D4.5 - Final Grid-side ESS Control System

Deliverable ID	**D4.5**
Deliverable Title	**Final Grid-side ESS Control System**
Work Package	**WP4**
Dissemination Level	**PUBLIC**
Version	**1.0**
Date	**2019/09/13**
Status	**Final**
Type	**Prototype**
Lead Editor	**LIBAL**
Main Contributors	**LIBAL**

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	2019-06-03	LiBal	First draft including TOC and most of the contents
0.2	2019-06-18	LiBal	Modification to section 4, 7-10
0.3	2019-06-19	LiBal	Add Site Controller section
0.4	2019-06-24	LiBal	Add new figures to section 4
0.5	2019-07-02	EDYNA	Review
0.6	2019-07-02	LiBal	Reply to the review questions and suggestions by EDYNA
0.7	2019-07-02	EDYNA	Reply to v0.6
0.8	2019-07-03	LiBal	Reply to the review by FIT (based on v0.4) Merge EDYNA and FIT's review
0.9	2019-08-01	LiBal	Modification on section 3-5 according to EDYNA and FIT's comments
1.0	2019-09-13	LiBal	Final version, ready for submission to the EC

Internal Review History

Review Date	Reviewer	Summary of Comments
2019-07-02 (v0.4)	FRAUNHOFER	Not approved: • Changes in the text • Information missing
2019-07-02 (v0.5)	EDYNA	Approved: • General minor corrections • Some comments to have a better comprehension
2019-07-03 (v0.8)	FRAUNHOFER	Not approved: • Changes in the text • Information missing
2019-09-XX (v1.0)	FRAUNHOFER	Approved:

Table of Contents

Executive Summary

D4.5 – "Final Grid-side ESS control system" presents the prototype of the grid side control system and its architecture. The grid-side control system has been divided by 2 types of control algorithm which are used in different use cases in the project: optimal control is used in Bolzano test site (residential and commercial) and Skive simulation case, on use cases HLUC2-PUC-1, HLUC2-PUC-2 and HLUC3-PUC-4 [D2.2]; andlocal real-time control is used in Skive grid-side storage test site, on use case HLUC3-PUC-3 [D2.2].

Optimal grid-side ESS control system (GESSCon) is a decision-making software component running in LiBal's Cloud that is deciding the energy storage system charge/discharge behavior in the next 24 hours based on current grid situation and some relevant forecast information. It works together with PROFEV/PROFESS (local agent) for a better control of the substation level.

Three types of Input data (Forecast inputs, telemetry input and site configuration input) are supported by DSF (Decision Support Framework) and SMX (Smart Meter Extension) through REST API and MQTT using the Data Broker [D3.2]. The output data – ESS (Energy Storage System) control setpoints for the next 24-hours, will be sent to PROFEV/PROFESS through the Control Broker. PROFEV/PROFESS [D4.3][D4.6] will execute these control setpoints based on its own local knowledge.

Local grid-side ESS control system (Site Controller) is a hardware unit located close to the energy storage system in Skive. It collects data from energy storage and exchange information with GESSCon and other Cloud components such as DSF. Site Controller is not primarily developed for S4G, but a control algorithm for Skive grid-side storage (HLUC3-PUC-3) is developed within the project.

This deliverable describes the final version of the grid-side ESS control system prototypes which includes GESSCon software and Site Controller control algorithm. Minor modifications and improvements might need to be implemented/developed during phase 3 integration and deployment actions.

1 Introduction

This document describes the grid-side energy storage system control system prototype in Storage4Grid project. Optimal grid-side ESS control system (GESSCon) is a decision-making software that take both grid status and local energy storage system condition into account and make charge/discharge suggestions for the next day. GESSCon will deliver a 24h control setpoints to the local control system (PROFEV/PROFESS) every day. Therefore, PROFEV/PROFESS [D4.3][D4.6] will control the energy storage system based on this schedule and its own optimization algorithms. Local grid-side ESS control (Site Controller) algorithm for Skive grid-side energy storage is also included in this deliverable.

1.1 Scope

This prototype deliverable has been developed under Task T4.2 – "Grid-side ESS control".

1.2 Related documents

ID	Title	Reference	Version	Date
[RD.1]	Final Storage Scenarios and Use Cases	D2.2	1.0	2018-07-31
[RD.2]	Updated S4G Components Interfaces and Architecture Specification	D3.2	1.5	2018-08-08
[RD.3]	Final User-side ESS Control System	D4.3	1.0	2019-06-13
[RD.4]	Initial Grid-side ESS Control System	D4.4	1.0	2018-08-30
[RD.5]	Initial Cooperative EV charging station control algorithms	D4.6		
[RD.6]	Updated DSF Connectors for external systems and services	D5.4	1.0	2018-09-03
[RD.7]	Final DSF Predictive Models	D5.7	0.7	2019-05-14

