ONENET
one network for Europe

Report on Technical specifications for data models/platform agnostic middleware

D5.5

Authors:

Apostolos Kapetainos (ED)

Ferdinando Bosco (ENG)

Konstantinos Kotsalos (ED)

Responsible Partner	ED
Checked by WP leader	**ENG** - Date: 28/01/2021
Verified by the appointed Reviewers	**EPRI** - Date: 27/01/2021 **E.ON** – Date 23/01/2022
Approved by Project Coordinator	**Stephan Gross (FhG)** – Date: 31/01/2022

Dissemination Level		
PU	Public	X
CO	Confidential, only for members of the consortium (including the Commission Services)	
CI	Classified, as referred to in Commission Decision 2001/844/EC	

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	31/01/2022
Actual date of delivery	
Status and version	V1.0

Version	Date	Author(s)	Notes
0.0	30/10/2021	Konstantinos Kotsalos	First structure of the deliverable with Table of Contents
0.1	4/01/2022	Apostolos Kapetanios (ED)	Chapters 3 & 4.1
0.1	10/01/2022	Konstantinos Kotsalos (ED)	First Draft on Chapter 2
0.2	15/01/2022	Apostolos Kapetanios (ED)	Executive Summary, Chapters 1 & 4.2
0.3	18/01/2022	Ferdinando Bosco (ENG)	Chapter 2

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

About OneNet

OneNet 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.

The project OneNet (One Network for Europe) is funded through the EU's eighth Framework Programme Horizon 2020. It is titled "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)".

While the electrical grid is moving 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. For this reason, the two major associations of grid operators in Europe, ENTSO-E and EDSO, have activated their members to put together a unique consortium.

OneNet will see the participation of a consortium of over 70 partners. Key partners in the consortium include already mentioned ENTSO-E and EDSO as well as Elering, E-Redes, RWTH Aachen University, University of Comillas, VITO, European Dynamics, Ubitech, Engineering, and the EUI's Florence School of Regulation (Energy).

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.

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

Table of Contents

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

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
ADM	Architecture Development Method
ASRs	Architecturally Significant Requirements
ATAM	Architecture Trade-off Analysis Method
COTS	Commercial Off-The-Shelf
DoA	Description of Action
DSO	Distribution System Operator
FSR	Florence School of Regulation
GA	Grant Agreement
IT	Information Technology
LAE	Lightweight Architecture Evaluation
RA	Reference Architecture
RFP	Request for proposal
SAAM	Software Architecture Analysis Method
TRL	Technology Readiness Level
TSO	Transmission System Operator
WP	Work Package

List of Figures

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

List of Tables

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

Executive Summary

One of the most important aspects of the OneNet network of platforms implementation effort is:

- To design an open conceptual architecture for effective yet seamless operation of a smarter pan-European electricity system where market and network technical operations are coordinated closer to real time among them and across countries.
- To provide requirements, functional and technical specifications, together with interoperable and standardized interfaces for an open scalable decentralized interconnection of platform, technology agnostic adaptable and flexible IT reference architecture which fully supports the OneNet concept and provides the necessary backbone for the WP6 subsequent implementation of the OneNet data sovereignty-preserving working space.

Based on the above, this deliverable aims to present in a clear and concise manner the project technical architecture with a view to enable interoperability and integration of the OneNet Interoperable Network of Platforms with all available platforms and systems. Integral parts of this technical architecture are well documented technical specifications and interfaces for data models/platform agnostic OneNet middleware. These technical specifications and interfaces constitute an important design block of the reference implementation of the OneNet solution:

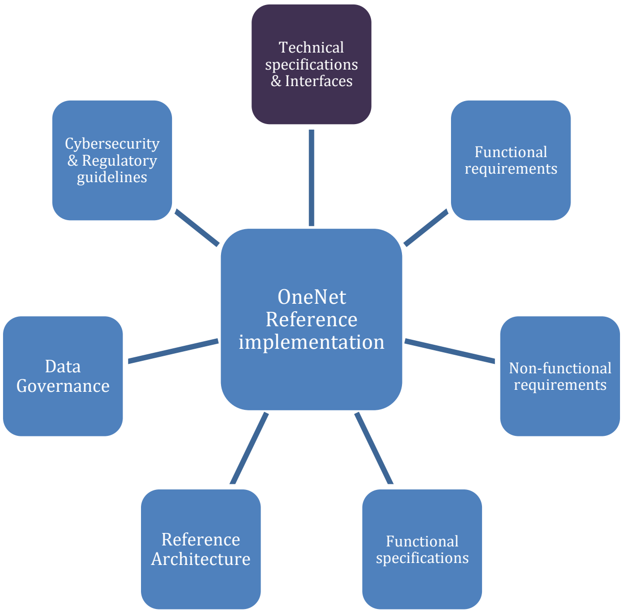

Technical specifications & Interfaces
Cybersecurity & Regulatory guidelines
Functional requirements
OneNet Reference implementation
Data Governance
Non-functional requirements
Reference Architecture
Functional specifications

Figure 1: OneNet design pillars

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

So, the results of this deliverable are necessary for the whole WP6 implementation phase but also bring us closer to a pan-European solution which is capable to interconnect any smart energy platform and involve energy stakeholders at any level (TSOs, DSOs, Customers, etc…).

From a technical perspective the specifications and interfaces analysis of D5.5 are closely connected with standard architectures and initiatives (IDSA/FIWARE) in order to create a OneNet framework which is:

- scalable, pluggable and fully decentralized
- provides a series of data harmonization services
- provides tools for Data and Services Orchestration and evaluation
- has monitoring and analytics features
- takes into account cybersecurity and data governance guidelines

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 presents the context in which the activities of WP5 and more specifically of T5.5 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 OneNet Scope

The OneNet will create a fully replicable, open, flexible and scalable architecture that enables the whole European electrical system to operate as a single efficient system in which a variety of markets allows the universal participation of stakeholders regardless of their physical location, at every level from small consumer to large producers. Also, by clearly define stakeholder interactions and bringing all possible data exchanges to a European level of harmonization it will fully unlock markets at every level and expand the possibilities for a real commercial exploitation.

The OneNet results will be:

- A data management framework which will support flexibility markets, but also monitor and optimize the overall European electrical infrastructure
- A clear and open architecture that will enable any player to participate at innovative market structures
- A smooth integration of the grid and market operation for TSO and DSO in the innovative market structure
- A new set of customer-centric business models to support next generation service-based markets

According to OneNet Description of Action (DoA), WP5 contributes to the direction of fulfilling the OneNet vision by striving to attain two objectives; First, to design an open conceptual architecture for effective yet seamless operation of a smarter pan-European electricity system where market and network technical operations are coordinated closer to real-time across countries, and second to provide requirements, functional and technical specifications, together with interoperable and standardisable interfaces for an open scalable decentralized interconnection of platforms, technology agnostic adaptable and flexible IT reference architecture which fully support the OneNet concept and

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

provides the necessary backbone for the WP6 subsequent implementation of the OneNet data sovereignty-preserving working space. The WP5, together with WP6, act as IT pillar of the overall OneNet project. The IT pillar it 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 four (4) Demonstration Clusters and the results of the evaluation will be used for adapting, improving, and enhancing the OneNet Solution.

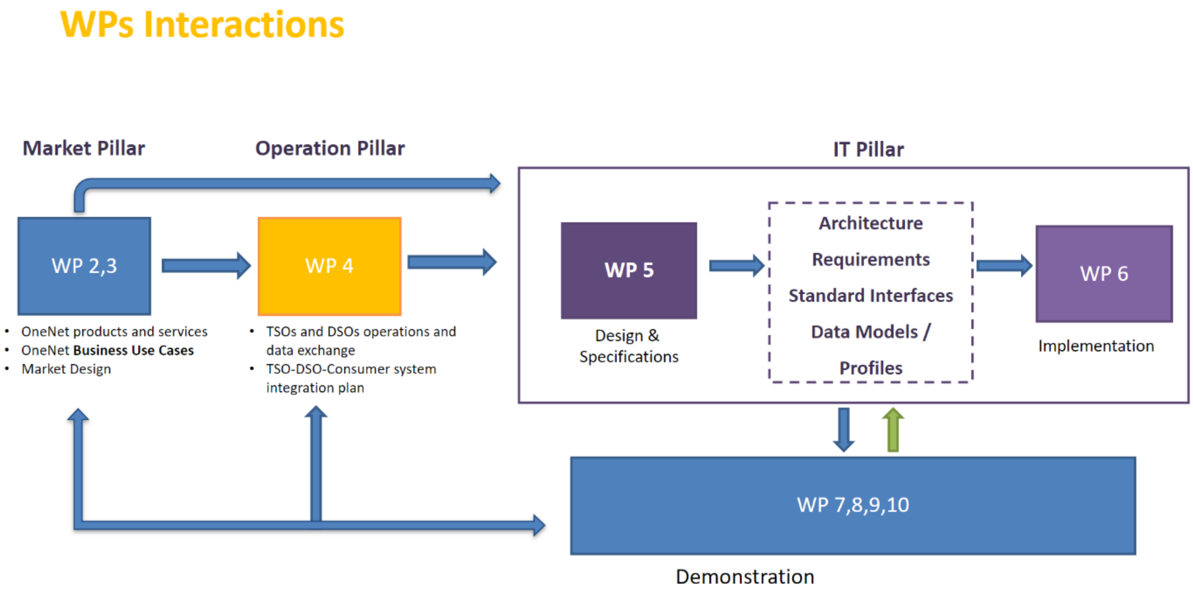


Figure 1: WP5 interdependencies

1.2 Task 5.5 within OneNet

Within the context described above, the main goal of the Task 5.5 is defining the OneNet technical architecture with a view to enable interoperability and integration of the OneNet Interoperable Network of Platforms with all available platforms and systems that generate data that could be required for different services by different actors. The deliverable 5.5 analyses the interfaces which will be developed with ICT tools and the data sources implemented by operators and common repositories to improve the collaboration among relevant stakeholders and systems based on previous successful experiences and existing systems. These appropriate interfaces will be developed as plugins to be described in WP6, endorsing open standards.

The deliverable provides also the IT services, which will support the above interfaces by describing the distinct patterns on how they will be developed/consumed keeping in mind that the specification of the service, the implementation of the service, and the consumption of the service are created by a different part of a platform. These services will be exposed publicly through REST based architecture allowing their use through HTTP request operations.

Task 5.5 can be considered an integral task of the whole WP5. This task provides necessary information of the OneNet Technical Architecture, by taking as input all the Use Cases, functional and non-functional requirements, and other useful information of the OneNet Reference Architecture. So T5.5 forms the basis for OneNet solution implementation, as a reference implementation for the Demonstration Clusters, but also with a view to being used as a future reference for the implementation of a unique pan-European solution for the provisioning of coordinated countries and stakeholders' market and grid operations.

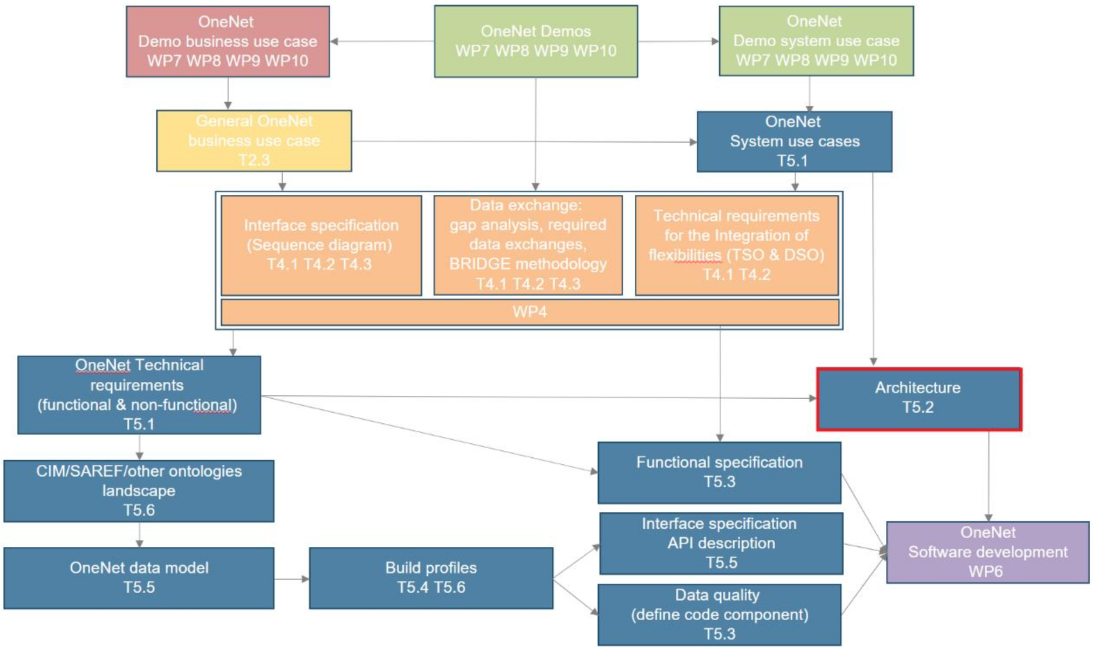

OneNet Demo business use case WP7 WP8 WP9 WP10
OneNet Demos WP7 WP8 WP9 WP10
OneNet Demo system use case WP7 WP8 WP9 WP10
General OneNet business use case T2.3
OneNet System use cases T5.1
Interface specification (Sequence diagram) T4.1 T4.2 T4.3
Data exchange: gap analysis, required data exchanges, BRIDGE methodology T4.1 T4.2 T4.3
Technical requirements for the Integration of flexibilities (TSO & DSO) T4.1 T4.2
WP4
OneNet Technical requirements (functional & non-functional) T5.1
Architecture T5.2
CIM/SAREF/other ontologies landscape T5.6
Functional specification T5.3
Interface specification API description T5.5
OneNet Software development WP6
OneNet data model T5.5
Build profiles T5.4 T5.6
Data quality (define code component) T5.3

Figure 2 WP5 interdependencies

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

1.3 Report Outline

This deliverable is structured in 6 different chapters.

Chapter 1 is the introductory one where the deliverable is contextualized within the WP5 and Task 5.5.

Chapter 2 analyses the concept of the OneNet decentralized middleware by presenting an overview of a) the reference architecture, b) cross-platform services and data model harmonization, c) system use cases and d) functional requirements.

Chapter 3 describes the OneNet interfaces based on the general use cases.

Chapter 4 focuses on the OneNet information model approaches both from standards (e.g. CIM) and technical initiatives point of view (e.g. IDSA, FIWARE).

Chapter 5 models the REST APIs which will be part of the OneNet middleware and attempts to provide technical specifications for the WP6 implementation activities.

Finally, *Chapter 6* concludes the document.

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

2 OneNet Decentralised Middleware

2.1 Overview of OneNet architectural approach

The analysis of the architectural approach (see Figure 3) and structure of OneNet is analytically provided in Deliverable 5.2. Hereby a brief overview is provided as follows. OneNet relies on data interoperability mechanism to all platforms as a matter of supporting data exchange for facilitating market and network operations and the cooperation between network operators, like TSOs and DSOs as well as the involvement of other players like prosumers and aggregators. To achieve the seamless interoperability, fundamental characteristics that shall be covered are adoption of open standards and interfaces, data privacy and data access to regulation for each stakeholder, definition of standard models and protocols for data exchange, provision of data management and dataflow monitoring, identification, authentication and authorization mechanisms. The decentralized approach and the use of standardized interfaces and mechanisms therefore assume fundamental importance to satisfy all these characteristics and in particular to ensure the necessary scalability for the near real-time data integration and management enabling multi-country and multi-stakeholder near real-time services.

The analysis of IDS reference model and FIWARE interfaces, bring to a hybrid solution using both the standard models for implementing the OneNet Decentralized middleware and the OneNet Connector. The usage of IDS Connector and FIWARE Context Broker was identified as the best solution to be adopted for ensuring a high level of standardization, interoperability, scalability and reuse of OneNet solution.

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

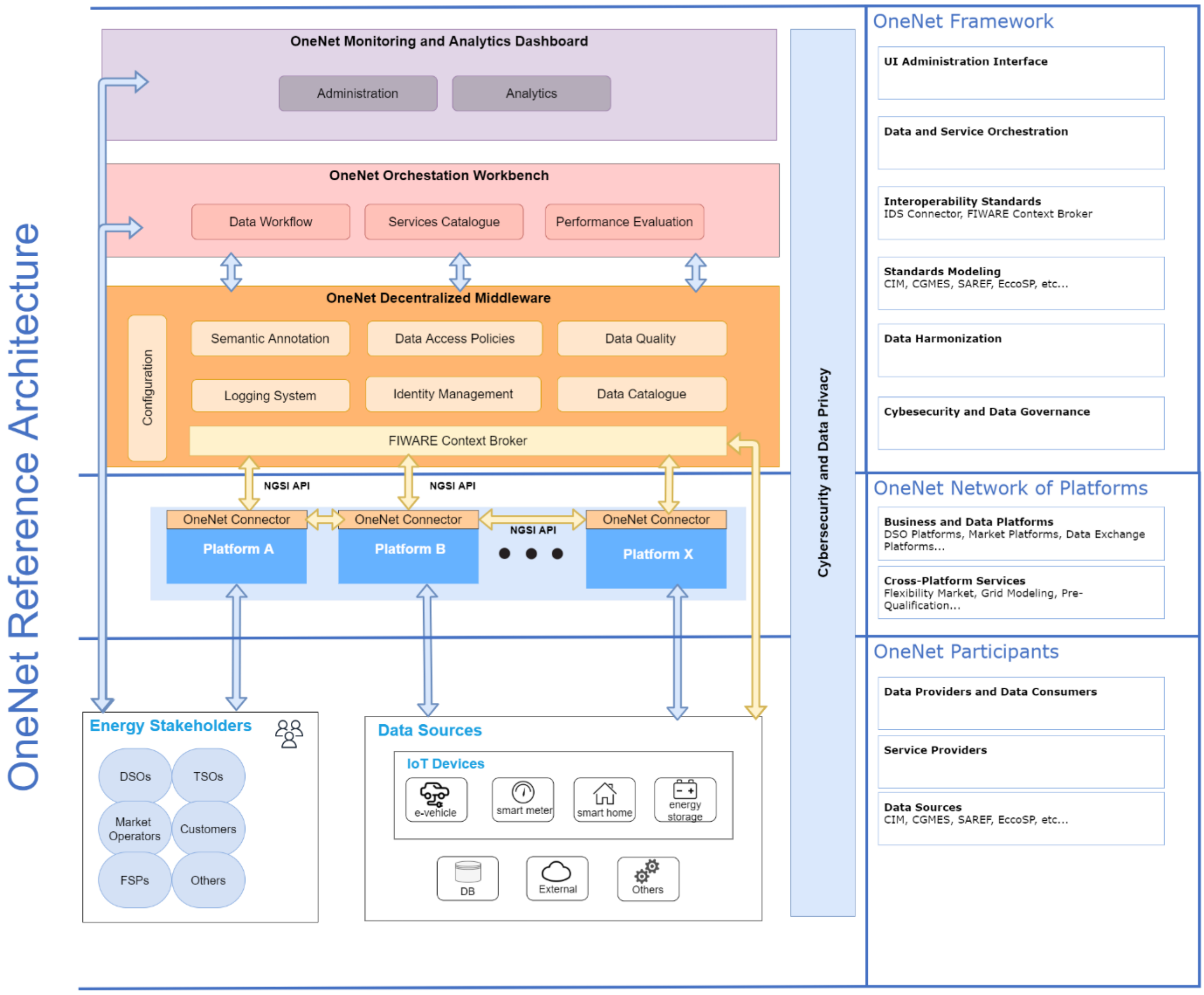

OneNet Reference Architecture
OneNet Monitoring and Analytics Dashboard
Administration
Analytics
OneNet Orchestation Workbench
Data Workflow
Services Catalogue
Performance Evaluation
OneNet Decentralized Middleware
Configuration
Semantic Annotation
Data Access Policies
Data Quality
Logging System
Identity Management
Data Catalogue
FIWARE Context Broker
NGSI API
NGSI API
OneNet Connector
OneNet Connector
NGSI API
OneNet Connector
Platform A
Platform B
Platform X
Cybersecurity and Data Privacy
Energy Stakeholders
DSOs
TSOs
Market Operators
Customers
FSPs
Others
Data Sources
IoT Devices
e-vehicle
smart meter
smart home
energy storage
DB
External
Others
OneNet Framework
UI Administration Interface
Data and Service Orchestration
Interoperability Standards
IDS Connector, FIWARE Context Broker
Standards Modeling
CIM, CGMES, SAREF, EccoSP, etc...
Data Harmonization
Cybesecurity and Data Governance
OneNet Network of Platforms
Business and Data Platforms
DSO Platforms, Market Platforms, Data Exchange Platforms...
Cross-Platform Services
Flexibility Market, Grid Modeling, Pre-Qualification...
OneNet Participants
Data Providers and Data Consumers
Service Providers
Data Sources
CIM, CGMES, SAREF, EccoSP, etc...

Figure 3 OneNet reference architecture

The OneNet Reference Architecture consists of three logical layers:

- the bottom layer includes data sources and energy stakeholders, the **OneNet Participants**;
- the middle layer is the one that in the OneNet ecosystem allows the creation of a **OneNet Network of Platforms** and includes all the platforms that participate in data exchange and the use of cross-platform services. In this layer there is the first component provided by OneNet, the OneNet connector;
- the top layer is the one properly defined as **OneNet Framework**. This is the core of the OneNet Architecture. It includes all the components that will be implemented in the reference implementation in WP6, as well as all the necessaries specifications for data harmonization, ontologies, data modelling, service orchestration, workflow monitoring, analytics, etc.

The OneNet Network of Platforms layer focuses in the integration of external platforms, such as DSO platforms, Market platforms and other data exchange platforms into the OneNet system. This

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

integration is to be made regardless of the technology of these platforms in order to remain platform-agnostic.

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

From the OneNet perspective, the more important component included in this layer is the OneNet Connector.

A OneNet Connector is a specific instance of the OneNet Decentralized Middleware, will be placed inside each platform and will allow an easy integration and cooperation among the platforms, maintaining the data ownership and preserving access to the data sources.

As described in the D5.4, OneNet connector will rely on the FIWARE TRUE Connector (FTC), a connector for the IDS (International Data Space) ecosystem. FTC enables the trusted data exchange in order to be active part of an IDS Ecosystem, a virtual data space leveraging existing standards and technologies, as well as governance models well-accepted in the data economy, to facilitate secure and standardized data exchange and data linkage in a trusted business ecosystem. This connector's implementation is compliant with the latest IDS specifications and can be easily customized to fit a wide spread of scenarios thanks to the internal separation of Execution Core Container and Data App. It is integrable with a lot of existing IDS services and totally configurable in terms of internal/external data format (multipart/mixed, multipart/form, http-header) and protocols (HTTP, HTTPS, Web Socket over HTTPS, IDSCPv2). The FTC includes the Execution Core Container, based on the IDS Reference Model for the integration of the IDS based services and metadata exchange as well as the NGSI-LD Data App, able to enable the data exchange using the FIWARE NGSI-LD Context Broker.

