ETSI GR CIM 023 V1.1.1 (2024-01)

Context Information Management (CIM);
NGSI-LD;
Case Study of NGSI-LD Adoptions

Disclaimer

The present document has been produced and approved by the cross-cutting Context Information Management (CIM) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference

DGR/CIM-0023

Keywords

application, smart city, smart water, use case

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00   Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Important notice

The present document can be downloaded from:
https://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

If you find a security vulnerability in the present document, please report it through our
Coordinated Vulnerability Disclosure Program:
https://www.etsi.org/standards/coordinated-vulnerability-disclosure

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Group Report (GR) has been produced by ETSI Industry Specification Group (ISG) cross-cutting Context Information Management (CIM).

Modal verbs terminology

In the present document "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Executive summary

The NGSI-LD specification [i.1] has been implemented and deployed globally. The present document provides several adoption cases as a proof of disseminations and serves as the reference material for the future users of NGSI-LD. Each adoption case illustrates what is a deployed system and services, how the NGSI-LD standard is used. Not all the adoptions are not listed here but there are more open sources that implement NGSI-LD and more deployments out there. Therefore, the present document would be revised later to have more cases.

Introduction

The NGSI-LD specifications provide data management for different systems (e.g. smart city) leveraging Linked Data. While the NGSI-LD Primer provides the developer guide, the present document provides where and how NGSI-LD are used in real deployments.

Having adoption cases in the present document is helpful for delegates of ETSI ISG CIM group as well as users of NGSI-LD. With the description for each adoption, it is better understandable how to apply the standard for data management platform in different services. The feedback from the deployments such as extensions, potential API and each conclusion are for the delegates to consider future contributions to the specifications.

So far, the adoption cases cover Korean, Indian and European deployments including smart city, data marketplace and water management domains. This would be extended to cover other regional adoptions soon with different open source or commercial implementations.

1 Scope

The present document provides the case studies of NGSI-LD adoptions. Basically, each case study consists of each deployed system introduction with its architecture and adopted specifications. Then it is followed by implementation extensions other than the NGSI-LD specifications, also interworking and modelling aspects. Based on the analysis, the case study provides new functional requirements and recommendations for the future NGSI-LD standardizations if there is any.

2 References

2.1 Normative references

Normative references are not applicable in the present document.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI GS CIM 009 (V1.6.1): "Cross-cutting Context Information Management (CIM); NGSI-LD API".

[i.2] ETSI GS CIM 013 (V1.1.1): "Context Information Management (CIM); NGSI-LD Test Purposes Descriptions".

3 Definition of terms, symbols and abbreviations

3.1 Terms

Void.

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AI Artificial Intelligence
AMQP Advanced Message Queuing Protocol
API Application Programming Interface
CCTV Closed-Circuit TeleVision
CPS Cyber-Physical System
CRUD Create Read Update and Delete
DNA Data, Network and AI

EISS	Epidemiological Investigation Support System
ETL	Extract, Transform, Load
ID	IDentifier
IoT	Internet of Things
IUDX	India Urban Data eXchange
KDCA	Korea Disease Control & Prevention Agency
KETI	Korea Electronics Technology Institute
LPWA	Low Power Wide Area
ML	Machine Learning
MQTT	Message Queuing Telemetry Transport
NGSI-LD	Next Generation Service Interface-Linked Data
ODALA	Collaborative, Secure, and Replicable Open Source Data Lakes for Smart Cities
OIDC	OpenID Connect
R&D	Research and Develoment
RDF	Resource Description Framework
REST	REpresentational State Transfer
SAREF	Smart Applications REFerence ontology
SEAS	Smart Energy Aware System
TLS	Transport Layer Security
UI	User Interace
URI	Uniform Resource Identifier
URN	Uniform Resource Name

4 Adoption Case Study 1: Smart City Data Hub (South Korea)

4.1 Introduction

In 2018, the National Smart City Strategic Program was launched and more than 100 organizations and 2 pilot cities joined the program. It consists of 3 projects having 1 R&D project and 2 pilot projects in Siheung and Daegu city. The role of the 1st project in the program is the technology provider and the scope of the core technology includes smart city data hub, massive IoT and digital twin. The Smart City Data Hub has been deployed by the 2nd and 3rd projects in the two cities with smart city service implementations (e.g. disaster management).

The main motivation of the smart city program is to build data-driven smart cities upon the digital transformed cities. To enable data-driven smart cities, a data centric smart city platform is the key and in case of South Korea, it is Smart City Data Hub. The main role of the city data platform is to gather city data from heterogeneous data sources (e.g. public data portal, legacy systems) and then process, analyse and distribute data to city services and external systems.

There are many different data, in terms of syntax, semantics and periodicity, that gets collected to the city data platform. Converting those data into common data models is necessary process for better data usage and less cost by service applications. Also, for further usability of various domain data, linked data concept is considered. From ETSI GS CIM 009 [i.1], NGSI-LD standard interfaces which fulfilled basic requirements to manage city linked data. To leverage the linked data in the Smart City Data Hub, semantic web technologies with a smart city core ontology have been used to see the potential of linked data.

Figure 4.1-1: Overview of National Smart City Strategic Program in South Korea

4.2 System Architecture

Korea Electronics Technology Institute (KETI) consortium in the 1st project designed the reference architecture of Smart City Data Hub as illustrated in Figure 4.2-1, which has been used the other two projects for their pilot deployments. The architecture has been implemented by the consortium and has been released as open source, named City Data Hub.

As intended by the NGSI-LD technology, the data hub collects city data from existing data systems. There are different types of data sources, for instance IoT platforms (e.g. oneM2M), open data from public portals, legacy databases. that are providing data to the data hub. The data hub manages collected data and utilizes the data in different purposes (e.g. analytics) so it can be used in different city services. Also as a data platform, it provides security and other management capabilities.

From the reference architecture, there are eight functional modules including the API gateway. City data gets collected by the data ingest module and be stored in and managed by the data core module. The analytics, marketplace and semantic modules consume data from the data core via NGSI-LD APIs and provides their functionalities to users. Other non-data-handling modules provide platform managements.

The API gateway protects the platform from external applications/clients with token authentication with the security module, API routing/blocking and request rate limit. The security module provides user account and client management so it supports authentication and authorization with OAuth 2.0. The cloud management module enables system administrator to monitor and manage the cloud infrastructure to run the Smart City Data Hub. Hybrid cloud, which uses private and public clouds together, infra can be set up and managed such as monitoring and metering.

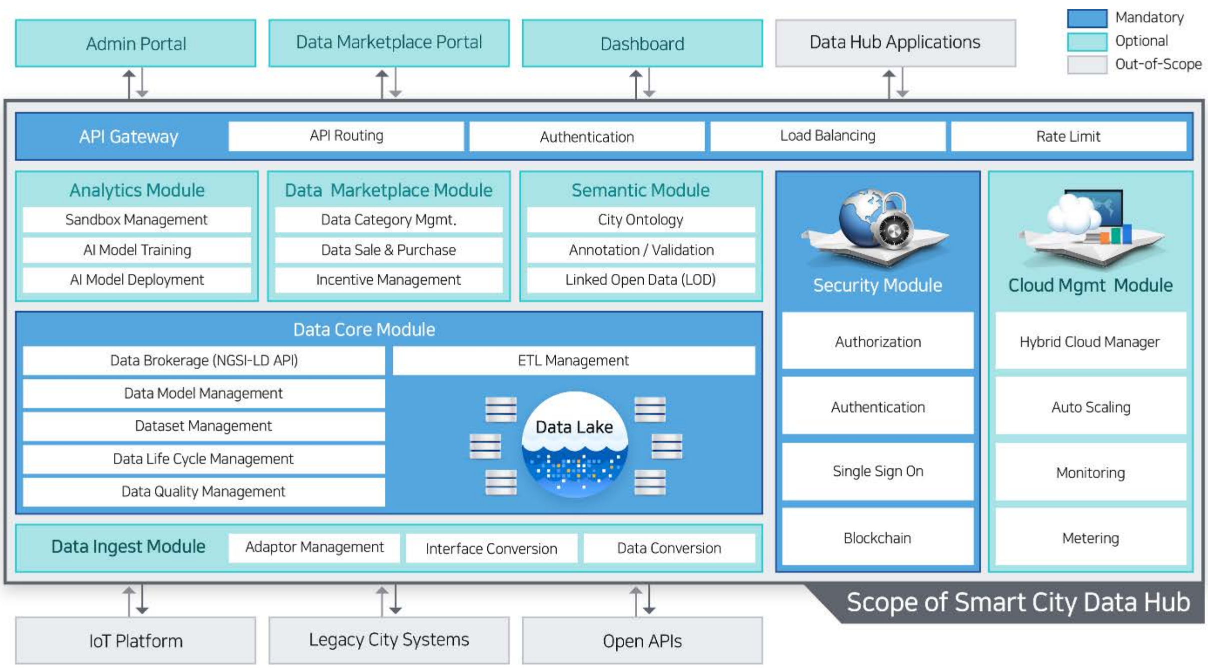

Figure 4.2-1: Reference Architecture and Data Flow of Smart City Data Hub

The data stored by the data broker, that implements NGSI-LD APIs, interworks with other modules in the data hub. Therefore, NGSI-LD APIs been used also to provide interoperability internally to the system. It is an important aspect of extensibility since other data utilization modules can be added, and they implement NGSI-LD APIs for interoperability. Also, the data core module serves city data to the external services/systems so having standard interfaces is beneficial.

The analytics module provides AI/ML based analytics and prediction while providing data pre-processing, model training/testing and model batch deployment capabilities. Data for model training and batch inference is provided by the data core module. During the training dataset preparation, multiple datasets, possibly from different domains, get mashed-up (e.g. parking events and weather). Several data pre-processing and training algorithms are supported so user can build AI/ML models for different service needs. Each trained models can be deployed with batch schedules. On each schedule, a model gets inference input data from the data core and ETL procedure and perform inference. Inference result (e.g. parking congestion prediction) from ML models can be also stored into the data core, so it can be used by other city applications as well.

