TwinEU

# TwinEU Open Data Space v1

Deliverable No: D4.2
Work package: WP4

Official delivery date: 31.08.2025
Actual delivery date: 28.08.2025

Dissemination level: Public

Co-funded by the European Union

Project: 101136119 | HORIZON-CL5-2023-D3-01 | www.twineu.net

# Disclaimer





| Document Authors | |
|---|---|
| Ferdinando Bosco (ENG)<br>Marcantonio La Franca (ENG)<br>Silvia Clementina Santoro (ENG)<br>Giovanni Di Marco (ENG) | Dimitris Apostolidis (ED)<br>Apostolos Kapetanios (ED)<br>Beyza Cizmeci (Fraunhofer) |



| Version | Date | Author(s) | Notes |
|---|---|---|---|
| 0.1 | 27.06.2025 | Ferdinando Bosco (ENG) | Table of Contents |
| 0.2 | 25.07.2025 | Marcantonio La Franca (ENG)<br>Silvia Clementina Santoro (ENG)<br>Giovanni Di Marco (ENG)<br>Dimitris Apostolidis (ED)<br>Apostolos Kapetanios (ED) | First draft of the document with Ch.3, Ch.4 and Ch.5. |
| 0.3 | 07.08.2025 | Ferdinando Bosco (ENG)<br>Marcantonio La Franca (ENG)<br>Beyza Cizmeci (Fraunhofer) | Second draft of the document with all the contents. |
| 0.4 | 08.08.2025 | Ferdinando Bosco | Document revised and ready for internal review |
| 0.4.1 | 15.08.2025 | Jawana Gabrielski (TUDO)<br>Vasiliki Kontargyri (UoA) | Document revised by internal reviewers |
| 0.4.2 | 28.08.2025 | Ferdinando Bosco (ENG)<br>Apostolos Kapetanios (ED) | Comment and feedback from reviewers addressed. Abbreviations and references added. |
| 0.5 | 28.08.2025 | Ferdinando Bosco (ENG) | Document ready for final quality check and submission. |

| | |
|---|---|
| **Responsible Partner** | ENG |
| **Checked by WP leader** | Apostolos Kapetanios (ED), 28.08.2025 |
| **Verified by the appointed Reviewers** | Jawana Gabrielski (TUDO), 15.08.2025<br>Vasiliki Kontargyri (UoA), 14.08.2025 |
| **Approved by Project Coordinator** | Padraic McKeever (Fraunhofer), 28.08.2025 |

# Table of Contents

# List of Figures

# List of Tables

# List of Abbreviations and Acronyms

| Acronym | Meaning |
|---|---|
| CEEDS | Common European Energy Data Space |
| DAPS | Dynamic Attribute Provisioning Service |
| DERA | Data Exchange Reference Architecture |
| DSO | Distribution System Operator |
| DSP | Data Space Protocol |
| DSSC | Data Space Support Centre |
| DT | Digital Twin |
| EDC | Eclipse Dataspace Connector |
| GUI | Graphical User Interface |
| IATP | Identity and Trust Protocol |
| IDS | International Data Space |
| IDSA | International Data Space Association |
| IMM | Interoperability Maturity Model |
| MQTT | Message Queuing Telemetry Transport |
| PDP | Policy Decision Point |
| PEP | Policy Enforcement Point |
| RAM | Reference Architectural Model |
| RBAC | Role Based Account Control |
| TSG | TNO Security Gateway |
| TSO | Transmission System Operator |

# Executive Summary

The work reported in this deliverable focuses on the design and implementation of the first release of the **TwinEU DT Federator**, a core component of the TwinEU platform enabling **Digital Twin (DT) federation** through a **Data Space-based architecture**. The Federator is conceived to support **interoperable, secure, and sovereign data and model exchange** across heterogeneous energy systems and stakeholders.

The development of the TwinEU DT Federator has led to the creation of a robust and modular infrastructure that enables secure and interoperable data and model exchange across the energy sector. One of the most significant achievements is the extension of the OneNet Data Space Framework, which now includes the **TwinEU Data Space Protocol (DSP) Connector**. This component has been designed to support both traditional pull-based data access and more advanced push-based mechanisms, allowing real-time and event-driven data flows—essential for grid operations and time-sensitive energy services.

The TwinEU DT Federator also introduces the **TwinEU Middleware** and a semantic interoperability layer, enabling participants to describe their data using shared vocabularies and ontologies. This ensures that data exchanged between DTs is not only technically compatible but also semantically meaningful.

To facilitate user interaction, a new **Graphical User Interface (GUI)** was developed. This interface allows users to manage services, subscriptions, and data exchanges intuitively, while also supporting advanced features like **NATS protocol** integration for real-time messaging.

Through the design and implementation of the TwinEU DT Federator, several important insights emerged:

- Federated DT ecosystems require flexible and modular architecture. The ability to integrate heterogeneous systems and evolve with emerging standards is crucial for long-term sustainability.
- Interoperability must be addressed holistically. Technical compatibility alone is not sufficient; semantic alignment, governance frameworks, and legal compliance are equally essential to ensure meaningful and trustworthy collaboration.
- Real-time responsiveness is a critical requirement in energy systems. The integration of push-based data exchange and protocols like NATS demonstrated the importance of low-latency communication for operational efficiency and reliability.
- Data sovereignty and governance are foundational principles. The Federator's mechanisms for access control, identity verification, and policy enforcement reflect the growing need for secure and accountable data sharing in line with European regulations.
- Alignment with European initiatives such as IDSA, GAIA-X, BRIDGE, and DSSC provides a strategic advantage. By leveraging on these frameworks, TwinEU ensures compatibility with broader data space ecosystems and facilitates cross-sectoral integration.
- These lessons not only validate the architectural choices made but also provide a roadmap for future enhancements and broader adoption of federated digital twin technologies across Europe.

The two core components of the TwinEU DT Federator (TwinEU Middleware and TwinEU DSP Connector) aim to be released as open source by the end of the project.

While the TwinEU DSP Connector is already planned to be released as open source by the end of the project and the first version of the source code is already available in the internal project repository, the TwinEU Middleware will be provided in a free-to-use version and the possibility to have an open-source release will be investigated by the end of the project.

# 1 Introduction

In the era of the digital transformation, the energy sector faces unprecedented challenges and opportunities. The integration of distributed energy resources, the rise of prosumers, and the increasing complexity of grid operations demand new paradigms for data exchange, system coordination, and real-time decision-making. Among the most promising innovations is the concept of Digital Twins—virtual representations of physical systems that enable simulation, monitoring, and optimization across the energy value chain.

The true potential of Digital Twins can be unlocked only when they operate in federated ecosystems, where diverse stakeholders—grid operators, service providers, regulators, and consumers—can share data and models in a secure and interoperable way. This vision requires more than technical connectivity; it demands semantic alignment, data sovereignty, and governance frameworks that respect European values and regulations.

The TwinEU DT Federator, a modular infrastructure built upon the OneNet Data Space Framework and aligned with initiatives like IDSA, GAIA-X, BRIDGE, and the Data Space Support Centre (DSSC), can be the solution of the challenges of the DT federation. It introduces key concepts such as semantic interoperability, event-driven data exchange, real-time data streaming, and outlines how the Data Space approach can support a trusted, scalable, and sovereign Digital Twin federation at pan-European level.

## 1.1 Tasks 4.3 and 4.5

Task 4.3 "Coordination with demos and data sharing strategies" aims to establish the bridge between the design and development of the TwinEU framework and its deployment in real-world pilot environments. This task can be divided into several activities, each of them trying to introduce the project stakeholders in every TwinEU concept and software product. Specifically, the T4.3 activities include i) a structured pilot use case analysis which is focused on pan-EU scenarios, ii) presentation and demonstration activities followed by continuous support and feedback so that stakeholders understand the Data Space and Federated Digital Twin concept and iii) the creation of the validation scenarios which will be used by T4.6 for the operational validation of the TwinEU data space. This process is being implemented through the creation of the pan-EU scenarios validation matrix (Chapter 3.3). A core goal of this task is to harmonize the operation of the TwinEU technological enablers with specific activities from a business perspective. To achieve meaningful DT federation and demo sharing strategies, Task 4.3 focuses on defining the connections and interfaces among these diverse DTs, enabling seamless exchange of data and digital models. Practically, T4.3 will guide T4.6 through the established methodology on how the pan-EU scenarios participate within the federated Digital Twin Data space ecosystem.

The outcomes of Task 4.3 are multidimensional. First, we ensure that the TwinEU goal of a federated and interoperable Digital Twin ecosystem is not only conceptually designed and implemented but also practically validated through coordinated demonstrations. Additionally, in collaboration with Task 4.6, Task 4.3 also contributes to the preparation and implementation of pan-European validation scenarios as mentioned above. This operation (which will occur in the next period) will enable validation of the TwinEU solutions achieving technical refinements, and the broader scaling of the TwinEU platform. Moreover, Task 4.3 aligning demo activities with data-sharing strategies, it provides the necessary foundations for achieving interoperability, reliability, and trust in the pan-European Digital Twin federation.

Task 4.5 focuses on the implementation of TwinEU DT Federator, as core part of the integrated TwinEU Platform for DT federation. It aims to:

- Define digital model sharing
- Implement data sharing governance (data access, usage control and traceability)
- Integrate data in harmonized way (semantic, ontology, vocabularies…)
- Extend data space connector for supporting TwinEU features

To satisfy these objectives, the following activities were planned during the whole duration of the task:

- Identification of TwinEU Use Cases and Requirements for Data Space Framework
    - Analysis of TwinEU Use Cases (WP2), as well architectural, functional and non-functional requirements (WP3) related to Data Space
    - Identification of Use Cases for adoption of the Data Space Framework
    - Analysis of Technical Requirements for DT Federation and Data Space
- Definition of Data Governance in DTs Federated environment
    - Data Exchanged (Actors, Data Models, etc.)
    - Data Access and Usage Control
    - Vocabularies
- Evolution and adaptation of the open-source OneNet Data Space Framework (including OneNet Connector)
    - Setup of the TwinEU Data Space Framework (TwinEU DT Federator) environment for TwinEU
    - Releasing of extended version of OneNet Connector (including new features e.g. new GUI, external service integration and blockchain transaction notarization), the TwinEU DSP Connector
    - Implementation of Data Space Protocol
    - Real-time data integration (additional protocols and interfaces)
    - DT models exchange
    - Extended Semantic Interoperability (Data models, standards, ontologies and vocabulary)

The activities include the whole duration of the task. This document reports on the activities, and the results achieved so far. Final results and outcomes will be reported in D4.2 v2 at M33.

## 1.2 Objectives of the Work Reported in this Deliverable

The objectives of the outcomes reported in this document focus on the design, development, and implementation of the first version of the TwinEU Digital Twin (DT) Federator. The TwinEU DT Federator is a central component of the TwinEU platform, based on Data Space, aimed at enabling a federated ecosystem of Digital Twins in the energy sector across Europe, ensuring that diverse DTs from different stakeholders and platforms can interoperate within a standardized framework.

To achieve these objectives a two-step analysis methodology was implemented:

- Step 1: Analysis of Use Cases and Scenarios
    - Based on WP2 outcomes, all relevant data exchanges were identified, categorized, and mapped to energy service categories.
    - Metadata such as data formats, exchange methods (PULL/PUSH), and frequency were documented to ensure interoperability and traceability.
- Step 2: Identification and Prioritization of Requirements
    - Derived from WP2 and WP3, both user and technical requirements were analyzed.
    - A subset of these requirements was prioritized using the MoSCoW method, guiding the implementation roadmap for the DT Federator.

Finally, the first version of TwinEU DT Federator, including the TwinEU Middleware and TwinEU DSP Connector was released. This version will be tested and evaluated within the different pilot and scenario of the TwinEU project.

## 1.3 Outline of the Deliverable

In addition to this chapter, the deliverable includes the following chapters:

- Chapter 2 reports an in-depth analysis about data space initiatives, architectures and related projects.
- Chapter 3 analyses the output of WP2 and WP3such as use cases, scenario and requirements, for designing the TwinEU DT Federator.
- Chapter 4 describes the concept of TwinEU DT Federator, its architecture, components and main functionalities.
- Chapter 5 reports the functionalities implemented so far and the technical specifications about the first release of TwinEU DT Federator with a big focus on the TwinEU DSP Connector.
- Chapter 6 concludes this deliverable.

## 1.4 How to Read this Document

This document is self-consistent and provides a big overview about data space and the first release of TwinEU DT Federator. For completeness, D2.2, D3.1 and D3.2, which describe respectively the use cases, the reference architecture and the functional requirements can provide some additional information.

# 2 Data Space

## 2.1 Introduction and Main Concepts

Data spaces are flexible and open IT infrastructures that enable trustful and transparent use of decentral organised data according to previously defined scopes of use while guaranteeing the full sovereignty of the actors involved. They are based on a federated organisational principle.

Data spaces create equal framework conditions for the sovereign exchange of data. This also means that every actor can benefit from the use of the data in the same way.

The main objectives of the Data Space implementation are:

- Value creation, New data-based applications, business models and collaboration.
- Self-determination, Control over the subject matter, scope, duration and actors in the context of data sharing.
- Efficiency, data exchange takes place exclusively for a specific purpose.

In order to implement the main objectives, the data space implementation leverages on 6 main building characteristics:

- Decentralisation
- Federation
- Interoperability (technical and sematic)
- Sovereignty
- Trust
- Transparency

**Decentralisation** refers to the distribution of control and decision-making across multiple actors rather than relying on a central authority. In data ecosystems, this approach enhances resilience, reduces dependency on single points of failure, and empowers participants to manage their own data and infrastructure. It fosters innovation and autonomy by allowing organizations and individuals to operate independently while still contributing to a shared environment.

**Federation** is essential for enabling collaboration among diverse systems and actors. Federation allows each participant to maintain their own infrastructure and governance while still being part of a larger network.

**Interoperability** ensures that these different systems can communicate and exchange data effectively through common standards and protocols.

In the context of Data Space there are two crucial levels of Interoperability: technical and semantic. Both are essential to ensure that data can be shared, understood, and used effectively across different systems, organizations, and domains.

**Technical Interoperability** refers to the ability of different systems, platforms, and technologies to connect, communicate, and exchange data reliably and securely. It includes standardized APIs and protocols; data formats: authentication and authorization mechanisms.

**Semantic interoperability** ensures the standardization of data exchanged in a common way. It involves common vocabularies and ontologies; metadata standards that describe the context, structure, and meaning of data; data models that define how data is organized and related.

**Sovereignty** ensures that data owners retain full control over their data, including decisions about access, usage, and storage. This principle is closely aligned with legal frameworks such as the GDPR, which emphasize the rights of individuals and organizations over their personal and proprietary data. Sovereignty builds confidence in data sharing by guaranteeing that participants can enforce their own policies and preferences regarding data governance.

**Trust** is the foundation of any successful data ecosystem. It is built through secure technologies, robust control mechanisms, and the use of verifiable digital identities. Trust also depends on the quality and integrity of the data being shared. When participants can rely on the systems and the data, they are more likely to engage in meaningful collaboration. Governance frameworks and accountability structures further reinforce trust by ensuring that all actors adhere to agreed-upon rules and standards.

**Transparency** promotes accountability and ethical behavior by making data-related operations visible and traceable. Through the use of digital identities and audit trails, participants can understand who accessed data, when, and for what purpose. This visibility helps prevent misuse, supports compliance with regulations, and fosters a culture of openness. Transparency is crucial for building trust and ensuring that data ecosystems operate fairly and responsibly.

In the last years, the Data Space concept arose in the research landscape, with very relevant associations and industry clusters pushing for it also from the ICT sector. The next chapters report and analyse the most relevant initiatives and projects at European level.

## 2.2 Main Initiatives and Reference Architectures

### 2.2.1 International Data Space Association (IDSA)

The International Data Spaces Association (IDSA) [1] is a non-profit organization focusing on establishing and promoting standards for data spaces creating trusted environments where organizations can share data while retaining full control over its use. The IDSA achievements establish the foundation for a fair data economy, setting clear rules for trusted data sharing.

IDSA has developed a comprehensive Reference Architecture Model (RAM) [2] to guide the design and implementation of data spaces.

At its core, International Data Space (IDS) functions as a Virtual Data Space leveraging established technological standards and governance models. It aspires to facilitate secure data exchange within a trusted Business Ecosystem. By doing so, IDS becomes the backbone for developing Smart Service Business Processes while empowering participants to unlock the full value of their data. IDSA drives this vision by articulating and implementing a new reference architecture that guarantees security, sovereignty, and trust. This RAM interconnects data clouds, enterprises, business applications, and Internet of Things (IoT) devices, establishing a cohesive IDS ecosystem. The key technical building block in this ecosystem is the IDS Connector, a dedicated software component (described in Ch. 2.2.1.4), functioning as an intermediary between participating entities.

#### 2.2.1.1 Key Features, requirements, and principles

The IDS prides itself on several distinctive features:

- **Sovereignty of Data Assets**: Owners retain control by setting individualized usage policies governing data access and utilization.
- **Security of Data Exchange**: A robust protection layer ensures data security, particularly during its exchange.
- **Decentralized Organization and Federated Architecture**: IDS amalgamates all endpoints utilizing an IDS connector, eliminating the necessity for centralized data management or governance. This approach presents an alternative architectural design compared to both centralized data management concepts and decentralized data networks lacking shared rules.
- **Governance and Shared Rules**: In the absence of a central supervisory authority, data governance principles evolve as shared rules, delineating rights and obligations for effective data management, derived from user requirements.
- **Network of Platforms and Services**: IDS acts as a bridge connecting data providers and users. Entities ranging from companies to individual IoT entities like vehicles, machines, and equipment can function as data providers.
- **Scaling and Network Effects**: IDS facilitates data services for secure exchange and seamless data linking. Its decentralized nature enables scalability without reliance on central authority, fostering network effects through increased data availability across diverse ecosystems.
- **Openness**: Driven by user input and participative development, the International Data Spaces initiative operates under the umbrella of the IDSA.
- **Trust Protection**: Participants in IDS must rely on the identity verification of data providers and users, ensured through mandatory software certification. IDS also offers specialized connectors featuring extended encryption for secure data exchange.

For the IDS to succeed, specific strategic requirements need fulfilment:

- **Trust**: Trust forms the bedrock of the ecosystem, mandating evaluation, and certification of all participating entities by trusted authorities.
- **Security and Data Sovereignty**: Rigorous evaluation and certification of technical components ensure security. Data owners dictate usage restrictions for data transfers.
- **Ecosystem of Data**: IDS's decentralization approach negates the need for centralized data storage, ensuring that data remains with respective owners until transferred to trusted third parties.
- **Comprehensive Data Source Description**: Thorough descriptions of data sources are mandated to maintain clarity and transparency.
- **Integration of Domain-Specific Data Vocabularies**: Usage of domain-specific vocabularies aids in data annotation and description.
- **Real-time Data Search Services**: Facilitation of quick and efficient data discovery and retrieval.
- **Standardized Interoperability**: IDS connectors act as technological components ensuring seamless communication and interaction.
- **Value-Adding Apps**: Integration of IDS connectors with business applications offers a myriad of services atop data exchange processes, including data processing, format alignment, exchange protocols, and analytics.

- **Data Markets**: IDS enables the provision of clearing mechanisms, billing functions, domain-specific broker solutions, and marketplaces, along with information regarding usage restrictions and legal aspects.

The IDSA steers its goals guided by specific principles:

- **Open Development Process**: The IDSA operates as a non-profit organization open to all participants adhering to common work principles.
- **Re-Use of Existing Technologies**: Emphasis is laid on leveraging and reusing established technologies and standards wherever feasible.
- **Contribution to Standardization**: IDSA aims to establish an international standard by supporting standardized architecture stacks.

#### 2.2.1.2 IDS RAM

At the core of IDSA's mission lies the RAM - a meticulously designed framework that outlines the structure, interactions, and functionalities within the International Data Spaces ecosystem. The RAM [2], now at version 4, serves as a guiding blueprint, offering a systematic approach to designing and implementing secure data spaces, ensuring compatibility, interoperability, and trust among diverse stakeholders.

In compliance with common system architecture models and standards (e.g., ISO 42010, 4+1 view model), the Reference Architecture Model uses a five-layer structure expressing various stakeholders' concerns and viewpoints at different levels of granularity as illustrated in Figure 1.

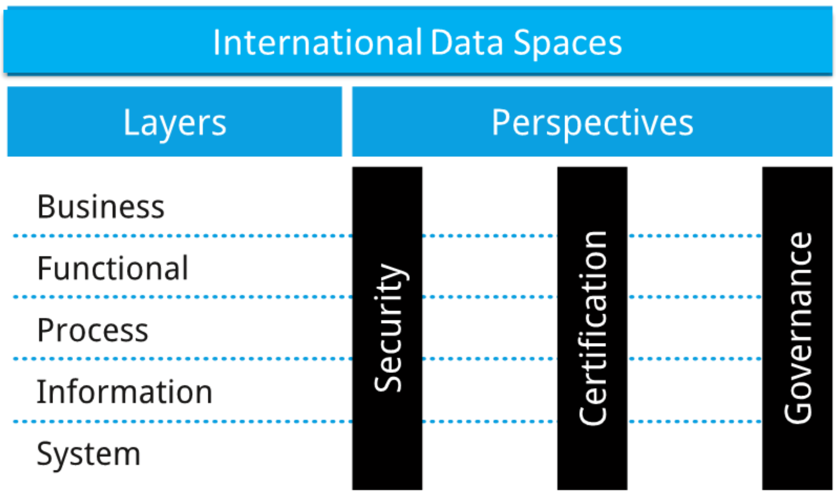



*Figure 1: IDS RAM 4 [3]*

The **Business Layer** specifies and categorizes the different roles which the participants of the International Data Spaces can assume, and it specifies the main activities and interactions connected with each of these roles.

The **Functional Layer** defines the functional requirements of International Data Spaces, plus the concrete features to be derived from these.

The **Information Layer** defines a conceptual model which makes use of linked-data principles for describing both the static and the dynamic aspects of the International Data Space's constituents.

The **Process Layer** specifies the interactions taking place between the different components of the International Data Spaces; using the BPMN notation, it provides a dynamic view of the Reference Architecture Model.

The **System Layer** is concerned with the decomposition of the logical software components, considering aspects such as integration, configuration, deployment, and extensibility of these components.

In addition, the Reference Architecture Model comprises three perspectives that need to be implemented across all five layers:

- Security
- Certification
- Governance

#### 2.2.1.3 IDS Components

The technical components of the IDS RAM are shown in Figure 2.

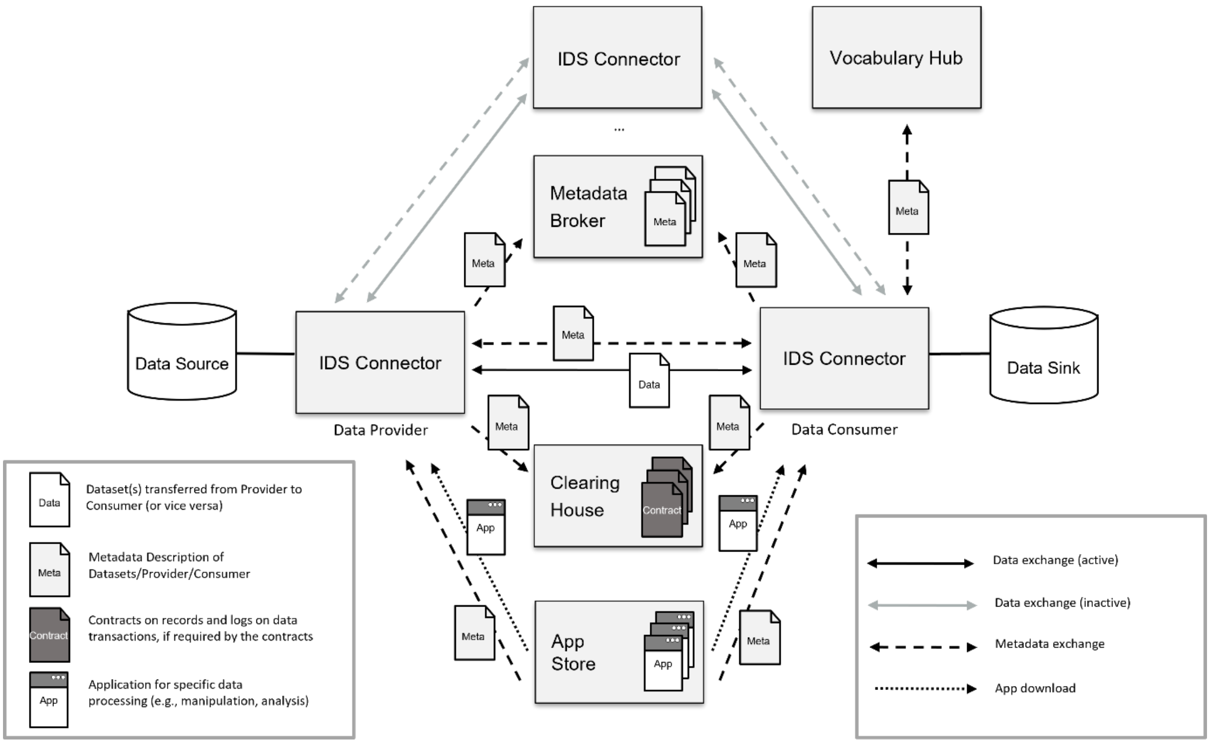



*Figure 2: IDS RAM 4 components [4]*

The IDS Reference Architecture consists of the following core components:

- the Identity Provider
- the IDS Connector
- the App Store and Data Apps
- the Metadata Broker
- the Clearing House
- the Vocabulary Hub

A distributed network like the International Data Spaces relies on the connection of different participants where IDS Connectors or other core components are hosted (an IDS Connector comprising one or more Data Endpoints). The IDS Connector is responsible for initiating a data exchange from and to the internal data resources and enterprise systems of the participating organizations and the International Data Spaces. It provides metadata to the Metadata Broker as specified in the IDS Connector self-description, e.g., technical interface description, authentication mechanism, and associated data usage policies. Usage Contracts can be transferred via the IDS Connector to the Clearing House to ensure trust. Also, the data transfer can be logged at the Clearing House for trust reasons, or for clearing reasons. Vocabularies can be interpreted by getting more details from the Vocabulary Hub. Additional IDS Apps can be downloaded to the IDS Connector to run operations on the data.

#### 2.2.1.4 IDS Connector

The International Data Spaces network is constituted by the total of its IDS Connectors. Each IDS Connector allows the exchange of data via the Data Endpoints it exposes. Applying this principle, there is no need for a central instance for data storage. An IDS Connector must be reachable by IDS connectors from other organisations. Due to organizational security policies, this may require changing network policies. It should be possible to reach an IDS Connector using standardized protocols, and to operate it in any appropriate environment. A Participant may operate multiple IDS Connectors (e.g., to meet load balancing or data partitioning requirements). IDS Connectors can be operated on-premises or in a cloud environment. A dedicated section about interoperability of Connectors and Data Space Protocol is provided on Ch.2.2.1.5.

The Connector provides metadata to the Broker, which includes description of the technical interface, authentication mechanism, exposed data sources, and associated data usage policies. Data is transferred between the Connectors of the Data Provider and the Data Consumer (peer-to-peer network concept).

**Connector Architecture**

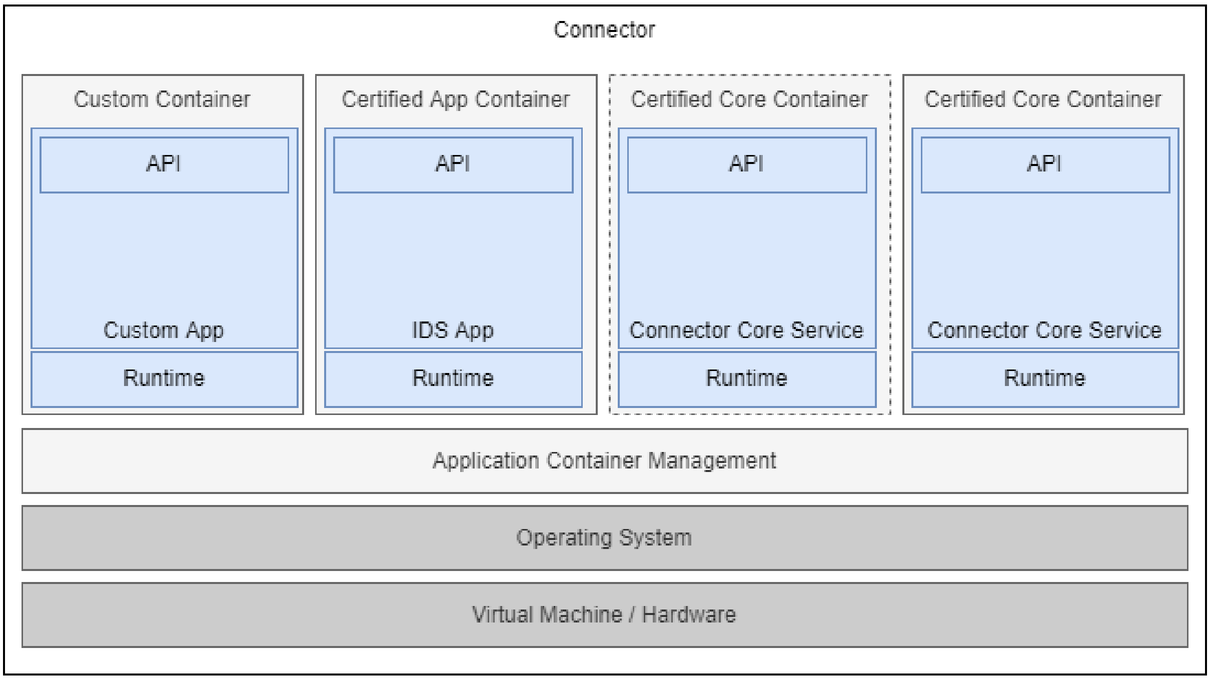



*Figure 3: IDS Connector Architecture [5]*

The Connector Reference Architecture shown uses application container management technology to ensure an isolated and secure environment for individual data services which offer an API to store, access or process data.

#### 2.2.1.5 Data Space Protocol

While the IDS Connector is a strong reference implementation that embodies the principles of the Data Space Protocol and serves organizations that want to implement a ready-to-use solution, in the ecosystem of Data Space, it is important to allow different implementations to interoperate without needing identical architecture.

In this context, the Dataspace Protocol (DSP) [6] defines how data space connectors communicate within a data space. It covers essential functions like data catalog access, contract negotiation, and data transfer management, all built on proven HTTPs technologies and fully RESTful.

The DSP has been designed with huge flexibility allowing various authorization mechanisms and multi-tenant deployments, as well as the possibility to explain how the information is provided within the data exchange.

It leverages on three main pillars:

**Data Catalogue**. It outlines how datasets are presented into a Catalogue and how rules about data usage are expressed.

**Contract Negotiation**. It defines how agreements between providers and consumers governing data usage are written and negotiated electronically.

**Data Transfer**. It identifies how datasets are accessed using specific protocols.

The interaction of Participants in a Dataspace is conducted by the Participant Agents, so-called **Connectors**, which implement the protocols described above. While most interactions take place between Connectors, some interactions with other systems are required. Figure 4 below provides an overview of the context of this specification.

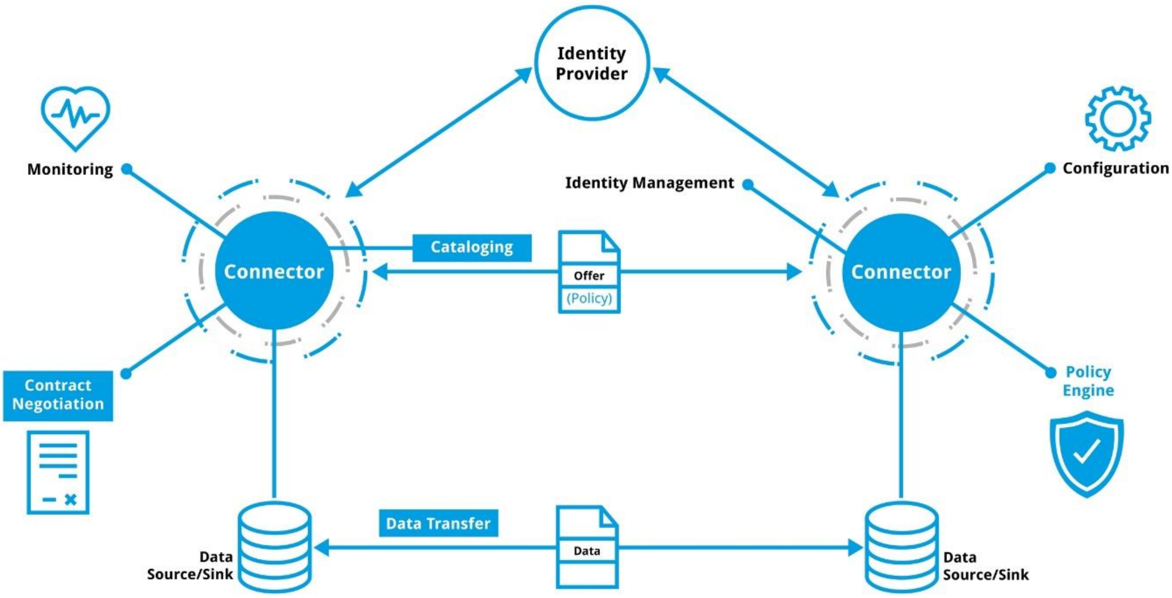



*Figure 4: Data Space Protocol Architecture [6]*

An Identity Provider realizes the required interfaces and provides required information to implement the Trust Framework of Dataspace. The validation of the identity of a given Participant Agent and the validation of additional claims is a fundamental mechanism. The structure and content of such claims and identities may, however, vary between different Dataspaces, as well as the structure of such an Identity Provider, e.g. a centralized system, a decentralized system or a federated system. Other specifications, like the Identity and Trust Protocol (IATP), define the respective functions.

## 2.2.2 GAIA-X

Gaia-X [7] is a European initiative, launched in 2019, designed to establish a federated and secure digital infrastructure that promotes data sovereignty, transparency, and interoperability. At its core, Gaia-X envisions a decentralized ecosystem where participants—ranging from cloud providers to data consumers—can interact under a shared governance model. This model is not a single platform but a framework that enables multiple federations to coexist and interoperate.

The Gaia-X architecture [8] as shown in Figure 5 below, is structured around three conceptual layers: the **Trust Plane, the Management Plane, and the Usage Plane.**

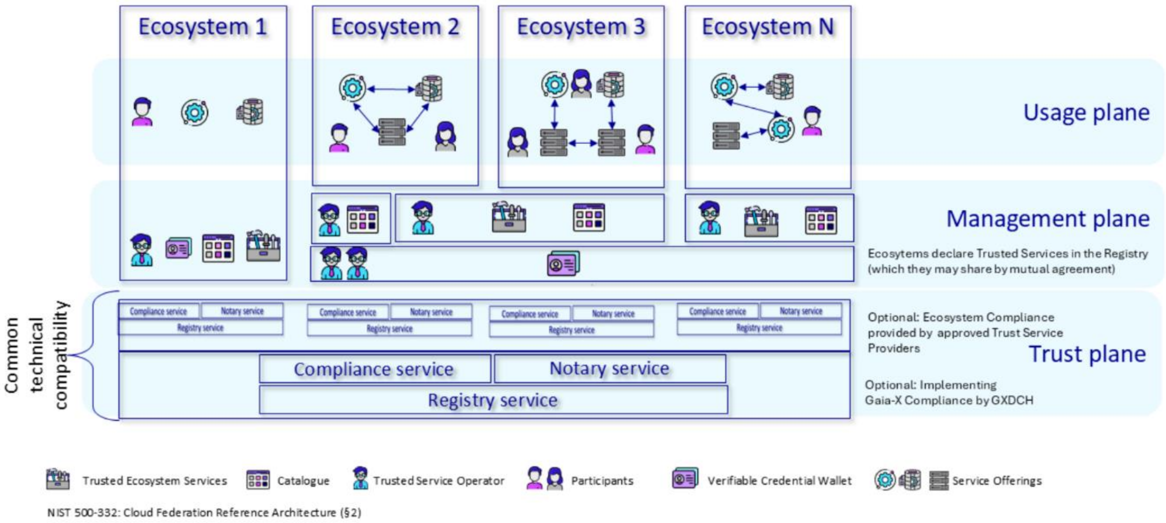



*Figure 5: Gaia-X Reference Architecture [8]*

These layers are inspired by the NIST Cloud Federation Reference Architecture [9] and adapted to Gaia-X's goals. The Trust Plane governs the global rules and compliance mechanisms, the Management Plane allows for domain-specific extensions, and the Usage Plane ensures technical interoperability among services and data products.

