Extended Interoperability and Management with FIWARE

D6.3

Authors:

Ferdinando Bosco (ENG)

Angelo Triveri (ENG)

Pierluigi Linardi (ENG)

Juan Galeano Martinez (RWTH)

Iris Koster (RWTH)

Apostolos Kapetainos (ED)

Konstantinos Kotsalos (ED)

Vassilis Sakas (ED)

Responsible Partner	ENG
Verified by the appointed Reviewers	Rui Pestana (REN), 26.07.2023 Kos Anton (EC), 20.07.2023
Approved by Project Coordinator	Padraic McKeever (Fraunhofer), 31.07.2023

Dissemination Level	Public

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

Issue Record

Planned delivery date	31.07.2023
Actual date of delivery	31.07.2023
Status and version	V1.0

Version	Date	Author(s)	Notes
0.1	05/04/2023	Ferdinando Bosco (ENG)	Table of Contents
0.2	30/05/2023	Juan Galeano Martinez (RWTH) Iris Koster (RWTH)	Contribution on Ch. 3.1, 3.5 and 4.3
0.3	17/07/2023	Apostolos Kapetainos (ED) Konstantinos Kotsalos (ED) Vassilis Sakas (ED) Ferdinando Bosco (ENG) Pierluigi Linardi (ENG) Angelo Triveri (ENG)	Contribution on Ch. 3.2 and 4.1 Contributions on Ch. 1, 2, 3.3, 3.4, 4.2, 5. Consolidation of the entire document. Document ready to be reviewed.
0.4	26/07/2023	Kos Anton (EC) Rui Pestana (REN) Ferdinando Bosco (ENG)	Document reviewed by internal reviewers and consolidated for final quality check.

Disclaimer:

All information provided reflects the status of the OneNet project at the time of writing and may be subject to change. All information reflects only the author's view and the European Climate, Infrastructure and Environment Executive Agency (CINEA) is not responsible for any use that may be made of the information contained in this deliverable.

About OneNet

The project OneNet (One Network for Europe) will provide a seamless integration of all the actors in the electricity network across Europe to create the conditions for a synergistic operation that optimizes the overall energy system while creating an open and fair market structure.

OneNet is funded through the EU's eighth Framework Programme Horizon 2020, "TSO – DSO Consumer: Large-scale demonstrations of innovative grid services through demand response, storage and small-scale (RES) generation" and responds to the call "Building a low-carbon, climate resilient future (LC)".

As the electrical grid moves from being a fully centralized to a highly decentralized system, grid operators must adapt to this changing environment and adjust their current business model to accommodate faster reactions and adaptive flexibility. This is an unprecedented challenge requiring an unprecedented solution. The project brings together a consortium of over seventy partners, including key IT players, leading research institutions and the two most relevant associations for grid operators.

The key elements of the project are:

1. Definition of a common market design for Europe: this means standardized products and key parameters for grid services which aim at the coordination of all actors, from grid operators to customers;
2. Definition of a Common IT Architecture and Common IT Interfaces: this means not trying to create a single IT platform for all the products but enabling an open architecture of interactions among several platforms so that anybody can join any market across Europe; and
3. Large-scale demonstrators to implement and showcase the scalable solutions developed throughout the project. These demonstrators are organized in four clusters coming to include countries in every region of Europe and testing innovative use cases never validated before.

Table of Contents

Copyright 2023 OneNet

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

List of Figures

Copyright 2023 OneNet

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

List of Tables

Copyright 2023 OneNet

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

List of Abbreviations and Acronyms

Acronym	**Meaning**
API	Application Programming Interface
CaPe	Consent-based Personal Data Suite
CEF	Connecting Europe Facility
CIM	Common Information Model
COSMAG	Comprehensive Architecture for Smart Grid
ETSI	European Telecommunications Standards Institute
HTTP	HyperText Transfer Protocol
IDS	International Data Space
JSON	JavaScript Object Notation
NGSI-LD	Next Generation Service Interface - Linked Data
RAM	Reference Architectural Model
RDF	Resource Description Framework
REST	Representational state transfer
UML	Unified Modeling Language
XML	Extensible Markup Language
XSD	XML Schema Definition

Copyright 2023 OneNet

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

Executive Summary

The OneNet Solution mainly relies on the concept of a pan-European ecosystem, which, through the adoption of standardised interfaces and data models, enables a seamless integration and cooperation among energy stakeholders for cross-platform market and network operation services.

The analysis conducted for the definition of the OneNet Reference Architecture highlighted as, among the main requirements for its implementation, that the adoption of FIWARE assumes fundamental importance for ensuring a standardized, secured, and trusted data exchange, maintaining the data control and ownership.

The main components of the OneNet Solution are the OneNet Decentralized Middleware and the OneNet Connector. The OneNet Connector enables decentralized data exchanges in the OneNet ecosystem, supporting the creation of the OneNet Network of Platforms.

An important part of the OneNet Connector is the Data Integration & Homogenization sub-layer, which manages the end-to-end data exchange process and provides a number of additional data-based services, directly at the connector layer.

This document is therefore focused on the detailed description of the Data Integration & Homogenization sub-layer and on how the adoption of FIWARE can facilitate a decentralized and interoperable approach in the implementation of the OneNet Solution.

The Data Integration & Homogenization sub-layer includes: the FIWARE Data App (integrated with the Orion NGSI-LD Context Broker for managing the data exchange in a standardized way and the OneNet Data Service Layer, which includes a Data Homogenization Tool, which supports the integration of the CIM standards with the FIWARE NGSI-LD information model, as well as two of the main building blocks of the IDS architectural model for the data management aspects: the Clearing House and the Usage Control App.

- The FIWARE Data App manages the data exchange process, leveraging on the NGSI-LD standard, and offers standardized interfaces for the integration of the external platforms and systems through the OneNet Connector GUI and/or Local API components.
- The Data Harmonization tool is capable to map the CIM standards I NGSI-LD standard and convert XML data format in JSON-LD formation.
- The Clearing House service plays the role of intermediary that logs all activities performed during data exchange in the IDS ecosystem and provides clearing and settlement services for all financial and data exchange transactions.
- The Usage Control App allows definition of Usage Policies and Usage Enforcement. Each Data Owner and Data Provider can define access and usage control policies for their data, applied during each data transaction.

Copyright 2023 OneNet

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

The Data Integration & Homogenization tool is an important sub-layer of the OneNet Connector because it integrates several interoperability and standardization aspects fundamental for the implementation of a data space ecosystem such as OneNet.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

1 Introduction

This chapter describes the context in which the activities of WP6, and more specifically the T6.3, are placed and how they are coordinated and linked within the other project activities. In addition, a detailed description of the structure and objectives of this document is provided.

1.1 Scope

One of the main goals of OneNet is to implement an open and flexible architecture to transform the actual European electricity system, which is often managed in a fragmented country- or area-level way, into a pan-European smarter and more efficient one, while maximizing the consumer capabilities to participate in an open market structure.

WP6 "Reference IT Implementation for OneNet" contributes to fulfilling the OneNet vision by striving to attain four objectives:

- Develop the OneNet Decentralised Middleware (and OneNet Connector) that will enable the interconnection of energy actors and seamless data management and exchange.
- Incorporate the data interoperability and cyber security aspects into the OneNet System.
- Incorporate the Big Data and the FIWARE layers into the OneNet System.
- Develop and test the OneNet Interoperable Network of Platforms.

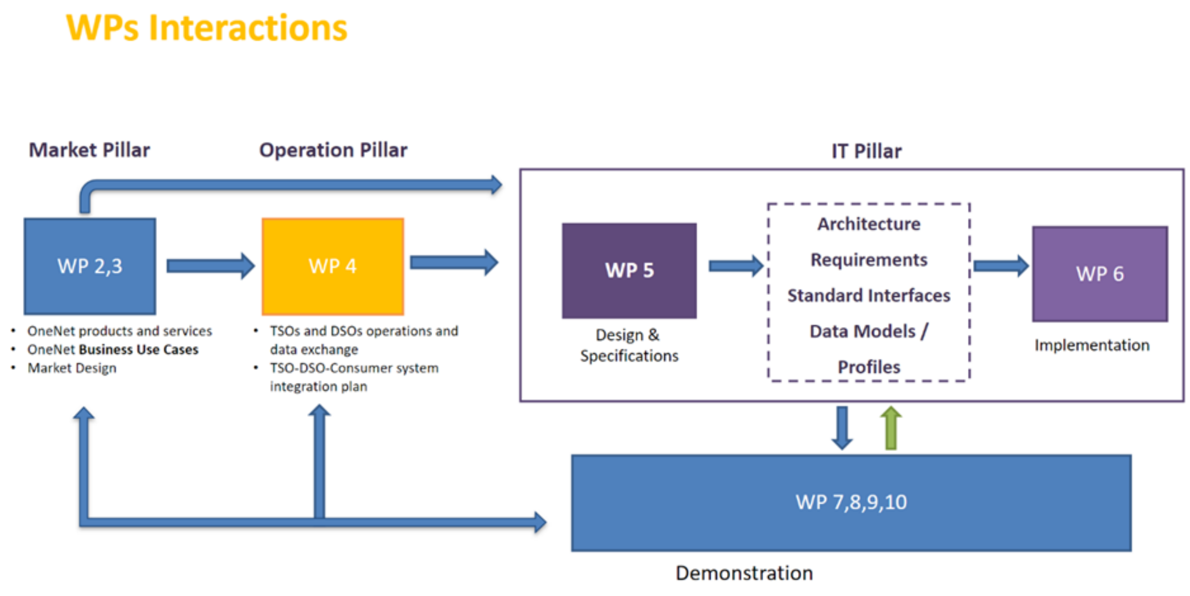


Figure 1: WP6 interactions

Copyright 2023 OneNet

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

WP6, following the design phase of the WP5, acts as IT main pillar of the overall OneNet project, covering the implementation phase.

The IT pillar is closely linked to all the other pillars of the project, as shown in Figure 1. It takes into consideration all the results provided in the Market Pillar (WP2 and WP3) as well as the Operation Pillar (WP4). In addition, the OneNet Solution, implemented in WP6 will be tested and evaluated in 4 Demonstration Clusters and the results of the evaluation will be used for adapting, improving, and enhancing the OneNet Solution.

1.2 Task 6.3

The main goal of the Task 6.3 "Data and Service Interoperability, Integration with FIWARE, and Homogenization Management" is to incorporate the FIWARE layers and the data interoperability tools into the overall OneNet System, implementing and integrating the Data Integration & Homogenization sub-layer.

Task 6.3, which started in M7 of the OneNet project, collects all the necessary information provided from the WP5 ”Open IT Architecture for OneNet” tasks and in particular: the requirements and use cases (from T5.1), the OneNet Reference Architecture (T5.2) the FIWARE interoperable interfaces (T5.3) and the interoperability specifications (T5.6).

All these inputs were used as reference for the implementation of the Data Integration & Homogenization sub-layer.

In addition, T6.3 strictlty collaborated with the T6.1 for the implementation of the OneNet Decentralised Middleware and OneNet Connector, as the Data Integration & Homogenization sub-layer is an integral part of the connector itself, as well as with the T6.5 for the integration and validation phase.

Finally, the T6.3 produced the following results:

- the **first release of the Data Integration & Homogenization sub-layer** included in the release of the OneNet Connector in **July 2022** and reported in D6.5 and MS12.
- the **intermediate release of the Data Integration & Homogenization sub-layer**, integrated with the OneNet Connector and within the overall OneNet System (including other components) in **January 2023**, reported in D6.1 and MS16.
- the **final release of the Data Integration & Homogenization sub-layer**, **at July 2023** and reported in this document.

1.3 Outline of the deliverable

This deliverable is structured in 5 different chapters.

The Introduction, Chapter 1, puts the deliverable into context within the WP6 and task T6.3.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

Chapter 2 analyses the FIWARE Smart Energy Architecture and its relationship with the OneNet System and with the OneNet Connector.

Chapter 3 analyses the most relevant Standardised Data Models and Ontologies.

Chapter 4 describes the implementation of the Data Integration & Homogenization sub-layer and its relationship with the OneNet Connector from an architectural perspective.

Finally, chapter 5 concludes the document.

To facilitate the readability of D6.3, it might be useful to refer to D5.2 [2], D5.4 [3] for information about the design phase and D6.1 [20], D6.5 [30] for some references about the implementation and integration phase with the OneNet Connector.

2 FIWARE Smart Energy Architecture

2.1 Introduction

FIWARE is an Open-Source initiative defining a set of standards for context data management that facilitate the development of smart solutions for different domains, including smart energy.

The FIWARE Foundation encourages the adoption of a transparent, common, collaborative data sharing framework capable of reaching its full potential in developing smart applications.

The main goals of the FIWARE initiative are:

- accelerating the development and adoption of Future Internet technologies in Europe,
- advancing the European market for smart infrastructures, and
- increasing the effectiveness of business processes through the Internet.

2.2 FIWARE and Smart Energy

The Energy Transition is transforming the energy sector from straight supply chain to a complex ecosystem.

The digitalisation of the energy sector requires higher levels of operational excellence with the adoption of disruptive technologies to enable cross-sector data sharing and data-driven applications. The following key aspects in data management must be fulfilled:

- Data model/semantics: Defining an appropriate data model appropriate data model beyond a single sector is a key ingredient for interoperability.
- Context information: Defining the context is a key ingredient for bridging the gap between different verticals.
- Data sovereignty: The ability of a data owner to define what a third party is allowed to do with its data.
- Open APIs: Closed solutions will not create a truly open and open and competitive market. Open APIs offer the perfect bridge between the private infrastructure spaces.

Modern grid applications need to accommodate the interaction of a multitude of stakeholders. The main goal of a smart grid implementation is the creation of an automatic process, where data exchange, semantics and related communication protocols are fundamental.

FIWARE can be a solution for the implementation of standardized and secure data exchange in the smart energy sector.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

Reference Architecture Model

Figure 2 below shows the FIWARE Reference Architecture for Smart Energy solutions.

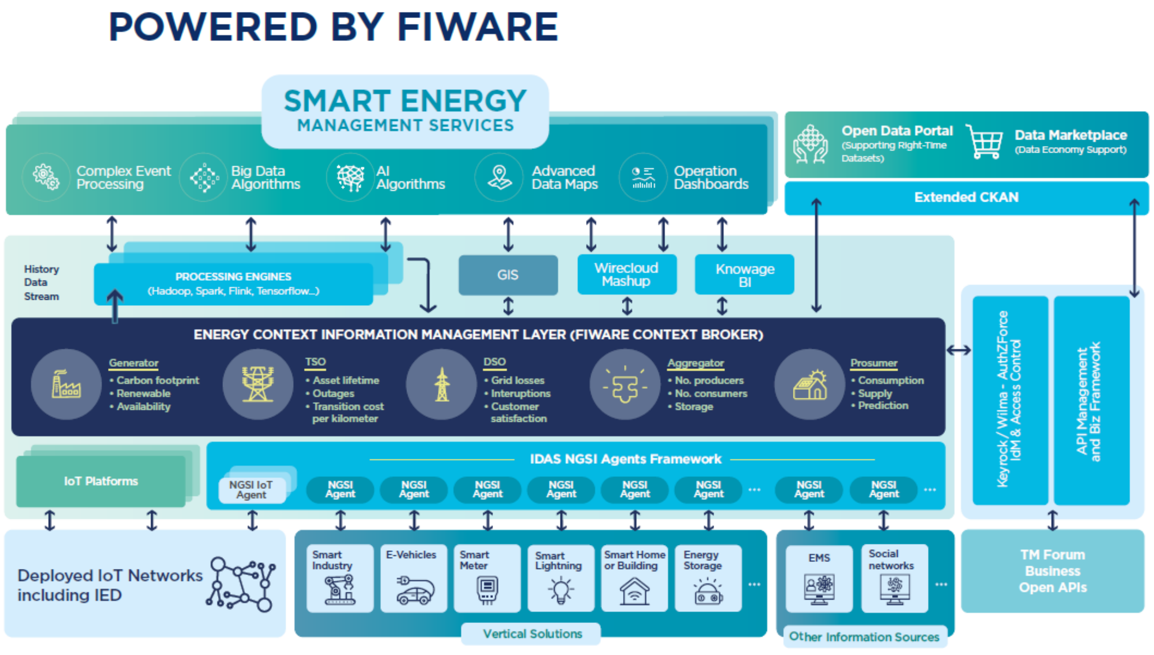


Figure 2: FIWARE Smart Energy Reference Architecture Model [1]

The FIWARE Smart Energy Reference Architecture Model has been considered during the design of the OneNet system reference architecture and the gap analysis which conducted to the implementation of the OneNet Connector, mixing IDS and FIWARE reference model and technologies. For additional information about the design phase please refer to D5.2 [2] and D5.4 [3].

2.3 Components and Tools

FIWARE is a framework of Open-Source components that can be integrated with third party components to build innovative solutions. The main and only mandatory component of this open-source framework is the **Orion Context Broker** [6]. The Context Broker enables to manage all the whole lifecycle of context information including updates, queries, registrations, and subscriptions. FIWARE NGSI is the standard interface exported by a FIWARE Context Broker, used for the integration of platform components within a FIWARE solution and by applications to update or consume context information. FIWARE NGSI API specifications have evolved over time, initially matching NGSI-v2 specifications, now aligning with the ETSI NGSI-LD standard [5].

