ENPOWER

POWER IN OUR HANDS

D4.1 | Energy Data Space adaptation and DLT-blockchain digital P2P marketplace documentation

DST

August 2024

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement Nº 101096354

Energy Activated Citizens and Data-Driven Energy-Secure Communities for a Consumer-Centric Energy System

Grant Agreement Number 101096354		Acronym ENPOWER	
Full Title	Energy Activated Citizens and Data-Driven Energy-Secure Communities for a Consumer-Centric Energy System		
Topic	HORIZON-CL5-2022-D3-01: Sustainable, secure, and competitive energy supply		
Funding scheme	HORIZON Innovation Actions		
Start Date	1st of September 2023	Duration	36 months
Project Coordinator	NTUA		
Deliverable	D4.1 - Energy Data Space adaptation and DLT/blockchain digital P2P marketplace for a consumer-centric energy system (1st technology release)		
Work Package	WP4 - Share: 'Energy Data Space adaptation as enabler of consumers individual and community-level activation and DR'		
Delivery Month (DoA)	August 2024	Version	1
Actual Delivery Date	August 30, 2024		
Nature	Report	Dissemination Level	PU
Lead Beneficiary	DST		
Authors	Carlos Pereira (INESC TEC), Moisés Fernández Muiña (DST), Klemen Bregar (COMS), Ioannis Madourarakis (ED)		
Quality Reviewer(s)	Vasilis Michalakopoulos (NTUA), Stathis Sarantinopoulos (NTUA), Nektarios Matsagkos (NTUA), Eleni Kanellou (NTUA), Filippos Ioannidis (GEARS), Tatiani Georgitsioti (GEARS)		

Copyright Message

Preface

ENPOWER will design, develop, and demonstrate SSH-driven methodologies, interactive and closed-loop tools, and data-driven services for energy-activated citizens and energy-secure cross-sector communities towards a citizen-centric energy system. We combine leading-edge ICTs with social/behavioural dimensions and with sharing economy and value stacking business models to deploy: 1) a Social Science Framework for energy citizens activation and innovative multidimensional incentives for citizens' participation in energy markets; 2) AI-based consumers clustering and segmentation algorithms; 3) interactive tools and facilitation services for energy community planning; 4) Energy Data Space adaptation to support community-level data-driven activation and interoperable automated privacy and sovereignty-preserving Demand Response (DR); 5) P2P DLT/Blockchain digital marketplace for tokenized energy and non-energy assets reciprocal compensation; 6) data-driven services and apps for energy efficiency and activation performance management; 7) ICT services and Digital Twins for community-level energy optimization and aggregated flexibility management to trade off local self-consumption against grid service provisioning, while boosting a beyond-energy community social welfare; 8) Edge monitoring hubs for automated DR; 9) Business Sandbox for novel sharing economy and social innovation-based business models; 10) methodologies for evaluating different energy community setups, upscaling and replication. ENPOWER framework will be validated along 4 Front Runners energy communities pilots and further replicated in 2 Early Adopters, covering different levels of maturity of communities, to demonstrate increased RES local self-consumption and consumers participation to the energy markets, while nurturing increased local security of supply. ENPOWER Leadership Programme and blueprints will support EU-wide replication and advice regulatory bodies on communities-friendly enabling frameworks.

Who we are

No	Participant Name	Short Name	Country	Role
1	National Technical University of Athens	NTUA	GR	Coordinator
2	Design Strategy Technology S.r.l	Dst	IT	Technology providers
3	INESC TEC - Institute for Systems and Computer Engineering, Technology and Science	INESC TEC	PT	
4	Fundación CARTIF (CARTIF Technology Center)	CARTIF	ES	
5	COMSENSUS D.O.O.	COMS	SL	
6	European Dynamics Luxembourg SA	ED	LU	
7	Audencia Business School	AUDENCIA	FR	Business modelling
8	Regulatory Assistance Project	RAP	BE	Regulatory assistance
9	ICLEI Europasekretariat GmbH	ICLEI	DE	Policy & capacity
10	University of Basel*(associated partner)*	UNIBAS	CH	Social framework
11	Blueprint Energy Solutions GMBH	BLUEPRINT	AT	Pilot 1 - Austria (Front runners)
12	OurPower Energiegenossenschaft SCE	OurPower	AT	
13	Cooperativa de Desenvolvimento Sustentável, CRL	COOPERNICO	PT	Pilot 2 - Portugal (Front runners)
14	WATT-IS SA	WATT-IS	PT	
15	Cooperativa Eléctrica do Vale D'Este CRL	CEVE	PT	
16	Energy Community of Chalki‘Chalkion’	ChalkiON	GR	Pilot 3 - Greece (Front runners)
17	Hellenic Association for Energy Economics	HAEE	GR	
18	Green Energy Aggregator Services SA	GEARS	GR	
19	Parity Platform P.C.	PARITY	GR	

20	Electric Power Research Institute Europe DAC	EEU	IE	Pilot 4 - Ireland (Front runners)
21	University College Cork (MaREI Centre for Energy, Climate and Marine, Environmental Research Institute)	UCC	IE	
22	DC Six Technologies Limited	DCSix	IE	
23	MOL TEIC - Dingle Hub	DINGLE HUB	IE	
24	ESB Innovation ROI Limited	ESB	IE	
25	Békéscsaba Energia ESCO Kft.	BCS	HU	Pilot 5 - Hungary (Early adopters)
26	Enasco Cleantech Alliance LLC.	ENASCO	HU	
27	Montajes Eléctricos Cuerva SL	CUERVA	ES	Pilot 6 - Spain (Early adopters)
28	VERGY COMMUNITY S.L. *(affiliated partner)*	VER	ES	

Executive Summary

This document presents Deliverable 4.1 of the ENPOWER project, titled "Energy Data Space Adaptation and DLT/Blockchain Digital P2P Marketplace for a Consumer-Centric Energy System (1st Technology Release)." The ENPOWER project aims to empower energy-activated citizens and develop data-driven energy-secure communities, fostering a consumer-centric energy system through innovative technologies and methodologies.

The primary objective of this deliverable is to outline the initial architecture and implementation of the ENPOWER system, focusing on privacy and security frameworks, data model interoperability, and the creation of tools compliant with Energy Data Space standards for sovereign data sharing and federated planning. This report details the establishment of an open, interoperable tokenized marketplace designed to facilitate reciprocal compensation of energy and data assets among consumers and communities.

The report outlines various data-driven initiatives and architectures that inspire the design of the ENPOWER system, including BRIDGE, IDS, GAIA-X, FIWARE, and the CEEDS Architecture Blueprint. A high-level architecture is provided, illustrating how these components integrate to support a consumer-centric energy marketplace.

Privacy, security, and legal compliance frameworks are addressed, including GDPR adherence, identity and access management, and security measures tailored to AI technologies and data protection. The deliverable also explores data models and approaches for achieving semantic interoperability within the system, emphasizing the SEMAPTIC approach and the ENCOM ontology for energy communities, designed to enable seamless data exchange and integration across different platforms.

Tools and connectors such as the FIWARE Context Broker and OneNet Connector support secure data sharing and collaborative planning within the Energy Data Space, enhancing interoperability and data sovereignty. The establishment of a tokenized marketplace is a core component of ENPOWER, enabling P2P transactions and reciprocal compensation of energy and data assets. The report describes the structure, objectives, and technological foundations of this marketplace, highlighting its impact on the broader ENPOWER ecosystem.

This initial technology release sets the foundation for subsequent developments within the ENPOWER project, aiming to increase local self-consumption of renewable energy sources, boost consumer participation in energy markets, and enhance local energy security. Future releases will build upon this groundwork, leading to the deployment of a robust, consumer-centric energy system across multiple European communities.

Table of Contents

Figures

Tables

List of acronyms

ACID	Atomicity, Consistency, Isolation, and Durability
AI	Artificial Intelligence
API	Application Programming Interface
BE	Back-End
CECs	Citizen EnCs
CH	Clearing House
CSV	Comma-Separated Values
DER	Distributed Energy Resources
DLT	Distributed Ledger Technology
DR	Demand Response
DSO(s)	Distribution System Operator(s)
DTs	Digital Twins
EC	European Commission
ECC	Execution Core Container
EDS	Energy Data Space
EnC(s)	Energy Community(ies)
ENCOM	Energy Community Ontology
FACT	Failure Attack and Countermeasure
FTP	File Transfer Protocol
GDPR	General Data Protection Regulation
GUI	Graphical User Interface
IAM	Identity and access management
ICT	Information Communication Technology
IDS	International Data Space
IDSA	International Data Spaces Association
IDSCP	IDS Communication Protocol
IEC	International Electrotechnical Commission
IoT	Internet of Things
ISG	ETSI Industry Specification Group
JSON	JavaScript Object Notation

KPIs	Key Performance Indicators
MQTT	Message Queuing Telemetry Transport
MVCC	Multi-Version Concurrency Control
NGSI	Next Generation Service Interfaces
OAuth2	Open Authorization
OIDC	OpenID Connect
OP	OpenID provider
ORDBMS	object-relational database management system
P2P	Peer-to-peer
PAT	Protection API Access Token
PCT	Persisted Claims Token
PEP	privacy enforcement point
PETs	Privacy Enhancing Technologies
PKCE	Proof Key for Code Exchange
PoA	Proof of Authority
RAM	Reference Architecture Model
RBAC	role-based access control
RP	Relying Party
RPT	Requesting Party Token
SAI	Secure AI
TFEU	Treaty of Functioning of the European Union
TSO(s)	Transmission System Operator(s)
UC	Usage Control
UMA2	User-Managed Access
VPN	virtual private network
WP	Work Package

1. Introduction

1.1 About this document

This document presents Deliverable 4.1: "Energy Data Space Adaptation and DLT/Blockchain Digital P2P Marketplace for a Consumer-Centric Energy System (1st Technology Release)" as part of Work Package 4 (WP4): SHARE. This deliverable marks the first of three key technological releases aimed at enabling the adaptation of the Energy Data Space (EDS) and the development of a decentralized peer-to-peer (P2P) marketplace leveraging distributed ledger technology (DLT) and blockchain. The overarching goal of WP4 is to empower consumers and communities through active participation in energy markets and demand response initiatives.

The primary objective of this document is to outline the initial architecture and implementation of the ENPOWER system, focusing on the integration of data-driven initiatives and architectures, privacy and security frameworks, data model interoperability, and the creation of tools compliant with Energy Data Space standards for sovereign data sharing and federated planning. Additionally, the document details the establishment of an open, interoperable, tokenized marketplace designed to facilitate reciprocal compensation of energy and data assets.

1.2 Document Structure

This deliverable is structured as follows:

- **Section 2 - Global Architecture of the ENPOWER System**: Discusses the various data-driven initiatives and architectures that inform the design of the ENPOWER system, including BRIDGE, IDS, GAIA-X, FIWARE, and CEEDS Architecture Blueprint.
- **Section 3 - Privacy, Security, and Legal Compliance Framework**: Covers the essential frameworks and protocols ensuring the privacy, security, and legal compliance of the system, including GDPR adherence and identity/access management.
- **Section 4 - Data Models and Service/Platform Interoperability**: Explores the models and approaches for achieving semantic interoperability within the system, with a detailed examination of the SEMAPTIC approach and the ENCOM ontology for energy communities.
- **Section 5 - Energy Data Space Compliant Tools for Sovereign Data Sharing and Federated Planning**: Details the tools and connectors, such as the FIWARE Context Broker and OneNet Connector, that support data sharing and collaborative planning.
- **Section 6 - Open Interoperable Tokenized Marketplace**: Describes the structure, objectives, and technological foundation of the tokenized marketplace, including its impact and relevance within the ENPOWER system.

This initial technology release lays the groundwork for subsequent developments, setting the stage for the final release and deployment of a robust, consumer-centric energy system.

2. Global architecture of the ENPOWER system

The aim of this section is to:

- illustrate most prominent work on software architectures and platforms that are of interest for and will be leveraged by ENPOWER High level Architecture.
- design the high-level architecture of the ENPOWER solution. Each architectural layer will be described accordingly.

2.1 Data-Driven initiatives and Architectures

This section focuses on major ongoing EU-level initiatives addressing data-driven solutions in the energy sectors which are envisaged to be leveraged by ENPOWER.

2.1.1 BRIDGE

BRIDGE is a European Commission initiative which brings together research and innovation projects focused on Smart Grids, Energy Storage, Islands, and Digitalization[1]. Its primary goal is to create a structured view of cross-cutting issues in these areas. The project is financed by the Horizon 2020 and Horizon Europe programs and involves 155 projects, among which 58 are completed as of 1 July 2023. The work on the project is structured through four working groups: Data Management, Regulation, Business Models, and Consumer and Citizens Engagement; these groups collaborate on addressing key challenges and sharing best practices in their respective areas.

The ENPOWER project plays a significant role in fostering collaboration, knowledge sharing, and policy support in energy innovation and smart grids within the European Union, while taking part to the main groups and getting as well significant input to align the project activities.

BRIDGE collaborates with external initiatives such as ETIP Smart Networks for Energy Transition, Smart Grid Task Force, Clean Energy Transition Partnership, and 2ZERO Partnership, allowing the BRIDGE community to stay informed about their work. Moreover, ENTSO-E and the EU DSO Entity introduced their joint preparatory work for promoting digital twins of electricity grids, emphasizing technological advancements in the energy sector.

Central for the ENPOWER activities is the Data Exchange Reference Architecture v3.0, which has been developed and fine-tuned within the Data Management Working Group (Figures 1 and 2, below), showing a clear alignment with IDS and GAIA-X.

[1] https://op.europa.eu/en/publication-detail/-/publication/dc073847-4d35-11ee-9220-01aa75ed71a1/language-en

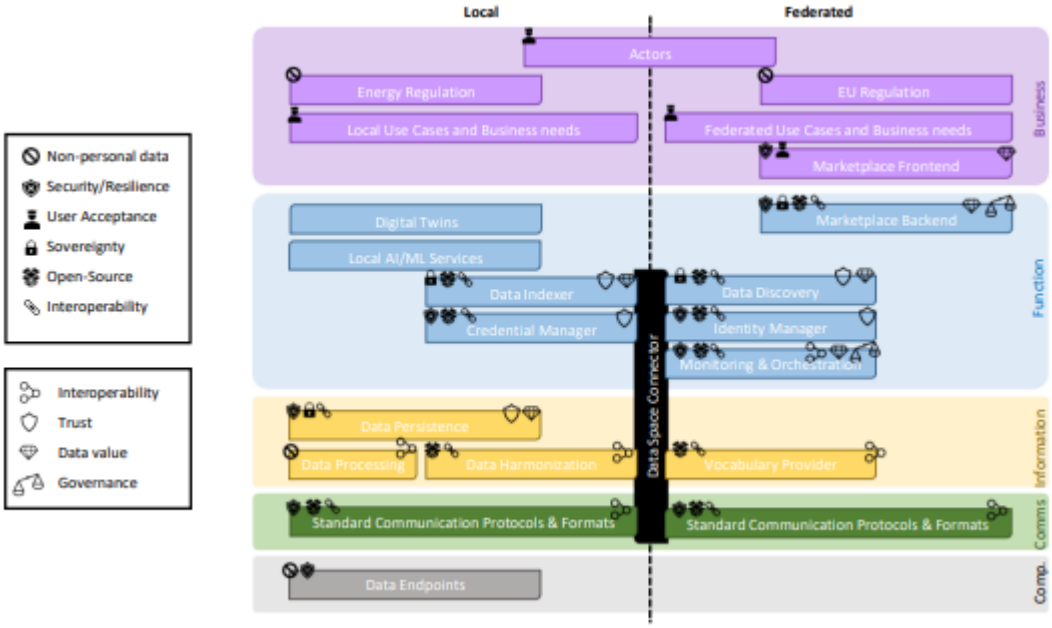

Figure 1: BRIDGE DERA v3.0.

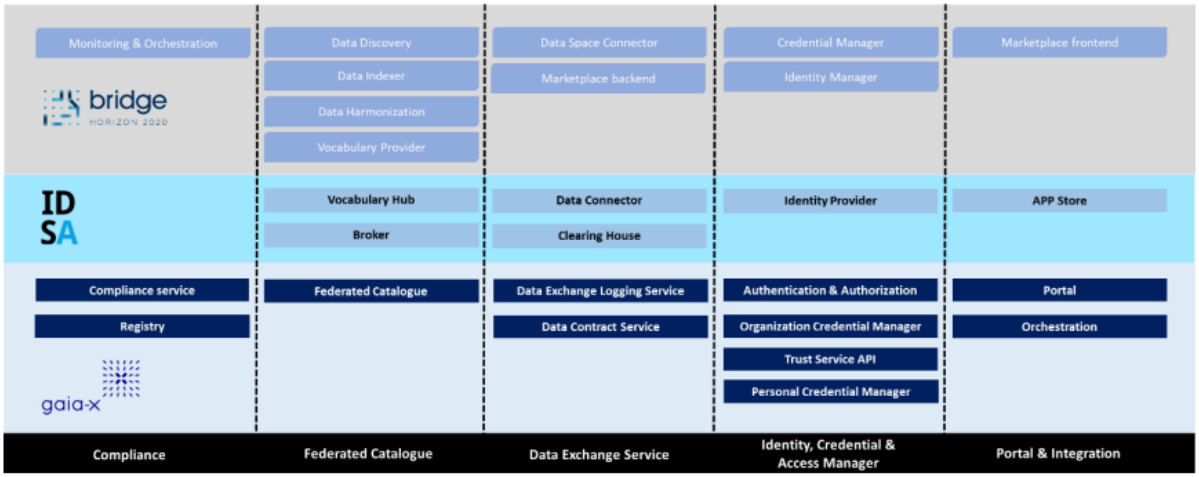

Figure 2: BRIDGE DERA alignment with IDSA and GAIA-X.

2.1.2 IDS (International Data Space)

A Data Space is a decentralized infrastructure for trustworthy data sharing and exchange in data ecosystems based on commonly agreed principles [1]. Data Spaces have been proposed by the European Union as a way to support data sharing, while respecting the rules that safeguard European values2. A Data Space ensures trust and security. Moreover, data sovereignty is ensured to guarantee the control of the data by their owner: a data owner can define usage restriction to their data, before sharing it, and data consumers must accept the usage restrictions.

[2] https://digital-strategy.ec.europa.eu/en/policies/strategy-data

The International Data Spaces Association (IDSA)[3] is a powerful alliance to drive the advancement of data sovereignty, secure data exchange, and standardized data interoperability in the digital ecosystem.

The International Data Space (IDS) is a virtual data space leveraging existing standards and technologies, as well as governance models well-accepted in the data economy, to facilitate secure and standardized data exchange and data linkage in a trusted business ecosystem. It thereby provides a basis for creating smart-scenarios and facilitating innovative cross-company business processes, while at the same time guaranteeing data sovereignty for data owners.[4]

IDSA has created the IDS Reference Architecture Model (RAM) (Fig.3), version 4.0[5] at the time of writing. Data and endpoints are described semantically in the IDS Information Model[6], and certification ensures that every system follows the architecture and uses the information model in the defined way.

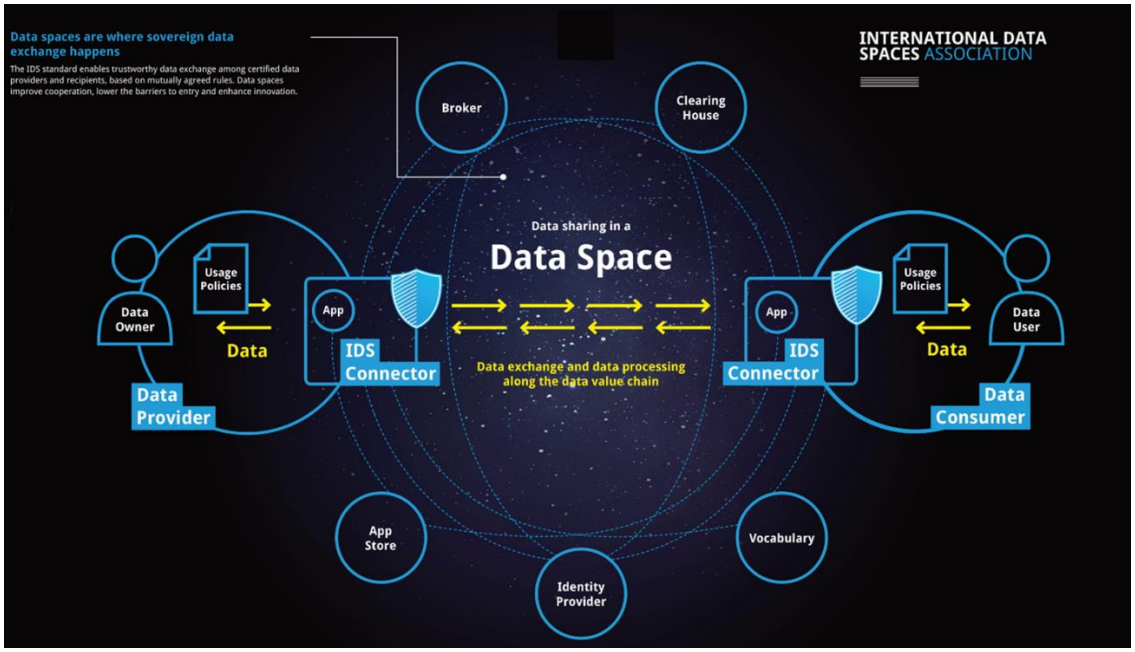

Figure 3: IDS Reference Architecture.

The IDS Connector is the core of the Data space. Is the gateway to connect existing systems and their data to an IDS ecosystem. The IDS Connector allows to exchange data and enrich it with metadata.

The Identity Provider should offer a service to create, maintain, manage, monitor, and validate identity information of and for participants in the IDS.

[3] https://internationaldataspaces.org/
[4]https://docs.internationaldataspaces.org/ids-knowledgebase/v/ids-ram-4/introduction/1_1_goals_of_the_international_data_spaces
[5] https://docs.internationaldataspaces.org/knowledge-base/ids-ram-4.0
[6]https://docs.internationaldataspaces.org/ids-knowledgebase/v/ids-ram-4/layers-of-the-reference-architecture-model/3-layers-of-the-reference-architecture-model/3_3_informationlayer

IDS Connectors and Identity Provider are mandatory components of the dataspace.

A Metadata Broker acts as a mediator between data providers offering data, and data users requesting data. It also acts as a data source registry.

The Clearing House provides clearing and settlement services for all financial and data exchange transactions. The Clearing House logs all activities performed in the course of a data exchange. The logging information can also be used to resolve conflicts.

The Industrial Data Space promotes the development of a business ecosystem in which participants may develop software (especially data services) and make this software available via the App Store. AppStore consists of a registry for available IDS Apps in this AppStore. IDS Apps are independent, functional, and re-usable software asset that is deployable, executable, and manageable on an IDS Connector.

In order to better define one's own data, domain-specific vocabularies can be created and made available for all IDS participants.

2.1.3 GAIA-X

Gaia-X association enables the transition from disjoint data & infrastructure ecosystems, to composable, interoperable & portable cross-sector data sets (or dataspace) and services based on a framework (DE-CIX Management GmbH, Eggers, & Fondermann, 2020). The association covers a large set of services including Energy and is based on three pillars:

- Compliance: for a common digital governance based on European values.
- Federation: enables interoperable & portable (Cross-) Sector datasets (within a dataspace) and services.
- Data exchange: A mean to perform data exchange and anchor data contract negotiation results into the infrastructure.