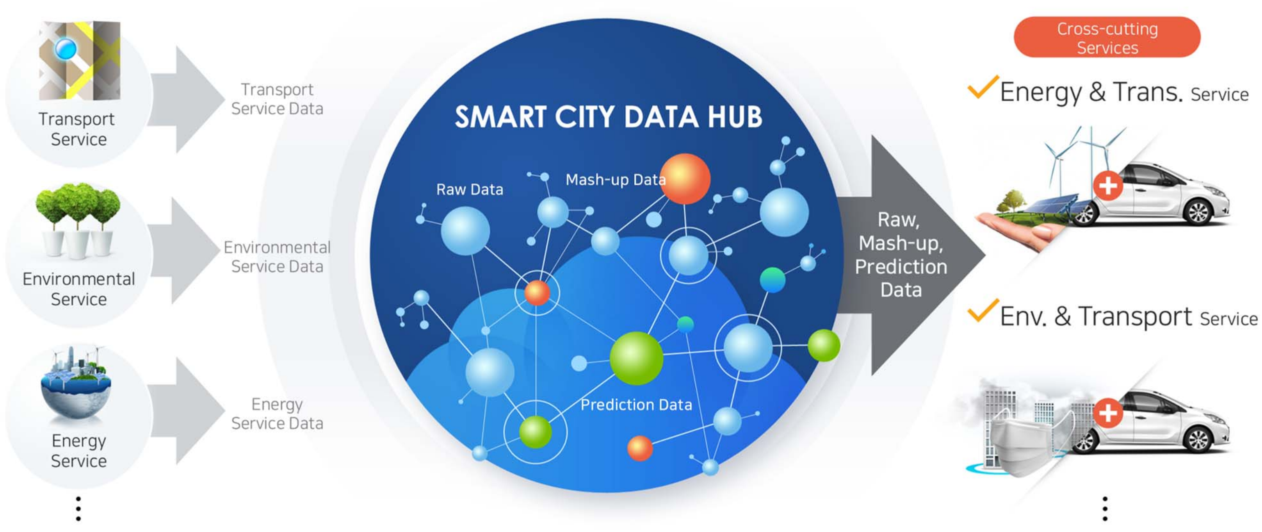

Figure 4.2-2: City Data Utilization on Smart City Data Hub

The data marketplace module is the back-end of the data marketplace portal. The data portal is also implemented to boost data distribution between data providers and consumers. It has been said that distributing data to 3rd parties to realize data-driven smart city is more important than just gathering big data from city infrastructures. A data provider can sell data products which is stored in the data core. The marketplace module sends NGSI-LD queries to find user's dataset, so selected data can be prepared as data products. When preparing the product, by the provider's preference, specific attributes can be selected and even attribute names can be changed from the original NGSI-LD entity instances. Purchased data can be used by fetching and receiving asynchronous notification which leverages NGSI-LD entity retrieval and subscription/notification features.

The semantic module includes a triple store which get RDF triple data from a semantic annotator. The annotator gets data over NGSI-LD APIs from the data core module and creates annotations with smart city ontologies. There has been defined the smart city core ontology for domain extensions (e.g. parking ontology).

Following the reference architecture designed in the program, in Daegu, commercial data hub implementation, other than open source based, has been deployed. Figure 4.2-3 describes the reference architecture of the Daegu case. As the extension to the reference model to meet the requirements for Daegu such as legacy system interfaces and video stream management in the data ingest module. For the data hub in the pilot city, several AI-based services have been tested. City bus routes have been optimized for citizens and safety services like police patrol routes planning and CCTV deployment optimization also been successfully deployed. With the data driven safety services as one of the best practices for smart city data hub, the services have been disseminated in other cities with data hub deployments.

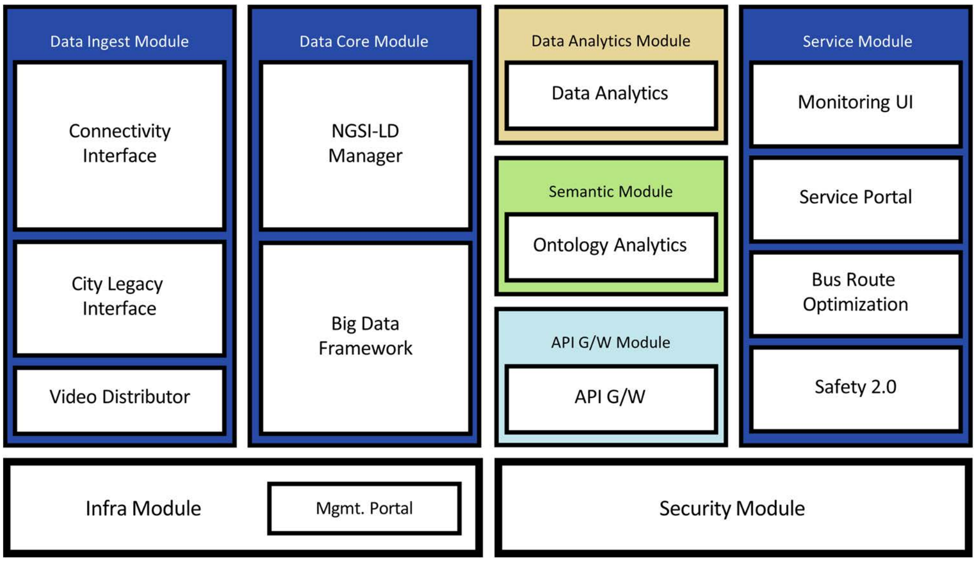

Figure 4.2-3: Reference Architecture of Daegu Smart City Data Hub

4.3 NGSI-LD Adoptions

City Data Hub, open source implementation of the Smart City Data Hub, adopts the essential NGSI-LD APIs as summarized in Table 4.3-1. The NGSI-LD API specification defines temporal data manipulation such as POST to "/temporal/entities/{entityId}" resource. In case of City Data Hub, as the specification also allows as a deployment scenario, data manipulation gets handled over "/entities/" and its sub-resources. Only the historical data query is used via "/temporal/entities" and "/temporal/entities/{entityId}" resources.

Table 4.3-1: Supported NGSI-LD APIs by City Data Hub

REST Resource	Operation	Notes
/entities/	POST, GET	
/entities/{entityId}	GET , DELETE	
/entities/{entityId}/attrs	POST, PATCH	
/entities/{entityId}/attrs/{attrId}	PATCH, DELETE	
/temporal/entities/	GET	Only GET operation is supported, context consumption is provided with "/entities"
/temporal/entities/{entityId}	GET	Only GET operation is supported, context consumption is provided with "/entities/{entityId}" and its sub-resources
/subscriptions	POST, GET	
subscriptions/{subscriptionId}	GET, PATCH, DELETE	
/csourceRegistrations	POST, GET	
/csourceRegistrations/{registrationId}	GET, PATCH, DELETE	
/csourceSubscriptions	POST, GET	
/csourceSubscriptions/{subscriptionId}	GET, PATCH, DELETE	

Additional features like multi-attribute support with datasetId attribute and multi-typing are not supported in the City Data Hub v1. The first version as minimum deployable solution focuses more on necessary capabilities for service deployments over the Smart City Data Hub.

The context broker implementation has been validated with ETSI NGSI-LD tester prototype based on robot framework and included test suites which implements conformance test cases in ETSI GS CIM 013 [i.2].

4.4 Extensions

Extending the API specifications in ETSI GS CIM 009 [i.1], the data hub architecture provides additional features. Data schema management is one of the data core feature which also provides more expressivity than JSON Schema with respect to data validation.

One major role of Smart City Data Hub is to provide data which fits to common data models. To use a new type of data in the data hub, a new schema needs to be registered in the data core module. In ETSI GS CIM 009 [i.1], @context is defined to represent linked data. However, there is no concept of schema management which is basically the binding between Entity type information with corresponding attributes (i.e. Property and Relationship). To guarantee data-level interoperability, when there is a new data ingested into the data core, before storing it in the data broker, optional data schema validation can be performed.

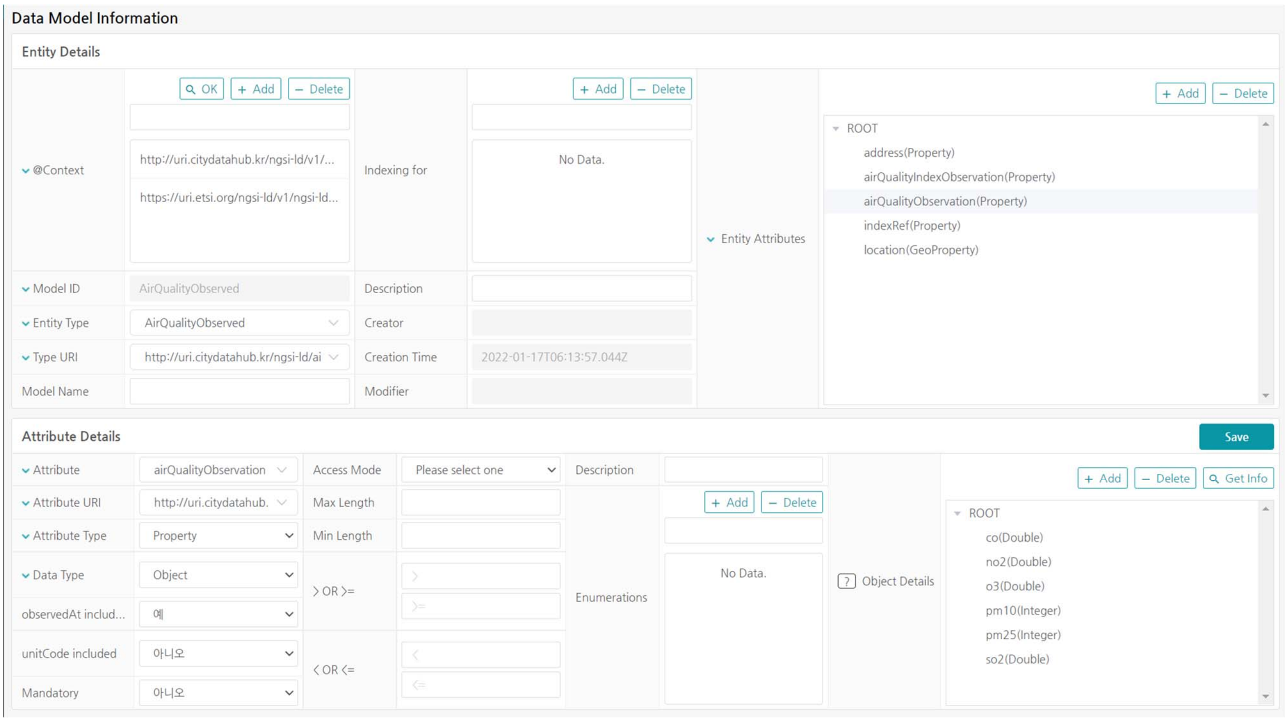

Figure 4.4-1: Data schema management UI

From the deployment experiences, mostly bulk data in the same data model is sent by the data sources. For instance, fine dust sensor measurements from hundreds of devices are provided by an IoT platform. From the proxy point of view, it is much simpler to implement not to remember whether a measurement, NGSI-LD Entity instance, has been provided to NGSI-LD broker. Mapping between the IoT sensors and Entity instances is not required when Upsert operation is supported. Therefore, Upsert operation per group of NGSI-LD instances was supported by the name of dataset. After this feature implementation, different concept but same terminology as dataset has been added to the NGSI-LD specification to support the multi attribute concept. It would be better to be discussed as data group for example to avoid any confusion.

