D6.1 Testbed Design

Revision: v.1.0

Work package	WP 6
Task	Task 1
Due date	30/11/2023
Submission date	30/11/2023
Deliverable lead	ALMAVIVA
Version	1.0
Authors	ALMAVIVA (Antonio Retico, Samantha Hine, Sergio Sestili, Vincenzo Cirillo) CEFRIEL (Alessio Carenini, Ilaria Baroni)

WWW.TEADAL.EU

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

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

	MARTEL (Andrea Falconi) POLIMI (Mattia Salnitri, Pierluigi Plebani) UW (Jorge Catarino)
Reviewers	Pierluigi Plebani (POLIMI) Andrè Ostrak, Eduardo Brito (CYB)
Abstract	This deliverable presents the design of the project Testbed infrastructure for deploying the six project Pilots. It is composed of one Pilot Testbed deployment for each pilot, addressing its own specific characteristics.
Keywords	

Document Revision History

Version	Date	Description of change	List of contributor(s)
V0.1	24/08/2023	First ToC version (draft)	AlmavivA
V0.2	2/10/2023	First draft of the document	AlmavivA
V0.3	16/11/2023	Final draft ready for review	ALMAVIVA (Antonio Retico, Samantha Hine, Sergio Sestili, Vincenzo Cirillo), CEFRIEL (Alessio Carenini, Ilaria Baroni), MARTEL (Andrea Falconi), POLIMI (Mattia Salnitri, Pierluigi Plebani), UW (Jorge Catarino)
V1.0	28/11/2023	Final version of the document	ALMAVIVA (Antonio Retico, Samantha Hine, Sergio Sestili, Vincenzo Cirillo), CYB (Andrè Ostrak, Eduardo Brito), POLIMI (Mattia Salnitri, Pierluigi Plebani)

DISCLAIMER

Funded by
the European Union

Funded by the European Union (TEADAL, 101070186). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

COPYRIGHT NOTICE

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

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

DEM: Demonstrator, pilot, prototype, plan designs

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

DATA: Data sets, microdata, etc.

DMP: Data management plan

ETHICS: Deliverables related to ethics issues.

SECURITY: Deliverables related to security issues

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

Executive Summary

The **TEADAL** (“Trustworthy, Energy-Aware federated Data Lakes along the computing continuum”) project aims to develop key cornerstone technologies to create stretched data lakes spanning in the cloud-edge continuum and in the multi-cloud. The TEADAL results are in the form of a toolset for data lake technologies capable of providing trusted, verifiable, and energy-efficient data flows, both in a **stretched data lake** and across a **trustworthy mediator-less data lake federation**, based on a shared approach for defining, enforcing, and tracking privacy/confidentiality requirements balanced with the need for energy reduction.

The present deliverable describes the design of the project Testbed infrastructure for deploying the six defined project Pilots. The document starts with an introduction of the architecture design methodology, as well as a glossary of the main key concepts defined and studied during the project. In addition, an overview of the main elements of TEADAL Node resulting from architecture is presented. All above is aimed to enable the reader to better understand the presented Testbed deployment architecture, one for each one of the defined Pilot Cases.

The core of the current deliverable is then to present all the **Pilot Testbed Architectures**, each one with its own characteristics based on the corresponding use case formalised during the activities in the initial phases of the project.

Each Pilot Testbed is initially presented by describing its **topology** in terms of the required TEADAL Nodes. Then the Pilot **deployment architecture** is described as well as some **Technical Notes** and the **Connectivity Map**. The document ends with a summary of the resources made available by the Testbed sites and the way they are allocated and distributed across the described pilot Testbeds.

Funded by the European Union

TABLE OF CONTENTS

Funded by the European Union

Funded by the European Union

LIST OF FIGURES

Funded by the European Union

List Of Tables

Funded by
the European Union

ABBREVIATIONS

4C	4C software architecture model (Cloud/Data Center, Cluster, Container, Code)
AOI	Area Of Interest
API	Application Programming Interface
AWS	Amazon Web Service
CI/CD	Continuous Integration/Continuous Deployment
DAG	Directed Acyclic Graphs
DAS	Direct-Attached Storage
DC	Data Center
DevOps	Development and Operations
DL	Data Lake
FDP	Federated Data Product
GitOps	Operational approach applicable to DevOps practices
HW	Hardware
IaC	Infrastructure As Code
IAM	Identity Access Management
KPI	Key Performance Indicator
KYC	Know Your Customer
NAP	National Access Point
NAS	Network-Attached Storage
PDP	Policy Decision Point
RAP	Regional Access Point
REST	Representational State Transfer
S3	Simple Storage Service
SFDP	Shared Federator Data Product
SW	Software
TB	Tera Byte
TCP	Transmission Control Protocol
TOGAF	The Open Group Architecture Framework
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VCPU	Virtual CPU
VM	Virtual Machine
WP	Work Package

Funded by the European Union

1 INTRODUCTION

The TEADAL ("Trustworthy, Energy-Aware federated Data Lakes along the computing continuum") project mission is to provide key technologies to enable sharing data and computation among the cloud-edge continuum (gravity) managed by a single organisation and in a data lake federation environment (friction) where the nodes of such federation are managed by different entities. All this by enabling private, confidential, and energy-efficient data management.

TEADAL main goal is to develop a software toolset for data lake technologies to provide trusted, verifiable, and energy-efficient data flows, both in a stretched data lake and across a trustworthy mediator-less federation of data lakes, based on a shared approach for defining, enforcing, and tracking privacy/confidentiality requirements balanced with the need for energy reduction.

As a key aspect of the TEADAL project, different Pilot cases are defined to cover eight of our nine European common data spaces. Each Pilot case is represented in the consortium by one or more partners that provide data and are interested in obtaining a solution to address pressing data management needs at the end of the project. The selected Pilot cases refer to different domains, allowing the project to cover a broad spectrum of possible industrial scenarios. When managing different types of data, the solution developed in such scenarios may be faced with different needs for sharing them, as well as with different requirements in terms of privacy, resources allocation, data analytics, data movement, policy/access control management, energy efficiency, etc.

Moreover, the project utilises four main *Testbeds sites* that share resources to enable the deployment, in-depth testing, and operation of the TEADAL tools by creating the data lake federation needed to validate the project results in a DevOps fashion. BOX2M, MARINA, POLIMI, and TERRAVIEW are the project partners that will provide resources (e.g., storage, computation) to host data and applications. During the progress of the project other partners may contribute offering new resources if needed.

In this context the present deliverable describes the definition of the *Testbeds* environments used to deploy and run the different Pilots in all the planned project iterations to validate the Project results.

This document describes the six Pilot Testbeds, one for each Pilot, designed during the activities of the project. They are presented in their estimated final topology, also based on direct interviews with Pilot owners and on the "*D2.1 Requirements of the Pilot Cases*" and "*D2.2 Pilot cases' intermediate description and initial architecture of the platform*".

Each Pilot will address only some of the project KPIs defined in the proposal, but the totality of them will cover all the defined KPIs, to validate the Project results, according to the planning defined in the project proposal.

Funded by the European Union

2 TESTBED DESIGN METHODOLOGY

Although IT solution and infrastructure design relies on well consolidated patterns and methodologies, it is not infrequent that among different people, speaking different languages, with different IT backgrounds and knowhow, misunderstandings arise with respect to expressed concepts and terminology. Just to give an example, the name itself of *Federation* is prone to be given a different meaning by - say - a data analyst, an IT security technician, or a non-IT-savant, regional politician.

This risk is particularly high within a research project such as TEADAL, where the need to introduce entirely new IT concepts goes along with the need to make the same concepts understandable and familiar to a wide and heterogeneous audience: this is often done making use of analogies which end up sometimes "overloading" the semantic of names and to foster misinterpretation. This section is meant to minimise such risk by:

1) introducing a graphical notation - the ArchiMate language - to describe architectural concepts (section 2.1);
2) providing a glossary of the TEADAL-specific names, entities and concepts introduced and adopted within the TEADAL research project (section 2.2).

Furthermore, section 2.3 synthetically explains the approach used to produce the Testbed design.

2.1 TOGAF-ARCHIMATE® AS ARCHITECTURE DESIGN METHODOLOGY

To mitigate possible misunderstanding risks, it is very important that the project sets for itself the objective to make every concept clear and unambiguous, especially when it comes to software and contents still to be fully designed and developed. We tried in this document to match this objective by making an extensive use of the **ArchiMate®** architecture description language.

ArchiMate® is a graphical language that provides the foundations for a good and complete architecture description. It is the standard proposed by **The Open Group** (author of TOGAF framework) adopted by most specialists in the Enterprise Architecture discipline and supported by several architectural tool vendors. The language allows describing unambiguously the construction and operation of business processes, organisational structures, information flows, IT systems, and technical infrastructure. This insight helps stakeholders to design, assess, and communicate the consequences of decisions and changes within and between business domains.

This section briefly introduces the basics of this language and the conventions used throughout this document.

2.1.1 ArchiMate® for Architecture Description

In this section we'll provide a basic explanation of the language metamodel and the *legend* of the elements used in the subsequent sections.

Funded by the European Union

This section can be skipped by the reader already comfortable with ArchiMate® notation.

2.1.2 Relations

In ArchiMate®, decomposing an element into other elements of the same type is usually done through the **Composition** relationship.

This Composition relationship can be omitted if nesting is used. The following two diagrams convey the same meaning.

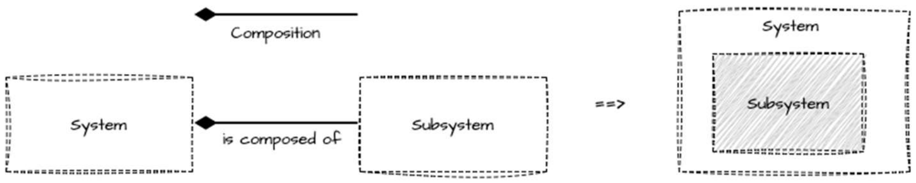


FIGURE 1 - COMPOSITION RELATION

As for **Composition**, the **Assignment** relationship can be omitted if nesting is used:

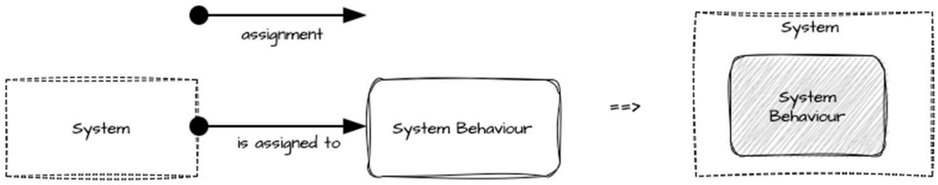


FIGURE 2 - ASSIGNMENT RELATION

For example, a *Service* is in ArchiMate® an external behaviour; it can be seen as an abstraction of *Functions* or *Processes.* Such abstraction is modelled through the **Realisation** relationship.

As for *Composition* and *Assignment*, this *Realisation* relationship can be omitted if nesting is used. These two diagrams are equivalent:

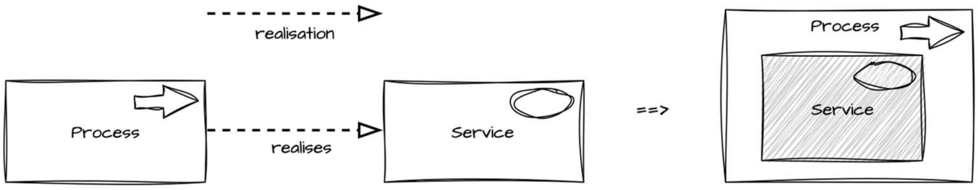


FIGURE 3 - REALISATION RELATION

Systems can exchange or share stocks through **Flows.** *Flows* can be modelled between systems:

Funded by the European Union

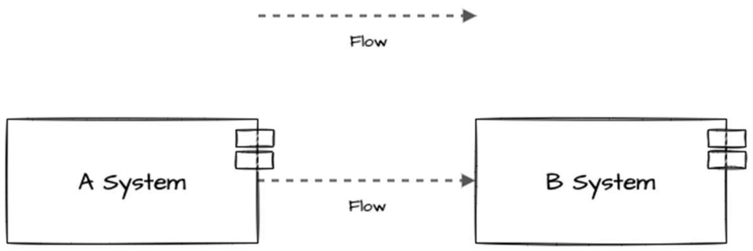


FIGURE 4 - FLOW RELATION

Between (or inside) systems, some behaviours can affect other behaviours. This impact can be immediate or delayed, intentional or unintentional. For all those cases, in ArchiMate®, we'll use the *Triggering* relationship which is used to model the temporal or causal precedence of elements. Like *Flows*, *Triggering* can be modelled between systems or processes:

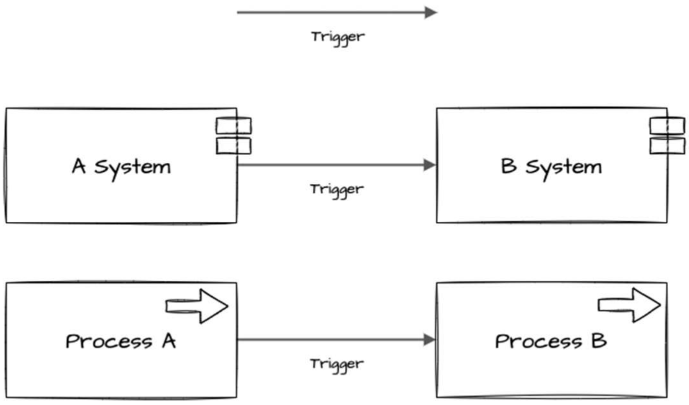


FIGURE 5 - TRIGGER RELATION

Systems exchanging stocks (i.e., Passive Structure) and affecting each other lead to dependency. In ArchiMate®, **Serving** relationship is used to denote that some systems provide its functionality (i.e., External Behaviour or Service) to other systems.