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

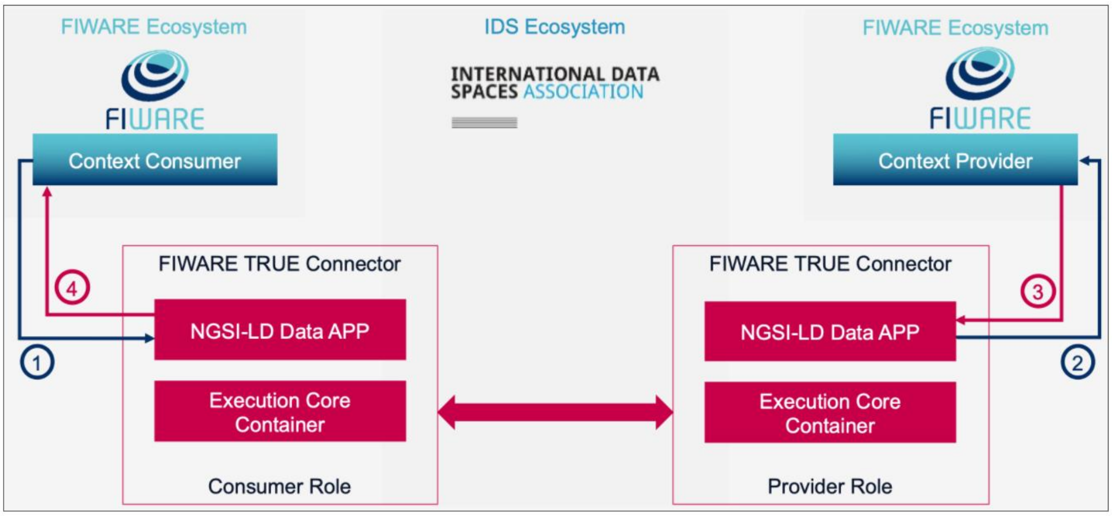


Figure 4 Technical composition of FIWARE TRUE Connector [2]

2.2 Cross-platform services and OneNet data-models harmonization

The goal of the OneNet System is to facilitate data exchanges among existing platforms, services, applications, and devices by the power of interoperability techniques. To ensure that system requirements are technically - implementable and widely adopted, internationally standardized file formats, metadata, vocabularies and identifiers - are required.

2.2.1 OneNet harmonised cross-platform services

Based on extensive analysis OneNet, the need to accommodate the following categories were defined to render a taxonomy of cross-platform data services that will be enabled by OneNet decentralized middleware. The provided categories, and subsequently the defined cross-platform data service as defined in Deliverable 5.3 of OneNet are open to further extensions and additions followed by harmonized semantics.

Table 1: Categories of cross-platform services in OneNet, [3]

No	Category Name	Description
1	Authentication & Authorization	Activities related to cross-platform authentication and authorization. This category is different from the other categories of cross-platform services, since it specifies cross-domain services for authentication and data access policies.
2	Measurements & Monitoring	Exchanging measurements or other data related to monitoring, e.g., state estimation results

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

3	Forecasts	Exchanging forecast of any kind
4	Reports & invoices	Activities related to reporting or invoicing of system or other services, incl. Reporting energy/flexibility settlement
5	(Flexibility) Market participation	Activities related to participation in market, e.g., sending bids, market clearing etc.
6	Grid models	Exchange of grid models, for example for grid reconfiguration
7	Simulation results	Exchange of simulation results, for example power flow results
8	Resource (pre-) qualification	Activities related to the (pre-) qualification of resources, incl. qualification of product's/ service's technical parameters
9	System service activation	Ask system operator to activate/ start certain system service
10	Resource control	Sending set points to assets/ flexibility sources etc.

2.2.2 Harmonised OneNet business objects for platform agnostic middleware for electricity sector

To serve the cross-platform services among different connected platforms in the OneNet data ecosystem, there is an evolving work in Task 5.6 of OneNet to suggest commonly accepted data profiles. This process details the collection of all relevant business objects that are used in the different proposed cross-platform services. The target is to, finally, describe the appropriate data profiles that consider the transaction of the defined (i.e., in cross-platform services) business objects, creating a set of harmonized semantics for the OneNet ecosystem.

OneNet harmonized semantics will be a collection of existing, commonly adopted data profiles merely based on IEC Common Information Model (CIM), that among others cover extensive business processes as previously taxonomized (in Section 2.2.1). Therefore, domain specific profiles are described such as data profile IEC 62325, 62926, 61970, 61968 (exploiting also common requirements for TSO and DSO) with the likelihood to enhance them. The harmonized data profile will accompany the specification and documentation of the interfaces for the development of cross-platform services.

2.3 OneNet Demo and General System Use Cases

OneNet Deliverable 5.1 [1] provides the overall concept of the OneNet System based on the collection of the Demo System Use Cases and General OneNet System Use Cases.

Starting from these SUCs, the D5.1 also provides the list of Functional and Non-Functional Requirements for the overall OneNet System. In this chapter, we analysed all the information related to OneNet Decentralised Middleware.

Starting from the analysis of the System Use Cases, it is evident that most relevant SUCs for the purpose of this deliverable are the “GSUC_01: Cross-Platform Energy Data Exchange for market-based flexibility management” and the “GSUC_03: Integration of devices and other data sources to OneNet using FIWARE “.

As described in the GSUC_01 the main characteristics for enabling a cross-platform data exchange should be:

- Ensure a standardized connection of different platforms
- Allow the discovery of data sources and services
- Definition of common vocabularies for improving interoperability
- Manage data exchange in a secure and trusted way

All those characteristics are crucial for the implementation of the OneNet Decentralised Middleware, that is the core components in charge of managing the data exchange and the cross-platform integration.

The cross-platform integration and cooperation for market and network operation services is based on IDS reference model and foresees an interaction between the OneNet Participants (platforms, applications or services that act as Data Provider or Data Consumer) and the OneNet Decentralised Middleware.

This use case also describes which kind of components and tools should be present in the OneNet Middleware and in the OneNet Connector for fulfilling the expected goals and follow the identified specifications.

More in detail, the following tools and components should be part of the OneNet Decentralised Middleware and OneNet Connector following IDS reference model:

Identity Provider offers a service to create, maintain, manage and validate identity information of and for participants in the OneNet System. This is imperative for secure operation and to avoid unauthorized access to data.

Broker Service Provider is an intermediary that stores and manages information about the data sources available in OneNet system. The activities of the Broker Service Provider mainly focus on receiving and providing metadata. The Broker Service Provider must provide an interface for Data Providers to send their metadata. The

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

metadata should be stored in an internal repository (Broker Service Registry) for being queried by Data Consumers in a structured manner.

Clearing House is an intermediary that provides clearing and settlement services for all financial and data exchange transactions. In OneNet, clearing activities are separated from broker services, since these activities are technically different from maintaining a metadata repository. The Clearing House logs all activities performed in the course of a data exchange. After a data exchange, or parts of it, has been completed, the Service Provider (see below) confirms the data transfer by logging the details of the transaction at the Clearing House. Based on this logging information, the transaction can then be billed. The logging information can be used also to re-solve conflicts (e.g., to clarify whether a data package has been received by the Data Consumer or not). The Clearing House also provides reports on the performed (logged) transactions for billing, conflict resolution, etc.

Vocabulary Provider manages and offers vocabularies that can be used to annotate and describe datasets. In particular, the Vocabulary Provider provides the Information Model of the OneNet, which is the basis for the description of data sources. In addition, other domain specific vocabularies can be provided.

The GSUC_03 adds a further relevant element for the design and implementation of the OneNet Decentralised Middleware, the adoption and evolution of the FIWARE Context Broker for providing a data-model agnostic connector based on NSGI-LD.

Context Broker will be part of the OneNet Connector following the in FIWARE-based implementations of the IDS Architecture. In fact, the Context Broker offers the FIWARE NGSI APIs and associate information model (entity, attribute, metadata) as the main interface for sharing data by the OneNet participants. Data Providers use the APIs to publish or to expose the data they offer (normally through a System Adaptor) and Data Consumers retrieve or subscribe (to be later notified) to the data offered.

More detail in the characteristic of the FIWARE Context Broker and the NGSI-LD standard are also provided in D5.4 [4].

Other relevant considerations could be extracted in the analysis of the Demo System Use Case. Indeed, most of the needs collected in the Demo System Use Cases clearly address the OneNet Decentralised Middleware for implementing a secure and trusted platform integration and data exchange.

The next paragraph, report the functional requirements relevant for the implementation of the OneNet Decentralised Middleware.

2.4 OneNet Decentralized Middleware functional requirements

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

Requirement ID	Requirement Name	Description	Reference
OneNet_FUR_01	The OneNet system must enable exposure of list of data/ services from vertical WPs to third parties	System of the OneNet Participant has many features/roles and data. Those can be accessed through API's by third party. The list of services/data and their properties can be retrieved automatically by special API - "Catalogue service". This list can be provided by API to OneNet system, in order to be exposed to potential third parties.	Northern Cluster, General SUC_01, General SUC_02
OneNet_FUR_02	The OneNet system must enable role-based access for data/service to authenticated users.	Every data/service responds to authenticated requests only. In case third party need the access then the authentication/secure channel needs to be established.	Northern Cluster
OneNet_FUR_03	The OneNet system must visualize and provide analysis tool for activity logs.	User activity trace logs, technical performance or problem related logs are generated and could be exposed from demonstrator's implementation to third parties, through the OneNet system	Northern Cluster, GSUC_01
OneNet_FUR_04	The OneNet system must facilitate the communication of the SO's flexibility needs to external interested stakeholders	The SOs, i.e., DSO and TSO, shall be able to make available to stakeholders their flexibility needs in different timeframes, e.g., Day-ahead and Intra-day, through the utilization of the OneNet system.	DSUC_WE_PT_02
OneNet_FUR_05	The OneNet system must facilitate market results to be disseminated to external interested stakeholders	The local market platform publishes collected market results through OneNet system to external interested parties.	DSUC_SP_01
OneNet_FUR_06	The OneNet system must facilitate data exchange amongst SOs, MOs, and FSPs participating in the market-based flexibility	Prequalified limits in the interface between the HV/MV (TSO) and MV/LV (DSO) that FSPs exist are sent to the market (TSO market or local DSO market) in order to be taken into consideration by the market operator in the	DSUC_SO_CY_02, DSUC_SO_CY_03, DSUC_SO_CY_04

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

	procurement process, for prequalification, market clearing, evaluation and real-time control purposes.	allocation of the awarded bids to the FSPs. In addition, MO publishes the awarded bids to the operators through the OneNet. After the activation of the flexibility, evaluation report of the FSP's performance is sent to the market operator through the OneNet platform. Finally, communication between the DSO control centre (ABCM-D platform is develop in the context of Cypriot Demonstrator) and the local FSPs connected to the distribution grid through the OneNet system	
OneNet_FUR_07	The OneNet system must connect the involved in the Demo parties to external actors responsible for TSO-DSO coordination	TSO/DSO coordination process takes place through the utilization of OneNet. This specifically includes: DSO demand finalization, flexibility registration, bid prequalification and market result broadcasting	DSUC_EA_HU_01 DSUC_EA_HU_02 DSUC_EA_HU_03
OneNet_FUR_08	OneNet system must be able to manage and certificate the identity of each OneNet Participant	OneNet system manage the identities of all the OneNet participants offering an Identity Provider	
OneNet_FUR_09	OneNet system must be able to register/unregister a OneNet connector	OneNet Connector need to register itself before starting any data exchange process	
OneNet_FUR_10	Each OneNet Participant must be uniquely identified using certification	OneNet Participants are uniquely identified within the OneNet ecosystem, using certification process and establishing trust among all participants.	
OneNet_FUR_11	Each OneNet Connector have a unique certificate and identifier		
OneNet_FUR_12	Each OneNet Connector is able to verify the identity		

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

	of the other OneNet Connectors		GSUC_01 GSUC_03
OneNet_FUR_13	OneNet participant must be able to run the OneNet connector in its own environment	OneNet Middleware leverage on the IDS decentralized approach. The OneNet Connector provided by OneNet must be deployable in any environment	
OneNet_FUR_14	The OneNet Participant must be able to configure its own OneNet Connector	OneNet connectors are configurable by the OneNet participants using specific interfaces	
OneNet_FUR_15	The OneNet connector must be able to send metadata of a data source to one or more Brokers	Once the connector is configured it is able to connect the Brokers for starting data exchange. The connector is able to provide and/or search metadata as well as discover for new data sources and participants.	
OneNet_FUR_16	The OneNet Participant must be able to search and discover other OneNet Participants		
OneNet_FUR_17	The OneNet Connector must be able to search for metadata connecting to a Broker		
OneNet_FUR_18	The OneNet Connector must be able to exchange data with other connectors using pull and/or push mechanisms	The data exchange process happens end-to-end exploiting pull or push mechanisms.	
OneNet_FUR_19	The OneNet system must be able to support the creation, management and usage of vocabularies	A feature provided by OneNet system is the Vocabulary Provider. It manages and offers vocabularies (i.e., ontologies, reference data models, or metadata elements) that can be used to annotate and describe datasets.	
OneNet_FUR_20	The OneNet participant could use vocabularies for creating and structuring its metadata		

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

OneNet_FUR_21	The OneNet system should offer data services/apps for data processing and transformation	One of the main features of the OneNet system is the possibility to enrich, transform, validate and harmonize the data processed. In addition, the OneNet allow to log all the data transaction.	
OneNet_FUR_22	The OneNet system should be able to log any data transaction between any OneNet participant		
OneNet_FUR_23	The OneNet system should be able to assess the quality of data processed		
OneNet_FUR_24	The OneNet system should be able to perform a semantic validation of the data processed		
OneNet_FUR_25	The OneNet system could use AI mechanism for empowering Data services	For improving the Data Services offered by the OneNet system, some AI mechanism could be implemented.	
OneNet_FUR_26	The OneNet system should be able to integrate any kind of data sources using Context Broker	The usage of the FIWARE context broker could facilitate the integration of any kind of data source, using a standard API based approach.	GSUC_03

2.5 OneNet Interfaces

It was necessary to define for the overall OneNet System and in particular for the OneNet Decentralised Middleware, many interfaces (internals and externals) for the integration of the external platforms and components as well as the provisioning of the OneNet Data services. The Figure 5 represents the interfaces schema of the OneNet Connector and Middleware.

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

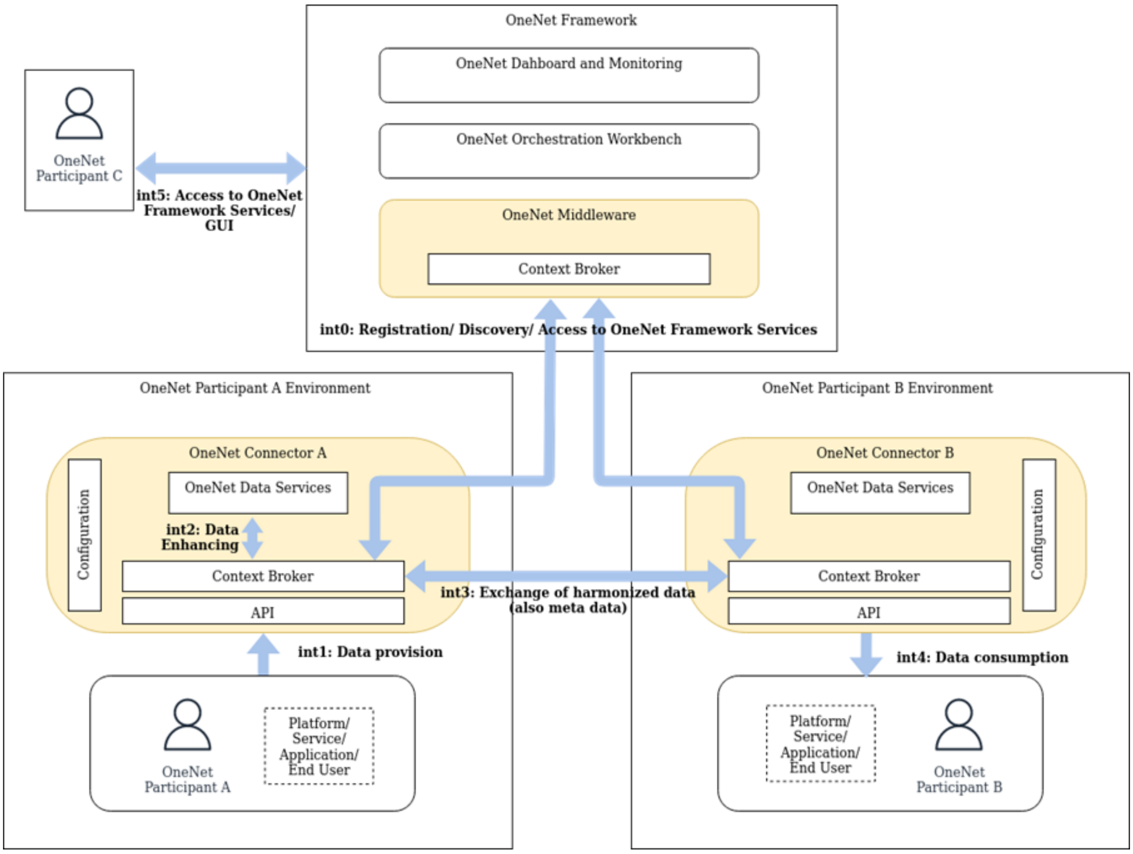

OneNet Framework
OneNet Dahboard and Monitoring
OneNet Orchestration Workbench
OneNet Middleware
Context Broker
OneNet Participant C
int5: Access to OneNet Framework Services/ GUI
int0: Registration/ Discovery/ Access to OneNet Framework Services
OneNet Participant A Environment
OneNet Connector A
Configuration
OneNet Data Services
int2: Data Enhancing
Context Broker
API
int1: Data provision
OneNet Participant A
Platform/ Service/ Application/ End User
int3: Exchange of harmonized data (also meta data)
OneNet Participant B Environment
OneNet Connector B
OneNet Data Services
Configuration
Context Broker
API
int4: Data consumption
Platform/ Service/ Application/ End User
OneNet Participant B

Figure 5: Interfaces schema of the OneNet Connector and Middleware

The complete list of the interfaces was defined in the D5.3 [3] (for the OneNet Decentralised Middleware and OneNet Connector) and in the D5.4 [4] (for the OneNet Orchestration Workbench and OneNet Monitoring and Analytics Dashboard).

Table 2 and Table 3 below report respectively the interfaces for the OneNet Decentralised Middleware and Connector and for the OneNet Orchestration Workbench and OneNet Monitoring and Analytics Dashboard identified and to be implemented in the OneNet System.

Table 2: Identified Interfaces for OneNet Decentralised Middleware and Connector

Interface ID	From	To	Bidirectional	Description
int0	OneNet Connector (NGSI broker)	OneNet Middleware (NGSI broker)	Yes	Interface between OneNet Middleware and the OneNet Participant's connector for data and meta-data exchange e.g., register data information availability, contact with Certification Body, Evaluation

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

				facilities, Dynamic Attribute Provisioning Service, or access to OneNet Framework Services (optional)
int1	OneNet Participant (data source/ data provider)	OneNet Connector (NGSI broker)	No	Interface between data provider platform and local OneNet Connector: OneNet network of platforms access for data provision (OneNet API)
int2	OneNet Connector broker (NGSI broker)	OneNet Data Services	Yes	Interface between OneNet Connector broker and OneNet Data Services
int3	OneNet Connector (NGSI broker)	OneNet Connector (NGSI broker)	Yes	Interface between local OneNet Connectors (data consumer's – data provider's) for data exchange and meta-data exchange
int4	OneNet Connector (NGSI broker)	OneNet Participant (data exchange/ consuming data)	No	Interface between OneNet Participant's local OneNet Connector (broker) and data consumer platform: OneNet network of platforms access for data exchange/ consuming data
int5	OneNet Participant (User)	OneNet Framework	Yes	Interface between OneNet Participant (user account) and OneNet Framework through OneNet Middleware: OneNet network of platforms access for non-automated data exchanges and graphical user interface

Table 3: Identified Interfaces for OneNet Orchestration Workbench and OneNet Monitoring and Analytics Dashboard

Interface ID	From	To	Type	Description
int0b	OneNet Orchestration Workbench	OneNet Middleware (NGSI broker)	Broker/API	Interface between OneNet Middleware and the OneNet Orchestration Workbench. The OneNet Orchestration Workbench should exploit FIWARE Context Broker implementation for integrating data sources and services.
int1b	OneNet Middleware	OneNet Orchestration Workbench	Open API	Interface offered by OneNet Orchestration Workbench for including additional services in the OneNet Decentralized Middleware (e.g., register news services or retrieving Service Catalogue information).
Int2b	OneNet Participant (User)	OneNet Monitoring and Analytics Dashboard	GUI	Graphical Interfaces for accessing, configure and utilize the OneNet Monitoring and Analytics Dashboard
Int3b	OneNet Participant (User)	OneNet Monitoring and Analytics Dashboard	Open API	Standard OpenAPI interfaces for exploiting the services offered by the OneNet Monitoring and Analytics Dashboard in automated way
Int4b	OneNet Participant (User)	OneNet Orchestration Workbench	Open API	Standard OpenAPI interfaces for exploiting the services offered by the OneNet Orchestration Workbench in automated way

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

3 OneNet Middleware Information Model

3.1 Introduction to OneNet information model approaches (CIM, domain-specific)

The OneNet information model will be a NGSI-LD information model raises a hybrid solution using both 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 details the definition of the NGSI-LD model.

3.1.1 Definition

NGSI-LD 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** [1] (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". 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**[2] 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**[3] 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 on the basis of RDF and the semantic web framework. It can be serialized using JSON-LD. Every entity and relationship are given a 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).

[1] https://www.etsi.org/
[2] https://en.wikipedia.org/wiki/Open_Mobile_Alliance
[3] https://www.fiware.org/

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

- Properties (similar to 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. [1]

3.1.2 Architecture

The NGSI-LD specification consists of an information model and an API. The API provides functionalities to support the architectural roles described in the following Figure:

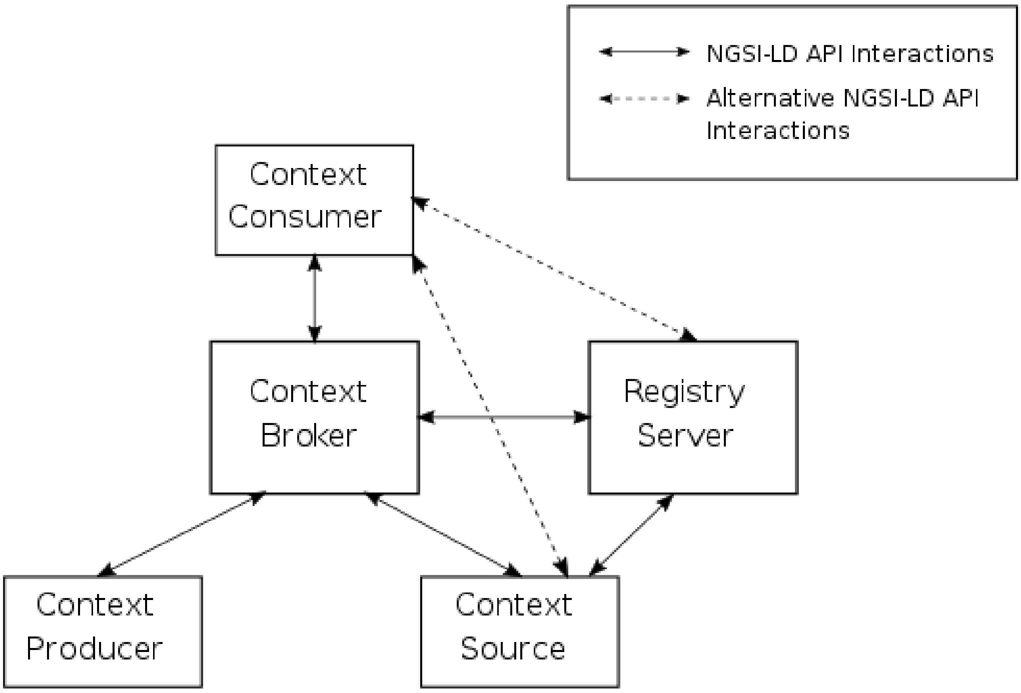


Figure 6: NGSI-LD Architecture Interactions

- **Context Consumer**: A Context Consumer consumes NGSI-LD Entities from a Context Broker (or possibly directly from a Context Source) using the Context Information Consumption functionalities of the NGSI-LD API.It can retrieve a specific NGSI-LD Entity or query relevant NGSI-LD Entities using synchronous requests. It can also subscribe for relevant NGSI-LD Entities and receive asynchronous notifications whenever there are changes in the requested NGSI-LD Entities.