Access control per data group, called "dataset" in the Smart City Data Hub, also been implemented. A new data schema is applied to the data core module, if needed, and then create a new data group for new data ingest. Then the data is accessible by the data owner, who is authenticated by the security module and so is authorized for the data group in terms of access control.

The data hub dataset manages metadata as depicted in Figure 4.4-2. REST APIs for dataset, as well as data model, management have been defined and implemented. Per dataset schema validation and lifecycle can be configured. Also targeted data storage per dataset can be selected (e.g. PostgreSQL, Hive, HBase).

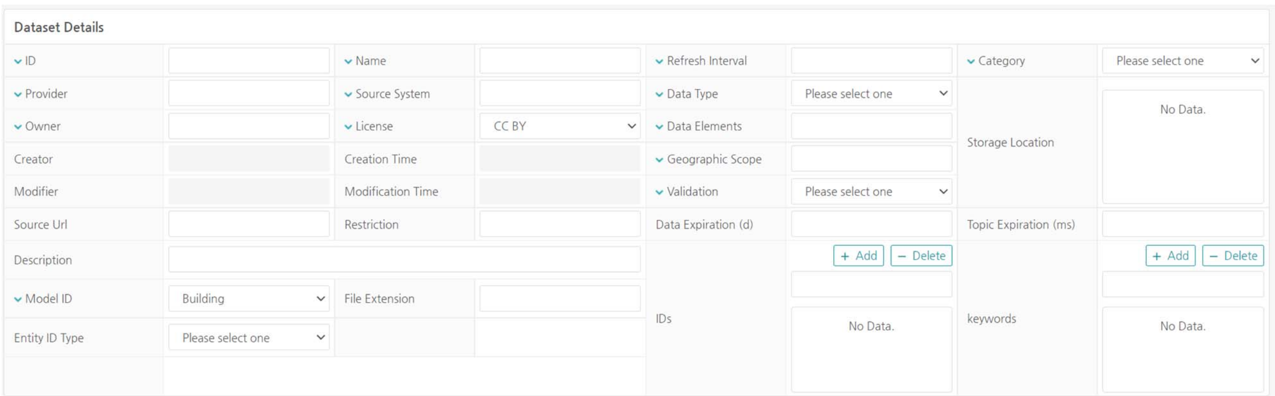

Figure 4.4-2: Dataset management UI

4.5 Interworking and Modelling Requirements

In the data broker, that uses NGSI-LD REST APIs, data models been defined using core NGSI-LD ontologies and those includes service specific Entity attributes. In case of the semantic module, the smart city core ontology was designed by analysing on SAREF4City and Smart Energy Aware Systems (SEAS) ontologies. As the proof-of-concept, parking service data has been semantically described in RDF format with the parking ontology by extending the smart city core ontology.

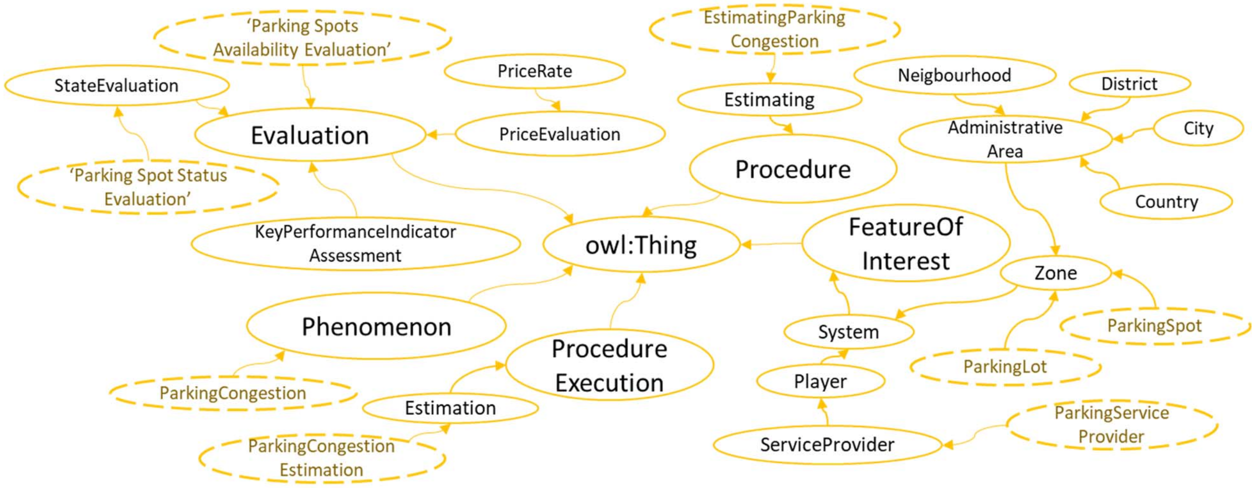

Figure 4.5-1: Parking ontology extending smart city core ontology

4.6 Potential API Requirements

From the Korean Smart City Data Hub system development, additional functional requirements for the NGSI-LD system have been derived as Table 4.6-1.

Table 4.6-1: Potential API Requirements

Number	Requirement
API-1	The NGSI-LD System shall be able to support data collection with grouping methods.
API-2	The NGSI-LD System shall be able to manage meta-data for a group of data.
API-3	The NGSI-LD System shall be able to manage schemas of data models.
API-4	The NGSI-LD System shall be able to support access control mechanisms per data group.
API-5	The NGSI-LD System shall be able to support the OAuth protocol for group-based access control.

4.7 Conclusions and Recommendations

Smart City Data Hub systems have been successfully deployed with smart city service pilots in Korean. For the City Data Hub open source, parking congestion prediction service was implemented as proof-of-concept. Six different data models were defined and data has been stored into the data core module. Collected data has been used by the analytics module to build the AI model and feed the model for inference. Inference result, which is the parking congestion level, also gets stored into the core via NGSI-LD API and shown as graphs on a parking service application.

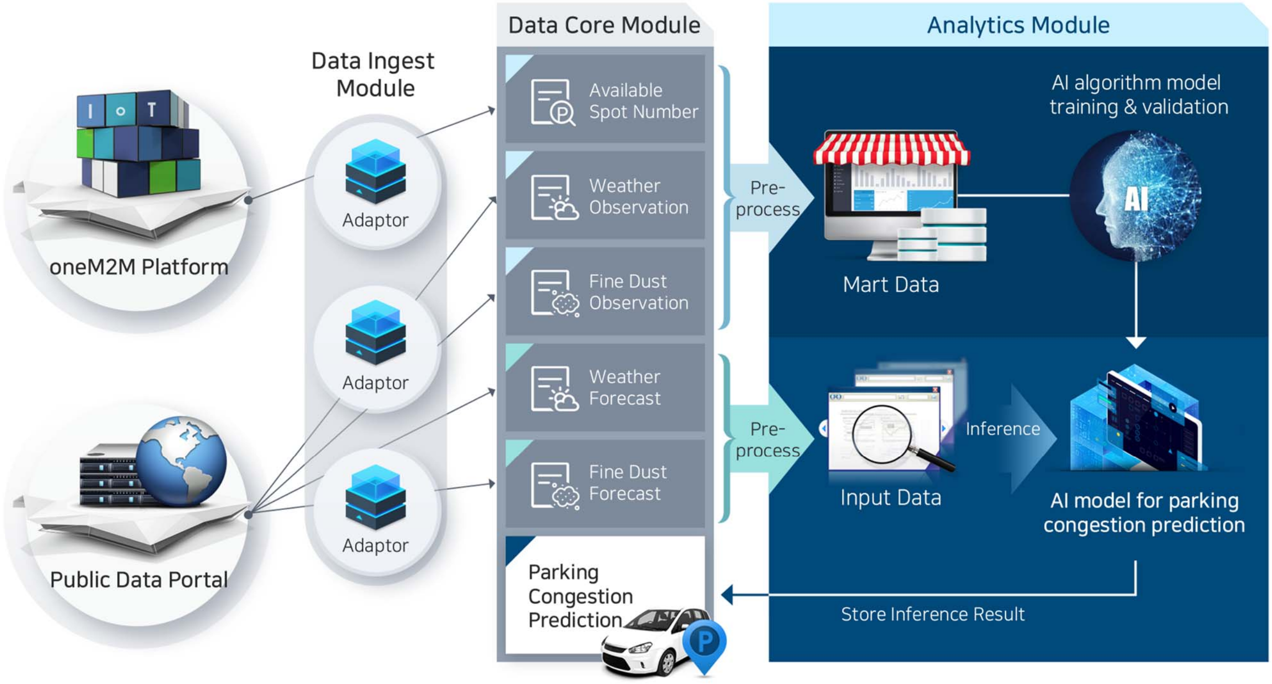

Figure 4.7-1: Parking Congestion Prediction with City Data Hub

In 2020, COVID-19 EISS (Epidemiological Investigation Support System) was built upon City Data Hub and being operated by Korea Disease Control & Prevention Agency (KDCA). The motivation was to provide data-driven epidemiological investigation support system which shortens the COVID-19 confirm case tracking process with the help of City Data Hub. It collects data from external systems such as mobile operators, credit card companies. Then pre-processing filtered the raw data so past few days routes of a confirmed person get created for investigators. With further analytics, intersections have been analysed for different confirmed case and been used to prepare preventive measures against epidemics by authorities. Those processed data get stored into the data core module and be used by the EISS portal over NGSI-LD APIs.