As seen with Flows and Triggering, Serving can be modelled between systems (i.e., Active Structure), between their Behaviours or a mix of both. By the way, unlike Flows and Triggers which are often used between parts of the same system, Servings are almost always used between two different systems.

Funded by the European Union

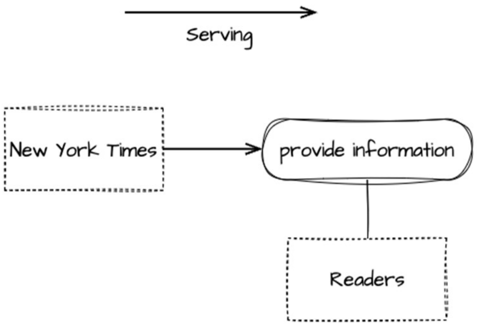


FIGURE 6 - SERVING RELATION

2.1.3 ArchiMate® objects

In our exploration of the TOGAF-ArchiMate® as an architectural design methodology, the following diagram presents a concise visualisation of the various ArchiMate® objects employed throughout this documentation.

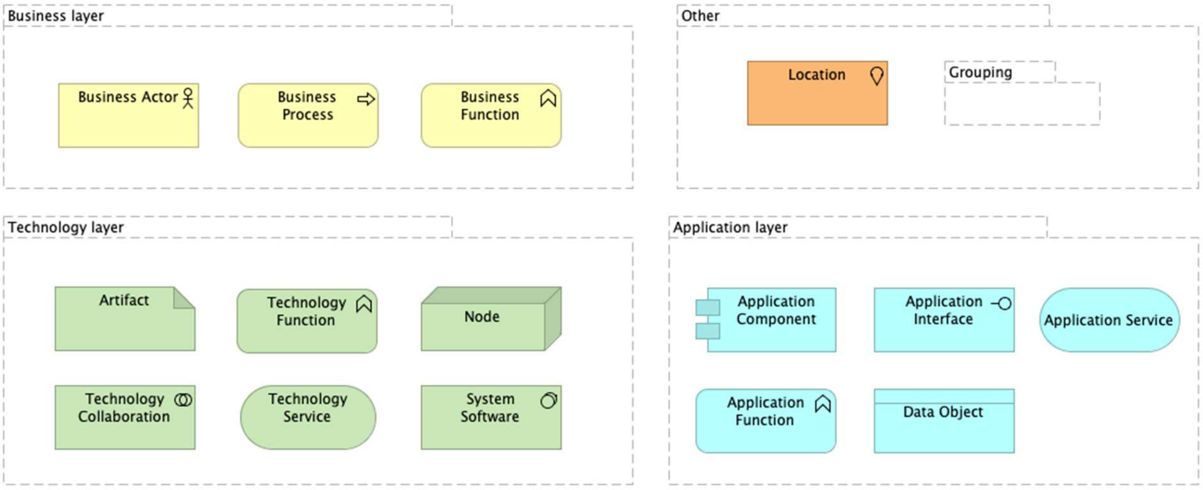


FIGURE 7 - ARCHIMATE® OBJECTS

Funded by the European Union

- **Business Layer:** This layer addresses the core operations and behaviours of the organisation:
  - **Business Actor:** Represents a business entity capable of performing behaviour;
  - **Business Process:** A series of business behaviours realised as a response to a specific trigger;
  - **Business Function:** A functional unit that can be distinguished from others based on its purpose or task.
- **Technology Layer:** Here, we delve into the technical components that support and manage the business operations:
  - **Artefact:** Physical entities that realise application or technology elements;
  - **Technology Interface:** The point of access where technology services are made available to other components;
  - **Node:** A computational or physical resource that hosts, manipulates, or interacts with other computational resources;
  - **Technology Collaboration:** The aggregation of two or more nodes that work together to offer technology services;
  - **Technology Service:** Exposes the functionality of nodes, through interfaces, to its environment;
  - **System Software:** The software environment for specific types of components.
- **Application Layer:** This layer encapsulates the application structure and its interactions:
  - **Application Component:** A modular, deployable, and replaceable part of a software system;
  - **Application Interface:** The point of access where application services are provided to other components;
  - **Application Service:** Exposes the functionality of components, through interfaces, to its environment;
  - **Application Function:** A unit of functionality that offers specific behaviour;
  - **Data Object:** A representation of entities manipulated by application components.
- **Other:**
  - **Location:** Represents a conceptual point or extent in space;
  - **Grouping:** Enables the grouping of any number of specific objects or relations.

By understanding these foundational elements, readers can better comprehend the intricate relationships and dynamics captured in the detailed architectural diagrams of the document.

Funded by the European Union

2.2 TEADAL GLOSSARY

This section introduces the main TEADAL concepts defined throughout the life of the project, in order to make the reader able to fully understand the TEADAL Testbed deployment architecture. They come as results from the ongoing project activities.

2.2.1 "Testbed Site"

A Testbed site refers to a virtual, physical, or hybrid Data Center managed by a single organisational entity, e.g. POLIMI.

The Testbed sites within the TEADAL project are single administrative entities that act as resource providers by delivering the necessary hardware, computational and network resources, and so serving as the fundamental building blocks for the various Pilot cases.

As per the project proposal, there are four main Testbed Sites corresponding to four partners that share resources. They are POLIMI, MARINA, BOX2M, TERRAVIEW.

During the project activities, other partners can share new resources if needed.

2.2.2 "TEADAL Node Baseline"

As discussed in deliverable "*D3.1 Gravity and friction-based data governance*", the TEADAL Node baseline includes three canonical data zones typical of generic data lake architecture, plus the data sharing zone which is a peculiar element of the TEADAL proposal. They have been considered by the TEADAL Architecture at the level of a common background (baseline) to be used as the foundation to build the new TEADAL-enhanced, federated stretched data lakes.

The deliverable D3.1 gives the agreed definition of such zones which are represented, in ArchiMate® syntax, as Data Object blocks as shown in the following diagram:

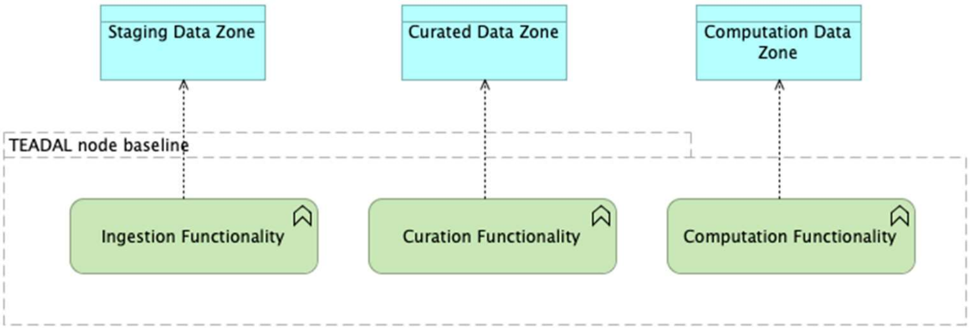


FIGURE 8 - TEADAL NODE BASELINE DATA LAKE

The diagram illustrates the conceptual layout of a baseline TEADAL data lake. It is divided into four primary zones or areas, each with a distinct purpose and function:

- **Ingestion Functionality:** Structured, unstructured, or semi-structured data from external data sources or other data lakes is fed into the data lake through this functionality. The data lake supports either batch data ingestion or continuous data streaming. Data is ingested into a **Staging data zone** that is the entry point for data into the data lake. All incoming data, whether structured or unstructured, is initially directed to this zone. It acts as a staging area where raw data is collected before undergoing any processing or transformation.

- **Curation Functionality:** Once the data has been ingested, it undergoes various curation processes. Data curation ensures that the data is cleaned, enriched, and made ready for analytical processes in a **Curated data zone**. This zone stores data that has been processed to some extent.
- **Computation Functionality:** Computation functionality allows complex analysis and transformations to be performed on one or more data contained in the Curated Data Zone. Data is transformed, analysed, and processed in the **Computation zone**. This zone can be visualised as the 'workbench' of the data lake where the heavy computational tasks, data mining, and analytics processes take place. It's the zone that facilitates data's journey from raw information to actionable insight.

Each zone plays a well-defined role in ensuring that the functional data journey, from ingestion to sharing, is smooth, efficient, and serves the purpose of the whole system. This modular structure also allows for scalability, adaptability, and separation of duty as the data governance requirements grow or change.

2.2.3 “TEADAL Tools”

With **TEADAL Tools** we encompass the full set of TEADAL project results, namely the set of products and services that, by enhancing the TEADAL node baseline, realise the envisioned continuum of stretched and federated data lakes.

Examples of TEADAL tools that are under development include a Data Catalog for FDP, the Control Plane to optimise the data distribution among the locations and organisations, the Policy Management tool to improve the data governance, and the blockchain-based Trust Plane.

The complete taxonomy and definition of the TEADAL tools will be referenced in the final deliverable “*D2.4 Final general architecture*” to be produced at the end of the project. Increasingly refined descriptions of these tools are expected to be made available in the other deliverables of the architecture series (“*D2.2 Pilot cases’ intermediate description and initial architecture of the platform*” and “*D2.3 Pilot cases’ final description and intermediate architecture of the platform*”) according to the progresses of the project activities.

2.2.4 “TEADAL Node”

The complex of **TEADAL Node baseline** and **TEADAL Tools** is defined as **TEADAL Node** (ref. “*D2.2 Pilot cases’ intermediate description and initial architecture of the platform*”).

This section specifies and extends the concept of TEADAL node from the technical architecture point of view.

Funded by the European Union

The TEADAL Node model is inspired by the C4 model for visualising software architecture[1], where each layer offers a higher level of abstraction and encapsulation. It is shown in the following image:

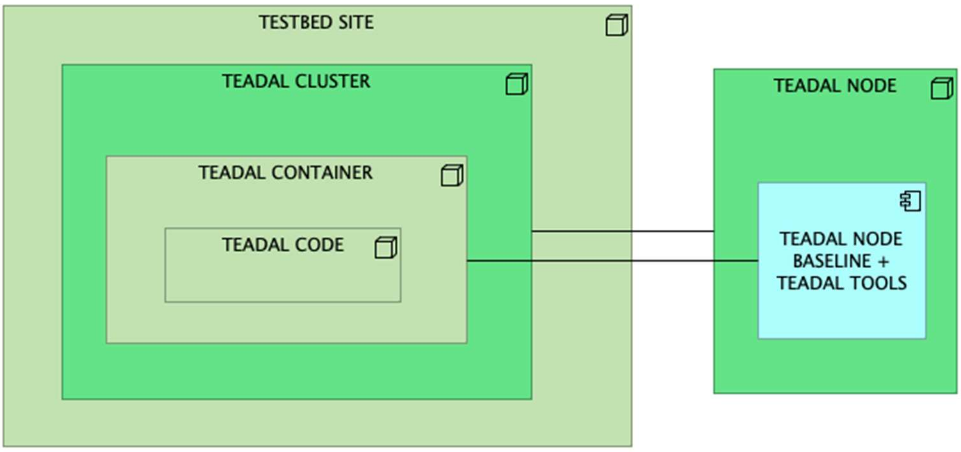


FIGURE 9 - TEADAL NODE LAYERED INFRASTRUCTURE (4C MODEL)

The diagram conceptualises the layered organisation of the TEADAL infrastructure. The outermost layer is the “Testbed Site”, that is the main infrastructure environment in which all TEADAL-related activities and operations are housed.

Within this environment, the next layer of encapsulation is the "TEADAL CLUSTER." This signifies a specific collection of resources and services designated for TEADAL functions. It provides a unified system that ensures resources are used efficiently and services are delivered seamlessly. It is to be remarked that the fact that the TEADAL Node sits at cluster level has consequences and significance in terms of its level of entanglement within the infrastructure. This will be better explained in the following Testbed Drivers subsection.

Inside the cluster, we find the "TEADAL CONTAINER." As the name suggests, it is a standardised environment encapsulating the TEADAL applications and services. Containers are known for their portability and lightweight nature, ensuring that the applications run consistently across different computing environments.

At the core of this hierarchy lies the "TEADAL CODE." This layer represents the raw computational logic and functionalities that drive the TEADAL platform. It's the essence of the TEADAL infrastructure, embodying the primary tasks and operations.

Adjacent to the main structure, the diagram showcases the "TEADAL NODE," which, in this representation, is constituted by the "BASELINE DATA LAKE + TEADAL TOOLS." A TEADAL Node essentially represents a unit or an instance of the TEADAL infrastructure that interacts with the data lake and utilises the tools provided. The fact that it is portrayed at the cluster level emphasises the concept that a node operates at this granularity.