2.3.1 FIWARE NGSI-LD Context Broker

FIWARE Orion Context Broker, is the core component of the FIWARE platform and implements the Next Generation Service Interface NGSI standard to deploy a Publish/Subscribe architecture.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

In the FIWARE framework, Orion Context Broker [6] is a key service to ease the development and provisioning of smart and innovative applications that require context information and data stream management, processing, and exploitation. The Orion Context Broker is an HTTP Publish/Subscribe implementation—based on the NGSI standard—that enables management of the entire lifecycle of context information including updates, queries, registrations, and subscriptions. Orion Context Broker has been recognized as a CEF Building Block, which is one step forward on its path towards becoming a global standard for large-scale contextual information management. Orion allows defining a model of data (i.e. entity) to which publishers update values to be obtained by subscribers. Orion uses the NoSQL MongoDB database to store these entities and the last value recorded on them.

In NGSI information model, contextual elements are referred to as entities. Entities are physical objects (sensor, actuator, etc.), hardware, and software as presented in Figure 3. Based on this conceptual model, entities are uniquely identified by an Entityid. Each entity can have attributes that are related to an entity's characteristics. These attributes can have static or dynamic values and represent by the triplets <Name, Type, Value>.

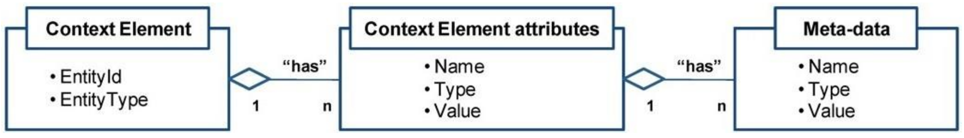


Figure 3: Context element conceptual model [7]

2.3.2 Evolution of FIWARE Context Broker NGSI into NGSI-LD

The FIWARE NGSI v2 information model has been extended to support Linked Data (LD) which is including entities relationships, property of graphs, and semantics. This evolution has been conducted under the ETSI CIM initiative and named NGSI-LD. NGSI-LD targets semantic context information, its specification creates models of real-world entities, relationships, and properties and is expressive to connect and federate other existing information models, using JSON-LD as a lightweight linked data format. Linked data is structured data that is interlinked with other data, so it becomes more useful through semantic queries.

According to Figure 4, NGSI-LD is defined at two levels consisting of the Core Meta-Model, the Cross-Domain Ontology. The NGSI-LD Core-Meta Model represents Entities, their Relationships, and their properties with the value according to the "property graph" model [9]. The main constructs of NGSI-LD are Entity, Property, and Relationship. The classes in Cross-Domain are a set of generic and transversal classes to avoid redundant definitions in the Domain-Specific Ontologies. Indeed, NGSI-LD information model contains the core terms needed to uniquely represent the key concepts of the NGSI-LD information model as well as the terms that define the API-related data types. These are encoded using JSON-LD, which provides the advantage of being familiar and accessible to developers.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

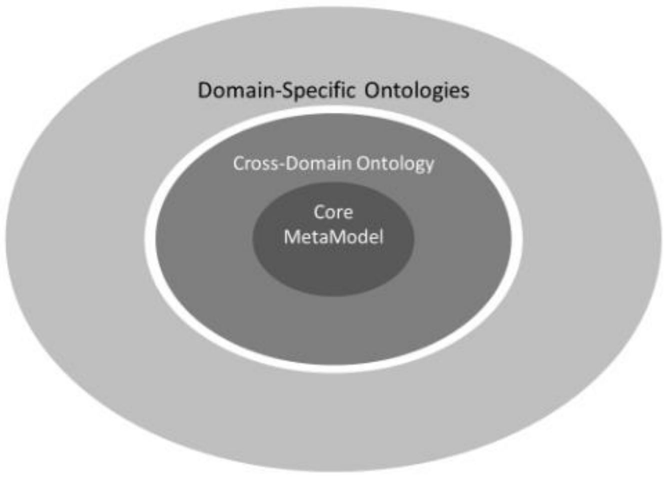


Figure 4: Overview of the NGSI-LD information model structure [8]

For example, an Entity, representing a device or other data source, is encoded using a JSON-LD object. The data in NGSI-LD are structured like a data graph model and linked with each other. The relationships between entities are easily handled in NGSI-LD in a similar way to graph databases. It also provides the update, query, and subscription support needed to allow applications to automatically access data from various data sources. Currently, there exist 3 implementations of NGSI-LD that are summarized in the table:

Table 1: FIWARE Context Broker implementation

Name	NGSI v2	NGSI-LD	Description
Orion[1]	Supported	Not Supported	The original Orion and supporting NGSI V2
Orion-LD[2]	Supported	Supported	A fork from the original Orion repository and aims to be merged back at some point. It is the only context broker which can service both NGSI-v2 and NGSI-LD
Scorpio[3]	Not Supported	Supported	pure NGSI-LD brokers which does not require the compromises of having to serve both syntaxes. Positions itself as the heavyweight broker, with a strong interest in federations.
Stellio[4]	Not Supported	Supported	pure NGSI-LD brokers which does not require the compromises of having to serve both syntaxes. It is somewhere in the middle between Scorpio and Orion.

[1] Orion: https://fiware-orion.readthedocs.io/en/master/
[2] Orion-LD: https://github.com/FIWARE/context.Orion-LD
[3] Scorpio: https://scorpio.readthedocs.io/en/latest/
[4] Stellio: https://github.com/stellio-hub/stellio-context-broker

Copyright 2023 OneNet

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

2.4 OneNet System and FIWARE Alignment

As described in D5.2 [2], while the OneNet Decentralised Middleware offers central features to all the OneNet participants like identity management, sources discovery, semantic annotation, vocabularies and ontologies, the OneNet Connector is a decentralised instance of the OneNet Middleware itself and is responsible for the execution of the complete data exchange process.

Following the analysis conducted during the design phase for the OneNet Connector (reported in D5.2), it was clear that while the IDS reference architecture model architecture (RAM) and its own connector implement all the aspects and characteristics necessary to implement a data space ecosystem, the FIWARE ecosystem is needed to fulfil all the needs and goals of the OneNet System, introducing the standardization aspects for the data exchange process.

The OneNet Connector will therefore consist of a hybrid integration of the IDS Reference Model (IDS connector) and FIWARE Context Broker, using NGSI-API.

The Figure 5 below shows how the FIWARE ecosystem and IDS environment integrate each other for implementing a complete end-to-end data exchange process.

The FIWARE Data App and the Orion Context Broker, described in Ch. 4.2 are the main components responsible for the integration of the FIWARE ecosystem.

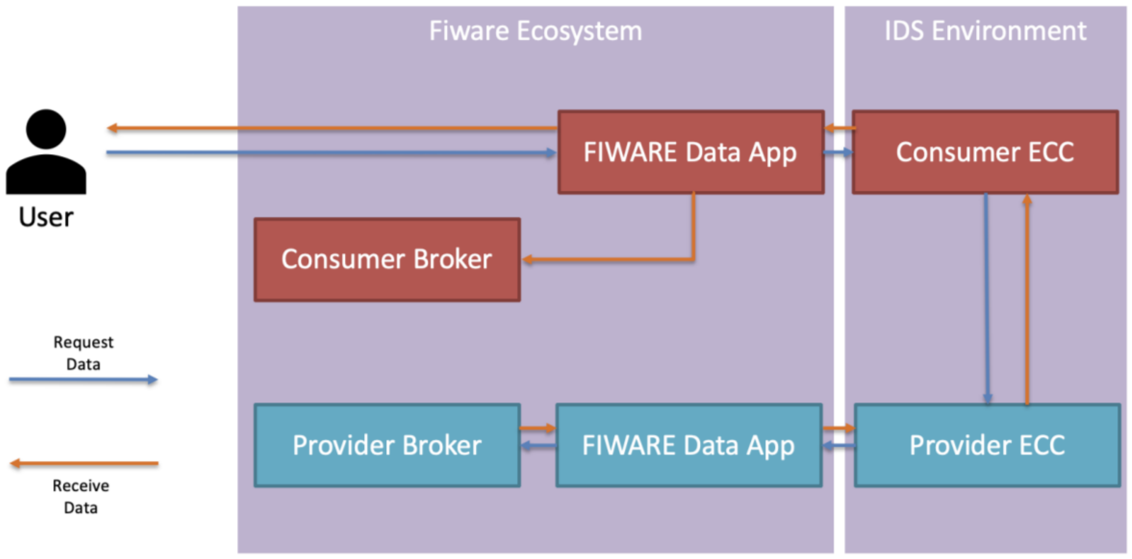


Figure 5: FIWARE and IDS integration in OneNet Data Exchange

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

3 Standardized Data Models and Ontologies

3.1 CIM Data Models

As part of Task 5.6, activities were conducted to understand the data models utilized by the different participants. During the activities mentioned before, CIM data models were defined or discovered to be utilized in different services in the OneNet ecosystem. These data models are an important step towards having well-defined data and associations that are not tied to specific technology or vendor and can ensure interoperability within smart grids.

3.1.1 IEC 62325 as JSON schema

The IEC 62325-451 standard was initially acquired as a compilation of XML Schema definition files (XSD files) sourced from the ENTSOE webpage [10]. As an initial step towards the implementation of the Data Homogenization tool (section 4.3), these XSD files underwent conversion into JSON schema files. Figure 6 provides an overview of the XSD files that define the CIM standard IEC 62325-451.

Table 2: IEC 62325-451 standard tested in the tool

Payload Type	Version
ResourceCapacityMarketUnit_MarketDocument	1.2
Balancing_MarketDocument	4.5 - 3.0
ImplicitAuctionResult_MarketDocument	7.1 - 7.0 - 7.0
CapacityAllocationConfiguration_MarketDocument	1.3 - 1.0 - 1.0 - 1.1
Weather_MarketDocument	1.1
AnomalyReport_MarketDocument	5.3
Activation_MarketDocument	6.0
ConstraintNetworkElement_MarketDocument	1.0 - 1.0
ProblemStatement_MarketDocument	3.1 - 3.0
AllocationResult_MarketDocument	7.2
Unavailability_MarketDocument	4.1 - 3.0
StatusRequest_MarketDocument	4.1 - 4.0
Bid_MarketDocument	7.1 - 7.0
AreaConfiguration_MarketDocument	1.1 - 1.0

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

Table 2 presents a list of the payloads and the respective versions that were successfully tested with the tool; however, it is important to note that there is a broader range of payloads that have been tested and can be found in the repository mentioned in Chapter 4.3.6.

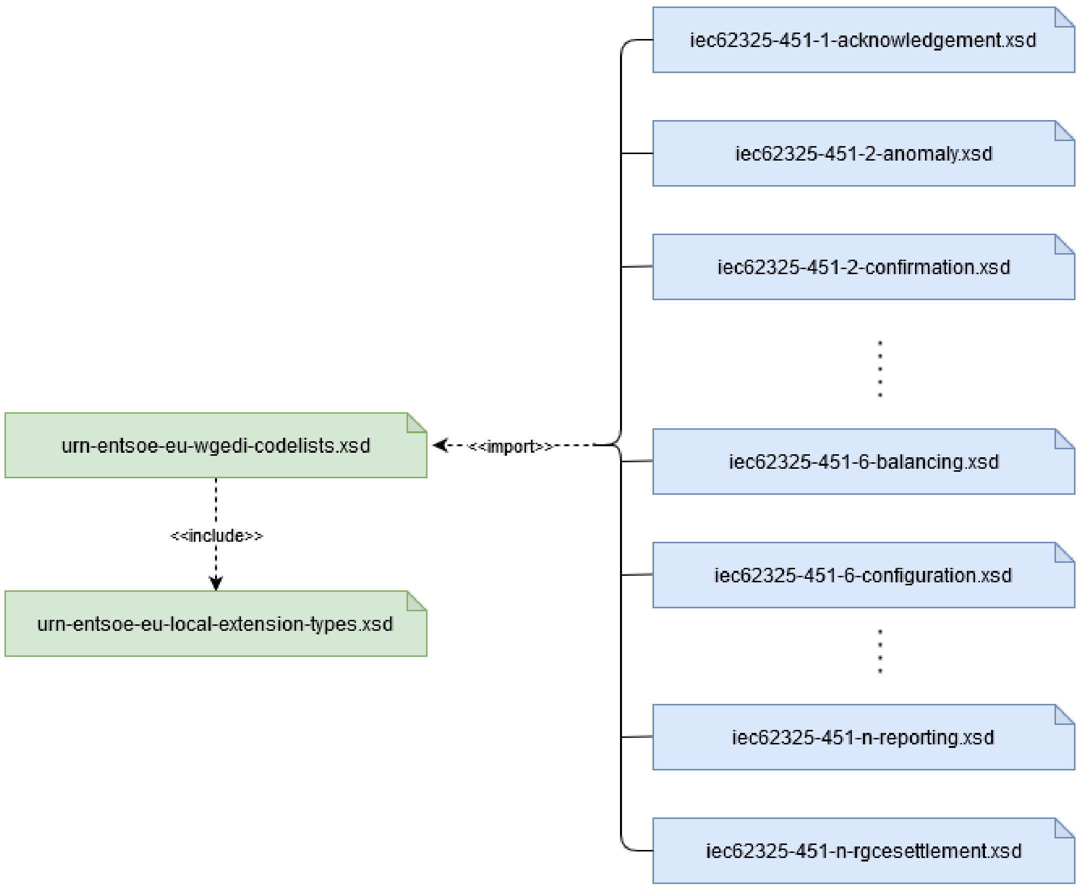

iec62325-451-1-acknowledgement.xsd
iec62325-451-2-anomaly.xsd
iec62325-451-2-confirmation.xsd
urn-entsoe-eu-wgedi-codelists.xsd
<<import>>
iec62325-451-6-balancing.xsd
<<include>>
iec62325-451-6-configuration.xsd
urn-entsoe-eu-local-extension-types.xsd
iec62325-451-n-reporting.xsd
iec62325-451-n-rgcesettlement.xsd

Figure 6 : IEC 62325-451 File structure XML Schema Definitions

Depicted on the left side, there are two files containing generic definitions that are imported by the schemas on the right side. These definition files include enumerations that, for example, list options for data types. On the right side of Figure 6, an excerpt of the XML schemas is listed, with each XSD file defining an energy market document.

To verify whether an energy market data entity in XML format complies with the CIM standard, the entity can be validated against the corresponding XSD file. To enable validation of entities in JSON format as well, the XML schemas were mapped to JSON schemas. For this purpose, a Python script was written, which takes the XSD files from the IEC 62325-451 as input and generates a corresponding JSON schema for each XSD file. Initially, the script extracts the definitions from the files "urn-entsoe-eu-local-extension-types.xsd" and "urn-entsoe-eu-wgedi-codelists.xsd" (Figure 6, left) and caches the definitions for further use. Afterwards, the schemas for the distinct energy market documents are processed in a loop and a JSON schema file is written for each XSD. Since

Copyright 2023 OneNet

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

XSD and JSON schema are different schema languages with distinct syntax and features, a direct one-to-one mapping was not feasible, but an approach was devised. The mapping included the following steps:

- Mapping XSD data types to their equivalent in JSON schemas, for instance, xs:string to JSON schema “type”: “string”.
- Translating XSD constraints to their JSON schema counterparts, such as converting minOccurs="1" in XSD to "minItems": 1 in JSON schema.
- Converting complex types in XSD into a nested structure in JSON using JSON schema's properties.
- Mapping `xsd:enumeration` to JSON schema`s `enum` type.

The resulting JSON schemas were validated using an online validator as well as the Python package jsonschema.

3.2 ETSI Context Information management and NGSI-LD

The OneNet information model as a NGSI-LD information model uses both the IDS and FIWARE NGSI-LD standard models for implementing the OneNet Decentralized middleware and the OneNet Connector. The usage of IDS Connector and FIWARE Context Broker was identified as the best solution to be adopted for ensuring a high level of standardization, interoperability, scalability and reuse of OneNet solution. Therefore, this chapter summarises the definition of the NGSI-LD model according to ETSI specifications, based on D5.5 “Report on Technical specifications for data models/platform agnostic middleware” [4].

NGSI-LD [5] is an information model and API for publishing, querying and subscribing to context information. It is meant to facilitate the open exchange and sharing of structured information between different stakeholders. It is used across application domains such as Smart Cities, Smart Industry, Smart Agriculture, and more generally for the Internet of Things, Cyber-Physical Systems, Systems of systems and Digital Twins. NGSI-LD has been standardized by ETSI [11] (European Telecommunications Standardization Institute) through the Context Information Management Industry Specification Group, following a request from the European Commission. Its take up and further development are spelled out in the EU's "Rolling plan for ICT standardization" [31]. NGSI-LD builds upon a decades-old corpus of research in context management frameworks and context modelling. The acronym NGSI stands for "Next Generation Service Interfaces", a suite of specifications originally issued by the OMA [12] which included Context Interfaces. These were taken up and evolved as NGSIv2 by the European Future Internet Public-Private-Partnership (PPP), which spawned the FIWARE [13] open-source community.

The NGSI-LD information model represents Context Information as entities that have properties and relationships to other entities. It is derived from property graphs, with semantics formally defined based on RDF and the semantic web framework. It can be serialized using JSON-LD. Every entity and relationship are given a

Copyright 2023 OneNet

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 957739

unique IRI (International Resource Identifier) reference as identifier, making the corresponding data exportable as Linked data datasets. The -LD suffix denotes this affiliation to the Linked Data universe.