A central element of this architecture is the Federation of Ecosystems, where each ecosystem can define its own governance while remaining interoperable with others. This is achieved through standardized credentials, schemas, and policy frameworks that are universally applicable across sectors and regions.

### 2.2.2.1 Gaia-X Trust Framework

The Trust Framework is the backbone of Gaia-X's architecture. It ensures that all participants in the ecosystem can trust each other through verifiable credentials, policy enforcement, and transparent governance. This framework is designed to support automated onboarding and offboarding of services and participants, making it scalable and adaptable.

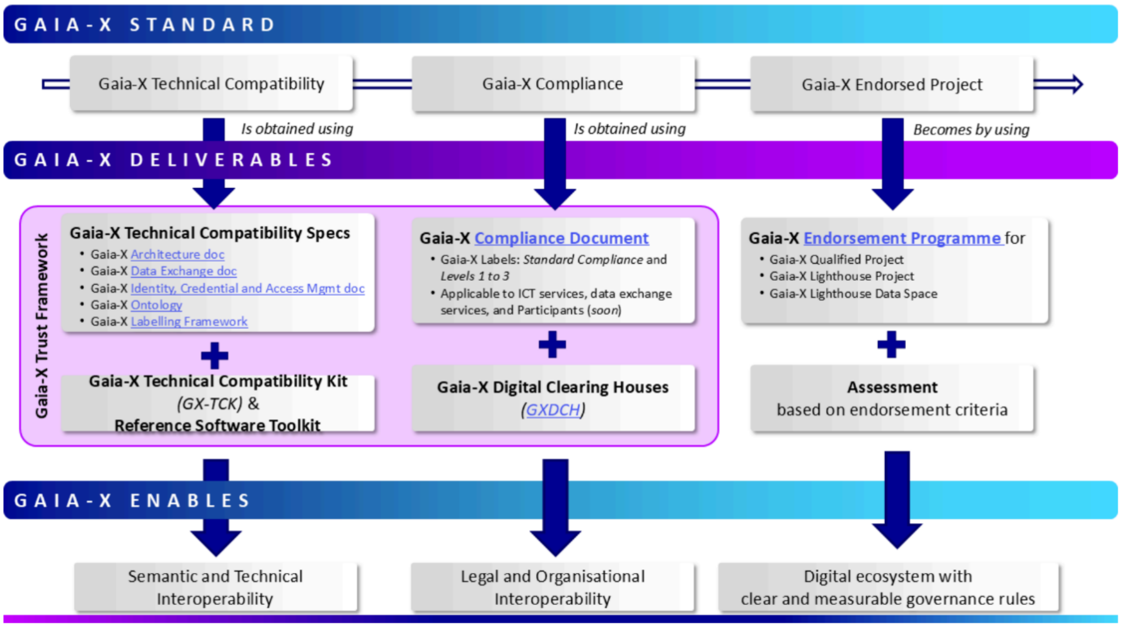



*Figure 6: Gaia-X Trust Framework [8]*

At the core of the Trust Framework are Gaia-X Credentials, which encapsulate identity, service attributes, and compliance proofs. These credentials are issued and verified using OpenID Connect protocols, ensuring secure and standardized interactions. The framework also includes a Policy Decision Point (PDP), which evaluates access and usage policies dynamically, allowing for fine-grained control over data and service interactions.

The Trust Framework is not limited to technical compatibility; it also enforces compliance with legal and ethical standards. This dual approach—technical and regulatory—ensures that Gaia-X ecosystems are not only interoperable but also trustworthy and aligned with European values.

Main building blocks and key elements of the Gaia-X Framework are:

- **Verifiable Credentials**: Used for identity, service attributes, and compliance proofs.
- **Linked Data and Ontologies**: Enable semantic interoperability.
- **Gaia-X Schema**: Standardized data model for describing services and participants.
- **OpenID Connect Extensions**: For secure credential issuance and presentation.
- **GXDCH (Gaia-X Digital Clearing House)**: A reference implementation for federation services.
- **Data Usage Agreements (DUAs)**: Define how data can be used, shared, and monetized.
- **Trust Anchors**: Entities that validate and certify compliance within the ecosystem.

### 2.2.3 BRIDGE

BRIDGE [10] was introduced by the European Commission in 2016 as an EU initiative aiming to enhance cooperation and consolidate findings across research & innovation projects centred on smart grids, energy storage, islands, and digitalization of the energy system funded under the Horizon 2020 program. This initiative operates through four primary Working Groups—Data Management, Business

Model, Regulation, and Customer Engagement—which compile reports and propose recommendations to the European Commission regarding various themes pertinent to the future of the energy sector.

In April 2021, the Data Management Working Group, focusing on both technical and non-technical aspects concerning stakeholder data exchange handling, released a report outlining an EU data exchange reference architecture [11].

Multiple BRIDGE projects contributed to this report. The reference architecture, aligned with the Smart Grid Architecture Model (SGAM) [12], defines foundational elements essential for inter-domain exchanges. Its purpose is to facilitate cross-domain and cross-border interoperability, empowering existing and future projects in this pursuit.

The escalating growth of distributed renewable energy generation and energy storage systems, coupled with an anticipated surge in active customer engagement in demand response and electric mobility, presents several challenges to prevailing planning and operational methodologies of system operators. An imperative aspect in transitioning towards an energy shift is integrating demand flexibility services from these new assets and players into the energy market. This incorporation is crucial for addressing technical issues and ensuring resilience, efficiency, and reliability in modern electricity grids. This evolution anticipates bringing flexibility products—inclusive of those from residential consumers—to the forefront of system operation, thereby facilitating market adoption.

Both traditional retail procedures and emerging flexibility services demand accessible data and information exchange among a diverse array of stakeholders, networks, systems, devices, applications, and components. The cooperation and data exchange among existing and new systems are vital to enable present, evolving, and future energy services. Establishing a common European reference architecture serves as a pivotal driver toward committing to demand flexibility, allowing utility coordination across national borders and diminishing market entry barriers.

Designating interoperable data exchange solutions forms a cornerstone in defining a reference architecture. Diverse solutions—bilateral point-to-point arrangements, data hubs/warehouses, and data exchange platforms—exist alongside each other. The interoperability of these solutions necessitates comparability, appropriate standardization, and governance. Apart from decentralized solutions, platform-oriented solutions (hubs, DEPs) have recently emerged across Europe to streamline processes, enhance data quality, and ensure minimal delay, primarily in retail markets. Recent initiatives, guided by the Clean Energy Package and other relevant directives, emphasize the active inclusion of end-users in both retail and wholesale energy markets, intensifying the demand for data exchange among all stakeholders. An integrated data exchange architecture is pivotal in the global energy system, signalling a convergence of wholesale and retail markets.

#### 2.2.3.1 Bridge DERA (Data Exchange Reference Architecture)

In July 2023, the third version of Data Exchange Reference Architecture – DERA 3.0 was released by BRIDGE Data Management Working Group [13]. It represents a big step forward in the alignment with Data Space ecosystem.

Similarly to the previous versions, DERA 3.0 is a modular architecture composed of five layers: Communications, Components, Information, Functions, and Business. It emphasized the need for standardized protocols and formats to enable seamless data exchange across energy systems.

The major innovation in 3.0 was the distinction between local platforms and federated data space stacks. Local platforms referred to existing data infrastructures, while federated stacks enabled data discoverability and trading through a Data Space Connector.

DERA 3.0 also began integrating data governance across its layers, recognizing the importance of harmonized policies and metadata for cross-sector interoperability. However, it was still considered a work in progress, with recommendations to enhance semantic interoperability and governance mechanisms.

In October 2024, a minor update of the architecture was released as DERA 3.1. This last version represents a significant step forward in aligning energy data exchange with the broader European vision of interoperable, sovereign, and federated data spaces. The architecture is designed to support cross-domain interoperability, enabling energy data to be shared securely and efficiently across sectors such as mobility, manufacturing, and smart cities.

**1. Stronger Alignment with European Data Space Initiatives**

DERA 3.1 explicitly integrates building blocks from OpenDEI and Gaia-X, ensuring compatibility with broader European data space frameworks. This includes support for verifiable credentials, federated identity management, and data usage policies, making it easier to connect energy data spaces with other domains like mobility and manufacturing.

**2. Enhanced Governance and Compliance**

While DERA 3.0 introduced governance concepts, version 3.1 formalizes them. It includes mechanisms for policy enforcement, compliance verification, and auditability, enabling trust and accountability across federated ecosystems. These features are essential for regulatory alignment and for building confidence among stakeholders.

**3. Semantic and Technical Interoperability**

DERA 3.1 deepens its commitment to semantic interoperability by adopting linked data principles, ontologies, and standardized metadata schemas. This ensures that data exchanged across platforms is not only syntactically compatible but also semantically meaningful, facilitating automated processing and integration.

**4. Improved Data Space Connector Functionality**

The Data Space Connector, introduced in 3.0, is further refined in 3.1 to support dynamic data discovery, policy negotiation, and real-time data exchange. It acts as a bridge between local platforms and federated data spaces, enabling scalable and secure data flows.

**5. Cross-Sector and Cross-Border Integration**

DERA 3.1 expands its scope to support cross-sectoral and cross-border data exchange. It encourages the creation of interoperable data ecosystems that can span national boundaries and industry verticals, aligning with the EU's vision for a unified digital market.

One of the key advancements in DERA 3.1 is its explicit alignment with the **OpenDEI Data Space building blocks** [14]. As shown in Figure 7, these include components for identity management, data governance, semantic interoperability, and trust frameworks. By incorporating these elements, DERA 3.1 ensures that energy data exchange is not only technically compatible but also compliant with the European Data Strategy and Gaia-X principles.

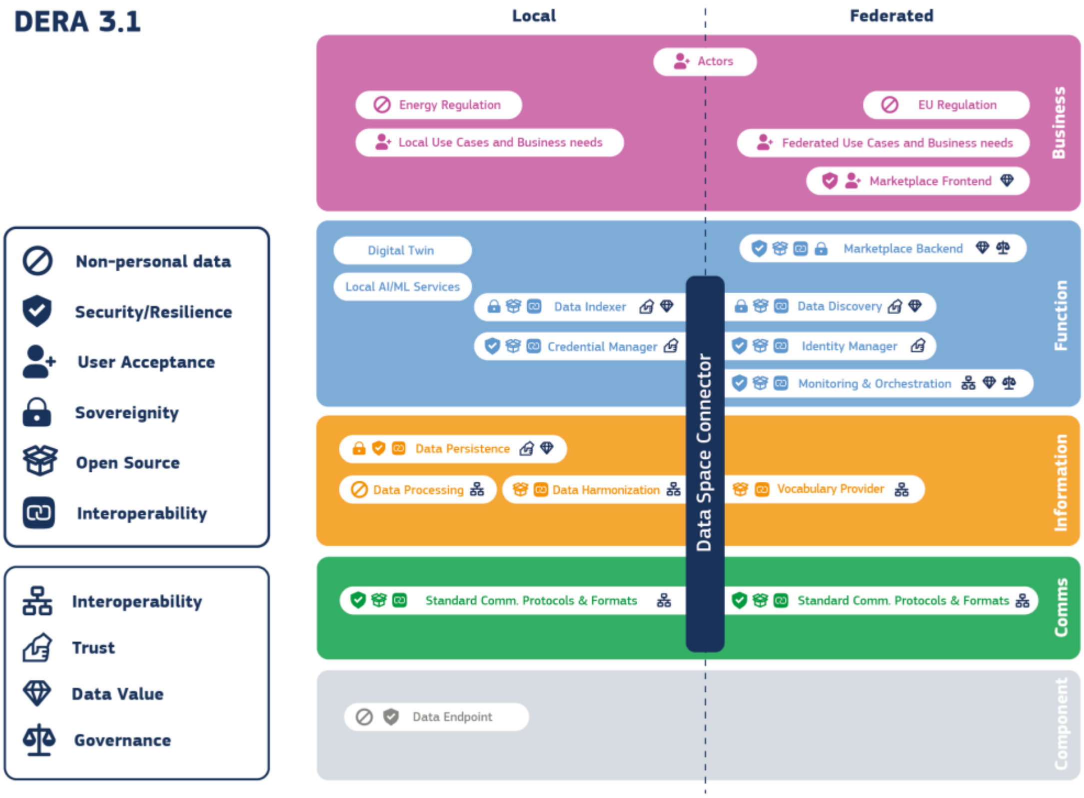



*Figure 7: BRIDGE DERA 3.1 [14]*

The architecture clearly insists on the concept of Federated Data Space (Local and Federated layers) and in the Data Space Connector as key technological enabler of the trusted and secure data exchange.

The Data Space Connector, as transversal element, allows different IT systems/platforms and data using applications to connect and share data with each other. This can be useful for integrating data from different sources, or for allowing multiple applications to access the same data without having to duplicate it in multiple places.

The local side of the architecture refers to any kind of data platform that is locally deployed and can be integrated via data space (e.g., the data platform, energy community platform, IoT platform, Flexibility Register etc.). These platforms are already capturing and persisting their own data, which is usually fed into local services for tailored applications. The data space connector should be incorporated into these platforms to enable identification, data harmonization and integration towards data space.

The federated data space part of the architecture refers to where data is indexed, making it discoverable and providing a sort of marketplace for trading both data and data services. In order to do so, Data Space will rely on multiple actors and data platforms (the previously described ones) federating through the Data Space Connectors and offering their data under pre-recorded policies.

DERA 3.1 emphasizes the use of standardized APIs, semantic models, and metadata schemas to facilitate seamless data sharing. It also supports data usage policies and access control mechanisms, which are essential for maintaining data sovereignty and trust among participants.

As already described, DERA 3.1 architecture promotes a federated model, where multiple stakeholders—such as grid operators, energy providers, and consumers—can participate in data exchange without relying on a central authority. This is achieved through a combination of distributed registries, verifiable credentials, and policy enforcement engines.

DERA 3.1 also introduces mechanisms for data traceability, auditability, and compliance verification, which are crucial for building trust in energy data ecosystems. These features are particularly important for enabling real-time data exchange and automated decision-making in smart grid environments.

### 2.2.4 DSSC - Data Space Support Centre

The Data Space Support Centre (DSSC), established in 2022 under the Digital Europe Programme, acts as a central hub for advancing the European data-space vision. Its mission is to provide technical guidelines, legal frameworks and best practices to help communities design and operate data spaces that comply with data-sovereignty principles [15]. An early deliverable of DSSC is the Data Spaces Blueprint, which defines a conceptual model, a set of organizational and technical building blocks, and a common vocabulary to harmonize data-space developments. The blueprint emphasizes that data spaces enable secure and trusted data transactions, support data sovereignty, and foster interoperability and standardization across sectors. It provides the foundation for cross-domain projects like GAIA-X, IDSA and BRIDGE to converge towards common architectures.

#### 2.2.4.1 Objectives and Blueprint Development

A primary goal of the DSSC is to develop common standards and architectural building blocks for data spaces in collaboration with all relevant stakeholders. Rather than reinventing the wheel, the DSSC analyses existing solutions and best practices from initiatives like IDSA and GAIA-X, integrates what works, and identifies gaps where new solutions or standards are needed [16]. The outcome of this work is captured in the DSSC Blueprint [17], an evolving reference architecture for data spaces. This blueprint is built on a set of modular building blocks covering technical, semantic, legal, operational, and business aspects of data sharing ecosystems. Notably, the DSSC built on the 12 design principles (building blocks) defined by the earlier OPEN DEI project, expanding them (especially the business and organizational blocks) and providing detailed specifications for implementing data space components [18]. In this way, the Blueprint offers a toolkit of architecture components and design patterns that any sectoral data space can adopt to ensure compatibility with the European common framework.

DSSC also publishes a Data Spaces Glossary to harmonize concepts and terminology across initiatives, ensuring that interoperability is supported not just by code but by a common language and understanding among stakeholders [19].

DSSC actively engages with domain-specific data space pilots (in energy, manufacturing, mobility, etc.), gathers their requirements, and feeds back common needs into the development of standards. The DSSC Glossary and its working groups on architecture are results of co-creation with experts across domains. The DSSC also works closely with policy bodies to ensure alignment with evolving regulations (for instance, the European Data Act, or sectoral legislation for data sharing). In summary, the DSSC plays a pivotal role in Europe’s data space ecosystem by providing the reference architecture, common building blocks, and coordination required to establish a network of decentralized, interoperable data spaces.

Beyond the reference architecture documentation, the DSSC is establishing a support platform (web portal) as a one-stop shop for data space initiatives. This includes a knowledge base with best practices, templates, and how-to guides for designing data spaces according to Blueprint. It also features a help desk or advisory service to assist new data space projects in interpreting requirements or troubleshooting interoperability issues. Furthermore, the DSSC curates various toolboxes. For example, collections of open-source software components that implement Blueprint building blocks. These could range from reference implementations of an IDS/GAIA-X compatible connector to tools for semantic mapping (like SAREF or ECLASS vocabularies in energy), to legal templates for data-sharing agreements. By providing these resources, the DSSC lowers the barrier to entry: even organizations with less experience in data sharing can quickly boost a data space node that is compliant and interoperable.

#### 2.2.4.2 Key Architectural Building Blocks

The blueprint organizes building blocks into three layers: Governance, Data Space Services, and Infrastructure by reflecting the layered approach adopted by the IDSA and GAIA-X. Within these layers, several building blocks are particularly relevant to TwinEU DT federation:

1. **Data Space Connector**: The connector is the gateway through which participants join a data space. It enables participants to publish and consume data while enforcing usage policies, identity management and data-sovereignty rules. The DSSC blueprint promotes a federated connector architecture where each participant runs their own connector instance, thus avoiding centralized control. A connector comprises:

    a. Data Provider/Consumer services: interfaces to internal data sources and applications. A connector can implement transformation services to expose data as standardized APIs or as NGSI-LD context information for IoT/energy data.

    b. Policy Enforcement Point (PEP) and Usage Control: modules that enforce data sharing agreements using contracts and policies, ensuring that data is only used for authorized purposes.

    c. Identity and Trust: integration with identity providers, trust registries and certification authorities to verify the identity of participants and ensure compliance with trust frameworks (e.g., GAIA-X self-descriptions).

    d. Transaction Logging: secure logging of data exchanges, often using distributed ledgers for notarization and audit.

    Different connector implementations exist. The IDS connector uses the IDS Reference Architecture Model and emphasizes trust, usage control and metadata exchange (Ch. 2.2.1.4). The Eclipse Data Space Components (EDC) connector, adopted by projects such as OMEGA-X, is a modular, open-source implementation aligned with GAIA-X and IDSA; it supports plug-and-play integration, policy management and data exchange across heterogeneous platforms [20]. Both connectors implement standardized protocols (IDSCP, AS4) and can be extended with NGSI-LD adapters for smart grid data.

2. **Interoperability Services:** Data spaces must connect heterogeneous data sources and systems. The DSSC blueprint recommends services for semantic and syntactic interoperability:

    a. Common Vocabulary and Ontology: the blueprint includes a glossary and encourages using sector-specific ontologies (e.g., CIM for power systems, NGSI-LD for context information) to facilitate machine-readable semantics.

   b. Data Harmonization and Transformation: connectors or mediators transform local formats into standard representations. For energy data, this may involve mapping measurement data to CIM or ENTSO-E transparency formats.

   c. Discovery and Catalogue: building blocks for indexing datasets, services and metadata. The Data Space Connector Consumer architecture building block in the Interoperable Europe portal describes an application service that gathers data over available sources, checks the content and structure quality, and enables technical interoperability through discovering and connecting to data. The DSSC blueprint extends this concept with a federated catalogue where each data space publishes metadata to facilitate cross-space discovery.

3. **Identity and Trust Framework**: The blueprint reuses GAIA-X trust services and the IDSA certification scheme to ensure that only accredited participants can join the data space. Trust anchors maintain self-descriptions that record participants' credentials, privacy policies and compliance with regulations. A federated identity management system allows single sign-on and cross-domain authentication.
4. **Governance and Compliance**: The DSSC emphasizes governance processes that define roles (data provider, consumer, mediator, operator), business models, data licenses and dispute resolution. Compliance with the EU Data Governance Act and other regulations (GDPR, NIS2) is integrated into the blueprint.

#### 2.2.4.3 Decentralized/Federated Architecture

The blueprint advocates for a decentralized architecture where data remains with the owner and is accessed through connectors rather than copied into central repositories. Federated services such as catalogues, identity providers and monitoring can be hosted by different organizations, enabling federation of data spaces across domains. For instance, the energy data space described by the InterSTORE and OMEGA-X projects uses a data space middleware customized from OneNet's decentralized middleware; it implements identity management, catalogue/discovery, vocabularies and data access management, and includes an IDS-based connector combined with a FIWARE context broker (NGSI-LD) for real-time energy data [31]. This example demonstrates how the DSSC blueprint can be instantiated for the energy domain by combining IDS connectors with FIWARE and blockchain notarization to ensure trust.

#### 2.2.4.4 Alignment with Existing Initiatives and Relevance to TwinEU

The DSSC blueprint is designed to be compatible with, and build upon, existing initiatives such as IDSA, GAIA-X and BRIDGE:

- **IDSA** provides the IDS Reference Architecture Model (RAM 3.0), which specifies connectors, metadata, and trust components. The DSSC blueprint adopts many elements of IDS RAM, including the separation between control plane (negotiation, policy management, logging) and data plane (data transfer), and reuses the IDS certification scheme.
- **GAIA-X** defines a federated infrastructure and trust framework for cloud providers. The DSSC blueprint reuses GAIA-X self-descriptions and identity federation to enable cross-cloud interoperability. It encourages implementing connectors on sovereign cloud infrastructures compliant with GAIA-X to guarantee transparency and data protection.
- **BRIDGE** is an EC initiative that coordinates demonstration projects in the energy sector. BRIDGE working groups have defined reference architectures for data exchange in electricity

networks; these architectures emphasize decentralized and hierarchical grid-control systems. The DSSC blueprint enables BRIDGE projects to align with cross-sector data-space architecture by adding connectors and federated governance on top of existing energy platforms.

TwinEU can leverage the DSSC blueprint as a guideline for designing its data space:

- **Connector-centric integration:** Each digital twin provider will operate its own data space connector to expose twin data (e.g., grid models, real-time measurements, simulation results) while enforcing usage policies. The connectors will support standard interfaces (NGSI-LD, CIM) and policy enforcement to ensure that data is consumed only for agreed purposes.
- **Semantic interoperability:** The DSSC emphasizes harmonized vocabularies. *TwinEU architecture adopts the Common Information Model (IEC 61970/61968), NGSI-LD, and possibly the Digital Twin Definition Language (DTDL) to represent grid assets, states and events.*
- **Federated catalogue and marketplace:** A distributed catalogue will allow participants to advertise digital twin data and simulation services. Using the Data Space Connector Consumer concept, digital twins can discover and connect relevant data from other participants.
- **Trust and compliance:** Identity and trust services will be aligned with GAIA-X to ensure only certified operators can access digital twins. Usage control and logging will guarantee accountability and facilitate auditing.

#### 2.2.4.5 Challenges and Ongoing Developments

Although the DSSC blueprint provides a robust foundation, several challenges remain:

- **Interoperability across multiple connectors and standards**: Energy projects use different connector implementations (IDS, EDC, FIWARE DSBA components). Harmonizing connectors and achieving plug-and-play interoperability is an active research area.
- **Cross-domain interoperability:** Achieving interoperability between data spaces from different domains is critical for the European data economy. The DSSC blueprint has to evolve to include cross-domain mapping and common governance guidelines.
- **Performance and scalability:** Decentralized architectures may face latency issues when retrieving data from distributed connectors. Caching, edge computing and data summarization strategies will be needed for real-time DT operations.
- **Governance and business models**: Data spaces require sustainable business models to incentivize participation. The blueprint mentions generic roles but does not prescribe specific revenue models.

## 2.3 Relevant Projects

### 2.3.1 OneNet

OneNet was a project co-funded by the European Commission under the Horizon 2020 program. It ended in 2023, and the key achievements of the project have been:

- Definition of a common market design for Europe: this means standardized products and key parameters for grid services which aim at the coordination of all actors, from grid operators to customers.

- Definition of a Common IT Architecture and Common IT Interfaces: this means not trying to create a single IT platform for all the products but enabling an open architecture of interactions among several platforms so that anybody can join any market across Europe.
- Large-scale demonstrators that implemented and validated the scalable solutions developed throughout the project. These demonstrators were organized in four clusters coming to include countries in every region of Europe and tested innovative use cases never validated before.

In the context of data-driven architecture, the OneNet Reference Architecture represented a very innovative solution that leverage on IDS reference model and FIWARE interfaces, bring to a hybrid solution using both the standard models for implementing the OneNet Decentralized middleware and the OneNet Connector. The usage of IDS Connector and FIWARE Context Broker ensured a high level of standardization, interoperability, scalability and reuse of OneNet solution.

#### 2.3.1.1 OneNet Reference Architecture

The OneNet Reference Architecture [21], shown in Figure 8, consists of three logical layers:

- OneNet Participants
- OneNet Network of Platforms
- OneNet Framework

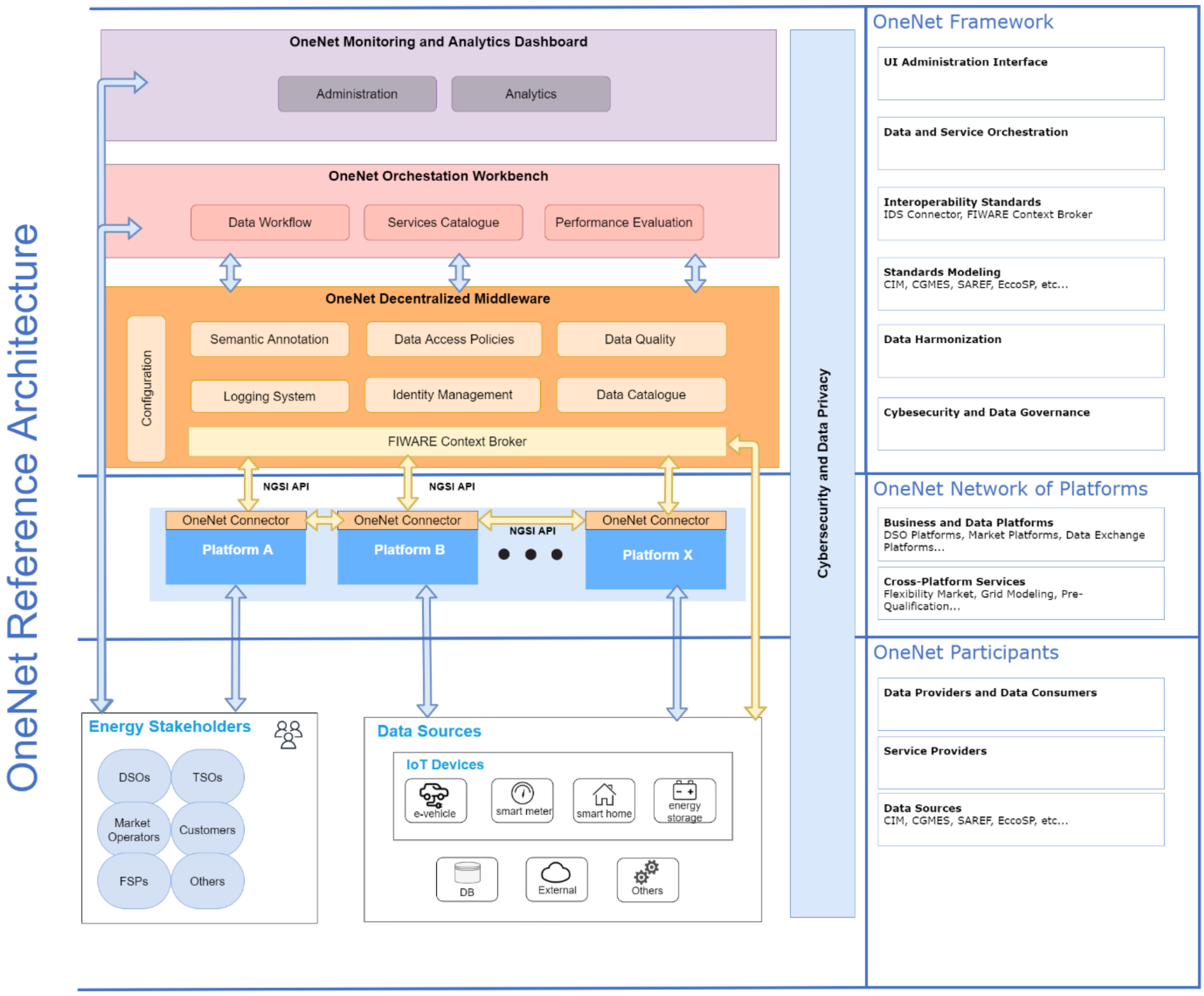



*Figure 8: OneNet Reference Architecture [21]*

**OneNet Participants**

The bottom layer of the OneNet Reference Architecture, the OneNet Participants layer includes the Business Actors and Data Sources.

The Business Actors are named according to their business role, and this can be DSO, TSO, Market Operator, FSP, Customer or others possible Energy Stakeholders. Due to the focus of the architecture in the Data exchange, these actors are classified following a data-driven set of roles such as Data Owner, User, Consumer, Provider, and others.

The following data-driven Business Actors and Roles were identified in OneNet:

- Data Provider is a specific OneNet participant that provide data to the system. To submit metadata to a Broker, or exchange data with a Data Consumer, the Data Provider uses software components (OneNet connector) that are compliant with OneNet Framework. To facilitate a data request from a Data Consumer, the Data Provider should provide proper metadata.
- Data Consumer receives data from a Data Provider. From a business process perspective, the Data Consumer is the mirror entity of the Data Provider; the activities performed by the Data Consumer are therefore similar to the activities performed by the Data Provider. Before the connection to a Data Provider can be established, the Data Consumer can search for existing datasets in a data catalogue.
- Service Provider is a specific OneNet participant that provides services or tools. The Service Provider registers its services in the OneNet Framework in order to be used, integrated and tested within any cross-platform integration or orchestration process.

The Data Sources refer to all the assets that produce or process data. These can be sensors, actuators, gateways, edge computing nodes and they can come from various energy sectors like electric mobility, energy storage and residential energy monitoring.

This layer is connected with the OneNet Network of Platforms layer since it represents the source of data for all the business and data platforms, as well as any integrable services or applications. It could be also directly connected to the OneNet Framework, since based on needs and requirements, there may be a need to integrate a data source directly into the OneNet Framework, using the OneNet Middleware and its interoperability mechanisms based on the FIWARE architecture and on the FIWARE Orion Context Broker.

Furthermore, the business actors will have the possibility to access the Orchestration Workbench for the evaluation and testing of apps, tools and services and the OneNet Monitoring Dashboard for exploiting monitoring and analytics features offered by the OneNet Framework.

**OneNet Network of Platforms: Business and Data Platforms, Application, Services**

The OneNet Network of Platforms layer focuses on the integration of external platforms, such as DSO platforms, Market platforms and other data exchange platforms into the OneNet Framework. This integration is to be made regardless of the technology of these platforms in order to remain platform-agnostic. From a technical point of view, the term platform means any software environment (e.g., applications, services or tools) able to connect with the OneNet Middleware using the OneNet Connector.

The main goal of this layer is to create a P2P fully decentralised system for interoperability. In such an infrastructure, two systems (OneNet Participants) can interact directly with each other, without intermediation by a third party. The results of this fully decentralised approach will create the OneNet Network of Platforms.

**OneNet Framework**

The OneNet Framework is the core of the OneNet Architecture. It consists of three main components: the OneNet Middleware (including OneNet Connector), the OneNet Orchestration Workbench and the OneNet Monitoring and Analytics Dashboard.

**OneNet Middleware and OneNet Connector**

The **OneNet Middleware** [21] is main component of the OneNet Framework which enable the creation of the data space ecosystem. It implements some crucial centralized features for setup the data space and provides central information to the OneNet Participants. In particular in the OneNet Decentralized Middleware are implemented:

- Identity Management
- Data Catalogue and Data source discovery
- Vocabulary Provider

The OneNet Connector is a specific instance of the OneNet Decentralized Middleware, to be deployed inside each platform environment. It allows an easy integration and cooperation among the platforms, maintaining the data ownership and preserving access to the data sources. The OneNet Connector is essential for connecting and integrating a platform within the OneNet ecosystem.

The **OneNet Connector** [21] has been developed following a decentralised approach, ensuring the necessary scalability for the near real-time data integration and management enabling multi-country and multi-stakeholder near real-time decision-making services.

The OneNet connector is based on IDS connector concept and follows all the IDS specifications and requirements. The Broker included in the OneNet Connector is implemented using the FIWARE Orion-LD Context Broker.

The FIWARE context Broker offers the FIWARE NGSI APIs and associate information model (entity, attribute, metadata) as the main interface for sharing data by the OneNet Participants. Data Providers can use the FIWARE APIs to publish or to expose the data they offer, and Data Consumers can retrieve or subscribe (to be later notified) to the data offered.

**OneNet Orchestration Workbench**

The OneNet Orchestration Workbench [22] is the component able to orchestrate and evaluate the performance and scalability of the cross-platform services that will be integrated and implemented in the OneNet System.

Any OneNet Participant and in particular the Service Providers will be able to test and evaluate its own service exploiting the OneNet Orchestration Workbench, that allows to integrate data coming from the OneNet Middleware and to implement a data pipeline orchestration.

From a functional perspective supports the integration and the evaluation of the performance and scalability of the AI, IoT and Big Data cross-platform services for market and grid operations.

The OneNet Orchestration Workbench allows to integrate data coming from the OneNet middleware and the interaction with the service providers which are able to register and evaluate their own services, implementing services orchestration, data pipeline and evaluation schema.

The OneNet Orchestration Workbench also include:

- Job Scheduling;
- App/Service registry and discovery;
- Error/Retries management; and
- SLAs tracking, alerting and notification.

**OneNet Monitoring and Analytics Dashboard**

The OneNet Monitoring and Analytics Dashboard [22] is the component that offers a GUI for facilitating the OneNet Participants in the management, monitoring and analytics of the data exchanges.

The OneNet Monitoring and Analytics Dashboard is the main User Interface that allows the access to the OneNet Participants for monitoring and data analytics features, as well as for the OneNet Administrators for configuration and administration tools.

The main features of the OneNet Monitoring and Analytics Dashboard are:

- administrative and configuration tools;
- easy integration with the OneNet Orchestration Workbench and OneNet Middleware;
- data-analytics dashboard;
- monitoring and alerting dashboard for data processes and platform integrations; and
- user-friendly selection of data sources and services from the catalogues.

### 2.3.2 ENERSHARE

ENERSHARE is a Horizon Europe-funded initiative aimed at developing a reference implementation of the Common European Energy Data Space (CEEDS). With a consortium of 30 partners across 11 EU countries, the project is designed to foster interoperable, secure, and sovereign energy data exchange across sectors such as electricity, heat, gas, and mobility [23].