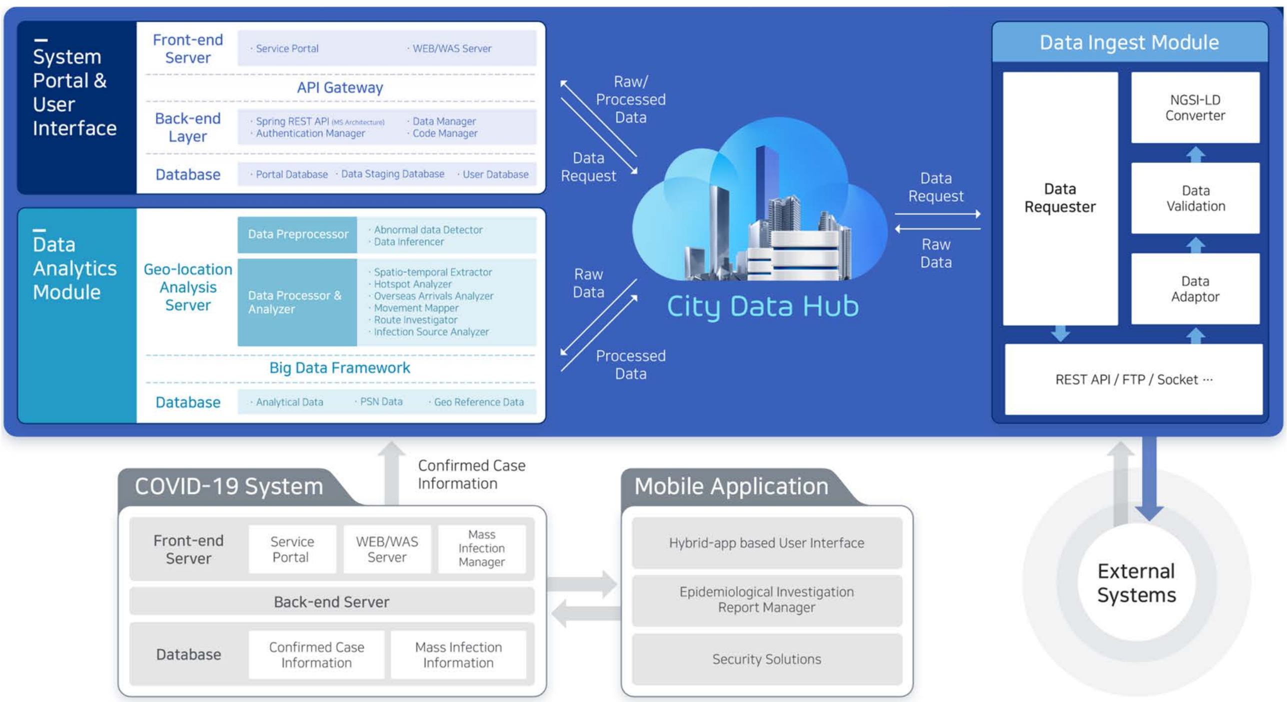

Figure 4.7-2: COVID-19 EISS over City Data Hub

NGSI-LD, adopted for in the Smart City Data Hub system, has potential with linked data since there are ton of scattered and fragmented data in smart cities. Therefore linking cross-domain data from different sources is being encouraged to have value-added data/information so it can be used to solve city problems.

Since NGSI-LD APIs support JSON as well as JSON-LD format, ordinary web developers could have easily implemented their services with minimum developer guides. From their point of view, someone else manages @context so they do not worry about @context definitions but use a common URI per system guide.

Leveraging linked data like graph traversal over multiple NGSI-LD Entities could be candidate scenario to be considered. Currently NGSI-LD REST APIs provides query per single Entity instance.

Federation of the data hub is one of the main features for the next version of the system and that elaborates the different deployment configurations concepts (e.g. federated context brokers) in the ETSI CIM GS 009 [i.1]. There would be many use cases for data sharing or federation among data hubs (e.g. public transportation across several cities), it is one of the key requirements for the successful data hub deployments.

5 Adoption Case Study 2: Water DNA

5.1 Introduction

From 2019 to 2022, the intelligent urban water resource management program, funded by the Ministry of Environment has been performed in South Korea. The program consists of the projects such as a pilot for intelligent urban water resource management operation with digital twin, integrated water data management, and decision-making support with CPS simulations. The water data management platform with the NGSI-LD APIs has been deployed for the testbed, which is the Haemil district in Sejong city.

Figure 5.1-1: Urban water resource management with digital twin

From the data management point of view the goal of the project is to collect, store and serve water resource data for the entire urban water cycle. In brief, when raw water is taken at pump stations, it gets purified. To distributed to households and buildings, the water firstly is delivered to reservoirs. Storm water is also used in a city and sewage is treated at sewage treatment plants. After that, the treated water is released in rivers.

Each component of the water cycle is operated by different organizations hence water resource data is fragmented in different systems. The required data for the pilot services in the program, data from different sources has been collected and stored in the data platform named Water DNA (Data, Network and AI). The Water DNA platform gathers the water data from different systems and provides them to different applications (e.g. CPS simulations) in the program.

Figure 5.1-2: Urban water cycle and water resources

5.2 System Architecture

Heterogeneous water resource data from different data sources are collected with each data adapter.

Data pertaining to the water cycle segments other than the testbed (e.g. households) is publicly available with open APIs. The flowrate data per building is gathered from the local proprietary system interworking to the Water DNA. IoT devices (e.g. water quality sensor) are also deployed in the pilot area and sensing data was transmitted over a private LPWA network. All the gathered data is stored in the NGSI-LD based data platform. In the Water DNA, the NGSI-LD compatible water data models have been defined, which are contributed to water domain data models in Smart Data Models initiative (http://smartdatamodels.org). With the data model schemas, data quality management have been done.

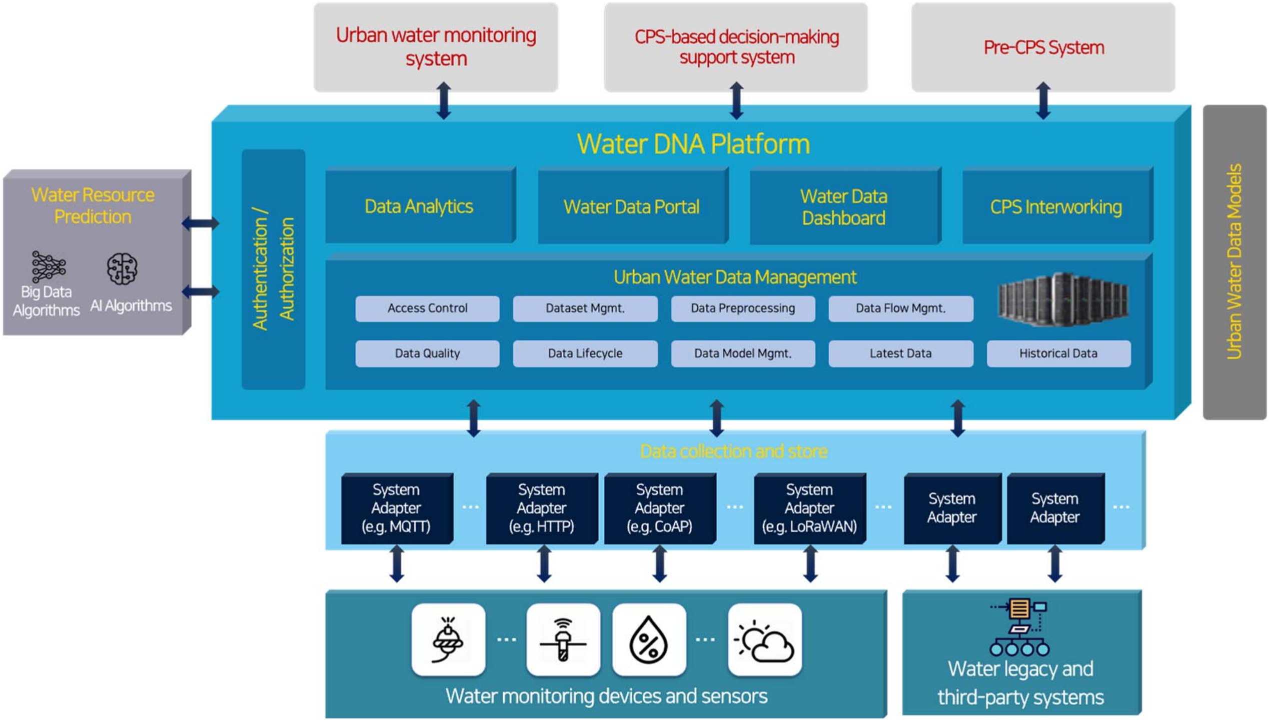

Figure 5.2-1: Water DNA with urban water management systems

The platform also provides common applications (e.g. water data portal) in terms of water data management. The user data portal provides 3D facility (e.g. water purification plant) management as well as data query and monitoring. The common data models applied in the water data platform has been guarantee data-level interoperability with other systems as depicted in Figure 5.2-1.

Not just raw data collected but synthetic data has also been generated and used. In the platform, AI data analytics feature creates water resource prediction (e.g. water demands in next 24 hours) and the results get stored in the platform. The CPS decision-making support system performs simulations and generate data which cannot be physically monitored (e.g. pressure in the pipe without physical sensors) due to limitations in the testbed. Those synthetic data and physical raw data are all monitored on the digital twin based urban water operation system.

5.3 NGSI-LD Adoptions

The Water DNA platform leverages the NGSI-LD open source implementation in Korea, which is the City Data Hub (see clause 5.2). In the system deployment, the central broker model was deployed and the NGSI-LD APIs summarized in Table 5.3-1 that is a subset of Table 4.3-1, are used.

Table 5.3-1: Supported NGSI-LD APIs by Water DNA

REST Resource	Operation	Notes
/entities/	POST, GET	
/entities/{entityId}	GET , DELETE	
/entities/{entityId}/attrs	POST, PATCH	
/entities/{entityId}/attrs/{attrId}	PATCH, DELETE	
/temporal/entities/	GET	Only GET operation is supported, context consumption is provided with "/entities"
/temporal/entities/{entityId}	GET	Only GET operation is supported, context consumption is provided with "/entities/{entityId}" and its sub-resources
/subscriptions	POST, GET	
subscriptions/{subscriptionId}	GET, PATCH, DELETE	

5.4 Extensions

Since the Water DNA platform is based on the City Data Hub open source S/W, extensions to the NGSI-LD are basically the same. In terms of query capabilities, the entity query per dataset ID and data storage type is supported.

5.5 Interworking and Modelling Requirements

The targets of NGSI-LD Entity modelling are two types: water facilities and water sensors. A water facility has a Property which is related to Properties from sensors. For instance, at a pump station, an intake flowrate is defined for the pump station which is measured by a flowmeter in the plant. NGSI-LD Relationships are defined for this facility-sensor relationships.

From the pilot services, a common NGSI-LD Property "isVirtual" defined. This can be applied to any virtual data observation. For example, a water pressure value as a simulation result is marked as virtual. By some applications it could be an important information where real vs. virtual data distinction is important. For different simulation scenarios, sometimes physical observation can be used as input, but to test other scenarios, virtual data can be generated, stored in the NGSI-LD data platform and finally served as simulation inputs.