[1] https://c4model.com/

Funded by the European Union

To further clarify the concept of the TEADAL Node, let's consider its practical realisation within the TEADAL Testbed ecosystem. It is worth noticing that in that respect a generic Testbed Site can and will host more than one TEADAL Node, as shown on the following diagram:

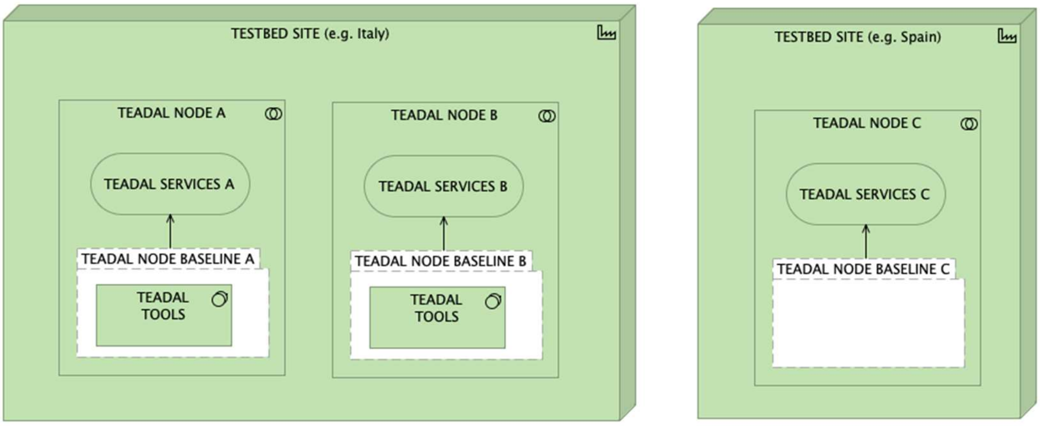


FIGURE 10 - TESTBED SITES

Such a diagram illustrates two distinct Testbed Sites, each one of them hosting one or more TEADAL Nodes. In detail, this diagram shows:

TESTBED SITE 1

- **TEADAL NODE A and B:** This serves as the primary hub for managing, storing, and processing data within Testbed site 1. It is the core of operations for this site;
- **TEADAL SERVICES A and B:** These are the specialised services offered within the node, tailored to handle specific tasks and operations. These services are based on the baseline TEADAL node and the TEADAL tools. An example is the FDP/SFDP services as described in D2.2;
- **TEADAL NODE BASELINE A and B + TOOLS:** The data lakes where all data gets stored, processed, and managed, featured with the set of TEADAL tools, ensures TEADAL-“powered” data operations within the node.

TESTBED SITE 2

- **TEADAL NODE C:** Mirroring the functions of Node A and Node B is the primary hub for Data Center 2, ensuring all data-related operations are centralised;
- **TEADAL SERVICES C:** Like its counterpart in Testbed site 1, these services fulfil specific requirements of this site, providing TEADAL solutions and functionalities;
- **TEADAL NODE BASELINE C:** This repository is designed to meet the storage and processing needs of Testbed site 2. In this example no extra set of TEADAL tools is deployed within the node.

In synthesis, this representation underscores the modularity and adaptability of the TEADAL Node concept. Whether within Testbed Sites 1 or 2, each TEADAL Node works as an autonomous unit, complete with its services and data lake. Yet, it remains an integral part of

Funded by the European Union

the larger data ecosystem, emphasising the node's role as both an individual entity and a part of the broader data infrastructure.

2.2.5 “Pilot Testbed”

The Pilot Testbed is the deployment environment where a generic Pilot is installed and run. As already anticipated, in the TEADAL project each Pilot will be deployed in a dedicated Pilot Testbed that fits its characteristics and needs. It is the main subject of this deliverable, and it will be defined in detail in section 5.

2.3 Testbed Design Approach

During the project activities all the six defined Pilots have been analysed to proceed with their Testbed definition.

The gathered information comes from previous deliverables "*D2.1 Requirements of the Pilot Cases*" and “*D2.2 Pilot cases’ intermediate description and initial architecture of the platform*”, various meetings with Pilot partners and Testbed site partners. In such meetings internal documents have been produced, shared and compiled in order to have a complete collection of the needed data to proceed.

Available resources have been gathered through an internal document that has been filled by resource providers: POLIMI, BOX2M, TERRAVIEW and MARINA. The available resources table is included in section 6.1. During the life of the project, as previously anticipated, other partners are allowed to share resources if needed.

Hardware resource needs for each Pilot has been estimated, considering hardware requirements for a single TEADAL Node, the Pilot topology and storage requirements for each data lake in the Pilot topology.

Hardware requirements for a single TEADAL Node have been formalised during project activities. The proposed sizing is defined referring to the estimated needs of the Pilots looking at the expected final project configuration.

The Pilot topology has been derived from D.2.1 and D2.2. Then a proposal of Pilot Testbed topology has been shared with Pilot owners though an internal document to gather their observations and technical details.

Storage requirements for the total amount of data that will be loaded on each data lake while running the Pilot have been estimated by Pilot owners. The Pilot storage requirement has been added to the TEADAL Node storage requirement for each VM in the Pilot topology. The VM list associated with each Pilot is included in section 5.

Partner available resources have been assigned to Pilots, identifying the candidate Testbed sites for hosting the VMs. The site-pilot matrix is included in section 6.2.

Funded by the European Union

3 Central TEADAL Services

In this section, Testbed Resources for TEADAL central services are described.

By TEADAL central services we mean those services that are used in a shared way by all Pilots. They also include components that are not actually part of the TEADAL software distribution but are nevertheless functional to the realisation and operations of a working Pilot Testbed.

Among the central services we can distinguish at least two categories:

1) Services that can be considered part of the stable Testbed infrastructure. These services include the solution supporting the CI/CD process and could e.g., include a web portal giving access to the TEADAL distribution (not in scope at the time of this writing). These components, as said, can be regarded as an infrastructural value brought to the project but they are not, by themselves, objects of the research;
2) Services that are objects of research and development and, although initially deployed in a centralised fashion, are evolving toward distributed/federated deployment models. These categories include e.g., the TEADAL Data Catalog, deployed as a central service in the initial deployment and validation phases, but moved to be part of the TEADAL distribution and so integrated in the TEADAL node later.

Two phases of the TEADAL Testbed infrastructure evolution are described in the pictures below.

The first shows the initial TEADAL Testbed configuration, represented as a composition of TEADAL nodes and central services. In this configuration both the Data Catalog and the CI/CD systems are centralised services that support all the Pilot Testbeds, grouped on the left.

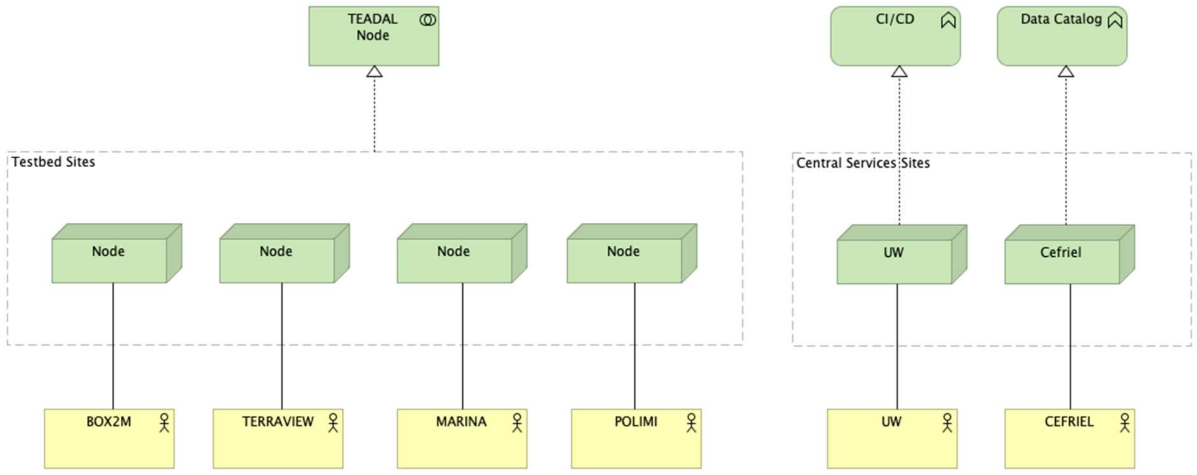


FIGURE 11 - THE INITIAL TEADAL TESTBED CONFIGURATION

In the final project configuration, as illustrated by the picture below, the “Federated” Data Catalog will have evolved into a distributed service and therefore “absorbed” inside the TEADAL tools, becoming part of all the TEADAL Nodes (left).

Funded by the European Union

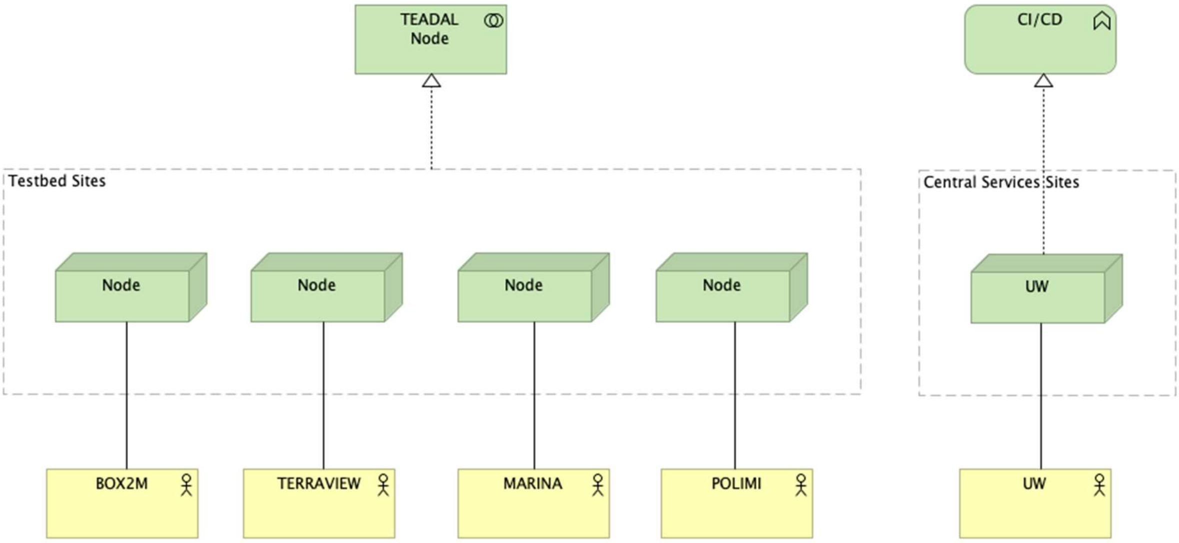


FIGURE 12 - THE FINAL TEADAL TESTBED CONFIGURATION

3.1 CI/CD - Resources dedicated to CI/CD Central Services

This section shall define the CI/CD Process used in the TEADAL project. It will also describe which resources are available and where they are deployed.

3.1.1 The CI/CD Process definition in TEADAL

The CI/CD Process is the set of tools and methodologies employed that aim to guarantee the integrity of the TEADAL code base, as well as its continuous deployment to a given target. The rest of this section will describe how this is achieved.

Continuous deployment in TEADAL is assured by using GitOps. In this context, GitOps means that a Git repository is used as the single source of truth for the status of the infrastructure. This repository contains configuration files that describe the TEADAL architecture in a format appropriate to be parsed by the CI/CD tools, an approach known as Infrastructure as Code (IaC). In this way, there is a guarantee that each deployment is in an easily auditable, known state. Additionally, this approach has the advantage that different versions of the infrastructure are kept in the version control system, allowing the developer to roll-back to previous versions in case the current one is failing.

Whenever a developer merges an update on a repository monitored by the GitOps deployment tool, it triggers a chain of processes with the objective of updating existing TEADAL deployments in production.

Funded by the European Union

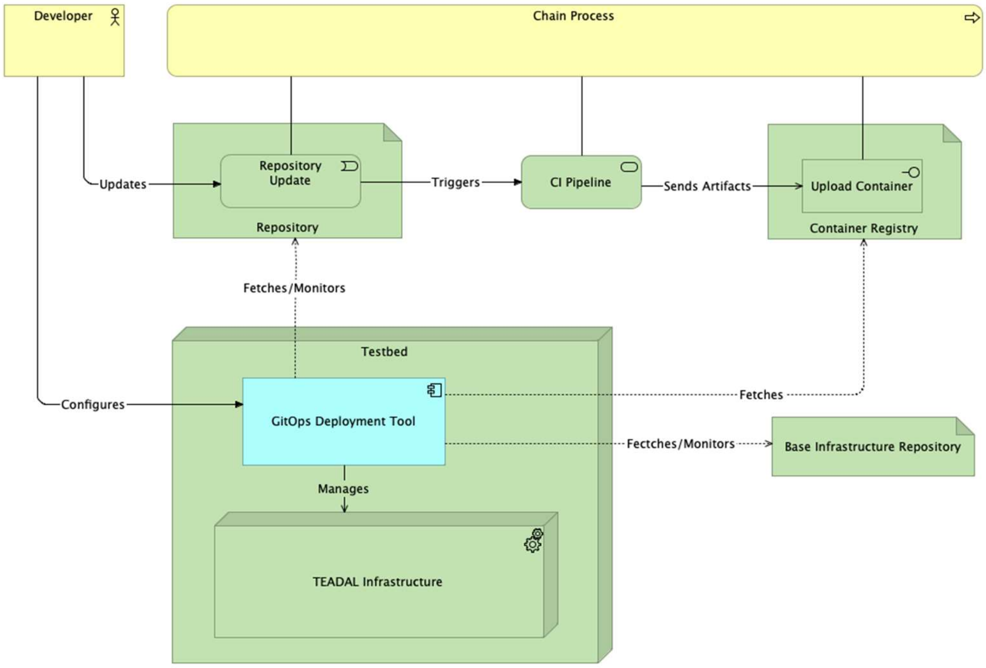


FIGURE 13 - CI/CD WORKFLOW REPRESENTATION

A representation of the proposed CI/CD workflow is given on figure 13. As it stands, a GitOps Deployment tool needs to be installed in each Testbed. This tool needs to point, at least, to the repository that contains the descriptions of the base TEADAL node infrastructure. This repository shall be hosted on an accessible version control, Git-based, platform.