The project supports the European energy transition by enabling trusted cross-sector and cross-border data sharing, empowering data sovereignty, and facilitating the emergence of new energy market structures. ENERSHARE aligns with the CEEDS blueprint and contributes to the broader European Data Space strategy, which promotes seamless data flow across domains while respecting EU values and regulations.

#### 2.3.2.1 Architecture and Technical Building Blocks

ENERSHARE adopts a federated architecture based on the Dataspace Protocol (DSP), fully compliant with the IDSA Reference Architecture Model (RAM v4), Gaia-X principles, and CEEDS structures. The architecture is peer-to-peer, supporting decentralized identity and federated data exchange, with core reliance on **TNO's Security Gateway (TSG) and Energy Data Space Connectors**.

Figure 9 below represents the ENERSHARE Reference Architecture, mapped on the BRDIGE DERA 3.0 layers [24].

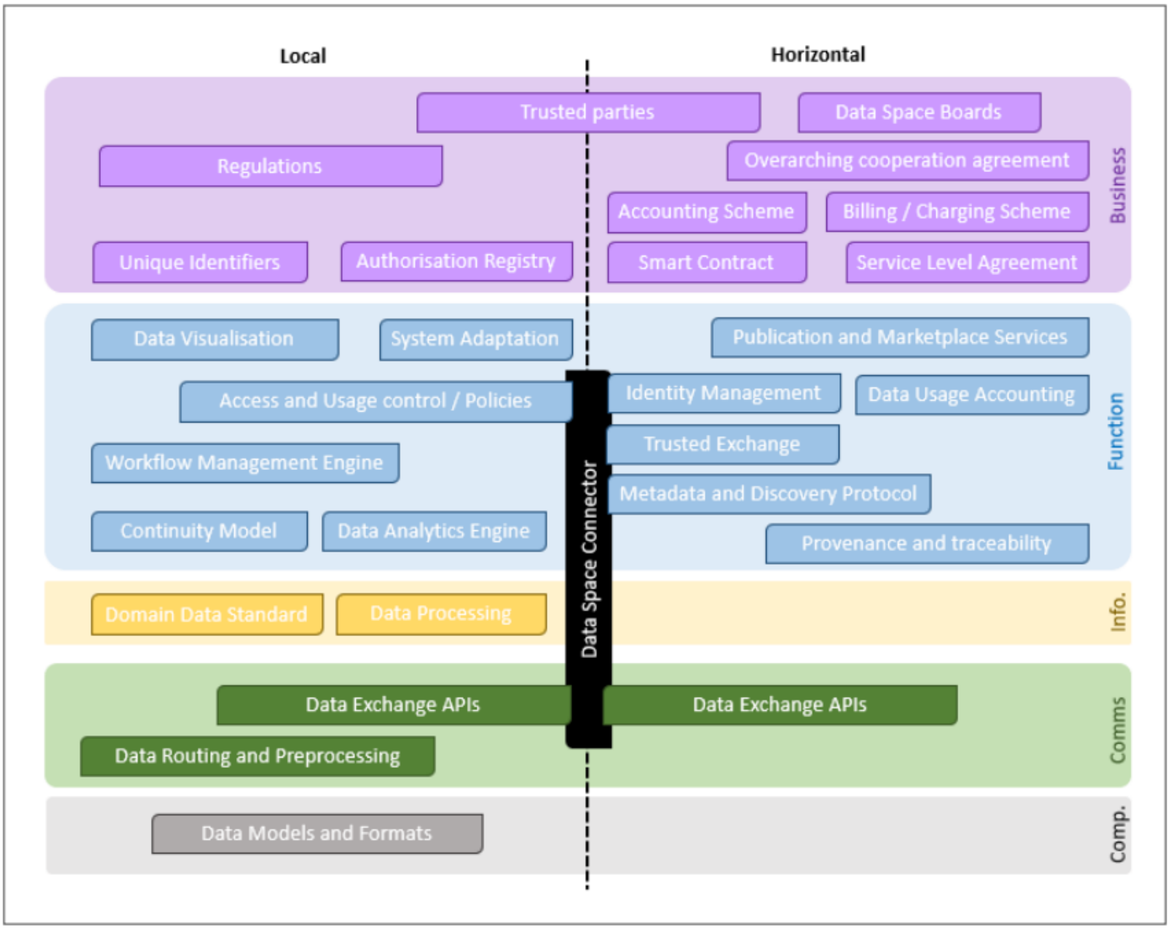



*Figure 9: ENERSHARE Reference Architecture based on BRIDGE DERA [24]*

Key components include:

- Federated Identity Management (SSI)
- Usage Control Enforcement
- Blockchain-based Policy Notarization

These elements are mapped to DSSC blueprints for access control, identity, and trusted exchange.

Technical building blocks are:

- **Data Space Connectors**: TNO Security Gateway (TSG v1), Energy Data Space Connector v1.1 (based on OneNet)
- **Identity Management**: TSG Identity Provider (IDP), Keycloak
- **Logging & Clearing**: Integrated Clearing House with Marketplace for transaction logging and payment validation
- **Vocabulary Hub**: Semantic Treehouse, supporting RDFS/OWL, SHACL, JSON schema, SKOS
- **Contracting Framework**: Blockchain-based smart contracts using ERC-20 tokens
- **Metadata Broker**: Based on TNO Security Gateway
- **Policy Enforcement**: XACML and ODRL modules integrated into connectors

#### 2.3.2.2 Interoperability Aspects

ENERSHARE contributed to the **Interoperability Framework in Energy Data Spaces v2**, a position paper developed by the IDSA Energy Interoperability Task Force [25].

This framework outlines a multi-dimensional approach to interoperability, addressing technical, semantic, organizational, and legal interoperability, each of them critical for enabling trusted, efficient, and scalable data sharing across borders and sectors.

**Technical Interoperability**

ENERSHARE contributes by implementing and testing standardized data exchange protocols, federated identity systems, and secure connectors such as the TNO Security Gateway and Energy Data Space Connector. These components ensure that systems can communicate reliably and securely, even when operated by different stakeholders or across national boundaries 1.

**Semantic Interoperability**

To address the fragmentation of data models across pilots, ENERSHARE integrates the Semantic Treehouse, a tool that supports alignment of data vocabulary using standards like RDFS/OWL, SHACL, and SKOS. This enables consistent interpretation of data, regardless of its origin, and supports real-time validation and harmonization of energy-related datasets1.

**Organizational Interoperability**

ENERSHARE developed a governance model for facilitating coordinated data sharing and service development. This organizational layer ensures that interoperability is not just technical but also embedded in the workflows and decision-making structures of data space participants.

**Legal Interoperability**

The project also addresses legal challenges by embedding policy enforcement mechanisms based on XACML and ODRL, and by developing smart contracts using ERC-20 tokens. These tools help enforce data usage rights, privacy preferences, and contractual obligations, ensuring compliance with EU regulations such as the GDPR and the Data Governance Act.

### 2.3.3 int:net - Interoperability Network for Energy Services

The int:net (Interoperability Community for the Energy Sector) project, launched in July 2022 under Horizon Europe as a Coordination and Support Action project. It acts as an enabler for the Common European Energy Data Space (CEEDS) (Ch. 2.4) by coordinating interoperability activities across multiple innovation projects (DATA CELLAR, EDDIE, Enershare, OMEGA-X and SYNERGIES) and by fostering collaboration with other initiatives such as IDSA, GAIA-X, and BRIDGE.

int:net creates an open, cross-domain community to develop, test and implement interoperable energy services. It recognizes that while technical interoperability in energy systems is relatively mature, functional and business interoperability need more attention [26]. int:net brings together researchers, ministries, regulatory authorities, standardization bodies, software vendors and operators to foster consensus on policies and standards. A key outcome is an Interoperability Maturity Model (IMM) and a methodology for assessing the interoperability of products and services.

int:net acknowledges that interoperability in the energy sector is multidimensional: beyond technical compatibility, it encompasses semantic, organizational and legal aspects. The project's main objectives are to:

- Establish a cross-sector community involving transmission system operators (TSOs), distribution system operators (DSOs), aggregators, technology providers, regulators and researchers. This community develops a knowledge platform and fosters consensus on standards and best practices for energy data interoperability [27].
- Develop an Interoperability Maturity Model (IMM) and assessment methodology to evaluate products and services across technical, semantic, organizational and legal dimensions. Solutions that meet defined criteria receive a quality seal, signalling readiness for integration within the European energy data space [28].
- Coordinate interoperability testbeds and certification through a network of testing facilities. The project harmonizes test procedures and promotes the use of open standards, thereby ensuring that connectors and services can be assessed under common conditions
- Harmonize efforts across projects in the energy data space cluster by identifying common building blocks, governance schemes and datasets. At the cluster workshop in March 2024, it was noted that the cluster works towards the interoperability of solutions implemented by the inner projects through defining system use cases and promoting common frameworks [29].

#### 2.3.3.1 Interoperability

**Cross-Sector Interoperability**

Interoperability is not confined to the electricity domain. int:net emphasizes cross-sector collaboration, particularly with mobility, buildings, industry and weather data providers. The energy data space cluster identified system use cases such as electromobility roaming, flexibility services, residential energy management and renewables operation & maintenance. Digital twins of grids will need to integrate data from these sectors to simulate and optimize network operation. By ensuring that connectors and data models are cross-domain, int:net facilitates integration of digital twins with broader energy ecosystems.

**Multidimensional Interoperability:**

int:net addresses four dimensions of interoperability [30]:

- **Technical interoperability:** ensuring that systems can physically connect and exchange data. This involves standard protocols (IDSCP, AS4, MQTT, OPC UA), secure transport (TLS), and open API frameworks. The cluster workshop highlighted the need to harmonize existing APIs and protocols and leverage standards such as CIM, SAREF, and CGMES
- **Semantic interoperability:** guaranteeing that the meaning of exchanged data is preserved. int:net promotes the use of common ontologies and vocabularies, notably CIM for power system models, SAREF for IoT devices and flexibility assets, and NGSI-LD for dynamic context data. At the workshop, stakeholders agreed that standardized data models (e.g., CIM) are fundamental for TSO–DSO cooperation
- **Organizational interoperability:** aligning business processes, workflows and roles. int:net facilitates dialogue among TSOs, DSOs and market actors to harmonize processes such as data access requests, flexibility procurement and settlement. Stakeholders stressed the need for complementary standardization of communication protocols, data formats and business processes to enable peer-to-peer exchanges without central hubs
- **Legal and regulatory interoperability:** ensuring compliance with EU and national regulations (GDPR, Data Governance Act, network codes). int:net engages regulators to address issues

such as data ownership, liability and consent. Panellists at the workshop emphasized that trust and clear governance rules are essential for customer participation and data sharing

**Interoperability Maturity Model (IMM):**

To assess progress across these dimensions, int:net develops the Interoperability Maturity Model (IMM). The IMM assigns maturity levels to products and services, guiding stakeholders from basic syntactic interoperability to full semantic, organizational and legal alignment. The model supports continuous improvement and underpins the certification process; solutions tested in int:net testbeds that reach a certain maturity level receive the int:net quality seal, signalling readiness for integration in the energy data space.

#### 2.3.3.2 Data-Space Connectors and Minimum Interoperability Mechanisms (MIMs)

**Evolution from OneNet and Enershare**

Within the energy data-space cluster, int:net leverages the decentralized middleware and connectors developed by earlier projects, most notably OneNet and Enershare. OneNet's middleware supports identity management, discovery/catalogue services, and data access management; its connector includes an IDS-based trust framework, transaction log, access and usage control, integration of a FIWARE NGSI-LD context broker, open API interfaces, blockchain notarization and data harmonization supporting standards such as the CIM and IEEE2030.5 [31]. Enershare emphasizes data sovereignty and uses IDSA connectors aligned with GAIA-X. int:net works with these projects to evolve the connector design, ensuring compatibility with both IDS and Eclipse Data Space Components (EDC) implementations.

**Key Characteristics**

int:net focuses on connectors as modular gateways enabling participants to publish and consume energy-related data. Critical characteristics include:

- Trust and identity integration: Connectors interface with identity providers and trust registries to verify participants and ensure compliance with certification schemes, drawing on GAIA-X self-descriptions and IDSA certification processes.
- Usage control and policy enforcement: Usage control mechanisms enforce data-sharing agreements by embedding policies into contracts and controlling data flows. Transaction logs provide audit trails and support dispute resolution.
- Semantic and syntactic adapters: To bridge heterogeneous systems, connectors implement adapters for CIM, IEC 61850, IEC 62325, NGSI-LD and other domain-specific formats. This allows them to translate proprietary data into standardized models and vice versa, facilitating interoperability across TSOs, DSOs, aggregators and prosumers.
- Extensibility: Connectors can be extended with NGSI-LD adapters to support real-time contextual data required by smart grids and digital twins. They also integrate blockchain notarization to guarantee data integrity and non-repudiation and can connect to context brokers or streaming platforms for high-frequency data exchange.

#### 2.3.3.3 Architecture, Infrastructure, and Governance

The int:net community advocates the adoption of Minimum Interoperability Mechanisms (MIMs) or Minimum Viable Products (MVPs) for energy data spaces. These building blocks, such as connectors, identity services, vocabularies, and data models, provide a starting point for pilot deployments. During the energy data space cluster workshop, participants emphasized leveraging existing mature key building blocks identified as MIMs and stressed the importance of mixed data-model architectures (central, federated and decentralized) [30]. This approach allows flexibility in integrating legacy systems while moving towards federated data spaces.

int:net also advocates a federated architecture where data remains with the owner and is made accessible through connectors. This architecture balances the benefits of centralized hubs (for certain national data hubs) with peer-to-peer exchanges. The cluster workshop stressed the need to support mixed data-model architectures (central, federated and decentralized) to accommodate different national contexts and legacy systems. For example:

- Centralized data hubs (e.g., Elia's data hub and Energinet's DataHub 3.0) aggregate data for visibility and market operations. They often expose open APIs and marketplaces, enabling third parties to build services
- Federated and decentralized models rely on agreements between local actors and standardized protocols. The Austrian DSO Österreichs Energie's approach focuses on decentralized data exchange using peer-to-peer mechanisms based on eIDAS and open standards for authentication and authorization. This model avoids a central hub and emphasizes semantic alignment, making it suitable for our digital-twin federation.

int:net supports hybrid models combining these approaches, enabling interoperability across national implementations while preserving data sovereignty.

**Infrastructure and Technology**

int:net does not prescribe a single technology stack; rather, it encourages the use of open-source and standardized components. Connectors can be implemented using IDSA connectors, EDC, or extended FIWARE data space connectors. Identity management may rely on eIDAS, GAIA-X IAM or other federated identity services. For message transport and event streaming, protocols such as AS4, MQTT, Kafka and REST/GraphQL are considered, depending on latency and throughput requirements. For data analytics and digital twins, edge computing and local caching are explored.

**Governance and Trust**

Decentralized architectures necessitate robust governance. int:net promotes governance frameworks that:

- Define roles and responsibilities (data owners, service providers, operators, governance bodies)
- Provide mechanisms for consent management, dispute resolution and audit
- Integrate identity and trust services to ensure only authorized entities can join the data space
- Promote open-source implementations of generic building blocks to foster transparency and avoid vendor lock-in.

These governance principles align with those of IDSA and GAIA-X and are also critical for TwinEU digital-twin federation, where multiple stakeholders will exchange sensitive grid data.

#### 2.3.3.4 Alignment with Existing Initiatives and Relevance to TwinEU

int:net leverages the IDSA Reference Architecture Model and GAIA-X principles. IDSA's focus on data sovereignty and usage control inspires the design of connectors; GAIA-X's federated identity and self-description models inform the trust framework. By adopting these standards, int:net ensures that energy data spaces can interoperate with other sectoral data spaces using a common vocabulary and governance approach.

The BRIDGE initiative provides a repository of use cases and architectures from energy demonstration projects. int:net incorporates feedback from BRIDGE's working groups on data management and interoperability. OneNet's middleware and connectors form a basis for technical realization and are being evolved through int:net to align with DSSC specifications. Enershare contributes its experience in implementing IDSA connectors and protecting data sovereignty. Through collaboration, int:net ensures that our digital-twin federation will integrate seamlessly with these platforms.

For TwinEU DT federation, aligning with int:net ensures compatibility with IDSA and GAIA-X standards, leverages lessons from OneNet and Enershare, and supports the integration of local digital twins into a European-wide energy data space while maintaining data sovereignty and trust. int:net provides concrete guidance and tools:

- **Connector alignment:** We will implement connectors that adhere to int:net's specifications and MIMs, ensuring compatibility with IDSA and EDC connectors. Adapters for CIM, NGSI-LD and other standards will enable us to integrate diverse digital-twin data sources.
- **Interoperability assessment:** Using the IMM, we can evaluate the maturity of our digital-twin services and identify gaps. Possible participation in int:net testbeds will allow us to certify our connectors and services, instilling trust among stakeholders.
- **Hybrid architecture:** By adopting mixed central/federated architectures, we can combine national data hubs with decentralized peer exchanges. This will support real-time simulation and control while preserving data sovereignty.
- **Cross-sector integration:** int:net encourages integration of energy data with mobility, buildings and other domains; TwinEU DT federation will benefit from connecting to these sectors through standardized connectors and vocabularies.

## 2.4 Common European Energy Data Space – CEEDS

The Common European Energy Data Space (CEEDS) [32] is a strategic initiative designed to foster a federated, interoperable, and sovereign ecosystem for energy data exchange across Europe. It aims to overcome fragmentation in existing data infrastructures by interconnecting heterogeneous systems—operated by diverse stakeholders—through a shared architecture that supports secure, standardized, and policy-compliant data transactions.

CEEDS is built upon the foundational principles of data sovereignty, semantic and technical interoperability, trust, security, and governance. These principles are operationalized through components such as Data Space Connectors, Federated Catalogues, and Policy Enforcement Points, which enable dynamic data discovery, contract negotiation, and real-time data exchange. The architecture supports both local platforms and federated layers, allowing seamless integration of distributed energy resources (DERs), smart grids, and consumer-facing services.

The Blueprint of CEEDS v2.0 [33] emphasizes a multi-dimensional framework that includes:

- **Security & Privacy**: Ensuring confidentiality and integrity of exchanged data.

- **Governance & Policy**: Defining rules for data access, sharing, and compliance with energy regulations.
- **Business Models**: Enabling new services such as flexibility markets and energy communities.
- **Legal Frameworks**: Supporting contractual instruments and organizational structures.
- **Operational Processes**: Facilitating use cases like residential energy management and electromobility.
- **Functional Architecture**: Aligning with standards like SGAM and ISO/IEC 42042 to ensure interoperability.

CEEDS is closely aligned with initiatives such as **Gaia-X, IDSA, and DERA 3.1**, and is envisioned as a specialization of the Data Spaces Support Centre (DSSC) reference architecture.

Finally, CEEDS aims to create a common reference for business use cases, exploring the practical applications of CEEDS across the energy sector and its potential for integration with other domains. CEEDS enables a wide range of use cases, including:

- **Grid flexibility and balancing**, where real-time data exchange supports dynamic load management and distributed energy resource (DER) coordination.
- **Renewable energy integration**, facilitating predictive analytics and optimization of solar, wind, and storage systems.
- **Energy market transparency**, enabling secure and standardized access to bidding, pricing, and consumption data.

Beyond the energy domain, CEEDS is designed to interoperate with sectors such as **mobility, smart cities, and** manufacturing, supporting cross-sectoral data flows and collaborative innovation. The architecture supports event-driven scenarios, such as automated demand response and fault detection, through real-time data streaming and push-based mechanisms.

Main use cases are enabled by the **Data Space Protocol (DSP),** which governs catalog access, contract negotiation, and data transfer.

# 3 TwinEU Data Space Design

## 3.1 Data Space as enabler for DT Federation

In the rapidly evolving landscape of digital transformation, the concept of Digital Twin technology has emerged as a pivotal innovation. Digital Twins, virtual replicas of physical entities, enable real-time monitoring, simulation, and optimization of systems and processes. However, the true potential of Digital Twins is unlocked when they are interconnected across stakeholders and sectors, forming a Federated Digital Twin (FDT) ecosystem.

The implementation of Federated Digital Twin technology presents several challenges that must be addressed to ensure its success.

1. **Interoperability**: Digital Twins must communicate and share data seamlessly by using standardized protocols, data formats, and communication mechanisms. This interoperability allows for the integration of diverse systems and platforms, facilitating cross-stakeholder collaboration.
2. **Data Sovereignty**: DT owners must maintain control over data and models by implementing mechanisms, retaining sovereignty over their data, in a trust and cooperative ecosystem
3. **Scalability**: It is important to provide scalable architecture and infrastructure capable of handling large volumes of data, in order to ensure that the FDT ecosystem can grow and adapt to increasing demands, supporting the integration of new business models, services, applications, and technologies such as AI, XR, and IoT.
4. **Security**: It is necessary to implement robust end-to-end security measures to protect sensitive data exchanged within the FDT ecosystem. These security measures maintain trust among stakeholders and ensure the integrity and confidentiality of shared data.

To achieve these goals TwinEU proposes adopting the **Data Space** concept for supporting an interoperable and secure Digital Twin Federation with an easy and effective integration of various data sources and infrastructures.

The dataspace-enabled structure for data and model sharing is the pivotal element of the TwinEU architecture. It involves the adaptation of the Energy Data Space to the specific context of TwinEU Digital Twin ecosystem, creating a trusted and sovereignty-preserving layer for data and model exchange.

The solution proposed is the **TwinEU DT Federator** which aims to:

- **Integrating** heterogeneous **Digital Twin systems**
- **Orchestrating services for data and models sharing**
- **Incorporating real-world data** to enhance simulation and analysis capabilities

## 3.2 Design Methodology

In the context of designing a federated and interoperable energy data space, it was essential to gain a detailed understanding of the needs of the business use case as well as the technical specifications for enabling a secure and trusted DT federation ecosystem.

Starting from the outcomes of WP2 and WP3 a two steps analysis was implemented.

1. Analysis of TwinEU Business Use Cases and Scenario
2. Identification and prioritization of User and Technical Requirements

The first step consisted of the analysis of all the use cases and scenarios, described in D2.2, which will be enabled by Data Space. During the analysis, a systematic methodology was developed and applied to analyze all relevant data flows contextualizing them with data space principles.

In order to identify all data exchanges, every use case was examined in detail, identifying and cataloguing the individual data exchanges involved and for each of them a unique identifier (ID) was assigned to ensure traceability and consistent referencing across systems and documentation.

The analysis focused on identifying the key actors involved in each exchange, clearly distinguishing the data provider (data owner) and data consumer, and mapping the interaction to the corresponding energy service category. Additionally, each data offering was characterized by specifying the type of data exchanged, the data model used (if any), and the exchange method—whether based on a PULL or PUSH mechanism.

Further metadata was collected to describe the data format (e.g., CSV, JSON, XML) and the exchange frequency (e.g. event, periodic, others), enabling the assessment of integration needs, technical compatibility, and performance requirements. This comprehensive approach ensures a coherent view of the data flows within the energy data space and facilitates the alignment with interoperability standards and data governance principles.

The second step consisted of the analysis of User and Technical Requirements elicited in WP2 and WP3, with a particular focus on the Data Space aspects. For each of these requirements a feasibility analysis was conducted, allowing to define a subset of requirements for the TwinEU DT Federator and prioritize the implementation of each of them using the MoSCoW method [34].

## 3.2.1 TwinEU Use Cases and Scenarios

In order to provide a comprehensive overview of the data interactions within the TwinEU project, this section presents a detailed mapping of all data exchanges associated with the use cases identified across the project’s pilot sites. Each exchange has been analyzed and recorded to ensure full traceability and alignment with the overall architecture and data space approach.

### 3.2.1.1 Energy Service Categories

To support better classification and organization of data exchanged services within the energy data space, a set of standardized Energy service categories has been defined. These categories help to clearly identify the nature and purpose of each service, enabling more consistent cataloguing, easier discovery, and improved interoperability across systems. The Table 1 below provides an overview of the identified categories, along with their corresponding codes and short descriptions. It reports and extends the list of categories already reported in D3.1.

*Table 1: TwinEU Data Service Categories*

| Categories | | |
|---|---|---|
| **Name** | **Source** | **Description** |
| **Measurements & Monitoring** | OneNet | Exchange of measurement data, sensor readings, and monitoring information |
| **Forecasts** | OneNet | Exchange of forecasts and predictive data |
| **Constraint/Limit Data** | New | Exchange of data specifying operational or technical limits and constraints |
| **Grid models (Asset/Network Topology Data)** | OneNet (extended scope) | Exchange of data related to grid structure, network topology, or asset configurations, including visualization and enrichment of network data |
| **Simulation results** | OneNet | Exchange of simulation results, for example, power flow results |
| **Reports & invoices (Analysis/Result Data)** | OneNet (extended scope) | Exchange of analytical reports, invoices, or settlement documents related to energy and flexibility services |
| **Resource (pre-) qualification** | OneNet | Activities related to the (pre-) qualification of resources, including qualification of product's/ service's technical parameters |
| **Resource control (Control/Command Data)** | OneNet (extended scope) | Sending operational commands or control signals to assets and flexibility resources |
| **System service activation** | OneNet | Instructions or requests to activate system services |
| **(Flexibility) Market participation** | OneNet | Exchange of data for market interactions |
| **Status/Alert Data** | New | Exchange of real-time or periodic status updates, alerts, alarms, health status, fault notifications, and maintenance alerts |
| **Other** | New | Reserved for data exchanges not clearly fitting into the defined categories |
| **Cybersecurity** | New | Data exchanges related to cybersecurity information |

#### 3.2.1.2 Data exchanged

Table 2 below captures the complete set of data flows, including relevant metadata such as the use case reference:

- UC: A reference to the specific Use Case in which the data exchange takes place. It helps contextualize the scenario within the broader set of pilot activities.
- ID: A unique identifier assigned to each individual data exchange, used for tracking and cross-referencing.
- Data Provider: The actor (organization, system, or component) responsible for making the data available within the data space.
- Data Consumer: The actor (organization, system, or component) that receives or accesses the data from the data space.
- Service Category: The functional classification of the service, based on a predefined list of categories (e.g., Measurements, Forecasts, Market).
- Service: The specific data offering through which the data is shared, usually associated with a DSP service.
- Data/Model Exchanged: A short description of the actual content being exchanged — this may include raw data, processed information, or predictive models.
- Method (Push/Pull): Indicates whether the data is made available via a push mechanism (automatically sent) or a pull mechanism (retrieved on request).
- Format: The structure and encoding of the data being exchanged, such as CSV, JSON, XML, etc.
- Frequency: The expected interval or timing of the data exchange, which could be real-time, periodic (e.g., hourly, daily), or event-driven.

This mapping serves as a foundational reference for the implementation of interoperable data sharing mechanisms across the TwinEU ecosystem.

*Table 2: Use Cases Data Exchange*

| UC | ID | Data Provider | Data Consumer | Service Category | Service | Data/Model Exchanged | Method (Push/Pull) | Format | Frequency |
|---|---|---|---|---|---|---|---|---|---|
| BG01 | BG01-1 | IT provider | System operators | 06 - Grid Models | Digital Twin delivery | Developed Digital Twins of Bulgarian power system | Push | CSV/JSON/XML | Event (once per version) |
| BG01 | BG01-2 | System operators | IT provider | 09 - Service Activation | Quality confirmation of DTs | Confirmation of the quality of the DTs | Push | CSV/JSON/XML | Event (once per version) |
| BG01 | BG01-3 | IT provider | System operators | 06 - Grid Models | Communication codes delivery | Developed codes for communication between the DTs | Push | CSV/JSON/XML | Event (once per version) |
| BG02 | BG02-1 | Weather forecast provider | IT provider | 00 - Generic | Weather forecast request | Request for the weather forecasts | Pull | CSV/JSON/XML | Event (Once per forecast cycle) |
| BG02 | BG02-2 | Weather forecast provider | IT provider | 03 - Forecasts | Weather forecast delivery | High-resolution numerical weather forecast | Push | CSV/JSON/XML | Event (Per forecast cycle) |
| BG02 | BG02-3 | IT provider | System operators | 03 - Forecasts | WPP production forecast | Forecast of the WPP production power | Push | CSV/JSON/XML | Event (Per forecast cycle) |
| BG03 | BG03-1 | Weather forecast provider | IT provider | 00 - Generic | Weather forecast request | Request for the weather forecasts | Pull | CSV/JSON/XML | Event (Once per forecast cycle) |
| BG03 | BG03-2 | Weather forecast provider | IT provider | 03 - Forecasts | Weather forecast delivery | High-resolution numerical weather forecast | Push | CSV/JSON/XML | Event (Per forecast cycle) |
| BG03 | BG03-3 | IT provider | System operators | 03 - Forecasts | SPP production forecast | Forecast of the SPP production power | Push | CSV/JSON/XML | Event (Per forecast cycle) |
| BG04 | BG04-1 | Weather forecast provider | IT provider | 00 - Generic | Weather forecast request | Request for the weather forecasts | Pull | CSV/JSON/XML | Event (Once per forecast cycle) |
| BG04 | BG04-2 | Weather forecast provider | IT provider | 03 - Forecasts | Weather forecast delivery | High-resolution numerical weather forecast | Push | CSV/JSON/XML | Event (Per forecast cycle) |
| BG04 | BG04-3 | IT provider | System operators | 03 - Forecasts | Ampacity forecasts | Ampacity forecasts for the selected lines | Push | CSV/JSON/XML | Event (Per forecast cycle) |
| BG05 | BG05-1 | System operators | IT provider | 03 - Forecasts | Ampacity Forecast Request | List of critical lines for ampacity forecasts | Push | CSV/JSON/XML | Event |
| BG05 | BG05-2 | IT provider | System operators | 03 - Forecasts | Ampacity Forecasts | Ampacity forecasts for critical lines | Push | CSV/JSON/XML | Periodic/as needed |

| UC | ID | Data Provider | Data Consumer | Service Category | Service | Data/Model Exchanged | Method (Push/Pull) | Format | Frequency |
|---|---|---|---|---|---|---|---|---|---|
| BG06 | BG06-1 | IT provider | System operator | 03 - Forecasts | Forecasts of WPP, SPP and ampacity | Forecasts of production and ampacity for RES connection optimization | Push | CSV/JSON/XML | Event |
| BG06 | BG06-2 | FSP | System operator | 03 - Forecasts | Forecasts of WPP, SPP and ampacity | Forecasts of production and ampacity for RES connection optimization | Push | CSV/JSON/XML | Event |
| BG06 | BG06-3 | System operator | IT provider | 06 - Grid Models | Updated grid models | Updated grid models including RES and forecast data | Push | CSV/JSON/XML | Event |
| BG06 | BG06-4 | IT provider | System operator | 07 - Simulation Results | Results of the optimization | Optimized connection points and related data | Push | CSV/JSON/XML | Event |
| BG07 | BG07-1 | System operator | IT provider | 03 - Forecasts | Request for forecasts | Request including list of critical regimes | Push | CSV/JSON/XML | Event |
| BG07 | BG07-2 | IT provider | System operator | 03 - Forecasts | Forecasts for critical regimes | Forecasted parameters for critical regimes | Push | CSV/JSON/XML | Event |
| BG08 | BG08-1 | System operator | IT provider | 03 - Forecasts | Request for forecasts | Request for forecasts for critical regimes | Push | CSV/JSON/XML | Event |
| BG08 | BG08-2 | IT provider | System operator | 03 - Forecasts | Forecasts for critical regimes | Forecasted parameters for critical regimes | Push | CSV/JSON/XML | Event |
| BG09 | BG09-1 | IT provider | System operator (DSO) | 03 - Forecasts | Forecasts delivery | Forecasts of WPP, SPP production and ampacity | Push | CSV/JSON/XML | Event |
| BG09 | BG09-2 | System operator (DSO) | FSPs | 09 - Service Activation | Request for services | Information on needed services | Push | CSV/JSON/XML | Event |
| BG09 | BG09-3 | FSPs | System operator (DSO) | 09 - Service Activation | Service offers submission | Offers regarding required services | Push | CSV/JSON/XML | Event |
| BG10 | BG10-1 | IT provider | TSO | 03 - Forecasts | Flexibility exchange to avoid the congestions | Forecasts of the WPP and SPP production and the ampacity | Push | CSV/JSON/XML | Event |
| BG10 | BG10-2 | DSO | TSO | 03 - Forecasts | Flexibility exchange to avoid the congestions | Information on the needed services | Pull | CSV/JSON/XML | Event |
| BG10 | BG10-3 | FSPs | DSO | 03 - Forecasts | Flexibility exchange to avoid the congestions | Information on the needed services | Pull | CSV/JSON/XML | Event |
| BG10 | BG10-4 | FSPs | DSO | 03 - Forecasts | Flexibility exchange to avoid the congestions | Offers regarding the required services | Push | CSV/JSON/XML | Event |

| UC | ID | Data Provider | Data Consumer | Service Category | Service | Data/Model Exchanged | Method (Push/Pull) | Format | Frequency |
|---|---|---|---|---|---|---|---|---|---|
| BG10 | BG10-5 | DSO | TSO | 03 - Forecasts | Flexibility exchange to avoid the congestions | Offers regarding the required services | Push | CSV/JSON/XML | Event |
| EACL-IT-01 | EACL-IT-01-1 | TSO | MO | 00 - Generic | Regulations and Standards | TSO regulations and standards | Push | CSV/JSON/XML | Periodic |
| EACL-IT-02 | EACL-IT-02-1 | MO | TSO | 00 - Generic | Design proposal | BIM model and loin | Push | CSV/JSON/XML | Periodic |
| EACL-IT-02 | EACL-IT-02-2 | TSO | MO | 04 - Report and Invoices | Project validation | Validation outcome | Push | CSV/JSON/XML | Periodic |
| EACL-IT-03 | EACL-IT-03-1 | MO | TSO | 04 - Report and Invoices | Scalability analysis | Scalability analysis for the design validation tool | Push | CSV/JSON/XML | Periodic |
| EM-CY-02 | EM-CY-02-1 | MO | TSO | 05 - Market | Regional coordination between interconnected systems | Day-Ahead market results | Push | CSV/JSON/XML | Event |
| EM-CY-02 | EM-CY-02-2 | TSOs | MOs | 05 - Market | Regional coordination between interconnected systems | FCR products procurement | Push | CSV/JSON/XML | Event |
| EM-CY-02 | EM-CY-02-3 | TSOs | RSC | 06 - Grid Models | Regional coordination between interconnected systems | System constraint submission | Push | CSV/JSON/XML | Event |
| EM-CY-02 | EM-CY-02-4 | FSP, HVDC operator | MO | 00 - Generic- User defined service | Regional coordination between interconnected systems | Bids submission to FCR markets | Push | CSV/JSON/XML | Event |
| EM-CY-02 | EM-CY-02-5 | RSC | MO and HVDC operator | 00 - Generic- User defined service | Regional coordination between interconnected systems | Submission of the two system constraints | Push | CSV/JSON/XML | Event |
| EM-CY-02 | EM-CY-02-6 | MOs | FSPs, HVD operators | 05 - Market | Regional coordination between interconnected systems | Information on Market clearing results | Push | CSV/JSON/XML | Every three hours |
| EM-CY-02 | EM-CY-02-7 | HVDC operator (HVDC P&Q controller), RSC (wide area regional controller) | HVDC operator (HVDC Vdc&Q controller) | 00 - Generic- User defined service | Real-time operation for frequency stability enhancement of interconnected systems | RSC wide area coordination signals | Push | CSV/JSON/XML | Event |
| GE07 | GE07-1 | FSP | TSO | 08 - Pre-Qualification | Master Data information request | Newly registered electricity producing unit, including type, location, and rated power | Pull | JSON/XML | weekly |
| GE07 | GE07-2 | FSP | TSO | 02 - Measurements and Monitoring | Live Data Stream | Produced electricity of a unit in MWh in 1min interval | Pull | JSON/XML | Every minute |
| GE07 | GE07-3 | FSP | TSO | 02 - Measurements and Monitoring | Live Data Stream | P produced electricity of a unit in MWh in 15min interval | Pull | JSON/XML | Every 15 minutes |

