STREAMING FLEXIBILITY TO THE POWER SYSTEM

D3.1 sDATA OPEN-PLATFORM DEVELOPED

Delivery Date: September 2024

STREAMING FLEXIBILITY TO THE POWER SYSTEM

DISCLAIMER

The work described in this document has been conducted within the STREAM project. This document reflects only the STREAM Consortium view, and the European Union is not responsible for any use that may be made of the information it contains.

COPYRIGHT STATEMENT

This project has received funding from the European Union's Horizon Europe research and innovation programme under the grant agreement N° 101075654.

More information available at https://stream-he-project.eu/

Title	Document Version
D3.1 sDATA open-platform developed	0.6

Project Number	Project Acronym	Project Title
Grant agreement N°: 101075654	STREAM	STREAM: Streaming flexibility to the power system

Contractual Delivery Date	Actual Delivery Date	Deliverable Type*-Security**
September 2024 (M24)	September 2024 (M24)	R-PU

*Type: P: Prototype; R: Report; D: Demonstrator; O: Other.

**Security Class: Public - fully open (automatically posted online); Sensitive - limited under the conditions of the Grant Agreement; EU classified - RESTREINT-UE/EU-RESTRICTED, CONFIDENTIEL-UE/EU-CONFIDENTIAL, SECRET-UE/EU-SECRET under Decision 2015/444

Responsible	Organisation	Contributing WP
Andraž Žertek	IND	WP3

Authors (organisation)

Andraž Žertek (IND), Valerija Jordan Tršinar (IND), Denis Sodin (COMS), Daniel Pastor Pascual (ETRA), Diego García-Casarrubios Gálvez (ETRA), Klemen Peter Kosovinc (KOL), Gašper Artač (KOL), Borut Jereb (KOL), Gašper Mis (KOL), Dušan Krovinovič (KOL), Bernardo Di Chiara (OPT), Alessio Cavadenti (ASM), Prashanth Kumar Pedholla (ASM), Janez Gregor Golja (IRI UL), Jan Jeriha (IRI UL), Gregor Skrt (EPR), Efrén Guilló Sansano (ENER), Manuel Rebollo Reviriego (OMIE), Janko Blatnik (EPR), Jan Jokelainen (OPT)

Abstract

This deliverable outlines the work and contributions done in the scope of Task 3.1 and Task 3.2, which include the sDATA requirements and specifications, based on a review of previous projects, Open Data Spaces, and existing standards along with an analysis of use cases, data users' needs and the data to be exchanged within the sDATA platform at each pilot site.

Keywords

sDATA, data exchange, Open Data Spaces, interoperability, security, data sovereignty, Open APIs

Revision	Date	Description	Author (Organisation)
V0.1	06.05.2024	New document & first version of the Table of Contents	Andraž Žertek (IND), Valerija Jordan Tršinar (IND)
V0.2	16.08.2024	Overview of similar previous projects, Open Data Spaces overview and IEC Standards overview	Denis Sodin (COMS), Daniel Pastor Pascual (ETRA), Valerija Jordan Tršinar (IND)
V0.3	06.09.2024	Merge of the pilot sites' contributions into the document and general chapters – preparation of the document for the internal review	Klemen Peter Kosovinc (KOL), Gašper Mis (KOL), Dušan Krovinovič (KOL), Bernardo Di Chiara (OPT), Denis Sodin (COMS), Alessio Cavadenti (ASM), Daniel Pastor Pascual (ETRA), Diego García-Casarrubios Gálvez (ETRA), Valerija Jordan Tršinar (IND), Janez Gregor Golja (IRI UL), Jan Jeriha (IRI UL), Gregor Skrt (EPR)
V0.4	18.09.2024	Internal review	Ismo Heimonen (VTT), Janez Gregor Golja (IRI UL), Daniel Pastor Pascual (ETRA), Klemen Peter Kosovinc (KOL), Darja Skrt Georgiev (CYG)
V0.5	25.09.2024	Review comments addressed	Valerija Jordan Tršinar (IND), Bernardo Di Chiara (OPT), Daniel Pastor Pascual (ETRA), Jan Jeriha (UL)
V0.6	26.09.2024	Final version, ready for review	Valerija Jordan Tršinar (IND)

EXECUTIVE SUMMARY

This document outlines the **development of the sDATA tool**, guided by two tasks within WP3: Task 3.1 ICT Architecture and Task 3.2 Data driven open platform. These tasks focused on defining the sDATA **requirements** and **specifications**, based on a review of **previous projects**, **Open Data Spaces**, and **existing standards**. Each pilot site analysed **use cases** defined in WP2, **data users' needs**, and **data** to be exchanged within the sDATA platform. All pilots contributed to defining the requirements and specifications for data exchange, which led to the implementation of the sDATA tool. The requirements and specifications also demonstrate compliance with EU regulations regarding security and privacy. This document also covers **edge computing**, implemented separately from sDATA, as sDATA is used solely for data exchange for analytical purposes.

Despite **differing use cases and tools across pilot sites**, a set of **general requirements** was established, categorized into **functional** and **non-functional** groups. The sDATA tool **specification** was then defined for each pilot site, with a unified specification for the IT and SI pilots, where sDATA was developed by the same project partner.

In addition to functional and non-functional specifications, the document describes the tool's **architecture**, **Open APIs**, and alignment with **Open Data Spaces requirements and principles**. The pilot sites implemented **different architectural solutions**, demonstrating that identical requirements can be met with different architectures while **maintaining core principles**. However, in both SI and IT pilots, the same sDATA solution was implemented, highlighting its **flexibility, interoperability, and adaptability for other applications** despite varied use cases, data clients, and types.

The sDATA platform also supports the **development of other STREAM tools**, such as sFLEX and sGRID, by providing **reliable** and **consistent data**.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 PURPOSE OF THE DOCUMENT

This document provides an overview of the development of the sDATA tool. The sDATA tool is a data-driven platform that facilitates data exchange among multiple stakeholders. The development process was guided by tasks T3.1 ICT Architecture and T3.2 Data driven open platform. The implementation of sDATA enables other tools that exchange data with it to have access to all the necessary data required for performing tasks and analyses in the ongoing and future developments within the scope of the STREAM project and potential applications in other relevant sectors.

All pilots have contributed to the related tasks. Their contributions helped shape the general requirements that all pilot sites follow, ensuring the sDATA solution is robust enough to address diverse use cases and data needs.

1.2 SCOPE OF THE DOCUMENT

This document covers a range of topics, including an overview of previous projects and Open Data Spaces, the identification of UCs and data user needs, the definition of requirements and the complete specifications of the tool.

1.3 STRUCTURE OF THE DOCUMENT

- **Section 2** reviews previous projects, Open Data Spaces and IEC standards to identify best practices and outlines the specific requirements for sDATA, derived from an analysis of UCs, data, and user needs.
- **Section 3** outlines pilot-specific implementations based on requirements from Section 2. It includes functional specifications tailored to each pilot, non-functional specifications, and system specifications, covering architecture, Open APIs, and the sDATA solution's compliance with requirements and principles of the Open Data Spaces.
- **Section 4** concludes the document outlining the key findings and insights.
- **Annex A** – Complete list of sDATA requirements contains the complete list of sDATA requirements of the pilots.

2 METHODOLOGY AND REQUIREMENTS

The Methodology and requirements section introduces and motivates the development of the sDATA tool within the STREAM project.

The sDATA tool serves as a crucial data-driven platform that facilitates seamless data sharing and access control for end-users, enabling flexible interactions between upstream entities, such as flexibility aggregators, service providers, and DSOs.

As the foundation for data exchange within the STREAM ecosystem (Figure 1), sDATA provides data for analytics, user profiling, flexibility forecasting, demand and production insights, and the development of consumer services. This shared data is utilized by other STREAM tools, such as sFLEX and sGRID.

This section provides an overview of previous projects to extract lessons learned, good practices, and what has already been accomplished, forming the foundation for the tool.

Additionally, it includes an overview of Open Data Spaces, laying the groundwork for the development of sDATA based on their requirements and principles.

This section also outlines the general requirements of the sDATA tool, derived from a thorough analysis of UCs, data, and users' needs, which are also detailed in this section.

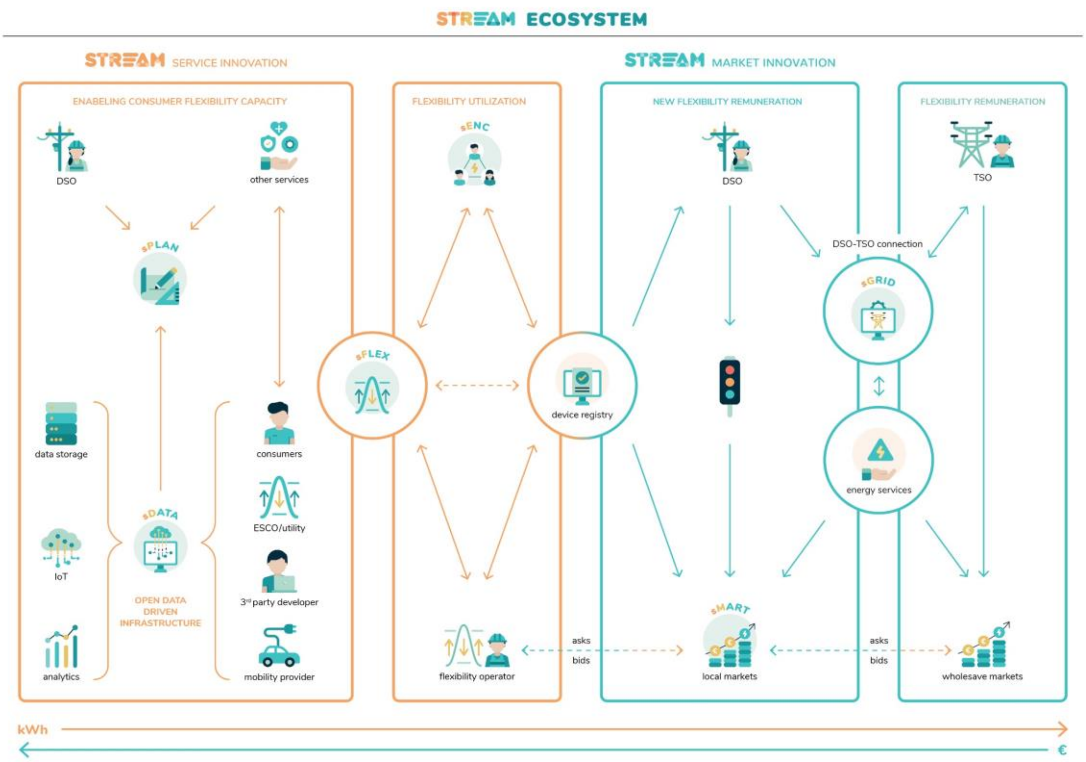

Figure 1: STREAM Ecosystem

2.1 OVERVIEW OF PREVIOUS PROJECTS

In this section, an overview of the data sharing and management practices that have emerged from the previous projects is provided. The analysis focuses on the methodologies, tools and frameworks used to ensure efficient and secure data processing. By examining past experiences, the aim is to identify best practices, highlight areas for improvement and provide a solid foundation for future initiatives. This overview covers the entire data management lifecycle, from collection and storage to processing and sharing, ensuring that the lessons learned are both practical and applicable.

2.1.1 BD4NRG

The data architecture of the BD4NRG project was presented in detail in the project's Work Package 3, deliverable 3.3 [1]. The document served as a master plan detailing the location, description, flow and availability of the data. This type of architecture was developed to support data quality, management, integration, delivery and collaboration across different systems and applications, as shown in the Figure 2. The architecture describes how the data moves from the raw data in the Large-Scale Pilots through various layers of transformation and quality checking to become the Golden Data Source and finally to the business intelligence and analytics applications.

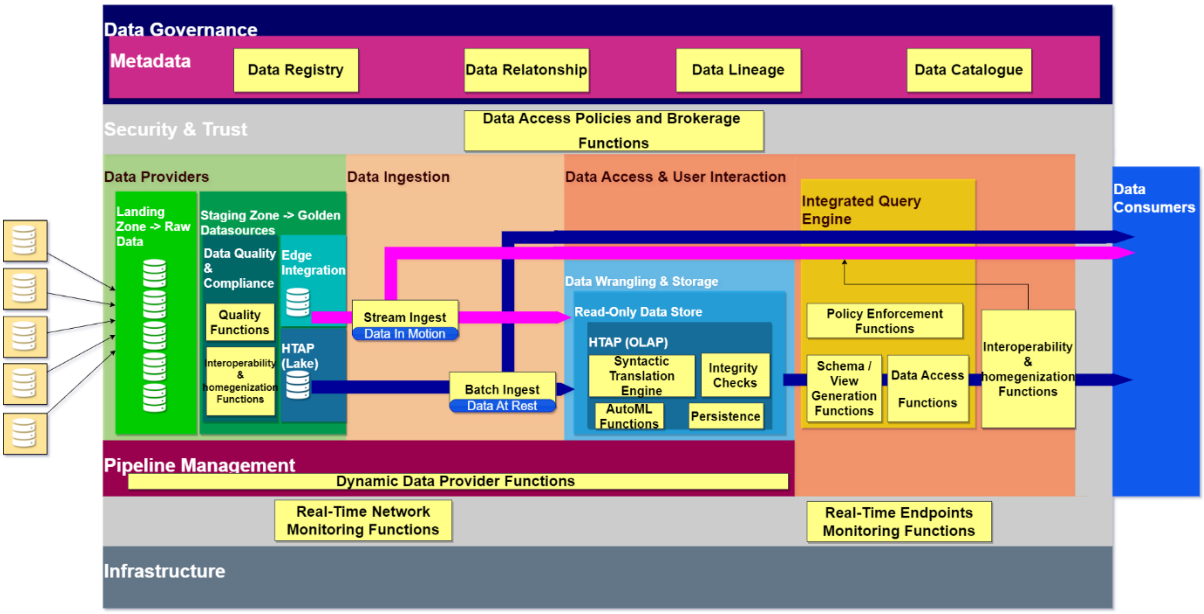

Figure 2: BD4NRG data system overall architecture, Source [1]

The purpose of each layer is briefly described:

- **Data Providers:** introduces new data into the BD4NRG Data system for discovery, transformation, analysis, and access. The data can come from different sources such as static files (e.g., Comma Separated Values (CSV) or JavaScript Object Notation (JSON)), social media, third-party services as well as sensory data from intelligent devices. Raw data is – within this layer – properly processed to create the so-called golden datasets. These datasets are unique data, including all the necessary meta-data, the required quality and following the created interoperability standards to be ingested by the BD4NRG data system.
- **Data Ingestion & Management:** where the data starts their journey i.e., how data can be delivered and distributed within the BD4NRG data system for immediate use or storage. Two main mechanisms are considered, namely: real-time streaming ingestion (for data in motion) and batch ingestion (for data at rest). The specific data ingestion mechanism strictly depends on the application UC.
- **Data Access & User Interaction:** facilitates access and navigation of the data, allowing the generation of different business views to be represented by the data. At the centre of this layer lies a high-performance real time analytics database to support interactive dashboards and analytics platforms while being used as data warehouse. A query engine is also part of the layer and serves as Rapid Application Development (RAD) by presenting fast logical views of data to consumer applications.

- **Data Governance:** all considered data is channelized through data governance processes to ensure data stewardship, data classification, data transparency, data discovery, data ownership and data quality by adding business and technical metadata.
- **Security & Trust:** is responsible to support and maintain security and trust beyond anonymization and privacy. Data access policies are here defined and enforced as well as dedicated monitoring functions will be also deployed at infrastructure level.
- **Infrastructure:** is the core infrastructure of the BD4NRG data system. The layer deals with networking, computing and storage needs to ensure that large and diverse formats of data can be stored and transferred in a cost-efficient, secure and scalable way. The key requirement of any Big Data storage is that it can handle massive quantities of data and that it keeps scaling with the growth of the organization, and that it can provide the input/output operations per second (IOPS) necessary to deliver data to applications.

2.1.2 BRIGHT

A detailed description of the BRIGHT consortium's approach to interoperability, in particular the standardization of cross-domain access to data and the standardization of service interoperability, can be found in the project’s Deliverable D2.4 – Cross domain data & service interoperability [2]. This document sets out the methodology for harmonizing the approaches of the different stakeholders. The main objective of the document was to identify and analyse different standards based on the needs of pilots and service developers. The document also presents clearly defined approaches with practical examples focusing on the following:

- An overview of existing technologies.
- The creation of a BRIGHT-specific interoperability solution.
- The analysis of different data models, from low-level data models used in data sources to high-level models.
- Collecting and harmonizing information about existing and planned information objects.
- Preparation of the harmonized data for the development of a BRIGHT data model solution.

In BRIGHT, interoperability refers to the system's ability to map different data sources to the common data model. Such a system can freely create, consume and exchange data, knowing exactly the context and meaning of the data. To achieve interoperability of data from the data source in BRIGHT, the components of the data flow were similar to those in the BD4NRG project:

- **Data source:** is any source that generates data. This can be simple devices (sensors, Intelligent Electronic Device (IED), Internet of Things (IoT) devices, Energy Management Service (xEMS) devices, etc.), databases or even non-technical data acquisition obtained through various methods. For example, the description of socio-economic aspects using analytical methods or the manual direct or indirect collection of data from stakeholders using various questionnaires.
- **Data Acquisition:** This is the step responsible for integrating heterogeneous data sources and facilitating their integration. The data acquisition component must handle the so-called 3Vs: Volume, Velocity and Variety. In addition, the data can be structured or unstructured and can come from different sources with limited connectivity.
- **Data curation:** The next step in the typical data flow is the process that can be carried out with the raw data or big data. It is responsible for pre-processing the data such as cleansing, anonymization and semantic enrichment.

- **Data storage:** Various types of data storage are required. Therefore, most of them need to be considered to achieve full interoperability. In most cases, the data is stored in databases. However, the closer you get to the data source, the more likely other types of data storage come into question, which usually work with a small data package in real time (e.g. temporary ones such as registers and memories or persistent ones such as files). In the case of a shared data lake or distributed storage, it is important to achieve autonomous real-time or batch processing. Therefore, the storage should not only cover one database type, but many different database types such as relational, non-relational and time series databases.
- **Data security:** In parallel to all the components described above, data security is of paramount importance. Data protection must be guaranteed at all levels. In addition, anonymization must be considered with all possible ethical issues.

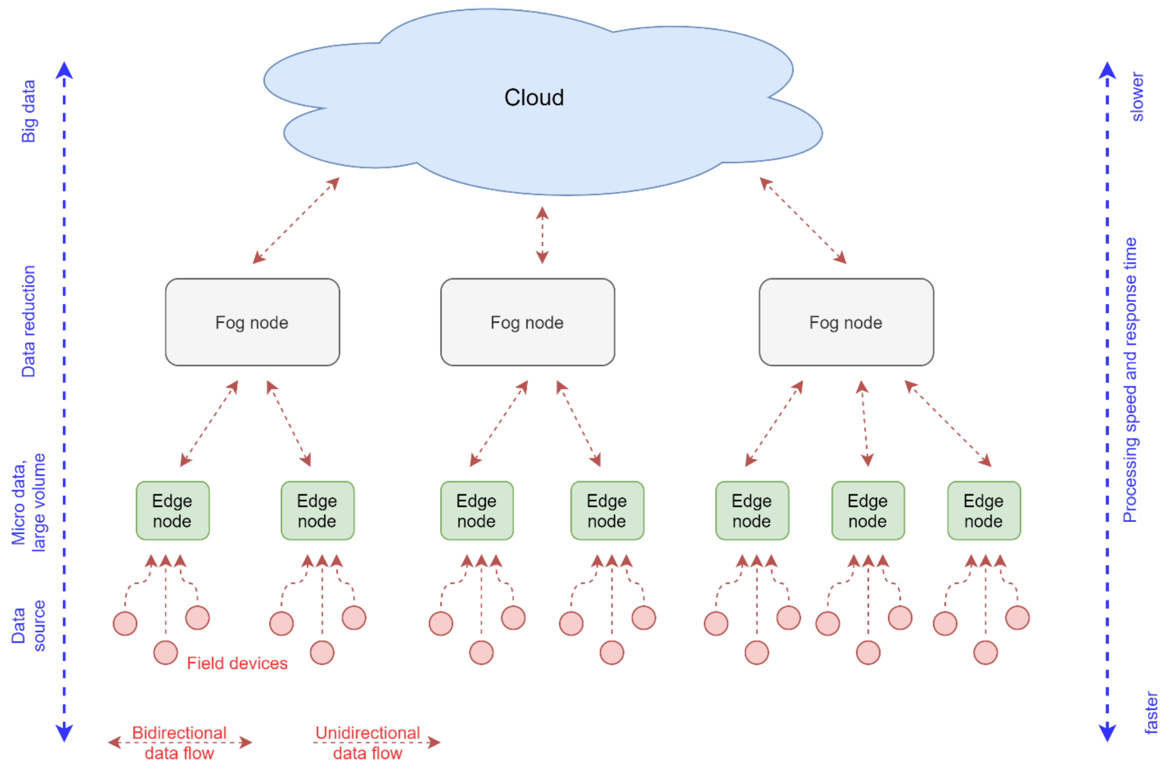

Figure 3: Processing layers in BRIGHT, Source [2]

Figure 3, taken from the deliverable mentioned above, shows the data layer stack. The data from the source (e.g. sensors) can transform and evolve as it passes through the different processing layers (edge, fog and cloud). To cope with the speed and achieve interoperability, the data has gradually evolved from raw data to big data, which was semantically enriched with selected ontologies during the process of acquisition and curation. Originally, the raw data was processed by sensors and other producers and temporarily stored at the edge. The data at the edge could be used for real-time processing and the creation of real-time loops. The processed data could be transferred to the fog or cloud for further processing to create better observability and awareness of the system as a whole. The fog layer extended the edge layer to include the cloud, but still ensured decentralized data processing. This provided a better overview of the overall system compared to the edge layer, while also creating an additional intelligent layer that brought cloud computing functions closer to the edge layer. With edge and fog computing, sensitive data could remain in the network and only non-sensitive data was transferred to the cloud for centralized processing. The cloud fulfils the need for fast access to large amounts of data and ensures easy scalability.

2.1.3 Platone

As part of the Horizon 2020 program "A Single, Smart European Electricity Grid", the Platone project tackles the issue of "Flexibility and Retail Market Options for the Distribution Grid" [3]. Power grids today are evolving from centralized, infrastructure-intensive transmission system operators (TSOs) to more adaptable and flexible distribution system operators (DSOs) that can handle a variety of renewable energy sources.

The Platone Open Framework, a layered architecture solution, shown in Figure 4, aimed to develop an open, flexible, and secure system that supports distribution grid flexibility and congestion management through new energy market models that involve multiple stakeholders, including DSOs, TSOs, customers, and aggregators. This open-source framework facilitates the secure and flexible integration of external solutions, such as legacy systems, and supports the incorporation of external services via standardized Open APIs. It is designed to comply with current regulations, making it easier for small power producers to get certified and sell excess energy back to the grid. Additionally, it includes an open-market system to connect with traditional TSOs.

The core features of the Platone Open Framework are interoperability, adaptability, and flexibility. These principles are integral to its design, ensuring that it includes multiple standardized and interoperable mechanisms for integration. The framework is composed of three main components, each of which can be used independently or together.

These are:

- **Market Platform,** a key component of the Platone Open Framework, is a blockchain-based system for managing the flexibility market involving TSOs, DSOs, aggregators, and customers. It registers and certifies all market operations within the blockchain service layer, ensuring transparency, security, and trustworthiness among all the market players. The platform also includes an innovative incentive mechanism for customer engagement based on blockchain technology, smart contracts, and tokenization.
- **Blockchain Access Layer (BAL)** which enables secure, standardized integration of energy data from physical infrastructure (smart meters, external servers), granting access to DSOs and other stakeholders. It collects and certifies various types of energy data using blockchain and smart contracts, ensuring data integrity and preventing tampering. Through smart contracts, it is possible to track who shared what, with whom, when, by what means and for what purposes in a verifiable fashion. The data, harmonized and certified, is accessible to stakeholders and external platforms via protocols like MQTT, HTTPS, and REST APIs. It includes the Blockchain Access Platform and the Shared Customer Database (SCD). In the Blockchain Access Platform, Data Management Tool is included for identity management and data access management and control. The blockchain technology within the BAL plays a key role in addressing important challenges, such as ensuring data privacy, managing data access and usage control and incentivizing secure data sharing.
- **DSO Technical Platform** helps DSOs manage the distribution grid securely, efficiently, and stably by evaluating the current grid state and activating local flexibility. This platform, built on Horizon 2020 project SOGNO's work, features an open-source extensible microservices architecture for easy deployment of additional services. The Data Bus layer within the DSO Technical Platform facilitates the integration of other Platone framework components and

external systems (e.g., DSO Management System) by establishing a direct connection to the traditional SCADA system used by the DSO, utilizing standard communication protocols.

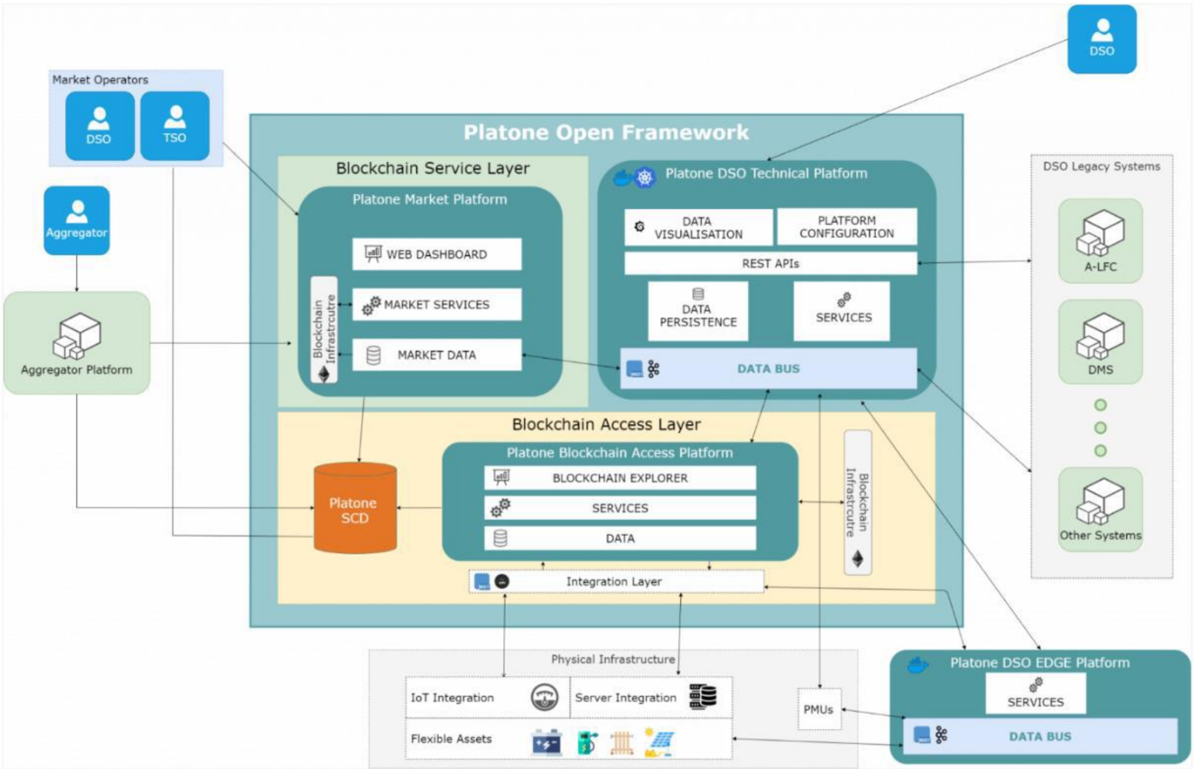

Figure 4: Platone Open Framework Architecture, Source [4]

2.2 OVERVIEW OF DATA SPACES

An energy flexibility market ecosystem such as the one that is being developed in the STREAM project requires the participation of several stakeholders (DSOs, TSOs, prosumers, aggregators, market operators (MO), etc.). The inclusion of such variety of stakeholders requires an extensive and challenging data exchange, with huge amount of data that needs to be managed somehow. Moreover, the data that is being exchanged across the different pilot sites is generally sensitive, since the misuse of these data may jeopardize the security of the energy supply, and data from consumers is protected by General Data Protection Regulation (GDPR). All in all, Energy Data Spaces stands as an innovative solution to tackle these challenges in data exchange when so many stakeholders and sensitive data are involved.

According to [5], 'a data space can be defined as a federated data ecosystem within a certain application domain and based on shared policies and rules. The users of such data spaces are enabled to access data in a secure, transparent, trusted, easy and unified fashion. These access and usage right can only be granted by those persons or organisations who are entitled to dispose of the data'. The European Commission, in their report on Common European Data Spaces [6], underlines Data Spaces as a key enabler to facilitate flexibility in energy delivery due to the interoperable and governable nature of Data Spaces, which will be further explored in the next sections. Indeed, governability and interoperability are fundamental to enabling stakeholders to easily access data, upon the required agreements with data owners, making possible the development of innovative solutions such as those being developed in this project.

2.2.1 State-of-the-art and reference architectures

In the following sections, the status of EU's most important initiatives and projects related to data spaces will be explored. The reference architectures and the requirements specific to Energy Data Spaces as defined in each initiative will be pinpointed, which sDATA should aim to fulfil in its development.

2.2.1.1 European Commission

The European Commission (EC) views Data Spaces as a cornerstone for achieving the digitalisation of the energy sector, while ensuring data security and compliance with GDPR, and the confidentiality of sensitive data, overcoming legal and technical barriers and ensuring data-driven innovation. The aim is to create a single market for data [7]. That is why the EC is promoting the creation of Common European Data Spaces, which will ensure the availability of data to be used by other stakeholders, while allowing data owners to keep control over their data. One of the common data spaces the EC supports is the Common European energy data space, 'to promote a stronger availability and cross-sector sharing of data, in a customer-centric, secure and trustworthy manner, as this would facilitate innovative solutions and support the decarbonisation of the energy system' [8].

In the first Staff Working Document of the European Commission [9], the EC documents the state of play of the Data Spaces that are being developed. This document gathers the features required for the development of a European Data Space, which are listed below:

- A secure and privacy-preserving infrastructure to pool, access, share, process and use data, favouring the development of common open standards and findable, accessible, interoperable and reusable (FAIR) principles.
- A clear and practical structure for access to and use of data in a fair, transparent, proportionate and/non-discriminatory manner and clear and trustworthy data governance mechanisms.
- European rules and values, in particular personal data protection, consumer protection legislation and competition law, are fully respected.
- Data holders will have the possibility, in the data space, to grant access to or to share certain personal or non-personal data under their control.
- Data that is made available can be reused against compensation, including remuneration, or for free.
- Participation of an open number of organisations/ individuals.

2.2.1.2 BRIDGE Data Management WG

The BRIDGE initiative is a cooperation group created by the EC that aims to foster the exchange of information, experience, knowledge and insights across the projects funded under the programs Horizon 2020 and Horizon Europe. These objectives are expected to be reached by encouraging the members to exchange their feedback, lessons learned and recommendations from the projects they have participated in. BRIDGE is formed by different activities named Working Groups (WG). Among them, the Data Management WG works on the Communication Infrastructure, Cybersecurity and Data Privacy, and Data Handling. To reach these objectives, the WG has elaborated and polished through the years the European Data Exchange Reference Architecture (DERA). The latest architecture defined so far is the third version of DERA, DERA 3.0, which is depicted in Figure 5 and explained below.

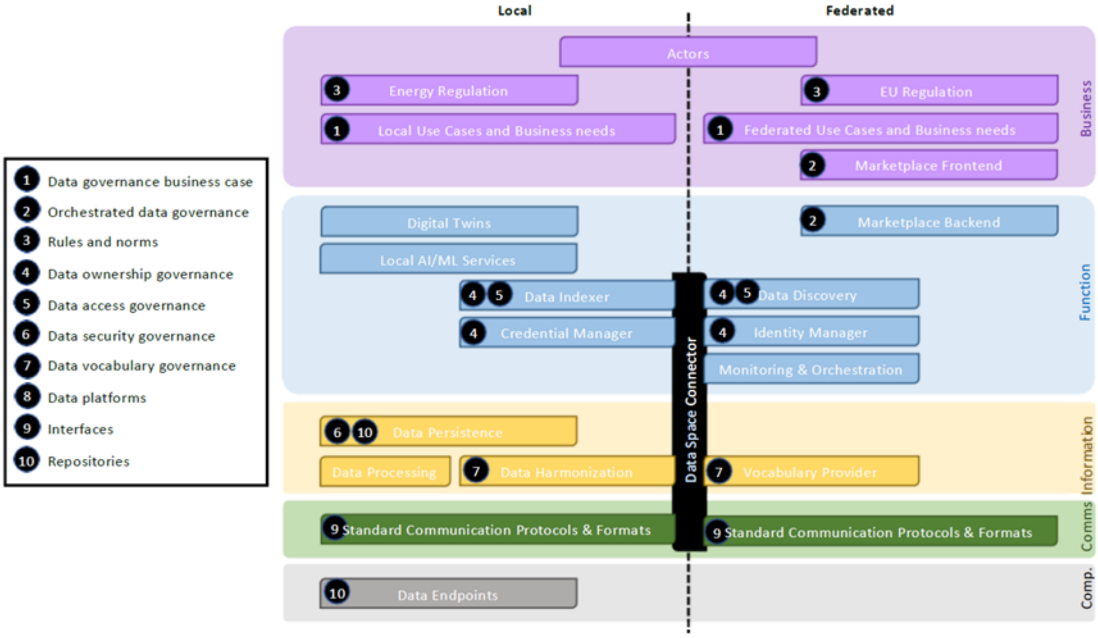

Figure 5: BRIDGE DERA 3.0 link to data governance, Source [10]

While the Local side of the architecture involves data platforms that are already deployed and using proprietary data (e.g. SCADA systems, Energy Community applications, etc.) to support specific applications, the Federated side encompasses the deployment of data spaces, where data and data services can be traded. Both layers are interconnected through the Data Space Connector, facilitating the data exchange and enabling identification, data harmonization and brokerage across various systems and applications, both federated and local [10].

The different modules of the architecture intend to deliver the following requirements, which are aligned with the aforementioned requirements of the EC for European Data Spaces. In Table 1 the components are explained, indicating how the requirements are delivered. Since in the STREAM project both local and federated sides of the architecture are represented, all the components are included in this section.

Table 1: DERA 3.0 modules and requirements

Layer	Component	Requirements
Component Layer: Origin of the data being handled by the system.	**Data Endpoint:** Any data source that can be incorporated to the data exchange	**Non-personal data:** if personal data is to be shared, it has to be anonymized or aggregated, signing a form of consent.
		Security/Resilience: identify directly at the origin potential business or operation critical elements, being carefully handled and replicated just when strictly needed.
Communication Layer: enables the different components of the system to exchange data and coordinate their actions.	**Standard communication protocols and formats:** on the local side, requirements should not be too demanding but should integrate standard and open protocols and formats to ease the process of federation. On the Data Spaces side, using open and standard protocols and formats is critical.	**Security/Resilience:** the protocol selected should ensure the highest levels of cybersecurity to ensure data sovereignty and confidentiality.
		Open Source and Interoperability: The use of open standards make sure easy access and use of data.
Information Layer: this layer acts as a repository for information, allowing it to be accessed and used by different parts of the system.	**Data Harmonisation and Vocabulary provider:** on the local side, data harmonisation assure the sharing format is appropriate using a common vocabulary. On the Data Spaces side, it provides information about the language used for data, and checks that data provided is compliant with the vocabulary.	**Open Source:** it is important to rely on open-source modules for this to allow any stakeholder to connect and understand the language and performing correct data exchanges.
		Interoperability: there should be a consensus on the syntactic format for data exchanges and queries to the Data Space, allowing easy connection.
	Data Processing: includes data quality and cleaning, data collection, data anonymization, personal data handling and metadata management.	**Non-personal data:** the module oversees providing the needed anonymization.
	Data Persistence: this module on the local side makes sure that data is stored and compliant with the vocabulary and syntactic rules.	**Security/Resilience:** the module must ensure a secure, trusted and sovereign storing by deploying cybersecurity mechanisms to protect data access and usage.
		Sovereignty: sovereignty should be guaranteed.
		Interoperability: interoperability needs to be specified in the definition of how data is accessed.

Function Layer: contains the components responsible for carrying out the tasks and operations needed to achieve the desired outcomes (e.g. analysis, decision-making, etc.).	**Credential Manager and Identity Manager:** on the local side, credential manager allows the identification of the user for data indexing. On the Data Spaces level, identity manager checks identities of federated nodes to start data indexing, discovery and transaction.	**Security/Resilience:** ensured by the identity provision and management. **Open Source:** the way to implement identification should be kept as simple and as open as possible. **Interoperability:** important to make sure the identification mechanism proposed is aligned at EU level, maximizing interoperability.
	Data Indexer and Data Discovery: on the local level, data indexer refers to the ability of Local data platforms to push relevant data into the Data Space, making sure it is in the correct format. On the Data Spaces level, data discovery gathers and processes data received through the local platforms.	**Sovereignty:** they should be indexed preventing its misuse or leaking. **Open Source:** the technology used on both modules should be open enough to allow stakeholders to federate in the Data Space with minimum technical barriers. **Interoperability:** the usage of common ontologies and data models to capture and index the data is key to allow understandability and replicability.
	Monitoring and orchestration: makes sure the federated nodes connected and using the system are performing well. The module should provide transparency to the Data Space.	**Security/Resilience:** cybersecurity is crucial to this module for guaranteeing privacy and sovereignty in the system. **Open Source and Interoperability:** the module relies heavily in the openness of the technologies used not directly in this module but also regarding underlying ingestion, communication and formatting modules.
	Marketplace backend: through digital marketplaces, the data and services made available in the Data Space should be prompted to the users. The module includes functionalities such as **monetization and payment, contracting and clearing.**	**Security/Resilience:** the marketplace should implement the highest standards of security, since it is the entry gat of the data. **Sovereignty:** data interactions and transactions occur within this module, so it is crucial to ensure that these exchanges happen under the data owners' premises and conditions. **Open Source and Interoperability:** using open-source technologies and APIs in the marketplace's development will ease the interaction with all stakeholders.

	Digital Twins and Local AI/ML services: placeholder for potential Local services, either local/pre-existing, or additional ones purchased from the App Store of the Marketplace and deployed locally.	No specific requirements.
Business Layer: this layer enables different business units and systems to interoperate and exchange data in a consistent and standardised manner, supporting the flow of information across the enterprise and facilitating interoperability between different business processes.	**Marketplace frontend:** this is the part related to the marketplace interface for the users.	**Security/Resilience:** like the backend part, adding considerations regarding the way actors are accessing the interface.
		User Acceptance: the frontend has to be simple and user-friendly to maximize its acceptance and user engagement.
	Local/Federated Use cases (UCs) and business needs: traditional local UCs (data collection and acquisition, storing, processing, etc.) and introduction of new federated UCs, combining data from sources and using services provided by third parties to increase the value of data.	Not directly linked to any requirements.
	Energy/EU regulation: relevant for the final operation of both Local and Federated sides, this module supports the compliance with energy and EU data usage regulation.	Not directly linked to any requirements.
	Actors: stakeholders and other entities extracting the information and triggering data exchange.	Not directly linked to any requirements.
	Marketplace frontend: this is the part related to the marketplace interface for the users.	

2.2.1.3 IDSA

The International Data Spaces Association (IDSA) is a non-profit organization that aims to create the required standards to ensure the interoperability of data spaces, keeping the focus on data sovereignty. Data sovereignty has been already mentioned several times in previous sections, and according to IDSA it refers to the capability of a person or an entity of being self-determined about its data [11]. The main proposal of IDSA is the Reference Architecture Model (IDS-RAM), which comprises the standards for secure and sovereign data exchanges. The general structure of RAM is depicted in Figure 6.

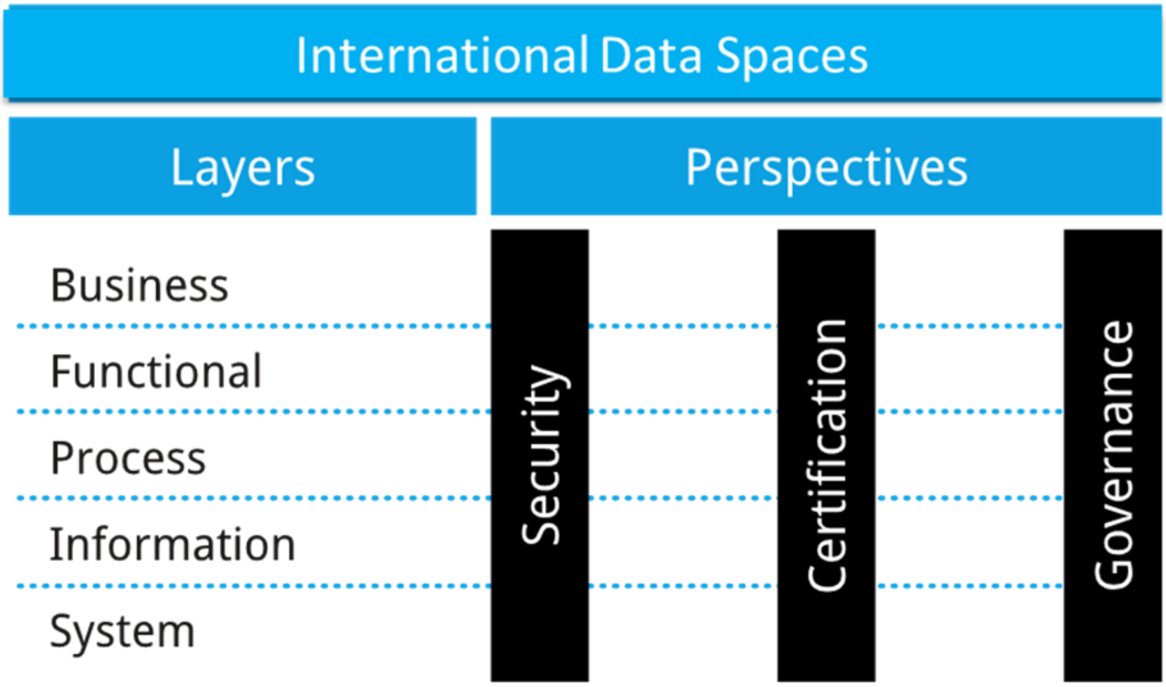

Figure 6: General RAM structure, Source [11]

From this structure, it is important to highlight and delve into the Functional Layer, under which the IDSA's functional requirements for data spaces are defined. These requirements are grouped in six groups of software functionalities as can be seen in Figure 7, and are summarized below:

Figure 7: Software functionalities and functional requirements, Source [11]

- **Trust:** Data spaces integrate numerous stakeholders into a common data environment. Thus, the participants must be identified and certified before being granted access to the trusted business ecosystem.
- **Security and data sovereignty:** state-of-art security measures must be integrated in the data space to ensure security, and all the technical components must be evaluated and certified. Each participant must have a certificate, enabling any participant in the data space to verify the capabilities of the other participants. Data sovereignty, as previously mentioned, is a

fundamental principle for IDSA, so data owners must be able to restrict the information to their data, and data consumers must accept the data usage policies set by the provider.