The core concepts are:

- A property graph is a directed multigraph, made up of nodes (vertices) connected by directed links, where nodes and arcs both may have multiple optional attached properties (i.e. attributes).
- Properties (like attributes in object models) have the form of arbitrary key-value pairs. Keys are character strings and values are arbitrary data types. By contrast to RDF graphs, properties are not arcs of the graph.
- Relationships are arcs (directed edges) of the graph, which always have an identifier, a start node and an end node [4].

3.3 FIWARE Smart Data Models

The FIWARE Foundation together with TMForum, and IUDX are leading the Smart Data Models initiative [14] for supporting the adoption of reference architecture and compatible data models for multiple sectors including, Smart Cities, Smart Agrifood, Smart Environment, Smart Sensoring, Smart Energy, Smart Water and other domains. The data models are intended to be used wherever you want, but specifically, they are designed to be compliant to FIWARE NGSI V2 and NGSI-LD (see chapter 3.2).

A smart data model includes three elements [15]:

- **the schema**, or technical representation of the model defining the technical data types and structure;
- **the specification** of a written document for human readers;
- **the examples** of the payloads for NGSIv2 and NGSI-LD versions.

As described in the D5.4 [3], the FIWARE Smart Data Models considered for the FIWARE integration in the OneNet systems are those related to the Smart Energy Domain.

In the analysis phase, conducted for the definition of the requirements and the implementation of the FIWARE Data App and Data Integration and Homogenization sub-layer, 5 different subjects have been identified:

- Battery
- Energy
- EnergyCIM
- GreenEnergy
- Weather

For completeness, a new subject, which includes two smart data models, relating to energy consumption has been added in the meantime.

Copyright 2023 OneNet

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

The new smart data models are:

- **ConsumptionCost**. Information of energy consumed and its cost by consumption point.
- **ConsumptionPoint**. Information on a given consumption point.

Addition information about all the smart data models can be found in the Smart Data Models initiative official GitHub repository [16].

3.4 COSMAG Specifications

Comprehensive Architecture for Smart Grid (COSMAG) refers to the analysis and collection of specifications, which can define the possible process of data exchange between energy stakeholders. The definition of COSMAG is based on some fundamental requirements:

- The set of interactions are defined to support the implementation of the vision of the European Commission as from the Clean Energy for all Europeans Package [17].
- The architecture is built to offer open interfaces, i.e., data interaction points that can be used for future expansions and novel use cases.
- COSMAG investigate and exploit existing standard, starting from results of previous projects or standardization activities.
- COSMAG support the interactions of multiple stakeholders.

Below a set of specifications and recommendations supported by COSMAG initiative [18]:

- Many standards are available and data exchanges are already formalized. Nevertheless, emerging changes in the electricity market structure may bring some significant element of novelty (e.g. local markets).
- Most of the open points are around the customer/prosumers level. In the future, it is expected that a large amount of data will be related to the customer level. In this respect, data platforms able to aggregate data are an important part of the picture. Those data will play a key role in every element of the energy system.
- An important element to be considered is the emerging role of cross-sector applications. In this respect new standards as emerged from the recent work of ETSI (see Chapter 3.2).
- It is important that data platforms will be based on open standards to support open competition. In this respect solutions such as FIWARE can be considered as the right approach.
- Data models are also a critical aspect. In this respect, SAREF extended to cover the whole energy value chain is a very valuable candidate.

COSMAG recommendations and guidelines were analysed during the design of the OneNet Reference Architecture and taken into consideration for the gap analysis. Additional details can be found in D5.2 [2].

3.5 SAREF Ontology

It is important to remark that deliverable 5.6 [32] started the matching of business objects with existing data profiles, which provided not only the semantic definition regarding data with respect to IEC profiles but also insights and recommendations that can help to enhance IEC standard profiles in some cases.

Data exchange in OneNet relies on well-established defined standard data profiles that can be seen as a set of vocabularies that can be formalized as ontologies. For a data exchange to occur between participants, the procedure is to define and agree on the type of service and data in advance by the two actors involved. Semantic interoperability can be accomplished with different approaches; the approach taken in OneNet is to provide an interface to explain the data. As a first step, this interface will make it easy for participants to understand the data that will be received by checking the metadata of the data presented in the service description. This interface, called Semantic Definition, then lets the participants define the type and form of the data that will be accepted to send. This metadata can give many facilities on the data exchange at a semantic level. Furthermore, this can be used later to create a vocabulary database grouping similar already known data according to models or using semantic similarity techniques to accomplish schema matching. Schema matching is a technique to find semantic correspondences between two or more schemas or data structures.

Further semantic solutions can be given using SAREF [33], which is a reference ontology for smart appliances and can be seen as an umbrella to improve the integration of semantic data across vertical domains in IoT. SAREF can be further extended to more concepts, as it was already done with SAREF4ENER [34], which extends to some concepts in the energy domain. In addition, SARGON [35] ontology incorporates more classes, properties, and instances that cover concepts from the building and electrical grid automation domain.

These ontologies can be used to perform semantic annotation, which is a process of adding information to data payloads. The SISEG tool [36] can receive raw data and an ontology, and with these inputs, can annotate, or in other words, tag these raw data to have the attributes' names aligned with some ontologies.

With this set of ontologies and a tool like SISEG, data can be annotated in the premises of the participant and a score can be given to know the number of attributes that were annotated. This process is considered semiautomatic because these technologies are still undergoing changes and research, and at the beginning, there needs to be a mechanism to check and measure the errors that might arise. This means that human intervention will still be needed to verify that the data is correctly annotated, nevertheless, this can be leveraged and used to feed the corrections into the model, and increased accuracy is expected within this phase.

After that, further standardization should be checked on the data values. There needs to be considered the range when it comes to numbers, and the abbreviations and identifiers values that are foreign keys to other tables of information. The alignment of different data models and ontologies is another very important task, as it also is to maintain the versions organized.

Copyright 2023 OneNet

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

4 Data Integration & Homogenization sub-layer

4.1 OneNet Decentralized Middleware and OneNet Connector

The OneNet decentralized middleware is the key component of the OneNet Framework since it provides the necessary functionality for managing and sharing information in a controlled and administrative environment. It is important to remark that this framework does not have access to the data shared by the OneNet participants, nor does it process such data. In this way, data producers can have total ownership and control (sovereignty of their data). The OneNet Decentralized Middleware orchestrates processes such as the identity management, data catalogue (including the administration of OneNet Cross-Platform Services list), data quality process, logging processes and data access policies (including connector's discoverability based on meta-data information).

The OneNet middleware includes the following modules:

- Interfaces between involved actors to enable the handling and management of data.
- Semantics layer interface able to support the definition of data sharing requirements and how these will be implemented in the components of the system.
- Data Quality and homogenization module with a capability to "tag" exchanged datasets with specific parameters as metadata.
- The Linked Data (LD) Context Broker responsible for information context management.
- The Clearing house and Usage Control modules responsible for the data exchange process logging and exchange process policies (contracts) between data produces and consumer.

As for the OneNet Connector component, it is a deployment instance of the OneNet Decentralized Middleware and, once deployed and integrated within the platforms of each OneNet participant, in conjunction with the accompanying GUI, it allows for a trusted pan-European data space, defined as OneNet Network of Platforms, to be created. The development of the final version of the OneNet Connector considered all functional and non-functional requirements collected starting from the different use cases of OneNet.

The Figure 7 below represents the OneNet Connector Architecture, which includes, among other things, the components related to the Data Integration and Homogenization sublayer, described in the following chapters:

- FIWARE Data App (and FIWARE Context Broker)
- Data Homogenization Tool
- Clearing House and Usage Control App

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

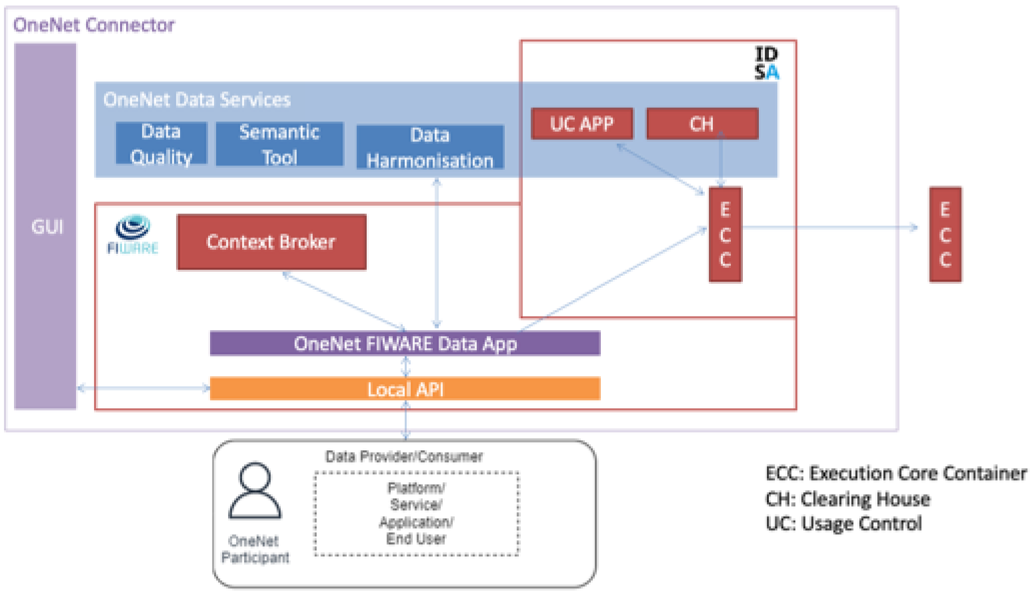


Figure 7: OneNet Connector Architecture

4.2 FIWARE Data App and NGSI-LD Context Broker Integration

The FIWARE Data App and the NGSI-LD Context Broker are two core components of the OneNet connector as can be seen in Figure 8, and it is responsible for implementing the complete end-to-end data exchange process leveraging on the NGSI-LD standard, as well as for offering standardized interfaces for the integration of the external platforms and systems through the OneNet Connector GUI and/or Local API components.

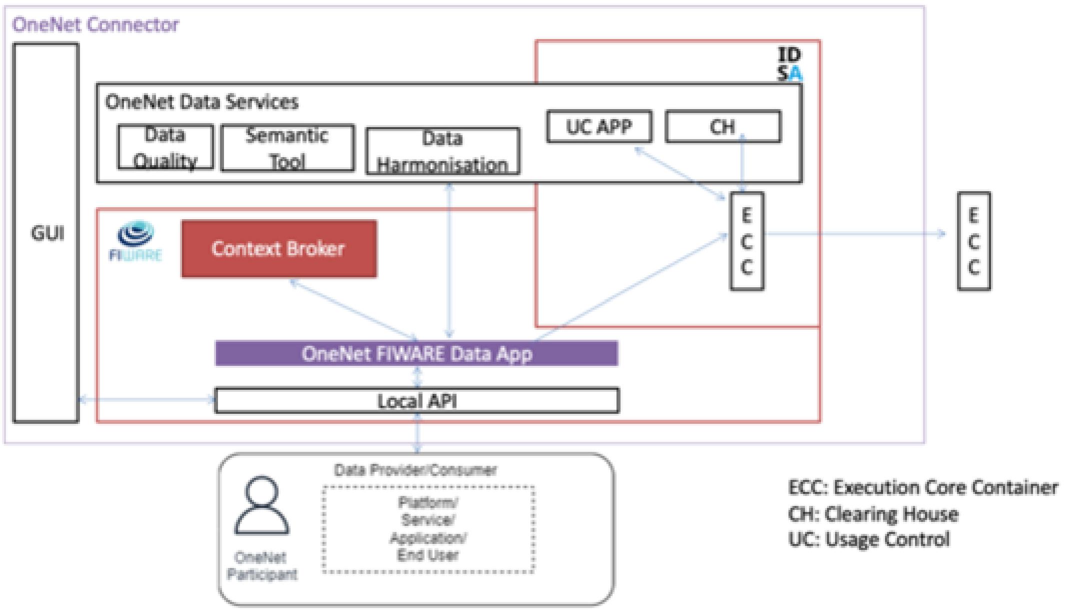


Figure 8: FIWARE Data App and Context Broker in OneNet Connector Architecture

Copyright 2023 OneNet

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

The FIWARE Data App is compatible with the NGSI-LD standard, supporting the integration with the NGSI-LD Orion Context Broker as well as with the Execution Core Container of the OneNet Connector (as described in D6.1 [20]) for implementing the IDS-based Data Exchange process.

In addition, the FIWARE Data App, as part of the Data Integration and Homogenization sublayer is also integrated with the OneNet Data Services, including the Data Homogenization tool, Usage Control App and Clearing House (additional data services were not implemented).

4.2.1 Architecture

The FIWARE Data App consists of a back-end application, based on Spring Boot Framework [19], ready to be integrated with any data provider and/or consumer platform, providing standardized interfaces.

The Figure 9 below shows the architecture of the FIWARE Data App, based on standard Spring Boot one.

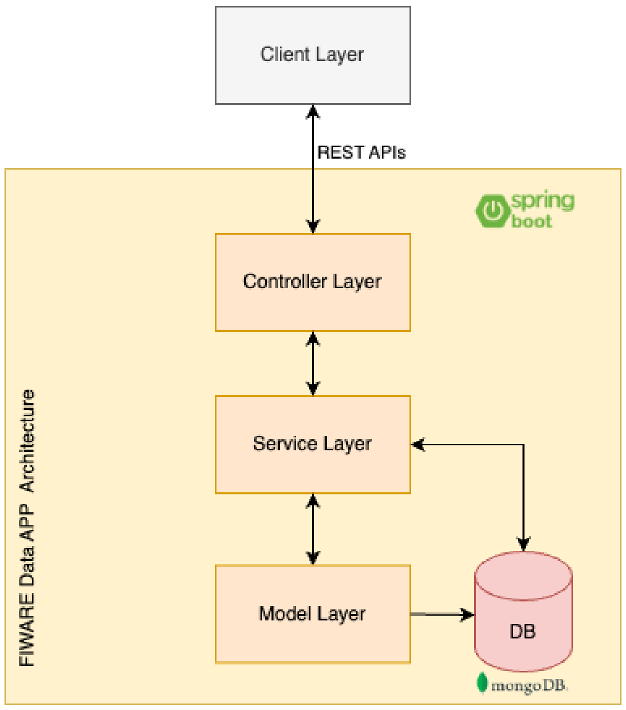


Figure 9: FIWARE Data App - Spring Boot basic architecture

Client Layer

The client layer is not properly part of the FIWARE Data App itself but indicates any possible client capable of interacting with the FIWARE Data App itself. In particular, the client layer should be capable of interacting with the FIWARE Data App using the standardized APIs provided by the Controller Layer. In the OneNet Connector overall architecture the Client is represented by the OneNet Connector GUI and the LocalApi instance (see D6.1 [20] for additional details).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

Controller Layer

The Controller layer is responsible for intercepting all the REST APIs, process them, validate the requests (including authentication and authorization) and interact with the service logic. The APIs interfaces are described and documented in Ch. 4.2.3.

Service Layer

The Service Layer implements the business logic and the main functionalities of the FIWARE Data App. The implemented functionalities are described in detail in Ch. 4.2.2. In addition, the Service Layer interacts with the Model Layer for retrieving the data models and the Database for accessing the stored data.

Model Layer and Database

The Model Layer implements the Data Models and the Data Access Objects, allowing the Service Layer to interact with the database using queries. For the Database instance, a NoSQL databased has been selected (MongoDB) [21].

4.2.2 Functionalities

The main concept underlying the data exchange implemented in OneNet ecosystem is the possibility of registering data sources for the implementation of standard cross-platform services as described in D5.3 [37] and D6.1 [20].

Each OneNet Participant can act as Data Provider (or Data Source), offering its own data sets (namely Data Offering) to any other OneNet Participants which act as a Data Consumer.

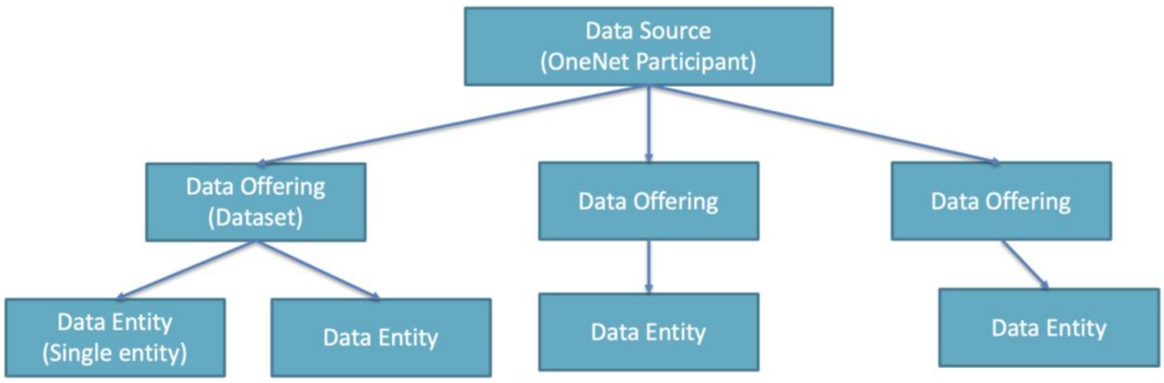


Figure 10: Data Offering representation schema

A Data Offering is based on a specific cross-platform service, mapped to the relative data profile, where possible. It can include one or more data entities, all belonging to the same dataset.