| UC | ID | Data Provider | Data Consumer | Service Category | Service | Data/Model Exchanged | Method (Push/Pull) | Format | Frequency |
|---|---|---|---|---|---|---|---|---|---|
| GE07 | GE07-4 | DSO | TSO | 08 - Pre-Qualification | Master Data information request | Newly registered electricity producing unit, including type, location, and rated power | Pull | JSON/XML | weekly |
| GE07 | GE07-5 | DSO | TSO | 02 - Measurements and Monitoring | Live Data Stream | Produced electricity of a unit in MWh in 1min interval | Pull | JSON/XML | Every minute |
| GE07 | GE07-6 | DSO | TSO | 02 - Measurements and Monitoring | Live Data Stream | P produced electricity of a unit in MWh in 15min interval | Pull | JSON/XML | Every 15 minutes |
| GE07 | GE07-1 | FSP | TSO | 08 - Pre-Qualification | Master Data information request | Newly registered electricity producing unit, including type, location, and rated power | Pull | JSON/XML | weekly |
| HU01 | HU01-1 | DSO | TSO | 00 - Generic- User defined service | Power line selection | Name of involved power lines | Push | CSV/JSON/XML | Event |
| HU01 | HU01-2 | DSO | DSO | 00 - Generic- User defined service | Power line selection | Name of involved power lines | Push | CSV/JSON/XML | Event |
| HU01 | HU01-3 | DSO | DSO | 00 - Generic- User defined service | Configuration of the hardware part of the monitoring system | Spans for monitoring equipment installation | Push | CSV/JSON/XML | Event |
| HU01 | HU01-4 | DSO | Technology provider | 00 - Generic- User defined service | Collection of technical data | Phase conductor data | Push | CSV/JSON/XML | Event |
| HU01 | HU01-5 | TSO | Technology provider | 00 - Generic- User defined service | Collection of technical data | Phase conductor data | Push | CSV/JSON/XML | Event |
| HU01 | HU01-6 | DSO | Technology provider | 00 - Generic- User defined service | Field data collection (short period) | DT structure | Push | CSV/JSON/XML | Event |
| HU01 | HU01-7 | TSO | Technology provider | 00 - Generic- User defined service | Field data collection (short period) | DT structure | Push | CSV/JSON/XML | Event |
| HU01 | HU01-8 | Technology provider | Academic partner | 00 - Generic- User defined service | Evaluation of DT performance (short training period; same system) | DT performance | Push | CSV/JSON/XML | Event |
| HU01 | HU01-9 | Technology provider | Academic partner | 00 - Generic- User defined service | Evaluation of DT performance (long training period; same system) | DT performance | Push | CSV/JSON/XML | Event |
| HU01 | HU01-10 | Technology provider | Academic partner | 00 - Generic- User defined service | Evaluation of DT performance (long training period; different system) | DT performance | Push | CSV/JSON/XML | Event |
| HU01 | HU01-11 | Technology provider | Academic partner | 00 - Generic- User defined service | Evaluation of DT performance (long training period; different system) | DT performance | Push | CSV/JSON/XML | Event |
| HU01 | HU01-12 | TSO | Academic partner | 00 - Generic- User defined service | Evaluation of real-time capacity gain | DLR (dynamic line rating) | Push | CSV/JSON/XML | Event |

| UC | ID | Data Provider | Data Consumer | Service Category | Service | Data/Model Exchanged | Method (Push/Pull) | Format | Frequency |
|---|---|---|---|---|---|---|---|---|---|
| HU01 | HU01-13 | DSO | Academic partner | 00 - Generic- User defined service | Evaluation of real-time capacity gain | DLR (dynamic line rating) | Push | CSV/JSON/XML | Event |
| HU01 | HU01-14 | Technology provider | Academic partner | 00 - Generic- User defined service | Evaluation of real-time capacity gain | DLR (dynamic line rating) | Push | CSV/JSON/XML | Event |
| HU02 | HU02-1 | TSO | TP | 02 - Measurement and Monitoring | Flow based data | Grid model, BC demand, BC supply and energy data | Push | CSV/JSON/XML | Periodic |
| HU02 | HU02-2 | AP | TP | 02 - Measurement and Monitoring | DLR Data | DLR data | Push | CSV/JSON/XML | Periodic |
| HU02 | HU02-3 | TP | TSO | 02 - Measurement and Monitoring | Capacity calculation | Flow based data integrated with DLR based capacity calculation | Push | CSV/JSON/XML | Periodic |
| HU02 | HU02-4 | TP | MO | 02 - Measurement and Monitoring | Capacity calculation | Flow based data integrated with DLR based capacity calculation | Push | CSV/JSON/XML | Periodic |
| HU03 | HU03-1 | TSO | MO | 05 - Market | BC bids | BC bids | Push/Pull | CSV/JSON/XML | Periodic |
| HU03 | HU03-2 | TSO | MO | 05 - Market | Bids | Public Sources Energy Bids | Push/Pull | CSV/JSON/XML | Periodic |
| HU03 | HU03-3 | AP | MO | 05 - Market | Bids | Public Sources Energy Bids | Push/Pull | CSV/JSON/XML | Periodic |
| HU03 | HU03-4 | MO | TSO | 05 - Market | Market results | Market results | Push | CSV/JSON/XML | Periodic |
| HU03 | HU03-5 | MO | TP | 05 - Market | Market results | Market results | Push | CSV/JSON/XML | Periodic |
| IB01 | IB01-1 | TSO | RC | 02 - Measurement and Monitoring | Historical data | Historical data | Push | CSV/JSON/XML | Periodic |
| IB01 | IB01-2 | TSO | RC | 02 - Measurement and Monitoring | Generation data | Generation data | Push | CSV/JSON/XML | Periodic |
| IB01 | IB01-3 | RC | TSO | 03 - Forecast | Forecasted demand and supply | Forecasting data | Push | CSV/JSON/XML | Periodic |
| IB01 | IB01-4 | RC | TSO | 04 - Report and Invoices | AI reporting | Report of AI analysis | Push | CSV/JSON/XML | Periodic |
| IB01 | IB01-5 | RC | TSO | 04 - Report and Invoices | Grid optimization reports | Grid optimization suggestions | | | |
| IB02 | IB02-1 | TSO | RC | 02 - Measurement and Monitoring | Historical power system and weather data | Historical power system and weather data | Push | CSV/JSON/XML | Periodic |
| IB02 | IB02-2 | RC | TSO | 04 - Report and Invoices | Model training report | Status update on the training of the AI model | Push | CSV/JSON/XML | Periodic |

| UC | ID | Data Provider | Data Consumer | Service Category | Service | Data/Model Exchanged | Method (Push/Pull) | Format | Frequency |
|---|---|---|---|---|---|---|---|---|---|
| IB02 | IB02-3 | RC | TSO | 04 - Report and Invoices | AI generated scenario report | Summary of generated scenarios and insights for planning | Push | CSV/JSON/XML | Periodic |
| IB04 | IB04-1 | MO | TSO | 05 - Market | Wholesale market abnormal participation detection | Parameters identifying potential abnormal participation | Push? | JSON/XML | Near real-time / Event |
| IB04 | IB04-10 | DSO | MO | 02 - Measurements and Monitoring | Monitoring protocol effectiveness | Metrics related to the result of the protocol activation | Pull | CSV/JSON/XML | Periodic (e.g., hourly) |
| IB04 | IB04-11 | MO | TSO | 02 - Measurements and Monitoring | Monitoring protocol effectiveness | Metrics related to the result of the protocol activation | Pull | CSV/JSON/XML | Periodic (e.g., hourly) |
| IB04 | IB04-12 | MO | DSO | 02 - Measurements and Monitoring | Monitoring protocol effectiveness | Metrics related to the result of the protocol activation | Pull | CSV/JSON/XML | Periodic (e.g., hourly) |
| IB04 | IB04-2 | TSO | MO | 05 - Market | Wholesale market abnormal participation detection | Parameters identifying potential abnormal participation | Push? | JSON/XML | Near real-time / Event |
| IB04 | IB04-3 | MO | DSO | 05 - Market | Local flexibility market abnormal participation detection | Parameters identifying potential abnormal participation | Push? | JSON/XML | Near real-time / Event |
| IB04 | IB04-4 | DSO | MO | 05 - Market | Local flexibility market abnormal participation detection | Parameters identifying potential abnormal participation | Push? | JSON/XML | Near real-time / Event |
| IB04 | IB04-5 | MO | TSO | Cybersecurity (NEW) | Activation of mitigation protocol | Proposed measures to activate the protocol | Push | JSON/XML | On protocol activation |
| IB04 | IB04-6 | TSO | MO | Cybersecurity (NEW) | Activation of mitigation protocol | Proposed measures to activate the protocol | Push | JSON/XML | On protocol activation |
| IB04 | IB04-7 | MO | DSO | Cybersecurity (NEW) | Activation of mitigation protocol | Proposed measures to activate the protocol | Push | JSON/XML | On protocol activation |
| IB04 | IB04-8 | DSO | MO | Cybersecurity (NEW) | Activation of mitigation protocol | Proposed measures to activate the protocol | Push | JSON/XML | On protocol activation |
| IB04 | IB04-9 | TSO | MO | 02 - Measurements and Monitoring | Monitoring protocol effectiveness | Metrics related to the result of the protocol activation | Pull | CSV/JSON/XML | Periodic (e.g., hourly) |
| IB14 | IB14-1 | FSP | TSO | 08 - Pre-Qualification | Prequalification application | Technical data and documentation | Push | CSV/JSON/XML | Periodic |
| IB14 | IB14-10 | RU | TSO | 02 - Measurements and Monitoring | Electrical measurements | Power output collected by reserve units and grid | Pull | CSV/JSON/XML | Periodic |
| IB14 | IB14-2 | FSP | TSO | 08 - Pre-Qualification | Prequalification request | Prequalification info, technical data and test request | Pull/Push | CSV/JSON/XML | Periodic |

| UC | ID | Data Provider | Data Consumer | Service Category | Service | Data/Model Exchanged | Method (Push/Pull) | Format | Frequency |
|---|---|---|---|---|---|---|---|---|---|
| IB14 | IB14-3 | FSP | TSO | 08 - Pre-Qualification | Reports and Records | Reports, test results, performance data | Push | CSV/JSON/XML | Periodic |
| IB14 | IB14-4 | TSO | FSP | 08 - Pre-Qualification | Prequalification results | Notification on reserve unit status | Push | CSV/JSON/XML | Periodic |
| IB14 | IB14-5 | TSO | MO | 03 - Forecast | Forecast | Forecast of expected FFR guiding FSP to prepare bids | Push | CSV/JSON/XML | Periodic |
| IB14 | IB14-6 | MO | FSP | 03 - Forecast | Forecast | Forecast of expected FFR guiding FSP to prepare bids | Push | CSV/JSON/XML | Periodic |
| IB14 | IB14-7 | FSP | MO | 05 - Market | Bids | Capacity and price offers in response to forecast | Push | CSV/JSON/XML | Periodic |
| IB14 | IB14-8 | MO | FSP | 05 - Market | Market results | Accepted bids, contract details, activation terms | Push | CSV/JSON/XML | Periodic |
| IB14 | IB14-9 | MO | TSO | 05 - Market | Market results | Accepted bids, contract details, activation terms | Push | CSV/JSON/XML | Periodic |
| IT01 | IT01-1 | DSO | TSO | 02 - Measurement and Monitoring | MV feeder data | MV feeder data (voltage, current, power, frequency) | Push | CSV/JSON/XML | Periodic |
| IT02 | IT02-1 | DSO | TSO | 09 - Service Activation | Disconnection strategy data | Disconnection strategy data | Push | CSV/JSON/XML | Periodic |
| IT02 | IT02-2 | TSO | DSO | 09 - Service Activation | Disconnection request | Disconnection request | Push | CSV/JSON/XML | Periodic |
| IT02 | IT02-3 | DSO | DER | 09 - Service Activation | DERs intervention activation | DERs intervention activation | Push | CSV/JSON/XML | Periodic |
| IT02 | IT02-4 | DSO | TSO | 02 - Measurement and Monitoring | MV grid performance | MV grid performance for post contingency analysis | Push | CSV/JSON/XML | Periodic |

### 3.2.2 User and Technical Requirements

The second step foreseen in the analysis of results coming from WP2 and WP3 activities was the identification of user and technical requirements related to Data Space.

In the context of T2.3, one of the main activities focused on the identification of technical requirements for DTs coordination and cooperation in the TwinEU Continuum. The complete list of requirements is reported on D2.2.

In addition, general functional requirements for the implementation of the TwinEU Platform and DT Federation were identified and in the context of T3.2. The complete list of these requirements is reported on D3.2

A subset of these requirements was considered impactful for the implementation of TwinEU DT Federator and prioritized using the MoSCoW methodology. The prioritized list is reported in Table 3 (Technical Requirements) and Table 4 (Functional Requirements) below.

*Table 3: TwinEU DT Federator Technical Requirements*

| Technical requirement ID | Name | Description | Priority |
|---|---|---|---|
| Tech_01 | All DTs should have the capacity to be connected to the TwinEU's dataspace | Task 4.5 will adapt the current dataspace based on IDSA to work with the TwinEU continuum. This adaptation will define data models, the data sharing governance, the data integration and homogenization mechanisms and the extension of a data connector to support the required features. | MUST |
| Tech_02 | Real-time Data Synchronization | TwinEU DTs may implement real-time data synchronization mechanisms to keep all DTs updated with the latest information. | MUST |
| Tech_03 | Decentralized Data Architecture | Use decentralized data architectures to enable efficient data sharing and reduce bottlenecks in communication. | MUST |
| Tech_05 | TwinEU must ensure a certain level of Data Quality and its validation | The project architecture should establish mechanisms for data validation and quality assurance to ensure the accuracy and reliability of inputs used in DT simulations and analyses. | SHOULD |
| Tech_06 | Automated Data Exchange Interfaces | The DTs in the TwinEU continuum must be provided with automated interfaces for data exchange via de data space, reducing manual intervention and ensuring timely updates. | MUST |
| Tech_07 | Dynamic Data Sharing Policies | Develop dynamic data sharing policies to control access and ensure data is shared appropriately among DTs. | SHOULD |
| Tech_08 | Hybrid employment of edge and cloud | TwinEU's DT can be architecturally conceived with a balance between using cloud-based platforms for scalable and flexible data exchange among DTs and employing edge computing to process data locally, | COULD |

| | | | |
|---|---|---|---|
| | computing mechanisms | always aiming at reducing latency and improving response times for the DT coordination. This does not exclude any DT to be developed fully locally or in the cloud. | |
| Tech_09 | Data Exchange Performance Metrics for Continuous Improvement | The TwinEU dataspace should implement monitoring performance metrics for data exchange to ensure efficiency and effectiveness in DT collaboration. | SHOULD |
| Tech_10 | Data Exchange Feedback Loops for Continuous Improvement | The TwinEU dataspace must implement feedback loops to continuously improve data exchange processes and address emerging challenges in DT coordination. | SHOULD |
| Tech_11 | Data Space federation techniques and data exchange infrastructure | TwinEU's data space must employ data federation techniques to provide unified access to distributed data sources across DTs as well as design a scalable infrastructure to support growing data exchange needs as more DTs are integrated. | MUST |
| Tech_12 | Inter-Twins Data Sharing Agreements | Beyond enabling a context-aware data sharing to provide relevant information to DTs based on their operational needs and scenarios, TwinEU must implement data sharing agreements among stakeholders to define roles, responsibilities, and data access rights. | COULD |
| Tech_13 | Secure Data Transmission Protocols | TwinEU must employ secure data transmission protocols to protect data integrity and confidentiality during exchanges. The implementation of multi-layer security measures to protect data at rest and in transit between DTs should be studied. | MUST |
| Tech_14 | Event-Driven Data Exchange | TwinEU's architecture must facilitate event-driven triggers based on specific conditions or changes in the system. | MUST |
| Tech_15 | API-driven data space connectors compatibility | The TwinEU Data Space Connectors must provide an API for the generation of offers and contracts | MUST |
| Tech_16 | Access Control Mechanisms | Establish robust access control mechanisms to ensure that only authorized users and systems can access the TwinEU continuum and its services. | MUST |
| Tech_17 | Secure Software Development | The TwinEU DTs and related services (via the data space implementation) must be developed according to the secure software development principles to minimize vulnerabilities in the ecosystem. | MUST |
| Tech_18 | Compliance with Data Protection Laws | All Use Cases must ensure compliance with relevant data protection laws and regulations, such as GDPR, to safeguard personal and sensitive information. | MUST |

*Table 4: TwinEU DT Federator Functional Requirements*

| Requirement ID | Name | Description | Priority |
|---|---|---|---|
| TwinEU_GFUR_01 | TwinEU system must be able to manage and certificate the identity of each TwinEU Participant | TwinEU system manage the identities of all the TwinEU participants offering an Identity Provider. | MUST |
| TwinEU_GFUR_02 | TwinEU system must be able to register/unregister a Data Space connector | Data Space Connector need to register itself before starting any data exchange process. | MUST |
| TwinEU_GFUR_03 | Each TwinEU Participant must be uniquely identified using certification | TwinEU Participants are uniquely identified within the TwinEU ecosystem, using certification process and establishing trust among all participants. | MUST |
| TwinEU_GFUR_04 | Each Data Space Connector has a unique certificate and identifier | | MUST |
| TwinEU_GFUR_05 | Each Data Space Connector is able to verify the identity of the other Data Space Connectors | | MUST |
| TwinEU_GFUR_06 | TwinEU participant must be able to run the Data Space connector in its own environment | TwinEU Middleware leverage on the IDS decentralized approach. The Data Space Connector provided by TwinEU must be deployable in any environment. | MUST |
| TwinEU_GFUR_07 | The TwinEU Participant must be able to configure its own Data Space Connector | Data Space connectors are configurable by the TwinEU participants using specific interfaces. | MUST |
| TwinEU_GFUR_08 | The Data Space connector must be able to send metadata of a data source to one or more system components | Once the connector is configured it is able to provide and/or search metadata as well as discover for new data sources and participants. | MUST |
| TwinEU_GFUR_09 | The Participant must be able to search and discover other TwinEU Participants | | MUST |

| | | | |
|---|---|---|---|
| TwinEU_GFUR_10 | The Data Space Connector must be able to search for metadata published within the data space. | | MUST |
| TwinEU_GFUR_11 | The Data Space Connector must be able to exchange data with other connectors using pull and/or push mechanisms. | The data exchange process happens end-to-end exploiting pull or push mechanisms. | MUST |
| TwinEU_GFUR_12 | The TwinEU system must be able to support the creation, management and usage of vocabularies | A feature provided by TwinEU system is the Vocabulary Provider. It manages and offers vocabularies (i.e., ontologies, reference data models, or metadata elements) that can be used to annotate and describe datasets. | MUST |
| TwinEU_GFUR_13 | The TwinEU participant could use vocabularies for creating and structuring its metadata | | SHOULD |
| TwinEU_GFUR_14 | The TwinEU system should offer data services/apps for data processing and transformation | One of the main features of the TwinEU system is the possibility to enrich, transform, validate and harmonize the data processed. In addition, the TwinEU allow to log all the data transaction. | COULD |
| TwinEU_GFUR_15 | The TwinEU system should be able to log any data transaction between any TwinEU participant. | | MUST |
| TwinEU_GFUR_16 | The TwinEU system should be able to assess the quality of data processed. | | COULD |
| TwinEU_GFUR_17 | The TwinEU system should be able to perform a semantic validation of the data processed | | COULD |
| TwinEU_GFUR_18 | The TwinEU system could use AI | For improving the Data Services offered by the TwinEU system, | COULD |

| | | | |
|---|---|---|---|
| | mechanism for empowering Data services. | some AI mechanism could be implemented. | |
| TwinEU_GFUR_19 | The TwinEU Orchestration Workbench must be able to manage data and service orchestration | The TwinEU Orchestration Workbench aims to support the data orchestration for the evaluation of the performance and scalability of the AI, IoT and Big Data cross-platform services for market and grid operations. The TwinEU Orchestration Workbench allows to integrate data coming from the TwinEU middleware and implement a data pipeline orchestration. It also should include: Job Scheduling App/Service registry and discovery Error/Retries management SLAs tracking, alerting and notification. The system should provide a centralized AI & Big Data marketplace for discovering, integrating, and utilizing AI-driven services and analytics tools. The system should support service registration and discovery through an IDSA-compliant Federated Catalogue. | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_20 | The TwinEU Orchestration Workbench must be able to integrate data using the TwinEU Middleware | | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_21 | The Service Provider must be able to register its service in the TwinEU Orchestration Workbench | | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_22 | The Service Provider must be able to create a data workflow using the Orchestration Workbench | | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_23 | The Service Provider must be able to evaluate the performance of its own service | | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_24 | The TwinEU Orchestration Workbench should provide a service catalogue to the TwinEU Participants | | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_25 | The TwinEU system should offer a UI dashboard to TwinEU Participants for monitoring and analytics | The TwinEU system should implement a GUI for facilitating the TwinEU Participants in the management, monitoring and analytics of the data transactions. | N/A (Not strictly related with TwinEU DT Federator) |

| | | | |
|---|---|---|---|
| TwinEU_GFUR_26 | Digital Twin Federation Integration | The system must enable the creation of a Digital Twin Federation by integrating local Digital Twins into a system-of-systems. | MUST |
| TwinEU_GFUR_27 | Bidirectional Data Exchange for Physical and Virtual Entities | The system should provide bidirectional data flow between physical and virtual entities. | SHOULD |
| TwinEU_GFUR_28 | Seamless Interoperability Across Digital Twins | The system should ensure seamless interoperability between different Digital Twins, regardless of their underlying data structures and models. | SHOULD |
| TwinEU_GFUR_29 | Secure Digital Twin Data Exchange via Data Space Connector | The system should utilize the Data Space Connector to enable secure and controlled data sharing between Digital Twins. | SHOULD |
| TwinEU_GFUR_30 | Scalable Federated Computing Architecture Support | The system should support distributed and federated computing architectures to enhance scalability. | SHOULD |
| TwinEU_GFUR_31 | Federated Digital Twin Data Space Orchestration | The system should implement a Data Space Framework for orchestrating data sharing, model exchange, and real-world data integration across Digital Twin clusters. | SHOULD |
| TwinEU_GFUR_32 | Regulatory-Compliant Data Exchange Governance | The system should ensure that all data exchanges follow data governance policies and regulatory compliance requirements. | SHOULD |
| TwinEU_GFUR_33 | Integration of heterogeneous data sources | The system shall enable integration of heterogeneous data sources regardless of their structure or format. | SHOULD |
| TwinEU_GFUR_34 | Service Orchestration | The system should allow dynamic orchestration of services, enabling users to define and manage workflows. | SHOULD |
| TwinEU_GFUR_35 | Trusted and authorized Digital Twins | The system shall verify that only trusted and authorized Digital Twins are permitted to participate in data exchange | SHOULD |
| TwinEU_GFUR_36 | Data versioning and traceability | The system could enable data versioning and traceability to | COULD |

| | | | |
|---|---|---|---|
| | | track data provenance and integrity. | |
| TwinEU_GFUR_37 | Role-Based Access Control for Secure Data Exchange | The system should allow role-based access control (RBAC) for data exchange, ensuring that only authorized entities can access or modify data. | SHOULD |
| TwinEU_GFUR_38 | Error-handling mechanism | The system could include an error-handling mechanism that enables automatic retries and logs failures for further analysis. | COULD |
| TwinEU_GFUR_39 | Identity management | The system should implement a robust identity management framework to authenticate and authorize Digital Twin actors. | SHOULD |
| TwinEU_GFUR_40 | Governance model | The system should include a governance model to manage interactions and security policies across the Digital Twin Federation. | SHOULD |
| TwinEU_GFUR_41 | Validation of DT | The system could support real-world validation of Digital Twin interoperability through simulation environments. | COULD |
| TwinEU_GFUR_42 | Policy-driven optimizations | The system should allow policy-driven optimizations to ensure regulatory compliance and efficient energy market operations. | SHOULD |
| TwinEU_GFUR_43 | Enforced Data Retention Policy for Sensor Data | The system will not store raw sensor data indefinitely but will implement a data retention policy aligned with regulatory and operational needs. | W'ONT |
| TwinEU_GFUR_44 | AI & Big Data Marketplace for providers | The system should allow service providers to publish and manage AI models, services, and data assets. | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_45 | AI & Big Data Marketplace for consumers | The system should enable data consumers to access and integrate available AI-driven services via standardized APIs. | N/A (Not strictly related with TwinEU DT Federator) |

| | | | |
|---|---|---|---|
| TwinEU_GFUR_46 | Predictive Analytics and Data-Driven AI Solutions | The system could offer real-time anomaly detection to identify inconsistencies in energy data and prevent disruptions. | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_47 | Data Streaming Consumption and Querying | The system should allow data consumers to retrieve data entities based on specific queries. | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_48 | Data Streaming Provision | The system shall allow data providers to create, modify, and delete data entities. | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_49 | IoT Indexing and Discovery | The system should enable registration and management of IoT devices as data entities. The system should maintain a catalogue of all registered entities and their associated providers. | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_50 | IoT Device Integration | The system should support the registration of IoT devices. The system should map each registered IoT device to TwinEU data entity associated with a Data Provider. | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_51 | Validation against regulatory standards, directives, laws, and codes | The system should validate all exchanged data against applicable regulatory standards, directives, laws, and codes. | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_52 | Regulatory Compliance Enforcement | The system should enforce compliance validation before data is accepted or shared within the TwinEU system. | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_53 | Secure and Transparent Data Exchange with CIM in two SGAM Layers | The system must enable secure and transparent data exchange using CIM standard definitions within the Communication and Information layers of SGAM. | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_54 | CIM-Based Interoperability for SGAM Communication and Information Layers | The system should implement the CIM standard in the Communication and Information layers of the SGAM communication model to | N/A (Not strictly related with TwinEU DT Federator) |

| | | | |
|---|---|---|---|
| | | ensure interoperability and standardized data exchange. | |
| TwinEU_GFUR_55 | Compliance Validation | The system could provide compliance validation for all TwinEU components. | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_56 | Monitoring of Regulatory Changes | The system must track regulatory changes at EU, national, and regional levels. | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_57 | Secure and Transparent Data Exchange | The system must implement access control mechanisms for regulatory data, ensuring only authorized entities can access or modify it. | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_58 | Alerts issuing | The system could issue alerts for any detected non-compliance in data exchanges. | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_59 | Report Generation | The system could generate periodic compliance reports detailing regulatory adherence and potential violations. The system could allow authorized users to export compliance reports in standard formats (e.g., PDF, CSV, XML). | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_60 | Logging Activity | The system could log all compliance validation activities for audit purposes. The system could ensure that all compliance-related data exchanges are logged and auditable. | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_61 | Grid Resilience Simulation | The system should provide simulation tools to assess the resilience of the energy grid under normal and abnormal conditions. | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_62 | Grid Stability Analysis for RES and DER Integration | The system must detect and analyse grid stability issues caused by the integration of RES and DER. | N/A (Not strictly related with TwinEU DT Federator) |

| | | | |
|---|---|---|---|
| TwinEU_GFUR_63 | Proactive Grid Planning | The system should support decision-making by grid operators through predictive analytics for grid stability. | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_64 | Scenario-Based Strategic Planning for Infrastructure Expansion | The system should enable scenario-based strategic planning for infrastructure expansion and adaptation. | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_65 | Abnormal Conditions and Disruptions Modeling | The system should detect and model potential physical disruptions in the energy grid. The system should provide response strategies for mitigating abnormal conditions. | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_66 | Infrastructure Bottleneck Identification | The system shall identify grid infrastructure bottlenecks that impact the efficient transmission and distribution of energy. | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_67 | Large-Scale Integration of RES and DER | The system should evaluate the impact of RES and DER on grid stability and adjust control mechanisms accordingly. | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_68 | Pan-European Energy Market Scenario Modeling | The system should model interconnected energy markets across Europe to simulate cross-border energy exchanges, market trends, and impact of regulatory frameworks. | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_69 | Report Generation | The system could generate reports on potential vulnerabilities in the grid infrastructure. Furthermore, the system should generate dynamic reports to support real-time and historical analysis. | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_70 | Wanning System | The system could provide early warnings for grid disturbances and potential failures. | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_71 | High-quality Real-time Data Visualization | The system should support high-quality real-time visualization of Digital Twin (DT) data, allowing users to possibly | N/A (Not strictly related |

| | | | |
|---|---|---|---|
| | | view and analyze live grid performance. | with TwinEU DT Federator) |
| TwinEU_GFUR_72 | Multiuser XR Collaboration | The system should allow multiple users to interact and collaborate within the same Extended Reality (XR) environment in real time. | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_73 | Unity3D-Based Plugin Integration | The system must integrate a Unity3D-based plugin to streamline user interaction with DT data and enhance XR visualization capabilities. | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_74 | Immersive XR Toolkit | The system should provide pre-built components and building blocks to facilitate rapid creation of XR applications by developers in an easy way. The system should provide reusable building blocks to enable rapid development of XR applications | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_75 | Advanced VR Visualization Features | The system could support Virtual Reality (VR) visualization for immersive energy infrastructure simulations and interactive visualization of grid anomalies, bottlenecks, and infrastructure risks. | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_76 | Immersive User Interface (UI) | The system should provide an intuitive XR user interface with easy navigation, tooltips, and context-sensitive controls. | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_77 | Security | The system should enforce secure authentication and authorization mechanisms to protect sensitive energy data and simulation parameters. | N/A (Not strictly related with TwinEU DT Federator) |
| TwinEU_GFUR_78 | Full IoT Edge-Network Control | The system will not provide detailed edge device control | N/A (Not strictly related with TwinEU DT Federator) |

## 3.3 Coordination with demos and data sharing strategies methodological approach

Part of the activities of WP4 and an integral part of the design and implementation of the TwinEU Open Data space is the process of demo coordination in the context of data sharing strategies based on the T4.3 activities. The main idea behind this activity is to provide input to Task 4.6 for the successful preparation and implementation of the pan-European scenarios. Based on this i) the connection and interfaces among the different local DTs were defined and evaluated (initial information in current document) and ii) a methodological approach was established on how we can align the pan-EU scenarios with technological enablers (Data space) in order to provide a complete validation scenarios to T4.6 “Validation of platform & implementation of pan-European scenarios”.

The 1st methodological step was the creation of the Pan-European scenarios chain of activities which aligned the efforts between WP2, WP3 & WP4 as shown in the Figure 10 below.

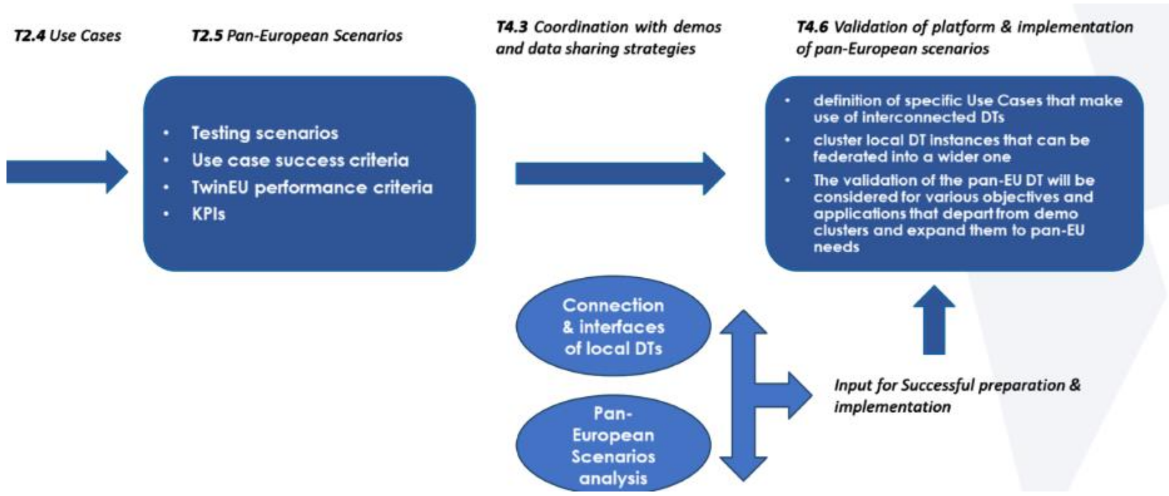



*Figure 10: Pan-European scenario activities chain*

Through this chain of activities, T4.3 efforts along with T4.5 produced the following results which were used in the design of the TwinEU Open Data Space:

- Analysis of the pilot use cases focusing on the pan-EU scenarios
- Support the base connector adaptation based on the needs of the pilot use cases
- Support the process of the federated Digital Twin concept definition
- First preparatory cycle of coordination with demos regarding the TwinEU Middleware and data sharing strategies

The 2nd methodological step is the creation of the pan-EU scenarios matrix template which will align the pan-EU scenarios with TwinEU Data space technological enablers and success criteria. The template is presented below using the 1st scenario as an example:

| A. Scenario: The grid hosting capacity map | | | | | | |
|---|---|---|---|---|---|---|
| **Step** | **Step Description** | **Technological Enablers** | **Functional Specification** | **Succes Criteria** | **Validation** | **Comments** |
| A1 | A1.Description of Pan EU Activity step | A1.TE1.Name | TE1 Functionality in the PanEU senario context | Expected Outcome | Pass/Fail | |
| | | A1.TE2.Name | TE2 Functionality in the PanEU scenario context | Expected Outcome | Pass/Fail | |
| A2 | | | | | | |
| A3 | | | | | | |

*Figure 11: Validation Scenario template*

- **Step:** The pan-EU scenario step (considering that the pan-EU scenario can be described a series of subsequent steps: A1-A2 etc).
- **Step Description:** The activity description of the step through a business perspective.
- **Technological Enablers:** The specific TE which is connected with the actual pan-EU scenario activity step.
- **TE Functional Specification:** The technological enabler functional description.
- **Success criteria:** Description of the expected outcome which denotes the successful operation as expected by the user.
- **Validation result:** Pass/Fail depending on the outcome.
- **Comments:** Potential comments regarding the specific user activity and the technological enabler behaviour.

The completed filled-in matrix (a major activity during the upcoming period) will form the operational guide which will be used by T4.6 for the TwinEU Open Data space validation process.

# 4 TwinEU Data Space Implementation

## 4.1 Introduction and main concepts

As already introduced in the Ch. 3.1, TwinEU DT Federator is the TwinEU proposed solution for implementing a federated ecosystem for Digital Twin in the energy sector at pan-European level.

