Ref. Horizon-CL5-2023-D3-03
Grant Agreement 101160614

EU-DREAM

Effective Uptake of Digital Services to Repower European Consumers and Communities as Active Participants in Energy Transition and Markets

D1.2 System Architecture

EU-DREAM is supported by the European Union's Horizon Europe programme under agreement 101160614. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

DOCUMENT INFORMATION

DOCUMENT/ DELIVERABLE ID	D1.2 System Architecture
TYPE	Report
DISTRIBUTION LEVEL	Public
DUE DELIVERY DATE	30/06/2025
DATE OF DELIVERY	30/06/2025
VERSION	V1.0
DELIVERABLE RESPONSIBLE	LINKS
AUTHOR (S)	Marco Sacchet (LINKS) Nicolò Bertozzi (LINKS)
OFFICIAL REVIEWER/s	Amedeo Buonanno (ENEA) Giovana-Milenka Ciprian Herrera (AKKODIS)

DOCUMENT HISTORY

VERSION	AUTHORS	DATE	CONTENT AND CHANGES
v0.1	Marco Sacchet (LINKS)	28/03/2025	Table of Contents
v0.2	Marco Sacchet (LINKS)	19/05/2025	First Draft
v0.3	Marco Sacchet (LINKS) Nicolò Bertozzi (LINKS)	12/06/2025	Consolidated
v0.4	Amedeo Buonanno (ENEA) Giovana-Milenka Cipriàn Herrera (AKKO)	19/06/2025	Revision
v0.5	Marco Sacchet (LINKS) Nicolò Bertozzi (LINKS)	26/06/2025	Release Candidate
v1.0	Marco Sacchet (LINKS) Nicolò Bertozzi (LINKS) João Catalão (UPO)	30/06/2025	Final Version

Horizon-CL5-2023-D3-03
Grant Agreement 101160614

ACKNOWLEDGEMENTS

NAME	PARTNER
Nicolò Bertozzi	LINKS
Marco Sacchet	LINKS
Alessandro Mozzato	LINKS
Lucio Rocco Inglese	LINKS
Silvio Meneguzzo	LINKS
Alfredo Favenza	LINKS
Christian Destro	LINKS
Marco Pagliarini	LINKS
Babak Arbab Zavar	AAU
Amedeo Buonanno	ENEA
Martina Caliano	ENEA
Salvatore Fabozzi	ENEA
Valeria Palladino	ENEA
Maria Valenti	ENEA
Giovana-Milenka Ciprièn Herrera	AKKODIS
Ali Nasri	AKKODIS
Wail El Bani	AKKODIS
Maroua-Dorsaf Djelouat	AKKODIS
Philippe Szczech	AKKODIS
Marialaura Di Somma	ENSIEL/UNINA
Gianfranco Chicco	ENSIEL/POLITO
Angela Russo	ENSIEL/POLITO
Pandelis Biskas	AUTH
Stratos Keranidis	DOMX
Joao Catalao	UPORTO

A special thanks to all the colleagues listed in the "Acknowledgements" table for their valuable contributions during the WP Meetings, Consortium Meetings, and open discussions that helped shape a cohesive and comprehensive EU-DREAM Architecture.

Their expertise, insights, and commitment were instrumental in formalising the content presented and described throughout this document.

CONSORTIUM

Participant No.	Participant Organisation Name	Acronym	Country
1 (Coordinator)	Universidade do Porto	**UPO**	PT
2 (Partner)	Cleanwatts Digital SA	**CWD**	PT
3 (Partner)	EPRI Europe DAC	**EEU**	IE
4 (Partner)	SSE Airtricity LTD	**SSEA**	IE
5 (Partner)	DCSix Technologies Limited	**DCS**	IE
6 (Partner)	Aalborg Universitet	**AAU**	DK
7 (Partner)	Teknologian Tutkimuskeskus VTT OY	**VTT**	FI
8 (Partner)	AKKA High Tech – Akkodis Group	**AKKO**	FR
8.1 (Affiliated)	AKKA I&S	**AKKIS**	FR
8.2 (Affiliated)	MODIS France	**MODIS**	FR
9 (Partner)	Universidad de Castilla - La Mancha	**UCLM**	ES
10 (Partner)	Agenzia Nazionale per le Nuove Tecnologie, L'energia e lo Sviluppo Economico Sostenibile	**ENEA**	IT
11 (Partner)	Consorzio Interuniversitario Nazionale per Energia e Sistemi Elettrici	**ENSIEL**	IT
11.1 (Affiliated)	Politecnico di Torino	**POLITO**	IT
11.2 (Affiliated)	Università degli Studi di Napoli Federico II	**UNINA**	IT
11.3 (Affiliated)	Università degli Studi di Salerno	**UNISA**	IT
12 (Partner)	Fondazione Links - Leading Innovation & Knowledge for Society	**LINKS**	IT
13 (Partner)	IREN SPA	**IREN**	IT
13.1 (Affiliated)	IREN Mercato SPA	**IME**	IT
14 (Partner)	Aristotelio Panepistimio Thessalonikis	**AUTH**	EL
15 (Partner)	DOMX Idiotiki Kefalaiouchiki Etaireia	**DOMX**	EL
16 (Partner)	Vlaamse Instelling Voor Technologisch Onderzoek N.V.	**VITO**	BE

EXECUTIVE SUMMARY

This deliverable presents the work conducted within Task 1.2, titled “Overall System Architecture Design”, which spans from M1 (July 2024) to M12 (June 2025), with submission scheduled at M12.

It defines the **Reference Architecture** of the EU-DREAM platform, which aims to enable the effective uptake of digital energy services by empowering consumers and communities to actively participate in the energy transition and flexibility markets. Developed in accordance with the ISO/IEC/IEEE 42010:2022 standard and structured following the 4+1 View Model, the document provides a comprehensive description of the system’s logical, development, deployment, process, and scenario views.

It introduces a modular and scalable infrastructure integrating key components such as Digital Twin Orchestrators, AI-based optimisation algorithms, NLP-driven user interfaces, and secure, interoperable data services built on technologies like Docker, Kubernetes, Apache Kafka, and MongoDB.

The architecture addresses the needs of diverse stakeholders, including prosumers, DSOs, energy communities, and vulnerable users, through role-based access control, real-time monitoring, and compliance with EU regulatory frameworks. It provides a robust foundation for the technical implementation of the platform, ensuring replicability, user-centric design, and alignment with EU-DREAM’s goal of democratising digital energy participation across Europe.

For this reason, the work presented in this document will be validated across the Living Labs.

Table of Contents

Horizon-CL5-2023-D3-03
Grant Agreement 101160614

Horizon-CL5-2023-D3-03
Grant Agreement 101160614

LIST OF FIGURES

LIST OF TABLES

Introduction

EU-DREAM – Effective Uptake of Digital services to Repower European consumers and communities as Active participants in energy transition and Markets – aims to accurately reflect end-users' preferences and expectations in a simplified manner, improving customers' awareness, trust, and confidence in their interactions with the energy market by introducing a smart intermediator – as an energy attorney – for end-users.

Real empowerment will come from translating all the complex details of the energy market into simple and transparent language that relates to the user's everyday experience, like comfort, temperature, and utility bills. Citizens need to be aware of the implications of their preferences and routines associated with energy use, rather than knowing all technical details.

Therefore, an Artificial Intelligence (AI)-based assistant tool and Natural Language Processing [1] based intermediator will be created in EU-DREAM, as shown in Figure 1, so that end-user preferences can be discussed in a common layperson's language. A Digital Twin[2] of the household will be developed based on real-time data being acquired by sensors.

The digital service of the household, which is federated into wider DTs of energy communities, is a unique tool to examine the implications for energy and flexibility markets, proposing new market design guidelines supported by novel consumer-oriented business models. The AI tool, in turn, will control EMS technical settings, and the NLP interface will greatly simplify the experience of managing energy services, shielding end-users from all the technical jargon and the sheer complexity of the smart grid and the services it allows, which often act as engagement hindrances.

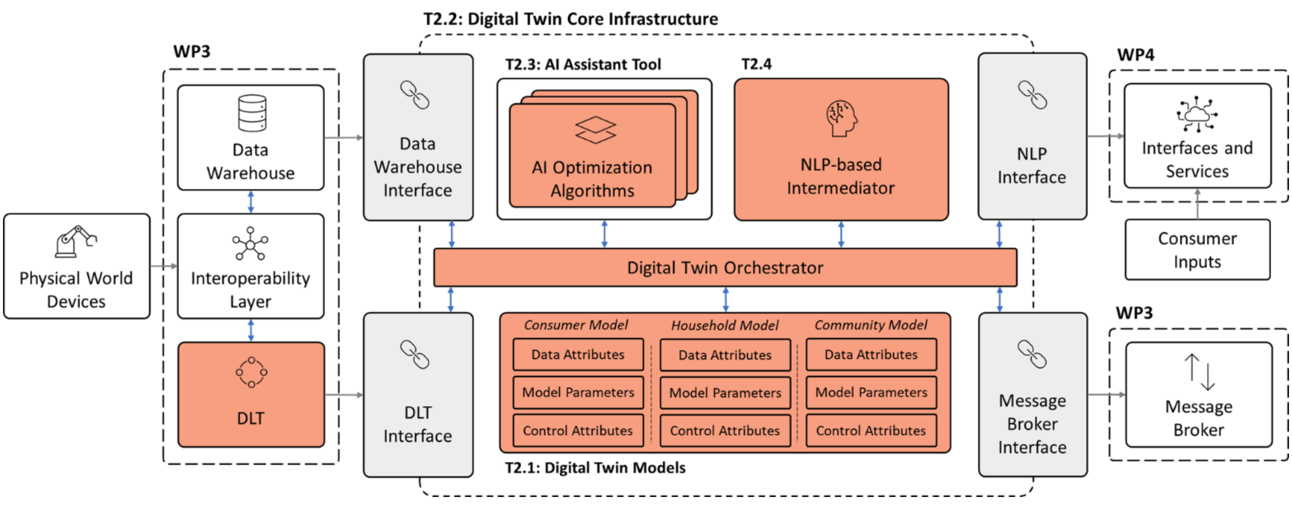

Figure 1: Digital Twin Proposal Architecture

[1] NLP is the field of study focused on teaching computers to understand and interact using human language in its natural, everyday form.