Applications that need to be deployed on a TEADAL node need to be containerised. The containerisation process can be automated using a CI pipeline. These describe a certain number of steps that are executed whenever a new version of the code is available. After being containerised, this application needs to be sent to a centrally available container registry, so that the GitOps Deployment tool can fetch it.

For more information about the architecture of a TEADAL node and its software stack, see "*D2.2 Pilot cases' intermediate description and initial architecture of the platform*".

3.1.2 CI/CD Testbed resources

This section describes what CI/CD resources are available and where they are available. Due to our GitOps approach, as referenced in section 3.1.1, most of the tools are centralised on the repository manager and DevOps platform, GitLab.

Both the repositories containing the baseline configuration of the Data Lake and the base TEADAL tools are hosted on this platform, making it the single source of truth for this software.

Among its set of DevOps features, Gitlab includes both a container registry, GitLab Registry, and a pipeline engine, GitLab CI. Consequently, the CI part of the TEADAL CI/CD process can be centralised on just a single platform.

This platform is hosted by UW[2], and available to all partners. The GitOps Deployment tool used in TEADAL is ArgoCD. Unlike GitLab, this software is not centrally deployed. Instead, each TEADAL node will host its own instance of ArgoCD, configured according to its needs.

3.2 Data Catalog

As also described in "*D3.1 Gravity and friction-based data governance*" and "*D4.1 Stretched data lakes first release report*", the Data Catalog is a service devoted to archiving structured descriptions (metadata) of various kinds of "digital assets". In particular, it allows describing Datasets and Federated Data Products, the two central objects involved in the gravity and friction model promoted by TEADAL.

The functionalities of the Catalog can be accessed both via API and via a dedicated UI. Using the UI, users of a data lake can:

- browse through the available assets,
- search Datasets and Data Products,
- engage in the governance of the lifecycle of the aforementioned assets, according to the lifecycle processes specified via BPMN.

The first release of the Catalog is tailored to be used as a centralised and shared component among the participants of a federation. As such, users, groups, and lifecycle processes are centralised, and there exists only one instance of the Catalog for all the members of the federation. This setup is partly motivated by the desire of providing a ready-to-use instance for the benefit of all the project partners, and also by the experience accumulated by Cefriel in managing collaboration ecosystems[3]. An effective collaboration (in a sense, a federation) can be obtained by having a trusted entity run a shared catalog. Such a catalog is then operated according to a clear governance model, and such model also covers all the steps required to ask for the permission to access an API for a specific purpose.

The initial configuration of a central TEADAL Catalog therefore allows obtaining a federation via a centralised component shared between all the parties and run by a third-party company.

In the next iterations we will instead focus on a fully peer-to-peer federation, where each company will run its own data lake comprising the Data Catalog, and these catalogs will be made aware of one another. As a result, such "Federated" Catalogs will allow seamless searching through the assets in the whole federation, and a special emphasis will be put on catalog-to-catalog interactions for the processes by which a specific catalog user will obtain the permission to use an asset hosted on a different data lake.

[2] https://gitlab.teadal.ubiwhere.com

[3] https://www.e015.regione.lombardia.it

Funded by the European Union

4 ELEMENTS OF TEADAL NODE ARCHITECTURE

This section provides the reader with a quick overview of the TEADAL software architecture mainly with the purpose of better understanding what the TEADAL installation will look like once deployed on a Testbed site. We believe that a more in-depth understanding of the architecture structure and behaviour of the TEADAL solution may help Testbed sites administrators to figure out, based also on their personal experience, which drivers and constraints for their sites will come from the deployment of the TEADAL node on their site.

4.1 TAXONOMY OF TEADAL TOOLS

As previously introduced in the section 2.2.3, the complete taxonomy and definition of the TEADAL tools will be referenced in the final deliverable "*D2.4 Final general architecture*" to be produced at the end of the project. We summarise here the main concepts established at the current time of writing.

In brief, TEADAL architecture builds upon the data mesh concept to enable data sharing between different organisations. Hence, the architecture is envisioned to manage the life cycle of the main application service featured by the TEADAL architecture, the Federated Data Product (FDP).

The FDP is an application service for sharing or distributing processed data, enabling collaboration or data export to other systems. It is produced out of data available in the staged dataset, it has got policies attached and is accessible through a REST API. The FDP is registered in the federation Data Catalog for discoverability.

The object that is actually accessed by the Federation consumer is the Shared Federated Data Product (SFDP). The SFDP is a particular instantiation of an offered FDP also bundling the specificities of the agreement between the owner and the consumer.

The picture 14 shows in detail how a single TEADAL node deployed on a Testbed site - in the example a site distributed over two physical locations - provides all the technical functionalities (data ingestion, curation, computation, and sharing) apt to create and share the FDP and the SFDP.

Funded by the European Union

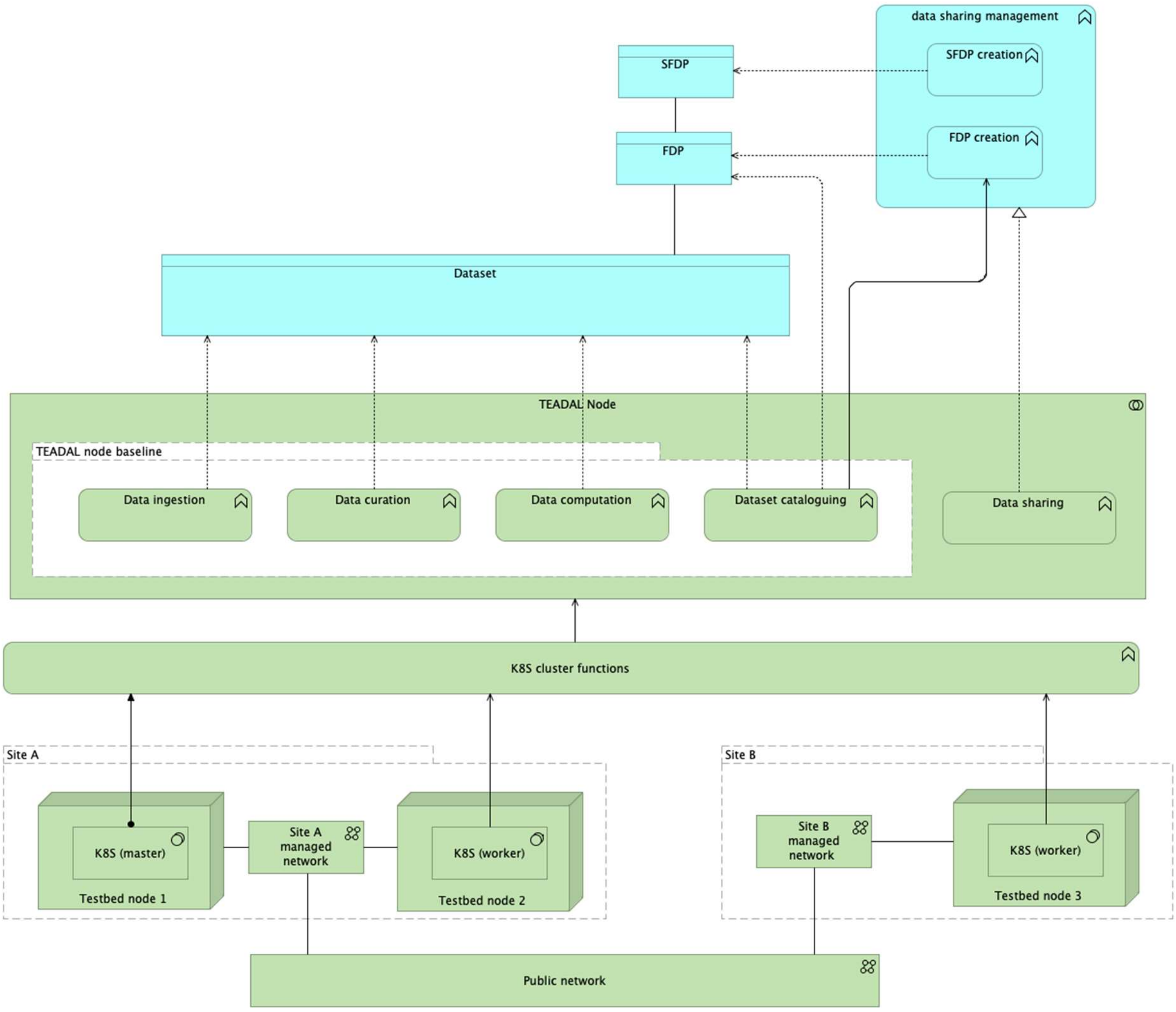


FIGURE 14 - EXAMPLE: SINGLE TEADAL NODE ON A DISTRIBUTED TESTBED SITE

The technical functionalities described above are realised through an integrated solution based on a set of system software components included in the TEADAL node distribution. The current list of system software packages is reported for reference in the table below as it is extracted from the deliverable "*D2.2 Pilot cases' intermediate description and initial architecture of the platform*" section 9.5.3. The list will most likely be subject to future variations throughout the continuation of the project and it will be kept up to date in the relevant architecture deliverables "*D2.3 Pilot cases' final description and intermediate architecture of the platform*" and D2.4, to be always considered as the final authoritative reference for that matter.

Name	Category	Functionality
ArgoCD	DevOps	GitOps IaC tool for Kubernetes clusters
Istio	Networking	Service mesh network
KeyCloak	Security	Access/authentication manager

Funded by the European Union

OPA	Security	Policy enforcement tool
Reloader	Security	Check changes in config files
MinIO	Storage	Object storage
PostgreSQL	Storage	SQL database
Grafana	Monitoring	Platform for data visualisation
Prometheus	Monitoring	Tool for collection and storage of computing metrics
Kiali	Monitoring	Istio console to monitor and control the service mesh
Jaeger	Monitoring	Tracing tool to map data flows and requests
Airflow	DAG	Management platform to define pipelines and workflows on data
Kubeflow	DAG	Management platform to define pipelines and workflows on data related to machine learning

TABLE 1 - TEADAL NODE DISTRIBUTION

In addition to the project documentation, the de-facto definition of the TEADAL node distribution is available in the project git repository[4].

4.2 Example use-case of TEADAL tools: Creation of a SFDP

As a complement to the static views provided above, in the figure below we show an example of dynamic interaction of the TEADAL node application architecture components, giving a representation of how the application elements collaborate for the **use case of creation of an SFDP** (Shared Federated Data Product).

[4] https://gitlab.teadal.ubiwhere.com/teadal-tech/teadal.node

Funded by the European Union

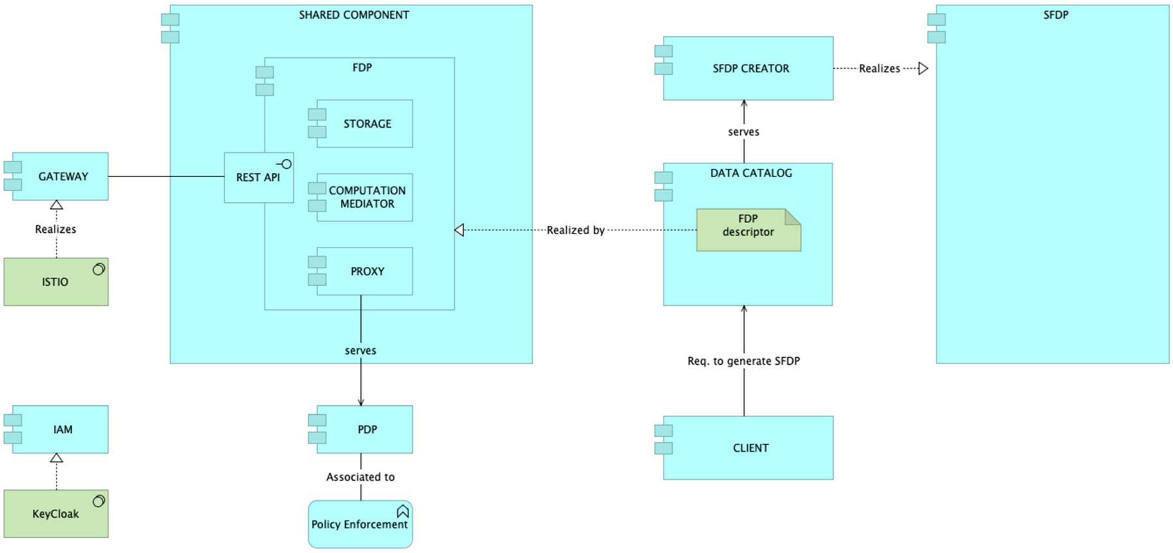


FIGURE 15 - SFDP CREATION PROCESS

This view describes the process that starts from a client request to create an SFDP, through the retrieval of a descriptor from the Data Catalog to the actual creation of the SFDP, all while ensuring security, policy adherence, and efficient data flow.

The view is useful also to understand the nature of inter-sites communication interfaces that need to be enabled in order for the required communication interactions to work (e.g., between client application and Catalog or gateways).

