D5.5 - Final DSF Connectors for external systems and services

Deliverable ID	**D5.5**
Deliverable Title	**Final DSF Connectors for external systems and services**
Work Package	**WP5**
Dissemination Level	**PUBLIC**
Version	**1.0**
Date	**10/09/2019**
Status	**Final**
Type	**Prototype**
Lead Editor	**UNINOVA**
Main Contributors	**UNINOVA (Fábio Januário, Vasco Delgado-Gomes), LINKS (Michele Ligios, Orlando Tovar), UPB (Mihai Sanduleac, Mihaela Albu, Marta Sturzeanu), LiBal (Yini Xu)**

Published by the Storage4Grid Consortium

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731155.

Document History

Version	Date	Author(s)	Description
0.1	2019-02-26	UNINOVA	First TOC draft.
0.2	2019-07-22	UNINOVA	Update TOC and UNINOVA inputs.
0.3	2019-08-14	UPB	UPB inputs.
0.4	2019-08-21	UNINOVA	UNINOVA updates.
0.5	2019-08-27	UPB	UPB updates.
0.6	2019-08-28	LINKS	LINKS updates.
0.7	2019-08-28	LiBal	Inputs to LiBal System interface.
0.8	2019-08-29	UNINOVA	Version ready for internal review.
1.0	2019-09-10	UNINOVA	Ready for submission to the EC.

Internal Review History

Review Date	Reviewer	Summary of Comments
2019-09-06 (v0.8)	Yini Xu (LIBAL)	Approved • Corrections • Comments
2019-09-06 (v0.8)	Elia Kinigadner (ALPERIA)	Approved • Minor Corrections

Legal Notice
The research work leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731155 - Storage4Grid project. The information in this document is subject to change without notice. The Members of the Storage4Grid Consortium make no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Members of the Storage4Grid Consortium shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material. The European Union and the Innovation and Networks Executive Agency (INEA) are not responsible for any use that may be made of the information contained therein.

Table of Contents

Executive Summary

This deliverable presents the final results of Task 5.2 – "DSF Interoperability with 3rd party and DSO systems" and Task 5.3 – "DSF Hybrid Simulation Support" of WP5 – "Decision Support Framework". The main goal of Task 5.2 was to develop a set of connectors to allow the Decision Support Framework (DSF) to operate with third-party and Distribution System Operator (DSO) systems. Task 5.3 objective was the integration of the Decision Support Framework Simulation Engine (DSF-SE) with the storage systems interfaces for realistic conditions evaluation.

The following DSF connectors were developed in the S4G project, namely:

- DSO connectors:
  - DSO Supervisory Control and Data Acquisition (SCADA) system.
  - DSO Grid models.
  - DSO Electric Vehicle (EV).
- Third-party systems connectors:
  - Geolocation.
  - Weather forecast.
  - Energy price.
  - Professional Realtime Optimization Framework for Energy Storage Systems (PROFESS)/ Professional Realtime Optimization Framework for Electric Vehicles (PROFEV) solar radiation.
- Control connectors:
  - Energy Router (ER).
  - LiBal System.
  - Local Energy Storage System Agent (LESSAg).
  - Hardware-in-the-loop (HIL).
- DSF-SE hybrid simulator connector.
- DSF predictive models connector.

This deliverable provides all the necessary information and requirements to install and deploy the DSF connectors. Moreover, it is also presented their software dependencies and Application Programming Interface (API) to enable the connectors interoperability. D5.5 is a prototype deliverable reporting the final results of Task 5.2 and Task 5.3, so no updates on this deliverable are expected.

1 Introduction

The deliverable D5.5 – "Final DSF Connectors for external systems and services" documents the final results of Task 5.2 and Task 5.3 describing the Decision Support Framework (DSF) connectors to operate with third-party and Distribution System Operator (DSO) systems, and the Decision Support Framework Simulation Engine (DSF-SE) integration with its connectors.

Section 2 presents the DSF, detailing the D5.5 architecture. Section 3 gives an overview of the DSF related S4G prototypes to foster interoperability and data collection. The DSF connectors are presented in section 4. The installation and deployment of the connectors are presented in section 5. Section 6 shows the software dependencies and requirements, whereas section 7 presents the DSF connectors API reference. Lastly, section 8 discusses the final conclusions.

1.1 Scope

Following the S4G project workplan, the effort in WP5 – "Decision Support Framework" was focused on the development of the necessary components necessary to gather data from the three test sites Bolzano (IT), Fur/Skive (DK), and Bucharest (RO).

The work in WP5, and more specifically in Task 5.2 – "DSF Interoperability with 3rd party and DSO systems" and Task 5.3 – "DSF Hybrid Simulation Support", are directly connected with WP3 – "S4G Architecture". Since this deliverable report Task 5.2 and Task 5.3 final results, so no updates on this deliverable are expected.

1.2 Related documents

ID	Title	Reference	Version	Date
D2.2	Final Storage Scenarios and Use Cases	[S4G-D2.2]	1.0	2018-07-31
D3.3	Final S4G Component Interfaces and Architecture Specification	[S4G-D3.3]	0.9	2019-09-05
D4.5	Final Grid-side ESS control system	[S4G-D4.5]	0.9	2019-08-01
D4.10	Final USM extensions for Storage Systems	[S4G-D4.10]	0.4	2019-09-02
D5.4	Updated DSF Connectors for external systems and services	[S4G-D5.4]	1.0	2018-09-04
D5.7	Final DSF Predictive Models	[S4G-D5.7]	0.9.2	2019-08-26
D6.3	Phase 3 Test Site Plans	[S4G-D6.3]	0.25	2019-08-30

2 Final DSF Connectors for External Systems and Services Prototype Overview

The DSF high-level structure is shown in Figure 1. D5.5 – "Final DSF Connectors for external systems and services" prototype describes the DSF connectors, namely: DSO connectors, third-party systems connectors, control connectors, DSF-SE hybrid simulator connector, and DSF predictive models connector. The remaining DSF components are described in other deliverables of WP5 – "Decision Support Framework".

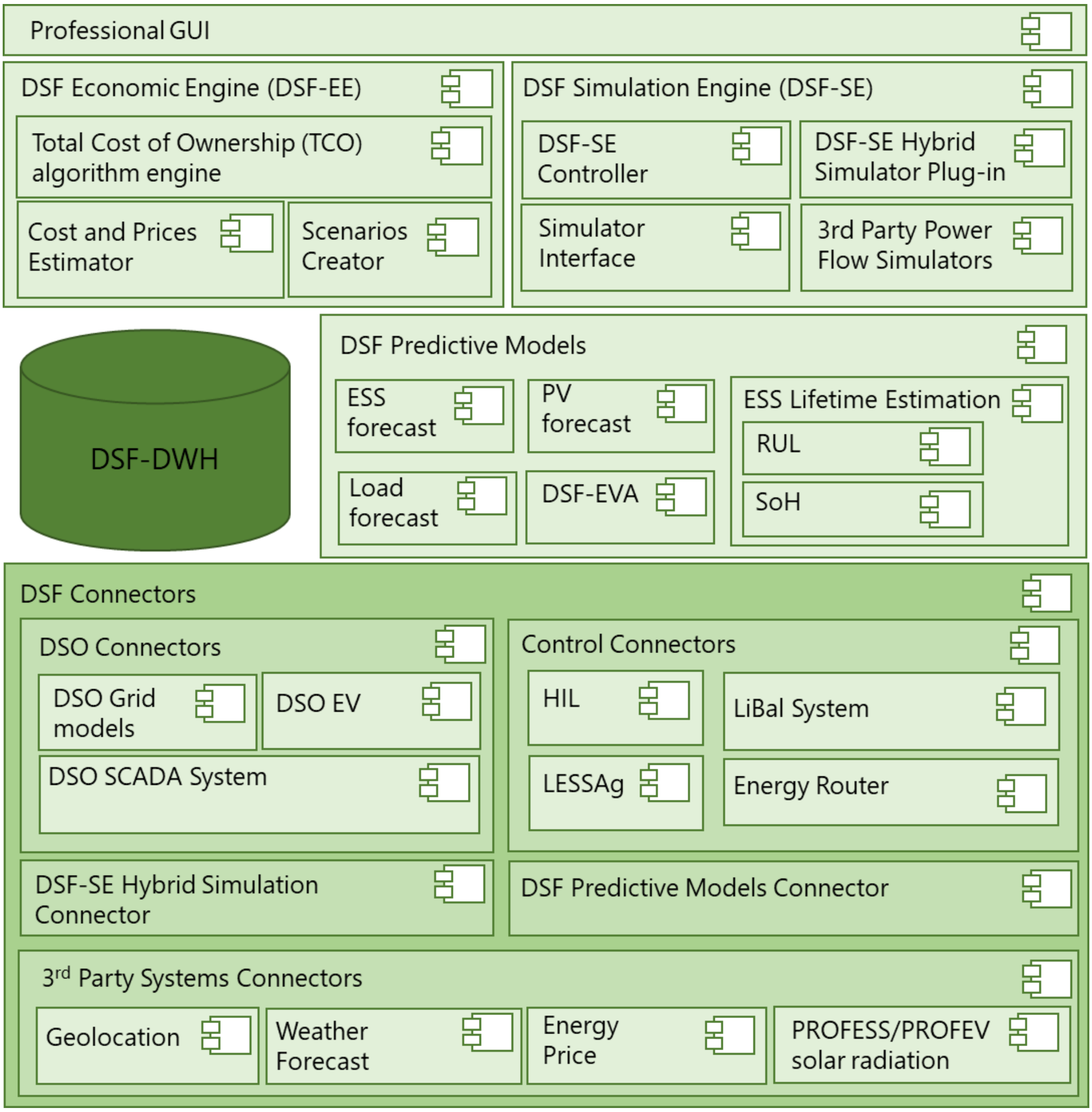

Figure 1. D5.5 prototype diagram [S4G-D3.3].