Gaia-X develops the software components necessary to sets up a federated system that inter-connects several Participants with each other, aiming to develop new services and innovative products.

Such ecosystems consist of joined interconnected data and infrastructure ecosystems, aggregated as 'Federators' by the concept of a Gaia-X Federation, and individually orchestrated and operated by a set of 'Federation Services' (Fig. 4).

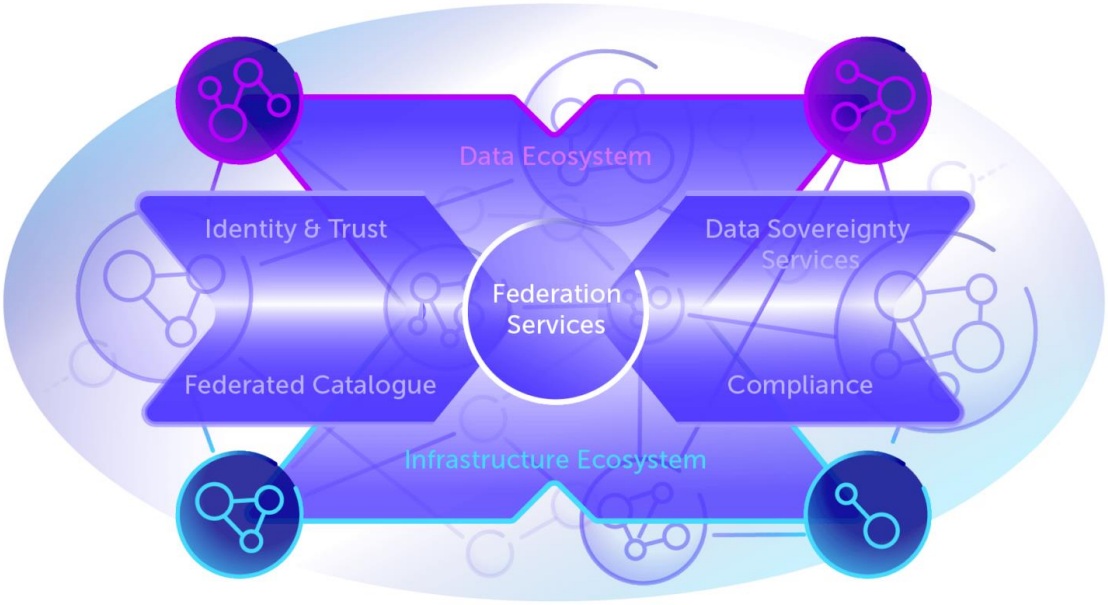

Figure 4: Federation Services (htt1111).

The reference architecture describes concepts and is not focused on specific technologies, required to be compatible with Gaia-X participant to the Data sharing Ecosystem. In other words, Gaia-X explains the Providers, Consumers, and essential Services to interact all togethers to share data. This data space notion is that data are not stored centrally, but rather at the source. Thus, they are only transferred through semantic interoperability as necessary. These Services comprise ensuring identities, implementing trust mechanisms, and providing usage control over data exchange and Compliance – without the need for individual agreements. In term of compatible technologies:

Identity systems standards supporting federation are OIDC/OAuth2 (draft), SAML, SPIFFE/SPIRE.

Self-Descriptions are W3C verifiable presentations in Json-LD format. Also, verifiable Credentials themselves contain a set of Claims: assertions about Entities expressed in the RDF data model.

2.1.4 FIWARE

FIWARE Foundation (https://www.fiware.org/) curates a catalogue of more than 60 open-source components for the management of data across multiple sectors including Energy. These components shared a common API, NGSI-LD, which is an open standard evolved by ETSI[7]. These components include several functions, from Identity and access management (with full alignment with DSBA and Gaia-X IAM approaches) with keyrock[8], keycloak[9]. AuthZForce[10], etc., to the gathering of information from heterogenous data sources, the context broker, with Orion-LD[11] Scorpio[12] and Stellio[13], or for the big data processing of data, Cosmos[14] and the retrieval of

[7] https://www.etsi.org/committee/cim
[8] https://fiware-idm.readthedocs.io/en/latest/.
[9] https://github.com/keycloak/keycloak#readme.
[10] https://authzforce-ce-fiware.rtfd.io/
[11] https://github.com/FIWARE/context.Orion-LD/tree/develop/doc/manuals-ld.
[12] https://scorpio.rtfd.io/.
[13] https://stellio.rtfd.io/
[14] https://fiware-cosmos-flink.readthedocs.io/ and https://fiware-cosmos-spark.readthedocs.io/

information from sensor and IoT devices. The core component, the context broker allows the gathering of this information and its querying in very different ways, geographically, time series and other thanks to the NGSI-LD standard. NGSI standard is a powerful general-purpose API and its full horizontal and vertical scalability. Federation between instances of brokers allows diverse configurations and scalability options.

Besides this alignment, that is related to the implemented mechanisms for the processing of data by using shared standards there is another alignment at the semantic level. FIWARE together with IUDX, OASC, and TMForum curates more than 1000 data models, at the Smart Data Models Initiativei most of them in the Energy domain.

The Smart Data Models are also included in the reference architecture. They provide a quick mechanism to create standardized data models, compiling, the attribute names of the object or classes defined, the data types of the information stored in them, the written description about their content and URIs for their use in linked data-based systems. Besides this, these data models are available in a python package, pysmartdatamodels[15], that include all the information for their integration into different systems but also some functions to allow its manipulation, extracting information, validation of payloads or even the insertion of data model compliant information into NGSI-based systems

The Smart Data Models is an initiative that will remain after the end of the ENERSHARE project and therefore, provides actual standardization for the domain

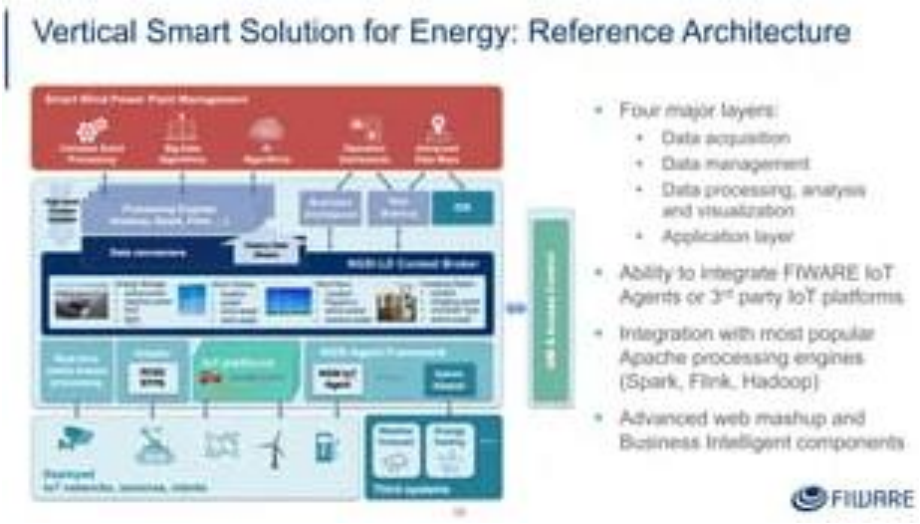

Figure 5: Vertical Smart Solution for Energy: Reference Architecture.

2.1.5 CEEDS Architecture Blueprint

The **int:net** project (https://intnet.eu/) has been cooperating with the five Energy DataSpace sister projects (ENERSHARE[16], OMEGA-X[17], DATA CELLAR[18], SYNERGIES[19], EDDIE[20]), hence forming

[15] https://pypi.org/project/pysmartdatamodels/.
[16] https://enershare.eu/
[17] https://omega-x.eu/
[18] https://datacellarproject.eu/
[19] https://synergies-project.eu/
[20] https://eddie.energy/

the **Energy Data Space Cluster Projects (EDSCP)**, and the Energy Community (EnC) to identify the specific vertical capabilities that are needed in an energy data space supporting environment. The result is the **CEEDS architecture blueprint** (Figure below) that is expected to meet the need of the domain.

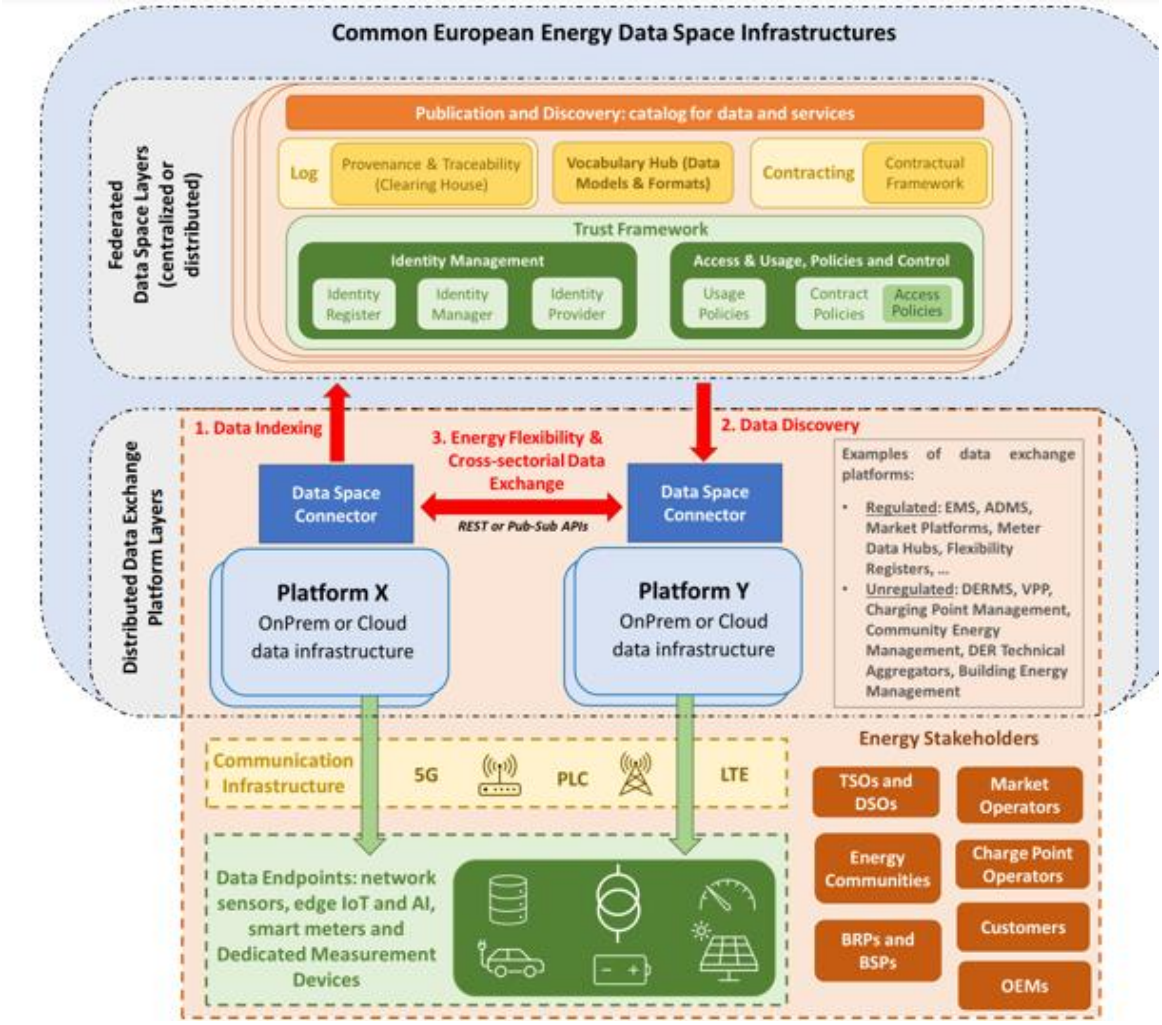

Figure 6: int:net CEED Architecture blueprint[21].

2.1.6 Data Connectors to support Interoperability

IDSA and FIWARE Foundation are cooperating to drive the advancement of data sovereignty, secure data exchange, and standardized data interoperability in the digital ecosystem. Standardized interoperability is essential to build up a Data Space because different data ecosystems will exchange different kinds of data in different formats and protocols. With standardized interoperability, every system can be interoperable in the IDS.

The **TRUsted Engineering (TRUE) Connector** (Figure 7) enables the trusted data exchange in order to be active part of an IDS Ecosystem. The TRUE Connector is composed of three components:

[21] https://intnet.eu/resources/technical-resources

- Execution Core Container (ECC), an open-source project designed by Engineering Ingegneria Informatica (ENG)22 in charge of the data exchange through the IDS ecosystem representing data using the IDS Information Model and interacting with an external Identity Provider. It is also able to communicate with an IDS Broker for registering and querying information.
- Back-End (BE) Data Application, an open-source project designed by ENG. It represents a trivial data application for generating and consuming data on top of the ECC component.
- Usage Control (UC) Data Application, a customized version of the Platoon project base application for integrating Usage Control functionality.

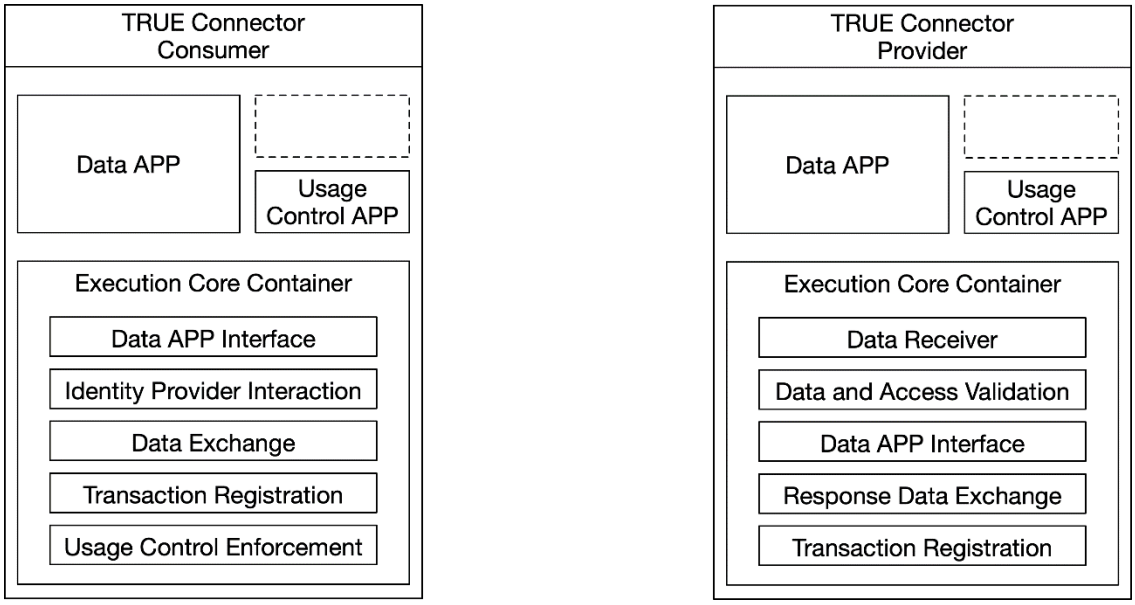

Figure 7 TRUE Connector architecture

The FIWARE TRUE Connector[23] is a customization of TRUE Connector that includes a NGSI-LD Data App to enable the data exchange using the FIWARE NGSI-LD Context Broker. As can be seen in the Fig. 8, data providers and data consumers have their own FIWARE platform with the Context Broker and the NGSI-LD Data App of the Prosumer converts in NGSI-LD standard context information and send it to the Consumer and vice versa.

[22] http://www.eng.it
[23] https://fiware-true-connector.readthedocs.io/en/latest/

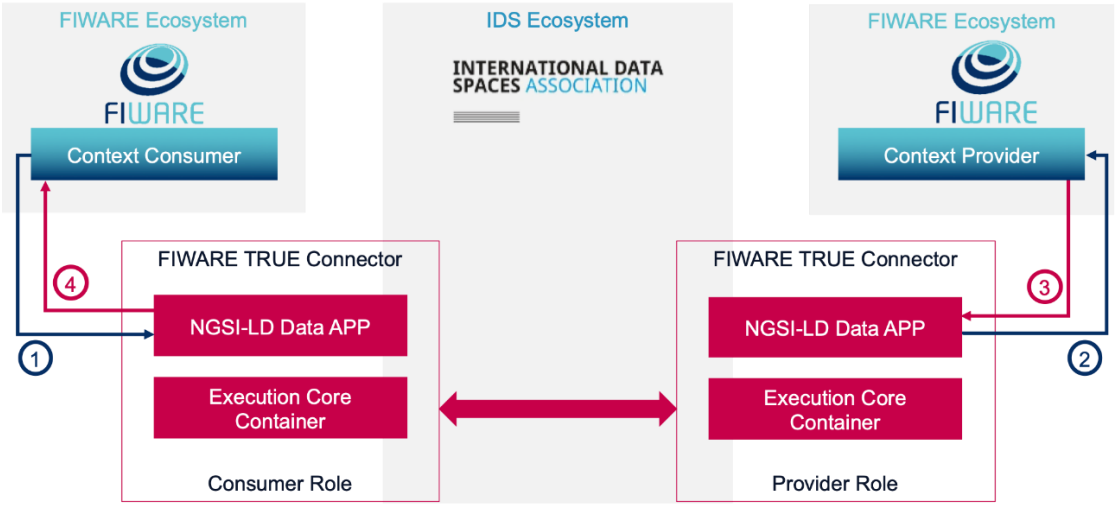

Figure 8: FIWARE TRUE Connector[23]

The **OneNet** connector[24], based on TRUE Connector and developed in the context of OneNet project[25], aims to enable a European Energy Data Space, combining the IDS principles with the advantages of the FIWARE ecosystem ensuring a seamless and secure data exchange in a completely end-to-end decentralized approach[26]. The OneNet Connector is ready to be deployed and integrated in any existing platform and offers user-friendly interfaces (both as REST APIs and GUI) enabling users and platforms to share data. In addition, the OneNet Connector offers a pre-defined, dynamically evolving list of Cross Platform Services, business objects and corresponding Data Profiles being available for enabling semantic and data interoperability.

2.2 ENPOWER High Level Architecture

The objective of this section is to illustrate and describe the high-level architecture of ENPOWER, delineating the specifications of each architectural layer:

- Business Layer
- Functional Layer (AI Services, DTs, DLT)
- Information Layer
- Communication Layer
- Component Layer
- Generic Security Layer, including Energy Data Space compliant tools for enforcing trust and sovereignty

Through this architecture, the ENPOWER solution lays the groundwork for informed decision-making and strategic alignment with organizational objectives. A comprehensive understanding of its architecture will be useful for effective implementation and utilization.

[24] OneNet Consortium, "Extended Interoperability and Management with FIWARE D6.3", 2023, [Online]. Available: https://www.onenet-project.eu//wp-content/uploads/2023/09/OneNet_D6.3_v1.0.pdf
[25] https://www.onenet-project.eu/
[26] International Data Space Association, "Data Connector Report", 2023, [Online]. Available: https://internationaldataspaces.org/wp-content/uploads/dlm_uploads/IDSA-Data-Connector-Report-5_March-2023.pdf

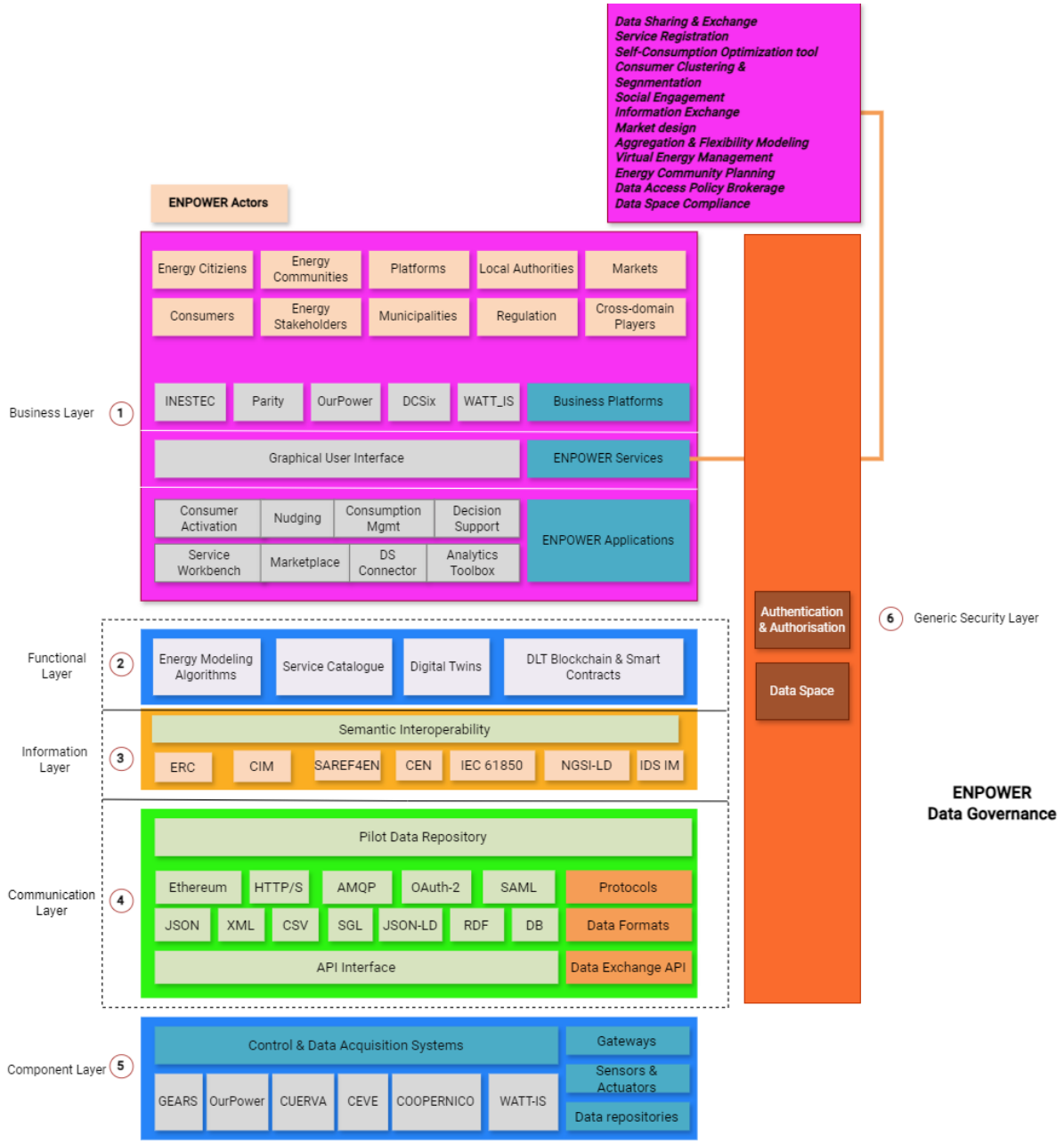

Figure 9: ENPOWER High Level Architecture.

Each layer of the Architecture represented in Figure 9 is described in a top-down way.

The **Business Layer** (number 1 in the figure) layer consists of

- all the value-added ENPOWER Services and Applications that enhance the functionality and capabilities of the ENPOWER project, while being tailored to the specific ENPOWER use cases and requirements.
- Local Business Platforms serving as interfaces for users to interact with the system, presenting insights, recommendations, and data in a user-friendly manner. They include 6 local platforms: OurPower Platform, CUERVA Platform, CEVE Platform, COOPERNICO Platform, Watt-IS Platform.