[2] a DT is a virtual replica of a physical object, system, or process that helps us monitor, analyse, and optimise its real-world counterpart.

Horizon-CL5-2023-D3-03
Grant Agreement 101160614

Digital platforms and tools, represented in Figure 2, will be designed and developed for both non-experienced and experienced users to significantly simplify their participation in the energy market.

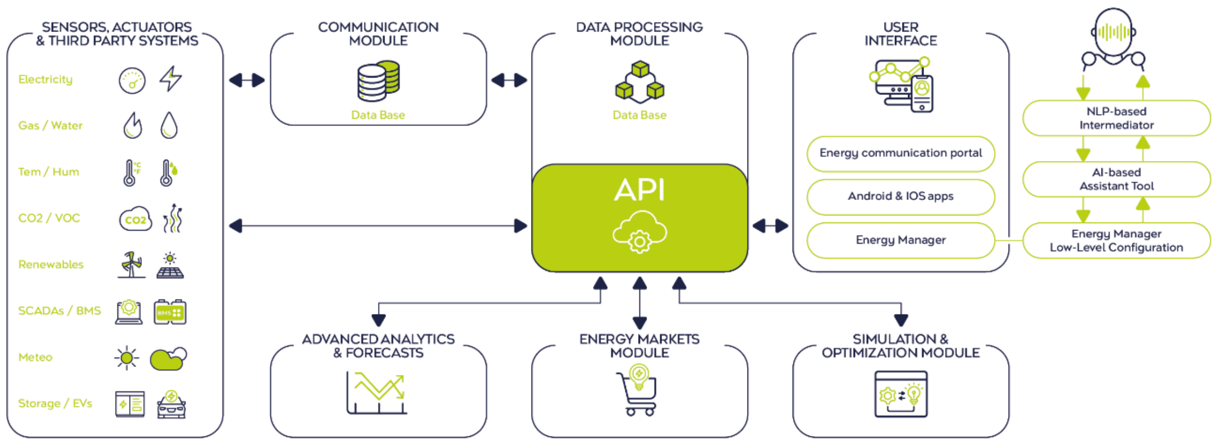

Figure 2: Digital Platform Holistic Architecture in a high-level view with logical and architectural components together with representation of possible workflows

A Living Lab (LL) is a real-world testing environment. These LLs will be realised in 6 EU countries (Portugal, Belgium, Italy, Ireland, Greece, and Denmark) with the following denominations and goals. The solutions developed within the project will be validated in 5 LLs, with different levels of integration and deployment. In addition to these ones, the Belgium LL will co-develop with vulnerable households to gather needs and motivations, informing socially inclusive energy tools. All these characteristics are listed in Table 1.

Table 1: Living Lab Descriptions.

LL Number	**Description**	**Location**
LL1	AI-Powered Renewable Energy Communities	Coimbra, *PORTUGAL*
LL2	Energy Poverty and Vulnerable Consumers Impacts	Genk, *BELGIUM*
LL3	Energy Flexibility at Residential Scale and Interoperability	Northern Italy, *ITALY*
LL4	Consumers Empowerment for Energy Management	Dublin, *IRELAND*
LL5	Smart Energy Services for Multi-Energy Vectors	Thessaloniki, *GREECE*
LL6	DT-enabled Residential IoT Microgrid	Aalborg, *DENMARK*

Architecture Design

This chapter provides an overview of the ISO/IEC/IEEE 42010:2022 (ISO/IEC/IEEE, 2022) standard as a core methodology for developing architectural descriptions (ADs) in software-intensive systems. It explores how this standard applies to the 4+1 Architectural View Model, incorporating views such as Logical, Physical, Process, Development, and Scenarios. This approach seeks to address diverse stakeholder concerns and promote a holistic understanding of software architecture.

ISO/IEC/IEEE 42010:2022 Standard

The current standard for architecture design, ISO/IEC/IEEE 42010:2022, evolves from previous standards, particularly IEEE 1471, published in 2000 as the first significant approach to software architecture design. This standard, titled "Systems and software engineering - Architecture description," provides a structured methodology for creating architectural descriptions (ADs) for software-intensive systems, offering a step-by-step workflow.

The process involves identifying and documenting stakeholders, understanding their architecture-related concerns, selecting architecture viewpoints to address these concerns, generating architecture views with corresponding models, ensuring view consistency, and documenting the rationale behind architectural decisions.

ISO/IEC/IEEE 42010:2022 heavily relies on viewpoints and views to capture different facets of a software system, allowing a focused examination of specific concerns while ensuring overall architectural consistency. This approach has proven effective in several instances, delivering higher-quality artifacts and deliverables compared to less structured methods that attempt to address all issues simultaneously.

Viewpoints are essentially collections of patterns, templates, and conventions designed to construct specific types of views, such as the functional viewpoint, which includes the necessary functions for architecture description. IEEE 1471 aimed to establish best practices for the software and systems communities, addressing the fragmented landscape of architectural practices and description languages prevalent at the time. Its adoption led to recognising the benefits of a common conceptual foundation and terminology within the software architecture community.

The standard has evolved over the years, influenced by a decade of use of the first edition in industry, academia, and government.

In its current iteration, the standard defines a comprehensive methodology for the architectural description of software-intensive systems, including steps like identifying stakeholders, architecture-related concerns, selecting viewpoints, creating views with models, and analysing consistency. It continues to use viewpoints and views to document different aspects of a software system, ensuring a coherent architectural design. The relationships between the components of this standard, described below, are pictured in Figure 3.

Horizon-CL5-2023-D3-03
Grant Agreement 101160614

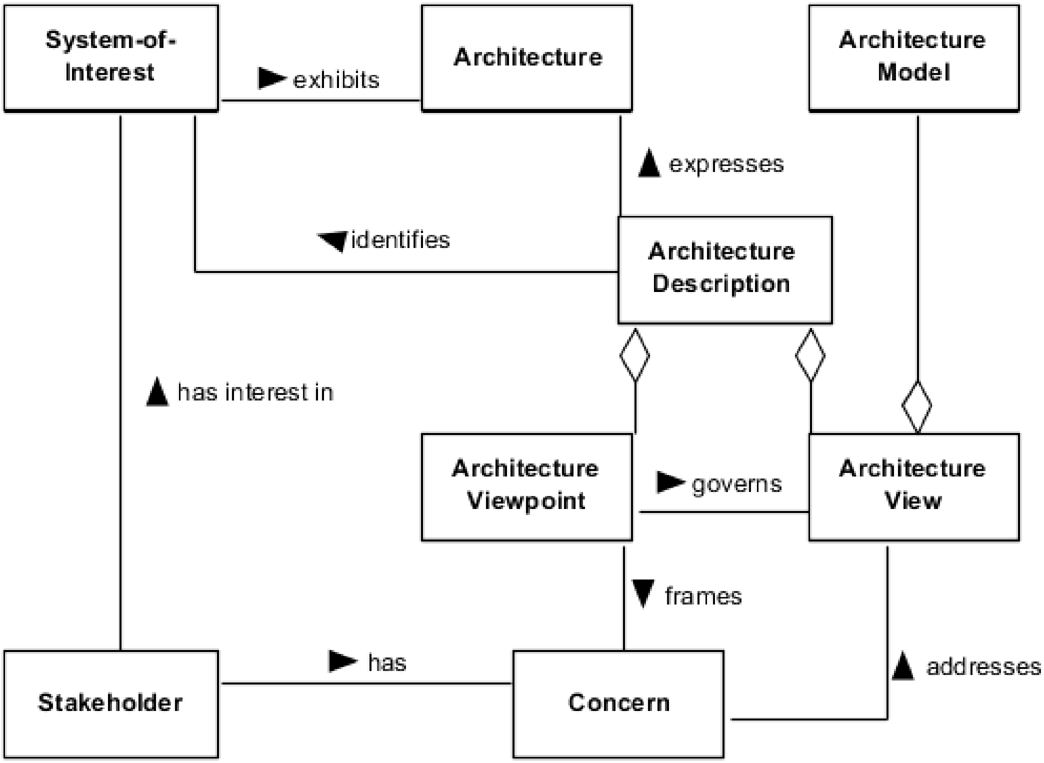

Figure 3: Architecture concepts and their relationships from ISO/IEC/IEEE 42010:2022

General Definitions

System of Interest

The system under investigation, whose architecture is being studied or considered.