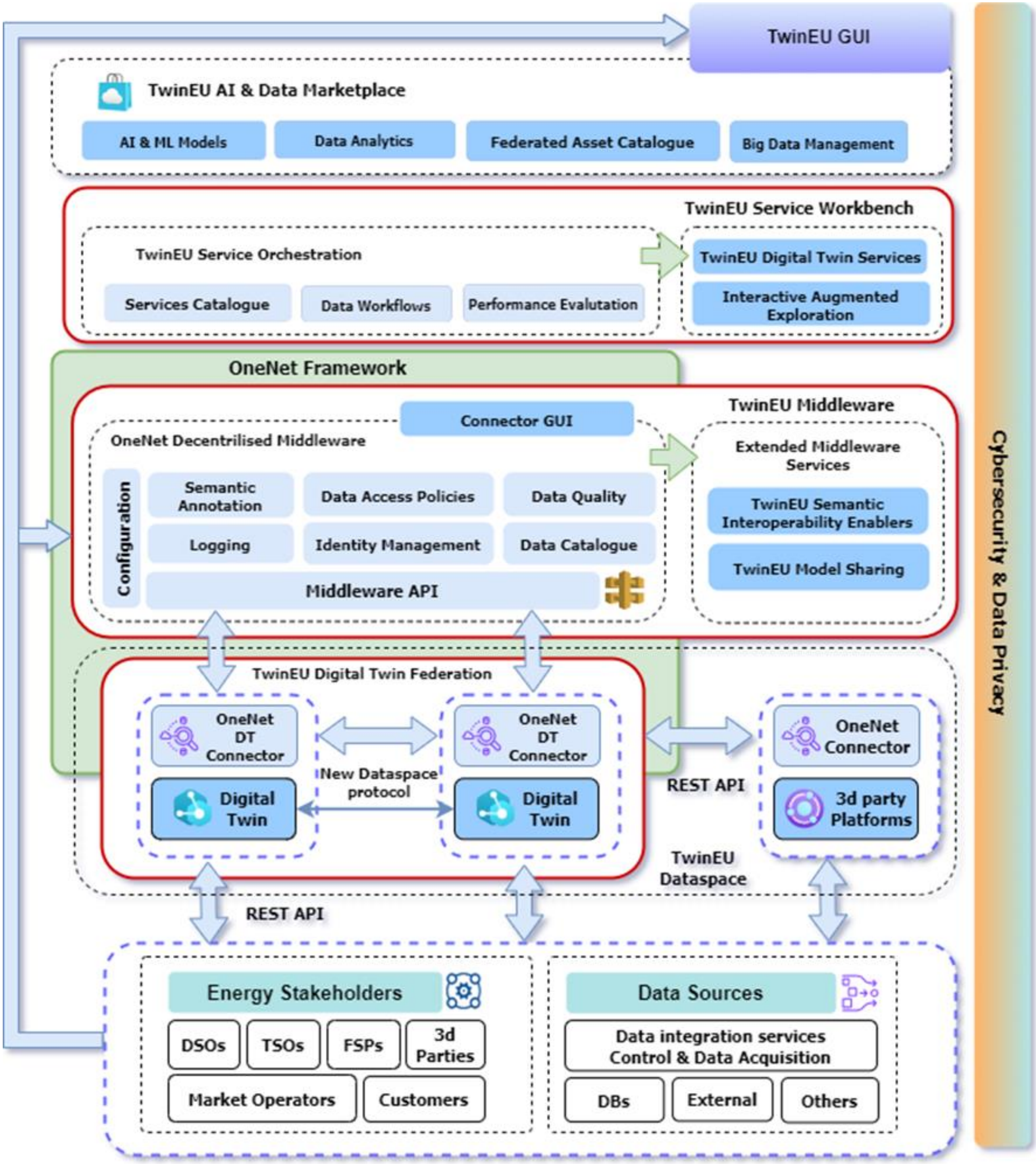



*Figure 12: TwinEU Reference Architecture*

As shown in Figure 12, the TwinEU Digital Twin Federation is a part of the TwinEU overall architecture, in charge of integrating existing or new Digital Twins, leveraging the **Data Space** concept for supporting an interoperable and secure Digital Twin Federation with an easy and effective integration of various data sources and infrastructures.

The dataspace-enabled structure for data and model sharing is the pivotal element of the TwinEU architecture. It involves the adaptation of the Energy Data Space to the specific context of TwinEU

Digital Twin ecosystem, creating a trusted and sovereignty-preserving layer for data and model exchange.

The solution leverages on the most relevant initiatives and architectures described in the previous chapters (2.2) and in particular extends and improves the OneNet Framework, a **Data Space Framework tailored for the Energy domain,** which supports the standardized data exchange among energy stakeholders at any level.

As already described in the analysis of the projects, in Ch. 2.3.1, it includes two main components: the **OneNet Middleware and the OneNet Connector.**

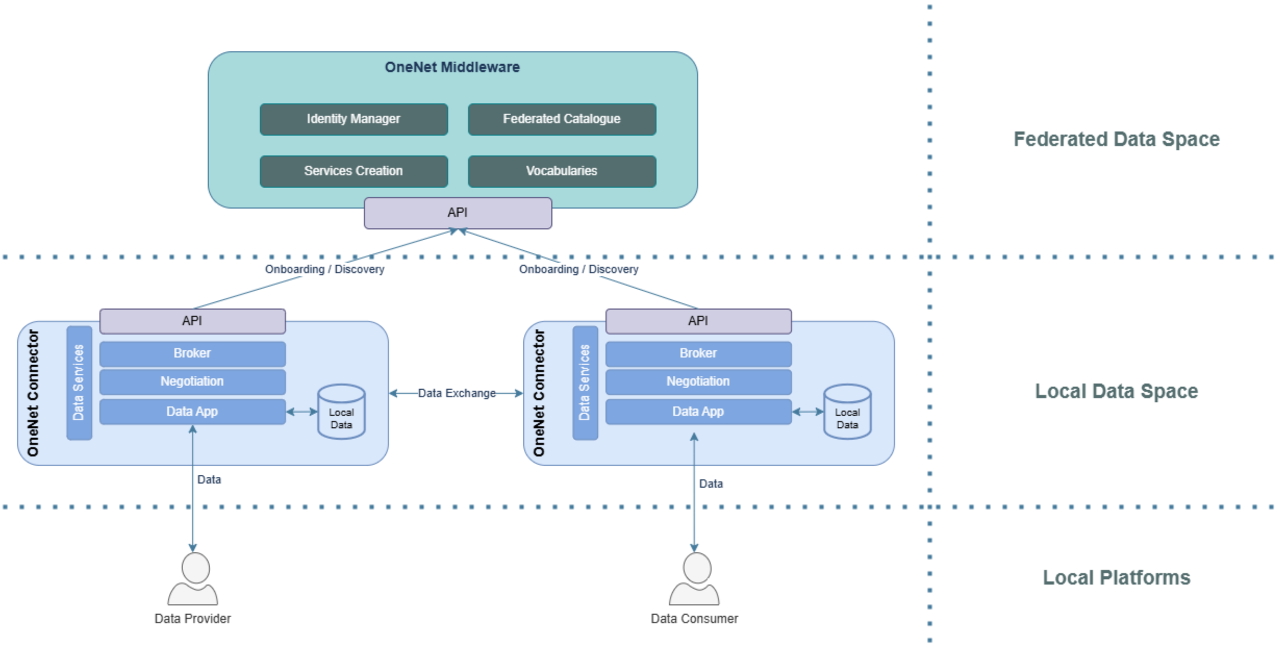



*Figure 13: OneNet Middleware and Connector - Implementation of a Federated Data Space*

As shown in Figure 13, The OneNet Middleware stands on top of the common infrastructure, ensuring secure and standardized communication between assets, systems, data sources, models, and energy stakeholders. This middleware facilitates seamless, efficient, and transparent sharing of data across the entire ecosystem, creating the concept of **Federated Data Space.**

The OneNet Connector implements the core features for a completely decentralized end-to-end data exchange while maintaining full control over data access and usage. In addition, it communicates with OneNet Middleware for onboarding and data service discovery.

The **OneNet Middleware and Connector were extended for supporting DTs Federation**, keeping its core characteristics and supporting DT data and model exchange, including new communication protocols, vocabularies and standards.

## 4.2 TwinEU DT Federator

Enabling a federated ecosystem for DTs in the energy sector is a complex task, but the TwinEU project has a clear vision of it: implementing a Digital Twin Federator with the main goals:

- Integrating heterogeneous DT systems
- Orchestrating services for data and models sharing
- Incorporating real-world data to enhance simulation and analysis capabilities

To achieve these goals, the TwinEU DT Federator leverages the Data Space concept for supporting an interoperable and secure DT Federation with easy and effective integration of various data sources and infrastructures. A dataspace-enabled structure for data and model sharing is a pivotal element of the TwinEU architecture. It adapts the Data Space concept to the specific context of TwinEU DT ecosystem, creating a trusted, sovereignty-preserving layer for data and DT model exchange.

### 4.2.1 Architecture

The TwinEU DT Federator extends and improves the **OneNet Data Space Framework**, a Data Space solution tailored for the Energy domain, which support the standardized data exchange among energy stakeholders at any level.

From the architectural point of view, the main changes are related to the integration of Data Space Protocol, which impact in the main interaction between the Connector and Middleware as well as the connector-to-connector communications.

Figure 14 below shows the architecture of the TwinEU DT Federator, including:

- **TwinEU Middleware,** adapted and configured from the OneNet Middleware,
- **TwinEU Connector,** completely revolutionized for integrating the Data Space Protocol specifications.

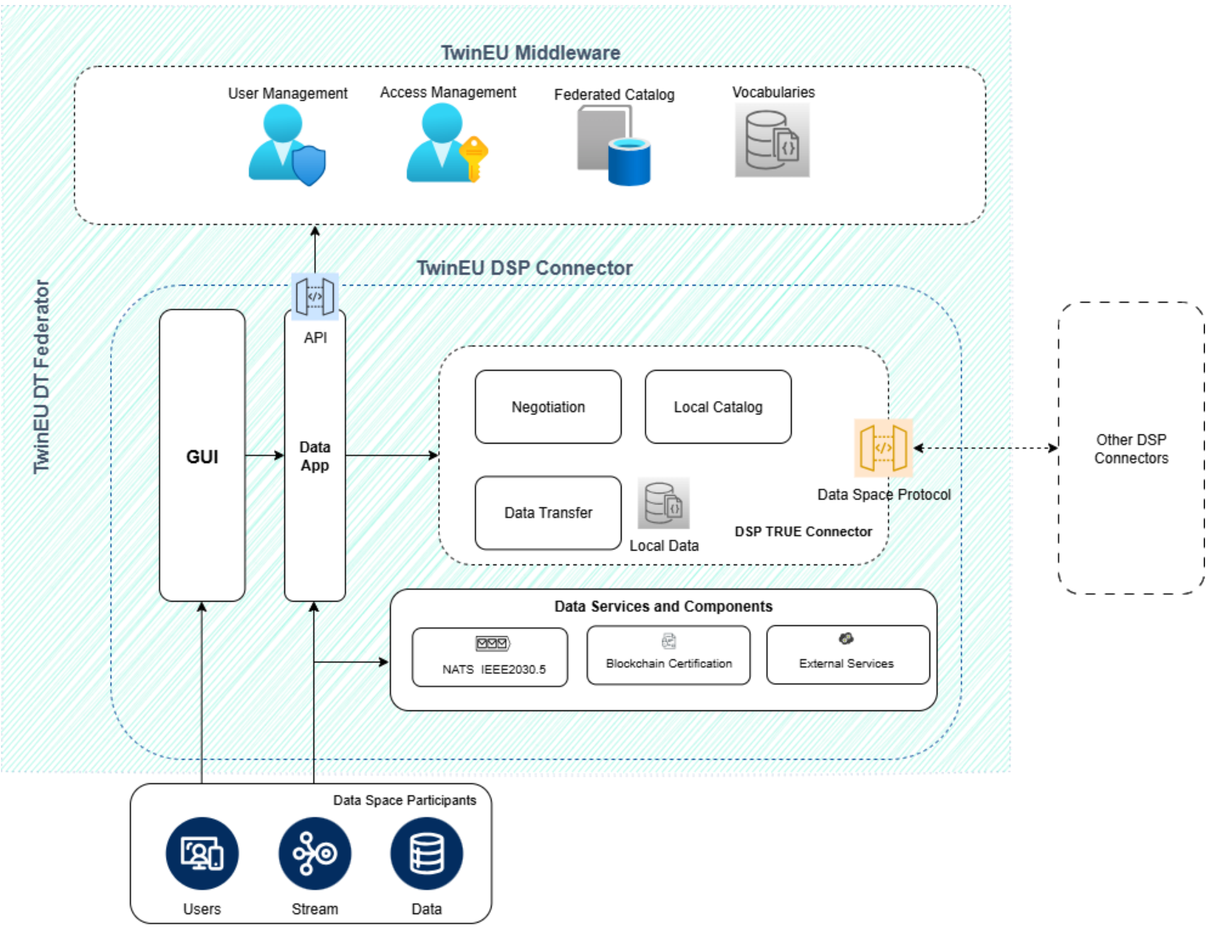



*Figure 14: TwinEU DT Federator Architecture*

Within the TwinEU Connector several components were adapted or developed from scratch**:**

- **DSP Data App,** new module compliant with DSP specifications able to integrate TwinEU Middleware and external platforms/applications.
- **DSP TRUE Connector,** new version of the TRUE Connector compliant with DSP 2024.1.
- **Data Service and Components layer,** which additional services for data integration and logging.
- **GUI,** adapted and extended for the latest version of OneNet Connector.

A TwinEU Middleware instance will be available for the partner of the consortium in a free-to-use version and an open-source release will be investigated by the end of the project.

The TwinEU DSP Connector will be fully available for the partner of the consortium through the TwinEU Github repo (see Ch. 5.4), in which the packaged version and source code of the first version was already published. An open-source version in a public repository will be available by the end of the project.

In the following chapters, additional details about the TwinEU DT Federator components are provided.

### 4.2.2 TwinEU Middleware

The TwinEU Middleware is the administration component of a federated Data Space system that performs the distributed data exchange within the TwinEU Framework. Specifically, the TwinEU Middleware ensures management, cohesion, and collaboration among the distributed IDSA Connectors that serve the data exchange between Digital Twins. Finally, the TwinEU Middleware operates as a provider of a list of the available Data Offerings services among the TwinEU participants, creating a Data Space environment for the provision, discovery and consumption Data Offering services, enriched with data access, role management and user settings features.

The TwinEU Middleware is responsible for the following functions:

- Enables the setup and integration of new Connectors tailored to the needs of TwinEU participants.
- Supports interoperability defining the principles, methodologies and standards that should be followed by the TwinEU stakeholders.
- Supports federated authentication and authorization, allowing participants to retain control over their data and identities while enabling secure, role-based access across services.
- Facilitates the orchestration of distributed services and allows users to discover and consume services offered by other members of the TwinEU.
- Provides collection of metrics, logs, and traces from all participating systems for performance monitoring and compliance auditing.

#### 4.2.2.1 Semantic Interoperability and Annotation features

The Semantic Interoperability service in the TwinEU Middleware is performed in a two-layer structure. The first layer includes a predefined categorization of the Data Offerings services that are available within the TwinEU framework. The user can select a defined set of Data Offering categories combined with a more specific Business Object codes, during the configuration and provision of a new Data Offering service. This process enables a high-level interoperability type regarding the Data offering services and offers easy navigation and consumption of Data Offering services.

The second layer of Interoperability includes a set of fields that the user should complete regarding the semantic structure and annotation of the data that will provide. Specifically, every user that creates a Data Offering service through TwinEU Middleware can provide the file schema of the data, a file schema example, the format of the file and a description of the data. This information is very important for the TwinEU users that will consume the data comprehending the structure, meaning and context of the information.

The information and annotation that the user can provide for its data includes:

- Ontology Mapping that associates data with standardized concepts and classes to unify semantics across TwinEU framework.
- Metadata Enrichment that augments datasets with contextual metadata.
- Semantic Harmonization that ensures that data from different Digital Twins can be understood and processed uniformly by the TwinEU participants.

Utilizing the aforementioned information the data consumers can easily utilize and process the data or use tools to transform the data to the desired structure and format to be compatible with their applications.

Finally, the TwinEU Middleware, since it is perfectly aligned with the Data Space standards and architecture, integrates by default interoperability features that are incorporated by default by the Data Space concept. These features include semantic interoperability standards (Shared Data Models and Ontologies, Vocabulary Alignment) and syntactic interoperability standards (Standardized Formats and API Specifications).

The interoperability components and features that are mentioned above play a pivotal role in ensuring data interoperability and semantic consistency across the federated network of Digital Twins within the TwinEU Middleware/platform and generally for the TwinEU framework. As Digital Twins exchange diverse datasets from heterogeneous sources it becomes critical to annotate, define, and enrich this data with an understandable meaning. Hence, the TwinEU Middleware facilitates seamless integration of data and models across Digital Twins, enables the service orchestration using annotations to match services and models with relevant data inputs and forms the foundation for interoperability with external platforms.

#### 4.2.2.2 Data Access Policy features

In the TwinEU platform where data sovereignty, interoperability, and secure exchange are foundational principles, Data Access Policies play a central role in governing how data can be shared, accessed, and used across the federated Digital Twin ecosystem. These policies are essential for enabling trusted, controlled, and sovereign-preserving data exchange, aligned with the IDSA Reference Architecture Model (IDS-RAM).

The Data Access Policy within the TwinEU platform defines the conditions, constraints, and permissions under which data assets can be accessed by other actors in the ecosystem. They ensure that data owners retain control over how their data is consumed and by whom, while also allowing for seamless and scalable integration of heterogeneous systems.

The Data access policy and sovereignty combined with the dynamic management of them regarding the TwinEU data access and exchanges, happens in multilayer way. First, every user of the TwinEU Middleware is an authenticated TwinEU and Data Space participant that has a specific identity and attributes and is constantly authenticated with the proper techniques when it is using Middleware.

Moreover, when the Data provider creates a Data offering service through the Middleware, optionally can define specific rules and obligations for the data consumers and the usage of data. Specifically, the Data provider can set specific rules regarding the Country Restrictions, namely the user's countries that can have access to the data, Role Restrictions, namely the specific predefined roles that are created within TwinEU framework that can have access to the data and the Market Restrictions, namely specific attributes of the market that will be used for the data provision that should be prerequisite to access the data. Finally, the user can define a specific time duration that the data will be available for the data consumers.

Additionally, the Data access policy is implemented by the TwinEU Middleware using a robust handshake subscription method that provides security and data sovereignty. This process requires specific steps consequently by the Data consumer and the Data provider. These steps are the following:

1. A subscription request from the Data Consumer to the provider of the Data offering service.
2. The response to the specific subscription request by the Data offering service provider (approval or denial)
3. If there is an approval the Subscription between them is certified by the TwinEU Middleware.

The Data Offering service provider can cancel the subscription between every user that has a subscription or deleting the whole Data offering service.

Finally, the TwinEU Middleware, since it is aligned with IDS standards, integrates by default some specific operations regarding data access and security. In the context of IDSA, policies are defined and attached to Contract Offers and Contract Agreements between Connectors, when this information is stored in the back end of the Middleware. The TwinEU Middleware's backend also stores all the policy types of the IDS network. Specifically, the Policy Enforcement Point (PEP) that ensures that only requests that comply with agreed policies can retrieve or use data, the Policy Decision Point (PDP) that evaluates the access request against the stored policies and provides a decision (e.g., Permit, Deny, Obligate) and the Policy Information Point (PIP) that examines sources with additional contextual information needed for decision-making, such as user roles, time constraints, or geographic location. Concluding the TwinEU Middleware embeds PEP, PDP and PIP allowing integration with IDS Data Sovereignty Services (such as the Clearing House, Identity Provider, and Vocabulary Provider).

#### 4.2.2.3 Data Logging features

In the TwinEU federated digital twin ecosystem, where data traceability, sovereignty, and compliance are critical, a Data Logging system serves as an essential foundation for enabling trust, accountability, and verifiability of data transactions. The TwinEU Middleware is in line with the IDS Reference Architecture Model supporting a compatible logging system. This logging system ensures that all data-related events are transparently recorded across the ecosystem while respecting the decentralized and sovereign nature of data exchange.

The logging system that is described above is supported by the TwinEU Middleware both through storage and management and through the provision of the appropriate information whenever is needed. The components and processes that compose this system are the following:

- A logging engine, which collects logs from multiple services across the middleware (e.g., data access and policy evaluation).
- Log Storage and Archival Module, which incorporates encryption to ensure confidentiality.

- Access Control and Identity Logging, which captures information about who accessed what data, when, from where, and under what credentials. These components should be closely tied to the Identity Provider and Access Control components, which are integrated to the Connector and defined by the IDSA, supporting trusted identity verification.
- Interface with the IDSA Clearing House, which plays a crucial role in ensuring that log records are verifiable and archived in a neutral, trusted environment.
- Audit & Reporting Service, which supports automated audits, notifications of non-compliance, and reporting dashboards for operational oversight.

#### 4.2.2.4 User settings and Connector Configuration

The TwinEU Middleware encompasses a group of heterogeneous participants that manages a wide variety of data entities, Digital Twins, and IDS Connectors. These sets of entities should be configured either by the administrator or by its owner to be certified and operational parts of the TwinEU Middleware.

The TwinEU participants configuration is processed through a request to the Middleware administrator that creates and configures a user profile with the id, username, password and mail of the user, followed by default user settings. Since the TwinEU Middleware follows the IDS Architecture principles, it should support a Role-Based Access Control (RBAC) component and connect with IDS Identity Provider and Dynamic Attribute Provisioning Service (DAPS) for trusted credential verification, to define the roles and identity of the user. Under these prerequisites, the user can have access to the Middleware and customize the settings regarding the access policy preferences of its data, the rules of its subscriptions and the settings of its Connectors.

Regarding the configuration and management of Data Offering services, Digital Twin data, Digital Twin models and the Digital Twins as a distinct entity, the TwinEU Middleware implementation will include future features for customized options. Additionally, the owner of the entities, mentioned above, would be able to request for customized features to secure and manage of them. The goal of the TwinEU Middleware is to provide all the necessary tools to support a functional federated pan-European Digital Twin and Data ecosystem and will offer the creative cooperation, secure interaction and interoperability of participants and Digital Twins. This configuration in combination with IDS alliance is especially critical in heterogeneous ecosystems such as TwinEU, where users, digital twins, data services, and energy systems vary in structure, format, and access control policies.

Regarding the configuration of data, as it has been mentioned the data provider can define usage conditions, rights, restrictions, and purposes as it is presented in Chapter 4.2.2.2 and define who can access which data under what conditions. Additionally, the user can define technical and interoperability configuration (including rules or suggestions) as it is presented in Chapter 4.2.2.1. Once the Data offering services and respectively the data, are configured from technical, interoperable and policy aspect, the Data Offerings become parts of the Data Offering Catalogue and Endpoint Management components, that configure how datasets are exposed to, discovered and consumed by consumers (e.g., via RESTful APIs, SPARQL endpoints). This component should operate with IDS Metadata Brokers and Connectors to register, publish, and manage data offerings in a discoverable way in an IDS compatible way.

Finally, a core configuration that needs to be done by the TwinEU Middleware is the Connector Configuration Interface (IDSA Integration). This provides a configuration dashboard or interface to define the proper operation of IDS Connectors, with specific instructions of the ports that need to be used. As a future additional feature, the TwinEU Middleware could include plug-ins or adapters for other domain-specific energy connectors.

#### 4.2.2.5 Data Offering services Catalogue and the Integration with IDSA Architecture

In the context of the TwinEU federated Digital Twin Middleware, Data Offering services Catalogue components serve as foundational elements for enabling structured, discoverable, and controlled access to data assets across the TwinEU ecosystem. They provide the mechanisms by which data providers can describe, publish, and manage their datasets, while data consumers can search, understand, and request access to the data, all in compliance with data sovereignty, trust, and interoperability principles. Aligned with the IDSA Reference Architecture Model, these components are crucial for creating a harmonized environment where energy-related participants and subsequently digital twins can interact securely with a predefined structure and logic.

The core functionality of the Data Offering Service Catalogue is to describe data assets using standardized metadata schemas, expose data assets to the ecosystem in a searchable and interpretable manner and enable secure discovery, filtering, and matchmaking between providers and consumers.

The principal component of the Data Offering Service Catalogue is the Metadata Management Layer. This layer supports the creation, storage, and validation of metadata describing each dataset or service. This process regarding the first version of the Middleware is processed by the administrator of the platform. This layer of metadata should be compliant with the IDS Information Model and semantic standards. The Metadata Management Layer is completely complementary with a Data Offering registration and publishing Interface. Through this interface the user can register and publish its Data Offering services defining some metadata fields that are required. These fields are the following:

- “Category”, which is the high-level energy domain of the Data Offering service (from a predefined set of categories).
- “Service type Code”, which is a specific code that characterizes every Data Offering service.
- “Business Object”, which is a more detailed description of the business object of the Data Offering service (from a predefined set of Business Objects).
- “Description”, which is an analytic text about the content of the Data Offering.
- “Owner” of the Data Offering service namely the participant that has published the Data Offering.
- “Status”, which presents if the Data offering service is active and available for the TwinEU participants.
- “Created on”, which stands for the date and time that the Data Offering has been published.
- “Title”, which is set by the publisher of the Data Offering.
- Profile format, which presents semantically the structure and type of the Dataset that is provided.

As we have mentioned before, in a federated environment like TwinEU, where multiple independent actors (TSOs, DSOs, market participants) expose datasets with varying levels of abstraction and access rights, the data catalogue becomes the central registry for organizing and managing the availability and usability of shared information. The TwinEU Middleware should associate each catalogue entry with specific usage policies and contracts, leveraging IDSA Usage Control mechanisms. Additionally, the TwinEU Middleware should provide mechanisms to ensure that the discoverability is separated from accessibility, assets may be visible in the catalogue but only accessible after policy-compliant subscription and agreement.

Additionally, the Data Offering service Catalogue of the Middleware supports a search and Discovery engine. This engine facilitates keyword-based and semantic search over registered assets. Moreover, this engine supports filtering by category, business object and Data offering service code fully integrated with the IDS Metadata Broker, which acts as a federated discovery point across different Data Space participants.

Finally, the TwinEU Middleware supports components for data source federation and interconnection that are very important for the progression of the Data Offering service Catalogue. Specifically, it already supports and should add more relevant features in the future about the management, importing and sharing of catalogues distributed across different TwinEU tools maintaining a federate view. Additionally, the final version of the Middleware should be able to provide RESTful APIs and user-friendly interfaces for developers and non-technical users to explore and consume the catalogue and integration points for external platforms (e.g., marketplaces, digital twin dashboards).

### 4.2.3 TwinEU DSP Connector

The TwinEU DSP Connector is the core part of the TwinEU DT Federator. It extends the OneNet Connector with additional features and adaptation for supporting the Data Space Protocol and the DT Federation. Figure 15 below shows the TwinEU DSP Connector architecture.

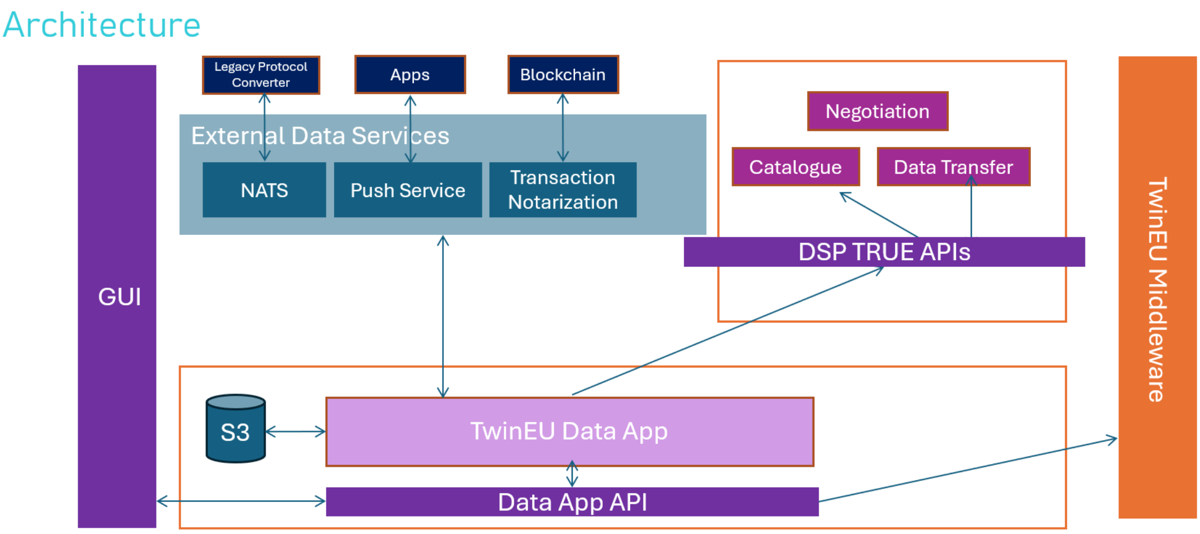



*Figure 15: TwinEU DSP Connector*

#### 4.2.3.1 Data Space Protocol (DSP)

TwinEU DSP Connector implements the Data Space Protocol (DSP), version v2024.1, to enable standardized, interoperable data exchange across the ecosystem. The DSP defines a common framework and semantics for describing, requesting, and delivering data in a trusted and secure way, aligned with data space principles such as sovereignty and governance. The Connector exposes DSP-compliant interfaces that allow participants to publish their data offerings, negotiate access, and consume datasets through well-defined interactions. This ensures that both human and automated consumers, as well as third-party services, can seamlessly discover and access data in accordance with agreed policies and contracts, fostering an open yet controlled data sharing environment. By leveraging DSP v2024.1, the TwinEU DSP Connector integrates the system into the broader data space ecosystem, making data streams available as standardized resources and ensuring full interoperability with other certified connectors that support DSP.

The DSP TRUE Connector [37] is the evolution of the original TRUE Connector, redesigned to comply with the Data Space Protocol (DSP) standard and to support interoperable, sovereign data exchange in federated environments. It encapsulates the core functionalities required to operate as a participant in a data space, organized into a set of specialized modules that together implement the DSP specifications. The catalogue module handles the logic for processing and exposing the participant's data catalogue, enabling discovery and indexing of available resources. The negotiation module manages the contract negotiation process, orchestrating the exchange of offers and agreements between participants to establish usage policies before data transfer begins. The data-transfer module is responsible for maintaining and monitoring the secure, policy-compliant transfer of data assets once an agreement has been reached. To facilitate deployment and execution, the connector module serves as the application entry point, coordinating the initialization and startup of the various components. Finally, a tools module provides shared utilities and helper functions that support the operations of the other modules, promoting modularity and code reuse. Together, these modules form a robust, extensible implementation of the DSP standard, enabling the DSP TRUE Connector to seamlessly integrate into the federated digital twin ecosystem.

#### 4.2.3.2 Push data exchange mechanism (with External API service call)

*Overcoming the Limits of Pull-Based Data Exchange in Grid Operations*

In a data space context, the common interaction pattern—where data consumers actively request information from providers when needed—is sufficient for many energy sector applications. However, this model shows its limitations in scenarios that require immediate, real-time data exchange. For instance, in real-time grid balancing, energy providers and grid operators must react within seconds to fluctuations in supply or demand. Situations like an unexpected drop in renewable generation (e.g., wind or solar) or a sudden surge in consumption call for the instant transmission of critical parameters such as frequency deviations or load changes. In these cases, a push-based data delivery mechanism ensures that such vital information is sent to relevant stakeholders as soon as it is generated. This enables prompt, often automated, responses—such as activating storage systems, adjusting demand, or redistributing energy resources. Relying solely on pull-based interactions in these time-sensitive scenarios could introduce delays, reduce system efficiency, raise operational costs, and risk compromising grid reliability.

*Limitations of Pull-Based Data Access in Dynamic Energy Scenarios*

Traditional data access models rely on consumers periodically querying providers to retrieve information—a pull-based approach. While sufficient in static or less time-sensitive contexts, this method can introduce significant delays and hinder the timely processing of data. In highly dynamic environments, such as real-time energy consumption tracking or smart grid operations, where decisions must be made within seconds, this model falls short. The inability to receive data as soon as it becomes available can compromise system responsiveness and reduce overall efficiency.

*Advantages of Push-Based Data Exchange and Support for Event-Driven Integration*

The push-based approach addresses the limitations of traditional pull mechanisms by delivering data to consumers as soon as it becomes available—eliminating the need for explicit requests. This paradigm shift enables faster, more responsive, and more efficient data integration across energy systems. Key benefits include:

- Seamless Integration with External Services: Simplifies connections with third-party platforms and systems, enabling automation and real-time interactions.
- (Near) Real-Time Data Delivery: Ensures immediate transmission of data at the moment it is generated—critical for time-sensitive operations such as grid balancing and demand-response.
- Lower Latency: Minimizes delays between data creation and consumption, enhancing system responsiveness and supporting real-time analytics.
- More Efficient Resource Utilization: Eliminates the need for constant polling, reducing computational overhead and network traffic on the consumer side.
- Support for Event-Driven Scenarios: Ideal for use cases that require immediate action based on specific events—such as equipment malfunctions, sudden demand surges, or variable renewable generation.
- Enabling Proactive Decision-Making: Provides stakeholders with timely alerts and notifications, empowering them to act quickly, anticipate issues, perform preventive maintenance, and optimize energy flows.

To further support advanced, event-driven use cases, a new push service feature has been introduced, allowing one or more data providers to actively push data to a service defined and hosted by an external service provider.

The process works as follows:

1. *Service Creation*: A service provider defines a new service by implementing a RESTful web service that adheres to a standard interface. The endpoint URL is registered via the Connector's Push Service creation interface.
2. *Subscription*: Data providers request to subscribe to the service. Once the request is approved, they can begin pushing data using the Connector, as with standard data exchanges.
3. *Data Loading*: During data upload, the configured REST endpoint (hosted by the service provider) is called synchronously. Both the data file and its associated metadata are transmitted, making the data immediately available on the provider's platform—even outside the Connector environment.
4. *Response Handling*: The system returns a response to the data provider, confirming successful delivery or indicating an error with a descriptive message and status code.

This mechanism enhances the flexibility and speed of data integration, offering a scalable foundation for modern, event-driven architectures within the energy domain.


*Figure 16: Push data scenario flow*

*Push Mechanism Workflow: Actors and Sequence*

The complete flow of the push-based data exchange mechanism can be represented through a sequence diagram involving two main types of participants:

- *Service Provider*: Defines and publishes the push-type service and consumes the incoming data.
- *Data Provider(s)*: One or more entities that upload data to the defined service.