The schema represented in the Figure 10 describes the Data Offering concept.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

The Figures below represent the sequence diagrams for the overall Data Provisioning process (Figure 11) and Data Consumption process (Figure 12).

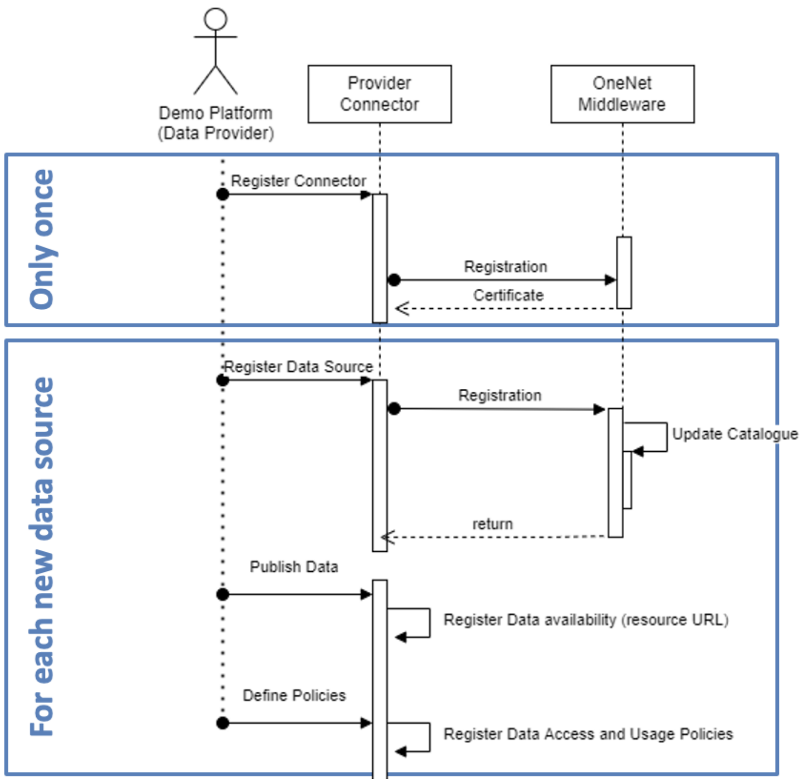


Figure 11: Data Provisioning - Sequence Diagram

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

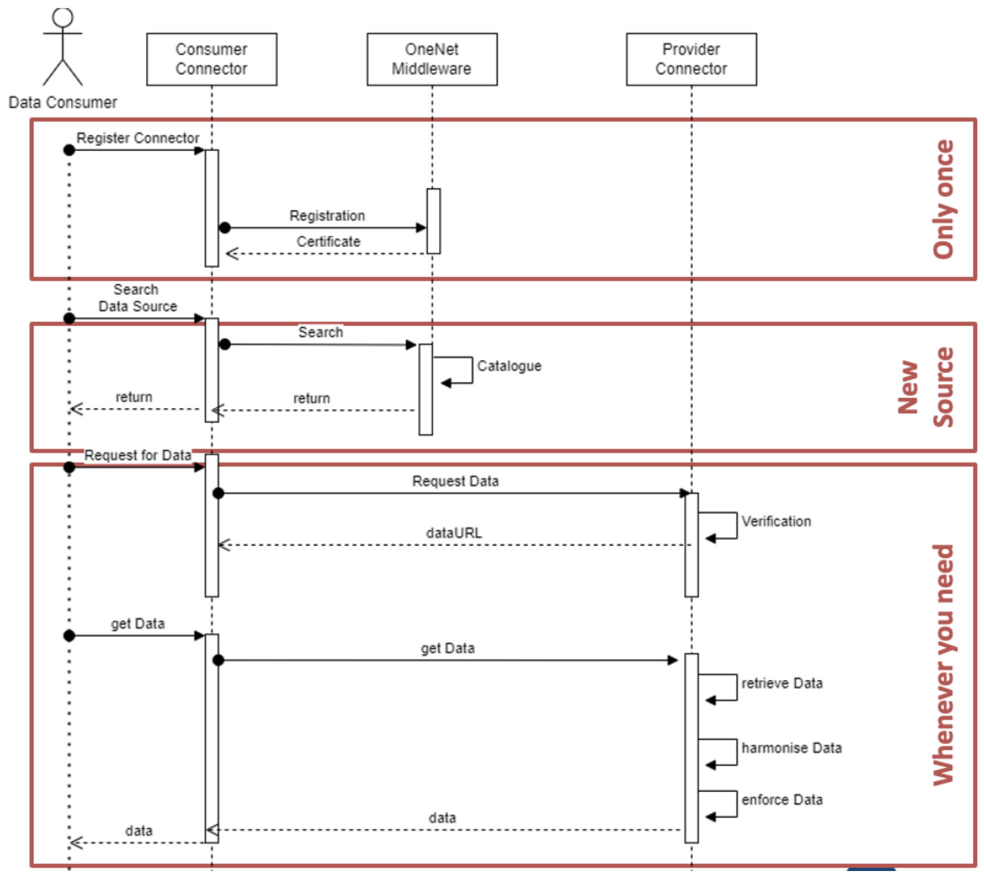


Figure 12: Data Consumption - Sequence Diagram

To implement this schema, the creation of Data Offering in the Data Catalogue was implemented in the OneNet Connector GUI, while the workflow for creation and retrieving of data entities was implemented in the FIWARE Data App and through three main functionalities:

- **Create Entity**: a Data Provider can create new entities (in a specific data offering) within its own environment and makes them available to all the other OneNet Participants.
- **Registration**: a Data Consumer can register to specific data offering (and related data entities) for receiving data in automatic or manual way, after the acceptance of the Data Provider. The Data Consumer can register itself to many data providers and many data offerings.
- **Get Entity**: the Data Consumer, after the registration, can retrieve specific data entity from the data offering.

For each functionality, the paragraphs below report the interaction with the OneNet Connector GUI and a detailed sequence diagram specifically related to the FIWARE Data App.

Copyright 2023 OneNet

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

Create Entity

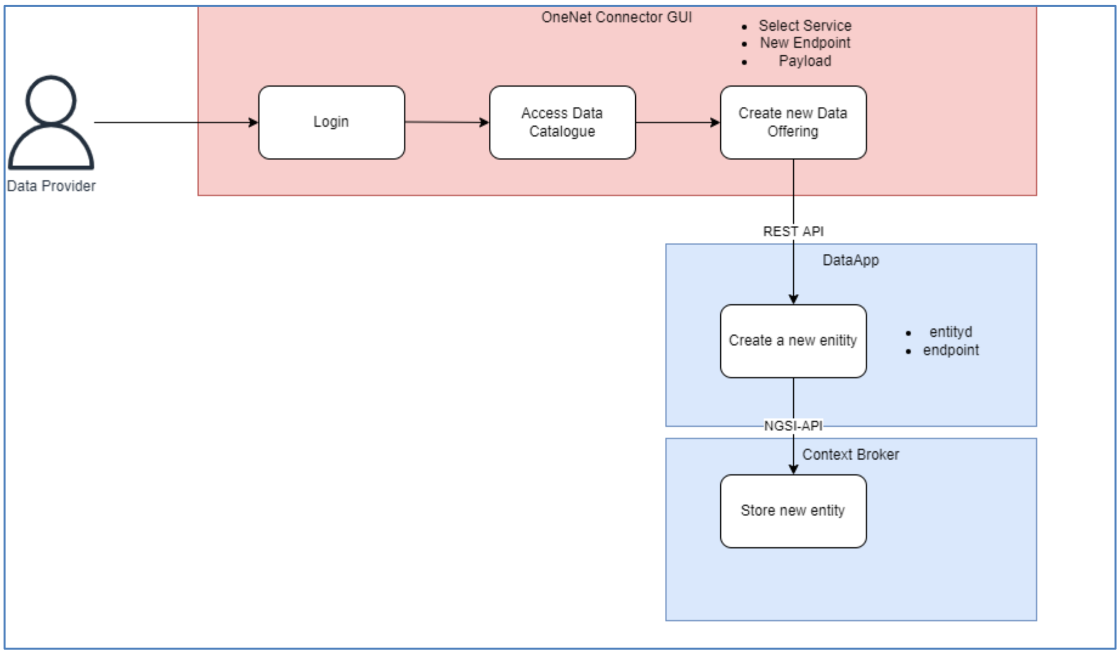


Figure 13: Create Data Offering and new data entities

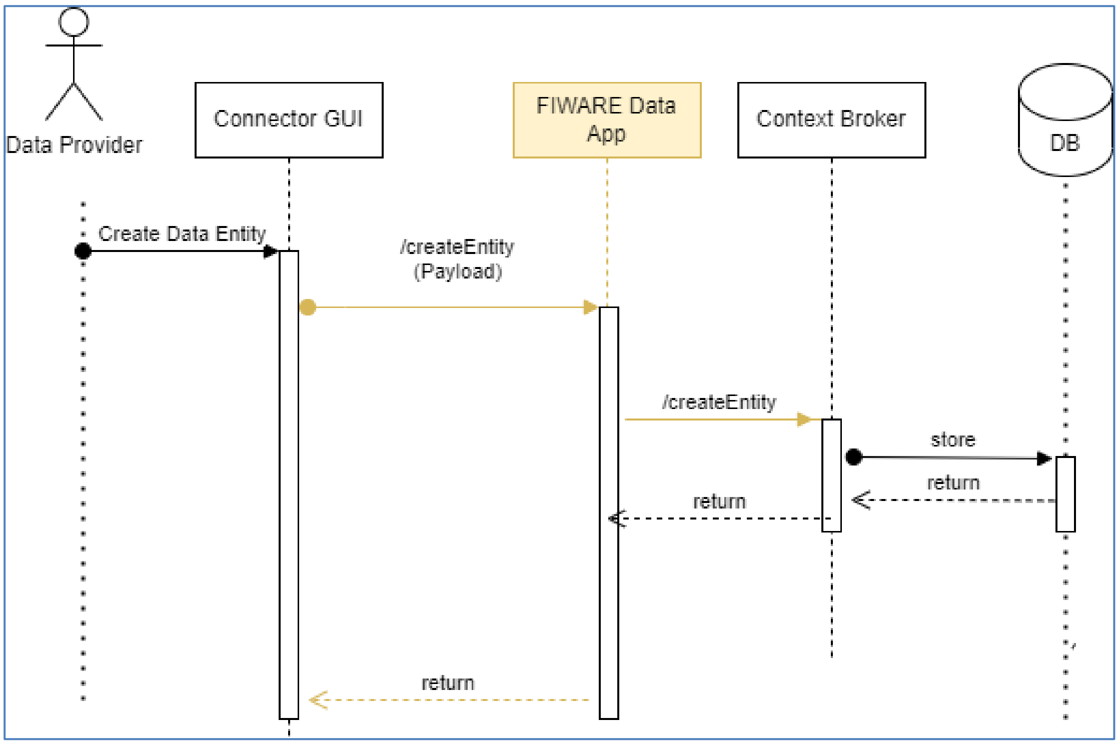


Figure 14: Create Entity - Sequence Diagram

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

Registration

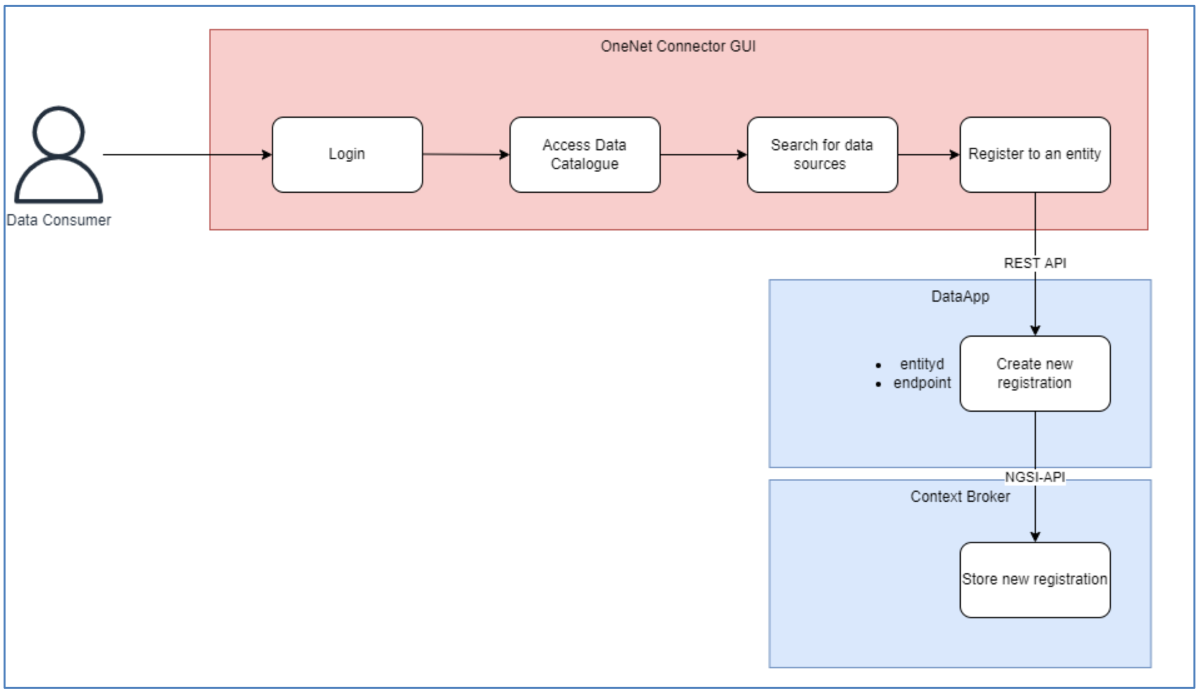
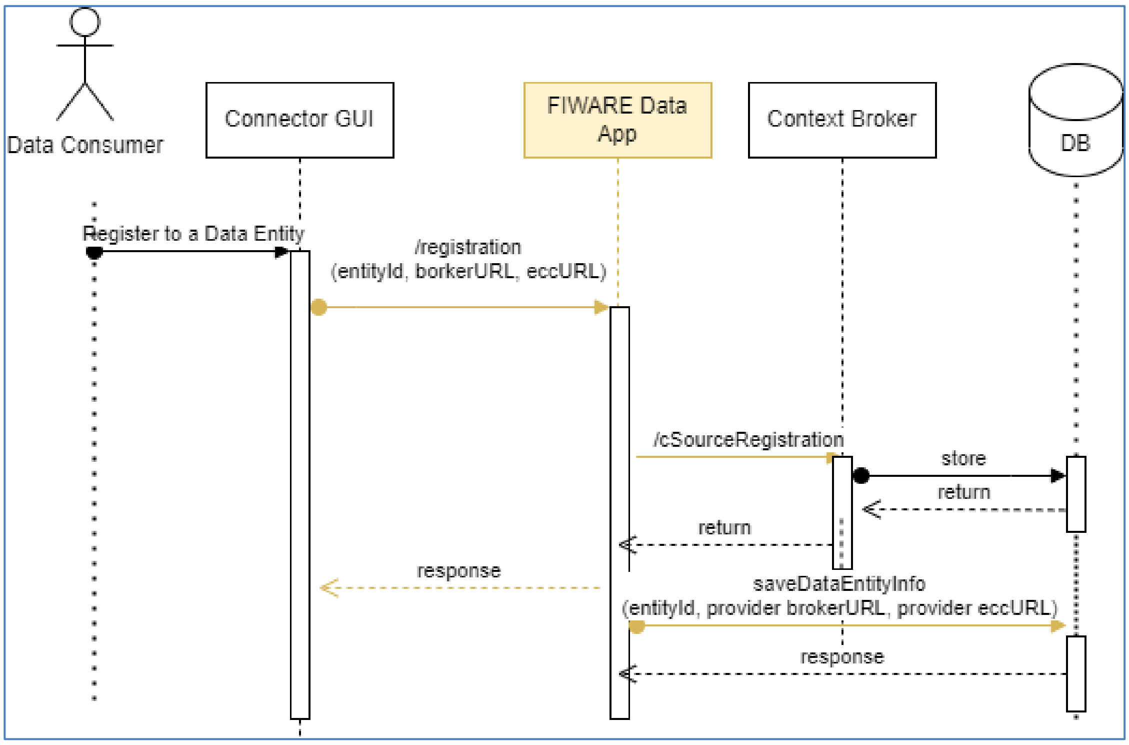


Figure 15: Registration to a specific Data Offering



Data Consumer
Connector GUI
FIWARE Data App
Context Broker
DB
Register to a Data Entity
/registration (entityId, borkerURL, eccURL)
/cSourceRegistration
store
return
return
response
saveDataEntityInfo (entityId, provider brokerURL, provider eccURL)
response

