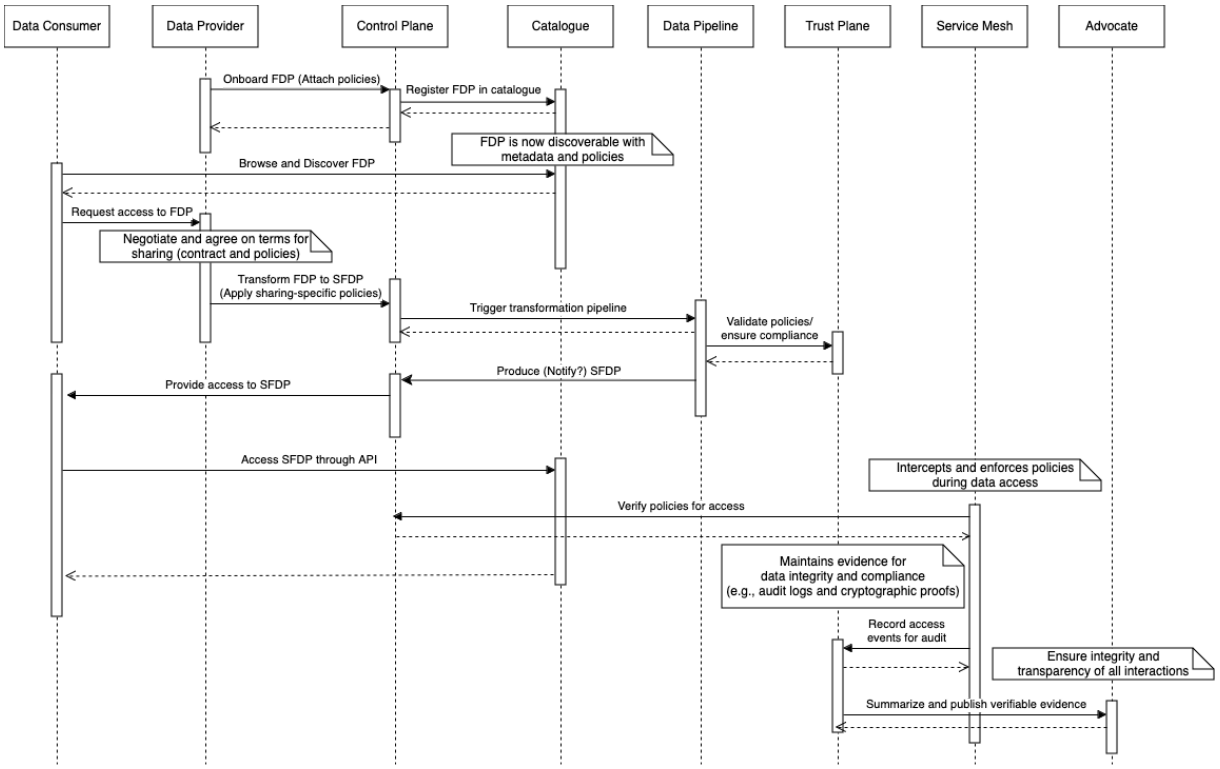


FIGURE 19: INTERACTION BETWEEN TEADAL COMPONENTS

The following describes how the TEADAL components interact in an end-to-end interaction flow.

1. **Onboarding the FDP**
   The Data Provider begins the process by deploying the FDP (Federated Data Product) to the TEADAL platform. During this phase, policies, which are defined as Rego files, are prepared by the Data Provider and applied through the platform's GitOps-based orchestration, which ensures they are correctly attached to the FDP These policies define governance rules, data ownership details, access restrictions, and metadata. Metadata includes descriptive attributes such as the type, format, and

Funded by the European Union

intended use of the data. This step ensures that the FDP is properly configured and aligned with the federation's standards, enabling controlled operations downstream.

2. **Registering FDP in the Catalogue**
   Once the FDP is onboarded, the provider registers it in the Catalogue. The Catalogue acts as a central repository that enhances discoverability within the federation. It integrates with an identity management system, which restricts the visibility of data assets to authorized users only. By registering the FDP, the platform updates its metadata and policies, making it discoverable by potential consumers. This step also ensures that the FDP can be explored through federated searches, allowing members to assess the data's suitability for their needs.

3. **Requesting Access to FDP**

   The Data Consumer initiates the next phase by sending a request to access the FDP via the Catalogue, which serves as the primary discovery means. This request is handled by the Control Plane, which acts as an intermediary between the provider and the consumer. During this step, the platform facilitates negotiations for terms and agreements, such as usage rights, compliance requirements, and pricing (if applicable). These agreements are stored in the Catalogue and act as a reference for subsequent interactions. This phase ensures that both parties have clearly defined expectations and responsibilities.

4. **Transforming FDP into SFDP**

   After the terms are agreed upon, the Control Plane transitions the FDP into an SFDP (Shared Federated Data Product). This transformation involves applying sharing-specific policies that adhere to the negotiated terms. Policies may include additional encryption, redaction, or partitioning of data to ensure compliance with the agreed-upon usage conditions. The SFDP is essentially a derivative of the FDP, tailored for secure and governed sharing with the Data Consumer.

5. **Triggering the Data Pipeline**

   The Control Plane triggers the Data Pipeline to handle the processing of the SFDP. The pipeline is responsible for tasks such as data preparation, transformation, and delivery. These tasks may include reformatting data for compatibility, applying encryption for secure transfer, or leveraging privacy-preserving mechanisms like Trusted Execution Environments (TEEs) or Secure Multi-Party Computations (MPC) for confidential data processing, enabling secure computations on distributed datasets, or enforcing strict access controls without exposing sensitive information. The pipeline ensures that the SFDP is optimized and ready for access, fulfilling both technical and contractual requirements.

6. **Validating Policies and Ensuring Compliance**

   During the transformation process, the Trust Plane validates the applied policies to ensure compliance with governance rules. This validation checks that the SFDP adheres to sharing agreements, data usage constraints, and legal requirements. By

Funded by the European Union

ensuring compliance, the Trust Plane fosters trust among federation members and prevents potential misuse or unauthorized sharing of data. Detailed processes described in D3.3 and D5.3.

7. **Providing Access to SFDP**

   Once the SFDP is processed and validated, the Data Pipeline provides access to the Data Consumer. As soon as the SFDP is ready to receive requests, its URL is registered through the contract facilitation flow in the Catalog, ensuring discoverability and controlled access. Access is granted through APIs or specific data channels as defined in the agreement. This ensures that the consumer receives the data securely and in a usable format. This phase completes the primary sharing process, making the data available for the consumer's intended purposes.

8. **Intercepting and Enforcing Policies in Real-Time**

   As the Data Consumer accesses the SFDP, the Service Mesh intercepts and validates each request against predefined access control policies. This is achieved through interception proxies such as Istio and Envoy. The Service Mesh ensures that only authorized requests are granted access to the SFDP. Simultaneously, the Trust Plane records all access events, providing a detailed log of data usage.

9. **Generating Audit Logs and Cryptographic Proofs**

   The Trust Plane generates audit logs and cryptographic proofs for every significant interaction with the SFDP. These records are stored immutably and contain details such as timestamps, user actions, and compliance status. Cryptographic proofs ensure the integrity of the logs, enabling independent verification. These artifacts support audits and provide transparency for regulatory or contractual compliance.

10. **Summarizing and Publishing Evidence**

    The Advocate component aggregates and summarizes evidence related to the data-sharing operation. This evidence includes audit logs, compliance records, and cryptographic proofs. The Advocate publishes this information, making it accessible for audits or further verification. By providing a verifiable summary, the Advocate ensures accountability and builds trust among all federation members. This step concludes the data-sharing lifecycle while maintaining transparency and adherence to governance rules.