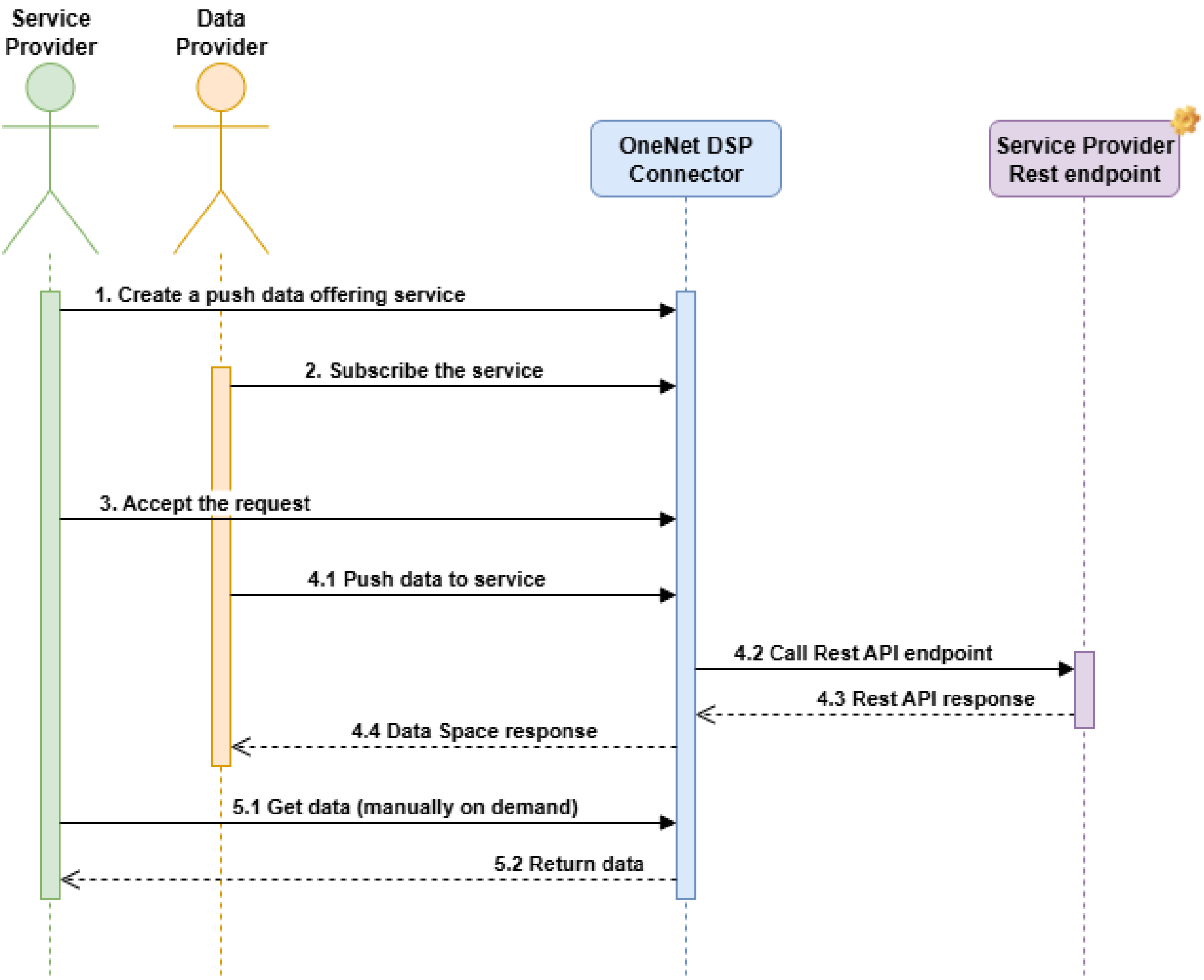



*Figure 17: Push mechanism Sequence Diagram flow*

Below is a step-by-step description of the interaction flow:

1. *Service Definition*: The Service Provider creates a new push-type offering, specifying a REST endpoint that will receive incoming data. This endpoint must implement a predefined interface supported by the Connector.
2. *Subscription Request*: A Data Provider interested in contributing data browses the list of available push-type services and submits a subscription request for the desired one.
3. *Request Approval*: The Service Provider reviews incoming subscription requests and either approves or rejects them based on predefined criteria or access policies.
4. *Data Upload*: Once approved, the Data Provider selects the service and uploads data through the Connector. During the upload process, the REST endpoint defined by the Service Provider is called synchronously. The request includes both the data file and its metadata. A response is returned to the provider, indicating whether the upload succeeded (OK) or failed (FAILED), along with optional error codes or messages.

5. *Data Consumption*: The Service Provider receives the data in real time via the configured REST API. In addition to programmatic access, the provider can also explore the list of received data, inspect metadata, and download data files manually through the Connector's user interface or APIs.

### 4.2.3.3 Integration of NATS Protocol with the Data Space

#### 4.2.3.3.1 NATS as a Messaging Backbone for Data Spaces

NATS [38] is a high-performance, lightweight, open-source messaging system purpose-built for cloud-native and distributed environments. It leverages a publish/subscribe (pub/sub) communication model that is inherently scalable, resilient, and easy to manage. Thanks to its low latency, minimal resource footprint, and ability to process millions of messages per second, NATS has become a widely adopted solution in domains such as IoT, edge computing, and microservice-based architectures.

Within the context of a data space, where multiple independent stakeholders need to exchange data securely, reliably, and in near real-time, NATS offers a natural and effective foundation. Its decoupled pub/sub model supports flexible and dynamic data flows between producers and consumers, without requiring direct dependencies or tight integration. This design aligns seamlessly with the core principles of data sovereignty, interoperability, and modularity that define data spaces, making NATS an ideal choice for enabling scalable and policy-compliant data exchange.

#### 4.2.3.3.2 (near) Real-Time Data Exchange via NATS in the Data Space

A data space like the TwinEU DSP Connector is designed to enable controlled data exchange among diverse actors—including data providers, consumers, and intermediaries—while enforcing trust, governance, and access policies. However, many operational systems and devices in the energy domain (such as smart meters, DER controllers, and EV chargers) are already built to transmit real-time data using lightweight messaging protocols.

The integration of NATS with the DSP Connector effectively bridges this gap between real-time systems and governance-driven data spaces:

- Seamless Device Communication: Edge devices and systems can continue to publish telemetry and status updates to NATS topics, without requiring any knowledge of data space structures or protocols.
- Middleware Transformation: A middleware component within the data space can subscribe to selected NATS topics, perform necessary data transformations or filtering, and load the resulting data into configured data offering services exposed by the DSP Connector.
- Bidirectional Exchange: The data space can also send control signals or data back to the devices by publishing messages to specific NATS topics, which target the intended recipients on the edge.

This hybrid approach enables organizations to leverage their existing NATS-based infrastructure while gaining the added value of data governance, traceability, policy enforcement, and monetization offered by the data space architecture. It promotes interoperability without disrupting existing real-time data pipelines, aligning operational efficiency with strategic data management goals.

#### 4.2.3.3.3 Device-to-Dataspace Integration via NATS

In a device-to-dataspace integration scenario, edge devices or gateways publish data to specific NATS topics, while the data space ingests this information and makes it accessible to authorized consumers through managed data services.

Typical data flows include:

- Smart Meter Readings: Smart meters publish consumption data to topics such as meters.<id>.readings. The data space middleware subscribes to these topics, enriches the data with contextual metadata (e.g., location, timestamp), and loads it into a data offering service targeted at utilities or energy aggregators.
- DER Status Updates: Distributed energy resources—such as solar inverters, battery systems, or wind turbines—broadcast real-time status and output on topics like der.<type>.<id>.status. This data is ingested by the data space to support use cases like virtual power plant (VPP) optimization or grid balancing.
- Environmental Sensors: Sensors measuring air quality, temperature, or humidity send their readings via NATS. The data space aggregates and structures this information into data services for applications such as urban planning or energy efficiency monitoring.

In all these cases, the middleware layer within the data space is fully configurable. It defines which NATS topics to subscribe to, how to transform or enrich incoming data, and which data offering service the data should be routed to—ensuring a flexible, scalable, and standards-aligned integration of real-time data sources into the data space ecosystem.

##### 4.2.3.3.4 Dataspace-to-Device Integration via NATS

In the dataspace-to-device scenario, the direction of data flow is reversed: the data space publishes commands or information to specific NATS topics, while devices—or their associated controllers—subscribe and react to these messages in real time.

Example use cases include:

- Demand Response Signals: The data space publishes load reduction instructions to topics such as devices.<id>.commands during peak demand periods. Subscribed devices respond autonomously by adjusting their energy usage.
- Firmware Updates or Configuration Changes: New firmware versions or configuration parameters are sent through NATS topics, enabling a coordinated update process across large fleets of devices.
- Dynamic Price Signals: Real-time energy pricing is published to inform consumer behavior, either through smart home interfaces or automated control systems that optimize energy consumption based on cost.

This bidirectional integration enables responsive, governed interactions between the physical energy infrastructure and the data-driven services of the data space.

By integrating NATS with the TwinEU DSP Connector, stakeholders can harness NATS's lightweight, high-speed messaging capabilities to create seamless, two-way communication between devices, applications, and data services. This architecture supports real-time, configurable data flows in both directions—from devices into the data space, and from the data space back to devices—while preserving key principles such as data sovereignty, security, and interoperability that underpin the data space model.

***Architectural Overview: Data Space and NATS integration***

The architectural diagram below presents a data exchange ecosystem built upon the TwinEU DSP Connector, integrated with a NATS messaging server to enable real-time, scalable communication across a variety of participants and systems.

Key components of the architecture include:

- ***IoT Devices***: Multiple edge devices operate as both data producers and consumers, publishing telemetry data or receiving control signals via NATS topics.
- ***NATS Messaging Core***: A single instance of the NATS server serves as the messaging backbone, providing low-latency, high-throughput communication between all connected entities.
- ***NATS Middleware***: This component acts as a bridge between the NATS environment and the Data Space Protocol (DSP) platform, transforming and routing data streams into standardized DSP-compliant data services.
- ***Human Interfaces***: Data providers and consumers can access the system through graphical interfaces or APIs, enabling them to contribute new data, explore available services, and retrieve relevant datasets.
- ***Third-Party Services***: External systems are integrated as automated clients capable of publishing and subscribing to data streams. These services support added-value features such as advanced analytics, real-time monitoring, and interoperability with other platforms.

This architecture supports flexible, real-time, and governed data exchange across a heterogeneous ecosystem, aligning with the core principles of data sovereignty, interoperability, and scalability promoted by the TwinEU Data Space concept.

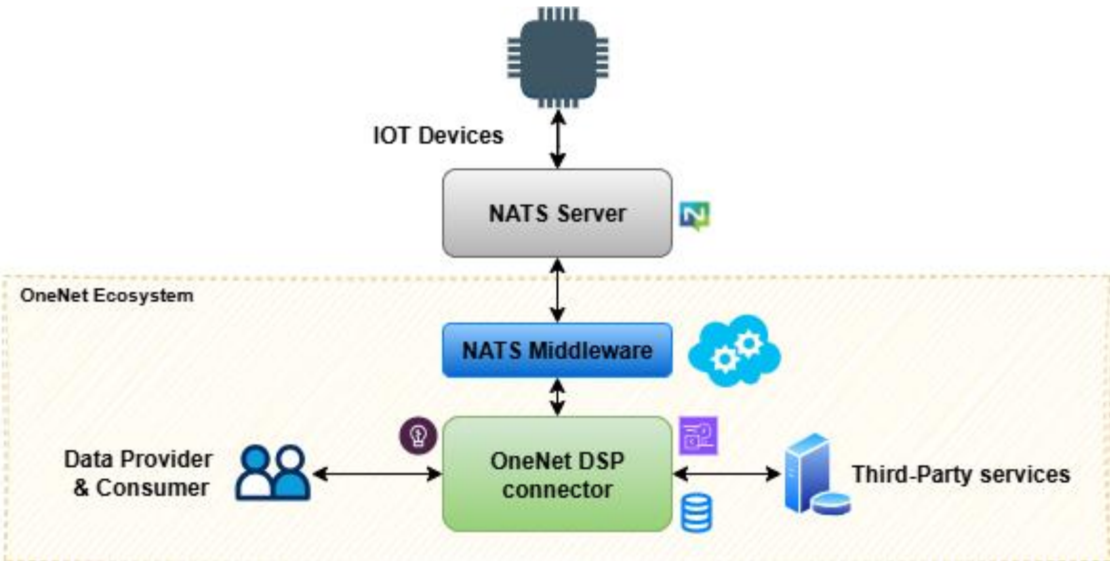



*Figure 18: Integration Diagram of TwinEU DSP Connector with NATS*

This architecture ensures seamless, real-time data flow among heterogeneous devices, systems, and users.

***Key considerations***

- ***Data Sharing***: The data space can control who has access to what data based on offered services and relative subscriptions. For instance, utilities can access specific data streams, while other stakeholders, like consumers, can view their own data.
- ***Data Sovereignty***: Implement policies to ensure that data shared in the data space adheres to sovereignty, privacy, and compliance rules.

***Example Flow***

1. ***IOT Device***: A solar panel or storage system sends energy production data to a NATS topic like `energy.production.solarpanel1`.
2. ***NATS Server***: NATS acts as pub/sub message broker and broadcasts the message to all subscribers to the `energy.production.solarpanel1` topic.
3. ***Data Space***: The data space subscribes to this topic, ingests the data, and publish them to one or more configured offered services.
4. ***Consumers***: A utility or grid operator, that subscribed these services, retrieves the data, enabling grid optimization.

# 5 First Release of TwinEU DT Federator

This chapter presents the first release of the TwinEU DT Federator which includes the first version of the TwinEU DSP Connector and an adapted version of the TwinEU Middleware, detailing its key features and implementation aspects.

This first release of the TwinEU DT Federator focused on implementing the Data Space Protocol for ensuring a high level of interoperability in the context of Energy Data Space, as well as on the evolution of services and UI of the Middleware and Connector.

All the Data Space Protocol (DSP) modules and interfaces were successfully integrated in the first version of TwinEU DSP Connector, with a dedicated UI and a new Data Application, which allow the integration of legacy systems.

In addition, an adaptation for the compatibility of Push mechanism and NATS protocol was performed both in the Connector GUI and in the TwinEU Middleware.

## 5.1 Functionalities

The first version of the TwinEU DT Federator included the new following features:

- Data Space Protocol integration (DSP v2024.1) [6]
- New version of TwinEU Middleware for DSP protocol compatibility.
- New version of Open Graphical User Interface.

### 5.1.1 Data Space Protocol

The first version of TwinEU DT Federator is compliant with the standard workflow implemented in the Data Space Protocol. This workflow can be structured into 5 steps:

1. Configuration (Onboarding)
2. Creation of Data Offering (Indexing)
3. Data Catalogue (Discovery)
4. Subscription and access management (Negotiation)
5. Provisioning and Consuming Data (Data Exchange)

Figure 19 provides a high-level sequence diagram of the five main phases between Data Provider and Data Consumer, showing the integration of the TwinEU DT Federator, including both the Middleware and the Connector.

**Configuration (Onboarding)**

This initial step consists of setting up the data space environment via the TwinEU DSP Connector. It includes configuring the necessary endpoints, defining participant identities and ensuring that all participants are properly onboarded into the Data Space. This step ensures that the data space is secure and ready for use.

**Creation of Data Offering (Indexing)**

In this step, Data Providers create and index their data offerings. This involves describing the data, setting metadata, and making it discoverable within the TwinEU Middleware. Indexing helps in organizing the data so that it can be easily searched and accessed by users through the Data Catalogue.

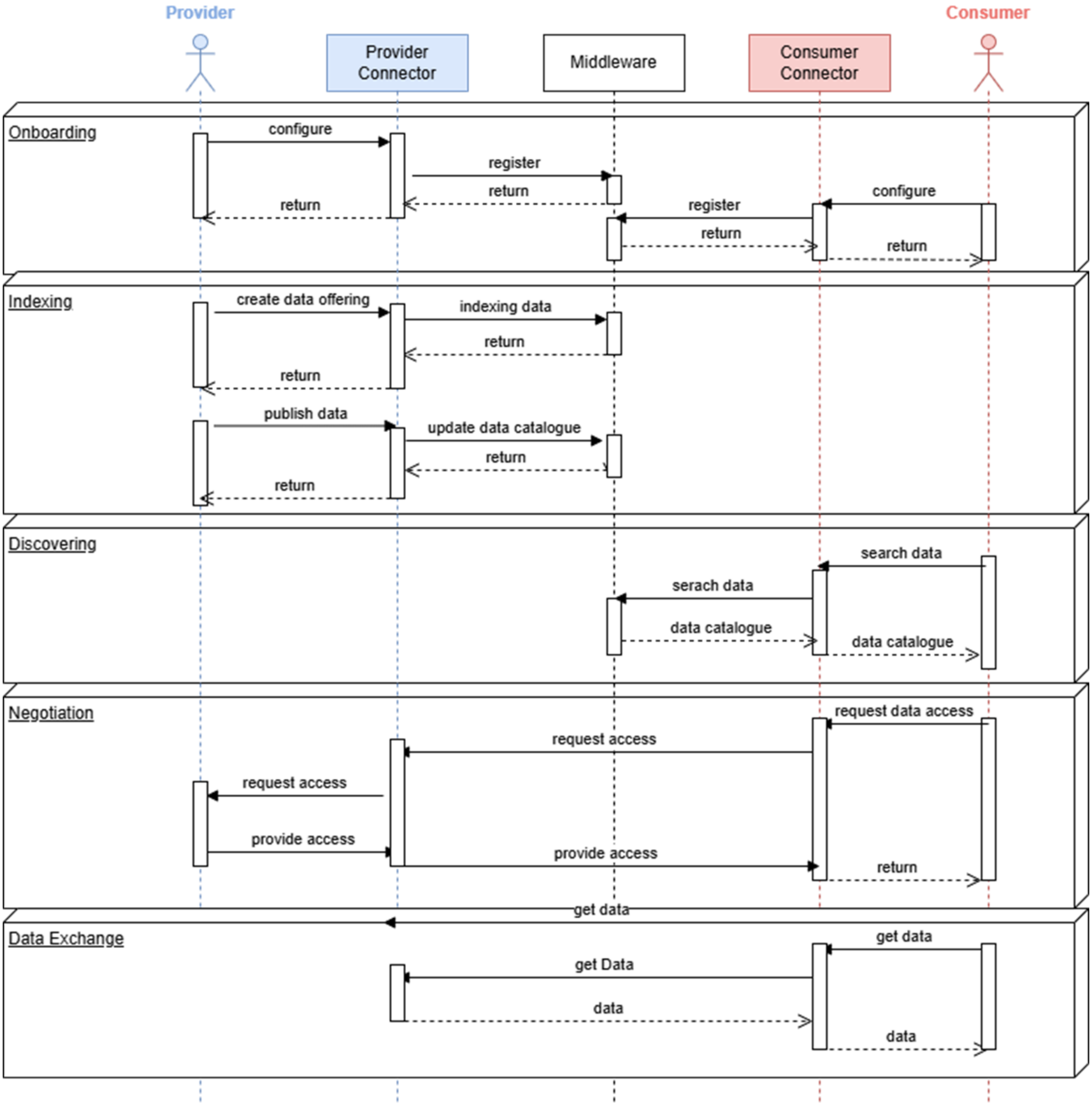



*Figure 19: DSP sequence diagram*

**Data Catalogue (Discovery)**

The Data Catalogue is a centralized repository where all indexed data offerings are listed. Users can browse, search, and discover data sets that are relevant to their needs. This step is crucial for facilitating easy access to data and ensuring that users can find the data they need efficiently. Within the TwinEU Middleware, the Data Catalogue also provides categorization mechanisms and a set of vocabularies to be applied to any data offering. The Data Catalogue is implemented in a standardized way, following the Data Space Protocol specifications. The DSP indicate the Catalogue as a collection of entries representing Datasets and their Offers that is advertised by a Provider Participant. The DSP Catalogue uses the standard DCAT model [35].

**Subscription and access management (Negotiation)**

Once Data Consumers find the data they need, they can subscribe to the data offerings. This step involves negotiating access terms, such as usage rights, and data sharing agreements. Effective subscription and access management ensure that data is shared securely and in compliance with agreed-upon terms. The negotiation step is implemented in accordance with the Data Space Protocol negotiation steps and workflow, which includes six distinct steps, as shown in Figure 20 below.

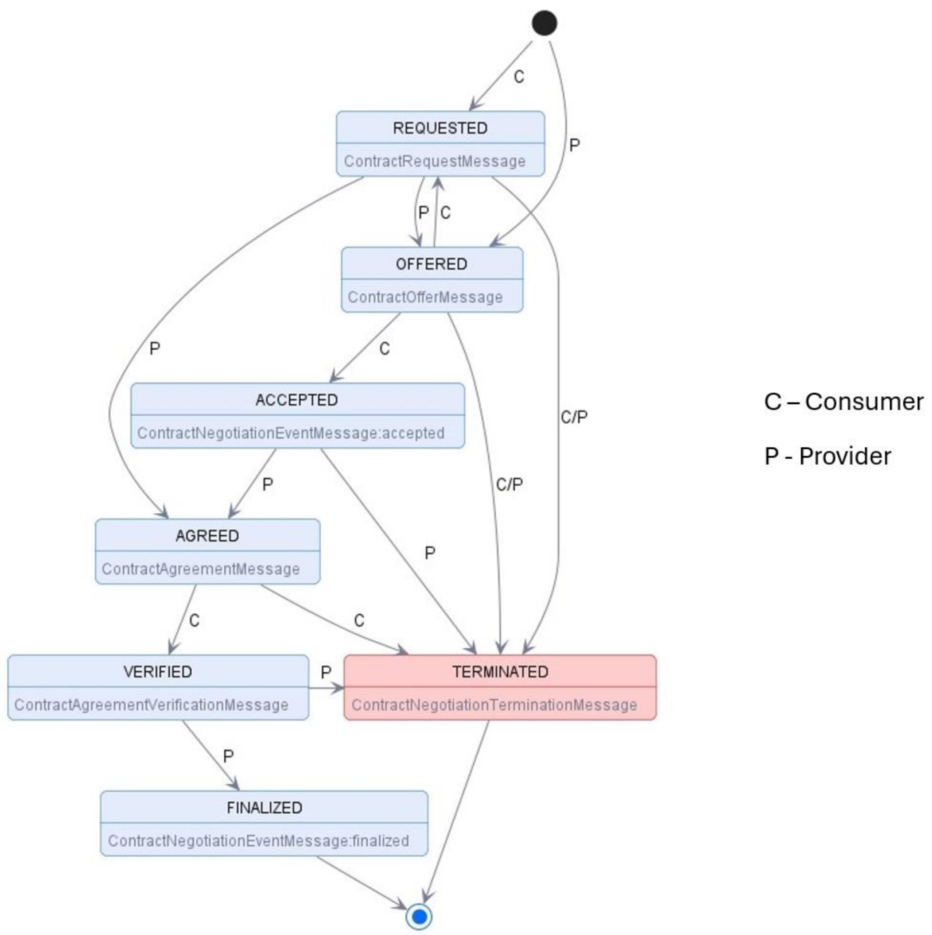



*Figure 20: DSP Negotiation Workflow*

**Provisioning and Consuming Data (Data Transfer)**

In the final step, the subscribed data is provisioned to the users. This involves the actual transfer of data from the provider to the consumer, ensuring seamless and efficient data exchange in an end-to-end manner, without passing data through any central system. In the DSP protocol the Data Transfer can be implemented in two-way approach:

- **Push process** – provider push data on consumer service
- **Pull process** – consumer request data to provide

### 5.1.2 TwinEU Middleware functionalities

To enable extended functionalities in alignment with the Dataspace Protocol (DSP) and modern data exchange patterns, the TwinEU Middleware module has been enhanced with the following main capabilities:

- DSP Support (version 2024.1)
- Push Flow Mechanism Support
- NATS Integration

#### 5.1.2.1 DSP Support

Support for the Dataspace Protocol (DSP) version 2024.1 has been implemented through new APIs and data model extensions. These changes enable interoperability with DSP-compliant connectors and services. The following modifications have been applied:

- Enhancements to the Data Entity model, including the addition of:
    - *File Type (varchar, 10)* – Specifies the type of file (e.g., CSV, JSON).
    - *Dataset Id (varchar, 50)* – Unique identifier for the dataset within the DSP True Connector.
    - *Policy (text)* – JSON data usage policy.
    - *Format (varchar, 50)* – Specifies the Transfer Type (PULL/PUSH)
- Enhancements to the User Provider model, including:
    - *DSP API URL (varchar, 500)* – The endpoint for DSP API communication.
    - *Connector URL (varchar, 500)* – The URL of the DSP connector instance associated with the user.

#### 5.1.2.2 PUSH Flow support

The PUSH flow mechanism has been introduced to enable event-driven data exchange without the need for explicit data pull requests. This includes:

- Creation or Modification of REST APIs to handle:
    - Creation and management of offered services with push capability.
    - Subscription requests to push-based data offering services.
    - Upload of data entities in push scenarios.
- Enhancements to the Data Offering model, with the following fields:
    - *Type (varchar, 20)* – Indicates whether the offering service is of type "PUSH" or "PULL".
    - *Push URI (varchar, 1000)* – Target endpoint for receiving data via push.

#### 5.1.2.3 NATS Integration

To support real-time and scalable messaging, the middleware now includes support for NATS as a lightweight publish-subscribe messaging protocol. The following changes were made:

- Enhancements to the Data Offering model, with new fields:
    - *Topic (varchar, 1000)* – NATS topic to which the data is published or from which it is subscribed.
    - *Updating Frequency (double)* – Defines how often the data is expected to be updated or published (in seconds).

### 5.1.3 Graphical User Interface

The graphical interface, originally developed within the Interstore project [36], has been evolved and adapted to support the DSP standard, optimizing the user experience and ease of use of the connector.

The starting point was the user interface already designed for the OneNet project, which was completely rewritten as a modern single-page application, fully decoupled from the BackEnd API.

The interface was extended and evolved within the TwinEU project to integrate the additional functionalities developed for the TwinEU DT Federator. The new interface includes the following features:

- User management and user authentication,
- Dashboards,
- Connector Settings,
- Services Catalog
- Viewing the macro services of the Energy domain,
- Creation and management of Services,
- Subscription of services with acceptance workflow,
- Data exchange (upload/download) on a service,
- Timeline.

In the next chapter all the technical information regarding the new user interface module will be included.

## 5.2 Interfaces and Communication Mechanisms

All modules within the TwinEU DSP Connector primarily communicate through REST APIs.

APIs can be divided into the following three groups:

- Centralized APIs
- TwinEU DSP Connector APIs
- DSP True Connector APIs

The next sections describe the aforementioned APIs and the main data models, which follow the Open API [39] specification.

### 5.2.1 Centralized APIs

This set of APIs supports all operations related to:

- User management (including authentication, user information, and connector settings),
- Service management,
- Data exchange.

The following sections provide a detailed description of these APIs.

#### 5.2.1.1 User Management

Table 5 below reports the list of REST APIs for the user management, including authentication and profile settings.

*Table 5: Centralized APIs - User Management*

| Name | URL | Method | Parameters | Responses |
|---|---|---|---|---|
| User Authentication | /api/user/auth | POST | In body:<br>username: String<br>password: String | Success (200)<br>accessToken: String<br>refreshToken String<br>user: {User}<br><br>Error (XXX)<br>Error code - String<br>Error Message - String |
| User Info | /api/user/current | GET | - | Success (200)<br>user: {User}<br><br>Error (XXX)<br>Error code - *String*<br>Error Message - *String* |
| Get Connector Settings | /api/custom-query/data-objects/?id=e48046c9-0b94-41d2-9ad4-206f1604b821 | GET | In Request:<br>Id: String | Success (200)<br>Connector settings info<br><br>Error (XXX)<br>Error code - *String*<br>Error Message - *String* |
| Save Connector Settings | /api/custom-query/data-objects/?id=74c9e3bc-4e26-4d74-aefb-a5ab4b364c1e | POST | In Request:<br>Id: String<br>local_api_url: String<br>endpoint_connector_url: String | Success (200)<br><br>Error (XXX)<br>Error code - String<br>Error Message - String |

#### 5.2.1.2 Service Management

Table 6 below shows the list of REST APIs for the management of energy services, including creation, categorization, subscription and access request.

*Table 6: Centralized APIs - Service Management*

| Name | URL | Method | Parameters | Responses |
|---|---|---|---|---|
| Cross Platform Service List | /api/datalist/cross_platform_service/page//{{page-number}} | GET | In Path:<br>page : int<br>In Request:<br><filters>: string | Success (200)<br>listContent: Array[CrossPlatformService] |

| | | | | |
|---|---|---|---|---|
| | | | | Error (XXX)<br>Error code - *String*<br>Error Message - *String* |
| Cross Platform Service Get by id | api/dataset/cross_platform_service/{{entity_id}} | GET | In Path:<br>entity_id: string | Success (200)<br>“data_catalog_business_object”: {CrossPlatform Service}<br><br>Error (XXX)<br>Error code - *String*<br>Error Message - *String* |
| Offered Services List | /api/datalist/my_offered_services/page/{{page-number}} | GET | In Path:<br>page: int<br>In Request:<br><filters>: string | Success (200)<br>listContent: Array[OfferedService]<br><br>Error (XXX)<br>Error code - String<br>Error Message - String |
| Push Offered Services List | /api/datalist/my_push_offered_services/page/{{page-number}} | GET | In Path:<br>page: int<br>In Request:<br><filters>: string | Success (200)<br>listContent: Array[OfferedService]<br><br>Error (XXX)<br>Error code - String<br>Error Message - String |
| Offered Services Get By Id | /api/dataset/my_offered_services/{{entity_id}} | GET | In Path:<br>entity_id: string | Success (200)<br>“data_catalog_business_object”: {OfferedService}<br><br>Error (XXX)<br>Error code - *String*<br>Error Message - *String* |
| Offered Services Create/Update | /api/dataset/my_offered_services | POST | In Body:<br>data_catalog_data_offerings: {OfferedService} | Success (200)<br>id of { OfferedService}<br><br>Error (XXX)<br>Error code - String<br>Error Message - String |
| Subscription List | /api/datalist/my_subscriptions/page/{{page-number}} | GET | In Path:<br>page: int<br>In Request:<br><filters>: string | Success (200)<br>listContent: Array[Susbscription]<br><br>Error (XXX)<br>Error code - String<br>Error Message – String |
| Push Subscription List | /api/datalist/my_push_sub/page/{{page-number}} | GET | In Path:<br>page: int<br>In Request:<br><filters>: string | Success (200)<br>listContent: Array[Susbscription]<br><br>Error (XXX) |

| | | | | |
|---|---|---|---|---|
| | | | | Error code - String<br>Error Message – String |
| Subscription Get By Id | /api/dataset/my_subscriptions/{{entity_id}} | GET | In Path:<br>entity_id: string | Success (200)<br>“data_catalog_data_requests”: {Subscription}<br><br>Error (XXX)<br>Error code - String<br>Error Message - String |
| Subscription Create | /api/dataset/my_subscriptions | POST | In Body:<br>data_catalog_data_requests: {Subscription} | Success (200)<br>id of {Subscription}<br><br>Error (XXX)<br>Error code - String<br>Error Message – String |
| Request List | /api/datalist/requests_on_offered_services/page/{{page-number}} | GET | In Path:<br>page: int<br>In Request:<br><filters>: string | Success (200)<br>listContent: Array[Request]<br><br>Error (XXX)<br>Error code - String<br>Error Message – String |
| Push Request List | /api/datalist/push_subscription/page/{{page-number}} | GET | In Path:<br>page: int<br>In Request:<br><filters>: string | Success (200)<br>listContent: Array[Request]<br><br>Error (XXX)<br>Error code - String<br>Error Message – String |
| Request Get By Id | /api/dataset/requests_on_offered_services/{{entity_id}} | GET | In Path:<br>entity_id: string | Success (200)<br>“data_catalog_data_requests”: {Request}<br><br>Error (XXX)<br>Error code - String<br>Error Message - String |
| Request Create | /api/dataset/requests_on_offered_services | POST | In Body:<br>data_catalog_data_requests: {Request} | Success (200)<br>id of {Request}<br><br>Error (XXX)<br>Error code - String<br>Error Message – String |

#### 5.2.1.3 Data Exchange

Table 7 below shows the list of REST APIs for the data exchange, including data provisioning and consumption.

*Table 7: Centralized APIs - Data Exchange*

| Name | URL | Method | Parameters | Responses |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| Data Provided List | /api/datalist/data_provided/page//{{page-number}} | GET | In Path:<br>page: int<br>In Request:<br><filters>: string | Success (200)<br>listContent: Array[Data Provided]<br><br>Error (XXX)<br>Error code - String<br>Error Message – String |
| Data Provided Get By Id | /api/dataset/data_provided/{{entity_id}} | GET | In Path:<br>entity_id: string | Success (200)<br>"data_send": {Data Provided}<br><br>Error (XXX)<br>Error code - String<br>Error Message - String |
| Data Consumed | /api/datalist/data_consumed/{{page-number}} | GET | In Path:<br>page: int<br>In Request:<br><filters>: string | Success (200)<br>listContent: Array[Data Provided]<br><br>Error (XXX)<br>Error code - String<br>Error Message – String |
| Push Data Consumed | /api/datalist/push_data_consumed/{{page-number}} | GET | In Path:<br>page: int<br>In Request:<br><filters>: string | Success (200)<br>listContent: Array[Data Provided]<br><br>Error (XXX)<br>Error code - String<br>Error Message – String |
| Data Consumed Get By Id | /api/dataset/data_consumed/{{entity_id}} | GET | In Path:<br>entity_id: string | Success (200)<br>"data_send": {Data Consumed }<br><br>Error (XXX)<br>Error code - String<br>Error Message – String |
| Timeline List | /api/timeline/data/?id=d6342c52-8995-4a0d-b42d-894ffc600a3d&enabled=1 | GET | In Request:<br>created_after: data<br>created_before: data<br>currentPage: int | Success (200)<br>resultList: Array[DataEntity]<br><br>Error (XXX)<br>Error code - String<br>Error Message – String |

### 5.2.2 TwinEU DSP Connector APIs

The TwinEU DSP Connector APIs are used to manage data entities within the TwinEU DSP Connector, the DSP True Connector and the TwinEU Middleware.

The APIs are described in detail in the following table.

*Table 8: TwinEU DSP Connector APIs*

| Name | URL | Method | Parameters | Responses |
|---|---|---|---|---|
| Data Provide | /api/provide-data | POST | In Body:<br><br>title: String<br>description: String<br>filename: String<br>file: String<br>data_offering_id: String<br>code: String | Success (200)<br>Result List:<br>“id”: {fileid}<br>“retrieved”: boolean<br><br>Error (XXX)<br>Error code - String<br>Error Message – String |
| Data Consume | /api/consume-data/{{fileid}} | GET | In Request:<br>fileid: String | Success (200)<br>Result List:<br>“fileData”: String<br>“responseCode”: boolean<br><br>Error (XXX)<br>Error code - String<br>Error Message – String |
| Data Delete | /api/consume-data/{{fileid}} | DELETE | In Request:<br>fileid: String | Success (200)<br><br>Error (XXX)<br>Error code - String<br>Error Message – String |

### 5.2.3 DSP TRUE Connector APIs

The DSP TRUE Connector API is used to interact directly with the DSP TRUE Connector [37], in accordance with the Dataspace Protocol.

This set of APIs supports all operations related to:

- Catalog,
- Contract Negotiation,
- Transfer Process.

The APIs are described in detail in the following sections.