Figure 16: Registration - Sequence Diagram

Copyright 2023 OneNet

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

Get Entity

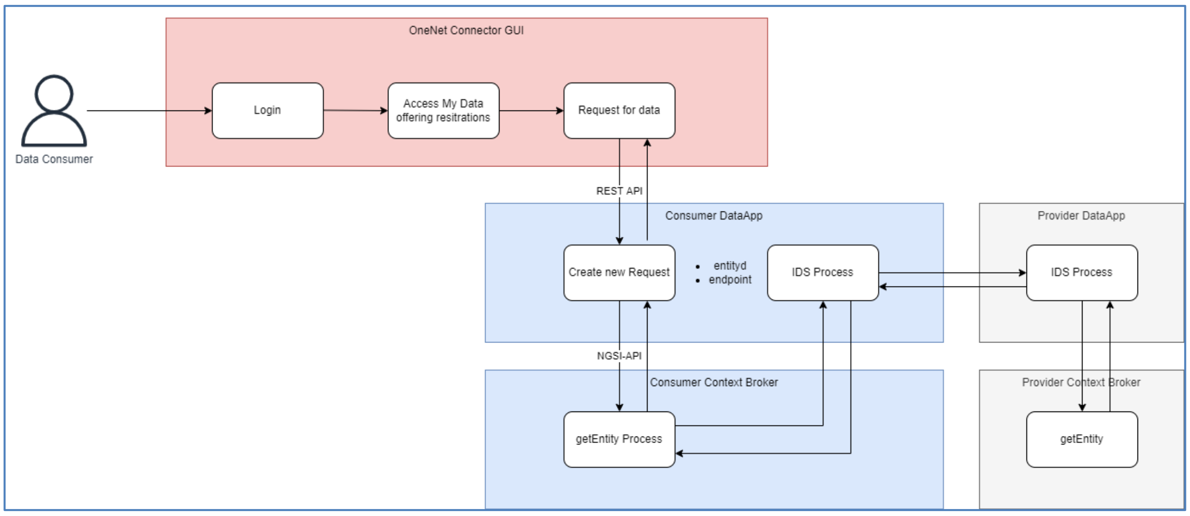
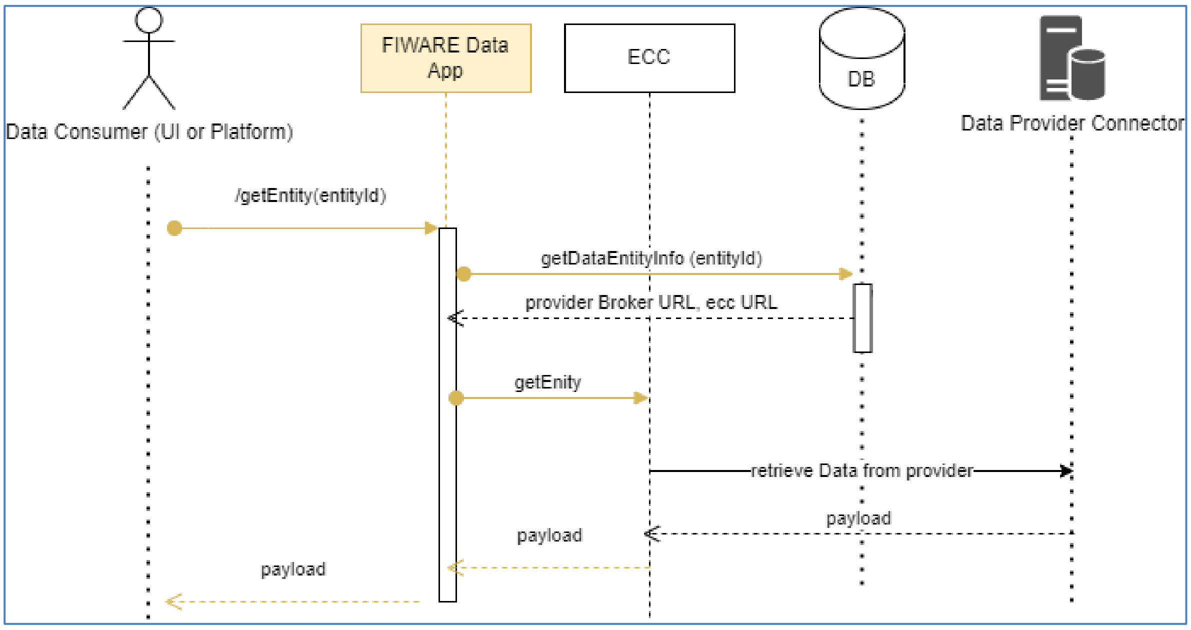


Figure 17: Get Entity from OneNet Connector GUI (or Local API)



Data Consumer (UI or Platform)
FIWARE Data App
ECC
DB
Data Provider Connector
/getEntity(entityId)
getDataEntityInfo (entityId)
provider Broker URL, ecc URL
getEnity
retrieve Data from provider
payload
payload
payload

Figure 18: Get Entity - Sequence Diagram

4.2.3 Interfaces and APIs

As described in the previous chapter, three main functionalities have been implemented in the FIWARE Data App. For each of these features, a specific API has been implemented. All the APIs are available both from the OneNet Connector GUI or Local API and accessible using credentials and certificates used in the Connector itself.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

The Table 3 below reports the schema of the three APIs. In addition, all the endpoints are documented using OpenAPI [22] standards and available in each connector instance at *<fiware-data-app-url:port/swagger-ui.html>* (e.g., https://smart-energy.eng.it:8080/data-app-consumer/swagger-ui.html).

Table 3: FIWARE Data App endpoints

Endpoint	Method	Description	Input	HTTP code	Response
/createentity	POST	Create a new Entity in a FIWARE standardized data model	JSON – Entity Payload	200 OK	-
				400 Bad Request	Error response
/getentity/{entityId}	GET	Return a specific entity, given an Id.	String	200 OK	Entity
				400 Bad Request	Error response
/registration	POST	Register a consumer to a specific data entity service	JSON – { eccUrl, String entityId, String}	200 OK	pid – string (process id, identifier of the registration)
				400 Bad Request	Error response

4.2.4 Packaging and delivery

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

The FIWARE Data App and Context Broker docker containers are released together with the OneNet Connector. Please refer to the OneNet Connector official guidelines [23] for the deployment and installation of the software.

4.3 Data Homogenization tool

The OneNet data homogenization tool CIM2FIWARE has the purpose of integrating IEC 62325-451 entities with the FIWARE NGSI-LD information model. To perform this integration, its core functionalities include the validation of these entities as well as the conversion from XML or JSON formats into NGSI-LD valid format. This means that not only the previously mentioned data models can be the input of this tool, but it can also accept other data models that conform to XML or JSON format.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

4.3.1 Architecture

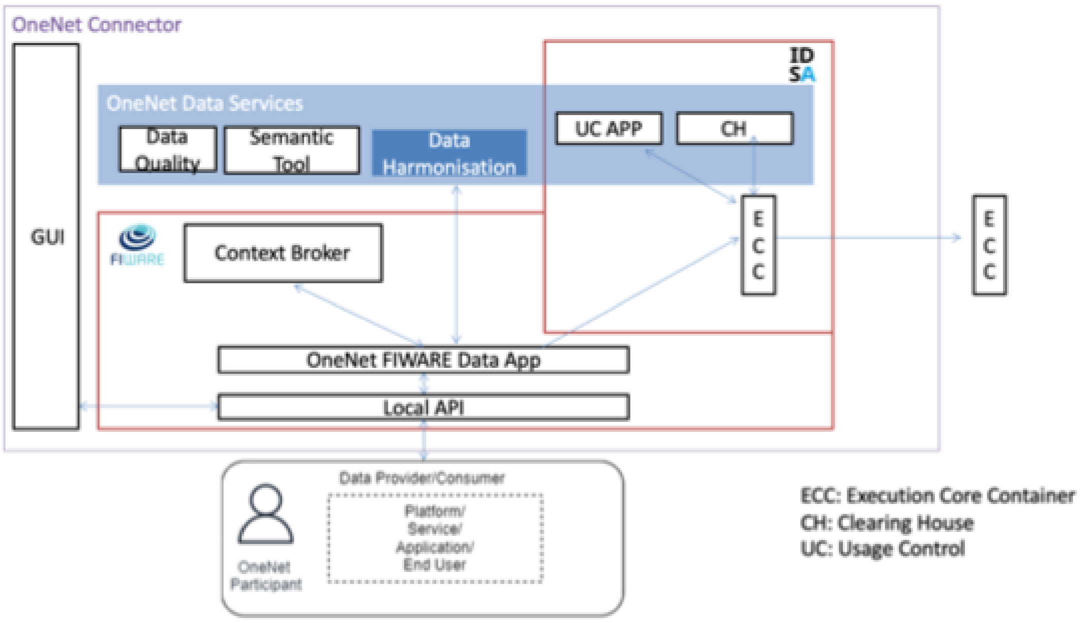


Figure 19 - Data Homogenization tool inside the OneNet Connector

The data homogenization tool is one of the data services that is part of the OneNet connector as can be seen in Figure 19, and it is responsible for the validation and conversion of data into a format that can be accepted by the FIWARE context broker. The architecture of the tool can be seen in Figure 20, and the components are explained below.

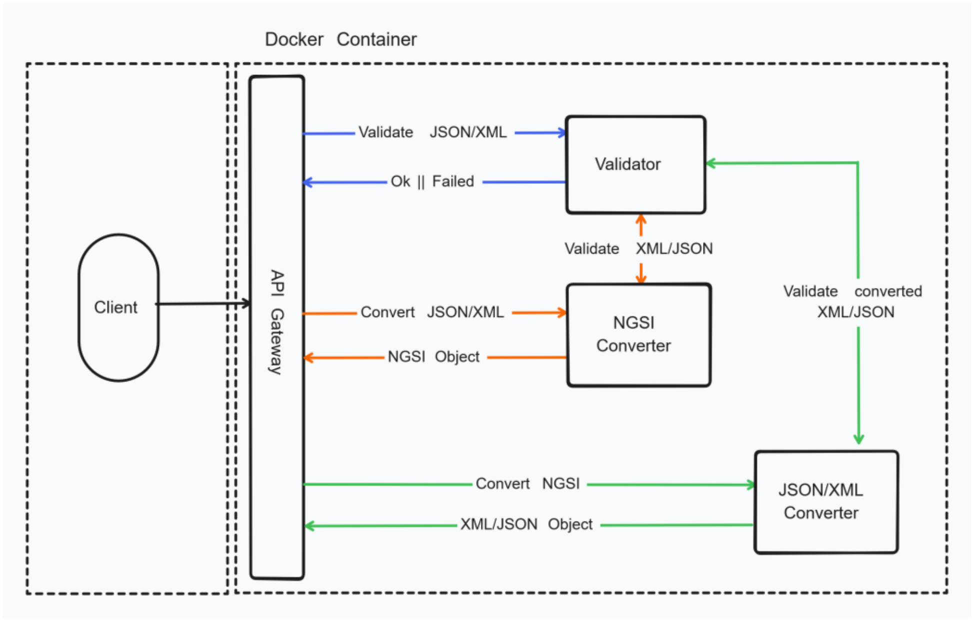


Figure 20 - Data Homogenization Tool Architecture

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

The NGSI-LD information model leverages the advantages provided by JSON-LD and provides support for linked data. The main constructs of NGSI-LD encompass Entity, Property, and Relationship, which enable a comprehensive representation of the data. In alignment with the overarching principle of sector-agnostic data formats, as highlighted in deliverable D5.2, section 3.1.1.3 [2] Reference Architecture Model, the tool has been equipped with support for the universally recognized formats XML and JSON. Furthermore, the software has been designed for the potential inclusion of other data formats, such as CSV, in a similar manner, ensuring future adaptability and extensibility.

4.3.2 Functionalities

The tool encompasses two primary functionalities: the validation and conversion of market data entities. Upon reception of an XML or JSON entity, a validation is initiated to ensure compliance with the IEC 62325-451 standard. Successful validation facilitates subsequent entity conversion to NGSI-LD. In case of validation failure, the tool returns an error response to the user. Error handling will be described below.

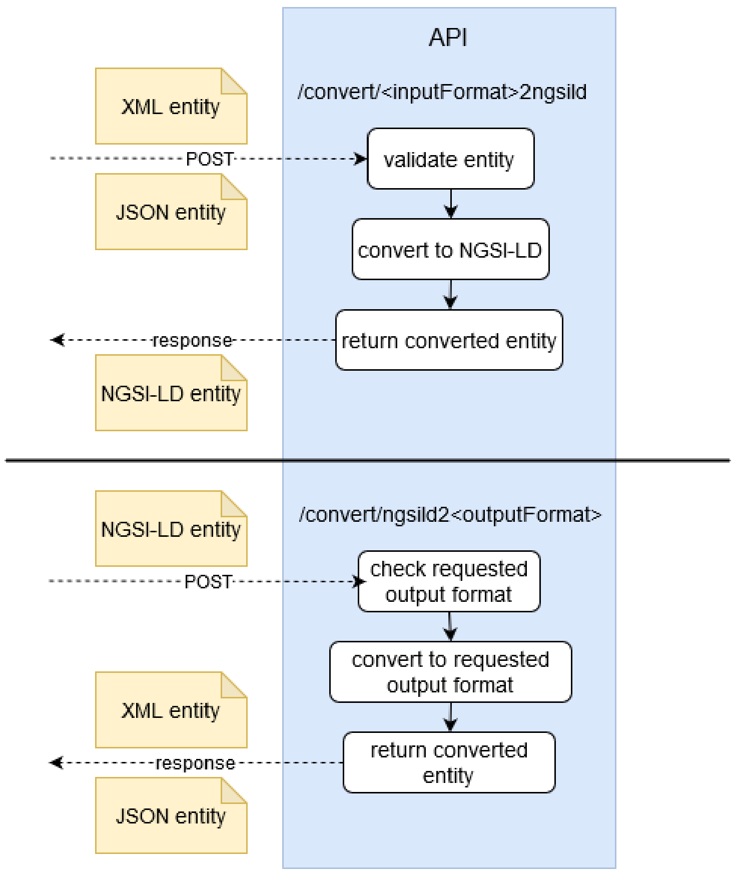


Figure 21 – Data conversion flow

To access the data homogenization tool, an Application Programing Interface (API), which conforms to the design principles of REST, is provided and has been documented in an OpenAPI specification.

Validation

Request data is validated before conversion into NGSI-LD format. Additionally, the API of the Data Homogenization Tool provides an endpoint for validation only. The API user can send a request to the endpoint /validate/xml or /validate/json (see Table 5) to check whether an entity corresponds to the IEC standard. On successful entity validation, the API returns the HTTP status code 200 [OK] to confirm the validity of the request data.

Conversion

Figure 21 illustrates the data conversion flow of the data homogenization tool. An entity (XML or JSON format) is sent to the API with a POST request, it gets validated and converted into NGSI-LD format. After successful conversion, the converted entity is returned, facilitating entity storage in the FIWARE context broker.

The lower half of Figure 21 illustrates reconversion from NGSI-LD format: the POST request includes the entity to convert, the format to convert to is determined by the API endpoint which is addressed in the request (see Table 5 in section 4.3.3). The entity in NGSI-LD format is then converted to the requested format (XML or JSON) and returned to the caller of the API.

Figure 22 gives a more detailed overview of the API's conversion endpoints. An XML entity, which is sent to the API endpoint /convert/xml2ngsild, is first validated with the use of an XML Schema Definition file that corresponds to the market data type of the XML entity. Similarly, when a JSON payload is sent to the endpoint /convert/json2ngsild, the JSON entity is validated with the use of the corresponding JSON schema file for the market document type. In case validation fails, the API returns an error response. In case of successful validation, the committed entity is converted into NGSI-LD format and returned in the HTTP response.

The conversion API also entails endpoints to convert a NGSI-LD entity to the formats XML and JSON. A payload that is directed to the endpoint /convert/ngsild2xml is thus converted to XML. In case the conversion fails, an error response is returned. Otherwise, the successfully converted XML entity is returned to the API user. The /convert/ngsild2json endpoint works in an analogous fashion to convert a payload from NGSI-LD to JSON.