5.6 Potential API Requirements

From the Water DNA platform that interworks with other water systems deployment, additional functional requirements for the NGSI-LD system have been derived as Table 5.6-1.

Table 5.6-1: Potential API Requirements

Number	Requirement
API-1	The NGSI-LD System shall be able to support meta data to indicate synthetic/virtual data over physical data.
API-2	The NGSI-LD System shall be able to support common methods to associate synthetic data with physical data.

5.7 Conclusions and Recommendations

In water domain, which is still a single domain, data systems in the water cycle are fragmented. The Water DNA platform with the NGSI-LD standard interfaces and common data models integrate heterogeneous water data from different data sources to enable integrated urban water resource management. When the water data is managed by the standard compatible data platform, other water applications do not need to collect data each time with different interfaces/protocols.

From urban water management perspective, the other environmental data and smart city data should be considered for value-added service implementations in the future. Water data modelling which related to those domain data models and leveraging those Relationships with the extended standard interfaces would be the one of the way forwards.

6 Adoption Case Study 3: ODALA

6.1 Introduction

The Connecting Europe Facility (https://cinea.ec.europa.eu/programmes/connecting-europe-facility_en) is a key EU funding instrument to promote growth, jobs and competitiveness through infrastructure investment at European level. In this context, the ODALA (Collaborative, Secure, and Replicable Open Source Data Lakes for Smart Cities) project has been funded.

ODALA was launched in September of 2020 to run for a duration of three years. It is a collaboration between European cities, different private companies and research institutes, and the goal is to improve data management and sharing between cities and regions. According to Jonas Dageförde, Chief Digital Officer, City of Kiel, and coordinator of ODALA, the goal is to create "a virtual place to bring data together in a secure and trusted way and to make it usable for the city but also for third parties". A core goal is to facilitate decision making in public administrations, with a focus on mobility and environmental aspects of a smart city.

To achieve this, ODALA aims to provide a set of components for cities to pick and choose for their use cases. The ODALA Connected Broker implemented based on the Scorpio NGSI-LD Broker is at the core of the ODALA architecture as shown in Figure 6.1-1. Input is provided by many different sources and output is provided to many components and applications that extract higher-level information and make it available to users in a suitable way, enabling better decision making. ODALA uses NGSI-LD compliant Smart Data Models (https://smartdatamodels.org/).

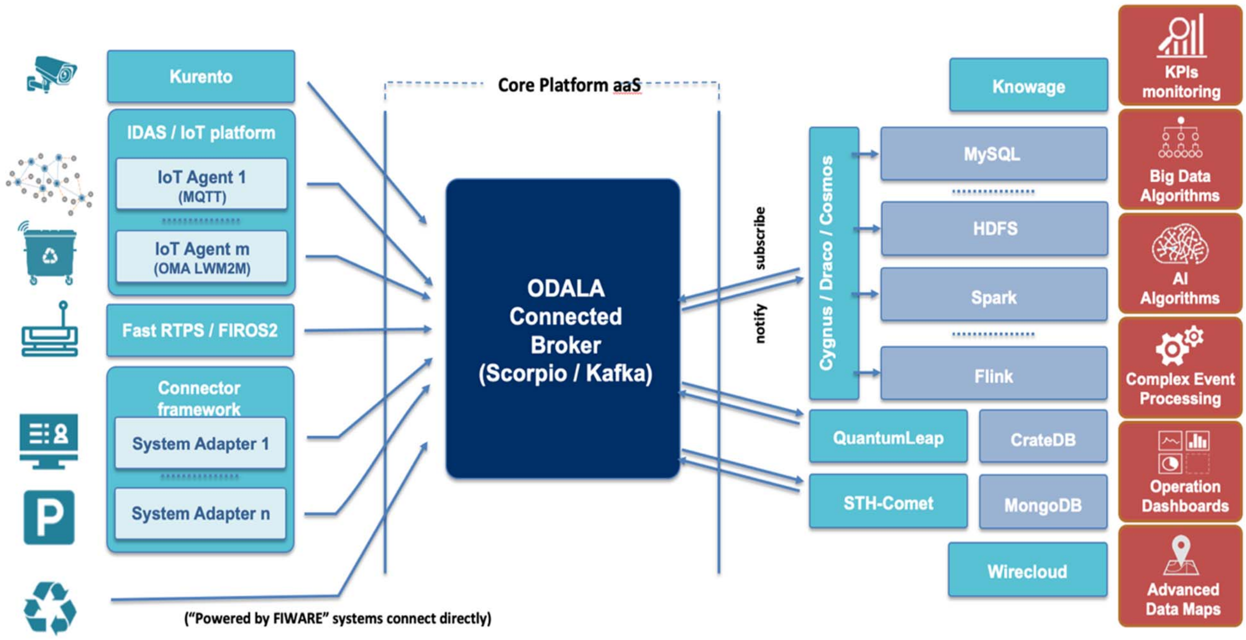

Figure 6.1-1: High-level ODALA Architecture

6.2 System Architecture

Figure 6.2-1 shows the layered ODALA System Architecture, where the Scorpio NGSI-LD Broker is part of the Core Data Management.

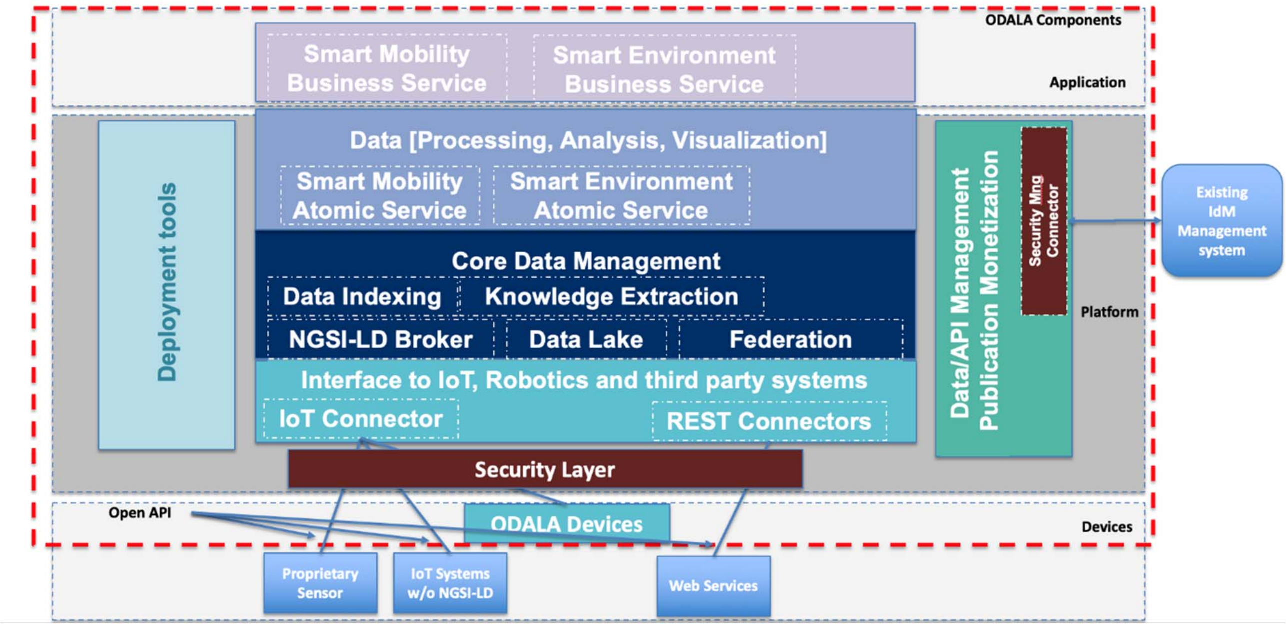

Figure 6.2-1: ODALA System Architecture

ODALA supports federated deployments across different organizational units, which can be different departments or areas within the same city or multiple cities or regions. ODALA has also defined a security framework for such a federated setup of NGSI-LD Brokers.

6.3 NGSI-LD Adoptions

The ODALA platform leverages NEC's open source Scorpio Broker. In general, the whole NGSI-LD functionality can be used. For federated setups, retrieval, query and subscription functionalities are supported, whereas the management of information (creation, update, deletion) is only supported for each broker individually, i.e. manage information locally, but make it available globally.

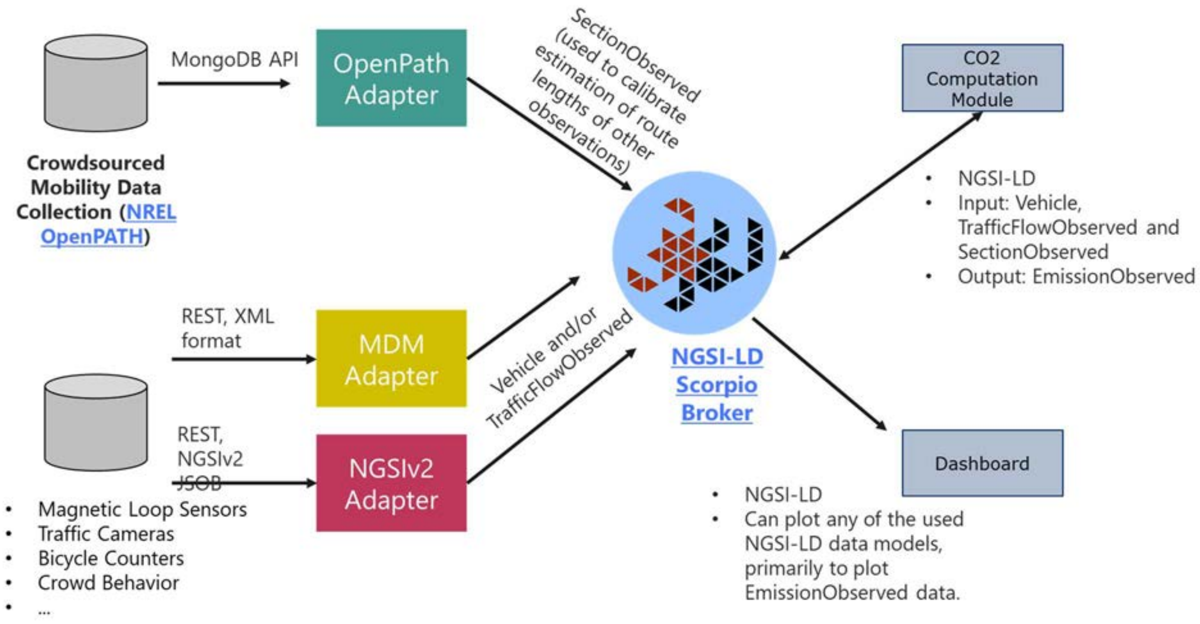

Figure 6.3-1: ODALA's Mobility Use Case

Figure 6.3-1 shows ODALA's mobility use case. Different sensors are used, e.g. magnetic loop sensors, traffic cameras and bicycle counters. These are provided to the NGSI-LD Scorpio Broker through adapters. In addition crowd sourced mobility data collection is used for calibration purposes. The provided information is used by a CO_2 computation module that feeds it information back into the NGSI-LD Scorpio Broker. The information is thus made available to a dashboard that is used for visualization.

