Ref. Ares(2021)8022613 - 31/12/2021

interconnect

interoperable solutions connecting smart homes, buildings and grids

WP4 – Smart Grids Framework for an Interoperable Energy System

D4.2

Technical Specification of DSO Interface

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant agreement No 857237

DOCUMENT INFORMATION

DOCUMENT	D4.2 – Technical Specification of DSO Interface
TYPE	Report
DISTRIBUTION LEVEL	Public
DUE DELIVERY DATE	30/09/2021
DATE OF DELIVERY	31/12/2021
VERSION	V2.2
DELIVERABLE RESPONSIBLE	E-Redes
AUTHOR (S)	João Falcão (E-Redes) Carlos Damas Silva (E-Redes) Diogo Silva (E-Redes) Mário Pascoal Pereira (E-Redes) Rafael Oliveira Rodrigues (EDP Labelec) Manuel Pio Silva (EDP Labelec) Bernardo Almeida (E-Redes) José Eduardo Sousa (E-Redes) Carlos Cândido (E-Redes) José Miguel Cruz (E-Redes) Alexandre Lucas (INESC TEC) Fábio Coelho (INESC TEC) Gil Sampaio (INESC TEC) Mojtaba Eliassi (3E) Antonio Kung (TRIALOG) Olivier Genest (TRIALOG)
OFFICIAL REVIEWER/s	Andraž Andolšek (cyberGRID)

DOCUMENT HISTORY

VERSION	AUTHORS	DATE	CONTENT AND CHANGES
1.0		17/12/2021	Version submitted for Internal Review
1.1	Andraž Andolšek	22/12/2021	Review

DISCLAIMER:

EXECUTIVE SUMMARY

H2020 InterConnect project WP4 sets the ambition to design and implement a standard DSO Interface, which is an API interface that enables and allows the communication between the DSO to market platforms and entities.

The innovation brought by the DSO interface is very important to the future of flexibility provisioning to the energy system and infrastructure. This system allows for semantic interoperable mechanisms to take place at the energy distribution level, by connecting downwards to the household devices in terms of data gathering and operation of devices, to foster an open bilateral data perspective from the DSO and Service providers that can exchange data to allow for a more efficient and reliable system operation, and to integrate the consumer into the energy value chain. The DSO interface will act as one solution that will support multiple scenarios, multiple interaction, with multiple entities, raising the connectivity levels of the entire energy ecosystem.

This deliverable sets the ground for the development and implementation of the DSO interface, following the functional specification carried out in the previous deliverable, *D4.1 Technical specification of DSO standard interface application*, of this work package, which detailed the functional specification of DSO standard interface application.

The document is divided into 3 core parts, which are:

- DSO Interface requirements, leveraging InterConnect interoperability framework
- Data exchange requirements
- Operating environment requirements and implementation

Furthermore, the DSO interface must be replicable to the partners of the project, which show interest in using this software in their InterConnect pilot, therefore it's set as a challenge to cope with the multitude of existing systems, not often standard in terms of communication and technologies amongst DSO companies. Being so, the DSO interface should focus on the innovation and enablement of new services, such as flexibility as observability of the grid, to demonstrate the potential to unlock innovation to smart grids while leveraging the proposed InterConnect interoperability framework, aiming at using the SAREF ontology for enabling interaction with market entities.

This document starts by specifying the use cases that will be supported by the DSO interface, subsequently deriving the core modules that shall exist to enable the operation of these use cases, with regards to data storage, analytics, communication, and operational modules.

Then it focusses on the first conceptual layer of the system, concerning the communication with external entities/platforms/systems, both considering current and proven methods to perform the communication – REST APIs – and the communication though the InterConnect interoperability framework. Having both ways to perform this communication will, on one side, increase the DSO Interface flexibility in adopting new services and communication with other entities, but also to build the baseline for the integration of these new services into the IFA. In this case, the DSO interface is seen as one interface directly connecting and registered to the

Service Store (WP5), performing this communication via the Knowledge Engine with SAREF based messages.

Lastly, the basis for the implementation of the interface is set, detailing the conditions for this the system deployment in a real scenario.

It's worth to note that the current version of this deliverable is as a specification document that will be complemented with a later update after the implementation phase of the DSO interface, which will be carried out by a specialized team, with regards to the technical specificities of the implementation, messaging, and chosen technologies to support the proposed DSO interface. The supporting technical documentation, will be made available at the end of T4.2, after the development has occurred, in a generic and public format.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS AND ACRONYMS

ADMS	Advanced Distribution Management System
API	Application Programming Interface
DER	Distributed Energy Resources
DSO	Distribution System Operator
DTC	Distribution Transformer Controller
FSP	Flexibility Service Provider
GIS	Geographic Information System
HEMS	Home Energy Management System
HV	High Voltage
IFA	InterConnect Interoperability Framework
LV	Low Voltage
MV	Medium Voltage
OT	Operational Technology
SAREF	Smart Applications REFerence
SCADA	Supervisory Control and Data Acquisition

1. INTRODUCTION

The Distribution System Operator (DSO) plays a key role in the development of a consumer-centric energy system, enabling new standardized flexibility products provided by smart homes, buildings, and communities.

This deliverable provides the initial technical specification and architecture of the DSO Interface, which ensures a fully interoperable and replicable interface with new marketplaces and actors.

1.1 DELIVERABLE OBJECTIVES AND STRUCTURE

This deliverable provides the technical specification and architecture of the DSO Interface, which ensures a fully interoperable and replicable interface with new marketplaces and actors.

The current version of this deliverable is as a working document that will be updated and completed with more technical details during the implementation phase of the DSO interface. This will be carried out by a specialized team, with regards to the technical specificities of the implementation, messaging and chosen technologies to support the proposed DSO Interface.

The document is structures as follows:

- Chapter 2 – Details the functional requirements to be covered by the DSO Interface, according to D4.1 and the use cases to be supported. Additionally, the architecture for the DSO interface is included, through the identification, connections, description of the main modules to be developed and implemented. In summary, this chapter covers the vision for the development and implementation of the system.
- Chapter 3 – This chapter focuses on the communication towards external systems and entities. Firstly, a communication layer design is presented, which will enable the data exchange with external parties. After that, the data exchanges happening on the DSO interface border (i.e., direct interactions between the DSO and other entities/platforms) from the previously identified use cases are explored and deeply detailed, specifying the needed REST APIs, both hosted by the DSO interface and by external entities/platforms, and the correspondent data models to be developed and implemented. Lastly, the required IFA integration process for the enablement of interoperability through SAREF is detailed to be followed by the upcoming development procedures.
- Chapter 4 – The last part of the document investigates the operational part and implementation of the DSO interface, or in other words, the internal components of the DSO interface in terms of requirements for implementation (hardware and software) and the linkage to DSO OT systems and data storage.

1.2 THE INTERFACE CONTEXT

DSOs take the role of interconnecting stakeholders within and outside the Energy System. Although energy is the main domain involved, economic and regulation context are paying more and more attention to DSOs Data Management role. This comes to fact as digitalization is reaching new boundaries regarding Energy System's control and interoperability.

Besides energy exchange, DSOs are leveraging their energy services through data operation, while providing users their consumption data, forwarding it to third parties on user's consent, managing grid flexibilities and coordinating actions regarding energy distribution with other stakeholders.

For the sake of reliability and scalability this requires an infrastructure to manage all this data and involved stakeholders. As data grows and changes in time, repositories should be kept coherent, so data intra-replication is not desirable. On the other hand, security and privacy issues on data access need to be assured and continuously assessed.

Based on these requirements a DSO interface concept was agreed, managing the access to data repositories to assure the objectives described in D4.1: Allow universal access to DER, microgrids and energy communities to different flexibility market models; accommodate flexibility services designed according to the needs of the DSO; comply with GDPR; assure cyber secure implementation.

Figure 1 summarizes the DSO interface, which will be detailed in the following sections. It is based on a multi-layered software architecture pattern, namely the three-tier design pattern, which relies on a presentation layer, an application layer and the data layer.

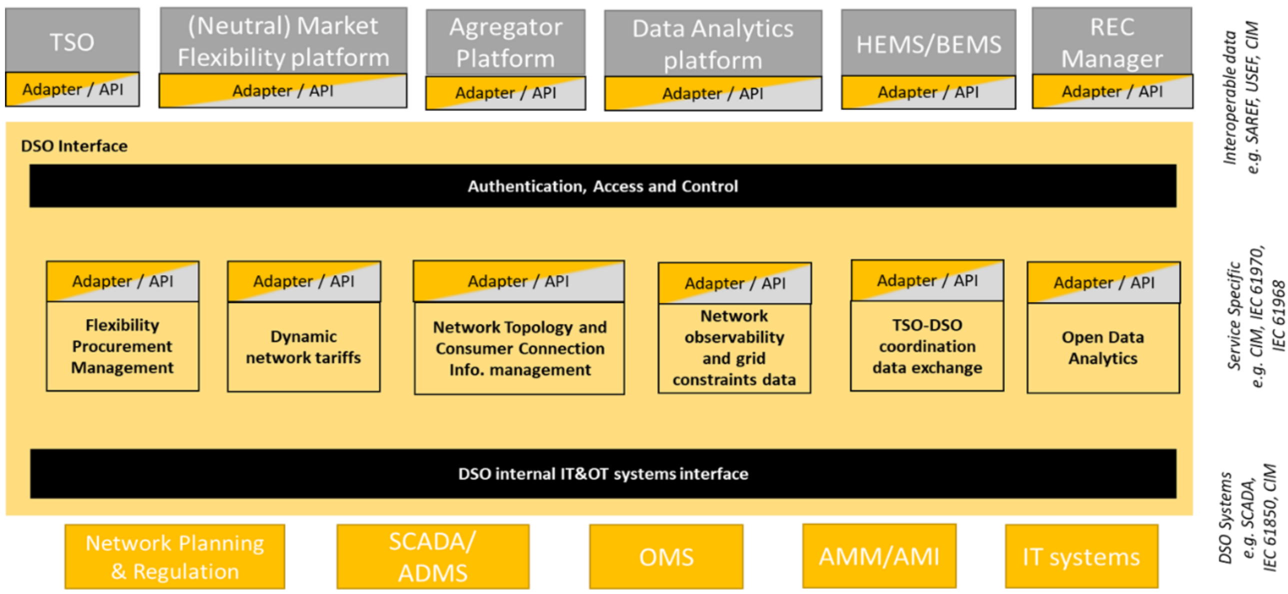


FIGURE 1 - DSO interface building blocks

The DSO interface is planned to be implemented in a cloud environment, bridging DSO's on-premise data lakes (data layer) and the several data clients (interoperability layer), through a set of APIs and adapters (application layer).

The DSO interface will be designed to implement seamless machine-to-machine communication addressing interoperability and cyber-security aspects by design.

2. FUNCTIONAL REQUIREMENTS

To accomplish the interconnected platform that will integrate the several energy and data services described in *D1.3 (System use cases for smart buildings and grids*) additional details on the functional requirements need to be defined. In this section an overview of the functional requirements for the distribution grid demonstrators use cases are described.

2.1 USE CASES TO BE SUPPORTED

As the DSO Interface will serve as a gateway between the DSO and external agents, a series of use cases within this scope were selected from D1.3. This section describes the functionalities which the DSO Interface will support:

- HLUC 5 - DSO Data Sharing
  - PUC 1 - Data Access Authorization Management
  - PUC 2 - DSO Data Sharing Services
- HLUC 10 - Flexibility management for distribution grid support
  - PUC 1 - Day-ahead grid operation planning with flexibility
  - PUC 2 - Intraday adjustment of flexibility needs
- HLUC 11 - Enhancing distribution grid observability with end user data
  - PUC1 - Using HEMS/BEMS data to support distribution network fault location
  - PUC2 - Quantification of consumers load elasticity
  - PUC3 - Assessing LV network operation status and the impact of load types

The above functionalities were selected based on the concept for the interface and its connection to the Interconnect pilots, namely the Portuguese one. However, the DSO interface is designed and conceptualized as a system that can be leveraged and expanded by other system operators who have the ambition to incorporate both the services being presented as well as new ones. Therefore, when specifying the architecture, it must be considered this implementation flexibility and the margins for new use cases to be supported on top of chosen technologies and recommend structure.