- **Ecosystem of data:** IDSA proposes decentralized data storage in its architecture. Therefore, data must be stored locally by the data owner until it is transferred to the data consumer. These data must be provided with metadata describing the syntax and the semantics. These metadata must be accessible by any other participant to understand the data of the others. Data providers can use vocabularies to define the semantics of their data, and a Vocabulary Hub is made available for easy understanding.
- **Standardized interoperability:** IDSA has designed a Dataspace protocol with the aim of defining a set of specifications meant to facilitate the interoperability of data exchange between the different stakeholders and participants of a data space [12]**.** IDSA puts at the centre of the communication protocol (see Figure 8) the International Data Spaces Connector. This connector is a software installed by the participating company or a platform, thereby providing technical access to the IDS ecosystem, connecting many data endpoints to increase the pool of available data and enabling secure and effective communication [13]. The connector must be able to run in the IT environment of any participant to ensure interoperability. The users of the connectors must be identifiable and manageable.
- **Value adding apps:** Data apps available in the Data Space should be able to provide an added value to the data, transforming input data into a desired output to comply with the requirements of Data Consumers. Data Apps must define their interfaces, dependencies and access requirements, and should be made available through the App Store.
- **Data markets:** Data owners can define the pricing model and the price of their data. The data market must ensure simple and standardized clearing and billing procedures. Standard contracts for these transactions are necessary.

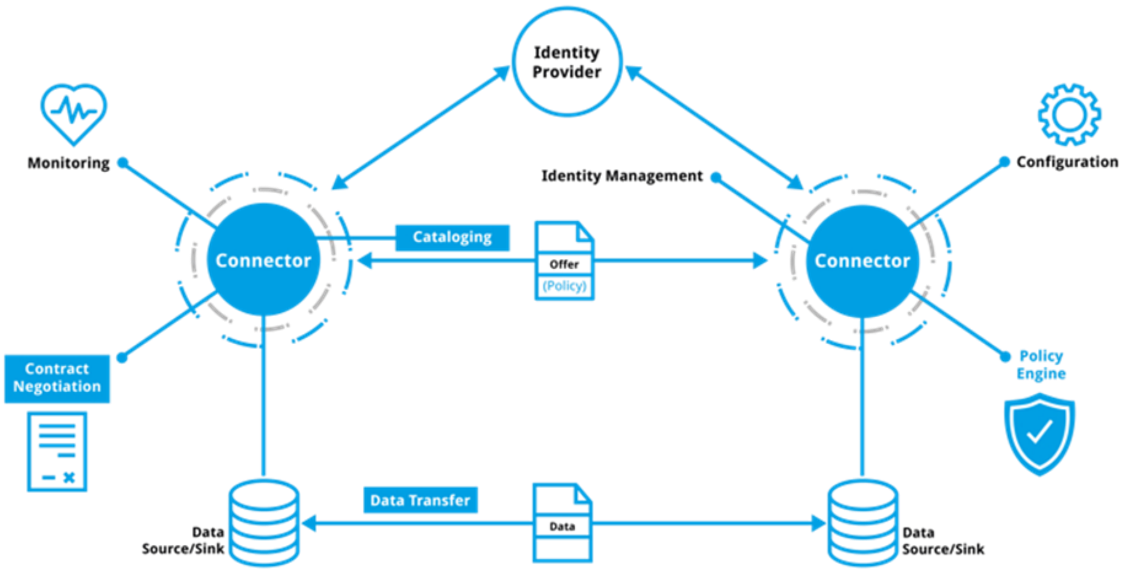

Figure 8: IDSA protocol overview, Source [12]

2.2.1.4 GAIA-X

GAIA-X is an association with the objective of establishing an ecosystem for data sharing in a trustworthy environment in collaboration with other parties [14]. Similarly to the previous initiatives, this ecosystem intends to give the users the control over their data (data sovereignty), with a federated system that links the cloud services of data providers and consumers in a transparent environment. For such purposes, GAIA-X proposes a different architecture that can be seen in Figure 9.

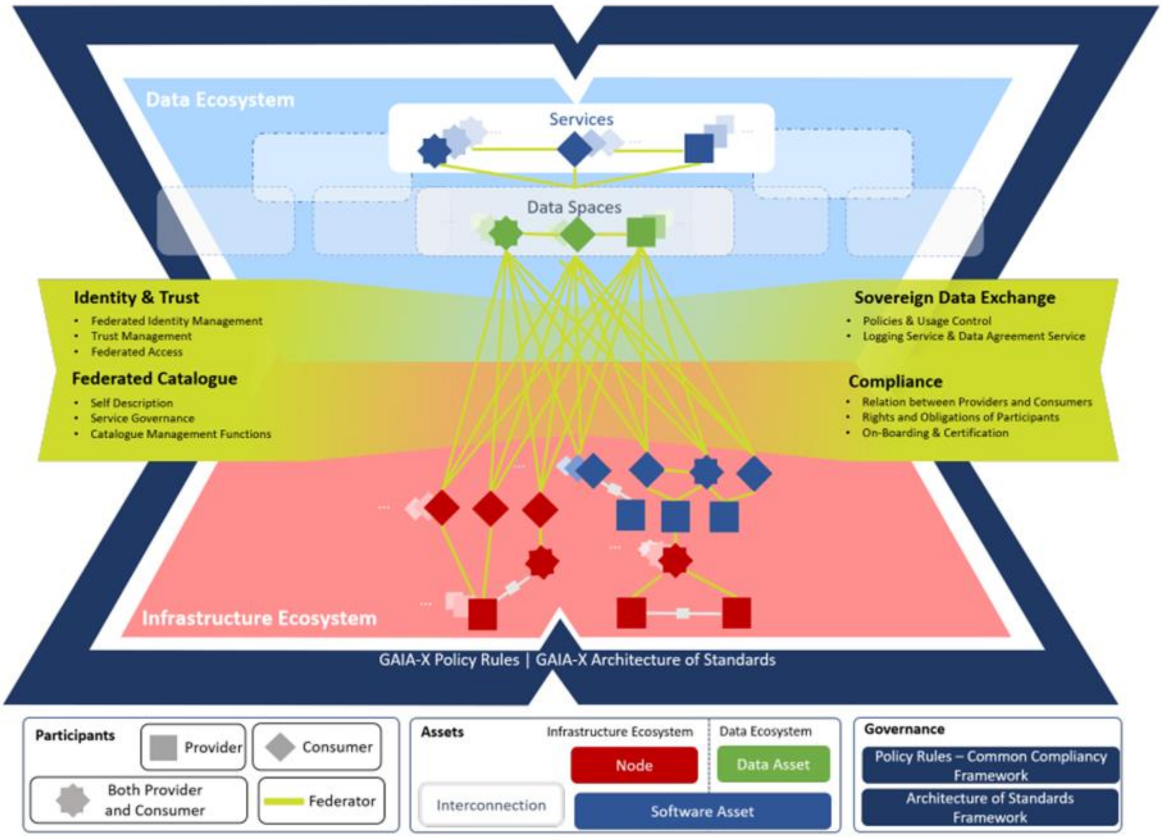

Figure 9: GAIA-X Architecture, Source [15]

As can be seen in the architecture, GAIA-X subdivides its ecosystem into two different ones, which are complementary: data ecosystem, which includes services and data spaces, where the data assets are made available in a decentralized manner; and infrastructure ecosystem, which computes, stores and interconnects elements. Both ecosystems are interconnected by federated services, which comprises both infrastructure services and organisational support functionalities. These federated services enable connections among different parties in the ecosystem, enabling the matching between data providers and consumers for data exchange. In this context, GAIA-X introduces a new actor, which is the federator. Federators are actors that directly or indirectly consume, produce or provide related resources, and are in charge of the Federation Services.

The architecture proposed by GAIA-X is meant to cover the requirements, presented in Table 2 [15]:

Table 2: GAIA-X requirements

Requirement	Definition
Data and services interoperability	Several systems and services must be able to exchange information to use it in mutually beneficial ways.

Portability of data and services	Standardized protocols must be used to enable transfer and processing of the data to increase its usefulness. Services can be migrated without significant changes.
Sovereignty over data	Participants must be able to retain absolute control and transparency over their data and its usage by other participants.
Security and trust	Security technology must be put at the core of the ecosystem to protect every participant and system. An Identity management is needed for authentication, selective disclosure and revocation of trust if needed.

2.2.1.5 OpenDEI

OpenDEI project was a part of the Horizon 2020 program. The project's target was to define the design principles for data spaces implementation in a unified way [16]. In its position paper [5], the project underlined the importance of data spaces in creating the future data economy. The most relevant findings of this document for sDATA's development are analysed in the following paragraphs.

OPENDEI understands data spaces a decentralised infrastructure for trustworthy data sharing and exchange. This data exchange is made under data ecosystems, which are based on commonly agreed principles between data providers and consumers. In the position paper, three building blocks are identified as essential in a data space:

- Data platforms, for effective data sharing and exchange.
- Data marketplaces, for data trading between providers who offer data and consumers who request data.
- Data sovereignty, for each stakeholder to control their data specifying terms and conditions.

From these building blocks, it can be inferred that several actors may be part of a data space:

- Data consumers: they access the data space to use data.
- Data providers: they make data available in the data space.
- Data producers: the ones creating the data.
- Data owners: the ones that can provide or deny access and use of data.
- Data application providers: developers of applications that ingest data that is available in the data space, processes the data and visualises it.
- Data platform providers: developers that provide capabilities that allow operation of data platforms.
- Data marketplace providers: developers that provide capabilities that allow operation of data marketplaces.
- Identity providers: provide capabilities for identifying parties.

OpenDEI provides in [5] a thorough and comprehensive analysis of the design principles that are required to meet the needs and concerns of the aforementioned stakeholders. These principles are:

- **Data sovereignty:** As has been already defined, data sovereignty refers to the capability of a person to keep control over their data. OpenDEI expands this definition adding the benefit that the data owner gains from enhanced possibilities to view, process, manage and secure their data.
- **Data level playing field:** New entrants must not face any barriers when seeking admission to a data space. Players compete on the quality of their data and services, not on the amount of data they control.

- **Decentralised soft infrastructure:** As European data spaces are understood, the infrastructure of a data space is made of the implementation of interoperable API-based IT platforms.
- **Public-private governance:** In European data spaces, both public and private interests should be represented. The public interests and values should be ensured by following the already established legislation and regulation (eIDAS, GDPR, PSD), as well as the specific upcoming regulation (DGA, DSA, DMA).

To meet these four principles, OpenDEI identifies seven architecture requirements. These requirements and their subdivisions are summarised in Table 3:

Table 3: OpenDEI architecture requirements

Requirement	Definition	Subdivision
Data-sharing empowerment	Ensuring that decisions can be made by appropriate stakeholders.	Governance in data spaces
		Citizen engagement support
		Data sovereignty support
		Federation
Data-sharing trustworthiness	Ensuring that data spaces operate according to expected requirements.	Security-by-design
		Privacy-by-design
		Assurance-by-design
Data-sharing publication	Enabling data to be published so it can be easily located by consumers.	N/A
Data-sharing economy	Creating the conditions for data sharing and exchange.	Non-financial incentive mechanisms
		Financial incentive mechanisms
		Agreement mechanisms
Data-sharing interoperability	Providing the ability for all applications in data spaces to create, use, transfer and exchange data.	Semantic interoperability
		Behavioural interoperability
		Policy interoperability
Data-space engineering flexibility	Providing the ability for engineers to add customised features in data-processing applications and data platforms.	Flexibility in terms of interoperability
		Flexibility in terms of trustworthiness
		Flexibility in terms of data processing

Data-space community	Fostering maximum reuse of data space solutions.	Open solutions and specifications
		Reusability
		Open source
		Sustainability solutions

Figure 10 relates these architecture requirements with the design principles defined by OpenDEI.

	Data spaces design principles			
Data space architecture requirements	Data sovereignty	Level palying field	Descentralised soft infraestructure	Public-private governance
Data-sharing empowerment	●	○	○	●
Data-sharing trustworthiness	●	●	●	●
Data-sharing publication	●	○	○	○
Data-sharing economy	●	●	●	○
Data-sharing interoperability	○	●	●	○
Data space engineering flexibility	○	●	●	○
Data space community	○	●	●	○

Figure 10: Relation between OpenDEI's data space design principles and architecture requirements, Source [5]

2.2.2 Conclusion and sDATA compliance with other initiatives and requirements

As a summary of all the initiatives described in section 2.2.1, data spaces are an IT infrastructure that enables the exchange of data between several stakeholders, both in a centralized and decentralized manner. The main requirements of this technology, which appear in all the analysed initiatives, are:

- **Interoperability and standardization,** in order to ensure easy integration and access for already existing systems and interconnection of various data spaces.
- **Data security and trust,** fundamental to ensure secure and safe data exchange.
- **Data sovereignty,** which is a cornerstone of data spaces. Data providers must have full control over their data, and data consumers must stick to the policies and conditions established by the providers.
- **Use of non-personal data and anonymization of personal data,** required to be compliant with GDPR.

The architecture and specifications of data spaces developed in EU projects, particularly in STREAM, should adhere to these requirements in order to be compliant with the current legislation regarding data exchange.

The sDATA solutions comply with the principles and requirements of the Open Data Spaces, as detailed in sections 3.1.3.5 (Finnish sDATA architecture) and 3.3.3.3 (IT and SI sDATA architecture).

2.3 IEC STANDARDS FOR INFORMATION EXCHANGE

IEC standards are internationally recognized guidelines that ensure the quality, safety, and efficiency of electrical and electronic systems worldwide [17].

Key standard series that address different aspects of electrical system management, used in STREAM, are IEC 61970 and IEC 61968. Both standards are part of the Common Information Model (CIM) framework, which ensures interoperability and continuous development through collaboration among various working groups [18].

- **IEC 61970**: The core packages of CIM are defined in IEC 61970, which focuses on the needs of electricity transmission, including applications such as energy management systems, SCADA, planning and optimization. It also covers a definition of an XML format for network model exchanges using Resource Description Framework (RDF) [19].
- **IEC 61968**: The standard extends CIM to meet the needs of electrical distribution, addressing applications such as distribution management, outage management, metering, asset management, and more. The series begins with distribution management standard, which sets the foundation for Distribution Management Systems interfaces and the data model standard CIM, which standardizes the integration of smart meter measurement data, without defining specific communication protocols for data exchange. The standard is designed to facilitate the integration of various applications within a utility enterprise, enabling the collection of data from both legacy and new systems, each with its own unique interfaces and run-time environments [20], [21].

The figures below (Figure 11, Figure 12, Figure 13) provide an example of how a dristribution network can be modelled using CIM. Figure 11 illustrates a sample distribution network model, while Figure 12 represents the corresponding CIM presentation of network. Figure 13 offers a detailed CIM model of the substation part shown in Figure 12.

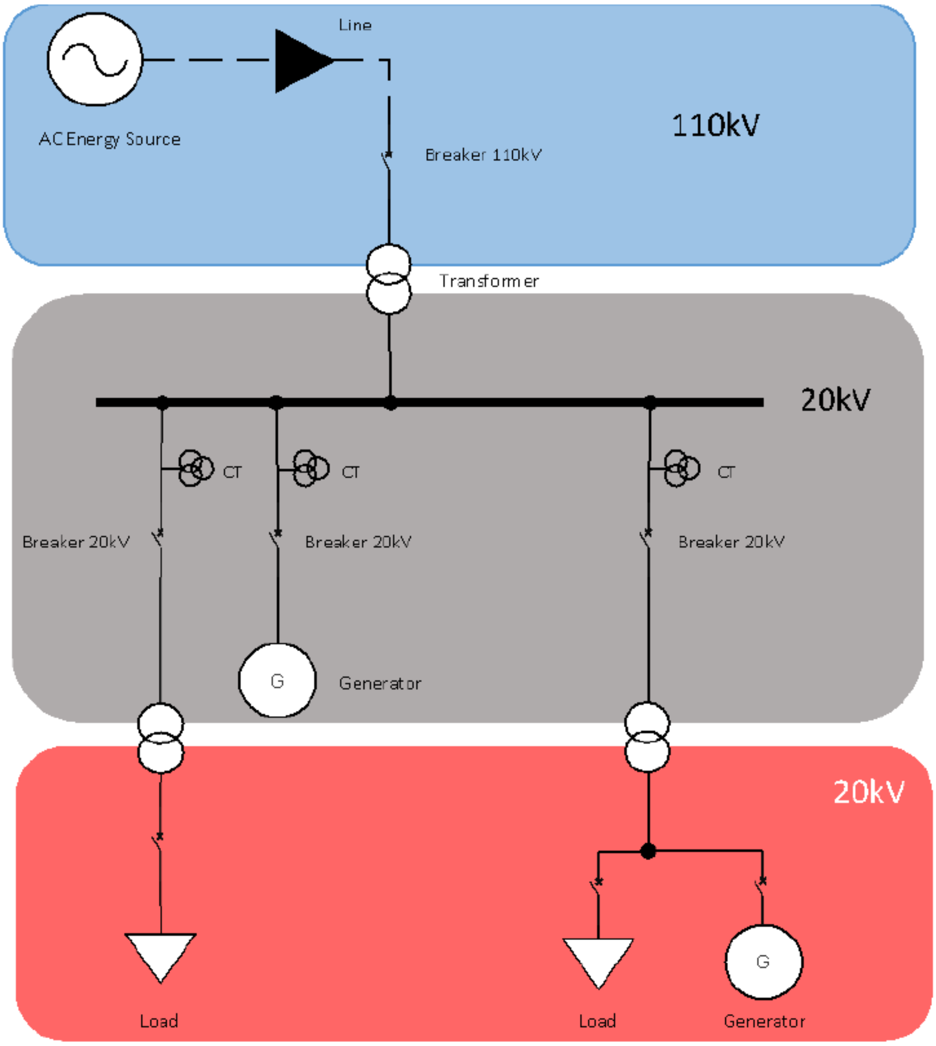

Figure 11: An example of a distribution network model

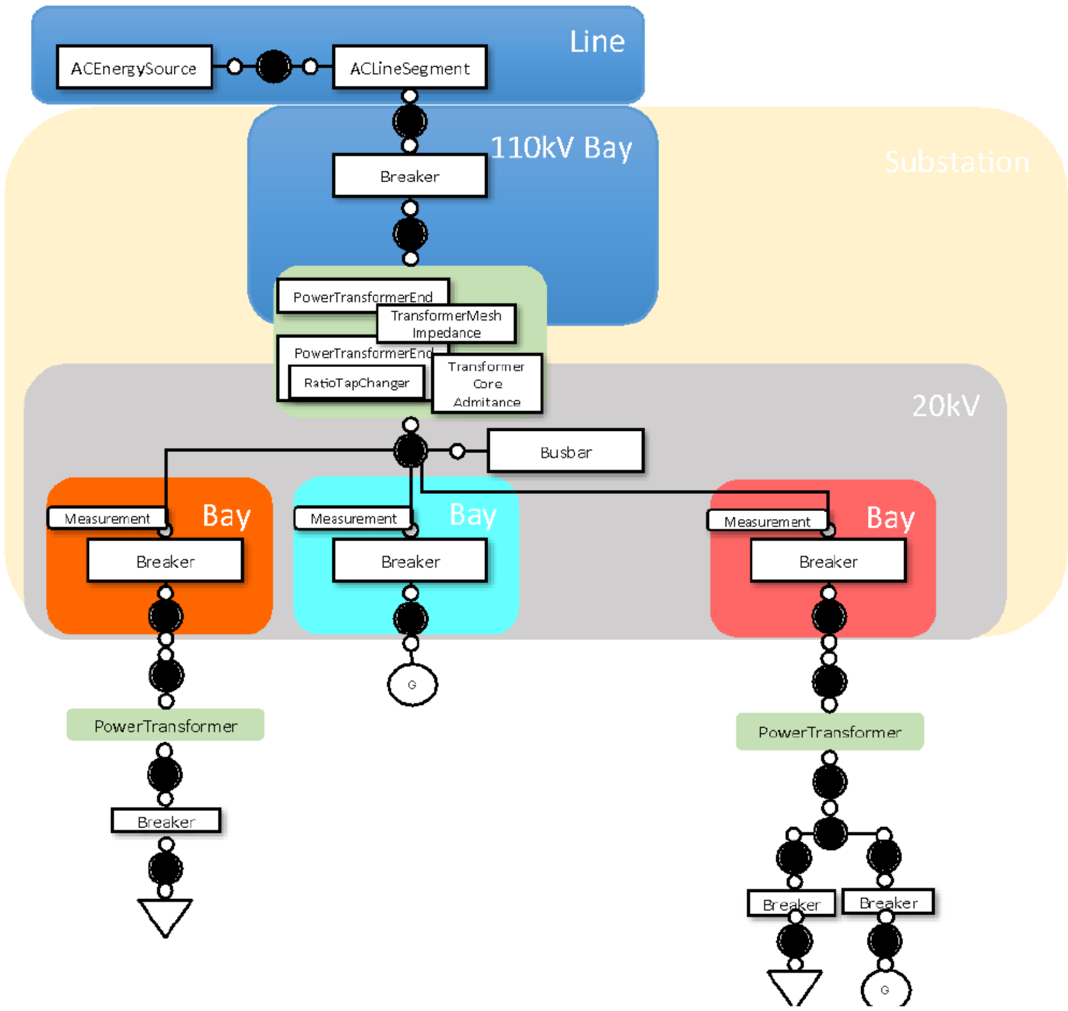

Figure 12: CIM presentation of the distribution network model example

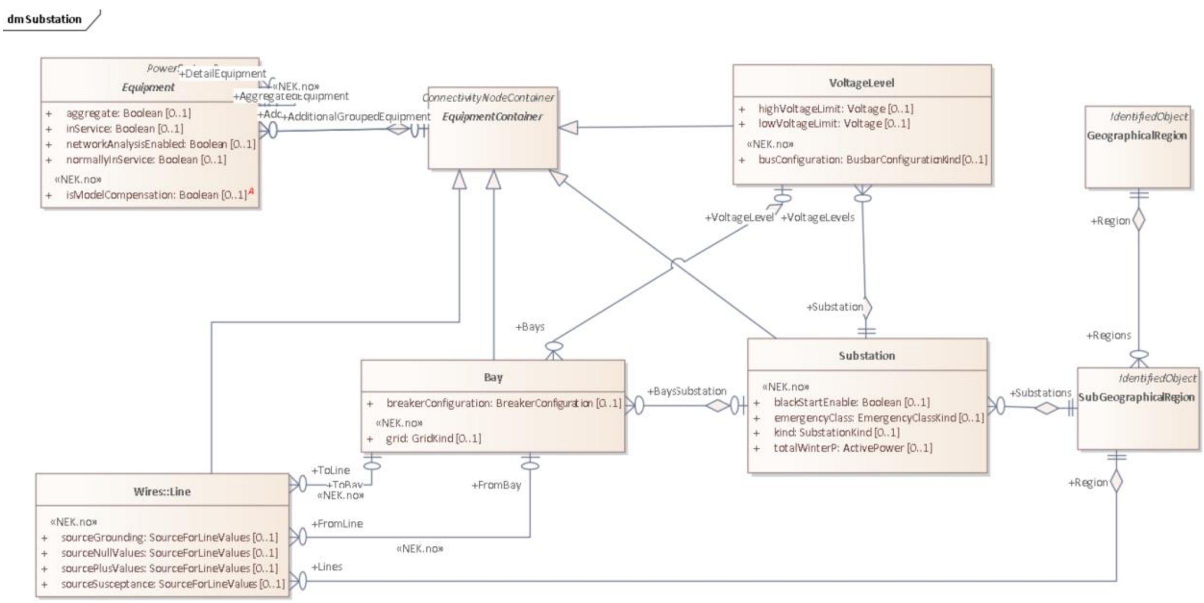

Figure 13: CIM model of the distribution network model example

2.4 USE CASES AND DATA ANALYSIS

This chapter presents a comprehensive analysis of UCs, data, and user needs, serving as the foundation for defining the sDATA requirements. As sDATA facilitates data exchange within the STREAM ecosystem, this analysis is essential for identifying the key functionalities and requirements the tool must address. The chapter is organized by pilot, with separate analyses of UCs, data, and user needs conducted for each of the four STREAM pilot sites.

The UCs analysis section provides an overview of the UCs from WP2, which form the basis for defining the requirements for sDATA. The Finnish and Slovenian pilots have already incorporated sDATA into their UCs, while the Italian UCs from WP2 were redefined to include sDATA. The Spanish pilot's, contribution focuses on UCs where data exchange is crucial. These sections include detailed descriptions and flowcharts of the UCs.

The Data and data users' analysis sections conduct an analysis of the pilots' and data users' needs. Both raw and processed data are examined for sharing with various stakeholders, including DSOs, consumers, and energy and non-energy service providers, each of whom has different requirements for data representation. These sections also address data protection needs and explore the challenges associated with data sharing.

2.4.1 Finnish pilot

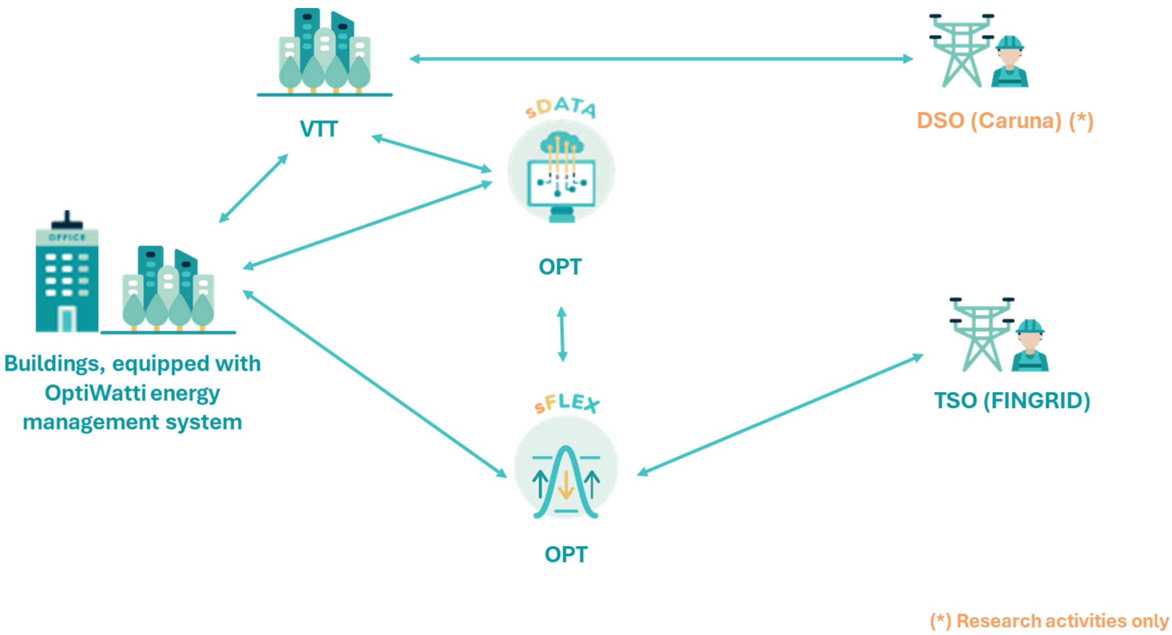

Figure 14: Finnish pilot set-up

The scope of the Finnish pilot is studying the possibility of using heating load as a source of flexibility in the demand response market.

Optiwatti builds the logic to run explicit flexibility services (sFLEX) and enhances the content of its databases to store the necessary information for the operation of such services (sDATA) as shown in Figure 14. Optiwatti also builds a simulated interface with Fingrid, the Finnish TSO. A selected group of existing Optiwatti customers, whose buildings already have Optiwatti energy management system, are equipped with additional meters, whenever needed.

VTT supports the pilot project by performing market status analysis and literature reviews. Also, VTT leads customer engagement activities and several research activities, including customer experience research.

The sDATA specification and implementation work follows the implementation of the sFLEX part and, since development is done by following Agile Methodology, the full detailed and final specification will be available step by step.

2.4.1.1 Use cases analysis

In the Finnish pilot, five UCs concerning data exchange and sDATA were defined in WP2, as shown in Table 5.

UC's **priority** has been originally organized as **classes** as described below.

Critical: The project would fail if the requirement were not met at this level.

Essential: The project should meet the requirement at this level, but the project could still meet its goals without it, possibly after renegotiation with the stakeholders.

Optional: The project wouldn't fail to meet its goals if the requirement weren't met at this level. The project will try to meet this level, but may elect not to, for example, because of time or resources constraints.

Nice: Meeting the requirement at this level would help it along, but in general this level should only be met if it can be done cheaply, for example, as a side-effect of meeting higher levels of other requirements. These requirements can contribute a great deal to customer satisfaction and goodwill. Also known as "extra credit".

Those priorities have been mapped to the STREAM priorities as shown in Table 4.

Table 4: Mapping between STREAM priority and Optiwatti priority

Optiwatti priority	STREAM priority
Critical	5
Essential	4
Optional	3
Nice	2
not used	1

Table 5 represents sDATA-related UCs in the FI pilot:

Table 5: List of UCs of the Finnish pilot

ID	Name	Type	Priority
FI.01	Technical proof of concept: using heating for flexibility	Technical	Critical (5)
FI.02	Maximizing flexibility potential	Technical	Essential (4)
FI.03	Defining the buildings potential on flexibility	Analysis	Optional (3)
FI.04	User experience when using flexibility	UX	Optional (3)
FI.05	Evaluating the profitability for the customer	Monetary	Critical (5)

All the defined UCs need sDATA. The details are listed in sections 2.4.1.2 and 2.4.1.3.

The UCs are presented in decreasing priority order. Figure 15 shows the main UC of the Finnish pilot project. This is also the UC that is going to be implemented first.

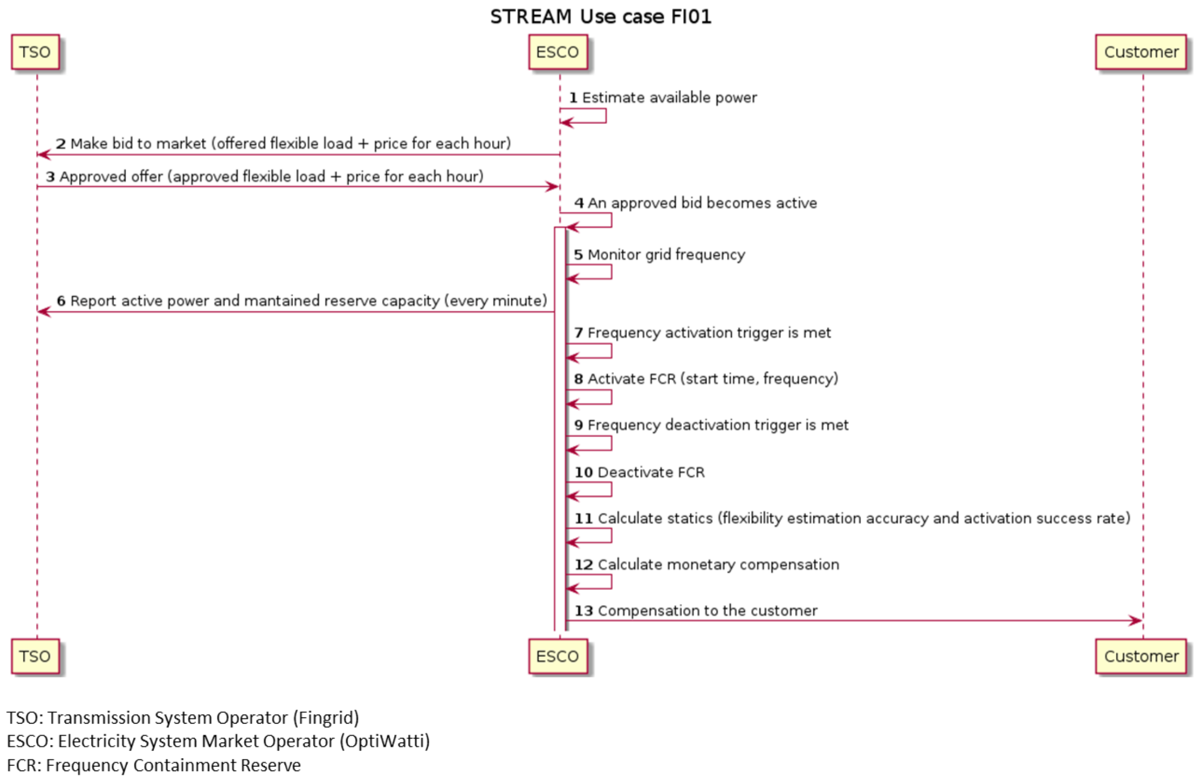

Figure 15: Main UC of the Finnish pilot: using heating for flexibility

The UC FI.01 can be in turn divided into 3 sub-UCs (which will be developed in this order):

- FCR-D-up
- FCR-D-down
- FCR-N

The UC consists of the following main steps:

- **Estimate** the combined available flexible load by taking into consideration implicit flexibility
- Make a **bid** to the TSO (flexible load, price)
- **Monitor** the frequency and report to the TSO
- **Activate** a FCR when the trigger is met
- **Deactivate** the FCR
- **Analyse** and calculate the monetary compensation to the customer

The following figures (Figure 16, Figure 17, Figure 18 and Figure 19) show the remaining UCs Those are the remaining UCs in decreasing order of priority.

UC FI.05 has the scope of evaluating the profitability for the customer as shown in Figure 16.

Customers might get compensated to be part of a FCR. Also, load cutting due to FCR activations involves less costs for the customer temporarily.

However, heating back after a load cut involves certain costs. Also, extra heating due to load activation due to FCR activation involves extra costs for the customer.

Spot prices contracts might also have an effect on those calculations since energy prices might be different each hour.

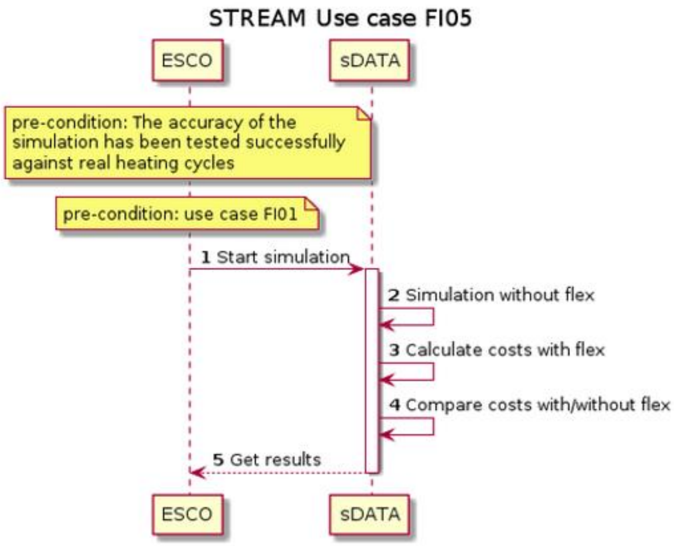

Figure 16: Evaluating the profitability for the customer

UC FI.02 aims to maximize the use of explicit flexibility while at the same time ensuring a satisfactory level of customers' satisfaction. This is achieved by using certain optimizations in deciding when to activate a FCR based on certain customer settings.

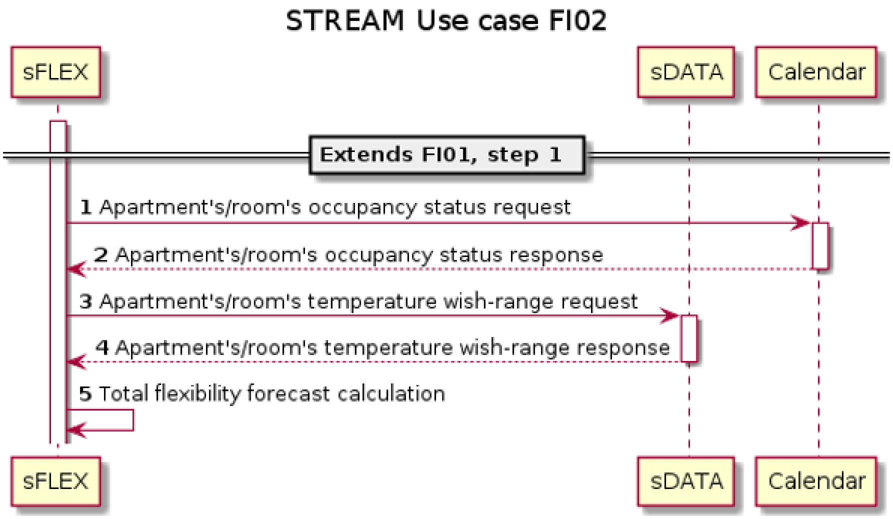

Figure 17: Maximizing flexibility potential and customer comfort

For commercial usage, when the number of customers being part of a FCR grows, it is necessary to have an automatic mechanism to get customer feedback. This is what UC FI.04 is about. Customer feedback is asked to customer whose buildings are being part of a FCR and to customers whose buildings are not part of a FCR in order to have a control group.

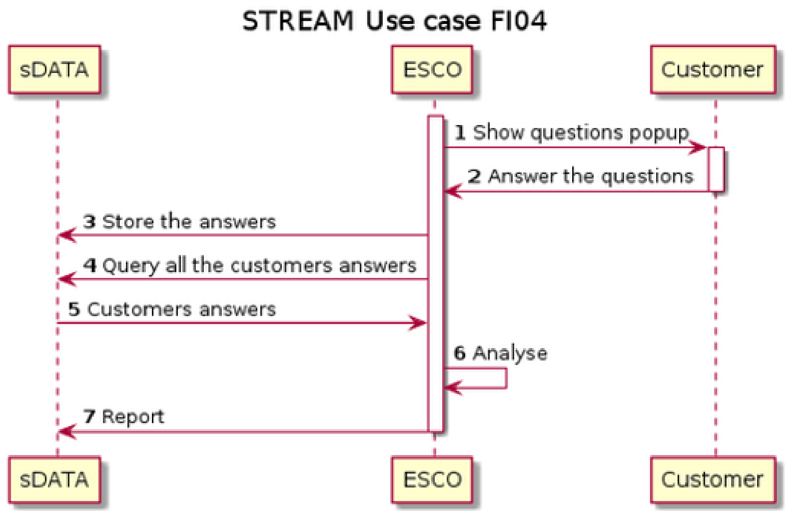

Figure 18: User experience feedback

For commercial usage of this functionality, it is also important to have an understanding of what building types and what heating types are most suitable for FCR services. UC FI.03 is about comparing the efficiency of direct heating versus indirect heating, fast non storing heating versus slow storing heating, etc. Also, it is studied how long the flexible load can be used in different kinds of environments and how long it takes for the room to regain the temperature drop caused by the flexibility service.

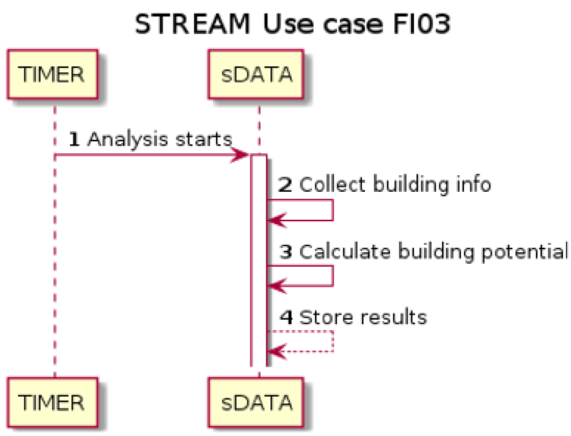

Figure 19: Studying building types and heating types potential

2.4.1.1.1 UC FI.01: Using heating for flexibility

Figure 20 shows the data scheme for the new variables that need to be stored in the system to implement the basic FCR-D upward, FCR-D-downward and FCR-N services.

It is a relational database scheme. The target is to store most control data such as information about flex client (controller), flexibility group, bid, etc. into a relational database (MySQL). Log data (“datalog” and “datalogrow” tables) are more efficiently stored into a time series database (PostgreSQL-based Timescale).

In addition, data about apartment, room, heating devices, contracts, etc… is (not shown in the figure below) fetched from the existing (MySQL) control database.