6.4 Conclusions and Recommendations

The ODALA project uses an NGSI-LD Context Broker as the core of its data management, in a smart city and across smart cities. It has been shown that federated deployments of NGSI-LD Context Brokers are feasible, making information available globally, but keeping management locally - this is both related to the use case, enabling the city or even individual departments in control of its own data and to the available functionality, as, at the time of deployment, the Scorpio NGSI-LD Broker was the only one to support federated deployments, but at the time only for making information available in a federated setting. The latest implementation of the Scorpio NGSI-LD Broker supports all distributed operations, i.e. also distributed management would be possible.

7 Adoption Case Study 4: SALTED

7.1 Introduction

The Connecting Europe Facility is a key EU funding instrument to promote growth, jobs and competitiveness through infrastructure investment at European level. In this context, the Situation-Aware Linked heterogeneous Enriched Data (SALTED) project (https://salted-project.eu) has been funded.

SALTED aims to add value to existing datasets and data-streams by enriching them through the application of the principles of linked-data, semantics and Artificial Intelligence (AI). It uses Scorpio at its core and the smart data models to find/establish links between NGSI-LD Entities. SALTED is aiming to enrich existing entities based on their data model.

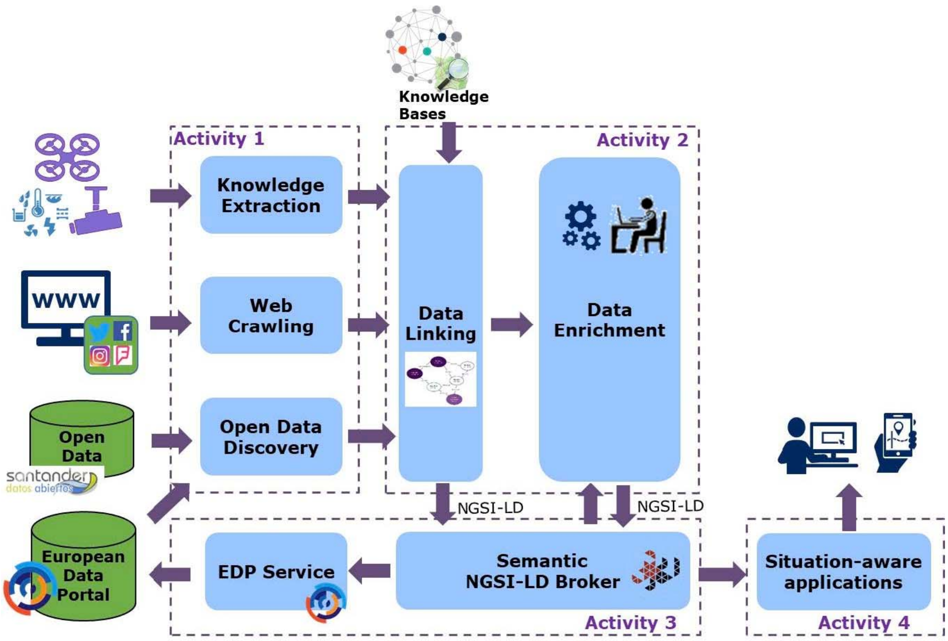

Figure 7.1-1: High-level SALTED Architecture

As shown in Figure 7.1-1, SALTED extracts data from many sources (IoT, open data, WWW), enriches and maps them to NGSI-LD entities, based on smart data models. The NGSI-LD entities are stored in the Scorpio NGSI-LD Broker. They data can be exported to the European Data Portal and used by Situation-aware applications.

7.2 System Architecture

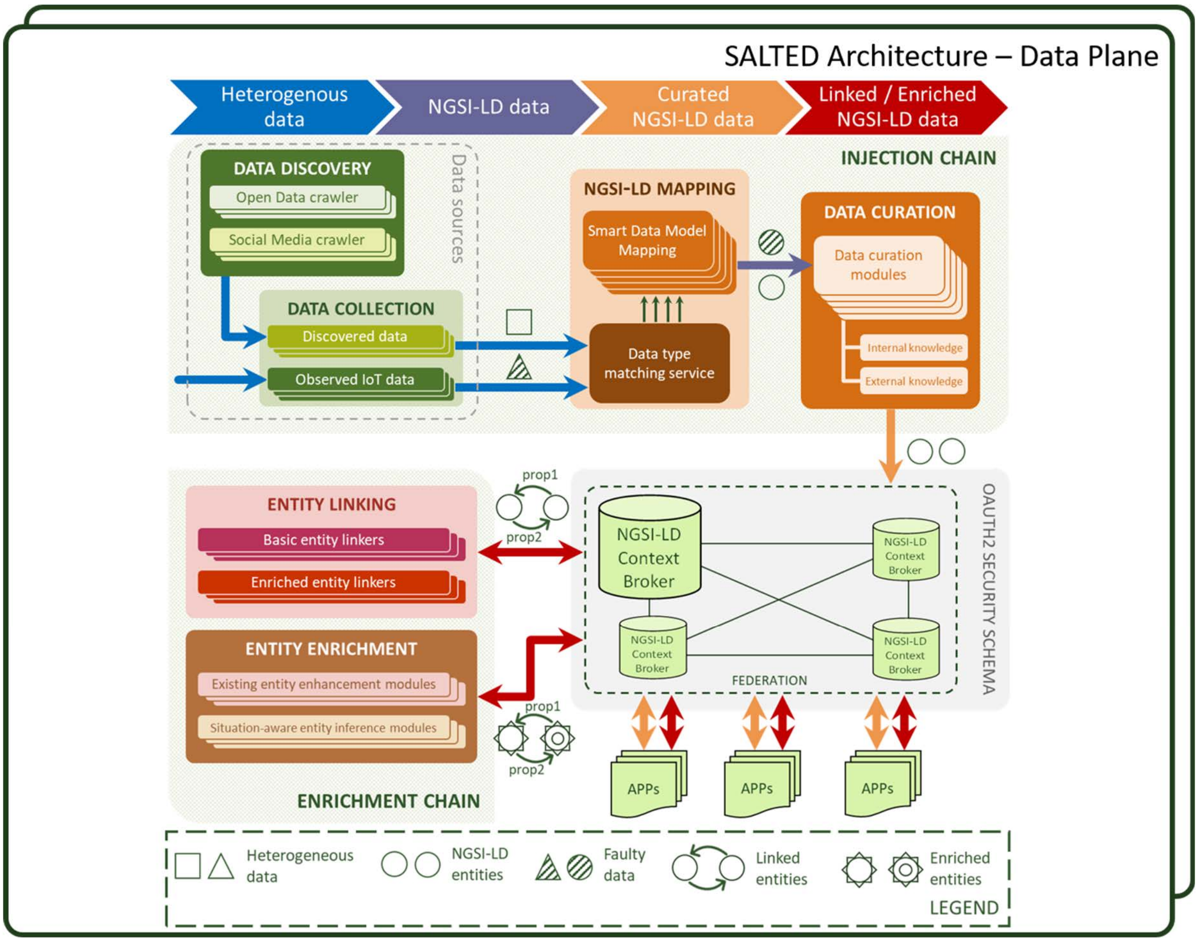

Figure 7.2-1: SALTED System Architecture

Figure 7.2-1 shows the SALTED System Architecture. At the core, a federation of Scorpio NGSI-LD Brokers is shown that make data accessible to applications and are used as a basis for entity enrichment and entity linking.

7.3 NGSI-LD Adoptions

SALTED leverages NEC's open source Scorpio Broker. In general, the whole NGSI-LD functionality can be used. As the project is still running, it uses an up-to-date version of the Scorpio Broker, which also supports management operations for distributed deployments. Still, the general setup assumes that data is injected locally, but made available globally to applications by a Scorpio Federation Broker as shown in Figure 7.3-1.

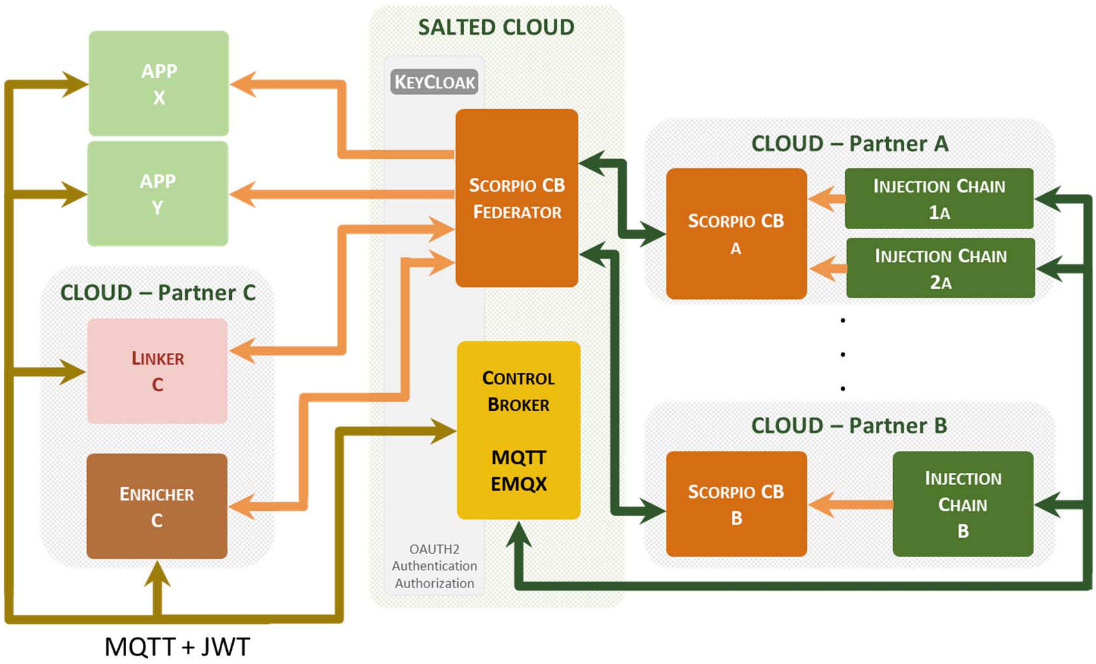

Figure 7.3-1: SALTED Deployment

7.4 Conclusions and Recommendations

The SALTED project uses an NGSI-LD Context Broker as the core of its data management, locally for each partner, mapping, enriching and linking entity information, and as a Federation Broker, making information available to applications, and providers of higher-level information that further link and enrich information. The Scorpio NGSI-LD Broker is supporting distributed operations, enabling the federation of NGSI-LD information from different partners.