The component interactions illustrate how TEADAL ensures secure, compliant and auditable data sharing across the entire lifecycle of a federated data product. By orchestrating the roles of the Control Plane, Trust Plane, Service Mesh, Data Pipelines and Catalogue, the platform enforces governance rules at every step, such as onboarding, registration, access control and evidence creation. These tightly integrated processes form the operational backbone of the TEADAL architecture and are the foundation for building trust, enabling compliance and supporting transparency within the data network.

Funded by the European Union

7. INTERACTION BETWEEN TEADAL NODES

Teadal is a cloud computing platform for sharing data among organisations. This sharing happens by federating clusters running Teadal software. In this section we review the concept of a Teadal federation and look at interaction patterns among federation sites.

7.1 Federated data sharing

Teadal delivers contract-bound, trusted, verifiable, and efficient data sharing among organisations. It achieves this by allowing organisations to make their data available in a distributed and decentralised data-sharing environment, where Teadal software enforces data governance policies. This environment can be modelled as a graph, where each node represents a federation site controlled by an organisation, and each edge represents a communication link between two sites.

Each federation site has its own computing resources, such as a cluster or data centre, and data products, typically assembled from a data lake. Additionally, each federation site runs the Teadal cloud computing platform (i.e., the runtime comprising Teadal's services and tools) which allows that organisation's site to share resources and data in a controlled manner. We refer to each federation site as a "Teadal node", given the fact that we model federation through a graph and a node in the graph represents a site equipped with the Teadal runtime. Thus, the term "Teadal node" highlights the critical role of the Teadal runtime in enabling data sharing within the federation.

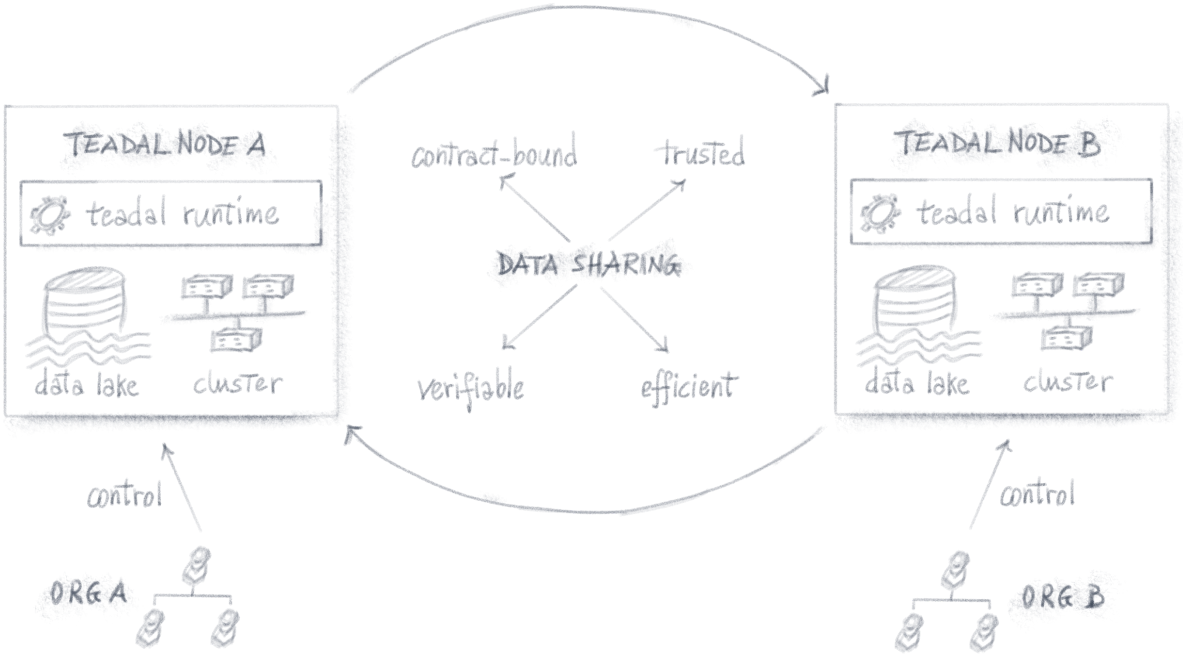


FIGURE 20: FEDERATION BETWEEN TEADAL NODES

Funded by the European Union

7.2 Interaction Patterns

The Teadal architecture supports several interaction patterns among federation sites (Teadal nodes). This section provides an overview of each pattern, focusing on interactions among Teadal nodes, and excluding the more complex interactions within individual nodes. Note that the following description outlines the types of inter-node interactions which the Teadal architecture is designed to support. Some of these patterns are already being tested in field trials, while others are still potential possibilities. In other words, this section describes the intended capabilities of the Teadal design, rather than the specific features currently implemented in the field trials.

7.2.1 Data product sharing

The FDP/SFDP concept is the cornerstone of data product sharing in a Teadal federation. An FDP sources its data from an organization's internal data product, making it available to consumers within a Teadal federation. However, consumers, who typically belong to another organisation, cannot access an FDP directly. Rather, they have to agree with the producer on specific sharing terms regarding a subset of the data which the FDP holds. The SFDP encapsulates these sharing terms and makes a specific subset of data from the FDP that is agreed upon for sharing available to the consumer.

FDPs and SFDPs are RESTful services, whereas consumers are RESTful clients. Therefore a consumer is an HTTP client process making an HTTP request to an SFDP server to query some data product. In turn, an SFDP retrieves data from its associated FDP over HTTP.
Depending on gravity and friction rules, the SFDP may be deployed either in the consumer's Teadal node or in another federation node, typically the one hosting the FDP. Consequently, there are two interaction scenarios between the Teadal nodes involved in a data product sharing transaction:

- Decentralised SFDP. The FDP and SFDP are in separate Teadal nodes. In this case the consumer typically is in the same Teadal node as the SFDP. Thus, the interaction between the two nodes consists of HTTP traffic between the SFDP and the FDP.
- Centralised SFDP. Both the FDP and SFDP are in the same Teadal node. In this scenario, the consumer is deployed in a separate Teadal node than the FPD/SFDP pair and the interaction between the nodes consists of HTTP traffic between the consumer and the SFDP.

Funded by the European Union

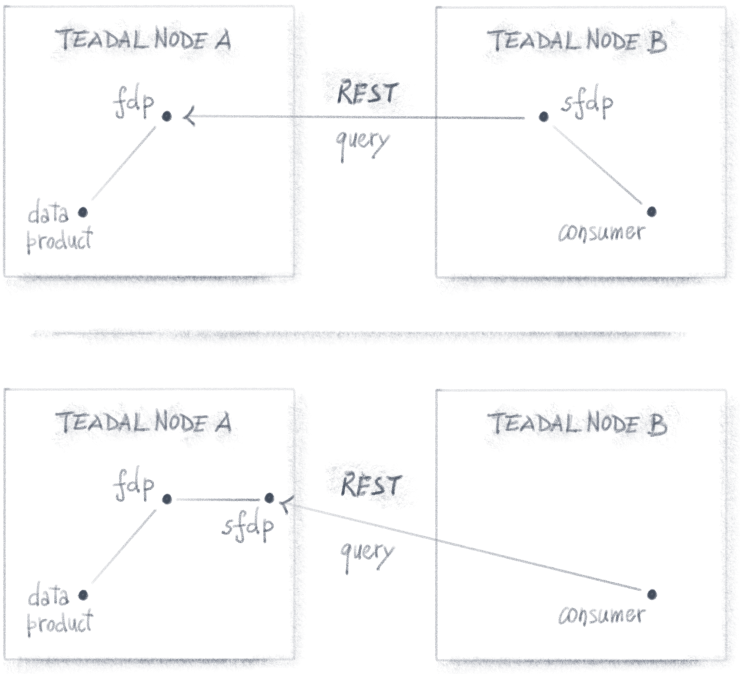


FIGURE 21: FEDERATED DATA PRODUCT SHARING BETWEEN NODES

7.2.2 Product discovery

The Catalogue is a central element of the Teadal architecture, facilitating data product discovery, contract management, and other essential functions. It stores all the data needed for consumers to discover FDPs within a Teadal federation and offers convenient means to query that information.