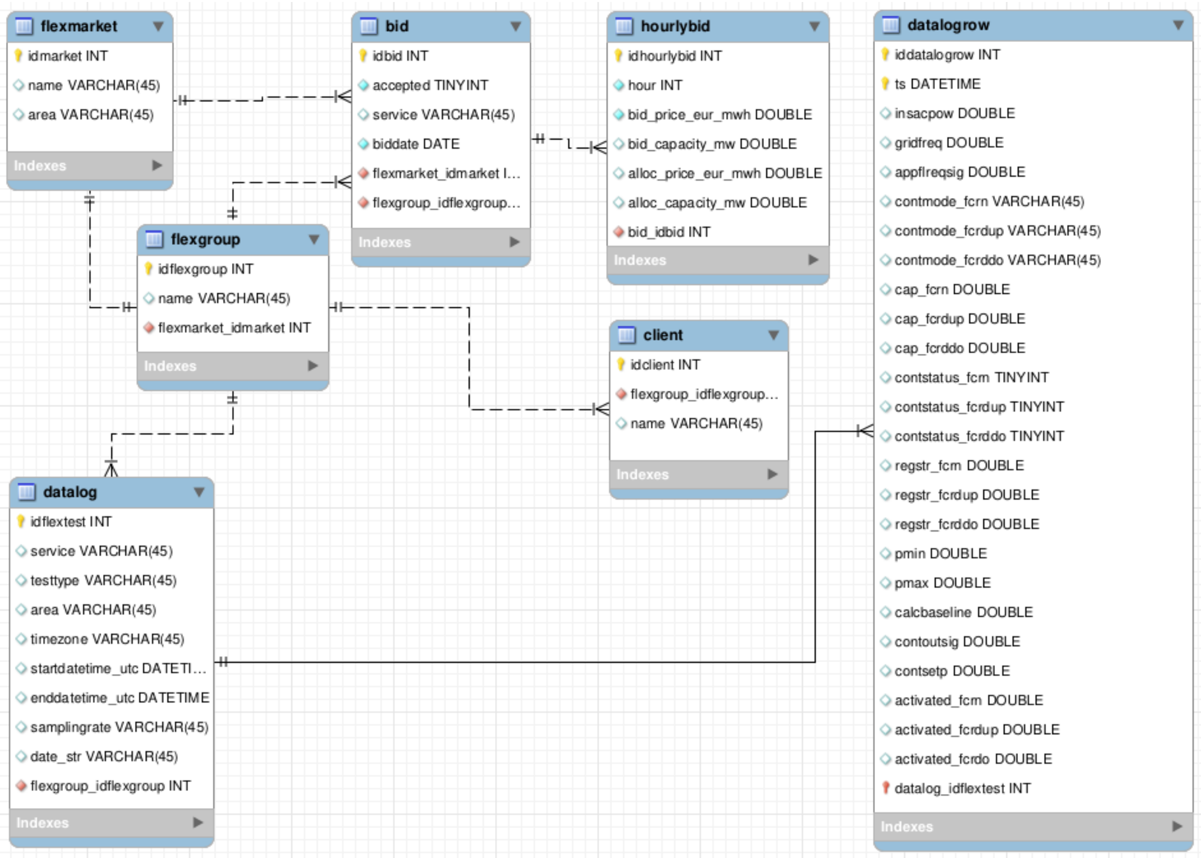

Figure 20: Data schema for the new needed variables (UC FI.01)

In addition to the above information elements, the following measurements are needed in order to calculate the KPI (FI01_01 activation success rate):

- Number of successful FCR-D-up activations
- Total number of FCR-D-up activation requests
- Number of successful FCR-D-down activations
- Total number of FCR-D-down activation requests
- Number of successful FCR-N activations
- Total number of FCR-N activation requests

All those measurements are stored in the STREAM dedicated database.

The required data log is documented in ENTSO-E documentation [22].

The information about the needed data log that needs to be stored in order to be exchanged with Fingrid is summarized in the following tables. Table 6 shows the general data log variables and Table 7 shows the detailed content of the log. The variables in orange colour are also contained in Fingrid specifications.

Table 6: Logged data

Variable	Description
Resource	identifier for the resource
Service	The type of service (FCR-N. FCR-D-up, FCR-D-down)

TestType	For normal operation it has the value “Operation”.
Area	The bidding area (Finland)
Timezone	For example, UTC or EET or EEST
Interval	The time interval for which data is delivered in the format YYYYMMDDThhmm-YYYYMMDDThhmm
SamplingRate	Nominal time difference between samples given in seconds. If the time difference between samples is less than 1 second, it is specified in milliseconds.
Date	The day data is extracted in the format YYYYMMDD.

Table 7 Logged data: details

Variable	Description	Type
InsAcPow	Instantaneous active power injection (negative for absorbed power) [MW] – positive for load cut and negative for added load	Double
GridFreq	Measured grid frequency [Hz]	Double
ApplFreqSig	Applied frequency (only used during test) [Hz]	Double
ContMode_Fcrn	Control mode (parameter set) FCR-N [id] (recommended)	Alphanumeric identifier
ContMode_FcrdUp	Control mode (parameter set) FCR-D up [id] (recommended)	Alphanumeric identifier
ContMode_FcrdDo	Control mode (parameter set) FCR-D down [id] (recommended)	Alphanumeric identifier
Cap_Fcrn	Maintained capacity FCR-N [MW]	Double
Cap_FcrdUp	Maintained capacity FCR-D up [MW]	Double
Cap_FcrdDo	Maintained capacity FCR-D down [MW]	Double
ContStatus_Fcrn	Status FCR-N [on/off]	Binary
ContStatus_FcrdUp	Status FCR-D up [on/off]	Binary
ContStatus_FcrdDo	Status FCR-D down [on/off]	Binary
RegStr_Fcrn	Regulating strength FCR-N [MW/Hz]	Double
RegStr_FcrdUp	Regulating strength FCR-D up [MW/Hz]	Double

RegStr_FcrdDo	Regulating strength FCR-D down [MW/Hz]	Double
Pmin	Minimum power [MW] - the minimum power level that the reserve entity can operate at (for example, - 1MW)	Double
Pmax	Maximum power [MW] - the maximum power level (for example, 0 MW)	Double
CalcBaseline	Power baseline [MW] - the active power of the reserve entity minus the reserve activations (it demonstrates the consumption of the devices if no reserves were activated)	Double
ContOutSig	Controller output signal - the value that the controller is forwarding to the devices (recommended)	Double
ContSetP	Setpoint before FCR [% or MW] - the same as power baseline, but for some entities it can be reported as a static value as it is not changing during the tests or operation (recommended)	Double
Activated_Fcrn	Activated FCR-N [MW]	Double
Activated_FcrdUp	Activated FCR-D up [MW]	Double
Activated_FcrdDo	Activated FCR-D down [MW]	Double

The **maintained capacity** is the maintained flexible power. If all the bid capacity is available, then it is the same as the offered flexible capacity.

The **regulating strength** is the reserve capacity divided by the frequency range of the product in question, which for FCR-D is 0.4 Hz. This is a static value for the whole market time unit, unlike the maintained capacity that should be a dynamic value.

The **activated power** is the power that is actually used for FCR.

2.4.1.1.2 UC FI.05: Evaluating the profitability for the customer

The following variables are needed in order to implement UC FI.05, as part of the periodic measurements data:

- forecasted room temperature
- forecasted electric power usage for heating with no explicit flexibility
- forecasted electric power usage for heating with explicit flexibility
- customer compensation
- aggregator compensation

Those variables will be stored in the STREAM dedicated database in Timescale.

In addition, since UC FI.05 is an extension of UC FI.01, also all the variable used in UC FI.01 are needed here.

2.4.1.1.3 UC FI.02: Maximizing flexibility potential and customer comfort

UC FI.02 needs additional variables in the control database for each room:

- Demand-response settings (at room level)
  - Possible values: no flexibility, economy, comfort
    - In the minimum implementation the two values flexibility and no flexibility could be enough
  - For each of the last two options it shall be possible to set the values of the min and max temperature deviation
    - temp_delta_plus_comfort (default 1 °C)
    - temp_delta_minus_comfort (default 1 °C)
    - temp_delta_plus_economy (default 2 °C)
- temp_delta_minus_economy (default 2 °C)
- For the "economy" and the "comfort" option it shall be possible also to define the time range of the day when flexibility can be used (by default all the hours of the day can be used)

Currently, it is planned to add three profiles for the explicit flexibility: no flexibility, economy and comfort. For each of the last two profiles the user can define in Optiwatti user interface how much the set temperature for the room can be changed in case of overheating (temp_delta_plus_x) and in in case of decreasing the heating load (temp_delta_minus_x).

In addition, the following variables are needed in the STREAM-specific measurement database in order to calculate the KPI (FI02_01 additional gained flexibility):

- flexibility with variation at room level,
- flexibility with no variation at room level.

2.4.1.1.4 UC FI.04: User experience feedback

UC FI.04 requires storing customer answers.

In addition, the following variables are needed in order to calculate the KPI (FI04_01 answer rate percentage):

- count of answered questionaries
- count of sent questionaries

The new flex measurement database in Timescale will be used for this purpose.

2.4.1.1.5 UC FI.03: Studying building types and heating types potential

For storing the building and heating types specific information, a JSON file could be used.

This file will contain data needed to study the impact of building type and heating type on explicit flexibility.

Periodic measurement data migration

While for the first prototype the data described so far might be temporarily developed into a local database, such a temporary solution will not be enough for commercializing the feature.

Currently, over 20 million new periodic measurements are stored each day in the periodic measurement tables. Adding additional functionality requires storing even more data and this would put too much stress on the current MySQL relational database. Therefore, even the existing periodic measurements shall be migrated from the current MySQL relational database into a more efficient, faster and cost-effective database optimized for time series.

The new database is a Timescale cloud-based solution. A big part of the existing measurements has been migrated to the new database already.

2.4.1.2 Pilots' and data user needs analysis

Data privacy and data sharing

The data contained in the Finnish pilot database consists of customers' personal data and therefore it is protected by GDPR.

The Finnish pilot consists of two partners: Optiwatti and VTT. Data needs to be shared with VTT for research purposes. Only data related to the STREAM pilot apartments can be shared with VTT after data has been anonymized.

Data related to the operation of the FCR services is also shared with Fingrid, the Finnish TSO. This data is anonymized and does not contain data about individual apartments but rather only data about groups of apartments.

Data visualization

In addition to the existing plots in Optiwatti user interface, a new data visualization layer shall be added. Grafana has been selected for this visualization later.

Grafana is the market leader among data visualization tools for time series. It has a free and open-source version, an easy-to-use GUI and very handy built-in functions to handle the time range of data series. It supports a wide range of graphs in addition to time series and a wide range of data transformations.

2.4.1.3 Data analysis

Apart from data analysis in this subchapter, the data analysis is also handled in tasks 3.4.1 Data analytics and 3.4.2 User profiling in the scope of deliverable D3.3. Some data analysis is also provided section 2.4.1.1. Below is a short summary of data analysis for the purpose of the sDATA development.

Data analysis: User profiling

Sending any bid to the TSO, sFLEX needs to estimate the available power that can be used for explicit flexibility (by taking into consideration also implicit flexibility). This is done based on an estimation of the required electric power. The sFLEX Simulator performs a **baseline forecast** by leveraging historical data. The sFLEX Manager **forecasts the available total flexibility** for the Flex Group based on simulation results, weather forecast, spot prices, nominal powers of the devices, etc. (UC FI01 using heating for flexibility).

Data analysis: Heating costs calculations

Customer costs with and without using explicit flexibility are calculated and the results are compared (UC FI.05 evaluating the profitability for the customer).

Data analysis: Building types and heating types potential

Calculations and estimations are done in order to evaluate how suitable different building types and different heating types are for explicit flexibility (UC FI.03 defining the buildings potential).

Standardization

Data that shall be logged and exchanged with the TSO is standardized by ENTSO-E and Fingrid specifications. Additionally, there are certain Optiwatti added value features that require storage of additional information.

ETL pipelines

For the process of moving the existing periodic measurements from the MySQL relational database into the more efficient PostgreSQL-based Timescale database, an ETL (Extract-Transform-Load) pipeline is needed. Apache Ni-Fi has been selected for this task.

Apache Ni-Fi is a free and open-source tool with a highly configurable GUI which is based on flow-based programming. It has a wide range of connectors, good flow management capabilities (templates, processor groups) and good data tracking capabilities (data provenance module to track and monitor data through all the flow). It is fast and it supports clustering through embedded zookeeper. Supported security protocols include SSL, SSH and HTTPS.

Those are the required data transformation types:

- Partitioning the source table
- Filtering fields
- Renaming fields
- Adding fields
- Omitting original fields
- Changing field values
- Reordering fields

Figure 21 provides an example of data pipeline.

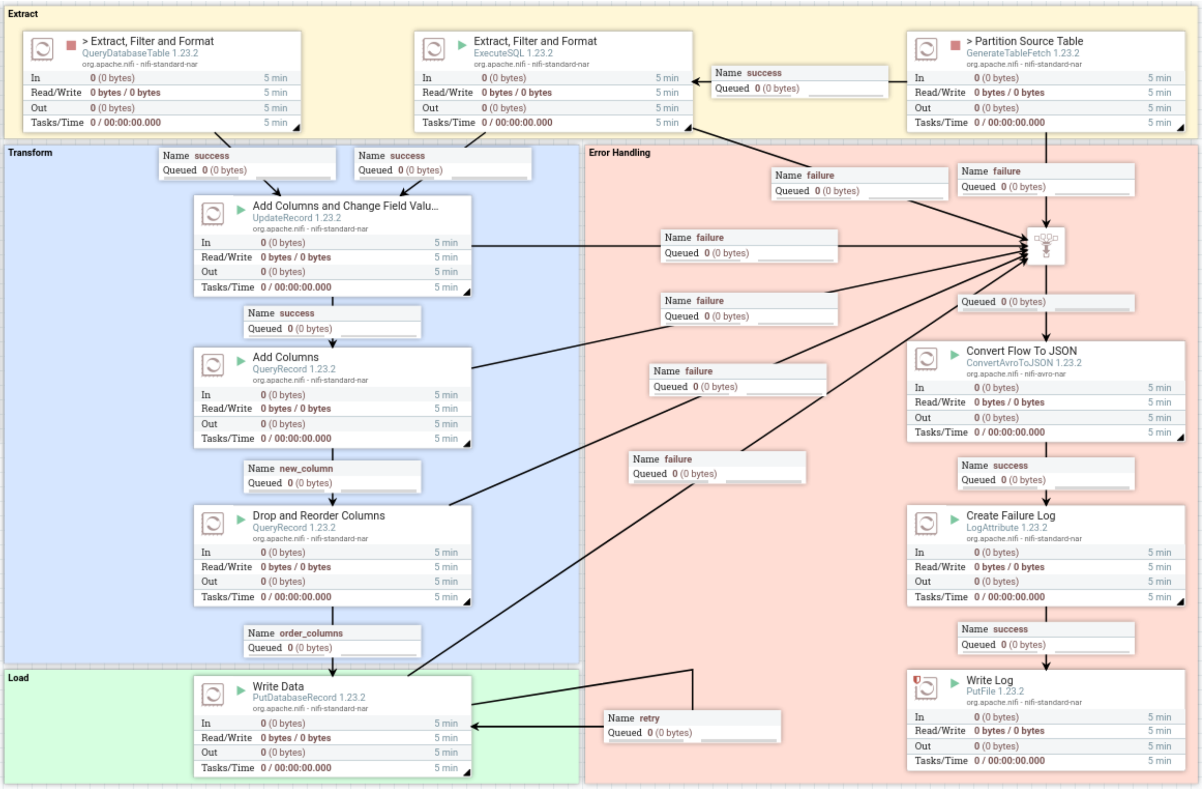

Figure 21: An example of Apache Ni-Fi data pipeline used for migrating the periodic measurements

The processor "Partition Source Table" allows using big source tables by retrieving data based on the configured Partition Size field. The rows in the source table are fetched 10 million at time. This processor also changes the name of certain columns and the format of the timestamp for compatibility with the target database.

The processor "Extract, Filter and Format" has the task of extracting data from the MySQL database using the "Connector to AWS MySQL OW DB" controller service. 100.000 rows are included in each flow file.

Data transformations are performed by the processors "Add Columns and Change Field Values" and "Add Columns". The first processor adds a column which identifies the source database and changes all the Null values of the columns that are used as a composite key in the destination database with 0 value. The second processor implements an algorithm that allows calculating the value of each metric based on the input coming from the source database.

The original and the created columns are selected and ordered in the processor "Drop and Reorder Columns".

The processor "Write Data" inserts the data into the Timescale database by using the controller service "Connector to alfa Managed Timescale" by using UPSERT.

A few other processors ("Convert Flow to JSON", "Create Failure Lod" and "Write Log") are used to create a failure log in case of errors.

The above-mentioned controller services contain the credentials to connect to the source and target databases.

Periodic measurement data processing

After the data has been extracted, filtered, transformed and loaded from the MySQL database into the new periodic measurements database, some further processing is needed before applying the scheduled aggregation, compression and retention policies.

<u>Verification</u>

First, the data is verified with python script in order to check if there are any Ni-Fi failure flows and if the amount of entries at destination is the expected one.

<u>Aggregation</u>

Due to the huge amount of data, data is processed in chunks in order not to require unreasonable amount of CPU capacity and disk space.

Data shall be first aggregated according to the defined aggregation rules. Data is aggregated by using hour, day and week intervals. The following statistics are calculated:

- Count
- Average
- Standard deviation
- Min
- Max
- Sum

Not all those aggregates are useful or make sense for all metric types. Figure 22 shows which aggregates are calculated for which metric. Marks in green show the most meaningful statistic for each metric type.

Metric	Count	Average	Standard Deviation	Median	Inter Quartile Range (IQR)	Min	Max	Sum
1 TEMPERATURE	√	√	√			√	√	
2 HUMIDITY	√	√	√			√	√	
6 SETTEMP	√	√	√			√	√	
7 LEAK	√						√	
16 CO2_PPM	√	√	√			√	√	
4 ONTIME	√	√						√
5 HEAT_ELEC_POWER_W	√	√	√			√	√	
19 HEAT_WATER_POWER_W	√	√	√			√	√	
161 GRID_V_L1 162 GRID_V_L2 163 GRID_V_L3	√	√	√			√	√	
171 PRODUC_V_L1 172 PRODUC_V_L2 173 PRODUC_V_L3	√	√	√			√	√	
121 GRID_P_L1_KWH / 131 GRID_M_L1_KWH 122 GRID_P_L2_KWH / 132 GRID_M_L2_KWH 123 GRID_P_L3_KWH / 133 GRID_M_L3_KWH						√	√	
141 PRODUC_P_L1_KWH / 151 PRODUC_M_L1_KWH 142 PRODUC_P_L2_KWH / 152 PRODUC_M_L2_KWH 143 PRODUC_P_L3_KWH / 153 PRODUC_M_L3_KWH						√	√	
181 GRID_KVARH_P_TOTAL 182 GRID_KVARH_M_TOTAL						√	√	

Figure 22: Metric-specific aggregates

When bulk loading data, aggregation is a very CPU-intensive and disk space consuming process.

Compression

Then, data is compressed by using python scripts in order to reduce the occupied disk space.

Purging

Further disk space savings is obtained by applying manually the defined retention policy by using python scripts.

Verification

A final verification phase is performed manually on a selected data subset by using Grafana visualization layer.

Data visualization

Grafana is used as a data visualization layer.

Dashboards that mirror the existing plots in Optiwatti user interface are needed. This allows seeing several plots in a single dashboard and changing easily the target apartment and room with no need for new authentication (which would be required by using the legacy Optiwatti user interface). This is also useful in the testing phase since it allows to test the new measurement API which reads from the periodic measurements in Timescale by comparing the plots in Optiwatti user interface which use the old MySQL database to the plots in Grafana which use the new API to the new measurement database.

The following figures (Figure 23, Figure 24) contain examples of such Grafana dashboards:

Figure 23: An example of a prototype Grafana dashboard which is connected to the new periodic measurements database in Timescale I

Figure 24: An example of a prototype Grafana dashboard which is connected to the new periodic measurements database in Timescale II

2.4.2 Italian pilot

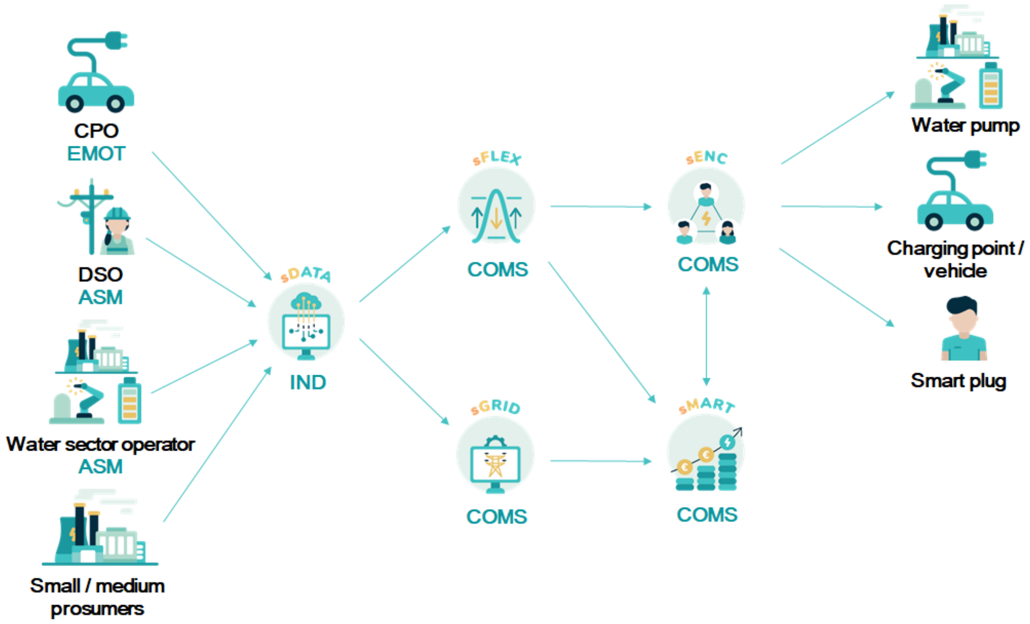

Figure 25: IT pilot schema

In the Italian pilot, sDATA will enable the seamless exchange of information between different partners and systems as shown in the Figure 25. EMOT, the electric mobility provider, will supply data from charging stations and electric vehicles (EVs) for analysis. In addition, sDATA will store historical consumption and generation data, and historical water pumps data, provided by ASM, which will be utilized in the sGRID and sFLEX tools for grid analysis and forecasting of baseline load and flexibility needs. The main objective is to create a forecasted time series that outlines the anticipated consumption and availability of flexibility within the grid. These insights will help determine the optimal timing and extent to which water pumps, EVs charging stations and energy communities can be used to prevent grid congestion and maximize the use of local renewable energy. sDATA will ensure efficient and timely data exchange, supporting both the analytical and operational requirements of the pilot.

In short, the integration of sDATA into this pilot is anticipated to significantly improve collaboration between the various stakeholders by providing a unified platform for data exchange. This seamless data exchange will enable a more comprehensive and accurate analysis of energy consumption patterns and the availability of renewable energy sources required by advanced analytics and forecasting tools to optimize the management of the grid.

2.4.2.1 Use cases analysis

In the Italian pilot, the UCs presented in this section are updated versions of those from WP2. The sDATA component was added at a later stage to relevant UCs:

- UC IT.01 - Grid model reconstruction for power quality and congestion assessment on MV/LV substation level
- UC IT.02 - Day-ahead congestion forecast,
- UC IT.04 Water-energy cross-sector model for flexibility potential assessment,
- UC IT.05 - Community-level flexibility potential assessment based on anonymized smart metering data.

2.4.2.1.1 UC IT.01 - Grid model reconstruction for power quality and congestion assessment on MV/LV substation level

Data flow will be streamlined from the DSO to sDATA and then to sGRID as shown in Figure 26. Consequently, the parts of the UC where sGRID requests historical data from the DSO will be updated to retrieve data from sDATA instead. The historical data will consist of loads and generation data, since this data is required for this UC. Moreover, some data cleansing will be performed within the sDATA tool to provide quality data for further data analysis by the sGRID component.

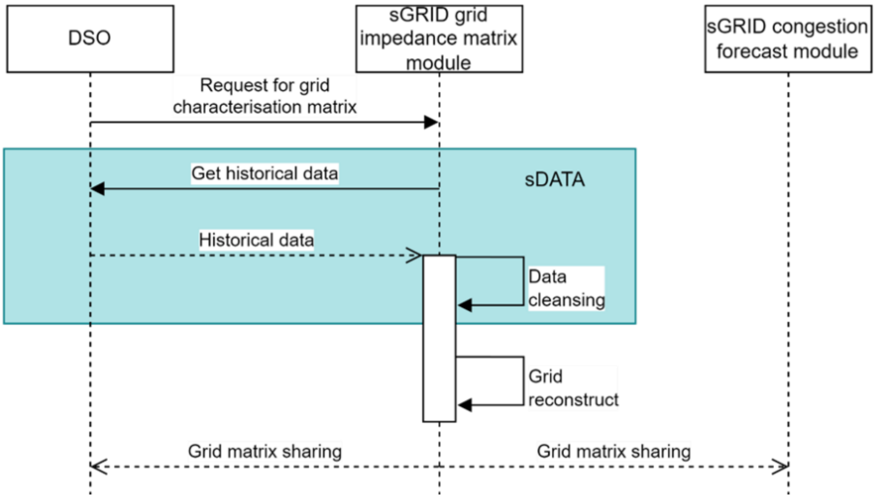

Figure 26: UC IT.01 sequence diagram

2.4.2.1.2 UC IT.02 - Day-ahead congestion forecast

In the updated UC, presented in Figure 27, sDATA will provide sGRID with historical data, replacing the direct communication between sGRID and the DSO for obtaining this information.

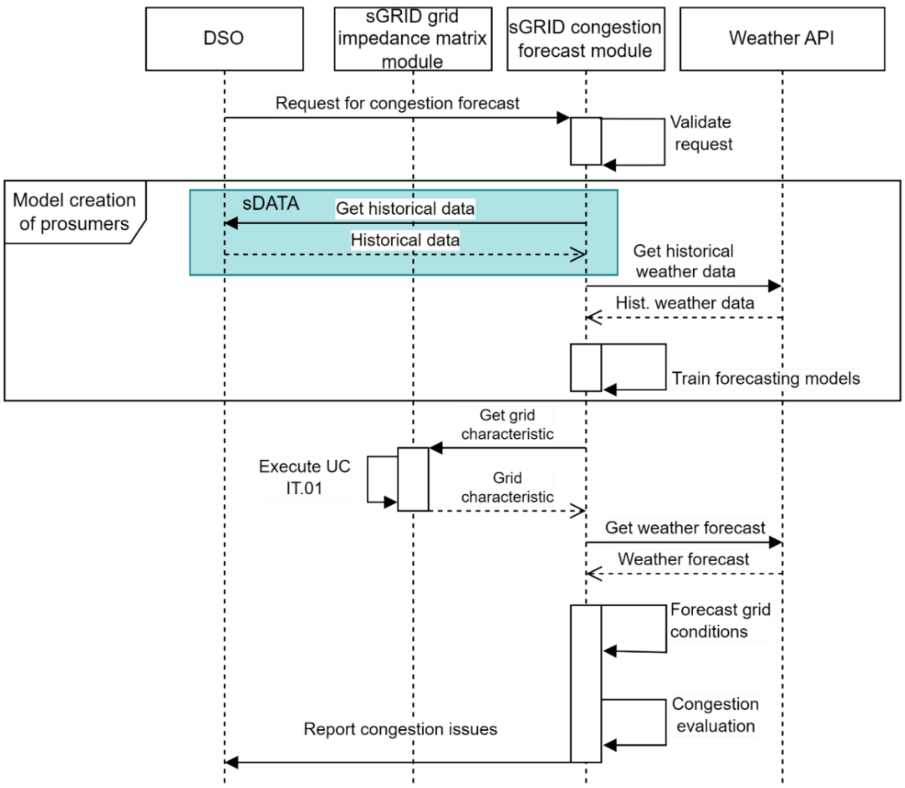

Figure 27: UC IT.02 sequence diagram

2.4.2.1.3 UC IT.04 Water-energy cross-sector model for flexibility potential assessment

Initially, a direct connection from ASM to sFLEX was anticipated in the UC. However, it is now planned that historical water pump data will be stored in sDATA and shared with sFLEX to develop a water-energy cross-sector model for assessing flexibility potential. The sDATA inclusion in this UC is shown in Figure 28.

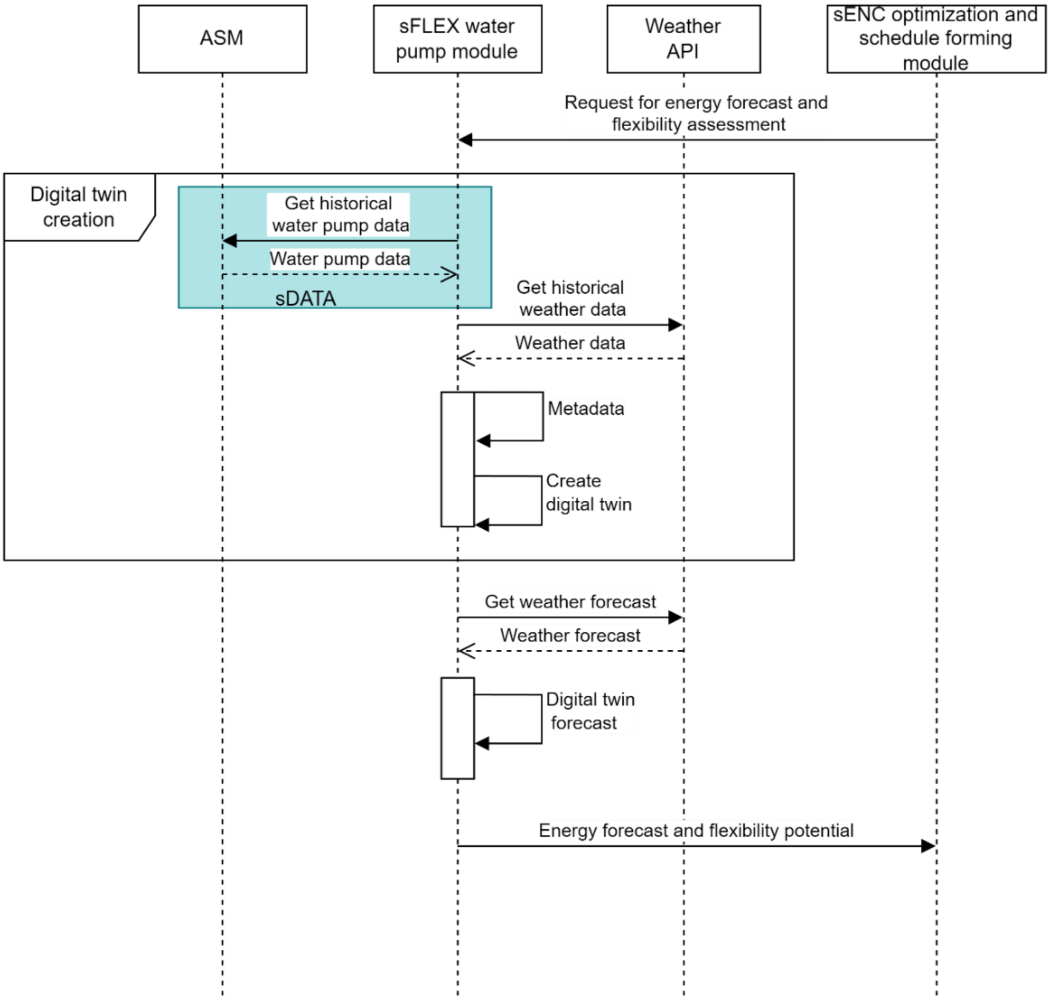

Figure 28: UC IT.04 sequence diagram

2.4.2.1.4 UC IT.05 - Community-level flexibility potential assessment based on anonymized smart metering data

Similarly, to UC IT.04 (see Figure 29), in this scenario, a direct connection from ASM to sFLEX was originally foreseen. Instead, historical user data is now planned to be retrieved through the sDATA for the Community-level flexibility potential assessment based on anonymized smart metering data in the sFLEX tool.

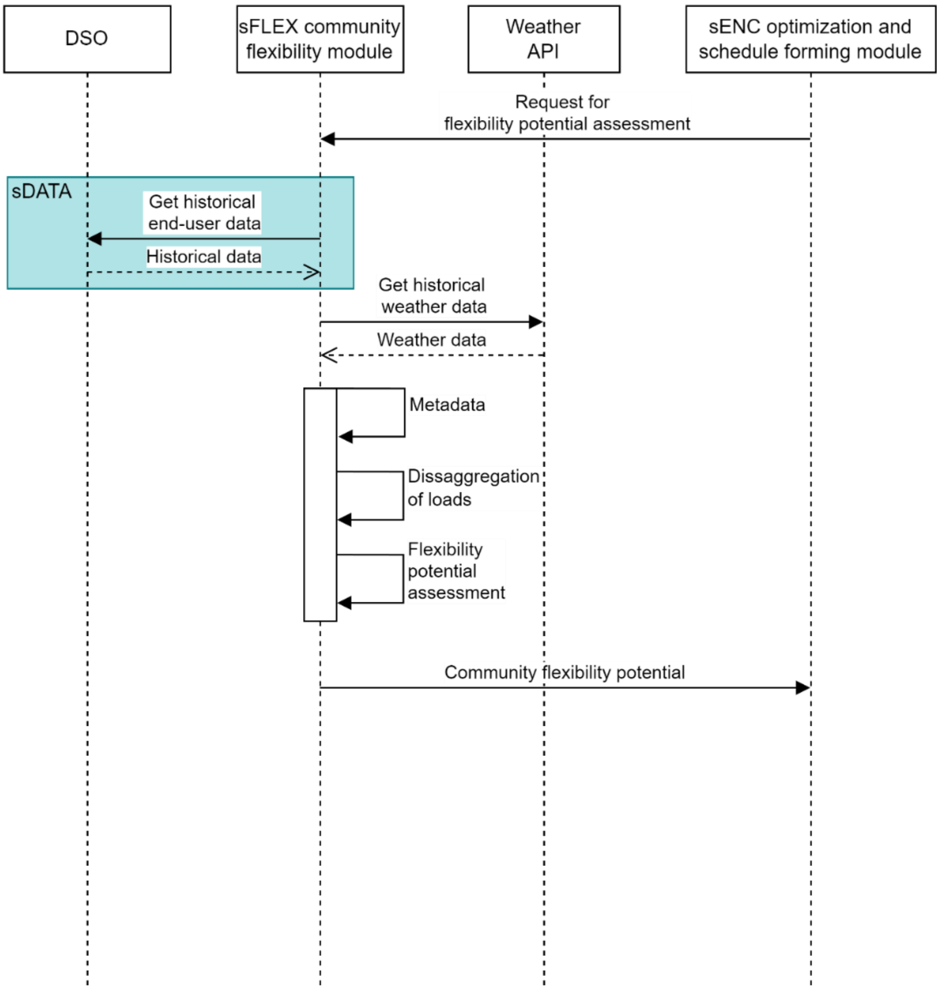

Figure 29: UC IT.05 sequence diagram

2.4.2.2 Pilots' and data user needs analysis

Centralizing data exchange in sDATA is crucial for the project's success, ensuring efficiency, consistency, integration, privacy, security, and quality. Streamlining the data flow from the DSO to sDATA and then to sGRID eliminates repeated direct requests from sGRID to the DSO, making the process more efficient and reducing waiting times. This centralization ensures that sGRID always uses up-to-date, quality-controlled data, improving the reliability of analyses and supporting accurate forecasting.

sDATA will provide sGRID with necessary historical data, replacing the need for direct communication between sGRID and the DSO. This ensures consistent and consolidated data, essential for system scalability and reducing the burden on the DSO. Additionally, storing historical water pump data in sDATA and sharing it with sFLEX facilitates the development of an integrated water-energy model, promoting interoperability between different sectors and tools.

Storing historical anonymized user data in sDATA before its use in sFLEX ensures compliance with privacy and data security regulations. This centralized approach allows sFLEX to easily access the necessary data for community-level flexibility assessments, improving efficiency and reducing access times. Overall, centralizing data in sDATA significantly enhances the pilot's effectiveness and reliability.

In this context, several users require data for their operations. DSO ensures the necessary data is transferred to sDATA and is available for downstream use by sGRID and sFLEX. They require accurate historical data to plan and manage grid infrastructure efficiently. Water Management Operators provide historical water pump data for use in sFLEX to develop the water-energy cross-sector model. Community-Level Users use anonymized smart metering data to assess the flexibility potential at the community level, enabling more effective local energy management. Grid Operators need historical load and generation data to reconstruct grid models, assess power quality, and forecast congestion. Analysts use historical data to perform detailed analyses for grid optimization and to support decision-making processes.

This structured data flow ensures that all relevant data is available where and when needed, supporting the various analytical and operational tasks required. Historical load and generation data include information on electricity consumption and production, crucial for grid modelling and congestion assessment.

ASM

ASM (DSO) collects this data from various points within the electrical grid, such as substations and meters as shown in Table 8. Water pumps data include historical operational data of water pumps, necessary for developing cross-sectoral water-energy models. The DSO collects this data from the water management systems it operates. Smart metering data are anonymized and consists of energy usage information collected from smart meters installed at consumer premises. The data is anonymized to protect user privacy before being used in community-level flexibility potential assessments. The sDATA platform stores all the collected data, performs data cleaning, and ensures that the data is consistent and ready for analytical use.

It will use Operational Data as real-time and historical data of the specific portion of grid, data on energy consumption at the household level, raw data detailed and unprocessed data sets for modelling and analysis. Visualization tools (like Grafana) could play a crucial role in helping users make sense of the data, monitor systems in real time, and support informed decision-making.

ASM provides an agreement with the partner in order to ensure security and privacy for the data shared, also to third parties. By addressing these safety and security concerns, and meeting the technical and operational requirements, the pilot can ensure that data is exchanged securely, efficiently, and in compliance with relevant regulations. This will help maintain the integrity, confidentiality, and availability of the data throughout the project.

By proactively addressing these challenges and implementing robust mitigation strategies, the pilot can achieve its objectives and demonstrate the viability and benefits of advanced data exchange and processing in the energy sector.

Table 8: Data provided by ASM

Data	Source of data	Data exchange format
Energy consumption	Smart meters	.json
PV data	Power quality analyser	.json, .csv
Water feeder	Power quality analyser	.json, .csv
Building energy management system	Platform	.json, .csv
Water pump data	Sensors of water pumping station	.csv

COMS

The main data required for COMS arises from its role as developer of several STREAM tools (sGRID, sFLEX and sENC), with the required data being provided by EMOT and ASM via the sDATA tool. The most important type of data required for the development and operation of the STREAM tools in the Italian pilot is the time series measurement data of various variables (consumption and production of electrical energy, water pressure, water level, weather data), while metadata also plays a role, albeit a lesser one.

As most of the data exchanged on the Italian pilot site is sensitive personal data, the information obtained on the pilot site must be treated confidentially. To ensure this, the data must be anonymized and encrypted, and credentials are required to transmit the data to the tools to ensure security. Only the owners of the data and authorized third parties will have access to the data. Since COMS is only a recipient of the data, the primary responsibility for ensuring data security and integrity lies with ASM, EMOT and sDATA. As data providers, ASM and EMOT must define the conditions under which the data can be accessed and used. COMS, on the other hand, as a data user, undertakes to strictly adhere to these conditions and ensure the confidentiality and security of the data received.

2.4.2.3 Data analysis

ASM

The stored and available data are related to the ASM database and are not openly available. Data can only be reused only by the partners involved in the STREAM project through single agreement partner by partner. Data are stored in servers, either on site of the pilots or in locations indicated by the technology provider. A partial list of standards used includes OpenADR 2.0; ASCII (American Standard Code for Information Interchange); MQTT; RESTful services. FTP/HTTP, MQTT, Rest API / MQTT, and MODBUS are relevant protocols.

Data stored includes:

- raw data from sensors/devices,
- processed data and analytics,
- historical energy consumption data.

Storage format collected and shared by the DSO are CSV and JSON or by direct access to the database.

The data from smart meters are:

- negative reactive power (Q-) [kvar],
- instantaneous voltage (U) in phase L3 [V],
- instantaneous voltage (U) in phase L2 [V],
- instantaneous voltage (U) in phase L1 [V],
- positive active power (A+) [kW],
- negative active power (A-) [kW],
- positive reactive power (Q+) [kvar].

Water pumping station data:

- water pressure [bar],
- reservoir levels [cm],
- water flow rate [l/s].

Along the water feeder, ASM has a various Power Quality Analyser that collect:

- phase currents, values instant/min/median/max on programmable interval,

- active, reactive, and apparent power, values instant/min/median/max on programmable interval,
- frequency values instant/min/median/max,
- temperature, values instant/median on programmable interval,
- relative humidity, values instant/median on programmable interval.

PV plants collect:

- instantaneous voltage (U) in phase L3 [V],
- instantaneous voltage (U) in phase L2 [V],
- instantaneous voltage (U) in phase L1 [V],
- apparent instantaneous power (S+) [kVA],
- positive reactive instantaneous power (Q+) [kvar],
- positive active instantaneous power (A+) [kW].

The challenges in data exchange span technical, operational, and governance aspects. Ensuring data interoperability, maintaining data quality, safeguarding security and privacy, achieving seamless technical integration, and managing governance issues are all critical hurdles that organizations must address to ensure effective data exchange. Overcoming these challenges requires a combination of advanced technologies, clear policies, and ongoing collaboration between stakeholders.

COMS

COMS is responsible for the development of the tools sGRID, sFLEX and sENC in the Italian pilot site. These tools require a large number of data sets in order to provide the data analytics necessary for the needs of the defined UCs. The required data, such as historical consumption and generation profiles of prosumers, state of charge of EVs, water level and pressure in water pumping stations, etc., are provided by other Italian partners, namely EMOT and ASM. As the consistency, coherence and notation of the data in a standard format is of utmost importance for the development of the tools, some pre-processing is foreseen in the sDATA tool before it is further used by COMS. To achieve this, the first step is to define and standardize the structure, the type of data, the scope of the data and the data exchange protocols in the pilot site to avoid data inconsistencies and communication problems.

EMOT

Data provided by EMOT are related to Italian pilot charging stations (Voltage, Current, Power, Energy, Plugs Status) and EVs (State-of-Charge, Speed, Odometer, Charge Status). Data is already standardized and is collected each second in JSON format using WebSocket protocol. Data is already available and is shared with technical partners via MQTT broker (for real-time data) or via RESTful API (for historical data).