The diagram includes three primary elements:

- The **Shared Component** bundling core services related to data storage, computation, and interfacing;
- The **SFDP Creation Process**, which involves the SFDP Creator interfacing with the Data Catalog and the eventual realisation of the SFDP;
- The **Foundational services** that include the Gateway, IAM, and PDP which provide foundational capabilities such as identity and policy enforcement.

1. **Sharing Component:**
    - The **REST API** serves as the central interfacing component;
    - Storage is where all data pertinent to this process is persisted;
    - **Computation Mediator** is responsible for orchestrating the computation tasks;
    - The **Proxy** acts as an intermediary that serves requests, directs the flow of data, and ensures secure access.

2. **SFDP Creation Process:**

    The process starts with a **Client** issuing a request to generate an SFDP.

    - This client interacts with the Data Catalog to find an appropriate FDP Descriptor;

Funded by the European Union

- Following this, the SFDP Creator kicks into action. It takes the chosen FDP Descriptor, processes it, and then realises (or creates) an SFDP.

3. **Foundational Services:**

   - **Gateway** serves as the primary access point, realised by **Istio**, an open-source service mesh;
   - **IAM (Identity and Access Management)** is facilitated through **KeyCloak**, ensuring proper authentication and authorization;
   - **PDP (Policy Decision Point)** is working in accordance with the Policy Enforcement component, ensuring that data shared in the federated environment adheres to agreed-upon policies.

4.3 Testbed Drivers

The TEADAL consortium put a lot of attention into the definition of tools that are readily available and easily deployable on the Testbed sites. One of the main approaches adopted to simplify the scenario and to make the deployment procedures as independent as possible from the infrastructure choices is to leverage on well-established CI/CD best practices such as containerisation and container orchestration. The info provided at sections 3.1 and 4.1 of this document well demonstrate this point, showing that the TEADAL node can be abstracted as a logical unique element. It can though be realised through very different configurations of the underlying Testbed site, e.g., by physical nodes in a data centre, virtual machines from a cloud provider, or even by a distributed cluster of physically remote sites. Please refer to the example provided in subsection 4.1. Such flexibility translates in the requirement to run in a containerised, orchestrated environment.

On the other end, the design and engineering done in TEADAL has been focused towards making the tools future-proof. So, by design, sufficient modularity is sought to allow independent scalability as well as to enable separation of duty in the administration of the different services. For this reason, as an example, the design choice has been adopted to define separate namespaces for different TEADAL components. This choice translates into an important deployment constraint for the site to have a dedicated Kubernetes cluster for a TEADAL node (please see section 3.1).

Testbed sites (as well as sites possibly joining in the near future) will have to assure enough operational flexibility to work at the level of virtual machines and so be able to instantiate and manage multiple local Kubernetes clusters, especially if they aim to supply more than one Pilot Testbed. Once this is granted, the inherent horizontal scalability allowed by a containerised solution should assure adequate responsiveness to possibly changing requirements in terms of computing and storage resources.

Funded by the European Union

5 TEADAL Pilots Testbeds

TEADAL is featuring six Pilot use cases to demonstrate the outcomes and benefits in terms of decreased characteristics of gravity and friction in sharing data across data lakes. The Pilot use cases, focusing on the different sectors of Healthcare, Mobility, Agriculture, Industry, Finance, and Energy, are selected and designed in TEADAL.

The reference for the complete Pilot Testbeds description is mainly the series of deliverables "D2.1 *Requirements of the pilot cases*" and "*D2.2 Pilot cases' intermediate description and initial architecture of the platform*". The information hereby provided is derived from the sampling of said document along with interviews and meetings with pilot partners to gather specific information needed to design their testbed. The deliverables above should be always considered as the authoritative source of truth with respect to Pilot use cases and architecture topology.

For the demonstration of each one of the six use cases, an integrated solution is set-up based on the appropriate use of TEADAL features (TEADAL Baseline Node and TEADAL Tools) within the domain-specific applications.

As anticipated in section 2.2.5, the implemented, integrated, and deployed solution realising the specific Pilot use cases in one of the domains is defined as **Pilot Testbed**.

This section provides the design of the six Pilot Testbeds. First, generally applicable concepts such as the supported deployment scenarios and the TEADAL node hardware requirements are outlined. Then, the section breaks down into six subsections where the specifics of each pilot, in terms of *pilot topology* (high-level description of the specific integrated solution for the pilot case), *deployment architecture*, *technical information*, and *connectivity matrix* are provided.

The deployment architecture chapter identifies the Pilot significant entities such as the

- **Pilot Owner**: accountable subject for the integration, operations, as well as execution of the pilot cases, managing the work of all involved partners;
- **Pilot Case Entities**: the key entities constituting the pilot, each one corresponding to one TEADAL Node;
- **Testbed site**: the designation of the "physical" site or sites where the TEADAL nodes are deployed;
- **Resource Provider Owner**: responsible for overseeing and providing the necessary infrastructure resources for the Pilot testbed(s).

and their relations.

The technical information chapter reports technical specificities of the Pilot Testbed.

The connectivity matrix provides a tabular description of the communication interfaces and relevant protocols that need to be enabled among the nodes.

Funded by the European Union

Pilot Testbed Deployment scenarios

A generic Pilot Testbed is composed of one or more TEADAL Nodes, leveraging cloud and, eventually, edge resources. Cloud resources are provided on a private or public cloud environment. A Virtual Machine (e.g., a local VM or an AWS EC2) is assigned to each TEADAL Node. This virtual node represents an amount of physical resources available for a single TEADAL Node.

Each Pilot chooses whether to have its virtual nodes deployed in the same Testbed Site or to split them over more than one site, to experiment some network complexity. In the following picture a generic example of general pilot topology is represented.

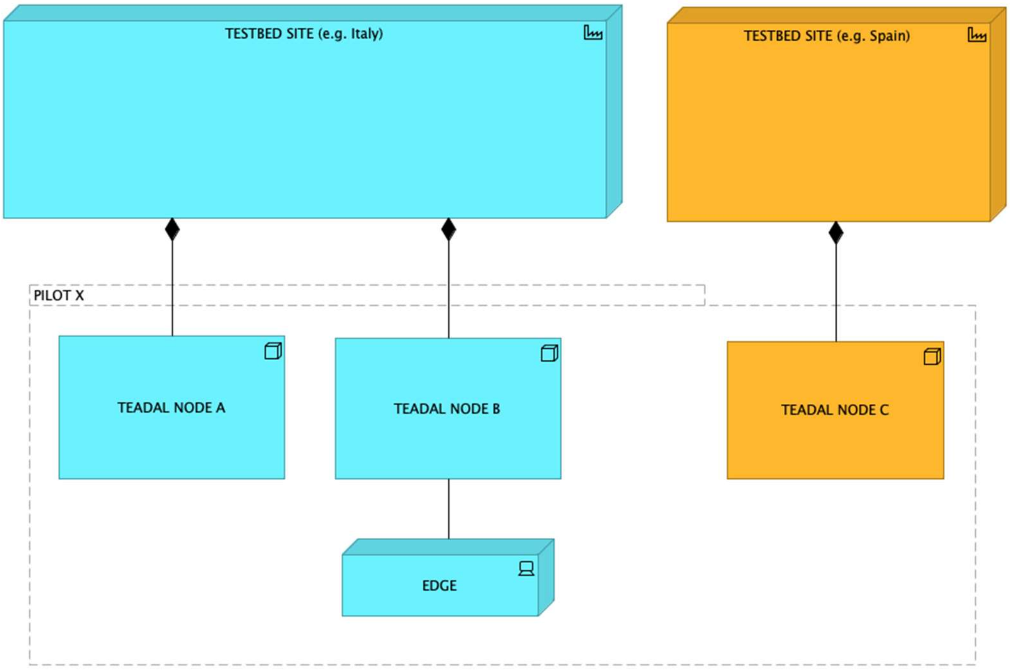


FIGURE 16 - EXAMPLE: PILOT TESTBED SPANNING OVER TWO TESTBED SITES

Figure 16 is an example showing the pilot topology of the PILOT X, based on two TEADAL Nodes (A, B) hosted in one Testbed site (Italy), and one TEADAL node (C) hosted in a different Testbed Site (Spain). Furthermore, the pilot X relies on an EDGE device extending the continuum of TEADAL Node B.

TEADAL Node Hardware Requirements

The virtual node, as introduced in the previous section, is the building block for the definition of the Pilot Testbeds.

At deployment architecture level, it is composed of a Virtual Machine, or a Virtual Service, having the following specifications, as formalised during the project activities and expressed in TEADAL project software repository[5].

[5] https://gitlab.teadal.ubiwhere.com/teadal-tech/teadal.node/-/blob/main/docs/bootstrap/dev-cluster-base.md?ref%5C_type=heads#hardware

Funded by the European Union

Component	Minimum requirement
Virtual CPU	8 at 2,3 GHz
RAM	32 GB
Storage	100 GB
OS	Linux

TABLE 2 - VM SPECIFICATION

The requirements stated in the above table includes resources needed for both the TEADAL Node Baseline and the TEADAL Tools to be installed and run. The amount of storage needed to host the TEADAL Node must be integrated with the storage needed for the pilot data to be hosted on each specific node. The above requirements are the ones that, at this stage of the project, are deemed to be sufficient to support the final implementation of the project results. However, given the flexibility level of the defined Testbed resources, in case of necessity, it will be possible to scale up or down such requirements.

5.1 Pilot #1 - Evidence-Based Medicine

The Evidence-Based Medicine Pilot aims to improve the current status of data analytics in healthcare, easing the process of sharing medical data. The pilot has a focus on data privacy constraints that are the main handicap in healthcare data sharing between data providers, such as hospitals, and researchers. In this section the dedicated Pilot Testbed is described.

5.1.1 Pilot Topology

This section presents the designed solution for the Evidence-Based Medicine Pilot Testbed. A full description of the Pilot objectives and planned topology can be found in deliverable D2.1, under the “USE CASE PILOT #1: EVIDENCE-BASED MEDICINE” chapter.

The described topology is the one foreseen for the final Pilot Testbed configuration, to provide the most complete indication of all the needed resources.

It is composed by three nodes:

- two TEADAL Nodes, each one for an Hospital (multiple data Providers);
- one TEADAL Node for the Researcher (data Consumer).

For the first project iteration just one node will be set up as a data provider node with the TEADAL Node baseline, while the data consumer will act from a client (browser/service query client) for querying the Data Catalog and accessing the data.

Funded by the European Union

5.1.2 Testbed Deployment Architecture

The deployment architecture for the final set-up of the Evidence-Based Medicine pilot is illustrated in the following diagram.

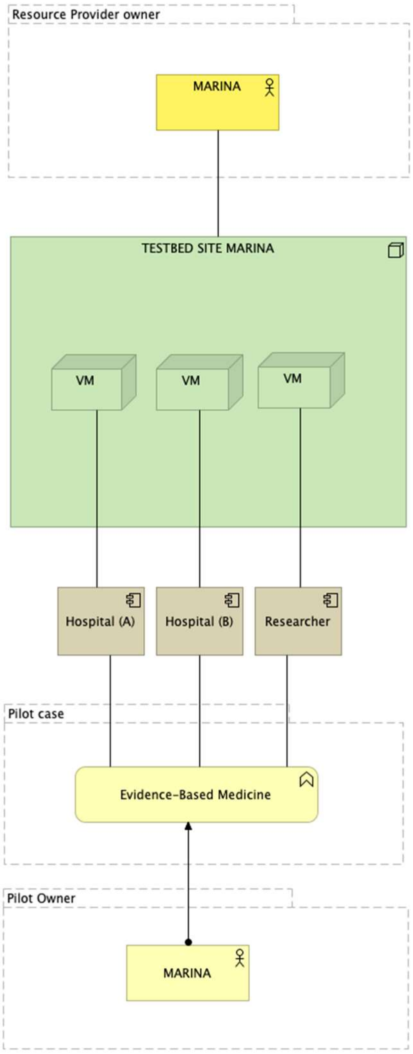


FIGURE 17 - PILOT #1 - TEADAL DEPLOYMENT ARCHITECTURE

Here's a breakdown of the depicted components and their relationships:

1. **Pilot Case:** The business scenario representing the 'Evidence-Based Medicine' Pilot as described above;
2. **Pilot Owner:** 'MARINA' is the owner of this specific pilot, ensuring its alignment with the overarching objectives and providing direction. MARINA is ultimately accountable for the integration, operations, as well as execution of the pilot cases, managing the work of all involved partners;
3. **Pilot Case Entities:** Relying on the cloud infrastructure, there are three key entities directly involved in the pilot, each one deploying one TEADAL Node:

Funded by the European Union

   a. **Hospital (A) & Hospital (B):** These are the two primary healthcare institutions participating in the pilot acting as data producer;
   b. **Researcher:** An individual or team that accesses and analyses the data to derive meaningful insights, contributing to the medical research component of the pilot, acting as data consumer/s.

4. **TESTBED SITE MARINA:** This is the designated site where the Testbed for the pilot is located. Within this site, the core infrastructure is hosted on the AWS cloud, the public cloud platform provided by Amazon Web Services;
5. **Resource Provider Owner:** Besides being the pilot owner MARINA acts also as Resource Provider Owner'. As such the owner is responsible for overseeing and providing the necessary resources for the Testbed.

5.1.3 Technical Notes

All the TEADAL Nodes are hosted in MARINA site and are implemented by AWS cloud virtual machines resources. Their sizing is the same as specified in the introduction of this section. Additional memory can be added at each node on the basis of the specific Pilot needs. In the current Pilot, the expected amount of storage for each Data Lake is 5 GB.