Copyright 2023 OneNet

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

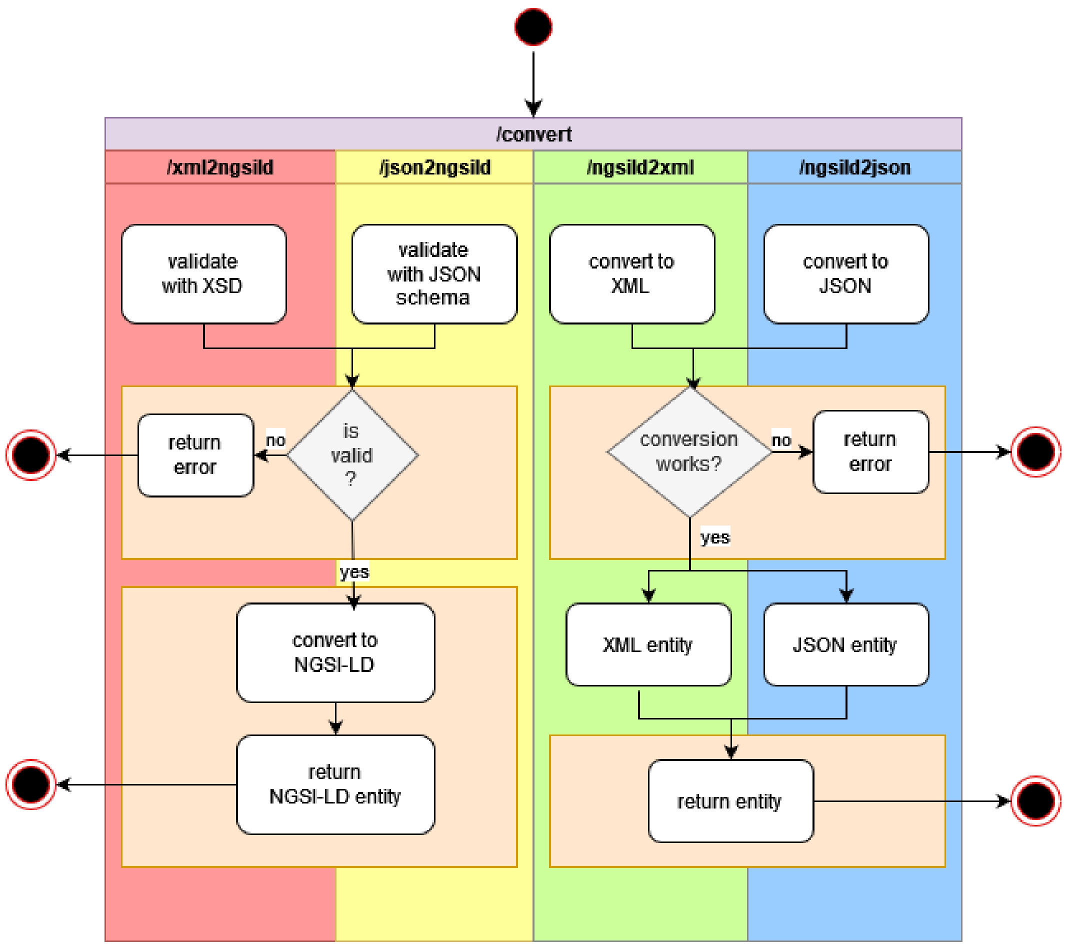


Figure 22 - Conversion endpoints

Error handling

An error response from the API has the following format:

```
    HTTP/1.1 422 Unprocessable Entity

    Content-Type: application/problem+json

    Content-Language: en

    {

     "status": 422,

     "title": "Request data cannot be converted",

     "detail": "Entity conversion not possible because XML payload did
not validate with corresponding XML Schema Definition",

    }
```

Figure 23 - Error response example

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

The HTTP response includes a body with an error response in JSON format. The error messages follow the RFC 7807 standard [24] and contain the members:

- **status** (number): The HTTP status code
- **title** (string): A short, human-readable summary of the problem type
- **detail** (string): A human-readable explanation specific to this occurrence of the problem

Table 4 lists the errors responses returned by the API.

Table 4 - Error responses

status	title	detail
400 Bad Request	Request data invalid	XML input could not be validated with corresponding XML Schema Definition
		JSON input could not be validated with corresponding JSON schema
412 Precondition Failed	Request data does not match content-type header	XML to NGSI-LD conversion endpoint was called with Content-Type header other than "application/xml"
		JSON to NGSI-LD conversion endpoint was called with Content-Type header other than "application/json"
422 Unprocessable Entity	Request data cannot be converted	Entity conversion not possible because XML payload did not validate with corresponding XML Schema Definition
		Entity conversion not possible because JSON payload did not validate with corresponding JSON schema

4.3.3 Interfaces and APIs

In Table 5 can be seen the different endpoints provided for entity validation and conversion plus the corresponding API responses.

Table 5 - Endpoints of data homogenization tool

Endpoint	Method	Description	Input	HTTP code	Response
/validate/xml	POST	Validator for XML documents against XSD schemas. Receive the document and try to validate it against its proper schema version.	XML	200 OK	-
				400 Bad Request	Error response
	GET	Get a list of available XSD schema files and their version		200 OK	List of available XSD schemas with the versions.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

/validate/json	POST	Validator for JSON documents against JSON schema. Receive the document and try to validate it against its proper schema version.	JSON	200 OK	-
				400 Bad Request	Error response
	GET	Get a list of available JSON schema files and their version		200 OK	List of available XSD schemas with the versions.
/convert/xml2ngsild	POST	Convert an XML entity to normalized NGSI-LD format.	XML	201 Created	Entity in NGSI-LD format
				412 Precondition Failed	Error response
				422 Unprocessable Entity	Error response
/convert/ngsild2xml	POST	Convert the JSON-LD entity into XML	NGSI-LD	201 Created	Entity in XML format
/convert/json2ngsild	POST	Convert a JSON entity to normalized NGSI-LD format.	JSON	201 Created	Entity in NGSI-LD format
				412 Precondition Failed	Error response
				422 Unprocessable Entity	Error response
/convert/ngsild2json	POST	Convert the JSON-LD entity into JSON	NGSI-LD	201 Created	Entity in JSON format

4.3.4 Entity conversion example

To illustrate the conversion output of the Data Homogenization Tool, Figure 24 shows a minimal example entity of type Activation_MarketDocument, encompassing the involved market participants (sender, receiver), and a TimeSeries object, which represents time-dependent data that is related to the market document.

Copyright 2023 OneNet

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

```xml
<?xml version="1.0" encoding="UTF-8"
standalone="yes"?>
<Activation_MarketDocument
xmlns="urn:iec62325.351:tc57wg16:451-7
       :activationdocument:6:3">
  <mRID>POGXELEKTROLJ058W-1_20230305-39</mRID>
  <revisionNumber>2</revisionNumber>
  <type>A40</type>
  <process.processType>
    A47
  </process.processType>
  <sender_MarketParticipant.mRID codingScheme="A01">
    28XSODO---LJS
  </sender_MarketParticipant.mRID>
  <sender_MarketParticipant.marketRole.type>
    A04
  </sender_MarketParticipant.marketRole.type>
  <receiver_MarketParticipant.mRID
  codingScheme="A01">
    28XELEKTROLJ058W
  </receiver_MarketParticipant.mRID>
  <receiver_MarketParticipant.marketRole.type>
    A27
  </receiver_MarketParticipant.marketRole.type>
  <createdDateTime>
    2023-03-05T18:15:00Z
  </createdDateTime>
  <activation_Time_Period.timeInterval>
    <start>2023-03-05T18:30Z</start>
    <end>2023-03-05T19:30Z</end>
  </activation_Time_Period.timeInterval>
  <TimeSeries>
    <mRID>POGXELEKTROLJ058W-1</mRID>
    <resourceProvider_MarketParticipant.mRID
    codingScheme="A01">
      28XELEKTROLJ058W
    </resourceProvider_MarketParticipant.mRID>
    <businessType>A97</businessType>
    <acquiring_Domain.mRID codingScheme="A01">
      10YSI-ELES—O
    </acquiring_Domain.mRID>
    <connecting_Domain.mRID codingScheme="A01">
      10YSI-ELES—O
    </connecting_Domain.mRID>
    <measurement_Unit.name>
      MAW
    </measurement_Unit.name>
    <flowDirection.direction>
      A01
    </flowDirection.direction>
    <marketObjectStatus.status>
      A10
    </marketObjectStatus.status>
    <Period>
      <timeInterval>
        <start>2023-03-05T18:30Z</start>
        <end>2023-03-05T19:30Z</end>
      </timeInterval>
      <resolution>PT60M</resolution>
      <Point>
        <position>1</position>
        <quantity>0.02</quantity>
      </Point>
    </Period>
  </TimeSeries>
</Activation_MarketDocument>
```

Figure 24: Entity of type Activation_MarketDocument (XML format)

The entity is given in XML format with nested elements. Converted into NGSI-LD format, the entity adopts the format in Figure 25. The NGSI-LD document includes a @context, id, type and all the elements from the original XML document as Properties. The context provides defining terms for understanding the data in the document. The 'id' element represents a globally unique identifier of the entity. It is a Uniform Resource Identifier (URI) that serves as a globally unique identifier. The 'id' is a composition of the market entities' type, revision number and mRID. The 'type' element in the converted NGSI-LD document specifies the entity type.

All elements from the XML document are stored as Properties in the converted NGSI-LD format.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

```
{
  "@context": [
    "https://schema.lab.fiware.org/ld/context",
    "https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-
context.jsonld"
  ],
  "id":
"Activation_MarketDocument:2:POGXELEKTROLJ058W-
1_20230305-3977578",
  "type": "Activation_MarketDocument",
  "mRID": {
    "type": "Property",
    "value": "POGXELEKTROLJ058W-1_20230305-39"
  },
  "revisionNumber": {
    "type": "Property",
    "value": "2"
  },
  "process.processType": {
    "type": "Property",
    "value": "A47"
  },
  "sender_MarketParticipant.mRID": {
    "type": "Property",
    "value": {
      "codingScheme": "A01",
      "mRID": "28XSODO---LJS"
    }
  },
  "sender_MarketParticipant.marketRole.type": {
    "type": "Property",
    "value": "A04"
  },
  "receiver_MarketParticipant.mRID": {
    "type": "Property",
    "value": {
      "codingScheme": "A01",
      "mRID": "28XELEKTROLJ058W"
    }
  },
  "receiver_MarketParticipant.marketRole.type": {
    "type": "Property",
    "value": "A27"
  },
  "createdAt": "2023-03-05T18:15:00Z",
  "activation_Time_Period.timeInterval": {
    "type": "Property",
    "value": {
      "start": "2023-03-05T18:30Z",
      "end": "2023-03-05T19:30Z"
    }
  },
  "TimeSeries": {
    "type": "Property",
    "value": [
      {
        "mRID": "POGXELEKTROLJ058W-1",
        "resourceProvider_MarketParticipant.mRID": {
          "codingScheme": "A01",
          "value": "28XELEKTROLJ058W"
        },
        "businessType": "A97",
        "acquiring_Domain.mRID": {
          "codingScheme": "A01",
          "value": "10YSI-ELES-----O"
        },
        "connecting_Domain.mRID": {
          "codingScheme": "A01",
          "value": "10YSI-ELES--O"
        },
        "measurement_Unit.name": "MAW",
        "flowDirection.direction": "A01",
        "marketObjectStatus.status": "A10",
        "Period": [
          {
            "timeInterval": {
              "start": "2023-03-05T18:30Z",
              "end": "2023-03-05T19:30Z"
            },
            "resolution": "PT60M",
            "Point": [
              {
                "position": "1",
                "quantity": "0.02"
              }
            ]
          }
        ]
      }
    ]
  }
}
```

Figure 25: Activation market document in NGSI-LD format

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

4.3.5 Constraints

For storing market document entities in the FIWARE Orion Context Broker, two distinct approaches were considered:

a) Storing all data pertaining to the market document entity within a single NGSI-LD entity, with each data point represented as a NGSI-LD Property, as shown in the example above (Figure 25);

b) Dividing the market document into its constituent components and establishing relationships between them.

The first approach offers a less complex solution to map the CIM standard IEC 62325-451, reducing the likelihood of errors. The second approach provides a closer alignment between the NGSI-LD data and the IEC standard. However, it introduces additional complexities during the processes of adding, updating, and retrieving entity data from the FIWARE context broker because several NGSI-LD entities need to get created and linked together.

Considering the lack of interlinking among distinct market document entities, approach a) was chosen for the implementation of the data homogenization tool, as it strikes a balance between adherence to standards and practicality within the given context.

To provide more clarity regarding the distinct approaches, the document type Activation_MarketDocument is reviewed more closely. The Activation_MarketDocument contextual model, depicted in Figure 26, has been extracted from the ENTSO-E documentation "Activation document – UML model and schema (Version 1.2)." It shows the UML representation of the document type with its components and their connections. The Activation_MarketDocument comprises several basic properties (mRID, revisionNumber, type, createdDateTime) and links to several subparts, some of which are optional (e.g., Process, Domain, TimeSeries) while others are required (Time_Period, Sender_MarketParticipant, Receiver_MarketParticipant).

Each of these constituent elements could be stored as separate entities and linked together, given that in the CIM standard they are treated as separate entities rather than direct properties of the Activation_MarketDocument. When looking at the depicted boxes in Figure 6, this approach would entail the creation and linkage of up to 17 distinct entity types. Since TimeSeries and its corresponding parts (Series_Period, Point, Reason) can be featured multiple times, an Activation_MarketDocument entity converted into NGSI-LD might be split up into a large set of linked NGSI-LD entities.

However, it is conceivable to reduce the number of distinct entities, such that only the more complex elements, such as TimeSeries, are stored as distinct entities, while the simpler elements (e.g., Process with a single string property) are not considered distinct NGSI-LD entities but instead stored as Properties within the entity assigned the type 'Activation_MarketDocument,' as illustrated in Figure 25.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

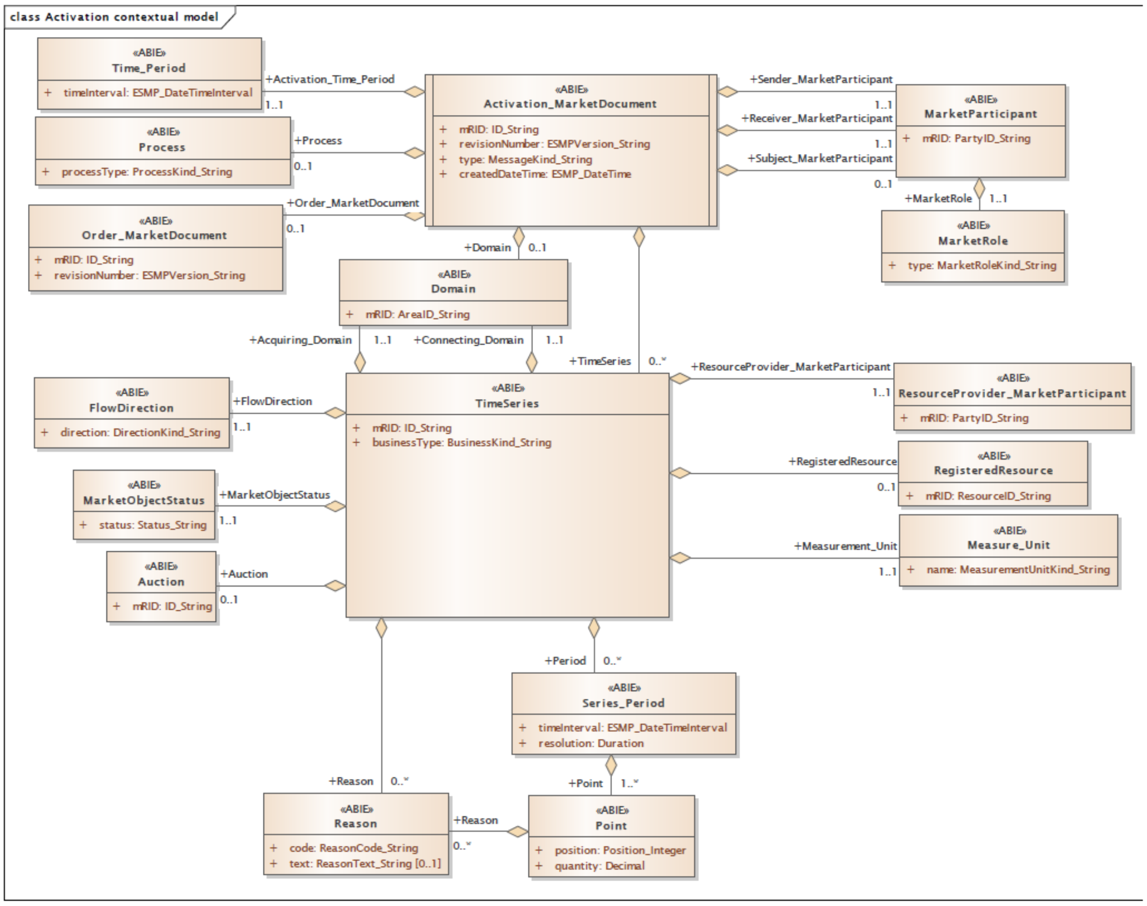

class Activation contextual model
«ABIE» Time_Period
+ timeInterval: ESMP_DateTimeInterval
+Activation_Time_Period 1..1
«ABIE» Process
+ processType: ProcessKind_String
+Process 0..1
«ABIE» Order_MarketDocument
+ mRID: ID_String
+ revisionNumber: ESMPVersion_String
+Order_MarketDocument 0..1
«ABIE» Activation_MarketDocument
+ mRID: ID_String
+ revisionNumber: ESMPVersion_String
+ type: MessageKind_String
+ createdDateTime: ESMP_DateTime
+Sender_MarketParticipant 1..1
+Receiver_MarketParticipant 1..1
+Subject_MarketParticipant 0..1
«ABIE» MarketParticipant
+ mRID: PartyID_String
+MarketRole 1..1
«ABIE» MarketRole
+ type: MarketRoleKind_String
+Domain 0..1
«ABIE» Domain
+ mRID: AreaID_String
+Acquiring_Domain 1..1
+Connecting_Domain 1..1
+TimeSeries 0..*
+ResourceProvider_MarketParticipant 1..1
«ABIE» ResourceProvider_MarketParticipant
+ mRID: PartyID_String
«ABIE» FlowDirection
+ direction: DirectionKind_String
+FlowDirection 1..1
«ABIE» TimeSeries
+ mRID: ID_String
+ businessType: BusinessKind_String
+RegisteredResource 0..1
«ABIE» RegisteredResource
+ mRID: ResourceID_String
«ABIE» MarketObjectStatus
+ status: Status_String
+MarketObjectStatus 1..1
+Measurement_Unit 1..1
«ABIE» Measure_Unit
+ name: MeasurementUnitKind_String
«ABIE» Auction
+ mRID: ID_String
+Auction 0..1
+Period 0..*
«ABIE» Series_Period
+ timeInterval: ESMP_DateTimeInterval
+ resolution: Duration
+Reason 0..*
+Point 1..*
«ABIE» Reason
+ code: ReasonCode_String
+ text: ReasonText_String [0..1]
+Reason 0..*
«ABIE» Point
+ position: Position_Integer
+ quantity: Decimal

Figure 26 - Activation contextual model

This approach would still result in a significantly higher complexity than approach a). In future work, a closer evaluation should be conducted to determine which data parts are reused and how to divide the data optimally.