3 S4G prototypes to foster interoperability and data collection

3.1 Introduction

This chapter provides a high-level overview of the implementation of the S4G Data Warehouse (DWH), i.e. the main component set-up to collect all the test sites data in S4G, together with the S4G components enabling bi-directional communication and fostering interoperability.

The main purpose of these components is to enable the feeding of data from the physical layer into the data warehouse and to enable the forwarding of control messages from both the edge and the service layer towards the Physical layer, as described in the following subsections.

3.2 DSF-DWH

The Decision Support Framework Data Warehouse (DSF-DWH) enables the data collection. It is an open source distributed platform providing ad-hoc services for smart grid applications to support monitoring and control systems, as well as to support analysis, planning, forecast and optimization of storage systems behaviour. It is a time-series database used to store raw data from the S4G test-sites. Moreover, it provides core services to accumulate, analyse, and act on time-series data with strictly data access control policies to preserve privacy. It is installed and maintained either as a private cloud service or in the premises of the entity operating S4G-based services. Further details are provided in D3.3 [S4G-D3.3].

3.3 Data Dispatcher

The Data Dispacthes enables the forwarding of messages (measurements and controls) from the physical layer towards the edge layer and vice versa. It subscribes to the Smart Meter eXtension (SMX) Broker to receive data coming from the Physical Layer and to forward them to the upper layers such as Edge Layer. Further details about this component are presented in D3.3 [S4G-D3.3].

3.4 OGC Wrapper

The Open Geospatial Consortium (OGC) Wrapper enables the forwarding of messages (measurements and controls) from the edge layer towards the service layer and vice versa. It is a software based on Message Queue Telemetry Transport (MQTT), which transmits SMX data to the Service Layer, from Service Layer to the Aggregator services, and from the Aggregator to a specific SMX. It is compliant with OGC Sensor Things standard[i] and allows to keep a Common Information Model (CIM) among S4G components. Moreover, it describes components in Device and Edge Layers and generates a virtual association between specific SMXs and the Aggregator hosting it, i.e., the Aggregator is considered as a single resource hosting N SMXs with specific functionalities. Finally, it supports interoperability among S4G components exploiting the registration of its "resources" to the OGC Sensor Things Server. Further details about this component are described in D3.3 [S4G-D3.3].

4 DSF Connectors

This chapter details all the DSF connectors developed within the S4G project, namely:

- **DSO connectors**: enabling the data collection from a DSO system, i.e., monitoring data from the SCADA system, the DSO grid model, and the EV monitoring and control (section 4.1).
- **Third-party systems connectors**: enabling the data input from third-party services, i.e., geolocation, weather forecast, energy price, and solar radiation for PROFESS/PROFEV (section 4.2).
- **Control connectors:** enables the different DSF systems to interact S4G prototypes, e.g., ER, LESSAg, and HIL (section 4.3).
- **DSF-SE hybrid simulator connector:** allows the interaction with real physical components as the resulting output of the DSF-SE hybrid simulator (section 4.4).
- **DSF predictive models connector:** retrieves the output of the DSF predictive models, e.g., load forecast, Photovoltaic (PV) production forecast, Energy Storage System (ESS) status forecast (section 4.5).

4.1 DSO Connectors

4.1.1 DSO SCADA System

This connector was initially designed to enable the DSF real-time data injection of the DSO SCADA information using the IEC 60870-5-104[ii], as shown in Figure 2.

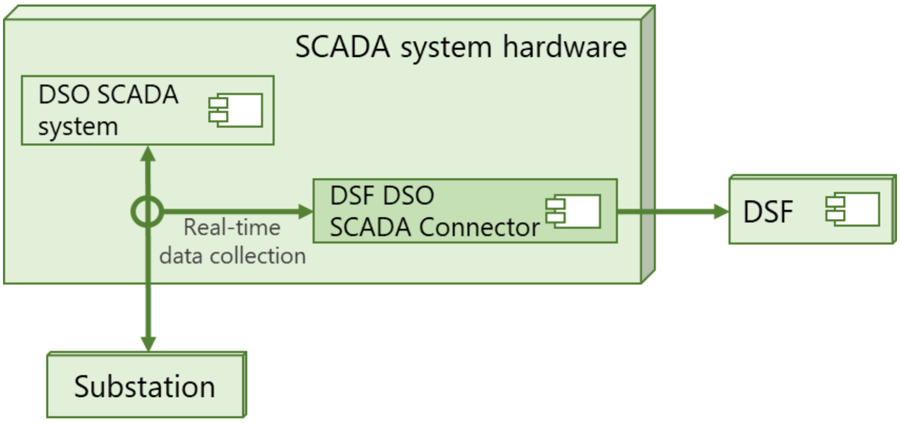

Figure 2. DSO SCADA connector deployed on-site.

However, there where operational and security concerns in deploying additional software and hardware in a DSO SCADA system. Therefore, it was necessary to implement an alternative that does not affect the DSO SCADA operation.

Since EDYNA uses the Fink WinPP Protocol Simulator[iii] software to monitor the substations, a .lg4 file can be extracted containing all the exchanged information between the DSO SCADA system (TCP client) and the substations (TCP server).

The developed solution is presented in Figure 3 and enables the reading of a .lg4 file in the DSO SCADA connector, which then formats and forward the information to the DSF. The .lg4 file is IEC60870-5 standard compliant and an excerpt is show in Figure 4.

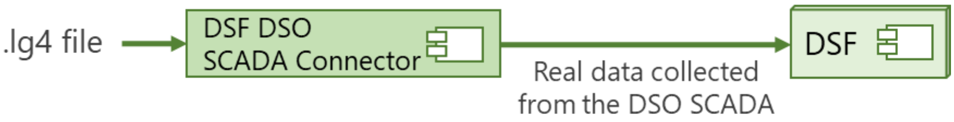

Figure 3. DSO SCADA connector developed solution.

```
/172.20.197.131:2404 -> /192.168.129.11:61447
Type ID: 9, M_ME_NA_1, Measured value, normalized value
Cause of transmission: SPONTANEOUS, test: false, negative con: false
Originator address: 0, Common address: 241
IOA: 234
Normalized value: 0.5037841796875
Quality, overflow: false, blocked: false, substituted: false, not topical: false, invalid: false
Unnormalized Value = 16508

/172.20.197.131:2404 -> /192.168.129.11:61447
Type ID: 9, M_ME_NA_1, Measured value, normalized value
Cause of transmission: SPONTANEOUS, test: false, negative con: false
Originator address: 0, Common address: 241
IOA: 246
Normalized value: 0.19598388671875
Quality, overflow: false, blocked: false, substituted: false, not topical: false, invalid: false
Unnormalized Value = 6422Single Point, is on: false, blocked: false, substituted: true, not
topical: false, invalid: false
```

Figure 4. Excerpt of a .lg4 file from the DSO SCADA system.

4.1.2 DSO Grid Models

The DSO grid model is a connector designed to access and read distribution network model from DSO database. Due to security reasons, the accessed database is a backup copy (located on the DSF) of the DSO ones. These models are available in the S4G Virtual Private Network (VPN) through Representational State Transfer (REST) requests. The configuration of the distribution feeder of Bolzano's test site is demonstrated in Figure 5.

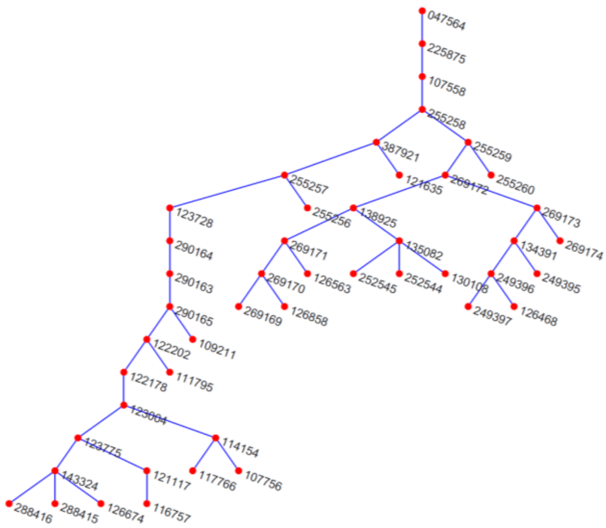

Figure 5. Bolzano test site feeder.