As the pilot is based on AWS service, MARINA will be allowed to choose - among the predefined instance types offered by AWS - the more appropriate sizing for the progressing load, to be possibly dynamically upgraded if more resources will be needed, in order to comply with budget and environmental opportunity.

5.1.4 Connectivity Map

Connectivity requirements for the Pilot Testbed are specified in Table 3:

Producer	Consumer	Interface Protocol	Notes
Hospital A	Researcher	https	
Hospital B	Researcher	https	
CI/CD (Argo CD)	Hospital A	https	
CI/CD (Argo CD)	Hospital B	https	
CI/CD (Argo CD)	Researcher	https	
Hospital A	Data Catalog	https	only in the initial configuration
Hospital B	Data Catalog	https	only in the initial configuration

Funded by the European Union

Data Catalog	Researcher	https	only in the initial configuration

TABLE 3 - PILOT #1 - CONNECTIVITY MAP

5.2 Pilot #2 - Mobility Federated Access Point

The Mobility Federated Access Point pilot is set to experiment data sharing, finalised to the creation of a National Access Point (NAP) of mobility data. Since regional data collection initiatives in urban areas are limited due to disparate cross-border cooperation, Italy has delegated transport data collection to regions, creating a three-level system, where a Regional Access Point (RAP) collects data from transport operators and infrastructure managers and makes it available to the National Access Point. In this section the dedicated Pilot Testbed is described.

5.2.1 Pilot Topology

This section presents the designed solution for the Mobility Federated Access Point Pilot Testbed. A full description of the Pilot objectives and planned topology can be found in deliverable D2.1, under the “USE CASE PILOT #2: MOBILITY” chapter.

The described topology is the one foreseen for the final Pilot Testbed configuration, to provide the most complete indication of all the needed resources.

It is composed by three nodes:

- one node for the Local Transport (data provider and data consumer), AMT. The data lake is periodically fed with transportation data; consumes data from NAP node;
- one node for the RAP (aggregator). The data lake is periodically fed with new data coming from the data provider. The node acts also as data provider for the NAP node;
- one node for the NAP (aggregator). The data lake is periodically fed with new data coming from the RAP node. The node also acts as a data provider for the Local Transport Company node.

For the first project iteration just one node will be set-up as a data provider node (AMT) with the TEADAL Node baseline, while data consumers will act from a client (browser/service query client) for querying the Data Catalog and accessing the data.

5.2.2 Testbed Deployment Architecture

The deployment architecture for the Mobility Federated Access Point pilot is visually outlined in the following diagram.

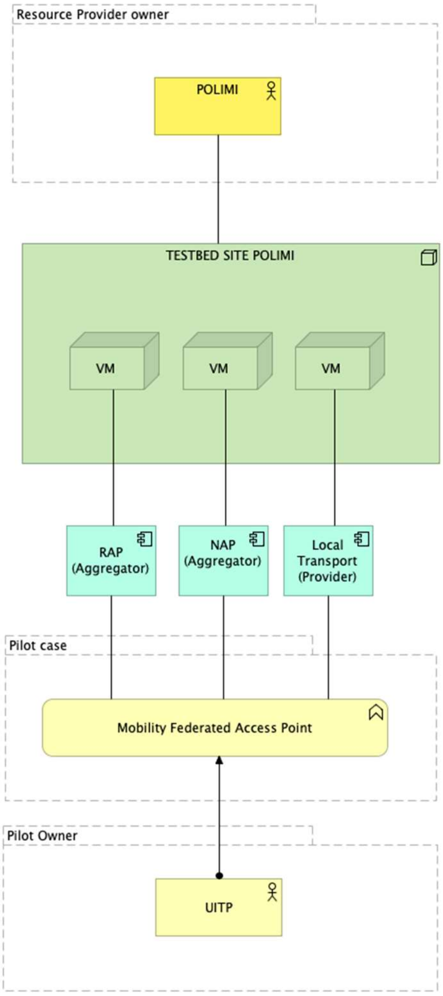


FIGURE 18 - PILOT #2 - TEADAL DEPLOYMENT ARCHITECTURE

Here's a breakdown of its components and their interactions:

1. **Pilot Case**: The business scenario representing the 'Mobility Federated Access Point' Pilot as described above;
2. **Pilot Owner**: The overall responsibility to guide, integrate and manage the pilot belongs to 'UITP'. As the owner of the pilot, they ensure the initiative remains aligned with the consortium's overarching goals and objectives. They are ultimately accountable for the integration of the pilot Testbed and the execution of the Pilot use case;
3. **Pilot Case Entities**: three are the primary entities of the pilot that will be featuring a TEADAL Node:
   a. Local Transport (Data Provider/Data Consumer): This is the TEADAL Node corresponding to AMT that acts as data producer and as data consumer;
   b. RAP (Aggregator): this node collects data from local transport data providers at regional level and makes it available for NAP at national level;
   c. NAP (Aggregator): this node collects data from regional RAP and makes it available for users.

Funded by the European Union

The TEADAL nodes are used to store and manage large amounts of mobility data, offering flexibility and scalability for the pilot needs.

4. **TESTBED SITE POLIMI:** This is the designated location for the pilot's Testbed. Within this domain, the three distinct TEADAL Nodes are deployed on separate VMs (Virtual Machines);
5. **Resource Provider Owner**: The final responsibility of providing infrastructural resources stands with the 'Resource Provider Owner', 'POLIMI'. The Resource Provider Owner is the "supplier" of the Pilot Owner holding the final responsibility of procuring the necessary resources and infrastructure.

In synthesis, this diagram provides a holistic view of the interactions and collaborations among different stakeholders in the Mobility Federated Access Point pilot, all supported by the infrastructure put forth by 'POLIMI'.

5.2.3 Technical Notes

All the TEADAL Nodes come from resources shared by POLIMI Testbed Site. There is no particular requirement regarding storage capacity, the one provided with the allocated VM is more than suitable for the pilot needs. In the current Pilot, the expected amount of storage for each Data Lake is less than 1 GB, not considering historical records. If the need arises to keep historical records the storage will need to be scaled up accordingly, depending on the required want to be kept.

5.2.4 Connectivity Map

Connectivity requirements for the Pilot Testbed are specified in the table below:

Producer	Consumer	Interface Protocol	Notes
Local Transport	RAP	https	
RAP	NAP	https	
NAP	Local Transport	https	
CI/CD (Argo CD)	Local Transport	https	
CI/CD (Argo CD)	RAP	https	
CI/CD (Argo CD)	NAP	https	
Local Transport	Data Catalog	https	only in the initial configuration
RAP	Data Catalog	https	only in the initial configuration

Funded by the European Union

NAP	Data Catalog	https	only in the initial configuration

TABLE 4 - PILOT #2 - CONNECTIVITY MAP

5.3 Pilot #3 - Smart Viticulture Data Sharing

The Smart Viticulture Data Sharing pilot aims to enable data sharing between different Vineyards, especially those placed next to each other, with the goal of quickly monitoring the changes that can have an impact in decision-making processes and in giving them adequate and timely warnings to deal with adverse conditions like weather-related challenges. Each vineyard is equipped with a Terraview Crate, a sort of infrastructure in a box, where the hardware is local and offloaded from the “cloud”. A crate represents a decentralised Edge deployment of TerraviewOS, the SaaS platform managed by TERRAVIEW, that offers vineyard operators a system for better managing their assets. In this section the dedicated Pilot Testbed is described.

5.3.1 Pilot Topology

This paragraph describes the solution for the Smart Viticulture Data Sharing Pilot Testbed. A full description of the Pilot objectives and planned topology can be found in deliverable D2.1, under the “USE CASE PILOT #3: SMART VITICULTURE” chapter.

The described topology is the one foreseen for the final Pilot Testbed configuration, the one at the end of the project, to provide the most complete indication of all the needed resources.

The pilot topology is composed by three nodes:

- 2 Vineyards (data providers): the nodes are fed by data streamed from sensors on the field;
- 1 TVCore (data consumer): central TERRAVIEW component for collecting and analysing the data from the vineyards.

For the first project iteration only two nodes will be set-up, one Vineyard acting as data provider and the TVCore node as a data consumer.

5.3.2 Testbed Deployment Architecture

Outlined in the provided diagram is the deployment architecture for the Smart Viticulture Data Sharing pilot.

Funded by the European Union

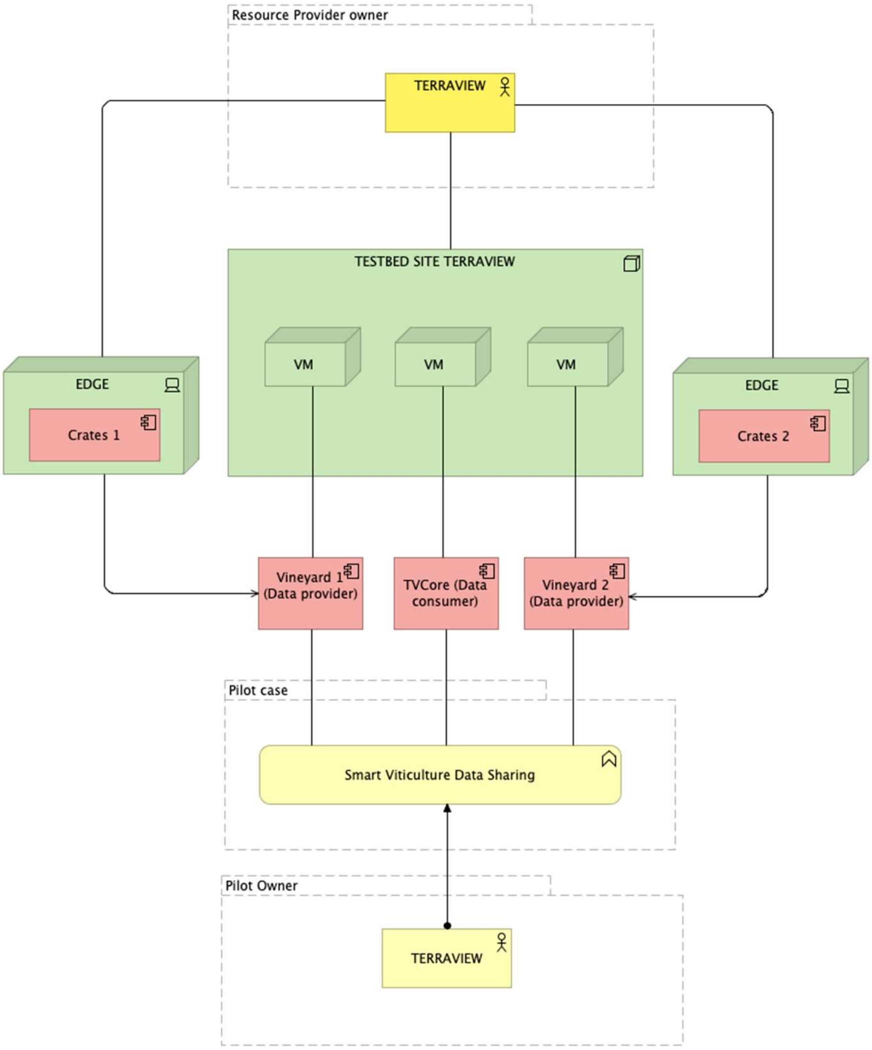


FIGURE 19 - PILOT #3 - TEADAL DEPLOYMENT ARCHITECTURE

Here's a comprehensive breakdown of its elements and their interactions:

1. **Pilot Case:** The business scenario representing the ‘Smart Viticulture Data Sharing’ Pilot as described above;
2. **Pilot Owner:** The accountability of the entire pilot sits with 'TERRAVIEW'. As the principal owner, they make sure the project progresses in line with its intended goals and objectives. It is ultimately accountable for the integration of the pilot Testbed and the execution of the pilot use case;

3. **Pilot Case Entities:** the pilot Testbed consists of three major entities, corresponding to TEADAL nodes:
    a. **TVCore (Data Consumer):** It is the node that utilises the data for various purposes, such as analytics, visualisation, or other relevant tasks;
    b. **Vineyard (two Data Providers):** Representing the primary source of data for the project. Through the TEADAL tools the vineyard ensures that accurate and relevant data is available to the TVCore system.

Funded by the European Union

4. **EDGE:** An important technical entity of the pilot Testbed is the 'Crate', which in TEADAL terminology represents an EDGE component. The EDGE extends the reach of the Vineyard data lake bringing it closer to the location where it is needed;
5. **TESTBED SITE TERRAVIEW:** This box represents the dedicated location for the pilot's Testbed. The site hosts the 'VM' (Virtual Machine) providing computing processes, storing resources, ensuring the required performances;
6. **Resource Provider Owner:** the 'Resource Provider Owner', TERRAVIEW in this case. In the context of the Pilot Testbed, it can be considered as the "supplier" of the Pilot Owner holding the accountability for procurement and operations of the necessary hardware and software resources needed for the pilot Testbed integration and for the execution of the use case.

In synthesis, this diagram offers a perspective on the collaborations and interactions among different stakeholders in the TERRAVIEW pilot, emphasising the support provided by 'TERRAVIEW' as a Testbed site and Edge provider.

5.3.3 Technical Notes

The resource requirements for the baseline have been defined as three VMs hosted at TERRAVIEW Testbed Site with the following sizing:

- 16 VCPU;
- 64GB RAM min;
- 500 GB local storage.