2.4.3 Slovenian pilot

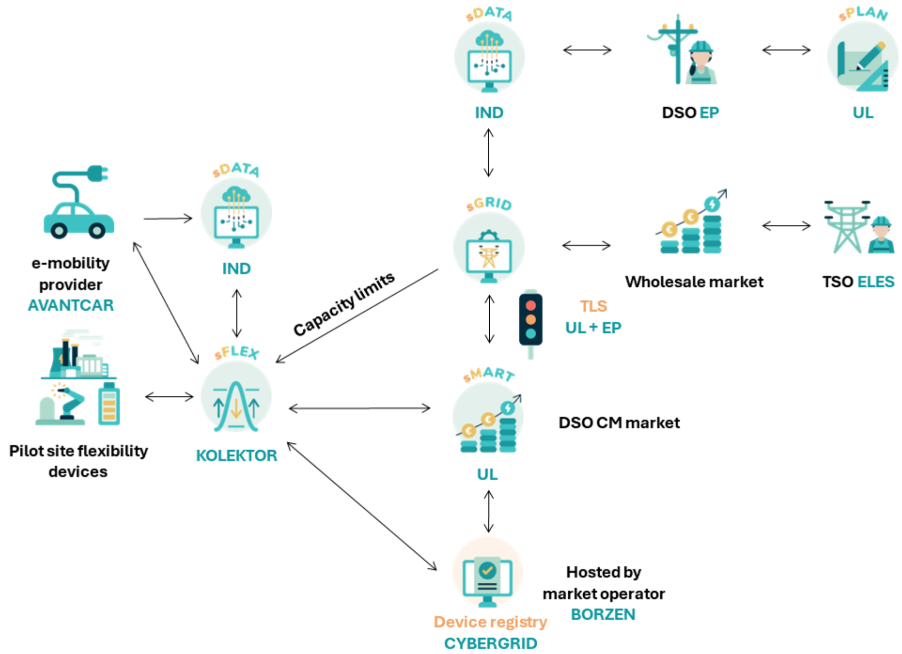

Figure 30: Slovenian pilot site scheme

In the SI pilot, sDATA facilitates data exchange among various users and systems as shown in Figure 30. sDATA is provided by the technology provider, Inden (IND). Avantcar (AVA), the e-mobility provider, will supply EV chargers' data for analytic purposes to Kolektor sETup (KOL) – aggregator. Additionally, a direct integration for edge computing will be implemented between flexibility assets and Kolektor sETup to enable rapid data exchange. This approach is essential for near real-time analysis, as routing data through the sDATA platform would introduce additional latency, making it unsuitable for time-sensitive operations.

sDATA will also store grid and time series data provided by Elektro Primorska (EPR) – DSO, which will be used in the sGRID tool for simulations and grid analysis. This setup ensures efficient and timely data sharing, supporting both analytic and operational needs within the pilot.

2.4.3.1 Use cases analysis

In the Slovenian pilot, four UCs concerning data exchange and sDATA were defined in WP2. These are:

- UC sDATA.01: Interoperable Data Exchange,
- UC sDATA.02: Distributed Ledger for Data Access Policies,
- UC sDATA.03: Metadata Management,
- UC SI.05: Grid assessment utilizing Traffic Light System (TLS).

The UCs together with data user needs and data analysis establish the foundation for the overall sDATA requirements. The three sDATA UCs (UC sDATA.01, UC sDATA.02, and UC sDATA.03) were broadly established, meaning they also apply to the IT pilot site, where sDATA is provided by the same technology provider, IND.

2.4.3.1.1 UC sDATA.01: Interoperable Data Exchange

The first UC sDATA.01 focuses on enabling interoperable data exchange within the sDATA platform, emphasizing robust authentication and authorization to ensure secure data usage and distribution among various partners.

Given the involvement of multiple data users, data interoperability is crucial. Standardized data exchange protocols and data models, such as the CIM, facilitate uniform communication across various platforms, allowing different systems to consistently interpret data. This standardization also simplifies the integration of new partners and enables the use of the open data exchange platform in other sectors, such as gas and water distribution, with minimal modifications.

Additionally, this UC highlights the need for diverse data storage solutions to accommodate various data structures. This versatile approach ensures that the system can adapt to the specific needs of each application, promoting a more efficient and scalable interoperable data exchange environment.

The UC's sequence diagram is shown in Figure 31.

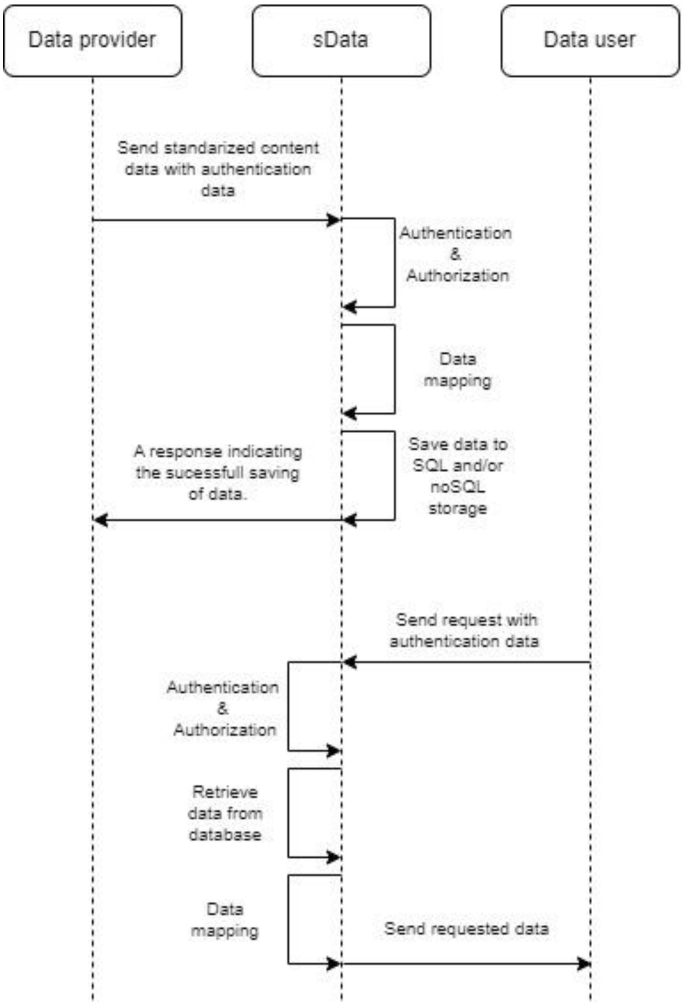

Figure 31: UC sDATA.01 sequence diagram

2.4.3.1.2 UC sDATA.02: Distributed Ledger for Data Access Policies

UC sDATA.02 addresses accountability, transparency, and security of data within the sDATA platform during data exchange, viewing, and data transformation. Distributed ledger technologies, such as blockchain, enable the recording of all access events, providing a transparent and immutable record. This allows users to easily access and review the history of data-related activities, ensuring full visibility into access changes and system operations.

The UC's sequence diagram is shown in Figure 32.

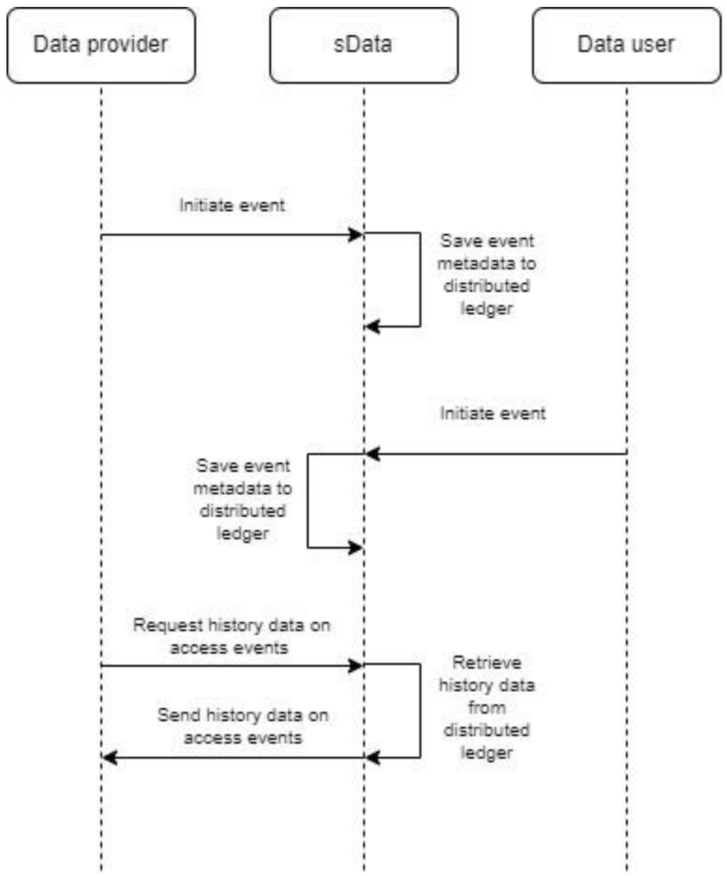

Figure 32: UC sDATA.02 sequence diagram

2.4.3.1.3 UC sDATA.03: Metadata Management

To enable smooth communication and data exchange, the sDATA platform requires an established user management system and a metadata management system. The user management system should handle user accounts and access rights, ensuring secure and controlled access to sDATA. The metadata management system will store metadata across various categories, such as time series, measurement points, grid models, and code lists, providing users with all relevant information about the data they use.

This approach includes managing the metadata lifecycle, addressing key aspects like retention policies and data stewardship. By promoting a systematic approach to metadata, this UC enhances data quality, compliance, and overall data management within the project ecosystem.

The UC's sequence diagram is shown in Figure 33.

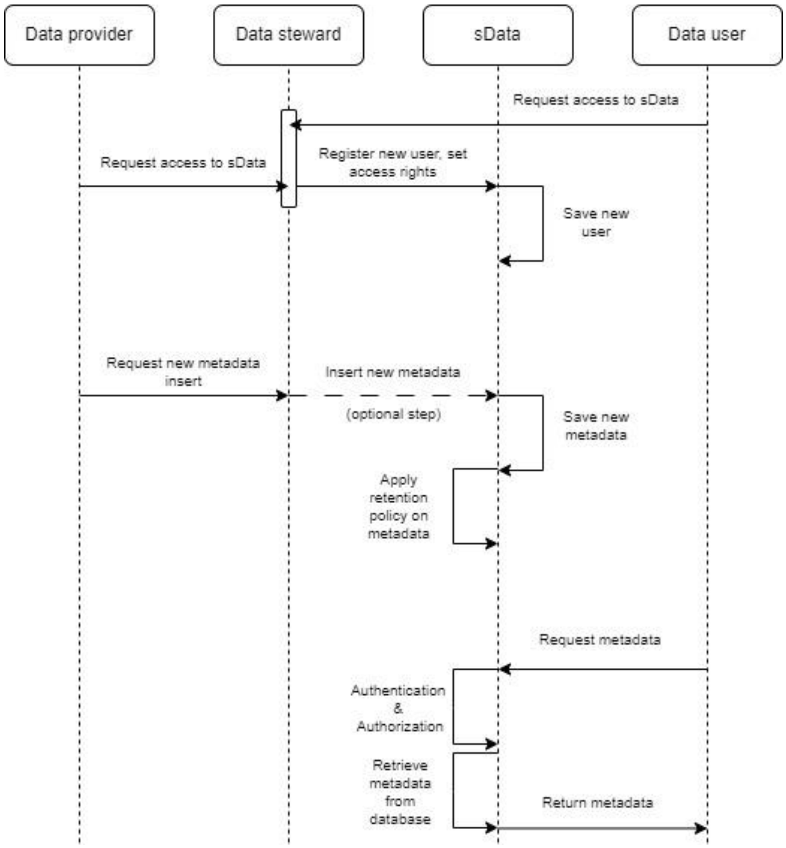

Figure 33: UC sDATA.03 sequence diagram

2.4.3.1.4 UC SI.05: Grid assessment utilizing Traffic Light System (TLS)

sGRID is designed to forecast active and reactive power for monitored grid nodes using historical power data and weather forecasts. This tool generates predictions for grid parameters such as power (P and Q), current (I), and voltage (U) for the next 24 hours, updating periodically for accuracy. Initially, sGRID conducts renewable energy sources and load forecasts to inform congestion predictions. Advanced algorithms simulate future power flows, aiming to detect potential congestion issues in advance. These forecasts help grid operators and energy providers address congestion, optimize local flexibility, and support renewable energy growth. The TLS framework within sGRID outlines actions for managing grid congestion and voltage issues, allowing DSOs to assess and request the necessary flexibility to resolve these problems.

All the data provided by DSO for the TLS calculation will be distributed to the sGRID tool via sDATA. This includes timeseries measurement data such as historical P and Q data and grid topology data.

The UC's sequence diagram is shown in Figure 34.

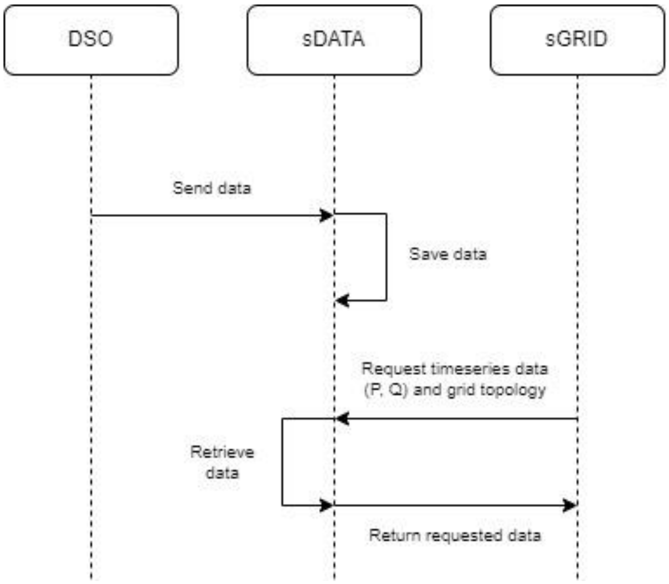

Figure 34: UC SI.05 sequence diagram

2.4.3.2 Pilots' and data user needs analysis

EPR

EPR, as a DSO, requires various types of data and data-connected tools for normal operations, including:

- Distribution grid model
- Power system measurements (voltage, current, active and reactive power, etc.)
- Integration and data management tools (from multiple sources such as SCADA, GIS, AMI)
- Simulation tools:
  - Power flow calculations (congestion management, element loading)
  - State estimation calculations
  - Reliability calculations
  - Short circuit calculations
- Power system observability and reporting tools (time series management tools and visualization)
- Analytics (data quality tools, batch processing, voltage profile management tools)
- Power quality measurements and reliability (SAIFI, SAIDI, harmonics)

In the Slovenian pilot, EPR will share both the grid model and timeseries measurement data with sDATA. This data will then be utilized in STREAM tools such as sGRID and sMART, both of which are essential for establishing the STREAM ecosystem. The data exchange must be secure, as the measurement data is very sensitive and needs to be anonymized to exclude any personal data. Additionally, all queries must be logged.

In return, EPR will receive data from sGRID that will enhance its grid observability and controllability, facilitated through sMART, including:

- TLS state for each grid element
- Remaining grid capacity
- Required flexibility (magnitude, direction and location) in case of congestion

For EPR, no edge computing devices will be used within the scope of STREAM. However, it is anticipated that some form of edge computing will be necessary in the future for the stable operation of distribution networks. The main UC for this would be real-time grid observability (TLS), as the current measurement data is received with a one-day delay due to the way the concentrator works (PLC technology). Edge computers could calculate the grid state using some measurements and combine this with state estimation calculations to enable the DSO to observe the state of the LV network in real-time.

KOL

Kolektor sETup will leverage the Avantcar data exchanged over sDATA to perform advanced user profiling. This process includes the application of baseline and flexibility forecast methodologies. The primary objective is to generate a forecasted time series detailing the consumption and flexibility availability of electric charging points. The insights derived will inform decisions regarding the optimal timing and extent to which Avantcar's fleet of e-charging points can participate in flexibility markets.

Upon receipt of the data, Kolektor sETup will execute the following processes:

1. **Baseline forecasting**: Identifying a consumption pattern.
2. **Flexibility forecasting**: Predicting the potential flexibility in power consumption, identifying periods where the consumption can be adjusted.

The output will be a detailed forecasted time series that reflects the expected consumption and available flexibility of Avantcar's charging points fleet.

Since Kolektor sETup will solely act as the recipient of the data, the primary responsibility for ensuring data security and integrity lies with Avantcar and sDATA. Avantcar, as the data provider, must define the conditions under which data can be accessed and used. Kolektor sETup, as the data user, commits to adhering strictly to these conditions and ensuring the confidentiality and security of the data received.

UL

UL's main data needs arise from its role as a developer of several STREAM tools (sPLAN, sGRID, and sMART), with the necessary data being provided by EPR through the sDATA tool. The two main types of data required for the development and operation of these STREAM tools are the grid model and timeseries measurement data.

This data will be utilized by the sGRID tool, a grid management tool, to perform congestion forecasts by predicting consumption and running load flow simulations. The results will be compared against the grid element limits defined by EPR, which will determine the TLS state of the grid element over a given period. The TLS state, along with information about remaining grid capacity or required flexibility (magnitude, direction, and location) in the case of detected congestion, will be communicated back to the DSO.

The received timeseries data will be raw and will require processing to insert missing values and remove measurement errors.

2.4.3.3 Data analysis

EPR

EPR will share two types of data with the sDATA tool: grid topology and timeseries measurements. The main information included in these data types is described below:

1. **Grid topology** (Information on grid elements and their connections):
   - Nodes: node ID, nominal voltage (Un), geolocation info

- Lines/feeders: line ID, start node ID, end node ID, Un, maximum current (Imax), line material parameters
- Switches: switch ID, status, start node ID, end node ID
- Transformers: transformer ID, nominal power (Pn), primary and secondary voltages (Up, Us), maximum short-circuit impedance (Uk_max), maximum copper losses (Pcu_max), maximum iron losses (Pfe_max), start node ID (primary side), end node ID (secondary side)
- Metering Points: node ID, smart meter number (SMM), Un

2. **Timeseries data**:
   - Smart meter measurements (P, Q, U): timestamp, value, SMM (or node ID)
   - Control measurements (for medium voltage (MV) feeders, MV/LV secondary substations): timestamp, value, node ID, control measurement ID.

The topology will be shared in CIM format, and the timeseries data in JSON format (optionally in CSV).

The two types of data that will be shared through sDATA include a number of challenges/specifics that will need to be addressed in some manner:

Distribution grid model can include **topological mistakes** such as:

- Incorrect geo topological location of consumers
- Consumer phase and zero impedance model definition for unbalanced load flow calculations (this is currently not available)
- Connections between lines (geo topology transformation problems)
- Measurement point location issues
  - SCADA to GIS connections (solved manually for the SI pilot site)
  - Consumer locations and data availability
  - MV/LV measurement location
- Missing parameters for grid components:
  - missing materials for lines
  - missing transformer types

Timeseries data challenges:

- Missing data
- Mistakes in the data (but outliers are not necessarily mistakes)
- Frequency of receiving data (for the previous day, not real-time)
- Timeseries granularity (15 minutes in LV network is insufficient for real control) and the use of averages instead of actual values, which can cause issues in simulations

Ensuring secure data exchange is crucial. This includes protecting user data and logging every query when using sensitive data (e.g., consumption measurements). To enhance data usability, EPR has developed a dedicated ETL process, including a reporting module for timeseries visualization.

While the required data is available, the CIM grid model has not been tested for load flow simulations. The timeseries data will need refinement by filling in missing values. The grid model will need validation for convergence, and the timeseries measurements will be used to validate the grid model.

KOL

Comprehensive metadata needs to be obtained from sDATA for each electric charging point. The metadata should encompass the following parameters:

- **Charging point name**: A unique identifier for each charging point.
- **Charging point ID**: A distinct identification code for each charging point.
- **Maximal power**: The maximum power output capability of the charging point.
- **Additional parameters**: Other relevant attributes identified at a later stage that could influence analytic outcomes.

For a thorough analysis, detailed time series data of power consumption for each observed electric charging point is required. This data should be raw and unprocessed to ensure that Kolektor sETup can apply its proprietary processing techniques tailored to different strategic UCs.

The data shared with Kolektor sETup is JSON data format.

UL

UL will utilize the grid model and timeseries measurement data for the development and operation of sGRID. The specifics of this data are outlined in the EPR section above.

AVA

For analytical purposes, Avantcar's EV charger data is shared with sDATA. The data is exchanged in JSON format and includes metadata for both the chargers and the EVs (battery). For each charger, metadata such as the charger's ID, name, and status are included. For the EVs, metadata includes the car ID, battery percentage or State of Charge (SoC), and the timestamp of the last data capture. Since each charger can have one or two electric plugs, the plug data is also exchanged.

Each electric plug has its own metadata, including the plug ID, associated EV charger ID, current type, maximum power, maximum voltage, maximum amperage, and status. If a plug is in charging mode, additional data is exchanged, such as the charging session ID, car ID, starting and current SoC of the battery, start and end times, energy in Wh, power in kW, amperage, and the timestamp of data collection.

When a charger is in charging mode, data is updated every 5 minutes from the start of charging. Battery data is updated every 8 minutes during charging, but less frequently when the car is not charging, depending on the limitations of the EV. Electric plug status is updated in near real-time.

2.4.4 Spanish pilot

In the next subchapters, the data that the Spanish pilot site and its users require for the operation and testing of the UCs will be analysed. The requirements of the data exchange procedure will be built upon these needs.

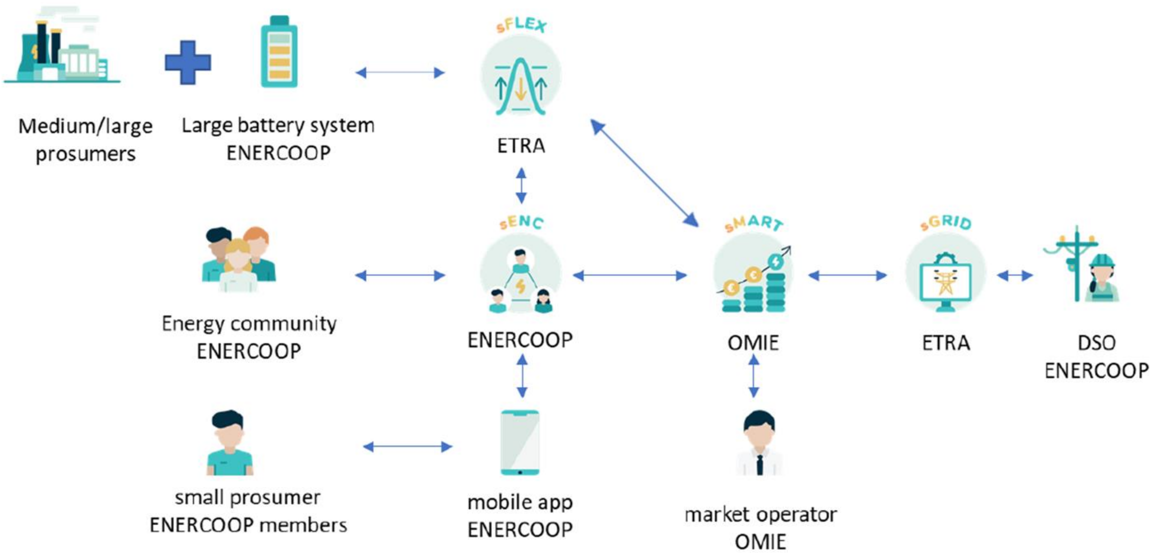

Figure 35: Spanish pilot site scheme

A Local Flexibility Market (LFM) is a complex procedure in which, as seen in the analysis of the Spanish UCs in Section 2.4.4.1, requires the participation and coordination of several tools and stakeholders as shown in Figure 35. The different tools that are used by the main actors of the LFM need a huge amount of data to run the analytics necessary to provide outputs that will be used across the STREAM ecosystem. Thus, data exchange is essential in the pilot site for the correct operation of the DSO (Enercoop), the MO (OMIE) and the Flexibility Service Providers (FSPs) and aggregator (Enercoop), as well as for the technology provider (ETRA). The data needed by each user and the source of this data is further analysed in the following sections.

2.4.4.1 Use cases analysis

Since sDATA is not being deployed under the Spanish pilot site, this section will study and analyse the UCs developed within D2.2 [23] that involve data exchange between different stakeholders of the pilot site and/or different tools. In this case, as the main focus of most of the UCs is the interaction between the different tools deployed in the pilot site, several UCs involve data exchange and are analysed below:

2.4.4.1.1 UC ES.01 – Congestion management in the MV electricity distribution network through the participation in local flexibility markets

This UC comprises the coordination of all the tools deployed within the pilot site. It begins with the detection of a congestion in the MV grid of the DSO through sGRID, either in real-time or in a forecast. Therefore, data is needed both to detect a congestion in real-time and to feed the congestion forecasts engine of sGRID. For such purposes, as starting point, the following data needs to be provided from the DSO (Enercoop) to sGRID:

- Metadata of the network assets (substations capacity in MW, impedance and longitude of the cables, voltages of the nodes, etc.).
- GIS data of the network to build the grid topology.
- Real-time and historical data of the network assets (voltage, current, power) through SCADA and Advanced Metering Infrastructure (AMI).
- Metering data of the production and supply points connected to the grid (energy generation and demand) obtained through smart meters and AMI.

Once the congestion is detected, sGRID sends the flexibility request to sMART to launch the flexibility market mechanisms. For this, the quantity of flexibility required to address the congestion is sent from sGRID to sMART which, in turn, opens an auction session, sending the required flexibility to the FSPs both through sENC and sFLEX. The FSPs send the bidding offers with data of flexibility quantity and flexibility price. Once all the offers are gathered, sMART sends to sGRID the solution, indicating the identification of the assets that will be activated and the flexibility quantity that will be activated. On the other hand, sMART send this same information to sENC and sFLEX indicating the results and the flexibility that needs to be activated.

A sequence diagram of the data exchange of this UC is provided in Figure 36:

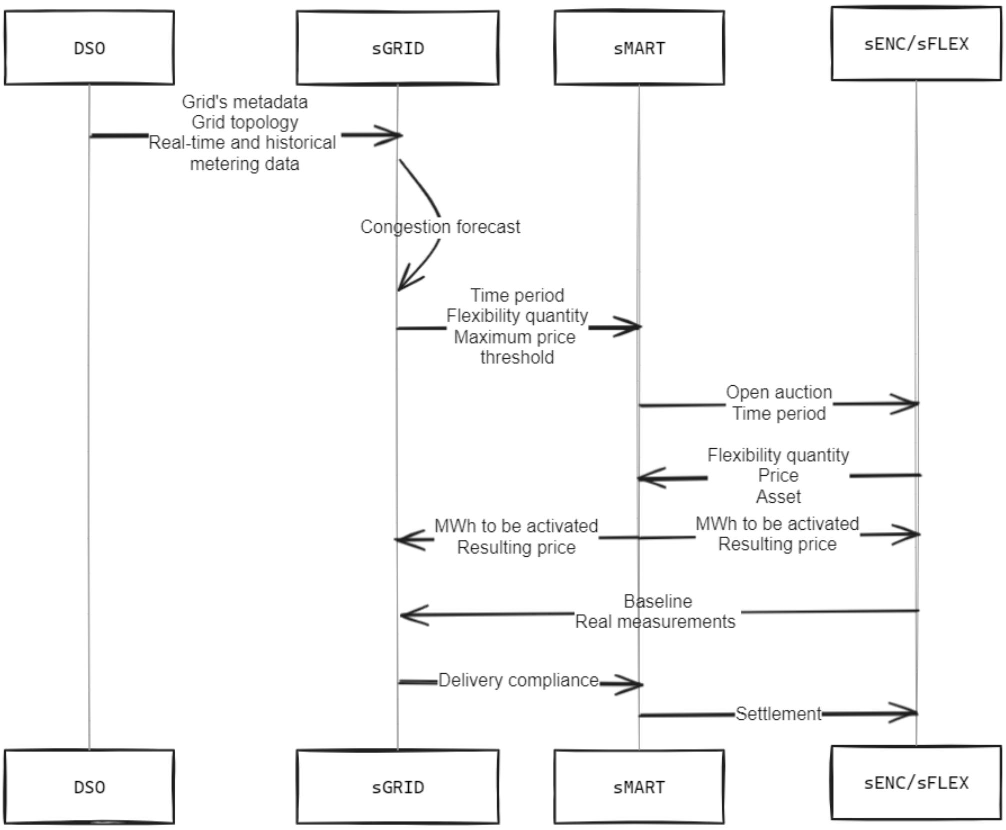

Figure 36: ES_UC.01 data exchange diagram

2.4.4.1.2 UC ES.02 – Demand response-enabled voltage control in the LV grid

The data exchanged through the different steps of this UC is similar to the data exchanged in UC ES.01. The dataflow of the UC is depicted in Figure 37:

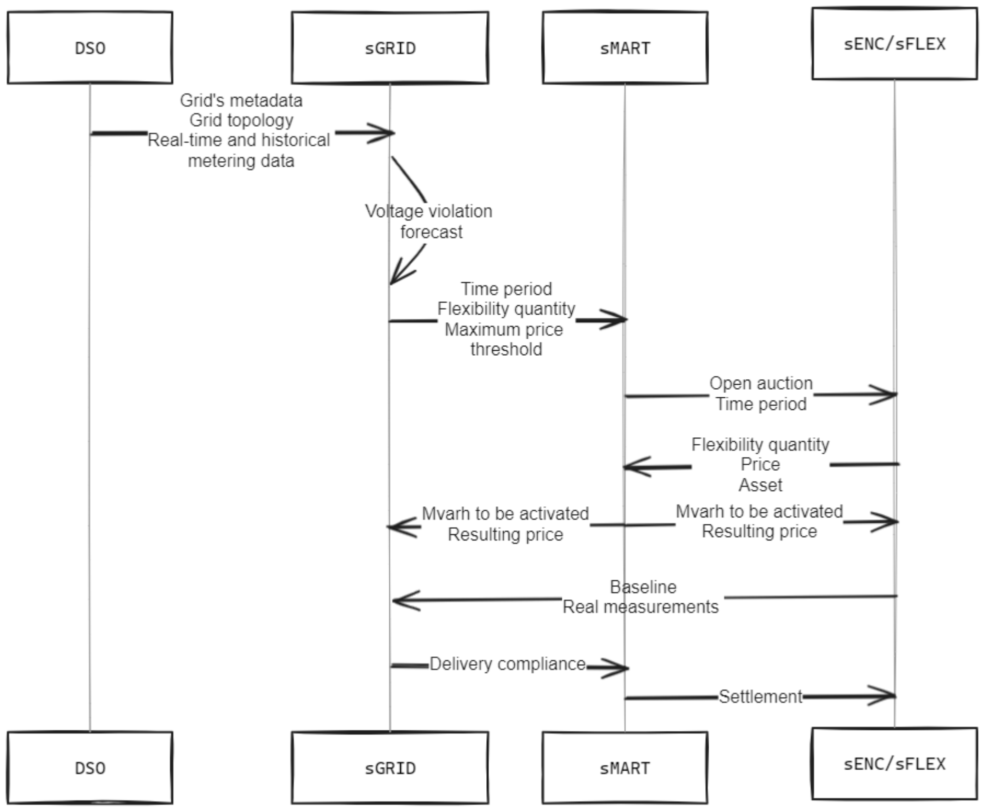

Figure 37: ES_UC.02 data exchange diagram

2.4.4.1.3 UC ES.03 – Registration and pre-qualification of resources (assets) in the Local Flexibility Market

The registration and pre-qualification process is a prerequisite to participate in the LFM. To register an asset, there are two options: either be registered directly by the asset owner for large prosumers through sFLEX, or by the aggregator for small prosumers through sENC. In both cases, metadata of the asset is manually introduced to the MO through sMART, and by using either sENC or sFLEX to the DSO through sGRID. The metadata that needs to be provided is listed below in Table 9:

Table 9: Metadata exchanged for UC ES.03

Field	Data	Description
Installation	Name	Name given to the device.
	Description	Brief description of the device.
	Contact phone	Contact phone of the device owner.
	Asset ID	Universal Supply Point Code (CUPS) or Production Installation Code (CIL).
	DSO	DSO owner of the grid in which the asset is connected.
	Agent code	Identification of the agent owner of the asset. An agent can own more than one asset.
	Status	Indicates if the asset has been already prequalified.

Location	**Region**	
	Municipality	
	Address	
	Postal code	
	Latitude	
	Longitude	
Technical requirements	**Flexibility capacity (MW)**	Maximum flexibility the asset can offer.
	Type of asset	Battery, Photovoltaic Energy (PV), HVAC, etc.
	Activation mode	Automatic or Manual.
	Activation time	Timelapse between the request of flexibility and the activation of the asset.

Once the assets have been registered a communication and measurement test should be performed as part of the prequalification procedure to ensure activations are feasible. For assets adscripted to control centres (larger than 1MW in Spain) this is not required as these are already compliant though the communication with the control centres. However, for the case of smaller assets and LV networks, this is not typically the case.

Figure 38 represents the dataflow of the UC:

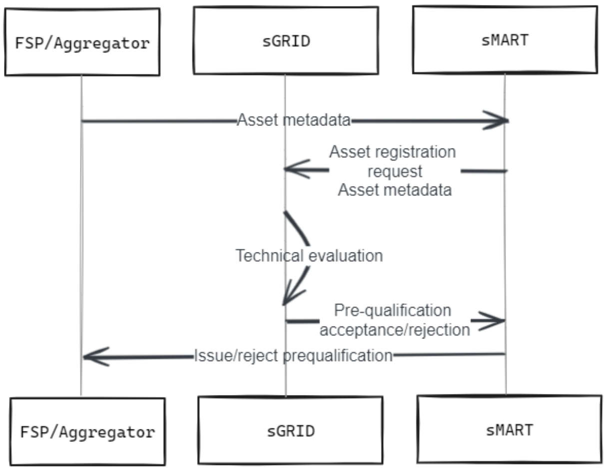

Figure 38: ES_UC.03 data exchange diagram

2.4.4.1.4 UC ES.04 – Provision of flexibility services to the local flexibility market

This UC takes as starting point the continuous monitoring of FSPs. For such monitoring, sENC and sFLEX retrieves data as follows:

- Energy consumption and generation of the FSPs: sENC and sFLEX retrieve data from the users through the AMI system of the DSO through an FTP once per day with the hourly measurements of the smart meters.
- EV charging station parameters: sFLEX and sENC communicate with the charging points of the EVs mainly through the OCPP protocol, being able to obtain measurements every 5 minutes.

The data feeds the analytics engines of sENC and sFLEX, which process the data and generate consumption and generation forecasts and, in turn, baseline profiles in order to obtain a business-as-usual profiles. For the assets in which the data is ingested by sFLEX, it is afterwards sent to sENC, which aggregates the data of the users. Once the Local Flexibility Market Operator (LFMO) receives a flexibility requirement from the DSO, the sENC and sFLEX send the flexibility availability of the assets managed by each tool to offer their flexibility. After matching FSPs and DSO bids, the market operator, through sMART, notifies the energy that must be activated by the FSPs to attend the required flexibility energy. After communicating the firm flexibility that has to be provided to each asset, sENC and/or sFLEX monitors the actual activation of flexibility, which is validated by the DSO through sGRID. In this validation process, sENC and sFLEX provide to sGRID the real measurements of the devices and the baseline profile, so the DSO can cross-check the final consumption/generation of the asset with the baseline, as stated in UC ES.01. sMART then settles the market, informing about the revenues of the session to sENC and sFLEX.

The data exchange procedure is represented in Figure 39:

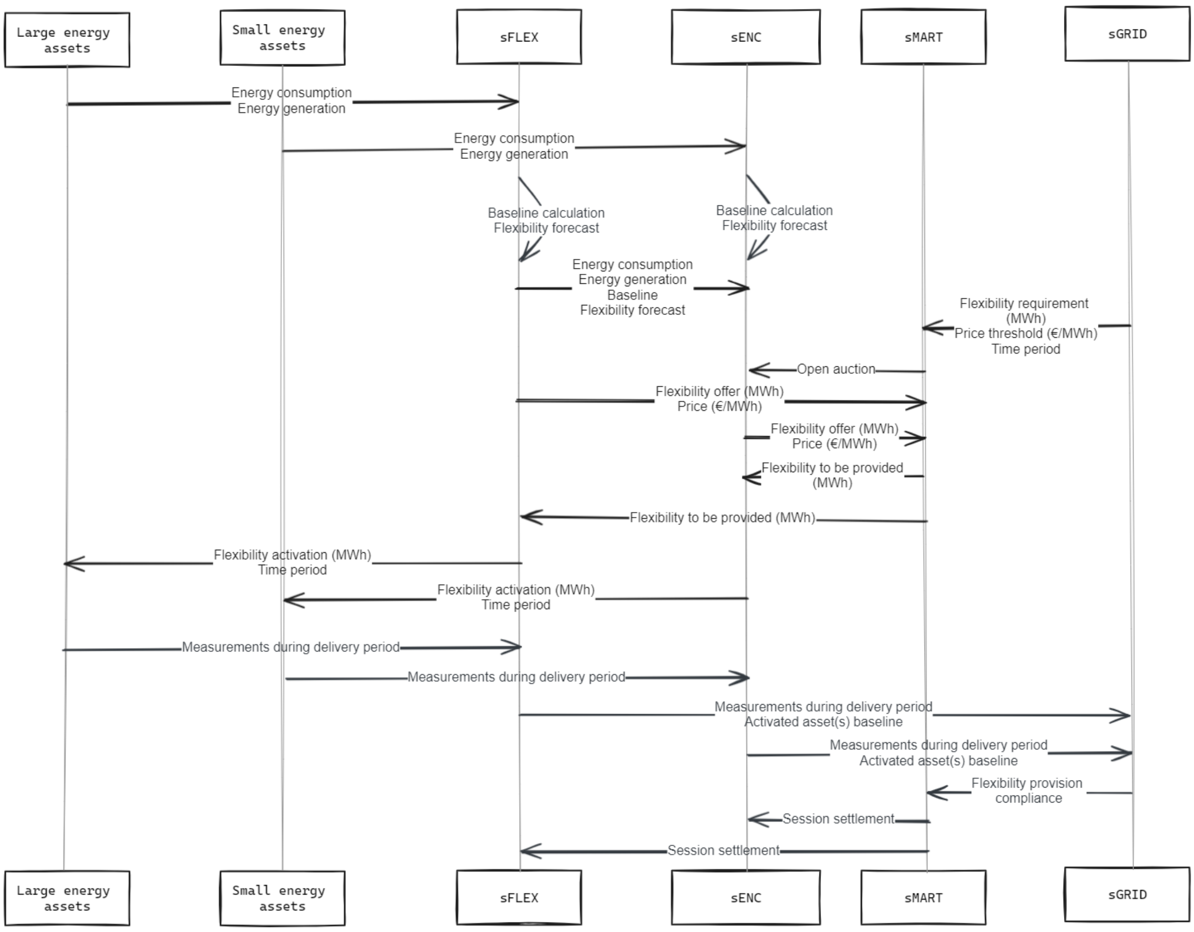

Figure 39: ES_UC.04 data exchange diagram

2.4.4.1.5 UC ES.06 – Local Flexibility Market establishment and operation between peers of the Energy Community

This UC begins when an energy generator of the pilot site receives a notification indicating a curtailment in their production. This curtailment signal is notified by the Transmission System Operator (TSO) when there is an excess of generation in the system or when an asset of the grid is expected to exceed their technical constraints. This curtailment creates an energy surplus that the generator will try to sell through the LFM in order to avoid wasting generation capacity. The generator sends the data regarding the hour and the energy surplus they will try to sell to the aggregator through sENC. Data is then exchanged between sENC and sMART, indicating the energy quantity and price that needs to be sold in the local market. Seamless communication is then needed between sMART and sENC in order to allow consumers to detect the opportunities to buy this surplus energy at a much lower price than the wholesale market. Offers to buy that energy is sent by the consumers through sENC, and sMART sends back to sENC the result of the session, indicating the quantity of energy sold to each consumer.

Figure 40 depicts the data exchange across the different tools:

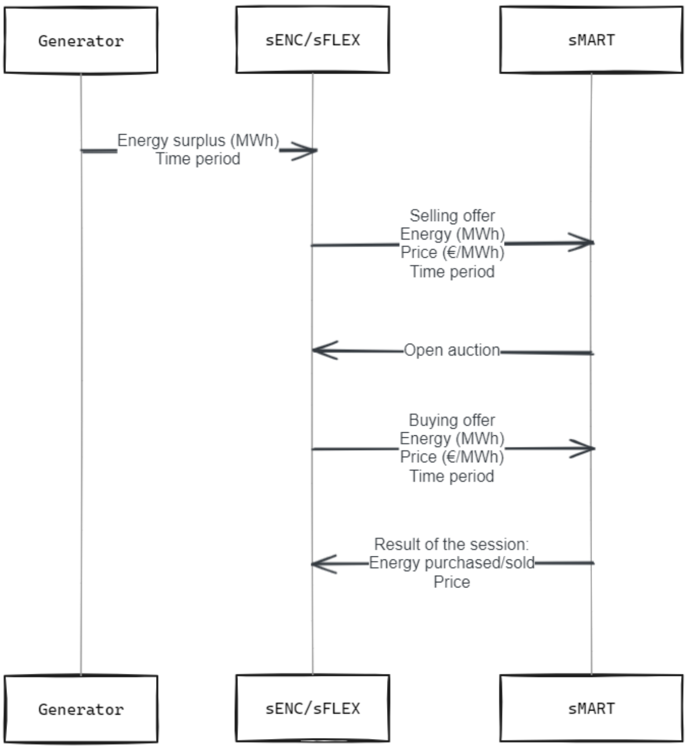

Figure 40: ES_UC.06 data exchange diagram

2.4.4.1.6 UC ES.07 - Coordinated operation of the Community assets to minimise the EnC members' cost of electricity

This UC seeks the optimal operation of the Energy Community (EnC) assets to optimise the energy costs of the members. To do so, the collected data by Enercoop's AMI is sent to sENC so the Energy Community Manager monitors and is assisted by the decision support system of the tool to perform the optimal operation. This data is exchanged with sENC in two different ways:

- For the large assets of the EnC, the data from the meters is collected by sFLEX and afterwards is sent to sENC.
- For small assets and prosumers, the data is collected directly by sENC.

After running the optimization algorithm, sENC or sFLEX notifies the assets their optimal operation schedule.