The ENPOWER Applications comprises the following components:

- the Consumer Activation, Consumption Management and Nudging which are aimed to influence the end consumer at individual and collective level, activate it to actively participate in the energy management of the community and manage their consumption.
- Decision Support to elaborate and predict different energy management scenarios to be presented and shared among the community members.
- The Services Workbench, which is a collection of tools, utilities, and services provided within the ENPOWER ecosystem. It offers a comprehensive set of functionalities for developing, deploying, and managing applications and services within the ENPOWER architecture.
- Tokenized blockchain-based marketplace to orchestrate the P2P heterogeneous tokenized assets sharing and compensation in the community
- Analytics Toolbox, to offer a riche environment to orchestrate and facilitate the tailored deployment of energy analytics.
- DS or DataSpace Connector to manage the interaction with the Energy and/or other external DataSpace with a view to gain benefits from the User Access Control Trust and sovereignty enforcing mechanisms as made available by the ENERSHARE Energy Data Space (NTUA, COMS, INESC TEC, ED common partners).

The ENPOWER Services includes all the technological capabilities which are offered to the EnCs and/or technology providers to manage EnCs, as reported in the Figure.

Building upon the lower foundational layers is the Functional Layer (number 2 in the figure), housing advanced AI services, Digital Twins (DTs), and DLT blockchain/smart contracts-based components, which harness data to derive actionable insights and optimize processes. The Functional Layer (AI Services, DTs, DLT/Blockchain/Smart Contracts) has the goal to allow the upper Business Layer to deploy data-driven EnC application and services that are part of the project. Such layer incorporates added-value libraries of functionalities and technology enablers which will be customized to support ENPOWER Application and Services, and includes

- Energy & Flexibility Modeling Data-Driven Algorithms for predicting Energy generation, consumprion and management at community level.
- Service Catalogue to access the appropriate service.
- Digital Twins (DTs) as virtual representations of physical objects, processes, or systems, which mimic the behaviour of their real-world counterparts, allowing for simulation, analysis, and optimization of various scenarios without directly impacting the physical environment.
- DLT/Blockchain/smart Contracts capabilities to support ENPOWER Data Governance & Management.

The Information Layer (number 3 in the figure) acts as a bridge, enabling smooth communication and semantically enriched data exchange. Such layer represents information objects and/or data models necessary to fulfil functions and to be shared/exchanged.

Such layer encompasses the capability to manage Semantic interoperability, which aims to enable mutual understanding among two or more systems by employing shared data models. This entails structuring data exchanges and encoding data with standardized vocabularies, allowing

receiving systems to interpret the data in accordance with the intended semantics. It comprises the capability to manage ontologies like SAREF4ENERGY, CIM families of models, including IEC61850, NGSI-LD and the ENPOWER Ontology. These components are instrumental in structuring and organizing domain-specific knowledge and data, enabling intelligent decision-making and analysis within the system.

The Information Layer relates to the Energy Data Space facilitating interoperability with external data spaces, allowing integration with external systems and data sources.

The following two layers establish the groundwork for upper layers to leverage and build upon, enabling higher-level functionalities such as data analysis, decision-making, intelligent management and smart control.

The Communication Layer (number 4 in the figure) serves as conduits for data collection and transmission, ensuring seamless integration into the system and offering the necessary southbound infrastructure backbone for upper technical and semantic interoperability. This layer includes protocols, mechanisms (such as Data Formats, Data Exchange APIs) for exchanging information among virtual data repositories/hubs, including all the data sources generated at local pilot -level from IoT and/or physical devices and from third party Control & Data Acquisition Systems (es. SCADA)

Main communication protocols and data format are, among the others:

- The MQTT (Message Queuing Telemetry Transport) protocol is a lightweight messaging protocol designed for constrained devices and low-bandwidth, high-latency, or unreliable networks. The MQTT protocol is well-known in the IoT environment due to its features.
- The FTP (File Transfer Protocol) protocol it is a standard network protocol designed for transferring files between a client and a server.
- CSV (Comma-Separated Values) is a file format used for storing tabular data.
- JSON (JavaScript Object Notation) is a lightweight data interchange format based on the key-value pairs paradigm. It is well-known for its simplicity, readability, and ease of use across different programming languages and platforms
- Ethereum for the blockchain-based transactions

The Communication Layer addresses data technical interoperability, ensuring efficient handling and distribution of data across various layers and components. The technical interoperability comprises the FIWARE Context Broker, which collectively manage and facilitate the exchange of data within the system.

The Component Layer (number 5 in the figure) addresses the Basic Connectivity and includes physical components (physical objects, like smart meters, sensors, IoT devices, actuators with local host capabilities and the underlying low-level communication capabilities, including the & Data Acquisition Systems, serving as conduits for data collection and transmission.

Each pilot encompasses a distinct array of devices, offering the potential to gather disparate datasets through the Component Layer.

The Generic Security Layer (number 6 in the figure) is positioned as a transversal layer across the entire architecture; hence it is linked with all the other five above-described layers. It ensures the security and integrity of data, processes, and interactions within the system, implementing robust

security measures to safeguard against threats and vulnerabilities. The layer comprises two main components:

- The Authentication & Authorization component, aimed to
    - Implement security measures to ensure the integrity, confidentiality, and authenticity of data transmitted between IoT devices and the system.
    - Enforce authentication and authorization mechanisms to control access to sensitive data and functionalities.
- The Energy Data Space component, which virtualize the access to the ENERSHARE Energy Data Space to exploit the Usage Access Policy Control Building Blocks to increase trust and effectively manage to keep end users full in control of their data.

Such component is a two way one, making possible via Open APIs to integrate the EnC's interoperability extensions planned in the ENPOWER projects and some of the above identified ENPOWER services and applications in the Energy Data Space

This architecture provides a comprehensive framework for building and deploying a system that leverages IoT data, knowledge management, value-added services, and presentation interfaces while prioritizing security and interoperability.

3. Privacy, security and legal compliance framework (Task 4.1)

The privacy, security and legal compliance framework is a horizontal component providing guidelines, methodologies, techniques, and tools to ensure privacy, security and legal compliance during the design and operational phases. The framework cowers aspects, such as identity management, access management, encryption, authentication, authorization, anonymisation, auditing and others. Special care is taken in case of AI technologies present and developed in the ENPOWER project by referencing the definitions and guidelines defined by the legal framework for AI (EC COM /2021/206[27]).

3.1 Privacy Protection

The processing of personal data must be explicitly allowed by law or consent given by the involved data subject (data owner). Consent is one of the legal bases for data processing introduced by a GDPR. A data subject must provide consent prior to any personal data collection and processing. Consent of the data subject as defined in Article 4 of GDPR is “any freely given, specific, informed and unambiguous indication of the data subject’s wishes by which he or she, by a statement or by a clear affirmative action, signifies agreement to the processing of personal data relation to him or her” [1].

Consent must be willingly and clearly given for it to be valid under GDPR. This means the person must freely offer it without any coercion, fully understanding what they're agreeing to, specifically what data will be used and how. They need to know who is collecting their data and why, including their right to withdraw consent anytime. Consent requires a clear, affirmative action, meaning it can't just be assumed. While there's no set duration for consent, revoking it should be as easy as giving it. If a third party provides consent, they must prove their authority to do so.

Privacy can be represented by the ability to hide information and disseminate information by choice, especially when the information is sensitive. Any data handling is associated with a certain degree of risk of data leakage. Data can be exposed both while in storage as well as during transit. Adequate security measures must be developed to prevent unauthorized data access. In one hand, data access policies may be used to define the responsibilities and roles of actors who are granted access to data, on the other hand, data encryption is a very effective method for data protection. It is a process of encoding information by converting a plaintext into a ciphertext usually performed by algorithms based on pseudo-random keys.

Auditability is defined as an auditor’s capacity to realize a comprehensive review of records. The rise of blockchain technologies enable improved accountability and audit capabilities. Blockchain technologies ensure authenticity of records and non-repudiation using public key infrastructure and digital signatures. The immutability of ledgers with the transactions provides integrity by preventing alteration of the records.

[27] https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:52021PC0206

Privacy protection can also be achieved through anonymization, where identifying and quasi-identifying attributes are removed for data. It can be also achieved by replacing the identifiers by pseudonymity, but the permanent unique identifiers can still pose a privacy risk.

3.2 GDPR

Privacy and data protection are recognised as fundamental rights in EU, which makes the protection of privacy and personal data a priority in the project. According to article 4 of GDPR [1], personal data is defined as "any information relating to an identified or identifiable natural person ('data subject'); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person".

The identification of the privacy and data protection requirements should consider the following:

Smart grid and smart meters belong to complex IT infrastructures that require not only to gather, collect, and analyse data, but also to exchange the same data with other IT infrastructures. In that respect, considering the source from which such data will be gathered/collected/extracted it is possible that among those data also personal information, i.e. personal data, might be included in the information exchange cycle. It is necessary that a balancing operation among two conflicting interests (privacy and data protection on one side and security on the other side) is performed.

Privacy and data protection rights are two fundamental rights recognized at European level. In particular, the inclusion of protection of personal data in EU primary legislative texts such as the Treaty of Functioning of the European Union (TFEU), as well as the Charter, enabled the same EU to provide for effective legislative instruments (e.g. among the other, GDPR) to protect personal data.

The interests or the values protected by the right to privacy and data protection rights for the sake of clarity are not the same. Privacy can be considered as providing for a general prohibition of interference in the private life of an individual. On the other hand, the protection of personal data, can be instead intended as a complete system of rights, to be balanced and to be exercised and activated only when personal data (i.e. information that can directly or indirectly identify a person) are processed. This means that data protection rules and requirements shall be held in check even when the processing personal data do not interfere with the privacy of the individual.

As of today, GDPR is the most relevant EU legislative source in terms of providing the rules and principles that should be respected when personal data are processed.

The main privacy and data protection requirements are presented in Table 1.

Table 1 Privacy and data protection requirements.

Requirement	Description
Transparency	The purposes of the data processing should appear clear and intelligible for the data subject. This can be ensured providing all the appropriate and necessary information to data subjects to exercise their rights, to data controllers to evaluate their processors, and to Data Protection Authorities to monitor according to responsibilities. **The technology solutions and their data models should ensure that a data subject have easy**

	access to that information at any time. All that information should be made available to the data subjects in a clear and intelligible way.
Lawful data collection	The data processing shall originate only from the personal data that have been collected with a lawful ground. Particular attention should be paid when implementing the components that will help to collect and get the data subject's consent. In this respect, the relevant partner should ensure the possibility to map the data flow. Particular attention should be given in case of secondary processing (even if, at the time of submission, this kind of operations are not foreseen).
Personal data collected are: adequate, proportionate, relevant to the objectives of the system.	The implementation of the principle of purpose limitation and data minimisation (core principles in GDPR) requires that the amount of data collected shall be proportionate to the purposes to be achieved, and at the same time, the purpose itself shall be legitimate. In this respect, data should be gathered if and only if it is strictly necessary for achieving the specified purpose.
The personal data collected are accurate	The data to be processed needs to be accurate, i.e. data are correct and up to date in all details.
Storage limitation	The development team of the technology solution should define and implement an infrastructure to which it is possible to foresee for how long the personal data will be stored. Data subjects must be informed about it. In case the data are no longer necessary to fulfil the scope of technology solution such data should be immediately erased and/or anonymised according to the best standards and practices.
Procedures for granting individual rights	The components of the technology solutions should be designed taking also into consideration how concretely the relevant data subject might exercise his/her rights in connection with the data processing. The relevant partner should be aware of all the rights that GDPR grants to data subjects, and for each of them tailor a specific solution (e.g. data subjects have the right to rectify their data and to request their erasure).
Accountability principle and technical implementation	The implementation of the accountability principle entails that the technology solutions should allow a clear identification of the responsibilities related to the data processing. Examples of accountability measures are related to tracking of personal data access and tracking of communications with external systems. In addition, the accountability principle implies the set-up of internal audits and handle complaints procedures.
Implementation of security measures	Information security addresses integrity, confidentiality, and availability concerns. IT measures such as privacy enhancing technologies (PETs) represent an important tool for privacy and data protection, in terms of implementing technology solutions able to restrict access to personal data only to authorized people (e.g. permissions), and to ensure that the data is trustworthy and accurate (e.g. based on provenance information). The relevant partner should: Regularly conduct privacy risk assessment and audit processes.

	Regularly run reviews of the security measures implemented. Design an ad hoc procedure to be followed in case of data breach.

3.3 Legal Framework for AI

EC COM /2021/206[28] is the regulation regarding a legal framework for AI that takes a risk-based approach to solve challenges of AI related to safety and liability. In a lifecycle of AI-based solutions it is important to ensure that security, traceability and explainability are handled throughout the lifecycle. The AI lifecycle contains four major steps: development, training, deployment, and operation. ENPOWER consortium will address the following guidelines:

Implement systematic software development practices that focus on quality management and establishing confidence in the maturity of data collection and training processes and deployment of AI.

Prioritise explainable AI even at the expense of accuracy.

When applicable, use robust inference solutions as countermeasures to deal with unmodelled phenomena or adopt uncertainty presentations.

Conduct risk assessment during the design phase, mapping the AI-based system internal decision-making process to the system control using Failure Attack and Countermeasure (FACT) graph[29] and mitigation strategies from ongoing initiatives as ETSI Industry Specification Group (ISG) Secure AI[30] (SAI).

Assess robustness and reliability with digital twins before deployment of solutions (it is difficult to predict system behaviour for scenarios vastly different from what the system was trained to and tested against).

Ensure that humans are active part of the design and safety assessment of AI in the pilots' validation to undergo functional testing to validate safety mechanisms and make sure that safety of installations and humans is always the highest priority.

The regulatory framework on Artificial Intelligence (EC COM /2021/206) has the following objectives:

- Ensure that AI systems placed on the EU market and used are safe and respect existing law on fundamental rights and EU values.
- Ensure legal certainty to facilitate investment and innovation in AI.
- Enhance governance and effective enforcement of existing law on fundamental rights and safety requirements applicable to AI systems.
- Facilitate the development of a single market for lawful, safe and trustworthy AI applications and prevent market fragmentation.

The framework presents regulatory approach to AI that is limited to the minimum necessary requirements to address the risks and problems linked to AI, without unduly constraining the

[28] https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:52021PC0206
[29] Cui, J., & Sabaliauskaite, G. (2017). On the alignment of safety and security for autonomous vehicles. Proc. IARIA CYBER, 1-6.
[30] Industry Specification Group (ISG) Securing Artificial Intelligence (SAI), https://www.etsi.org/committee/sai

hindering technological development or otherwise disproportionately increasing the cost of placing AI solutions to the market.

For non-high-risk AI systems, it is obligatory to flag the use of an AI system when interacting with humans. For high-risk AI systems, the following requirements are strictly necessary to mitigate risks to fundamental rights and safety posed by AI and that are not covered by the existing legal frameworks: high quality data, documentation and traceability, transparency, human oversight, accuracy and robustness.

The legal framework for AI enhances and promotes the protection of the rights protected by the EU Charter of Fundamental Rights:

- The right to human dignity.
- Respect to private and family life.
- Protection of personal data.
- Non-discrimination.
- Equality between men and women.

3.3.1 Prohibited AI Practices

The legal framework for AI defines the prohibited AI practices which are the following:

- Use (placing on the market, putting into service or use) of an AI system that deploys subliminal techniques beyond a person's consciousness to materially distort a person's behaviour in a manner that causes or is likely to cause that person or another person physical or psychological harm.
- Use of an AI system that exploits any of the vulnerabilities of a specific group of persons to materially distort the behaviour of a person pertaining to that group in a manner that causes or is likely to cause that person or another person physical or psychological harm.
- Use of AI systems by public authorities or on their behalf for the evaluation or classification of the trustworthiness of natural persons over a certain period of time based on their social behaviour or known or predicted personal or personality characteristics, with the social score leading to either:
    - detrimental or unfavourable treatment of certain natural persons or whole groups thereof in social contexts which are unrelated to the contexts in which the data was originally generated or collected.
    - detrimental or unfavourable treatment of certain natural persons or whole groups thereof that is unjustified or disproportionate to their social behaviour or its gravity.
- The use of real-time remote biometric identification systems in publicly accessible spaces for the purpose of law enforcement, unless and in as far as such use is strictly necessary for one of the specific purposes.

3.3.2 High-risk AI systems

AI systems are considered high-risk AI systems, if they are listed in any of the following areas:

- biometric identification and categorisation of natural persons
- management and operation of critical infrastructure
- education and vocational training
- employment, workers management and access to self-employment
- access to and enjoyment of essential private services and public services and benefits
- law enforcement

- migration, asylum, and border control management.
- administration of justice and democratic processes.

AI system is also considered a high-risk AI system, when both of the following conditions are fulfilled:

- the AI system is intended to be used as a safety component of a product, or is itself a product, covered by the list Union harmonisation legislation listed in the supplementary material.
- the product whose safety component is the AI system, or the AI system itself as a product, is required to undergo a third-party conformity assessment with a view to the placing on the market or putting into service of that product.

The AI system shall be considered a high-risk AI system, when it is intended to be used in any of the area listed in first eight points previously listed in the document and when AI system poses a risk of harm to the health and safety, or a risk of adverse impact on fundamental rights, that is equivalent to or greater than the risk of harm or of adverse impact posed by the high-risk AI systems from the 8 areas previously listed in the document.

3.4 Identity and Access Management

Identity and access management (IAM) is a framework of policies and procedures that provide the security to the applications and services by supporting concepts such as authentication and authorisation. IAM system provides security of identities and attributes ensuring the right individuals have access to the right resources at the right time and for the right reasons. The IAM systems can perform functions such as administration, discovery, policy enforcement, information exchange and authentication and basically, they are used to authenticate users, devices or services and grant or deny the rights to access data and other system resources.

The key components of IAM systems include identity management where digital identities of users are being created and managed, access management with the concepts of authentication (verifying identity of the user) and authorisation (granting or denying access to the resource), governance and compliance with policy management and audit functionality and federation component supporting federated identities allowing users to use their credentials from other IAM provider (Google, Facebook, etc.) to access resources on a selected domain.

3.4.1 OAuth2 Protocol

OAuth2 (Open Authorization) is an authorization framework that enables applications to access resources hosted by resource servers on behalf of a user. It works by delegating user authentication to the service and authorising third-party applications to access that user's account. OAuth2 protocol defines four actor roles in the authorization framework:

- Resource Owner: The user who authorizes an application to access their account.
- Client: The application requesting access to user's resources.
- Resource Server: The server hosting the protected resources.
- Authorization Server: The server that authenticates the user and issues access tokens to the client.

The OAuth2 authorization protocol flow contains the following steps, which are presented in Figure 10.

1. The application (Client) requests authorization to access protected resource owned by the user (resource owner - RO).

2. If RO authorizes the request, the application receives the authorization grant.
3. Client (application) makes a request to authorization server (AS) to get an access token by presenting authentication for its own identity and the authorization grant from the previous step.
4. If both client identity authentication and authorization grant are valid, the AS issues an access token to the client.
5. The client requests the protected resource from the RO by presenting the access token.
6. If the access token is valid, the RO serves the protected resource to the client (application).

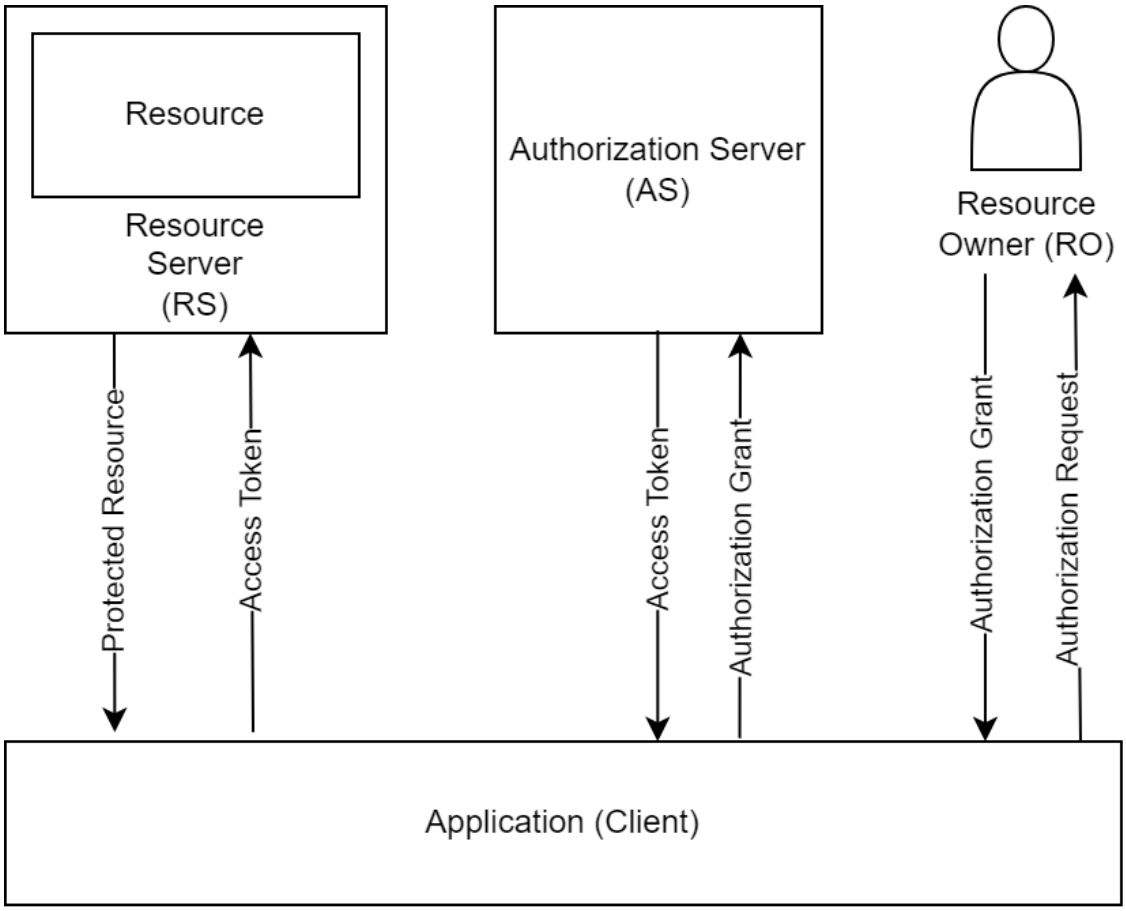

Figure 10: OAuth2 authorization protocol flow

The authorization grant type depends on the method used by the client and the grant types supported by the API. OAuth2 defines the following grant types:

- **Authorization Code Grant**: AS returns a single-use Authorization Code to the Client, which is later exchanged for an Access Token. This is best suited for traditional web apps where the exchange happens securely on the server side without exposing Client Secret.
- **Implicit Grant**: Access token is returned directly to the Client. The AS may return the Access Token as a parameter in the callback URI or as a response to a form post. The first option is deprecated due to potential token leakage.
- **Authorization Code Grant with Proof Key for Code Exchange (PKCE)**: This authorization flow is the extended version of Authorization Code Grant authorization flow

with the additional steps that make it more secure for mobile and native apps, where app doesn't run at server side.

- **Resource Owner Credentials Grant Type**: This grant requires the Client to acquire the RO's credentials, which are passed to the AS. It is limited only to the Clients that can be completely trusted. No redirect to the AS is involved.
- **Client Credentials Grant Type**: Used for non-interactive applications such as automated processes, microservices, etc. The application is authenticated by using its client id and Client Secret.
- **Device Authorization Flow**: This grant enables the use of authorization flow by apps on input-constrained devices (e.g., smart TV, etc.).
- **Refresh Token Grant**: It involves the exchange of a Refresh Token for a new Access Token.