These data essentially contain buses and lines as the configuration definition and in an objective framework describes the nodes' connected elements such as transformers, loads and generators.
The grid related data have been stored in a JavaScript Object Notation (JSON) format in the database. An example of hierarchical grid object in the database is followed presented.


```
{
 "hv_mv_substation": {
  "mv_substations": {
   "substation_001": {
    "feeders": {
     "feeder_001": {
      "nodes": {
       "bus_047564": {
        "tech_id": "047564",
        "latitudine": 0,
        "longitude": 0,
        "from_root": 1,
        "parent": "substation_02",
        "children": {
         "TR1": {
          "tech_id": "TR1",
          "phases": 3,
          "windings": 2,
          "xhl": 0.014,
          "parent": "substation_02",
          "bus1": "047564",
          "kva1": 400,
          "conn1": "delta",
          "kv1": 16,
          "r1": 0.5,
          "bus2": "225875",
          "kva2": 400,
          "conn2": "wye",
          "kv2": 0.4,
          "r2": 0.5
         }
        }
       },
...
```


Such data are being used by DSF for simulation purposes, in a specific format adapted to the DSF-SE. As a result, the DSO grid model connector reads the JSON formatted grid information and convert them in a second phase to the DSF-SE adapted format.

4.1.3 DSO EV

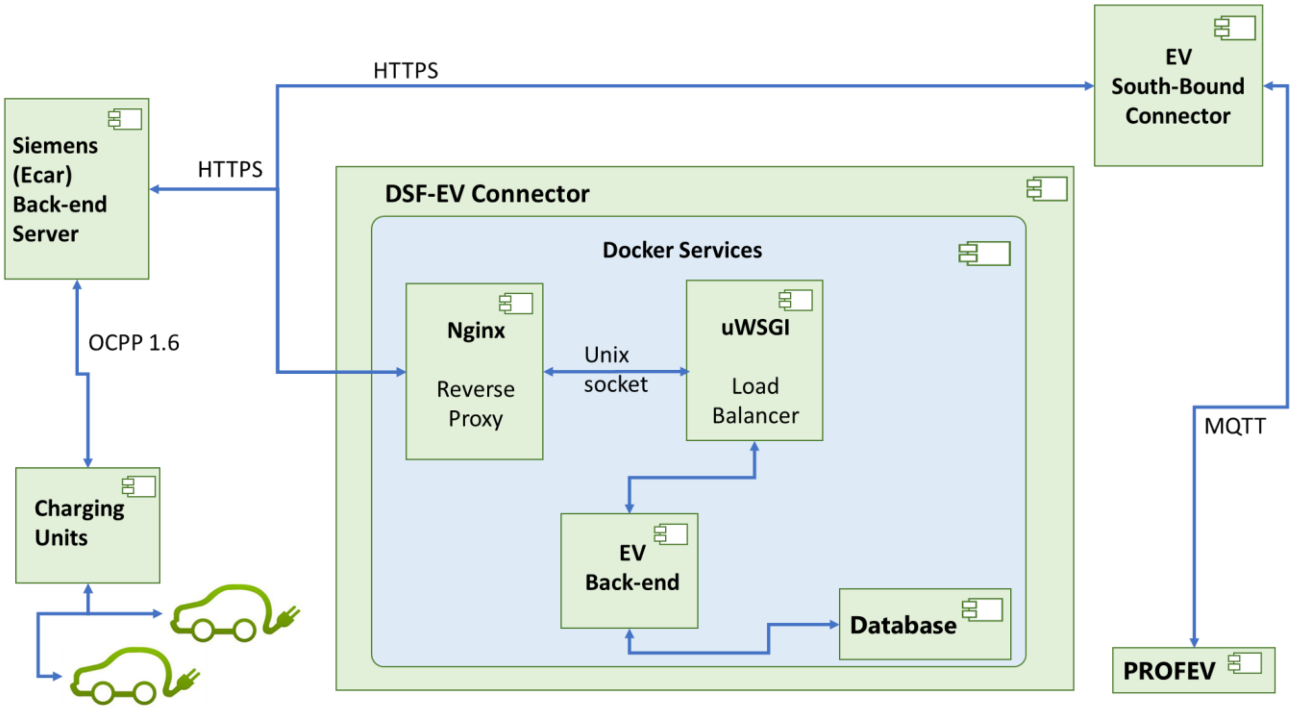

Figure 6. EV connector high-level architecture.

The EV connector is designed to specifically interact with the SIEMENS Energy Service and with PROFEV. The main role of the current connector is to behave as a transparent and secure interface between the two systems. It is built as a REST Hypertext Transfer Protocol Secure (HTTPS) public server to intercept all the registration/configuration and real-time measurements provided by the SIEMENS system. It is role of this private system to directly interact with the Charging Units (CU) of interest via Open Charge Point Protocol (OCPP)[iv] (1.6). By following this approach, one of the EV connector tasks is to provide a secure public interface towards the SIEMENS system.

To improve portability and scalability, it has been built as a standalone docker container integrating reverse proxy (Nginx) and load balancing (uWSGI) functionalities. To ensure a robust implementation it involves also a persistent Structured Query Language (SQL) database (sqlite3). Consequently, all the information provided by the Energy Service are properly modelled into different SQL tables for later uses. Due to current limitations of the quoted service, and PROFEV requirements, the connector performs real-time estimations of the EV's storage system State of Charge (SoC) during each recharging sessions. These calculations are made through a discrete formula exploiting the available information provided such as the electrical power and amount of time passed on each step.
The discrete formula used is presented in equation (1).

$$Updated\_SoC = PreviousSoC + \frac{Efficiency \cdot Power\_Injected \cdot dt}{Storage\_Capacity \cdot 3600} \quad (1)$$

Exploiting the EV South Bound connector it will forward via MQTT these SoC estimations exploiting the Sensor Measurements Lists (SenML)[v] format towards PROFEV. During each recharging session, it provides also ways to catch the required information to generate the proper modulation commands to control the power injection setpoints used by the CU of interest.
Due to some limitation of the optimization tool, each of the recharging session is mapped internally over a fixed amount of anonymized id. All the messages content exchanged in both directions is deeply controlled to

prevent wrong interactions with the Energy Service, by coupling the real-time data with the registration information of the CU hardware capabilities. In order to enhance the debugging functionalities of this service, the EV connector provide also ways to:

- Store on dedicated files each one of the raw messages received from the Energy Service system.
- Retrieve the structured data stored on the SQL database from specific devices inside the VPN, exploiting dedicated REST endpoints.
- Different configurable logging levels about its real-time behaviour.

4.2 Third-party System Connectors

4.2.1 Geolocation

The Geolocation aspects are considered to properly identify the location area of interest by the S4G services and to retrieve the relevant data specific for each country. In order to do so, the functionalities exposed by the current connector are only used by the S4G cloud services requiring it. The service exploited to provide this functionality is based on the GeoPy[vi] library.

4.2.2 Weather Forecast

This connector is built upon a free API offered by weatherunlocked[vii] that provides necessary information for solar irradiation calculation such as cloud density and temperature. The data is available for seven days in three hours' intervals.

This information is available inside the S4G environment exploiting a dedicated REST server providing updated forecast information to enhance other services evaluations e.g. the production forecasts of photovoltaic panels over the locations and dates of interest. Partial results of a request are shown in the Figure 7.

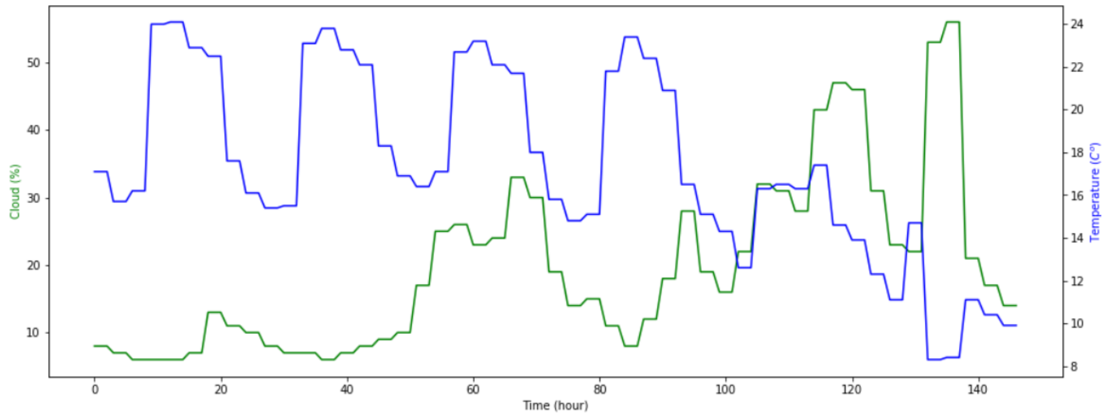

Figure 7. Cloud and temperature for seven days in Bolzano test site.