Each Teadal node in a federation may deploy its own Catalogue. In this decentralised setup, all Catalogue instances share the same information about the available FDPs across the entire federation. For example, if Teadal node A offers FDP#1 and node B offers FDP#2, both Catalogue instances—one in node A and one in node B—will contain data about both FDPs.

When a centralised catalogue is not used in a federation, there are multiple means of achieving replicating metadata on different nodes and therefore allow advertisement of data offerings in the federation:

- Sharing a common middleware
- Polling all the members of the federation
- Notifying all the members of the federation

The main difference in the approaches relies on what the members of the federation are willing to share in terms of software components. If the federation wants and agrees on how to split the effort required to maintain a shared infrastructure, a middleware such as a Redis database or a message broker (like Kafka) can be used. In this case, a common identity

Funded by the European Union

system could also be employed to guarantee the identity of the members of the federation, like a common Keycloak server federated with each identity system belonging to participants.

An example of this approach is represented by the case of Catalogues sharing FDP information by replicating FDP data in their own Redis databases. For example, the Catalogue in node A could be configured as a master whereas the Catalogue in node B would be a replica, pulling data from the master.

Another possibility in the case of a shared infrastructure is to have a centralised Catalogue serve a whole Teadal federation. In this case the Catalogue runs in only one Teadal node but it still holds information about the available FDPs across the entire federation. For example, if Teadal node A offers FDP#1 and Teadal node B FDP#2, a centralised Catalogue deployed in node A will reference both FDP#1 and FDP#2.

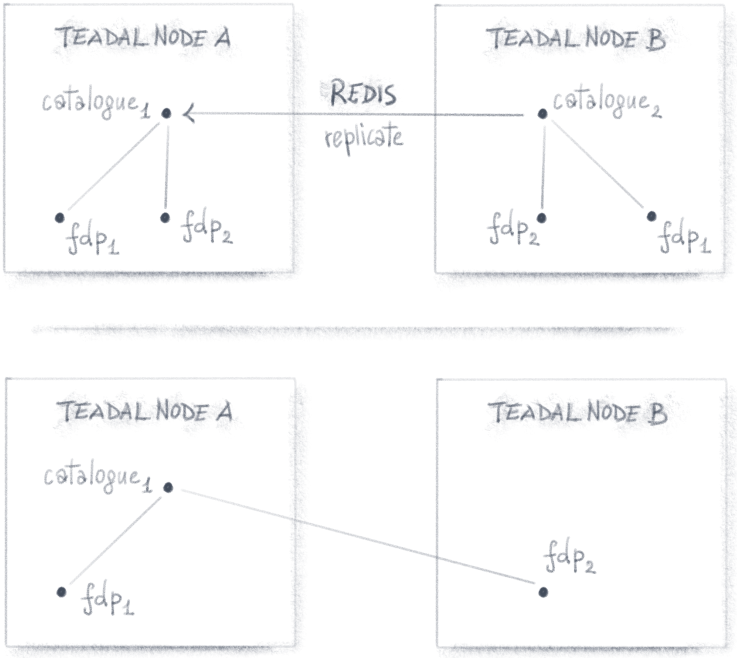


FIGURE 22: CATALOGUE FEDERATION BETWEEN TWO TEADAL NODES

Sharing infrastructural elements in a federation is usually considered easy from the technological point of view, but very difficult in organisational terms. The main problem is indeed the fact that to be able to share costs, a new joint company or initiative should be created to take care of the common infrastructure, and that is perceived as an organisational burden to be avoided if possible. Minimum commitment approaches are possible in such cases, compensating the loss of efficiency from the technological point of view with increased coordination and information exchange between humans.

In case of a zero (technological) commitment federation where participants do not want to share any infrastructure, achieving effective coordination is still possible. The main problems to be solved are:

- The list of the federation members must be known. This can be achieved by distributing periodically a document listing all the members to every member of the

Funded by the European Union

federation. The number of participants is a slowly changing list therefore the effort for manually updating the systems of each member is low.

- The identity systems of each federation member must include service accounts to be used by the other parties. Even if sharing an identity system is advisable, from a technological point of view this is still achievable. The service accounts would be used to perform API calls required to align the contents of all the Catalogues.
- A shared metadata model must be used. This is a requirement for a healthy federation to ensure that the assets to be shared are all described in the same way.

Once those requirements are fulfilled (with the identity of the federation members being the only truly critical issue), it is possible to achieve alignment of the Catalogues belonging to each federation member via polling or via notification. Both in the case of polling or notification, each Catalogue just requires a list of endpoints and identities to be used when calling each other Catalogue. In the case of polling, an internal periodical task inside the Catalogue would then call all the federation endpoints to collect “external” metadata (so, a list of FDP made available in the federation). In the case of notification, the lifecycle process linked to the publication of FDPs would include a task like “notify the members of the federation”, whose implementation would cycle through the federation members and call the notification endpoints of each participant to inform them of the availability of a new FDP.

As explained in D2.3 there is also another way of achieving the result of obtaining a federation with minimum (not zero) technological commitment, that is via Dataspaces. In Dataspaces, metadata about data offerings (FDPs) would be shared by means of Dataspace Connectors, agents which share a common identity system and a common metadata model which are able to perform secure communication of both data and metadata (as depicted in Figure 23).

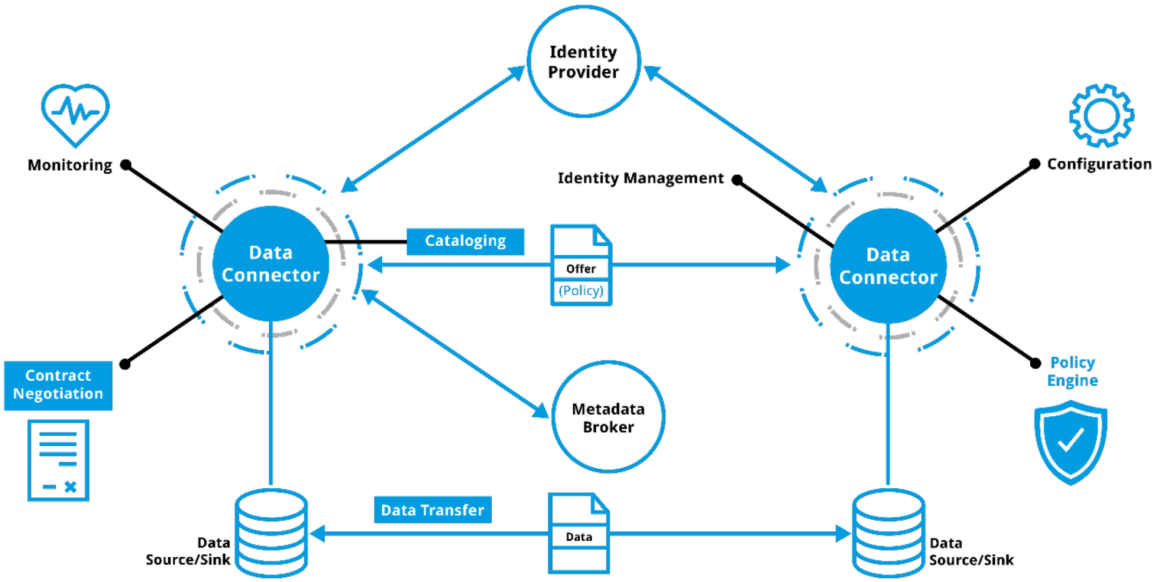


FIGURE 23: FEDERATION VIA DATASPACE

Funded by the European Union

Such an architectural layout allows addressing the main problem related to zero technological commitment, represented by identity management of the participants to a federation. As we already described in the previous deliverable, TEADAL architecture is compatible with such an approach and therefore is conceptually and technologically possible to achieve metadata alignment via Dataspace Connectors. Dataspaces also cover the actual data transfer, so it is also possible to add an extra secure layer to the consumption of sFDPs.