Stakeholders

An individual, group, organisation, or other entity that has a vested interest in the development, deployment, or operation of the system, and may affect—or be affected by—its realisation.

Architecture

The fundamental concepts and characteristics of a system within its operational environment, expressed through its components, their relationships, and the principles governing their design and evolution. It defines the system's structure, key properties, modes of interaction with external entities, and the rationale underlying architectural decisions.

Architectural Description

A collection of artifacts intended to document and communicate the architecture of a system, including models, views, and diagrams (e.g., component diagrams, data flow diagrams) that describe its components, relationships, and guiding design principles.

Concern

An interest or area of importance to one or more stakeholders that pertains to the system. Concerns may encompass functional or non-functional requirements, quality attributes, design constraints, or broader objectives that stakeholders expect the system to fulfil or address.

View

A representation of a system from the perspective of a specific set of concerns, comprising one or more models and accompanying rationale or explanations. Each view is constructed in accordance with a defined.

Viewpoint

A set of conventions, templates, and modelling patterns that guide the construction of a specific type of view.

Model

A simplified representation of one or more aspects of a system's architecture often conforms to a specific notation or language, such as UML.

4+1 Architectural View Model

Rozanski and Woods (Rozanski & Woods, 2012) defined a multi-view framework for describing software architecture, designed to address the concerns of different stakeholder groups and to offer a comprehensive understanding of the system. It comprises four primary views: Logical, Development (or Implementation), Process, and Physical. Each one captures distinct aspects of the architecture. The '+1' refers to the Scenario View, which illustrates how the elements, from the four primary views, collaborate to realise key use cases or scenarios, thereby validating and integrating the other views through concrete examples of system behaviour. The 4+1 Architectural View Model includes the following perspectives: logical view, physical view, process view, development view and scenario view, detailed below.

Logical View (Functional Viewpoint)

A representation of the system that addresses its functional requirements by modelling key abstractions such as classes, objects, or components and their relationships. This view emphasises the responsibilities, interactions, and collaborations among these elements, capturing the high-level design and structure of the system's functionality.

Physical View (Deployment Viewpoint)

A representation of the system that describes the mapping of software components onto the underlying hardware infrastructure. This view captures the physical deployment architecture, including processing nodes, network topology, and communication links. It addresses concerns related to system distribution, performance, and scalability.

Process View

A representation of the system that focuses on its dynamic and runtime behaviour, particularly regarding concurrency, synchronisation, and inter-process communication. This view models the system's processes, threads, and their interactions, detailing how components execute in parallel and how they coordinate and exchange information.

Horizon-CL5-2023-D3-03
Grant Agreement 101160614

Development View

A representation of the system from a software engineering perspective, focusing on the organisation, structure, and management of the software's implementation. This view describes the software's modular architecture, such as packages, modules, or subsystems, and includes guidance on development practices, coding standards, toolchains, and version control strategies.

Scenarios (Use Cases) View

A view that illustrates the system's behaviour through representative scenarios or use cases, emphasising interactions between the system and external actors under various conditions. This view serves to validate and integrate the other architectural views by demonstrating how the system fulfils specific functional requirements in practical, real-world contexts.

Methodology

In this section, we present the methodology adopted by EU-DREAM for the architectural design of its software systems in alignment with the ISO/IEC/IEEE 42010:2022 standard. The objective is to elucidate the foundational principles and conceptual framework that inform the consortium's architectural decisions. This clarification serves to make the rationale underpinning the design choices made throughout the project transparent and well-grounded.

Architecture Design Process

The foundation for this process is the ISO/IEC/IEEE 42010:2022 standard. The EU-DREAM project adopted the process, aligning it with the mentioned standard (see Figure 4).

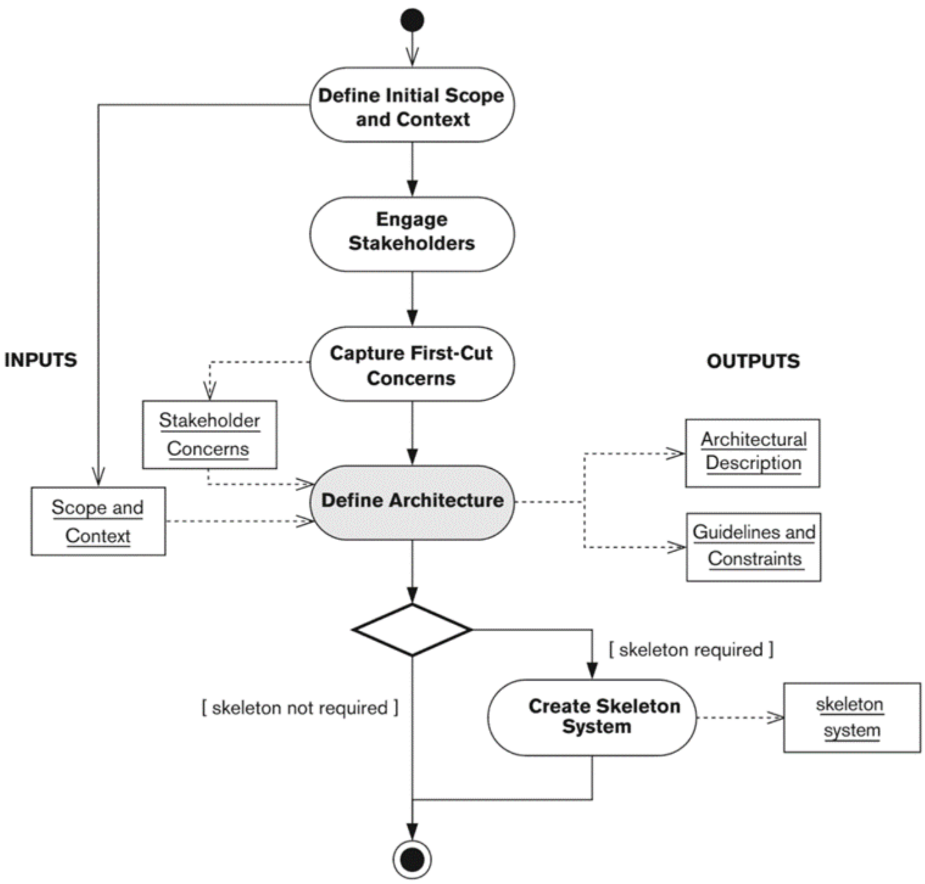

Figure 4: Activities Supporting Architecture Definition from ISO/IEC/IEEE 42010:2022

The architectural process began with the definition of the system's Scope and Context. This initial phase aimed to establish a clear understanding of the platform's intended environment and operational boundaries. A preliminary document was produced to provide an overview of a range of tools potentially applicable to the platform, presenting a comparative analysis of their respective advantages and limitations. The purpose of this document was to support informed decision-making by identifying tools that best align with the platform's customisation requirements, based on the defined scope and context.

Following this, a stakeholder analysis was conducted, as detailed in the subsequent subsection. This analysis focused on identifying and characterising the various individuals, groups, and organisational entities involved in the operational and production workflows of the participating factories. These stakeholders, representing the platform's end users, exhibit diverse expectations regarding system functionality, leading to differentiated access rights and permission levels.

The insights gained from the stakeholder analysis, along with the previously defined scope and context, served as inputs for the architectural design phase. This culminated in the formal definition of the platform's architecture, including both the architectural description and a set of guidelines and constraints to steer the system's evolution and ensure alignment with stakeholder needs.

The final step, described in the Architecture Views, involves the development of a system skeleton. This phase translates the gathered requirements and architectural decisions into an initial, structured implementation. The system skeleton serves as a foundational framework, ensuring that all previously collected inputs are coherently integrated into a cohesive and extensible platform.

Analysis of Stakeholders and Requirements

As previously discussed, a diverse range of stakeholders is involved in the organisational and operational ecosystem addressed by the EU-DREAM project. These stakeholders occupy different roles and exhibit distinct information needs, expectations, and technical capabilities. From this broader landscape, several categories can be identified—each engaging with the EU-DREAM platform in different ways. These users anticipate varied functionalities and services, leading to differentiated access levels, permissions, and interface requirements. This diversity underscores the importance of designing a flexible, user-centric architecture capable of accommodating multiple stakeholder profiles and ensuring equitable access to the digital tools and energy services provided by the platform.

The list of stakeholders involved in the use of the platform can be found in Table 2.

Table 2: Stakeholders.

Stakeholder Groups	**Role**
Individual Energy Consumers	Use simplified digital platforms to manage energy usage and improve efficiency.
Digital Services Customers	Access digital services that inform and optimise energy-related decisions.
Prosumers	Benefit from tools that support both energy consumption and local energy production.
Low-literacy Energy Users	Gain access through simplified interfaces and guided digital services.
Vulnerable Consumers	Receive tailored support to reduce energy poverty and promote equitable access.
Energy Communities	Adopt and test flexible, community-driven energy solutions and digital platforms.
Consumer & Environmental Organisations	Represent social and ecological interests, helping align the project with public values and fairness.