2 Final optimal grid-side ESS control system prototype overview

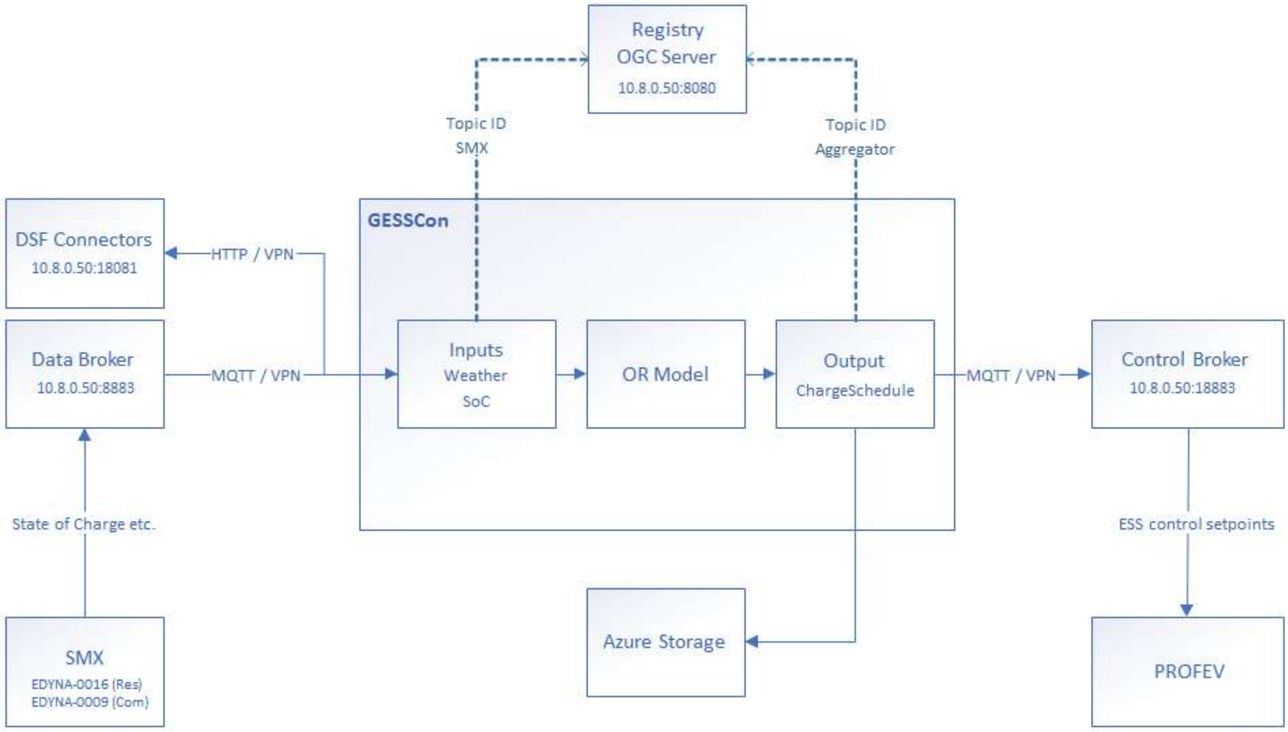

Figure 1. GESSCon architecture overview

GESSCon is formed by GESSCon Core and its input and output handlers to other components in the Storage4Grid system. As illustrated in Figure 1, GESSCon Core is composed of an input and output handler which has interactions to Data Broker, DSF connectors, OGC server and Control Broker within the project. Data Broker, DSF connectors and Control Broker are used for receiving inputs and sending outputs, OGC server is used as a "map" for GESSCon to locate test sites and find the correct MQTT topics.

2.1 GESSCon#OpenVPN

The S4G systems used by GESSCon must be accessed with VPN. GESSCon uses OpenVPN with obtained certificates from the S4G project.

2.2 GESSCon#OGC Server

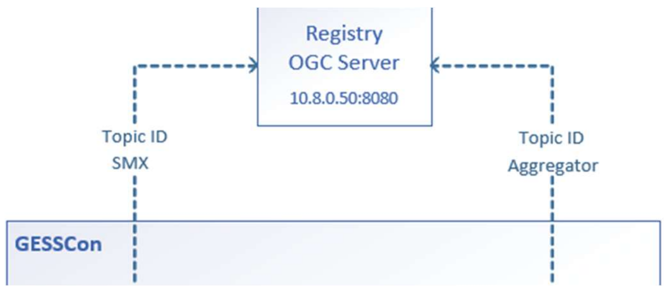

Figure 2. GESSCon interfaces to OGC server

As show in Figure 2, The OGC Server[i] is a REST API compliant with the OGC open source standard. The S4G project implements the SensorThings[ii] layer of the standard, which contains metadata about locations, things, sensors and datastreams. The OGC Server is located at 10.8.0.50:8080.

A VPN connection must be established (as described in section 2.1). The GESSCon uses the OGC Server to look up the broker topics to subscribe, as shown in Figure 3.


```
// 20180927163100
// http://10.8.0.50:8080/v1.0/Things(20)/Datastreams?$filter=contains(name,%27Control%27)

{
  "value": [
    {
      "@iot.id": 92,
      "@iot.selfLink": "http://10.8.0.50:8080/v1.0/Datastreams(92)",
      "name": "EDYNA-0013_1_Control_S4G-GW-EDYNA-0013",
      "description": "Electrical Flow about Photovoltaic System",
      "unitOfMeasurement": {
        "definition": "http://S4G-Measurement-Electrical.eu",
        "name": "Electrical",
        "symbol": "E"
      },
      "observationType": "http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_Measurement",
      "Thing@iot.navigationLink": "http://10.8.0.50:8080/v1.0/Datastreams(92)/Thing",
      "Sensor@iot.navigationLink": "http://10.8.0.50:8080/v1.0/Datastreams(92)/Sensor",
      "Observations@iot.navigationLink": "http://10.8.0.50:8080/v1.0/Datastreams(92)/Observations",
      "ObservedProperty@iot.navigationLink": "http://10.8.0.50:8080/v1.0/Datastreams(92)/ObservedProperty"
    },
    {
      "@iot.id": 91,
      "@iot.selfLink": "http://10.8.0.50:8080/v1.0/Datastreams(91)",
      "name": "EDYNA-0013_Aggregator chunk_Control",
      "description": "Stream limited to Aggregator<->Cloud Communication",
      "unitOfMeasurement": {
        "definition": "S4G-dedicated",
        "name": "data"
```


Figure 3. OGC server look up.

GESSCon queries OGC server everytime about the broker topics of a specific test-site during operation time and uses the obtained topic IDs to fetch relevant site telemetry data from the Data Broker. After GESSCon finishes its internal calculations, it will query OGC server again about the control broker topic ID for the same test-site and use this ID to send ESS control setpoints.

2.3 GESSCon#DSF connectors

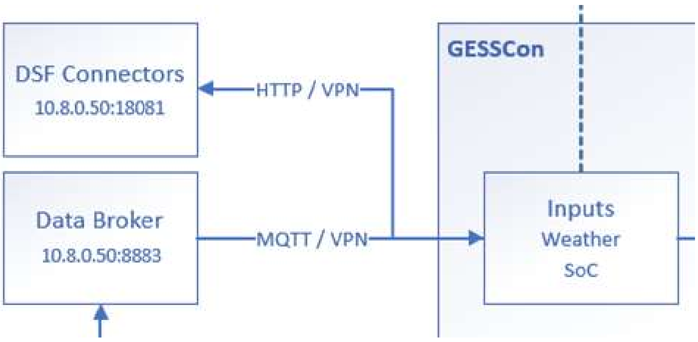

Figure 4. GESSCon interfaces to DSF connectors and Data Broker

The DSF Connectors are REST APIs located at 10.8.0.50:18081. The GESSCon uses the DSF Connectors to fetch “Raw Data” of electricity prices, Demand (load and EV) and PV production for each site. A brief overview is provided in Figure 4. More descriptions are available in Table 1.

Data Source	Url	Comment
Electricity Prices	http://10.8.0.50:18081/{latitude},{longitude}/prices/{date_from}/{date_to}	Site location in latitude and longitude has been fetched from OGC Server.
Demand	http://10.8.0.50:18081/EDYNA/residential/aggregatedloads/{DateFrom}/{NumberOfDays}	Demand forecast is only available for Bolzano residential site. It is assumed that there is no additional load than EV charging station in Bolzano commercial site.
EV	http://10.8.0.50:18081/EVA/{WeekDay}	Since EV forecast data are static, EV connector is only providing data for every single day. The data horizon is from 00:00 to 23:45 with 15 minutes resolution.
PV	http://10.8.0.50:18081/pv/{dateFrom}/{dateTo}/{lat},{lon}/	PV data has 1-minute resolution.

Table 1. Forecast data endpoints provided by DSF connector

2.4 GESSCon#Data Broker

The Data Broker is a topic-based broker using MQTT protocol, and is located at 10.8.0.50:8883. A VPN connection must be established (as described in section 2.1) to use the Data Broker.
The GESSCon uses the Data Broker to subscribe to battery system datastream topics (as described in section 2.2) to read for example, latest SoC and other values, as show in Figure 4. Currently due to the implentation of energy storage data connector is not ready, mocked SoC is used for GESSCon operation. Once the connection is in place, real SoC will be feed into GESSCon.