The latest implementation of the Scorpio NGSI-LD Broker supports all distributed operations, i.e. not just distributed retrieval, but also distributed management would be possible.

8 Adoption Case Study 5: India Urban Data Exchange (IUDX)

8.1 Introduction

In late 2018, the Smart Cities Mission in the Government of India came together with the Indian Institute of Science, Bangalore to jointly develop and deploy the India Urban Data Exchange (IUDX). IUDX is now deployed as a production cloud service in the cities of Surat, Varanasi and Pune and in more than 30 cities in India. The intent is to make IUDX broadly available across all Indian cities in the next coming years, with an ecosystem of collaborating partners. The development of IUDX as described in (iudx.org.in) is collaborative with contributors from various organizations.

IUDX is and will be completely open source, based on an underlying framework of open APIs, data models, and the security, privacy and accounting mechanisms will facilitate an easy and efficient exchange of data among disparate urban data silos. IUDX draws on ideas and, where feasible, code from best-of-breed global projects such as FIWARE (https://www.fiware.org). However, it is skewed towards the Indian ecosystem, cultural norms, city nomenclature, payment and identification systems, etc. Therefore, the solution is particularly suitable for Indian Smart Cities.

IUDX is and will facilitate secure, authenticated and managed exchange of data amongst various data platforms, 3rd party authenticated & authorized applications and other data sources, data producers and consumers, within a city to begin with and scaled up across cities eventually at a national level, in a uniform & seamless way. The platform gives the data owners full control of what data to expose and to whom. Built-in accounting mechanisms enable connection with payment gateways to form the foundation for a data marketplace. The whole platform is developer friendly, via definitions of open APIs (application program interfaces) and data schema templates (formats for interpreting data), so that a whole new application ecosystem gets created.

IUDX addresses two key stakeholders: data providers and data consumers. A data provider is the provider/custodian of the data resources made available through the data exchange. A data consumer consumes the data resource available through the data exchange for the purpose of application/solution development. A data provider has full authority to control access and provide authorizations to consumers wishing to access data for resources it controls. IUDX strives to enable seamless, secure access to data to the consumers while respecting access control policies set by the provider.

The key concerns of the data providers are as follows:

- Be able to control access to its data resources by authorized consumers only.
- Be able to onboard data into the exchange in a secure fashion.
- Be able to provide and manage meta-information about its data resources in order to improve discoverability, data usage and data interoperability (to make its data resource more attractive to a wide spectrum of data consumers).
- Be able to interact with the data exchange in an easy and secure fashion.

Key concerns of data consumers are as follows:

- Be able to discover data resources of interest.
- Be able to access the relevant data from the available data resources in a secure fashion.
- Be able to get access tokens in case authorization is required by the provider of a resource.
- Be able to understand the data from a resource to be able to use it easily in application development.
- Be able to get additional context (meta-information) for the data (e.g. to be able to use data from different resources, to be able to connect and correlate data from different sources, etc.).
- Be able to interact with the data exchange in an easy and secure fashion.

8.2 System Architecture

8.2.1 Architecture Overview

IUDX addresses the above concerns by implementing the interfaces and open APIs as described in the 'Unified Data Exchange Architecture' specifications. IUDX is an open data exchange platform and consists of three main components:

- Catalogue service provided by a Catalogue server.
- Authorization and Authentication services are provided by an Authorization server.
- Data access and data ingestion services provided by one or more Resource Servers.

Data resources managed by a data provider are hosted on one or more Resource Servers. A data consumer can access a data resource via open and standard data access APIs. The Resource Servers also provide publication services enabling data providers to ingest data from their respective resources.

A data consumer can discover the data resources relevant to its application using the search APIs provided by the catalogue service. The catalogue hosts information (e.g. data formats, units, type of resource, etc.) for each data resource. This information is registered and managed completely by the provider of the given resource using open management APIs provided by the catalogue service. The meta-information, which is both human and machine understandable, enables the consumer to understand data and get additional context required for intelligent data usage.

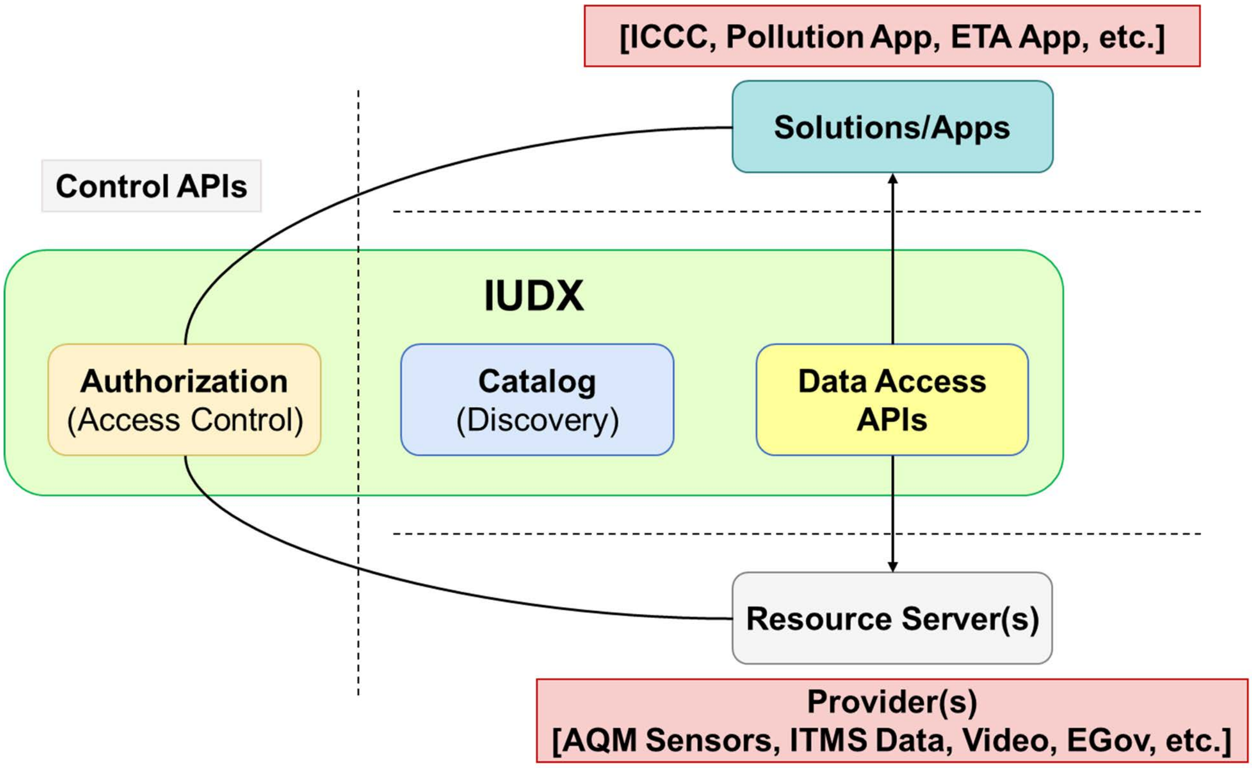

Figure 8.2-1: IUDX Reference Architecture

The Authorization server provides management APIs to register and modify access control policies associated with a protected data resource. The authorization server also provides services to get access tokens in case the data consumer needs to access a protected resource. Using token validation services provided by the authorization server, a resource server can ensure compliance with the data access policy set by the provider.

8.2.2 Catalogue Server

The main functions of the IUDX catalogue service are:

- Search and discovery of data resources.
- Provide an unambiguous description of data from a given resource thereby leading to improved data understandability and interoperability.
- Provide additional context for a given resource that may improve its usage in applications.

The IUDX catalogue is a store of meta-information associated with the data resources available in the data exchange. Some illustrative meta-information attributes associated with a resource are:

- associated provider details;
- resource server details where the data can be accessed;
- location of a sensor device;
- descriptions associated with the resource;
- datasheets, if applicable;
- information whether the resource is public or private.

The meta-information for a given resource is provided and managed by the data provider of the associated resource. The catalogue uses concepts of linked data to provide semantic contexts for the attributes describing meta-information which leads to improved machine readability, interpretability, and operational interoperability and enables vocabulary reuse from other data-model stores and taxonomies. The catalogue items are stored using JSON-LD format, which is a lightweight linked-data format based on the existing and popular JSON format and is easily readable and writable by humans. Furthermore, each catalogue item adheres to a schema that specifies a set of mandatory attributes and that may contain additional custom attributes. A catalogue item may be related to another catalogue item by providing explicit references to one another.

One of the key meta-information objects is the 'data descriptor' object which unambiguously describes each attribute contained in the data from the associated resource. It may contain descriptions of data types, units, value constraints, text descriptions, semantic context etc., for these attributes. The use of JSON-LD enables linking each attribute to a vocabulary, thus providing semantic understanding for these attributes in the data. The IUDX catalogue services, built on top of the meta-information store, provides APIs for:

- Search and discovery: Text search, Geo-spatial search, Attribute search and Relationship search.
- Management: Create, read, update and delete meta-information objects.

8.2.3 Authorization Server

The IUDX Authorization Server allows data sharing while respecting ownership, privacy and compliance requirements. The IUDX authorization server ensures that only the authorized person can get tokens to access the private/protected data.

The main functions of the authorization service are:

- Resource access authorization to grant access to access-controlled resources.
- Resource access policy management to manage policies which specify access rules.
- Authentication and registration services.

To access a protected data resource, the consumer needs to present a valid authorization token to the resource server. A token can be obtained by using token access APIs provided by the authorization server. Any access token request for a resource by a consumer will be checked against the existing access control policies. These policies are set and managed by the provider of the resource. If the policy decision is a success, an access token is provided to the consumer using which a data access request can be made. The access token is also logged to ensure auditability of consent grant. If no policy exists or no decision can be made, the provider and the consumer may negotiate, outside the scope of IUDX, to agree to consent terms after which an access policy can be set by the provider. The authorization service also provides token introspection APIs using which a resource server can check the validity of a token presented by the consumer at the time of data access.