Horizon-CL5-2023-D3-03
Grant Agreement 101160614

Local Distribution System Operators (DSOs)	Facilitate real-world deployment and testing of platform services on local grids.
Regulatory Organisations	Oversight of market behaviour and support for policy development based on platform insights.

According to the roles and responsibilities assigned to each stakeholder, the functionalities expected from the platform vary significantly across user groups. This variation leads to the formulation of a corresponding set of system requirements for the EU-DREAM infrastructure. These requirements, which can be found in Table 3, are derived from stakeholders' expectations and are further informed by the functional needs identified for each module of the platform. As a result, they are translated into structured architecture comprising discrete functional blocks, modules, or components. Each one of these, designed to deliver specific capabilities aligned with the users' needs.

Table 3: Platform Requirements.

ID	Definition	Description
R1-001	Modular and Scalable Architecture	The system architecture shall be modular and designed to allow for future expansion and adaptation. This includes the integration of additional services, tools, or pilot cases without disrupting existing functionality. Scalability shall support replication across different geographic locations, user types, and energy systems.
R1-002	Data Interoperability	The platform shall support interoperability with heterogeneous data sources, including legacy systems, IoT devices, building management systems, weather forecast and smart meters. It should utilise open standards and APIs to enable seamless data exchange and integration with both internal and third-party components.
R1-003	Simulation and Forecasting Tools	The platform shall include predictive tools capable of forecasting energy demand and production and pricing trends. These tools support stakeholders in strategic planning, operational decision-making, and proactive grid engagement.
R1-004	Role-Based Access Control	The system shall implement a secure access management framework that provides differentiated rights and permissions based on the user.
R1-005	Real-Time Monitoring	The platform shall be able to receive, manage and process real-time data, enabling users and operators to observe energy production, consumption, and flexibility indicators.

Horizon-CL5-2023-D3-03
Grant Agreement 101160614

R1-006	Energy Flexibility Services	The platform shall deliver functionalities that enable energy flexibility management, such as automated demand response, load shifting, and the coordination of distributed energy resources.
R1-007	Integration with Market Platforms	The system shall support integration with external energy markets, local flexibility markets, or aggregators' platforms.
R1-008	Compliance with Regulatory Frameworks	All components of the platform shall adhere to relevant European and national regulations, including but not limited to GDPR for personal data protection, energy market rules, and standards for cyber-security and interoperability.

The requirements listed above may be further detailed or adapted as needed to better align with the specific objectives and context of the project's implementation.

Required Components

In Figure 5 it is possible to see the functionalities within which all the EU-DREAM Infrastructure components are included. These functionalities, and therefore these components, have been selected based on the requirements listed and fully described in Table 3. Also, the connections between functionalities and the project work's partition are represented in Table 4.

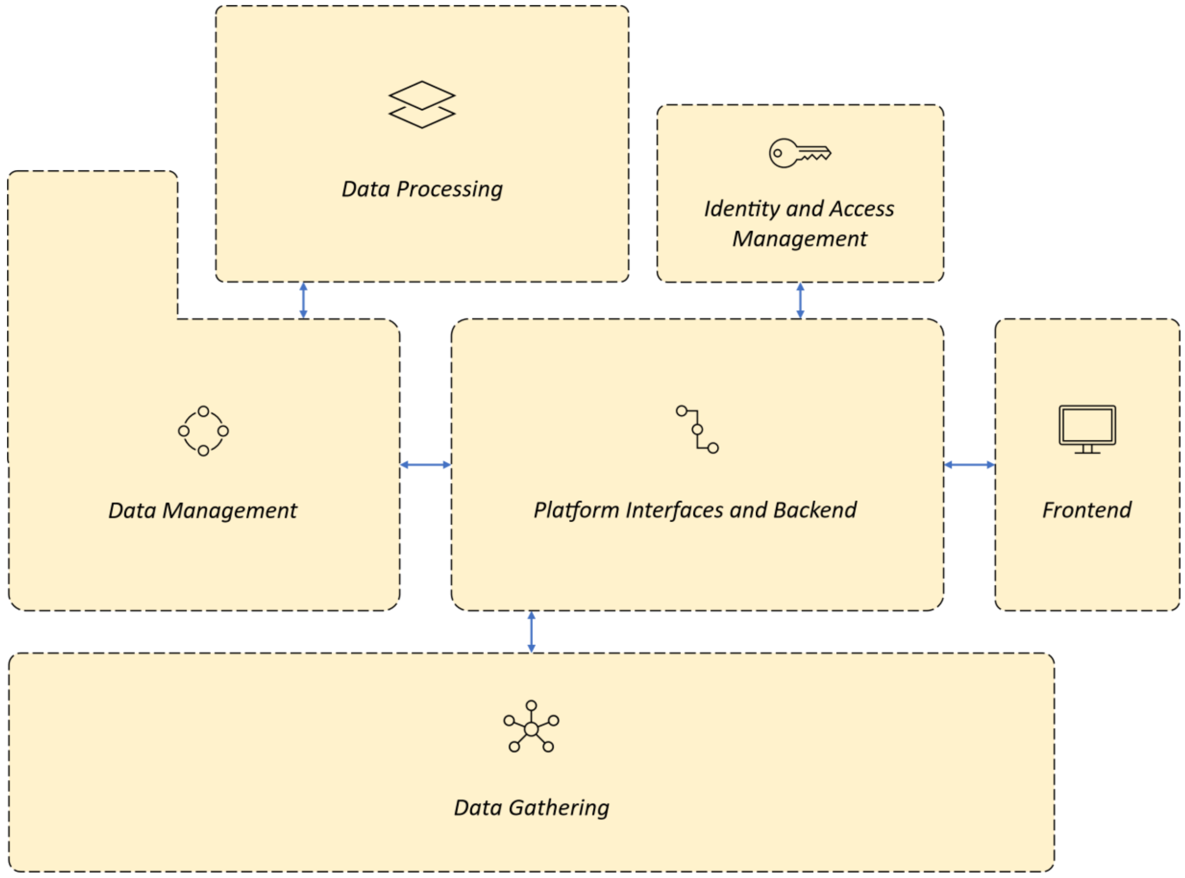

Figure 5: Macro-level modules of the EU-DREAM infrastructure.

Horizon-CL5-2023-D3-03
Grant Agreement 101160614

EU-DREAM

Table 4: Mapping between infrastructure functionalities and tasks.

Functionality	Associated Task	Associated WP
Data Processing	T2.1, T2.2, T2.3	2
Data management	T3.1, T3.2, T3.3, T3.4	3
Platform Interfaces and Backend	T2.4, T4.2	2,4
Frontend	T4.4	4
Data Gathering	T3.4, T7.2, T7.3, T7.4, T7.5, T7.6, T7.7	3,7

Data Processing

The Data Processing layer of the EU-DREAM infrastructure provides intelligent decision-making and personalised energy services. This functionality integrates the modelling of energy consumer behaviours through Digital Twins, enabling the platform to simulate, predict, and respond to user-specific consumption patterns. These models are continuously updated with real-time and historical data, allowing for dynamic adaptation to changing conditions. The core orchestration infrastructure coordinates simulation cycles and manages the synchronisation of data across edge and cloud environments, ensuring consistency and responsiveness. On top of this, AI-based tools analyse processed data to generate forecasts, optimise energy usage, and provide user-specific recommendations.

Data Management

The Data Management functionality within the EU-DREAM infrastructure is the foundation for secure, interoperable, and scalable handling of energy-related data across all platform layers. The trustworthiness, integrity, provenance and tamper resistance of the data are guaranteed by Distributed Ledger Technology (DLT). Building upon this layer, the platform incorporates a semantic data space that standardises heterogeneous data sources and enables interoperability between components and external systems. This semantic layer facilitates seamless data sharing while ensuring alignment with European data governance principles. A dedicated set of services is responsible for managing the acquisition, storage, access, and delivery of data in a modular and efficient manner, supporting both real-time and historical analytics. These services are integrated into a comprehensive data platform architecture that constitutes a robust data management stack.

Platform Interfaces and Backend

The Platform Interface and Backend layer of the EU-DREAM infrastructure enables user interaction with the system while coordinating internal operations across distributed components. This layer includes a natural language-based intermediator that facilitates intuitive communication between users and the platform, translating user inputs into structured queries that can be processed by the backend services. The backend itself orchestrates the execution of requests, manages workflows, and ensures that data and services are appropriately routed through the platform's modular architecture.

Frontend

The Frontend of the EU-DREAM platform, by means of a mobile application specifically developed, provides users with a coherent and accessible interface through which they can visualise, interpret, and interact with energy-related data and platform functionalities. It enables users to monitor consumption patterns, receive AI-generated recommendations, participate in energy markets, and manage preferences in real time.

Data Gathering

The Data Gathering functionality in the EU-DREAM infrastructure is dedicated to the systematic collection of operational, environmental, and behavioural data from the various pilot sites deployed across the project's Living Labs.

Architecture Views

Scenario

The scenario serves as the foundational reference for the logical, process, physical, and development views of the architecture. It provides a unifying narrative that captures the platform's intended functionality and outlines the sequence of technical steps required for its realisation. Within the EU-DREAM project, this scenario is anchored in the Grant Agreement, which defines its scope through work packages and task-level specifications. From a high-level perspective, the scenario envisions a citizen-centric energy ecosystem in which various actors interact with a digital platform to manage their energy usage more efficiently. To achieve this goal, a tight integration between real data and forecasting is essential.