2.5 GESSCon#Control Broker

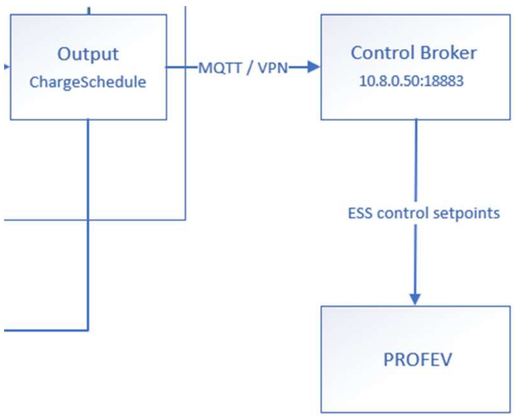

Figure 5. GESSCon interfaces to Control Broker

The Control Broker is a topic-based broker using MQTT protocol, and is located at 10.8.0.50:18883 as shown in Figure 5. A VPN connection must be established (as described in section 2.1) to use the Control Broker.

The GESSCon uses the Control Broker to publish to the PROFESS/PROFEV's control datastream topic (as described in section 2.2), in order to deliver the computed setpoints. The format of the setpoint structure is compliant with the SenML standard[iii].

An example is shown below, where "t" is an epoch (as described in Appendix A) formatted timestamp:


```
[{
	"n": "GESSCon Profile",
	"t": 1.538697600e+09,
	"v": 0.0,
	"u": "kW"
},
{
	"n": "GESSCon Profile",
	"t": 1.538701200e+09,
	"v": 0.0,
	"u": "kW"
},
...
{
	"n": "GESSCon Profile",
	"t": 1.538712000e+09,
	"v": 0.0,
	"u": "kW"
}]
```


2.6 GESSCon#DSF-SE

In Storage4Grid project besides real site ESS control, GESSCon also participates in a simulation use case HLUC3-PUC-3 [D2.2], where GESSCon will cooperate with simulation engine (DSF-SE) for grid planning.

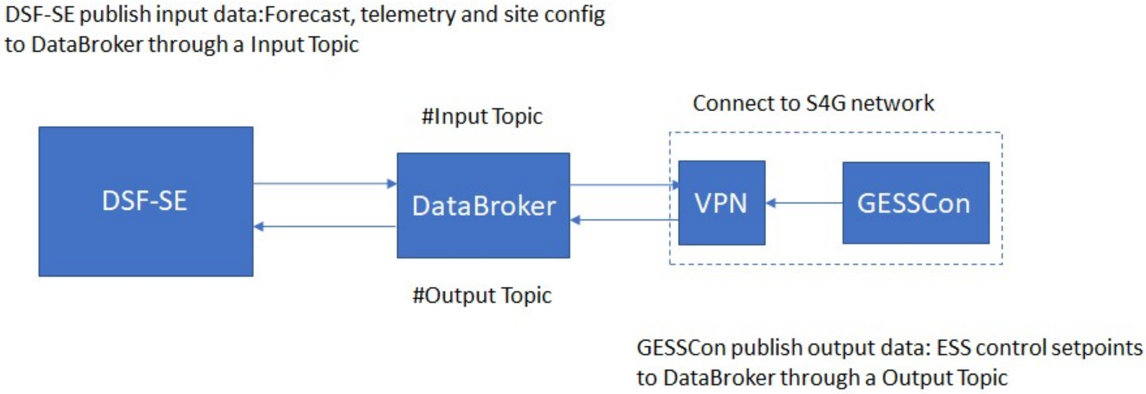

Figure 6. GESSCon interfaces to DSF-SE

As shown in Figure 6, GESSCon receives input data from DSF-SE and return ESS control setpoint back to DSF-SE in form of MQTT messages through the Data Broker.

3 Input data

There are three types of input data for GESSCon. One is real-time data of ESS status, which is published by SMX in the Data Broker. Another one is the site configuration data, which contains data describing the site physical limitation such as battery capacity, battery power and grid capacity. And the last one is forecasted data produced by DSF-Connectors. In this section, three types of data (Forecast data, telemetry data and site configuration data) are introduced.

3.1 Forecast data

Forecast data are provided by DSF and they are developed within T5.4 in Storage4Grid.

3.1.1 Demand/Load forecast

Demand or loads forecasts are provided by a DSF-connector for a specific test site. In Bolzano residential site, as GESSCon is a grid-side control system, the demand input to GESSCon is not a single user demand. In fact, it is an aggregated load forecast at the feeder line of the test site. In Bolzano commercial site, since there is no other consumption than EV charging station, the demand input to GESSCon is an aggregated EV load forecast. In the simulation case, DSF-SE simulates the topology in Skive/Fur residential grid, therefore the demand input is load forecast of Skive/Fur residences.

3.1.2 PV production forecast

PV production forecast in Bolzano is provided by a DSF-Connector in Storage4Grid. The returned data are targeting a specific area which is Bolzano city in this case. The raw data has 1-minute resolution with a unit of Watt/m^2. It is further processed by GESSCon's input handler to fit it with the correct PV panel size (provided by PV owner) on the test-site and convert it into readable data format for GESSCon core program.
In the simulation use case, PV production forecast is provided by DSF-SE, the data has 24 hours horizon with 1-hour resolution.

3.1.3 Electricity price forecast

Electricity price forecast in Bolzano is supported by another DSF-Connector. It is spot market price obtained through a third-party service (transparency.entsoe.eu). [iv]
In the simulation case, electricity price forecast of Skive/Fur site is provided by DSF-SE.

3.1.4 Tariff

Feed-in tariff is assumed to be zero in all use cases in Storage4Grid project, which is for the purpose of maximizing the utilization of the renewable energy.

3.2 Real-time Battery telemetry

Battery telemetry inputs are battery related real-time parameters measured by BMS (Battery Management System). In Storage4Grid project, SMX uses a smart meter to accomplish on-site real-time measurements of the ESS. It publishes this data into specified MQTT-topics, in order to be used by other components in the system through the Data Broker. Currently, energy storage data for Bolzano residential and commercial site are not available in topics of the Data Broker. For testing purposes, GESSCon is operating with mocked SoC (0%) for now and it will switch to real telemetry data once the implementation is completed.

3.3 Site configuration

Site configuration inputs are site related physical limitations of the local grid and the ESS as shown in Table 2. These values are site specific and currently they are stored in GESSCon. Once there is a physical change on the test-site (such as battery replacement, change of grid limitation) the corresponding site configuration data in GESSCon should be updated as well.