4.2.3 Energy Price

The Energy Price connector has been developed as a cloud REST service accepting input in a parametrized way, allowing to retrieve relevant and updated data for specific location of interest. Energy prices within S4G perspective are provided exploiting the transparency.entsoe.eu third-party service. The results exposed are in JSON format. This approach has been chosen since none of the project DSOs are directly exploiting spot market. The energy prices are still provided by DSOs as kind of price table in which various contractual and timing framework determine the specific price contents.

4.2.4 PROFESS/PROFEV Solar Radiation

This connector was fully described in the previous version of this deliverable (D5.4 [S4G-D5.4]), and no modifications were needed.

4.3 Control Connectors

4.3.1 Introduction

This section presents the control connectors related to DSF, which includes the hybrid simulation controlled by DSF and by the HIL connector. The DSF-SE have been developed in order to provide realistic conditions for the phase 3 evaluation. While DSF-SE is a general approach which is used for all the tests sites, the HIL connector used in the Advanced Prosumer (AP) allows specific tests which are needed only in the Bucharest test site. The section presents the connection of different S4G project field components with their cloud related components.

4.3.2 ER

The connector that enables the ER control and monitoring is fully described in D4.10 [S4G-D4.10].

4.3.3 LiBal System

LiBal System is a battery system including Battery Management System (BMS) and inverter. In S4G project LiBal System is considered as a closed system, which can only receive limited external control commands and it has been deployed in Bolzano commercial site during phase 3. In use-case HLUC-2-PUC-2 [D2.2], the LiBal System interfaces with PROFEV. It receives active power control setpoints from PROFEV through the Data Broker in SenML format and publishes ESS SoC every 10-seconds to other components in the S4G network through the Data Broker, as illustrated in Figure 8.

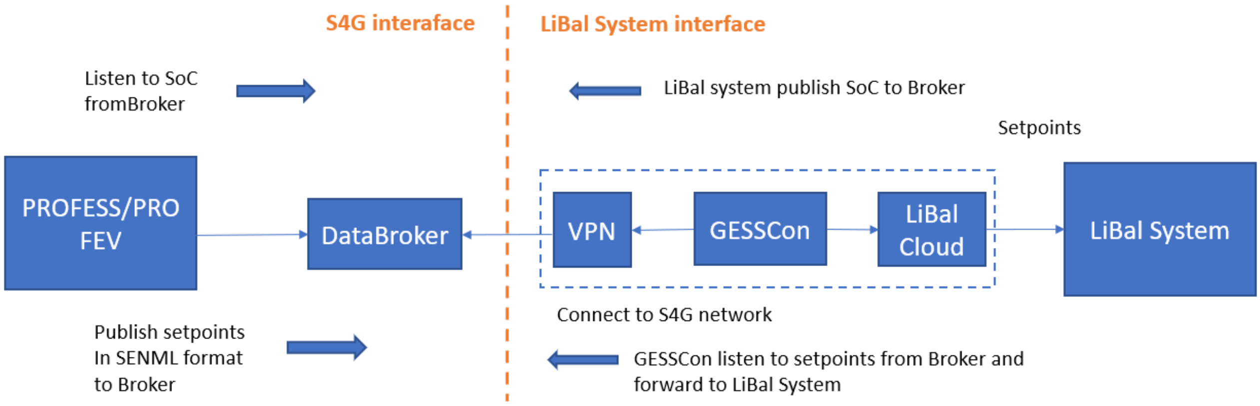

Figure 8. LiBal System interface to PROFEV in Bolzano commercial site.

4.3.4 LESSAg

The LESSAg supports the functionalities needed for the AP with all its extensions, which include the energy transfer to a neighbour and the resiliency behaviour.

LESSAg is stand-alone in order to be able to provide AP resiliency in various situations. It is an agent which can control the energy resources of the AP through the ER. Moreover, in case that AP is willing to provide storage services for the grid, such as accepting to store excess energy from the grid or helping the grid with local power during peak hours (excess consumption which may bring congestions on lines or on the MV/LV transformer), LESSAg is able to ensure local energy needs while providing also energy storage services which have been requested from a grid integrator, such as Grid Energy Storage System Control (GESSCon). Such storage services can be received by LESSAg as MQTT messages of storage orders for selected periods of time (15-, 30- or 60-minutes timeframes), allowing technical services to be implement with different business models. In this way, LESSAg is balancing local needs with external services which have been previously accepted on voluntary basis.

In order to be compatible with the project applications, LESSAg is providing local data in JSON format and sending through MQTT messaging towards the DSF engine. For this, a dedicated connector is provided by LESSAg, which sends data towards DSF through the SMX Broker.

Moreover, LESSAg has an additional connector for being able to make HIL connections, allowing complex tests with simulated grid and in various production and consumption patterns. This connector is described in the following section.

4.3.5 Description of new interfaces for storage systems

4.3.5.1 HIL concept for low TRL tests

The low Technology Readiness Levels (TRL) applications, as it is the Bucharest test site demonstrator of AP, need comprehensive analysis which cannot be done in real grid environment, due to compliance to the grid codes and to potential influence to other consumers connected to it.

For this purpose, the interaction between the AP and the grid is difficult to be assessed. In addition, the UPB network is designed for a much higher load than actually is presently used, thus bringing a situation where the influence in the grid of a small demonstrator cannot be valued.
Moreover, as the project is focusing on different use-cases, including Storage as a Service (SaaS) transactions and executed services, it is needed to have more than one actor in the tested grid, and a second or a third is not available in the trial.

As an additional aspect, and list is open, meaningful load profiles and Renewable Energy Sources (RES) production are needed to be merged, in order to value different AP situations, from comfortable situations up to those which use the energy resources at their limit. It is therefore needed to make possible contexts of loads, production and storage evolutions to allow a realistic assessment and Key Performance Indicators (KPIs) deduction.
Complex trial tests can be therefore accomplished by combining real-time control of ER with different scenarios of production, consumption and storage scheduling, including for providing SaaS, also being embedded in local AC grids which are meaningful for the use-cases.
The test and their evaluation asked therefore for a more complex setup, which combines ER and LESSAg, as real systems, with appropriate environment to evaluate appropriate situations. This requirement brought the necessity of a HIL approach.
UPB has already a network simulation tool based on Open Distribution System Simulator (OpenDSS) load-flow program and on Grid Monitoring and Control Knowledge (GridMonK) application, the latest being an open source software with its basic functionality developed by UPB.
In order to enable such HIL-oriented tests, UPB decided to develop and adapt the following components:

- **LESSAg**: embedded and made compatible in SMXCore, needs a HIL connector to deal with an external real-time simulation tool.
- **GridMonK**: as interface between the grid simulation tool OpenDSS and users, need also an adapter to combine off-line simulations with on-line connection with the ER and LESSAg.

With ER as real device (the tested hardware) and OpenDSS-GridMonK as real-time simulation device, the loop is completed and allows HIL tests.

4.3.5.2 HIL connector for AP demonstrator

The HIL connector for AP demonstrations is embedded in the LESSAg application and is using SMXcore technologies for interacting with OpenDSS-GridMonK external system.
For this purpose, a set of data needed for the UPB tests is provided by LESSAg in the general database of SMXcore, using the database centric approach of SMXcore. Moreover, with essential LESSAg data in the database, additional functionalities are tailored for the purpose of the HIL demonstration. Specific data are

recorded on the disc, for evaluation purposes, while specific data is exchanged between HIL connector and external OpenDSS-GridMonK, in order to close the loop.
The HIL connector allows special interaction between LESSAg and the external simulation system. While in normal mode the behaviour of LESSAg is decided through the parameterisation file of LESSAg, which is stored on SMX, the HIL connector allows to override different parameters. The following main information can be changed through the HIL connector:

- **Type of strategy for the storage use**: with acronym "EMS_strategy", having as default value "UniRCon", which means that the AP has no generation back to the grid. This parameter can have also the status "PV_forecast_driven" or "PV_CONS_forecast_driven".
- **Storage service order**: for both requested power storage in case of excess power in the grid or requested power production towards the grid, in case it is needed to be performed for a storage resources integrator.
- **Consumption measurement**: which can be either the real consumed power measured locally or a pre-recorded consumption profile which can be overwritten in the LESSAg module. In addition, a scaling factor can be used for modifying the consumption power while the load profile keeps similarity with the real power profile.
- **PV production measurement**: which can be either the real produced power measured locally by ER or a pre-recorded generation profile which can be overwritten in the LESSAg module. In addition, a scaling factor can be used for modifying the production power while the generation profile keeps similarity with the real generated power profile.