Logical View

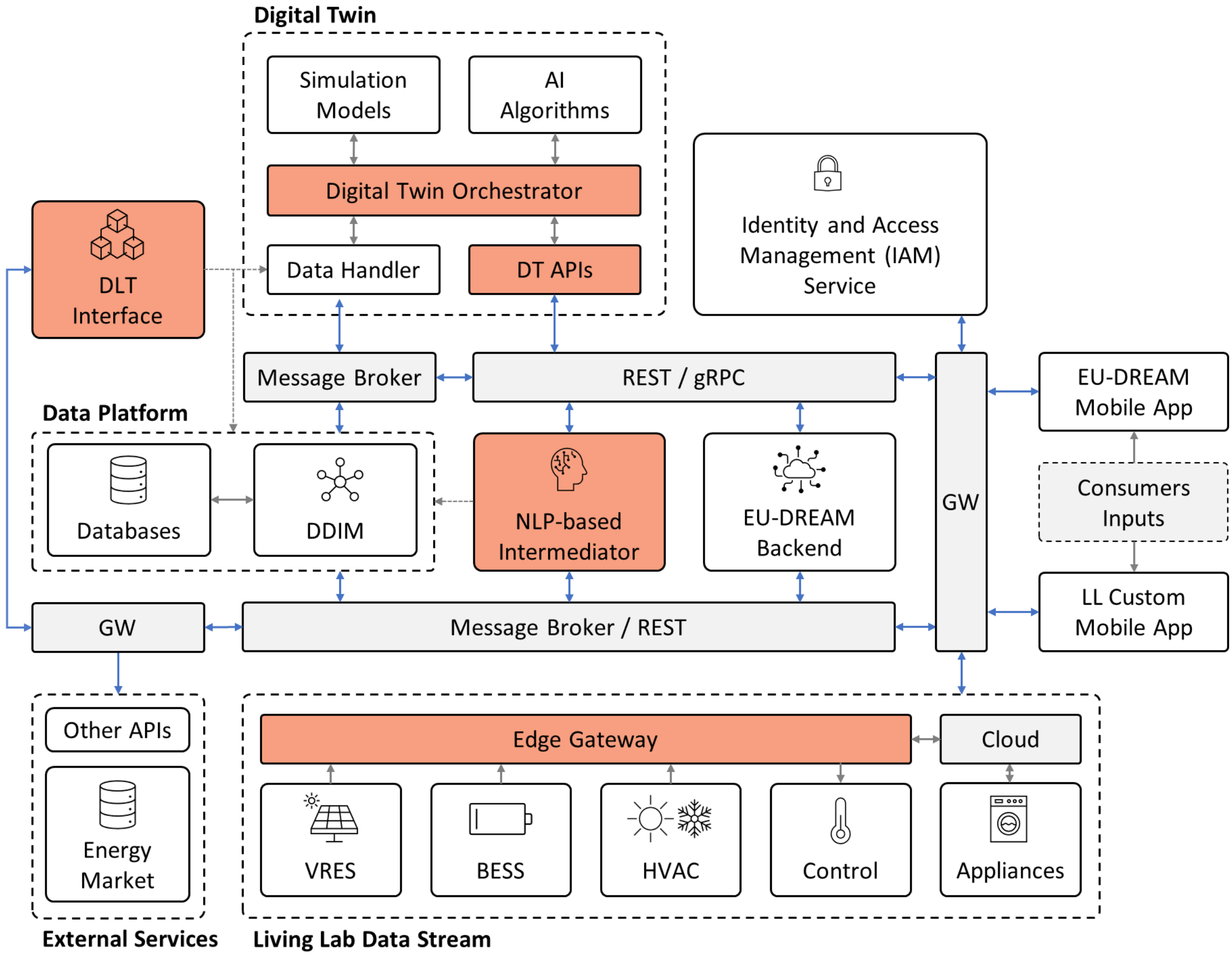

Figure 6: Logical View.

The primary objective of Logical View (Figure 6) is to represent the core functionalities offered by the platform to its various user groups. This view captures the system's key abstractions (such as services, components, and their interactions), focusing on how the platform fulfils user needs and functional requirements, independently of implementation or deployment details.

Horizon-CL5-2023-D3-03
Grant Agreement 101160614
EU-DREAM

It is possible to define a relationship between each component of the Logical View and the project tasks, as listed in Table 5.

Table 5: Mapping between components and task.

Component	Associated Task	Associated WP
Simulation Model	T2.1	2
Digital Twin Orchestrator	T2.2	2
AI Algorithms	T2.3	2
Data Handler	T2.2	2
Digital Twin APIs	T2.2	2
NLP-Based Intermediator	T2.4	2
DDIM	T3.2	3
Database	T3.3, T3.4	3
Message Broker	T3.3	3
DLT Interface	T3.1	3
Identity and Access Management Service	T3.3	3
Energy Market	T4.2, T4.3	4
Gateway	T3.4	3
EU-DREAM Backend	T4.2	4
EU-DREAM Mobile App	T4.4	4
Living Lab Data Stream	T3.4, T7.2, T7.3, T7.4, T7.5, T7.6, T7.7	3, 7

In the following sections, the roles of each component will be discussed.

Simulation Model

The Simulation Model component enables virtual replication of household or grid energy behaviour under varying conditions. It allows users or services to test scenarios, such as peak demand or control strategy effects, through real-time or historical data simulations. It helps in exploring energy savings, testing comfort thresholds, or evaluating policy impacts, forming a core analytical engine of the platform.

Digital Twin Orchestrator

The Digital Twin Orchestrator coordinates the interaction of various simulation and optimisation agents, enabling streamlined execution of user-triggered or automated tasks. It ensures that data flows from external sources are securely ingested, verified, and routed to the correct agent, supporting concurrent tasks and modularity. Its abstraction layer simplifies interactions across distributed Living Labs and heterogeneous data contexts.

Horizon-CL5-2023-D3-03
Grant Agreement 101160614

AI Algorithms

These agents perform data-driven analysis and decision support by applying machine learning models to predict, optimise, and recommend energy usage strategies. They allow the system to offer personalised suggestions (e.g., when to shift loads or charge batteries), supporting goals like cost reduction or sustainability. Their modularity allows the integration of new models as needs evolve.

Data Handler

The Data Handler acts as a mediator between the Digital Twin and the platform's data sources. It ensures that data requests are properly routed, validated via the blockchain interface, and formatted for consumption by the simulation or AI algorithms. It also flags non-verified data, enhancing trust in outputs.

Digital Twin APIs

These RESTful APIs enable external components (e.g., NLP-based Intermediator, EU-DREAM backend) to trigger Digital Twin functions such as launching simulations or retrieving results. They form the primary interface layer between higher-level platform logic and the DT core, ensuring standardised, secure, and traceable communication.

NLP-Based Intermediator

This natural language interface allows users, regardless of technical expertise, to interact with the platform via conversational input. It translates spoken or written queries into actionable API calls, then reformats the responses back into human-readable language. This component significantly lowers the barrier for citizen engagement.

Database

The EU-DREAM platform employs a multi-database architecture (Samarta, Gunawan, & Syahputra, 2024) to support different data types and use cases. MongoDB handles unstructured data, InfluxDB stores time series from sensors, and PostgreSQL manages structured information like user profiles and configurations. This setup ensures optimal performance and flexibility in data storage, enabling scalable and efficient access for all platform components.

DDIM

The Data and Device Interoperability Module (DDIM) is responsible for standardising data across heterogeneous sources by leveraging semantic models and ontologies. Acting as an interoperability layer, it ensures consistent data representation and enhances compatibility among components such as the Data Platform, NLP interface, and external services. DDIM supports seamless integration by translating device and source-specific data into a common, machine-readable format.

Message Broker

A key enabler of event-driven architecture, the Message Broker manages asynchronous communication between platform components like the DT, Data Platform, NLP-based Intermediator, and AI algorithms. It ensures decoupled, real-time data exchange and service responsiveness.

Horizon-CL5-2023-D3-03
Grant Agreement 101160614

DLT Interface

The Distributed Ledger Technology (DLT) Interface provides cryptographic verification of data using Merkle trees and a private, permissioned blockchain (Hyperledger Besu). It ensures tamper-proof traceability and accountability for data ingested, used, or shared across the platform, particularly for simulation, market participation, and compliance needs.

Identity and Access Management

The Identity and Access Management (IAM) service governs secure authentication and fine-grained authorisation across users and services. It ensures that data access complies with GDPR and platform-specific policies by managing roles, credentials, and access logs.

Gateway

This gateway enables bidirectional communication between the EU-DREAM platform and external energy markets or aggregators. It supports tasks like sending bids, receiving pricing signals, and managing contractual interactions, enabling user participation in flexibility and trading services.

EU-DREAM Backend

The backend component routes requests between the user interface, the NLP-based Intermediator, and internal services. It handles data streams, request authentication, and asynchronous orchestration of services. Once a task is initiated, it allows downstream services to communicate autonomously, minimising latency and overhead.