Site name	EDYNA-0018 (Residential)	EDYNA-0019 (Commercial)

Contractual limit in/Grid capacity in (kW)	11/86.6	90/124.7
Contractual limit out/Grid capacity out (kW)	16/86.6	150/124.7
System efficiency (Round trip efficiency) (%)	96%	96.1%
Battery max operational capacity (kWh)	12(Usable 9.6)	80(Usable 75)
Battery min operational capacity (kWh)	12*0.07(Usable 0)	0
Battery/inverter max charge power (kW)	6.4	33
Battery/inverter max discharge power (kW)	6.4	33
PV panel size (m^2)	76.8	1151.17

Table 2. Site configuration input from Bolzano sites for GESSCon

4 Output data

GESSCon will send a 24h ESS control setpoints to PROFEV/PROFESS once per day, in order to reach GESSCon's objective which can maximize renewable self-consumption and minimize the energy payment (shown in section 5). These setpoints will be further processed by local control unit PROFEV/PROFESS to control the objective ESS. The process method is documented in D4.7-Final Cooperative EV charging station control algorithms.

Figure 7, Figure 8, Figure 9 and Figure 10 are plots made by real input data of 18th June 2019.

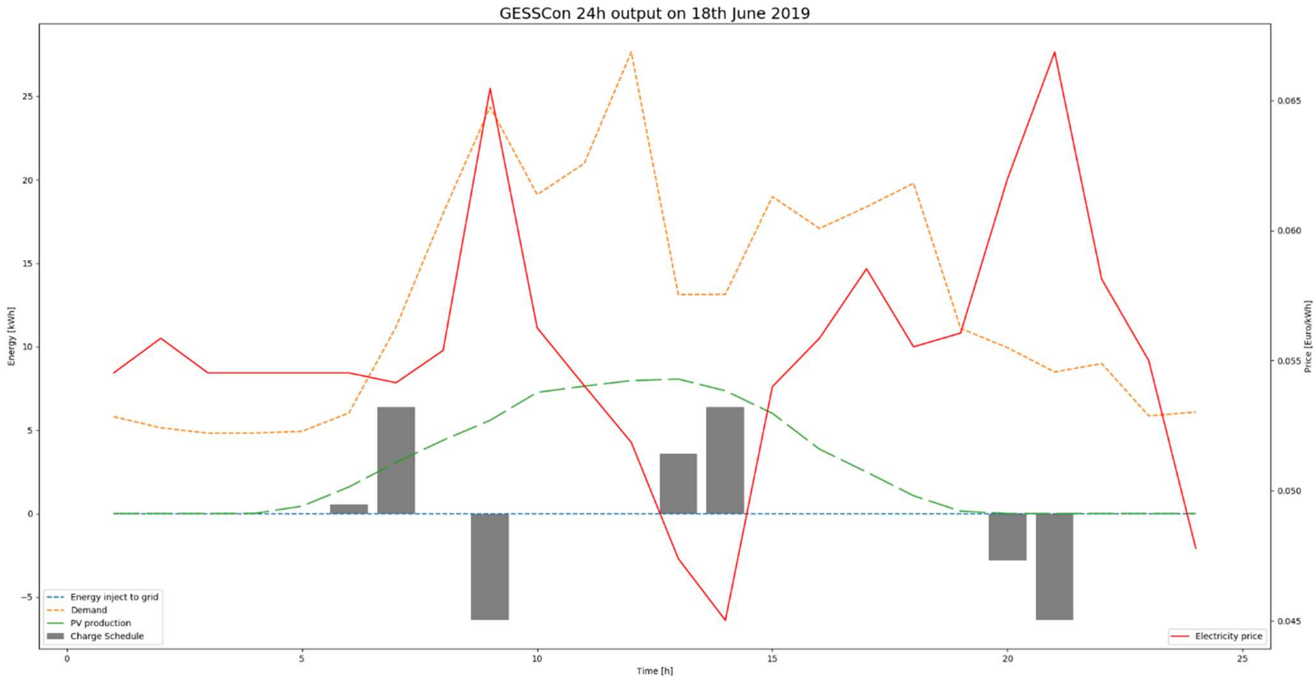

Figure 7. GESSCon output for Bolzano residential site on 18th June 2019

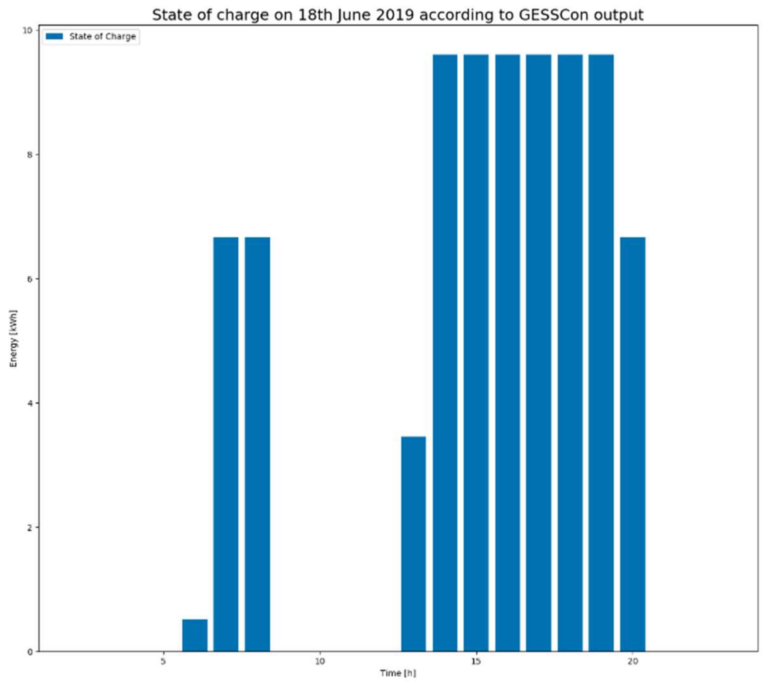

Figure 8. ESS state of charge according to GESSCon output for Bolzano residential site on 18th June 2019