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

- **Context Producer**: A Context Producer creates, updates and deletes NGSI-LD Entities, NGSI-LD Properties and NGSI-LD Relationships in the Context Broker using the Context Information Provision functionalities of the NGSI-LD API.
- **Context Source**: A Context Source makes NGSI-LD Entities available through the Context Information Consumption functionalities of the NGSI-LD API. To make the information discoverable for a Context Broker, it registers the kind of context information it can provide with a Registry Server using the Context Source Registration functionality of the NGSI-LD API.
- **Context Broker**: A Context Broker acts as the primary access points to context information for Context Consumers. NGSI-LD Entity information can be stored by the Context Broker itself, if it has been provided by a Context Producer using the Context Information Provision functionalities of the NGSI-LD API, or the Broker can request is from Context Sources using the Context Information Consumption functionalities of the NGSI-LD API. The Context Broker aggregates all NGSI-LD Entity information related to a request and returns the aggregated result to the Context Consumer. In the case of a subscription, it sends notifications whenever there are relevant changes, potentially as a result of receiving notifications from Context Sources. To find Context Sources that may have NGSI-LD Entities relevant to a Context Consumer request, the Context Broker uses the Context Source Discovery functionality of the NGSI-LD API implemented by the Registry Server.
- **Registry Server**: The Registry Server stores Context Source Registrations provided by Context Sources using the Context Source Registration functionalities of the NGSI-LD API. Context Source Registrations contain information about what *kind* of Context Information a Context Source can provide, but not actual values. The kind of context information can be provided on different granularity levels ranging from very detailed information, e.g. certain properties or relationships of a specific NGSI-LD Entity, to any information of a specific NGSI-LD Entity, or to the level that it can provide NGSI-LD Entities that have a certain Entity Type, possibly for a given geographic area. The Context Source Discovery functionality of the NGSI-LD API allows the Context Broker (or possibly a Context Consumer) to find Context Sources that may have relevant NGSI-LD Entities.

The architectural roles allow the implementation of different deployment architectures. In a centralized architecture, there is a central Context Broker that stores the context information provided by Context Producers. In a distributed setting, all context information can be stored by Context Sources. In a federated architecture, Context Sources can again be Context Brokers that make aggregated information from a lower hierarchy level available. These architectures are not mutually exclusive, i.e. an actual deployment may combine them in different ways. [1]

3.1.3 NGSI-LD Information Model Structure

The NGSI-LD Information Model prescribes the structure of context information that shall be supported by an NGSI-LD system. It specifies the data representation mechanisms that shall be used by the NGSI-LD API itself. In addition, it specifies the structure of the Context Information Management vocabularies to be used in conjunction with the API.

The NGSI-LD Information Model is defined at two levels (see Figure 7): the foundation classes which correspond to the Core Meta-model and the Cross-Domain Ontology. The former amounts to a formal specification of the "property graph" model. The latter is a set of generic, transversal classes which are aimed at avoiding conflicting or redundant definitions of the same classes in each of the domain-specific ontologies. Below these two levels, domain specific ontologies or vocabularies can be devised. For instance, the SAREF Ontology ETSI TS 103 264 can be mapped to the NGSI-LD Information Model, so that smart home applications will benefit from this Context Information Management API specification.

The version of the cross-domain model proposed by ETSI is a minimal one, aimed at defining the classes used in the current release of the API specification. It has been extended by other work items like ETSI GS CIM 006, with classes defining extra concepts such as mobile vs. stationary entities, instantaneous vs. static properties, etc. [6], [7], [8], [9]

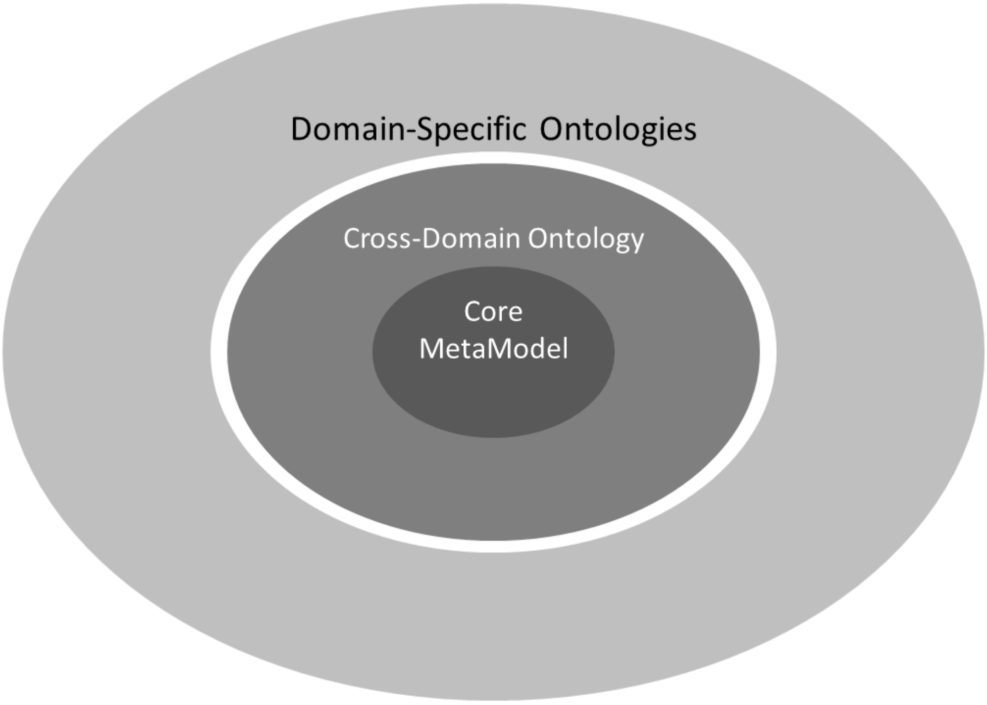


Figure 7 : Overview of the NGSI-LD Information Model Structure

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

3.1.4 NGSI-LD Meta Model

The NGSI-LD meta-model formally defines these foundational concepts (Entities, Relationships, Properties) on the basis of RDF/RDFS/OWL, and partially on the basis of JSON-LD.

- An NGSI-LD **Entity** is the informational representative of something (a *referent*) that is supposed to exist in the real world, outside of the computational platform using NGSI-LD. This referent need not be something strictly physical (it could be a legal or administrative entity), nor self-contained (it may be a distributed system-level construct). Any instance of such an entity is supposed to be uniquely identified by an IRI, and characterized by reference to one or more NGSI-LD Entity Type(s). In property-graph language, it is a node.
- An NGSI-LD **Property** is an instance that associates a characteristic, an NGSI-LD Value, to either an NGSI-LD Entity, an NGSI-LD Relationship or another NGSI-LD Property. Properties of properties are explicitly allowed and are encouraged e.g. to express the accuracy of a particular measured value.
- An NGSI-LD **Relationship** is a directed link between a subject (starting point), that may be an NGSI-LD Entity, an NGSI-LD Property, or another NGSI-LD Relationship, and an object (end-point), that is an NGSI-LD Entity. A NGSI-LD Relationship from a Property to an Entity can for example be used to express that the Property was measured by that Entity (Provenance of the measurement).
- An NGSI-LD **value** is a JSON value (i.e. a string, a number, true or false, an object, an array), or a JSON-LD typed value (i.e. a string as the lexical form of the value together with a type, defined by an XSD base type or more generally an IRI), or a JSON-LD structured value (i.e. a set, a list, or a language-tagged string).
- An NGSI-LD **type** is an OWL class that is a subclass of either the NGSI-LD Entity, NGSI-LD Relationship, NGSI-LD Property or NGSI-LD Value classes defined in the NGSI-LD meta-model. NGSI-LD pre-defines a small number of types, but is otherwise open to any types defined by users.

Figure 8 provides a graphical representation of the NGSI-LD Meta-Model in terms of classes and their relationships. To provide additional clarity an informal (non-normative) mapping to the Property Graph Model is also presented.

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

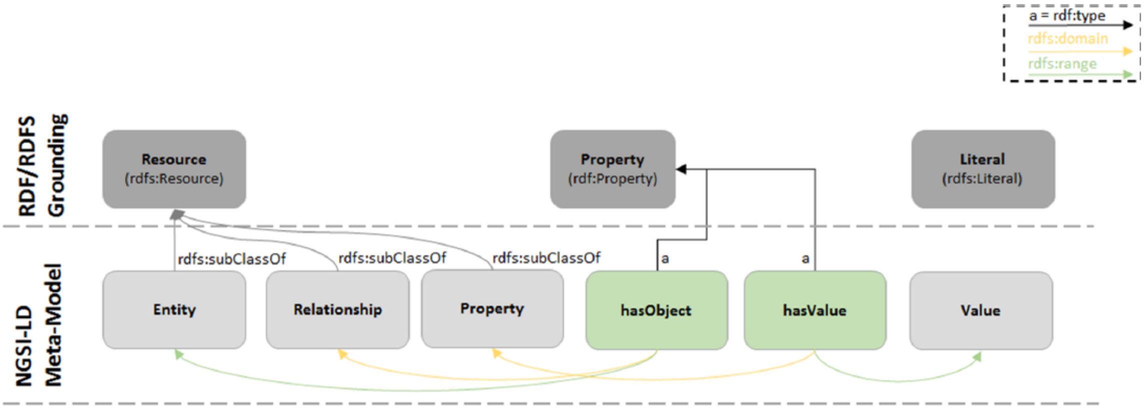


Figure 8: NGSI-LD Core Meta-Model

Implementations should support the NGSI-LD Meta-model as follows:

- An **NGSI-LD Entity** is a subclass of rdfs:Resource.
- An **NGSI-LD Relationship** is a subclass of rdfs:Resource.
- An **NGSI-LD Property** is a subclass of rdfs:Resource.
- An **NGSI-LD Value** shall be either a rdfs:Literal or a node object (in JSON-LD language) to represent complex data structures.
- An **NGSI-LD Property** shall have a **value,** stated through *hasValue*, which is of type rdf:Property.
- An **NGSI-LD Relationship** shall have an **object** stated through *hasObject* which is of type rdf:Property. [1], [6], [10]

3.1.5 Cross Domain Ontology

Complementing this metamodel, the NGSI-LD information model specification also provides a **cross-domain ontology** that defines key constructs related to spatial, temporal or system-composition characteristics of entities. The flexible information model allows the specification of any kind of entity. In order to allow interoperability between NGSI-LD users, standardized entities are collaboratively defined at Smart Data Models Program and made available at its repository with an open-source license.

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

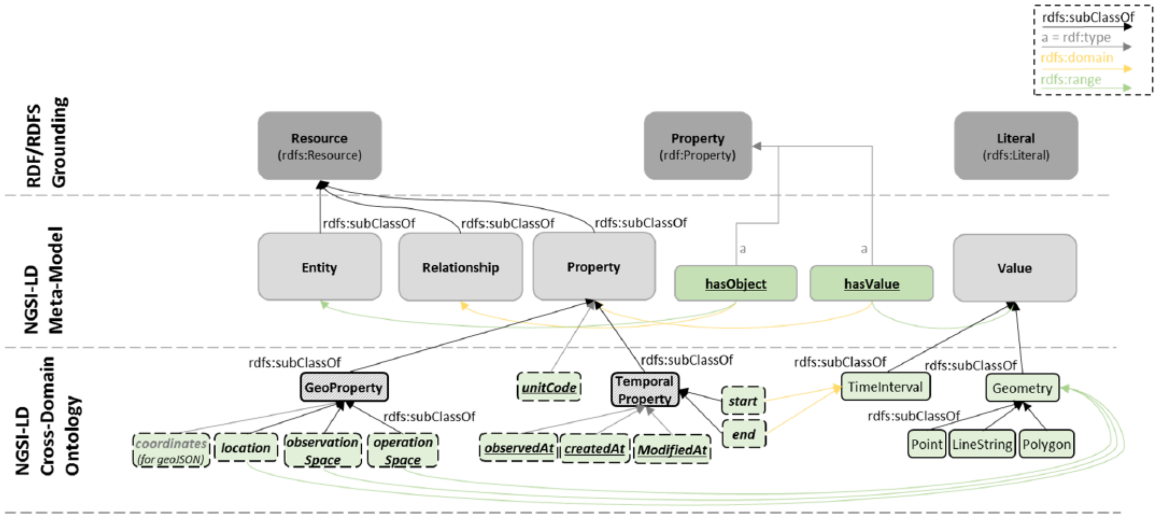


Figure 9: NGSI-LD Core Meta-Model plus the Cross-Domain Ontology

Figure 9 describes the concepts introduced by the NGSI-LD Cross-Domain Ontology, which shall be supported by implementations as follows:

- **Geo Properties:** Are intended to convey geospatial information.
- **Temporal Properties:** Are intended to convey temporal information.
- **"unitCode" Property:** A Property intended to provide the units of measurement of an NGSI-LD Value.
- **Geometry Values:** They are a special type of NGSI-LD Value intended to convey geometries corresponding to geospatial properties.
- **Time Values:** They are a special type of NGSI-LD Value intended to convey time instants or intervals representations. [1][6]

3.1.6 NGSI-LD domain-specific models and instantiation

Figure 10 intends to illustrate the relationship between the NGSI-LD Information Model and NGSI-LD Domain-specific models by showing an example of an NGSI-LD domain-specific model. Domain-specific models introduce the specific entity types required for a particular domain. Figure 10 shows the types *Car*, *Parking, Street, Gate*. Entity types can have further subtypes, e.g. *OffStreetParking* as subtype of *Parking*.

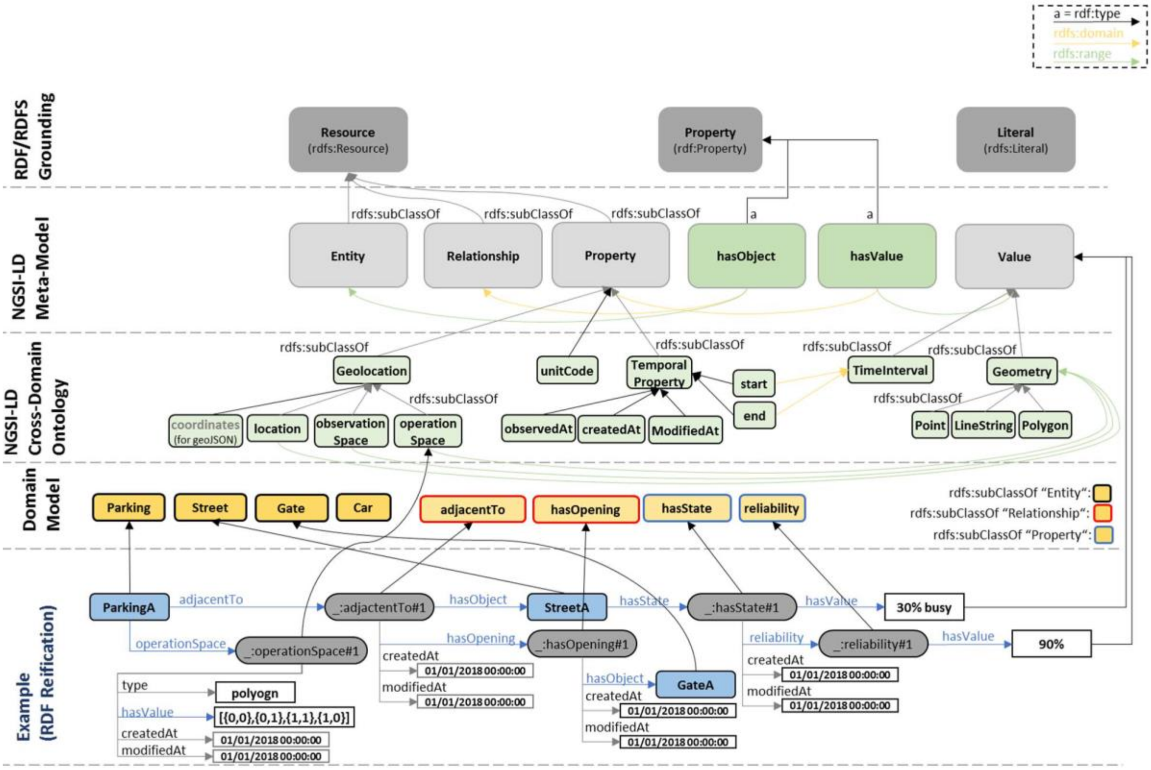


Figure 10: Cross-Domain Ontology and instantiation

In addition, two different NGSI-LD Properties are introduced ('*hasState*', '*reliability*'). The '*adjacentTo*' Relationship links entities of type '*Parking*' with entities of type '*Street*'. [6]

The definition of domain specific data models is an abstract model that organises elements of data and standardises how they relate to one another and to the properties of real-world entities. They play a crucial role because they define the harmonised representation formats and semantics that will be used by applications both to consume and to publish data. The adoption of Standardized Data Models is fundamental for facilitating interoperability within the community.

The FIWARE Foundation together with TMForum, and IUDX, have launched the Smart Data Models initiative where data models are made available for the benefit of all. In it FIWARE Data Models have been harmonised to enable data portability for different application domains including, Smart Cities, Smart Agrifood, Smart Environment, Smart Sensoring, Smart Energy, Smart Water and others 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. More information and smart applications on Smart Energy Data Models and implications for potentials are described D5.4.

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

3.1.7 UML representation

This section is informative and is intended to show how the NGSI-LD information model could be described using UML diagrams. The aim of this diagram is to help those readers less familiar with ontology representations or RDF to understand the NGSI-LD Information Model. In Figure 11 NGSI-LD Entity, Relationship, Property and Value are represented as UML classes. UML associations are used to interrelate these classes while keeping the structure and semantics defined by the NGSI-LD Information Model. [6]

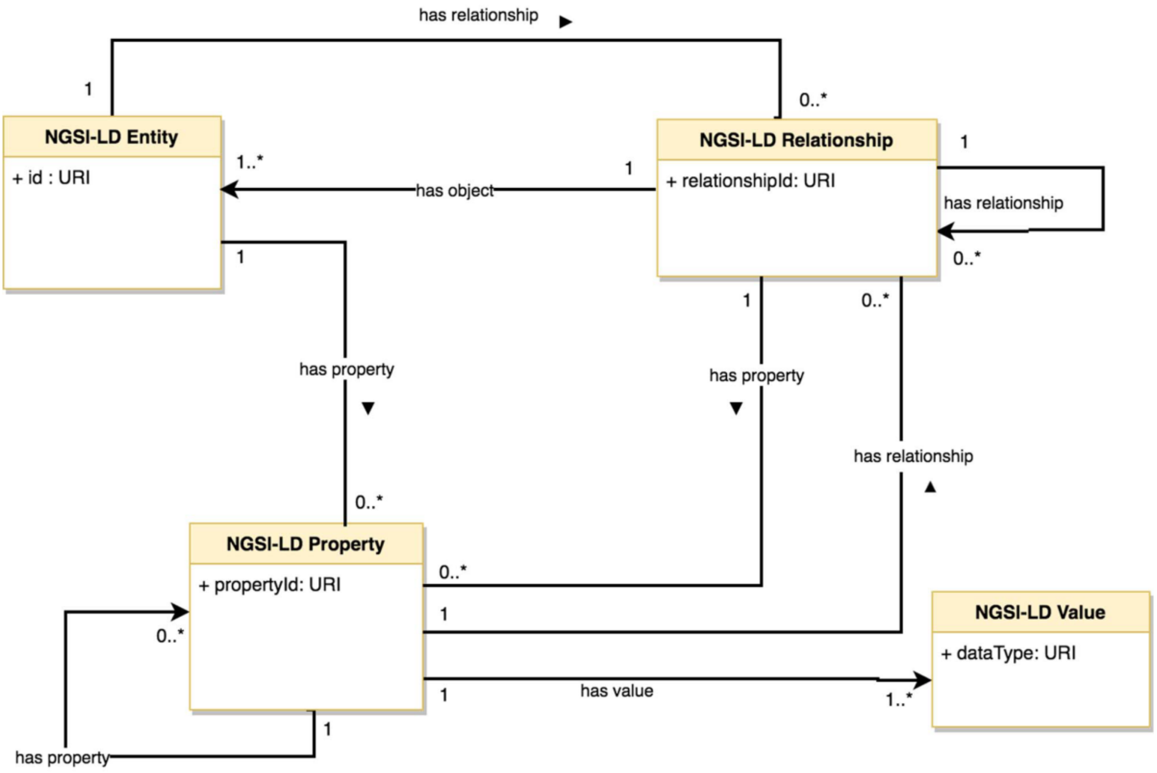


Figure 11: NGSI-LD information model as UML

3.1.8 Core NGSI-LD @context

NGSI-LD is based on JSON-LD, a JSON-based format to serialize Linked Data. The @context in JSON-LD is used to map terms provided as strings to concepts specified as URIs. The Core NGSI-LD (JSON-LD) @context is defined as a JSON-LD @context which contains:

- The core terms needed to uniquely represent the key concepts defined by the NGSI-LD Information Model.
- The terms needed to uniquely represent all the members that define the API-related Data Types

NGSI-LD compliant implementations should support such Core @context, which shall be implicitly present when processing or generating context information. Nonetheless, when rendering NGSI-LD Elements, the Core @context shall always be referenced, so that, if needed, JSON-LD processors can properly expand the resulting JSON-LD documents provided by API implementations. The NGSI-LD @context is publicly available at [6], [11]

3.2 Cross-Platform Energy Data Exchange for market-based flexibility management (domain-agnostic: IDS based/FIWARE information model)

3.2.1 Definition

The IDS Information Model is an RDFS/OWL-ontology covering the fundamental concepts of the International Data Spaces (IDS), i.e. the types of digital contents that are exchanged by participants by means of the IDS infrastructure components. The ontology and its documentation are published at https://w3id.org/idsa/core. The model development is led by the Fraunhofer Institutes for Applied Information Technology FIT and Intelligent Analysis and Information Systems IAIS with support by members of the International Data Spaces Association in the context of the Information Model sub-working group (SWG4). The group is chaired by Christoph Lange (Fraunhofer FIT) and Sebastian Tramp (eccenca GmbH).