EU-DREAM Mobile App

A mobile application offering user-centric access to platform functionalities. It allows consumers to visualise energy usage, interact with AI recommendations, configure preferences, and communicate with the DT via the NLP-based Intermediator. It simplifies participation in energy services for non-technical users.

Living Lab Data Stream

This component encompasses the continuous flow of real-time and historical data collected from sensors, appliances, HVAC systems, and other IoT devices deployed across EU-DREAM's distributed Living Labs. These data streams feed into the platform's core services, such as simulation, optimisation, and analytics, via message brokers and are validated for integrity through the DLT interface before being stored and made available for reuse by other modules.

Development View

The Logical View illustrated in Figure 7 and explained in Table 6 can be operationalised through the implementation of widely adopted tools and technologies commonly used in distributed system architectures, as exemplified in the general Development View shown in Figure 7. While this deployment representation provides a more concrete instantiation of the abstract architecture described in the previous subsection, it remains at a high level.

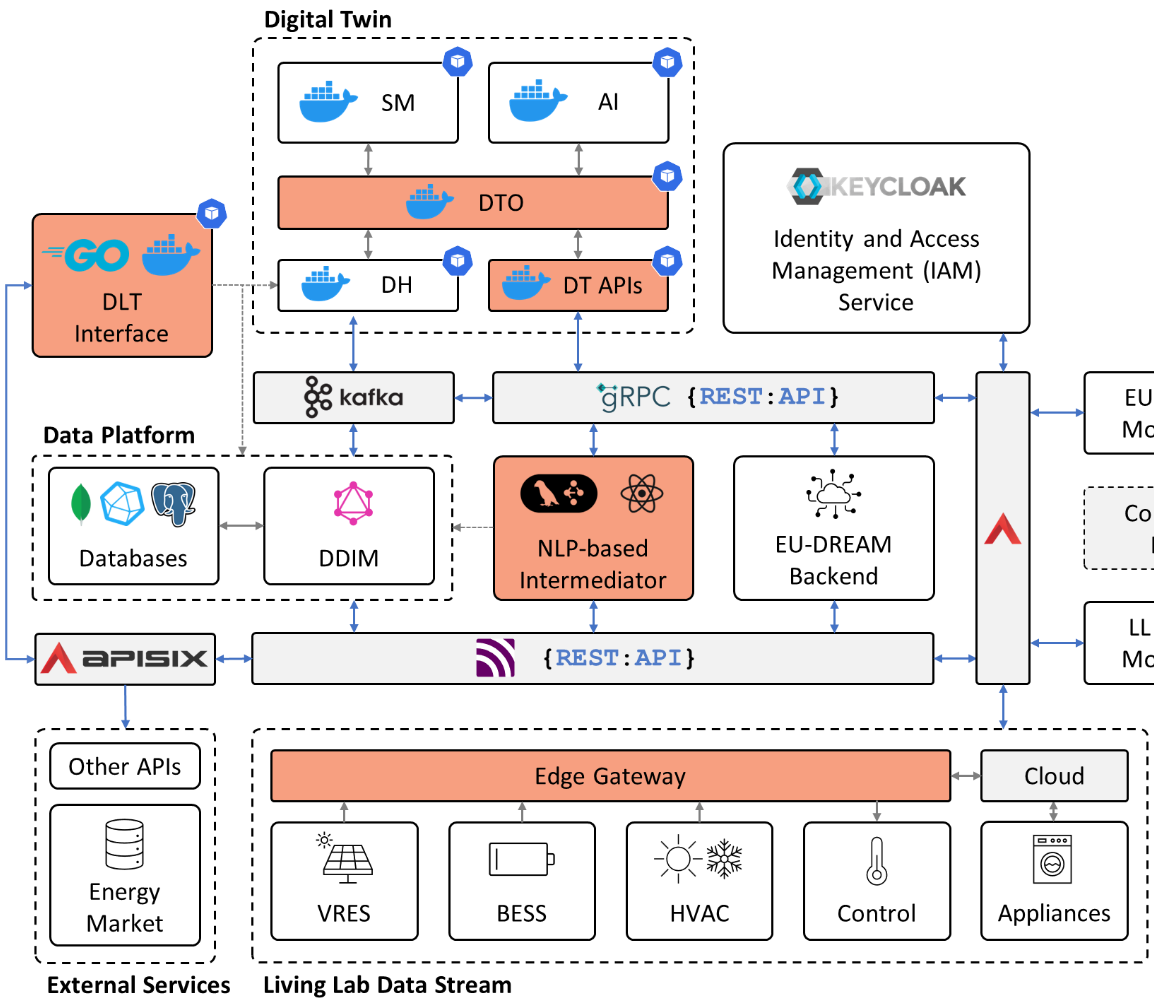

Figure 7: Every component with the corresponding technology used to implement it.

From a standards-based perspective, a comprehensive Deployment View would typically incorporate additional technical specifications relevant to distributed systems such as: Load balancing strategies, secret and credential management, SSL encryption, high-availability and replication mechanisms, disaster recovery protocols, and dynamic service instantiation. These aspects, however, will be detailed extensively in the technical deliverables associated with Work Packages 2, 3, and 4. This document instead focuses on presenting the set of tools considered during the design phase and outlines their integration into the architectural framework from a high-level, conceptual standpoint.

Table 6: Components-technology relation

Component	**Technology**
Digital Twin	Docker Containers
	Kubernetes
	Kafka
	Eclipse Mosquitto
	Python
Data Platform	MongoDB
	Influx DB
	PostgreSQL
	GraphQL
	Kafka
	Eclipse Mosquitto
DLT Interface	Docker Containers
	GO
Gateway	Apache APISIX
Identity And Access Management	Keycloak

Docker Containers

Docker (Acharya & Suthar, 2021) serves as a foundational tool for containerising and deploying applications across heterogeneous environments, including both edge devices and cloud infrastructures. Its strength lies in its modular integration of key container technologies—such as *runc*, which implements the Open Container Initiative (OCI) runtime specifications, and *containerd*, a robust container runtime that manages the full container lifecycle. It also provides a standardised and reproducible deployment environment, which is critical for ensuring portability and consistency across the platform's distributed components. In order to support more complex and scalable deployments, Docker seamlessly interoperates with orchestration tools such as Docker Compose and Kubernetes. These enable declarative service definitions, automated scaling, and fault tolerance.

Kubernetes

Kubernetes (Burns, Grant, Oppenheimer, Brewer, & Wilkes, 2016) is a leading orchestration platform for managing containerised applications in a scalable, distributed environment. It follows a declarative configuration model that automates deployment, scaling, and operational tasks while abstracting hardware-level details. Within the EU-DREAM platform, Kubernetes operates in tandem with Docker, using containers as its primary packaging format and enhancing them with orchestration capabilities. Tools such as kubectl and Helm further streamline service configuration and lifecycle management. Meanwhile are also incorporated networking components such as Kubernetes Service for load-balanced endpoints, kube-proxy for traffic routing, and the Ingress Controller for managing external access and routing policies. These features collectively enhance performance, scalability, and reliability across the containerised infrastructure.

Apache Kafka

Apache Kafka (Kreps, Narkhede, & Rao, 2011) is a distributed, open-source event streaming platform designed for high-throughput, real-time data processing across scalable infrastructures. At its core, Kafka organises data into topics: this enables persistent, fault-tolerant message storage and decoupled communication between producers and consumers. Producers publish data to topics, while consumers subscribe to them asynchronously, all mediated by Kafka brokers that manage routing, durability, and load balancing. This architecture supports fast, reliable, and scalable data exchange.

MongoDB

MongoDB is a NoSQL, document-oriented database designed for flexibility, scalability, and high performance in handling semi-structured data. It stores data in JSON-like documents, allowing for dynamic schemas that adapt easily to evolving application needs. It is particularly well-suited for heterogeneous data sources. This tool enables rapid access to complex datasets while supporting horizontal scaling and seamless integration with other data services in the architecture.

Influx DB

Influx DB is a high-performance, time-series database optimised for storing and querying data points indexed by time. It is specifically designed to handle high-throughput ingestion of metrics, events, and sensor readings. These characteristics made it ideal for energy monitoring, environmental tracking, and IoT applications.

PostgreSQL

PostgreSQL is a powerful, open-source relational database known for its reliability, ACID (Atomicity, Consistency, Isolation, and Durability) compliance, and extensibility. It excels in managing structured data with complex relationships, making it well-suited for storing user credentials, access policies, regulatory metadata, and configuration tables. PostgreSQL serves as the backbone for structured data storage, supporting transactional integrity and advanced querying.

Apache APISIX

Apache APISIX is a high-performance, open-source API gateway designed for scalable and dynamic API management in distributed systems. It enables efficient routing, traffic control, and protocol transformation between platform components and external services. Key features include dynamic load balancing, hot reconfiguration via RESTful APIs, and native support for plugins such as authentication, rate limiting, and observability.

Keycloak

Keycloak (Divyabharathi & Cholli, 2020) is an open-source Identity and Access Management (IAM) solution developed by Red Hat, offering centralised user authentication, single sign-on (SSO), authorisation, and social login capabilities. It supports industry-standard protocols such as OAuth 2.0 and OpenID Connect. However, while this tool excels at human-facing identity management, it is not optimised for securing service-to-service communication in distributed microservice environments.