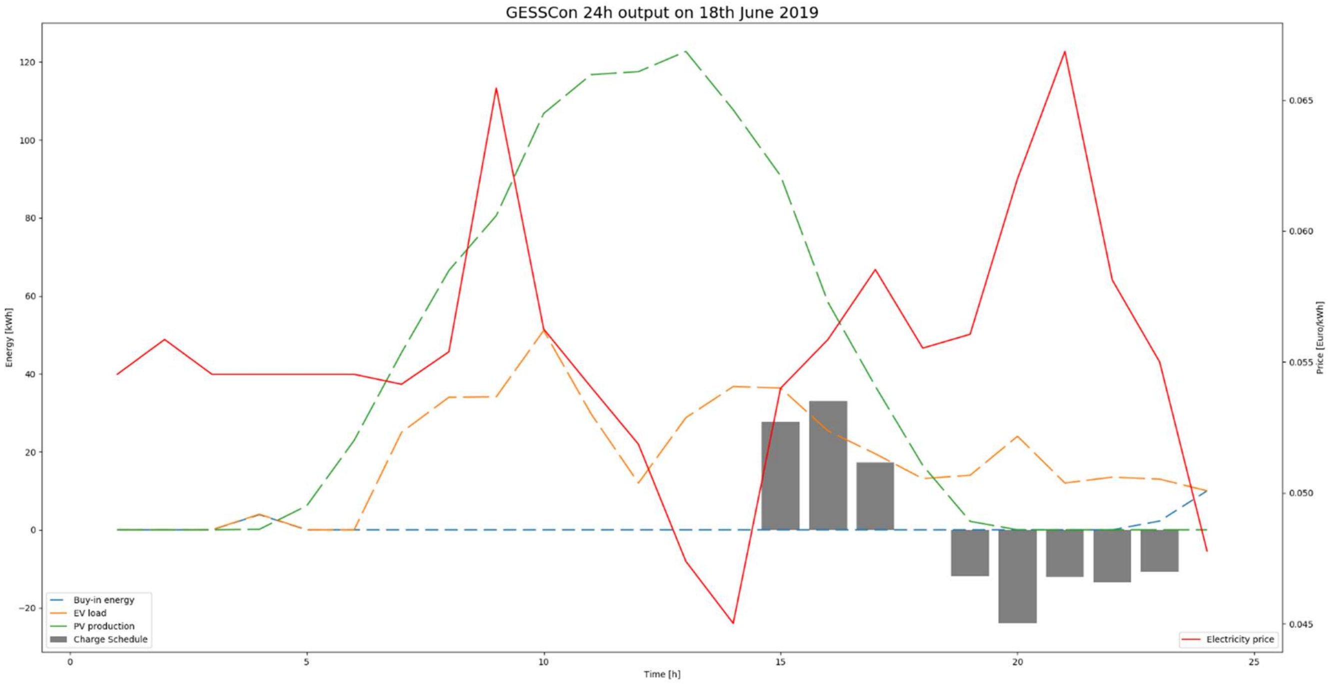

Figure 9. GESSCon output for Bolzano commercial site on 18th June 2019

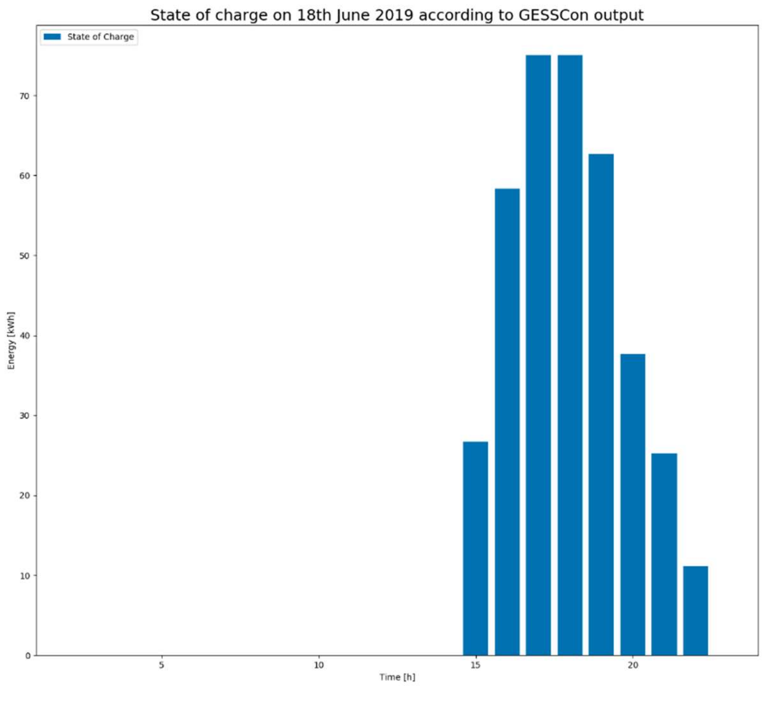

Figure 10. ESS state of charge according to GESSCon output for Bolzano commercial site on 18th June 2019

As illustrated in Figure 7 and Figure 9, the control setpoint from GESSCon are the grey bars called "Charge Schedule". Charge Schedule has a resolution of 60 minutes, so there are 24 setpoints output by GESSCon in each iteration. Besides energy storage SoC is still mocked data (as described in section 3.2), all the other inputs including site configuration and forecast data are specific for the test site in real time.

In Figure 7 the demand (aggregated load) is much higher than the PV production, therefore the exchange power (demand – PV production) are negative most of the time during the day. It means that there will be power taken from the grid (buy-in) most of the time. In this case the electricity price (red line) has more impact to the battery charge/discharge behavior (Charge Schedule). ESS will charge when the price is low and discharge when the price is high. ESS state of charge can be seen in Figure 8.

In Figure 9 the PV production is higher than the demand (EV load), therefore the exchange power (demand – PV production) are positive most of the time during the day. It means that there will be excess PV power injected to the grid most of the time. In this case the EV load (dashed green line) has more influence on the Charge Schedule. ESS will reach fully charged during high PV production hours and supply the hours when PV production is lower than the EV load. ESS state of charge is shown in Figure 10.

5 GESSCon Core

The objective of the GESSCon Core optimization algorithm is to minimize the overall cost of energy consumed. This is done by operating the ESS (charging and discharging the energy storage) in the most optimal way, based on the available information introduced in section 3. The optimization is based on forecasted information which present uncertainties. Therefore, the operation of the ESS may turn out not to be optimal under the actual conditions (actual consumption and production).

The GESSCon calculates a schedule for the local PROFESS/PROFEV (one schedule for each PROFESS/PROFEV). The actual control of the ESS is performed by the PROFESS/PROFEV and specified in "D4.3-Final User-side ESS control system" and "D4.7-Final Cooperative EV charging station control algorithms".

The GESSCon algorithm is formulated as a mixed integer linear programming (MILP) problem and consists of an objective function and a set of constraints.

The objective function contains 3 sections which can reach the final goal of T4.2:

- maximizing renewable self-consumption: Direct influenced by objective function
- relieve power congestion in the grid: Indirect influenced by objective function
- charging ESS when electricity market price is low: Direct influenced by objective function
- prolonging battery lifetime: Direct influenced by objective function
- remain grid stability by preventing systems discharging to the neighboring systems which charge is high: Direct influenced by objective function

$$\min \sum_{i=\mathrm{i}_0}^{I} \sum_{t=\mathrm{t}_0}^{T} \left(\lambda_{t,i}^{b} \cdot P_{t,i}^{B} - \lambda_{t,i}^{s} \cdot P_{t,i}^{S} + (\mathrm{P}_{t,i}^{d} + \mathrm{P}_{t,i}^{c}) \cdot \lambda_{t,i}^{o}\right)$$

where I is the number of test sites, T is the time. $P_{t,i}^{S}$ and $P_{t,i}^{B}$ are the selling (inject to the grid) and buying (taking from the grid) powers. $P_{t,i}^{d}$ and $P_{t,i}^{c}$ are discharge and charge powers for the battery. $\lambda_{t,i}^{s}$ and $\lambda_{t,i}^{b}$ are the buying and selling power prices. $\lambda_{t,i}^{o}$is battery operational cost per kWh.