7.2.3 Identity federation

In a Teadal federation, each Teadal node operates independently, with its own Identity Management (IdM) system and policy framework, managing the authentication and authorisation of users. Each IdM maintains a local user directory and supports identity-related services for its node's users. However, when nodes need to collaborate, it's possible for them to share user identities securely without replicating user databases across nodes. This is accomplished through identity federation, a concept that allows one Teadal node to import users from another node's IdM system.

Teadal supports identity federation through the OpenID Connect (OIDC) protocol, a widely adopted standard for authentication. Using OIDC, one IdM (the federated IdM) can authenticate users from another IdM (the imported IdM) by leveraging a secure identity token exchange. When an identity federation is established between two Teadal nodes, one IdM becomes the identity provider (IdP) and the other the service provider (SP). The identity provider authenticates the user and provides an OIDC-compliant identity token to the service provider, which is then passed on to the policy framework. The policy framework validates the token signature and decides whether to grant access to the data and services available at that node, depending on the permissions found in the token.

This method of federating user identities offers several benefits:

- Seamless user access. Users in one Teadal node can access resources in another Teadal node without needing separate credentials, simplifying user management across the federation.
- Security and trust. Since identity information is transmitted securely via the OIDC protocol, Teadal nodes can ensure that only authenticated users from trusted sources are granted access to resources. Specifically, as mentioned earlier, in a Teadal federation, a product consumer is typically located on a different node than the one where the data product is maintained. With identity federation, the policy framework can securely identify and grant access to consumers across the federation.
- Decentralised control. Each Teadal node retains control over its own user directory, while still being able to federate identities with other nodes as needed.

As an example, consider two Teadal nodes: node A and node B. Each node runs its own Keycloak instance for managing users. IdM#1, the Keycloak instance in node A, holds user#1 and user#2, while IdM#2, the Keycloak in node B, holds user#3 and user#4. To enable collaboration between these nodes, IdM#1 can be configured as the identity provider (IdP) for node B. Using the OIDC protocol, IdM#2 (acting as the service provider, or SP) can request user identities from IdM#1 whenever an authenticated user needs access to resources in node B. As a result, user#1 and user#2 become available to IdM#2, and thus to the policy framework software running in node B. This identity federation allows for seamless,

Funded by the European Union

secure data sharing between the two nodes without the need for redundant user management.

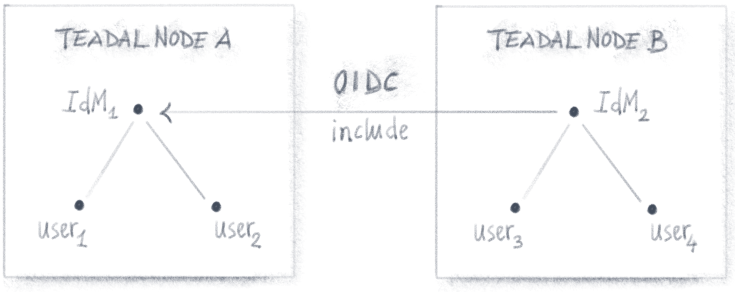


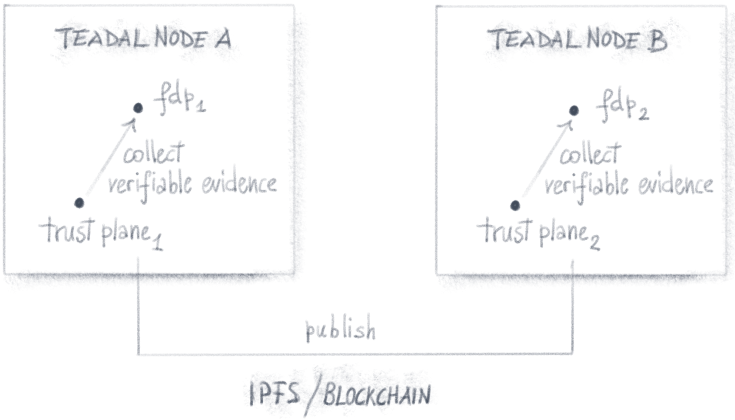
FIGURE 24: IDENTITY FEDERATION BETWEEN TWO TEADAL NODES

7.2.4 Trust Establishment

How to establish trust in a Teadal federation? While federated data products, the Catalogue, identity federation, and security policies provide the means to securely and efficiently share data among Teadal nodes according to agreed-upon terms, an additional layer of trust is often necessary. Organisations involved in data sharing must have confidence that data are stored, handled, and exchanged in compliance with their specified terms. Trust, therefore, must extend beyond the boundaries of a single Teadal node to encompass all federation nodes involved in a data transaction.

The Teadal Trust Plane, a groundbreaking innovation, offers a solution for establishing trust within a federation. This Trust Plane is an evidence-based trust framework that enables all parties involved in the data exchange process to review verifiable evidence ensuring that the Teadal runtime executes data transactions as expected. By utilising Advocate and privacy-preserving technologies, the Trust Plane generates and collects verifiable evidence throughout the Teadal infrastructure. Importantly, this evidence is linked back to the individuals responsible for each data transaction. The collected evidence is published on a decentralised, distributed storage system (IPFS) and anchored to the Teadal blockchain for immutability and transparency. This storage and blockchain infrastructure spans across all the nodes in the federation.

Thus, the Trust Plane plays a pivotal role in establishing trust in the system by ensuring that the observed behaviour of the system aligns with predefined expectations.



Funded by the European Union

FIGURE 25: ESTABLISHING TRUST IN A TEADAL FEDERATION

7.2.5 Resource allocation

In a TEADAL federation, efficient resource allocation is essential to optimize the performance and scalability of distributed data services. While each TEADAL Node operates its own Kubernetes cluster independently, coordination across nodes is needed to enable the federation to act cohesively when deploying and executing Federated Data Products (FDPs) and Shared Federated Data Products (sFDPs).

The key capability required to support resource coordination across a TEADAL federation is resource pooling, a feature supported by several existing cluster federation and multi-cluster orchestration solutions. This concept resembles *cloud bursting*, a practice envisioned in the early phases of cloud computing, where the resources of one cloud provider could be used to execute workloads admitted by another, under infrastructure-sharing agreements.

In TEADAL, this can be illustrated by a scenario shown in Figure 26 that involves two Teadal nodes—node A and node B—each running its own Kubernetes cluster. Node A has available computational resources: resource #1 (CPU) and resource #2 (GPU), while Node B has computational resources #3 (memory) and #4 (storage). With resource pooling in place, node B can utilise resources from node A, such as CPU (#1) or GPU (#2), despite the fact that these resources reside in a different physical cluster. Resource pooling abstracts the boundaries between clusters, allowing Kubernetes in node B to schedule tasks onto resources that exist in node A as though they were part of the same local resource pool. This capability is particularly beneficial in environments where resource demands fluctuate or where a single Teadal node might not have enough resources to meet the needs of specific workloads. With resource pooling, a Teadal federation can improve resource utilisation and ensure that computational resources are allocated where they are most needed, regardless of the cluster or node they physically reside in. This enhances overall efficiency and provides more flexibility in resource management, allowing organisations to dynamically scale workloads in a distributed environment.

In earlier iterations, TEADAL technology stack included integration with Kubestellar, an open-source project designed to enable cross-cluster workload federation and resource pooling through shared scheduling abstractions such as *spaces*. However, as the architecture evolved, this integration was discontinued for two primary reasons:

Industry maturity: Production-grade environments increasingly rely on vendor-backed solutions that provide robust support for workload federation, service meshes, and secure identity propagation. Re-implementing such capabilities using open-source tooling like Kubestellar offered limited innovation potential and would have introduced significant operational overhead, extending beyond TEADAL's scope.