Through this connector it is possible to make different combinations between real and simulated data.
The interaction is made through an additional MQTT client which accepts only data exchange specific to HIL interaction. The external system is described in the next section.

4.3.5.3 External system design for HIL purpose

As previously mentioned, an OpenDSS-GridMonK package is used to orchestrate the HIL process. Figure 9 shows the concept of HIL and how the different components interact.

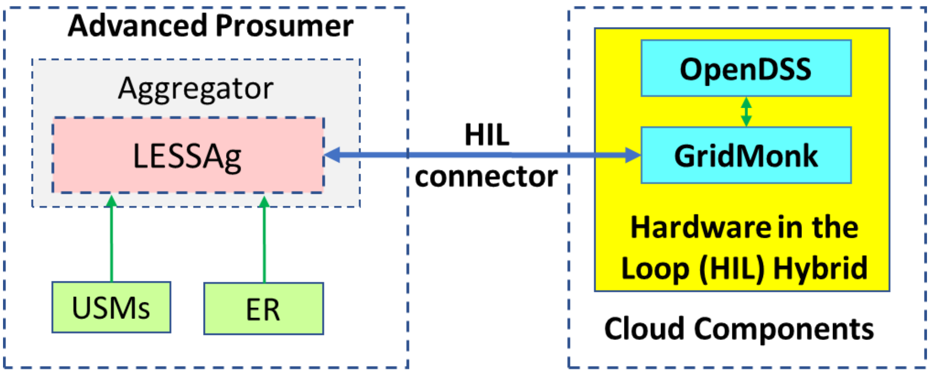

Figure 9. HIL simulation for the AP.

The interaction with LESSAg is provided through the HIL connector. The MQTT messages are exchanged between LESSAg and the external cloud component GridMonK / OpenDSS. GridMonk exchanges specific data with LESSAg, as previously described, through a specific connector on GridMonK, which acts as orchestrator of the HIL process, including interaction with OpenDSS in order to simulate the distribution grid.

4.4 DSF-SE Hybrid Simulator Connector

The Hybrid Simulation integrates physical devices (ER and ESS) in test sites as HIL elements. The ESS and ER are being monitored and controlled trough already developed interfaces with the LiBal System and SMX. The

Hybrid simulator requires a publish-subscribe messaging pattern that handle measurements and setpoints from hybrid simulator.

In order to provide the optimization functionalities in real-time, the hybrid simulator is composed by two main components, namely the *realtimesimulator* and the *ER connector*. The ER connector is capable of, by subscribing to the south-bound ER connector MQTT data flow, performing messages aggregation of the overall sensed value received over different input messages and by calculating the average of the received sequence of values. This approach allows to configure a programmable down sampling of the data generated by the real hardware devices to face different scenarios. The aggregation functionality is required because the hybrid algorithm requires an analysis of simultaneous measurements about each one of the sensed values. Afterwards, the interface forwards the resulting values to the realtimesimulator component via MQTT. This core component performs the optimization of interest and publishes the control messages towards the remote ER exploiting its south-bound ER connector (described in D4.10 [S4G-D4.10]. A general overview of S4G Hybrid Simulation is depicted in Figure 10.

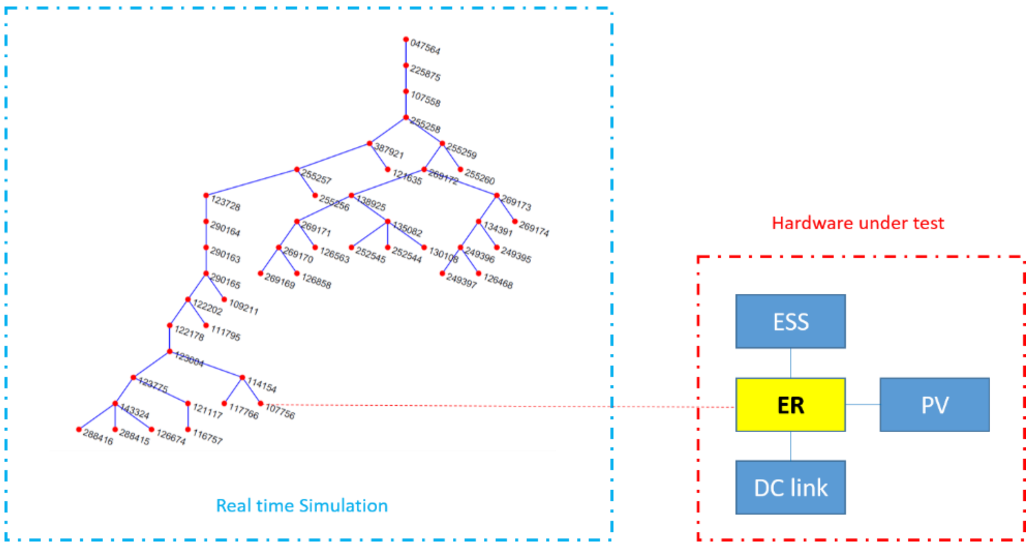

Figure 10. Hybrid HIL simulation.

4.5 DSF Predictive Models Connector

The Predictive models' connector provides different forecast information dynamically generated exploiting real-time and historical data collected by the S4G project. It exposes the DSF predictive models to other S4G components. Details about the predictive models are described in D5.7 [S4G-D5.7].
The forecast information is available within the S4G environment exploiting a dedicated REST backend. These functionalities are exposed to provide updated inputs to the cloud services enabling easy system integration. In particular, the predictive models exposed are about:

- Load (consumption) forecasts.
- EV (consumption) forecasts.
- PV (production) forecasts.
- ESS status forecasts.

4.5.1.1 DSF-Load connector

Concerning the S4G differences over the involved test sites, the loads (provided as aggregated load values) on the EDYNA residential test site is estimated as function of the overall loads present. The aggregated results are exposed in a JSON format.

4.5.1.2 DSF-EVA connector

On the EDYNA commercial test site, the aggregated load is estimated as function of the loads introduced by the EVs CU. The aggregated results are exposed in a JSON format.

4.5.1.3 DSF-PV connector

Whenever the current connector receives a solar irradiation forecast request about a specific time and geographical area, it gets use of the information received from third-parties (as mentioned in section 4.2.2) and calculates the day-ahead irradiation. The process of this calculation method is explained in the D5.7 [S4G-D5.7]. An example of such prediction result can be found in Figure 11. This irradiation then will be accessed from the service caller using the DSF-PV connector. The results are provided in a JSON format.

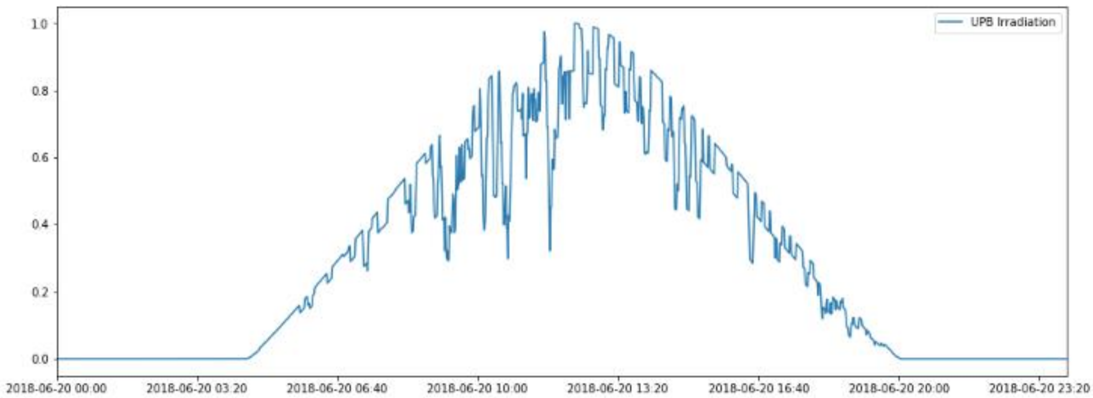

Figure 11. Normalized solar irradiation forecast by DSF predictive model.

4.5.1.4 DSF-ESS forecast connector

The ESS forecast models developed within the S4G project have been produced only on paper, due to the lack of necessity of these forecasts, and are described in D5.7 [S4G-D5.7].

5 Installation/Deployment instructions

5.1 DSO Connectors

5.1.1 DSO SCADA System

This connector was developed as a Java application and its installation is not need in an Operating System (OS) with Java. The .lg4 must be in the same folder as the Java application. Then, the connector enters the S4G VPN and starts to send the.lg4 file information to the DSF.

5.1.2 DSO Grid Models

The DSO grid model connector exposed contents have been embedded inside the DSF Connector server. In order to activate the quoted features, it is necessary to build and run the docker container on the cloud server exploited to provide the main shared and centralized services within the S4G VPN environment.

5.1.3 DSO EV

The DSO EV connector has been built as a standalone docker service. In order to activate the quoted features, it is necessary to build and run the docker container on the cloud server exploited to provide the main shared and centralized services within the S4G VPN environment.