The constraints are formed by different technical and economical limitations of the energy storage system, which includes energy balancing, energy storage operation constraints, grid-side constraints and feeder line topology related constraints.

5.1 Bolzano residential case and Skive simulation case

In Bolzano residential site and Skive simulation case HLUC-3-PUC-4, GESSCon act as an aggregator, it ensures energy storage system doesn't charge from its neighbouring system. In this case, GESSCon receives aggregated input such as aggregated load, aggregated PV production and it aims at optimizing the total electricity buy-in from the grid for the whole feeder line. Besides, each individual ESS specification are also known by GESSCon. This information is critical for GESSCon when disaggregating its output setpoints to each individual ESS. The local control unit for each individual ESS namely PROFESS/PROFEV receives individual setpoints profile from GESSCon and further process it [D4.7].

5.2 EV included optimization

In the original system architecture "D3.1 - Initial S4G Components, Interfaces and Architecture Specification", a parallel module to GESSCon - GEVChCon is in charge of all use cases that has EV involved in this project. However, during the components development, it was decided to integrate the EV aggregated profile into the GESSCon without using a separate component. Therefore, GESSCon will also include EV cases in its optimization model. Aggregated EV load prediction is an additional input to GESSCon for EV involved use cases. This prediction data is provided by a DSF-EVA connector, similar to other forecast data.

6 Site Controller for Skive grid-side storage

Site Controller is a local control unit which is placed close to the grid side energy storage delivered by LiBal. It has a CAN connection to the battery management system (BMS), a MODBUS TCP connection to the inverter and a TCP/IP connection to the GESSCon. This component is not developed within S4G project; however, it is needed for Skive grid-side storage use case "HLUC3-PUC-3 – Voltage and flux control at grid side storage" [D2.2]. Therefore, a control algorithm for this use case has been developed and implemented in the Site Controller.

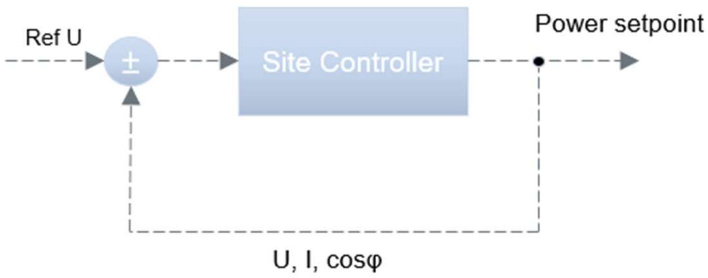

Figure 11. Skive grid-side storage control algorithm

As illustrated in Figure 11 the control algorithm takes real time measurement such as voltage (U), current (I), power factor (cosφ) from the point of common coupling of the site and computes control setpoints of active and reactive power to the inverter to control the ESS input/output in order to stabilize the feeder line voltage. The target voltage (Ref U) at common coupling point is 230V ±10% according to the requirement from DSO.

7 Implementation

The GESSCon mathematical algorithm is implemented in Python as a mix integer linear programming (MILP) problem using the Pyomo[v] package with GLPK[vi] as solver. The model contains continuous real variables, integer variables, and binary variables. A number of input parameters such as electricity price, demand, PV production, and feed-in tariff are used to determine the energy storage behaviors and return the optimal ESS charging/discharging schedule for the next day. The GESSCon solutions resides in a docker image based on python:3.6-jessie (linux/amd). In this project an Intel x86 processor has been chosen for the hardware implementation, with an Ubuntu Linux OS and Docker run time installed. In operational mode, the GESSCon collects needed data via REST calls from the DSF connectors and utilizes MQTT protocol to retrieve needed input data from the Data Broker such as ESS SoC value. Based on the inputs the GESSCon Core computes the

ESS control setpoints and publishes them as MQTT messages through the Control Broker under the S4G VPN network.

The control algorithm of Site Controller is implemented in Python. Based on current Site Controller environment no additional package is needed. This control algorithm will be presented as an operational mode that the user can choose to run inside the Site Controller. By external command, the Site Controller will enter the voltage support mode, reading inputs from a meter (external or internal to the Site Controller) through TCP/IP and sending control setpoints to the inverter through MODBUS TCP.

8 Installation/Deployment

The GESSCon uses a container architecture which simplifies its deployment. Images of the software are uploaded into an Azure Container registry. The image is downloaded by an Azure VM (Virtual Machine) which runs the docker image containing the GESSCon Core algorithm.

9 Software dependencies and requirements

As described in "D4.4 - Grid-side ESS control system", no special hardware requirements are needed to implement the GESSCon. Software dependencies can be found in Table 3.

Dependency	License	Role
Docker	Apache License 2.0	Docker is used to facilitate the build process of the container image.
Python, image based on "Jessie"	GNU General Public License (GPL) 2.0	The operating system of the optimization model (GESSCon Core algorithm) used as Core controller operating system.
Microsoft Azure	Pay-as-you-go	Connecting with OpenVPN to S4G and using Azure VM

Table 3. Software dependencies

10 API Reference

10.1 REST API

GESSCon doesn't present an API but it uses the DSF-as described in section 2.3 Table 1. GESSCon has adapted to the format of each specific endpoint.

10.2 MQTT/Broker

Table 4 and Table 5 are input and output data of GESSCon which should be applicable to all use cases as described in [D2.2]. GESSCon is able to handle several PROFESS instances at the same time, however it requires input data has been written into specific format for GESSCon to understand what these data are for. An example of DSF-SE simulation payload can be found in the Appendix B.

Variables (with units)	Data Type	Input Type	Description	Source	Periodicity	Note

Demand [kWh]	Float	Forecast	Forecast data from DSF connector	DSF	Once per 24 hours send a forecast array with 24 data points.	For the residential case where each house has individual PV and home battery, data should be an aggregated value.
EV load [kWh]	Float					
PV production [kWh]	Float					
Electricity Price [Local currency]	Float					
Feed-in Tariff [Local currency]	Float					
SoC	Float	Telemetry Data	Real time data from ESS	SMX/Data Broker	Real time updated (within 10 sec)	
Grid Connection In [kW]	Float	Site Configuration Data	Site specific data related to ESS and grid	Site/DSO	Static data	Line capacity and contractual capacity
Grid Connection Out [kW]	Float					
Number of ESS	Int					Number of ESS on the feeder line
ESS Efficiency	Float					Mainly inverter round trip efficiency
ESS Max Operation Limit [kWh]	Float					Recommended operating ESS between 20% to 80% of its capacity
ESS Min Operation Limit [kWh]	Float					
ESS Max Charge Limit [kW]	Float					
ESS Max Discharge Limit [kW]	Float					
PV panel size [m^2]	Float					For calculating total PV production
PV coefficient	Float					For calculating total PV production

Table 4. Input data for GESSCon for use cases and simulation

Output is an array of (0..23) setpoints for each ESS (1..N) on the feeder line.
POST / ESS_control_setpoints

Variables (with units)	Data Type	Output Type	Description	Directions	Periodicity	Note

Name	String	Chargeschedule	Description of the data	GESSCon → Control Broker	Once per 24 hours send a control array with 24 data points.	"ESS_Control"
Time	Float		Epoch time stamp			Convert to epoch time: ref1, ref2, ref3 Example: 1542208620.36428 -> 14, 2018 3:17:00.364 PM UTC
Charge/Discharge Power [kW]	Float		Charge/Discharge Control of ESS			SENML format MQTT message contains Time, Active Power and unit. Charge power shows in positive number, discharge power shows in negative number. Example of charge schedule is described in section 2.5.