The model development is based on GitHub, following a defined branching model. Contributions and community feedback are maintained via the GitHub ticketing system. The release process is aligned with the International Data Spaces Association architecture working group meetings, i.e. there are roughly 2 releases scheduled per year with intermediary updates to the development branch. The current release version is 4.1.0, with the latest revision 4.1.0. The Information Model and associated resources published on GitHub are available under the Apache License 2.0. [12]

3.2.2 Scope

The Information Model primarily aims at describing, publishing and detecting data products (Data Assets) and reusable data processing software (Data Apps) in the Industrial Data Space. Data Assets and Data Apps are the core resources of the Industrial Data Space, and are hereinafter referred to as resources. By means of a structured semantic annotation it is ensured only relevant resources are provided (i.e., resources appropriate to meet the requirements of the Data Consumer). Once the resources are identified, they can be exchanged and consumed via semantically defined service interfaces and protocol bindings in an automated way. Apart from those core commodities, the Information Model describes essential properties of Industrial Data Space entities, its participants, its infrastructure components, and its processes.

The Information Model is a generic model, with no commitment to any particular domain. Domain modeling is delegated to shared vocabularies and data schemata, as provided e.g. by domain-specific communities of the

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

Industrial Data Space. The Information Model does not provide a meta-model for defining custom structured datatypes comparable to the OData or OPC-UA standards. Considerations beyond the scope of modeling digital assets and their interchange are considered out-of-scope. The Information Model does not deal with the side effects of data exchange (on Data Consumer's side), for example in scenarios where data is used for real-time machine control. RPC (remote procedure call) semantics of data messages is also not covered by the Information Model. [13]

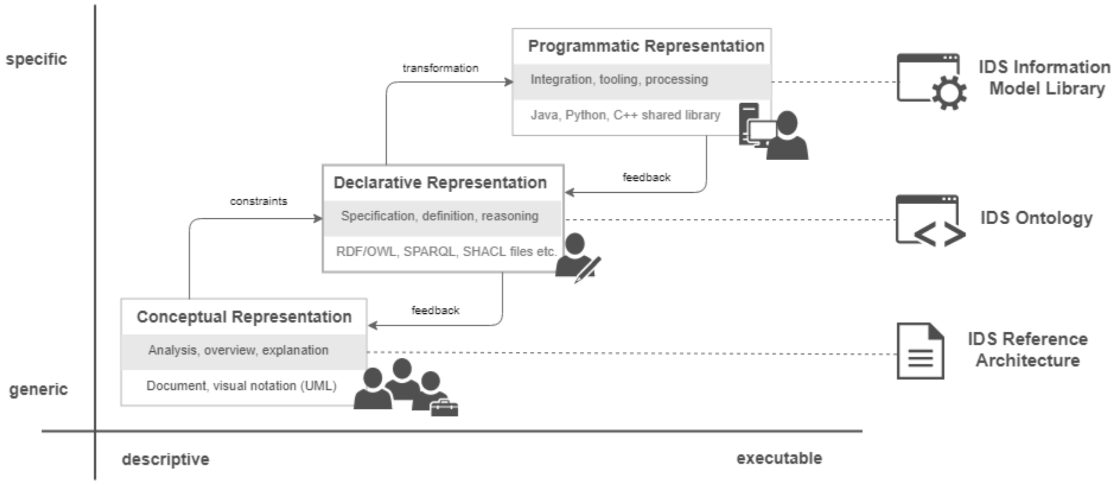

specific
generic
descriptive
executable
Programmatic Representation
Integration, tooling, processing
Java, Python, C++ shared library
Declarative Representation
Specification, definition, reasoning
RDF/OWL, SPARQL, SHACL files etc.
Conceptual Representation
Analysis, overview, explanation
Document, visual notation (UML)
transformation
constraints
feedback
feedback
IDS Information Model Library
IDS Ontology
IDS Reference Architecture

Figure 12: Representations of the Information Model

3.2.3 IDS Key Principles

Seamless collaboration and information exchange are the foundations of digital business models. Huge internet-based platforms have emerged, connecting people around the world and exchanging information in unprecedented speed. While end-users got used to such convenient communication and data exchange in their private interactions, they expect similar characteristics in their professional environment. However, data exchange in business-to-business relations faces a significant amount of still unresolved challenges. One example is the typical dilemma of digital strategies – sharing valuable data involves the risk of losing the company’s competitive advantage, whereas not participating prevents innovative business models and undermines upcoming revenue opportunities. There is currently no standardized, widely accepted means for a trustful exchange of business data that ensures traceability, data owner’s privacy and sovereignty. Privacy concerns and protection of proprietary information are critical factors of future data infrastructures. Such an infrastructure is a key prerequisite for a secure, standardized and fine-grained sharing of sensitive business data, unlocking the potential for novel value creation chains and the inception of intermediation platforms. The International Data Spaces initiative (IDS; formerly “Industrial Data Space”) targets the requirements mentioned above by promoting a standard for virtual data spaces for reliable data exchange among business partners. To achieve the goal of sovereign data exchange, aspects of data management, semantic data integration, and

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

security have to be addressed. The IDS propose a message-based approach to bridge syntactic differences. Still, a successful exchange of data objects requires sufficient understanding of its content and meaning. A shared information model is therefore needed. The IDS Information Model (IDS IM) is an RDFS/OWL ontology (as mentioned in 4.2.1), which defines the general concepts depicted in Figure 13 along with roles required to describe actors, components, roles and interactions in a data space. This ontology serves two purposes, (1) as a catalogue of machine-readable terms and data schema for IDS components and (2) as a shared language for all stakeholders. Each involved player needs to understand and be able to interpret this set of terms, thus enabling semantic interoperability in federated environments. The IDS IM therefore presents the backbone and common denominator for the data-sovereign ecosystem as envisioned by the IDS. [14]

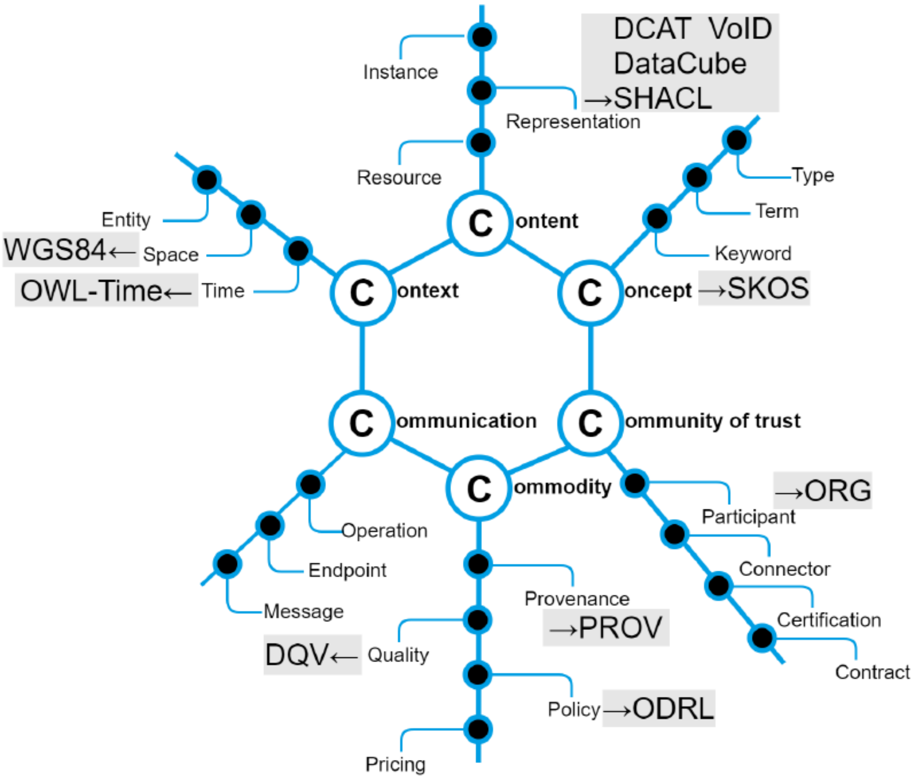


Figure 13: Partitions of the ontology by concern (pointing to standards reused)

3.2.4 Governance and Context of the IDS Information Model

The IDS has been designed in a systematic process with broad involvement of industrial stakeholders. Its specification and reference implementations are maintained and supported by the International Data Spaces Association (IDSA), a non-profit organization to disseminate and evolve the IDS views and principles.

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

The IDSA, with more than 100-member organizations meanwhile, serves as the institutional body for promoting the IDS in research projects and industrial applications. In particular, the IDSA ensures the sustainability of the ontology and provides the resources for future extensions.

The IDS Reference Architecture Model (RAM) defines the roles assumed and the responsibilities of organizations interacting in a data space. Figure 14 shows, for a broad initial overview, the core interactions and roles in the IDS Data Providers exchange messages with Data Consumers via standardized software interfaces, and use multiple services to support this. They can, for example, publish metadata about resources to a directory ("broker") and thus allow others to find these. At the heart of every IDS interaction is the adherence to the usage rules – accomplished by the connection of machine-readable usage policies with each interaction and the application of certified, trustworthy execution environments. The so-called IDS Connectors interpret and enforce the applied policies, thus creating a federated network for a trustworthy data exchange.

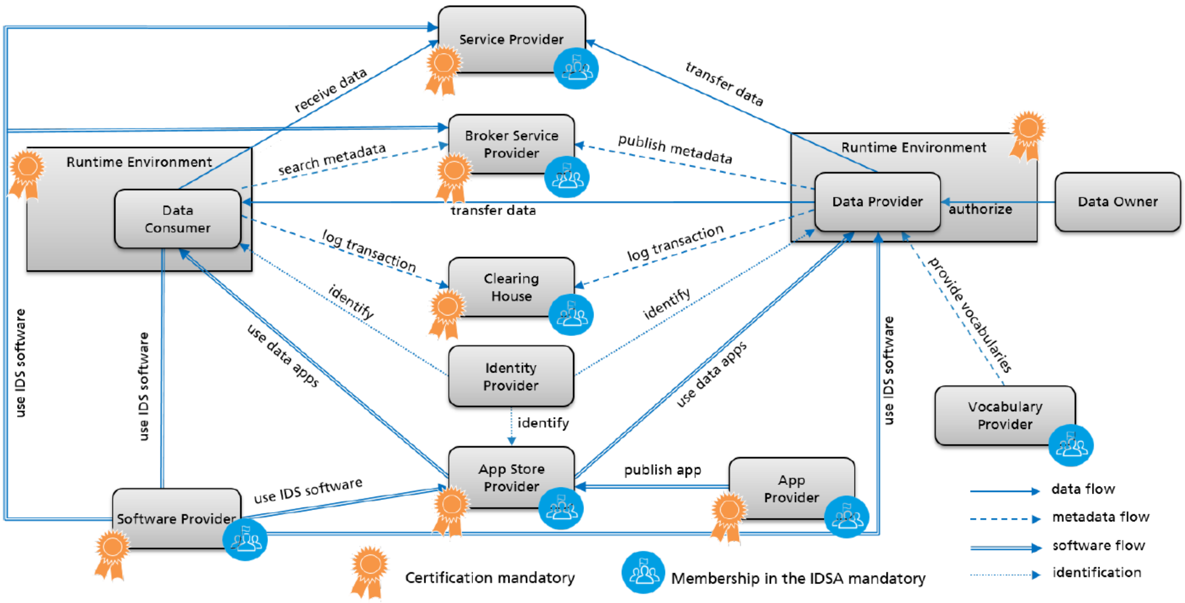


Figure 14: IDS Reference Architecture with its main roles and interactions.

The IDS IM specifies the domain-agnostic common language of the IDS. The IM is the essential agreement shared by the participants and components of the IDS, facilitating compatibility and interoperability. It serves the stakeholders' requirement "that metadata should not be limited to syntactical information about data, but also include data ownership information, general usage conditions, prices for data use, and information about where and how the data can be accessed" by supporting the description, publication and identification of

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

(digital) resources. It is, like other elementary IDS software components, available as open source to foster adoption (Table 4). The ontology, the normative implementation of the declarative UML representation in the IDS RAM, was originally created in 2017 and first released in 2018.

Table 4: Key facts about the IDS Information Model and related resources.

General	**License Size**	Apache License 2.0
	Size	278 classes, 149 object properties, 115 data properties, 684 individuals
	Total size	3912 triples
Reuse	**Reused ontologies**	CC, DCAT, DCMI Terms, FOAF, ODRL, OWL-Time, VoID, etc.
Documentation	**Ontology documentation**	https://w3id.org/idsa/core/
	Element description	Using rdfs:label, rdfs:comment
Availability	**Namespace**	ids: https://w3id.org/idsa/core/ idsc: https://w3id.org/idsa/code/
	Serialisations	Turtle, RDF/XML, JSON-LD, N-Triples
	GitHub	https://github.com/International-Data-Spaces-Association/InformationModel/
	VoCol Instance	https://vocol.iais.fraunhofer.de//ids/

Table 5: Namespace Declarations

ids	<https://w3id.org/idsa/core>
idsm	<https://w3id.org/idsa/metamodel>
code	<https://w3id.org/idsa/code>
dcam	<http://purl.org/dc/dcam>
ns	<http://creativecommons.org/ns>
owl	<http://www.w3.org/2002/07/owl>
freq	<http://purl.org/cld/freq>
xsd	<http://www.w3.org/2001/XMLSchema>
schema-org	<https://schema.org>

skos	<http://www.w3.org/2004/02/skos/core>
rdfs	<http://www.w3.org/2000/01/rdf-schema>
rfc3986	<https://tools.ietf.org/html/rfc3986>
shacl	<http://www.w3.org/ns/shacl>
docs	<https://postgis.net/docs>
rfc7519	<https://tools.ietf.org/html/rfc7519>
dcterms	<http://purl.org/dc/terms>
wgs84	<http://www.w3.org/2003/01/geo/wgs84_pos>
holdings	<https://www.loc.gov/marc/holdings>
dcat	<http://www.w3.org/ns/dcat>
locn	<http://www.w3.org/ns/locn>
vann	<http://purl.org/vocab/vann>
foaf	<http://xmlns.com/foaf/0.1>
geonames	<http://www.geonames.org/ontology>
spec-md	<https://github.com/cloudevents/spec/blob/master/spec.md>
void	<http://rdfs.org/ns/void>
org	<http://www.w3.org/ns/org>
resource	<http://dbpedia.org/resource>
voaf	<http://purl.org/vocommons/voaf>
url	<https://www.gnu.org/software/emacs/manual/html_node/url>
geosparql	<http://www.opengis.net/ont/geosparql>
rdf	<http://www.w3.org/1999/02/22-rdf-syntax-ns>
time	<http://www.w3.org/2006/time>
odrl	<http://www.w3.org/ns/odrl/2>

3.2.5 International Data Spaces Information Model: Overview

This ontology has the following classes and properties as shown in [13]

Table 6: Ontology Classes

Classes	Object Properties	Data Properties
Abstract Constraint	Accrual periodicity	access URL
Access Token Response	action	Accrual periodicity
AccessToken Request Message	action refinement	app documentation
Action	affected AppResource	app endpoint port
Agent	affected Connector	asset source
App Available Message	affected Participant	auth password
App Delete Message	affected Resource	auth username
App endpoint type	and operand	authService
App execution resources	Annex to contract	byte size
App Notification Message	app artifact reference	checksum
App Registration Request Message	app endpoint	content standard
App Registration Response Message	app endpoint media type	contentVersion
App Representation	App endpoint type	Contract date
App Resource	app route	Contract end
App Route	app route end	Contract Rejection Reason
App Store	app route start	Contract start
App Unavailable Message	asset refinement	corporateEmailAddress
App Upload Message	asset source	corporateHomepage
App Upload Response Message	assignee	created
Artifact	assigner	creation date
Artifact Request Message	aud	Custom License
Artifact Response Message	authInfo	Data type
Artifact State	Authorization token	date time

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

Asset	authService	description
Asset Collection	authStandard	duration
Audience	beginning	emailAddress
Audio Representation	broader LeftOperand	endpoint documentation
Audio Resource	certification level	endpoint information
Audit guarantee	component certification	exp
Authentication	configuration model log level	familyName
AuthInfo	connector Catalog	file name
AuthStandard	connector deploy mode	filename extension
Base Connector	Connector description	frame rate
BinaryOperator	Connector proxy	givenName
BoundingPolygon	Connector Status	has data
Broker	constraint	has duration
Catalog	Consumer	has PIP endpoint
Certification	consumer connector	height
Certification Level	content part	homepage
Clearing House	content type	http auth URI
Command Message	Contract document	iat
Component Certification	contract offer	inbound topic
Component Certification Level	curator	inboundModelVersion
Concept	data app information	iss
Configuration Model	Data Type Schema	issued
Connector	Default representation	key store

Connector Catalog	end	Key Value
Connector Certificate Granted Message	endpoint artifact	keyword
Connector Certificate Revoked Message	evaluation facility	last accessed
Connector Deploy Mode	exclusive or	last valid date
Connector Endpoint	extended guarantee	latitude
Connector Notification Message	geoPoint	longitude
Connector Status	has Agent	membership end
Connector Unavailable Message	has contract	model version
Connector Update Message	has data	modified
Connector-restricted Data Usage Agreement	has default endpoint	name
Connector-restricted Data Usage Offer	has endpoint	nbf
Connector-restricted Data Usage Request	has PIP endpoint	no proxy
Constraint	has state	outbound model version
Content type	has user	outbound topic
Contract	included certification level	path
Contract agreement	instance	phoneNumber
Contract Agreement Message	is included in	proxy URI
Contract offer	issuer connector	representation standard
Contract Offer Message	Key Type	requested Element
Contract Rejection Message	language	requesting application
Contract request	last accessed	Revocation Reason
Contract Request Message	leftOperand	route configuration
Contract Response Message	listed Connector	route deploy method
Contract Supplement Message	maintainer	route description

Custom Media Type	media type	sampling rate
DAT Payload	member	scope
DAT Request Payload	member participant	shapeGraph
Data representation	memberPerson	site address
Data Resource	obligation	source
DataApp	offered resource	Standard License
DataApp Endpoint	Operation Reference	sub
Delete After Interval Agreement	operator	temporal resolution
described	or operand	title
DescribedSemantically	participant catalog	tokenValue
Description Request Message	participant certification	transportCertsSha256
Description Response Message	participant refinment	trust store
Digital content	permission	unit
Distribute Encrypted Agreement	physicalLocation	usage duration
Distribute Encrypted Offer	post-duty	version
Distribute Encrypted Request	pre-duty	width
Duration	primarySite	
Duration Agreement	prohibition	
Duty	Provider	
Dynamic Attribute Provisioning Service	proxy Authentication	
Dynamic Attribute Token (DAT)	Public Key	
end to end route		
	publisher	
Endpoint		
	queryLanguage	
Evaluation Facility		

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

	queryScope	
event		
	recipient agent	
Event-restricted Data Usage Agreement		
	recipient connector	
Event-restricted Data Usage Offer		
	recipient scope	
Event-restricted Data Usage Request		
	referringConnector	
frequency		
	rejectionReason	
Frequency		
	representation	
GeoFeature		
	Requested Artifact	
Geometry		
	Requested Participant	
GeoPoint		
	requested resource	
HTTP Authentication		
	requesting application	
IANA Media Type		
	resource catalog	
Identity provider		
	resource endpoint	
Image Representation		
	resource part	
Image Resource		
	rightOperand	
InfrastructureComponent		
	rightOperandReference	
Instant		
	sample	
Integrity protection and verification		
	Security guarantee	
Integrity verification scope		
	Security token	
Interval		
	securityProfile	
Interval Usage Agreement		
	sender agent	

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

4 Specification of OneNet APIs

Our focus in this chapter is to further elaborate on the API standards which OneNet implementation will be based on, by providing an overview of the two different initiatives of OneNet Reference Architecture from a technical point of view (e.g. IDSA, FIWARE NGSI-LD).

4.1 Overview of the NGSI-LD APIs technical specification

Chapter 4.1 defines the resources and operations of the NGSI-LD API according to the **Context Information Management (CIM) ETSI Industry Specification Group (ISG)**[4] which will serve as an implementation guideline. The NGSI-LD API [20] is structured in terms of HTTP [15], [16] verbs, input and output payloads. A non-normative OAS specification [3] of the referred HTTP binding can be found at [18].

4.1.1 Global definitions and resource structure

All resource URIs of this API have the following root:

- {apiRoot}/{apiName}/{apiVersion}/

NOTE 1: The *apiRoot* for Context Source related aspects and the *apiRoot* for general Entity-related aspects can be different, e.g. the Context Source related aspects can be implemented by a Context Registry as shown for the distributed and federated architectures, whereas the Entity-related aspects would be implemented by a Context Broker.

NOTE 2: The apiRoot for Context Source related aspects and the apiRoot for general Entity-related aspects can be different than the apiRoot for temporal aspects, e.g. the temporal aspects can be implemented by an NGSI-LD subsystem specialized in historical data. The apiRoot includes the scheme ("http" or "https"), host and optional port, and an optional prefix string. The API shall support HTTP over TLS (also known as HTTPS - see IETF RFC 2818 [18]). TLS version 1.2 as defined by IETF RFC 5246 [19] is supported. HTTP without TLS is not recommended.

The apiName shall be set to "ngsi-ld" and the apiVersion shall be set to "v1". All resource URIs are defined relative to the above root URI. The structure of the resources under the root URI is shown in Figure 15 and methods defined on them are shown in Table 7.