3.4.2 UMA2 Protocol

UMA2 (User-Managed Access) is a federated authorization protocol built on top of OAuth2 which allows resource owner to control access to their resources by setting authorization policies on an authorization server. This enables fine-grained and user-centric approach to data sharing and access control. UMA2 protocol defines the following components and roles:

- **Resource Owner:** The entity that owns the resource and has authority to grant access.
- **Requesting Party:** The person or entity requesting access to the protected resource.
- **Client:** The application used by the Requesting Party to access the protected resource.
- **Resource Server:** The server that hosts the protected resource.
- **Authorization Server:** The server that handles authorization requests and enforces the policies set by the resource owner.

UMA2 authorization protocol contains the following key authorization steps:

1. **Resource Registration:** The resource owner registers resource with the resource server, which then communicates with the authorization server to obtain a Protection API Access Token (PAT). PAT token allows the resource server to manage resources and permissions.
2. **Permission Request:** When client requests access to a protected resource, the resource server issues a permission ticket if the request lacks a valid Requesting Party Token (RPT). This ticket is sent to the client along with the authorization server's address.
3. **Token Request:** The client presents the permission ticket to the authorization server, requesting an RPT. The authorization server evaluates the request against the policies set by the resource owner and, in case of approval, issues a RPT to the client.
4. **Access Granting:** The client uses the RPT to access the resource on the resource server. The resource server validates the RPT with the authorization server before granting access.

The UMA2 authorization flow can be further split into the following steps which are presented in Figure 11:

1. The client initiates a request to access the protected resource at the resource server.
2. The resource server requests a permission ticket from authorization server by sending it the requested scopes and registered resource id.
3. The resource server sends back an error response to the client including the authorization server address and a permission ticket.

4. The client optionally requests a Requesting Party Token (RPT) directly from the authorization server using the permission ticket.
5. The client redirects the user agent to the authorization endpoint on the Authorization Server to request a token.
6. The Authorization Server redirects the user-agent to the client redirection URI including an updated permission ticket.
7. The client requests a RPT token from the Authorization Server using the updated permission ticket.
8. The Authorization Server responds with the RPT and a Persisted Claims Token (PCT).
9. The client sends a request to Resource Server including RPT token.
10. Resource Server optionally requests Authorization Server to validate RPT or do that locally depending on the token's format.

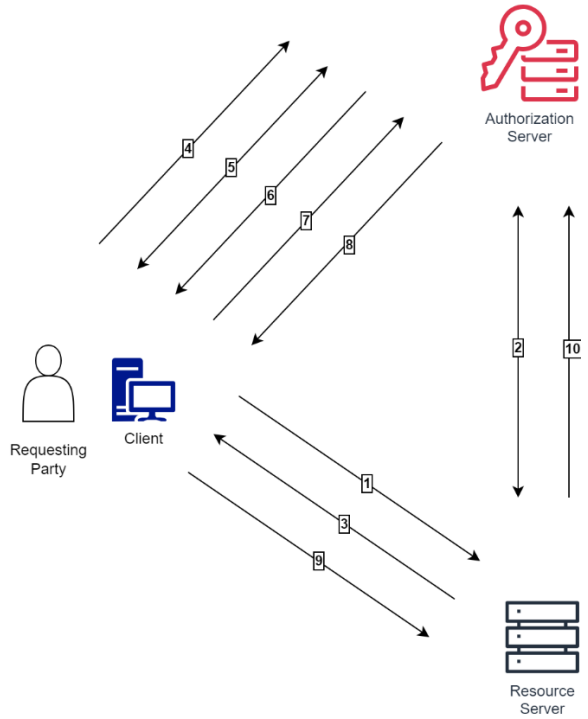

Figure 11: UMA2 authorization protocol flow.

The UMA2 standard introduces the separation of the requesting party and the resource owner in contrast to the OAuth2 protocol, where requesting party and resource owner are considered single person. This differentiation allows the following use-cases:

- A resource owner can share the resource with other people.
- An application owner can design rules allowing certain users and services to access a resource.
- A resource owner can aggregate the management of resource sharing under a single authorization server even in a case when different resources live in many domains.
- An application can get additional permissions and upgrade the access token scopes without involving the user agent if the authorization policies allow for it.

- All previous permissions can be granted asynchronously. The resource owner can grant or revoke access without direct interaction with the resource server at the time of request.

3.4.3 OpenID Connect

OpenID Connect (OIDC) is an identity layer built on top of the OAuth2 framework which allows third-party applications to verify the identity of the end-user and to obtain basic user profile information. It is used as a user authentication technology whose purpose is to give users one login for multiple sites.

Each time the user needs to log in using OIDC, the user is being redirected to the OpenID site, where the user logs in, and then is taken back to the website. The technology enables using other identity providers such as for example Google to authenticate with it and authorize the authorization framework (Auth0, Keycloak, etc.) to access user's information. Identity provider (e.g., Google) send requested user and authentication information back to authorization framework in a form of a JSON web token (JWT) and the user will get an access token.

OpenID Connect authentication flow contains the following steps:

1. **Authorization Request:** The relying party (RP) redirects the user to the authorization endpoint of the OpenID provider (OP).
2. **User Authentication:** The user authenticates with the OP using username and password.
3. **Authorization Response:** The OP redirects the user back to the client (RP) with an authorization code.
4. **Token Request:** The client (RP) exchanges the authorization code for an ID token and access token at the token endpoint (OP).
5. **Token Response:** OP returns the ID token and access token to the client (RP).
6. **UserInfo Request:** The client (RP) may use the access token to request additional user information from the UserInfo endpoint.

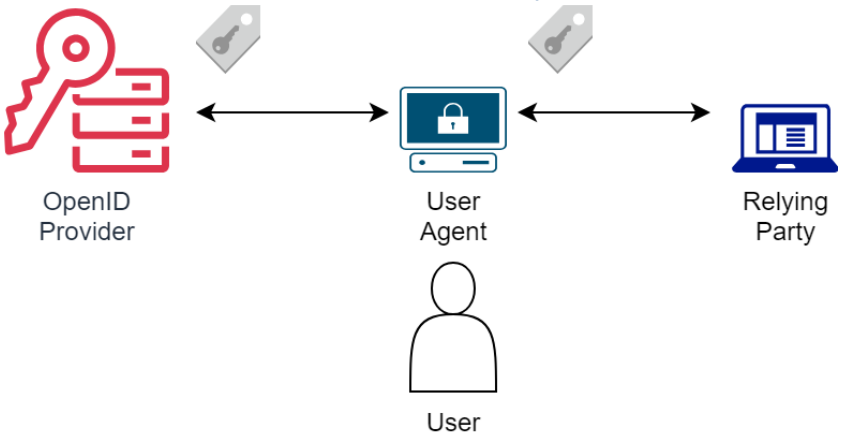

Figure 12: OpenID Connect actors.

3.5 Application Programming Interface Gateway

Implementing advanced security measures in services developed by various stakeholders imposes great deal of complexity and reduces the flexibility and scalability of the systems. Traditionally, developers must integrate all security measures in the applications and services

according to the system architecture which makes the system rigid and hard to adapt to new system configurations.

To separate the application development from the security mechanisms, typically, a design pattern from microservices architecture, particularly in the context of service meshes, sidecar proxies can be used. More generally for the API security, a concept of API gateway is introduced.

Both approaches separate the application from the security mechanisms where sidecar proxy or API gateway intercepts and manages all incoming and outgoing traffic for the API/service while taking care of security, observability and control, leaving application free from the security concepts. That enables developers to focus on implementing application's/service's business logic instead of dealing with the security aspects, leaving the security aspects to the data plane and control plane which enhances flexibility, control, and scalability of the system.

Sidecar proxies and API gateways provide load balancing and routing to the services, provides mTLS for encryption and decryption of traffic ensuring secure communication between services and provides authentication and authorization for access management. To provide observability, they typically provide logging, tracing and metrics on service performance.

General API Gateway architecture is presented in Figure 13, where application with its API resides behind the API gateway which provides the privacy and security protection to the otherwise unprotected application. Sidecar proxy such as for example Envoy is coupled with Keycloak for identity and access management (IAM), PostgreSQL as a persistent data layer, API control plane for API Gateway management and admin GUI to provide human-machine interface for privacy and security management.

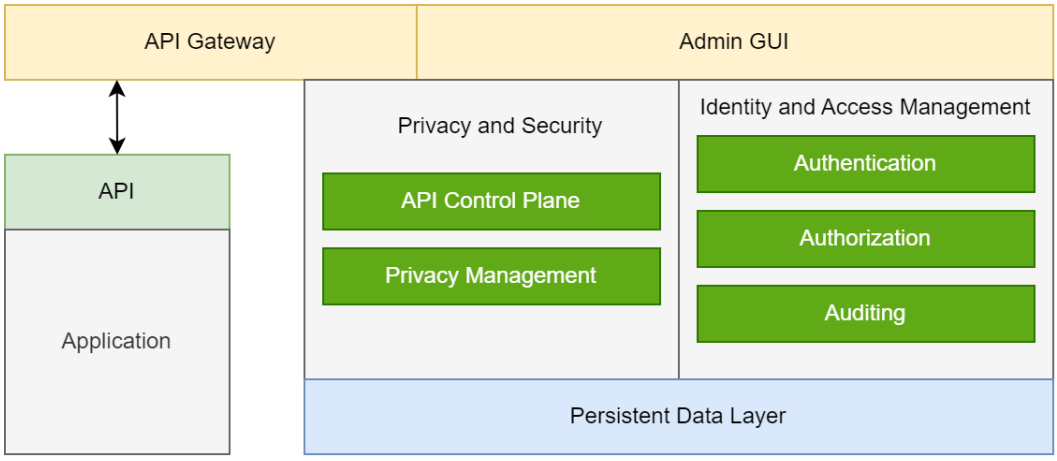

Figure 13: A concept of API Gateway system architecture.

3.5.1 API Gateway Data Flow

API Gateway provides mechanisms to protect otherwise unprotected application's API by acting as an intermediary component and a proxy. In a context of relations of different actors in the data flow, we can identify three significant roles in the API Gateway system. User is typically a data subject and resource owner because he/she owns his/her own data (personal data and other data). Resource owner also delegates, who can access his/her data. User data can be interpreted as a protected resource in a context of secure privacy-preserving application.

Protected resource is being exposed by unprotected API which is being protected by the API gateway containing privacy enforcement point (PEP) proxy (e.g. Envoy) in a data plane and a collection of tools in a control plane. Control plane contains IAM system (e.g. Keycloak), IAM and control user interface and auditing mechanisms.

Data in API Gateway implementation is exposed to external actors (clients) which act on behalf of external actors (users) such as data processors and data consumers. By using the IAM and control interface, data owner gives consent to selected users for specific use cases (data collection, data processing, etc.) and can also revoke the consent. The IAM and control interface gives user fine-grained control over his/her protected resources.

When the external actor makes a request to access the protected resources behind the API Gateway, the PEP proxy checks with the IAM provider (Keycloak) if the client/user was given a consent by the resource owner to access his/her data. If the client has been granted access to the requested resource by the resource owner, IAM provider gives authorization to the external actor to the protected resource and PEP proxy redirects external actor to the protected resource. A diagram of API Gateway and actors in a system is presented in Figure 14.

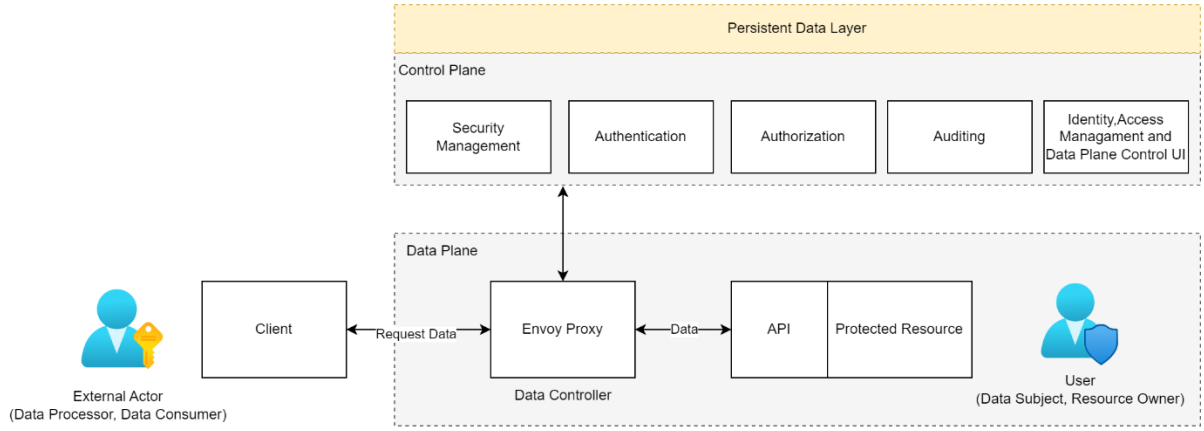

Figure 14: API Gateway architecture, relations and involved parties.

3.5.2 Technological Components

API Gateway component is built around verified and acknowledged open-source tools such as Envoy proxy, Keycloak, and PostgreSQL. Envoy in API Gateway configuration provides management and routing of external traffic to internal services, providing features like TLS termination, request/response transformation, rate limiting, content caching, authentication, and authorization. Keycloak is used as a reliable and configurable identity and access management provider where PostgreSQL data base is used as persistent data layer. Persistent data layer is used by Keycloak to store configurations, user identities and authorization settings (consents, access management, etc.). It also provides storage for all events to support auditing at the service-level, which include logins, data access events, data requests, requests for data access, consent requests, and others.

3.5.2.1 Keycloak

Keycloak is an open-source identity and access management solution which supports adding authentication to applications and secure services with minimal effort. It provides features such as single sign-on, identity brokering, and user federation, making it a comprehensive solution for managing user identities and access in both enterprise and development environments.

Its vast number of features include the following:

- **Single Sign-On (SSO)**: authenticate once and gain access to multiple applications.
- **Support for standard protocols**: OpenID Connect, OAuth2, SAML2.
- **Identity Brokering and Social Login**: supports integration with external identity providers (Google, Facebook, etc.) and enterprise identity providers.
- **User federation**: integrates with existing LDAP or Active Directory servers for user management and synchronizes user data between Keycloak and external databases.
- **Authorization services**: provides fine-grained authorization services and supports role-based access control (RBAC) and attribute-based access control (ABAC).
- **Customizable User Interfaces**: provide customizable login, registration and account management pages.
- **Admin console**: provides web-based admin console for managing users, roles and configurations.

3.5.2.2 Envoy Proxy

Envoy is a modern L7 proxy and communication bus designed for modern service-oriented architectures[31]. It brings the network transparency to applications by providing the following high-level features:

- **Out-of-process architecture**: Envoy is a self-contained process that runs alongside the application server. Envoy works with any application language (network separated from application) and supports quick and transparent deployments and updates across an entire infrastructure.
- **L3/L4 filter architecture**: A pluggable filter chain mechanisms allows filters to perform different TCP/UDP tasks.
- **HTTP L7 filter architectur**e: HTTP filters can be plugged to perform tasks such as buffering, rate limiting, routing, forwarding, etc.
- **First class HTTP/2 support**: Envoy supports both HTTP/1.1 and HTTP/2 and can operate as a transparent proxy in both directions.
- **HTTP/3 support**: Envoy supports HTTP/3 upstream and downstream with translations between any combination of HTTP/1.1, HTTP/2 and HTTP/3 in either direction.
- **HTTP L7 routing**: Envoy supports a routing subsystem capable of routing and redirecting requests based on path, authority, content type, etc.
- **gRPC support**.
- **Service discovery and dynamic configuration**.
- **Health checking**.
- **Advanced load balancing**.
- **Front/edge proxy support**: Envoy has a feature set that makes it well suited as an edge proxy including TLS termination, HTTP/1.1, HTTP/2 and HTTP/3 support and HTTP L7 routing.
- **Best in class observability**.

3.5.2.3 PostgreSQL

PostgreSQL is a powerful open-source object-relational database management system (ORDBMS) known for its robustness, scalability and standards compliance. It supports a wide

[31] https://www.envoyproxy.io

range of features that make it suitable for various applications, from simple web applications to complex data warehousing and analytics systems. PostgreSQL key features:

- **ACID compliance**: ensures reliable transactions through Atomicity, Consistency, Isolation, and Durability.
- **Extensibility**: supports extensions, custom functions, data types, operators, and procedural languages.
- **Advanced data types**: a variety of data types are supported including JSON/JSONB, XML, arrays and key-value pairs.
- **Full-Text Search**: built-in support for full-text search capabilities.
- **Concurrency and performance**: Multi-Version Concurrency Control (MVCC) handles concurrent transactions efficiently.
- **SQL compliance**.
- **Replication and high availability**: supports streaming replication, logical replication and various high-availability solutions.
- **Security**: features robust security with SSL, authentication methods, and row-level security.

4. Data models and service/platform interoperability (Task 4.2)

The main objective of this task is to provide the means and tools for the semantic interoperability among the energy services and platforms of the ENPOWER project.
To achieve this ambitious goal, we propose an innovative data model that we have named SEMAPTIC, which is used to add a semantic specification to the data exchanged by the energy services in order to ensure semantic interoperability.
Our semantic data model relies on well-defined ontologies, and rather than extending existing ones like SAREF, we propose the step further of creating a complete ontology specifically tailored to EnCs within the project.

This comprehensive ontology will accommodate and model all data parameters exchanged between services, reducing the need of using concepts from multiple ontologies, and facilitating the future maintenance of the semantic interoperability of the services.
Finally, we plan providing the SEMAPTIC BUILDER tool, to assist developers in aligning their data with the proposed ontological semantic model turning their services into semantically interoperable with much more limited development effort than existing approaches, as will be described in this deliverable.
Figure 15 illustrates how the different components and tools of our proposal are interconnected and interact to create the semantic map. As can be seen:

- The data structures used by the service to exchange data are uploaded to the SEMAPTIC BUILDER
- The service developer choses the ontologies that will be used to construct the semantic map SEMAPTIC.
- The semantic map is built by the service developer by associating concepts from the ontology selected to the variables exchanged.
- The map can then be integrated into the service code to be attached to the original data when the service is used by a data consumer.
- As considered in the ENPOWER project, data can be exchanged through a dataspace.

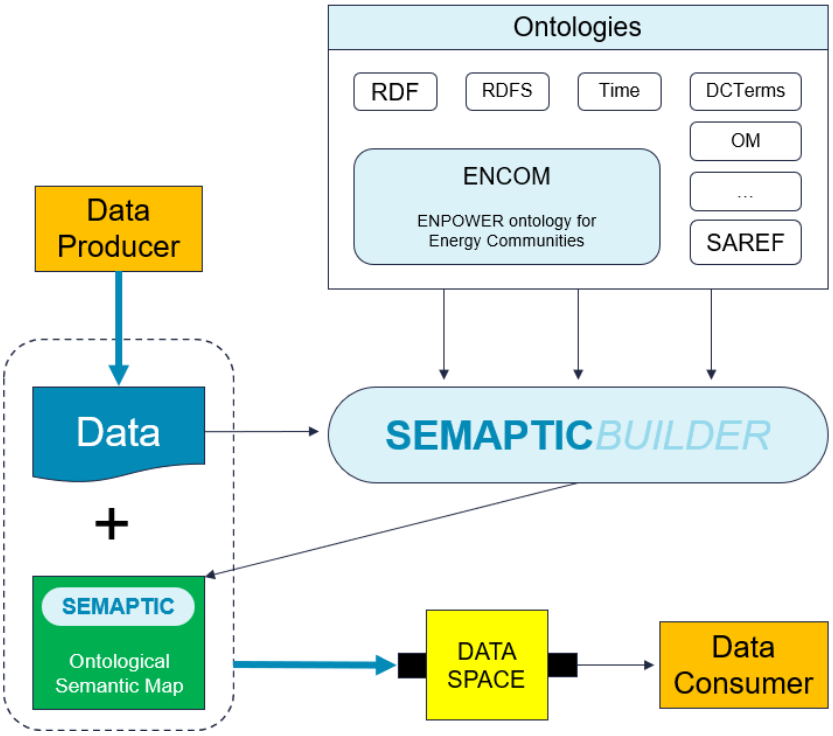

Figure 15: Tools to build the semantic map.

4.1 Interoperability

In the context of software and information and communications technologies, "interoperability" refers to the ability of different systems to work together, exchanging and using each other's information. There are several types of interoperability, including:

- **Organizational interoperability**: this involves alignment between different organizations, processes, policies, and procedures to support seamless collaboration and information sharing. It may require addressing cultural, legal, or organizational barriers.
- **Technical interoperability**: this involves ensuring that different systems of components can communicate with each other and exchange data. It includes compatibility with data formats, protocols, and communication interfaces.
- **Functional interoperability**: this involves ensuring that different systems or components can perform required functions or tasks effectively when integrated. It includes compatibility with APIs, interfaces, and service-oriented architectures.
- **Platform interoperability**: this refers to the ability of software applications or platforms to integrate with each other, allowing users to access and use functionality across different systems.
- **Temporal interoperability**: ensures data exchanged between systems remain accurate and relevant over time. It involves managing data currency, consistency, and validity across different systems.
- **Syntactic interoperability**: focuses on ensuring that data exchanged between systems follows a common syntax or structure. It involves adherence to agreed-upon data formats, such as XML, JSON, or CSV.
- **Semantic interoperability**: this refers to the ability of systems to understand and interpret the meaning of exchanged data accurately. It involves using common data models, ontologies, vocabularies, and standards to ensure shared understanding.

For the purpose of this document, and according to T4.2, we will focus on semantic interoperability.

4.1.1 Semantic Interoperability

The concept of semantic interoperability has been developed over time by many researchers and organizations. The introduction of the Semantic Web concept by Tim Berners-Lee in 1999 [4] has significantly contributed to the development and application of semantic interoperability.
Various conceptual models and representation models have been developed in different countries and sectors to address the principles and needs of semantic interoperability.
It's important to note that the evolution of semantic interoperability is a collective effort of many individuals, organizations, and communities worldwide. It continues to evolve as technology advances and the need for more efficient and meaningful data exchange grows.
Semantic interoperability builds upon syntactic interoperability. Indeed, if systems can't even understand the structure of the data, they can't hope to grasp its meaning. Once the format is understood, shared meaning can be established for accurate data exchange. This is accomplished by adding data about the data (metadata), linking each data element to a controlled, shared vocabulary, which is the ontology selected.
Therefore, the main focus of semantic interoperability is the meaning of the data being exchanged. When two systems are exchanging semantic data, they are also sharing understanding about that data. The consumer of that data can interpret it accurately and then use it correctly according to the exact meaning intended by the data provider.

4.1.2 Who needs semantic interoperability?

Any system that exchanges data with another system, in any area, should use some kind of semantic interoperability, to guarantee that the meaning of the data is the same for both systems. Beyond any specific domain, semantic interoperability offers several advantages:

- It is a requirement to enable machine computable logic, inferencing, knowledge discovery, and data federation between information systems.
- Improves data quality and consistency reducing errors and misinterpretations.
- Increased efficiency and productivity by reducing the manual effort required to translate data between systems.
- Opens doors for new applications and services that rely on seamless data exchange with well-defined semantics.