Table 5. Output data for GESSCon for use cases and simulation

11 Conclusions

This document "D4.5 – Final grid-side ESS control system" describes the final structure of GESSCon, Site Controller, and the relation between GESSCon, PROFEV/PROFESS, Brokers and DSF Components. GESSCon contains a core program and several connectors to other components in the S4G system, where the core program contains the optimization algorithm and the connectors are taking care of input/output data handling. A one-day result has been shown in section 4, which can reflect the purpose of GESSCon objectives. Site Controller is hosting the ESS control algorithm for use cases HLUC3-PUC-4 in S4G. The control algorithm is described in section 6 and implemented. A final evaluation "D6.12– Phase 3 Evaluation Report" will be presented in M39 which evaluates the quality of the GESSCon and the Site Controller control algorithm performance in Bolzano sites, Skive site and in the integration with DSF-SE simulation case. The current developed modules will be updated for further integration purpose within Storage4Grid project.

Acronyms

Acronym	Explanation
API	Application Program Interface
BMS	Battery Management System
DSF	Decision Support Framework
DSO	Distribution system operator
SE	Simulation Engine
ESS	Energy Storage System
EVA	Electric Vehicle Analyzer
GESSCon	Grid ESS Controller
GEVChCon	Grid Electric Vehicle Charging Controller
MQTT	Message Queuing Telemetry Transport
PROFESS	Professional Realtime Optimization Framework for Energy Storage Systems
PROFEV	Professional Realtime Optimization Framework for Electric Vehicle
RES	Renewable Energy Sources
REST	Representational State Transfer
S4G	Storage4Grid
SMX	Smart Meter eXtension
SoC	State of Charge
USM	Unbundled Smart Meter
VM	Virtual Machine

List of figures

List of tables

Reference

[i] OGC, Open Geospatial Consortium, http://www.opengeospatial.org/, accessed 26 February 2019.
[ii] OGC, OGC SensorThings APIhttp://www.opengeospatial.org/standards/sensorthings, accessed 26 February 2019.
[iii] SenML, Sensor Measurement Lists, https://tools.ietf.org/html/rfc8428, accessed 26 February 2019.
[iv] ENTSO-E, Transparency Platform restful API – user guide – A.10.Areas, https://transparency.entsoe.eu/content/static_content/Static%20content/web%20api/Guide.html#_areas, accessed 26 February 2019.
[v] Pyomo, Pyomo online documentation, https://pyomo.readthedocs.io/en/stable/, accessed 02 July 2019
[vi] GLPK, GNU Linear Programming Kit, https://www.gnu.org/software/glpk/, accessed 02 July 2019.

Appendix A *Epoch time*

Epoch time is the number of seconds since 1970 1. January UTC see https://en.wikipedia.org/wiki/Unix_time.

Using a converter such as https://www.epochconverter.com/, it is possible to convert, e.g., 1538063205942349056 = Thursday, September 27, 2018 3:46:45.942 PM.

Appendix B *DSF-SE#GESSCon payload example*

```
{
        "site": "EDYNA-0018",
        "time_start": "2018.10.01 00:00:00",
   "time_stop": "2018.10.01 23:00:00",
        "raw": {
                        "elprices": [{
                                "DateTime": "2018.10.01 00:00:00",
                                "elprice": 1.909825
                        },
                        {
                                "DateTime": "2018.10.01 01:00:00",
                                "elprice": 1.83985
                        },
```