[4] https://www.etsi.org/

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

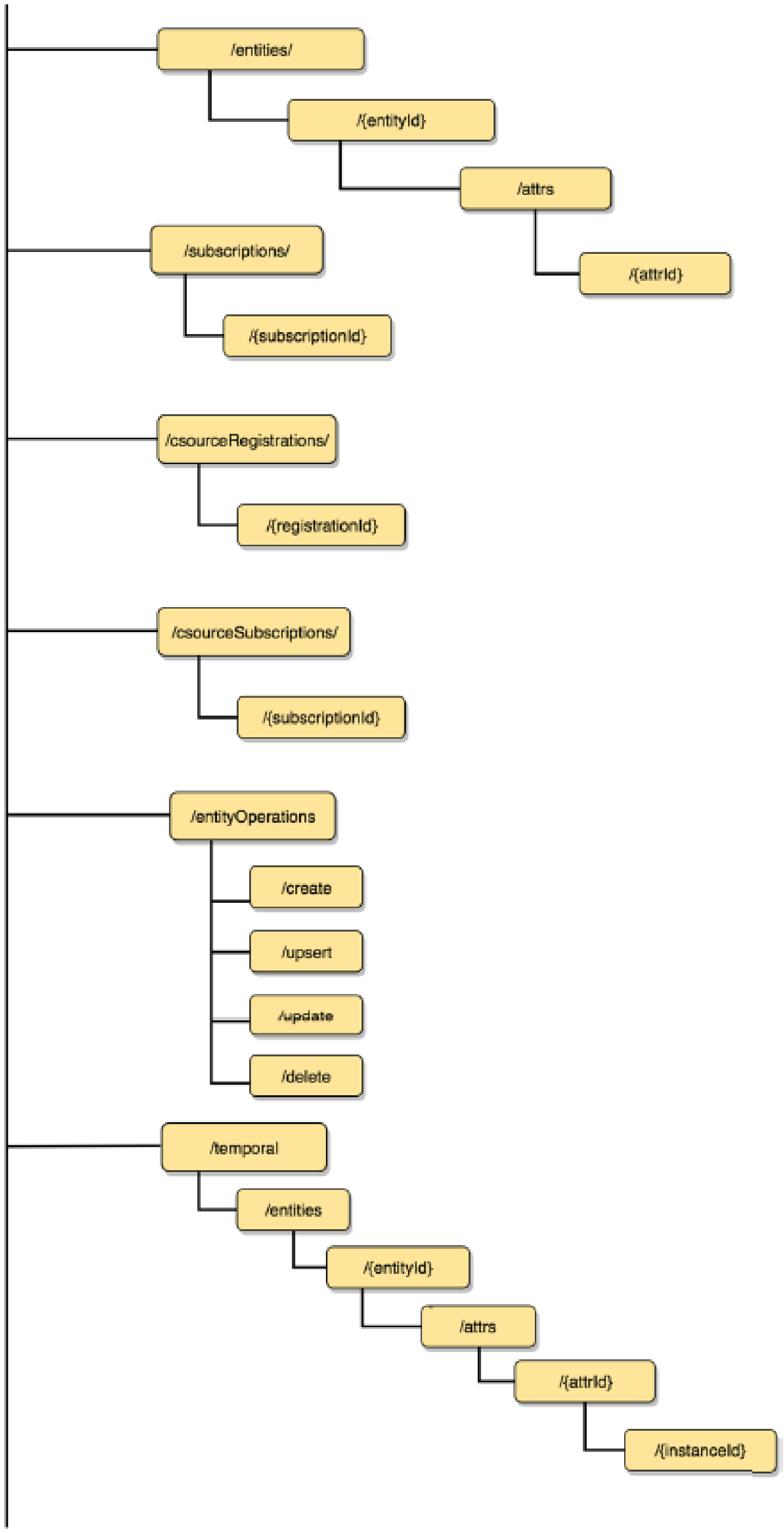


Figure 15: Resource URI Structure of NGSI-LD API

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

Table 7: Resources and HTTP methods defined on them

Resource Name	Resource URI	HTTP Method	Meaning
Entity List	/entities/	POST	Entity creation
		GET	Query entities
Entity by id	/entities/{entityId}	GET	Entity retrieval by id
		DELETE	Entity deletion by id
Entity Attribute List	/entities/{entityId}/attrs/	POST	Append entity Attributes
		PATCH	Update entity Attributes
Attribute by id	/entities/{entityId}/attrs/{attrId}	PATCH	Attribute partial update
		DELETE	Attribute delete
Subscriptions List	/subscriptions/	POST	Subscription creation
		GET	Subscription list retrieval
Subscription by Id	/subscriptions/{subscriptionId}	GET	Subscription retrieval by id
		PATCH	Subscription update by id
		DELETE	Subscription deletion by id
Context source registration list	/csourceRegistrations/	POST	Csource registration creation
		GET	Discover Csource registrations
Context source registration by Id	/csourceRegistrations/{registrationId}	GET	Csource registration retrieval by id
		PATCH	Csource registration update by id
		DELETE	Csource registration deletion by id
Context source Registration subscription list	/csourceSubscriptions/	POST	Csource registration subscription
		GET	Csource registration subscription list retrieval

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

Context source Registration subscription by Id	/csourceSubscriptions/{subscriptionId}	GET	Csource registration subscription retrieval by id
		PATCH	Csource registration subscription update by id
		DELETE	Csource registration subscription deletion by id
Entity Operations. Create	/entityOperations/create	POST	Batch Entity creation
Entity Operations. Upsert	/entityOperations/upsert	POST	Batch Entity create or update (upsert)
Entity Operations. Update	/entityOperations/update	POST	Batch Entity update
Entity Operations. Delete	/entityOperations/delete	POST	Batch Entity deletion
Entity Temporal Evolution	/temporal/entities/	POST	Temporal Representation of Entity creation
		GET	Query temporal evolution of Entities
Temporal Representation of Entity by id	/temporal/entities/{entityId}	GET	Temporal Representation of Entity retrieval by id
		DELETE	Temporal Representation of Entity deletion by id

Temporal Representation of Entity Attribute List	/temporal/entities/{entityId}/attrs/	POST	Temporal Representation of Entity Attribute instance addition
Temporal Representation of Entity Attribute by id	/temporal/entities/{entityId}/attrs/{attrId}	DELETE	Attribute from Temporal Representation of Entity deletion

Temporal Representation of Entity Attribute Instance by id	temporal/entities/{entityId}/attrs/{attrId}
	/{instanceId}
		PATCH	Attribute Instance update

		DELETE	Attribute Instance deletion by instance id

4.1.2 Error Types

This chapter extends the API common behaviors to the particularities of the HTTP REST binding. Table 8 associates API error types with HTTP status codes as shown below [20].

Error Type	**HTTP status**
http://uri.etsi.org/ngsi-ld/errors/InvalidRequest	400
http://uri.etsi.org/ngsi-ld/errors/BadRequestData	400
http://uri.etsi.org/ngsi-ld/errors/AlreadyExists	409
http://uri.etsi.org/ngsi-ld/errors/OperationNotSupported	422
http://uri.etsi.org/ngsi-ld/errors/ResourceNotFound	404
http://uri.etsi.org/ngsi-ld/errors/InternalError	500
http://uri.etsi.org/ngsi-ld/errors/TooComplexQuery	403
http://uri.etsi.org/ngsi-ld/errors/TooManyResults	403

Table 8: Mapping of error types to HTTP status codes

In addition, implementations support specific errors of the HTTP binding:

- "Method Not Allowed" (405) which shall be raised when a client invokes a wrong HTTP verb over a resource.

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

- "Request Entity too large" (413) which shall be raised when the HTTP input data stream provided by a client was too large i.e. too many bytes.
- "Length required" (411) which shall be raised when an HTTP request provided by a client does not define the "Content-Length" HTTP header.
- "Unsupported Media Type" (415) which shall be raised when an HTTP request provided by a client contains a payload which it is not "application/json" nor "application/ld+json". [20]

4.1.3 Resource: entities/

This resource represents a collection of entities known to an NGSI-LD system.

Resource URI: /entities/

Resource methods:

1. POST

This method is bound to the operation "Create Entity", taking the entity to be created from the HTTP request input payload.

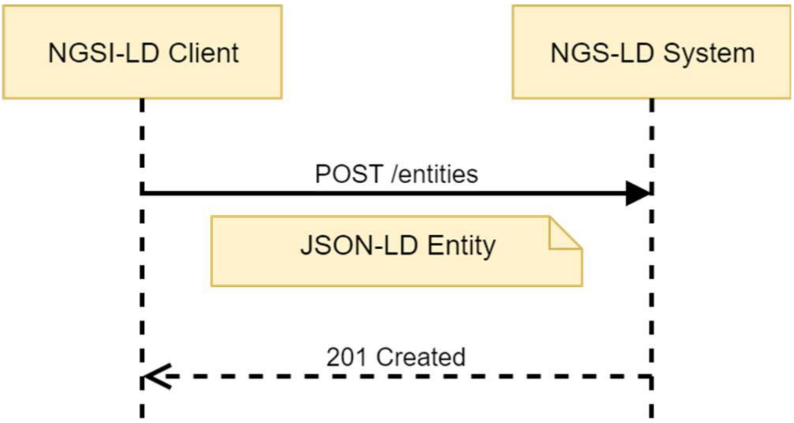


Figure 16: Create Entity interaction

2. GET

This method is associated to the operation "Query Entities", providing entities as part of the HTTP response output payload.

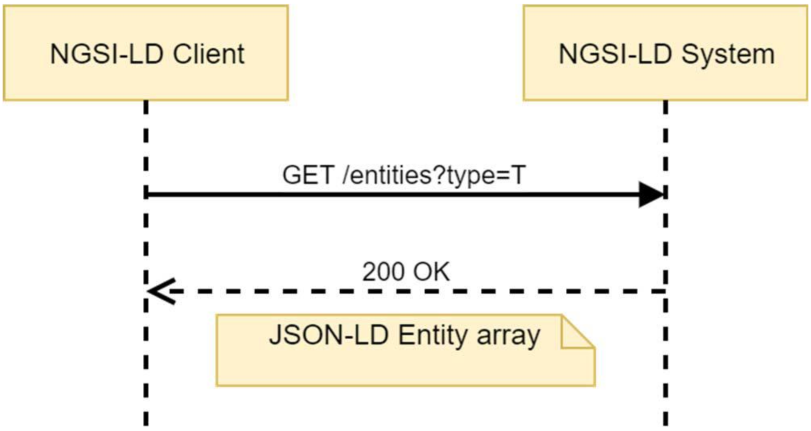


Figure 17: Query Entities interaction

4.1.4 Resource: entities/{entityId}

This resource represents an entity known to an NGSI-LD system.

Resource URI: /entities/{entityId}

Resource URI variables for this resource are defined as shown below.

Name	Definition
entityId Id	Id (URI) of the entity to be retrieved

Table 9: entities/{entityId} URI Variables

Resource methods:

1. **GET**

This method is associated to the operation "Retrieve Entity". The entity identifier is the value of the resource URI variable "entityId". Figure 18 shows the retrieve entity interaction.

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

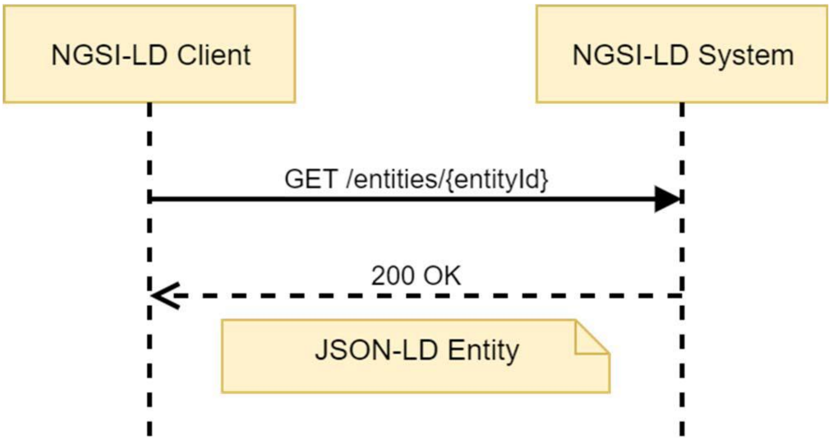


Figure 18: Retrieve Entity interaction

2. DELETE

This method is associated to the operation "Delete Entity". The entity identifier is the value of the resource URI variable "entityId". Figure 19 shows the delete entity interaction. [20]

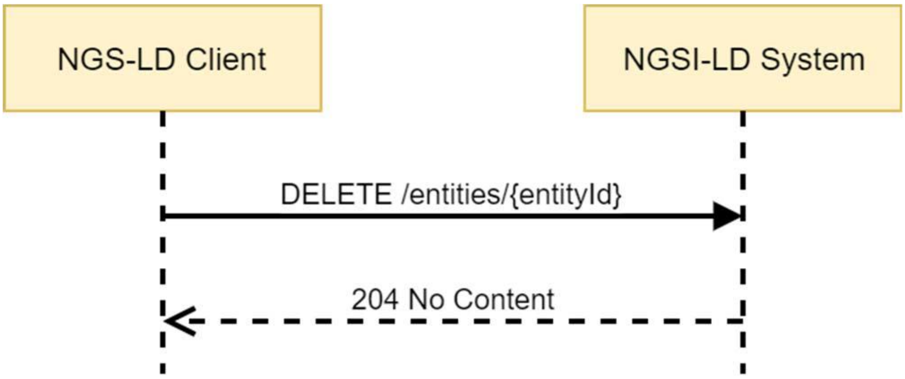


Figure 19: Delete Entity interaction

4.1.5 Resource: entities/{entityId}/attrs/

This resource represents all the Attributes (Properties or Relationships) of an NGSI-LD Entity.

Resource URI: /entities/{entityId}/attrs

Resource URI variables for this resource are defined in Table 10.

Name	Definition
entityId	Id (URI) of the concerned entity

Table 10: entities/{entityId}/attrs/ URI variables

Resource methods:

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

1. POST

This method is bound to the "Append Entity Attributes" operation. The entity identifier is the value of the resource URI variable "entityId". The data to be appended shall be contained in the HTTP request input payload. Figure 20 shows the append entity attributes interaction.

The "options" query parameter for this request can take the following values:

- "noOverwrite". Indicates that no attribute overwrite shall be performed.

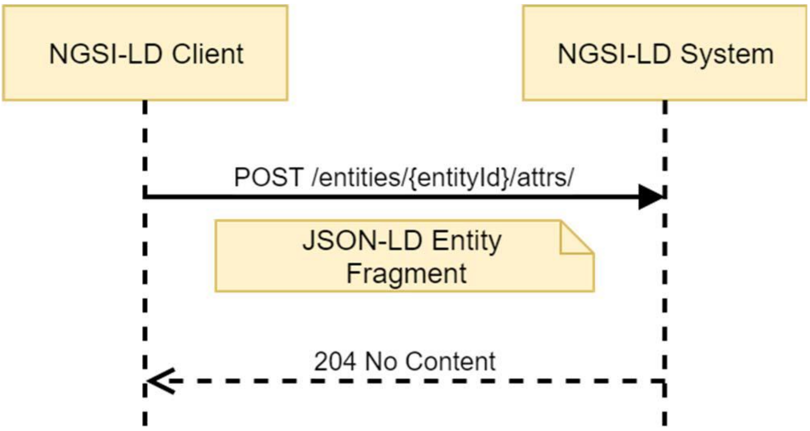


Figure 20: Append Entity Attributes interaction

2. PATCH

This method is bound to the "Update Entity Attributes" operation. The entity identifier is the value of the resource URI variable "entityId". The data to be updated shall be contained in the HTTP request input payload. Figure 21 shows the Update Entity Attributes interaction. [20]

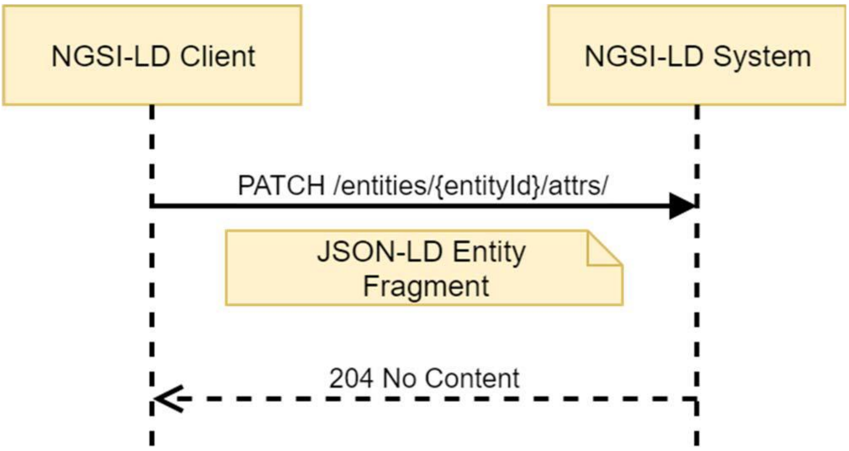


Figure 21: Update Entity Attributes interaction

4.1.6 Resource: entities/{entityId}/attrs/{attrId}

This resource represents an attribute (Property or Relationship) of an NGSI-LD Entity.

Resource URI: /entities/{entityId}/attrs/{attrId}

Resource URI variables for this resource are defined in Table 11.

Name	Definition
entityId	Id (URI) of the concerned entity
attrId	Attribute name (Property or Relationship)

Table 11: entities/{entityId}/attrs/{attrId} URI variables

Resource methods:

1. PATCH

This method is bound to the "Partial Attribute Update" operation. The entity identifier is the value of the resource URI variable "entityId". The attribute name is the value of the resource URI variable "attrId". The Entity Fragment shall be contained in the HTTP request input payload. Figure 22 shows the Partial Attribute Update interaction.

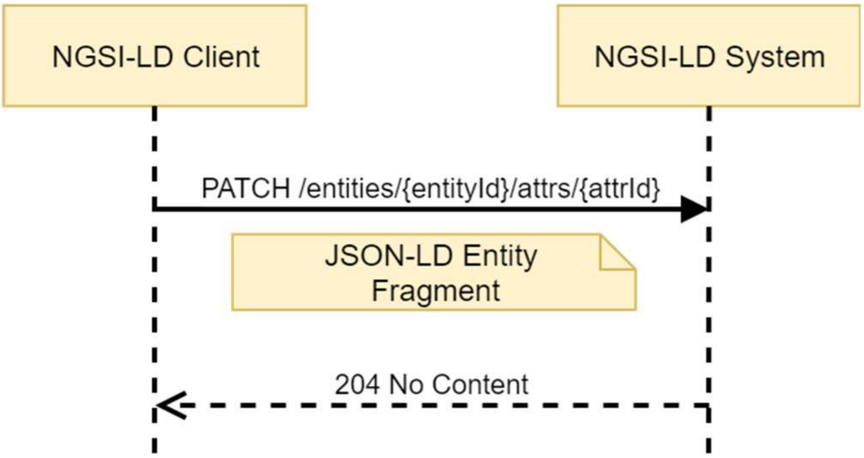


Figure 22: Partial Attribute Update interaction

2. DELETE

This method is associated to the operation "Delete Entity Attribute". The entity identifier is the value of the resource URI variable "entityId". The attribute name is the value of the resource URI variable "attrId". Figure 23 shows the Delete Entity Attribute interaction. [20]

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

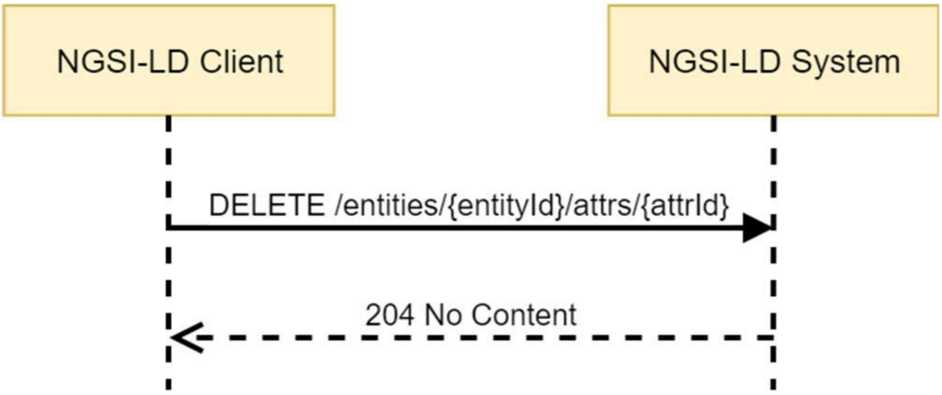


Figure 23: Delete Entity Attribute interaction

4.1.7 Resource: csourceRegistrations/

This resource represents a collection of context source registrations known to an NGSI-LD system.

Resource URI: /csourceRegistrations/

Resource methods:

1. POST

This method is bound to the operation "Register Context Source", taking the context source registration to be created from the HTTP request input payload. Figure 24 shows the Register Context Source interaction and describes the request body and possible responses.

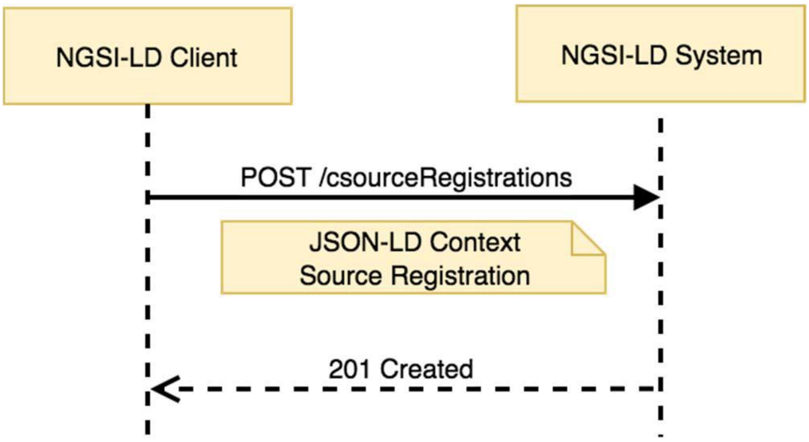


Figure 24: Register Context Source interaction

2. GET

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

This method is associated to the operation "Query Context Source Registrations". The parameters in the request describe entity related information, but instead of directly providing this entity information, the context source registration data, which describes context sources that can possibly provide the information, are returned as part of the HTTP response output payload. Figure 25 shows the Query Context Source Registrations interaction. [20]

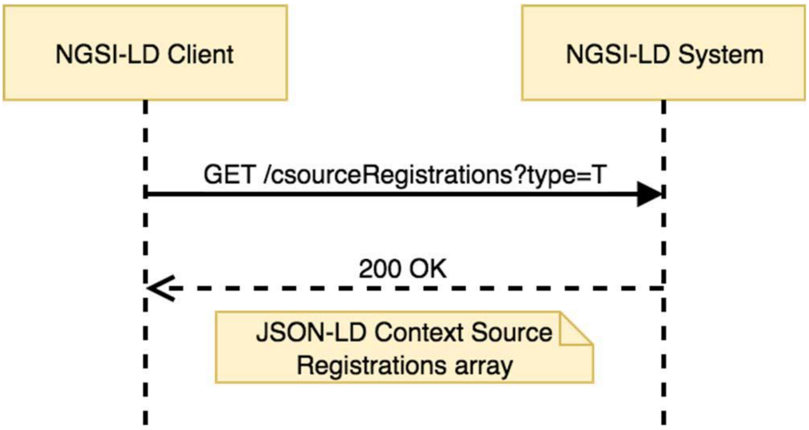

NGSI-LD Client
NGSI-LD System
GET /csourceRegistrations?type=T
200 OK
JSON-LD Context Source Registrations array

Figure 25: Query Context Source Registrations interaction

4.1.8 Resource: csourceRegistrations/{registrationId}

This resource represents a collection of context source registrations known to an NGSI-LD system.

Resource URI: /csourceRegistrations/{registrationId}

Resource URI variables for this resource are defined in Table 12.

Name	Definition
registrationId	Id (URI) of the context source registration

Table 12: csourceRegistrations/{registrationId} URI variables

Resource methods:

1. GET

This method is associated with the operation "Retrieve Context Source Registration". The registration identifier is the value of the resource URI variable "registrationId". Figure 26 shows the Retrieve Context Source Registration interaction.