SPIFFE/SPIRE

The platform may integrate Keycloak with SPIFFE (Secure Production Identity Framework for Everyone) and SPIRE (SPIFFE Runtime Environment), which provide cryptographically verifiable identities to services and enable secure, policy-driven communication across components. In this hybrid model, Keycloak would govern authentication and role-based access control for users, while SPIFFE/SPIRE ensures secure and dynamic identity management between backend modules, establishing a comprehensive and robust security architecture.

Eclipse Mosquitto

Eclipse Mosquitto is a fast, resource-efficient MQTT broker designed for lightweight publish-subscribe messaging. It can facilitate both edge-to-core communication and inter-service messaging within tightly constrained environments. Its asynchronous, decoupled architecture makes it suitable for connecting modular services that require low-latency, event-driven communication. While not a full replacement for more robust message brokers like Kafka, Mosquitto offers a simple and scalable option for handling real-time data exchange across lightweight components and distributed services.

Deployment View

The platform described in this deliverable is designed to operate through a hybrid deployment model that spans both cloud-based infrastructure and edge environments within Living Labs. Core services such as the Data Platform, DLT Interface, Digital Twin Orchestrator, and central AI algorithms are initially envisioned to reside in the cloud, providing scalability, centralised data governance, and interoperability across pilots. At the edge, components such as data acquisition services, device-level monitoring, lightweight analytics, and localised control interfaces are expected to be deployed directly within the Living Labs. These edge instances allow for reduced latency, enhanced responsiveness, and context-specific data processing close to the source.

However, this configuration, shown in Figure 8 may represent a worst-case scenario in terms of centralisation. The project will actively investigate opportunities to shift selected components to the edge, depending on the security constraints and integration requirements of each pilot site.

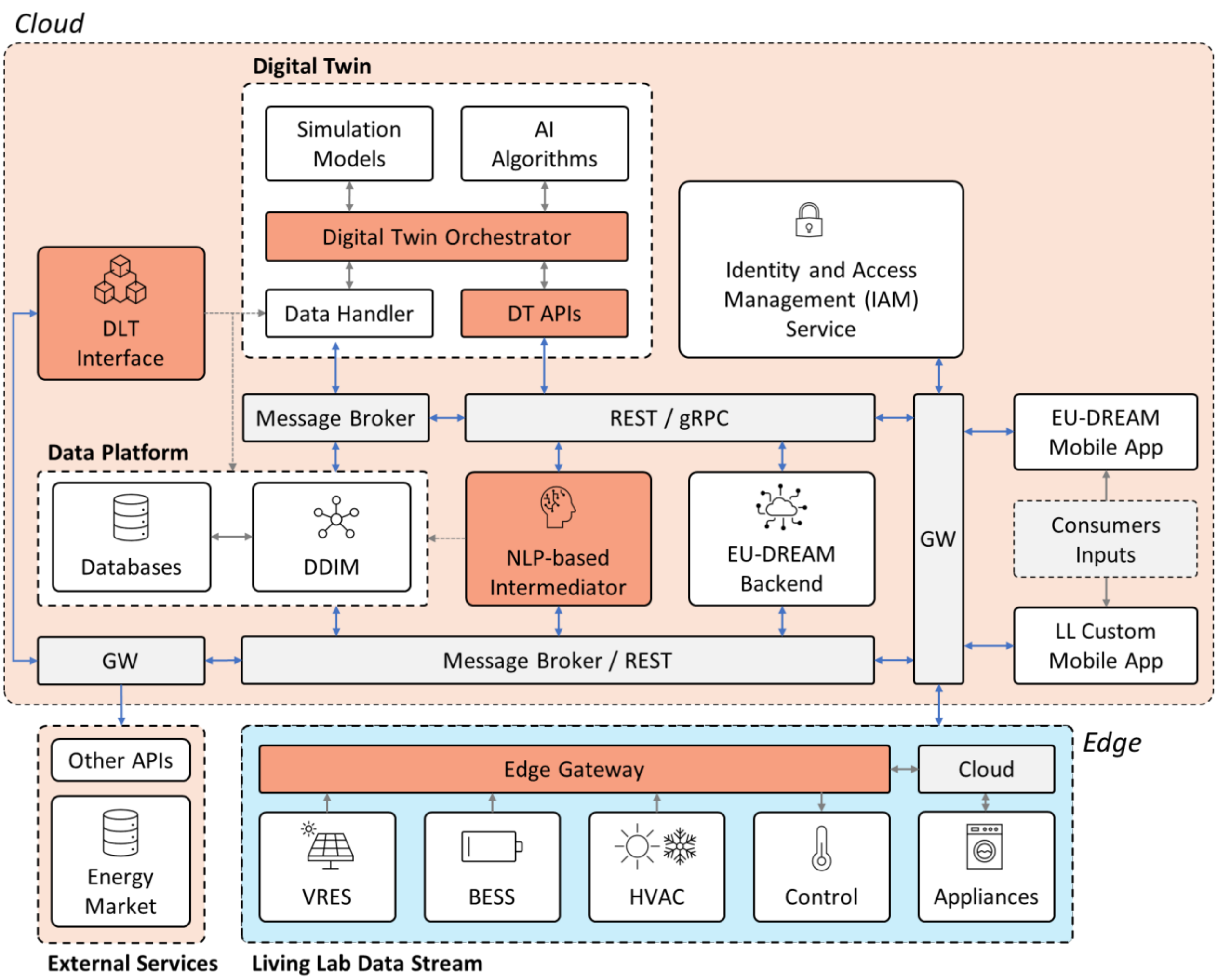

Figure 8: Deployment view.

Process View

Ensuring trust and traceability of data across the EU-DREAM platform is fundamental to the architecture's ability to support secure, decentralised, and verifiable energy management services. This section's aim is to present the data verification workflow, represented in Figure 9, as implemented through the integration of the Data Platform (composed of Database and DDIM), the DLT Interface, and the Digital Twin environment.

Data Ingestion

The process begins at the edge level, where raw data, originating from sensors, smart appliances, HVAC systems, or energy assets, is streamed into the Data Platform. Upon ingestion, each data batch is assigned a universally unique identifier (UUID) and persisted in the appropriate storage layer based on data type and structure.

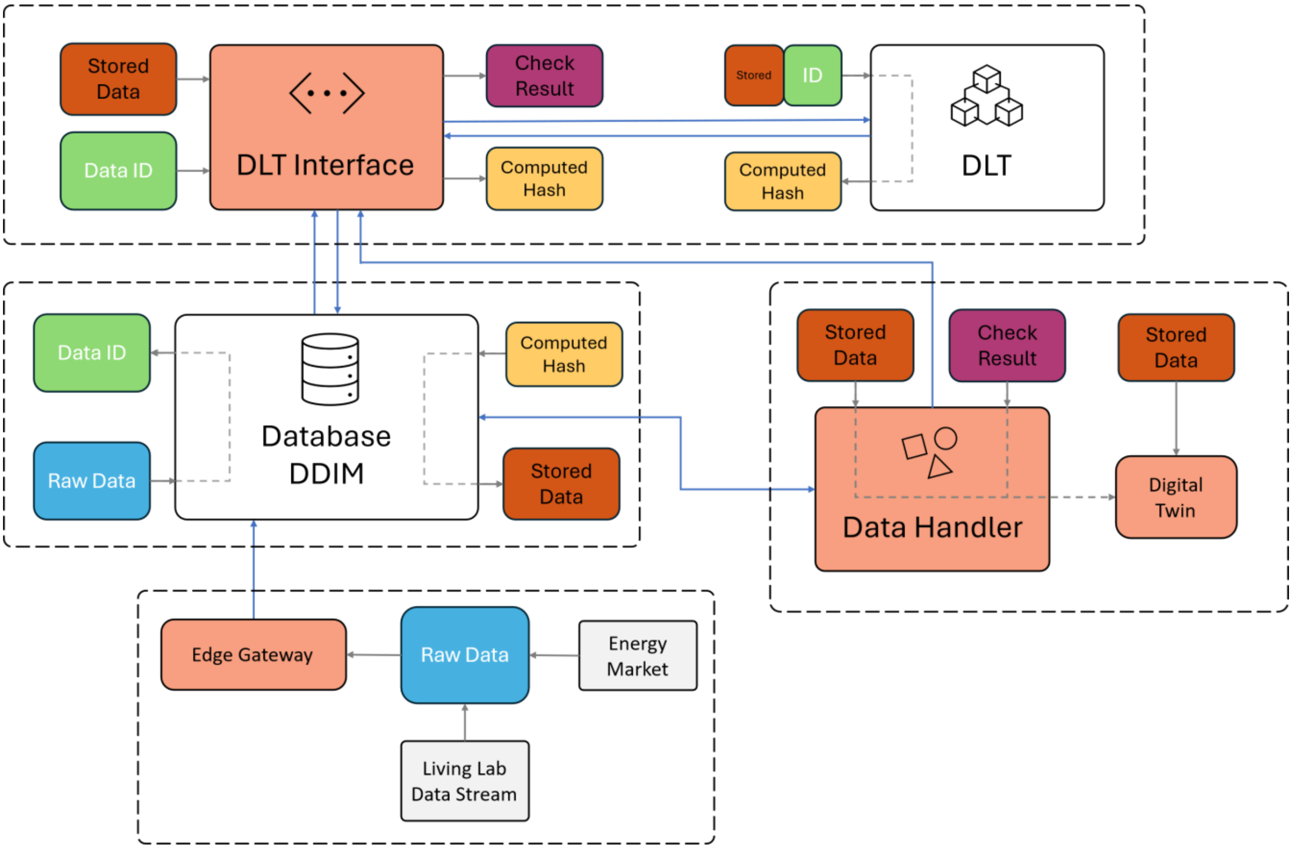

Figure 9: Process view.