{
    "DateTime": "2018.10.01 02:00:00",
    "elprice": 1.8422625
},
{
    "DateTime": "2018.10.01 03:00:00",
    "elprice": 1.7302375
},
{
    "DateTime": "2018.10.01 04:00:00",
    "elprice": 1.7302375
},
{
    "DateTime": "2018.10.01 05:00:00",
    "elprice": 1.897075
},
{
    "DateTime": "2018.10.01 06:00:00",
    "elprice": 2.02325
},
{
    "DateTime": "2018.10.01 07:00:00",
    "elprice": 2.286475
},
{
    "DateTime": "2018.10.01 08:00:00",
    "elprice": 2.286475
},
{
    "DateTime": "2018.10.01 09:00:00",
    "elprice": 2.2860125
},
{
    "DateTime": "2018.10.01 10:00:00",
    "elprice": 2.285825
},
{
    "DateTime": "2018.10.01 11:00:00",
    "elprice": 2.2866625
},
{
    "DateTime": "2018.10.01 12:00:00",
    "elprice": 2.285825
},
{
    "DateTime": "2018.10.01 13:00:00",
    "elprice": 2.2852625
},
{
    "DateTime": "2018.10.01 14:00:00",
    "elprice": 2.28415
},
{
    "DateTime": "2018.10.01 15:00:00",
    "elprice": 2.285175
},
{
    "DateTime": "2018.10.01 16:00:00",
    "elprice": 2.288525

```
},
{
        "DateTime": "2018.10.01 17:00:00",
        "elprice": 2.289825
},
{
        "DateTime": "2018.10.01 18:00:00",
        "elprice": 2.2875
},
{
        "DateTime": "2018.10.01 19:00:00",
        "elprice": 2.286575
},
{
        "DateTime": "2018.10.01 20:00:00",
        "elprice": 2.162075
},
{
        "DateTime": "2018.10.01 21:00:00",
        "elprice": 2.1362
},
{
        "DateTime": "2018.10.01 22:00:00",
        "elprice": 2.119275
},
{
        "DateTime": "2018.10.01 23:00:00",
        "elprice": 2.019425
}],
"demand": [{
        "DateTime": "2018.10.01 00:00:00",
        "Loads": 93.178825651650158
},
{
        "DateTime": "2018.10.01 01:00:00",
        "Loads": 84.278289960237004
},
{
        "DateTime": "2018.10.01 02:00:00",
        "Loads": 87.693150143429733
},
{
        "DateTime": "2018.10.01 03:00:00",
        "Loads": 84.113101260255831
},
{
        "DateTime": "2018.10.01 04:00:00",
        "Loads": 83.075137270556951
},
{
        "DateTime": "2018.10.01 05:00:00",
        "Loads": 82.475389730278067
},
{
        "DateTime": "2018.10.01 06:00:00",
        "Loads": 82.060492421070308
},
{
        "DateTime": "2018.10.01 07:00:00",
```

```
        "Loads": 85.073232467206253
    },
    {
        "DateTime": "2018.10.01 08:00:00",
        "Loads": 82.064489483182413
    },
    {
        "DateTime": "2018.10.01 09:00:00",
        "Loads": 84.786502672378987
    },
    {
        "DateTime": "2018.10.01 10:00:00",
        "Loads": 77.61181224509842
    },
    {
        "DateTime": "2018.10.01 11:00:00",
        "Loads": 83.064866107150792
    },
    {
        "DateTime": "2018.10.01 12:00:00",
        "Loads": 80.2605922598988
    },
    {
        "DateTime": "2018.10.01 13:00:00",
        "Loads": 81.720224649086973
    },
    {
        "DateTime": "2018.10.01 14:00:00",
        "Loads": 79.998597011714779
    },
    {
        "DateTime": "2018.10.01 15:00:00",
        "Loads": 81.163861756649681
    },
    {
        "DateTime": "2018.10.01 16:00:00",
        "Loads": 80.232652330399631
    },
    {
        "DateTime": "2018.10.01 17:00:00",
        "Loads": 84.529965768146127
    },
    {
        "DateTime": "2018.10.01 18:00:00",
        "Loads": 81.89492669255614
    },
    {
        "DateTime": "2018.10.01 19:00:00",
        "Loads": 84.42044783973769
    },
    {
        "DateTime": "2018.10.01 20:00:00",
        "Loads": 80.007100943038324
    },
    {
        "DateTime": "2018.10.01 21:00:00",
        "Loads": 85.302231514499354
    },
    {
```

```
		"DateTime": "2018.10.01 22:00:00",
		"Loads": 90.636472254697651
	},
	{
		"DateTime": "2018.10.01 23:00:00",
		"Loads": 95.130269168580256
	}],
	"pv": [{
		"DateTime": "2018.10.01 00:00:00",
		"pv": 0.0
	},
	{
		"DateTime": "2018.10.01 01:00:00",
		"pv": 0.0
	},
	{
		"DateTime": "2018.10.01 02:00:00",
		"pv": 0.0
	},
	{
		"DateTime": "2018.10.01 03:00:00",
		"pv": 0.0
	},
	{
		"DateTime": "2018.10.01 04:00:00",
		"pv": 0.0
	},
	{
		"DateTime": "2018.10.01 05:00:00",
		"pv": 0.0
	},
	{
		"DateTime": "2018.10.01 06:00:00",
		"pv": 0.0
	},
	{
		"DateTime": "2018.10.01 07:00:00",
		"pv": 0.0
	},
	{
		"DateTime": "2018.10.01 08:00:00",
		"pv": 0.0
	},
	{
		"DateTime": "2018.10.01 09:00:00",
		"pv": 552.5472511706563
	},
	{
		"DateTime": "2018.10.01 10:00:00",
		"pv": 556.1998308518424
	},
	{
		"DateTime": "2018.10.01 11:00:00",
		"pv": 556.1998308518424
	},
	{
		"DateTime": "2018.10.01 12:00:00",
		"pv": 556.1998308518424
	},
```

```
{
        "DateTime": "2018.10.01 13:00:00",
        "pv": 556.1998308518424
},
{
        "DateTime": "2018.10.01 14:00:00",
        "pv": 556.1998308518424
},
{
        "DateTime": "2018.10.01 15:00:00",
        "pv": 556.1998308518424
},
{
        "DateTime": "2018.10.01 16:00:00",
        "pv": 556.1998308518424
},
{
        "DateTime": "2018.10.01 17:00:00",
        "pv": 556.1998308518424
},
{
        "DateTime": "2018.10.01 18:00:00",
        "pv": 556.1998308518424
},
{
        "DateTime": "2018.10.01 19:00:00",
        "pv": 556.1998308518424
},
{
        "DateTime": "2018.10.01 20:00:00",
        "pv": 556.1998308518424
},
{
        "DateTime": "2018.10.01 21:00:00",
        "pv": 556.1998308518424
},
{
        "DateTime": "2018.10.01 22:00:00",
        "pv": 556.1998308518424
},
{
        "DateTime": "2018.10.01 23:00:00",
        "pv": 556.1998308518424
}],
"ev": [{
        "DateTime": "2018.10.01 00:00:00",
        "ev": 0.0
},
{
        "DateTime": "2018.10.01 01:00:00",
        "ev": 0.0
},
{
        "DateTime": "2018.10.01 02:00:00",
        "ev": 0.0
},
{
        "DateTime": "2018.10.01 03:00:00",
        "ev": 0.0
```

},
{
"DateTime": "2018.10.01 04:00:00",
"ev": 0.0
},
{
"DateTime": "2018.10.01 05:00:00",
"ev": 0.0
},
{
"DateTime": "2018.10.01 06:00:00",
"ev": 0.0
},
{
"DateTime": "2018.10.01 07:00:00",
"ev": 0.0
},
{
"DateTime": "2018.10.01 08:00:00",
"ev": 0.0
},
{
"DateTime": "2018.10.01 09:00:00",
"ev": 0.0
},
{
"DateTime": "2018.10.01 10:00:00",
"ev": 0.0
},
{
"DateTime": "2018.10.01 11:00:00",
"ev": 0.0
},
{
"DateTime": "2018.10.01 12:00:00",
"ev": 0.0
},
{
"DateTime": "2018.10.01 13:00:00",
"ev": 0.0
},
{
"DateTime": "2018.10.01 14:00:00",
"ev": 0.0
},
{
"DateTime": "2018.10.01 15:00:00",
"ev": 0.0
},
{
"DateTime": "2018.10.01 16:00:00",
"ev": 0.0
},
{
"DateTime": "2018.10.01 17:00:00",
"ev": 0.0
},
{
"DateTime": "2018.10.01 18:00:00",

```
                        "ev": 0.0
                },
                {
                        "DateTime": "2018.10.01 19:00:00",
                        "ev": 0.0
                },
                {
                        "DateTime": "2018.10.01 20:00:00",
                        "ev": 0.0
                },
                {
                        "DateTime": "2018.10.01 21:00:00",
                        "ev": 0.0
                },
                {
                        "DateTime": "2018.10.01 22:00:00",
                        "ev": 0.0
                },
                {
                        "DateTime": "2018.10.01 23:00:00",
                        "ev": 0.0
                }]
        },
    "tele": {
        "SOC" : [0.1, 0.2, 0.3]
        },
    "config": {
        "ESS_number": 3,
        "tariff" : 0,
        "grid_i" : 1000,
        "grid_o" : 1000,
        "ess_eff" : [0.96, 0.96, 0.96],
        "bmax" : [60, 60, 60],
        "bmin" : [0,0,0],
        "pcmax" : [30,30,30],
        "pdmax" : [30,30,30],
        "cycle" : [0,0,0],
        "loss" : 0
                }
}
```