Figure 41 depicts the data exchange of this UC:

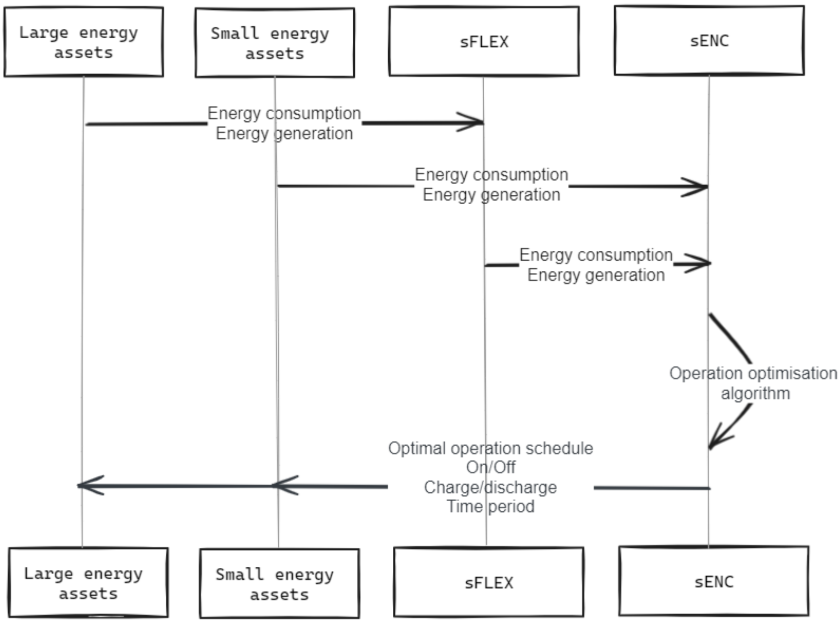

Figure 41: ES_UC.07 data exchange diagram

2.4.4.2 Pilots' and data user needs analysis

Enercoop as DSO

Enercoop, in its role of DSO within the Spanish pilot flexibility market, is in charge of ensuring the secure operation and supply of electricity within its network by identifying potential grid issues and activating the local flexibility market mechanisms to solve them. Additionally, Enercoop is co-responsible, together with OMIE, of the pre-qualification and qualification processes to allow a user to provide flexibility in the market. To meet these liabilities, the DSO requires several datasets as detailed in Table 10:

Table 10: DSO (Enercoop) data needs

Data	Source of data	Type of data
Energy consumption	Smart meters of supply points and AMI	Raw data, time series
Energy generation	Smart meters of generation and supply points (with self-consumption)	Raw data, time series
Voltage and power at grid buses	DSO SCADA system	Raw data, time series
Specifications of the assets of the grid	Assets specifications	Metadata
Grid topology	GIS model	Geospatial data
Specifications of the FSPs	Technical specifications of the assets	Metadata
Grid forecast (Power, voltage)	sGRID tool	Processed data, time series

OMIE

OMIE is the local MO in the Spanish pilot site. Its main duties are the management, administration of market platforms, matching, clearing and settlement processes of the local market upon the reception of a flexibility request coming from the DSO. Besides, as previously stated, OMIE is co-responsible of the pre-qualification and qualification processes to assess the suitability of an asset to provide flexibility services. The data needed to all these processes are listed in Table 11.

Table 11: MO (OMIE) data needs

Data	Source of data	Type of data
Flexibility offers (requests from DSO and bids from FSPs)	sGRID, sENC and sFLEX tools	Structured data
Specifications of the FSPs	Flexibility assets	Metadata
User log in data	Input by the users upon registration in the LFM	User data

Enercoop as Aggregator

Apart from its role of DSO, Enercoop, as manager of the EnC of Crevillent, acts as aggregator of the small prosumers to allow them to provide flexibility services. For such purpose, the data the EnC manager needs are (Table 12):

Table 12: FSPs and aggregator (Enercoop) data needs

Data	Source of data	Type of data
Energy consumption	Smart meter	Raw data, time series
Energy generation	Smart meter and/or inverter API	Raw data, time series
Baseline	sENC and sFLEX baseline calculation modules	Raw data, time series
Demand and production forecasts	sENC and sFLEX demand and production forecasts modules	Processed data, time series
Energy billing information	Energy bills, energy provider or market prices	Energy costs (€/kWh), time series

ETRA

ETRA is in charge of developing the sFLEX, sENC and sGRID tools in the Spanish pilot site. These tools require a variety of datasets in order to provide the data analytics and visualization tools necessary for all the actors in the Spanish pilot site to operate correctly. As the tools are meant to be used by the other users (DSO, aggregator and MO), all the raw data needed by the users are also needed by the technology provider to feed the tools. Apart from that data, user data for registration in the tool is needed (user, password, type of user, etc.), as well as weather data and forecasts as inputs of the consumption, generation and flexibility forecasts.

As can be inferred from the previous analysis, not only raw data is necessary. Data processing is vital to produce all the necessary intelligence and information. This processed data is essential for fulfilling the requirements and functions of a local flexibility market. The tools deployed in the pilot site ingests the raw data and provides these processed data. For example, sFLEX and sENC are fed by energy demand and generation time series, and produces energy demand, generation and flexibility forecasts, which are needed in order to estimate the future state of the grid and to be able to offer the optimal amount of flexibility to the market. The DSO also needs processed data, such as congestion forecasts in order to activate the flexibility market when needed, and the MO needs to know the baseline of the flexibility providers in order to confirm the correct activation of flexibility by the FSP, etc. These tools also integrate libraries such as Highcharts [24], Recharts [25], and Tremor [26] to visualize the information in graphical form, which allows the user to monitor their data.

The coordination of the tools' data exchange is crucial in order not to incur in operation errors, leading to a harm in the grid that may cause issues in the energy supply and comprise the energy security, or damaging the assets that provide flexibility services. To avoid this, data need to be transferred seamlessly across the STREAM ecosystem in the pilot site. Data must be exchanged in real-time and continuously between the tools and the SCADA and AMI systems of the DSO to detect potential grid issues as fast as possible. The structure, type of data, magnitude of the data and the data exchange protocols must be established and standardized in the pilot site in order not to incur in data mismatches and communication issues.

Since all the data exchanged across the Spanish pilot site implies sensitive data (energy consumption, user data, DSO's data, etc.), the information obtained in the pilot site has to be treated confidentially. To ensure this, the data must be anonymised and encrypted, and credentials are needed to both send the data to the tools and to log in to them in order to ensure security. Only data owners and authorized third parties will have access to the data. Security is especially crucial in the sMART tool, as the registration of the user also implies the deposit of guarantees and some financial data. In this case, a certificate is also required to access the data. A more detailed security analysis and the measures undertaken at the pilot site to safeguard sensitive data will be provided in upcoming deliverables (D3.2) of STREAM project.

The amount of variables gathered from the different systems, together with their own technical restrictions, entail some limitations in the frequency of the messages sent to the different STREAM tools, for example:

- DSO SCADA: 5 minutes
- Community battery: 1 minute
- EV chargers: 5 minutes
- AMI: hourly values, sent once a day

2.4.4.3 Data analysis

Based on the previously identified data user needs, the data sets coming from external sources are analysed here from a technical perspective, including formats, communication standards, ETL procedures, as well as their availability and challenges to integrate them.

Enercoop as DSO

Energy consumption and generation values come from Enercoop's AMI. Hourly records from the previous day are compiled from the smart metering system through a series of data concentrators and centralised in an FTP server. The data is stored in XML files organised per substation, concentrator, meter, and date. A dedicated service accesses the FTP periodically to look for updates, which are then transformed into JSON messages and sent via AMQP to the STREAM ecosystem. Historical records are available for several years, which are highly valuable for baselining and forecasting.

Voltage and power values at grid buses are retrieved from the SCADA system, which provides a SOAP service to access the values of its registers. An application developed in .NET Core interfaces this service to read these values every few minutes, transform them into JSON format, and send them to the STREAM tools via MQTT. The main challenge with the integration is not technical: linking the tag names provided by the SCADA to the parts of the grid topology they belong to. This context information needs to be handed by Enercoop, who has the knowledge of both their network and their systems.

The grid topology model and the specifications of each asset in the grid is provided by Enercoop in an offline manner, via Excel files with the details of each type of element and the relations among them. These files are fed to a process in Python that extracts the information to create XIIDM files (one for the MV network and one for each LV feeder) that are then imported into the PowSyBl[4] Network Store repository, from which sGRID extracts all topology data. The problem with this process comes when the topology of the network needs to be updated, which is a process that has to be performed manually. It is currently being explored with Enercoop the development of an API for their GIS system in order to enhance this process.

OMIE

The public message interface of sMART supports the connection of clients and their operations in the market. The MO needs to provide access to a new client by registering it in the platform and providing them with a unique identifier and a certificate, both of them to be used in the exchange of messages with the server.

The log in of the user is done via HTTPS using the certificate. If successful, the server returns the ID of the agent associated to the user, as well as the credentials (username and password) to connect to a RabbitMQ broker also using web services based on XML standards.

All subsequent communication in the market, including the operations for flexibility exchange, is performed through this RabbitMQ broker using AMQP over SSL, with JSON messages. There are three types of messages:

- **Inquiry:** Used by the client to request information about their certificate, agent, contracts, bids, transactions, etc. Inquiry requests are sent to an exchange and their response is received in a private queue.
- **Management:** Used by the client to perform operations in the market, such as creating, modifying or cancelling bids. The payload of management messages is signed using JSON Web Signature [27] and sent to an exchange. The client receives a response in their private queue, while changes in the market as a result of the operation are broadcasted to all involved clients.
- **Broadcast:** Used by the market to communicate updates to all parties in a specific zone, under a specific agent, or all connected clients (e.g. for server status notifications). A common broadcast exchange is used for general and zone-specific messages, while each agent has a dedicated exchange for their messages. Updates over contracts, transactions, and orders are sent as broadcast messages.

Finally, the specifications of flexibility assets registered in sMART are provided through its GUI. The client certificate is required to access the application, as well as the installation of a VPN-like software for security reasons. Once authenticated, the client is able to register their assets in the market, including their flexibility capacity and activation time. This information is subsequently validated by the MO and the DSO in order for the asset to participate in the market.

Enercoop as Aggregator

Different methods and technologies have been employed in order to integrate the wide range of flexibility assets in the Crevillent pilot. A summary is provided in Table 13.

Table 13: FSPs and aggregator (Enercoop) data

System/device	Technologies	Data formats
AMI system	FTP (source), AMQP	CSV files (source), JSON
Smart Meters	MQTT	JSON
Collective and domestic self-consumption	REST API (source), MQTT	JSON
PV + community battery	Modbus TCP (source), MQTT	JSON
EV chargers	OCPP 1.6, Modbus TCP, MQTT	JSON
Smart Plugs	OpenHAB, REST API, MQTT	JSON

The integration of some sources has been straightforward, especially those devices that presented IoT capabilities (Smart Meters, Smart Plugs). The maps of Modbus registers of some devices were sometimes not completely clear and required some empirical tests to assess the meaning or magnitudes of some of them. Regarding the EV chargers connected through OCPP, some wiring replacements had to be performed after the first tests, which added some unexpected efforts to the integration, but could eventually be implemented successfully.

2.5 FUNCTIONAL REQUIREMENTS

In this section, a list of *general* functional requirements for sDATA and data exchange is provided, covering the key requirements that apply across all pilot sites. The requirements are detailed in Table 14. Detailed requirements specific to each pilot are outlined in 6.1.

Each requirement has a given priority, where the priority scales from 1 to 5, with 1 as the lowest and 5 as the highest priority.

Table 14: General functional requirements

Name	Description	Priority
Data exchange	Users should have the capability to request and retrieve data to which they have been granted access. Data exchange should adhere to a predefined standardized format to ensure consistency and interoperability. Data exchange using standard communication protocols, such as MQTT, AMQP, RESTful APIs must be supported.	5
Anonymized and encrypted data	The data exchange must be highly secure, given the sensitivity of the measurement data, which requires strict anonymization to ensure no personal information is included. Additionally, robust encryption protocols must be implemented to protect the data during transmission and storage, safeguarding it from unauthorized access and potential breaches.	5

Data pre-processing	The data obtained from the data providers must be pre-processed by cleaning the data, deleting atypical and NaN values, and addressing data quality. This data pre-processing should be done either by the data management services (sDATA) or directly by the tool that uses the data, ensuring high-quality historical data is available for modeling and analytical purposes.	5
Data visualization	Data visualization tool for measurement data needs to be included to help users make sense of the data and support informed decision-making.	5
Data storage (retention policy)	Data received from external sources needs to be persisted for subsequent access both by end users and internal services (e.g. for data analysis). The data providers define the retention period and minimum data persistence duration for which the data must remain accessible in the repository.	5

2.6 NON-FUNCTIONAL REQUIREMENTS

In this chapter, *the general* non-functional requirements, focusing on availability, security, performance, connectivity, interoperability and scalability, for sDATA (and data exchange) in the STREAM project are described. These requirements represent a comprehensive overview, encompassing all non-functional aspects relevant to all pilot projects. Specific non-functional requirements for each pilot project are provided in 6.2.

2.6.1 Availability

Data availability

For the purpose of analysis in STREAM tools using data provided by sDATA, it is essential that the data be available upon request. The data exchange services must be operational continuously to ensure seamless communication.

Additionally, the system must collect data frequently enough to prevent any loss of operational information. This may require continuous data collection in some cases, while in others, a once-per-day data retrieval may be sufficient.

Data storage availability

The data storage solution must support high availability to provide a decent level of quality to the end users.

Data quality

sDATA must provide consistent and coherent data to ensure high-quality analysis by STREAM tools and serve as a reliable data source for analytical purposes.

2.6.2 Security

Authentication and authorization

sDATA and data exchange processes must implement authentication and authorization mechanisms to verify user identities and control access to data. Each user must have associated a domain of data that they can access. No unauthorized access can be permitted.

Secure data exchange

The platform must ensure secure data exchange to protect sensitive information. Security features, such as using API keys or user credentials for data exchange, must be implemented to ensure that only trusted users have access.

Data security

Since the data exchanged within STREAM project can be sensitive, all the data must be anonymised and encrypted to prevent misuse and ensure the security of the energy supply.

2.6.3 Performance

Data collection

The data collection process must be frequent and adhere to specific intervals defined by data providers to prevent any loss of data. It should be efficiently executed to ensure timely updates without disrupting other system operations or user interactions with the platform.

Data collection processes together with data storage solution must support high speeds of data ingestion.

Database

The sDATA platform must incorporate multiple databases to optimize data storage and operations. Specifically, a database designed for efficient time-series storage and operations, such as TimescaleDB, must be used due to its advanced handling of temporal queries and scalability. The database should be designed to efficiently store and manage a vast amount of measurement data, potentially reaching billions of records, over a predefined period, such as one year.

Data exchange

Data exchange among users must be efficient, ensuring fast transmission speeds to facilitate timely communication and collaboration. It's essential that data exchange processes do not disrupt other operations within the platform, maintaining optimal performance and responsiveness for all users.

2.6.3.1 Edge computing

Edge computing is implemented only in the Finnish and Slovenian pilots. While the objective in both cases is generally similar—to provide fast computation close to the data source—the implementations differ significantly. As a result, the edge computing requirements are described separately for each pilot.

Edge computing is not part of the sDATA tool. As previously mentioned, sDATA's primary function is to store and exchange data for analytical purposes, focusing on historical data. Attempting to transport near-real-time data through sDATA would significantly increase data transfer speeds and delay calculations, ultimately defeating the primary goal of edge computing—ensuring fast and efficient processing.

2.6.3.1.1 Slovenian pilot

The primary objective of STREAM is to establish an innovative and robust ecosystem that maximizes the use of small-scale flexibility and integrates it into existing wholesale power markets. Additionally, STREAM aims to develop new local flexibility markets. To begin harnessing this flexibility potential, it was essential to first identify the assets that could be utilized across various markets.

In the Slovenian pilot site, partners led by KOL identified the following flexibility assets, Table 15:

Table 15: Overview of the flexibility assets in the Slovenian pilot

Asset	Company	Nominal power	Integration option
Cold storage rooms	Incom Leone	1.500 MW	Siemens controller S7-1200
Mill	Mlinotest	0.450 MW	Siemens controller S7-1200
PV	Petrič	0.250 MW	REDUXI
PV	Tosla	0.250 MW	REDUXI
Cold storage rooms	Tosla	0.250 MW	Presumably Siemens controller S7-1200
Cold storage rooms	Mlinotest	0.215 MW	Siemens controller S7-1200
PV	Metal design	0.100 MW	REDUXI
E-charging stations	Tosla	0.044 MW	REDUXI
Heat pump	Tosla	0.001 MW	REDUXI
E-charging stations	Avantcar	/	Avantcar platform
Total	-	3.600 MW	-

KOL is currently integrating identified flexibility assets into the sFLEX aggregation platform. Four assets—cold storage rooms of Incom, Mlinotest, and Tosla, along with the Mlinotest mill—have been selected to participate in the ancillary services market for the TSO. Therefore, this integration must comply with the regulations set by the Slovenian TSO, ELES. For manual frequency restoration reserve (mFRR), measurements need to be collected every five minutes. Upon receiving an activation signal from the TSO, KOL must transmit this signal to the flexibility assets, which are required to respond within 12.5 minutes. For these purposes, KOL is/will be integrating the mentioned assets with the proven industrial Siemens controller S7-1200 to fulfil all requirements by the TSO. Integration via sDATA would not be optimal for the described purposes since the assets' measurements and management cannot be exchanged in the near to real-time.

Other assets—including the PV systems of Petrič, Tosla, and Metal Design, as well as Tosla's E-charging stations and heat pump—have been chosen to participate in the newly established local flexibility markets being developed within the project. Since the requirements for these markets are still being defined and specific characteristics of controller is not needed, the REDUXI home energy management system device was selected to integrate these assets into sFLEX. REDUXI facilitates the collection of measurements at one-minute intervals (at least) and manages the assets by sending the necessary set points. Again, sDATA would not be optimal for the predicted functioning of the local flexibility market, since the activations of the flexibility assets would need to be performed in the near to real-time, functionality not supported by sDATA.

E-charging stations were integrated into sFLEX through the Avantcar platform and concurrently connected via sDATA. The primary reason for this initial integration via sDATA is to enable the processing and analysis of consumption data. This allows for an assessment of the potential to utilize these assets in flexibility markets before addressing the complexities of full integration.

2.6.3.1.2 Finnish pilot

Optiwatti architecture is based on Edge computing.

Each apartment is equipped with a computing unit called the controller. Controllers get input from the temperature and humidity sensors and control the heating devices through relays. Controllers are able to store data temporarily in memory and communicate with the backend which contains the databases.

Controllers execute all the computation that is apartment specific. Algorithms perform the heat-up calculation and estimate when to switch on a certain device based on different input variables such as desired temperature, outdoor temperature, weather forecast, energy's spot price, occupancy information, etc.

In the same way controllers perform the heat-down calculation and estimate when to switch off a certain heating device.

Controllers are also responsible of estimating the available flexible power for FCR usage at apartment level. However, the calculation related to the flexibility groups which form an aggregated power for FCR market is done in the backend since such calculations are not apartment specific.

The communication between the Flex server which is located in the backend and the Flex client which is located in the controller uses MQTT/SSL.

Figure 42 shows the main components of Optiwatti architecture.

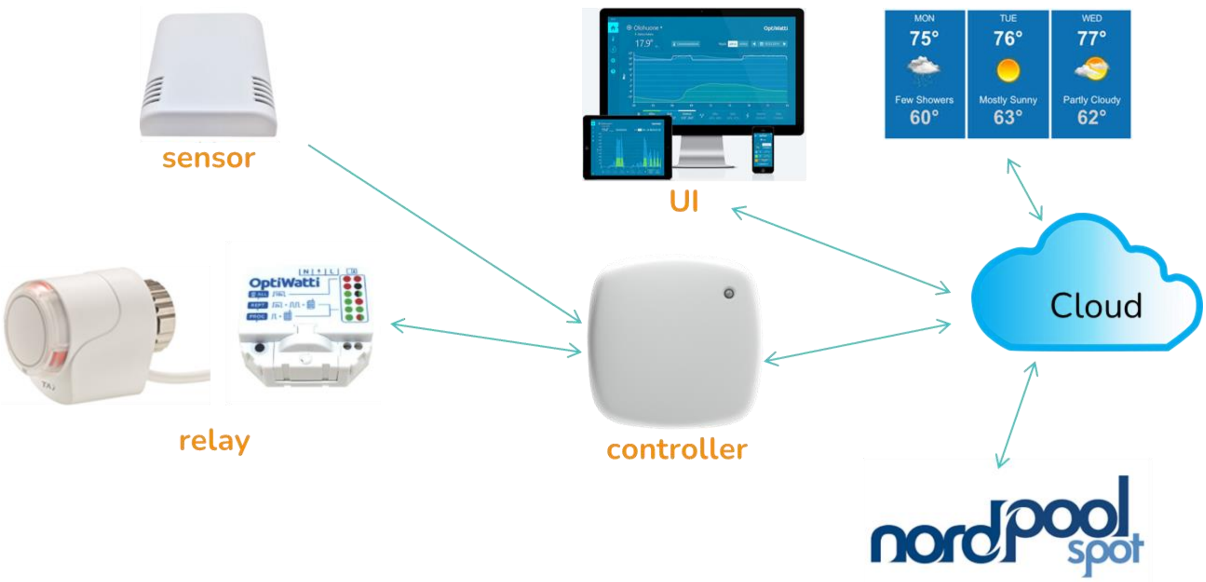

Figure 42: Optiwatti edge computing solution

2.6.4 Connectivity

The sDATA platform must support robust connectivity to ensure all tools can seamlessly gather data on request, provided they have sufficient rights. Reliable and secure connections are essential for efficient data exchange, enabling real-time access and updates.

The system should facilitate smooth connectivity, ensuring consistent data flow and interaction between various components and users across the platform. High availability and low latency are crucial to maintain optimal performance and user satisfaction. Connectivity uptime must be ensured for ingestion, storage, and retrieval of the data.

2.6.5 Interoperability

Standardized data formats

The sDATA platform must support data exchange in standard formats such as XML, JSON, and CSV to ensure interoperability.

The definition of data models depends on the communicational necessities of the tools.

API Compatibility

The sDATA platform must provide well-documented and consistent APIs that adhere to industry standards (e.g., REST, SOAP) to facilitate data exchange between diverse applications and platforms. Other standard exchange protocols, such as MQTT, AMQP are also acceptable.

2.6.6 Scalability

The sDATA tool must be designed to handle increasing loads seamlessly, ensuring scalability to accommodate a growing number of users, data volume and data traffic, even though the number of customers is not expected to grow for this specific project.

3 SPECIFICATION

The specification chapter provides a detailed framework for the pilot-specific implementations derived from the requirements. While the specifications are rooted in the general requirements, they are tailored to the unique needs of each pilot. This chapter is thus oriented towards the specific pilots, offering in-depth guidance on their implementation.

Each pilot section includes the following components:

1. **Functional Specification**: This outlines the implementation of functional properties specific to the pilot.
2. **Non-Functional Specification**: This details the non-functional aspects, categorized into several types: availability, security, performance, connectivity, interoperability, and scalability.
3. **System Specifications**: This section is divided into three main parts:
    - **Architecture**: This part focuses on the architecture of sDATA and its integration with other tools within the pilots. For pilots not implementing sDATA (Spanish), this section will instead detail the data exchange between various tools.
    - **Open APIs**: This section lists the APIs used for data exchange, emphasizing the interactions between sDATA and other tools. For pilots not using sDATA, this section will include APIs relevant to the data exchange among other tools.
    - **sDATA Compliance with Open Data Spaces**: This describes how the pilot's implementation of sDATA adheres to the key requirements and principles of Open Data Spaces.

The functional and non-functional specifications are presented in a table format. Each row in the table represents a distinct functionality, including its name and description. Non-functional requirements are further detailed in an additional column, which specifies the type of requirement such as availability, security, performance, connectivity, interoperability, and scalability.

3.1 FINNISH PILOT

3.1.1 Functional specification

Sections 2.4.1 and 6.1 already contain all the necessary details for the functional specification. Since the implementation directly follows these requirements, the specifications are not repeated here to avoid duplication.

Section 6.1 (the FI pilot part) contains a complete list of requirements for implementing the FCR services as well as for migrating the existing periodic measurements database. Sections 2.4.1.2 and 2.4.1.3 contain the details related to the implementations of the requirements for the FCR service and the related UCs, including all the needed fields.

Tables Table 16 and Table 17 contain the additional requirements that have been created during the implementation phase.

Table 16: Functional specification of sDATA in the FI pilot

Functionality	Description
Data warehouse	An aggregator needs a data warehouse containing all the relevant data of the pilot apartments including sufficient historical data with enough details for the periodic measurements in order to be able to fit the prediction models. Hour aggregates for the periodic measurements are needed for five years and the data for those apartments shall not be purged in any way. Data will be compressed. In addition, it might be beneficial to copy also control data (settings) related to those apartments in order to facilitate data analysis by having all the needed data in one database. The data shall be anonymized: no personal information shall be included in the data warehouse. If needed, data marts can be created to share data with the relevant parties. As an alternative to data marts, CSV files can be produced by filtering the needed information. The connection to the data warehouse shall be a secure connection (an encrypted SSL connection to a selected port).

3.1.2 Non-functional specification

Section 6.1 (the FI pilot parts) contains a comprehensive list of requirements. This section contains additional details related to security and privacy, performance and scalability. The sFLEX-sDATA interface is an Optiwatti internal interface so there are no further details about connectivity and interoperability. Since the implementation directly follows these requirements, the specifications are not repeated here to avoid duplication.

Table 17: Non-functional specification of sDATA in the FI pilot

Functionality	Description	Type
Secure access to the control database	The control database is accessible only through a server which is located in Optiwatti private AWS Cloud. Access to such server is possible only through SSH.	Security and Privacy
Secure access to the measurement database	A Virtual Private Cloud is created in AWS and peered with Timescale Cloud. Optiwatti private IP address range is configured in Timescale server.	Security and Privacy
Secure access to the data pipeline and data visualization servers	Servers are accessed by using SSH. Connections with RDP are allowed only by using a SSH tunnel.	Security and Privacy

Measurement database performance	Timescale is configured in order to have enough capacity to run the scheduled policies. The value of the parameter timescaledb.max_background_workers shall be equal at least to the number of the databases in the server + two times the number of databases which have timescale extension + the number of concurrent background workers multiplied by the number of databases that have automatic aggregate view policies. It has been set to 72. The value of the parameter pg.max_worker_processes shall be equal at least to the value of the previous parameter + the number of CPUs in the server. It has been set to 80.	Performance
ETL data pipeline server HW	The Apache Ni-Fi pipeline runs into an AWS EC2 instance of type m7g.2xlarge. It is a cost-effective ARM-based instance with 8 vCPUs, 32 GB RAM and network bandwidth up to 15 Gbps which provides enough capacity to run the ETL data pipeline.	Performance
ETL data pipeline performance	In order to be able to handle the required amount of data, the following changes have been done to the Ni-Fi default settings in order to increase the heap memory in the server and the amount of data that can be hold in each Apache Ni-Fi queue: • heap memory in bootstrap.conf o java.arg.2=-Xms512m > java.arg.2=-Xms16g o java.arg.3=- Xmx512m > java.arg.3=-Xmx16g • nifi.properties o nifi.flow.configuration.archive.max.storage=500 MB > nifi.flow.configuration.archive.max.storage=5 GB o nifi.queue.backpressure.size=1 GB > nifi.queue.backpressure.size=2 GB	Performance
Measurement database scalability	The measurement database is deployed in a Cloud-based Timescale solution which offers huge possibilities for scalability (from the current 4 CPUs, 16 GB RAM, 712 GB storage to 64 CPUs, 488 GB RAM, 10.240 GB).	Scalability

3.1.3 System specification

3.1.3.1 Architecture

Figure 43 shows the SW architecture of the Finnish pilot.

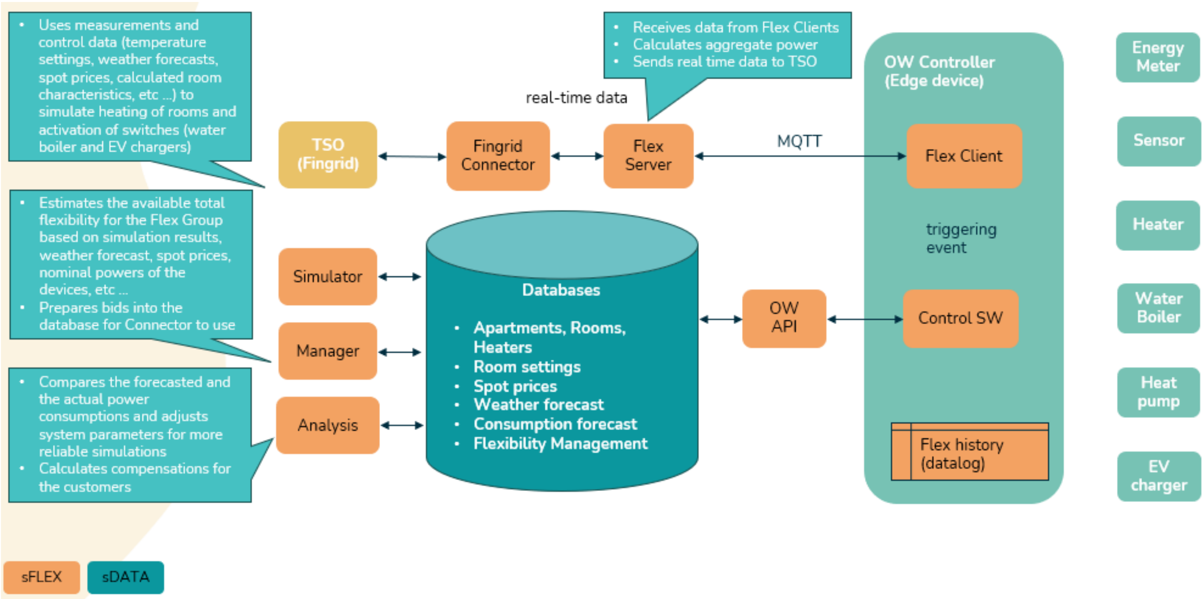

Figure 43: SW Architecture

Figure 44 shows the **general principles** to decide where to store data:

- Real time data is not stored in databases, but it stays in memory in Optiwatti backend (sFLEX).
- Control data, including any customer settings, is stored in a relational MySQL database in AWS cloud.
- All the periodic measurements are stored in a PostgreSQL-based Timescale database in Aiven cloud, which is optimized to handle time series.

Figure 44: Database types

In addition to the control database, three separate types of **periodic measurements databases** are used in Timescale:

- Legacy Optiwatti periodic measurements
- Explicit flexibility logs
- Data warehouse

Each database type has an alfa version and a production version.

In the minimum viable product (first prototype) the emphasis is on sFLEX implementation, and a local database is used.

3.1.3.2 Periodic measurements migration

Figure 45 shows the data integration solution that is used to copy existing periodic measurements into the new time-series-based database management system.

The solution is deployed completely into the Optiwatti private cloud in Amazon Web Services (AWS) and it uses free open-source tools.

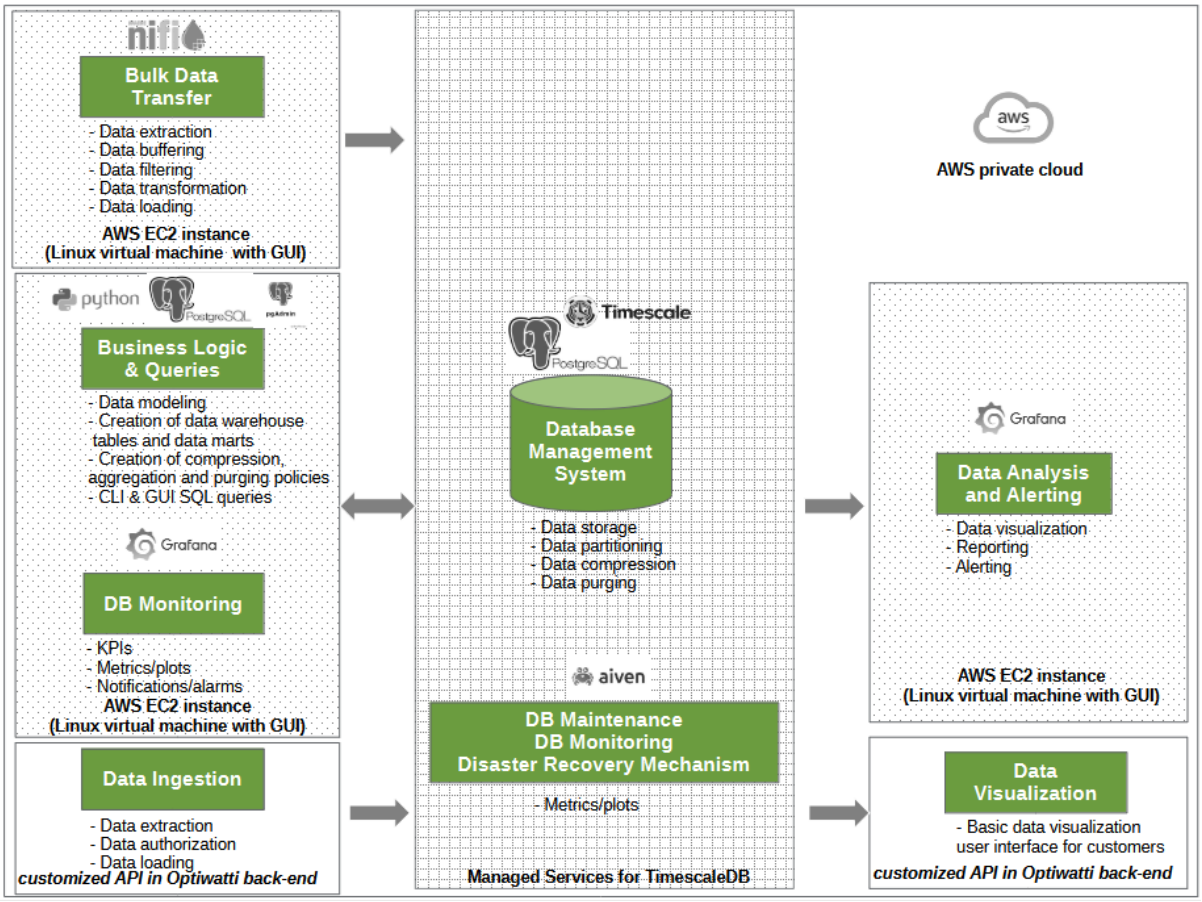

Figure 45: Data integration solution

The solution is based on **Timescale** database engine for data storage, partitioning, compression, time aggregation and purging. **Python** scripts are used to create the necessary tables and aggregate views as well as the compression, aggregation and retention policies. **Apache Ni-Fi** is used for copying the existing data and modifying it to fit into the target database model. A measurement API shall be built in the backend to load data in the database and to read the data for the UI. **Grafana** is used for in-house data visualization.

Aggregation, Compression and Retention Policy

Figure 46 summarizes the aggregation, compression and retention policy used in the new periodic measurements database.

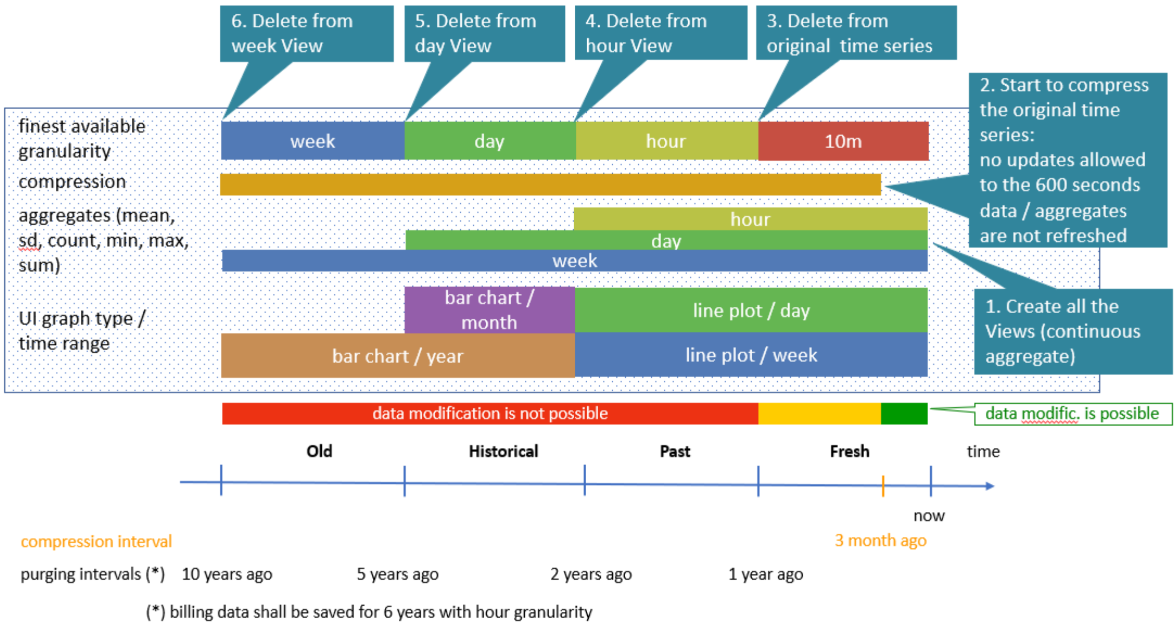

Figure 46: Aggregation, compression and retention policy of periodic measurements

3.1.3.3 Data warehouse

The Timescale database containing legacy Optiwatti periodic measurements does not contain detailed data for many years backward since, for optimizing database costs, data is purged regularly as shown in Figure 46. Currently, hour aggregates are kept only for 2 years whereas for data analytics purposes and for fitting prediction models it would be useful to have such aggregates for few more years.

Therefore, a data warehouse is built to contain data of the STREAM pilot apartments. Raw data for the years from 2019 to 2023 is contained in the periodic measurements part. The measurement interval in raw data is 10 minutes. This data, as well as the hour, day and week aggregates, are not purged.

The data has been retrieved from the historical archive contained in cold storage.

Figure 47 below shows the name of the data warehouse table and of the aggregate views containing periodic measurements related to the pilot apartments.

List of relations

Schema	Name	Type	Owner
public	dw_avg_day_aggregates	view	tsdbadmin
public	dw_avg_hour_aggregates	view	tsdbadmin
public	dw_avg_week_aggregates	view	tsdbadmin
public	dw_count_day_aggregates	view	tsdbadmin
public	dw_count_hour_aggregates	view	tsdbadmin
public	dw_count_week_aggregates	view	tsdbadmin
public	dw_max_day_aggregates	view	tsdbadmin
public	dw_max_hour_aggregates	view	tsdbadmin
public	dw_max_week_aggregates	view	tsdbadmin
public	dw_min_day_aggregates	view	tsdbadmin
public	dw_min_hour_aggregates	view	tsdbadmin
public	dw_min_week_aggregates	view	tsdbadmin
public	dw_stddev_day_aggregates	view	tsdbadmin
public	dw_stddev_hour_aggregates	view	tsdbadmin
public	dw_stddev_week_aggregates	view	tsdbadmin
public	dw_stream_pilot_apartments	table	tsdbadmin
public	dw_sum_day_aggregates	view	tsdbadmin
public	dw_sum_hour_aggregates	view	tsdbadmin
public	dw_sum_week_aggregates	view	tsdbadmin

Figure 47: Table and aggregate views of the data warehouse

Figures Figure 48 and Figure 49 show examples of the content of such tables:

mtimestamp	apartmentid	roomid	sensorid	metric	mvalue
2019-01-01 00:10:00+00	881	8307	0	6	22
2019-01-01 00:10:00+00	881	8308	0	6	22.5
2019-01-01 00:10:00+00	881	8308	17523	1	22.71100607161679
2019-01-01 00:10:00+00	881	8308	19661	2	23.19163616001628
2019-01-01 00:10:00+00	881	0	17521	1	-6.812500000000001
2019-01-01 00:10:00+00	881	0	19662	2	98.00000000000001
2019-01-01 00:10:00+00	881	8307	19660	2	23.79252875052683
2019-01-01 00:10:00+00	881	8307	17522	1	22.75
2019-01-01 00:10:00+00	880	8304	0	6	23
2019-01-01 00:10:00+00	880	8305	0	6	22.5

Figure 48: Content of the raw data table of the data warehouse

mtimestamp	apartmentid	roomid	sensorid	metric	mvalue
2019-01-07 14:00:00+00	4	10	119	2	36.3656291702653533
2019-01-01 19:00:00+00	882	8311	0	6	22.0000000000000000
2019-01-01 02:00:00+00	1014	9520	20587	2	33.2661525172208650
2019-01-08 20:00:00+00	4	5	5	1	20.5944179409732967
2019-01-02 11:00:00+00	4	11	10	1	19.5727191366446517
2019-01-09 03:00:00+00	1014	9516	0	6	18.0000000000000000
2019-01-06 20:00:00+00	1014	9526	20593	2	57.8094791374276133
2019-01-07 12:00:00+00	1014	9523	0	6	17.0000000000000000
2019-01-07 21:00:00+00	4	9	300	2	40.3576273299596683
2019-01-01 16:00:00+00	4	7	302	2	43.5447104288394067

Figure 49: Content of one aggregate view (hourly aggregates) of the data warehouse

The fields in Figures Figure 48 and Figure 49 have the following meaning:

- **mtimestamp** is the timestamp in UTC at the beginning of the measured interval in the format yyyy-mm-dd HH:MM:SS
- **apartmentid** is an integer which identifies the apartment
- **roomid** is an integer which identifies the room
- **sensorid** is an integer which identifies the sensor device (a unique identifier in Optiwatti system)
- **metric** is an integer which identifies the measured variable
- **mvalue** is the actual value of the measurement

mtimestamp, apartmentid, roomid, sensorid and metric are the composite primary key.

The possible values of the field metric and their meaning is described in Table 18.

Table 18: Metric types in the periodic measurements of the data warehouse

Metric value	Meaning	Notes
1	Measured temperature by the sensor in the measured interval.	Records with roomid = 0 are related to outside temperature values.
2	Measured humidity by the sensor in the measured interval.	Records with roomid = 0 are related to outside humidity values
6	Set temperature by the user in Optiwatti configuration UI.	

5	Calculated electric power consumed in the measured interval for heating by electrical-based heating.	
121, 122, 123	Cumulative value in kWh of the energy coming from the grid for phases 1, 2 and 3.	Used for solar panels installations.
131, 132, 133	Cumulative value in kWh of the energy coming to the grid for phases 1, 2 and 3.	Used for solar panels installations.
141, 142, 143	Cumulative value in kWh of the energy coming to from solar panels production for phases 1, 2 and 3.	Used for solar panels installations.