Once stored, the data (along with its UUID) is forwarded to the DLT Interface. This interface computes the corresponding Hash, representing a cryptographic fingerprint of the entire batch. This hash and its metadata are then anchored on the DLT, implemented as a permissioned blockchain, ensuring tamper-proof data provenance and auditability. The computed hash is subsequently returned from the DLT to the DLT Interface and then stored back in the Data Platform, now linked to the original dataset. This ensures that each dataset is paired with a verifiable cryptographic proof.

Horizon-CL5-2023-D3-03
Grant Agreement 101160614

Data Retrieval

When a service such as the Data Handler initiates a data request, typically on behalf of a Digital Twin simulation or AI algorithms, the platform performs a structured retrieval. The Data Platform provides both the requested data and the associated hash recorded during the ingestion phase. Then, the Data Handler sends this package (data + hash) to the DLT Interface, which re-computes the hash and queries the DLT to compare the data stored on-chain. This step ensures that the data has not been tampered with or altered since its original ingestion.

If the verification is successful, a validation result is sent back to the Data Handler, which can then securely pass the verified data to the Digital Twin. In the event of a verification failure, the system can alert the user or service, flagging the data as unverifiable, thereby maintaining trust without compromising system integrity.

Roles of Involved Partners

In Table 7, a preliminary listing is provided to identify the concrete inputs expected from the partners involved in Task 1.2. The same is graphically illustrated in Figure 10.

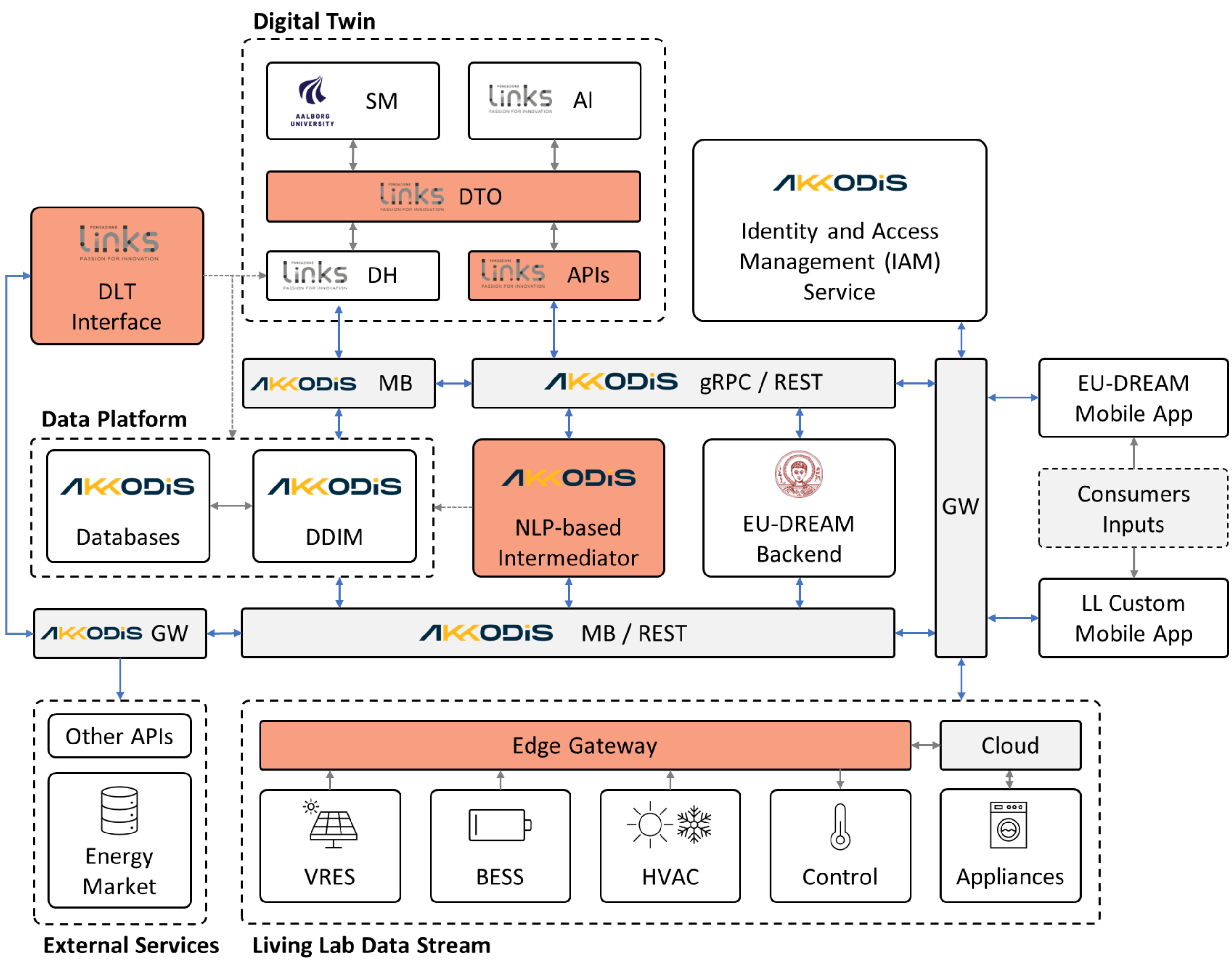

Figure 10: Partner View

Table 7: Contribution from Involved Partners.

Partner	Related Task(s)	Contribution Needed
AAU	T2.1 T7.7	Confirm if the proposed architecture ensures data availability, API consistency, and compatibility across Living Lab deployments. Provides requirements for the data gathering components.
AKKO	T2.4 T3.2 T3.3 T3.4	Confirm architectural compatibility with technologies planned for the semantic interoperability layer and NLP intermediator, ensuring alignment with integration specifications.

Horizon-CL5-2023-D3-03
Grant Agreement 101160614

EU-DREAM

AUTH	T4.1 T4.2	Provide input on architectural and integration requirements to ensure compatibility with user-centric components, including simplified interfaces and AI-powered platform features.
DOMX	T4.4 T7.6	Ensure alignment of digital visualisation and access tools with backend architecture, focusing on data flow, service interfaces, and access control. Provides requirements for the data gathering components.
CWD	T7.2	Provides requirements for the data gathering components.
VITO	T7.3	Provide insights on requirements from vulnerable users, which can be included in decision-making tools for energy management and other energy-related choices, aiming to enhance digital energy literacy.
IREN	T7.4	Provides requirements for the data gathering components.
SSEA	T7.5	Provides requirements for the data gathering components.
ENEA	T1.4	Provides detailed information on Living Labs' characteristics
ENSIEL	-	WP1 Leader
UPO	-	Project Coordinator

Conclusion

Task 1.2 has established the architectural foundation and integration strategy for the EU-DREAM platform, ensuring a coherent and modular framework that supports the development, interoperability, and deployment of all technical components across Work Packages. The defined architecture integrates core functionalities such as data acquisition, processing, AI services, Digital Twins, and user interaction layers, while maintaining flexibility to accommodate both cloud and edge deployments. In this deliverable were also outlined the interfaces, data exchange mechanisms, and validation pathways that guarantee traceability, integrity, and semantic consistency across modules.

Partner-specific contributions have been identified and aligned with the platform's integration requirements, setting the conditions for a scalable and federated ecosystem. The collaborative effort ensures that each component is designed with architectural compatibility in mind, facilitating its readiness for integration and operational deployment in the Living Labs.

Task 1.2 concludes with a fully defined integration framework that will serve as a reference throughout the subsequent development and demonstration phases of the EU-DREAM project.

References

Acharya, J. N., & Suthar, A. C. (2021). Docker container orchestration management: A review. *International Conference on Intelligent Vision and Computing*, 140-153.

Burns, B., Grant, B., Oppenheimer, D., Brewer, E., & Wilkes, J. (2016). Borg, omega, and kubernetes. *Communications of the ACM, 59*, 50-57.

ISO/IEC/IEEE. (2022, 11). Retrieved from https://www.iso.org/standard/74393.html

Kreps, J., Narkhede, N., & Rao, J. (2011). Kafka: A distributed messaging system for log processing. *Proceedings of the NetDB (Vol. 11)*, 1-7.

Rozanski, N., & Woods, E. (2012). *Software systems architecture: working with stakeholders using viewpoints and perspectives.* Addison-Wesley.

Samarta, M. T., Gunawan, A. A., & Syahputra, M. E. (2024). Systematic Literature Review and Comparative Performance Analysis of SQL and NoSQL Databases in Big Data Applications. *2024 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)* (pp. 218-222). Jakarta: IEEE.