Strategic focus: Given TEADAL's prototype nature and its focus on facilitating trusted and efficient inter-organizational data sharing, the project team prioritized investments in core innovations, such as secure data product modeling, automating service generation, and privacy-preserving transformations, over reproducing general-purpose multi-cluster infrastructure functionality.

Funded by the European Union

In the updated TEADAL design, data products are standalone, self-contained services. Upon creation, each service is annotated with Kubernetes manifests, policy files, and labels representing declarative constraints and resource profiles—such as preferences for compute capacity (CPU, GPU, memory), data locality, policy alignment, and compatibility requirements. These deployment artifacts are then used to guide placement decisions across available TEADAL Nodes. A lightweight controller—equipped with an Inventory of available resources and real-time monitoring of resource utilization and load—selects the most suitable deployment target based on current conditions and data product constraints.

This approach effectively mimics the core functionality of resource pooling, enabling meaningful experimentation and validation in the prototype phase, while leaving the door open for future integration with production-grade federated control planes when transitioning to real-world deployments.

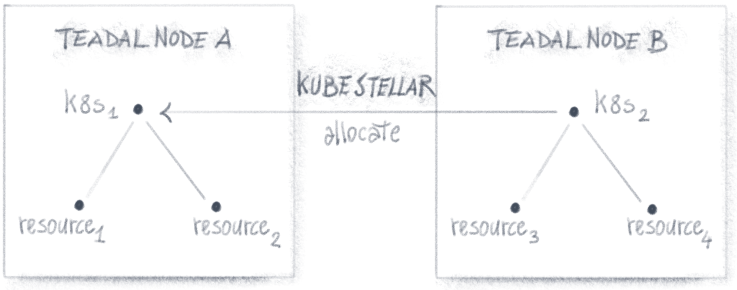


FIGURE 26: SHARING COMPUTING RESOURCES BETWEEN TWO TEADAL NODES

The interaction between the TEADAL nodes forms the operational backbone of the network and enables distributed data exchange, control and trust across organisational boundaries. The architecture supports scalable, secure and verifiable collaboration through clearly defined patterns, such as the shared use of data products and catalogue networks as well as identity networks, trust building and resource coordination. Each node works independently, but can be seamlessly integrated using common protocols and interoperable services. This modular yet coherent design allows TEADAL to accommodate different levels of technological and organisational commitment and ensure flexibility without compromising on compliance or performance. The federation model outlined here demonstrates TEADAL's ability to support real-world data ecosystems with different requirements and governance structures.

Funded by the European Union

8. DEPLOYMENT AND CI/CD INSTRUCTIONS

This section provides an overview of the deployment and CI/CD documentation for the TEADAL Node, as outlined in the README.md file available in the GitLab repository.

The README.md file on the Gitlab repository serves as a crucial resource, providing clear and structured deployment and CI/CD instructions for TEADAL. It is designed to assist individuals who have never worked with TEADAL, ensuring they can quickly understand and utilize the developed work.

This document includes step-by-step guidance on setting up the deployment environment, configuring CI/CD pipelines, and automating workflows. By following these instructions, users can efficiently integrate TEADAL into their infrastructure while maintaining consistency and reliability in deployments.

The README.md file is an essential part of the TEADAL Node documentation, enhancing accessibility and ensuring smooth adoption of TEADAL by new developers, teams or companies.

8.1 Prerequisites of the deployment

The prerequisites for deploying the TEADAL Node are defined by each Pilot. Since each Pilot has its own version of the TEADAL Node, it also has specific requirements. More information on this can be found in Deliverable D6.2.

To ensure a successful deployment, users should refer to the documentation provided by their respective Pilot, which outlines the necessary dependencies, environment setup, and configuration steps tailored to that version of the TEADAL Node.

8.2 Secrets configuration

The configuration of secrets should be done by the developer responsible for deploying the TEADAL Node. The baseline secrets configuration of the TEADAL Node can be found on the Quickstart guide in the GitLab repository.

8.3 ArgoCD Setup

Instructions for setting up ArgoCD are included in the Quickstart guide within the README.md. This guide walks users through the full setup process.

Once all configurations, such as secrets and ArgoCD, have been completed, the TEADAL Node deployment and CI/CD workflows should function as expected, providing a stable and consistent deployment pipeline.
For more detailed information or to explore specific aspects of the CI/CD deployment process, additional information can be found in related project deliverables, including D6.2.

Funded by the European Union

9. ARCHITECTURAL FIT FOR PURPOSE

This chapter evaluates how the TEADAL architecture effectively meets both the cross-cutting and pilot-specific requirements identified in Section 2. It analyzes the implementation of essential architectural principles, such as privacy, data sharing, and cataloguing, across all pilots through standardized mechanisms. Additionally, Figure 27 illustrates the interaction flow between TEADAL components, including the onboarding of FDPs, access negotiation, transformation into sFDPs, policy validation, secure data delivery, and audit logging. This integrated view highlights how the overal architecture supports federated data sharing in diverse application domains.

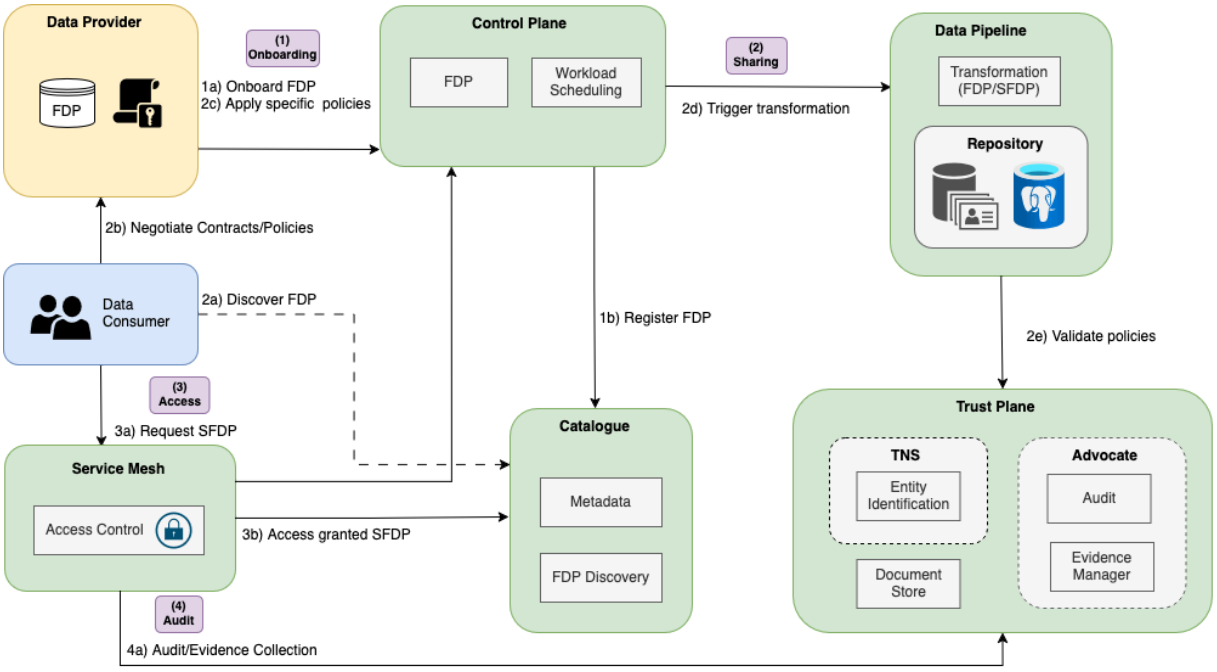


FIGURE 27: FIT FOR PURPOSE ARCHITECTURE

9.1 Implementation of the cross-cutting requirements in the components

The cross-cutting requirements (Section 2.2) form an essential basis for the implementation of all use cases in the TEADAL project. They relate in particular to the responsible handling of data, standardised processes for data processing and transfer, and transparency in data access. The following subsections explain how these requirements are specifically addressed within the technical and organisational infrastructure of TEADAL.