Formally, to guarantee the semantic interoperability of exchanged data, it's essential to have a set of well-defined and agreed-upon standards. These standards provide a common framework for both sending and receiving systems to comprehend the meaning, structure, and relationships within the data. This can be achieved by:

- Using existing standardized data models or ontologies relevant to the specific domain of the data exchange.
- Developing custom ontologies if existing ones do not meet specific requirements.
- Creating clear and unambiguous data dictionaries that define the meaning of each data element.

By following these formal elements, semantic interoperability becomes a powerful tool for seamless and accurate data exchange across diverse systems.

4.1.3 Disadvantages of semantic interoperability

Although semantic interoperability seems to be a powerful tool for data exchange, there are several drawbacks that may prevent many services developers to include it in their services specifications:

- Developing a common understanding of data, context, and relationships requires shared vocabularies and ontologies. This can be a very complex labour-intensive task, and requires trained personnel, making it a hurdle to overcome.
- The true value of semantic interoperability often lies in improved collaboration and efficiency over time. Short-term benefits might be less obvious, making it a less attractive option upfront for some companies.
- Many existing systems weren't designed with interoperability in mind. Upgrading them can be a costly and challenging endeavour.
- Especially in critical sectors, adopting new complex methodologies could have serious consequences. This risk makes companies hesitant to adopt new technology quickly.
- Although standards exist for semantic interoperability, they might not encompass all use cases or be interpreted consistently across different systems. Preferences for specific standards can vary, and the absence of a universal translation mechanism between them can further hinder data exchange.

4.2 SEMAPTIC - A new approach for Semantic Interoperability

Within the ENPOWER Project, one of the goals was to establish data models that would guarantee semantic interoperability for data exchanged between services. Extending existing ontologies or creating new ones helps ensure an interoperability chain. This approach guarantees that data produced by various services can be mapped to suitable classes and properties, facilitating its transformation into semantic data. Semantic data unlocks its full potential when serialized using formats like RDF/XML, Turtle, JSON-LD, or N-Tripes based on well-defined ontologies.
Apart from the avoidance of data interpretation ambiguities, one of the benefits of the semantical interoperability is the use of a semantic reasoner, or simply a reasoner. A reasoner is a service that performs automated reasoning on semantic data. When combined with ontologies, these automated reasoning engines perform three essential tasks:

- Reasoners can deduce hidden connections within the data, revealing implicit relationships that aren’t explicitly stated in the RDF triples.
- They identify inconsistencies between the data and the underlying ontology, ensuring adherence to established domain knowledge. This prevents data inconsistencies.
- Reasoners enhance existing data by adding inferred information based on the ontology, expanding the data’s value by incorporating additional insights and relationships.

A crucial aspect for this project is to know who may ultimately benefit from the data transformation, resulting from the incorporation of the semantic interoperability models, and to what extent semantic reasoners could extract deeper meaning from the data and add new value to them, or the services may simply use the transformed data to fulfil their designated functionalities.
Indeed, a major barrier to efficient data transformation for semantic interoperability is the complexity of data serialization required. Producers need expertise in ontologies to serialize data effectively, while consumers require de-serialization for service integration, resulting in a complicated process.

Therefore, recognizing that services may not use reasoners or semantic tools on the data in the short term, we have developed a novel approach that guarantees semantic interoperability without complex data transformation. This approach, called SEMAPTIC, relies on the construction of an ontological semantic map which is attached to the original data exchanged by the services. The following use cases help to understand how SEMAPTIC offers an alternative to "traditional" semantic interoperability, with benefits such as reduced processing burden and simpler implementation.

4.2.1 Semantic Interoperability Use Case - Traditional approach

To maintain simplicity in our use case, let's examine a scenario where one data producer and one data consumer are exchanging data (Figure 16).

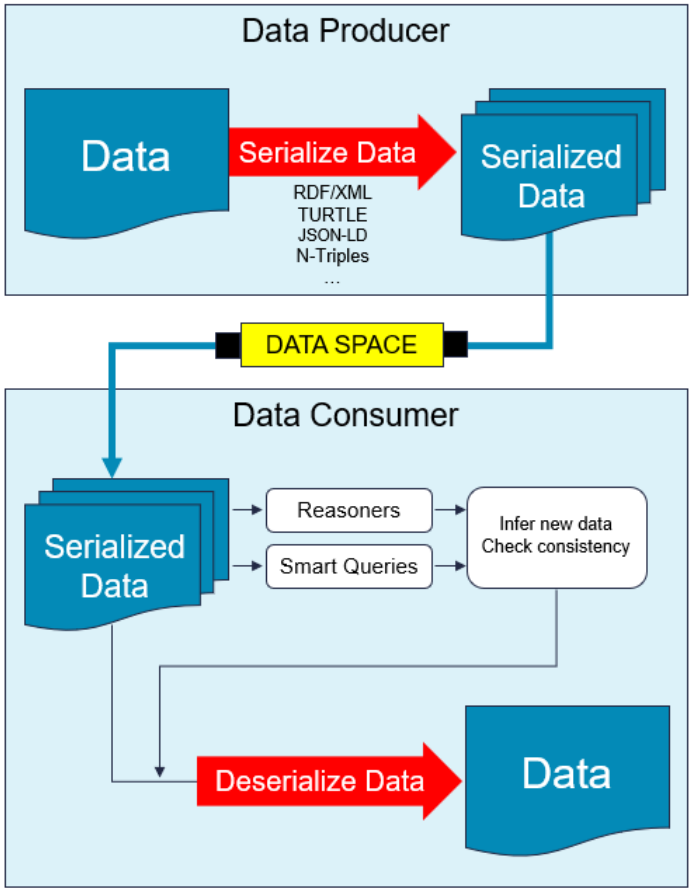

Figure 16: Serialization and deserializations in semantically interoperable data exchange.

This section outlines the steps involved in data exchange with traditional semantic interoperability:

Data Producer:

- Generate Data: Produce the data relevant to the exchange.
- Choose Ontology Format: Select a suitable format for representing the data's meaning (e.g., RDF/XML, Turtle).
- Serialize Data: Convert the data into the chosen format (one example in 5.2.1.1).
- Share Data: Make the serialized data accessible to the consumer (e.g., send or store for retrieval).

Data Consumer:

- Retrieve Data: Obtain the serialized data from the producer.
- Reasoning (Optional): If necessary, use semantic reasoners to infer new information or verify data consistency.
- Deserialize Data: Convert the serialized data back into a usable format.
- Utilize Data: Use the deserialized data for further processing or analysis.

Key Considerations:

- Agreement on Format: Both producer and consumer must agree on the serialization format for seamless exchange.
- Ontology Development: The producer is responsible for identifying or developing ontologies that accurately represent the data's parameters.
- Serialization Complexity: Serialization can be complex, requiring specialized algorithms and potentially increasing data size significantly.
- Deserialization is Necessary: Before utilizing the data, the consumer always needs to convert it back from the serialized format.

Conclusion:
If you don't need to use reasoning capabilities, you can simplify the data exchange process by avoiding serialization and deserialization.

4.2.1.1 Serialization example with Turtle format

Figure 17 illustrates one possible serialized version of the data shown in **Error! Reference source not found.**, using the Turtle format. This process converts the data into a different, machine-readable format. While serialization offers benefits, it's important to note:

- The specific conversion process can vary depending on the data structure and can be intricate.
- Direct conversion of serialized data back to the original format is not always possible, due to the initial transformation. This difficulty necessitates extracting the required information directly from the serialized data.
- Serialization can significantly increase data size, sometimes by a factor of three or more.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix saref: <https://saref.etsi.org/core/#> .
@prefix encom: <http://semanticweb.inesctec.pt/ontologies/encom/> .
@prefix time: <http://www.w3.org/2006/time#> .
@prefix om: http://www.ontology-of-units-of-measure.org/resource/om-2/ .
@prefix ex: <https://www.example.com/> .

<ex:2c70f765> rdf:type saref:Meter .
<ex:2c70f765> rdf:type encom:Identificator .
<ex:2c70f765> encom:hasIdentificator "abc123" .

<ex:2c70f765> saref:hasTimeseries <ex:data/1> .
<ex:data/1> rdf:type saref:Timeseries .
<ex:data/1> encom:hasTimestamp <ex:data/ts/1> .
<ex:data/ts/1> rdf:type time:interval .
<ex:data/ts/1> time:hasBeginning <ex:data/ts/b/1> .
<ex:data/ts/b/1> time:inXSDDateTimeStamp "2024-04-25T15:00:00Z" .
<ex:data/1> saref:hasValue <ex:data/value/1> .
<ex:data/value/1> rdf:type saref:Energy .
<ex:data/value/1> om:hasNumericalValue 0.12 .
<ex:data/value/1> saref:hasUnit om:kWh .

<ex:2c70f765> saref:hasTimeseries <ex:data/2> .
<ex:data/2> rdf:type saref:Timeseries .
<ex:data/2> encom:hasTimestamp <ex:data/ts/2> .
<ex:data/ts/2> rdf:type time:interval .
<ex:data/ts/2> time:hasBeginning <ex:data/ts/b/2> .
<ex:data/ts/b/2> time:inXSDDateTimeStamp "2024-04-25T15:15:00Z" .
<ex:data/2> saref:hasValue <ex:data/value/2> .
<ex:data/value/2> rdf:type saref:Energy .
<ex:data/value/2> om:hasNumericalValue 0.17 .
<ex:data/value/2> saref:hasUnit om:kWh .
```

Figure 17: Serialized data in Turtle format.

4.2.2 Semantic Interoperability Use Case - SEMAPTIC approach

The SEMAPTIC approach, developed for this project, emphasizes both simplicity and semantic preservation. It leverages an ontological semantic map based on established ontologies, providing meaning to all data parameters. As seen in **Error! Reference source not found.**, data p roducers can send data directly without any transformation, apart from adding the semantic map.

This ensures minimal complexity in data exchange while retaining the meaning and context (semantics) through the semantic map. Established vocabularies and ontologies within the map provide a shared understanding, eliminating ambiguity in the exchanged information, as with any other ontological format.

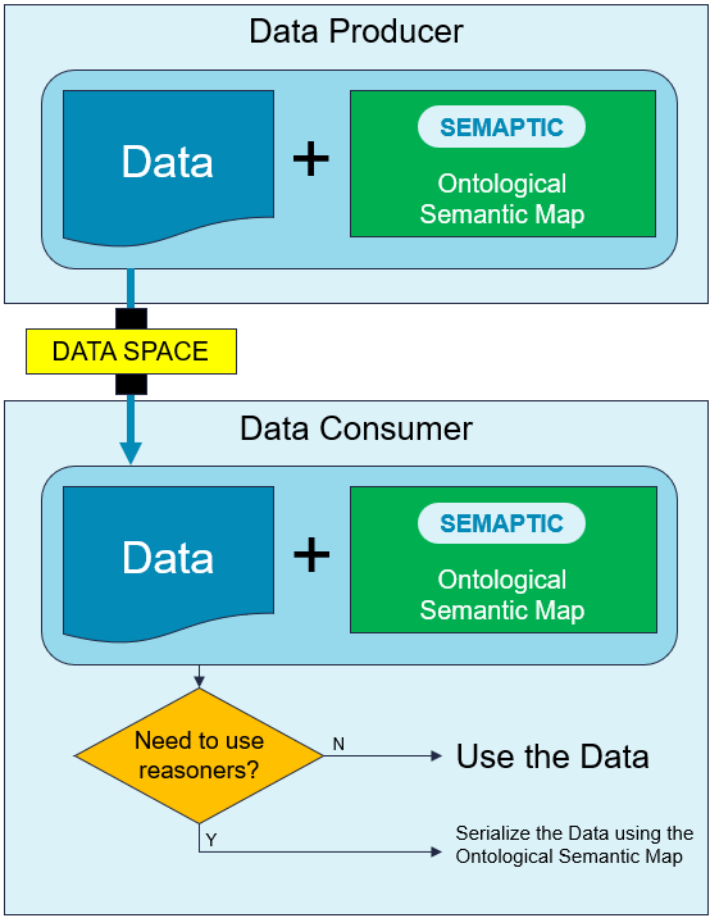

Figure 18: Exchanged data schema with Semaptic.

The steps involved in data exchange with SEMAPTIC framework are:

Data Producer:

- Data Preparation: Data production, relevant to the exchange.
- Ontology Selection/Development: Identification or development of ontologies that accurately represent the data's concepts.
- Semantic Annotation: Creation of a semantic map, using the chosen ontologies to define the data's meaning.
- Data Enrichment: Combination of the original data with the semantic map (e.g., by appending). Note, the data structure for the original data remains unchanged, and only a new data structure with the semantic map is added.

Data Consumer:

- Data Retrieval: Obtaining the enriched data containing both the data and the semantic map.
- Data Interpretation: Directly utilize the data and its semantic meaning for further processing.
- Reasoning (Optional): If reasoning capabilities are needed to infer new information or verify consistency, serialize the data using the provided semantic map to produce a conventional ontological format.

Key Points:

- Simplified Workflow: The producer builds the semantic annotations upfront, eliminating the need for consumer-side serialization unless reasoning is required.
- Direct Data Usage: The consumer can directly use the enriched data for most purposes, avoiding unnecessary deserialization steps.
- Reasoning Enabled: When reasoning is necessary, serialization using the provided map ensures a streamlined process.

Conclusion:
For scenarios where reasoning is not essential, this approach simplifies data exchange by eliminating the complexity of serialization and deserialization on the consumer side.

4.3 The Ontological Semantic Map

4.3.1 Basic structure

The basic structure of the semantic specification will have three main blocs (**Error! Reference source not found.**):

```
"@semantics": {
   "@prefixes": {   },
   "@context": {   },
   "@annotations": {   }
}
```

Figure 19: Semaptic basic structure.

Each component block within the specification map fulfils a distinct function, described in the next chapters.

4.3.2 Ontology References: the “@prefixes” block

The @prefixes block serves as a dictionary, defining key-value pairs where the "key" is a prefix, and the "value" is the corresponding ontology IRI (identifier). This block reveals the ontologies used in the ontological semantic map. Data producers, as the owners of their data, are responsible for selecting appropriate ontologies that effectively model and provide meaning to all their data parameters.

4.3.3 Parameter Mapping: the “@context” block

The @context block builds upon the chosen ontologies to define the specific meaning and context for all data parameters the producer wants to communicate. By leveraging the classes, properties, and individuals available within the selected ontologies, the producer can accurately model their data.

4.3.4 Annotation Section: the "@annotations" block

The @annotations block allows producers to provide additional metadata about the data being described. This metadata can leverage classes from the Dublin Core ontology and RDF Schema to provide details about the service itself or the data it produces.

4.4 Special characters

There are three reserved characters in this specification:

1. **@ (at sign):**
   a. Used before the keywords semantics, prefixes, context, and annotations, to indicate the main blocks of the semantic model.
   b. Used before a parameter key as a pointer to it. See example 4.1.
2. **? (question mark):**
   a. In some cases, auxiliary variables are required in the specification. The "?" character is used to prefix such variables. See example 3.
3. **$ (dollar sign):**
   a. Used before a parameter key, points to its value. See example 1.

4.5 Complete example

To provide a complete example of the specification, let's use a JSON structure that represents an array of energy values produced by a solar panel (Figure 20).

```
{
   "id": "abc123",
   "data": [
      {
         "timestamp": "2024-04-25T15:00:00Z",
         "value": 0.12,
         "unit": "kWh"
      },
      {
         "timestamp": "2024-04-25T15:15:00Z",
         "value": 0.17,
         "unit": "kWh"
      }
   ]
}
```

Figure 20: Example of JSON Structure.

To build the ontological semantic specification map we need to perform two initial steps:

a. Extract all key parameters from the JSON structure
b. Choose the ontologies that best model the data.

From the JSON structure que can easily check that the parameters are:

- "id"

- "data"
- "timestamp"
- "value"
- "unit"

Chosen ontologies:

- "**rdf**": "http://www.w3.org/1999/02/22-rdf-syntax-ns#"

RDF (Resource Description Framework) is a standard model for data interchange on the web. It structures data as triples (subject, predicate, object) to represent information about resources in a graph form.

- "**time**": "http://www.w3.org/2006/time#"

"Time" is an ontology of temporal concepts for describing the temporal properties of resources. It provides a vocabulary for expressing facts about relations among instants and intervals, durations, and date-time information.

- "**encom**": "http://semanticweb.inesctec.pt/ontologies/encom/"

The "Encom" ontology, in development within the ENPOWER project, is intended to provide semantic interoperability to facilitate data exchange between services for EnCs. It will model the complex relationships between the main involved such as DSOs, aggregators, members, energy sources, storage, consumption patterns, etc. This structured approach will serve as a foundation for data integration, knowledge discovery, and decision-making, ultimately promoting sustainable and efficient energy practices within these communities.

4.5.1 The "@prefixes" block example

After choosing the ontologies the "@prefixes" block is already complete since it only shows the prefixes of the IRI of the ontologies used for the specification.

```
"@prefixes": {
   "rdf": "http://www.w3.org/1999/02/22-rdf-syntax-ns#",
   "time": " http://www.w3.org/2006/time#",
   "encom": " http://semanticweb.inesctec.pt/ontologies/encom/"
}
```

4.5.2 The "@context" block example

The "@context" block is used to give meaning to all the parameters. Whether individually or together (when that makes sense), the parameters must be correctly modelled.

4.5.2.1 Example 1

The value of the "id" parameter, in our example, represents the identifier of a PV panel.

```
"id": "abc123"
```

We need to find in some ontology:

- A “class” to assign its type to the key “id”.
- A “property” to assign its value to “id”.

The character “$” is reserved. When used before the key it points to the value of that key. In this example, “$id” has value “abc123”. Putting all this together, a possible semantic specification for "id":"abc123":

```
"id": {
   "rdf:type": "encom:Meter",
   "rdf:type": "encom:Identificator",
   "encom:hasIdentificator": "$id"
}
```

4.5.2.2 Example 2

The "data" parameter, in our example, is the key that represents the identification of an array.

```
"data": [
   {
      "timestamp": "2024-04-25T15:00:00Z",
      "value": 0.12,
      "unit": "kWh"
   },
   …
]
```

To better model this parameter we must look inside the array to find any pattern. We can see that it is a timeseries, so we must find an appropriate ontological class.

```
"data": {
   "rdf:type": "encom:Timeseries"
}
```

4.5.2.3 Example 3

A “timestamp” can be a lot of things and must be modelled carefully. For example, a timestamp can represent an instant, a creation time, the beginning of an interval, the end of an interval, etc.

```
"timestamp": "2024-04-25T15:00:00Z"
```

In this example, "timestamp" represents the beginning of an interval. However, the value of the "timestamp", i.e. "$timestamp", has a specific format that needs to be modelled. To achieve this, we introduce an auxiliary variable that can hold additional properties.
The character "?" identifies these auxiliary variables. For instance, "?beg" serves as an auxiliary variable. This variable is specifically created to accommodate the "time:inXSDtimestamp" property, which defines the data type for "timestamp".
It's important to note that if an auxiliary variable is created, then it must also be modelled.

```
"timestamp": {
   "rdf:type": "time:Interval",
   "time:hasBeginning": "?beg",
   "?beg": {
      "time:inXSDtimestamp": "$timestamp"
   }
}
```

4.5.2.4 Example 4

Since the value and the unit parameters are related, they need to be modelled carefully.

```
"value": 0.12
"unit": "kWh"
```

The semantic specification of this example depends on the use case. There are at least two viable alternatives:

4.5.2.4.1 Example 4.1

In some use cases the unit may change in the timeseries. For example, we may have some values with unit “kWh” and others with unit “Wh”. In this case we model the two parameters separately, but linking the two using the special character “@”:

```
"value": {
   "rdf:type": "encom:ActivePower",
   "encom:hasValue": "$value",
   "encom:hasUnit": "@unit"
},
"unit": {
   "encom:hasValue": "$unit"
}
```

4.5.2.4.2 Example 4.2

When the unit is fixed then we can link the two related parameters avoiding the need to model the “unit” parameter separately.

```
"value": {
   "rdf:type": "encom:ActivePower",
   "encom:hasValue": "$value",
   "encom:hasUnit": "encom:kWh"
}
```

4.5.3 The “@annotations” block example

The "@annotations" block facilitates the inclusion of non-schematic metadata through free-text annotations. These annotations may reference terms from ontologies like Dublin Core (https://www.dublincore.org/specifications/dublin-core/dcmi-terms/) and RDF Schema (https://www.w3.org/TR/rdf12-schema/).

```
"@annotations": {
   "dcterms:creator": "Ricardo Silva",
   "rdfs:comment": "Service from INESC TEC"
}
```

The “@annotations” block provides a mechanism for including additional metadata beyond the original data content. As this block is independent of data transformations, it allows for the incorporation of supplementary information about the exchanged data and its originating service.

4.5.4 Complete ontological specification map

Figure 21 presents the complete specification map, integrating the elements illustrated in the preceding examples.

```
"@semantics": {
   "@prefixes": {
      "rdf": "http://www.w3.org/1999/02/22-rdf-syntax-ns#",
      "time": " http://www.w3.org/2006/time#",
      "encom": " http://semanticweb.inesctec.pt/ontologies/encom/"
   },
   "@context": {
      "id": {
        "rdf:type": "encom:Identificator",
        "encom:hasIdentificator": "@id"
      },
      "data": {
         "rdf:type": "encom:Timeseries"
      },
      "timestamp": {
         "rdf:type": "time:Interval",
         "time:hasBeginning": "?beg",
         "?beg": {"time:inXSDtimestamp": "$timestamp"}
      },
      "value": {
         "rdf:type": "encom:ActivePower",
         "encom:hasValue": "@value",
         "encom:hasUnit": "encom:kWh"
      },
   },
   "@annotations": {
      "dcterms:creator": "Ricardo Silva",
      "rdfs:comment": "Service from INESC TEC"

   }
}
```

Figure 21: Complete ontological semantic specification.

A new JSON structure (Figure 22), combining the original data with the semantic specification map, must be created and transmitted to the data consumer.