### 5.2.3.1 Catalog

*Table 9: DSP TRUE Connector APIs - Catalog*

| Name | URL | Method | Parameters | Responses |
|---|---|---|---|---|
| Catalog Request | /catalog/request | POST | In body:<br>@context: String<br>@type: String<br>dspace:filter: [<filters>: String] | Success (200)<br>Catalog info<br><br>Error (XXX)<br>Error code - *String*<br>Error Message - *String* |
| Catalog Get By Dataset Id | /catalog/datasets/{dataset_id} | GET | In body:<br>@context: String<br>@type: String<br>dspace:filter: [<filters>: String]<br>In request:<br>dataset_id: String | Success (200)<br>Catalog info<br><br>Error (XXX)<br>Error code - *String*<br>Error Message - *String* |

### 5.2.3.2 Contract Negotiation

*Table 10: DSP TRUE Connector APIs - Contract Negotiation*

| Name | URL | Method | Parameters | Responses |
|---|---|---|---|---|
| Contract Negotiation Get By Provider Pid | /negotiations/{provider_pid} | GET | In request:<br>provider_pid: String | Success (200)<br>Contract Negotiation info<br><br>Error (XXX)<br>Error code - *String*<br>Error Message - *String* |
| Contract Negotiation Start | /negotiations | POST | In body:<br>Forward-To: String<br>offer: Object<br>@id: String<br>target: String<br>assigner: String<br>permission: Array of Object<br>action: String<br>constraint: Array of Object<br>leftOperand: String<br>operator: String<br>rightOperand: String | Success (200)<br>Contract Negotiation info<br><br>Error (XXX)<br>Error code - *String*<br>Error Message - *String* |
| Negotiation Approve | /negotiations/{{provider_contract_negotiation_id}}/approve | PUT | In request: | Success (200) |

| | | | | |
|---|---|---|---|---|
| [Provider ] | | | provider_contract_negotiation_id: String | Error (XXX)<br>Error code - *String*<br>Error Message - *String* |
| Negotiation Verify [Consumer] | /negotiations/{{consumer_contract_negotiation_id}}/verify | PUT | In request:<br>consumer_contract_negotiation_id: String | Success (200)<br><br>Error (XXX)<br>Error code - *String*<br>Error Message - *String* |
| Negotiation Finalize [Provider ] | /negotiations/{{provider_contract_negotiation_id }}/finalize | PUT | In request:<br>provider_contract_negotiation_id: String<br><br>In body:<br>consumerPid: String<br>providerPid: String | Success (200)<br><br>Error (XXX)<br>Error code - *String*<br>Error Message - *String* |
| Negotiation Terminate [Consumer] | /negotiations/{{consumer_contract_negotiation_id}}/verify | PUT | In request:<br>consumer_contract_negotiation_id: String | Success (200)<br><br>Error (XXX)<br>Error code - *String*<br>Error Message - *String* |
| Negotiation Terminate [Provider ] | /negotiations/{{provider_contract_negotiation_id }}/finalize | PUT | In request:<br>provider_contract_negotiation_id: String | Success (200)<br><br>Error (XXX)<br>Error code - *String*<br>Error Message - *String* |

### 5.2.3.3 Transfer Process

*Table 11: DSP TRUE Connector APIs - Data Transfer*

| Name | URL | Method | Parameters | Responses |
|---|---|---|---|---|
| Transfer Get By Provider Pid [Provider] | /transfers/{provider_pid} | GET | In request:<br>provider_pid: String | Success (200)<br>Transfer Process info<br><br>Error (XXX) |

| | | | | |
|---|---|---|---|---|
| | | | | Error code - *String*<br>Error Message - *String* |
| Transfer Request [Provider] | /transfers/request | POST | In body:<br>@context: String<br>@type: String<br>dspace:consumerPid: String<br>dspace:agreementId: String<br>dct:format: String<br>dspace:callbackAddress: String | Success (200)<br>Transfer Process info<br><br>Error (XXX)<br>Error code - *String*<br>Error Message - *String* |
| Transfer Start [Provider] | /transfers/{{provider_pid}}/start | POST | In request:<br>provider_pid: String<br><br>In body:<br>@context: String<br>@type: String<br>dspace:consumerPid: String<br>dspace:providerPid: String | Success (200)<br><br>Error (XXX)<br>Error code - *String*<br>Error Message - *String* |
| Transfer Complete [Provider] | /transfers/{{provider_pid}}/completion | POST | In request:<br>provider_pid: String<br><br>In body:<br>@context: String<br>@type: String<br>dspace:consumerPid: String<br>dspace:providerPid: String | Success (200)<br><br>Error (XXX)<br>Error code - *String*<br>Error Message - *String* |
| Transfer Suspend [Provider] | /transfers/{{provider_pid}}/suspension | POST | In request:<br>provider_pid: String<br><br>In body:<br>@context: String<br>@type: String<br>dspace:consumerPid: String<br>dspace:providerPid: String<br>dspace:code: String<br>dspace:reason: Array of String | Success (200)<br><br>Error (XXX)<br>Error code - *String*<br>Error Message - *String* |
| Transfer Terminate [Provider] | /transfers/{{provider_pid}}/termination | POST | In request:<br>provider_pid: String<br><br>In body:<br>@context: String<br>@type: String<br>dspace:consumerPid: String<br>dspace:providerPid: String<br>dspace:code: String<br>dspace:reason: Array of String | Success (200)<br><br>Error (XXX)<br>Error code - *String*<br>Error Message - *String* |
| Artifact Get | /artifacts/{{transaction_id}} | GET | In request: | |

| | | | | |
|---|---|---|---|---|
| | | | transaction_id: String | |
| Transfer Start [Consumer] | /transfers/{{consumer_pid}}/start | POST | In request:<br>consumer_pid: String<br><br>In body:<br>@context: String<br>@type: String<br>dspace:consumerPid: String<br>dspace:providerPid: String<br>dspace:dataAddress: Object<br>@type: String<br>dspace:endpointType: String<br>dspace:endpoint: String<br><br>dspace:endpointProperties: Array of Object<br>@type: String<br>dspace:name: String<br>dspace:value: String | Success (200)<br><br>Error (XXX)<br>Error code - *String*<br>Error Message - *String* |
| Transfer Complete [Consumer] | /transfers/{{consumer_pid}}/completion | POST | In request:<br>consumer_pid: String<br><br>In body:<br>@context: String<br>@type: String<br>dspace:consumerPid: String<br>dspace:providerPid: String | Success (200)<br><br>Error (XXX)<br>Error code - *String*<br>Error Message - *String* |
| Transfer Suspend [Consumer] | /transfers/{{consumer_pid}}/suspension | POST | In request:<br>consumer_pid: String<br><br>In body:<br>@context: String<br>@type: String<br>dspace:consumerPid: String<br>dspace:providerPid: String<br>dspace:code: String<br>dspace:reason: Array of String | Success (200)<br><br>Error (XXX)<br>Error code - *String*<br>Error Message - *String* |
| Transfer Terminate [Consumer] | /transfers/{consumer_pid}}/termination | POST | In request:<br>consumer_pid: String<br><br>In body:<br>@context: String<br>@type: String<br>dspace:consumerPid: String<br>dspace:providerPid: String<br>dspace:code: String<br>dspace:reason: Array of String | Success (200)<br><br>Error (XXX)<br>Error code - *String*<br>Error Message - *String* |

### 5.2.4 Data Models

#### 5.2.4.1 User

Table 12 below shows the data model related to the user that can access the UI of the Connector.

*Table 12: Data Models - User*

| Field | Type | Description |
|---|---|---|
| username | String | **Required** Username of the user logged in |
| password | String | **Required** Password of the user logged in |
| Email | string | User's email provided during the registration |

#### 5.2.4.2 Connector Settings

Table 13 below shows the data models for Connector settings and configurations.

*Table 13: Data Models - Connector Setting*

| Field | Type | Description |
|---|---|---|
| Id | String | Id of the user logged in |
| name | String | Name of the company |
| Email | String | Email of the user logged in |
| username | String | Username of the user logged in |
| local_api_url | String | Local API URL |
| endpoint_connector_url | String | Endpoint Connector URL |

#### 5.2.4.3 Cross Platform Services

Table 14 below shows the data model for the energy services catalogue.

*Table 14: Data Models - Cross Platform Services*

| Field | Type | Description |
|---|---|---|
| Category | String | Category of the cross-platform service |
| Service | String | Name of the cross-platform service |
| Business object name | String | Name of the business object |
| Business object code | String | Code of the business object |
| Service description | String | Description provided for the service |
| Profile selector | String | Format of the file (e.g. csv, json etc..) |
| profile_description | string | Description of the format |
| Cross platform service id | String | Id of the cross-platform service |
| file_schema | String | File schema providing the format |

| | | |
|---|---|---|
| file_schema_sample | String | Example of file schema providing the format |
| file_schema_filename | String | Name of the file schema providing the format |
| file_schema_sample_filename | String | Name of file_schema_sample |
| data_catalog_service | data_catalog_service object | Information about the service |

#### 5.2.4.4 My Offered Services

Table 15 below shows the data model for the energy services created by users.

*Table 15: Data Models - My offered Services*

| Field | Type | Description |
|---|---|---|
| category | String | Category of the offering service |
| title | String | Required Title of the offering service |
| created_on | String | Creation date of the offering service |
| profile_selector | String | Format of the file (e.g. csv, json etc..) |
| profile_description | String | Description of the format |
| status | String | Status of the service (active \| disabled) |
| subscriptions | String | User subscribed the service |
| comments | String | Comments from the service applicant |
| id | String | Id of the offered service |
| data_catalog_service | data_catalog_service object | Information about the service object |
| file_schema | String | File schema providing the format |
| file_schema_sample | String | Example of file schema providing the format |
| file_schema_filename | String | Name of the file schema providing the format |
| file_schema_sample_filename | String | Name of file_schema_sample |
| active_from | String | Activation date dd/mm/yyyy hh:mm |
| active_to | String | Activated until this date dd/mm/yyyy hh:mm |
| data_catalog_business_object | data_catalog_business_object object | Business object information |
| type | String (data/push) | Offering service type: data or push |
| push_uri | String | URL of Rest API to call in case of Push type offering Service |
| topic | String | NATS topic to subscribe |
| updating_frequency | Double | Frequency, in seconds, at which data is retrieved from the NATS topic and loaded into the service |

#### 5.2.4.5 My Subscription

Table 16 below shows the data model for user subscriptions to specific data offering.

*Table 16: Data Models - My subscriptions*

| Field | Type | Description |
|---|---|---|
| category | String | Category of the offering service |
| title | String | Title of the offering service |
| user_offering | String | User offering the service |
| status | String | Status of the service |
| created_on | String | Creation date of the offering service |
| comments | String | Comments for the subscriptions |
| profile_selector | String | Format of the file (e.g. csv, json etc..) |
| profile_description | String | Format description |
| data_catalog_data_offering_id | String | Id of the service provided |
| data_catalog_data_offerings | data_catalog_data_offerings object | Information on the service provided |

### 5.2.4.6 Requests

Table 17 below shows the data model for user access request to specific data offering.

*Table 17: Data Models - Requests*

| Field | Type | Description |
|---|---|---|
| category | String | Category of the offering service |
| title | String | Title of the offering service |
| user_offering | String | User offering the service |
| status | String | Status of the service |
| created_on | String | Creation date of the offering service |
| comments | String | Comments for the subscriptions |
| category | String | Category of the offering service |
| title | String | Title of the offering service |
| modified_on | String | Date when the dd/mm/yyyy hh:mm |
| user_requesting | user_requesting object | Information about User requesting |
| data_catalog_data_offerings | data_catalog_data_offerings object | Data offerings information |
| id | string | Id of the request |
| user | user object | User that is accepting/rejecting information |

### 5.2.4.7 Data Provided

Table 18 below shows the data model for the data offering of data providers.

*Table 18: Data Models - Data Provided*

| Field | Type | Description |
|---|---|---|
| data_catalog_category_name | String | Name of the data category |
| id | String | Data id |
| title | String | Required Data title |

| | | |
|---|---|---|
| created_on | String | Creation date of the offering service dd/mm/yyyy hh:mm |
| description | String | Description of the data |
| fileName | String | Name of the file |
| created_by | String | User who creates the data |
| data_catalog_data_offerings_id | String | Required Service id for new data |
| message | String | Required String containing the file (Base64) |
| profile_selector | String | Format of the file (e.g. csv, json etc..) |
| profile_description | String | Format description |
| category | String | Category of the offering service |
| title | String | Title of the offering service |
| dataset_id | String | DSP True Connector Dataset ID |
| policy | String | DSP True Connector Data Exchange Policy |
| format | String | DSP True Connector Request Format |

#### 5.2.4.8 Data Consumed

Table 19 below shows the data model for data available for data consumers.

*Table 19: Data Models - Data Consumed*

| Field | Type | Description |
|---|---|---|
| category | String | Category of the offering service |
| user_offering | String | User offering the service |
| created_on | String | Creation date of the offering service dd/mm/yyyy hh:mm |
| title | String | Data title |
| description | String | Description of the data |
| filedata | String | String containing the file (Base64) |
| fileName | String | Name of the file |
| data_catalog_data_offerings_id | String | Service id for new data |
| data_catalog_data_offerings | data_catalog_data_offerings object | Information about the service for new data |

#### 5.2.4.9 Timeline

Table 20 below shows the data model for the data exchange timeline.

*Table 20: Data Models - Timeline*

| Field | Type | Description |
|---|---|---|
| left_side | number | If it is equal to 0 the item represents consumed data, if it is equal to 1 provided data |
| isTheLastPage | number | If it is equal to 0 is the last page and is not possible to expand anymore |

| | | |
|---|---|---|
| description | String | Idem description (e.g. category and file name) |
| title | String | Item title |
| nav | String | String contains the data id |

## 5.3 Graphical User Interface

All the main features implemented by the TwinEU DSP Connector are available via Graphical User Interface (in a dedicated web app) and via REST APIs (described in Ch.5.2.2).

All the sections of the GUI with description of the functionalities are reported below.

### 5.3.1 Login

The first page described in this chapter is the login page. Through it, users can access the TwinEU DSP Connector web platform. Credentials are transmitted to the backend server, which responds with a "login succeeded" message (if the credentials are correct) or an error (if the credentials are incorrect). When a user is logged in, two toasts show the health status of the connector and the TwinEU DSP Connector API.

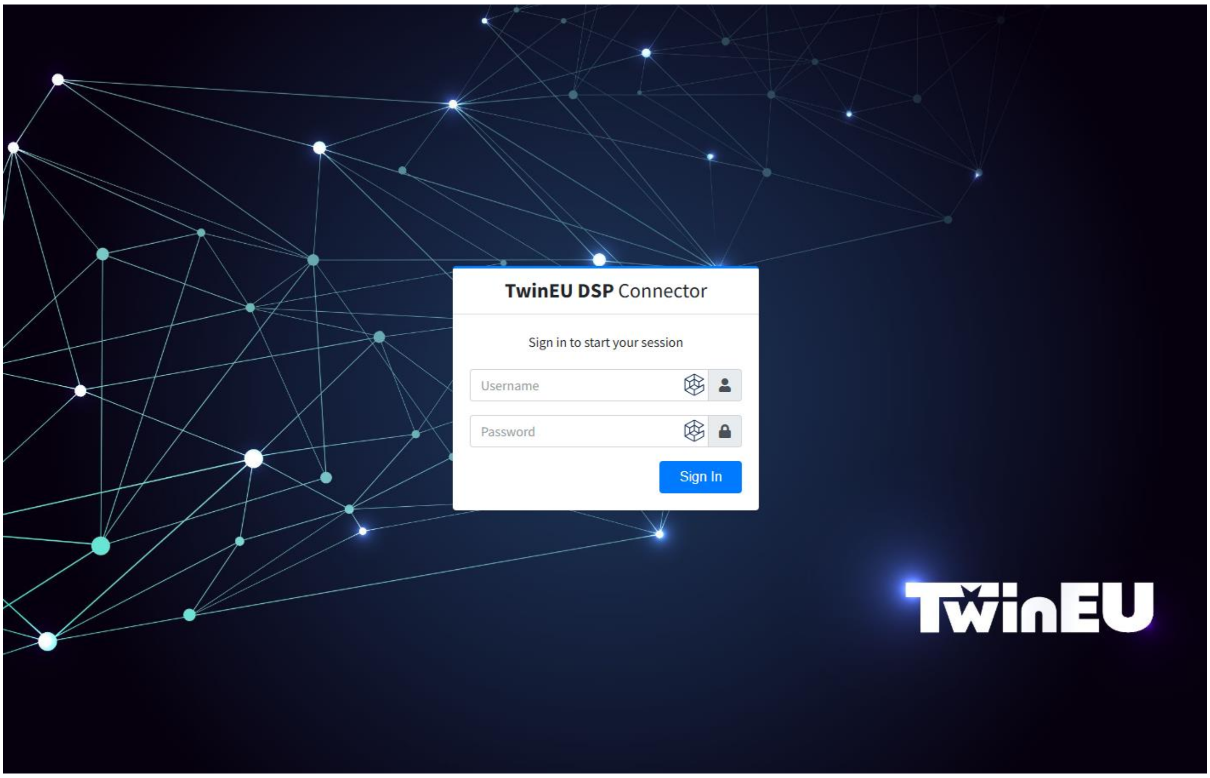



*Figure 21: Login of TwinEU DSP Connector*

### 5.3.2 Connector settings

This page gives an overview of the connector settings and provides information and the possibility to customize parameters.

The two buttons “Check” and “Check connector config” are used, respectively, to check the TwinEU DSP Api and the connector. Since the connector check is performed through the TwinEU DSP API, if the API is unavailable, the connector will also return an error.

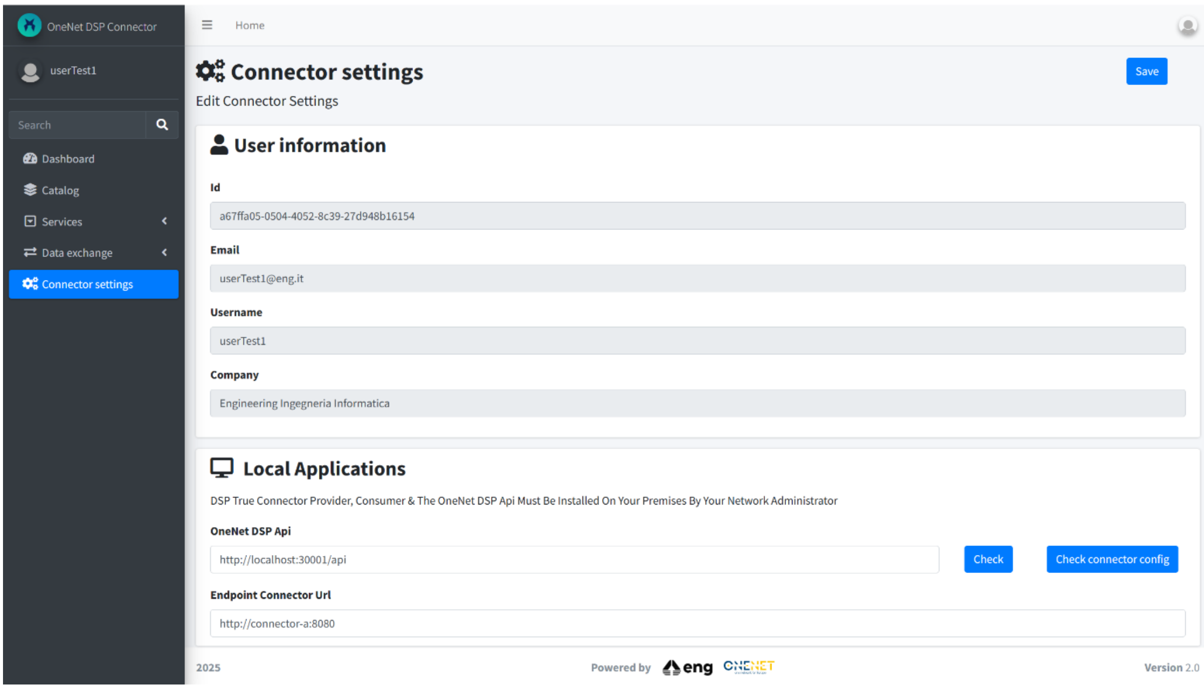



*Figure 22: Connector settings*

## 5.3.3 Dashboard

The Dashboard of the TwinEU DSP Connector web platform provides an overview about the user and the connector. In particular, this information is provided:

- Data consumed
- Data provided
- Offered services
- Subscriptions to services
- Pending subscriptions to services
- Connector check

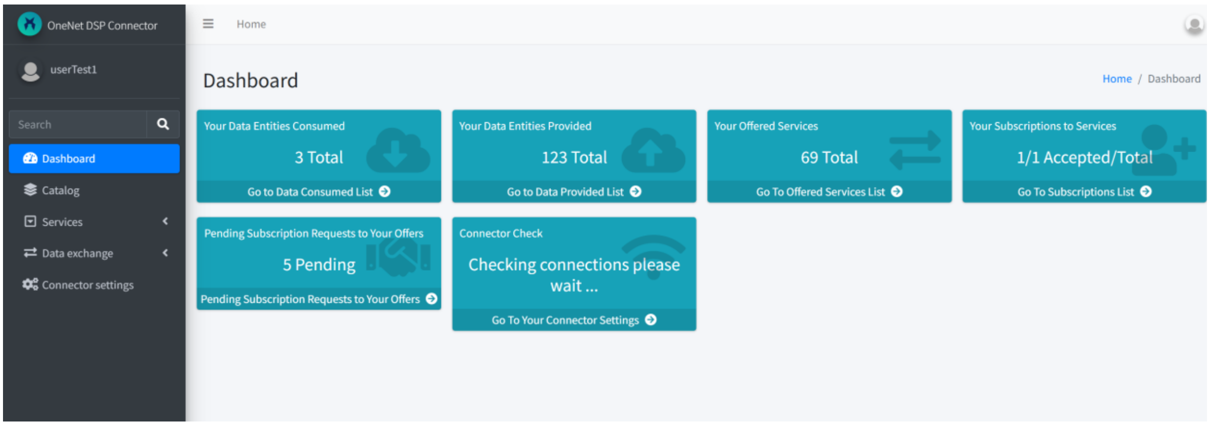
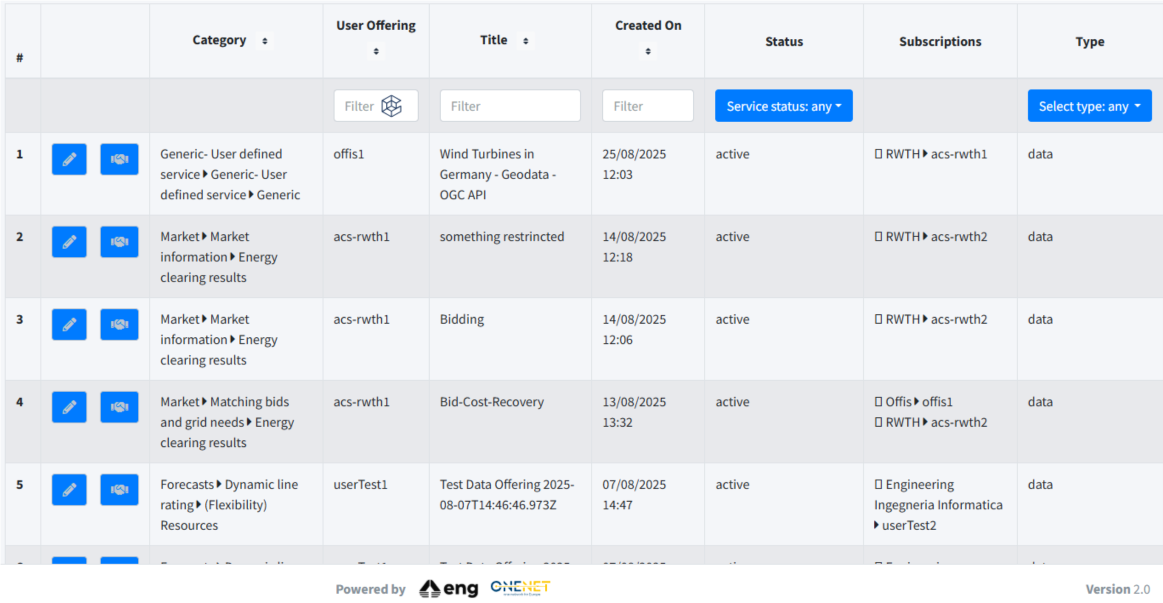



*Figure 23: Dashboard*

### 5.3.4 Service catalog

This page shows the list (divided by pages) of all offered services in the platform. Since the number of services can grow, to better navigate, some filters are provided. In particular, it is possible to filter with respect fields (by using a like filter) or filtering by category



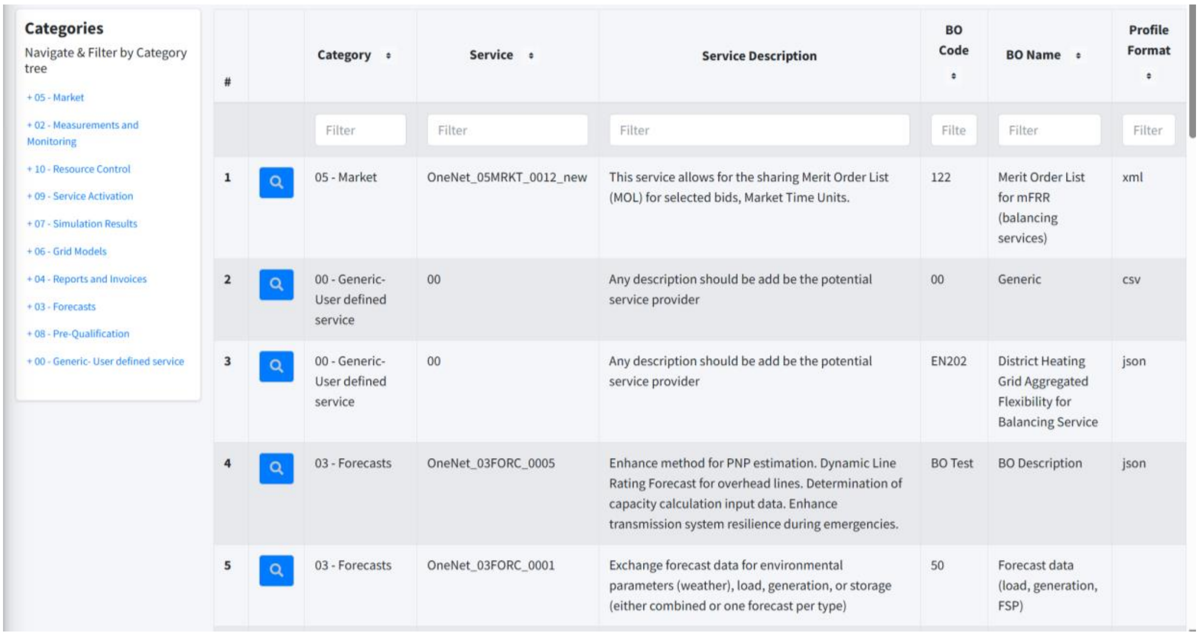
| # | | Category | User Offering | Title | Created On | Status | Subscriptions | Type |
|---|---|---|---|---|---|---|---|---|
| | | | Filter | Filter | Filter | Service status: any | | Select type: any |
| 1 | | Generic- User defined service ▸ Generic- User defined service ▸ Generic | offis1 | Wind Turbines in Germany - Geodata - OGC API | 25/08/2025 12:03 | active | RWTH ▸ acs-rwth1 | data |
| 2 | | Market ▸ Market information ▸ Energy clearing results | acs-rwth1 | something restrincted | 14/08/2025 12:18 | active | RWTH ▸ acs-rwth2 | data |
| 3 | | Market ▸ Market information ▸ Energy clearing results | acs-rwth1 | Bidding | 14/08/2025 12:06 | active | RWTH ▸ acs-rwth2 | data |
| 4 | | Market ▸ Matching bids and grid needs ▸ Energy clearing results | acs-rwth1 | Bid-Cost-Recovery | 13/08/2025 13:32 | active | Offis ▸ offis1<br>RWTH ▸ acs-rwth2 | data |
| 5 | | Forecasts ▸ Dynamic line rating ▸ (Flexibility) Resources | userTest1 | Test Data Offering 2025-08-07T14:46:46.973Z | 07/08/2025 14:47 | active | Engineering Ingegneria Informatica ▸ userTest2 | data |



*Figure 24: Catalog*

### 5.3.5 Service Categories

This page shows the list (divided by pages) of all service categories in the platform. Since the number of service categories can grow, to better navigate, some filters are provided. In particular, it is possible to filter with respect fields (by using a like filter) or filtering by category





| # | | Category | Service | Service Description | BO Code | BO Name | Profile Format |
|---|---|---|---|---|---|---|---|
| | | Filter | Filter | Filter | Filte | Filter | Filter |
| 1 | | 05 - Market | OneNet_05MRKT_0012_new | This service allows for the sharing Merit Order List (MOL) for selected bids, Market Time Units. | 122 | Merit Order List for mFRR (balancing services) | xml |
| 2 | | 00 - Generic- User defined service | 00 | Any description should be add be the potential service provider | 00 | Generic | csv |
| 3 | | 00 - Generic- User defined service | 00 | Any description should be add be the potential service provider | EN202 | District Heating Grid Aggregated Flexibility for Balancing Service | json |
| 4 | | 03 - Forecasts | OneNet_03FORC_0005 | Enhance method for PNP estimation. Dynamic Line Rating Forecast for overhead lines. Determination of capacity calculation input data. Enhance transmission system resilience during emergencies. | BO Test | BO Description | json |
| 5 | | 03 - Forecasts | OneNet_03FORC_0001 | Exchange forecast data for environmental parameters (weather), load, generation, or storage (either combined or one forecast per type) | 50 | Forecast data (load, generation, FSP) | |

*Figure 25: Service categories*

### 5.3.6 Offered Services

This page shows the list, divided by pages, of all offered services (provided by the user logged in) in the platform. Since the number of offered services can grow, to better navigate, some filters are provided. In particular, it is possible to filter with respect fields (by using a like filter) or filtering by category. In this page there are two tabs for the two types of services: Data and Push. For each offered service, there is a complete set of information and two buttons. The first one (represented by a pencil) redirects to a page that allows the service's editing. The second one (represented by a handshake) instead redirects to a page that shows the requests for that service.

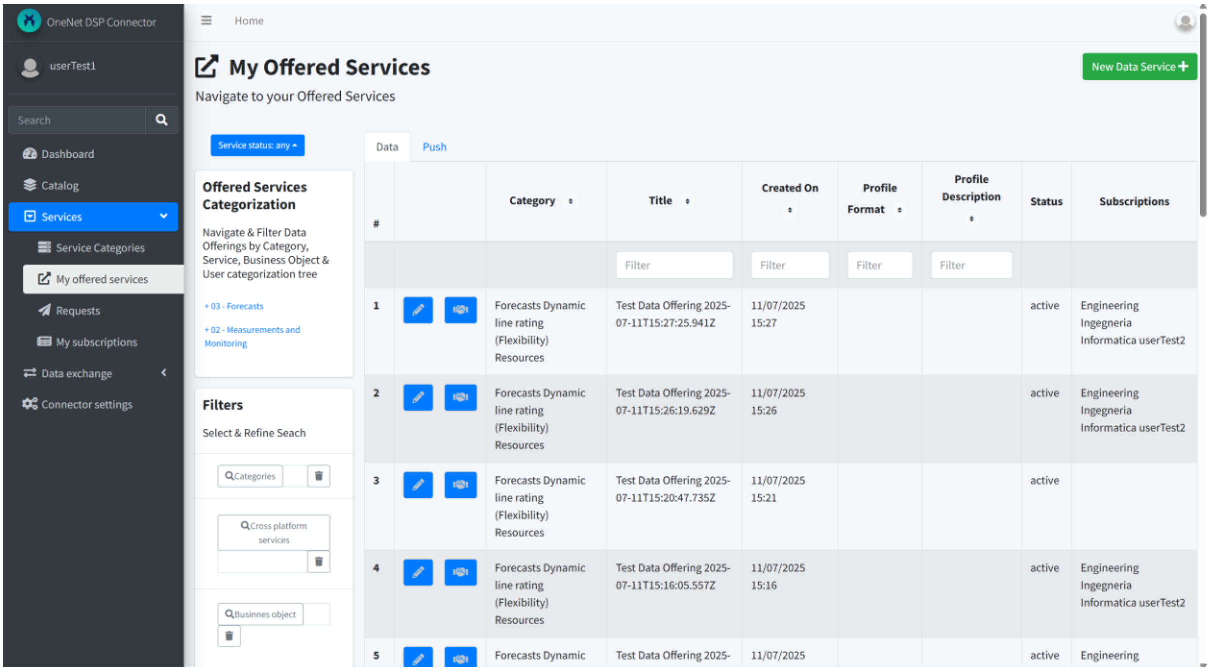



*Figure 26: "Data" type Offered Services*

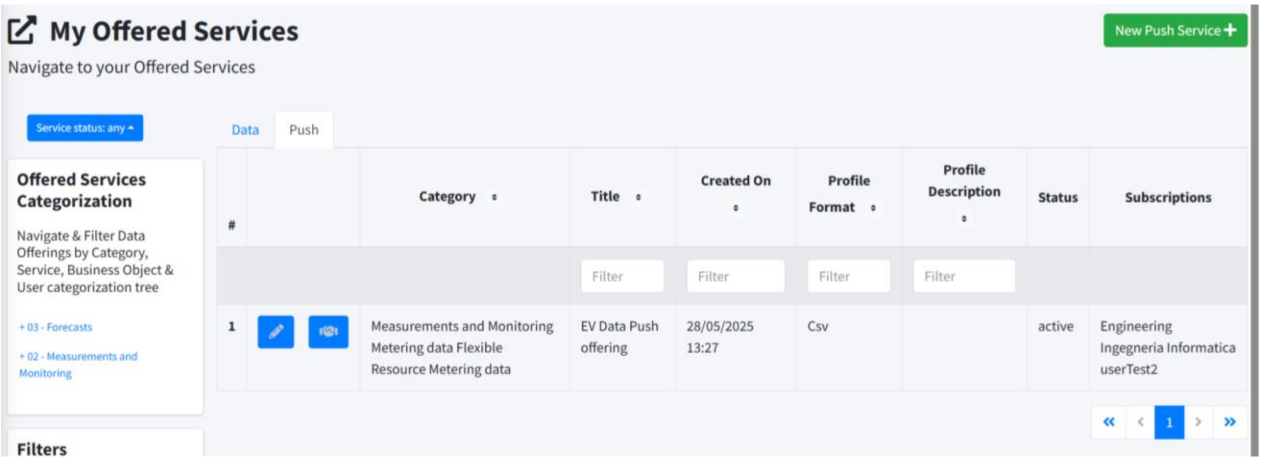



*Figure 27: "Push" type Offered Services*

In the Offered Service detail interface, mentioned before, it is possible to set data restrictions (active from and active to) and set NATS parameters (topic and uploading frequency).

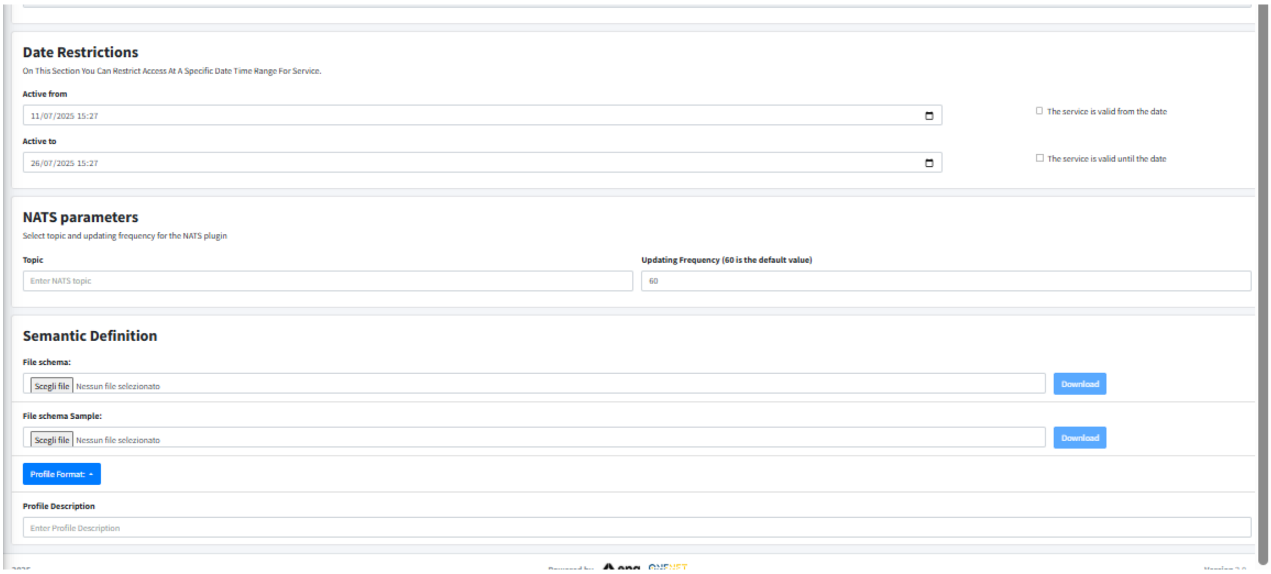



*Figure 28: Offered Service detail interface with NATS parameters configuration*

## 5.3.7 Requests

As mentioned before, each service can be requested; this page shows all the requests for all the user's services. Since services can be push type and data type it is possible to filter by type. By clicking the stencil button is possible to open the request's detail page and accept or reject the request.