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

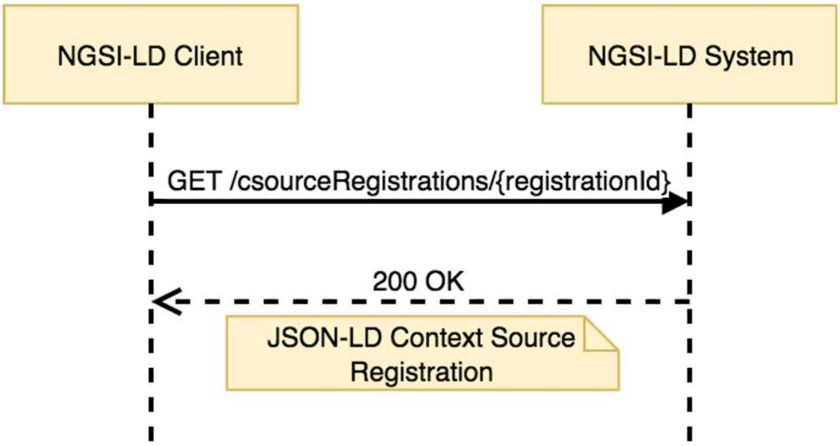

NGSI-LD Client
NGSI-LD System
GET /csourceRegistrations/{registrationId}
200 OK
JSON-LD Context Source Registration

Figure 26: Retrieve Context Source Registration interaction

2. PATCH

This method is bound to the "Update Context Source Registration" operation. The context source registration identifier is the value of the resource URI variable "registrationId". The context source registration to be updated shall be contained in the HTTP request input payload. Figure 27 shows the Update Context Source Registration interaction.

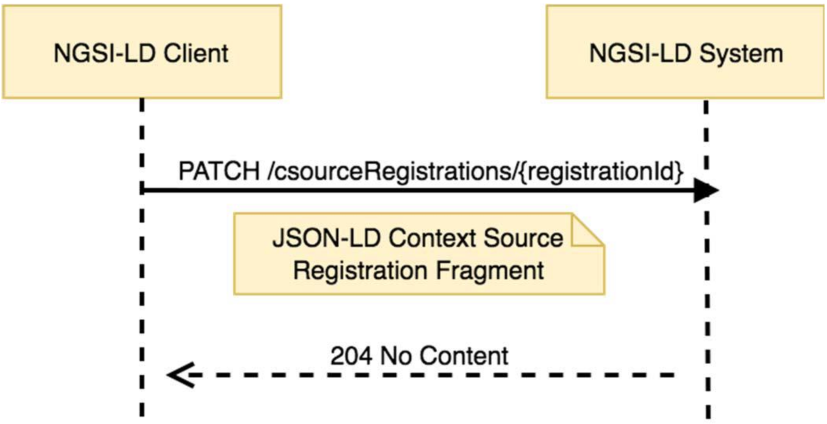

NGSI-LD Client
NGSI-LD System
PATCH /csourceRegistrations/{registrationId}
JSON-LD Context Source Registration Fragment
204 No Content

Figure 27: Update Context Source Registration interaction

3. DELETE

This method is associated to the operation "Delete Context Source Registration". The context source registration identifier is the value of the resource URI variable "registrationId". Figure 28 shows the Delete Context Source Registration interaction. [20]

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

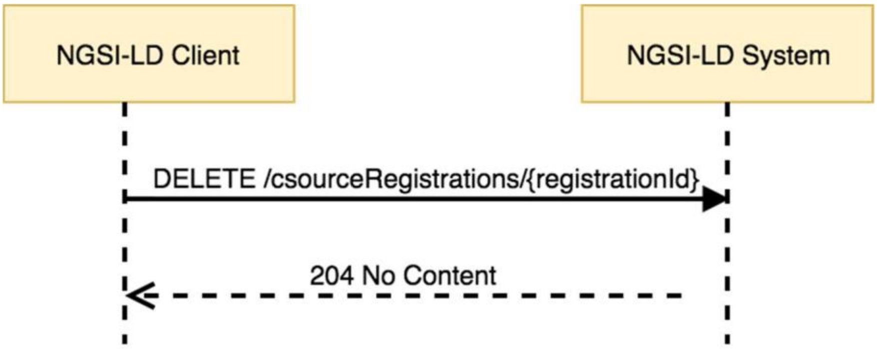


Figure 28: Delete Context Source Registration interaction

4.1.9 Resource: subscriptions/

This resource represents a collection of subscriptions known to an NGSI-LD system.

Resource URI: /subscriptions/

Resource methods:

1. **POST**

This method is bound to the operation "Create Subscription", taking the subscription to be created from the HTTP request input payload. Figure 29 shows the Create Subscription interaction.

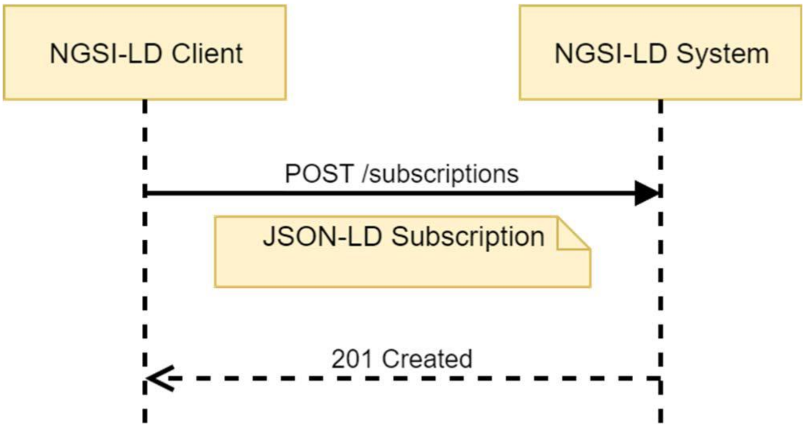


Figure 29: Create Subscription interaction

2. **GET**

This method is associated to the operation "Query Subscriptions", providing the subscription data as part of the HTTP response output payload. Figure 30 shows the Query Subscriptions interaction. [20]

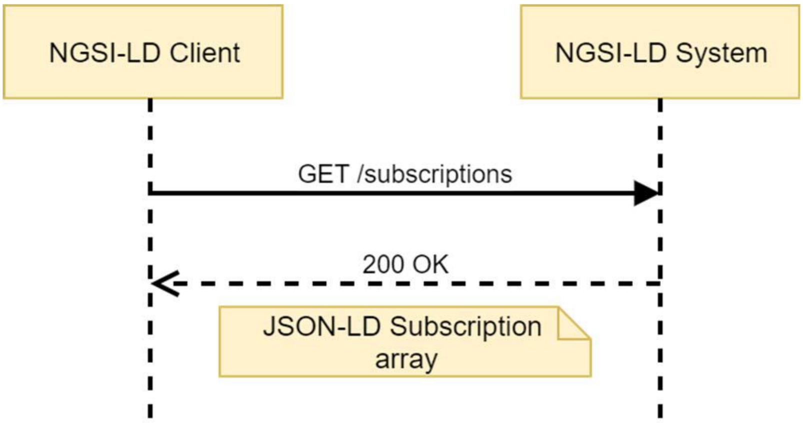


Figure 30: Query Subscriptions interaction

4.1.10 Resource: subscriptions/{subscriptionId}

This resource represents a subscription known to an NGSI-LD system.

Resource URI: /subscriptions/{subscriptionId}

Resource URI variables for this resource are defined in Table 13.

Name	Definition
subscriptionId	Id (URI) of the concerned subscription

Table 13: subscriptions/{subscriptionId} URI variables

Resource methods:

1. GET

This method is associated to the operation "Retrieve Subscription". The subscription identifier is the value of the resource URI variable "subscriptionId". Figure 31 shows the Retrieve Subscription interaction.

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

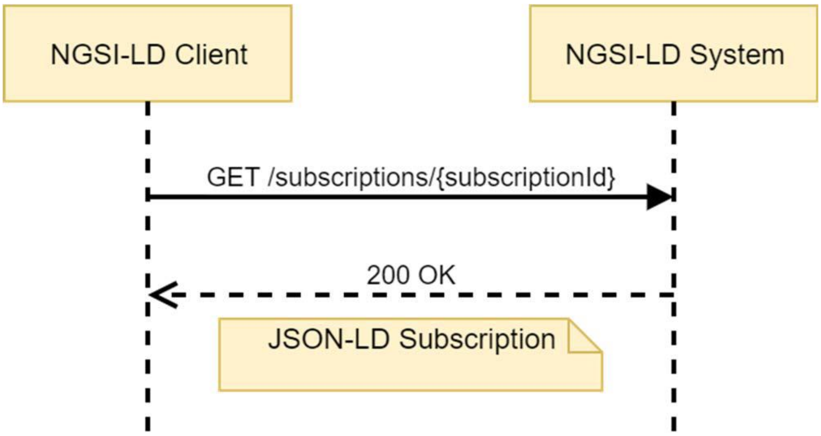


Figure 31: Retrieve Subscription interaction

2. PATCH

This method is associated to the operation "Update Subscription". The subscription identifier is the value of the resource URI variable "subscriptionId". Figure 32 shows the Update Subscription interaction.

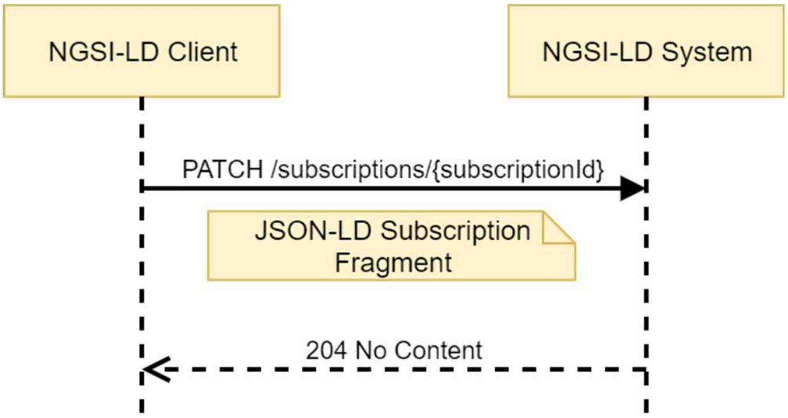

NGSI-LD Client
NGSI-LD System
PATCH /subscriptions/{subscriptionId}
JSON-LD Subscription Fragment
204 No Content

Figure 32: Update Subscription interaction

3. DELETE

This method is associated to the operation "Delete Subscription". The subscription identifier is the value of the resource URI variable "subscriptionId". Figure 33 shows the Delete Subscription interaction. [20]

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

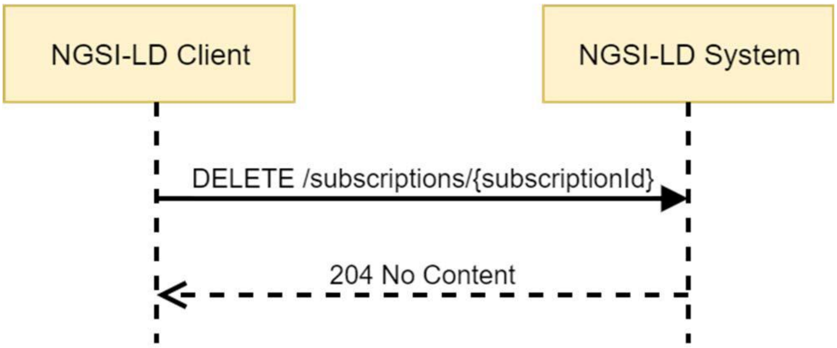


Figure 33: Delete Subscription interaction

4.1.11 Resource: csourceSubscriptions/

This resource represents a collection of context source registration subscriptions known to an NGSI-LD system.

Resource URI: /csourceSubscriptions/

Resource methods:

1. **POST**

This method is bound to the operation "Create Context Source Registration Subscription", taking the context source registration subscription to be created from the HTTP request input payload. Figure 34 shows the Create Context Source Registration Subscription interaction.

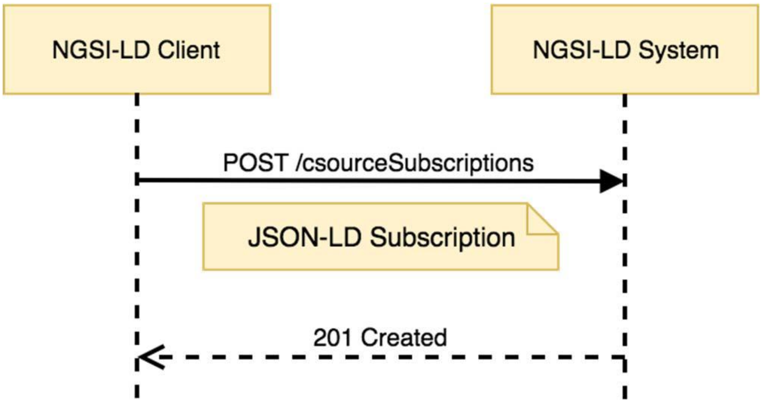


Figure 34: Create Context Source Registration Subscription interaction

2. **GET**

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

This method is associated to the operation "Query Context Source Registration Subscriptions", providing the context source registration subscription data as part of the HTTP response output payload. Figure 35 shows the Query Context Source Registration Subscriptions interaction. [20]

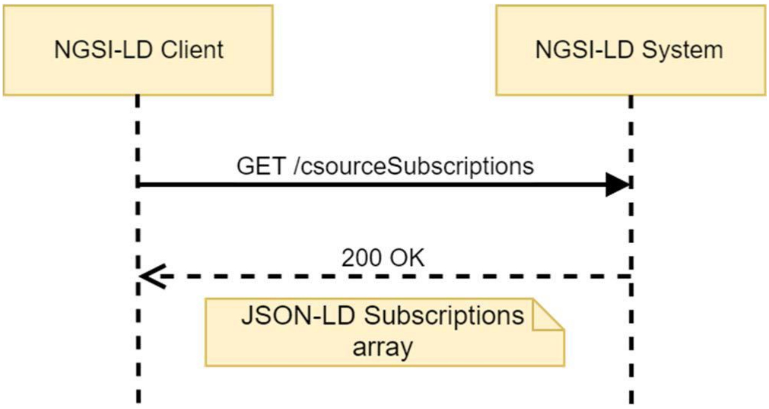


Figure 35: Query Context Source Registration Subscriptions interaction

4.1.12 Resource: csourceSubscriptions/{subscriptionId}

This resource represents a context source registration subscription known to an NGSI-LD system.

Resource URI: /csourceSubscriptions/{subscriptionId}

Resource URI variables for this resource are defined in Table 14.

Name	**Definition**
subscriptionId	Id (URI) of the concerned context source registration subscription

Table 14: csourceSubscriptions/{subscriptionId} URI variables

Resource methods:

1. GET

This method is associated to the operation "Retrieve Context Source Registration Subscription". The subscription identifier is the value of the resource URI variable "subscriptionId". Figure 36 shows the Retrieve Context Source Registration interaction.

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

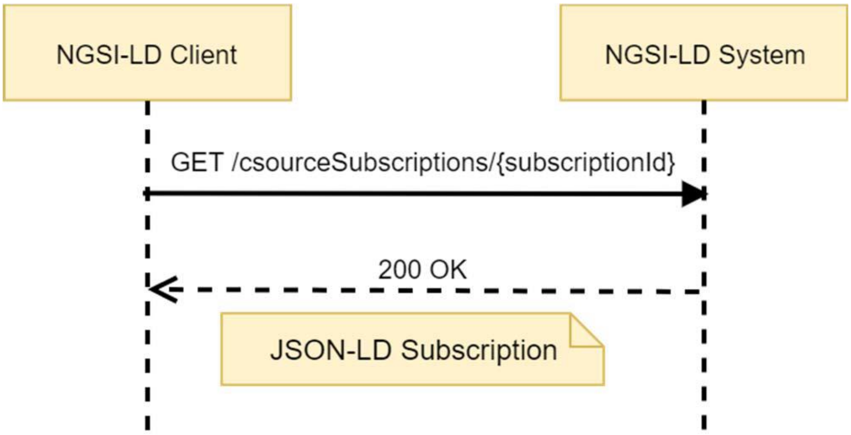


Figure 36: Retrieve Context Source Registration Subscription interaction

2. PATCH

This method is associated to the operation "Update Context Source Registration Subscription". The subscription identifier is the value of the resource URI variable "subscriptionId". Figure 37 shows the Update Context Source Registration Subscription interaction.

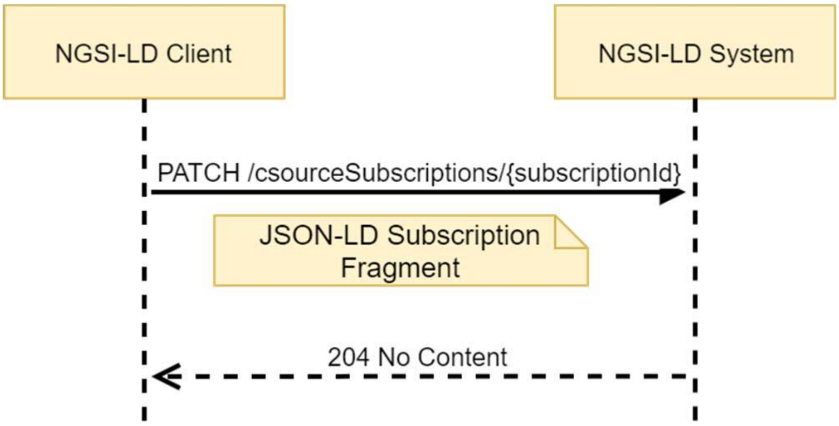

NGSI-LD Client
NGSI-LD System
PATCH /csourceSubscriptions/{subscriptionId}
JSON-LD Subscription Fragment
204 No Content

Figure 37: Update Context Source Registration Subscription interaction

3. DELETE

This method is associated to the operation "Delete Context Source Registration Subscription". The subscription identifier is the value of the resource URI variable "subscriptionId". Figure 38 shows the Delete Context Source Registration Subscription interaction. [20]

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

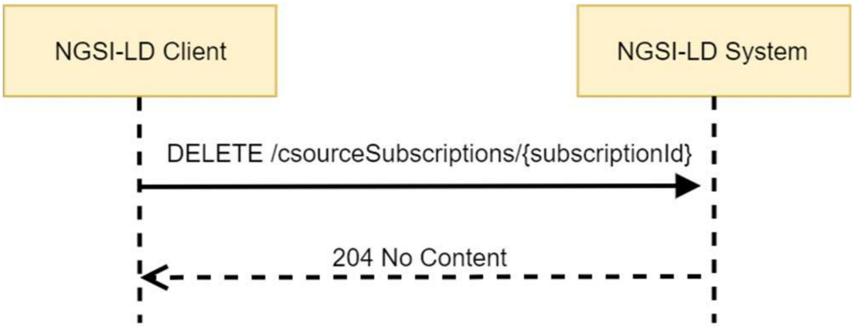


Figure 38: Delete Context Source Registration Subscription interaction

4.1.13 Resource: entityOperations/create

A sub-resource, pertaining to the *entityOperations/* resource, intended to enable batch entity creation for the NGSI-LD API.

Resource URI: /entityOperations/create

Resource methods:

1. **POST**

This method is associated to the operation "Batch Entity Creation". Figure 39 shows the operation interaction. [20]

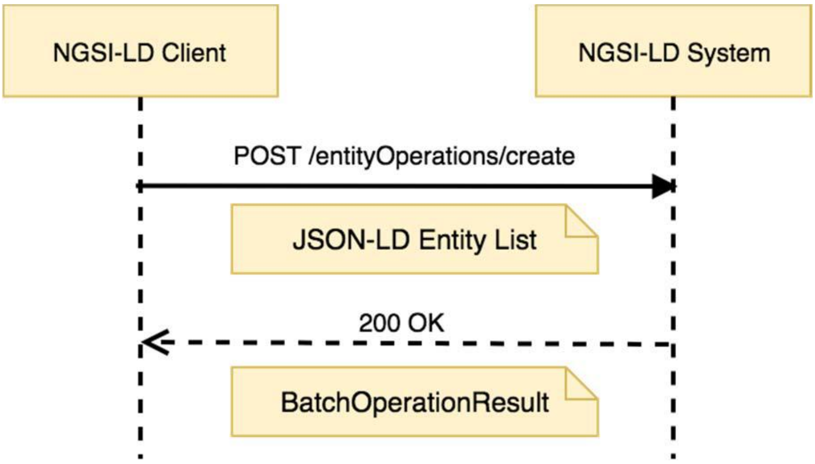


Figure 39: Batch Entity Creation Interaction

4.1.14 Resource: entityOperations/upsert

A sub-resource, pertaining to the *entityOperations/* resource, intended to enable batch entity creation or update for the NGSI-LD API.

Resource URI: /entityOperations/upsert

Resource methods:

1. POST

This method is associated to the operation "Batch Entity Creation or Update (Upsert)". Figure 40 shows the operation interaction.

The "options" query parameter for this request can take the following values:

- "replace". Indicates that all the existing Entity content shall be replaced (default mode).
- "update". Indicates that existing Entity content shall be updated. [20]

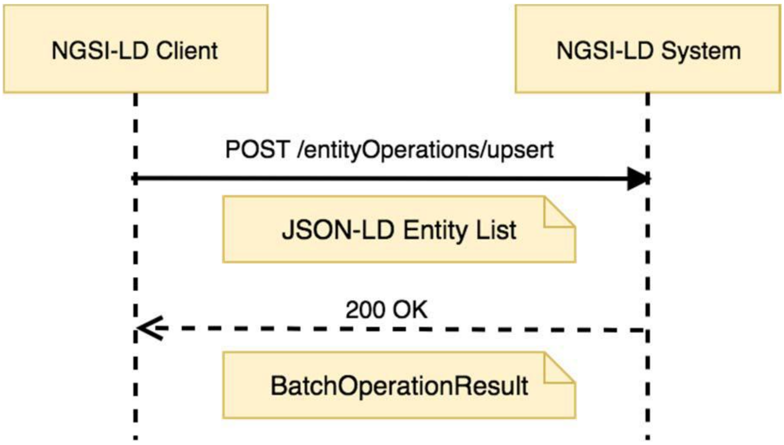


Figure 40: Batch Entity Creation or Update Interaction

4.1.15 Resource: entityOperations/update

A sub-resource, pertaining to the *entityOperations/* resource, intended to enable batch entity update for the NGSI-LD API.

Resource URI: /entityOperations/update

Resource methods:

1. POST

This method is associated to the operation "Batch Entity Update". Figure 41 shows the operation interaction.

The "options" query parameter for this request can take the following values:

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

- "noOverwrite". Indicates that no attribute overwrite shall be performed. [20]

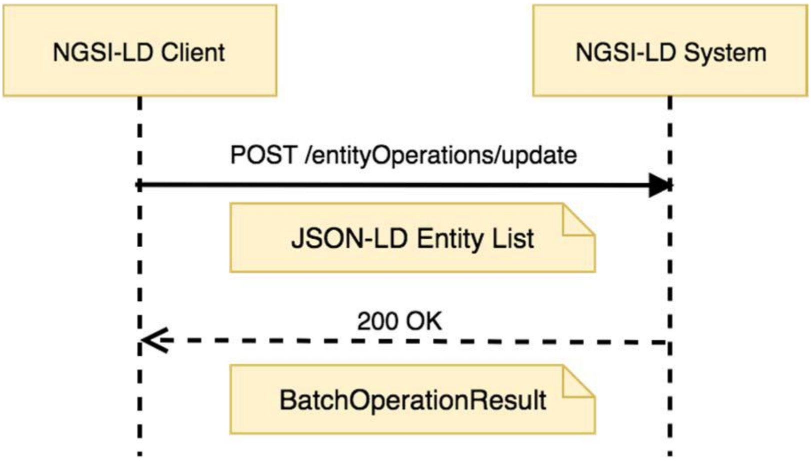


Figure 41: Batch Entity Update Interaction

4.1.16 Resource: entityOperations/delete

A sub-resource, pertaining to the *entityOperations/* resource, intended to enable batch entity deletion for the NGSI-LD API.