2.1.1 HLUC5 – DSO DATA SHARING

The following picture (Figure 2) shows the Sequence Diagram of PUC 1, where the DSO side systems will be integrated in the DSO Interface. The APIs/adapters that constitute the DSO Interface will have to:

- Receive requests from Service Provider to access end user data access authorization
- Request authorization to End User - consent
- Collect request validation
- Request permission change in the DSO Interface
- Return data access authorization to Service provider
- Receive requests from Service provider to access end user metering data
- Request validation of data access
- Return data access response

- Retrieve Metering Data
- Metering Data
- Return Metering Data

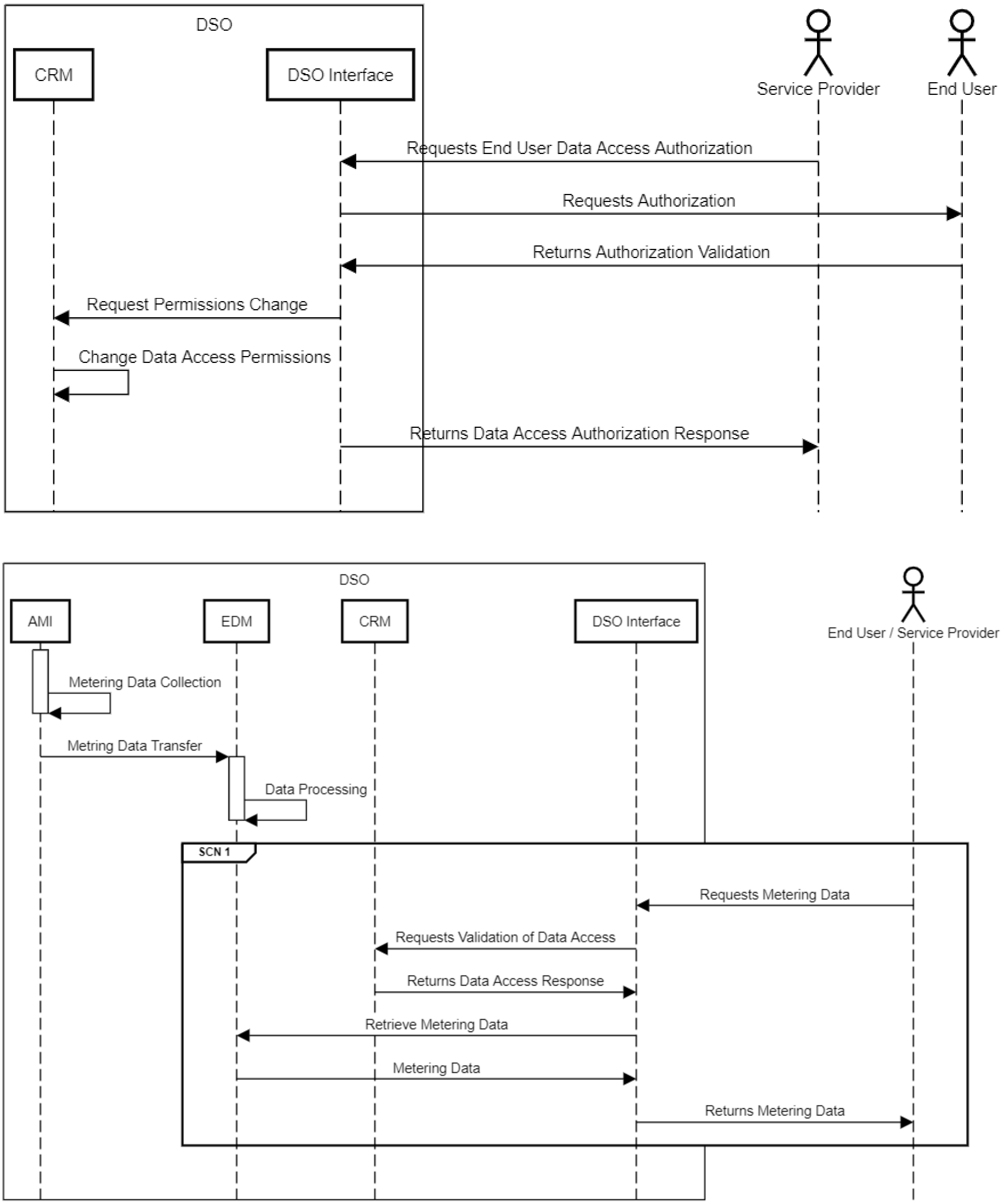


FIGURE 2 – PT PILOT HLUC5, PUC 1 SEQUENCE DIAGRAM.

The following picture shows the Sequence Diagram of PUC 2, where the DSO side systems (FMP, AMI, EDM, CRM) will be integrated in the DSO Interface. The APIs/adapters that constitute the DSO Interface will have to:

- (SCN1) Receive requests from service providers to access anonymized data
- (SCN1) Collect the necessary information from the DEP
- (SCN1) Return that information for the Service Provider
- (SCN2) Retrieve the contracted power incentives from DEP
- (SCN2) Send contracted power incentives to Service Provider/End user
- (SCN2) Receive the Service Provider/End user response (acceptance or rejection)
- (SCN2) Update information of user preferences on DEP
- (SCN3) Receive requests from service providers to access flexibility historical data
- (SCN3) Collect the necessary information from the DEP
- (SCN3) Return that information for the Service Provider

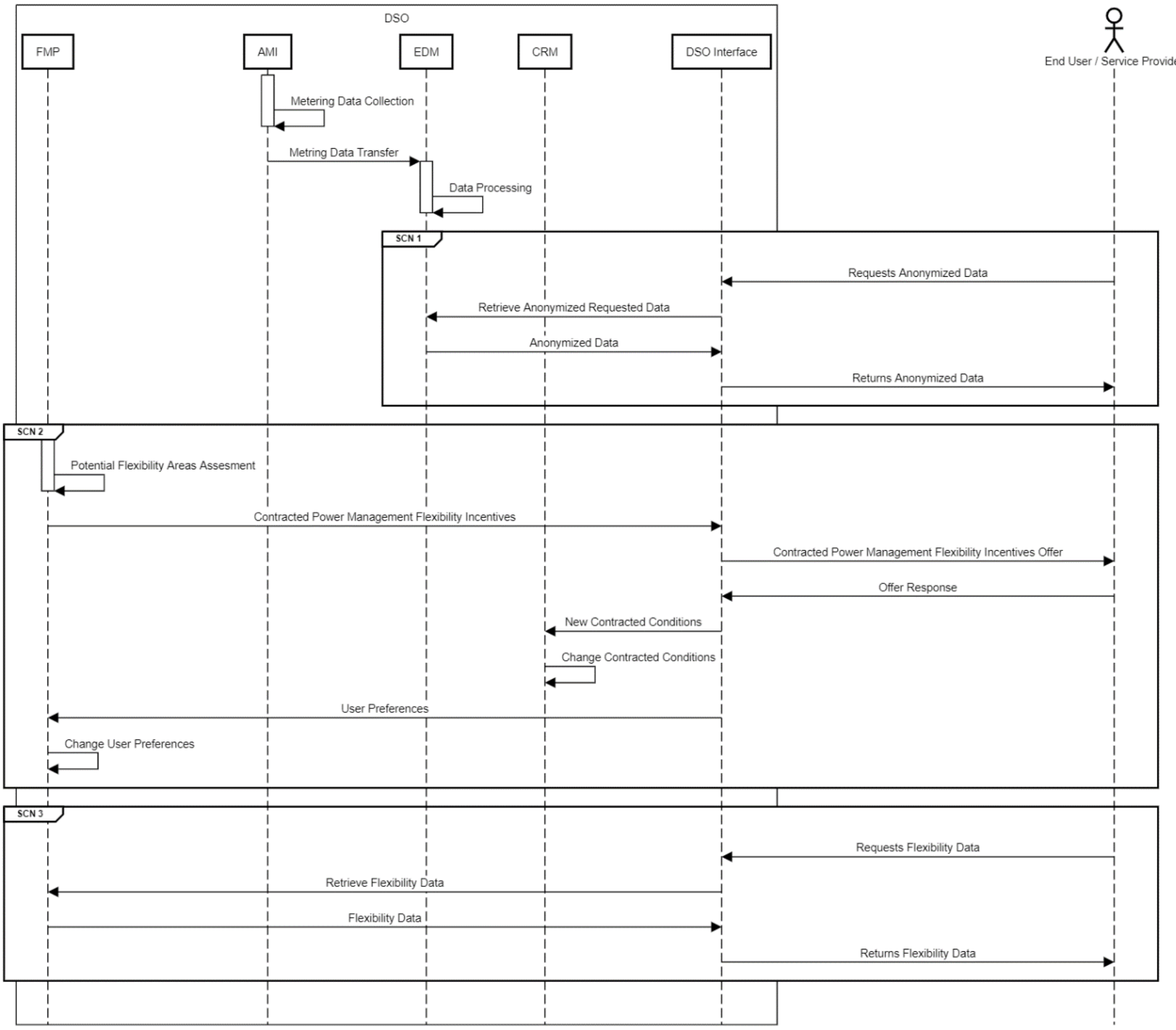


FIGURE 3 – PT PILOT HLUC5, PUC 2 SEQUENCE DIAGRAM.

2.1.2 HLUC10 – FLEXIBILITY MANAGEMENT FOR DISTRIBUTION GRID SUPPORT

The following picture (Figure 4) shows the Sequence Diagram of PUC 1, where the DSO side systems (FMP) will be integrated in the DSO Interface. The APIs/adapters that constitute the DSO Interface will have to support the following interactions in a continuous form:

- (SCN1) Receive Flexibility needs from the DSO Interface
- (SCN1) Send Flexibility needs to Service Providers
- (SCN2) Receive Requests of Flexibility needs from Service Providers
- (SCN2) Check Flexibility needs with the DSO Interface
- (SCN2) Send Flexibility needs to Service Providers
- Receive Flexibility offers from Service Providers
- Receive Flexibility activation plan from the DSO Interface
- Send Flexibility activation plan to Service Providers

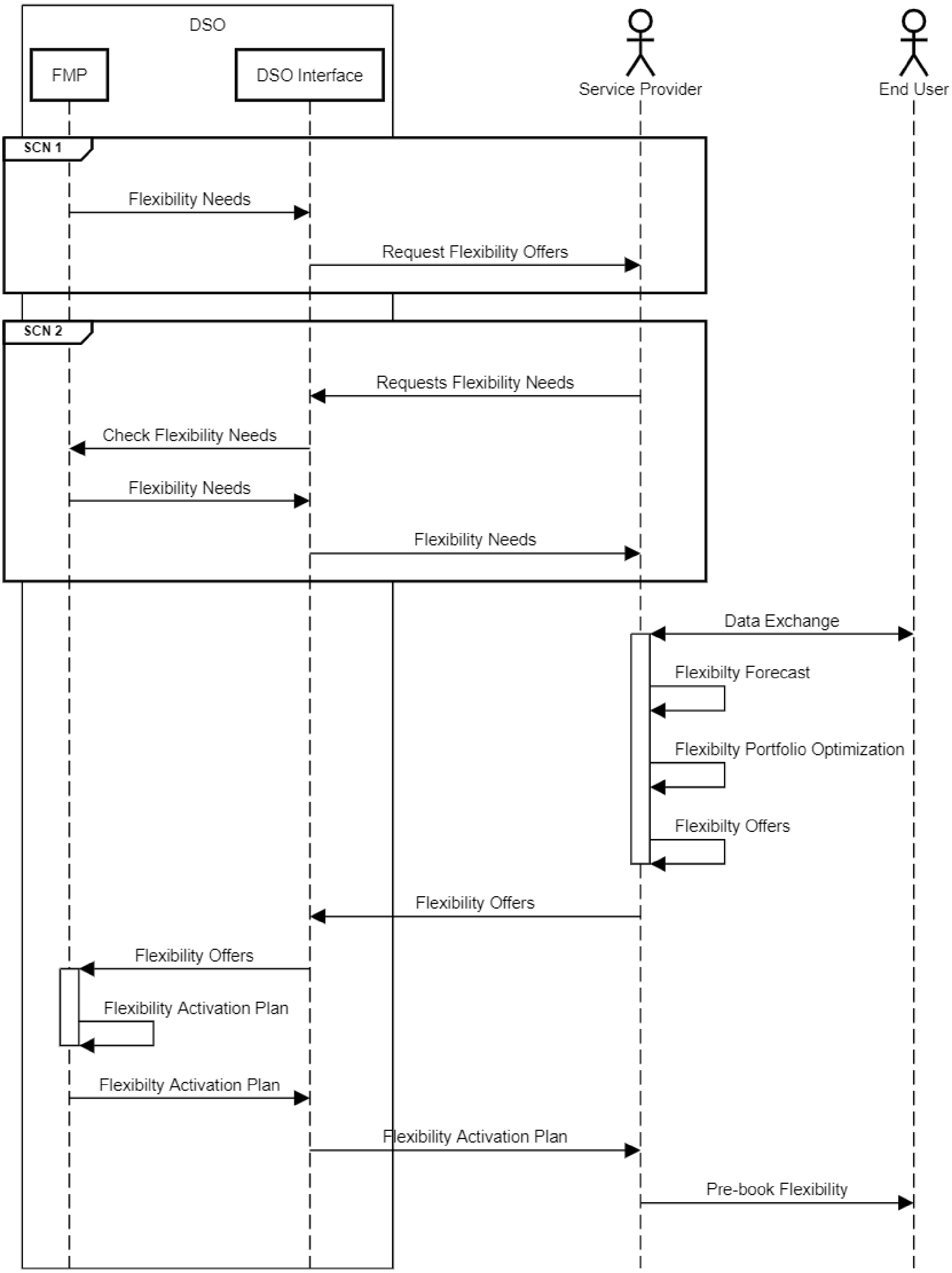


FIGURE 4 – PT PILOT HLUC10, PUC 1 SEQUENCE DIAGRAM.