Another aspect to consider is the differences between market document types. Figure 27 shows the context model of the Balancing market document type, which also includes the TimeSeries component. However, in this instance the properties and the links to other parts differ from those observed in the Activation_MarketDocument contextual model. This raises the question whether these are two different TimeSeries types and, in case the TimeSeries data is stored as a distinct NGSI-LD entities, it may necessitate storage in the FIWARE context broker with different TimeSeries types.

Copyright 2023 OneNet

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

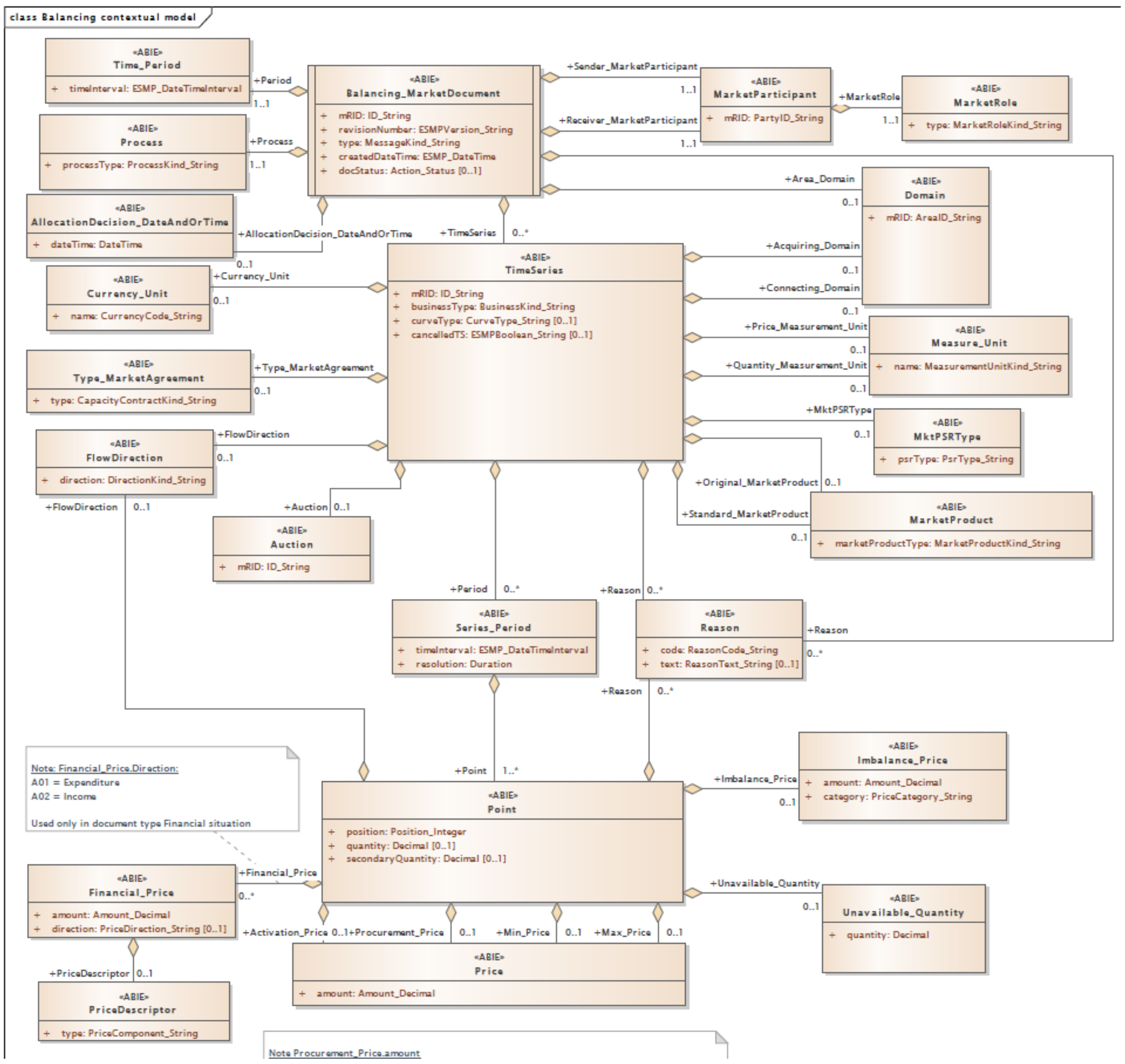

class Balancing contextual model
«ABIE» Time_Period
+ timeInterval: ESMP_DateTimeInterval
«ABIE» Process
+ processType: ProcessKind_String
«ABIE» AllocationDecision_DateAndOrTime
+ dateTime: DateTime
«ABIE» Currency_Unit
+ name: CurrencyCode_String
«ABIE» Type_MarketAgreement
+ type: CapacityContractKind_String
«ABIE» FlowDirection
+ direction: DirectionKind_String
«ABIE» Balancing_MarketDocument
+ mRID: ID_String
+ revisionNumber: ESMPVersion_String
+ type: MessageKind_String
+ createdDateTime: ESMP_DateTime
+ docStatus: Action_Status [0..1]
«ABIE» MarketParticipant
+ mRID: PartyID_String
«ABIE» MarketRole
+ type: MarketRoleKind_String
«ABIE» Domain
+ mRID: AreaID_String
«ABIE» TimeSeries
+ mRID: ID_String
+ businessType: BusinessKind_String
+ curveType: CurveType_String [0..1]
+ cancelledTS: ESMPBoolean_String [0..1]
«ABIE» Measure_Unit
+ name: MeasurementUnitKind_String
«ABIE» MktPSRType
+ psrType: PsrType_String
«ABIE» MarketProduct
+ marketProductType: MarketProductKind_String
«ABIE» Auction
+ mRID: ID_String
«ABIE» Series_Period
+ timeInterval: ESMP_DateTimeInterval
+ resolution: Duration
«ABIE» Reason
+ code: ReasonCode_String
+ text: ReasonText_String [0..1]
«ABIE» Imbalance_Price
+ amount: Amount_Decimal
+ category: PriceCategory_String
«ABIE» Point
+ position: Position_Integer
+ quantity: Decimal [0..1]
+ secondaryQuantity: Decimal [0..1]
«ABIE» Unavailable_Quantity
+ quantity: Decimal
«ABIE» Financial_Price
+ amount: Amount_Decimal
+ direction: PriceDirection_String [0..1]
«ABIE» PriceDescriptor
+ type: PriceComponent_String
«ABIE» Price
+ amount: Amount_Decimal
Note: Financial_Price.Direction:
A01 = Expenditure
A02 = Income
Used only in document type Financial situation
Note Procurement_Price.amount
+Period 1..1
+Process 1..1
+AllocationDecision_DateAndOrTime 0..1
+Currency_Unit 0..1
+Type_MarketAgreement 0..1
+FlowDirection 0..1
+Sender_MarketParticipant 1..1
+Receiver_MarketParticipant 1..1
+MarketRole 1..1
+Area_Domain 0..1
+TimeSeries 0..*
+Acquiring_Domain 0..1
+Connecting_Domain 0..1
+Price_Measurement_Unit 0..1
+Quantity_Measurement_Unit 0..1
+MktPSRType 0..1
+Original_MarketProduct 0..1
+Standard_MarketProduct 0..1
+Auction 0..1
+Period 0..*
+Reason 0..*
+Point 1..*
+Imbalance_Price 0..1
+Unavailable_Quantity 0..1
+Financial_Price 0..*
+Activation_Price 0..1
+Procurement_Price 0..1
+Min_Price 0..1
+Max_Price 0..1
+PriceDescriptor 0..1

Figure 27 - Balancing contextual model

4.3.6 Packaging and delivery

The data homogenization tool is written in the programming language Python and released in a public Gitlab repository [25] under Apache License 2.0. The tool can be run by either cloning the repository and executing the Python code or by pulling a docker image on which the tool is installed. Both installation types and corresponding testing methods are documented in the Gitlab repository. The repository includes the OpenAPI specification documenting the REST API of the data homogenization tool.

4.4 IDS-based data services: Clearing House and Usage Control App

The OneNet Connector offers additional data services based on IDS specifications and guidelines. In fact, the IDSA building blocks for the implementation of the data space includes among other two important components oriented to the data management: the Clearing House and Usage Control.

The **Clearing House** acts as an intermediary that logs all activities performed during data exchange in the IDS ecosystem and it therefore provides clearing and settlement services for all financial and data exchange transactions.

Usage Control App allows to define Usage Policies and Usage Enforcement**.** Each Data Owner & Data Provider can define usage control policies for their data, attached to the outbound data. Therefore, IDS participants can be sure that their data are treated according to their usage policies.

Clearing House and Usage Control App are strictly connected in the OneNet Connector implementation, since following the IDS specifications, the Clearing House can ensure that Data Provider and Consumer meet their contractual obligations, such as:

- The Data Provider sharing data with the Data Consumer according to Usage Contracts and Data Usage Policies defined.
- The Data Consumer using data according to Usage Contracts and Data Usage Policies defined.

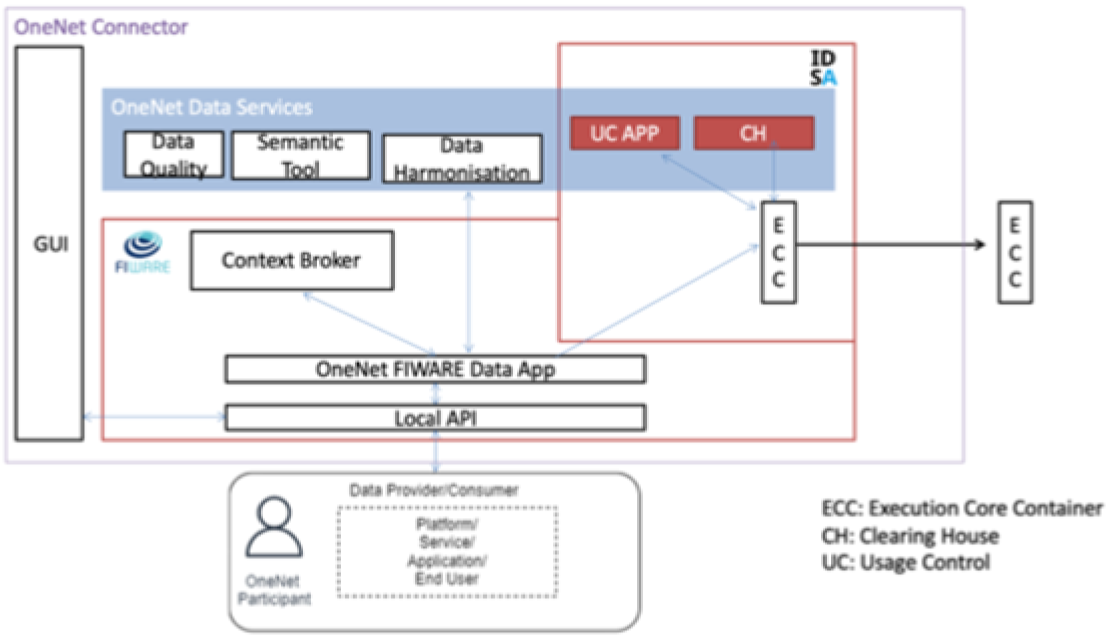


Figure 28: IDS-based OneNet Data Services - Clearing House and Usage Control App

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

For each data exchange transaction, the OneNet Connector can apply the contract negotiation between Data Provider and Consumer (e.g., data access, data usage restrictions, time of validity, etc.) logging all the authorized data transactions, making sure data sovereignty is guaranteed.

Figure 28 represents the Clearing House (CH) and Usage Control App (UC App) in the overall OneNet Connector Architecture.

4.4.1 Clearing House

The OneNet Clearing House is developed starting from the Fraunhofer IDS Clearing House Service.

Data in the Clearing House is stored encrypted and practically immutable. There are multiple ways in which the Clearing House enforces Data Immutability:

- Using the Logging Service there is no way to update an already existing log entry in the database.
- Log entries in the database include a hash value of the previous log entry, chaining together all log entries. Any change to a previous log entry would require rehashing all following log entries.
- The connector logging information in the Clearing House receives a signed receipt from the Clearing House that includes among other things a timestamp and the current chain hash. A single valid receipt in possession of any connector is enough to detect any change to data up to the time indicated in the receipt.

4.4.1.1 Architecture

As described in the official documentation, the core module of the IDS Clearing House Service is the logging-service. This service implements the business logic for all the logging mechanisms that are stored in a MongoDB instance. The logging-service needs an instance of the Trusted Connector to be deployed.

The other main module of IDS Clearing House Service is the document-API that provides standard interfaces for accessing the encrypted logged documents.

The Figure 29 below represents the containerized architecture of the IDS Clearing House Service.

Copyright 2023 OneNet

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

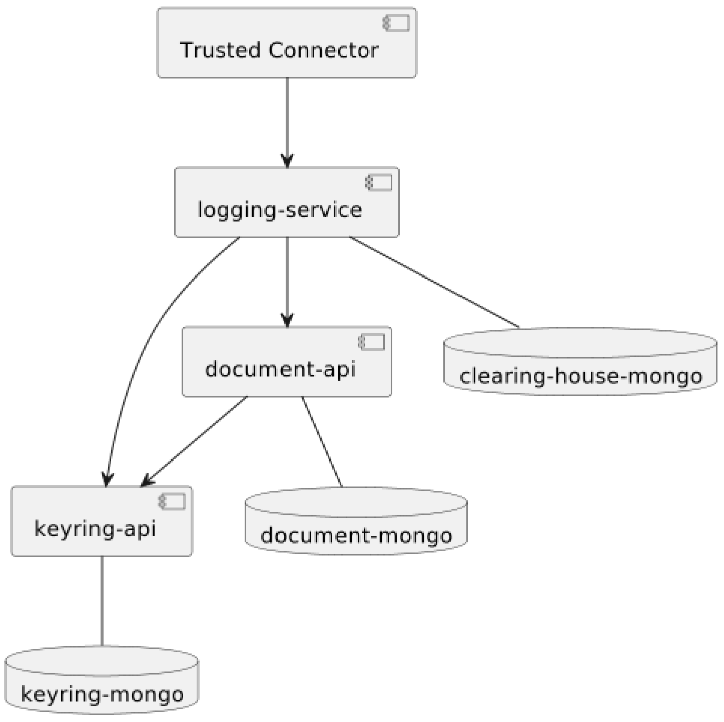


Figure 29: IDS Clearing House container architecture [26]

4.4.1.2 Functionalities

As described in the previous paragraph, the OneNet Clearing House is implemented starting from the Fraunhofer IDS Clearing House and takes advantage of its main functionalities.

- The logging-service can store data in the Clearing House in an **encrypted way and make them practically immutable** (it requires an access token to be retrieved and decrypted).
- The document-API allows retrieval of encrypted logs via REST APIs and to decrypt them using validated tokens.

In addition, the Clearing House is:

- **completely integrated in the FIWARE Data App** and could be enabled/disabled from config file.
- **requires a valid DAPS certificate** for logging system and data retrieving, as well as a contract negotiation between provider and consumer (see Ch. 4.4.2).
- The **contract negotiation is automatically activated** during the registration phase within the FIWARE Data App (see Ch. 4.2.2).
- OneNet Connector can log any data exchange both at consumer and provider level.

4.4.1.3 Packaging and delivery

Copyright 2023 OneNet
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

As described in the previous paragraph, the OneNet Clearing House acts as intermediary in the data exchange process and for this reason it is releases as a third-party component, completely integrated with the OneNet Connector (to be enabled at config level) but running in an external cloud environment.

The OneNet Clearing house is hosted in a public ENG cloud environment and available through authentication.

4.4.2 Usage Control App

Based on the analysis of the Access and Usage Control concepts conducted in D5.7 [27] the implementation of the UC App cover three main aspects:

- Data Control Management
- Data Policies Definition
- Data Policies Enforcement

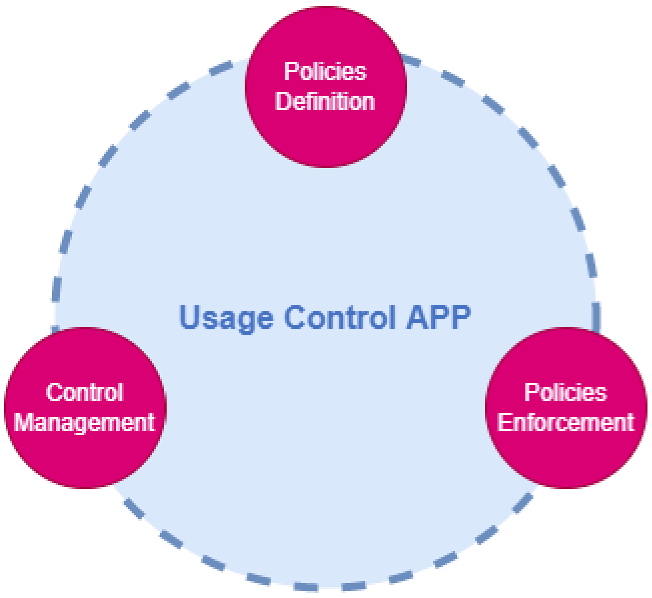


Figure 30: Usage Control App design concept

Data Control Management

The UC App should be able to apply all the defined policies to any data exchange interacting with the OneNet Connector and should allow the possibility to administrate the overall data access and usage control process.