5.2 Third-party System Connectors

5.2.1 Geolocation

The Geolocation features have been installed inside the DSF connector server from the GitHub cloud repository. In order to activate the quoted features, it is necessary to add the required GeoPy libraries inside the containers dockerfile requirements on the server exploited to provide the necessary services.

5.2.2 Weather Forecast

The Weather Forecast connector has been embedded inside the DSF connector server providing services via dedicated endpoints. In order to activate the quoted features, it is necessary to build and run the docker container on the cloud server exploited to provide the main shared and centralized services within the S4G VPN environment.

5.2.3 Energy Price

The Energy Price connector has been developed within DSF connector server via dedicated endpoints. In order to activate the quoted features, it is necessary to build and run the DSF connector docker container on the cloud server exploited to provide the main shared and centralized services within the S4G VPN environment.

5.2.4 PROFESS/PROFEV Solar Radiation

The installation/deployment instruction of the PROFESS/PROFEV solar radiation connector was fully described in the previous version of this deliverable (D5.4 [S4G-D5.4]).

5.3 Control Connectors

5.3.1 ER

The installation/deployment instruction of the ER connector is fully described in D4.10 [S4G-D4.10].

5.3.2 LiBal System

The LiBal System is a commercial product and its installation and deployment is made by the company technicians, which is not in the scope of this project.

5.3.3 LESSAg

LESSAg is an independent agent which is integrated in the SMXcore package in order to harmonize its interactions with SMXcore functionalities, therefore it is pre-installed with the SMXcore package. The functionality of LESSAg is described in a configuration file named "UniRConEMS_Conf". An excerpt of the information available in the LESSAg configuration file is as follows:

```
#UniRConEMS strategy
############ EMS parameters ###########
EMS_strategy=UniRCon
### E_bat_nominal [Wh] ####
E_bat_nominal=1000
### P_BAT_Nominal, meaning the power of inverters [W]
P_bat_max_inverters=500
```

Figure 12. Excerpt of a UniRConEMS_Conf file.

This excerpt describes the LESSAg input parameters, such as its strategy for handling the storage means and the nominal values of energy and power for the battery.

5.3.4 HIL

HIL is provided by GridMonK orchestrator which is interacting with the load-flow engine OpenDSS. The GridMonK orchestrator has a "HIL_Conf" file for the parameterisation, which includes the description of MQTT channels which allows remote connection with LESSAg. An excerpt of the information available in the LESSAg configuration file is available in Figure 13. This excerpt describes the LESSAg input parameters, such as the broker address for connecting to HIL functionality of LESSAg, scaling factors for values of the different powers, to be used in the simulated grid based on real data shapes, and the log file name settings.

```
MQTT_HIL_Broker=10.8.0.11
Cons_scaling_factor=1
Bat_scaling_factor=1
PV_scaling_factor=1
HIL_log=HIL_log.txt
```

Figure 13. Excerpt of a HIL_Conf file.

The simulated grid topology and nodes characteristics are described in a file with "dss" extension, having a syntax specific to OpenDSS software.

5.4 DSF-SE Hybrid Simulator Connector

The DSF-SE Hybrid Simulator connector is composed by two different developed components. In order to activate the quoted features, it is necessary to properly configure and run the python scripts exploited to provide these optimization features. The configuration phase it is only related to the IP addresses of the RealTimeSimulator, the ER connector and the south-bound ER connector brokers, other than the security parameters required. Consequently, in order to run these solutions, it is necessary to run only the main scripts exploiting the python3 interpreter.

5.5 DSF Predictive Models Connector

The set of predictive models developed within the S4G project have been embedded inside the DSF connector server. In order to activate the quoted features, it is necessary to build and run the docker container on the cloud server exploited to provide the main shared and centralized services within the S4G VPN environment. The quoted container contains all the necessary libraries and source code to effectively provide the services built.

6 Software dependencies and requirements

The connectors are designed to be lightweight, and therefore do not have special requirements in terms of RAM or processing power. Table 1 presents the connectors' software dependencies.

Table 1. Software Dependencies.

Dependency	License	Role
influxdb[viii]	MIT License	Part of DSF-DWH: Scalable time-series database for metrics, events, and real-time analytics
telegraf[ix]	MIT License	Part of DSF-DWH: The plugin-driven server agent for collecting & reporting metrics.
GeoPy[vi]	MIT License	Part of DSF connectors: provide geolocalization functionalities.
j60870[x]	GPLv3	This software is used to connect the DSO SCADA with the DSF DSO SCADA connector enabling the data exchange/decoding.

7 API Reference

7.1 DSO Connectors

7.1.1 DSO SCADA System

This connector does not have an API to interact with, since is reading an IEC 61870-5-104 standard compliant .lg4 file.

7.1.2 DSO Grid Models

The REST API provided by the connector rely on the grid data provided by DSO. The grid models are grouped together and divided by supplier:

- EDYNA
- ENIIG

For privacy reasons the quoted information is available only through the S4G VPN.
The format accepted to request the EDYNA grid is the following:

- http://10.8.0.50:18081/EDYNA/grid

It returns a JSON response with the entire EDYNA RESIDENTIAL grid.
Currently, the RESIDENTIAL information could be obtained by requesting punctually specific subsets data of interests, such as:

- lines: http://10.8.0.50:18081/EDYNA/lines
- linecodes: http://10.8.0.50:18081/EDYNA/linecodes
- loadshapes: http://10.8.0.50:18081/EDYNA/loadshapes
- loads: http://10.8.0.50:18081/EDYNA/loads
- nodes: http://10.8.0.50:18081/EDYNA/nodes
- PV_absorb_effs: http://10.8.0.50:18081/EDYNA/PV_absorb_effs
- PVs: http://10.8.0.50:18081/EDYNA/PVs
- PV_temp_effs: http://10.8.0.50:18081/EDYNA/PV_temp_effs
- source: http://10.8.0.50:18081/EDYNA/source
- storages: http://10.8.0.50:18081/EDYNA/storages
- substations: http://10.8.0.50:18081/EDYNA/substations
- transformers: http://10.8.0.50:18081/EDYNA/transformers

Information about the EDYNA COMMERCIAL grid are also provided via dedicated API:

- evs: http://10.8.0.50:18081/EDYNA/commercial/evs
- feeders: http://10.8.0.50:18081/EDYNA/commercial/feeders
- lines: http://10.8.0.50:18081/EDYNA/commercial/lines
- linecodes: http://10.8.0.50:18081/EDYNA/commercial/linecodes
- loadshapes: http://10.8.0.50:18081/EDYNA/commercial/loadshapes
- loads: http://10.8.0.50:18081/EDYNA/commercial/loads
- nodes: http://10.8.0.50:18081/EDYNA/commercial/nodes
- PVs: http://10.8.0.50:18081/EDYNA/commercial/PVs
- source: http://10.8.0.50:18081/EDYNA/commercial/source
- storages: http://10.8.0.50:18081/EDYNA/commercial/storages
- transformers: http://10.8.0.50:18081/EDYNA/commercial/transformers

Currently, the ENIIG grid information is only provided by requesting punctually specific subsets data of interests, such as:

- lines: http://10.8.0.50:18081/ENIIG/lines
- linecodes: http://10.8.0.50:18081/ENIIG/linecodes
- loadshapes: http://10.8.0.50:18081/ENIIG/loadshapes
- loads: http://10.8.0.50:18081/ENIIG/loads
- nodes: http://10.8.0.50:18081/ENIIG/nodes
- PV_absorb_effs: http://10.8.0.50:18081/ENIIG/PV_absorb_effs
- PVs: http://10.8.0.50:18081/ENIIG/PVs
- PV_temp_effs: http://10.8.0.50:18081/ENIIG/PV_temp_effs
- source: http://10.8.0.50:18081/ENIIG/source
- transformers: http://10.8.0.50:18081/ENIIG/transformers

7.1.3 DSO EV

The current connector exposes three different types of REST API:

1. Public endpoints to interact with SIEMENS energy service
2. One private endpoint to enable modulation commands from the South Bound EV connector
3. Private endpoints for debugging purposes

The public endpoints are the following:

- CU registration: https://dwh.storage4grid.eu:8082/EnergyService/api/cu
- CU modification: https://dwh.storage4grid.eu:8082/EnergyService/api/cu/detail/
- CU removal: https://dwh.storage4grid.eu:8082/EnergyService/api/cu/remove/
- CU status: https://dwh.storage4grid.eu:8082/EnergyService/api/event/recharge/push/
- Recharge start: https://dwh.storage4grid.eu:8082/EnergyService/api/event/recharge/new
- Recharge update: https://dwh.storage4grid.eu:8082/EnergyService/api/event/recharge/update
- Recharge end: https://dwh.storage4grid.eu:8082/EnergyService/api/event/recharge/remove