The following picture (Figure 5) shows the Sequence Diagram of PUC 2, where the DSO side systems (FMP) will be integrated in the DSO Interface. The APIs/adapters that constitute the DSO Interface will have to:

- (SCN1) Receive Additional Flexibility needs from the DSO Interface
- (SCN1) Send Additional Flexibility needs to Service Providers
- (SCN2) Receive Requests of New Flexibility needs from Service Providers
- (SCN2) Check New Flexibility needs with the DSO Interface
- (SCN2) Send Additional Flexibility needs to Service Providers
- Receive Flexibility offers from Service Providers
- Receive Flexibility activation plan from the DSO Interface
- Send Flexibility activation plan to Service Providers

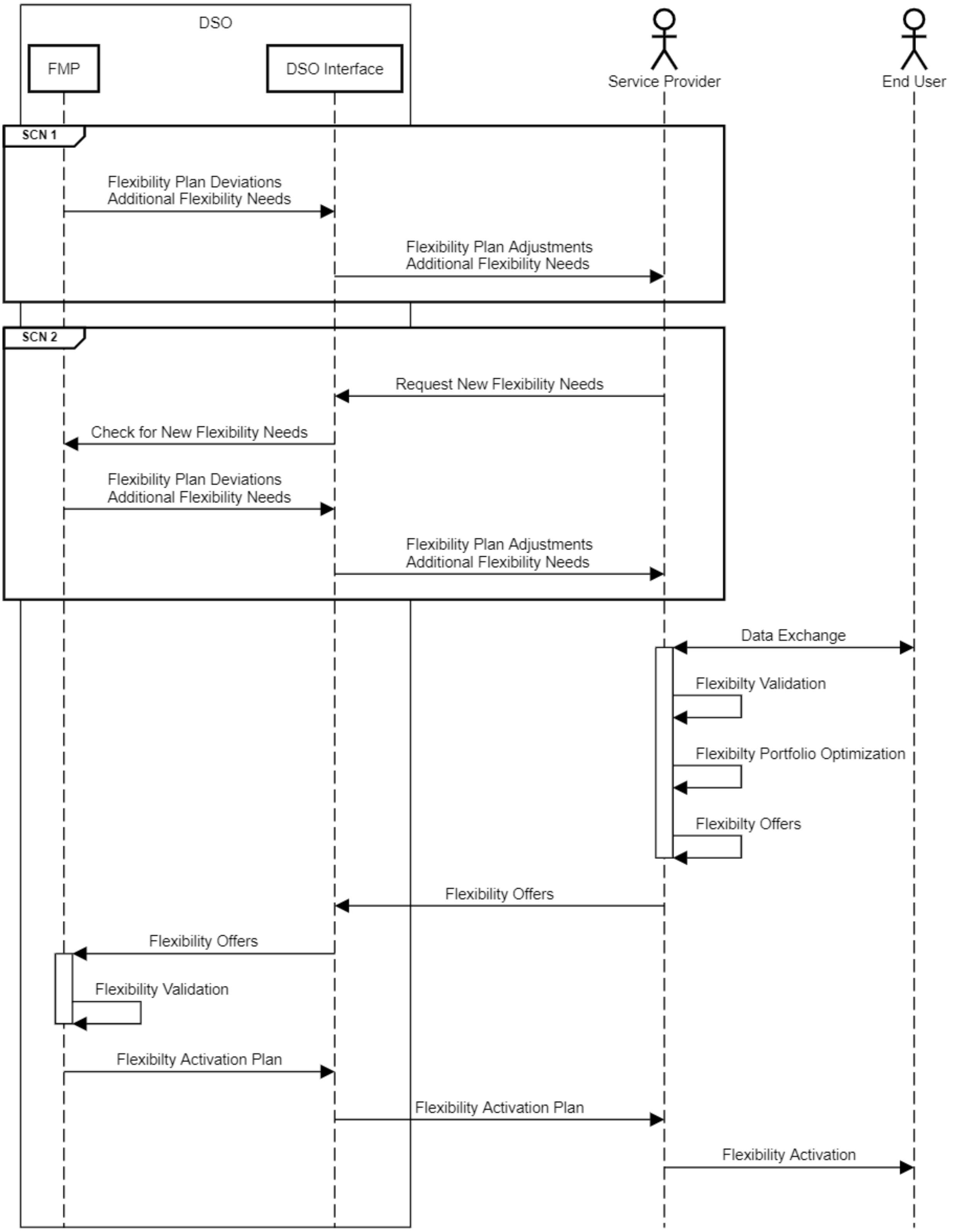


FIGURE 5 – PT PILOT HLUC10, PUC 2 SEQUENCE DIAGRAM.

In this section we will describe the functionalities to support HLUC11 and its respective PUCs 1, 2 and 3.

2.1.3 HLUC11 - ENHANCING DISTRIBUTION GRID OBSERVABILITY WITH END USER

The following picture Figure 6 shows the Sequence Diagram of PUC 1. The APIs/adapters that constitute the DSO Interface will have to:

- Send service subscription to Service Providers
- Receive Notifications of potential outages
- Receive Notification of abnormal voltages

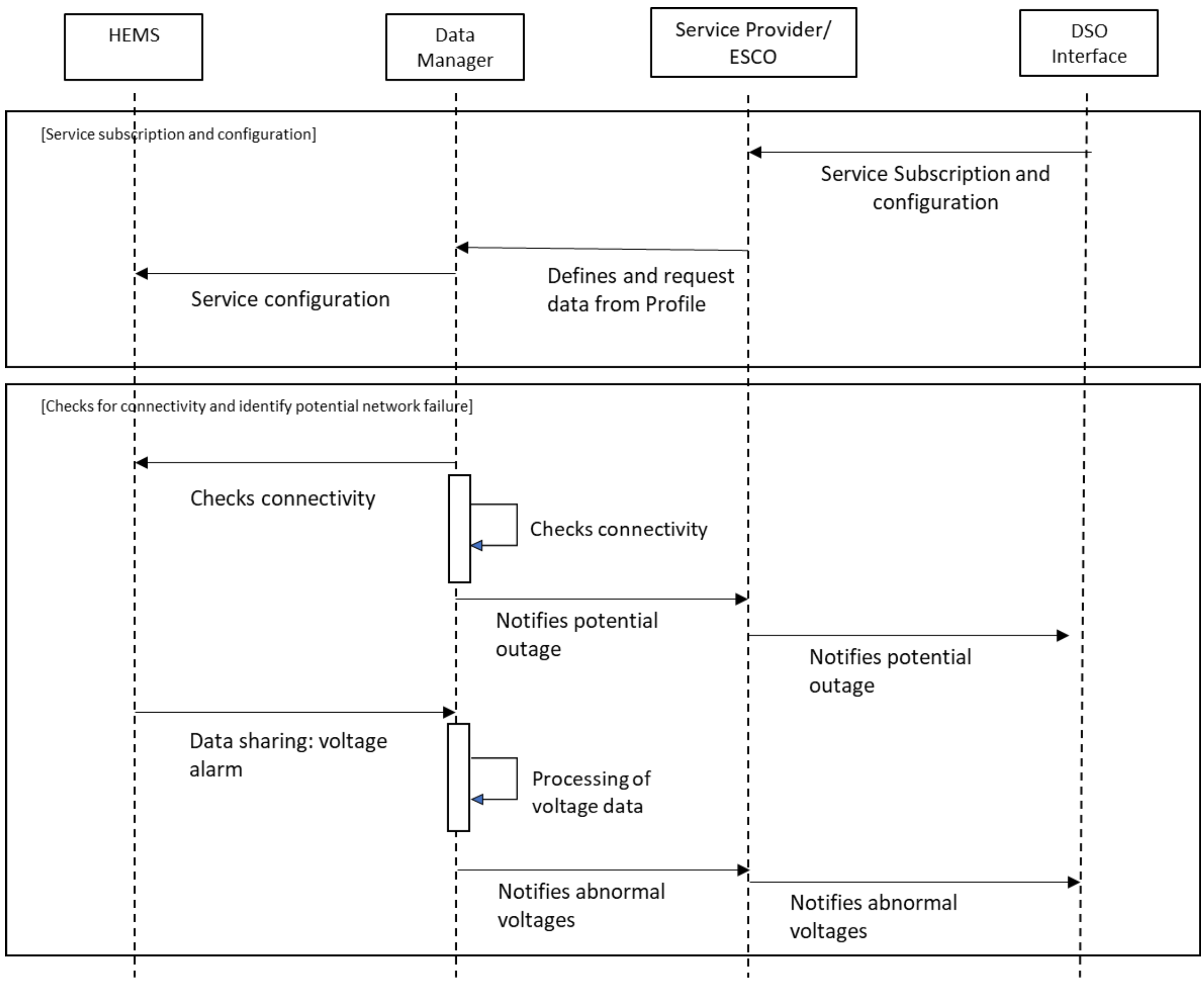


FIGURE 6 – PT PILOT HLUC11, PUC 1 SEQUENCE DIAGRAM.

The following picture (Figure 7) shows the Sequence Diagram of PUC 2 The APIs/adapters that constitute the DSO Interface will have to:

- Send service subscription to Service Providers
- Receive Elasticity of the customer

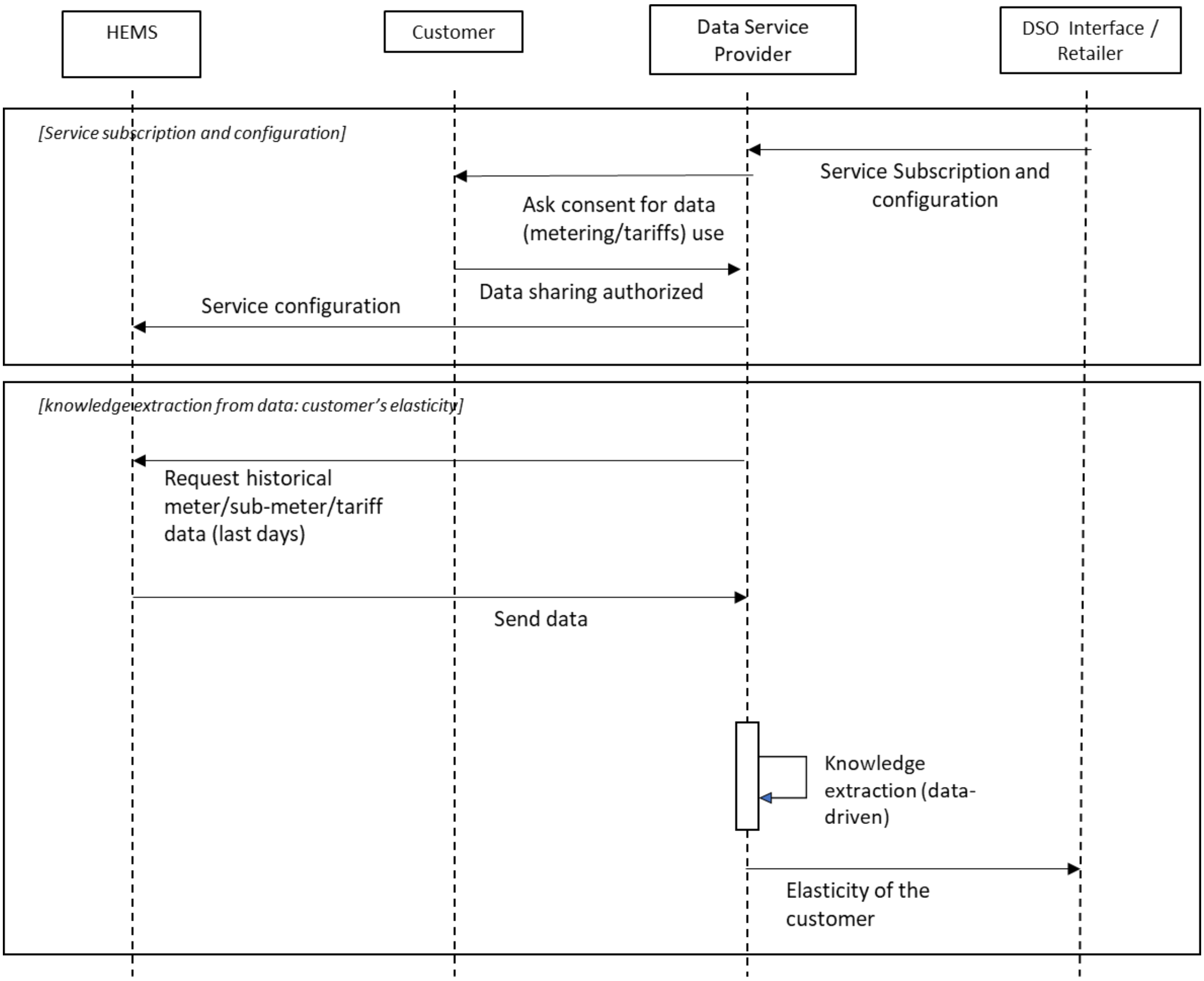


FIGURE 7 – PT PILOT HLUC11, PUC 2 SEQUENCE DIAGRAM.

The following picture (Figure 8) shows the Sequence Diagram of PUC 3. The APIs/adapters that constitute the DSO Interface will have to:

- Send service subscription to Service Providers
- Receive raw data
- Receive processed data

The difference between raw and processed data is that the first one sends the HEMS generated data directly to the DSO, while the latter adds business logic to the data in order to provide specific inputs to the DSO.