Additionally, configuration data about the pilot apartments can be added to the data warehouse by copying filtered data from the control database.

This data is anonymized since no personal details are contained in the data warehouse.

A separate table contains the energy hourly sport prices in the time range from the beginning of year 2019 till the end of year 2023. Figure 50 shows an example of the content of such a table.

```
data_warehouse=> SELECT COUNT(*) FROM energy_spot_prices;
 count
--------
 135941
(1 row)

data_warehouse=> SELECT * FROM energy_spot_prices LIMIT 10;
       mtimestamp       | originaldb | originalid | area | unit | pricemwh
------------------------+------------+------------+------+------+----------
 2018-12-31 00:00:00+00 |          3 |     132917 | FI   | EUR  |    49.13
 2018-12-31 01:00:00+00 |          3 |     132918 | FI   | EUR  |    48.28
 2018-12-31 02:00:00+00 |          3 |     132919 | FI   | EUR  |    45.92
 2018-12-31 03:00:00+00 |          3 |     132920 | FI   | EUR  |    45.47
 2018-12-31 04:00:00+00 |          3 |     132921 | FI   | EUR  |    47.11
 2018-12-31 05:00:00+00 |          3 |     132922 | FI   | EUR  |    49.12
 2018-12-31 06:00:00+00 |          3 |     132923 | FI   | EUR  |     49.8
 2018-12-31 07:00:00+00 |          3 |     132924 | FI   | EUR  |    49.74
 2018-12-31 08:00:00+00 |          3 |     132925 | FI   | EUR  |    50.06
 2018-12-31 09:00:00+00 |          3 |     132926 | FI   | EUR  |    50.76
(10 rows)
```

Figure 50: Number of entries and example of the content of the energy spot price table.

Each entry consists of the following **fields:**

- **mtimestamp** is the timestamp in UTC at the beginning of the measured interval in the format yyyy-mm-dd HH:MM:SS
- **originaldb** is an integer that uniquely identifies the source database in Optiwatti (it is added by the data pipeline that is used to copy the data in the data warehouse and it might be useful for trouble shooting reasons)
- **originalid** is an integer that uniquely identifies the record in the source database (the primary key of the MySQL source database)
- **area** is a text variable which logically is a categorical variable which identifies the geographical area
- **unit** is a text variable which logically is a categorical variable which gives the currency
- **pricemwh** is the actual value of the measurement, that is, the price per MWh

Eight aggregate views have been created for the energy prices. The names of these views are outlined in Figure 51. There are both daily and weekly aggregates for the following statistics: average, standard deviation, maximum and minimum. The fields of such views are the same as the fields of the originating energy_spot_prices table except that the fields originaldb and originalid are not included in the aggregate views.

```
data_warehouse=> \d
                       List of relations
 Schema |                 Name                 | Type  |  Owner
--------+--------------------------------------+-------+----------
 public | dw_avg_day_aggregates                | view  | tsdbadmin
 public | dw_avg_hour_aggregates               | view  | tsdbadmin
 public | dw_avg_week_aggregates               | view  | tsdbadmin
 public | dw_count_day_aggregates              | view  | tsdbadmin
 public | dw_count_hour_aggregates             | view  | tsdbadmin
 public | dw_count_week_aggregates             | view  | tsdbadmin
 public | dw_max_day_aggregates                | view  | tsdbadmin
 public | dw_max_hour_aggregates               | view  | tsdbadmin
 public | dw_max_week_aggregates               | view  | tsdbadmin
 public | dw_min_day_aggregates                | view  | tsdbadmin
 public | dw_min_hour_aggregates               | view  | tsdbadmin
 public | dw_min_week_aggregates               | view  | tsdbadmin
 public | dw_stddev_day_aggregates             | view  | tsdbadmin
 public | dw_stddev_hour_aggregates            | view  | tsdbadmin
 public | dw_stddev_week_aggregates            | view  | tsdbadmin
 public | dw_stream_pilot_apartments           | table | tsdbadmin
 public | dw_sum_day_aggregates                | view  | tsdbadmin
 public | dw_sum_hour_aggregates               | view  | tsdbadmin
 public | dw_sum_week_aggregates               | view  | tsdbadmin
 public | energy_spot_prices                   | table | tsdbadmin
 public | energy_spot_prices_day_avg           | view  | tsdbadmin
 public | energy_spot_prices_day_max           | view  | tsdbadmin
 public | energy_spot_prices_day_min           | view  | tsdbadmin
 public | energy_spot_prices_day_stddev        | view  | tsdbadmin
 public | energy_spot_prices_week_avg          | view  | tsdbadmin
 public | energy_spot_prices_week_max          | view  | tsdbadmin
 public | energy_spot_prices_week_min          | view  | tsdbadmin
 public | energy_spot_prices_week_stddev       | view  | tsdbadmin
(28 rows)
```

Figure 51: Aggregate views for the energy spot prices.

As part of the data warehouse, **data marts** can be created, containing data that can be shared within the Finnish pilot partners of the STREAM project.

See also section 3.1.3.4 "Open APIs" for further information about the external API of this data warehouse.

3.1.3.4 Open APIs

sFLEX – sDATA interface

The sFLEX-sDATA interface is an internal Optiwatti interface.

Connection to Fingrid

The connection to Fingrid is an SSL encrypted connection to a selected port that uses Active MQ or RabbitMQ (to be still decided).

Data exchange between Optiwatti and VTT

VTT will have access to a dedicated database server containing selected, filtered and anonymized data. The connection to the database will be an SSL encrypted connection to a selected port. The server will have dedicated access credentials and user profiles with dedicated passwords will be used to access the database. The server will be outside Optiwatti private cloud. The server will be maintained as long as there is budget from the STREAM project.

3.1.3.5 sDATA compliance with Open Data Spaces

This section describes the compliance of Finnish pilot to the open data space principles.

- Anonymization of personal data
  - Settings are contained in relational databases. This ensures that personal data is contained in a limited number of tables. Apartments are identified by an id in all the other tables. The contents of the tables containing personal data is not copied to the data warehouse used for the STREAM project. This ensures that the data that is shared with VTT and with the TSO is anonymized.
- Security and data sovereignty, trust
  - Security measures such as authentication and authorization are an integral part of sDATA. Data marts are created to contain data that can be shared with VTT. This data is contained in a server with dedicated credentials. Each participants have own login access credentials. Both the data exchange with the TSO and the connections with VTT are SSL encrypted connections to selected ports.
- Ecosystem of data and decentralized data architecture
  - Data is stored in the private cloud by Optiwatti. The data that can and will be shared is stored also in dedicated databases and server in sDATA outside Optiwatti private cloud. Then data can be transferred to the relevant stakeholders.
- Standardized interoperability
  - The data exchange is defined and specified in the Open APIs section. Access to data is allowed by means of a widely used encryption method. Data is stored in a SQL-friendly database which uses one of the most common SQL dialects.
- Value added apps
  - Data is checked before being copied in the data marts and some basic data cleansing is done.
- Data markets
  - All the data that is shared with VTT is shared free of charge. Note that the server used for sharing data with VTT is a temporary server and will be maintained as long as there is budget from the STREAM project. Data that is shared with the TSO is shared with no additional charges on top of what is agreed in the normal bidding process for the FCR services.

3.2 SPANISH PILOT

The present section outlines the specifications of the data exchange processes among the tools and systems in the Spanish pilot of STREAM. These specifications are based on the requirements described in previous sections.

3.2.1 Functional specification

The functional specifications of the data exchange services in the Spanish pilot site are presented in Table 19.

Table 19: Functional specification of data exchange in the ES pilot

Functionality	Description
Data ingestion	The services for data ingestion are interoperable, thus allowing the use of different communication protocols for third parties to send the data captured by their systems. Specifically, the following protocols are supported: • MQTT [28]: Message Queuing Telemetry Transport, an open, simple, and lightweight client-server publish-subscribe messaging transport protocol ideal for communication in Machine-to-Machine (M2M) and Internet of Things (IoT) contexts where a small code footprint and/or low network bandwidth are required. The service supports authentication and establishment of Quality-of-Service (QoS) as defined by the protocol. • AMQP [29]: Advanced Message Queuing Protocol, an open standard application layer protocol also based on the publish-subscribe pattern that separates the network transport from broker architectures and management. Its defining features are message orientation, routing, and queuing, as well as reliability and security. The service supports authentication and management of the different types of elements of the protocol (brokers, queues, bindings, etc.) • HTTP [30]: HyperText Transfer Protocol, a stateless, connectionless, and media-independent application-level protocol for distributed, collaborative, hypermedia information exchange. Its secure extension (HTTPS) provides encryption for secure communication. An Application Programming Interface (API) based on the Representational State Transfer (REST) architecture is provided on top for third parties to send new data to the tools. The API is secured in order to avoid unauthorised use.
Data storage	The services for data storage use two different repositories for different purposes: • An initial repository is used to maintain the most recent version of each element, including its metadata (name, properties, geolocation, etc.) and a copy of its last received set of measurements. This repository is based on MongoDB [31], a NoSQL database managing collections of documents in JSON format. • A secondary repository is used to store a copy of each updated measurement from each selected element in the system, in order to have a registry of the historical evolution of all data. This repository is based on InfluxDB [32], an open-source database optimized for fast, high-availability storage and retrieval of time series data.

Data retrieval and integration	The services for data retrieval depend on the repository to be accessed: • The real-time repository can be accessed via WebSocket, with a public-subscribe pattern. The client subscribes to one or more collections of elements, filtering by the selected criteria, and the server forwards to them any update over the matched documents. The Distributed Data Protocol (DDP) [33] as provided by the Meteor [34] JavaScript framework, is used for this purpose. The procedure is secured with credentials and the domains of the data are limited per user. • The historical repository can be accessed via REST API. The service allows to select the specific collection, entities, and metrics to be retrieved, as well as filtering per time span. Additionally, the time precision and integration function to use to format the results can be specified as well. The service is secured with credentials to avoid unauthorised access to data.
Data pre-processing	The data ingestion service allows to define specific rules to accept or discard an incoming piece of data based on its values. These rules can also be used to generate alarms when a received value or values fulfil the established condition, thus alerting the operator and/or a dedicated service about an abnormal operation status.
Data normalization	Although the technologies used by the data management services are agnostic in terms of normalization, the different wrappers that interface the systems and field devices take into account these specifications in order to ensure a coherent set of data for the STREAM tools to use.

3.2.2 Non-functional specification

The non-functional specifications of the Spanish pilot's data exchange services are defined in Table 20.

Table 20: Non-functional specification of data exchange in the ES pilot

Functionality	Description	Type
Retention policies	The historical repository is set up with a retention policy of 2 years for raw values and 5 years for aggregated 15-minute statistics.	Security and Privacy
Data access protection	The access to both data retrieval services (WebSockets and REST API) is protected with credentials. Human users are provided with a username and a password, while applications and services use an app ID and a token. Each set of credentials grants access to a subdomain of the data, so that each user can only access the data they own or have been allowed to. Each set of credentials can be modified and/or revoked at any time by the system administrator.	Security and Privacy
Data anonymisation	Sensitive data, especially that related to customers and members of the EnC, is anonymised in order to ensure privacy and prevent misuse. Personal names are replaced by generic ones (e.g. "Community member 1"), while exact addresses are replaced with more generic information, such as street, area, or postal code.	Security and Privacy

Data encryption	All ingestion and data access mechanisms (MQTT, AMQP, HTTP) provide encryption mechanisms over Transport Layer Security (TLS [35]) to ensure the security of the communications and prevent misuse by unauthorized parties.	Security and Privacy
Data ingestion format	All data ingestion methods expect payloads in JSON format, which makes their storage in the MongoDB repository seamless. For any other format of source data (e.g. XML), it is encouraged to create a service wrapper that transforms the information into JSON before sending it to the system.	Performance
Data ingestion message size	The platform used for MQTT and AMQP ingestion is RabbitMQ, which allows a maximum payload size of 512 MiB [36]. The HTTP protocol does not specify a limit for the body of the messages, but based on the storage technologies, the maximum document size in MongoDB 6.0 is 16 MB [37], while the default (not maximum) body size of an HTTP request to InfluxDB is 25 MB [38].	Performance
Connectivity settings	The services for data ingestion are deployed as a microservice architecture in a Kubernetes cluster, which contractually assures the availability of the infrastructure via Service Level Agreement (SLA). Moreover, it provides automated load balancing across different nodes and replicas of each microservice, which increments their availability.	Connectivity
Interoperability settings	The communication between the tools and third parties can be performed through different methods and communication protocols, which enhances the interoperability with equipment from different vendors.	Interoperability
Scalability and throughput	The services for data management in the STREAM Spanish pilot are deployed as a microservices in a Kubernetes cluster, which are easy to scale up and replicate. Database technologies support both replication (maintaining copies of the data in different machines) and sharding (distributing the data across different machines), useful techniques for scaling up the system by several orders of magnitude. Automated load balancing across nodes and replicas is provided, thus incrementing the performance.	Scalability, Performance

3.2.3 System specification

3.2.3.1 Architecture

The data management system in the STREAM Spanish pilot consists of a series of connected microservices, each of them dedicated to performing a specific task. The architecture is based on a platform developed by ETRA (known as CITRIC), built to cover the communication necessities in previous H2020 projects such as X-FLEX [39], ASSISTANCE [40], SAURON [41], and MATCHUP [42].

CITRIC is a Reactive, Interoperable, Visible, Elastic and Resilient architecture (RIVER) oriented to microservices and events. It is an open architecture with capacity to grow its service network, reactive because it is event-driven, interoperable because it is supported by standard protocols and agnostic models of data, visible because it is monitored in its operation, elastic because it can be scaled out and independently in each of its services to support any magnitude of city, and resilient because it is orchestrated and monitored to be fault tolerant. CITRIC's microservices distribution fully complies with norms IEC30182:2017 [43] and UNE178104:2017 [44] in its orientation towards functional layer.

The full architecture of CITRIC is presented in Figure 52. The functionalities used in STREAM are marked in the Figure and briefly described afterwards.

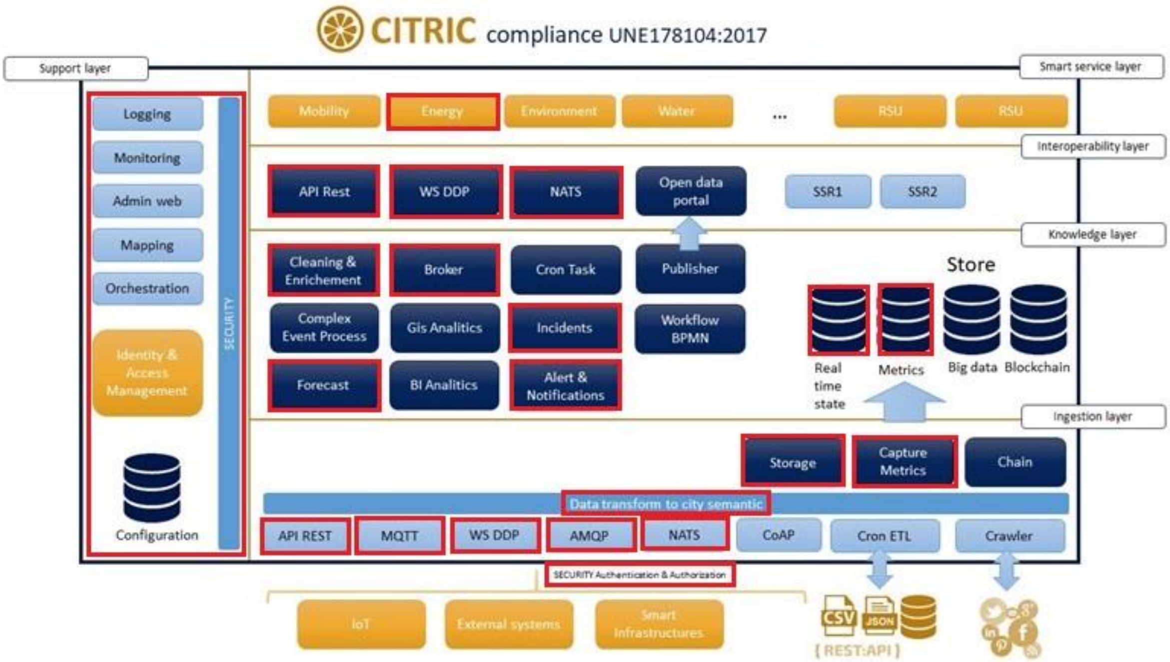

Figure 52: CITRIC architecture

- <u>**Support layer**</u>
  - **Orchestration:** The orchestration of all components is performed by Kubernetes [45] allowing flexible and monitored deployment.
  - **Monitoring:** The CITRIC management website allows to monitor all the microservices graphically, as well as the activity of the data flowing through the platform.
  - **Logging:** The entire logging system is handled by the orchestrator and stored for forensic analysis of errors and alerts.
  - **Admin web:** CITRIC offers a multilingual web interface for management and monitoring with presentations of metrics, mapping, incidents, data, multimedia, etc.
  - **Mapping:** A 3D map is available in the management interface to navigate platform data. This map allows you to make analytic representations of the data such as aggregations of means, maximums, deviations, etc., apart from the geolocated representation of the elements. The visualizer is optimized for bulk rendering of web data using WebGL [46] technology.
  - **Identification and access management:** CITRIC has a component for authentication, authorization, and profiling that allows to determine how the platform is accessed under particular credentials. It uses basic authentication, as well as SSO protocols OpenID [47], supported by a Keycloak [48] backend.
  - **Configuration:** All CITRIC configuration resides in a NoSQL database, particularly MongoDB. It stores all the information necessary for the operation of all its components.
- <u>**Ingestion layer**</u>

- **API REST:** CITRIC has a robust secure and protected HTTPS Rest API with rate limit control to prevent attacks and allows a lot of flexibility for data ingestion through it. Besides authentication, API security allows to authorize only a certain set of data to each authenticated user. The implementation of this API has been done with Express [49].
- **MQTT:** The MQTT protocol is widely used by lightweight devices (IoT) and supported by CITRIC. For this purpose, CITRIC uses RabbitMQ as the MQTT ingestion service.
- **AMQP:** It is possible to ingest data directly with AMQP protocol to ingest directly over queues that are consumed by microservices, which transform and store information in the platform storage system.
- **WS DDP:** It is possible to connect to the platform using WebSockets (e.g., from web browsers). The protocol over WS implemented is Distributed Data Protocol (DDP). A number of methods are implemented over this protocol that provide a very efficient way to do mass ingestion.
- **NATS:** NATS [50] is the underlying broker on the platform and is responsible for managing the entire microservices communication network. It acts as an enterprise service bus (ESB); communication through it makes use of topics that are mapped to services. CITRIC offers a set of topics that allow to interact directly with storage microservices with their corresponding level of security.
- **Transform:** Any ingestion process previously goes through a transformation process to normalize the data before entering it into the platform. Transformation schemas are pre-configured for each source in the configuration database and are particular to each platform deployment as they must be tailored to particular data sources and how they are stored in the storage service and then served to the higher layers.
- **Security:** Every ingestion process is authenticated and authorized by a layer of security in each microservice. Each credential per token or user/password has an authorization scheme to access a subset of platform data at three levels of security: read, write, and only public attributes.
- **Storage:** This microservice handles the storage of data when it is injected or modified. It manages the real-time database supported by MongoDB.
- **Metrics:** This microservice extracts the precise information to create KPIs from each type of modelled data. The information is stored in a time-oriented database such as InfluxDB.

- <u>Knowledge layer</u>
  - **Cleaning & Enrichment:** ETL batch process for cleaning and enriching the information stored in the big data repository.
  - **Forecast:** This microservice is responsible for generating Machine Learning models on data collections stored in the storage repository. Makes use of the XGBoost [51] library. This microservice creates models and makes predictions about data series. It offers NATS interaction to integrate with the entire CITRIC ecosystem and a Rest API.
  - **Broker:** The entire microservices architecture is supported by a robust and scalable messaging broker such as NATS. This key, fault-tolerant and redundant element is the one that sorts the entire service network on the platform.
  - **Incidents:** This microservice is responsible for observing incidents that occur in the installation; it has a complex notification system with different means such as Twitter, Telegram, Email, external systems by web services, or triggering of workflow processes that assist in the development of the incident.

- Alert and Notifications: Alerting and reporting events in the installation is done by various means. Like incidents, notifications are integrated with different means such as Twitter, Telegram, Email, external systems by web services, or triggering of workflow processes.

- **Interoperability layer**
    - **API Rest:** CITRIC has a portion of its REST API for CRUD operations that verticals can use to interact with the platform, with its corresponding level of security. This API is generally used by back-end services from the STREAM tools to retrieve data stored in the platform.
    - **WS DDP:** In the same way, customers can make use of WebSockets to interact with the platform. This service is generally used by front-end applications, as WebSockets are the only way to interact with external systems with bidirectional flow.
- **Smart service layer**
    - Includes any vertical service or application using the platform and its services. These external agents are usually grouped by its field of knowledge (Energy, in the case of STREAM), which defines the type of data (entities, metrics, etc.) the applications will use.

3.2.3.2 Open APIs

The services of the RESTful API as exposed by the CITRIC ecosystem is provided in Table 21. Authentication via login method is only required for regular users; applications must use the provided *userId* and *authToken* (there is no need for them to perform a login call).

Table 21: Open APIs for data exchange in the ES pilot

Path	Method	Description
/api/v2/health	GET	Path to check the service availability
/api/v2/login	POST	Authentication via user (ID, name or email) and password
/api/v2/:layer/:col	GET	Query documents from a collection
	POST	Insert one or multiple documents
	PATCH	Update one or multiple documents
	DELETE	Delete all the documents
/api/v2/:layer/:col/:id	GET	Get a single document
	PATCH	Update a document
	DELETE	Delete a document
/api/v2/config/:layer/:item	GET	Get the configuration of a collection
	PUT	Update the configuration of a collection
/api/v2/range/:layer/:col	GET	Query documents based on a range of dates
/api/v2/near/:layer/:col	GET	Query documents near a geographic point
/api/v2/gateway	**POST**	**Provides access to the additional microservices via NATS broker. This endpoint is used to access the long-term repository (InfluxDB).**

3.2.3.3 sDATA compliance with Open Data Spaces

Since sDATA is not deployed at the Spanish pilot site, and therefore there is no Data Space integrated, the pilot does not contemplate the Open Data Spaces principles and requirements on its data exchange services.

3.3 ITALIAN AND SLOVENIAN PILOT

IND developed the sDATA solution to support both the IT and SI pilot projects simultaneously. As a result, the specification provided applies to both pilots, demonstrating the versatility and adaptability of sDATA. This approach highlights how the same solution can be effectively implemented across two different pilot sites, each with distinct UCs, underscoring the interoperability and open nature of the sDATA framework.

Functional specifications (3.3.1) are given for sDATA platform, focusing on the Master Data Management (MDM) platform, which includes a user interface (UI). Other specifics, describing the data exchange and transformation processes are defined in non-functional specification section (3.3.2).

3.3.1 Functional specification

The functional specifications of sDATA in the IT and SI pilots are listed in Table 22.

Table 22: Functional specification of sDATA in IT and SI pilot

<table>
<tr><th>Functionality</th><th>Description</th></tr>
<tr><td>Login</td><td>MDM UI includes a secure and user-friendly login for registered users as shown in Figure 53. Users enter a valid login credentials, such as email address and password to access their accounts. The login system also features password recovery.

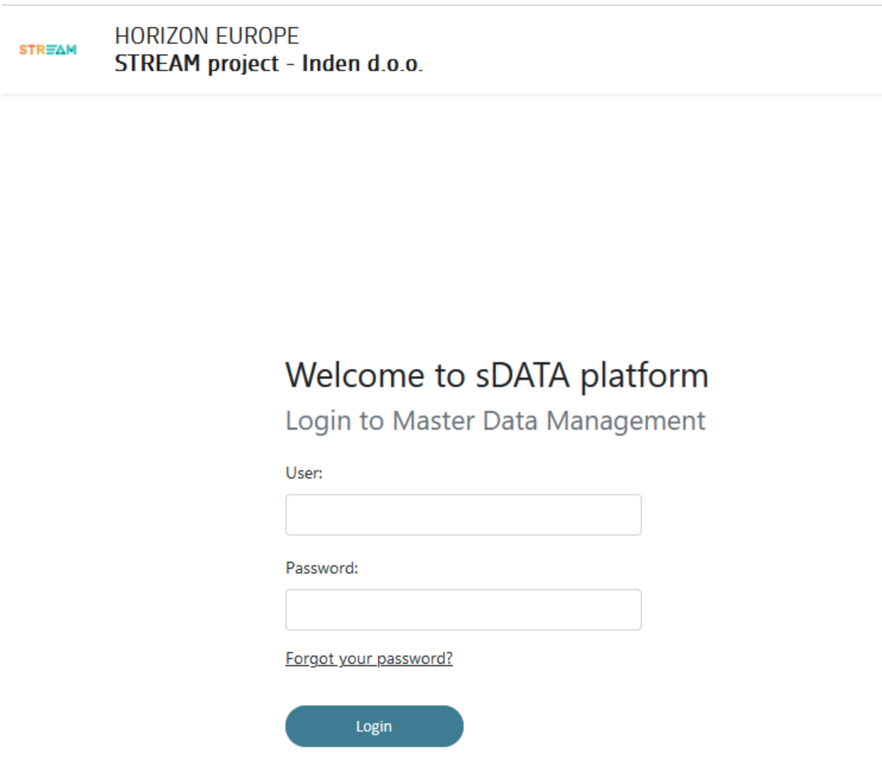

Figure 53: MDM login page</td></tr>
<tr><td>MDM structure</td><td>MDM system comprises three main components: Administration, Slovenian pilot, and Italian pilot as shown in Figure 54.

The Administration component is responsible for managing user records and user rights documentation and settings.</td></tr>
</table>

The Slovenian pilot component includes records of the EV charging stations, Power meter measurement points, SCADA measurement points and Elektro Primorska network model in CIM format.

The Italian pilot component contains records of the EV charging stations and Measurement Devices and Measurements.

Each component of the MDM contains master data relevant to its respective entities.

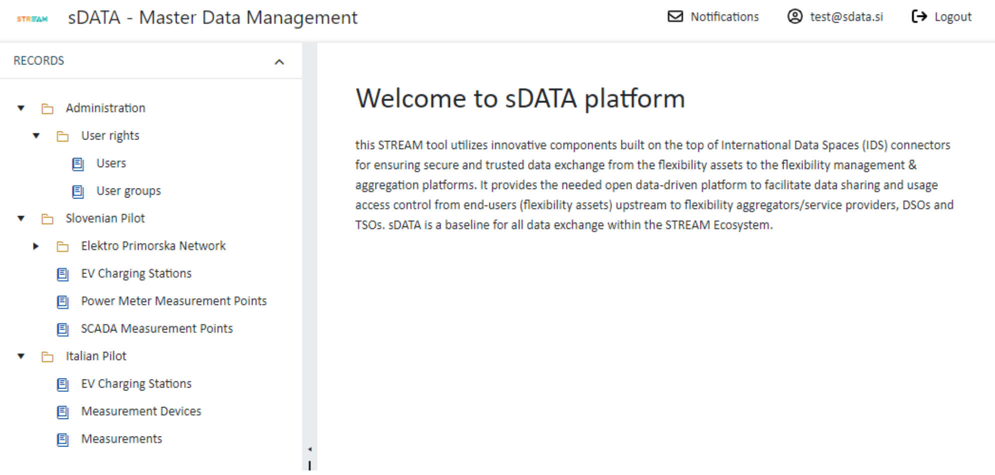

Figure 54: sDATA – MDM platform

MDM general functionalities

- **Main grid view**: When opening a record system, a grid displaying the master data for all records within that system is shown as the main view. Clicking on any individual record in the grid reveals the detailed information for that specific record. An example of the main grid view is shown in Figure 55.

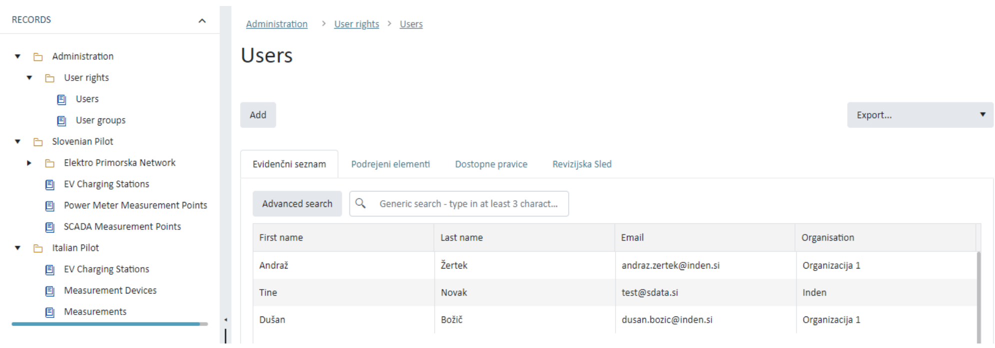

Figure 55: User record system main grid view

- **Master data export**: A list of all records within the system can be exported to a CSV file, which includes all the information displayed in the main grid view.
- **Record actions**: A record can be added, updated, or truncated by a user with Editor access rights. A user with Viewer rights can only view the record.
- **Audit trail**: Each element of the MDM features an Audit trail tab, which lists all actions performed on that element. This includes actions such as creation, detailed edits with descriptions of changed attributes (old vs. new values), and views of the element.

- **Generic search:** A generic search functionality enables users to search for records in the main grid view by matching column values with the search string. The search results are then displayed in the main grid.
- **Advanced search**: The platform provides advanced search functionality which allows users to find specific records based on search options, e.g. filtering by data values. The search results are then displayed in the main grid. An example of the advanced search is shown in Figure 56.

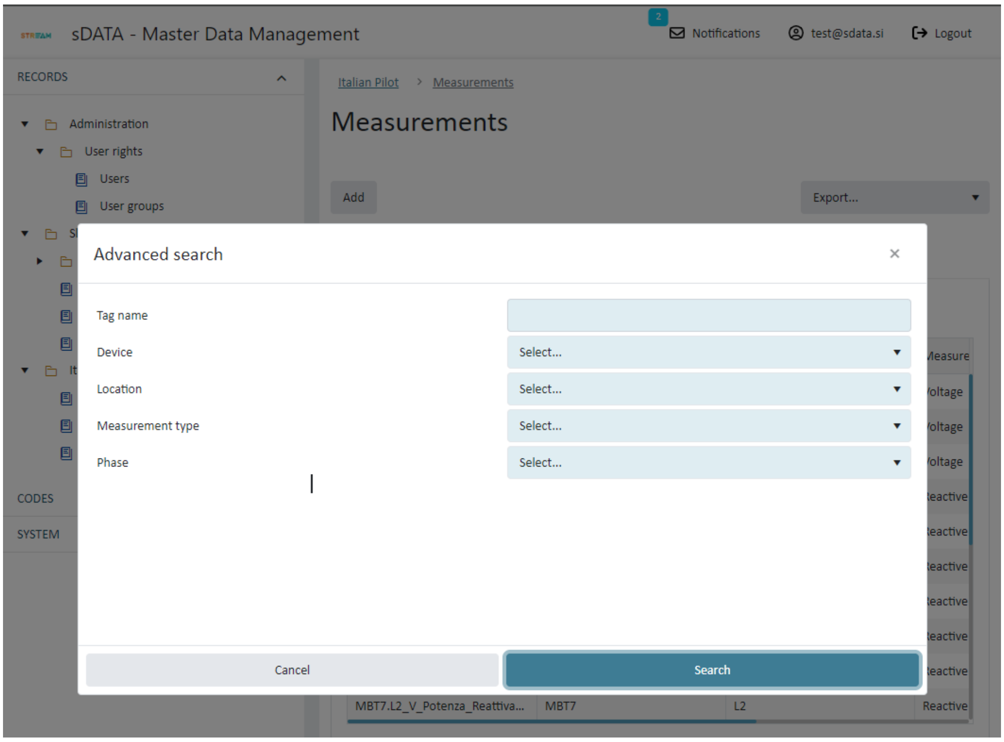

Figure 56: Advanced search window

User and User right settings

For each MDM user, a user record must be added to the user record system. Users can be added, edited, or removed from the system. Each user record includes basic user information such as:

- First name,
- Last name,
- Email,
- Organization.

Apart from that, each user record also includes the user rights. Users can be assigned different rights, such as Editor or Viewer (or none), for each record system (e.g., EV Charging Stations).

User groups consist of selected users and have access rights defined similarly to individual users.

For each record system, a user group can have Reader or Editor (or none) rights.

Each user can be assigned to one or more groups, inheriting all access rights granted to those groups.

If record systems are nested, access to a parent record system automatically grants access to all its child record systems.

Users can check the assigned access rights for any element of the MDM.

An example of a user record is shown in the following figures (Figure 57, Figure 58 and Figure 59).

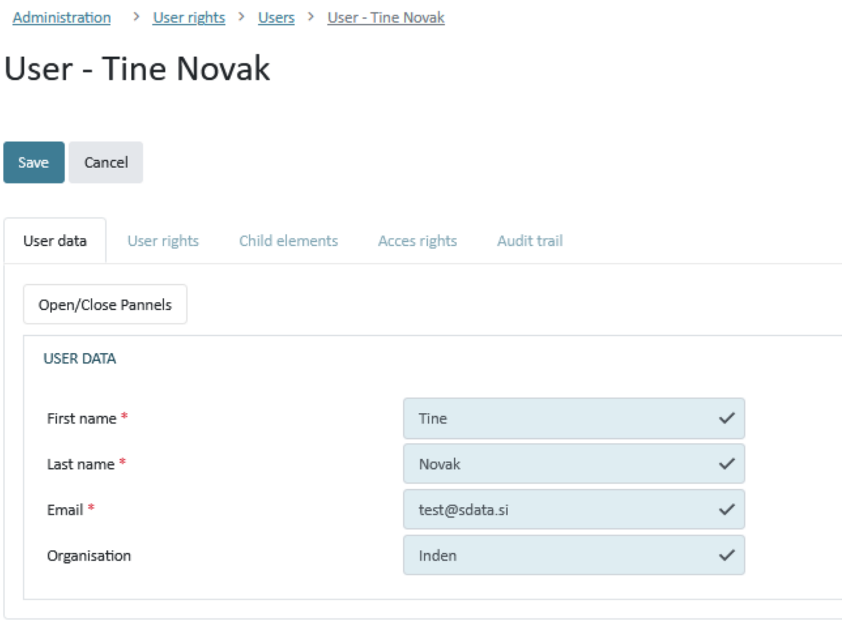

Figure 57: User master data

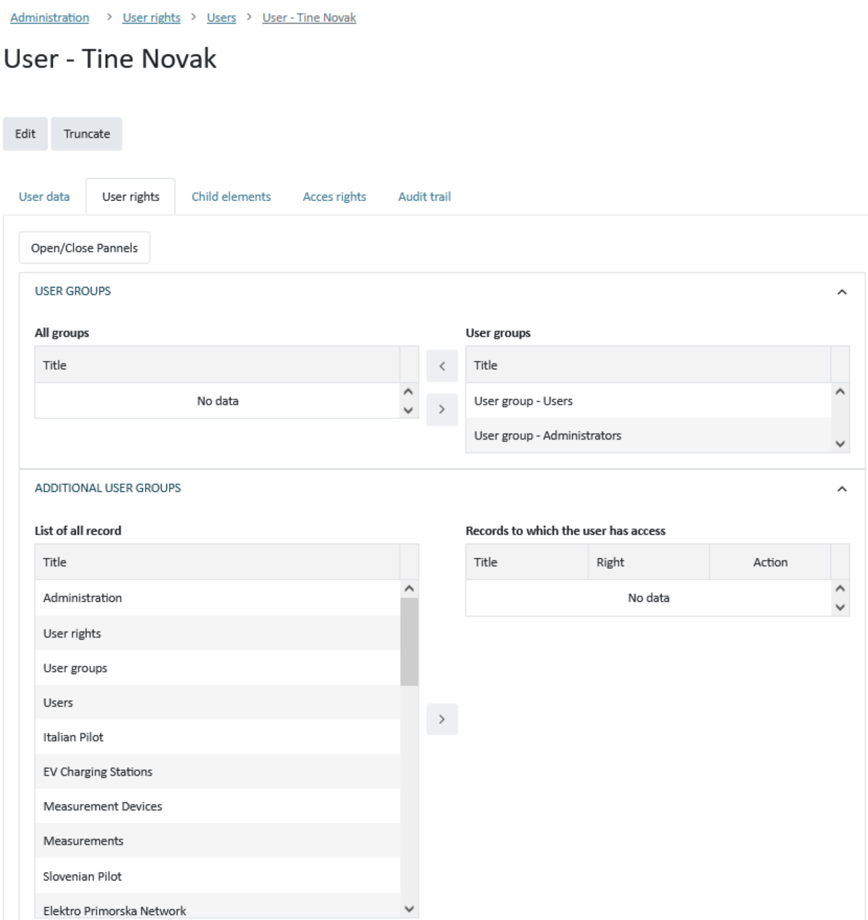

Figure 58: User access rights

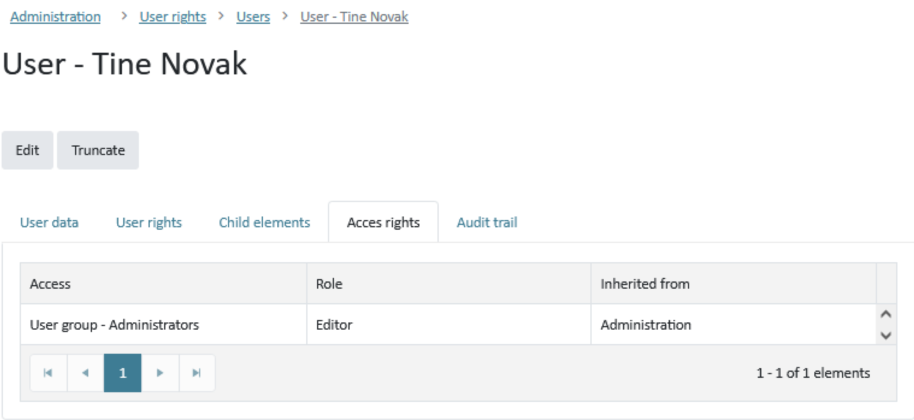

Figure 59: Record access rights

Slovenian pilot Master Data

The Slovenian pilot Master Data part includes records of the:

- EV charging stations,
- Power meter measurement points,
- SCADA measurement points,
- Elektro Primorska network model in CIM format.

EV charging stations records hold master data of the Avantcar charging stations in the Slovenian pilot site. The master data covers basic information about the EV charging station, such as:

- Id,
- Name,
- A list of charging plugs; where each charging plug is described by master data, such as Id, current type, max power, max voltage, max amperage and Link to chart.

The Link to chart property includes URL of the Grafana dashboard, which opens in a new tab of the browser, when clicking on it. The dashboard displays actual measurement data of the plug.

An example of EV charging station record is shown in Figure 60.

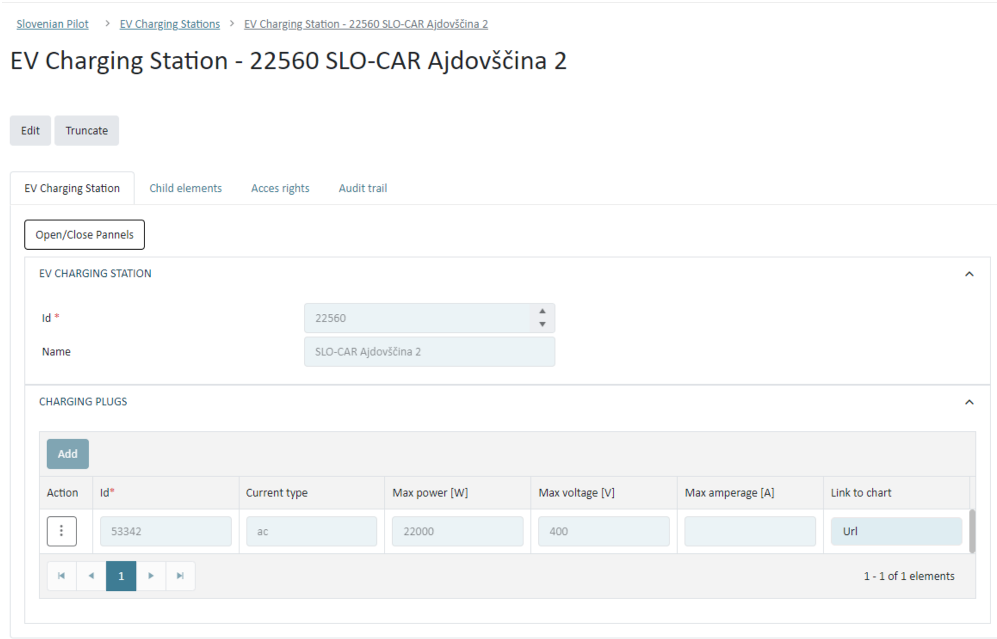

Figure 60: EV charging station record

Power meter measurement points records hold master data of the power meter measurement points in Slovenian site (Elektro Primorska) power network. The master data covers basic information about the power meter measurement point, such as:

- Id,
- Description,
- Primary element Id,
- Secondary element Id,
- Address,
- Production number,
- Voltage level,
- Rated power,
- Link to chart.

An example of Power meter measurement point record is shown in Figure 61.