IUDX specifications do not mandate any policy authoring language. In general, a policy should have a simple structure and should be able to specify: 'who' can access, 'what' resource and for 'how long', etc. IUDX consumer and provider need to authenticate themselves while interacting with Authorization server using authentication credentials obtained during registration process. Currently, IUDX uses OpenID Connect (OIDC) based authentication where the users are provided with ClientID and Client Secret during registration. Note that the entities need not authenticate themselves while interacting with a resource server or the catalogue server where all interactions with protected resources are secured using access tokens.

The authorization service provides APIs for:

- request an access token for a given resource;
- manage access policies for a given resource;
- list current access policies for a given resource;
- validate an access token;
- revoke an access token;
- audit access tokens.

To summarize, the authorization service enables asynchronous authorization grants and enables data providers and application developers to target a consistent set of APIs for authoring policies and accessing data for protected data resources.

8.2.4 Resource Server

The resource server constitutes the data plane for IUDX and provides data access for the resources available with the exchange. It serves data to data consumers in compliance with the access policy requirements set by the provider of the resource. For this compliance, the resource server is implementing a token introspection interface with the authorization server. For any access on a protected resource, the resource server requires the consumer to present an access token which it validates using token introspection APIs before serving the data.

Apart from data access APIs, AMQP subscriptions are supported for streaming data sources. In the future, support for callback subscriptions may also be provided. Data is always served over secure and encrypted channels using TLS channels.

The resource server may also provide data ingestion endpoints to allow data to be published into the resource server by data providers. IUDX resource access APIs are harmonized with ETSI NGSI-LD Specifications for data access.

The data access service provides APIs for:

- Search and count: Get data using spatial, temporal, attribute and complex searches.
- Subscriptions: Get access to streaming data using AMQP.

8.3 NGSI-LD Adoptions

IUDX's resource server uses the NGSI-LD APIs. The following are the APIs that are used

Table 8.3-1: NGSI-LD API Endpoints supported by IUDX

REST Resource	Method	Notes
/ngsi-ld/v1/subscriptions	POST	DATA SUBSCRIBER
/ngsi-ld/v1/subscriptions/{id}	GET	DATA SUBSCRIBER
/ngsi-ld/v1/subscriptions/{id}	DELETE	DATA SUBSCRIBER
/ngsi-ld/v1/subscriptions/{id}	PATCH	DATA SUBSCRIBER
/ngsi-ld/v1/entityOperations/query	POST	ENTITIES POST QUERY
/ngsi-ld/v1/temporal/entityOperations/query	POST	ENTITIES POST QUERY
/ngsi-ld/v1/entities/{id}	GET	LATEST ENTITY
/ngsi-ld/v1/entities	GET	SPATIAL ENTITIES
/ngsi-ld/v1/temporal/entities	GET	TEMPORAL ENTITIES

8.4 Extensions

8.4.1 API response extensions

The IUDX has introduced some new APIs and also incorporated more detailed error responses and standardized the response format based on feedback received from the developers. Table 8.4-1 shows the extensions implemented.

Table 8.4-1: IUDX extended responses

URN	Description
urn:dx:rs:success	Successful operation
urn:dx:rs:InvalidTemporalParam	Invalid temporal param for the given query
urn:dx:rs:InvalidTemporalRelation Value	Invalid temporal param value for the given query
urn:dx:rs:InvalidTemporalDateFormat	Invalid temporal param value date format for the given query
urn:dx:rs:InvalidGeoParam	Invalid geo param for the given query
urn:dx:rs:InvalidGeo Value	Invalid geo param value for the given query
urn:dx:rs:InvalidAttributeParam	Invalid attribute param for the given query
urn:dx:rs:InvalidAttribute Value	Invalid attribute param value for the given query
urn:dx:rs:InvalidOperation	Operation requested in the endpoint is not permitted
urn:dx:rs: UnauthorizedEndPoint	Access to the endpoint is not available
urn:dx:rs: UnauthorizedResource	Access to the resource is not available
urn:dx:rs:ExpiredAuthorizationToken	Token has expired
urn:dx:rs:MissingAuthorizationToken	Token needed and not presented
urn:dx:rs:InvalidAuthorizationToken	Token is invalid
urn:dx:rs:ResourceNotFound	Document of given id does not exists
urn:dx:rs:MethodNotAllowed	Method not allowed for given endpoint
urn:dx:rs: UnsupportedMediaType	Requested/Presented media type not supported
urn:dx:rs:responsePayloadLimitExceeded	Search operations exceeds the default response payload limit
urn:dx:rs:requestPayloadLimitExceeded	Operation exceeds the default request payload limit
urn:dx:rs:requestOffsetLimitExceeded	Operation exceeds the default value of offset
urn:dx:rs:requestLimitExceeded	Operation exceeds the default value of limit

Apart from the response codes specified in each API, the IUDX's context broker also responds with certain 4xx and 5xx error codes which are related to common API responses. The response schema has been standardized to have Type, Title and Results for 2xx calls and Type, Title and Details for 4xx or 5xx responses.

8.4.2 Additional APIs

[High frequency streaming & subscription API]

This additional API is an endpoint to register context sources sending and receiving data using streaming protocols (e.g. AMQP, MQTT, WebSocket). This is used for High Frequency Positioning. For example Ambulance positioning data.

The end point has both:

- Registration of Streaming context sources.
- Subscription of Streaming context sources.

[End to End encryption between Broker & Context Consumer]

This additional API can be used to securely communicate with consumers to receive data in an encrypted format and maintain data integrity. When a context consumer queries with the public key in the header the context broker encrypts the data and sends it back to the context consumer.

{"type":"urn:dx:rs:success", "title":"Success", "results":["encrypted-data"]}

[Async API]

Async APIs provide the ability to the user to query resources using larger temporal periods and greater spatial area. This will help to reduce the load on the context brokers. For example, if a query response is 1GB then it is not wise to send the data over the API response. Instead, a file will be created.

There are 2 endpoints:

- Search API which will return the Search ID based on temporal, spatial, attribute query submitted.
- Download data API which will return a file URL corresponding to the Search ID.

More information at https://rs.iudx.org.in/apis#tag/Async.

[Auditing/Metering API]

Auditing API allows context producers and consumers to understand the utilization of data and APIs using temporal queries. For example, an app developer (context consumer) can understand the total API calls made for an entity through the application to limit the access and also can understand the total number of consumers consuming their data.

This API has been very useful while introducing data marketplace where users can buy and sell data.

More information at https://rs.iudx.org.in/apis#tag/Metering.

8.5 Interworking and Modelling Requirements

The IUDX team together with IIIT, Hyderabad (International Institute of Information Technology, Hyderabad), has contributed to an architecture for Smart City Living Lab, which includes a live testbed to deploy, test and generate smart city solutions. A pilot was created under the project combining IUDX and oneM2M components.

As illustrated, live data that is received from different nodes for analysis through a live dashboard at the Smart City Living Lab. The data is stored in the Data Storage layer for long term storage and analysis.

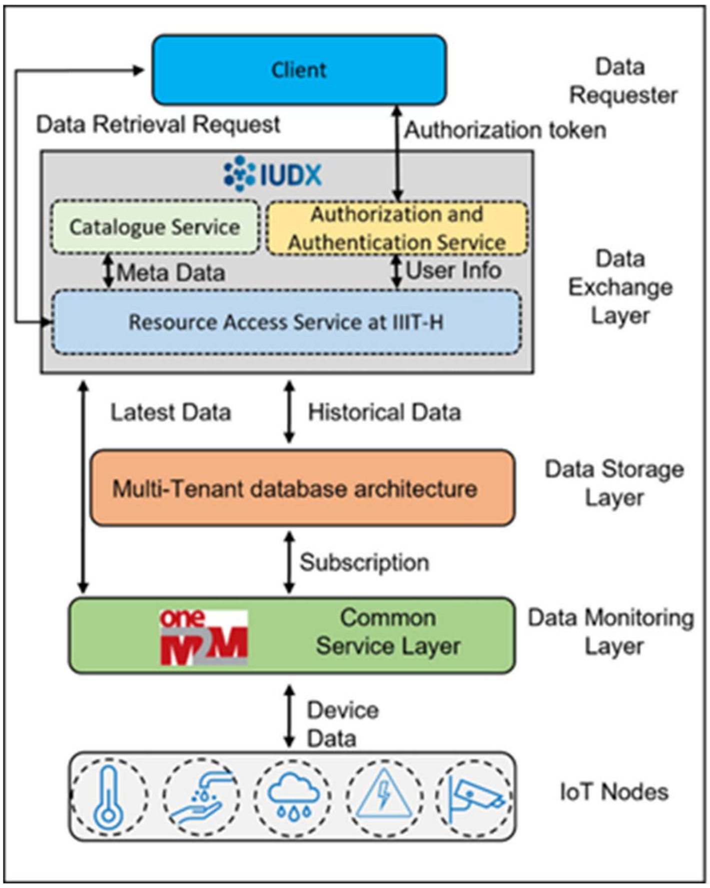

Figure 8.5-1: Smart City Living Lab (IIIH, Hyderabad)

The architecture uses the powerful interoperability of oneM2M capabilities at the IoT layer, and IUDX's components (Data Broker and Authentication) provide the benefit of NGSI-LD APIs and Data Models. This reduces time required to develop similar use cases in different places. In the context of the Indian market and smart city opportunities, the benefits of standardization are obvious.

9 Conclusion

In the present document, different system deployments are presented to show several global NGSI-LD adoption cases. Many of the cases cover smart city systems which necessarily require data management for existing systems and infrastructures in smart cities. Also, water domain adoption has been illustrated and that case proves usability of NGSI-LD based data management for fragmented legacy water management systems. In India, it has been proved that NGSI-LD can be used as a data marketplace enabler.

As the early deployments, the adoption cases so far use open source implementations. There are more open sources other than City Data Hub from KETI and Scorpio from NEC as described in the present document. It has been shown that the core APIs such as Entities CRUD operations are commonly used. Also in European cases, distributed operations over federated architecture configuration also has been proved.

As well as the standard API implementations from open sources, common data models have been used from Smart Data Models and also contributed to that data model initiative. Since NGSI-LD standard does not define domain specific data models, encouraging use of common data model needs to be highlighted.

Specifications with open sources and common data models are proven to be the key to global disseminations. Since there are more on-going NGSI-LD adoptions, there would be more global adoption cases written in the present document for others considering NGSI-LD for their systems.

History

Document history		
V1.1.1	January 2024	Publication