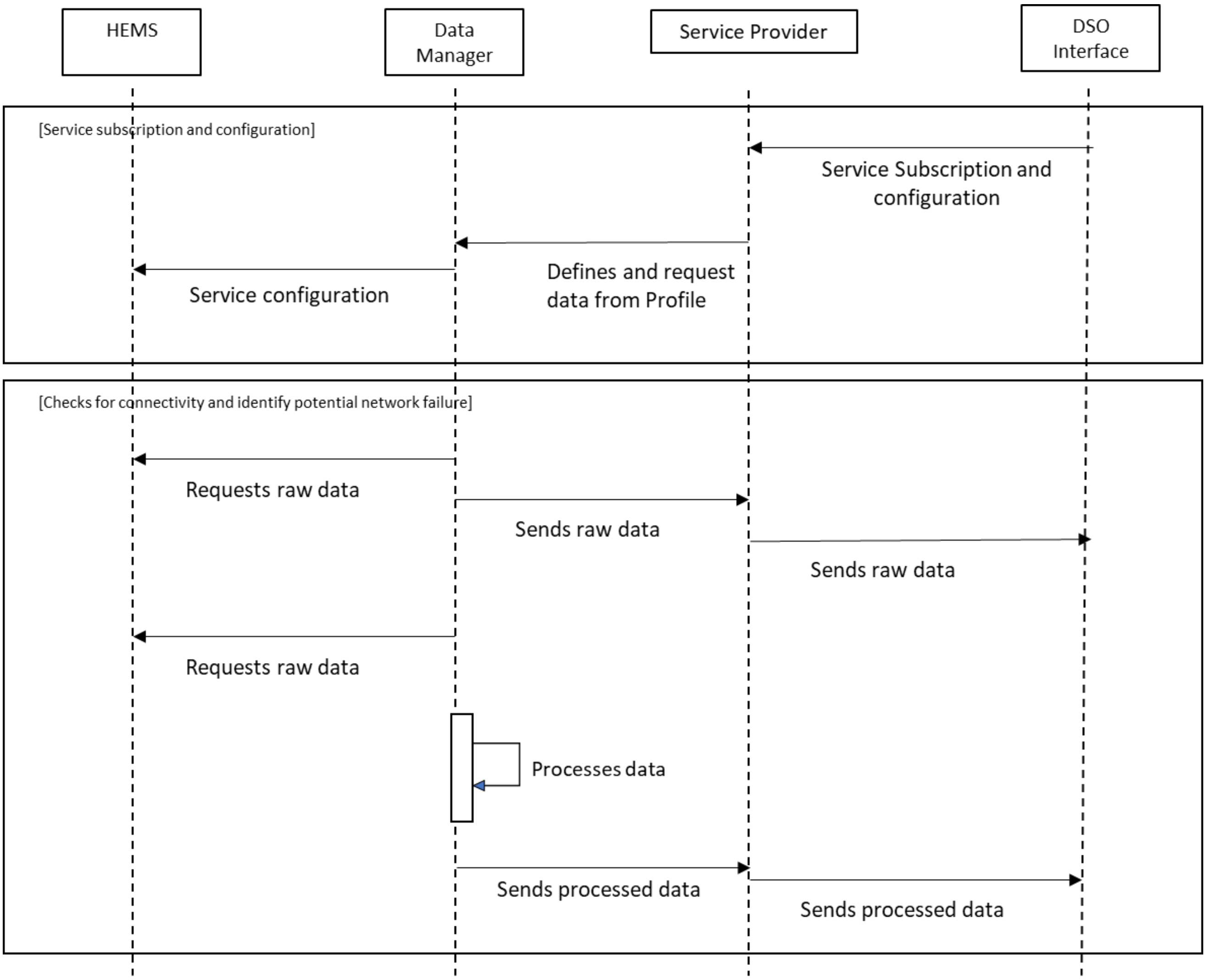


FIGURE 8 – PT PILOT HLUC11, PUC 3 SEQUENCE DIAGRAM.

2.2 FUNCTIONAL ARCHITECTURE

The goal of this activity is to develop and implement the DSO Interface, which will be a cloud system that will serve as a gateway between the internal systems of the DSO and external entities such as End Users and Service Providers. This system shall be capable of fulfilling the use cases stated in 2.1, by developing and implementing the necessary modules and information exchange mechanisms within a cloud environment. In Figure 9, the architecture of the DSO interface is detailed.

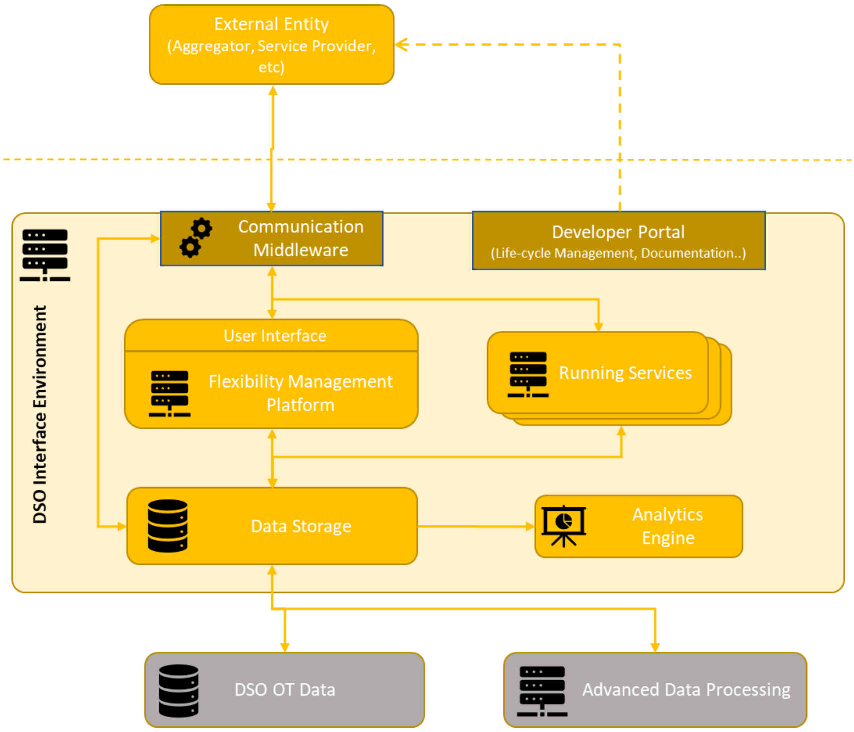


FIGURE 9 - DSO INTERFACE ARCHITECTURE

2.2.1 MODULES AND SERVICES

TABLE 1 - DSO INTERFACE MODULES

Modules	Functions	Potential technologies
Communication Middleware	- API Manager - Message Adaptations - Message Routing - Registration, Authentication, and Access Policies - InterConnect Interoperability Framework compatibility	- Apigee / Azure API Manager / WSO2 Together with - Middleware app: Python, Node.js
Developer Portal	- API Documentation - API Versioning - Traditional developer tools (e.g., code template download)	SwaggerHub, Tyk, Mulesoft Anypoint, WSO2 API Manager
Flexibility Management Platform	- FMP related activities (see 2.3.1) (HLUC 10 - PUC 1 & 2 and HLUC 5 – PUC 2)	- Webapp: React, Angular, Django Together with - backend service: Python, Nodejs, Java
Running Services	- Observability Service (HLUC 11 – PUC 1 & 2 & 3) - Data Exchange Service (HLUC 5 – PUC 1) - Additional Services to further extend the scope of DSO Interface operation	- Webapp (part of the previous one): React, Angular, Django - backend service: Python, Nodejs, Java
Data Storage	Set of relational DBs to store: - Running modules related data - Metering data - Registration, Authentication and Access Policies Data - Advanced analytics data - Logging and Monitoring of Requests	- SQL Server - MySQL - Postgresql

Analytics Engine	Data visualization of analytics regarding: - Flexibility - Observability - Data Exchange	- Kibana - Grafana - EKL Stack - PowerBI Server
DSO OT Data	Storage for structured and unstructured data from DSO internal OT Systems, namely metering and grid data	- Azure Data Lake Storage
Advanced Data Processing	Dedicated infrastructure to perform advanced analytics through algorithms	- Databricks - Apache Spark - Python

The DSO Interface Environment, which architecture is represented in Figure 9, has the requirement to hold the different components that will take part in this project. This environment will be deployed in the Cloud and all the required hardware will be provided by this environment. The DSO Interface layers and components are described in the following subsections.

2.2.2 DATA STORAGE

This layer will act as the point of connection for accessing DSO OT information, to avoid direct interaction with critical databases, and will be limited to relevant assets. Furthermore, to answer the other use cases, additional datasets/tables have to be developed (i.e., Flexibility DBs)

This will require input data from legacy systems, namely GIS and asset database for network modelling related data, SCADA measurements (to collect updated information regarding network assets). The AMI data correspond to load and voltage daily diagrams collected from the smart meters and from the Distribution Transformer Controller (DTC) installed at the MV/LV substation, that in addition to concentrate smart meter data also monitors the three-phase voltage at the LV bus and the active and reactive power at the MV/LV transformer. Furthermore, the DSO Database will concentrate all the data required by the tools as well as the outputs produced.

2.2.3 COMMUNICATION MIDDLEWARE

Implementation of a standard, agnostic, adaptable and modular API to link DSOs and market parties with flexibility market platforms, and other layers within the DSO Interface Environment. This approach will allow distributed communication without the need of managed platforms. The APIs will be publicly available for any stakeholder to adopt them or to develop new APIs concerning new services.

The Communication middleware also includes a support service to manage registration and access policies of entities who want to interact (send or receive requests to or from the DEP) with the DSO interface.

The Authentication service will handle the appropriate authentication mechanism, for example token-based authentication, to allow an application to access the several APIs which the platform will provide. It will also verify if a certain entity can request specific type of information.

2.2.4 FLEXIBILITY MANAGEMENT PLATFORM

The Flexibility Management Platform provides a central service between buyers and sellers to facilitate the communication and coordination of all processes related to the procurement of capacity and/or energy bids. The platform will be responsible for the registration, aggregation, and dispatch of the flexible resources.

2.2.5 ANALYTICS ENGINE

In this layer, the data gathered in the Storage layers can be processed to retrieve more value from it. This component will access both the DSO interface operational and DSO OT data in order to build interesting statistics from it.

2.2.6 RUNNING SERVICES

This layer is reserved for running services relevant to the Project and of interest to be further added to the DSO interface.

2.2.6.1DATA SHARING (HLUC5)

To enable the existence of a flexibility market and the provision of energy services by third party entities the DSO must create the mechanisms cable of provide to all the interested stakeholders the needed information for these services to surge and at the same time promote an open data energy ecosystem ensuring the compliance with GDRP and protection of DSO business information.

In the InterConnect project this data sharing services will mainly focus on creating and ensuring the existence of channels to provide clients metering both individual and aggregated. The provision of individual data is subjected to clients consent and the authorization management will be implemented trough the DSO existent channels (that are already used for the client to download their data and manage their account setting included give and revoke data access authorizations). Then API will be created so can service providers or other third-party entities can request and have access to client metering data.

One of the open data sharing services is for the DSO to also provide information that may help service providers designing new services. To that extent, a list of anonymized or aggregated data can be requested trough APIs which data can be from a specific area or type of population and the possible options will be available upon request trough the API.

For the flexibility service providers, historical data of the flexibility traded, and flexibility DSO needs for specific areas will also be available on the data sharing services so new service providers can access in advance the potential of a determined area.

2.2.6.2OBSERVABILITY (HLUC11)

Although most of countries already reached 100% of their smart meters roll-out, in most of the countries the base technology does not allow real time supervision and fast data transmission since most of them have a PLC-base smart metering infrastructure.

Since most of clients have an internet connection and is starting to be more common the use of connect smart home devices and appliances, one of the Interconnect goals is finding how the DSO can leverage from these devices to increase their observability of the low voltage grid.

In the scope of Interconnect and trough the APIs, each service provider will offer for each type of device (single appliance or aggregated trough HEMS), the DSO trough a kind of subscription service and standardized data structure can obtain information about voltage values and profiles or get to know which areas have devices that became offline (for fault detections). This information is then used by the correlation rules implemented on the ADMS and assist the DSO operations.

Information about flexibility potential of individual loads can also be included on the observability services.

3. DATA EXCHANGE REQUIREMENTS

3.1 COMMUNICATION MIDDLEWARE DESIGN

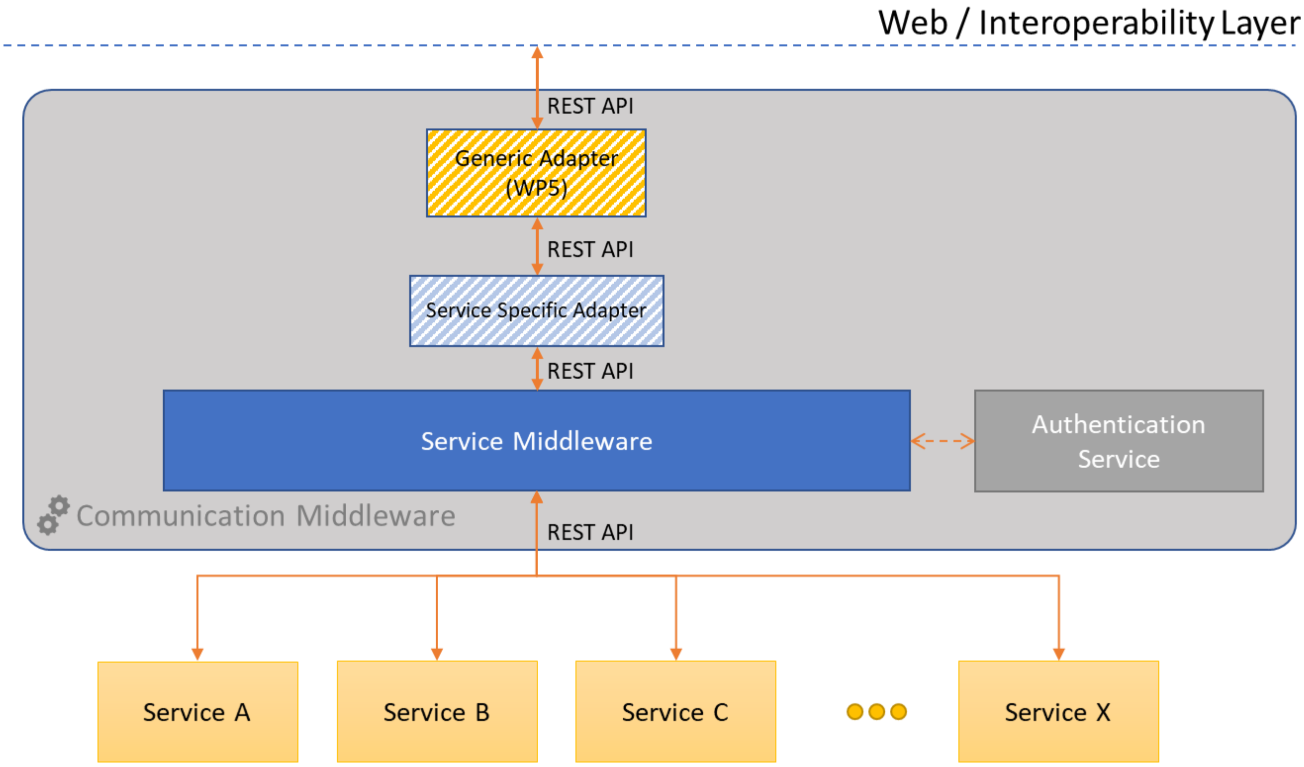


FIGURE 10 - COMMUNICATION MIDDLEWARE DESIGN

The Communication Middleware, which is represented in Figure 10, will manage the communications between the DEP and external entities by allowing communication through REST APIs and the Interconnect interoperability layer.