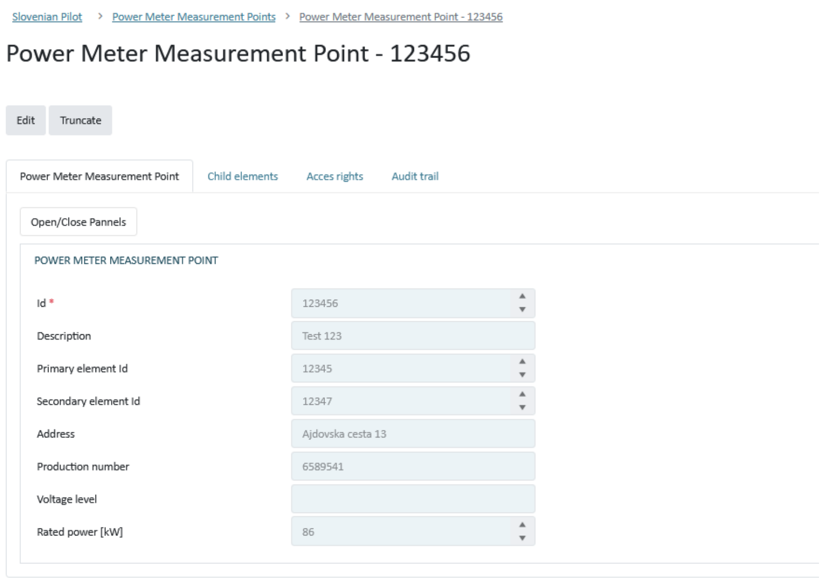

Figure 61: Power Meter Measurement Point record

SCADA measurement points records hold master data of the SCADA measurement points in Slovenian site (Elektro Primorska) power network. The master data covers basic information about the SCADA measurement points, such as:

- Id,
- Name,
- Description,
- Type,
- Unit.

An example of SCADA measurement point record is shown in Figure 62.

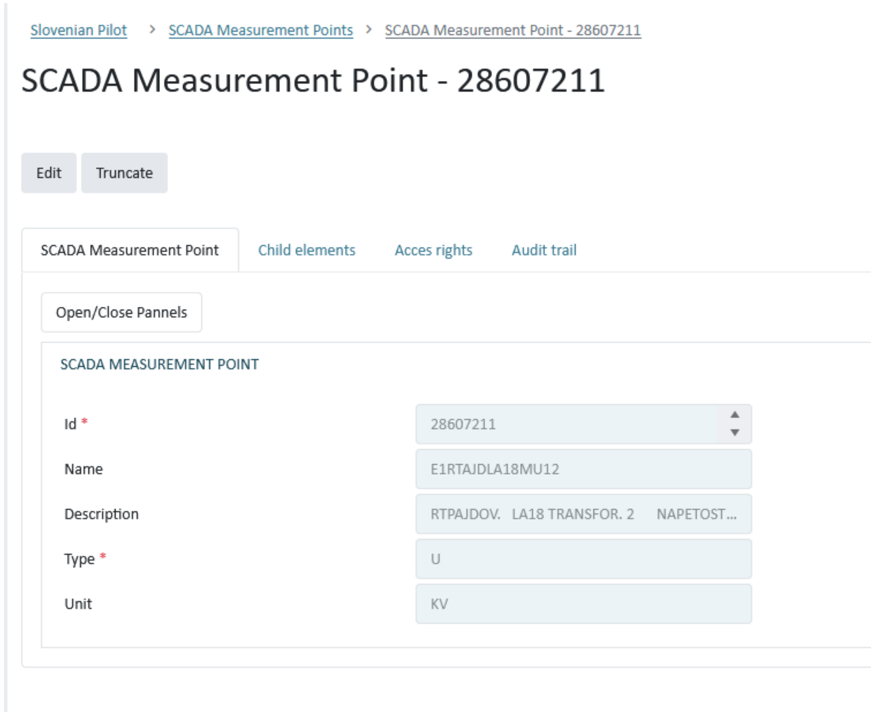

Figure 62: SCADA Measurement Point record

The Elektro Primorska network model is represented in a CIM format and is designed as a multi-record system for various types of entities within the model, such as:

- Breaker,
- Terminal,
- LinearShuntCompensator,
- SeriesCompensator,
- GeographicalRegion,
- SubGeographicalRegion, etc.

Each of these record groups contains multiple records of network elements, each described by its own set of parameters.

An example of Elektro Primorska network element record is shown in Figure 63.

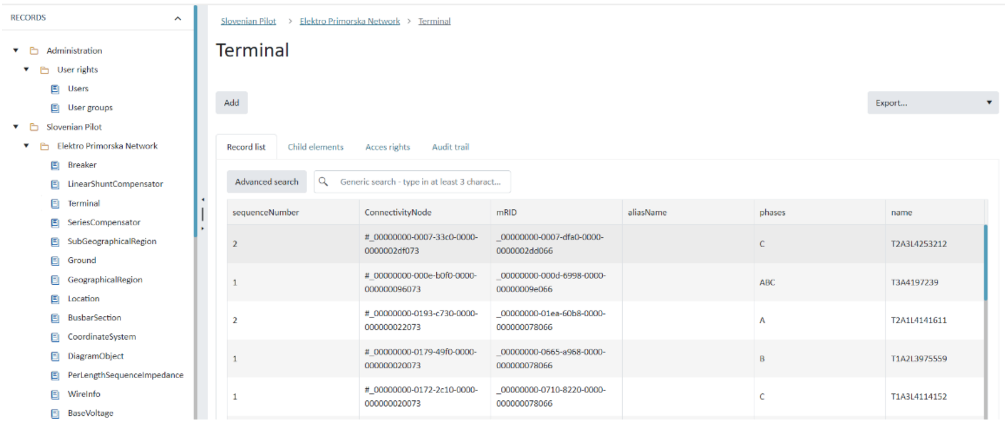

Figure 63: Elektro Primorska network model in MDM

Italian pilot Master Data

The Italian pilot Master Data part includes records of the:

- EV charging stations,
- Measurement devices,
- Measurements.

EV charging stations records hold master data of the Emotion charging stations in the Italian pilot site. The master data covers basic information about the EV charging station, such as:

- Id,
- Is active,
- Is public,
- Max power,
- AC max power,
- DC max power,
- Operator,
- Address,
- Coordinates,
- Location on map,
- Set of Charging plugs, where each charging plug is described with Id, Type, Status, Kwh price, Minute price, information whether the plug Is manually disabled or not.

Users can open Google maps directly from the record, clicking on the Location on map. The location of the charging station is displayed in a new tab of the browser, in Google maps as shown in Figure 65.

An example of EV charging station record is shown in Figure 64.

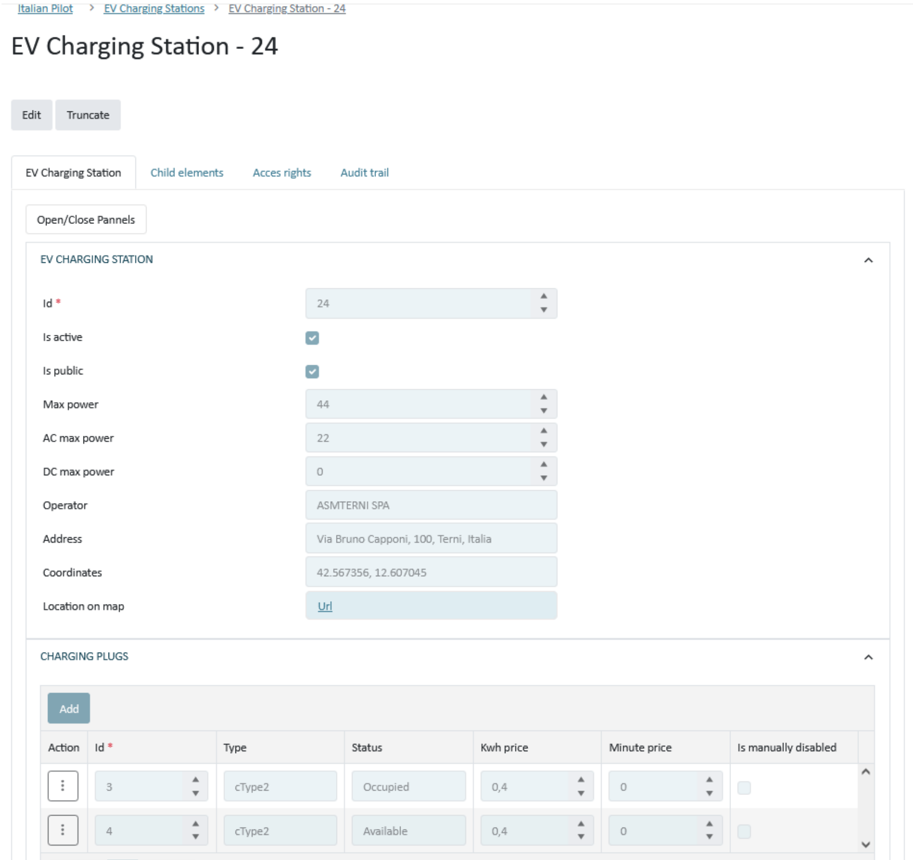

Figure 64: EV charging station record

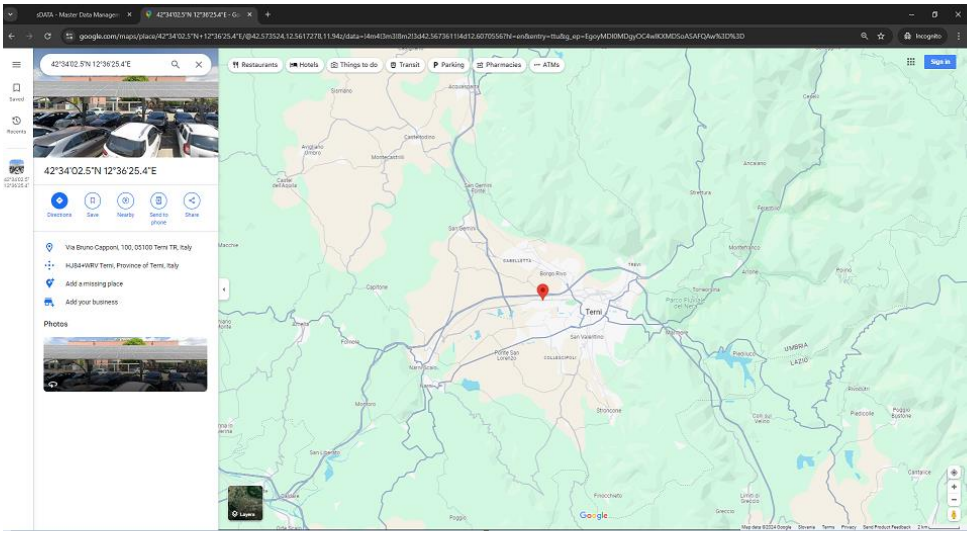

Figure 65: Display of the location in the Google maps, directly from MDM portal

Measurement devices records represent a set of measurement devices in the power system (of ASM) in the IT pilot for the purpose of the STREAM project. The records include following attributes:

- Id,
- Consumption P [kw],
- Production P [kW].

An example of Measurement device record is shown in Figure 66.

Figure 66: Measurement device record

Measurements records hold master data of the measurement quantities, obtained by the measurement devices in the Italian pilot. The master data is represented by a list of the following attributes:

- Tag name,
- Device,
- Location,
- Measurement type,
- Phase,
- Link to chart.

The device is directly linked to a record of Measurement device in the Measurement Devices record system.

An example of Measurement record is shown in Figure 67.

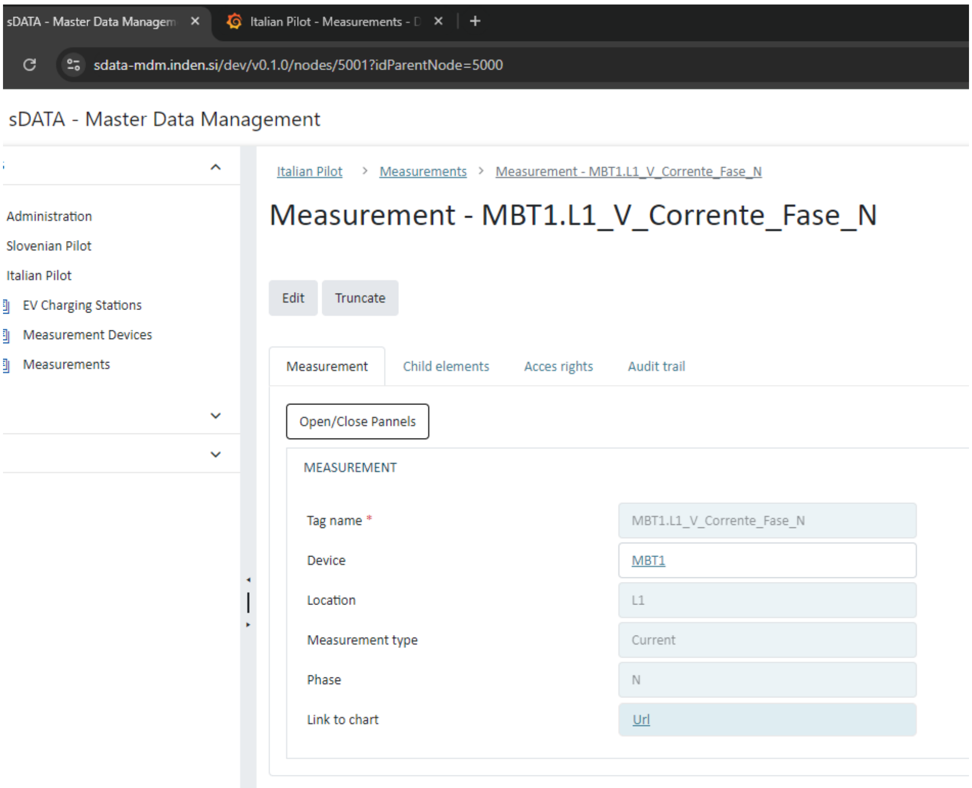

Figure 67: Measurement record

The Link to chart property includes URL of the Grafana dashboard, which opens in a new tab of the browser, when clicking on it. The dashboard displays actual measurement data of this measurement quantity, as shown in Figure 68:

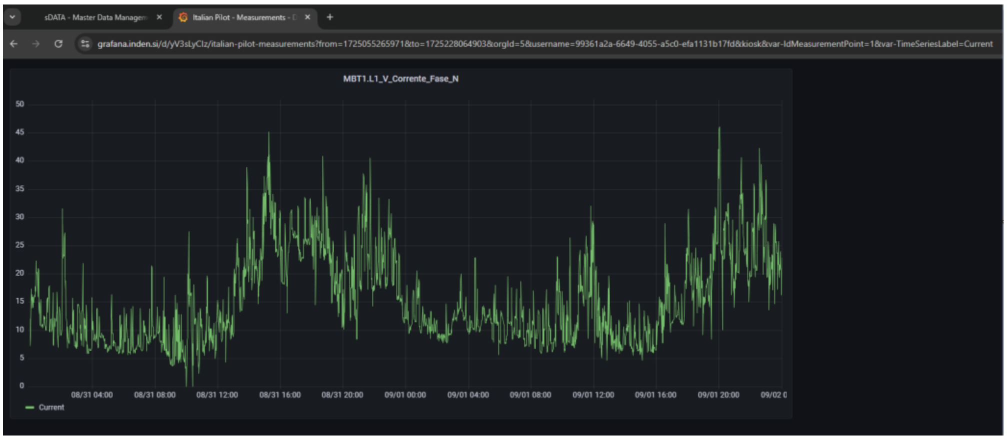

Figure 68: Example of simple Grafana dashboard for measurement

Data processing

The sDATA platform also supports robust data processing to ensure data quality and uniform distribution. Once data is retrieved from the source, it undergoes cleansing and normalization, as exemplified by the power grid measurement data in the Italian pilot. The data cleansing process is predefined, primarily involving methods like filling in missing values with the last known value, interpolation, or using values from similar days. After processing, the cleansed data is saved into the database, ensuring it is ready for analysis and further use.

3.3.2 Non-functional specification

The non-functional specifications of sDATA in the IT and SI pilots are listed in Table 23*Table 22*.

Table 23: Non-functional specification of sDATA in IT and SI pilot

Functionality	Description	Type
User authentication and authorization	sDATA includes a secure and user-friendly login for registered users. Users enter a valid login credentials, such as email address and password to access their accounts. This mechanism verifies user identities and control access to data. Security features, such as API keys and login credentials (username and password), are implemented at every data exchange endpoint to ensure that only trusted users have access to the data.	• Security • Privacy
Record of access and data exchange requests	All access events are recorded in the distributed ledger, ensuring a single source of truth and a complete, transparent history of user actions. This log is accessible to the system administrator for review and analysis. A more detailed specification of the distributed ledger will be covered in deliverable D3.2. In the MDM portal, all actions performed on a record are saved as Audit logs, allowing users to see who accessed the data and what actions were taken. Each action is logged with the following details: the user who performed the action, the name of the action, the timestamp, and a description that includes a detailed account of how the record was changed during this action. In addition to storing data in the distributed ledger, access events are also logged at certain data sources (e.g., Elektro Primorska). This is particularly important for points where data is more confidential or includes sensitive information, such as personal data.	• Security • Availability
Availability	Due to the time-sensitive nature of data collection from certain sources, the availability of sDATA must be ensured at all times. Data collection from providers to sDATA is implemented as frequently running jobs to capture all new data promptly, preventing any data loss. For more static or non-time-sensitive data, collection occurs once daily. This data is stored in the sDATA database in a standardized format and data model. The data gathering jobs are described in more detail in the Open APIs section of the System specifications. To monitor the system, enhance performance, ensure availability, and address any unexpected issues, logging system (such as Elasticsearch) is implemented for fast search and analytics.	• Availability • Performance

Connectivity	Data providers, sDATA, and data users are interconnected through multiple mechanisms, including RESTful APIs, MQTT brokers, direct database connections and file share with read access rights. Each data provider specifies its own access point, from which sDATA retrieves data at frequent, predefined intervals. sDATA offers a REST API for data users to access the data required for further analysis. Data exchanges use standard formats, primarily JSON. For grid topology data in the Slovenian pilot, an RDF schema is employed for sharing between DSO and sDATA. This data is then transformed into the Panda Power format to facilitate further analysis upon data user's request. All data exchange processes are secured with measures such as encryption, authentication, and access control to ensure safe and reliable connections. The data exchanged through sDATA is intended for analytical purposes, which means real-time data is not processed through this system to avoid increasing latency. For edge computing scenarios requiring near real-time data, direct connections (bypassing sDATA) are established to ensure timely data access.	• Connectivity • Interoperability
Database	The sDATA solution incorporates multiple databases to optimize data storage and operations. PostgreSQL with TimescaleDB is used for efficient time-series storage and operations. Additionally, an MS SQL relational database stores MDM data, such as user information, measurement points master data, EV charging stations master data, and grid topology data. The grid topology data model within the MS SQL database adheres to the standardized CIM format, ensuring interoperability and consistency in representing electrical grid structures. Furthermore, a NoSQL database, Elasticsearch, is used for logging purposes, as mentioned in the previous availability section. This multi-database approach ensures optimal performance and efficiency.	• Performance • Interoperability
API compatibility	The sDATA platform offers well-documented and consistent REST APIs to streamline data exchange between various applications and platforms. Detailed descriptions of these APIs can be found in the Open APIs section of the System Specifications chapter. The design of these APIs and the overall solution is versatile, allowing for easy adaptation to other UCs with minimal modifications. For instance, a water distribution system could be modelled using the CIM framework, imported into sDATA, and then exported using the existing APIs.	• Interoperability

Scalability	The sDATA portal is designed with scalability in mind. Both the MDM system and the Time Series components are designed to accommodate the addition of new entities, such as users, master data records, and time series data, which supports growing data volumes. Although scalability was not the primary focus for this project, the MDM system and Time Series component are equipped to handle significant data traffic and storage demands in production environments. Furthermore, both the MDM and Time Series components are designed for easy integration of new features, ensuring the platform can adapt and evolve to meet future requirements. The databases for both MDM and Time Series components utilize a generic data model, ensuring that the integration of new data sources or record systems can be achieved with minimal effort, without requiring changes to the existing data model.	• Scalability

Edge computing specification in the SI pilot:

The edge computing specifications of sDATA in the SI pilot are listed in Table 24.

Table 24: Non-functional specification of Edge computing in SI pilot

Functionality	Description	Type
Connectivity to different types of assets	The chosen PLC type, Siemens S7-1200 is an industry standard and therefore offers exceptional flexibility when connecting to different types of assets. Among others, the chosen PLC offers great support for different communication protocols, which is crucial when setting up communication.	• *Interoperability* • *Connectivity*
Different types of inputs and outputs	The chosen PLC type, Siemens S7-1200 is an industry standard and therefore offers exceptional flexibility when connecting to different types of assets. Among others, the chosen PLC offers great support for different types of data for input and output, which is crucial for data analysis.	• *Interoperability* • *Connectivity*
Security of connection	A private Access Point Name (APN) establishes a secure communication channel, independent of the internet, for devices using the MQTT protocol, which offers a sufficient level of security.	• *Security and Privacy* • *Connectivity*
Scalability	The low cost of used PLCs makes deploying a dedicated controller for each device connected to the aggregation platform a cost-effective solution.	• *Scalability*
Ease of connecting devices to the aggregation platform	To ensure seamless connectivity between PLCs and the aggregation platform, we leverage a private LTE mobile network. This private APN enhances security and simplifies device management.	• *Connectivity* • *Security and Privacy*

Collection of data on 1-min bases	Siemens S7-1200 guarantees reliable data collection and efficient asset management on a (at least) 1-min basis.	• *Performance*

3.3.3 System specification

3.3.3.1 Architecture

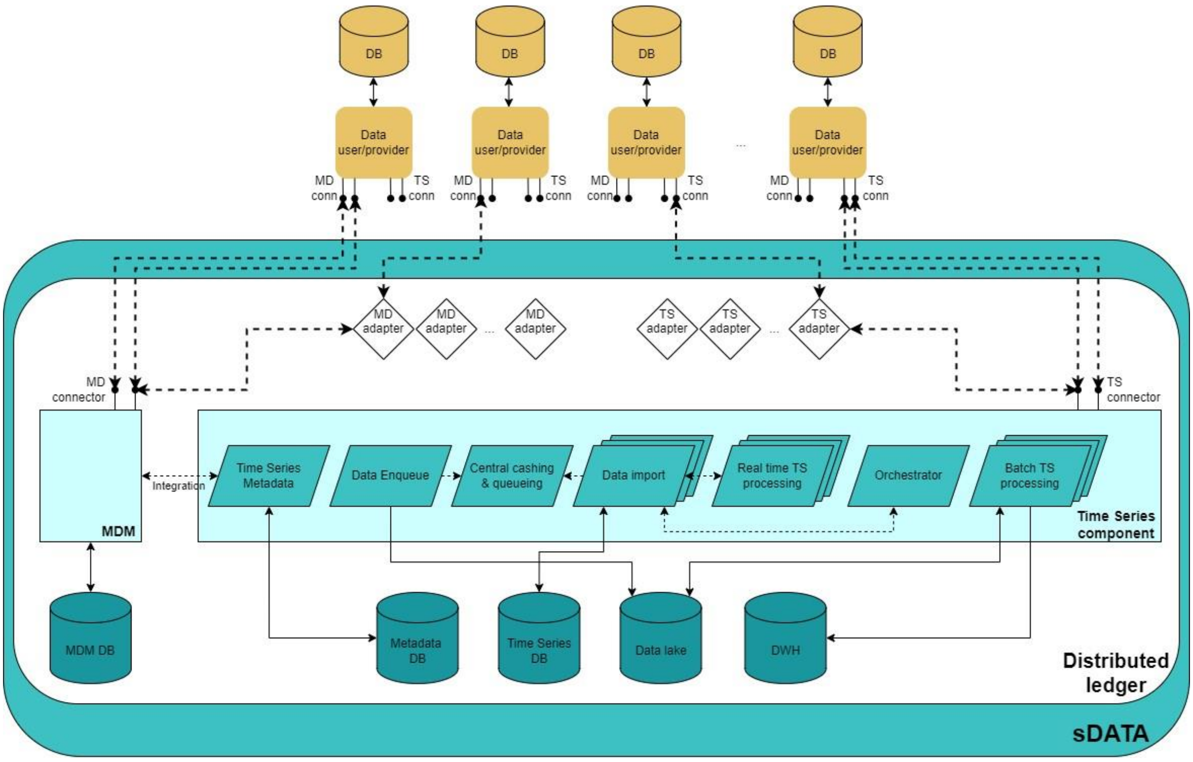

Figure 69: Architecture of the sDATA in SI and IT pilot

The sDATA tool comprises two primary components: the Time Series component and the MDM component as shown in Figure 69.

Additionally, a distributed ledger is integrated into sDATA, which records all access events across the solution. This distributed ledger, with its own dedicated databases for storing access events, will be thoroughly defined and described in the upcoming D3.2 deliverable.

sDATA exchanges data with both data providers and data users (data clients) through connectors, specifically Master Data connectors and Time Series connectors. Each connector group includes distinct Time Series and Measurement Point connectors.

Data exchange protocols among customers vary significantly (e.g., REST, SOAP, MQTT, SQL, File share), as do the data formats they use (such as XML, JSON, CSV, RDFS). This heterogeneity extends to security methods and authentication, which can differ by user, including API keys, username/password combinations, or digital certificates, etc.

If a data user or provider is Open Data Spaces-ready, meaning it can read and write data in the generic sDATA format, the connection can be made directly via the previously mentioned connectors. Otherwise, adapters are implemented to translate between the client's specific messages and protocols and sDATA (whether for MDM or the Time Series Component). These adapters also enable a pull mechanism, allowing data to be retrieved from sDATA. This scenario, involving adapter implementation, is the most common, as demonstrated in the SI and IT pilots. In these cases, each data provider employs its own data sharing methods and security measures. For more detailed information on these exchanges, please refer to the Open APIs section.

The Time Series component is built with the following main building blocks:

- **Time Series Metadata block** is designed to manage time series metadata and is integrated with the MDM portal for synchronization between master data in the portal and the Time Series component. It works with relational metadata database (e.g. PostgreSQL).
- **Data Enqueue block** receives, and offloads data received from data clients. In complex systems, like energy systems, where data loads can be substantial, this component ensures fast and efficient data integrations. After receiving the data, the Data Enqueue block offloads it into the Central cache and queuing block.
- **Central cashing & queueing block** stores the data received from the Data Enqueue block using the FIFO queue (First In, First Out).
- **Data import** block handles saving data into the time series-optimized relational database. It connects to the Central Caching & Queuing block, where it retrieves time series data and stores it in the database. Multiple instances of the Data Import block may operate simultaneously. If needed, data is sent from the Data Import block to the Real-Time Time Series Processing block, where it is processed (cleansed) and then imported into the time series database.
- **Real-time TS processing block** facilitates real-time time series data processing and cleansing to ensure high data quality. For instance, it can fill null values with the most recent known value.
- **Orchestrator** manages the data import by distributing all measurement points across all instances of Data Import.
- **Batch TS processing block** is designed for non-real-time time series data analysis, supporting large-scale and big data transformation, analysis, and preparation for reporting. It ensures high-quality data through various mechanisms such as filling null values with the last known value, interpolation, or the similar days method. It also handles unifying and transforming time intervals, data aggregation, and other ETL-related operations.

Multiple storages are used to support the operation of the Time Series component:

- Metadata database,
- Time Series database,
- Data lake,
- Data warehouse (DWH).

The Time Series and Metadata database employ a generic data model, enabling seamless integration of time series data and metadata from various data providers into a unified framework. This design enhances the system's interoperability and scalability, ensuring that integrating new partners is straightforward and does not require changes to the existing database model.

The MDM portal is crucial component of sDATA, responsible for handling all master data necessary for operations and user management. This data is stored in a dedicated relational database, which utilizes a generic data model. This flexibility allows for the seamless inclusion of new record sets or systems (e.g. EV fleet for a virtual battery), without requiring changes to the database structure. The portal also includes a user-friendly interface for easy interaction.

A NoSQL database (e.g. Elasticsearch) is used for logging, enhancing system monitoring and error detection. This is particularly valuable during the development phase, where bugs and errors are more frequent.

The use of the same sDATA solution in both pilots, SI and IT, demonstrates the interoperability and multi-purpose nature of the solution. Each of these pilots has different data and different UCs, while the generically designed architecture of the solution does not require changes in the implementation of core components. This proves that the platform is applicable for various cases and can easily be adapted for other purposes, such as gas and water distribution networks, or other scenarios where there is a need for processing large amounts of data and involving various stakeholders.

3.3.3.2 Open APIs

Avantcar – sDATA

Avantcar has developed a REST API for EV chargers' data exchange. A GET method was implemented for data exchange between Avantcar and sDATA, secured by an API key, as shown in Figure 70.

The API response contains metadata of the EV charger, its status and EV's metadata. When the charger is in charging mode, the status is updated every 5 minutes.

A data gathering job has been implemented in the sDATA component, which retrieves data from Avantcar every 5 minutes, transforms it to a standard predefined timeseries data model and saves it into a time series database. This data is then available for other partners to access through the sDATA API.

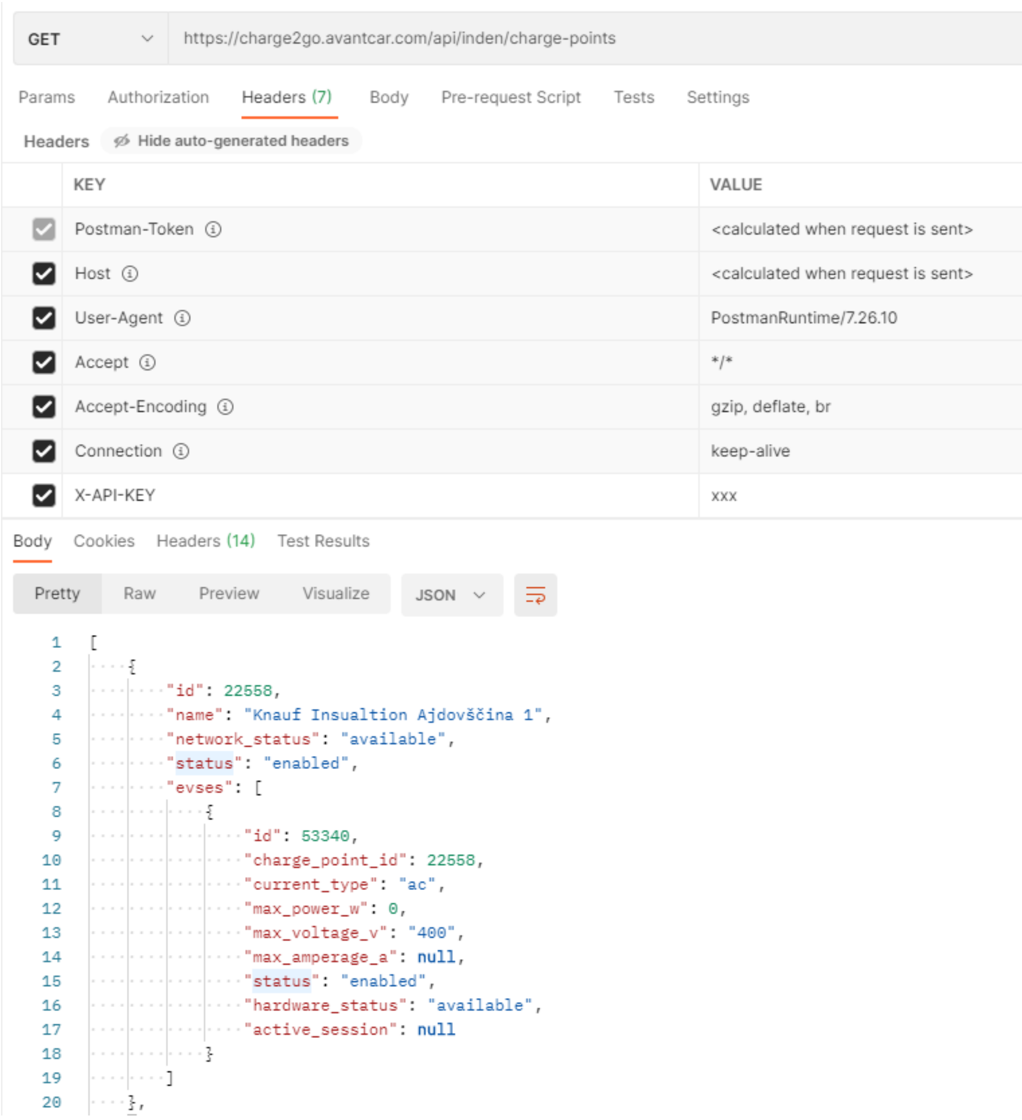

Figure 70: Avantcar's GET call for EV chargers' data exchange

Elektro Primorska – sDATA

Similar to Avantcar, Elektro Primorska (EPR) has implemented an API for data exchange as shown in Figure 71. Access to the data is protected by user authentication via login with username and password, which provides a JWT token. This token is required for user authorization to retrieve data from EPR. All requests are logged on the EPR side to provide an overview of user access and activity.

The exchanged data includes measurement point data (voltage and power) collected from:

- power meters (prosumer measurements),
- control points (medium voltage to low voltage transformer measurements),
- SCADA measurements.

Data for each measurement point can be retrieved by providing input parameters:

- measurement point ID,
- start time,
- end time.

A daily job has been configured to gather the data from EPR, transform it to a standard predefined timeseries data model and save it to the sDATA time series database. This data is then available for other partners to access through the sDATA API.

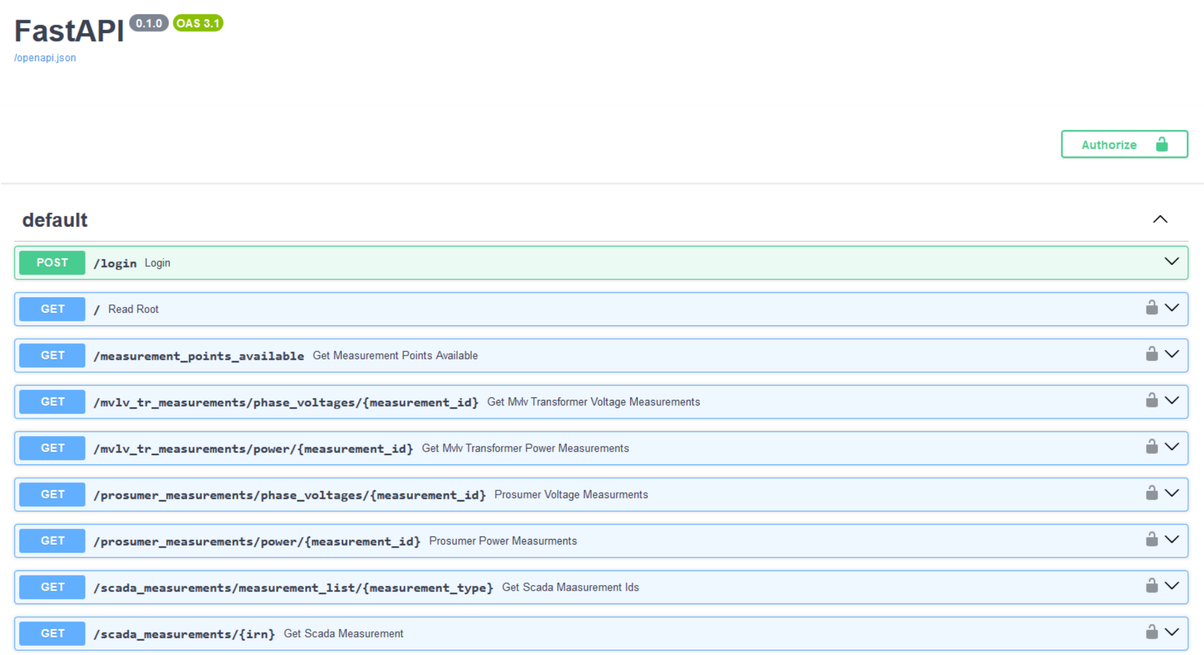

Figure 71: Elektro Primorska's API calls for data exchange in swagger

ASM – sDATA

ASM (the Italian DSO) and sDATA have established a direct connection, allowing sDATA to access the ASM database, which contains consumption and production data (power quality analysers data and smart meters data).

In the ASM database, each measurement point is characterized by the following attributes:

- TagName
- DateTime,
- Value.

sDATA has been granted read-only access to the database, based on the IP address of the machine where the sDATA application is deployed, for a predefined set of measurement point Tag names that are part of the STREAM project.

A daily job has been configured to gather the data from ASM database, pre-processes it to fill the missing values, transform it to a standard predefined timeseries data model and save it to the sDATA time series database. This data is then available for other partners to access through the sDATA API.

In addition to direct database access, an MQTT broker has been implemented on the ASM side for data acquisition. Since the sDATA portal is designed for analytical purposes rather than near real-time data processing, having real-time data in sDATA is not essential.

Additionally, performing data updates with daily jobs improves system performance. In future real-world scenarios, an MQTT connection could be integrated into the sDATA portal if it is upgraded to support near real-time data. However, for the purposes of this project, such an implementation is not required.

For other data, such as medium voltage production and consumption data, water reservoir data, and low voltage photovoltaics data, which are not provided through direct database access, the data is uploaded to sDATA in CSV format via integration endpoints.

Emotion – sDATA

Emotion (EMOT) has developed an API for electric vehicles' and EV chargers' (Towers) data:

- EVs data: https://emotion-projects.eu/api/vehicle-states/
- EV chargers data: https://emotion-projects.eu/api/tower-states/

A daily job has been configured to gather the data from ASM database, pre-processes it to fill the missing values, transform it to a standard predefined timeseries data model and save it to the sDATA time series database. This data is then available for other partners to access through the sDATA API.

In addition to the described API for EMOT data exchange, an MQTT broker has been implemented on the EMOT side for data acquisition. Since the sDATA portal is designed for analytical purposes rather than near real-time data processing, having real-time data in sDATA is not essential. Additionally, performing data updates with daily jobs improves system performance. In future real-world scenarios, an MQTT connection could be integrated into the sDATA portal if it is upgraded to support near real-time data. However, for the purposes of this project, such an implementation is not required.

sDATA – other STREAM tools (data export)

An Open API has been implemented in the sDATA portal to enable data exchange with other STREAM tools, such as sGRID or sFLEX for analytical purposes.

The Open API includes complete documentation of endpoints to gather data of the EV charging stations, timeseries measurement data from the grid and grid model data in pandapower format for easier further analysis as listed in Table 25 and shown in Figure 72.

Data access is secured using API keys, as interaction with the data will primarily be through applications. Each resource (endpoint) is individually secured and can be accessed with the appropriate API key.

Since not all STREAM tools have been fully developed, there is a possibility that the sDATA APIs could undergo slight changes if the tools' requirements evolve or new needs arise.

The Data Export API is accessible to authenticated users, with an OpenAPI JSON provided to simplify integration.

Data export endpoints have been implemented on the SI and IT pilot sites. To retrieve data successfully, each endpoint requires the dateTimeFrom and dateTimeTo query parameters.

Table 25: Endpoints for the data export

Path	Method	Description
/data-export/si/power-meters	GET	Gets power meter measurements for SI pilot in the given time range.
/data-export/si/controls	GET	Gets control measurements for SI pilot in the given time range.
/data-export/si/scada	GET	Gets SCADA measurements for SI pilot in the given time range.

/data-export/si/ev-charging-stations	GET	Gets EV charging station measurements for SI pilot in the given time range.
/data-export/it/ev-charging-stations	GET	Gets EV charging station measurements for IT pilot in the given time range.
/data-export/it/evs	GET	Gets EVs measurements for IT pilot in the given time range.
/data-export/it/grid	GET	Gets electric grid measurements for IT pilot in the given time range.

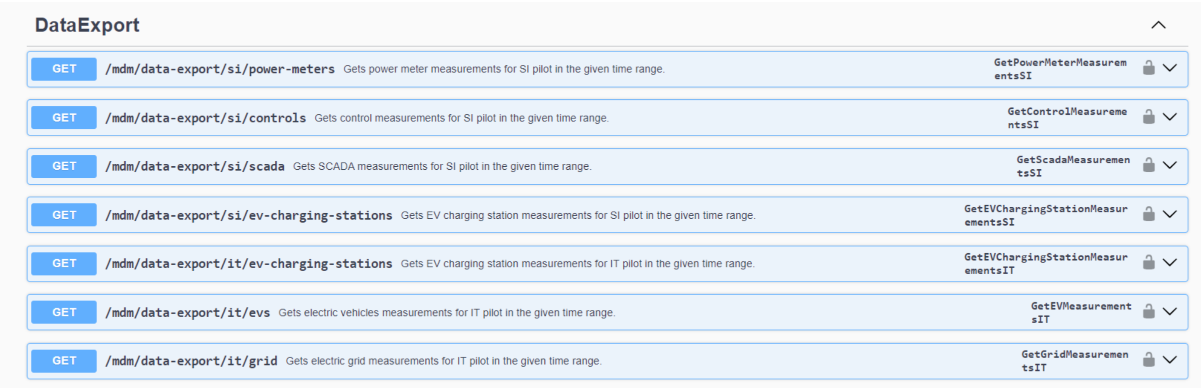

Figure 72: API calls for data exchange in swagger

3.3.3.3 sDATA compliance with Open Data Spaces

This section outlines how the sDATA developed in the SI and IT pilots aligns with the principles and requirements of Open Data Spaces. It begins by detailing the alignment with reference architectures (BRIDGE, IDSA, GAIA-X, and OPENDEI) and then explains how sDATA meets the overall requirements.

3.3.3.3.1 sDATA and BRIDGE

While sDATA is not structured exactly as proposed by the BRIDGE architecture, its functionalities still address most of BRIDGE's requirements and principles. Within the context of the BRIDGE architecture, sDATA handles the federated aspects, while other STREAM tools manage the local components.

- **Component layer**: On the local side, data providers in the STREAM project ensure that personal data is either anonymized or deleted to mitigate any risks. All data and exchange processes are securely managed using user credentials, API keys, and access restrictions based on specific IP addresses. Consequently, only authorized users with the appropriate credentials can access the data.
- **Communication layer**: The data formats supported in sDATA for data exchange, such as JSON, XML, CSV, are always standard, as are the communication protocols like MQTT, SOAP, File share and REST API. These open and standard protocols ensure data sovereignty and confidentiality, supported by the previously mentioned security mechanisms, including user credentials and API keys. A detailed description of the overall cybersecurity measures will be provided in Deliverable 3.2.
- **Information layer**: Open APIs are defined to serve as a common vocabulary for data exchange, facilitating seamless integration with sDATA. Data is pre-processed at the source and written in a pre-defined format. For certain UCs where data is not pre-processed or unified at the

source, sDATA handles this task (e.g. ASM data in IT pilot). Anonymization is managed at the data source, so sDATA remains agnostic about which data is personal. Metadata management is conducted through the MDM portal by data providers or data owners, with audit logs in the portal aiding in monitoring data access and usage. Cybersecurity measures to ensure data security will be detailed in Deliverable 3.2.