Within the VM multiple container processes will deliver the functionality defined by an Airflow DAG. Attached storage (DAS, NAS, Object Storage) needs to support at minimum the required storage. Amount of data: 1TB per AOI (historical 2013, per update 4GB), 3 sites: 3TB historical.

5.3.4 Connectivity Map

Connectivity requirements for the Pilot Testbed are specified in the table below:

Producer	Consumer	Interface Protocol	Notes
Vineyard1	TVCore	https	
Vineyard2	TVCore	https	
CI/CD (Argo CD)	Vineyard1	https	
CI/CD (Argo CD)	Vineyard2	https	
CI/CD (Argo CD)	TVCore	https	

Funded by the European Union

Crate1	Vineyard1	https	
Crate2	Vineyard2	https	
Data Catalog	TVCore	https	only in the initial configuration
Data Catalog	Vineyard1	https	only in the initial configuration
Data Catalog	Vineyard2	https	only in the initial configuration

TABLE 5 - PILOT #3 - CONNECTIVITY MAP

5.4 Pilot #4 - Industry 4.0 fast KPI calculation

The Industry 4.0 fast KPI calculation Pilot focuses on the need for calculating a set of KPIs that are shared between two ERT Group plants based in different countries (Portugal and Czech Republic). In this section the dedicated Pilot Testbed is described.

5.4.1 Pilot Topology

This section presents the designed solution for the Industry 4.0 fast KPI calculation Pilot Testbed. A full description of the Pilot objectives and planned topology can be found in deliverable D2.1, under the “USE CASE PILOT #4: INDUSTRY 4.0” chapter.

The described topology is the one foreseen for the final Pilot Testbed configuration, to provide the most complete indication of all the needed resources. It is composed by two nodes:

- Czech plant (data provider/consumer): data lake hosting data of the Czech plant;
- Portuguese plant (data provider/consumer): data lake hosting data of the Portuguese plant + analytics to create KPI report combining the data from both locations.

For the first project iteration just one node will be set up, the Portuguese plant, while data consumers will act from a client (browser/service query client) for querying the Data Catalog and accessing the data.

5.4.2 Testbed Deployment Architecture

The Testbed deployment architecture for the Industry 4.0 pilot is presented in the diagram provided below.

Funded by the European Union

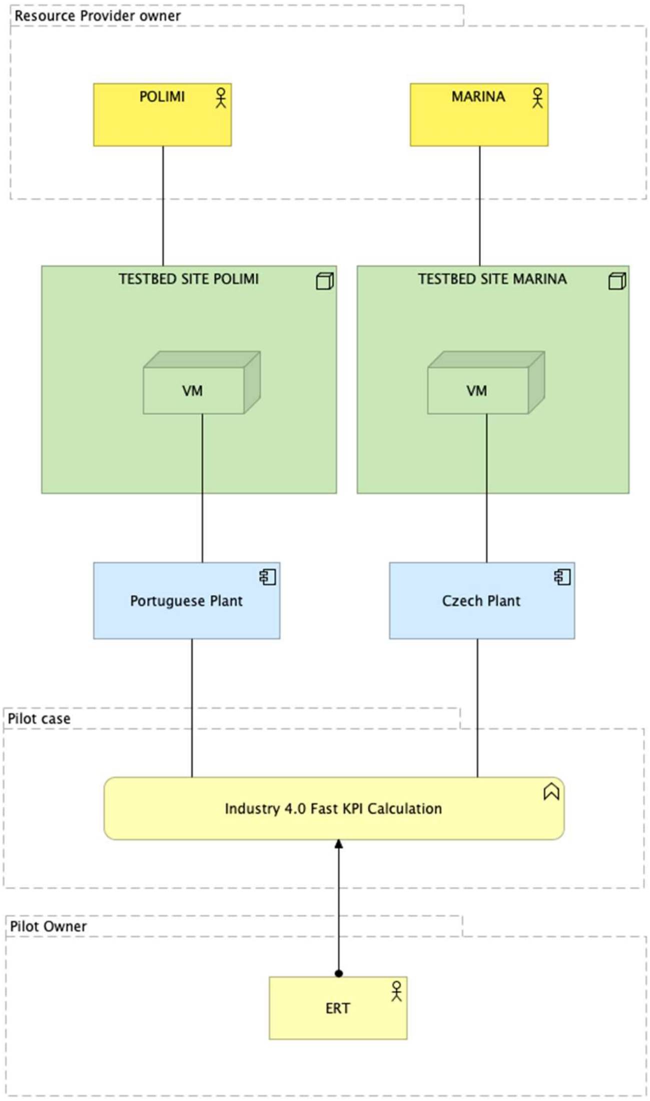


FIGURE 20 - PILOT #4 - TEADAL DEPLOYMENT ARCHITECTURE

Here's a detailed breakdown of its components and their interrelations:

1. **Pilot Case:** The business scenario representing the ‘Industry 4.0 Fast KPI Calculation’ Pilot as described above;
2. **Pilot Owner:** 'ERT’, as the pilot's primary owner, is accountable for the execution and alignment of the project alignment with the intended goals, objectives, and strategies, providing guidance and oversight throughout the pilot's lifecycle;
3. **Pilot Case Entities:** Two are the significant entities associated with the pilot, corresponding to TEADAL nodes:
   a. **Portuguese Plant:** This entity represents a manufacturing or production facility located in Portugal. Its operations and data play a double role in the Industry 4.0 pilot's objectives, acting both as data provider and consumer;

Funded by the European Union

   b. **Czech Plant:** This entity represents a production facility situated in the Czech Republic, contributing similarly mostly as data producer but receiving part of the final reports from the Portuguese Plant.

4. **TESTBED SITES**:
   a. **TESTBED SITE POLIMI:** In the administrative domain of the 'POLIMI', this site encompasses the 'VM' (Virtual Machine) acting as the primary computing environment, dedicated to tasks execution, simulations, and data storing; This Testbed site hosts the ‘Portuguese Plant’;
   b. **TESTBED SITE MARINA:** Associated with 'MARINA', this site offers a 'Private Cloud' featuring a secured and scalable environment for data storage and operations, optimising the computational footprint of the pilot. This Testbed site hosts the ‘Czech Plant.’
5. **Resource Provider Owner:** The diagram shows how this pilot relies on two distinct resource providers: 'POLIMI' and 'MARINA'. They are both responsible, each for its own part. for providing the Pilot Owner with the resources and infrastructure needed to operate this pilot's Testbed.

This diagram provides a clear and synthetic overview of the diverse stakeholders and components involved in the Industry 4.0 pilot.

5.4.3 Technical Notes

The two foreseen nodes are hosted in different Testbed Sites. The Czech TEADAL node is hosted in MARINA Testbed Site, whilst the Portuguese TEADAL Node is hosted at POLIMI.

The amount of foreseen data is 200 MB per week, 100 GB for total historical data.

5.4.4 Connectivity Map

Connectivity requirements for the Pilot Testbed are specified in the table below:

Producer	Consumer	Interface Protocol	Notes
Portuguese plant	Czech plant	https	
Czech plant	Portuguese plant	https	
CI/CD (Argo CD)	Portuguese plant	https	
CI/CD (Argo CD)	Czech plant	https	
Portuguese plant	Data Catalog	https	only in the initial configuration
Czech plant	Data Catalog	https	only in the initial configuration

TABLE 6 - PILOT #4 - CONNECTIVITY MAP

Funded by the European Union

5.5 PILOT #5 - SHARED FINANCIAL DATA GOVERNANCE

The Shared Financial Data Governance Pilot deals with the data sharing within ING group International Banks operating in multiple geographies. In particular, the Pilot case focuses on addressing both global and local policies due to multiple regulatory institutes, considering Enterprise/Global Domain located in the Netherlands and two Local Domains being Turkey and Australia. For compliant and efficient operation of the global financial institution there are a number of activities that need to leverage data and insights across different domains and geographies in a governed and efficient way. One of such activities is Know Your Customer (KYC), where the local domains need to provide data to the central unit in order to create a holistic view on the customer and detect any potential risks. In this section the dedicated Pilot Testbed is described.

5.5.1 Pilot Topology

This paragraph presents the solution for the Shared Financial Data Governance Pilot Testbed. A full description of the Pilot objectives and planned topology can be found in deliverable D2.2, under the “USE CASE PILOT #5: SHARED FINANCIAL DATA GOVERNANCE” chapter.

The described topology is the one foreseen for the final Pilot Testbed configuration, to provide the most complete indication of all the needed resources.

It is composed by three nodes:

- one Enterprise/Global Unit node (Consumer), where the data lake contains the global customer view needed for the KYC process. Foreseen storage capacity needed is 100GB for data;
- 2 Country/Domain nodes (Provider), with a data lake containing a number of data sets about customers from a certain domain/geography relevant for the KYC process. Foreseen storage capacity needed is 50GB for data.

Data flow will be bi-directional between Enterprise/Global Unit and 2 Country/Domains. First flow will require data from 2 Country/Domains to flow to the Enterprise/Global Unit where the global data set/report will be created. The second flow will showcase data/global set/report flowing from Enterprise/Global Unit to 2 Country/Domains. Additional storage capacity can be added for all elements when needed.

For the first project iteration just two nodes will be set-up. The first one is the Enterprise/ Global Unit acting as data Consumer and one Country/Domain node acting as data Provider.

Funded by the European Union

5.5.2 Testbed Deployment Architecture

In the diagram below the deployment architecture for Shared Financial Data Governance Pilot is presented.

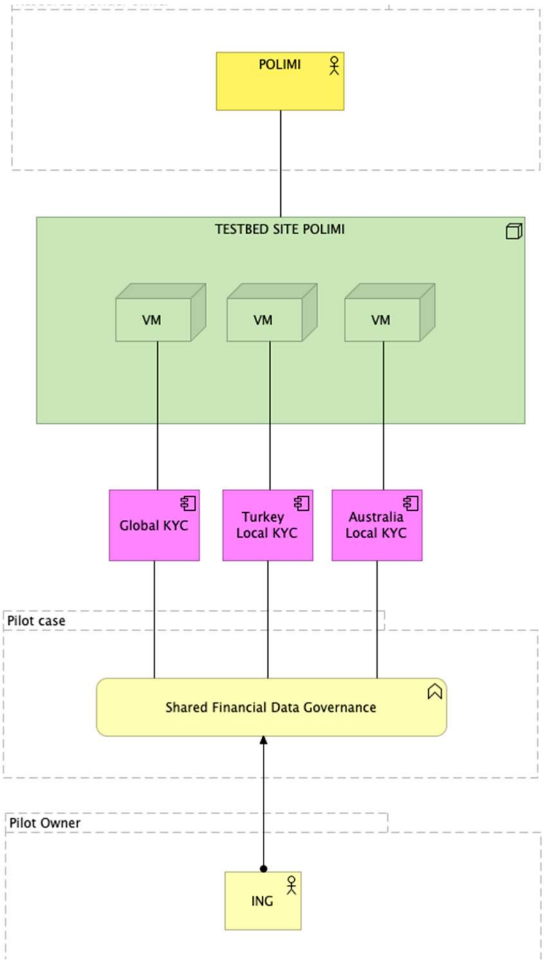


FIGURE 21 - PILOT #5 - TEADAL DEPLOYMENT ARCHITECTURE

The following is a breakdown of its components and their interactions:

1. **Pilot Case**: The business scenario representing the ‘Shared Financial Data Governance’ Pilot as described above.
2. **Pilot Owner**: The accountability of the entire pilot sits with ING. As the principal owner, they make sure the project progresses in line with its intended goals and objectives. It is ultimately accountable for the integration of the pilot Testbed and the execution of the pilot use case.
3. **Pilot Case Entities**: three are the primary entities providing data lake functionality to the pilot, by deployment of TEADAL Nodes:

   a) *KYC Enterprise/Global Unit*: This represents a global unit that is responsible for creation of the KYC models and global KYC reports. It relies on Countries/Domains to provide data needed for the KYC global process.

Funded by the European Union

Additionally, when created, Global Unit provides a global report back to the Countries/Domains;

b*) Country/Domain Australia and Country/Domain Turkey*. These are two countries where the data about the local customers is being registered. There are local KYC checks that happen in the countries themselves and additionally there is a subset of data that should be provided to the Global Unit for the global KYC process.

4. **TESTBED SITE POLIMI**: This is the designated location for the pilot's Testbed. Within this domain, there are three distinct Data Lakes, illustrated as 'VM' (Virtual Machines). These Data Lakes provide storage, governance, data movement for Shared Financial Data Governance pilot.
5. **Resource Provider Owner**: The 'Resource Provider Owner' is represented by ‘POLIMI’. This institution acts as a supplier for the Pilot Owner ING offering the needed resources and infrastructure needed to operate this pilot's Testbed.

5.5.3 Technical Notes

All the TEADAL Nodes for this pilot are hosted in POLIMI Testbed Site. The foreseen storage capacity needed is no more than 100 GB.

5.5.4 Connectivity Map

Connectivity requirements for the Pilot Testbed are specified in the table below:

Producer	Consumer	Interface Protocol	Notes
Turkey	Global	https	
Australia	Global	https	
CI/CD (Argo CD)	Turkey	https	
CI/CD (Argo CD)	Australia	https	
CI/CD (Argo CD)	Global	https	
Turkey	Data Catalog	https	only in the initial configuration
Turkey	Data Catalog	https	only in the initial configuration
Global	Data Catalog	https	only in the initial configuration

TABLE 7 - PILOT #5 - CONNECTIVITY MAP

Funded by the European Union

5.6 PILOT #6 - REGIONAL PLANNING FOR ENVIRONMENTAL SUSTAINABILITY

The Regional planning for environmental sustainability pilot's objective is to link sensor data from the BOX2M deployment of environment and energy consumption monitoring with building energy profiles administered by RT (Tuscany Region, Italy) public authority. In this section the dedicated Pilot Testbed is described.