One core component of the Communication Middleware is the Service Middleware, which will handle the routing of the requests to the appropriate internal running services. This design also includes the usage of an API Manager to manage the request's load, endpoints, and obtain real time insight through useful analytics on the status of the service.

The interaction with the Interconnect Interoperability Layer will be further detailed in 3.4. and the messaged specifications in 3.2 will serve as the baseline for the SAREFization process and integration with the IFA.

3.2 COMMUNICATION AND APIS REQUIREMENTS

The use cases from 2.1 were analysed in order to extract the data exchanges happening on the DSO interface border (i.e., direct interactions between the DSO and other entities/platforms), which analysis is detailed in Table 2.

TABLE 2 – USE CASE STEPS REQUIRING DATA COMMUNICATION BETWEEN THE DSO AND EXTERNAL ENTITIES

UC	Process	Step	Requester	Provider
HLUC 5	Authorization	Request end user data access authorization	Service Provider	DSO
HLUC 5	Authorization	Return data access authorization response	DSO	Service Provider
HLUC 5	Meter Data Sharing	Request Metering data	End User /Service Provider	DSO
HLUC 5	Meter Data Sharing	Return Metering Data	DSO	End User /Service Provider
HLUC 5	Anonymized data sharing	Request Anonymized Data	End User /Service Provider	DSO
HLUC 5	Anonymized data sharing	Returns Anonymized Request Data	DSO	End User /Service Provider
HLUC 5	Flexibility Data Requests	Request Flexibility Data	End User /Service Provider	DSO
HLUC 5	Flexibility Data Requests	Returns Flexibility Data	DSO	End User /Service Provider
HLUC10	Day ahead	Request Flexibility Offers	DSO	Service Provider
HLUC 10	Day ahead	Request Flexibility Needs	Service Provider	DSO
HLUC 10	Day ahead	Flexibility Needs	DSO	Service Provider
HLUC 10	Day ahead	Flexibility Offers	Service Provider	DSO
HLUC 10	Day ahead	Flexibility Activation Plan	DSO	Service Provider
HLUC 10	Intraday	Flexibility Plan Adjustments Additional Flexibility Needs	DSO	Service Provider
HLUC 10	Intraday	Request New Flexibility Needs	Service Provider	DSO
HLUC 10	Intraday	Flexibility Plan Adjustments Additional Flexibility Needs	DSO	Service Provider
HLUC 10	Intraday	Flexibility Offers	Service Provider	DSO
HLUC 10	Intraday	Flexibility Activation Plan	DSO	Service Provider
HLUC 11	Request	Requests Area flexibility potential to DSO as a service	Third party service requester	DSO
HLUC 11	Request	Requests data to be able to inputs complete datasets to the service provider	DSO	Data service provider
HLUC 11	Requests/Subscribes	Sends data to the service provider or subscribes service	DSO	Service provider

3.2.1 DATA SHARING SERVICE (HLUC 5)

3.2.1.1 INCOMING DSO INTERFACE REQUESTS

TABLE 3 - HLUC 5 - INCOMING DSO INTERFACE REQUESTS

PUC	Requester	Request	Response	Defined API
1	Service Provider	Requests End User Data Access Authorization	Returns Data Access Authorization Response	**Authorization**
1	End User / Service Provider	Requests Metering Data	Returns Metering Data	**Metering Data**
2	End User / Service Provider	Requests Anonymized Data	Returns Anonymized Data	**Metering Data**
2	End User / Service Provider	Requests Flexibility Data	Returns Flexibility Data	**Flexibility Data**

3.2.1.1.1 API DEFINITION

3.2.1.1.1.1 AUTHORIZATION

API to get authorization to access the data to the end user.

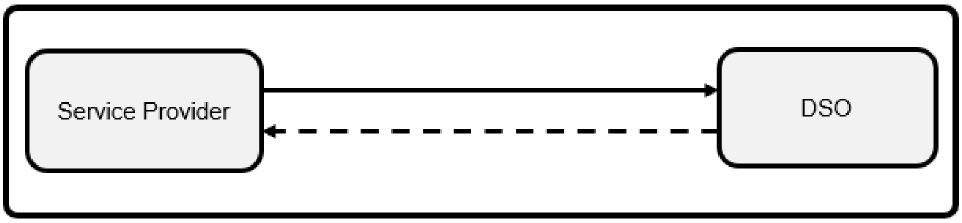
<table>
<tr><td>API</td><td>https://api.dso.com/1.0/
This are generic API addresses which will then be adapted upon implementation</td></tr>
<tr><td>Category</td><td>https://api.dso.com/1.0/Data/Authorization</td></tr>
<tr><td>Method</td><td>GET https://api.dso.com/1.0/ Data/Authorization</td></tr>
<tr><td>Producer</td><td>DSO</td></tr>
<tr><td>Receiver</td><td>Service Provider</td></tr>
<tr><td>Description</td><td>Service Provider sends a request to DSO to obtain access to data of a specific End User. DSO respond with success or unsuccess.</td></tr>
<tr><td>Data</td><td><table><tr><th>Field</th><th>Example</th></tr><tr><td>End User</td><td>U123456</td></tr><tr><td>Start Date</td><td>2021/09/27 10:00</td></tr><tr><td>End Date</td><td>2021/09/27 11:00</td></tr></table></td></tr>
<tr><td>Request</td><td>As need</td></tr>
<tr><td>Response</td><td>Authorization Token</td></tr>
</table>

3.2.1.1.1.2 METERING DATA

API to get metering data or anonymized data (with a parameter specifying the type of request).

Architecture Design

Logical Architecture Diagram (Request Flow)



Configuration

<table>
<tr><td>API</td><td>https://api.dso.com/1.0/</td></tr>
<tr><td>Category</td><td>https://api.dso.com/1.0/Data/Metering</td></tr>
<tr><td>Method</td><td>GET https://api.dso.com/1.0/Data/Metering</td></tr>
<tr><td>Producer</td><td>DSO</td></tr>
<tr><td>Receiver</td><td>Service Provider</td></tr>
<tr><td>Description</td><td>Service Provider sends a request to DSO to get metering data or anonymized data. DSO respond with metering data of an End User or a Zone.</td></tr>
<tr><td>Condition</td><td>Array of:
<table>
<tr><th>Field</th><th>Example</th></tr>
<tr><td>Type of Data
(User/Anonymized)</td><td>User</td></tr>
<tr><td>ID (User, Zone)</td><td>U123456</td></tr>
<tr><td>Authorization Token</td><td>49d259db-755c-4929-8410-15ede30dbc2d</td></tr>
<tr><td>Granularity [1-15min; 2-hourly; 3-daily)</td><td>1</td></tr>
<tr><td>Start time (ISO format)</td><td>2021/09/27 10:00</td></tr>
<tr><td>End time (ISO format)</td><td>2021/09/27 11:00</td></tr>
</table></td></tr>
<tr><td>Request</td><td>As need</td></tr>
<tr><td>Response</td><td>Metering Data
Array of:
<table>
<tr><th colspan="2">Field</th><th>Example</th></tr>
<tr><td colspan="2">User/Zone</td><td>U123456</td></tr>
<tr><td rowspan="3">Metering Array
(divided by the specific granularity)</td><td>Time</td><td>2021/09/27 10:00</td></tr>
<tr><td>Measurement</td><td>890</td></tr>
<tr><td>Unit</td><td>kW</td></tr>
</table></td></tr>
</table>

3.2.1.1.1.3 FLEXIBILITY DATA

API to get general historical Flexibility Data so Service Providers can evaluate the potential of providing flexibility on that given zone.

<table>
<tr><td>API</td><td>https://api.dso.com/1.0/</td></tr>
<tr><td>Category</td><td>https://api.dso.com/1.0/Data/Flexibility</td></tr>
<tr><td>Method</td><td>GET https://api.dso.com/1.0/Data/Flexibilty</td></tr>
<tr><td>Producer</td><td>DSO</td></tr>
<tr><td>Receiver</td><td>Service Provider</td></tr>
<tr><td>Description</td><td>Service Provider sends a request to DSO to get historical or forecasted information about flexibility needs on a given area in order to evaluate the flexibility provision potential. DSO responds with desired information.</td></tr>
<tr><td>Condition</td><td>Array of:
<table>
<tr><th>Field</th><th>Example</th></tr>
<tr><td>Zone ID</td><td>Z123456</td></tr>
<tr><td>Start time</td><td>2021/09/27</td></tr>
<tr><td>End time</td><td>2021/10/27</td></tr>
</table></td></tr>
<tr><td>Request</td><td>As need</td></tr>
<tr><td>Response</td><td>Flexibility Historical or Forecasted Data:
<table>
<tr><th>Field</th><th>Example</th></tr>
<tr><td>Zone ID</td><td>Z123456</td></tr>
<tr><td>Dates</td><td>[2021/09/27 10:00, 2021/09/27 11:00, …]</td></tr>
<tr><td>Needs</td><td>[100, 50, …]</td></tr>
<tr><td>Measurement Unit</td><td>[kW, kW, …]</td></tr>
<tr><td>Direction (Up/Down)</td><td>[Down, Up, …]</td></tr>
<tr><td>Type of Power (Active/Reactive)</td><td>[Active, Active, …]</td></tr>
<tr><td>Eligible Clients (Meter ID)</td><td>[ID Meter1_ID Meter2, ID Meter1_ID Meter2, …]</td></tr>
</table></td></tr>
</table>

3.2.1.2 OUTGOING DSO INTERFACE REQUESTS

TABLE 4 – HLUC 5 - OUTGOING DSO INTERFACE REQUESTS

PUC	Destination	Request	Response
1	End User	Requests Authorization	Returns Authorization Validation
2	End User / Service Provider	Contracted Power Management Flexibility Incentives Offer	Offer Response

The APIs referred in 3.2.1.2 are not detailed at the time of writing of the deliverable due to its implementation within the DSO not being foreseen. These will be evaluated at a later stage of the development procedures once the necessary technical data for the communication with the customers from the project pilots is provided.

3.2.2 FLEXIBILITY SERVICE (HLUC 10)

3.2.2.1INCOMING DSO INTERFACE REQUESTS

TABLE 5 - HLUC 10 - INCOMING DSO INTERFACE REQUESTS

PUC	Requester	Request	Response	Defined API
1	Service Provider	Requests Flexibility Needs	Flexibility Needs	**Flexibility Needs (GET)**
1	Service Provider	Flexibility Offers	Acknowledgement	**Flexibility Offers (POST)**
2	Service Provider	Request New Flexibility Needs	Flexibility Plan Adjustments Additional Flexibility Needs	**Flexibility Needs (GET)**

3.2.2.1.1 API DEFINITION

3.2.2.1.1.1 FLEXIBILITY NEEDS (GET)

API used by Service Provider to get flexibility needs from DSO.

Can be used within the day ahead (to request new flexibility needs to next day) or intraday (changes to agreed offers).

<table>
<tr><td>API</td><td>https://api.dso.com/1.0/</td></tr>
<tr><td>Category</td><td>https://api.dso.com/1.0/Flexibility/Needs</td></tr>
<tr><td>Method</td><td>GET https://api.dso.com/1.0/Flexibility/Needs</td></tr>
<tr><td>Producer</td><td>DSO</td></tr>
<tr><td>Receiver</td><td>Service Provider</td></tr>
<tr><td>Description</td><td>Service Provider sends a request to DSO to get flexibility needs. DSO reply with it.</td></tr>
<tr><td>Condition</td><td><table><tr><td>Field</td><td>Example</td></tr><tr><td>Start time</td><td>2021/09/27 00:00</td></tr></table></td></tr>
</table>

<table>
<tr><td></td><td><table><tr><td>End time</td><td>2021/09/28 00:00</td></tr></table></td></tr>
<tr><td>Request</td><td>As need</td></tr>
<tr><td>Response</td><td>Array of:<table><tr><th>Field</th><th>Example</th></tr><tr><td>Request ID</td><td>1</td></tr><tr><td>Needs</td><td>100</td></tr><tr><td>Measurement Unit</td><td>kW</td></tr><tr><td>Direction (Up/Down)</td><td>Down</td></tr><tr><td>Type of Power (Active/Reactive)</td><td>Active</td></tr><tr><td>Start time</td><td>2021/09/27 10:00</td></tr><tr><td>End time</td><td>2021/09/27 13:00</td></tr><tr><td>Eligible Clients (Meter ID)</td><td>[ID Meter1, ID Meter2, …]</td></tr></table></td></tr>
</table>

3.2.2.1.1.2 FLEXIBILITY OFFERS (POST)

API used by Service Provider to send to DSO flexibility offers.

Can be used within the day ahead (new flexibility offers to next day) or intraday (changes to agreed offers).