- **Function layer**: In sDATA, users are identified through their credentials via the MDM portal, allowing them to access master datasets and measurements/grid metadata in a standardized manner. All data shared through the sDATA platform is provided free of charge for this project. Within other STREAM tools, identified as the "local" components, digital twins (e.g. in the IT pilot) are planned to be implemented based on data exchanged through sDATA. Secure and private data exchange is ensured using credentials shared between sDATA and these tools. Additionally, anonymized user actions and data exchanges are monitored in the Elasticsearch database to ensure system performance and identify potential bottlenecks for further optimization.
- **Business layer**: The sDATA platform facilitates efficient data exchange, supporting various UCs, analyses, and studies within STREAM. The data sharing portal is essential for delivering high-quality data to users while ensuring system stability and preventing overload on data providers.

3.3.3.3.2 sDATA and IDSA

Like BRIDGE, the IDSA architecture outlines five distinct layers, emphasizing security, governance, and certification. Although sDATA is not structured exactly as the IDSA model, it fully meets the requirements of IDSA's functional layer, which is focused on Open Data Spaces:

- **Trust**: As previously noted, participants are identified and certified before gaining access to the trusted business ecosystem of sDATA.
- **Security and data sovereignty**: Security measures, including authentication and authorization, are integrated within sDATA and implemented by data providers to ensure robust protection. Each participant has unique login credentials, and data owners can assign specific access rights to users.
- **Ecosystem of data**: In line with IDSA's recommendations, the sDATA architecture incorporates decentralized data storage. Data is primarily stored locally by the data owner and can be held in sDATA if needed before being transferred to the data consumer. Additionally, the MDM portal oversees the management of all metadata related to the exchanged data.
- **Standardized interoperability**: Data exchange is defined and specified in the Open APIs section. The sDATA architecture aligns with several principles of the proposed Data Spaces Connector. As outlined, the sDATA solution supports the integration of multiple communication protocols and message formats, directly to the sDATA connectors or via adapters. If a new user with a different protocol needs to be integrated, only an additional adapter must be implemented (if not already available). This ensures that integrating new technologies and data clients requires minimal effort, while maintaining seamless interoperability.

- **Value adding apps**: The Time Series component of sDATA delivers the functionality described by value-adding apps. It pre-processes data, fills in missing values, and unifies it. Additionally, it transforms data into the required format to meet the needs of Data Consumers, such as converting data into the pandapower grid model for the sGRID in the SI pilot site. While these functionalities are not implemented as standalone applications—since that was not the objective of the STREAM project—they still provide significant benefits to data users.
- **Data markets**: As previously stated, all data exchanged through sDATA is provided free of charge. Data markets are not within the scope of this project.

3.3.3.3.3 sDATA and GAIA-X

GAIA-X features a more complex architecture compared to sDATA, but sDATA still meets the core requirements outlined by GAIA-X:

- **Data and services interoperability**: The primary objective of sDATA is to facilitate data exchange among multiple stakeholders while ensuring interoperability through standardized communication protocols and data models.
- **Portability of data and services**: As previously noted, standardized communication protocols and data models ensure interoperability. Within the STREAM project, two separate sDATAs (developed as one solution) have been implemented: one at the IT pilot site and another at the SI pilot site.
- **Sovereignty over data**: Participants maintain full control and transparency over their data and its usage by managing access rights, reviewing actions via the MDM portal, and accessing a history log of all access events recorded in the distributed ledger.
- **Security and trust**: Security is a fundamental feature of the sDATA portal, safeguarding both data and users. Comprehensive authentication and authorization mechanisms are embedded across all sDATA components to ensure that only authorized users can access the platform and its data. Additionally, the MDM portal includes a robust user management system to oversee user access and rights.

3.3.3.3.4 sDATA and OPENDEI

Like OPENDEI, sDATA incorporates many of the functionalities outlined in OPENDEI’s three building blocks. As a data exchange platform, sDATA integrates a Time Series component and a MDM portal, aligning with OPENDEI's data platform model. As previously mentioned, all data transmitted through sDATA is provided free of charge, so a marketplace component is not implemented.

The MDM portal, which includes user management, fulfils the Data Sovereignty component proposed by OPENDEI. Data providers specify the terms for data collection and storage, and retention policies can be updated as needed.

Table 26 compares participants in the STREAM project with those outlined in OPENDEI.

Table 26: sDATA and OPENDEI participants

OPENDEI participant	STREAM participants	Role description
Data consumers	Aggregators, DSOs, and other STREAM tools, such as sGRID, sFLEX.	They access the data space to use data.
Data providers	DSOs, E-mobility providers.	They make data available in the data space.
Data producers	DSOs, E-mobility providers, Flexibility devices, Weather forecast & data providers.	The ones creating the data.
Data owners	DSOs, E-mobility providers.	The ones that can provide or deny access and use of data.
Data application providers	IT providers, DSOs, STREAM tools developers.	Developers of applications that ingest data that is available in the data space, processes the data and visualises it.
Data platform providers	IT providers.	Developers that provide capabilities that allow operation of data platforms.
Data marketplace providers	/ *(*In the case of SI and IT pilot sites, there is no data marketplace.)	Developers that provide capabilities that allow operation of data marketplaces.
Identity providers	IT providers.	Provide capabilities for identifying parties

The following points outline how sDATA aligns with the requirements of OPENDEI:

- **Data-sharing empowerment**: sDATA enables data owners to control who can access their data. Access rights to the MDM portal are managed by administrators, while credentials for accessing data in the sDATA time series database are issued by the data owners. This ensures that unauthorized access to the data is not possible.
- **Data-sharing trustworthiness**: As previously detailed, sDATA employs various mechanisms to ensure trustworthiness in data sharing. These include robust safety and security measures, comprehensive user records, and other protective protocols.
- **Data-sharing publication**: Data exchange endpoints are detailed in the Open API section of this document. Authorized users of the STREAM project can access the data through these endpoints.
- **Data-sharing economy**: All data shared within the STREAM project is provided free of charge, and all data users are informed of this.
- **Data-sharing interoperability**: sDATA supports standardized data exchange, allowing users to seamlessly use, transfer, and exchange data in predefined formats.
- **Data-space engineering flexibility**: sDATA is designed for easy customization and is readily adaptable for new integrations and data users. Engineers, familiar with the sDATA data model, can efficiently integrate additional data sources (in the Time Series and MDM components),

or update records in the MDM portal. New data users can access the data promptly once their access credentials are provided.

- **Data-space community**: sDATA has been designed to align with the requirements and principles of Open Data Spaces, informed by a thorough review and analysis of existing open data space frameworks.

By meeting these requirements, sDATA also adheres to the core principles of OPENDEI, including Data Sovereignty, a Level Playing Field for Data, Decentralized Soft Infrastructure, and Public-Private Governance.

3.3.3.3.5 sDATA and General Requirements

Given its alignment with various Open Data Spaces designs described above, it is clear that sDATA meets the general requirements and principles of Open Data Spaces:

a) **Interoperability and standardization:** sDATA facilitates interoperable data exchange by supporting standardized data models for time series, master data, and grid topology data. The platform defines Open APIs for data exchange in predefined, standardized formats. New users can be added and granted access with ease, and integrating new data providers requires minimal effort.

b) **Data security and trust:** All data exchanged through sDATA is protected by robust security measures. Access is restricted to authorized users only, ensuring that data remains secure.

c) **Data sovereignty:** Data owners have full control over their data, including the ability to manage user access and review the audit log and history logs of data access. Retention policies can be easily applied and updated as needed. Consequently, data consumers must adhere to the policies and conditions established by the data providers.

d) **Use of non-personal data and anonymization of personal data:** Data is anonymized and pre-processed at the source, ensuring that no personal data is transmitted through the sDATA platform. This approach complies with GDPR regulations.

4 CONCLUSIONS

This deliverable focuses on guiding and detailing the **development process of the sDATA tool** and data exchange at the pilot sites.

It builds on a review of **previous projects,** established **Open Data Spaces,** and their requirements and principles, along with **UCs analysis** and **data user needs**. While specific UCs and data requirements **vary by pilot**, a set of **general requirements** applicable across all sites has been identified. The set is described in Table 27:

Table 27: A set of general sDATA requirements

Requirement	Short description	Type
Data exchange	**Authorized users** should retrieve data via **standardized formats and protocols**.	Functional
Anonymized and encrypted data	Data exchange must ensure **security** through **anonymization and encryption** to protect sensitive data.	Functional
Data pre-processing	Provider's data must be **pre-processed** to ensure **quality** for modelling and analysis.	Functional
Data visualization	A **data visualization** tool is needed to help users **interpret measurement data** and **support decision-making**.	Functional
Data storage (retention policy)	Data from external sources must be **retained for access by users and services**, with retention periods defined by data providers.	Functional
Availability	Data from sDATA must be **accessible on request**, with continuous data exchange and **frequent updates** to prevent loss.	Non-functional
Security	**Authentication and authorization** must verify **user identities** and **control data access**, with each user assigned specific data domains. **Data exchange** must be **secure**.	Non-functional
Performance	Data collection must be **frequent and efficient** to prevent loss and support **high-speed ingestion**. The platform should use **time-series databases** for managing large datasets and ensure fast, seamless data exchange without disrupting performance.	Non-functional
Connectivity	The sDATA platform must ensure **robust, secure connectivity** for tools to gather data on request, with reliable connections for real-time access and updates. High availability and low latency are essential for optimal performance and user satisfaction.	Non-functional
Interoperability	The sDATA platform must support **standardized data formats** and **exchange protocols** for interoperability and offer well-documented APIs compliant with industry standards.	Non-functional

Scalability	The sDATA tool must be **scalable to handle increasing loads**, ensuring it accommodates growing data volume and traffic, even if the number of customers remains stable.	Non-functional

Furthermore, sDATA tool must comply with the core principles and requirements of the Open Data Spaces:

- **Interoperability and standardization,**
- **Data security and trust,**
- **Data sovereignty,**
- **Use of non-personal data and anonymization of personal data.**

Based on both general and pilot-specific requirements, sDATA was developed as **two distinct solutions**: one for the FI pilot site and a unified solution for the SI and IT pilot sites (with IND as the technology provider for both SI and IT pilots). These solutions demonstrate how identical requirements can be met with **different architectures** while **maintaining core principles**. Specifically, the same sDATA solution illustrates its **flexibility and applicability** across the SI and IT pilots, despite their **varied UCs, data clients, and data types**.

The use of a single sDATA solution in both pilots highlights its **interoperability and multi-purpose nature**. The platform's **generic architecture** ensures that core components remain consistent, showcasing its **adaptability for other applications**.

The developed sDATA will also **support the development of other STREAM tools**, such as sFLEX and sGRID by providing **high-quality data in one place**. As new challenges in data exchange arise, sDATA will evolve, but its core requirements and architectural design have been robustly defined to support future developments.

5 REFERENCES AND ACRONYMS

5.1 REFERENCES

[1] BD4NRG, *D3.3: BD4NRG Governance – Final Technology Release*, 2023.

[2] BRIGHT, *Deliverable D2.4 Cross domain Data & Service Interoperability*, 2021.

[3] Platone, "D2.10 Specification of the interoperability and standard communication protocols (v2)," 2023. [Online]. Available: https://www.platone-h2020.eu/data/documents/864300_M48_D2.10.pdf.

[4] Platone, "D2.16 Platone Integrated Framework Prototype (v3)," 2023. [Online]. Available: https://www.platone-h2020.eu/data/deliverables/864300_M48_D2.16.pdf.

[5] L. Nagel and D. Lycklama, "Design of Principles for Data Spaces. Position Paper. Version 1.0.," 15 July 2021. [Online]. Available: http://doi.org/10.5281/zenodo.5105744.

[6] Directorate-General for Energy, Common European Energy Data Space, 2023.

[7] European Commission (EC), "A European strategy for data," 4 April 2024. [Online]. Available: https://digital-strategy.ec.europa.eu/en/policies/strategy-data.

[8] "COMMUNICATION FROM THE COMMISSION TO THE EUROPEAN PARLIAMENT, THE COUNCIL, THE EUROPEAN ECONOMIC AND SOCIAL COMMITTEE AND THE COMMITTEE OF THE REGIONS," 19 2 2020. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52020DC0066.

[9] European Commission, "Staff working document on data spaces," 14 2 2022. [Online]. Available: https://digital-strategy.ec.europa.eu/en/library/staff-working-document-data-spaces. [Accessed 9 7 2024].

[10] M. Couto, K. Kotsalos and K. Kukk, "European (energy) data exchange reference architecture 3.0," Publications Office of the European Union, July 2023. [Online]. Available: https://data.europa.eu/doi/10.2833/815043.

[11] International Data Spaces Association (IDSA), "IDS-RAM 4," 2023. [Online]. Available: https://docs.internationaldataspaces.org/ids-knowledgebase/v/ids-ram-4/introduction/1_1_goals_of_the_international_data_spaces.

[12] International Data Spaces Association (IDSA), "Datspace protocol," 2024. [Online]. Available: https://docs.internationaldataspaces.org/ids-knowledgebase/v/dataspace-protocol.

[13] International Data Spaces Association (IDSA), "Data Connector Report," April 2024. [Online]. Available: https://internationaldataspaces.org/wp-content/uploads/dlm_uploads/IDSA-Data-Connector-Report-No-14-April-2024.pdf.

[14] GAIA-X, "About GAIA-X," GAIA-X, [Online]. Available: https://gaia-x.eu/what-is-

gaia-x/about-gaia-x/.

[15] GAIA-X, "Gaia-X Architecture Document - 21.12 Release," April 2023. [Online]. Available: https://www.gaiax.es/sites/default/files/2023-04/Gaia-X_Architecture_Document_2112.pdf.

[16] OPENDEI Project, "OPEN DEI TASK FORCE 1," [Online]. Available: https://design-principles-for-data-spaces.org/.

[17] "International Electrotechnical Commission," [Online]. Available: https://en.wikipedia.org/wiki/International_Electrotechnical_Commission.

[18] "Common Information Model," [Online]. Available: https://www.entsoe.eu/digital/common-information-model/.

[19] "IEC 61970," [Online]. Available: https://en.wikipedia.org/wiki/IEC_61970.

[20] "IEC 61968," [Online]. Available: https://en.wikipedia.org/wiki/IEC_61968.

[21] "IEC 61968-1," [Online]. Available: https://webstore.iec.ch/en/publication/32542.

[22] ENTSOE, *Technical Requirements for Frequency Containment Reserve Provision in the Nordic Synchronous Area*, 2023.

[23] STREAM project, *D2.2 Use case and requirements definition*, 2024.

[24] "Highcharts," [Online]. Available: https://www.highcharts.com/.

[25] "Recharts," [Online]. Available: https://recharts.org/.

[26] "tremor," [Online]. Available: https://www.tremor.so/.

[27] "JSON Web Signature (JWS)," [Online]. Available: https://openid.net/specs/draft-jones-json-web-signature-04.html.

[28] "MQTT - The Standard for IoT Messaging," [Online]. Available: https://mqtt.org/.

[29] "AMQP," [Online]. Available: https://www.amqp.org/.

[30] "HTTP documentation," [Online]. Available: https://httpwg.org/specs/.

[31] "MongoDB," [Online]. Available: https://www.mongodb.com/.

[32] "InfluxDB: Purpose-Built Open Source Time Series Database | InfluxData," [Online]. Available: https://www.influxdata.com/.

[33] "Publications and Data Loading | Meteor Guide," [Online]. Available: https://guide.meteor.com/data-loading.html#publications-and-subscriptions.

[34] "Meteor," [Online]. Available: https://www.meteor.com/.

[35] "The Transport Layer Security (TLS) Protocol Version 1.3," 2018. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc8446.

[36] "RabbitMQ -- Messaging that just works," [Online]. Available: https://www.rabbitmq.com/.

[37] "MongoDB Docs: Document Limitations," [Online]. Available: https://www.mongodb.com/docs/v6.0/core/document/#document-size-limit.

[38] "influxdata Documentation: Max body size," [Online]. Available: https://docs.influxdata.com/influxdb/v1/administration/config/#max-body-size--25000000.

[39] "X-FLEX," [Online]. Available: https://xflexproject.eu/.

[40] "ASSISTANCE Project," [Online]. Available: https://assistance-project.eu/.

[41] "SAURON Project," [Online]. Available: https://www.sauronproject.eu/.

[42] "MATCHUP Project," [Online]. Available: https://www.matchup-project.eu/.

[43] "ISO/IEC 30182:2017," [Online]. Available: https://www.iso.org/standard/53302.html.

[44] "UNE 178104:2017," [Online]. Available: https://www.en-standard.eu/une-178104-2017-sistemas-integrales-de-gestion-de-la-ciudad-inteligente-requisitos-de-interoperabilidad-para-una-plataforma-de-ciudad-inteligente/.

[45] "Kubernetes," [Online]. Available: https://kubernetes.io/.

[46] "WebGL," [Online]. Available: https://get.webgl.org/.

[47] "OpenID - The Internet Identity Layer," [Online]. Available: https://openid.net/connect/.

[48] "Keycloak," [Online]. Available: https://www.keycloak.org/.

[49] "Express - Node.js web application framework," [Online]. Available: http://expressjs.com/.

[50] "NATS.io – Cloud Native, Open Source, High-performance Messaging," [Online]. Available: https://nats.io/.

[51] "XGBoost," [Online]. Available: https://xgboost.ai/.

5.2 ACRONYMS

Acronyms list	
AMI	Advanced Metering Infrastructure
AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
AWS	Amazon Web Services
BAL	Blockchain Access Layer

CIM	Common Information Model
CSV	Comma Separated Values
DDP	Distributed Data Protocol
DERA	Data Exchange Reference Architecture
DGA	Data Governance Act
DMA	Digital Markets Act
DSA	Digital Services Act
DSO	Distribution System Operator
DWH	Data Warehouse
EC	European Commission
eIDAS	Electronic Identification and Trust Services
EnC	Energy Community
ENTSO-E	European Network of Transmission System Operators for Electricity
ES	Spain
ETL	Extract Transform Load
EU	European Union
EV	Electric Vehicle
FAIR	Findable, Accessible, Interoperable and Reusable
FCR	Frequency Containment Reserve
FI	Finland
FIFO	First In, First Out
FSPs	Flexibility Service Providers
FTP	File Transfer Protocol
GDPR	General Data Protection Regulation
GIS	Geographic Information System
HTTPS	Hypertext Transfer Protocol Secure

ICT	Information and Communication Technology
IDSA	International Data Spaces Association
IEC	International Electrotechnical Commission
IED	Intelligent Electronic Device
IoT	Internet of Things
IT	Italia
JSON	JavaScript Object Notation
LFM	Local Flexibility Market
N/A	Not Available / Not Applicable
MDM	Master Data Management
MO	Market Operator
MQTT	Message Queuing Telemetry Transport
MV	Medium Voltage
OCPP	Open Charge Point Protocol
PSD	Payment Services Directive
PV	Photovoltaics
RAD	Rapid Application Development
RAM	Reference Architecture Model
RDF	Resource Description Framework
REST	Representational State Transfer
SCADA	Supervisory Control and Data Acquisition
SCD	Shared Customer Database
SI	Slovenia
SOAP	Simple Object Access Protocol
SoC	State of Charge
SQL	Structured Query Language

SSL	Secure Sockets Layer
TLS	Traffic Light Signal
TS	Timeseries
TSO	Transmission System Operator
UC	Use Case
UI	User Interface
VPN	Virtual Private Network
WG	Working Group
WP	Work Package
xEMS	Energy Management Service
XML	Extensible Markup Language

6 ANNEX A – COMPLETE LIST OF sDATA REQUIREMENTS

6.1 FUNCTIONAL REQUIREMENTS

Here, all functional requirements from the pilots are listed in Table 28.

In the Finnish pilot, a special requirements' syntax was used for requirement description as follows:

In order to have an informal and natural description of the requirements that always identify the user and the reason of the requirement, the **user story Connextra template syntax** is used to document the requirements.

```
As a <user> I want <requirement> in order to <benefit>
```

For refactored requirements, a less normalized syntax may be used, whenever it is more practical to do so. Example:

```
The table table_name shall contain the following columns: etc …
```

So, each high-level requirement shall contain the WHO, WHAT and WHY, whereas it is acceptable that lower-level requirements that add further details contain only the WHO and WHAT and inherit the WHY from the parent requirement.

Table 28: Full list of pilots' functional requirements

Name	Description	Priority	Pilot
Data ingestion (MQTT, AMQP)	The tools need to be able to receive data from field devices and systems using MQTT and AMQP communication protocols. Specific settings from each protocol need to be supported (e.g. QoS for MQTT, broker-queue management for AMQP, authentication for both).	5	ES
Data ingestion (HTTP)	A RESTful API needs to be available for third parties to send new data to the tools. The API needs to be secured in order to avoid unauthorised use.	5	ES
Data storage	Data received from external sources needs to be persisted for subsequent access both by end users and internal services (e.g. for data analysis). A minimum of 1 year of data is required to be accessible in the repository.	5	ES
Data retrieval (WebSockets, real time)	The most recent value of each received parameter needs to be accessible via WebSocket. Once the channel has been established, any updates must be automatically transmitted to the client. Access to data must be granted using credentials.	5	ES

Data retrieval and integration (HTTP, historical)	Series of historical data must be accessible through a RESTful API. Filtering and integration functionalities must be available, including the mathematical function to use for integration (e.g. min, max, sum, avg) and the time precision to use (e.g. 1 min, 15 min, 1 hour). The API needs to be secured in order to avoid unauthorised access to data.	5	ES
Data pre-processing	The data obtained from the pilot site assets must be pre-processed by cleaning the data, deleting atypical and NaN values, and addressing data quality. This data pre-processing should be done either by the data management services or directly by the tool that uses the data.	5	ES
Data normalization	Data must be normalized to ensure interoperability among the different tools of the pilot site (decimal separator, formats, magnitudes, etc.) and to prevent that could lead to failures in the flexibility market processes, potentially impacting energy security in the network.	5	ES
Regular-Sampling-Interval	As an IOT solution vendor I want my periodic measurement data storage solution to support regular sampling intervals since they are the current used sampling method (currently 10 minutes, in the future 15 minutes). A sampling interval is a measure on how frequently the observation of a certain variable is stored in a time series. When the frequency of the observation is constant, the sampling interval is a regular sampling interval.	5	FI
Irregular-Sampling-Interval	As an IOT solution vendor I want my periodic measurement data storage solution to support irregular sampling intervals to store the measurements when they come directly from the controller.	4	FI
Timestamp	As an IOT solution vendor I want my periodic measurement data storage solution to support an indexed dimension containing the time stamp since this is the main key to classify and query the entries.	5	FI
Apartment-id-Dimension	As an IOT solution vendor I want my periodic measurement data storage solution to support the apartment id dimension of type integer since this is a needed key to classify and query the entries.	5	FI
Room-id-Dimension	As an IOT solution vendor I want my periodic measurement data storage solution to support the room id dimension of type integer since this is a needed key to classify and query the entries.	5	FI

Sensor-id-Dimensions	As an IOT solution vendor I want my new periodic measurement data storage solution to support the sensor id dimension of type integer to be compliant with the current implementation.	4	FI
Measurement-Type	As an IOT solution vendor I want my periodic measurement data storage solution to support the measurement type, logically a categorical variable, since this is one of the key components of the measurements.	5	FI
Measurement-Value	As an IOT solution vendor I want my periodic measurement data storage solution to support the measurement value of type decimal since this is one of the key components of the measurements.	5	FI
FCR-bid-variables	As a demand-response aggregator I want my measurement database to support all the variables that are needed for the bidding process in order to be compliant with Fingrid specifications.	5	FI
FCR-log-variables	As a demand-response aggregator I want my measurement database to support all the necessary log variables for FCR-D-up, FCR-D-down and FCR-N in order to be compliant with ENTSO-E and Fingrid specifications.	5	FI
FCR-KPI-variables	As a demand-response aggregator I want my measurement database to support the necessary measurements in order to calculate the KPIs.	5	FI
FCR-profitability-variables	As a demand-response aggregator I want my measurement database to support the necessary variables to implement the evaluation of the profitability for the customer to support UC FI.05.	5	FI
FCR-comfort-variables	As a demand-response aggregator I want my control database to support the necessary variables for maximizing flexibility and customer comfort in order to support UC FI.02.	4	FI
FCR-user-feedback	As a demand-response aggregator I want my measurements database to support storing customers' answers in order to support UC FI.04.	3	FI
FCR-building-types	As a demand-response aggregator I want my system to be able to store building and heating types specific information in order to support UC FI.03.	3	FI
Aggregate-Data	As a smart energy management company, I want to have metric-specific aggregate types to save meaningful data after purging.	5	FI

Support-for-Append-Operation	As an IOT solution vendor I want my measurement data storage solution to support appending records at the end of the table since this is the basic write operation.	5	FI
Support-for-Insert-Operation	As an IOT solution vendor I want my measurement data storage solution to support inserting records in the middle of the table because this is needed when some controller goes off-line and then it comes back.	5	FI
Support-for-Update	As an IOT solution vendor I want my measurement data storage solution to support updating records since this is needed to override the original energy measurements values after validation, in certain cases.	5	FI
Support-for-Purging	As an IOT solution vendor I want my measurement data storage solution to support purging old data since this is a necessary functionality in a feasible solution.	5	FI
Support-for-Delete	As an IOT solution vendor I want my measurement data storage solution to support deleting records since this could be useful in case of mistakes in data insertion.	4	FI
Support-for-API-to-the-Back-end	As an IOT solution vendor I want my measurement data storage solution to support an API to the backend with a Python/Django friendly/compatible solution to get the source data coming from the sensors.	5	FI
Support-for-SQL-Like-Query-Language	As an IOT solution vendor I want my measurement data storage solution to support a SQL-like query language to support also manual queries and scripts running outside the backend.	5	FI
Support-for-Cloud	As an IOT solution vendor I want my measurement data storage solution to be able to run on the Cloud to have a low initial investment and great scalability.	5	FI
Support-for-On-Premises-Installation	As an IOT solution vendor I want my measurement data storage solution to be able to run on premises to be able to run it on docker during SW development.	4	FI
Ready-Made-Data-Pipeline-Tool	As an IOT solution vendor I want my measurement data storage solution to contain a ready-made ETL (Extract Transform Load) data pipeline tool to import the existing data in a quick, efficient, and trustable way.	4	FI
Data-Visualizations-and-Reporting-Capabilities	As an IOT solution vendor, I want my measurement data storage solution to include the capability to perform data visualization and reporting to support business intelligence operations.	5	FI

Monitoring-Functionality	As an IOT solution vendor, I want my measurement data storage solution to include alerting functionality that monitors, for example, too low temperature values or too high CO2 values to provide notifications to the end user.	4	FI
Data Search and Retrieval	The platform must provide a robust search functionality that allows users to easily find and retrieve specific data, such as timeseries data templates, assets, grid elements, etc. This functionality should support advanced search options, such as filtering by data type, source, and keywords.	4	IT, SI
Metadata management system	The metadata management system must store metadata across various categories, such as time series, measurement points, grid models, and code lists, providing users with all relevant information about the data they use.	3	IT, SI
User management system	The platform must include a comprehensive user management system that supports adding new users, updating existing users, and removing users. Each user's access rights and roles must be adjustable.	5	IT, SI
Data visualization	Data visualization tool for timeseries measurement data needs to be included within sDATA platform in helping users make sense of the data and support informed decision-making.	2	IT, SI
Data exchange	Users should have the capability to request and retrieve data to which they have been granted access. Data exchange should adhere to a predefined standardized format to ensure consistency and interoperability.	5	IT, SI
Anonymized and encrypted data	The data exchange must be highly secure, given the sensitivity of the measurement data, which requires strict anonymization to ensure no personal information is included. Additionally, robust encryption protocols must be implemented to protect the data during transmission and storage, safeguarding it from unauthorized access and potential breaches.	5	IT, SI
Data cleansing	For the modelling and analytical purposes, data cleansing must be performed within the sDATA platform, allowing other tools to focus on their primary tasks using high-quality historical data.		IT

6.2 NON-FUNCTIONAL REQUIREMENTS

6.2.1 Availability

6.2.1.1 Finnish pilot

Data-Availability

As an IOT solution vendor I want my measurement data storage solution to support **99,9% availability (9 hours downtime / year)** to provide a decent level of quality to the end users.

Priority: 5 (Critical)

Data-Quality

As an IOT solution vendor I want my measurement data storage solution to support a high level of data integrity and data consistency to provide a good quality of service to the end users, to provide a trustable data source for analytical purposes and to comply with requirements related to billing data.

Priority: 5 (Critical)

Support-for-Backup-and-Restore-Functionality

As an IOT solution vendor I want my measurement data storage solution to support backup and restore functionality to cope with disaster situations.

Priority: 5 (Critical)

Recovery-Point-Objective

As an IOT solution vendor I want my RPO (Recovery Point Objective) to be 15 minutes.

This means that the amount of data that is tolerated to be lost in case of database failure is the data related to 15 minutes.

Priority: 4 (Essential)

Database-Maintenance

As an IOT solution vendor I want the database maintenance of the periodic measurements done automatically based on pre-defined configurable windows to be able to define the most optimal time intervals.

Priority: 4 (Essential)

Database-Monitoring

As an IOT solution vendor I want database monitoring capabilities such as plots with CPU, memory usage, IOPS as well as configurable alarms to monitor the database usage and plan upgrades in time.

Priority: 4 (Essential)

6.2.1.2 Italian pilot

Data quality

The sDATA must provide consistent, coherent, and uniformly notated data to ensure high quality for analysis by other STREAM tools.

Data availability

For the purpose of analysis in STREAM tools using data provided by sDATA, it is essential that the data be available upon request, at least during regular business hours.

6.2.1.3 Slovenian pilot

Availability of the sDATA must be ensured at all times due to the time-sensitive nature of data collection, for example data provided by Avantcar. The system must continuously collect data to prevent the loss of operational data, thereby maintaining uninterrupted operations.

6.2.1.4 Spanish pilot

For every variable received from any field device or system, raw values must be stored for at least one year. For older values, integrated statistics can be stored if considered relevant (e.g. min, max, avg per 15 minutes for the last 3 years).

The service must be available for data retrieval from the tools at any time, ensuring seamless communication.

6.2.2 Security

6.2.2.1 Finnish pilot

Data-Security

As an IOT solution vendor I want my measurement data storage solution to be secure to prevent unauthorized access.

This means that data shall be encrypted both at rest and in transit.

Priority: 5 (Critical)

Data-Privacy

As an IOT solution vendor I want my measurement data storage solution to support data privacy to support general data privacy regulation and to protect the business.

Priority: 5 (Critical)

Servers-Location

As an IOT solution vendor I want all my servers to be located in Europe (possibly, in the OW AWS Cloud) in order to support general data privacy regulation and to protect the business.

Priority: 5 (Critical)

6.2.2.2 Italian pilot

Authentication and authorization

sDATA must implement authentication and authorization mechanisms to verify user identities and control access to data.

Secure data exchange

The platform must ensure secure data exchange to protect sensitive information. Security features, such as using API keys for data exchange, must be implemented to ensure that only trusted users have access.

Record of access and data exchange requests

The platform (the distributed ledger) should maintain a comprehensive and immutable record of all access events and data exchange requests for auditing purposes, ensuring transparency and accountability.

6.2.2.3 Slovenian pilot

The security requirements are the same as in the IT pilot.

6.2.2.4 Spanish pilot

All access to data must be protected using credentials (username and password for human users, app ID and token for applications and services). Each user must have associated a domain of data that they can access. No unauthorized access must be permitted.

Since the data of the Spanish pilot site is sensitive, all the data must be anonymised and encrypted to prevent misuse and ensure the security of the energy supply.

6.2.3 Performance

6.2.3.1 Finnish pilot

Support-for-Current-Record-Amount

As an IOT solution vendor I want my new measurement data storage solution to support **15 billion records** ($15*10^9$) to be able to contain the current stored data in the production database and in the historical archived snapshots.

Priority: 5 (Critical)

Support-for-Current-Storage-Space

As an IOT solution vendor I want my measurement data storage solution to support **700 gigabytes** ($700*10^9$ bytes) to be able to contain the current stored data in the production database and in the historical archived snapshots.

Such a “low” figure is possible thanks to the aggregation, purging and compression rules in Timescale

Priority: Priority: 4 (Essential)

Timestamp-Precision

As an IOT solution vendor I want my measurement data storage solution to support time stamps with one **second precision** because this precision is enough for the data visualization and reporting needs.

Priority 5 (Critical)

Appending-Frequency

As an IOT solution vendor I want my measurement data storage solution to support appending new records at a **10 minute intervals** for the same apartment, room and measurement type in order to support existing and future functionality.

For energy meters, **1 hour** duration interval would be enough at the beginning, then later it could go down to **15 minutes** and then, may be, down to **5 minutes**.

For FCR logs, the writing frequency is 1 second but a queuing mechanism will be used and the info does not need to be written in the database so frequently. The required interval is **1 minute.**

For on-demand investigations, **few seconds** duration intervals would be needed.

Priority: 5 (Critical)

Data-Retention-Period-for-Fresh-Data

As an IOT solution vendor I want that my measurement data storage solution supports storing full data (including **10 minutes interval** measurements) for at least **1 year** to allow users to get high quality and nice-looking plots when using Optiwatti user interface and allow detailed data analysis through defined API provided to the user or internally in OW.

Priority: 5 (Critical)

Data-Retention-Period-for-Past-Data

As an IOT solution vendor I want that my measurement data storage solution supports storing **hour level data** for at least **2 years** to allow users to get decent quality plots when using Optiwatti user interface and to support detailed analytical UCs either through defined API provided to the user or internally in OW.

For energy measurement data the limit is **6 years** due to regulatory reasons.

Priority: 5 (Critical)

Data-Retention-Period-for-Historical-Data

As an IOT solution vendor I want my measurement data storage solution to support storing **day level data** for at least **5 years** to allow users to get this granularity through some defined API and to support analytical UCs.

Priority: 5 (Critical)

Data-Retention-Period-for-Old-Data

As an IOT solution vendor I want my measurement data storage solution to support storing **week level data** for at least **10 years** to allow users to get this granularity through some defined API and to support analytical UCs.

Priority: 5 (Critical)

Data-Retention-Period-for-FCR-Logs

As a demand-response aggregator I want my FCR log to be kept in the measurement database for **14 days** since this is the requirement from Fingrid.

Priority: 5 (Critical)

Support-for-Current-Data-Ingestion-Speed

As an IOT solution vendor I want my measurement data storage solution to support a data ingestion speed of **300 records / second** to support current needs.

Priority: 5 (Critical)

Support-for-Bulk-Loading-Data-Query-Speed

As an IOT solution vendor I want my measurement data storage solution to support a data query speed of **at least 500 Read IOPS** to support the bulk loading operations.

Priority: 5 (Critical)

Data-Query-Latency-Fresh-and-Past-Data

As an IOT solution vendor I want my measurement data storage solution to support a data query latency of **few hundreds of milliseconds** for data **not older than 3 years** to support the current level of services to end users and to support analytical UCs.

Priority: 5 (Critical)

Data-Query-Latency-Historical-and-Old-Data

As an IOT solution vendor I want my measurement data storage solution to support a data query latency **not longer than few seconds** for data that is **older than 3 years** to support analytical UCs.

Priority: 5 (Critical)

Cardinality-Level

As an IOT solution vendor it is enough that my measurement data storage solution supports a low level of cardinality (number of distinct values in a table column divided by the number of rows in the table).

Priority: 5 (Critical)

6.2.3.2 Italian pilot

Data collection

The data collection process must be frequent and adhere to specific intervals defined by data providers to prevent any loss of data. It should be efficiently executed to ensure timely updates without disrupting other system operations or user interactions with the platform. Maintaining a seamless user experience while collecting and processing data is paramount to the system's operational efficiency and reliability.

Database

The sDATA must incorporate multiple databases to optimize data storage and operations. Specifically, a database designed for efficient time-series storage and operations, such as TimescaleDB, must be used due to its advanced handling of temporal queries and scalability.

Additionally, a relational database, such as MS SQL, must be utilized to store user-related data and grid topology data.

This dual-database approach ensures optimal performance.

Data exchange

Data exchange among users must be efficient, ensuring fast transmission speeds to facilitate timely communication and collaboration. It's essential that data exchange processes do not disrupt other operations within the platform, maintaining optimal performance and responsiveness for all users.

6.2.3.3 Slovenian pilot

The performance requirements for the SI pilot are an extended version of those established for the IT pilot. All requirements specified for the IT pilot site are also applicable to the SI pilot's performance criteria.

In the SI pilot, in addition to the database requirements outlined for the IT pilot, the data model for the grid topology within the relational database must support a standardized format, such as the Common Information Model (CIM). This is crucial to ensure interoperability and consistency in representing electrical grid structures.

6.2.3.4 Spanish pilot

Services for data ingestion must support the size and frequency of messages from systems and devices integrated in the pilot. Requirements identified include mainly JSON messages, ranging between 0.1 and 10 KB, and received from every minute to once a day.

6.2.4 Connectivity

6.2.4.1 Finnish pilot

Interface-to-sFLEX

sDATA shall be able to be read and written by the sFLEX component. The sFLEX-sDATA interface is an Optiwatti internal interface.

6.2.4.2 Italian pilot

The sDATA platform must support robust connectivity to ensure all tools can seamlessly gather data on request, provided they have sufficient rights. Reliable and secure connections are essential for efficient data exchange, enabling real-time access and updates. The system should also facilitate smooth general connectivity, ensuring consistent data flow and interaction between various components and users across the platform. High availability and low latency are crucial to maintain optimal performance and user satisfaction.

6.2.4.3 Slovenian pilot

The connectivity requirements are the same as in the IT pilot.

6.2.4.4 Spanish pilot

Services dedicated to the management of data in the Spanish pilot need to provide robust connectivity to the clients in order to avoid losing field measurements, which are the cornerstone of all data services of the tools. Connectivity uptime must be ensured for ingestion, storage, and retrieval of the data.

6.2.5 Interoperability

6.2.5.1 Finnish pilot

The sFLEX-sDATA interface is an Optiwatti internal interface.

6.2.5.2 Italian pilot

Standardized data formats

The sDATA platform must support data exchange in standard formats such as XML, JSON, and CSV to ensure interoperability.

API Compatibility

The sDATA platform must provide well-documented and consistent APIs that adhere to industry standards (e.g., REST, SOAP) to facilitate data exchange between diverse applications and platforms.

Data Model Consistency

The sDATA platform must implement a consistent and well-defined data model that aligns with industry standards, such as CIM, to ensure uniform representation and understanding of data across different systems.

6.2.5.3 Slovenian pilot

The interoperability requirements are the same as in the IT pilot.

6.2.5.4 Spanish pilot

The integration of equipment from different vendors and through different technologies is required to gather all necessary data for the tools to fulfil their objectives. The communication protocols identified so far (MQTT, AMQP, HTTP REST) have been defined as functional requirements.

Regarding communication among tools, the exchange of messages should be standardized in order to facilitate their integration and enhance their broadcasting. Exchange of JSON messages through AMQP are encouraged, while the definition of data models being dependant on the communicational necessities of the tools.

6.2.6 Scalability

6.2.6.1 Finnish pilot

Support-for-Future-Record-Amount

As an IOT solution vendor I want my measurement data storage solution to support **85 billion records** ($85*10^9$) to be able to contain the estimated needed amount of records for the future.

Priority: 4 (Essential)

Support-for-Future-Storage-Space

As an IOT solution vendor I want my measurement data storage solution to support **4 terabytes** ($4*10^{12}$ bytes) to be able to contain the estimated needed amount of data for the future.

Such a "low" figure is made possible by usage of Timescale aggregation, compression and purging rules.

Priority: 4 (Essential)

Support-for-Future-Data-Ingestion-Speed

As an IOT solution vendor I want my measurement data storage solution to support a data ingestion speed of **4.000 records / second** to support future needs.

This capacity should be enough also for current bulk loading (Write IOPS).

Priority: 4 (Essential)

Database-Scaling

As an IOT solution vendor I want to be possible to scale the database performances smoothly without migrating the data in order to avoid long service interruptions.

Priority: 4 (Essential)

6.2.6.2 Italian pilot

It must be designed to handle increasing loads seamlessly, ensuring scalability to accommodate a growing number of users and data volume, even though the number of customers is not expected to grow for this specific project. The system architecture must facilitate easy expansion and integration of new features while maintaining responsiveness and reliability, preparing the platform for real-world scenarios with a growing user base.

6.2.6.3 Slovenian pilot

The scalability requirements are the same as in the IT pilot.

6.2.6.4 Spanish pilot

Although the scope of the Spanish pilot in STREAM is delimited, the system should be able to support broader sets of elements and data, e.g. expanding the sGRID scope to a bigger network, managing more flexible assets and members in sFLEX/sENC, supporting a higher number of bids in sMART. The underlying systems for data management should assist the tools in these tasks.

6.2.7 Other requirements

6.2.7.1 Finnish pilot

Costs-Aspects

As an IOT solution vendor I want my measurement data storage solution to be cost effective to be sustainable from a business point of view.

Priority: 4 (Essential)

Reasonable-Storage-Costs

As an IOT solution vendor I want my measurement data storage solution to have low enough storage costs to be feasible from a business point of view.

The exact number is a company confidential value.

Priority: 4 (Essential)

High-Data-Compression-Rate

As an IOT solution vendor I want my measurement data storage solution to have high enough data compression rate to be feasible from a business point of view.

90% is a good target.

Priority: 4 (Essential)

Reasonable-Write-Read-Costs

As an IOT solution vendor I want my measurement data storage solution to have low enough data writing and reading costs to be feasible from a business point of view.

Priority: 4 (Essential)

Open-Source

As an IOT solution vendor I want my measurement data storage solution to be open source not to be locked to one specific vendor.

Priority: 4 (Essential)

Widely-Used-Tool

As an IOT solution vendor, I want my measurement data storage solution to have a big user base to be confident that the tool will be supported and developed in the future.

Priority: 4 (Essential)

6.2.7.2 Italian pilot

No other requirements are given.

6.2.7.3 Slovenian pilot

No other requirements are given.

6.2.7.4 Spanish pilot

Contractual agreements must be signed with the data owners (FSPs, DSO, etc.) explicitly allowing the use of their data for the purposes of the project. The use of the data must be compliant with GDPR.