Resource URI: /entityOperations/delete

Resource methods:

1. POST

This method is associated to the operation "Batch Entity Delete". Figure 42 shows the operation interaction. [20]

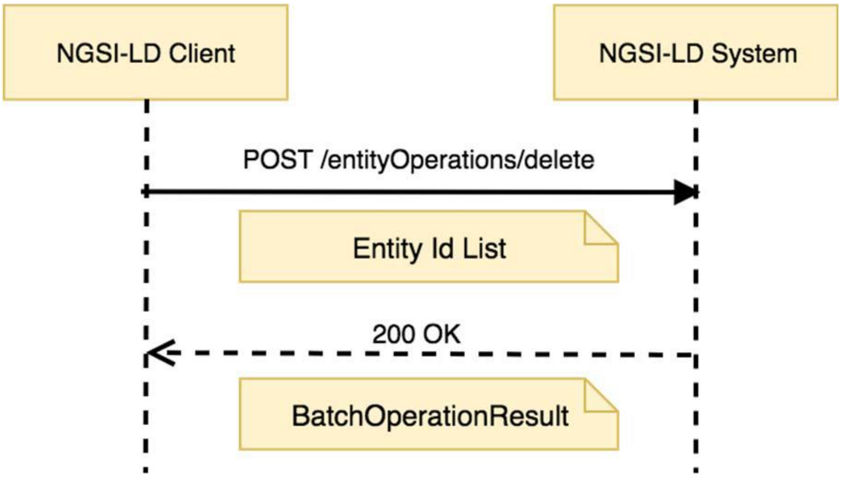


Figure 42: Batch Entity Delete Interaction

4.1.17Resource: temporal/entities/

This resource represents the temporal evolution of Entities known to an NGSI-LD system.

Resource URI: /temporal/entities/

Resource methods:

1. **POST**

This method is associated to the operation "Create or Update Temporal Representation of Entities", taking the temporal representation of entity to be created from the HTTP request input payload. Figure 43 shows this interaction (for creation).

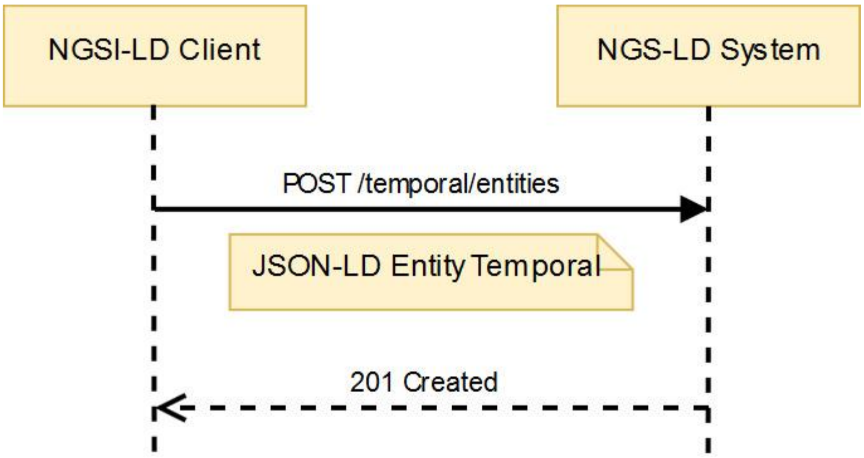


Figure 43: Create Temporal Representation of Entity interaction

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

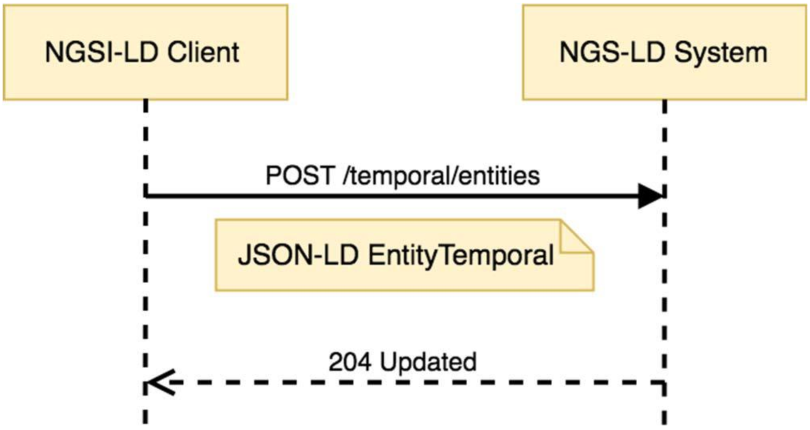


Figure 44: Update Temporal Representation of Entity interaction

2. GET

This method is associated to the operation "Query Temporal Evolution of Entities", providing the temporal evolution of the matching Entities as part of the HTTP response output payload. Figure 45 shows this interaction. [20]

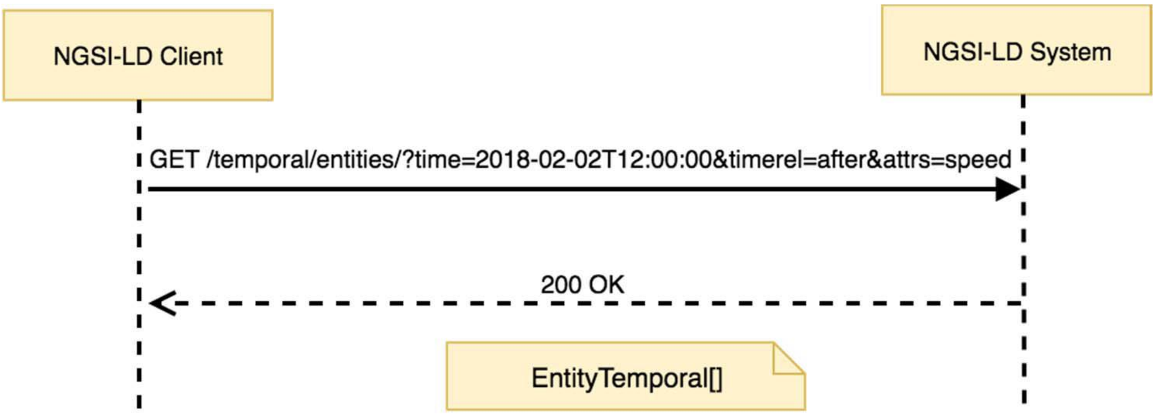

NGSI-LD Client
NGSI-LD System
GET /temporal/entities/?time=2018-02-02T12:00:00&timerel=after&attrs=speed
200 OK
EntityTemporal[]

Figure 45: Query Temporal Evolution of Entities interaction

4.1.18 Resource: temporal/entities/{entityId}

This resource is associated to the temporal representation of an Entity known to an NGSI-LD system.

Resource URI: /temporal/entities/{entityId}

Resource URI variables for this resource are defined in Table 15.

Name	Definition

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

entityId	Id (URI) of the entity to be retrieved

Table 15: temporal/entities/{entityId} URI variables

Resource methods:

1. GET

This method is associated to the operation "Retrieve temporal evolution of an Entity". The Entity identifier is the value of the resource URI variable *entityId*. Figure 46 shows the retrieve temporal representation of an entity interaction.

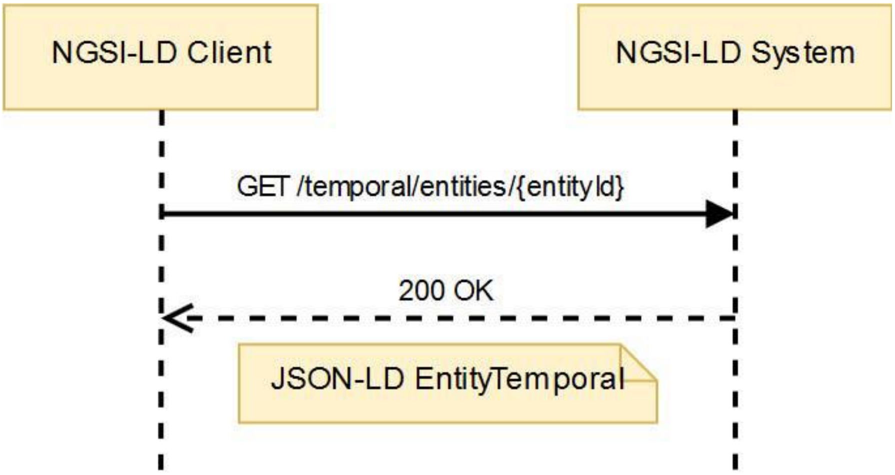


Figure 46: Retrieve Temporal evolution of an Entity interaction

2. DELETE

This method is associated to the operation "Delete Temporal Representation of an Entity". The Entity identifier is the value of the resource URI variable *entityId*. Figure 47 shows the delete entity interaction. [20]

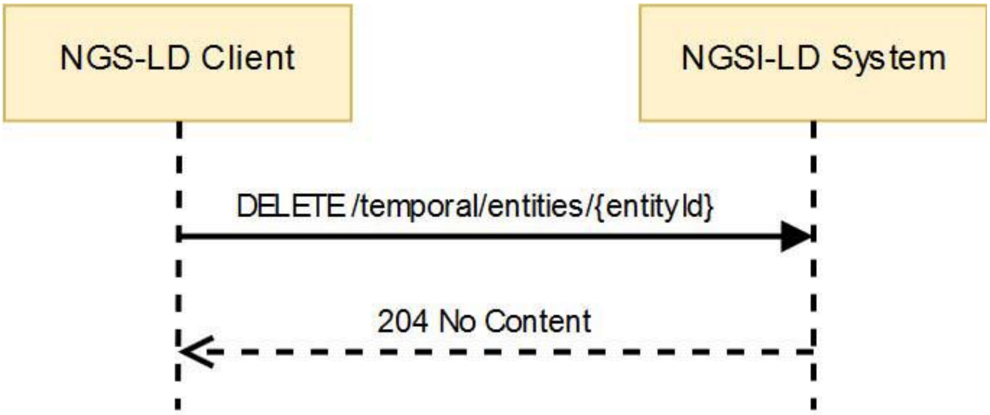


Figure 47: Delete Temporal Representation of Entity interaction

4.1.19 Resource: temporal/entities/{entityId}/attrs/

This resource represents all the Attributes (Properties or Relationships) of a Temporal Representation of an NGSI-LD Entity.

Resource URI: /temporal/entities/{entityId}/attrs/

Resource URI variables for this resource are defined in Table 16.

Name	Definition
entityId	Id (URI) of the concerned entity

Table 16: temporal/entities/{entityId}/attrs/ URI variables

Resource methods:

1. POST

This method is bound to the "Add Attributes to Temporal Representation of an Entity" operation. The Entity identifier is the value of the resource URI variable *entityId*. The data to be added shall be contained in the HTTP request input payload. Figure 48 shows the add entity attributes interaction. [20]

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

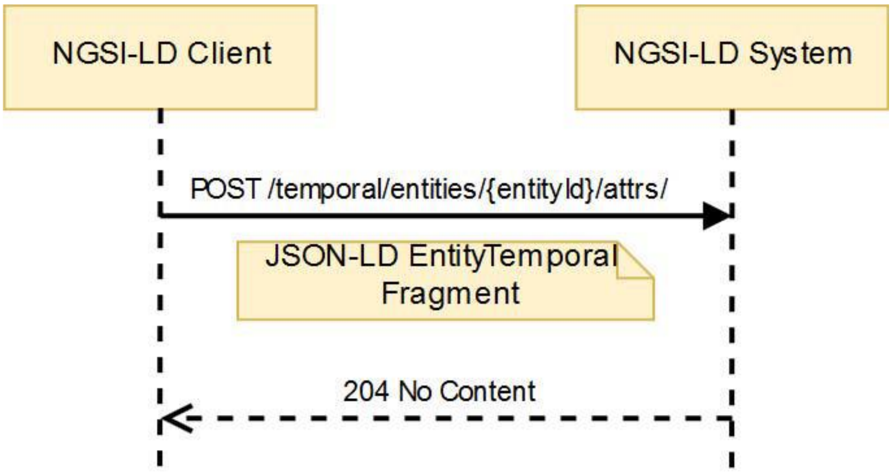

NGSI-LD Client
NGSI-LD System
POST /temporal/entities/{entityId}/attrs/
JSON-LD EntityTemporal Fragment
204 No Content

Figure 48: Add Attributes to Temporal Representation of an Entity interaction

4.1.20 Resource: temporal/entities/{entityId}/attrs/{attrId}

This resource represents an Attribute (Property or Relationship) of a Temporal Representation of an NGSI - LD Entity.

Resource URI: /temporal/entities/{entityId}/attrs/{attrId}

Resource URI variables for this resource are defined in Table 17.

Name	Definition
entityId	Id (URI) of the concerned entity
attrId	Attribute name (Property or Relationship)

Table 17: temporal/entities/{entityId}/attrs/{attrId} URI variables

Resource methods:

1. DELETE

This method is associated to the operation "Delete Attribute from Temporal Representation of an Entity". The Entity identifier is the value of the resource URI variable *entityId*. The Attribute Name is the value of the resource URI variable *attrId*. Figure 49 shows the Delete Attribute from Temporal Representation of an Entity interaction. [20]

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

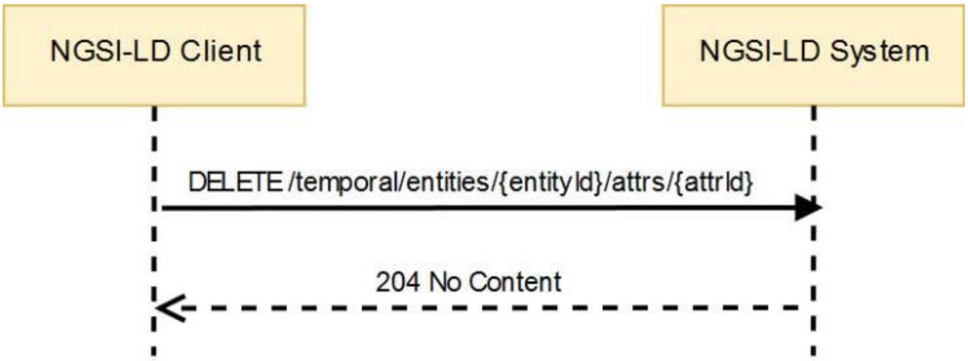


Figure 49: Delete Attribute from Temporal Representation of an Entity interaction

4.1.21 Resource: temporal/entities/{entityId}/attrs/{attrId}/{instanceId}

This resource represents an Attribute (Property or Relationship) instance of a Temporal Representation of an NGSI-LD Entity.

Resource URI: /temporal/entities/{entityId}/attrs/{attrId}/{instanceId}

Resource URI variables for this resource are defined in Table 18.

Name	Definition
entityId	Id (URI) of the concerned entity
attrId	Attribute Name (Property or Relationship)
instanceId	Id (URI) identifying a particular Attribute instance

Table 18: temporal/entities/{entityId}/attrs/{attrId}/{instanceId} URI variables

Resource methods:

1. PATCH

This method is associated to the operation "Modify attribute instance from Temporal Representation of an Entity". The Entity identifier is the value of the resource URI variable *entityId*. The attribute name is the value of the resource URI variable *attrId*. The instance identifier is the value of the resource URI variable *instanceId*. Figure 50 shows the Modify Entity Attribute instance interaction.

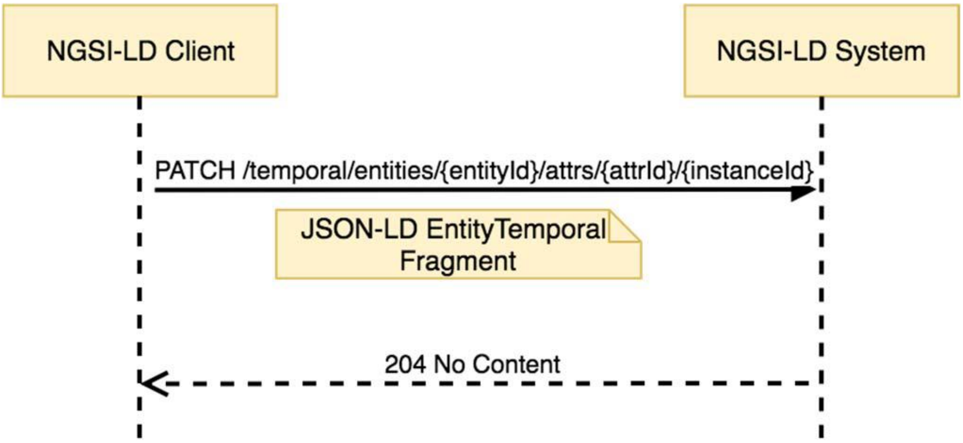

NGSI-LD Client
NGSI-LD System
PATCH /temporal/entities/{entityId}/attrs/{attrId}/{instanceId}
JSON-LD EntityTemporal Fragment
204 No Content

Figure 50: Modify Entity Attribute instance from Temporal Representation interaction

2. DELETE

This method is associated to the operation "Delete Attribute instance from Temporal Representation of an Entity". The Entity identifier is the value of the resource URI variable *entityId*. The Attribute Name is the value of the resource URI variable *attrId*. The instance identifier is the value of the resource URI variable *instanceId*. Figure 51 shows the Delete Entity Attribute instance interaction. [20]

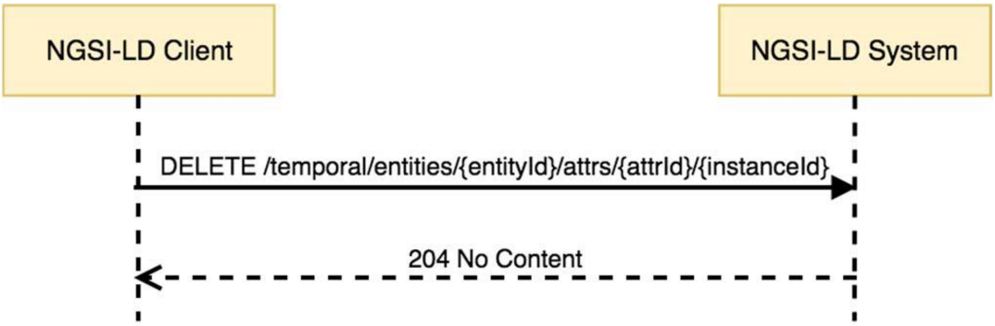

NGSI-LD Client
NGSI-LD System
DELETE /temporal/entities/{entityId}/attrs/{attrId}/{instanceId}
204 No Content

Figure 51: Delete Entity Attribute instance from Temporal Representation interaction

4.1.22 Core NGSI-LD @context definition

```
{
"@context": {
"ngsi-ld": "http://uri.etsi.org/ngsi-ld/",
"id": "@id",
"type": "@type",
"value": "http://uri.etsi.org/ngsi-ld/hasValue",
"object": {
"@id": "http://uri.etsi.org/ngsi-ld/hasObject",
"@type":"@id"
},
"Property": "http://uri.etsi.org/ngsi-ld/Property",
"Relationship": "http://uri.etsi.org/ngsi-ld/Relationship",
"DateTime": "http://uri.etsi.org/ngsi-ld/DateTime",
"Date": "http://uri.etsi.org/ngsi-ld/Date",
"Time": "http://uri.etsi.org/ngsi-ld/Time",
"createdAt": {
"@id": "http://uri.etsi.org/ngsi-ld/createdAt",
"@type": "DateTime"
},
"modifiedAt": {
"@id": "http://uri.etsi.org/ngsi-ld/modifiedAt",
"@type": "DateTime"
},
"observedAt": {
"@id": "http://uri.etsi.org/ngsi-ld/observedAt",
"@type": "DateTime"
},
"datasetId": {
"@id": "http://uri.etsi.org/ngsi-ld/datasetId",
"@type": "@id"
},
"instanceId": {
"@id": "http://uri.etsi.org/ngsi-ld/instanceId",
"@type": "@id"
},
"unitCode": "http://uri.etsi.org/ngsi-ld/unitCode",
"location": "http://uri.etsi.org/ngsi-ld/location",
"observationSpace": "http://uri.etsi.org/ngsi-ld/observationSpace",
"operationSpace": "http://uri.etsi.org/ngsi-ld/operationSpace",
"GeoProperty": "http://uri.etsi.org/ngsi-ld/GeoProperty",
"TemporalProperty": "http://uri.etsi.org/ngsi-ld/TemporalProperty",
"ContextSourceRegistration": "http://uri.etsi.org/ngsi-ld/ContextSourceRegistration",
"Subscription": "http://uri.etsi.org/ngsi-ld/Subscription",
"Notification": "http://uri.etsi.org/ngsi-ld/Notification",
"ContextSourceNotification": "http://uri.etsi.org/ngsi-ld/ContextSourceNotification",
"title": "http://uri.etsi.org/ngsi-ld/title",
"detail": "http://uri.etsi.org/ngsi-ld/detail",
"idPattern": "http://uri.etsi.org/ngsi-ld/idPattern",
"name": "http://uri.etsi.org/ngsi-ld/name",
"description": "http://uri.etsi.org/ngsi-ld/description",
"information": "http://uri.etsi.org/ngsi-ld/information",
"observationInterval": "http://uri.etsi.org/ngsi-ld/observationInterval",
"managementInterval": "http://uri.etsi.org/ngsi-ld/managementInterval",
"expires": {
"@id": "http://uri.etsi.org/ngsi-ld/expires",
"@type": "DateTime"
},
"endpoint": "http://uri.etsi.org/ngsi-ld/endpoint",
"entities": "http://uri.etsi.org/ngsi-ld/entities",
"properties": {
"@id": "http://uri.etsi.org/ngsi-ld/properties",
"@type": "@vocab"
```

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

```
},
"relationships": {
"@id": "http://uri.etsi.org/ngsi-ld/relationships",
"@type": "@vocab"
},
"start": {

"@id": "http://uri.etsi.org/ngsi-ld/start",
"@type": "DateTime"
},
"end": {
"@id": "http://uri.etsi.org/ngsi-ld/end",
"@type": "DateTime"
},
"watchedAttributes":{
"@id": "http://uri.etsi.org/ngsi-ld/watchedAttributes",
"@type": "@vocab"
},
"timeInterval": "http://uri.etsi.org/ngsi-ld/timeInterval",
"q": "http://uri.etsi.org/ngsi-ld/q",
"geoQ": "http://uri.etsi.org/ngsi-ld/geoQ",
"csf": "http://uri.etsi.org/ngsi-ld/csf",
"isActive": "http://uri.etsi.org/ngsi-ld/isActive",
"notification": "http://uri.etsi.org/ngsi-ld/notification",
"status": "http://uri.etsi.org/ngsi-ld/status",
"throttling": "http://uri.etsi.org/ngsi-ld/throttling",
"temporalQ": "http://uri.etsi.org/ngsi-ld/temporalQ",
"geometry": "http://uri.etsi.org/ngsi-ld/geometry",
"coordinates": "http://uri.etsi.org/ngsi-ld/coordinates",
"georel": "http://uri.etsi.org/ngsi-ld/georel",
"geoproperty": "http://uri.etsi.org/ngsi-ld/geoproperty",
"attributes": {
"@id": "http://uri.etsi.org/ngsi-ld/attributes",
"@type": "@vocab"
},
"format": "http://uri.etsi.org/ngsi-ld/format",
"timesSent": "http://uri.etsi.org/ngsi-ld/timesSent",
"lastNotification":{
"@id": "http://uri.etsi.org/ngsi-ld/lastNotification",
"@type": "DateTime"
},
"lastFailure":{
"@id": "http://uri.etsi.org/ngsi-ld/lastFailure ",
"@type": "DateTime"
},
"lastSuccess":{
"@id": "http://uri.etsi.org/ngsi-ld/lastSuccess",
"@type": "DateTime"
},
"uri": "http://uri.etsi.org/ngsi-ld/uri",
"accept": "http://uri.etsi.org/ngsi-ld/accept",
"success": {
"@id": "http://uri.etsi.org/ngsi-ld/success",
"@type": "@id"
},
"errors": "http://uri.etsi.org/ngsi-ld/errors",
"error": "http://uri.etsi.org/ngsi-ld/error",
"entityId": {
```

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

```
"@id": "http://uri.etsi.org/ngsi-ld/entityId",
"@type": "@id"
},
"updated": "http://uri.etsi.org/ngsi-ld/updated",
"unchanged": "http://uri.etsi.org/ngsi-ld/unchanged",
"attributeName": "http://uri.etsi.org/ngsi-ld/attributeName",
"reason": "http://uri.etsi.org/ngsi-ld/reason",
"timerel": "http://uri.etsi.org/ngsi-ld/timerel",
"time": {
"@id": "http://uri.etsi.org/ngsi-ld/time",
"@type": "DateTime"
},
"endTime": {
"@id": "http://uri.etsi.org/ngsi-ld/endTime",
"@type": "DateTime"
},
"timeproperty": "http://uri.etsi.org/ngsi-ld/timeproperty",
"subscriptionId": {
"@id": "http://uri.etsi.org/ngsi-ld/subscriptionId",
"@type": "@id"
},
"notifiedAt":{
"@id": "http://uri.etsi.org/ngsi-ld/notifiedAt",
"@type": "DateTime"
},
"data": "http://uri.etsi.org/ngsi-ld/data",
"triggerReason": "http://uri.etsi.org/ngsi-ld/triggerReason",

"values":{
"@id": "http://uri.etsi.org/ngsi-ld/hasValues",
"@container": "@list"
},
"objects":{
"@id": "http://uri.etsi.org/ngsi-ld/hasObjects",
"@type": "@id",
"@container": "@list"
}
}
}
```

4.2 Rest APIs definition (internal/external: connector's internal component, IDSA)

Chapter 4.2 provides an overview of both the IDSA Dataspace Connector[5] REST API and the IDSA HTTPS Generic API[6] which will provide the basis for the OneNet Connector API implementation.