<table>
<tr><td>API</td><td>https://api.dso.com/1.0/</td></tr>
<tr><td>Category</td><td>https://api.dso.com/1.0/Flexibility/Offers</td></tr>
<tr><td>Method</td><td>POST https://api.dso.com/1.0/Flexibility/Offers</td></tr>
<tr><td>Producer</td><td>Service Provider</td></tr>
<tr><td>Receiver</td><td>DSO</td></tr>
<tr><td>Description</td><td>Service Provider sends to DSO the flexibility offers.</td></tr>
<tr><td>Condition</td><td></td></tr>
<tr><td>Data</td><td>Array of:<table><tr><th>Field</th><th>Example</th></tr><tr><td>Request ID</td><td>1</td></tr><tr><td>Offer ID</td><td>1</td></tr><tr><td>Offer</td><td>20</td></tr><tr><td>Price (€/kWh)</td><td>0,1</td></tr><tr><td>Start time</td><td>2021/09/27 10:00</td></tr></table></td></tr>
</table>

	End time	2021/09/27 13:00
	Meter IDs	[ID Meter1, ID Meter2, …]
Request	PUC1 (Day-ahead) – 1x Day PUC2 – As need in the intraday	
Response	Acknowledgement	

3.2.2.2OUTGOING DSO INTERFACE REQUESTS

TABLE 6 - HLUC 10 - OUTGOING DSO INTERFACE REQUESTS

PUC	Destination	Request	Response	Defined API
1	Service Provider	Requests Flexibility Offers	Acknowledgement	**Flexibility Needs (POST)**
1, 2	Service Provider	Flexibility Activation Plan	Acknowledgement	**Activation Plan (POST)**
2	Service Provider	Flexibility Plan Adjustments Additional Flexibility Needs	Acknowledgement	**Flexibility Needs (POST)**

3.2.2.2.1.1 FLEXIBILITY NEEDS (POST)

API used by the DSO to publish flexibility needs to the Service Provider.

Can be used within the day ahead (new flexibility offers to next day) or intraday (changes to agreed offers).

API	https://api.service-provider.com/1.0/		
Category	https://api service-provider.com/1.0/Flexibility/**Needs**		
Method	**POST** https://api.service-provider.com/1.0/Flexibility/Needs		
Producer	DSO		
Receiver	Service Provider		
Description	DSO send flexibility needs to the service provider.		
Data	Array of:		
		Field	**Example**
		Request ID	1
		Needs	100
		Measurement Unit	kW
		Direction (Up/Down)	Down

	Type of Power (Active/Reactive)	Active
	Start time	2021/09/27 10:00
	End time	2021/09/27 13:00
	Eligible Clients (Meter ID)	[ID Meter1, ID Meter2, …]
Request	PUC1 (Day-ahead) – 1x Day PUC2 – As need in the intraday (a safety time interval between the request of flexibility and the activation shall be considered between the flexibility provider and the system operator)	
Response	Acknowledgement	

3.2.2.2.1.2 ACTIVATION PLAN (POST)

API used by DSO to send to Service Provider the activation plan from the previous sent offers.

Can be used within the day ahead (new flexibility offers to next day) or intraday (changes to agreed offers).

API	https://api.service-provider.com/1.0/		
Category	https://api service-provider.com/1.0/Flexibility/**ActivationPlan**		
Method	**POST** https://api.service-provider.com/1.0/Flexibility/ActivationPlan		
Producer	DSO		
Receiver	Service Provider		
Description	DSO send activation plan to the service provider		
Condition			
Data	Array of:		
	Field		**Example**
	Request ID		1
	Offer ID		1
	Activation	Yes/No	Yes
		Quantity	20 kW
	Start time		2021/09/27 10:00
	End time		2021/09/27 13:00
Request	PUC1 (Day-ahead) – 1x Day PUC2 – As need in the intraday		
Response	Acknowledgement		

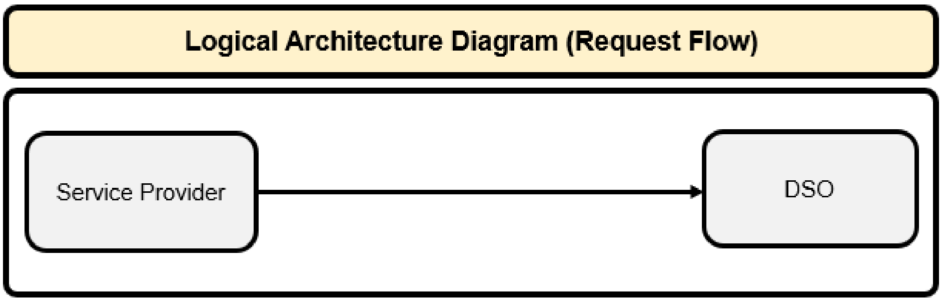
3.2.3 OBSERVABILITY SERVICE (HLUC 11)

3.2.3.1 INCOMING DSO INTERFACE REQUESTS

TABLE 7 - HLUC 11 - INCOMING DSO INTERFACE REQUESTS

PUC	Requester	Request	Response	Defined API
1	Service Provider	Posts fault notification and location (requests acknowledgement)	Acknowledgement	**Post Code_Fault (POST)**
3	Third Party	Requests flexibility power intervention areas to the DSO seeking business opportunities around a postal code	Report with recommendations/opportunities and corresponding flexibility values	**GET**

Architecture Design



3.2.3.1.1.1 API DEFINITION

The following bindings refer to Service 1 and 3 of the Observability HLUC 11 and detail the incoming methods used between the service requester and the service provider or third parties.

3.2.3.1.1.1.1 PostCode_Fault notification (Post)

The following method is used in Service 1, in which the service provider notifies the DSO of a potential fault and its location, through a POST:

API	https://inesc.interconnect.pt/GridObservability**/**
Category	https://inesc.interconnect.pt/GridObservability**/{Post Code_Fault}**
Method	**POST** https://inesc.interconnect.pt/GridObservability**/{Post Code_Fault}**
Producer	Service Provider
Receiver	DSO
Description	Notifies DSO of potential fault and location
Condition	DSO has subscribed the service and is always “listening” The conditions to be met are the settings of the DSO (min faulty Hems and min elapsed time.)

<table>
<tr><td>Data</td><td>Notification Type (potential Outage on Postal Code and corresponding nbr of potential faulty Hems)

<table>
<tr><th>Field</th><th>Example</th></tr>
<tr><td>Fault Location ID</td><td>1</td></tr>
<tr><td>Postal Code</td><td>[4405-123, 1200-000, …]</td></tr>
</table>

JSON version:
{
"Fault_Location_Identification" :
[
{
“Postal_Code”: “4405-123ID”, (string)
},
],
}</td></tr>
<tr><td>Request</td><td>As need</td></tr>
<tr><td>Response</td><td>Acknowledgement</td></tr>
</table>

3.2.3.1.1.1.2 Flexibility_Opportunities (GET)

The following method refer to Service 3 and details the methods used between a third party requesting an output (flexibility recommendation) from the DSO:

<table>
<tr><td>API</td><td>https://inesc.interconnect.pt/GridObservability/opportunities</td></tr>
<tr><td>Category</td><td></td></tr>
<tr><td>Method</td><td>GET</td></tr>
<tr><td>Producer</td><td>Third-party (e.g., aggregator)</td></tr>
<tr><td>Receiver</td><td>DSO</td></tr>
<tr><td>Description</td><td>A third-party requests the DSO for business opportunities in areas where flexibility increase would benefit the system operation, specifying zip codes of interest. The output contains opportunities (flexibility needs) in the specified zip codes and surrounding areas.</td></tr>
<tr><td>Condition</td><td></td></tr>
<tr><td>Input Data</td><td>{
"zip_codes": [</td></tr>
</table>

	"4200", "4250", "4265", "4280",] }

Output Data	{
	"opportunities": [
	{
	{
	"zip_code": "4200-013",
	"message": [
	"The number of EVs charging during 14h00 and 16h30 should increase by 6.",
	"Distributed storage devices were not found in this area."
	]
	},
	{
	"zip_code": "4260-339",
	"message": [
	"The number of EVs charging during 19h00 and 21h30 should decrease by 4.",
	"Distributed storage devices were not found in this area."
	]
	}
	}
	]
	}

Request	As need
Response	Acknowledgement

3.2.3.2OUTGOING DSO INTERFACE REQUESTS

TABLE 8 - HLUC 11 - OUTGOING DSO INTERFACE REQUESTS

PUC	Destination	Request	Response	API definition
1	Service Provider	Parameter setting (Min. nbr of faulty HEMS - Optional)	Acknowledgement	HEMS Nbr (POST)
1	Service Provider	Parameter setting (Min. Elapsed time for fault - Optional)	Acknowledgement	**HEMS Elps_Time (POST)**
1	Service Provider	Service Subscription Parameter setting (Postal Code)	Accepted/denied couldn't reach postal code or min. nbr. of HEMS	**Postal Code (POST)**
1	Service Provider	Hems status	(on/off)	**HEMS ID_Status (GET)**
1	Service Provider	Hems voltage	Last Single phase Voltage reading	**HEMS ID_Volt (GET)**
3	Service Provider	Requests disaggregated data to a Data service provider	Gets disaggregated data sets for the Ids provided	**DisaggregatedData (GET)**

3.2.3.2.1 API DEFINITION

The following bindings refer to Service 1 and 3 of the Observability HLUC 11 and details the outgoing methods used between the service requester and the service provider or third parties.

3.2.3.2.1.1 HEMS_NBR (POST)

API	https://inesc.interconnect.pt/GridObservability/
Category	https://inesc.interconnect.pt/GridObservability/{Postal Code}/ HEMS Nbr/
Method	**POST** https://inesc.interconnect.pt/GridObservability/{Postal Code}/ HEMS_Nbr/
Producer	DSO
Receiver	Service Provider
Description	Defines the minimum number of faulty hems to be considered to trigger the fault identification
Condition	
Data	Notification Type (Fault location input, Min_nbr_Hems_faulty) <table><tr><th>Field</th><th>Example</th></tr><tr><td>Min nbr HEMS faulty</td><td>*3*</td></tr></table> JSON version: *{* *"Fault_Location_Inputs" : [* *{* *"Min_nbr_Hems_faulty": "3", (integer)*

	}, *],* *}*
Request	As need
Response	Acknowledgement

3.2.3.2.1.2 HEMS ELPS_TIME (POST)

API	https://inesc.interconnect.pt/GridObservability/
Category	https://inesc.interconnect.pt/GridObservability/{Postal Code}/HEMS Elps_Time
Method	**POST** https://inesc.interconnect.pt/GridObservability/{Postal Code}/HEMS Elps_Time
Producer	DSO
Receiver	Service Provider
Description	Defines the minimum elapsed time to be considered to trigger the fault identification
Data	Notification Type (Fault location input, Min_time_of_interruption) **Field** / **Example** Minimum time of interruption / *180.00* Unit / *seconds* JSON version: *{* *"Fault_Location_Inputs" : [* *{* *"Min_time_of_interruption": "180", (float)* *"Min_time_of_interruption": "seconds"* *},* *],* *}*
Request	As need
Response	Acknowledgement

3.2.3.2.1.3 POSTAL CODE (POST)

API	https://inesc.interconnect.pt/GridObservability/
Category	https://inesc.interconnect.pt/GridObservability/**{Postal Code}**

Method	**POST** https://inesc.interconnect.pt/GridObservability**/{Postal Code}**
Producer	DSO
Receiver	Service Provider
Description	Defines the Postal code in which Hems should be monitored for fault notifications
Condition	
Data	Notification Type (Fault location input to monitor, Postal_Code) <table><tr><th>Field</th><th>Example</th></tr><tr><td>Postal Code</td><td>[“4120-233”, “4120-200”]</td></tr></table> JSON version: *{* *"Fault_Location_Inputs" : [* *{* *“Postal_Code”: “4405-123”, (string)* *},* *],* *}*
Request	As need
Response	Accepted/denied couldn’t reach postal code or min. nbr. of HEMS

3.2.3.2.1.4 HEMS ID_STATUS (GET)

API	https://inesc.interconnect.pt/GridObservability**/**
Category	https://inesc.interconnect.pt/GridObservability**/**HEMS/{HEMS ID}/Status
Method	**GET** https://inesc.interconnect.pt/GridObservability**/**HEMS/{HEMS ID}/Status
Producer	DSO
Receiver	Service Provider
Description	The DSO inquires the communication status of a specified HEMS
Condition	
Data	Notification Type (requests data on: Hems_ID), “Postal_Code”: “4405-123”, (string)
Request	As need
Response	<table><tr><th>Field</th><th>Example</th></tr><tr><td>HEMS Id</td><td>1234</td></tr><tr><td>Abnormal Voltage</td><td>Yes</td></tr></table>

<table>
<tr><td></td><td>
<table><tr><td>timestamp</td><td>"2021-12-03T15:11:20Z"</td></tr></table>
JSON Version:

*{
"Voltage_Monitoring" : [
{
"Hems_ID": "ID1234"
"Abnormal_Voltage":, Yes (boolean)
"timestamp": "2021-12-03T15:11:20Z", (timestamp)
},
]
}*

</td></tr>
</table>

3.2.3.2.1.5 HEMS ID_VOLT (GET)

<table>
<tr><td>API</td><td>https://inesc.interconnect.pt/GridObservability/</td></tr>
<tr><td>Category</td><td>https://inesc.interconnect.pt/GridObservability/HEMS/{HEMS ID}/Voltage</td></tr>
<tr><td>Method</td><td>GET https://inesc.interconnect.pt/GridObservability/HEMS/{HEMS ID}/Voltage</td></tr>
<tr><td>Producer</td><td>DSO</td></tr>
<tr><td>Receiver</td><td>Service Provider</td></tr>
<tr><td>Description</td><td>The DSO inquires the DB about the voltage magnitude of the specified HEMS</td></tr>
<tr><td>Condition</td><td></td></tr>
<tr><td>Data</td><td>Notification Type (abnormal voltage inquiry on Hems_ID with time stamp),
"Hems_ID": "ID1234" (string)</td></tr>
<tr><td>Request</td><td>As need</td></tr>
<tr><td>Response</td><td>
<table>
<tr><th>Field</th><th>Example</th></tr>
<tr><td>HEMS Id</td><td>1234</td></tr>
<tr><td>Voltage</td><td>240</td></tr>
<tr><td>Unit</td><td>Volts</td></tr>
<tr><td>timestamp</td><td>"2021-12-03T15:11:20Z"</td></tr>
</table>
JSON Version:

*{
"Voltage_Monitoring" : [
{*

</td></tr>
</table>

	"Hems_ID": "ID1234" *"timestamp": "2021-12-03T15:11:20Z", (timestamp)* *"Voltage_from_Hems: "240", (float)* *"Voltage_from_Hems: "volts",* *},* *]* *}*

3.2.3.3OTHER REQUESTS

Regarding PUC 2, this service will be internal do the service requester (e.g. DSO or to a retailer). For this reason, there is no need to implement SAREF in this case since the information is gathered by the service requester and the outputs are available for itself. The exchange of information, action flow and details of the service are presented in D4.3. The information to be gathered and to be submitted can be through an internal API. The inputs and outputs are detailed below in JSON format. These are obtained using the Post method, and the user can DELETE the dataset sent:

API	Data requirements for Service 2
Input Dataset	*{* *"user_profile" : [* *{* *"userID": "ID644", (string)* *"timestamp": "2021-04-12T15:11:20Z", (timestamp)* *"customer_consumption": 0.2, (float)* *"customer_consumption_unit: "kW",* *"signal_incentive_Class:Cost: 0.11, (float)* *"signal_incentive_unit: "€/kWh",* *"signal_incentive_Class:CO2_Emissions: 0.11, (float)* *"signal_incentive_unit: "gCO2/kWh",* *"signal_incentive_Class:Renew_Energy_Percentage: 0.11, (float)* *"signal_incentive_unit: "%/kWh",* *"signal_incentive_Class:maximum_power: 0.11, (float)* *"signal_incentive_unit: "kWh",* *"external_temperature": 19, (float)* *"external_temperature_unit": "degreesCelcius",* *"indoor_temperature_sensor": 24, (float)* *"indoor_temperature_sensor_unit": "degreesCelcius",* *"occupancy_rate": 4.3, (float)* *},* *...*

	] *}*
Output of the Service	*{* *“userID”: “ID644”, (string)* *"flexibility_response" : [* *{* *“flexibility_hour”: 7, (integer)* *“flexibility_hour_unit”: "h",* *“y_PowerD”: 2.1, (float)* *“y_PowerD_unit”: "kW",* *“y_lowerD”: 1.5, (float)* *“y_lowerD_unit: "kW”,* *“y_upperD”: 2.4, (float)* *“y_upperD_unit: "kW”,* *“y_Power_i1”: 2.1, (float)* *“y_Power_i1_unit: "kW”,* *“y_lower_i1”: 2.4, (float)* *“y_lower_i1_unit: "kW”,* *“y_upper_i1”: 2.4, (float)* *“y_upper_i1_unit: "kW”,* *“y_Power_i2”: 2.1, (float)* *“y_Power_i2_unit: "kW”,* *“y_lower_i2”: 1.5, (float)* *“y_lower_i2_unit: "kW”,* *“y_upper_i2”: 2.4, (float)* *“y_upper_i2_unit": "kW”* *… .* *“y_Power_i24”: 2.1, (float)* *“y_Power_i24_unit: "kW”,* *“y_lower_i24”: 1.5, (float)* *“y_lower_i24_unit: "kW”,* *“y_upper_i24”: 2.4, (float)* *“y_upper_i24_unit": "kW”* *},* *…* *],* *"Ranking" : [* *{* *“Ranking_hour”: 7, (integer)* *“Ranking_hour_unit”: "h",* *“rank_Flex_i1_Price)”: 85, (float)* *“rank_Flex_i1_Price_scale: "%”,* *“rank_Flex_i2_Price)”: 85, (float)* *“rank_Flex_i2_Price_scale: "%”* *… .* *“rank_Flex_i24_Price)”: 85, (float)*

	"rank_Flex_i24_Price_scale: "%" *},* *...* *],* *"Reliability_i1_hours" : [* *{* *"Intervention_hour": 7, (integer)* *"Intervention_hour_unit": "h",* *"i1_Signal_Power_reliability": 12, (float)* *"i1_Signal_Power_reliability_scale": "%"* *},* *...* *],* *"Reliability_i2_hours" : [* *{* *"Intervention_hour": 7, (integer)* *"Intervention_hour_unit": "h",* *"i2_Signal_Power_reliability": 23, (float)* *"i2_Signal_Power_reliability_scale": "%"* *},* *... .* *],* *"Reliability_i24_hours" : [* *{* *"Intervention_hour": 7, (integer)* *"Intervention_hour_unit": "h",* *"i24_Signal_Power_reliability": 17, (float)* *"i24_Signal_Power_reliability_scale": "%"* *},* *...* *]* *}*

3.2.3.3.1.1 DISAGGREGATED DATA (GET)

The following method refers to Service 3 (PUC 3) in which the DSO makes a request to a data service provider for data from Hems using a set of IDs. This data needs to be SAREFized. The remaining interactions between the user and service is detailed in T4.3.

API	https://inesc.interconnect.pt/GridObservability**/disaggregatedData**
Method	**GET**
Producer	DSO
Receiver	Data Service Provider
Description	The DSO requests disaggregated consumption area for a list of customers

Input Data	*{ "customers_ids": ["xvc989007", "ert552555", "uio999900", "alo092883", "fty838399", "eur989820"] }*
Request	As needed
Response	Gets disaggregated data sets for the customers_Ids provided

3.3 API/ADAPTERS LIFECYCLE MANAGEMENT

This section refers to the development and implementation of the REST APIs which will enable the DSO interface to communicate with external systems, and which will serve as basis for the integration in the IFA.

For this process it is important to consider the lifecycle process when it comes to developing this kind of communication mechanisms. The development and maintenance of the REST APIs will follow the API Life management presented below, which can be divided in 6 steps:

- Design
    - Architecture
    - Security
- Development
    - Test
    - Integration
    - Local deployment
- Publish
    - Documentation
    - Define Operations- Requests and responses
    - Final test iteration and code adjustments
    - Production Deployment
- Monitor
    - Analytics
    - Find errors
    - Generate errors report
- Update

    - Version control
    - Backup production
    - Development/Publish phases
- Deprecate (Retire)
    - New Version available
    - No more Support or bug fixes
    - May still run for legacy purposes

As referred in Figure 11 (Akana API lifecycle, s.d.), the API lifecycle consists of three primary phases — create, control, and consume. In the first one builds and documents the API, then in the control phase, one applies security policies and finally in the consume phase the API is published.

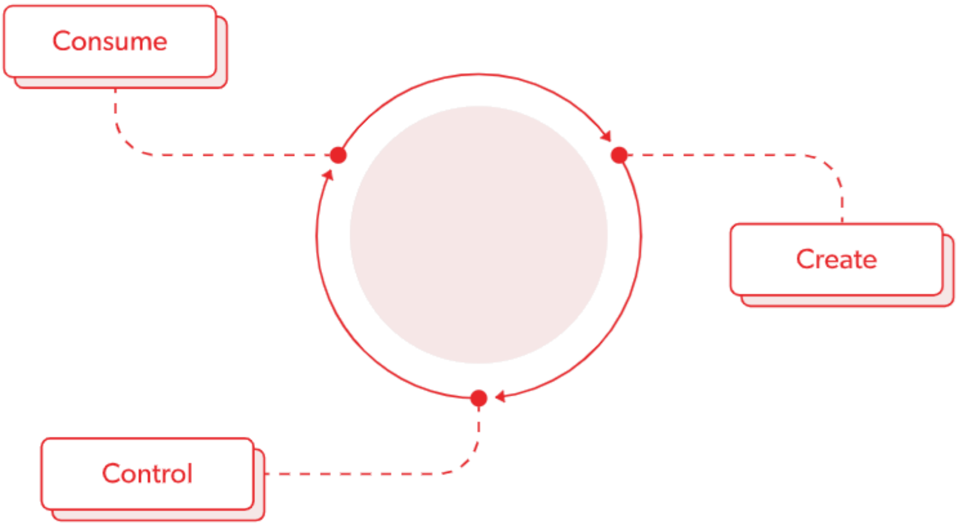


FIGURE 11 - API LIFE CYCLE AS SUGGESTED BY AKANA

To conclude the APIs previously presented and developed inside the DSO interface are going to incorporate these best practices in their different life cycle steps ensuring the continuous working until the end-of-life of the API.

3.4 INTERCONNECT INTEROPERABILITY LAYER INTEGRATION

The DSO interface integration in the Interconnect semantic interoperability layer (part of the InterConnect Interoperability Framework (Interconnect, 2020)) will be accomplished through the development of a Service Specific Adapter, which will translate the previously defined API messages into semantic triplets. The integration with the interoperability framework will allow the interaction with the wider interoperable ecosystem, particularly when assembling the pilot demonstrations. This service specific adapter will then interact with the IC Generic Adapter (entry point into IC Interoperability Framework) to communicate with the semantic interoperability layer.

From the perspective of the Interoperability Framework, the DSO interface acts as a digital service. This implies the need to deploy a Generic Adapter within its environment as depicted in Figure 12. A Service Specific Adapter is required as it deploys the client interface towards the Generic Adapter, as the integration point with the DSO interface's Service Middleware.

The Service Specific Adapter is built as the result of the SAREFization process, detailed in the scope of the WP3. The process consists of the following steps:

- Analysis of the DSO Interface API, identifying the interfaces that need to be made interoperable.
- Identification of the type of Knowledge exchange interactions (i.e., Post-Reach and Ask-Answer) required. The identified knowledge interactions assemble the knowledge exchange patterns possible.
- Assembling for each required knowledge interaction the respective graph pattern, offering the data modelling for each interaction according to the SAREF family of ontologies.

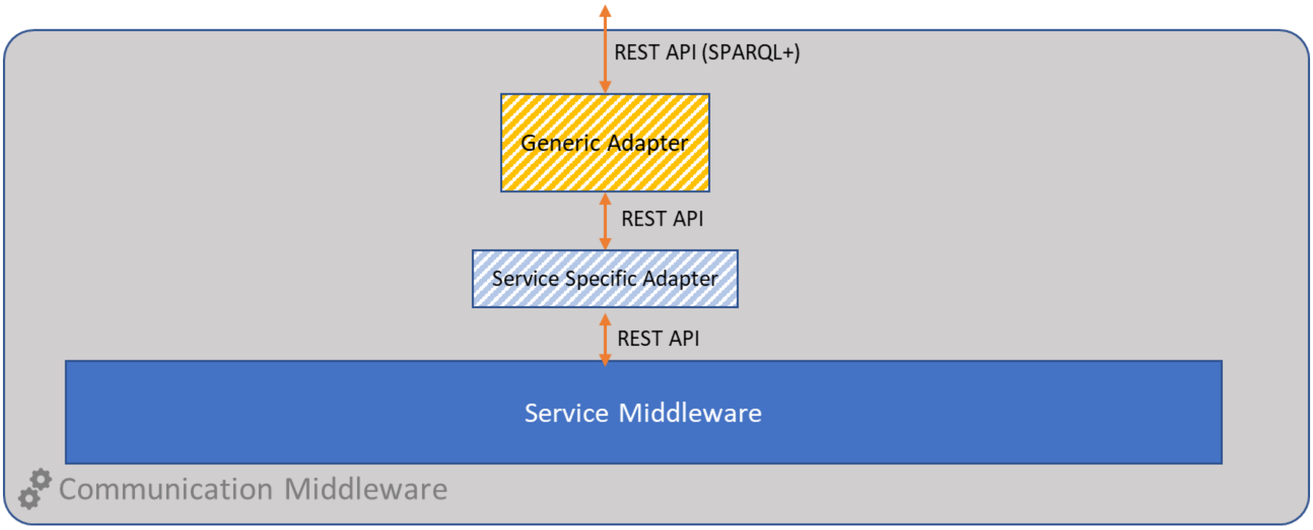


FIGURE 12 - COMMUNICATION MIDDLEWARE DESIGN

The integration of the DSO Interface with the Interoperability Framework will allow it also to consume information from other semantically interoperable digital services through the same interface. This differentiates the approach from a standard REST API interface, where external requests are just processed and served. The DSO Interface's requests to other interoperable services are possible by considering knowledge interactions of type *ASK* or *REACT* directed at the desired service. This renders this as a bidirectional interface that allows the DSO interface to serve data to the interoperable ecosystem and to consume data from other interoperable services.

The integration with the interoperability framework also provides the possibility to adjust or limit the privacy boundaries for the data exchanged through the interoperable interface, by using the Service Store to set the services that can interact with the DSO interface as an interoperable service. Moreover, as an intrinsic requirement for the Generic Adapter of the DSO interface to operate, it must undergo the authorization and authentication with the Interoperability Framework identity system provided by the Service Store.

3.5 PRIVACY AND SECURITY

This section provides guidance on how to address privacy and associated security requirements when the DSO interface includes data flow which might be require clearance in terms of privacy compliance. The Figure 13 shows a high-level view of the elements at stake:

- The flexibility service (e.g., HEMS/BEMS service) above
- The DSO level capability (e.g., management of smart meter data sets) in the DSO interface layer

The data flow which might include (or relate) to personal data (e.g., smart meter data) as well as related privacy management data (e.g., consent evidence).