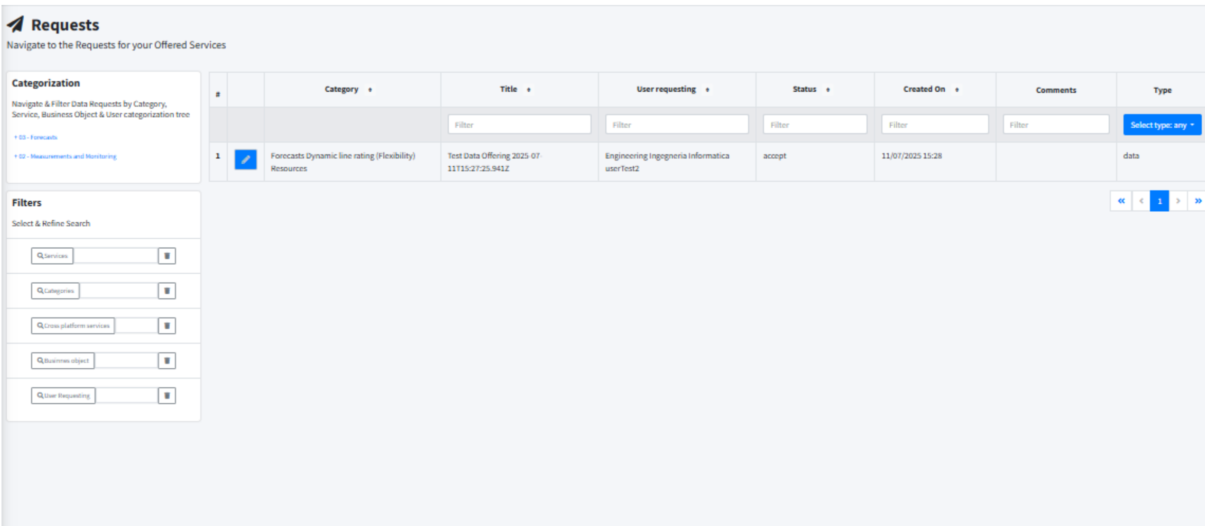



*Figure 29: Request list*

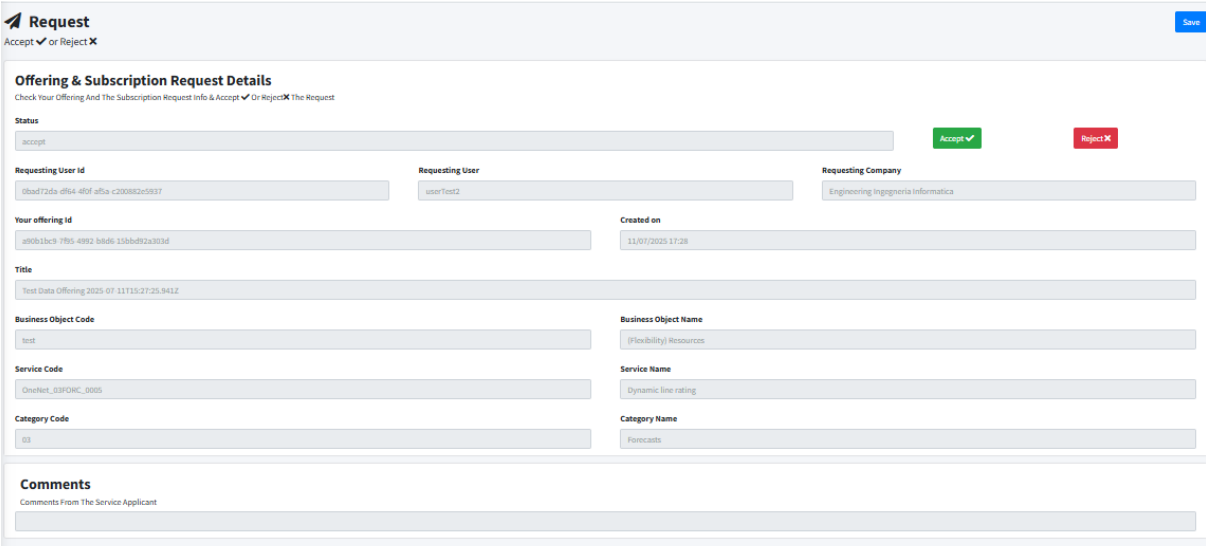



*Figure 30: Request details*

## 5.3.8 My subscriptions

This page provides the counterpart of the request page. Here is it possible to see the user's subscriptions and making a new one for both data and push services (by using the "new data subscription" and "new push subscription" buttons).

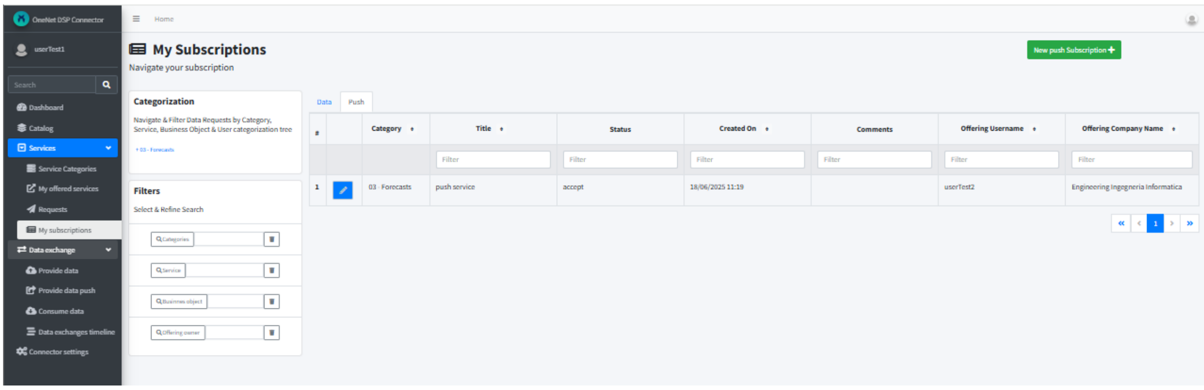



*Figure 31: My subscriptions*

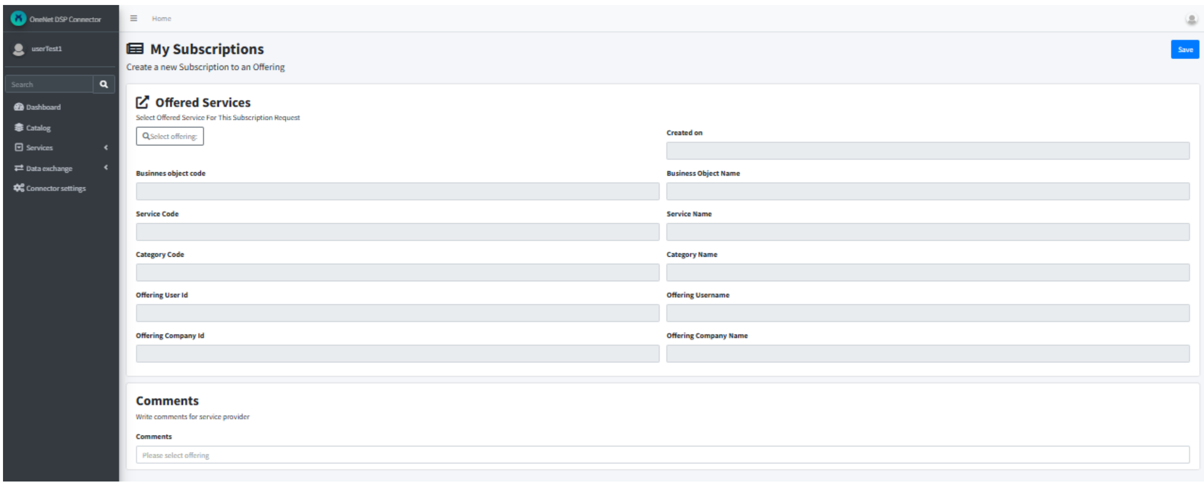



*Figure 32: New subscription*

### 5.3.9 Provide Data

On this interface, there are all the created data and the button to create new ones. There are two similar pages representing all the provided data, respectively for services of type data and type push. The button opens a form that provides the interface for uploading a new file, represented in the figure below.

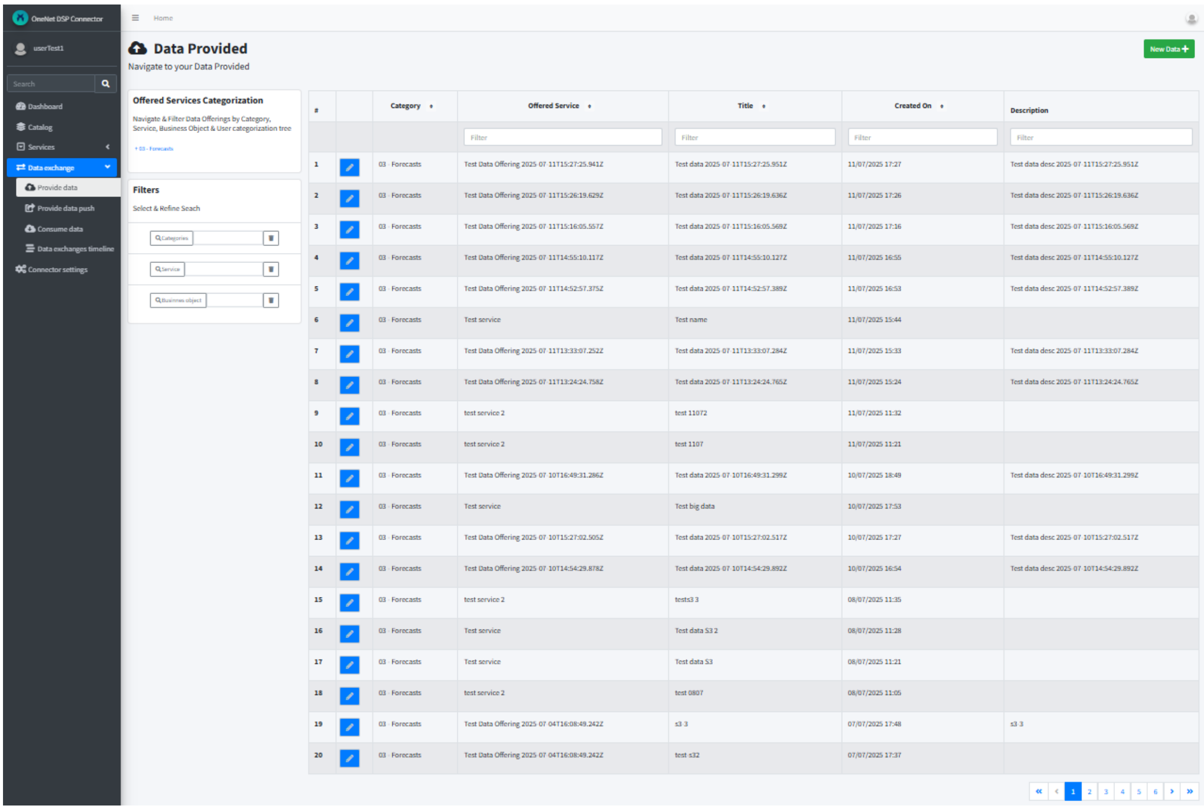



*Figure 33: Provide data*

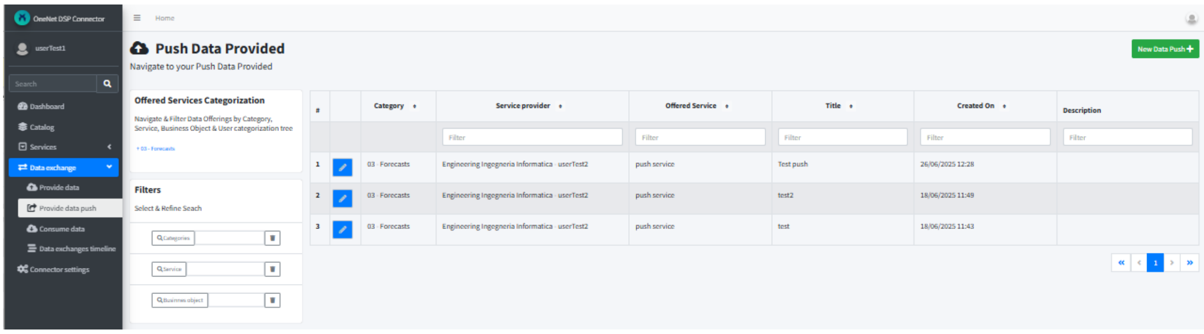



*Figure 34: Provide data push*

### 5.3.10 Consume Data

On this page, there are two lists representing all the data that is possible to consume, respectively for services of type data and type push.

By accessing the detail page, it is possible to see the information and download (consume) the file.

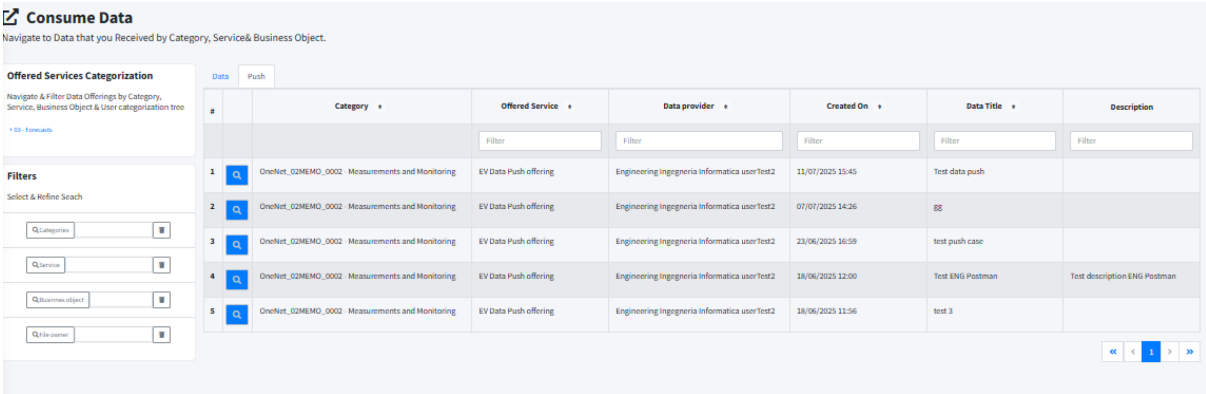



*Figure 35: Consume data list*

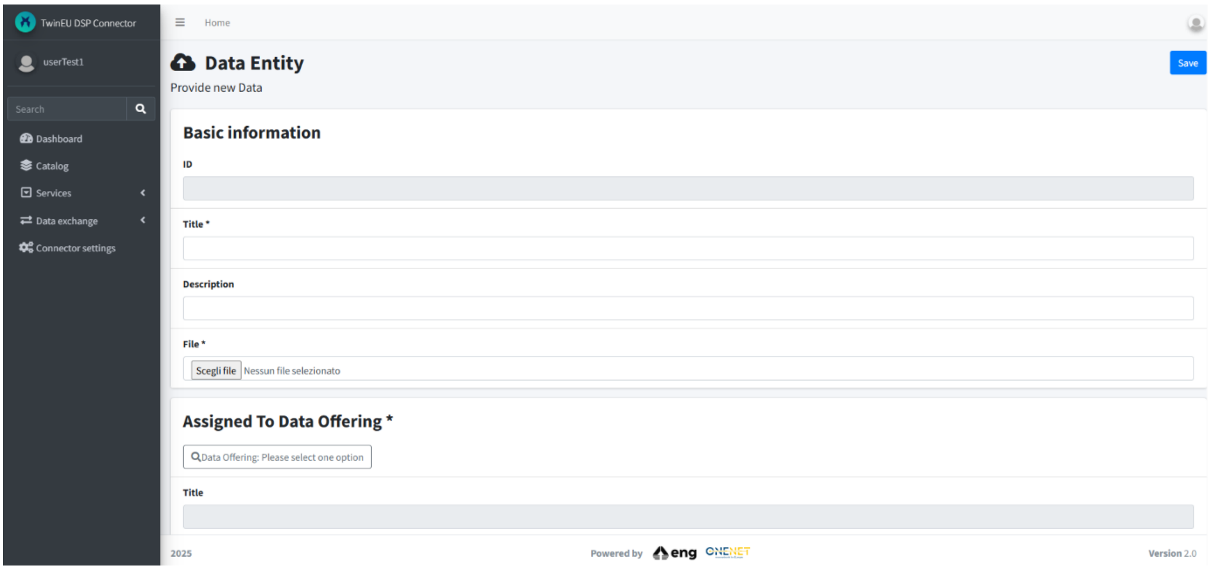



*Figure 36: Data entity detail*

## 5.3.11 Timeline

The timeline represents the data exchange flow. Each user can see the provided data on the right side (as a provider) and the consumed data on the left side (as a consumer). For each data transaction, there is a set of information about it, like the service type (data or push).

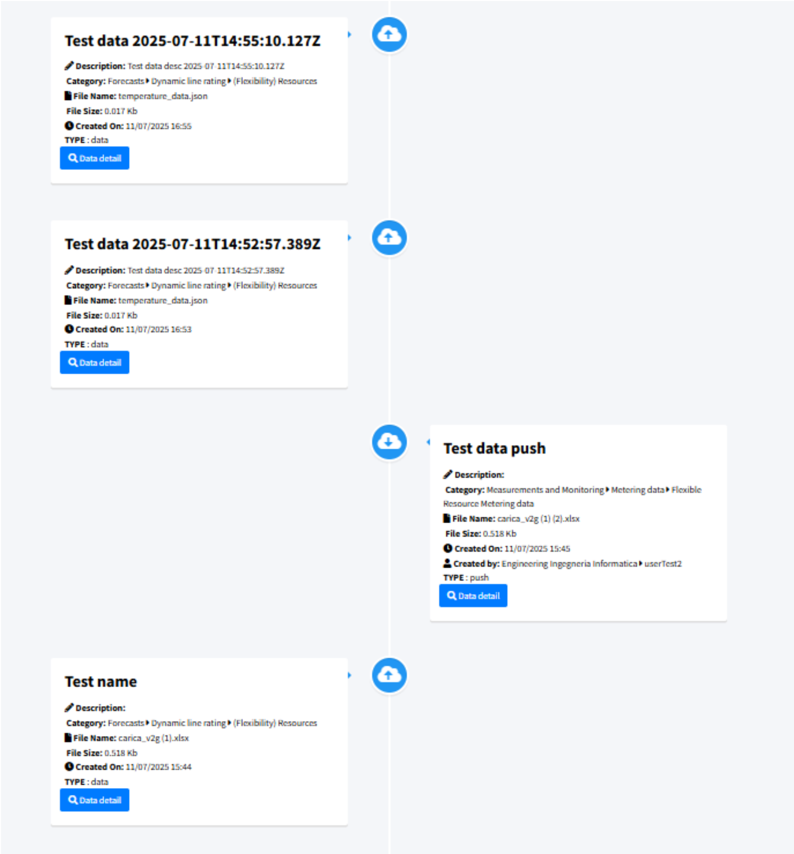



*Figure 37: Timeline*

## 5.4 Packaging, Deployment and Installation

The first version of TwinEU DSP Connector was released as open-source and both the docker containers and source code are available. This chapter reports all the aspects related to the deployment and installation of the solution, including guidelines and specifications.

### 5.4.1 Languages, Technologies and External Tools

Table 21 below reports some technical information about the tools used for the implementation, in order to facilitate the adoption and reuse of the solution by the partner of the consortium as wall as external and third-party stakeholders.

*Table 21: Languages, Technologies and Tools used for TwinEU DSP Connector development*

| Component/Service | Languages | Technologies/Framework | External Tools |
|---|---|---|---|
| Connector UI | Typescript<br>HTML5<br>CSS/SCSS | React<br>Vite<br>Bootstrap | Nginx<br>Docker |
| Middleware API | Java | Spring | Docker |
| TwinEU DSP API | Java | Spring | Docker |
| DSP True Connector | Java | Spring | Docker |
| Database | | MongoDB | Docker |

| | | | |
|---|---|---|---|
| NATS middleware component | Python | nats-py<br>requests<br>python-dotenv | Docker |
| MinIO Object Storage | | S3 compatible | Docker |
| E2E Automatic Tests | Typescript | Playwright | |

### 5.4.2 Deployment and Configuration

The deployment process involves the use of Docker containers. The use of Docker guarantees not only an easy deployment process and total portability of the solution, but also a high level of scalability of the released applications.

The hardware and operating system prerequisites are:

1. A 64bit 2-core processor
2. 8GB RAM Memory
3. 50GB of disk space or more

The software prerequisites include:

1 Linux or Windows (preferably Server edition) Operative System (OS)
2 Docker and docker-compose;

Energy Data Space Connector software and its components will be delivered utilizing the Docker containers functionalities. Firstly, the Docker platform has to be downloaded and installed accordingly to the OS of the server to host the deployment.

For the correct installation of docker and docker-compose, please refer to the official guides: https://docs.docker.com/get-docker/

***Energy data space installation on docker***

To proceed with the installation of Energy Dataspace Connector, the user must use the docker folder of the GitHub repository that contains all the necessary configuration.

1. The first step is to clone this repository https://github.com/TwinEU-Digital-Twin-for-Europe/onenet-dsp-connector in a specific folder *energy-data-space-connector*, by typing:

```
mkdir onenet-framework

cd onenet-framework

git clone https://github.com/TwinEU-Digital-Twin-for-Europe/onenet-dsp-connector
```

2. There is the *docker-compose.yml* file located under the docker folder that contains all the configuration of the Energy Data Space Connector containers. Go to that file by typing the command:

```
cd onenet-dsp-connector/docker
```

3. Start the containers with the below commands:

```
docker compose up -d
```

4. To show logs use the command:

```
docker compose logs -f
```

5. Alternatively, you can use Dozzle, a small lightweight application with a web-based interface to monitor Docker logs, for accessing the logs of each container. A Dozzle instance is included with Connector package and can be accessed at the following url on your browser:

```
http://localhost:8085
```

6. If no errors are seen, this means that Energy Data Space Connector was successfully deployed on your premises.
7. To stop all the containers use:

```
docker compose down
```

### *Login & Connector Settings*

The user interface is in a container that was installed on your premisses on the previous step. It can be accessed through the url:

```
http://localhost:8081/
```

1. You should see this interface, login using the username & password that you received from the TwinEU DSP Connector administrator.
2. Navigate to the connector settings by the sidebar menu & define the urls of your TwinEU DSP Api Url and Connector Url. Those 2 connector applications are running on the containers that you installed, so the urls must be configured accordingly as shown below.

Those Connector applications are running on the containers that you installed, so the URLs must be configured accordingly as shown below.

### *TwinEU DSP Api Url*

The url must be http://your_ip_where_the_containers_are_installed:30001/api In the default testing configuration two local-api are exposed to the URLs http://localhost:30001/api or http://localhost:30002/api, one for connector a and one for connector b.

### *Connector Url*

In the default testing configuration, the connectors are exposed to the URLs http://connector-a:8080 or http://connector-b:8080.

### *Environment Configuration*

Inside the /docker project folder, there is an environment configuration file (.env). This file allows you to set all Back End configurations of the TwinEU DSP Connector.

There are many environment settings, the following table shows some settings.

*Table 22: Environment Settings*

| Setting | Default value | Description |
|---|---|---|
| CONNECTOR_A_ENDPOINT_API_URL | http://connector-a:8080 | DSP True Connector A URL |
| CONNECTOR_B_ENDPOINT_API_URL | http://connector-b:8080 | DSP True Connector B URL |

| | | |
|---|---|---|
| CENTRALIZED_SERVICES_API_URL | https://twin-eu.eurodyn.com/api | API URL Configuration (centralized services). Not use localhost or 127.0.0.1, use the local IP address for local environment |
| PUSH_ENABLED | true | Push scenario: use true to enable, false to disable |

***IEEE2030.5 NATS component configuration***

NATS Component allows IEEE2030.5 integration with Energy Data Space Framework.

Inside the NATS component project, there is an *.env* environment configuration file. This file allows you to set all Back End configurations of NATS.

The following table shows and describe the settings.

*Table 23: NATS component environment configuration*

| Setting | Default value | Description |
|---|---|---|
| NATS_URL | nats://nats-server:4222 | NATS Server URL |
| DATA_AUTH_USR | **** | Data space account to use |
| DATA_AUTH_PSSWR | **** | |
| URL_LOCAL_API | http://host.docker.internal:30001/api/provide-data | Local APIs URL to upload data |
| URL_AUTH | https://smart-energy.eng.it/dev/api/user/auth | Centralized APIs Authentication URL |
| URL_SERVICES | https://smart-energy.eng.it/dev/api/datalist/my_push_offered_services/page/ | URL to get Data type Offering services |
| URL_PUSH_SERVICES | https://smart-energy.eng.it/dev/api/datalist/my_push_sub/page/ | URL to get Push type Offering services |
| BASE_FILENAME | Data_from_NATS_ | Base filename of data uploaded to the Data Space Connector (a timestamp and the .txt suffix are added automatically) |
| UPDATING_SERVICES_FREQUENCY | 60 | update services configuration frequency in seconds |
| PUB_ENABLED | false | Test Publisher component enabled |
| PUBLISHER_FREQUENCY | 2 | Test Publisher data send frequency in minutes |

### 5.4.2.1 Availability

The source code and the docker images necessaries for the deployment are available in the GitHub repository of the TwinEU project.

***GitHub Repository – Docker Version***

https://github.com/TwinEU-Digital-Twin-for-Europe/onenet-dsp-connector

***Source Code***

- GUI
- DSP Connector API
- DSP True Connector

# 6 Conclusions

The first release of the TwinEU DT Federator demonstrates the feasibility of a Data Space-enabled Digital Twin federation in the energy domain. It provides a robust, interoperable, and sovereign infrastructure for data and model sharing, laying the groundwork for future extensions and cross-sectoral integration. The work contributes significantly to the realization of the Common European Energy Data Space (CEEDS) and supports the broader goals of the TwinEU project in enabling a pan-European digital twin ecosystem

The TwinEU DT Federator introduces a modular architecture that supports both traditional and advanced data exchange mechanisms, including pull-based and push-based flows, as well as event-driven communication through the integration of the NATS protocol. These features are essential for supporting real-time responsiveness in energy systems, where latency and reliability are critical.

The TwinEU DSP Connector, which extend the OneNet Connector with the compliance to the Data Space Protocol, can provide a robust, interoperable, and sovereign infrastructure for secure data and model exchange across diverse stakeholders for DT federation.

The first release of the TwinEU DSP Connector is fully available for the partner of the consortium, both in a Docker version and with source code, in TwinEU GitHub repository, together with instruction for deployment, configuration and installation. During the next phase, it will be adopted by the pilot partners and will be tested and evaluated in 8 different demonstrations sites divided into 4 topics:

- Cyber-Physical Grid Resilience
- Grid Management, Operation and Monitoring
- Forecasting and Optimal Grid and Market
- Smart Coordinated Grid Planning

Through the validation of the main use cases and scenarios reported in this document, in the next phase of the project the TwinEU DT Federator requirements will be consolidated and enriched, with a focus on semantic interoperability and the definition of new vocabulary and standards for the DT context in the energy sector.

In collaboration with T4.3 and demo WPs a data sharing strategy will be defined and adopted, in order to define all the possible interface and data models for different DTs interconnection and federation. The outcomes of this activities will also fill in the Task 4.6 for the successful preparation and implementation of the pan-European scenarios.

The results of these activities, together with the final version of the TwinEU DT Federator, will be reported in the second version of this deliverable at M33.

In the final stage of the project, the whole TwinEU DT Federator will be public available for external stakeholders and the TwinEU DSP Connector will be released as open source.

# References


[1] IDSA - International Data Space Association Home Page. Online. Available: https://internationaldataspaces.org/ (Last Accessed August 27, 2025)

[2] IDSA - IDS Reference Model Architecture v4. Online. Available: https://github.com/International-Data-Spaces-Association/IDS-RAM_4_0/blob/main/README.md (Last Accessed August 27, 2025)

[3] IDSA - IDS RAM 4 Architecture Layers. Online. Available: https://github.com/International-Data-Spaces-Association/IDS-RAM_4_0/blob/main/documentation/3_Layers_of_the_Reference_Architecture_Model/3_Layers.md (Last Accessed August 27, 2025)

[4] IDSA - IDS RAM 4 system layer components. Online. Available: https://github.com/International-Data-Spaces-Association/IDS-RAM_4_0/blob/main/documentation/3_Layers_of_the_Reference_Architecture_Model/3_5_System_Layer/3_5_0_System_Layer.md#system-layer (Last Accessed August 27, 2025)

[5] IDSA - IDS Connector Architecture. Online. Available: https://github.com/International-Data-Spaces-Association/IDS-RAM_4_0/blob/main/documentation/3_Layers_of_the_Reference_Architecture_Model/3_5_System_Layer/3_5_2_IDS_Connector.md#ids-connector (Last Accessed August 27, 2025)

[6] IDSA - Data Space Protocol 2024.1. Online. Available: https://docs.internationaldataspaces.org/ids-knowledgebase/dataspace-protocol (Accessed August 27, 2025)

[7] GAIA-X – Home page. Online. Available: https://gaia-x.eu/ (Last Accessed August 27, 2025)

[8] GAIA-X – Architecture. Online. Available: https://docs.gaia-x.eu/technical-committee/architecture-document/25.05/pdf/document.pdf (Last Accessed August 27, 2025)

[9] Bohn, R. , Lee, C. and Michel, M. (2020), The NIST Cloud Federation Reference Architecture, Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD, [online], https://doi.org/10.6028/NIST.SP.500-332 , (Last Accessed August 27, 2025).

[10] European Commission – BRIDGE Smart Grid Storage Systems Digital Projects. Online. Available: https://bridge-smart-grid-storage-systems-digital-projects.ec.europa.eu/ (Last Accessed August 27, 2025)

[11] BRIDGE Data Management Working Group – European Energy Data Exchange Reference Architecture. Online. Available: https://bridge-smart-grid-storage-systems-digital-projects.ec.europa.eu/sites/default/files/bridge-reports/bridge_wg_data_management_eu_reference_architcture_report_2020-2021_0.pdf (Last Accessed August 27, 2025).

[12] CEN-CENELEC-ETSI Smart Grid Coordination Group - Smart Grid Reference Architecture. Online. Available: https://syc-se.iec.ch/wp-content/uploads/2019/10/Reference_Architecture_final.pdf, (Last Accessed August 27, 2025).

[13] European Commission: Directorate-General for Energy, Couto, M. J., Kotsalos, K. and Kukk, K., European (energy) data exchange reference architecture 3.0, Publications Office of the European Union, 2023, https://data.europa.eu/doi/10.2833/815043. (Last Accessed August 27, 2025).

[14] BRIDGE Data Management Working Group- European (energy) data exchange reference architecture 3.1 Online. Available: https://bridge-smart-grid-storage-systems-digital-projects.ec.europa.eu/sites/default/files/bridge-

reports/european%20energy%20data%20exchange%20reference%20architecture-HZ0124020ENN.pdf (Last Accessed August 27, 2025).

[15] Data Space Support Centre – About. Online. Available: https://dssc.eu/space/Partners/175472674/About+DSSC. (Last Accessed August 27, 2025).

[16] Data Space Support Centre – Fraunhofer ISST. Online. Available: https://www.isst.fraunhofer.de/en/expertise/DSSC.html. (Last Accessed August 27, 2025).

[17] Data Space Support Centre - Blueprint v2.0. Online. Available: https://dssc.eu/space/BVE2/1071251457/Data+Spaces+Blueprint+v2.0+-+Home. (Last Accessed August 27, 2025).

[18] IDSA - From Digitizing the European Industry (project OpenDEI) to the Data Spaces Support Centre: a clear path to the emergence of cross-sectoral European data spaces. Online. Available: https://internationaldataspaces.org/from-digitizing-the-european-industry-project-opendei-to-the-data-spaces-support-centre-a-clear-path-to-the-emergence-of-cross-sectoral-european-data-spaces/. (Last Accessed August 27, 2025).

[19] Data Space Support Centre - DSSC Glossary v2.0 September 2023. Online. Available: https://dssc.eu/space/Glossary/176553985/DSSC+Glossary+%7C+Version+2.0+%7C+September+2023. (Last Accessed August 27, 2025).

[20] Omega-X Project – The Pivotal Role of the Eclipse Data Space Components Connector in OMEGA-X Online. Available: https://omega-x.eu/2024/05/30/the-pivotal-role-of-the-eclipse-data-space-components-connector-in-omega-x/. (Last Accessed August 27, 2025).

[21] OneNet Project – OneNet Reference Architecture. Online. Available: https://www.onenet-project.eu//wp-content/uploads/2022/12/OneNet_D5.2_v1.0.pdf.

[22] OneNet Project – AI, Big Data, Orchestration Workbench. Online. Available: https://www.onenet-project.eu//wp-content/uploads/2023/09/OneNet_D6.4_v1.0.pdf.

[23] Enlit - Project ENERSHARE: Driving the energy transition through energy dataspaces. Online. Available: https://www.enlit.world/projects-zone/project-enershare-driving-the-energy-transition-through-energy-dataspaces/.

[24] ENERSHARE Project – ENERSHARE requirements, SSH-driven approach and reference architecture. Online. Available: https://enershare.eu/wp-content/deliverables/wp2/Enershare_D2.5_Final%20version%20of%20the%20ENERSHARE%20requirements%2C%20SSH-driven%20approach%2C%20and%20reference%20architecture%20v1.0.pdf.

[25] ENERSHARE Project - Position Paper: "Interoperability Framework in Energy Data Spaces v2" Has Been Published. Online. Available: https://enershare.eu/position-paper-interoperability-framework-in-energy-data-spaces-v2-has-been-published/.

[26] Fraunhofer Center Digital Energy - Project Launch »int:net« - Interoperability Network for the Energy Transition. Online: Available: https://www.digitale-energie.fraunhofer.de/en/presse-en/project-launch-int-net.html.

[27] Fraunhofer Center Digital Energy - The int:net model. Online. Available: https://www.digitale-energie.fraunhofer.de/en/projects/011-intnet-en.html.

[28] European Commission – Cordis Portal Int:net. Online. Available: https://cordis.europa.eu/project/id/101070086.

[29] Int:net project - The emerging Common European Energy Data Space - state of play and next steps. Online. Available: https://intnet.eu/news/54-the-emerging-common-european-energy-data-space-state-of-play-and-next-steps.

[30] Int:net project - Energy Data Space Cluster Workshop. Online. Available: https://intnet.eu/news/40-energy-data-space-cluster-workshop.

[31] Enlit - Energy data space framework: Interoperability and seamless data exchange in the energy sector. Online. Available: https://www.enlit.world/digitalisation/data-analytics/energy-data-space-framework-interoperability-and-seamless-data-exchange-in-the-energy-sector/.

[32] European Commission - European Commission advances development of common European energy data space (CEEDS). Online. Available: https://digital-strategy.ec.europa.eu/en/library/european-commission-advances-development-common-european-energy-data-space-ceeds.

[33] ENERSHARE Project – Blueprint of the Common European Energy Data Space. Online. Available: https://enershare.eu/wp-content/uploads/Blueprint_CEEDS_v2.pdf.

[34] Kravchenko, T., Bogdanova, T., Shevgunov, T. (2022). Ranking Requirements Using MoSCoW Methodology in Practice. In: Silhavy, R. (eds) Cybernetics Perspectives in Systems. CSOC 2022. Lecture Notes in Networks and Systems, vol 503. Springer, Cham. https://doi.org/10.1007/978-3-031-09073-8_18.

[35] W3C - Data Catalog Vocabulary (DCAT) - Version 3. Online. Available: https://www.w3.org/TR/vocab-dcat-3/.

[36] InterSTORE project – Open-source Data Space Connector UI. Online. Available: https://github.com/Horizont-Europe-Interstore/Data-Space-Connector-UI.

[37] Engineering Ingegneria Informatica S.p.a. – DSP TRUE Connector source code. Online. Available: https://github.com/Engineering-Research-and-Development/dsp-true-connector.

[38] NATS, Connective Technology for Adaptive Edge & Distributed Systems [Online]. Available: https://nats.io/.

[39] OpenAPI,[Online]. Available: https://www.openapis.org.