The private endpoint required to enable modulation command forwarding is the following:

- Modulate recharge: https://dwh.storage4grid.eu:8082/S4G/api/command/module

The private endpoints required for debugging are the following:

- Show overall status info: https://dwh.storage4grid.eu:8082/S4G/status/list
- Show single CU status info: https://dwh.storage4grid.eu:8082/S4G/status/cu/<id>
- Show single meter status info: https://dwh.storage4grid.eu:8082/S4G/status/meter/<id>
- Show single CU info: https://dwh.storage4grid.eu:8082/S4G/cu/<idCU>
- Show single meter info: https://dwh.storage4grid.eu:8082/S4G/meter/<idMeter>
- Show overall plug info: https://dwh.storage4grid.eu:8082/S4G/plug/list
- Show single plug info: https://dwh.storage4grid.eu:8082/S4G/plug/<idPlug>
- Show single session info: https://dwh.storage4grid.eu:8082/S4G/session/<idSession>
- Show overall running session: https://dwh.storage4grid.eu:8082/S4G/active/sessions
- Show overall started session: https://dwh.storage4grid.eu:8082/S4G/start/sessions
- Show single session info: https://dwh.storage4grid.eu:8082/S4G/start/session/<sessionID>
- Show overall update info: https://dwh.storage4grid.eu:8082/S4G/update/sessions
- Show overall session end info: https://dwh.storage4grid.eu:8082/S4G/end/sessions
- Show single session end info: https://dwh.storage4grid.eu:8082/S4G/end/session/<sessionID>

7.2 Third-party System Connectors

7.2.1 Geolocation

The provided geolocation feature exploits the GeoPy client. It provides functionalities for python applications only.

7.2.2 Weather Forecast

The current endpoint relies on the weatherunlocked third-party service. The format accepted of the GET request is:

- http://10.8.0.50:18081/weather/{lat},{lon}/

The {lat} accepted values are:

- Minimum: -90.00
- Maximum: +90.00

If no sign is given, positive value is assumed.

The {lon} accepted values are:

- Minimum: -180.00
- Maximum: +180.00

If no sign is given, positive value is assumed.

It returns a JSON response with the weather forecast about the given location for an entire week.

7.2.3 Energy Price

This connector relies on the energy price data provided dynamically after each request by the transparency.entsoe.eu third-party service. The selected time interval for the location of interest is about the current day.

The list of APIs provided are the following:

- http://10.8.0.50:18081/EDYNA/commercial/prices
- http://10.8.0.50:18081/ENIIG/commercial/prices
- http://10.8.0.50:18081/GENERIC/{AREA-CODE-ID}/prices
- http://10.8.0.50:18081/{latitude},{longitude}/{type}/prices
- http://10.8.0.50:18081/{latitude},{longitude}/prices/{date_from}/{date_to}

The JSON values returned by the first API (EDYNA commercial prices) forces requests about the current pricing about ITALY (10Y1001A1001A885).
The JSON values returned by the second API (ENIIG commercial prices) forces requests about the current pricing about DENMARK (10YDK-1--------W).
The JSON values returned by the third-party API (GENERIC prices) forces requests about the current pricing about the given area.
The JSON values returned by the fourth API will automatically convert the given latitude and longitude values to the country of interest and then to the transparency related pricing scheme.
The JSON values returned by the fifth API will automatically convert the given latitude and longitude values to the country of interest and then to the transparency related pricing scheme and will return the pricing values of interest in the given time window.
In case of values outside known countries or in case of country not considered in the current project, the API will respond with an error.
The type value quoted accept "residential" or "commercial" (or other strings values).
This option does not affect the evaluation because the information is not transparency related.
The list of accepted codes about areas of interest ({AREA-CODE-ID}) are provided at the following link:

- https://transparency.entsoe.eu/content/static_content/Static content/web api/Guide.html

The pricing values returned (as pointed out by the JSON content) are expressed in [EUR/MWh].

7.2.4 PROFESS/PROFEV Solar Radiation

The API of the PROFESS/PROFEV solar radiation connector was fully described in the previous version of this deliverable (D5.4 [S4G-D5.4]).

7.3 Control Connectors

7.3.1 ER

The API of the ER connector is fully described in D4.10 [S4G-D4.10].

7.3.2 LiBal System

The LiBal System is a commercial product and interfaces with external systems using the GESSCon API defined in D4.5 [S4G-D4.5].

7.3.3 LESSAg

The LESSAg connector to DSF provides MQTT messages which are regularly sent (each 2 seconds) to the SMX broker and are then taken by OGC Wrapper and sent to DSF.
The data structure of the information sent regularly (default each 2-seconds) to the SMX Broker is presented with the generic format followed described.

```
{
  "OBIS_code_1": {
    "-2": "value",
    "-5": "date",
    "-9": "Calculated_value"
  },
  "OBIS_code_2": {
    "-2": "value",
    "-5": "date",
    "-9": "Calculated_value"
  },
. . . . . . . ,
  "OBIS_code_n": {
    "-2": "value",
    "-5": "date",
    "-9": "Calculated_value"
  }
}
```

Where:

- **OBIS_code_k** is the OBject Identification System (OBIS) code of the meter measurement, which can be for instance "1-1-32-7-0-255" for a voltage code of phase A.
- **value** is the numerical value of the measurement, in text format, e.g., "231.3".
- **date** has the format of type "03/14/2019 04:25:41"
- **Calculated_value** is a result obtained from calculations of the data, e.g., when "value" needs to be multiplied by Current Transformer (CT) constant in order to obtain the primary circuit values, while "value" gives the secondary circuit values effectively measured by the meter.

For a certain "OBIS_code_k", "value" is a mandatory field, while the fields "date" and "Calculated_value" appear if the information is available, depending on factors such as meter type and calculation type.
The line with dots (. . . . . . . ,) shows the possibility of existing more objects with the same structure.

With these messages sent through the MQTT connector, DSF receives main data from the meter in the Point of Common Coupling (PCC), with programmable topic which contains the sender name code.

Data for providing storage services can be received by the connector as MQTT message in JSON format, in a vector array with the structure data followed presented.


```
[
  {
  "n": "ESS_Control",
  "t": 1.538697600e+09,
  "v":1.2,
  "u":"kW"
  },
  {
  "n": "ESS_Control",
  "t": 1.538701200e+09,
  "v":0.8,
  "u":"kW"
  },
  {
  "n": "ESS_Control",
   "t": 1.538704800e+09,
  "v":2.5,
  "u":"kW"
  },
        ...
]
```


Where each hourly time period has its own object, with the following meaning of the internal structure:

- **n**: "*ESS_Control*" is the specific key for an hourly object describing the storage service.
- **t**: gives the time corresponding to the beginning of the period when storage service is needed for the grid, e.g. 1.538704800e+09.
- **v**: *value*, gives the storage order (power to be absorbed or injected in the grid, as grid service), e.g., the value 2.5.
- **u**: *unit*, gives the unit of the power, e.g. "kW".

The dots at the end of the array shows that more objects can be present in the array.

7.3.4 HIL

The LESSAg connector to HIL exchanges MQTT messages which are regularly (each 1-minute) or event-based sent / received to/from the HIL, as part of DSF type cloud components. The general format of the messages is followed presented.


```
{
 "Item_ID": {
  "Action_ID": "action_ID_descriptor",
  "Attribute": "attribute_value",
  "Attribute_supl": "attribute_supl_value",
  "Value": "value",
  "Date": "date",
```

```
    "Information": "Suplementary_information"
   },
   "Item_ID": {
    "Action_ID": "action_ID_descriptor",
    "Attribute": "attribute_value",
    "Attribute_supl": "attribute_supl_value",
    "Value": "value",
    "Date": "date",
    "Information": "Suplementary_information"
   },
  . . . . . . . ,
   "Item_ID": {
    "Action_ID": "action_ID_descriptor",
    "Attribute": "attribute_value",
    "Attribute_supl": "attribute_supl_value",
    "Value": "value",
    "Date": "date",
    "Information": "Suplementary_information"
   },
  }
```


Where each object has an “Item_ID” identification and the internal structure is flexible and can contain one or more fields, namely:

- **Action_ID**: describes the type of action, e.g. “Overwrite” or “Data_report”, which allows to overwrite a variable in LESSAg or to report a variable towards HIL.
- **Attribute**: describes the type of data to be overwritten or reported (e.g., the “EMS_strategy”).
- **Attribute_supl** describes additional attributes to be described, such as the minute position in an hour. To be noted that additional fields with "Attribute_supl_k" can be present, as the additional attribute k.
- **Value**: describes the variable value of the “Action_ID” field.
- **Date**: gives a time+date stamp.
- **Information**: provides additional information, depending on “Action_ID” type.

The connector has been implemented in a flexible way, allowing that only part of the object fields are presented, or that some additional fields can be implemented (such as "Attribute_suplk").