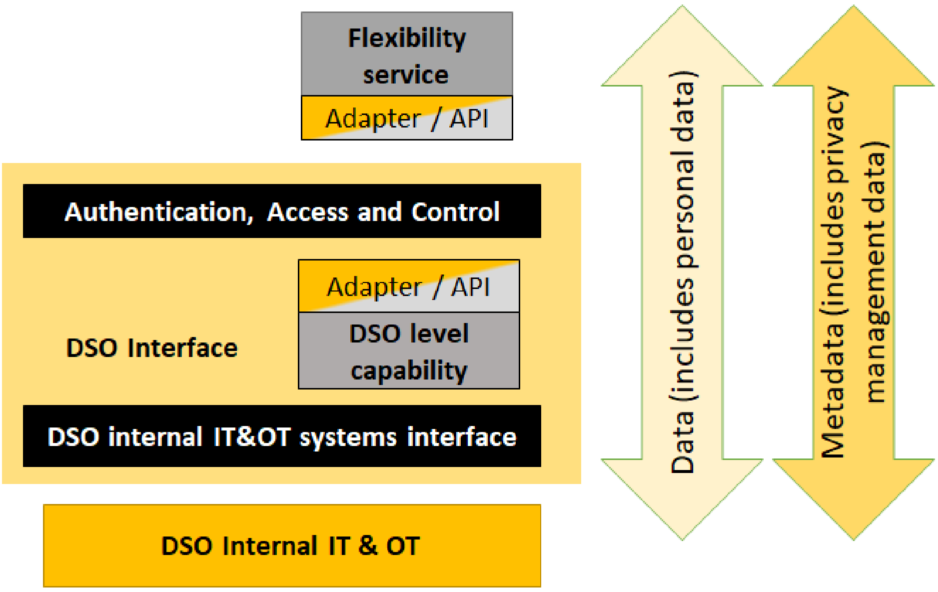


FIGURE 13 - DATA AND METADATA

The following guidance is provided:

- A security and privacy plan for the DSO interface is maintained by the DSO. It includes information that is made available to the service provider, based on the table below.

TABLE 9 - DSO SECURITY AND PRIVACY INFORMATION

Category	Description
Security and privacy capability information	• Provides a template of a security and privacy plan that the DSO would like the flexibility service provider to use (can be based on the guidance provided to the pilots in D5.3. Section 1.4 explains the five components of the plan: governance management, data management, risk management, engineering management and citizen management) • Provides a description of the security and privacy capability of the DSO interface. Can use templates provided in D5.3[1] • DSO data protection officer agreement on content

[1] Section 2 of D5.3 describes the security and privacy capability of the Interconnect interoperability framework which can be used as an example.

Practice information	• Explains the best practices the DSO is using to integrate privacy-by-design, as well as standards it is following (see https://ipen.trialog.com/wiki/ISO for an updated list of such standards) • Explains its resilience program. Maintains a list of privacy risk sources, potential consequences et processing capabilities
Privacy measures information	• Maintains a list of capabilities that it can provide or has already provided (e.g., privacy enhancing technologies, surveillance of privacy controls)
Datasets and privacy compliance procedures	• Maintains a list of datasets that can be provided, associated with privacy management datasets (e.g., consent evidence)
Operation information	• Contact points

- A security and privacy plan is maintained by the service provider. It includes not only a security and privacy analysis that will provide the privacy management capabilities, but also information that is made available to the DSO, based on the table below.

TABLE 10 - SERVICE PROVIDER SECURITY AND PRIVACY INFORMATION

Category	Description
Security and privacy capability information	Uses the template agreed with the DSO Security and privacy analysis of the service. Includes an overall privacy impact assessment considering the information provided on the DSO interface Provides a description of the security and privacy capability of the service and the service providers (technical, organizational) Service provider data protection officer agreement on content, and possibly conformance statement
Practice information	Explains the best practices the service provider is using to integrate privacy-by-design, as well as standards it is following Explains its resilience program. Maintains a list of privacy risk sources, potential consequences et processing capabilities
Privacy measures information	Describes the technical and organizational measures for privacy Include a registry of personal data operations
Operation information	Contact at citizen level Contact at service provider level Contact at DSO level

- A resulting API interface is generated and agreed between the DSO and the service provider. It includes management capabilities such as continual improvement, policy updates, based on the table below.

TABLE 11 - API INFORMATION

Category	Description
Policies	• List of policies that are managed by service • Datasets and agreed usage

	• Agreed risk analysis
Metadata	• Service provider – DSO agreement • Consent information • Usage log • Incident management (event, triaged event, incident handling)
Data	• According to agreed usage
Registry of operations	• Info for registry of operations

The Figure 14 shows the case where the flexibility service is based on the Interconnect interoperability framework. The difference with the previous case is that the system includes one more stakeholder: The Interconnect interoperability framework operator.

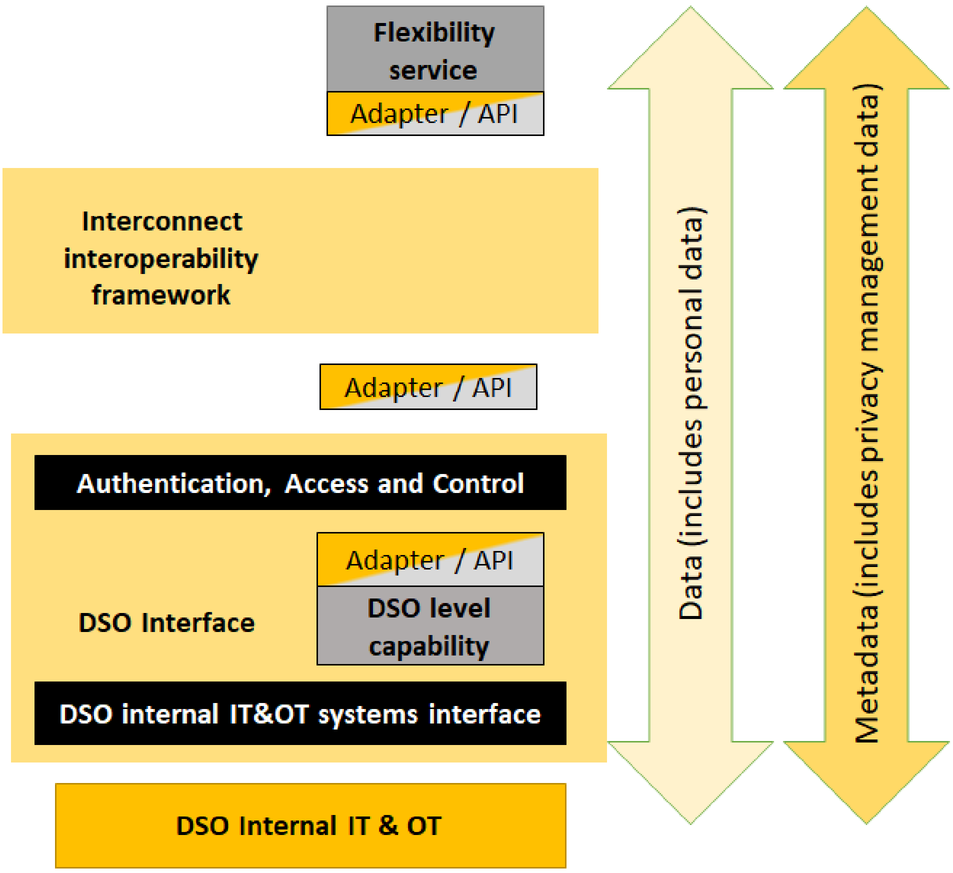


FIGURE 14 - DATA AND METADATAFLWO

4. OPERATING REQUIREMENTS AND IMPLEMENTATION

The overall goal of this activity is to develop the DSO Interface, which will be a cloud system that will serve as a gateway between the DSO's internal systems and external entities such as End Users and Service Providers.

The envisioned Architecture (Figure 9Figure 9 - DSO interface architecture) will be deployed in Microsoft Azure Cloud according to the DSO specification always ensuring a secure and GDPR compliant implementation.

The architecture consists of several modules that should be developed and implemented. Some of the modules are:

- **Communication/Running Services:** Enable the interaction with external parties through a set of APIs to cover the foreseen information exchange, as detailed in the use cases
- **Data Storage:** Host DSO OT information, to avoid direct interaction with critical databases, and will be limited to relevant assets. Furthermore, to answer the other use cases, additional datasets/tables must be developed (e.g. Flexibility Service)
- **Flexibility Management Platform:** Central service to enable communication between buyers and sellers to facilitate the coordination of all processes related to the procurement and activation of flexibility.
- **Analytics Engine** In this layer, the data gathered in the Data Storage layer can be processed to retrieve more dynamic value from it.

4.1 MODULES & SERVICES TECHNOLOGIES

The development and implementation process will be divided in a 5-step approach:

- Technical Criteria Evaluation
- Development
- Deployment
- Testing
- Implementation

This document provides the baseline for the beginning of the development process, however there are some technical restrictions and criteria that should be followed, decided, and provided during the next phases of the process.

The Operating Environment requirements can be found in the table Table 12.

TABLE 12 - DSO INTERFACE OPERATING REQUIREMENTS

Requirements	Possible Solution
Running place of the API	Internal or External
Machine	Virtual (Azure) or Physical

Endpoints	List of Endpoints
Documentation	List of API functions
Logs and Alerts	List of logs and alerts
Authentication	Token

The Operating Environment implementation definition should consider the criteria presented in Table 13.

TABLE 13 - DSO INTERFACE IMPLEMENTATION DEFINITION

Implementation definition	Criteria
Machine specification	Minimum viable number of hardware specification including vCPUs, Ram and Disk
Machine Operating system	Server OS either Linux long term support distribution or Windows
API Division	Internal communication and external
API Architecture	Rest API
API Requests	GET, POST, PUT and DELETE
API status codes	200 OK, 201-202 – accepted, 400- bad request, 401- unauthorized, 404- not found, 500- internal server error
Data exchange format	JSON
Logs and alerts	Saved locally and online
Documentation	Swagger or postman

Lastly, the non-functional requirements should also be considered, depicted in Table 14 - DSO Interface Non-functional requirements.

TABLE 14 - DSO INTERFACE NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements	Definition
Performance	How the API is coping with the request can be measured with response-time
Reliability	How long is the API is running using time measure (hours, days...) Difference between received request and answered (without error code) request.
Usability	How user friendly is the API, how good is documentation.

4.2 DATA MODEL & STORAGE

A link with the DSO internal system will be made resourcing to cloud databases with a different ecosystem from normal operation OT systems, that will store only the needed data for the demonstrator. This data will serve as input for the operations to be performed on the current architecture.

The data targeted to be deployed on the platform will be:

- Low Voltage (Participant clients)

TABLE 15 - LOW VOLTAGE CLIENTS DATA MODEL

Measure	Unit
Unique Identifier	Number
Active Power Injected	kW
Active Power Withdrawn	kW
Reactive Power Injected (Capacitive/Inductive)	kVAr
Reactive Power Withdrawn (Capacitive/Inductive)	kVAr
Voltage	V

- Special Low Voltage and Medium Voltage (Participant Clients)

TABLE 16 - SPECIAL LV CLIENTS AND MV CLIENTS DATA MODEL

Measure	Unit
Unique Identifier	Number
Active Power Withdrawn	kW
Reactive Power Injected	kVAr
Reactive Power Withdrawn	kVAr

- Secondary Substations (SS)

TABLE 17 - SECONDARY SUBSTATION DATA MODEL

Measure	Unit
Unique Identifier	Number
Active Power Injected	kW

Active Power Withdrawn	kW
Reactive Power Injected (Capacitive/Inductive)	kVAr
Reactive Power Withdrawn (Capacitive/Inductive)	kVAr
Voltage	V

- Primary Substations (PS)

TABLE 18 - PRIMARY SUBSTATION DATA MODEL

Measure	Unit
Unique Identifier	Number
Active Power	MW
Reactive Power	MVAr
Current	A
Voltage	kV
Tap Position	N/A

- Grid Topology (To understand the dependence between consumers and grid assets)

TABLE 19 - GRID TOPOLOGY DATA MODEL

Client/Producer	Secondary Substation	SS Feeder	Primary Substation	PS Feeder
LV Unique Identifier	Unique Identifier	Unique Identifier	Unique Identifier	Unique Identifier
MV Unique Identifier			Unique Identifier	Unique Identifier

However, this will not be the only present data on the DSO interface. As referred, several database schemas will be created to support the normal operation of the running modules (2.2.1).

4.3 FUTURE WORK

As referred in 1.1, this document is the first version of the technical specification, which will be complemented during the development and implementation efforts. For this future work, it is foreseen the following activities:

1. Technical criteria assessment:

   a. OpenAPI Specification of the API messages to be exchanged
   b. Definition of technical API authentication mechanisms and access policies mechanisms
   c. Integration with the IFA, according to the work from WP3 and WP5
   d. Definition of technologies to support the development of the several modules
   e. Definition of the databases schemas and system communications
   f. Final design of the DSO interface technical architecture

2. Development and implementation of the DSO Interface

Additionally, the developments should follow the requirements below:
1. The adopted technologies should be open source
2. The developed code shall be made available in a git environment (GitHub, Gitlab, etc..) and follow the coding best practices (commenting, indentations, variable naming, etc.) with the proper instructions for its deployment and replicability
   a. The DSO interface will be implemented on the Azure cloud environment, within a provided resource group a. Ubuntu VMs for hosting the modules (web app, middleware, and backend services)

This system represents the linkage between the consumers, service provides and the distribution system operation. The usage of semantic interoperable mechanisms between all of these parties is very important for the future agile and efficient usage of the distributed resource which are more and more present in the LV grids.

5. REFERENCES

Akana API lifecycle. (s.d.). https://www.akana.com/resources/api-lifecycle

Interconnect. (2020). *D2.1 - Secure interoperable IoT smart home/building.*

Interconnect. (2021). *D1.3 - System use cases for smart buildings and grids.*

Interconnect. (2021). *D4.1 - Functional Specification of DSO Standard Interface Application.*

Interconnect. (2021). *D5.3 - Applying practice for security and privacy policies compliance.*