Policy Definition

OneNet Participants that act as Data Provider must be able to create at runtime its Data Usage Control policies, to be applied to different data exchanges. A policy specification dashboard should support the Data Provider in the process of policy specification.

Policy Enforcement

The Policy Enforcement should be able to monitor, filter and mask data based on the predefined rules (e.g. anonymize personal data, remove specific information, send notifications and alerts). It should also be able to add additional enforcement rules based on external information like location, context, and purpose.

In D5.7 is also described the conceptual OneNet Data Access Policies Framework which describes how the Usage Control and Access Management should be implemented at Connector level.

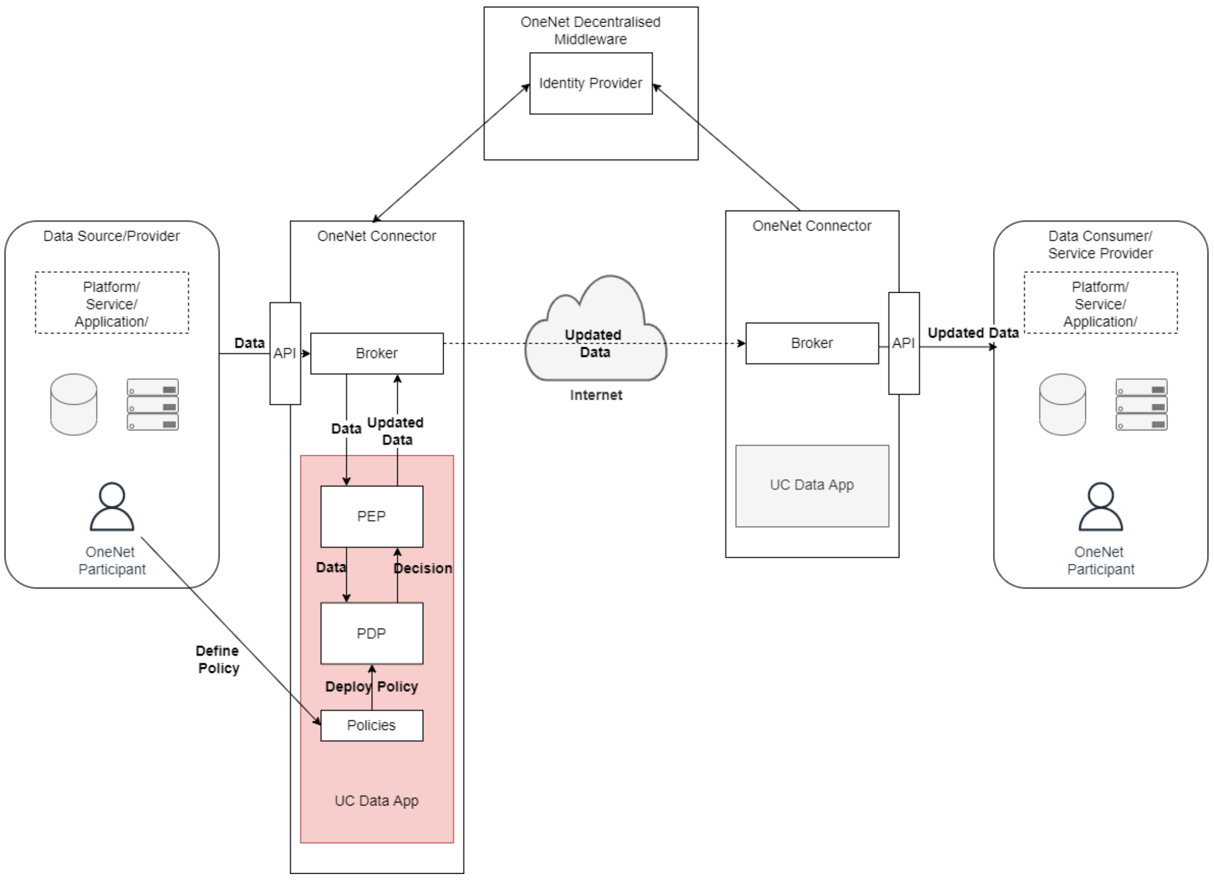


Figure 31: OneNet Data Access Policies Framework (DAP)

Figure 31 above is represents the architecture of the OneNet Data Access Policies Framework (DAP). The Usage Control on the Data Provider Side is applied whenever data is processed by the OneNet Connector. The OneNet connector is responsible for managing the data exchange and integrates the Usage Control App that can intercept the data exchanged. To ensure full data control, the Usage Control App must be invoked last before data is leaving the OneNet Connector at Data Provider Side.

Within the Usage Control App, the PEP intercepts the data flow, transforms it to a decision request and sends it to the PDP for the policy evaluation. At the same time, the OneNet Participant can define the policies that will

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

be deployed in the PDP in the form of IDS contracts. The PDP evaluates the decision request using the specific policy and returns the authorization decision to the PEP. Finally, the PEP enforces the decision in the intercepted data flow and sends back the data (accordingly updated).

The OneNet Connector currently supports the integration of different Usage Control App, among these the Fraunhofer MYDATA Framework [38] and open-source Platoon Usage Control App [29].

In this final release of the OneNet Connector, a customized version of the Platoon base application for integrating Usage Control functionality is available.

4.4.2.1 Architecture

The Data Usage Control is composed of the following components:

- Database store, where the Contract Agreements are stored. This database contains the following tables.
- The Contract Agreement Controller, which implements the REST services to get, update and remove the Contract Agreements in the database.
- The Usage Control module, which applies usage control enforcement over the input data according to the rules specified in the corresponding Contract Agreements.
- The REST Controller.

4.4.2.2 Functionalities

The Platoon Data Usage Control app [29] is implemented starting from the open-source IDS Dataspace Connector. In addition, it includes the following new functionalities:

- REST API to get, upload and remove the Contract Agreements from the Contract Agreements storage. The format of these Contract Agreements is the one specified by the IDS Information Model. These contracts will be used to apply the Data Usage Control enforcement.
- REST API to apply the Data Usage Control enforcement on the input data according to the Contract Agreements related to the pair consumer-producer indicated as input parameters.
- A new policy is supported which indicates if the data contains Personal Data, in which case the module invokes CaPe [28] for each data set belonging to a specific data subject. CaPe will filter out the content that is consented to by each data subject.

Enforcement

Policy enforcement is one of the core functionalities of the Usage Control App. Once the consumer and provider connectors have negotiated and established a Contract Agreement, this Contract Agreement is stored in the Data Usage Control by invoking the corresponding REST service.

The usage control enforcement REST service is invoked before transferring the data from the Provider Connector to the Consumer Connector and before transferring the data from the Consumer Connector to the Data App. This service will return the data according to the policies defined in the Contract Agreement.

The data Usage Control module supports usage policies written in the IDS Usage Control Language based on Open Digital Rights Language (ODRL)[5]. The policy patterns supported by the Data Usage Control module are the following ones:

- Allow the Usage of the Data: provides data usage without any restrictions.
- Prohibit the Usage of the Data: prohibits data usage.
- Interval-restricted Data Usage: provides data usage within a defined time interval. (interval defined with start end and end date)
- Duration-restricted Data Usage: allows data usage for a defined time period. (duration calculated from contract start date)
- Role-restricted Data Usage: allows data usage for a defined role. (for example http://example.com/ids-role:riskManager)
- Purpose-restricted Data Usage Policy: allows data usage for defined purposes. (for example http://example.com/ids-purpose:Marketing)
- Restricted Number of Usages: allows data usage for n times.
- Personal Data: filter out the contents of the data according to the data subject´s consents. To apply this rule, the Usage Control module interacts with CaPe.

4.4.2.3 Packaging and delivery

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

The Usage Control App docker container is released together with the OneNet Connector. Please refer to the OneNet Connector official guidelines [23] for the deployment and installation of the software. and to the Platoon Usage Control App for specific configuration and installation [29].

[5] https://www.w3.org/TR/odrl-model/

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

5 Conclusions

The work conducted in Task 6.3 and documented in this deliverable gives a complete overview on the implementation of the Data Integration & Homogenization sub-layer following the design and requirements defined in WP5. This implementation was conducted at three different stages: the first release of the Data Integration & Homogenization sub-layer was included in the release of the OneNet Connector in July 2022; the intermediate release was integrated with the OneNet Connector and within the overall OneNet System (including other components) in January 2023; the final release was released at July 2023.

The adoption of FIWARE Smart Energy Architecture and the usage and evolution of the FIWARE Context Broker together with the adoption of the standardised data models were fundamental for achieving the main objectives that OneNet Solution foresees. The FIWARE ecosystem was completely integrated in the OneNet Solution and in the OneNet Connector through the FIWARE Data App and the Orion Context Broker both supporting the NGSI-LD standard for data exchange.

In addition, the Data Integration & Homogenization sub-layer includes the OneNet Data Services Layer which offers the possibility to extend the OneNet Connector with specific data services, supporting the data governance and data exchange for the OneNet participants perspective (both Data Provider and Data Consumer).

The OneNet Data Services Layer included a Data Homogenization tool, capable to integrate IEC 62325-451 entities with the FIWARE NGSI-LD information model, validate the entities and converting from XML or JSON formats into NGSI-LD valid format. All these functionalities are provided through standardized REST APIs interfaces and integrated within the OneNet Connector during the data offering provisioning.

Task 6.3 also included all the activities related to the integration of two important building blocks of the IDS reference architecture model: the Clearing House and Usage Control App. The Clearing House can log all activities performed in the course of data exchange in the IDS ecosystem, while the Usage Control App allows to define Usage Policies and Usage Enforcement**.** Both the services are completely integrated in the OneNet Connector.

The aspects related to interoperability, standardization of data models and data access control are among the most relevant issues in the context of a data-based ecosystem such as that of OneNet.

The final version of the OneNet Connector, including now additional services like Data Homogenization Tool, Clearing house and Usage Control, strongly increases the interoperability of the connector itself and the overall OneNet System, now supporting the conversion from XML to JSON-LD of some of the most used CIM standards in the energy sector and at the same time adding some fundamental functionalities for energy stakeholders, who need to have control over the secure and traceable data access.

Copyright 2023 OneNet

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

References

[1] FIWARE Smart Energy Architecture https://www.fiware.org/wp-content/uploads/2018/10/Smart-Energy-Brochure-FIWARE-Web-1-2.pdf.

[2] OneNet Deliverable 5.2 “OneNet Reference Architecture” [Online]. Available: https://onenet-project.eu/wp-content/uploads/2022/12/OneNet_D5.2_v1.0.pdf.

[3] OneNet Deliverable 5.4 “D5.4 AI, Big Data, IoT Enablers and FIWARE-compliant interoperable interfaces for grid services” [Online]. Available: https://onenet-project.eu/wp-content/uploads/2022/12/OneNet_D5.4_v1.0.pdf.

[4] OneNet Deliverable 5.5 “Report on Technical specifications for data models/platform agnostic middleware” [Online]. Available: https://onenet-project.eu/public-deliverables/.

[5] NGSI-LD Standard – Guidelines for modelling with NGSI-LD. [Online]. Available: https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp_42_NGSI_LD.pdf.

[6] FIWARE Orion Context Broker, [Online]. Available: https://fiware-orion.readthedocs.io/en/master/.

[7] FIWARE-NGSI v2 Specification, “Next generation service interfaces architecture approved version [Online].Available:http://fiware.github.io/specifications/ngsiv2/stable.

[8] Context Information Management (CIM); NGSI-LD API,ETSI GS CIM 009 V1.4.1 (2021-02) https://www.etsi.org/deliver/etsi_gs/CIM/001_099/009/01.04.01_60/gs_cim009v010401p.pdf.

[9] Graph Databases: "New Opportunities for Connected Data". O'Reilly 2nd Edition. Webber, Robinson, et al. ISBN:1491930896 9781491930892.

[10] ENTSO-E – Electronic Data Interchange Library (EDI). [Online]. Available: https://www.entsoe.eu/publications/electronic-data-interchange-edi-library/.

[11] ETSI - European Telecommunications Standards Institute, https://www.etsi.org/.

[12] Open Mobile Alliance – Wikipedia, https://en.wikipedia.org/wiki/Open_Mobile_Alliance.

[13] FIWARE initiative, https://www.fiware.org/.

[14] FIWARE Smart Data Models, [Online]. Available: https://www.fiware.org/smart-data-models/.

[15] FIWARE Smart Data Models Repository, [Online]. Available: https://github.com/smart-data-models/data-models/tree/master/specs.

[16] FIWARE Smart Data Models - Smart Energy, [Online]. Available: https://github.com/smart-data-models/SmartEnergy.

[17] European Union, “Clean Energy for all Europeans”, 2019. [Online]. Available: https://op.europa.eu/en/publication-detail/-/publication/b4e46873-7528-11e9-9f05-01aa75ed71a1/language-en?WT.mc_id=Searchresult&WT.ria_c=null&WT.ria_f=3608&WT.ria_ev=search.

[18] European Commission, ”COSMAG - Background Document on big data and energy”, [Online]. Available: http://ec.europa.eu/research/participants/portal/doc/call/h2020/dt-ict-11-2019/1850117-background_document_on_big_data_and_energy_en.pdf.

[19] SpringBoot Framework, https://spring.io/projects/spring-boot.

[20] OneNet Deliverable 6.1 “Report on decentralized edge-level middleware for scalable platform agnostic data management and exchange” [Online]. Available: https://onenet-project.eu/wp-content/uploads/2023/02/D6.1-OneNet-v1.0.pdf.

Copyright 2023 OneNet

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 957739

[21] MongoDB, https://www.mongodb.com/.
[22] OpenAPI Specifications. [Online]. Avaialable: https://spec.openapis.org/oas/v3.1.0.
[23] OneNet Connector deployment and installation guidelines. [Online]. Available: https://github.com/european-dynamics-rnd/OneNet/blob/main/README.md.
[24] RFC 7807 Standard. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc7807.
[25] CIM2FIWARE – Public Repository. [Online]. Available: https://git.rwth-aachen.de/acs/public/cim2fiware.
[26] Fraunhofer AISEC IDS Clearing House. [Online]. Available: https://github.com/Fraunhofer-AISEC/ids-clearing-house-service.
[27] OneNet Deliverable 5.7 "Report on Data Enforcement Policies Design for Sovereignty preserving data access" [Online]. Available: https://onenet-project.eu/wp-content/uploads/2022/12/OneNet_D5.7_v1.0.pdf.
[28] CaPe — Consent-based Personal Data Suite. [Online]. Available: https://www.eng.it/en/case-studies/dati-sicuri-conformi-gdpr
[29] Platoon Data App Usage Control. [Online]. Available: https://github.com/Engineering-Research-and-Development/true-connector-uc_data_app_platoon.
[30] OneNet Deliverable 6.5 "OneNet Reference Platform First Release" [Online]. Available: https://onenet-project.eu/wp-content/uploads/2022/10/OneNet_D6.5_final_v1.1.pdf.
[31] European Commission, "The Rolling Plan on ICT Standardisation", 2018, https://digital-strategy.ec.europa.eu/en/library/rolling-plan-ict-standardisation.
[32] OneNet Deliverable 5.6 "Report on Extended Data, Platform and service Interoperability" [Online]. Available: https://onenet-project.eu/wp-content/uploads/2022/12/OneNet_D5.6_v1.0.pdfhttps://onenet-project.eu/wp-content/uploads/2022/10/OneNet_D6.5_final_v1.1.pdf.
[33] SAREF: the Smart Applications REFerence ontology, [Online]. Available: https://saref.etsi.org/core/v3.1.1/.
[34] SAREF4ENER: an extension of SAREF for the energy domain created in collaboration with Energy@Home and EEBus associations, [Online]. Available: https://saref.etsi.org/saref4ener/v1.1.2/.
[35] Maliheh Haghgoo, Ilya Sychev, A. Monti and Frank Fitzek, "SARGON-Smart energy domain Ontology", Smart Cities, 2020, https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/iet-smc.2020.0049.
[36] Haghgoo, M.; Nazary Aghche Mazary, A.; Monti, A. "SiSEG-Auto Semantic Annotation Service to Integrate Smart Energy Data",. *Energies* 2022, *15*, 1428. https://doi.org/10.3390/en15041428.
[37] OneNet Deliverable 5.3 "Data and Platform Assets Functional Specs and Data Quality Compliance" [Online]. Available: https://onenet-project.eu/wp-content/uploads/2022/12/OneNet-D5.3-v1.0.pdf.
[38] Fraunhofer IESE, MYDATA Framework, [Online]. Available: https://www.mydata-control.de.

Copyright 2023 OneNet
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739