9.1.1 Policies and privacy:

The protection of personal and sensitive data is consistently pursued in the TEADAL project. Table 10 presents cross cutting implementation regarding policies and privacy. This is based on generally applicable guidelines for handling data (C1-Pol01), which are based on European data protection standards (e.g. GDPR, ISO 27001). All project data, including pilot data, contact data and questionnaires, is processed exclusively on a project-related basis and in accordance with the applicable data protection regulations. If necessary, personal data

Funded by the European Union

is anonymised before further processing, whereby the respective data controller is responsible for compliance with these requirements.
To implement the confidentiality and access control requirements (C1-Pol02), access to data within the project is only granted to authorised partners. On the infrastructure side, this is achieved through dedicated Kubernetes clusters of the individual pilots, which are connected to the central TEADAL infrastructure via standardised interfaces. Access rights are clearly defined and regulated based on roles (e.g. read, edit, forward). In addition, the traceability and documentation of consent to data use (C1-Pol03) is guaranteed. Techniques such as data anonymisation and data synthesis are used for all sensitive data records. The procedures used are documented in Deliverable D2.1 and in the Data Management Plan (D1.2) and harmonised across the pilot.

ID	Description	How addressed
C1-Pol 01	Definition of clear rules for storing, copying, forwarding and deleting data in compliance with regulations.	Data processing is exclusively project-related and in accordance with the GDPR; personal data is anonymised before it is stored or shared. The data controller is responsible for all processing.
C1-Pol 02	Ensuring the confidentiality of sensitive information and regulations for access rights (read, edit, use, forward).	Access to data is possible in TEADAL pro Pilot and only for authorised partners. Access rights are regulated via infrastructure areas (Kubernetes Cluster, API Handler).
C1-Pol 03	Requirements for the documentation and traceability of consents to data use and disclosure.	Consent and transformation methods (e.g. data synthesis, anonymisation) are documented. The methods for data sharing are described in D2.1 and D1.2 (Data Management Plan).

TABLE 10: CROSS-CUTTING IMPLEMENTATION: POLICIES & PRIVACY

9.1.2 Data Processing & Sharing

Efficient and secure data processing is a central component of the project architecture. In order to meet the requirements for standardised data processes (C2-Proc01), a shared infrastructure is used that is based on a Kubernetes-based environment. This enables both real-time and batch processing, customised to the specific needs of the respective use cases. The data to be processed can have different formats (e.g. XML, JSON) and originate from both internal and publicly accessible sources (e.g. weather or map data).

Various mechanisms for data cleansing (sanitisation), normalisation and anonymisation are provided to meet the requirements of C2-Proc02. For example, synthetic data sets are generated on the basis of real data (e.g. UC 1) or structured attribute distributions (e.g. UC 2 and UC 5) in order to ensure data protection and at the same time enable realistic tests.
A standardised communication architecture has been implemented for the secure and harmonised exchange of data (C2-Proc03) between internal and external stakeholders. This is based on CI/CD API handlers and enables controlled interaction with the core

Funded by the European Union

infrastructure. When using public cloud services (e.g. AWS or Azure), additional security measures and automatic backup functionalities are used. Table 11 summarizes cross-cutting implementation details related to data processing and sharing.

ID	Description	How addressed
C2-Proc01	Provision of technical mechanisms for efficient, secure and standardised data processing and forwarding (batch and real-time processing).	The infrastructure is based on Kubernetes clusters in which data processing components can be operated flexibly. Real-time and batch processing are possible.
C2-Proc02	Support for different data storage systems (e.g. SQL) as well as methods for normalisation, data sanitisation and anonymisation of data.	The data formats used (e.g. XML, JSON) and connections with open data sources (e.g. weather data) are part of the processing. Synthesis and anonymisation are used to protect privacy.
C2-Proc03	Harmonisation of data processing and data exchange between internal and external stakeholders with secure transfer to third parties.	A harmonised exchange takes place via the common core infrastructure (e.g. CI/CD API handler), including connection to external cloud systems (e.g. AWS, Azure).

TABLE 11: CROSS-CUTTING IMPLEMENTATION: DATA PROCESSING & SHARING

9.1.3 Data access & cataloguing

To ensure structured and transparent access to project data, the introduction of a central data catalogue is planned (C3-Acc01). This serves as a platform for searching and classifying available databases and integrates domain-specific ontologies. This ensures that internal project data can be found consistently and utilised efficiently.
Transparency in data access (C3-Acc02) is realised through clearly defined access rights and the traceability of access. Each pilot manages its data infrastructure independently and ensures that only authorised persons have access to the respective data. Access takes place exclusively via secure interfaces.

With regard to metadata allocation (C3-Acc03), the systematic assignment of descriptive tags is planned, which, for example, depict the data source, data format, processing status or purpose reference. This enables semantic searchability of the data catalogue, which is particularly relevant for the further use and analysis of data. Table 12 summarizes cross-cutting implementation details related to data access and cataloguing.

ID	Description	How addressed
C3-Acc01	Provision of a data catalogue and domain-specific ontologies to make available data clear and efficient to find.	A central data catalogue system is provided for the use cases to facilitate structured searches and access to data.

Funded by the European Union

C3-Acc 02	Transparency and traceability of data access, e.g. insight into which data was accessed or used by whom.	Each pilot independently controls who can access which data. Access takes place via clearly defined interfaces.
C3-Acc 03	Assignment of Metadata with one or more tags describing the data contained.	Metadata is assigned during cataloguing, for example for categorisation by data type, source or processing status.

TABLE 12: CROSS-CUTTING IMPLEMENTATION:DATA ACCESS & CATALOGUING

9.2 Implementation of the pilot-specific requirements

For each of the pilot-specific requirements from the various use cases, the following section shows how the respective requirement was technically implemented within TEADAL. Both the specific implementation and the components responsible for it, as defined in Section 6 are discussed. The functional relationship between the formulated requirements from Section 2.3 and the services and mechanisms provided by TEADAL is shown.

9.2.1 Evidence-based medicine

In the Evidence-based Medicine use case, the focus is on the automation of study processes (P1-Arch02), data protection-compliant access to patient data (P1-Arch06, P1-Gen06) and the traceability of medical studies (P1-Arch09, P1-Gen03). The requirements are met by the use of data pipelines (P1-Mgmt02), a federated catalogue (P1-Mgmt01, P1-Gen05), the Trust Plane with the Advocate (P1-Gen03, P1-Arch09), dynamic access policies and the Service Mesh (P1-Gen06, P1-Arch06), among other things. Studies can be set up automatically (P1-Arch02), relevant patient data can be found in a targeted manner (P1-Mgmt01, P1-Mgmt03) and access can be controlled by sets of rules. In addition, all actions are systematically documented, which ensures integrity and traceability across multiple study steps. Table 13 list cross-cutting implementation details related to evidence-based medicine pilot use case.