The API of HIL is intended to be used only between LESSAg and HIL, thus it has an intimate structure which may evolve/change without affecting the opening for LESSAg towards any other S4G project components. The connector has symmetry in sending and receiving format.

7.4 DSF-SE Hybrid Simulator Connector

The Hybrid Simulator plugin has been built as a set of standalone software components to be installed and run on one or multiple machines to provide the required features. Consequently, this connector API is not available and cannot be used by other services within the project.

7.5 DSF Predictive Models Connector

7.5.1 DSF-Load connector

The REST API enables the integration of the systems with the aggregated Loads related information. Currently, it is filled by dynamic data produced exploiting information from the residential feeder of EDYNA (SCADA).

The format accepted is the following:

- http://10.8.0.50:18081/EDYNA/residential/aggregatedloads/{DateFrom}/{NumberOfDays}

Follows an example of request:

- http://10.8.0.50:18081/EDYNA/residential/aggregatedloads/2019-03-11/1

7.5.2 DSF-EVA connector

The REST API let enables the integration of the EVA System with the deployed devices. The format accepted are the following:

1. http://10.8.0.50:18081/EVA/{WeekDay}
2. http://10.8.0.50:18081/EVA/{OverallNumEV}

The {WeekDay} accepted values are: { Monday / Tuesday / Wednesday / Thursday / Friday / Saturday / Sunday / Today }. The "Today" value will force the connector to start the evaluation of the current day. All the above options will exploit internal predefined configuration built from older analysis. If the request is about a generic weekday, the date exposed will not consider values such as month and year.

The {OverallNumEV} accepted values are:

- Minimum: 1
- Maximum: 99999

It represents the amount of EV charging event to be considered in the current day and returns a JSON response with the EV profiles of interests.

7.5.3 DSF-PV connector

The REST API let users to request a PV production forecast targeting a specific area. The returned values are built consequently to dedicated algorithms exploiting weather forecast information extracted by the weatherunlocked service. Two different options are available:

1. Generic approach with some default values (tilt and horizon_declination) related to PV system deployment.
2. Specific approach with real values related to each PV system deployment.

The format accepted by the Generic API is the following:

- http://10.8.0.50:18081/pv/{dateFrom}/{dateTo}/{lat},{lon}/

Both the {dateFrom} and {dateTo} accepted format is the following:

- YYYY-MM-DD
- YYYY.MM.DD

Note that due to weather forecast service limitations, the time-window accepted must be less than a week.
The {lat} accepted values are:

- Minimum: -90.00
- Maximum: +90.00

If no sign is given, positive value is assumed.
The {lon} accepted values are:

- Minimum: -180.00
- Maximum: +180.00

If no sign is given, positive value is assumed.

It returns a JSON response with the PV production forecast about the given location concerning the dates of interests.
The evaluation will consider the given dates to perform the forecast as described here:

- Evaluation Starts from 00:00 of the {dateFrom} value inserted.
- Evaluation Ends at 00:00 of the {dateTo} value inserted.

- The predefined step between samples is about 60 seconds.

For instance, in order to select the Bolzano's forecast about (only) the day 2019-01-20 it is necessary to perform the following request:

- http://10.8.0.50:18081/pv/2019-01-20/2019-01-21/46.29,11.20/

The format accepted by the Specific API is the following:

- http://10.8.0.50:18081/pv/{dateFrom}/{dateTo}/{lat},{lon}/{tilt}/{horizon_declination}/

The parameters in common with the generic API are managed in the same way.
The {tilt} accepted values are:

- Minimum: 0
- Maximum: +180.00

If no sign is given, positive value is assumed.

The {horizon_declination} accepted values are:

- Minimum: -90.00
- Maximum: +90.00

If no sign is given, positive value is assumed.

It returns a JSON response with the PV production forecast with detailed deployment information enriching simulated results about the given location concerning the dates of interests.
The evaluation will consider the given dates to perform the forecast as described here:

- Evaluation Starts from 00:00 of the {dateFrom} value inserted.
- Evaluation Ends at 00:00 of the {dateTo} value inserted.
- The predefined step between samples is about 60 seconds.

For instance, in order to select the Bolzano's forecast about (only) the day 2019-01-20 it is necessary to perform the following request:

- http://10.8.0.50:18081/pv/2019-01-20/2019-01-21/46.29,11.20/0/90/

The resulting JSON about both requests (Generic and Specific) will contain an array of forecast values in which every value is distant 60 seconds from the previous one. This means that, concerning the current example, the amount of returned values will be 1440.
It is confirmed by dividing the overall number of seconds of a single day (86400) with the predefined step (60): 86400/60 = 1440

Note that, each of the returned value will represent the forecast about the production of a PV system on the specified area. The measurement unit to be considered for each value is: W/m^2

8 Conclusions

This deliverable presented the final results of Task 5.2 - "DSF Interoperability with 3rd party and DSO systems", which main goal was to develop a set of connectors to allow the DSF to operate with third-party and DSO systems. It describes the developed DSF connectors enabling data injection in the DSF-DWH, and routing control messages to the different layers and devices, DSO related information such as grid topologies, providing forecast services from third-party sources, and enabling the DSF predictive models to other S4G components.

The DSF connectors are cloud based and provide information for other S4G components. They can be installed using the installation instructions and the software dependencies available in this document. Their APIs are described enabling their interaction with other S4G components.

Acronyms

Acronym	Explanation
AC	Alternating Current
AP	Advanced Prosumer
API	Application Programming Interface
BMS	Battery Management System
CIM	Common Information Model
CT	Current Transformer
CU	Charging Units
DSF	Decision Support Framework
DSF-DWH	Decision Support Framework Data Warehouse
DSF-SE	Decision Support Framework Simulation Engine
DSO	Distribution System Operator
DWH	Data Warehouse
EMS	Energy Management System
ER	Energy Router
ESS	Energy Storage System
EV	Electric Vehicle
GESSCon	Grid Energy Storage System Control
GridMonK	Grid Monitoring and Control Knowledge
HIL	Hardware-in-the-loop
HTML	HyperText Markup Language
HTTPS	HyperText Transfer Protocol Secure
IEC	International Electrotechnical Commission
IP	Internet Protocol
IT	Information Technology
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
LESSAg	Local Energy Storage System Agent
LV	Low Voltage
MQTT	Message Queue Telemetry Transport
MV	Medium Voltage

Acronym	Explanation
OBIS	OBject Identification System
OCPP	Open Charge Point Protocol
OGC	Open Geospatial Consortium
OpenDSS	Open Distribution System Simulator
OS	Operating System
PCC	Point of Common Coupling
PROFESS	Professional Realtime Optimization Framework for Energy Storage Systems
PROFEV	Professional Realtime Optimization Framework for Electric Vehicles
PV	Photovoltaic
RAM	Random Access Memory
RES	Renewable Energy Sources
REST	Representational State Transfer
S4G	Storage4Grid
SaaS	Storage as a Service
SCADA	Supervisory Control and Data Acquisition
SenML	Sensor Measurement Lists
SMX	Smart Meter eXtension
SoC	State of Charge
SQL	Structured Query Language
TCP	Transmission Control Protocol
TRL	Technology Readiness Levels
UniRCon	Unidirectional Resilient Consumer
USM	Unbundled Smart Meter
VPN	Virtual Private Network

List of figures

List of tables

References

[i] Open Geospatial Consortium, OGC SensorThings API, https://www.opengeospatial.org/standards/sensorthings, accessed 6 August 2019.
[ii] International Electrotechnical Commission, IEC 60870-5-104:2006+AMD1:2016 CSV, Consolidated version, Telecontrol equipment and systems - Part 5-104: Transmission protocols - Network access for IEC 60870-5-101 using standard transport profiles, https://webstore.iec.ch/publication/3746, accessed 12 August 2019.
[iii] IPCOMM, Fink WinPP Protocol Simulator, https://www.ipcomm.de/product/FinkWinPP/en/sheet.html, accessed 21 August 2019.
[iv] Open Charge Alliance, Open Charge Point Protocol (OCPP), https://www.openchargealliance.org/, accessed 29 August 2019.
[v] Sensor Measurement Lists (SenML), https://tools.ietf.org/html/rfc8428, accessed 29 August 2019.
[vi] Geopy – PyPi, geopy 1.20.0, https://pypi.org/project/geopy/, accessed 29 August 2019.
[vii] WeatherUnlocked, Weather-Based Solutions for Advertisers, Retailers, and Developers, http://www.weatherunlocked.com/, accessed 29 August 2019.
[viii] InfluxDB, InfluxDB is the Time Series Database in the TICK Stack, https://www.influxdata.com/time-series-platform/influxdb/, accessed 29 August 2019.
[ix] Telegraf, Telegraf is the Agent for Collecting and Reporting Metrics & Data, https://www.influxdata.com/time-series-platform/telegraf/, accessed 29 August 2019.
[x] OpenMUC, j60870 Overview, https://www.openmuc.org/iec-60870-5-104/, accessed 29 August 2019.