```
   {
      {
         … original data …
      },
      "@semantics": {
         "@prefix": { … },
         "@context": { … },
         "@annotations": { … }
      }
   }
```

Figure 22: Data and Semantic Map.

4.5.5 Conclusions and main advantages

The advantages of the proposed approach include:

- Every data parameter is mapped based on well-defined ontologies, ensuring clear and unambiguous meaning.
- The ontological specification map is created only once for each service.
- The map is then appended and combined with the data produced by the service.
- No data transformation is required.
- The approach is scalable, accommodating both small (one day) and large (five years) datasets without altering the specification.
- Immediate data consumption by the receiver is possible due to the absence of data serialization.

4.6 SEMAPTIC Builder

Under development is a tool called SEMAPTIC Builder (Figure 23), to streamline the creation of semantic specification maps. This user-friendly tool simplifies the task of defining semantics for various data elements. SEMAPTIC Builder offers flexibility in ontology integration, allowing users to download existing online ontologies or upload their own custom ones. The tool then analyses the chosen ontology, presenting its classes and properties in a clear and easy-to-understand manner. With this information readily available, users can effortlessly map their data variables to the appropriate ontology elements.

Once the mappings are defined, SEMAPTIC Builder takes care of the rest, automatically generating a comprehensive semantic map based on the user's selections. In essence, SEMAPTIC Builder acts as a powerful assistant, facilitating the creation of semantic specifications by streamlining ontology integration, data variable mapping, and automatic map generation.

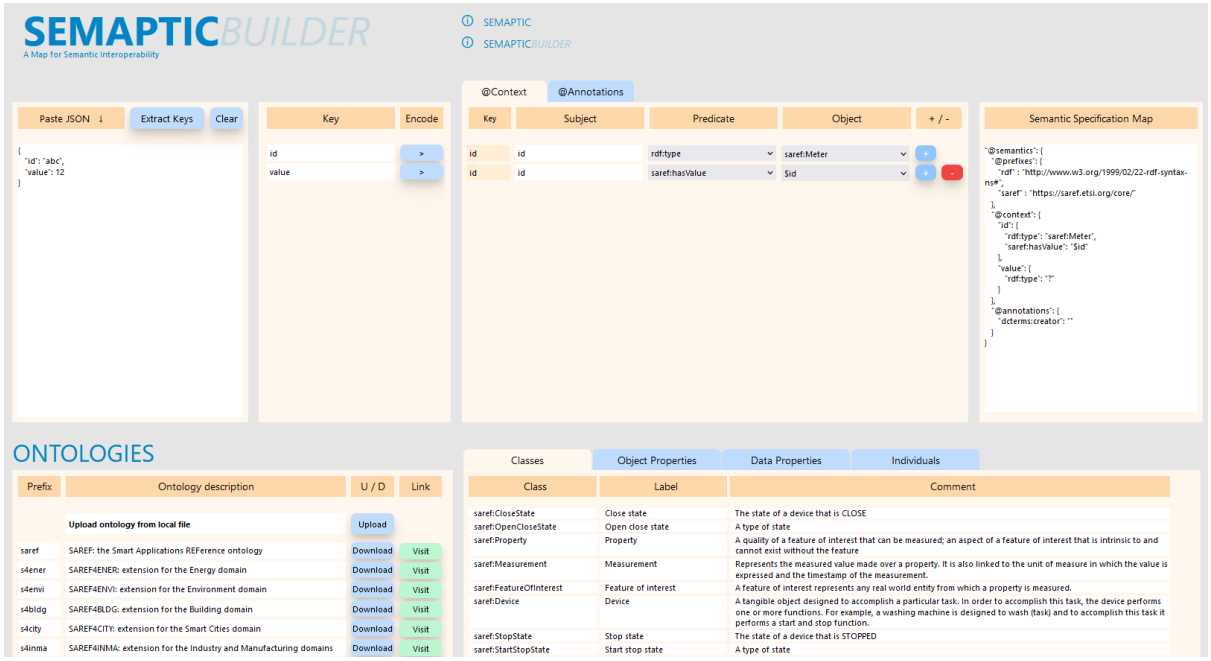

Figure 23: Semaptic Builder screenshot.

4.7 ENCOM – Ontology for Energy Communities

An ontology is described as “a formal, explicit specification of a shared conceptualization” [3]. By adding a layer of semantics, or meaningful relationships, ontologies greatly enhance existing data. This enhancement promotes interoperability, reusability, and shareability of information across various systems.

For well-defined groups such as EnCs, industrial parks, or large companies, creating a domain-specific ontology tailored to their specific needs is highly beneficial. This ontology acts as a centralized “data dictionary” that includes all relevant concepts within the group’s domain. By developing their own ontology, these groups can avoid the complexities of integrating or relying on external ontologies that may be designed for entirely different purposes. For example, an energy distribution network could create an ontology that defines power generation, distribution, and consumption according to their specific requirements.

The ENPOWER project illustrates the use of a domain-specific ontology for EnCs. Therefore, within ENPOWER project we are currently designing a custom ontology, ENCOM (Energy Community Ontology), to make the offered energy services semantically interoperable, facilitating data exchange and knowledge representation across them. ENCOM aims to support a wide range of services and functionalities, from complex resource sizing services used for community structure optimization to basic data communication mechanisms used by internal services and those interacting with external entities such as Distribution System Operators (DSOs), aggregators, and suppliers.

ENCOM is being developed with an incremental approach, allowing for the continuous integration of new concepts, classes, and properties relevant to the specific needs of the EnC and its services. This approach includes remodelling existing concepts found in other ontologies, avoiding the need to build from scratch, while ensuring their current and future usability. This eliminates the dependency on external entities for maintaining these concepts. Keeping imported parts up to date requires monitoring changes in the source ontologies, which can be an ongoing burden, especially if multiple external ontologies are involved.

By providing a comprehensive ontology that is readily accessible to non-ontologists and individuals with limited semantic interoperability expertise, ENCOM aims to empower service developers. With all relevant information available in a centralized location, developers can effectively assign meaning to their variables and construct semantic maps for their data. This simplifies knowledge representation and data exchange within the EnC.

5. Energy Data Space compliant tools for sovereign data sharing and federated planning (Task 4.3)

5.1 FIWARE Context Broker and OneNet Connector

In the context of the ENPOWER project, the FIWARE Context Broker is part of the interoperability middleware that is in charge of ensuring the proper communication between the local business platforms used by project pilots and the added-value services provided within WP5. Where applicable, the Context Broker should be able to send back data/information from the added-value services to the local business platforms.

The Context Broker can also provide semantic interoperability. Well-defined data format (JSON-LD) and assignation of unique IDs (URLs or URNs) for both data entities and the relationships between entities will ensure that semantic meaning can be programmatically retrieved from the data itself. JSON-LD, which is an extension of JSON, is a standard way of avoiding ambiguity when expressing linked data in JSON so that the data is structured in a format that can be analysed by machines. This method ensures that all data attributes can be easily compared when coming from a multitude of separate data sources. The FIWARE NGSIv2 information model has been evolved to better support linked data (entity-relationships), property graphs, and semantics by exploiting the capabilities offered by JSON-LD. This work has been conducted under the ETSI ISG CIM initiative and has been branded as NGSI-LD. DEDALUS pilots and partners in charge of the value-added services (WP4) will share the data models they use to be included in the Context Broker. As anticipated in section **Error! Reference source not found.**, FIWARE already provide a w ide range of data models as Smart Data Models but new data models can also be added by using a set of guidelines [32].

Finally, to ensure technical interoperability with Energy Data Spaces, the OneNet connector will be adapted suitably and deployed for exchanging/sharing data offered by Data Provider. This connector allows an easy integration and cooperation among the Data Providers and Data Consumers maintaining the data ownership and preserving access to the data sources. The Context Broker is the core component of the OneNet connector in FIWARE-based implementations of the IDS Architecture. The Context Broker offers the FIWARE NGSI APIs and associate information model (entity, attribute, metadata) as the main interface for sharing data in the Data Space. Data Providers use the APIs to publish or to expose the offered data to be retrieved by the subscribed Data Consumers.

Adhering to **IDS specifications**, the Context Broker is based on the **FIWARE Orion Context Broker** and **NGSI-API**. The Connector features a configuration tool, interoperable APIs for connecting with existing platforms, applications, and services, and provides Data Harmonization services. This setup ensures smooth and efficient data exchange across various platforms and stakeholders.

[32] https://smartdatamodels.org/index.php/guidelines/

5.2 OneNet Introduction

One of the primary objectives of OneNet is to establish an open and flexible architecture that transforms the European electricity system, which is traditionally managed at the national level, into a more integrated, pan-European smart grid. This transformation aims to enhance efficiency, reliability, and sustainability across the continent. The Reference Architecture, depicted in **Figure 1Figure 24**, is designed to support this vision and is structured into three logical layers:

- **OneNet Participants Layer**: This layer ensures that data from various sources are made available and accessible to energy stakeholders in a secure and trusted manner. It emphasizes data ownership and privacy, ensuring that data providers maintain control over their information. Participants in this layer include data providers, data consumers, and service providers who interact with the system to share and utilize data effectively. This layer is critical for fostering a collaborative environment where different stakeholders can contribute to and benefit from the shared data ecosystem.
- **OneNet Network of Platforms Layer**: This layer facilitates the integration and cooperation of various platforms to enable cross-platform market and network operation services. It includes demonstration platforms such as DSO platforms, Market platforms, and Data Exchange platforms. These platforms connect to the OneNet Middleware through the OneNet Connector, allowing for seamless data exchange and interoperability. The integration within this layer ensures that diverse systems can work together harmoniously, providing a foundation for advanced energy management and market operations. By supporting interoperability, the Network of Platforms layer plays a pivotal role in creating a cohesive and efficient energy market across Europe.
- **OneNet Framework Layer**: This layer serves as a scalable and pluggable solution that promotes platform cooperation and integration. It creates a unified ecosystem where any energy stakeholder can participate, leveraging standardized models and interoperability standards such as CIM, CGMES, SAREF, and IEC 61970. The framework includes components for data and service orchestration, ensuring that data flows smoothly between platforms and stakeholders. It also incorporates cybersecurity and data governance measures to protect data integrity and confidentiality. This layer's modular design allows for flexibility in accommodating new technologies and evolving market needs, making it a robust foundation for the future European energy system.

The OneNet Reference Architecture aims to address key challenges in the energy sector, such as the integration of renewable energy sources, the need for real-time data exchange, and the importance of maintaining data privacy and security. By leveraging advanced technologies and promoting collaboration among stakeholders, OneNet seeks to create a smarter, more resilient, and more sustainable energy system for Europe.

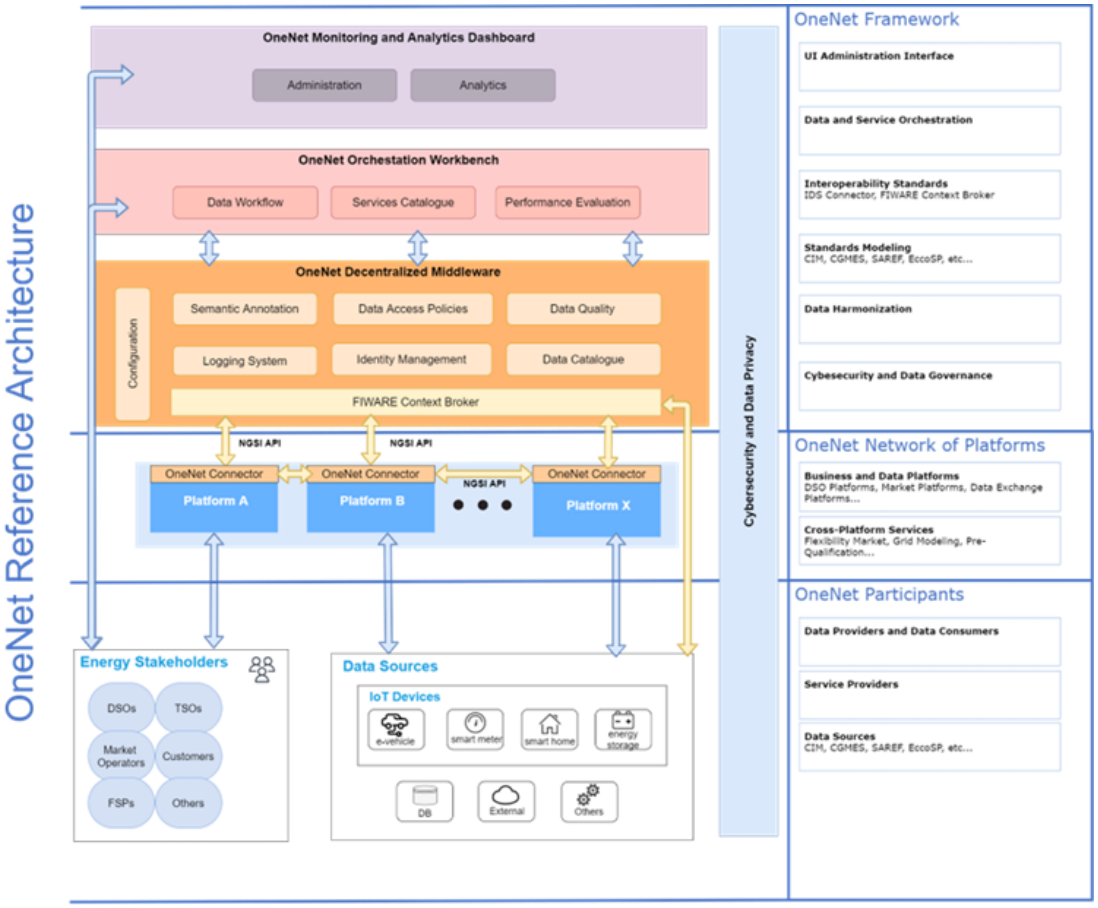

Figure 24 THE ONENET REFERENCE ARCHITECTURE.

5.2.1 OneNet Participants Layer

The OneNet Participants layer is a crucial component, consisting of *Data Sources* and *Energy Stakeholders*.

- ***Data Sources* component**: This includes all assets that produce or process data. Data can originate from various energy sectors such as electric mobility, energy storage, and residential energy monitoring. It is generated by devices like sensors, actuators, gateways, and edge computing nodes. The diverse range of data sources ensures a comprehensive and integrated approach to energy data management.
- ***Energy Stakeholders* component**: This encompasses stakeholders identified within the OneNet market, categorized based on their business roles, such as DSO, TSO, Market Operator, Customer, etc. For the architecture focused on data exchange, stakeholders are grouped into Service Providers, Data Consumers, and Data Providers. This categorization facilitates targeted data exchange and service provision.

This layer is intricately connected to the OneNet Network of Platforms layer, serving as the data source for business and data platforms as well as integrable services or applications. It connects directly to the OneNet Framework through the OneNet Middleware, utilizing interoperability mechanisms based on the FIWARE Architecture and the **FIWARE Orion Context Broker**. Additionally, energy stakeholders can access the Orchestration Workbench for evaluating and testing applications, tools, and services. They can also leverage the analytic features available through the OneNet Monitoring Dashboard provided by the OneNet Framework.

5.2.2 OneNet Network of Platforms Layer

The **OneNet Network of Platforms** layer is designed to integrate external platforms, including DSO platforms, market platforms, and other data exchange platforms, into the OneNet system using a fully decentralized approach. This infrastructure enables direct interaction between systems (OneNet Participants) without third-party mediation.
The central component in this layer is the **OneNet Connector**, which executes the complete data exchange process. The OneNet Connector is an instance of the OneNet Decentralized Middleware placed within each platform, facilitating seamless integration and cooperation among platforms. Adhering to **IDS specifications**, the Context Broker is based on the **FIWARE Orion Context Broker** and **NGSI-API**. The Connector features a configuration tool, interoperable APIs for connecting with existing platforms, applications, and services, and provides Data Harmonization services. This setup ensures smooth and efficient data exchange across various platforms and stakeholders.

5.2.3 OneNet Framework Layer

The **OneNet Framework** layer is the core of the OneNet Architecture, comprising the OneNet Decentralized Middleware, OneNet Orchestration Workbench, and OneNet Monitoring and Analytics Dashboard.

- ***OneNet Middleware***: This component ensures secure and reliable end-to-end data exchange between all assets and components integrated into the OneNet Network. It provides central features such as identity management, source discovery, semantic annotation, vocabularies, and ontologies. These functionalities are essential for maintaining data integrity, security, and interoperability within the network.
- ***OneNet Orchestration Workbench****:* This tool supports data orchestration for evaluating the performance and scalability of AI, IoT, and Big Data cross-platform services for market and grid operations. The workbench integrates data from the OneNet Middleware and implements data pipeline orchestration, facilitating complex data processing tasks and performance assessments.
- ***OneNet Monitoring and Analytics Dashboard***: Serving as an administrative and configuration tool, it seamlessly integrates with the OneNet Orchestration Workbench and OneNet Middleware. This tool provides a comprehensive data-analytics dashboard, a monitoring and alerting dashboard for data processes and platform integrations, and a user-friendly interface for selecting data sources and services from catalogues. These features enable effective monitoring, management, and optimization of data and services within the OneNet ecosystem.

By incorporating these detailed components, the OneNet Framework layer ensures a robust and scalable architecture that supports comprehensive data management, secure exchange, and efficient service integration, paving the way for a smarter and more interconnected European energy system.

5.3The OneNet Connector – Technical Characteristics

The OneNet connector is the core component designed to enable decentralized data exchanges within the OneNet ecosystem. As a deployment instance of the OneNet Decentralized Middleware, the OneNet connector, once integrated within the platforms of each OneNet participant, facilitates the creation of a trusted pan-European data space defined as the OneNet Network of Platforms.

In the context of the ENPOWER project, data providers refer to the entities or stakeholders that generate, collect, and supply data related to energy usage, production, and other energy-related metrics. These could include:

- **Energy Consumers and Prosumers:** Households, businesses, and other entities that not only consume energy but may also produce energy through renewable sources like solar panels or wind turbines. These prosumers provide data on energy consumption, production, and potential flexibility.
- **Energy Communities:** Localized groups of consumers and prosumers who share energy resources and data. Energy communities contribute data related to collective energy usage, sharing, and optimization strategies.
- **DSOs (Distribution System Operators):** Entities responsible for managing the distribution of electricity within specific regions. They provide data on the operation of local energy grids, including supply, demand, and grid flexibility.
- **TSOs (Transmission System Operators):** These organizations manage the high-voltage transmission of electricity across large geographic areas and provide data on grid stability, energy flows, and network conditions.
- **Aggregators:** These are entities that pool the energy consumption or production capacities of multiple users to act as a single entity in energy markets. Aggregators contribute data regarding energy loads, demand-response capabilities, and market participation.
- **Market Platforms:** Platforms involved in energy trading, where data related to energy prices, transactions, and market behaviors are provided.
- **Smart Devices and IoT Sensors:** Data providers also include the technological systems and devices that monitor energy usage, grid health, and environmental factors (e.g., smart meters, sensors, and IoT devices).

These data providers play a crucial role by supplying the data that will be used within the Energy Data Space to enable data-driven services, optimize energy usage, and enhance the overall operation of the energy market through secure and standardized data exchanges.

5.3.1 OneNet Connector Architecture

The overall architecture of the OneNet connector is depicted in Figure 25, illustrating its various components:

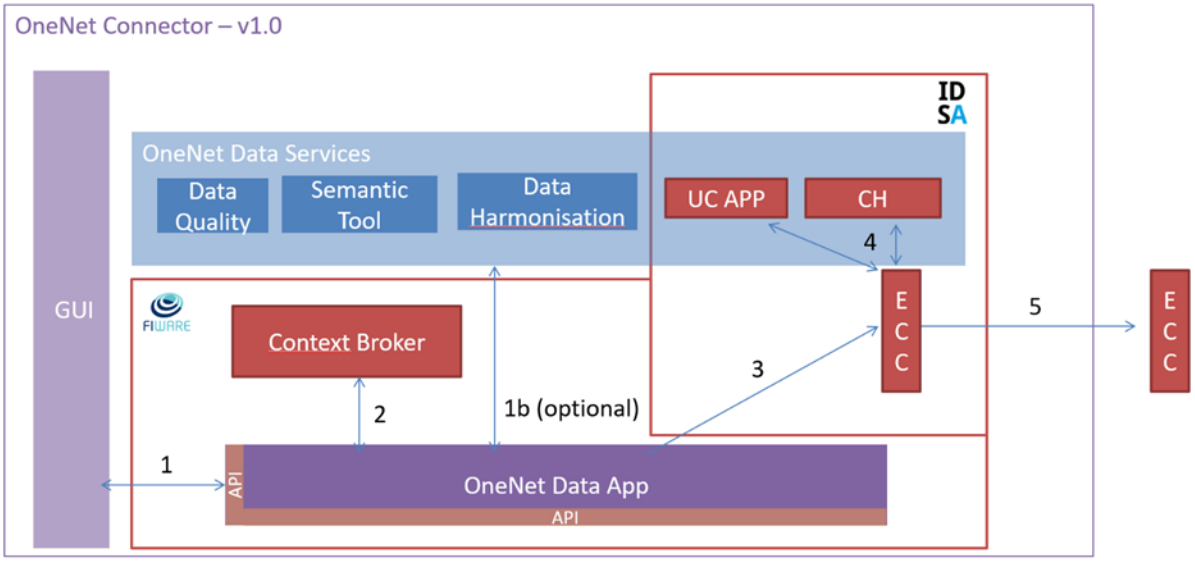

Figure 25: OneNet Connector Architecture - v1.

- **GUI (Graphical User Interface)**: This interface allows users to configure the OneNet Connector and utilize its main features. Each OneNet participant can access their own connector's GUI upon installation. The initial version of the GUI includes:
  - Registration as a data provider/consumer
  - Creation of new data offerings
  - Creation of data entities
  - Registration/consumption of data offerings and data entities
- **OneNet Data App**: This component provides integration APIs for both the GUI and external platforms. It utilizes the NGSI standard for API implementation and data exchange. In its first version, the OneNet Data App offers three main services:
  - Entity creation
  - Registration/Subscription
  - Data retrieving
- **Context Broker**: A FIWARE component that manages the overall data exchange in the standard NGSI-LD format.
- **IDS-Based Components**: These components and tools, extended from the open-source TRUE Connector, enable IDS processes for data exchange:
  - **Execution Core Container (ECC)**: Implements IDS-based data exchange processes, including authentication, metadata exchange, access control, logging, etc.
  - **Clearing House (CH)**: Logs all data exchanges (not available in the first version but being integrated).
  - **Usage Control App (UC APP)**: Verifies access and use of data (not available in the first version but being integrated).
- **OneNet Data Services**: Additional data services provided by the OneNet Connector (to be evaluated if to be included in future versions).

5.3.2 OneNet Connector – v1.0

5.3.2.1 OneNet GUI

The GUI enables OneNet users to:

- define configuration settings for the user and their OneNet connector,
- streamline service management (cross-platform service provision and subscription) and data exchange (provision and consumption), and
- utilize the OneNet connector's APIs.

A comprehensive presentation of the GUI in the form of a manual is presented in Annex A of this URI OneNet_D6.8_V1.0.pdf (onenet-project.eu) (pages 59 to 86). The most recent version of the GUI users' manual will be available online through a dedicated menu item on the GUI.

5.3.2.2 OneNet Connector Data App

The OneNet Data App serves as the entry point to the OneNet Connector, providing access to its main features, including data exchange, through standard interfaces based on the REST API.
Technically, the OneNet Data App is developed using a Java Architectural Stack, based on the Spring Boot framework, and a NoSQL database, MongoDB. All services offered by the OneNet Data App can be accessed via the connector's GUI or directly from external platforms needing integration with the OneNet Connector.
The first version of the OneNet Connector Data App supports three main interactions, provided as standard REST API and accessible through the GUI or API interfaces directly. These interactions are:

- **Create Entity**: Allows a data provider to create new entities within its environment, making them available to other OneNet Participants.
- **Registration**: Enables a data consumer to register for specific data entities to receive data automatically or manually after the provider's acceptance. A data consumer can register with multiple providers and entities.
- **Get Entity**: After registration, allows the data consumer to retrieve specific data entities from the provider.

For each interaction, a high-level sequence diagram is provided below in Figures 25, 26 and 27 to illustrate the process flow.

Create Entity

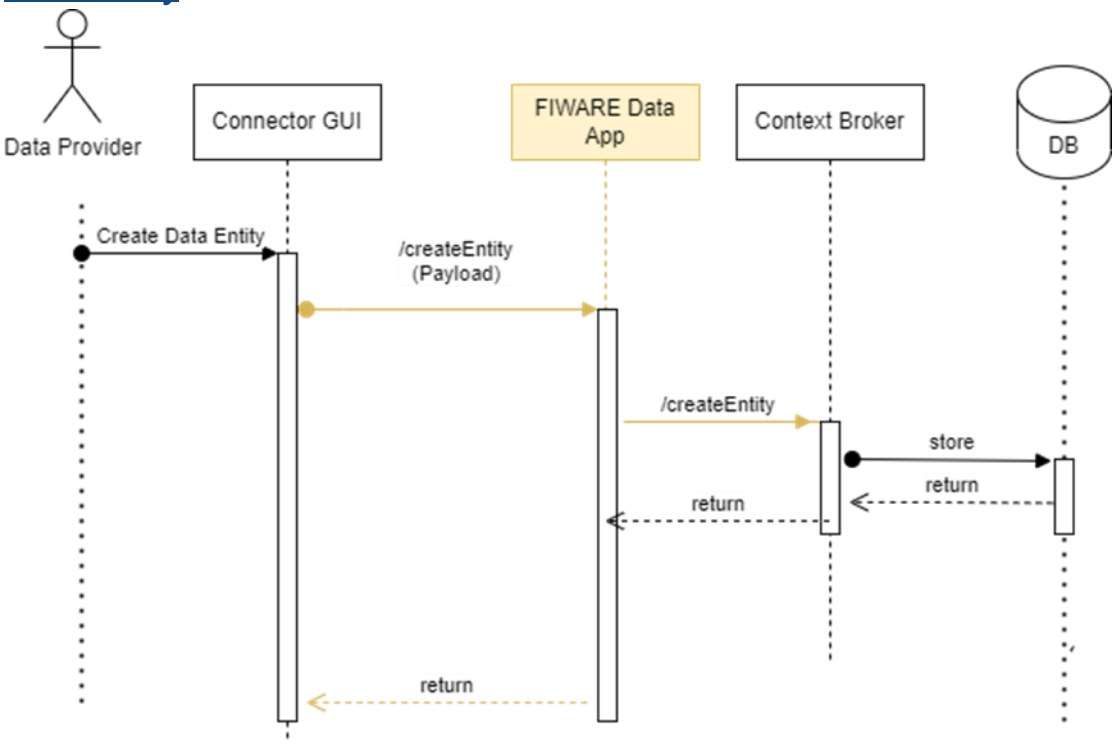

Figure 26: CREATE ENTITY - SEQUENCE DIAGRAM.

Registration

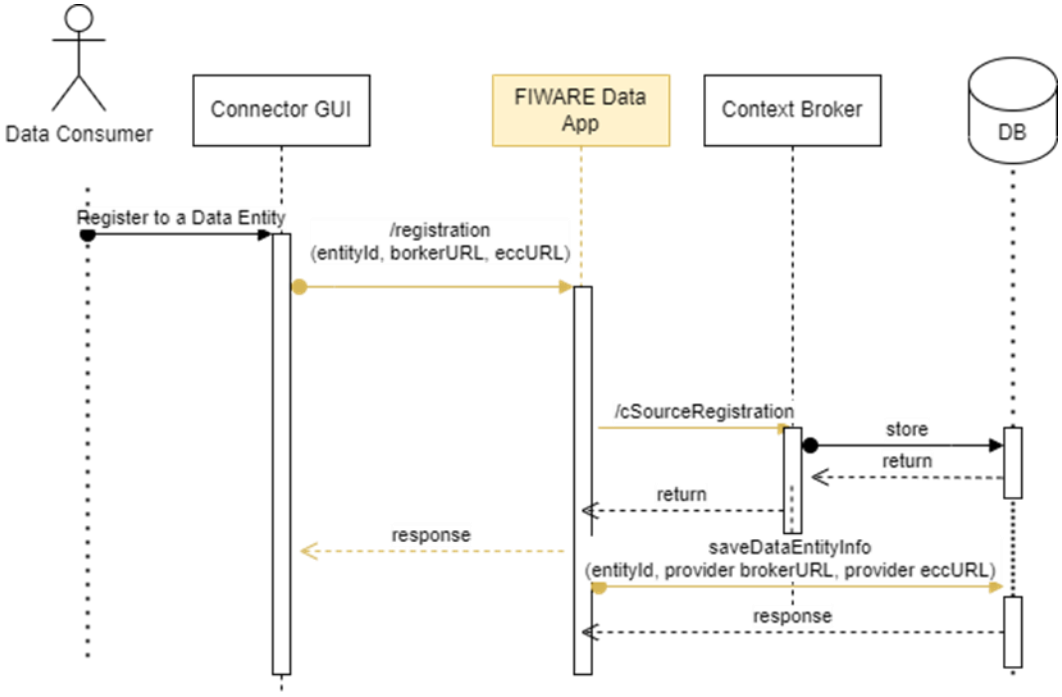

Figure 27: REGISTRATION - SEQUENCE DIAGRAM.

Get Entity

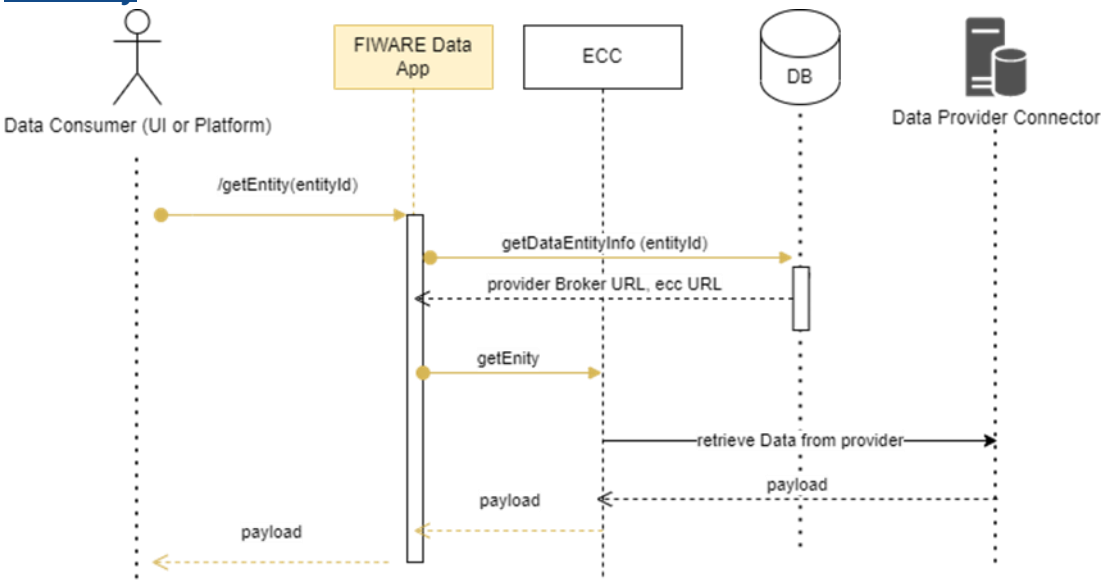

Figure 28: Get Entity – Sequence Diagram.

Create Entity (Figure 27)

This sequence diagram illustrates the steps involved in creating a new data entity within the OneNet ecosystem. The process involves the following components and steps:

- **Connector GUI**: The user initiates the creation of a new data entity through the graphical user interface (GUI) of the OneNet Connector.
- **FIWARE Data App**: The GUI sends a request to create the entity, including the necessary payload, to the FIWARE Data App.
- **Context Broker**: The FIWARE Data App forwards this request to the Context Broker, which handles the data exchange in NGSI-LD format.
- **Database (DB)**: The Context Broker stores the new entity in the database.
- **Return**: The successful creation of the entity is confirmed by returning the appropriate response through the Context Broker and the FIWARE Data App back to the GUI.

Registration (Figure 28)

This diagram explains the process of registering a data consumer to a specific data entity:

- **Connector GUI**: The user initiates the registration process by providing the entity ID, broker URL, and ECC (Execution Core Container) URL through the GUI.
- **FIWARE Data App**: This registration request is sent to the FIWARE Data App.
- **Context Broker**: The FIWARE Data App forwards the request to the Context Broker, which handles the source registration. The registration request related info is registered and checked against the database knowledge and the proper response is returned to the GUI.
- **ECC**: The Context Broker interacts with the ECC to manage the registration information.
- **Database (DB)**: The registration details are stored in the database.
- **Response**: A confirmation of the registration is sent back through the same path, from the database to the ECC, Context Broker, FIWARE Data App, and finally to the GUI.

Get Entity (Figure 29)

This diagram details the steps for retrieving a specific data entity by a registered data consumer:

- **FIWARE Data App**: The data consumer initiates the request to get the entity by providing the entity ID.

- **ECC**: The FIWARE Data App interacts with the ECC to get the entity information, which includes the broker URL and ECC URL.
- **Database (DB)**: The ECC retrieves the data entity information from the database.
- **Provider Broker URL, ECC URL**: Using the retrieved information, the FIWARE Data App contacts the provider's broker URL and ECC URL to get the actual data.
- **Payload**: The data payload is then retrieved and sent back through the ECC, FIWARE Data App, and finally to the data consumer.

These diagrams collectively illustrate the core interactions within the OneNet Connector for managing data entities, registration, and data retrieval, ensuring a comprehensive understanding of the data flow and integration within the OneNet ecosystem.

5.3.2.3 FIWARE and IDS-Based Components

The core implementation for handling data exchange in the OneNet Connector leverages the integration of FIWARE components (specifically, the FIWARE Orion Context Broker using NGSI-API) and the IDS Reference Model (IDS-based connector). This integration ensures that data exchange mechanisms are both efficient and standardized, promoting interoperability and security across the OneNet ecosystem.

The **FIWARE Orion Context Broker** is a critical component of the FIWARE framework, implementing the Next Generation Service Interface (NGSI) standard. This standard facilitates data exchange using standardized data models and interfaces. Within the FIWARE framework, the Orion Context Broker plays a pivotal role in simplifying the development and provisioning of smart, innovative applications that require robust context information and comprehensive data stream management, processing, and exploitation.

In connection with the OneNet Connector, the FIWARE Orion Context Broker-LD, which supports both NGSI-LD and NGSI-v2 APIs, manages all data exchanges among OneNet participants in a standardized manner. The broker is seamlessly integrated with the OneNet Data App and IDS-based components using standardized NGSI APIs, thus enriching its compatibility with the data models to be developed.

The OneNet Architecture adheres to the IDS reference model for creating a pan-European data space. This model leverages existing standards and technologies, as well as established governance models, to facilitate secure, standardized data exchange and linkage within a trusted business ecosystem. A cornerstone of this model is the IDS Connector, which the OneNet Connector extends via the IDS-based TRUE (TRUsted Engineering) Connector (Figure 30). This extension enables the implementation of a FIWARE-based Data Space, allowing existing FIWARE Ecosystems to integrate in a plug-and-play mode and supporting the creation and sustainability of IDS ecosystems.

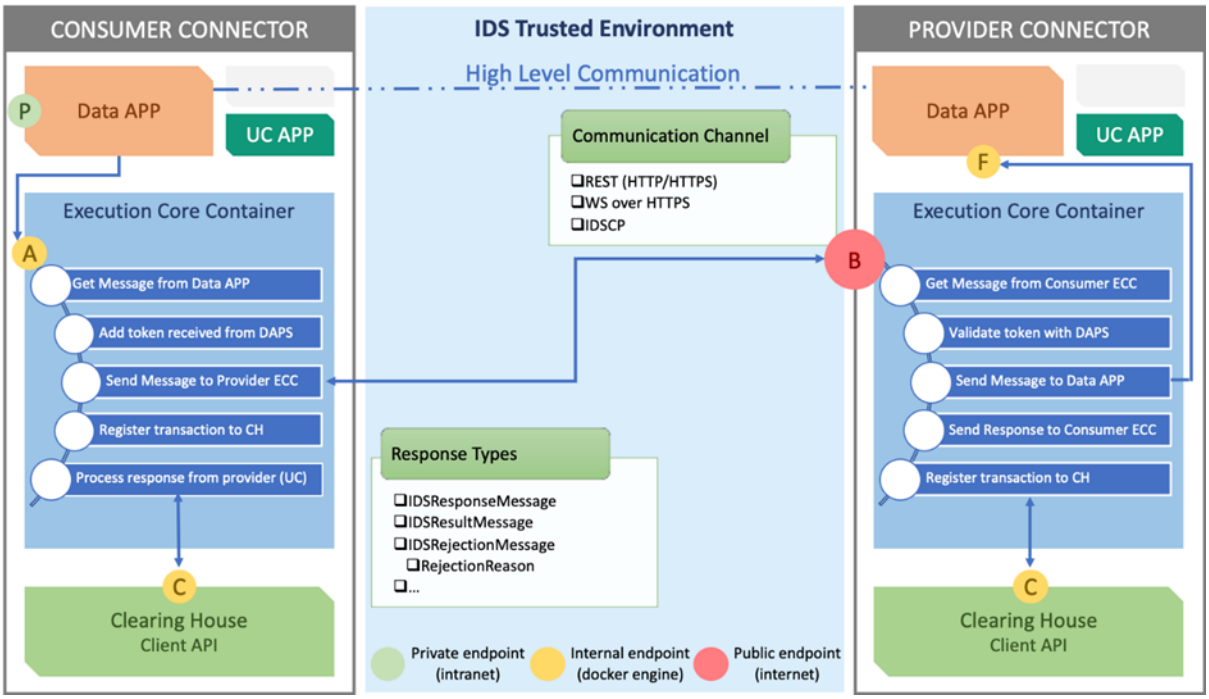

Figure 29: IDS based TRUE Connector

The core of the IDS implementation within the OneNet Connector is the **Execution Core Container**. This container complies with the latest IDS specifications (IDS Info Model) and can be easily customized to fit a wide range of scenarios. It is already integrated with the OneNet Data App and the FIWARE Orion Context Broker-LD is expected to be integrated with other IDS core services in the future, such as Identity Providers, Clearing House, and Usage Control App.

5.3.2.4 REST-APIs

All features available in the OneNet Data App are provided through standard REST APIs. These APIs are accessible both through the GUI and directly by external platforms, facilitating integration. The table below provides a brief description of the currently exposed APIs. Detailed and standardized OpenAPI documentation (https://swagger.io/specification/) will accompany the software release, ensuring comprehensive guidance for users and developers.

Table 2 OneNet REST-APIs.

Name	URL	Method	Parameters	Responses
Create Entity	/createentity	POST	In Body (JSON format) payload: String	Success (200), Error (500) Error Message - String
Registration	/registration	POST	In Body (JSON format) entityId: String, eccUrl: String brokerUrl: String	Success (200) Success Message - String, Error (500) Error Message - String
Get Entity	/getentity	GET	In Request: entityId: String	Success (200) entity,

				Error (500) Error Message - String

5.3.3 Communication Channels

The OneNet platform, adhering to the IDSA Reference Architecture, follows the principle of "Trustworthy Communication & Security by Design". This principle ensures that specific software components like Connectors, App Stores, and Brokers can verify whether the Connector of the connecting party is operating a trusted (i.e., certified) software stack. Communication between external Connectors is both encrypted and integrity-protected to maintain security. Each Data Owner and Data Provider must guarantee that their data is handled by the Data Consumer's Connector in accordance with specified usage policies; otherwise, the data will not be transmitted. To mitigate the risks associated with compromised applications, appropriate technical measures, such as isolating Data Apps from each other and from the Connector, must be applied. Additionally, Data Providers and Data Consumers have the flexibility to determine the security level required for their respective Connectors by deploying Connectors that support their chosen security profile.

The Execution Core Container is responsible for supporting communication and provides components for interfacing with Data Services, such as Data Routers or Data Buses connected to a Connector.

- **Data Router**: This component manages the communication with Data Services, ensuring they are invoked according to predefined configuration parameters. It handles the transmission and reception of data to and from the Data Bus. Participants have the option to replace the Data Router component with alternative implementations from various vendors. Differences in configuration can be managed by specialized Execution Configurator plug-ins. In scenarios where a Connector is part of a limited or embedded platform with a single Data Service or a fixed connection configuration (e.g., on a sensor device), the Data Router can be replaced by hard-coded software, or the Data Service can be exposed directly. The Data Router also invokes relevant components for enforcing Usage Policies, such as a Policy Enforcement Point (PEP), in compliance with the Connector or as specified in the Usage Policy.
- **Data Bus**: This component facilitates the exchange of data with Data Services and other Connectors' Data Bus components. It may also store data within a Connector. Typically, the Data Bus is responsible for exchanging data between Connectors. Similar to the Data Router, the Data Bus can be replaced by alternative implementations to meet the operator's requirements. The selection of an appropriate Data Bus may depend on various factors, such as cost, level of support, throughput rate, quality of documentation, or availability of accessories.

To ensure the confidentiality and authenticity of data transfers, communication between Connectors must be protected. When using the IDS Connector, two layers of security are implemented:

- **Point-to-point encryption:** This involves using an encrypted tunnel between Connectors.
- **End-to-end authorization:** This ensures authenticity and authorization based on actual communication endpoints, i.e., Data Apps.

Data transferred from one External Connector to another can be sent over the Internet or via a virtual private network (VPN), the specifics of which are beyond the scope of the IDS Security Architecture. The Security Architecture defines the IDS Communication Protocol (IDSCP), which must be supported by Trusted Connectors and can also be supported by any other Connector. The purpose of the IDSCP is to establish confidential, authenticated communication, facilitate data exchange between the Data Provider and the Data Consumer, and enable mutual remote attestation (if supported by the Connectors involved).

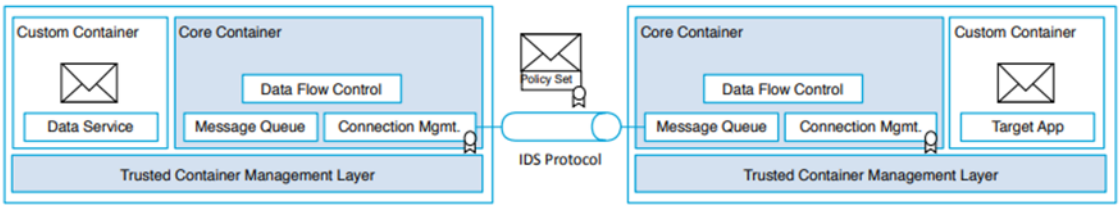

Figure 30: SECURE COMMUNICATION.

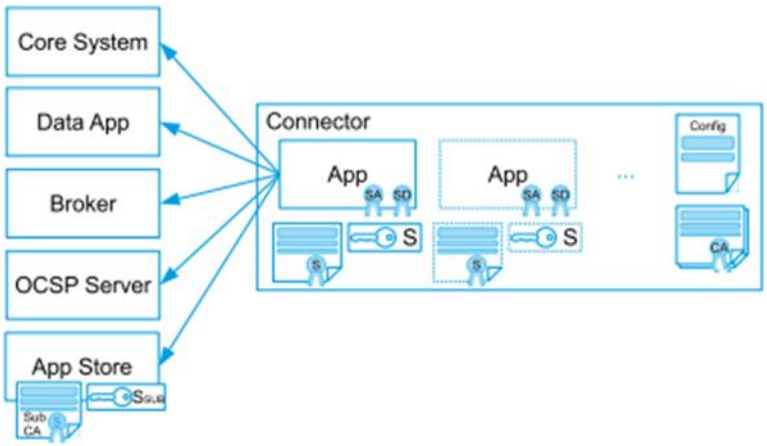

Figure 31: CONNECTOR ROLES AND MANIFESTATIONS.

Figures 31 and 32 illustrate the roles and manifestations of Connectors within the IDS Trusted Environment, demonstrating the high-level communication and interaction between consumer and provider Connectors, as well as the various components involved in ensuring secure and reliable data exchanges.

Future Steps

In the upcoming phases ED will focus on several key areas to enhance the architecture and functionality of the OneNet Connector, ensuring it aligns with the project's goals and integrates smoothly with pilot implementations.

- **Alignment with the ENPOWER Architecture:** ED will further refine the OneNet technological solution to align with the ENPOWER conceptual framework. This involves enhancing the design and structure of the OneNet Connector to ensure it meets the specific requirements of the project as part of a detailed software architecture
- **Knowledge Integration:** Leveraging the extensive knowledge and experience from the OneNet Connector, we will set up and further develop this core component to cater to the unique needs of the project. This includes adapting the connector to handle the specific data exchange and interoperability requirements outlined in the project scope.
- **Agile Development and Customization:** ED will undertake the necessary development work to customize the OneNet Connector, ensuring it adheres to the privacy, security, and

functionality specifications of ENPOWER. This will involve iterative development cycles, incorporating feedback and adjustments to optimize performance and reliability.

- **Pilot Integration:** A crucial next step will be organizing the integration of the OneNet Connector with the project's pilot sites. This will include setting up the connector, testing its interoperability with pilot platforms, and ensuring seamless data exchange between all involved stakeholders.
- **Ongoing Support:** Throughout the course of the project, we will provide continuous support and enhancements to the OneNet Connector, if any emerging challenges need to be addressed, to ensure that the system evolves in line with the project's progress and requirements.

6. Open interoperable tokenised marketplace for reciprocal compensation of energy and data assets (Task 4.4)

6.1 Overview

6.1.1 General Aspects

As stated in the Grant Agreement, task 4.4 focuses on delivering an interoperable marketplace using DLT/Blockchain technologies. In this marketplace, different types of energy and non-energy resources, both tangible and intangible, will be valued, shared, and exchanged. Tokens, which are a form of cryptocurrency, will be used as a reward mechanism for the services offered and will also enable the reciprocal exchange of heterogeneous tangible and intangible resources.

6.1.1.1 Summary of Objectives

The objective of this project is to develop an interoperable market platform designed to facilitate the efficient sharing, valuation, and exchange of both energy and non-energy resources. This platform will employ standardization and tokenization to digitally represent resources, utilizing ERC-20 for fungible assets and ERC-721 for non-fungible assets, ensuring compatibility across various implementations.
Tokens will serve as a reward and compensation mechanism, incentivizing users for their services and enabling the reciprocal exchange of resources. The marketplace will utilize blockchain technology with a Proof of Authority (PoA) consensus mechanism to ensure secure and trustworthy transactions. Smart contracts will automate policies related to resource access, use, and valuation, ensuring transparency and traceability.
The user interface will be intuitive and user-friendly, featuring management dashboards for visualization and asset management, thus making it accessible for all users. A centralized platform will manage data securely, providing tools for data analysis and auditing to monitor resource exchange accurately.
Fairness and inclusion are integral to the platform's design, ensuring equitable resource exchanges that benefits all participants. Additionally, the marketplace will offer customization options to meet the unique requirements of each pilot, allowing for seamless integration of tailored components

6.1.1.2 General Market Features

To ensure the efficient and equitable functioning of the marketplace, several key features will be integrated into its design. These features are fundamental for establishing a secure, transparent, and user-friendly environment that fosters trust and fairness. The following aspects outline how these features will be implemented and their roles in the market:

- **Trust Mechanisms:** Thanks to blockchain technology, the market will enable the implementation of mutual trust and security mechanisms among users. This will be achieved through the transparency and immutability of transactions recorded on the blockchain.

- **Anonymity and Transparency:** Although users may be relatively anonymous, the market structure and the use of smart contracts will allow direct trust between them, resulting in greater transparency and fairness in all transactions.
- **Smart Contracts:** Smart contracts will be used to automate and enforce the policies of access, use, and valuation of resources in the marketplace. These contracts will ensure that all transactions are fair and verifiable.
- **Fair and Reliable Valuation System:** The market will implement a resource valuation system that is valid and logical, ensuring that exchanges are fair and reliable for all participants.

6.1.1.3 Expected Benefits

The marketplace is designed to deliver several key benefits that enhance its overall value and usability for participants. These expected benefits are central to the platform's success and aim to address critical aspects such as transparency, security, and fairness. Following are outlined the anticipated advantages:

- **Transparency:** All market users will have access to transparent information about transactions and the value of resources.
- **Security:** Transactions will be protected against fraud and manipulation thanks to blockchain technology and the design of smart contracts.
- **Fairness:** The valuation system will ensure that resources are exchanged fairly, benefiting all market participants.

6.1.2 Market Division

Before explaining the marketplace, it is important to know two key elements, as it is divided into two main components: **Common Elements** and **Specific Elements**. This division allows for a solid and standardised base that can be customised to meet the unique requirements of each pilot.

6.1.2.1 Common Elements

These are the components of the marketplace that are shared by all pilots. These elements form the **core** of the system and ensure the overall interoperability and efficiency of the market. They can be described as follows:

Blockchain Platform and Smart Contracts:

The marketplace is built on a blockchain platform utilizing Proof of Authority (PoA), which provides a secure foundation for all transactions. This shared platform ensures that all member nodes operate under a unified security protocol, enabling consistent updates. PoA was chosen for its energy efficiency and alignment with our project's objectives.

Smart contracts automate and enforce policies for resource access, use, and valuation. These contracts are self-executing, with the terms of agreements embedded in code. They eliminate the need for intermediaries, reducing human error and ensuring all transactions are transparent and immutable. By automatically executing predefined actions when certain conditions are met, smart contracts foster trust and compliance among participants.

Resource Tokenization:

Tokenization is a crucial element, with tokens representing both fungible and non-fungible assets. The ERC-20 standard is employed for fungible assets, while ERC-721 is used for non-fungible assets, facilitating the digital representation of resources. This process supports the

reciprocal exchange of resources by using tokens for compensation and rewards within the marketplace.

Security and Trust Mechanisms:

Blockchain technology provides a high level of transparency and traceability, as all transactions are recorded immutably. This transparency enhances user trust, as participants can be confident in the security and integrity of the marketplace. Additionally, the robust design of blockchain and smart contracts protects transactions from fraud and manipulation.

User-Friendly Interface:

The marketplace includes a user-friendly interface with basic dashboards for visualization and asset management. These interfaces are designed to be intuitive, facilitating ease of use and participation for all users. They ensure that users can efficiently manage their assets and participate in resource exchanges without technical barriers.

Data Management:

Data within the marketplace is managed through a centralized platform that allows secure sharing among users. While the blockchain records transactions, it does not store data. Instead, the platform maintains a valid and visible record of all transactions and operations. Data analysis and auditing tools are provided to monitor resource exchanges, ensuring integrity and accuracy.

This definition will form part of the common elements of each market; this will ensure that despite the differences between markets, there are shared components that allow interoperability and thus the possible communication between several markets of this kind. This will facilitate the formation of communities of different markets, even if they have different functionalities.

6.1.2.2 Specific Elements

Each pilot has unique requirements and specific objectives that need to be addressed. These elements allow the marketplace to be customised to meet the needs of each region and community.

Austrian Pilot:

- **Renewable Energy Generation and Management:** Specific infrastructure for photovoltaics and wind.
- **Localised P2P Market:** Adaptations for the Energy Gemeinschaft-Regional Energy Community Dreißgen-Spörbichl .
- **Democratic Community Participation:** Design of specific tools for community decision-making.

Portuguese Pilot:

- **ESCO and E-Mobility Business Models:** Integration of multiple business models.
- **Dynamic Energy Management:** Specific mechanisms for data exchange and energy management.
- **Clean Mobility (V2X):** Specific solutions for the integration of clean mobility.

Greek Pilot:

- **Energy Self-Sufficiency:** Design of solutions, maximising the energy self-sufficiency of the community and the island.
- **Smart Charging and V2G:** Adaptations for electric vehicles and boats.
- **Community Participation:** Tools to promote democratic decision-making.

Irish Pilot:

- **Flexibility and Demand Response:** Specific tools for the Irish market.
- **Energy Data Management:** Integration with local platforms for data management.

Hungarian Pilot:

- **Local Generation and P2P Market:** Solutions adapted to local conditions.
- **Electric Mobility Services:** Customization of charging infrastructure and V2G.

Spanish Pilot:

- **Self-Consumption Optimization:** Tools to improve self-consumption efficiency.
- **Electric Mobility and V2G Services:** Integration of specific solutions for electric mobility.
- **Community Participation:** Adaptation of participation tools to the local culture.

6.1.3 Objectives

The objectives of the project are centred on creating a robust and efficient marketplace that enables the exchange and valuation of energy and non-energy resources across different EnCs. Outline of the key objectives:

Create an Interoperable Market Platform:

The primary goal is developing a platform that facilitates the exchange and valuation of both energy and non-energy resources between different EnCs. This platform will utilize standardized tokenization and smart contract protocols to ensure compatibility across various implementations.

Value and Tokenize Energy and Non-Energy Resources:

The platform will convert both tangible and intangible resources into digital assets through tokenization, using standards like ERC-20 for fungible assets and ERC-721 for non-fungible assets. Fair and transparent valuation mechanisms will be implemented to ensure equitable exchanges of resources.

Use of Tokens as a Reward and Compensation Mechanism:

Establishing a token system to reward users for their contributions and services is a key aspect of the marketplace. This system will also support the reciprocal exchange of resources, ensuring both goods and services are adequately compensated.

Implement Trust and Security Mechanisms:

To ensure the security and integrity of transactions, the platform will utilize blockchain technology with Proof of Authority (PoA). Smart contracts will be developed and deployed to automate and enforce policies related to resource access, use, and valuation, ensuring all transactions are transparent and traceable.

Develop an Intuitive and User-Friendly Interface:

Management dashboards and visualization tools will be provided that are easy to use for all marketplace users. These interfaces will facilitate participation and resource exchange, making them accessible and understandable for users at all levels.

Manage and Share Data Securely:

A centralized platform will be created for secure data management and sharing among users. Data analysis and auditing tools will be implemented to monitor resource exchanges, ensuring information integrity and accuracy.

Ensure Fairness and Inclusion in the Marketplace:

To promote fairness and inclusion, valuation mechanisms will be developed to ensure equitable exchanges for all participants. The platform will support diverse user participation, ensuring everyone can benefit from the marketplace.

Customise the Marketplace for Each Pilot:

Identifying and developing components tailored to the specific needs of each pilot project is a priority. These customizations will be seamlessly integrated with the core marketplace to maintain operational coherence and efficiency.

6.1.4 Implementation and Customization

To meet the generic and personalized elements of the project in its entirety, the following approaches have been adopted:

1. **Core Development:**
   The development of the marketplace involves designing and developing common components to ensure a robust and secure core system. Rigorous testing and audits will be conducted to validate the strength and security of the platform.
2. **Pilot Customization:**
   Each pilot's unique requirements will be identified to develop and implement the necessary components for specific regions and communities. These customizations will be seamlessly integrated with the marketplace core to ensure smooth and coherent operations.
3. **Deployment and Validation:**
   The marketplace will be deployed in test environments, and field validations will be conducted to gather feedback. Implementations will be adjusted based on the feedback received, and continuous improvements will be made. The system will be deployed in production, and ongoing monitoring of performance and security will be maintained.

6.1.5 Execution Checklist

Phase 1: Preparation and Planning

Initial Project Meeting:

- Gather development, management, and stakeholder teams.
- Define roles and responsibilities.

Requirements Definition:

- Collect general and specific requirements from each pilot.
- Document the requirements in detail.

Project Planning:

- Create a detailed project timeline.

- Define key milestones and deliverables.

Risk Analysis:

- Identify potential risks.
- Develop mitigation strategies.

Phase 2: Core Development

System Architecture Design:

- Design the system architecture, including blockchain, smart contracts, and interfaces.

Platform Development:

- Develop the core elements of the marketplace.
- Implement blockchain technology with PoA.
- Develop smart contracts for resource valuation and tokenization.

Token Design:

- Design and implement tokens (ERC-20, ERC-721, ...) for the marketplace.

Security Measures:

- Implement security measures to protect transactions
- Conduct security audits and penetration tests.

User Interface Development:

- Develop user-friendly dashboards and interfaces.

Phase 3: Pilot Customization

Requirements Analysis for Each Pilot:

- Analyse specific requirements for each pilot.

Customization Development:

- Develop specific components for each pilot.
- Integrate customizations with the core platform.

Testing and Validation:

- Conduct testing and validation for customizations.
- Perform security testing for each pilot.
- Adjust and optimise based on test results.

Phase 4: Field Deployment and Validation

Controlled Environment Deployment:

- Implement the marketplace in controlled environments.

Field Validation:

- Conduct field tests in the pilot locations.
- Collect user feedback.
- Make necessary adjustments based on the feedback.

Continuous Improvement:

- Implement improvements based on the feedback and results of the field tests.

Phase 5: Production Deployment and Maintenance

Production Deployment:

- Implement the marketplace in production environments.
- Ensure that all functionalities are operational.

Monitoring and Support:

- Establish monitoring systems for performance and security.

- Provide continuous technical support.

Maintenance and Updates:

- Develop a continuous maintenance plan.
- Implement updates and improvements as needed.

Review and Control

Periodic Reviews:

- Schedule periodic review meetings to monitor progress.
- Update the checklist and timeline as necessary.

Documentation and Reporting:

- Maintain up-to-date documentation of all activities and changes.
- Generate progress reports and share with stakeholders.

6.2 Marketplace Structure

6.2.1 Definition of the Marketplace and its technologies

To understand the structure of the marketplace that will be developed for Task 4.4, we must first review the previously discussed concepts and established objectives.

Marketplace: A tokenized marketplace is a digital platform where resources can be exchanged using digital tokens. These tokens represent values or rights over specific resources, whether they are energy or non-energy, tangible, intangible, or virtual (such as NFTs). The use of tokens facilitates transactions, making them fast, secure, and transparent.

Blockchain: Blockchain is a type of DLT (Distributed Ledger Technology) where transactions are recorded in blocks that are chronologically and immutably linked. This ensures that records are transparent and resistant to tampering.

The marketplace will serve as a system functioning like a virtual notary. In this system, all elements will be always secured, preventing any unauthorised modifications not pre-approved by the blockchain or the members themselves. This way, we follow a principle of anonymous trust, where users, despite not knowing who is, can be assured and maintain their information shared publicly without the risk of their data or integrity being compromised.

6.2.2 Impact and Relevance of the Marketplace in ENPOWER System

6.2.2.1 Impact of the Marketplace

Encouraging Active Participation:

The marketplace transforms passive energy consumers into active participants, enabling them to take full control of their energy usage. This active participation allows users to optimize their energy consumption, resulting in a more sustainable ecological impact.

Transparency and Trust:

The platform offers a transparent and secure system for resource exchange, ensuring the integrity of transactions through blockchain technology. This transparency builds trust among users by providing clear and immutable records of all exchanges.

Transaction Efficiency:

Digital tokens facilitate fast and secure transactions, significantly reducing the need for intermediaries. This streamlining lowers costs and enhances efficiency, making transactions more straightforward and accessible for users.

Interoperability:

The marketplace is designed to support interoperability between different communities and regions. By using common standards for tokenization and smart contracts, it ensures compatibility and ease of integration, allowing seamless interaction across various platforms.

Relevance of the Marketplace in the ENPOWER Project

Empowering Citizens:

The marketplace enables citizens to actively participate in managing their energy resources, providing them with tools for informed and responsible decision-making. This empowerment enhances user engagement and supports the transition to a more sustainable energy ecosystem.

Data-Driven Energy Security:

By leveraging data, the marketplace improves the security and efficiency of the energy system. It offers a secure framework for the exchange of data and resources, ensuring that energy transactions are reliable and transparent.

Innovation and Sustainability:

The platform promotes the adoption of innovative technologies such as blockchain and DLT. By encouraging efficient management of both energy and non-energy resources, the marketplace fosters sustainable practices and supports long-term ecological goals

Facing pilots' barriers

The marketplace architecture and the use of blockchain technology will facilitate the overcoming of some of the barriers that pilots are bringing back:

Table 3 Facing pilots' barriers.

Pilot	Barriers
Austria	Wind turbines ownership for EnCs Wind farms capacity and the size of EnC is mismatched Possible change in the law governing the electricity industry
Portugal	No significant legal barriers Legal framework does not support the integration of smart EV charging within RECs as the DSO cannot associate energy flows within smart chargers
Greece	None: Existing regulations do not significantly impact pilot implementation
Ireland	No support system exists for community PV for grid export TAM (Targeted Agricultural Modernisation Scheme) grants might be insufficient to cover the costs of necessary energy infrastructure or modernization
Hungary	Regulatory gaps: Energy can only be shared between properties with the same parcel number
Spain	No aggregator role defined in law Minimum capacity requirements for demand response programs hinder EnCs' participation Lack of technical ways to dynamically inform about energy exchanges

Legal aspects will not be addressed, as they are not technology dependent. However, the blockchain technology can help and be efficient in tracking, defining, and modifying roles and assets within EnCs.

It can help overcome barriers identified in the following three pilots:

Austrian Pilot

To handle challenges with wind turbines ownership and the mismatch between wind farm capacity and EnC size the solution could be the tokenization of wind turbine assets for collective ownership. Devices can be owned by EnCs and shared among participants. Excess energy can be tokenized and sold to other EnCs or consumers through the marketplace.

Spanish pilot

To address the minimum capacity requirements for demand response, the solution is to aggregate the capacity of multiple participants. Households combine their energy use to meet demand response thresholds. For example, several households within the EnC collectively meet the minimum requirement, allowing them to participate in demand response programs and share the benefits proportionally. The lack of dynamic P2P energy exchange information is solved by providing real-time updates and notifications through blockchain events. Members receive real-time transaction notifications, enabling informed energy trading decisions.

Hungarian pilot

Even though there is a regulatory gap, this issue could be resolved through virtual energy sharing agreements. The marketplace facilitates virtual agreements with smart contracts. A property with solar panels generates excess energy, which is tokenized and sold to a neighbour with a different parcel. The transaction is recorded on the blockchain, ensuring transparency and compliance with virtual agreements.

6.2.3 Marketplace Architecture

The marketplace will be structured based on the topology and interactions occurring in the EnCs. Since the previous deliverables (D2.1, D2.2, D2.3, D2.4), we can define the architecture of the protocol that will be implemented using smart contracts within the ENPOWER marketplace, delineating the roles of the actors. It elucidates the essential elements and interrelations constituting the marketplace's overarching architecture.

Entities

- **Region**: area in which one or more EnCs can coexist
- **Archetype**: refers to a specific model or pattern of EnCs that can be replicated or adapted based on specific characteristics e.g. Local Energy Market, E-Mobility Cooperative
- **Governance**:
    - **Cooperative EnCs**: Governed by cooperative principles where members have equal voting rights.
    - **Municipal EnCs**: Managed by local government entities with a focus on community-wide benefits.
- **EnC**: belongs to a Region, from which inherits Archetype and Governance, or can have specific ones; has a Marketplace, with which its EnC Users can interact; can own devices, thus shared with the entire community
- **EnC User**: e.g. Prosumer, DER Owner, etc.

- **EnC Super User**: e.g. EnC Manager, Oracle, Aggregator, Grid Operator, etc.
- **Device**: e.g. solar panel, wind turbine
- **EnC Marketplace**: in which trades take place within an EnC
- **Region Marketplace**: where trades take place among EnCs from the same region or between different regions

In the context of smart contracts, it is possible to define registries, contracts in which certain entities are registered, which can perform checks to validate specific characteristics

- Region Registry
- EnC Registry
- Device Registry
- EnC Marketplace Registry
- Region Marketplace Registry

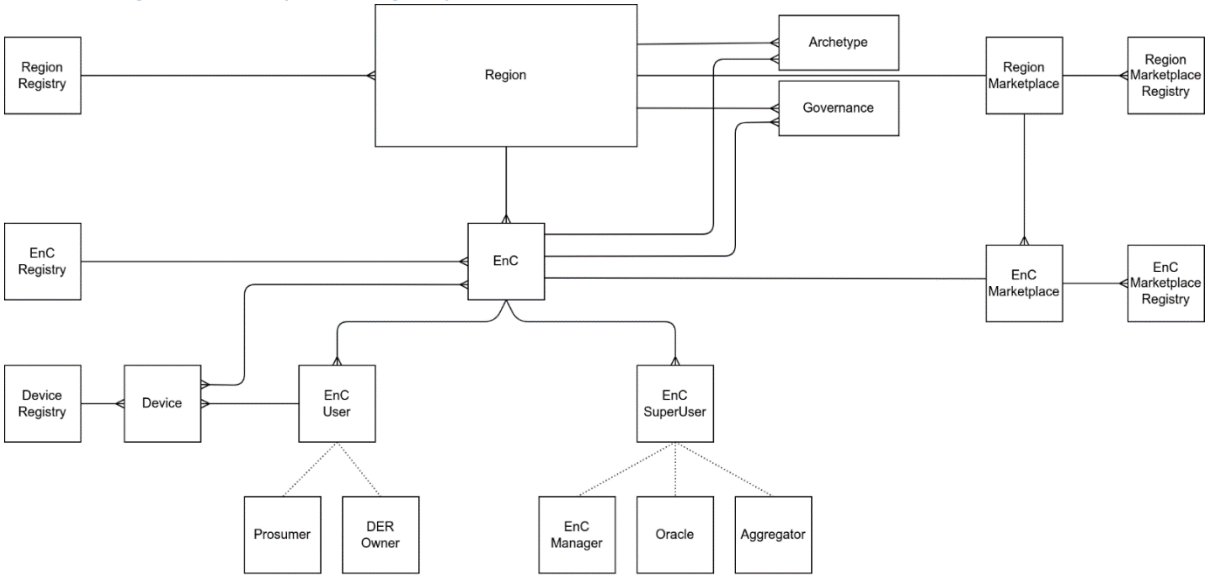

Figure 32: Marketplace architecture.

Logical View

The logical view outlines the structure of the ENPOWER system architecture, focusing on how the system components interact logically to achieve the defined use cases. It comprises layers, each with distinct responsibilities, ensuring modularity and separation of concerns.

- **Presentation Layer**: User Interface: Provides a user-friendly interface for users to monitor and manage energy production, consumption, and trading.
- **Business Logic Layer:**
  - **Energy Trading Engine:** Facilitates P2P energy trading between users, executes smart contracts, and ensures transaction integrity.
  - **Demand Response Module:** Manages demand response events, coordinating with grid operators and users to optimise energy consumption.
- **Data Access Layer:** Ensures secure and transparent transactions using blockchain technology, handles smart contract execution and reads.

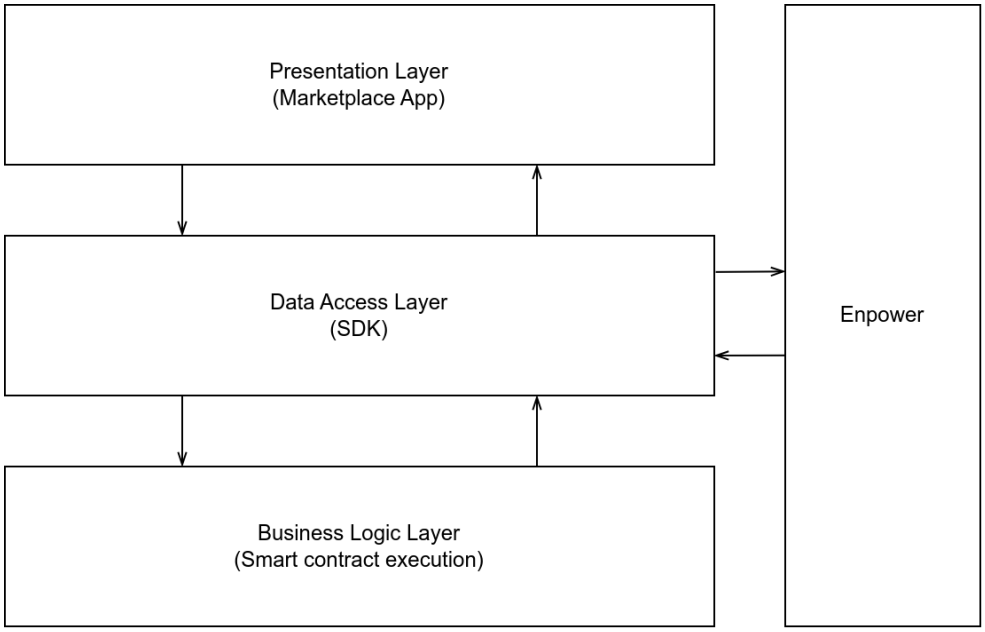

Figure 33: System components' interactions.

7. Conclusions

Deliverable 4.1 represents a significant milestone in the ENPOWER project, marking the first technology release within Work Package 4 (WP4): SHARE. This document outlines the foundational elements required to adapt the Energy Data Space (EDS) and establish a decentralized peer-to-peer (P2P) marketplace utilizing distributed ledger technology (DLT) and blockchain. By focusing on consumer-centric energy systems, this deliverable sets the stage for enhanced consumer and community-level participation in energy markets and demand response activities.

The comprehensive architecture of the ENPOWER system integrates multiple data-driven initiatives and technologies, including BRIDGE, IDS, GAIA-X, FIWARE, and the CEEDS Architecture Blueprint. These initiatives provide a robust framework for developing a secure, interoperable, and scalable system. The emphasis on privacy, security, and legal compliance ensures adherence to the highest standards, with particular attention to GDPR, identity, and access management protocols.

Furthermore, the deliverable delves into the critical aspects of data model and service/platform interoperability. The SEMAPTIC approach and the ENCOM ontology for EnCs are pivotal in achieving semantic interoperability, facilitating seamless communication and data exchange within the ENPOWER system. The development of tools compliant with Energy Data Space standards, such as the FIWARE Context Broker and OneNet Connector, underscores the commitment to sovereign data sharing and federated planning.

The creation of an open, interoperable, tokenized marketplace is a cornerstone of this deliverable. This marketplace not only enables the reciprocal compensation of energy and data assets but also drives consumer engagement and fosters innovation within the energy sector. By outlining the marketplace's structure, objectives, and technological foundation, this deliverable highlights its potential impact and relevance to the ENPOWER project.

As the first of three key technology releases, Deliverable 4.1 provides a solid foundation for the subsequent stages of development. The insights and advancements documented herein will guide the progression toward the final release, ultimately contributing to the realization of a consumer-centric, efficient, and sustainable energy system. The ongoing commitment to innovation, interoperability, and security will ensure that the ENPOWER project continues to address the evolving needs of energy consumers and communities.

In summary, the ENPOWER Energy Data Space and DLT-blockchain digital P2P marketplace project underscores the transformative potential of integrating advanced technologies into the energy sector. The implementation of energy data spaces, combined with blockchain-based marketplaces, offers a robust framework for enhancing data interoperability and enabling efficient, decentralized energy trading. This approach improves the accessibility and usability of energy data and supports the development of a more resilient and adaptable energy market. The project's outcomes pave the way for future advancements in the energy sector, offering a blueprint for integrating digital innovations into traditional energy markets to drive efficiency, sustainability, and resilience.

8. References

[1] B. Wolford, “What are the GDPR consent requirements? - GDPR.eu,” [Online]. Available: https://gdpr.eu/gdpr-consent-requirements/. [Accessed May 2024].

[2] European Union. "General Data Protection Regulation." Official Journal of the European Union, L 119, 4 May 2016, p. 1. Article 4. https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679

[3] R. Struder, R. Benjamins, and D. Fensel, ‘Knowledge Engineering: Principles and methods’, 1997.

[4] ‘Semantic interoperability’, *Wikipedia*. Jan. 28, 2024. Accessed: Apr. 14, 2024. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Semantic_interoperability&oldid=1199864205