[5] https://international-data-spaces-association.github.io/DataspaceConnector/

[6] https://app.swaggerhub.com/apis/idsa/ids-connector/0.3.2

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

4.2.1 Dataspace Connector

4.2.1.1 Introduction

The Dataspace Connector is an IDS connector that is being developed at Fraunhofer ISST. With the help of the Dataspace Connector, existing software can easily be extended by IDS connector functionalities in order to integrate them into an IDS data ecosystem. Furthermore, it is possible to use the Dataspace Connector as a basis for the development of own software that is to be connected to an IDS data ecosystem.

The Dataspace Connector uses the recent IDS Information Model version and the IDS Messaging Services for message handling with other IDS components. For managing datasets by means of their metadata as IDS resources, the Dataspace Connector provides a REST API. After an initial registration, IDS resources are persisted to an internal or external database of the connector. External data sources can be connected via REST endpoints, allowing the Dataspace Connector to act as an intermediary between the IDS data ecosystem and the actual data source.

Following the requirements of the International Data Spaces, TLS-encrypted communication with other IDS connectors and, for example, communication with an IDS broker are supported in the context of an IDS data ecosystem. The Dataspace Connector can simultaneously act as both a data provider and a data consumer, and thus both provide data in a data ecosystem and request it from other IDS connectors. The Dataspace Connector supports various usage control rules, which are implemented and enforced. This allows data in the IDS data ecosystem to be assigned usage control rules and ensures data sovereignty throughout the data lifecycle. Furthermore, identity management is supported by the integration of an identity provider in the IDS context, such as a DAPS.

The Dataspace Connector is an open source project whose development is being driven in collaboration with various research institutes and companies. Its architecture allows the existing implementation to be adapted as needed for domain-specific requirements. The deployment of the Dataspace Connector can be run in Docker as well as in Kubernetes [1].

4.2.1.2 REST API

Relations between Dataspace connector data model[7] objects are predefined and via the REST API, a data offer can thus be created very dynamically. Individual objects can be detached from each other, attached to other objects, and modified at any time as the mentioned data model is very modular.

Overview of all available endpoints reduced to generic endpoints:

[7] https://international-data-spaces-association.github.io/DataspaceConnector/Documentation/v6/DataModel

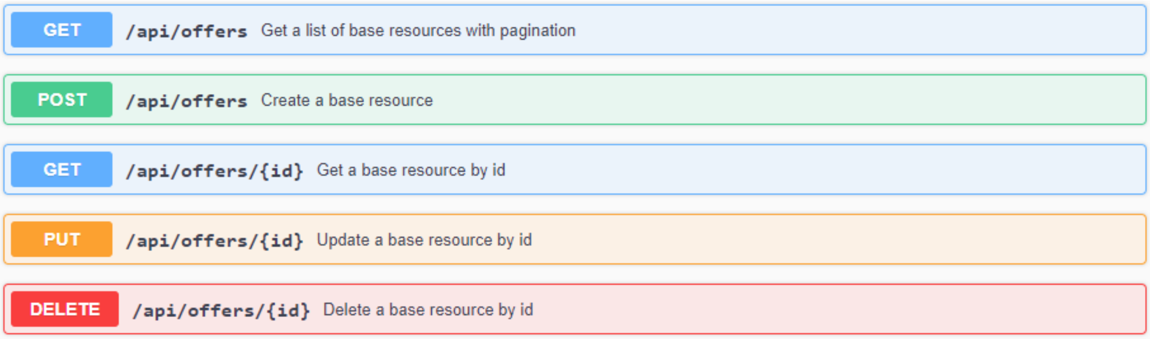
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

Table 19: Dataspace Connector REST API Endpoints

Method	Endpoint	Usage	Returns
GET	/	Get the connector	connector
POST	/Ts	Create a T	-
GET	/Ts	Get a list of all T	Ts
GET	/Ts/{id}	Get a T	T
PUT	/Ts/{id}	Change a T's details	-
DELETE	/Ts/{id}	Remove a T	-
GET	/Ts/{id}/Xs	Get a T's Xs	Xs
POST	/Ts/{id}/Xs	Add Xs to the T	-
PUT	/Ts/{id}/Xs	Replace Xs of the T	-
DELETE	/Ts/{id}/Xs	Remove Xs from the T	-

CRUD endpoints allow the creation and modification of both individual entities and the relations between objects - starting from the child and the parent. [21] [22]

Table 20: Swagger UI for creating offered resources



GET `/api/offers` Get a list of base resources with pagination

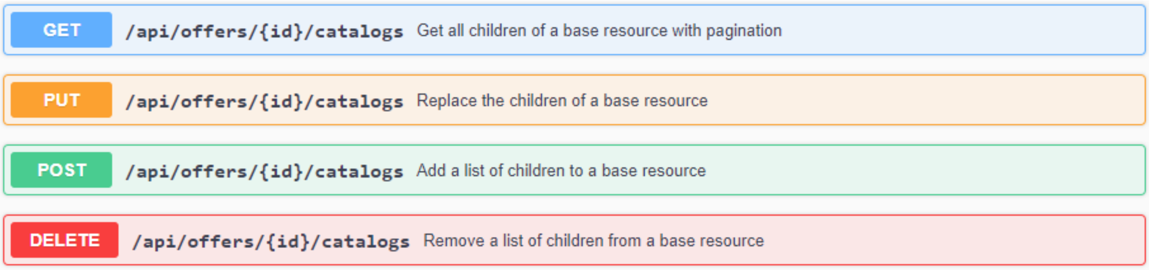
POST `/api/offers` Create a base resource

GET `/api/offers/{id}` Get a base resource by id

PUT `/api/offers/{id}` Update a base resource by id

DELETE `/api/offers/{id}` Delete a base resource by id

Table 21: Swagger UI for adding offers to catalogs

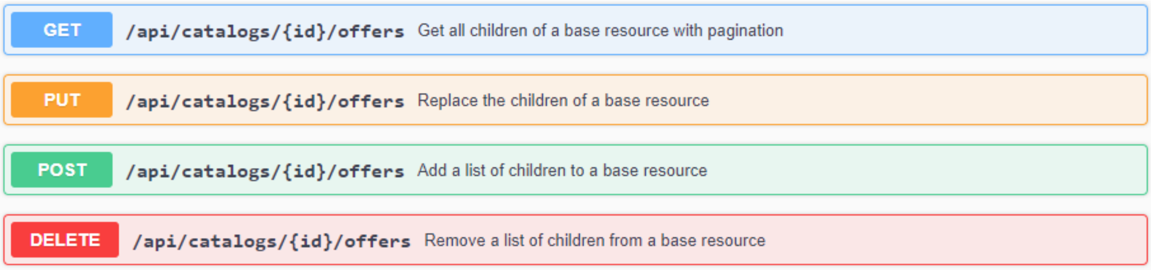


GET `/api/offers/{id}/catalogs` Get all children of a base resource with pagination

PUT `/api/offers/{id}/catalogs` Replace the children of a base resource

POST `/api/offers/{id}/catalogs` Add a list of children to a base resource

DELETE `/api/offers/{id}/catalogs` Remove a list of children from a base resource



GET /api/catalogs/{id}/offers Get all children of a base resource with pagination

PUT /api/catalogs/{id}/offers Replace the children of a base resource

POST /api/catalogs/{id}/offers Add a list of children to a base resource

DELETE /api/catalogs/{id}/offers Remove a list of children from a base resource

Table 22: Full list of endpoints

Method	Endpoint
GET	/api/agreements
GET	/api/agreements/{id}
GET	/api/agreements/{id}/artifacts
GET	/api/representations/{id}
PUT	/api/representations/{id}
DELETE	/api/representations/{id}
GET	/api/representations/{id}/requests
PUT	/api/representations/{id}/requests
POST	/api/representations/{id}/requests
DELETE	/api/representations/{id}/requests
GET	/api/representations/{id}/offers
PUT	/api/representations/{id}/offers
POST	/api/representations/{id}/offers
DELETE	/api/representations/{id}/offers
GET	/api/representations/{id}/artifacts
PUT	/api/representations/{id}/artifacts
POST	/api/representations/{id}/artifacts
DELETE	/api/representations/{id}/artifacts
GET	/api/representations
POST	/api/representations
GET	/api/requests/{id}
PUT	/api/requests/{id}
DELETE	/api/requests/{id}

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

Method	Endpoint
GET	/api/requests/{id}/representations
PUT	/api/requests/{id}/representations
POST	/api/requests/{id}/representations
DELETE	/api/requests/{id}/representations
GET	/api/requests/{id}/contracts
PUT	/api/requests/{id}/contracts
POST	/api/requests/{id}/contracts
DELETE	/api/requests/{id}/contracts
GET	/api/requests/{id}/catalogs
PUT	/api/requests/{id}/catalogs
POST	/api/requests/{id}/catalogs
DELETE	/api/requests/{id}/catalogs
GET	/api/requests
GET	/api/offers/{id}
PUT	/api/offers/{id}
DELETE	/api/offers/{id}
GET	/api/offers/{id}/representations
PUT	/api/offers/{id}/representations
POST	/api/offers/{id}/representations
DELETE	/api/offers/{id}/representations
GET	/api/offers/{id}/contracts
PUT	/api/offers/{id}/contracts
POST	/api/offers/{id}/contracts
DELETE	/api/offers/{id}/contracts
GET	/api/offers/{id}/catalogs
PUT	/api/offers/{id}/catalogs
POST	/api/offers/{id}/catalogs
DELETE	/api/offers/{id}/catalogs
GET	/api/offers
POST	/api/offers
GET	/api/catalogs/{id}

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

Method	Endpoint
PUT	/api/catalogs/{id}
DELETE	/api/catalogs/{id}
GET	/api/catalogs/{id}/offers
PUT	/api/catalogs/{id}/offers
POST	/api/catalogs/{id}/offers
DELETE	/api/catalogs/{id}/offers
GET	/api/catalogs
POST	/api/catalogs
GET	/api/artifacts/{id}
PUT	/api/artifacts/{id}
DELETE	/api/artifacts/{id}
GET	/api/artifacts/{id}/representations
PUT	/api/artifacts/{id}/representations
POST	/api/artifacts/{id}/representations
DELETE	/api/artifacts/{id}/representations
PUT	/api/artifacts/{id}/data
POST	/api/artifacts/{id}/data
GET	/api/artifacts
POST	/api/artifacts
GET	/api/artifacts/{id}/data/**
GET	/api/artifacts/{id}/agreements
POST	/api/ids/search
POST	/api/ids/resource/update
POST	/api/ids/resource/unavailable
POST	/api/ids/query
POST	/api/ids/description
POST	/api/ids/contract
POST	/api/ids/connector/update
POST	/api/ids/connector/unavailable
GET	/api/rules/{id}
PUT	/api/rules/{id}

Method	Endpoint
DELETE	/api/rules/{id}
GET	/api/rules/{id}/contracts
PUT	/api/rules/{id}/contracts
POST	/api/rules/{id}/contracts
DELETE	/api/rules/{id}/contracts
GET	/api/rules
POST	/api/rules
GET	/api/contracts/{id}
PUT	/api/contracts/{id}
DELETE	/api/contracts/{id}
GET	/api/contracts/{id}/rules
PUT	/api/contracts/{id}/rules
POST	/api/contracts/{id}/rules
DELETE	/api/contracts/{id}/rules
PUT	/api/contracts/{id}/requests
PUT	/api/contracts/{id}/requests
POST	/api/contracts/{id}/requests
DELETE	/api/contracts/{id}/requests
GET	/api/contracts/{id}/offers
PUT	/api/contracts/{id}/offers
POST	/api/contracts/{id}/offers
DELETE	/api/contracts/{id}/offers
GET	/api/contracts
POST	/api/contracts
GET	/api/configuration/pattern
PUT	/api/configuration/pattern
GET	/api/configuration/negotiation
PUT	/api/configuration/negotiation
POST	/api/examples/validation
POST	/api/examples/policy
GET	/api/configuration

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

Method	Endpoint
PUT	/api/configuration
GET	/api/connector
GET	/

4.2.2 IDSA HTTPS Generic API

The following table presents the SWAGGER UI for the IDSA HTTPS Generic API. [23]

Table 23: IDSA HTTPS Generic API

ROOT	Root endpoint of an IDS Connector	
GET	/	Endpoint for the self-description in JSON-LD
IDS-LDP	The IDS-LDP binding defines the RESTful interactions of the IDS. It is inspired by the Linked Data Platform W3C Recommendation and the IDS Communication Guide	
GET	/	Endpoint for the self-description in JSON-LD.
HEAD	/	Request the Headers
OPTIONS	/	Read the allowed operations
GET	/{catalog-id}/	Read the Catalog
HEAD	/{catalog-id}/	Request the Headers
OPTIONS	/{catalog-id}/	Read the allowed operations
PUT	/{catalog-id}/	Update the Catalog
PATCH	/{catalog-id}/	Update a Catalog entry
POST	/{catalog-id}/	Create a new Catalog entry
GET	/{catalog-id}/{resource-id}/	Read the Resource
HEAD	/{catalog-id}/{resource-id}/	Request the Headers
OPTIONS	/{catalog-id}/{resource-id}/	Read the allowed operations

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

PUT	/{catalog-id}/{resource-id}/	Update the Resource
PATCH	/{catalog-id}/{resource-id}/	Update a Resource entry
POST	/{catalog-id}/{resource-id}/	Create a new IDS Representation
DELETE	/{catalog-id}/{resource-id}/	Delete the Resource
GET	/{catalog-id}/{resource-id}/{representation-id}/	Read the Representation
HEAD	/{catalog-id}/{resource-id}/{representation-id}/	Request the Headers
OPTIONS	/{catalog-id}/{resource-id}/{representation-id}/	Read the allowed operations
PUT	/{catalog-id}/{resource-id}/{representation-id}/	Update the Representation
PATCH	/{catalog-id}/{resource-id}/{representation-id}/	Update a Representation entry
POST	/{catalog-id}/{resource-id}/{representation-id}/	Create a new IDS Representation
DELETE	/{catalog-id}/{resource-id}/{representation-id}/	Delete the Representation
GET	/{catalog-id}/{resource-id}/{contract-id}/	Read the Contract
HEAD	/{catalog-id}/{resource-id}/{contract-id}/	Request the Headers
OPTIONS	/{catalog-id}/{resource-id}/{contract-id}/	Read the allowed operations
PUT	/{catalog-id}/{resource-id}/{contract-id}/	Update the Contract
PATCH	/{catalog-id}/{resource-id}/{contract-id}/	Update a Contract entry
DELETE	/{catalog-id}/{resource-id}/{contract-id}/	Delete this IDS Contract

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

GET	/{catalog-id}/{resource-id}/{representation-id}/{artifact-id}/	Read the Artifact
HEAD	/{catalog-id}/{resource-id}/{representation-id}/{artifact-id}/	Request the Headers
OPTIONS	/{catalog-id}/{resource-id}/{representation-id}/{artifact-id}/	Read the allowed operations
PUT	/{catalog-id}/{resource-id}/{representation-id}/{artifact-id}/	Update the Artifact
POST	/{catalog-id}/{resource-id}/{representation-id}/{artifact-id}/	Add the Artifact Content
PATCH	/{catalog-id}/{resource-id}/{representation-id}/{artifact-id}/	Update an Artifact entry
DELETE	/{catalog-id}/{resource-id}/{representation-id}/{artifact-id}/	Delete this IDS Artifact
GET	/inbox/	Read the Inbox
HEAD	/inbox/	Request the Headers
OPTIONS	/inbox/	Read the allowed operations
POST	/inbox/	Send an IDS Notification
INFRASTRUCTURE	Endpoint for infrastructure communication (register, announce, unregister, request identity, etc.)	
POST	/infrastructure	waits for infrastructure-related messages
DATA	Endpoint for actual data-related interactions, for instance artifact requests, contract negotiations, etc.	
POST	/data	The data endpoint, waiting for Data Resource-related messages
HUMAN USER	Endpoints for human users interacting with the IDS Connector through a Web Browser.	
GET	/	Endpoint for the self-description in JSON-LD.
GET	/browse	Requests the frontend.

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

5 Final Remarks

One of the main purposes of Task 5.5 is to define the technical specification and interfaces for the OneNet project in order to successfully achieve the interoperability and integration with all available platforms and systems that generate data that could be required for different services by different actors. These interfaces will be developed as plugins to be described in WP6 by using ICT tools and endorsing open standards.

Based on the above, this deliverable presents in a clear and concise manner part of the project technical architecture with a view to enable interoperability and integration of the OneNet Interoperable Network of Platforms as expected above. Integral parts of this technical architecture are well documented technical specifications and interfaces for data models/platform agnostic OneNet middleware. These technical specifications and interfaces constitute an important design block of the reference implementation of the OneNet solution.

From a technical perspective the specifications and interfaces analysis presented are closely connected with standard architectures and initiatives (IDSA/FIWARE) in order to create a OneNet framework which is:

- scalable, pluggable and fully decentralized
- provides a series of data harmonization services
- provides tools for Data and Services Orchestration and evaluation
- has monitoring and analytics features
- takes into account cybersecurity and data governance guidelines

The work conducted in Task T5.5 so far and documented in this deliverable gives a complete overview on the NGSI-LD and IDSA Information models and API specifications which will form the basis of the aforementioned interfaces as NGSI-LD and IDSA API standards together with the adoption of the standardised data models is necessary and crucial for achieving the main objectives that OneNet Solution foresees.

In the next steps, the Task T5.5 will continue the analysis of mentioned technologies, taking a look in the implementation and evaluation phase of the project, including any possible feedback and results in the later stage. This analysis will be part of the development guideline and the integration plan of WP6 and its respective tasks before and during implementation.

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

6 References

[1] OneNet – Deliverable D5.1 “Concept and Requirements”, 2021

[2] https://fiware-true-connector.readthedocs.io/en/latest/

[3] OneNet – Deliverable D5.3 “Data and Platform Assets Functional Specs and Data Quality Compliance”

[4] OneNet – Deliverable D5.4 “AI, Big Data, IoT Enablers and FIWARE compliant interoperable interfaces for grid services”

[5] https://en.wikipedia.org/wiki/NGSI-LD

[6] https://www.etsi.org/deliver/etsi_gs/CIM/001_099/009/01.01.01_60/gs_cim009v010101p.pdf

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

[8] ETSI TS 103 264 (V2.1.1): "SmartM2M; Smart Appliances; Reference Ontology and oneM2M Mapping".

[9] https://portal.etsi.org/webapp/WorkProgram/Report_WorkItem.asp?WKI_ID=51351

[10] https://www.w3.org/TR/2014/REC-rdf-schema-20140225/.

[11] http://www.w3.org/TR/2014/REC-json-ld-20140116/.

[12] https://github.com/International-Data-Spaces-Association/InformationModel

[13] https://international-data-spaces-association.github.io/InformationModel/docs/4.1.0/index.html

[14] https://www.researchgate.net/publication/346501057_The_International_Data_Spaces_Information_Model_-_An_Ontology_for_Sovereign_Exchange_of_Digital_Content

[15] IETF RFC 7231: "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content". NOTE: Available at https://tools.ietf.org/html/rfc7231.

[16] IETF RFC 7232: "Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests". NOTE: Available at https://tools.ietf.org/html/rfc7232.

[17] OpenAPI Specification (Swagger). NOTE: Available at https://github.com/OAI/OpenAPI-Specification/blob/master/versions/3.0.2.md.

[18] IETF RFC 2818: "HTTP Over TLS". NOTE: Available at https://tools.ietf.org/html/rfc2818.

[19] IETF RFC 5246: "The Transport Layer Security (TLS) Protocol Version 1.2". NOTE: Available at https://tools.ietf.org/html/rfc5246.

[20] https://www.etsi.org/deliver/etsi_gs/CIM/001_099/013/01.01.01_60/gs_CIM013v010101p.pdf

[21] https://international-data-spaces-association.github.io/DataspaceConnector/

[22] https://app.swaggerhub.com/apis/benj-schol-test/dataspace-connector/6.0.0

[23] https://app.swaggerhub.com/apis/idsa/ids-connector/0.3.2

This paper reflects only the author’s view and the Innovation and Networks Executive Agency (INEA) is not responsible for any use that may be made of the information it contains.