5.6.1 Pilot Topology

In the current subsection the solution for the Regional Planning for Environmental Sustainability Pilot Testbed is presented. A full description of the pilot can be found in deliverable D2.1, under the "USE CASE PILOT #6: REGIONAL PLANNING FOR ENVIRONMENTAL SUSTAINABILITY" chapter.

The described topology is the one foreseen for the final Pilot Testbed configuration, to provide the most complete indication of all the needed resources.

It is composed by two actors, each one running a TEADAL node, each one running a TEADAL node.

- BOX2M (data consumer/data provider): has a Data Lake fed by devices deployed on a household BOX2M gathers part of the information from Regione Toscana and integrates them with IoT data from the household;
- RT (data consumer/data provider): owns data concerning the households and defines and validates analytics made by combining their own datasets with data produced by BOX2M nodes.

For the first project iteration just one node will be set-up, the BOX2M Node, while the data consumer will act from a client (browser/service query client) for querying the Data Catalog and accessing the data.

Funded by the European Union

5.6.2 Testbed Deployment Architecture

The diagram below presents the deployment architecture for the “Regional Planning for Environmental Sustainability” pilot.

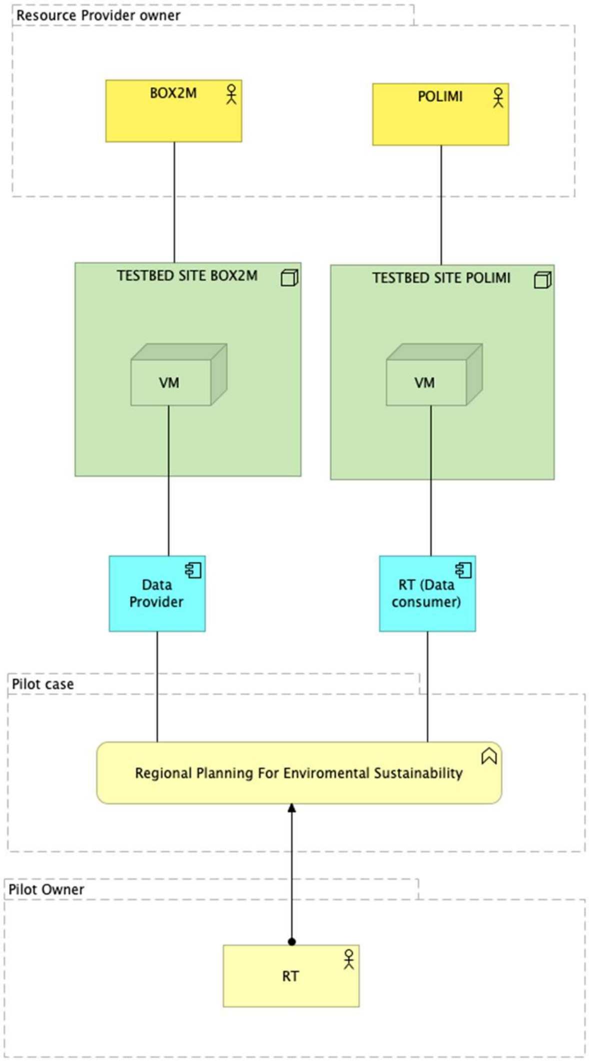


FIGURE 22 - PILOT #6 - TEADAL DEPLOYMENT ARCHITECTURE

The following is a breakdown of its components and their interactions:

1. **Pilot Case:** The business scenario representing the ‘Regional Planning for Environmental Sustainability’ Pilot as described above.
2. **Pilot Owner:** 'RT' is ultimately accountable for the entire pilot, from integration to operations to execution of use cases. RT ensures that the execution and outcomes of the project are aligned with the goals set, strategies, and objectives, providing guidance to involved partners throughout the pilot's lifecycle.
3. **Pilot case entities:** the system relies on two TEADAL Nodes

Funded by the European Union

   a. **BOX2M (Data Provider/Data Consumer):** This entity acts as a primary source of information, delivering data coming from the household; as a consumer, the node uses policy data received from RT;
   b. **RT (Data Provider/Data Consumer):** The node produces a data flow containing policies and consumes and utilises the data provided to make informed decisions and actions within the smart city framework.

4. **TESTBED SITES:**
   a. **TESTBED SITE BOX2M:** Provided by BOX2M, it houses a 'VM' (Virtual Machine) – the computing element for executing tasks, simulations, and data storage;
   b. **TESTBED SITE POLIMI:** Provided by POLIMI, it similarly deploys a 'VM', providing processing capabilities for the pilot.
5. **Resource Provider Owner:** The pilot Testbed relies on two administrative resource providers: both acting as a supplier for the Pilot Owner RT accountable for providing the relevant resources and infrastructure needed to operate this Pilot Testbed:
   a. **BOX2M;**
   b. **POLIMI.**

5.6.3 Technical Notes

The TEADAL Nodes for this pilot are hosted one in the POLIMI Testbed Site and one in the BOX2M Testbed Site. The amount of foreseen data does not exceed the capacity of 100 GB.

5.6.4 Connectivity Map

Connectivity requirements for the Pilot Testbed are specified in the table below:

Producer	Consumer	Interface Protocol	Notes
BOX2M	RT	https	
RT	BOX2M	https	
CI/CD (Argo CD)	RT	https	
CI/CD (Argo CD)	BOX2M	https	
BOX2M	Data Catalog	https	only in the initial configuration
RT	Data Catalog	https	only in the initial configuration

TABLE 8 - PILOT #6 - CONNECTIVITY MAP

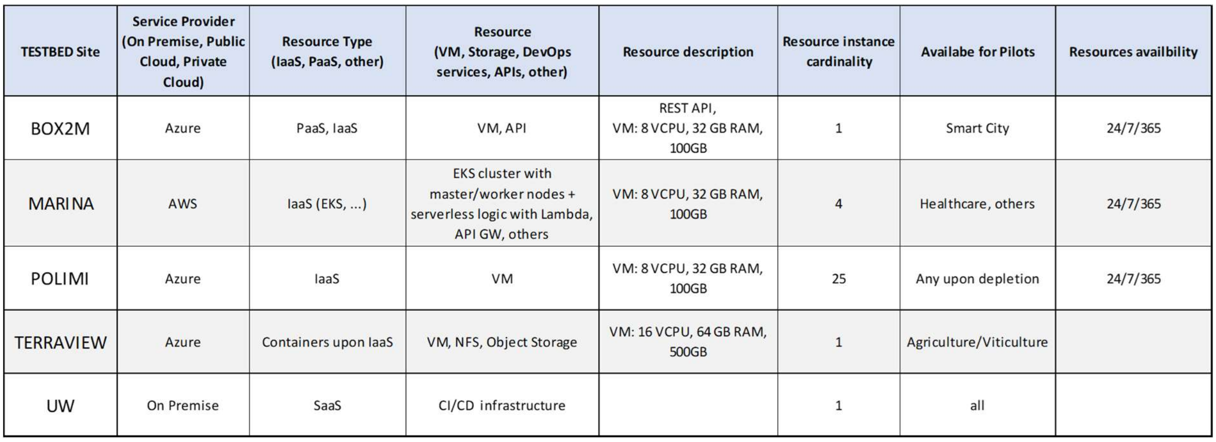
Funded by the European Union

6 "TESTBED SITE" RESOURCES AND TESTBED MATRIX

This section offers a synthetic summary of the resources collectively made available by the Testbed sites and the way they are allocated and distributed across the described pilot Testbeds.

6.1 AVAILABLE RESOURCES

The following table summarises the type of resources that the four TESTBED Sites can share for the different Pilots needs, as resulted from an interview with the corresponding partners in the initial phases of the project. Also included is the UW partner that provides resources for the CI/CD process.



TESTBED Site	Service Provider (On Premise, Public Cloud, Private Cloud)	Resource Type (IaaS, PaaS, other)	Resource (VM, Storage, DevOps services, APIs, other)	Resource description	Resource instance cardinality	Availabe for Pilots	Resources availbility
BOX2M	Azure	PaaS, IaaS	VM, API	REST API, VM: 8 VCPU, 32 GB RAM, 100GB	1	Smart City	24/7/365
MARINA	AWS	IaaS (EKS, ...)	EKS cluster with master/worker nodes + serverless logic with Lambda, API GW, others	VM: 8 VCPU, 32 GB RAM, 100GB	4	Healthcare, others	24/7/365
POLIMI	Azure	IaaS	VM	VM: 8 VCPU, 32 GB RAM, 100GB	25	Any upon depletion	24/7/365
TERRAVIEW	Azure	Containers upon IaaS	VM, NFS, Object Storage	VM: 16 VCPU, 64 GB RAM, 500GB	1	Agriculture/Viticulture	
UW	On Premise	SaaS	CI/CD infrastructure		1	all	

TABLE 9 - AVAILABLE RESOURCES PROVIDED BY TESTBED SITE PARTNERS

6.2 SITE VS. PILOT TESTBED MATRIX

To provide a general perspective and to demonstrate the interconnected nature of the project pilots, we present the following diagram that offers a comprehensive view of how each Pilot Testbed is deployed across the various Testbed Sites in the TEADAL operational infrastructure.

Funded by the European Union

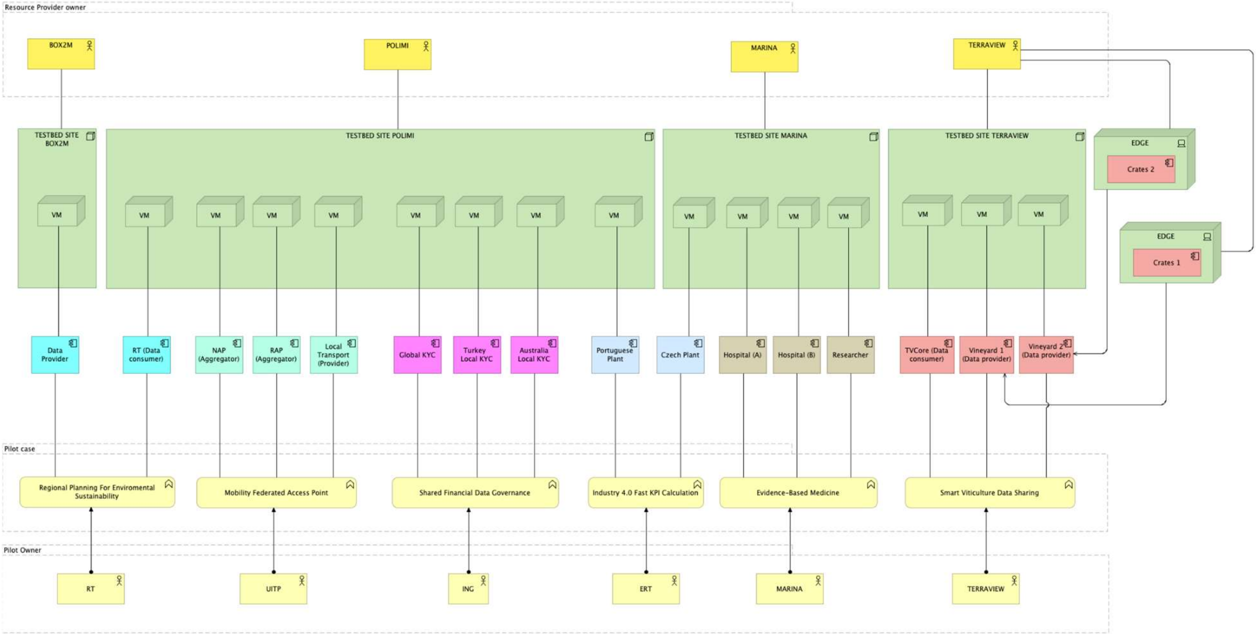


FIGURE 23 - THE TEADAL PROJECT'S PILOTS INFRASTRUCTURE OVERVIEW

As it can be observed from the diagram, the dynamic interrelation between pilots and TESTBED Sites demonstrates the adaptability and robustness of the designed Testbed. This visualisation also underscores the spirit of collaboration of the TEADAL consortium and the project's potential for scalable impact across diverse sectors.

Funded by the European Union

7 CONCLUSIONS

This deliverable describes the activities carried out for the definition of the TEADAL Testbed infrastructure to support the deployment of all the project Pilots.

Six Testbeds have been designed, one for each of the defined Pilots. Each Pilot Testbed has been described in terms of its topology and the resources allocated from the project Testbed Sites. They have been designed in their estimated final configuration at the end of the project. This estimate is based on a careful analysis of deliverables "*D2.1 Requirements of the Pilot Cases*" and "*D2.2 Pilot cases' intermediate description and initial architecture of the platform*" and on several meetings and interviews conducted with the pilot owners. However, if during next iterations of the project the conditions and configurations change, we may consider producing an update to this document for the sake of maintaining document consistency.

Once deployed, the Pilot Testbeds will host the Pilots according to the planned project iterations enabling them to validate the project results. Then, in the last months of the project, the final validation step will take place to validate the achievement of the project objectives.

Funded by
the European Union