ID	Description	Implementation	Component
P1-Arch02	The architecture has to facilitate as much as possible the automation of the process to set up the study.	Studies are set up automatically using data product lifecycle management and orchestrated pipelines (Kubeflow).	Control Plane, Data Pipelines
P1-Mgmt01	Provide a data catalogue that tells the study promoter where he will find the patients that it can study.	The federated catalogue enables search queries for suitable patient data and data records.	Catalogue

Funded by the European Union

P1-Mgmt02	Provide a variety of data sanitization methods with callbacks to data owners (some in TEADAL, others are plugged in by use-cases)	Sanitisation methods such as differential privacy and anonymisation are offered and integrated through data pipelines.	Data pipelines
P1-Gen03	Offer means to measure sustainability related metrics for each action taken by TEADAL.	All actions are monitored and measured; metrics are part of the monitoring and audit data.	Monitoring, Advocate
P1-Gen05	Facilitate the bureaucracy of a study proposal.	Automated workflows simplify study processes, and registrations run via federation services.	Catalogue, Control Plane
P1-Mgmt03	Allow to extract information about the amount of patients affected (SELECT COUNT) to know the strength of the study.	Direct queries (‘SELECT COUNT’) via standardised data access APIs on the data products.	Data product API
P1-Gen06	Allow to approve access to personal data to certain "study promoters" by comparing basic rules.	Access is rule-based and controlled by RBAC and dynamic policies via Service Mesh.	Service Mesh, Policies.
P1-Arch06	Provide technical components to enable GDPR like compliant data access, e.g., tools to anonymize, tools to check anonymity or tools to check informed consent.	Compliant data access is achieved through predefined policies and RBAC. Furhtermore by incooporating smart contracts Advocate ensures a olicy-based validation of claims in compliance with governance policies.	Service Mesh, Advocate

Funded by the European Union

P1-Arch09	Provide some kind of persistency for multiple-step studies.	Every step is documented and persistent evidence is stored.	Trust Plane (Advocate)

TABLE 13: PILOT-SPECIFIC IMPLEMENTATION: EVIDENCE-BASED MEDICINE

9.2.2 Mobility

For Mobility, the focus is on federated, controlled and transparent access to mobility data from various sources such as the National Access Point (NAP), Regional Access Point (RAP) and local transport providers. Requirements such as access to road network data (P2-Gen01), timetables (P2-Gen02) and arrival times (P2-Gen03) are implemented via the federation catalogue and standardised interfaces. Transparency in the use of data (P2-Gen07), price information (P2-Gen06) and controlled access and data release by transport service providers (P2-Privacy01, P2-Mgmt03, P2-Mgmt04) are controlled via policies and the service mesh. The traceability of all data accesses (P2-Mgmt02) and the validation of access restrictions (P2-Policy03) are carried out via the Trust Plane and the Advocate component. Access to the federation itself (P2-Arch01) and the visibility of data is regulated on a role-based basis. Table 14 list cross-cutting implementation details related to mobility pilot use case.

Req.	Description	Implementation	Component
P2-Gen01	Access Open Street Map for the street network of the city/region involved.	OSM data is integrated into the catalogue via external sources.	Catalogue
P2-Gen02	To be able to take information from the NAP about the schedules of all public transport operators in a specific area.	Connection of NAP providers to the federation catalogue for access to timetables.	Catalogue
P2-Gen03	To access data from the operator on the stop arrival times.	Access to current public transport arrival times via APIs.	Data Product API
P2-Gen06	Include also information about the average price of the trip, without the need to give the opportunity to book the trip.	Prices are attached to the travel data products as additional metadata.	Catalogue

Funded by the European Union

P2-Gen07	To offer transparency of data access across NAP, RAP, Local, etc. (e.g. allow the NAP to see what happened to the data downstream).	All transfers are transparently documented and verifiable via Advocate.	Trust Plane (Advocate)
P2-Privacy 01	Each Transport Service Provider (TSP) should be able to prevent a subset of his data to be shared with other TSPs (e.g. competitors).	Access to sensitive data is restricted by policies.	Policies (PDP, PEP)
P2-Arch01	To allow for joining the RAP or NAP TEADAL federation to access available data.	Access to federation is controlled over an identity management system, which restricts data asset visibility to authorized users with the appropriate credentials.	Catalogue
P2-Arch02	To provide only updated data, i.e. datasets can be dynamic so that only last versions should be available.	Data can be updated regularly in the data catalogue.	Catalogue
P2-Mgmt02	The TSP should have a way to check who accessed or requested data at the national level.	All accesses are logged and stored in an audit-proof manner within the trust plane.	Trust plane (Advocate)
P2-Mgmt03	The TSP should have a way to decide whether a dataset should be shared in the federation.	The TSP can decide on release through federated policies which can be defined in the Date Catalogue.	Catalogue
P2-Mgmt04	Each TSP should be able to access data belonging to other TSPs through the RAP or NAP.	Catalogue entries are federated and the access is regulated by policies.	Catalogue, Service Mesh
P2-Policy03	Enable a mechanism to prohibit access or sharing of data that should not be available to the NAP or RAP.	Policies prevent illegal access and complianc is ensured through validation within the trust plane.	Policies, Trust Plane

Funded by the European Union

TABLE 14: PILOT-SPECIFIC IMPLEMENTATION: MOBILITY

9.2.3 Smart Viticulture

The Smart Viticulture pilot focuses on the secure and controlled exchange of data between consumers, farms and companies. The requirements include the targeted support of consumer-to-business (P3-Gen01), consumer-to-consumer (P3-Gen02) and business-to-business (P3-Gen03) data sharing, which is secured by dynamic policies, data protection mechanisms and verification components such as the Advocate. The technical implementation takes place via data pipelines, federated policies and the Trust Plane. In addition, as part of P3-Arch01, the classification of data records along geographical and domain-specific characteristics is implemented in the Continuum via the catalogue. Table 15 list cross-cutting implementation details related to smart viticulture pilot use case.

ID	Description	Implementation	Component
P3-Gen01	Support C2B.	Data products can be released as SFDPs to companies by consumers through policies.	Policies, Data Pipeline
P3-Gen02	Support C2C (e.g. Vineyard operator-A to Vineyard operator-B).	Data Consumers can share data between themselves, with rules to protect privacy.	Policies, Data Pipeline
P3-Gen03	Support B2B sharing (e.g. Terraview to insurance company).	Companies share data securely via contractual and technical protection (policys + evidence from Advocate).	Policies, Data Pipeline, Trust Plane
P3-Arch01	The datasets should be placed in the continuum according to pilot-case specific notions, including geography.	The data is stored in the catalog with its specific metadata.	Catalogue

TABLE 15: PILOT-SPECIFIC IMPLEMENTATION: SMART VITICULTURE

9.2.4 Industry 4.0

In the ‘Industry 4.0’ use case, the focus is on standardised data formats, federated data storage and differentiated access mechanisms. The requirements include the provision of uniform data access via APIs for reporting purposes (P4-Arch02), the establishment of

Funded by the European Union

harmonised protocols for data exchange (P4-Arch03) and the standardisation of incoming data streams from different plants (P4-Gen01). Implementation takes place via the control plane and the federated catalogue. In addition, it is ensured that data processing takes place within the ERT infrastructure (P4-Gen05), while guidelines for the handling of KPIs (P4-Policy03) are implemented via the service mesh and corresponding access policies. Table 16 list cross-cutting implementation details related to industry 4.0 pilot use case.

ID	Description	Implementation	Component
P4-Arch02	Providing an easy-access interface for querying the data to generate the reports (e.g., API for accessing data).	Standardised APIs allow automated querying and reporting.	Data Product API, Catalogue
P4-Arch03	Create an harmonised/standard protocol for data collecting, processing and information sharing with different plants, departments and teams.	Standardised protocols for data recording and forwarding via federation nodes are set up during onboarding..	Control plane
P4-Gen01	Standardising information coming from different plants.	Incoming data records are standardised and converted to TEADAL-compliant formats during onboarding.	Control plane
P4-Gen05	Ensure that data is processed and stored using ERTs infrastructure.	Data is stored in a federation-wide and decentralised catalogue within ERTs intfrastructure, under Federation control.	Control plane, Catalogue
P4-Policy03	Define rules (aggregation, obfuscation, etc.) for visualising data, KPI values and/or KPI categories, reports that are accessed by users with restricted rights.	Access to KPIs and reports is aggregated and filtered by policies.	Service Mesh

TABLE 16: PILOT-SPECIFIC IMPLEMENTATION: INDUSTRY 4.0

Funded by the European Union

9.2.5 Financial Data Governance

The ‘Financial Data Governance’ pilot addresses the data protection-compliant handling of sensitive financial and production data, particularly in the context of KYC models, production data analyses and ROI calculations. Requirements such as the protection of privacy during calculations (P5-Privacy01) and the ability to dynamically activate confidential computing environments (P5-Arch01) are met by integrating Trusted Execution Environments (TEEs) into the data pipelines. The secure and traceable data flow is ensured by an interplay of guidelines (P5-Policy01), the Service Mesh and the Trust Plane, whereby national specifications for authorisation and certification are taken into account. The traceability of data processing and the preservation of evidence (P5-Mgmt01) are automated and cryptographically secured. In addition, metadata descriptions enable intelligent data forwarding in the mesh (P5-Mgmt04), while a coupled FDP-SFDP structure (P5-Gen01) supports the combination of production data with market information and analysis infrastructures. This is implemented via central components such as data pipelines, service mesh, trust plane and control plane. Table 17 list cross-cutting implementation details related to financial data governance pilot use case.

ID	Description	Implementation	Component
P5-Privacy01	Privacy should be preserved for computation tasks (for KYC model creation at least)	TEEs can be integrated into data pipelines, for executing confidential computations, and maintaining the integrity and privacy of sensitive data.	Data pipelines, Trust Plane
P5-Policy01	Data transfer and the processing of sensitive data must be defined in policies, controlled and authorised by national guidelines (including certification by authorities).	Before data transmission, the TEE environment is certified by national authorities and policies are set up.	Service Mesh, Policies
P5-Mgmt01	Data processing must be traceable and auditable; every transfer and processing must be backed up with evidence. After processing, the outputs must be securely stored and	All steps are automatically documented and evidence is secured cryptographically. After processing, the TEE is deleted and the results are stored in a secure environment.	Trust Plane (Advocate)

Funded by the European Union

	the TEE must be completely deleted (incl. memory footprint).		
P5-Mgmt04	Metadata description of the data should be used as an input in the mesh for smart data movements	TEEs are automatically integrated into processing pipelines.	Data Pipelines
P5-Arch01	It must be possible to integrate the use of Trusted Execution Environments (TEEs) into the privacy-preserving data pipelines and activate them dynamically.	Production data from edge devices is aggregated, combined with financial models and analysed.	Data Pipelines
P5-Gen01	Establishment of a coupled FDP-SFDP structure to evaluate production data, ROI and financial viability by combining edge devices and analytics infrastructure.	Market prices can be included during onboarding.	Control Plane, Catalogue

TABLE 17: PILOT-SPECIFIC IMPLEMENTATION: FINANCIAL DATA GOVERNANCE

9.2.6 Regional Planning

The 'Regional Planning' pilot focuses on the federated linking and data protection-compliant analysis of sensor data (BOX2M) and administrative data (RT). Requirements such as data-driven decision support (P6-Gen02), the integration of territorial sensor data (P6-Gen01) and the combination of different data sources (P6-Gen03) are implemented via data pipelines, catalogues and federated access interfaces. Data protection and control are realised through policies (P6-Privacy02, P6-Privacy03, P6-Privacy05) as well as through the service mesh and the trust enhancing Advocate. The federation structure defines clear roles for the actors (P6-Arch01), regulates access to federated data without replication (P6-Arch02) and enables the separate administration of RT and BOX2M nodes (P6-Arch03). In addition, incorrect data can be marked as invalid for a limited time and excluded from analyses (P6-Mgmt01). Table 18 list cross-cutting implementation details related to regional planning pilot use case.

Funded by the European Union

ID	Description	Implementation	Component
P6-Gen01	Data about plants and buildings coming from the Tuscany Region must be enriched with BOX2M dynamic data coming from sensors in an aggregated territorial perspective.	Sensor and management data are merged and combined via pipelines and federated interfaces.	Data Pipelines, Catalogue
P6-Gen02	The aggregated data must be used for decision support system development.	Analyses based on aggregated data provided as the TEADAL SFDP from the data pipeline can enable decisions to be prepared for regional planning.	Data Pipeline
P6-Gen03	Users must benefit from the combined use of RT and BOX2M data and from the analytics produced on top of these datasets.	BOX2M and RT data are combined and cleaned over strinc appliance of policies. It is ensured that no raw data is part of the result.	Policies, Data Pipeline, RT Data Analytics Engine
P6-Privacy02	The aggregation of data has a minimum threshold of 3 units. No analysis can be performed if less than 3 records are affected.	Queries with too few entries are automatically blocked, threshold value is configured based on policy.	Policies
P6-Privacy03	The system must forbid the users to see confidential data about buildings and plants stored in the RT data lake.	Access is controlled and technically secured depending on the sensitivity of the data.	Service Mesh, Identity Management System
P6-Privacy05	The must be 3 level of consent referring to P6-Privacy04: BOX2M is allowed to collect building's data and move them on cloud. (mandatory)	Consent is explicitly recorded, stored and taken into account in data access using Policies and recorded evidence.	Policies and Service Mesh, Trust Plane (Advocate)

Funded by the European Union

	Plant owners give to BOX2M the consent to analyse data, but not to share them (optional) Plant owners give to BOX2M the consent to analyse data and share them (optional)		
P6-Arch01	The solution must define a sort of ecosystem where RT is at the centre and BOX2M is one of the actors who is providing data. RT is one of the main consumers of data.	Federation nodes define clear roles: RT as consumer, BOX2M as data source.	Control Plane
P6-Arch02	The solution must define a logical component which federates SIERT dataset, open data and BOX2M sensor data without replication.	Data is made federated accessible without replication and its access is via catalogues.	Catalogue
P6-Arch03	The solution must define one node for RT and one node for BOX2M.	Nodes for RT and BOX2M are integrated and managed individually in the federation.	Control Plane
P6-Mgmt01	The solution must provide the ability to mark certain data elements as invalid in a particular interval, since some field sensors could be altered by malice, neglect or vandalism	Data can be marked in the catalogue as 'invalid' for a limited time and excluded from analyses.	Catalogue

TABLE 18: PILOT-SPECIFIC IMPLEMENTATION: REGIONAL PLANNING

This chapter has shown how the TEADAL architecture systematically fulfils both the overarching and pilot-specific requirements identified in the previous sections. Through a combination of modular components such as the Control Plane, the Trust Plane, the Catalogue, the Service Mesh and the Data Pipelines, the architecture enables secure, policy-compliant and verifiable data exchange across different domains. The individual requirements of each pilot are met through targeted technical implementations, while common requirements such as data protection, data processing and cataloguing are handled through standardised and reusable mechanisms. The realisation of the requirements shows that the TEADAL architecture is not only conceptually robust, but can also be practically adapted to a variety of real-world data management challenges.

Funded by the European Union

10. CONCLUSION

This deliverable provided an overview of TEADAL overall architecture by consolidating all technical developments into a comprehensive and integrated general architecture. Building upon earlier deliverables (D2.1, D2.2, and D2.3), it presents a mature and operationally aligned view that connects the TEADAL platform's capabilities with the concrete needs of the pilots. Each pilot has been reviewed to capture the latest requirements, and the document demonstrates how TEADAL components fulfill them both individually and in combination.It provides an in-depth overview of essential system-level features, including data automation, optimization, and trust, which are fundamental to the TEADAL project. The component descriptions, cluster runtime view, and patterns of federation interaction demonstrate clear architectural maturity and readiness for deployment. By implementing advanced mechanisms like policy enforcement, auditability, and distributed orchestration, TEADAL demonstrates its potential for long-term sustainability and interoperability.

In conclusion, this deliverable integrates the technical foundation and functional requirements of the project. It demonstrates that TEADAL's modular and federated architecture is not only technically strong but also suitable for a variety of domains.

Funded by the European Union