reschool

Creation, growing and management of energy communities

Community Energy Management System

Deliverable nº: D3.2

Deliverable name: D3.2 – Community Energy Management System

Version: v1 **Release date:** 18/06/2024 **Dissemination level:** PU

Author(s):

Pierre Kil (pierre@openremote.io). OpenRemote
Antoni Company (acompany@bambooenergy.tech). Bamboo
Adrian Brasero (abrasero@bambooenergy.tech). Bamboo
Yelena Vardanyan (yelena.vardanyan@ri.se). RISE
Nasos Vasilakis (avasilakis@coen.coop). COEN
Anton Belinskiy (a.belinskiy@uu.nl). Utrecht University
Annie Albage (annie.albage@electricityinnovation.se). ElectriCITY Innovation
Josefin Danielsson (josefin.danielsson@electricityinnovation.se). ElectriCITY Innovation

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101096490

Disclaimer

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101096490

Document history

V	Date	Beneficiary	Author
V0.1	05/04/2024	OpenRemote	Pierre Kil
V0.2	10/05/2024	OpenRemote	Pierre Kil
V0.3	11/6/2024	RISE, UU, BBEN, UdG, OpenRemote	Yelena Vardanyan, Anton Belinsky, Adrian Brasero, Sergio Herraiz Jaramillo, Pierre Kil
V1	18/6/2024	OpenRemote	Pierre Kil

Peer reviewed by

Partner	Reviewer
University of Girona (UdG)	Joaquim Melendez Frigola
University of Girona (UdG)	Sergio Herraiz Jaramillo
Bamboo (BBEN)	Adrian Brasero
Research Institute of Sweden (RISE)	Yelena Vardanyan
Bamboo (BBEN)	Antony Companion

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101096490

Executive Summary

This report describes the functionality of the Community Energy Management System (CEMS), and it's implementation as developed on top of OpenRemote. It starts with the data models and services as described in the Reschool platform architecture and data models (D3.1).

Furthermore, it specifies the data models, forecasting- and optimisation services and their usage within the different pilots, before detailing the actual implementation within the CEMS. In addition, the integration methods with both services as well as devices, and the visualisation tools are described.

For scalability and dissemination purposes we have added the references to the three locations which are implementing the open source CEMS based on OpenRemote, and added full documentation for users, as well as developers.

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101096490

Table of contents

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101096490

1 Introduction

The open source Community Energy Management System (CEMS) is intended to monitor and optimise a network of energy objects towards maximising self-consumption and/or minimising costs. It builds on the architecture as defined in D3.1. The CEMS consists of assets of different types as depicted in figure 1. The CEMS is described and documented in this task.

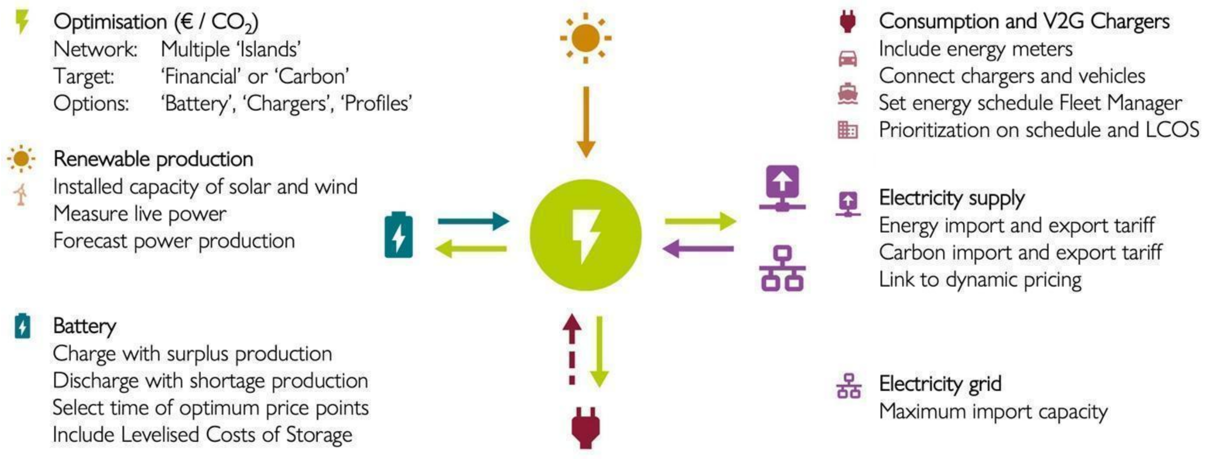

Optimisation (€ / CO₂)
Network: Multiple 'Islands'
Target: 'Financial' or 'Carbon'
Options: 'Battery', 'Chargers', 'Profiles'
Renewable production
Installed capacity of solar and wind
Measure live power
Forecast power production
Battery
Charge with surplus production
Discharge with shortage production
Select time of optimum price points
Include Levelised Costs of Storage
Consumption and V2G Chargers
Include energy meters
Connect chargers and vehicles
Set energy schedule Fleet Manager
Prioritization on schedule and LCOS
Electricity supply
Energy import and export tariff
Carbon import and export tariff
Link to dynamic pricing
Electricity grid
Maximum import capacity

Figure 1 the Community EMS connects the energy objects within a community with the intention to monitor and optimise them towards highest self-consumption or minimum costs

1.1 Objectives

Building on the architecture, data models and services outlined in D3.1, this document provides the description of the developed open source CEMS for energy communities which addresses the requirements at individual and community level, including the required forecasting and optimisation services. It also facilitates integration with flexibility services as developed in task 3.3.

To validate the reach of functionality, the open source CEMS will be fully deployed and tested in the Amsterdam pilot and partially in the Girona and Athens pilots. We acknowledge that technologies used for the actual pilot implementations will vary per pilot. Therefore, the broader validation will be partially based on functional mapping per pilot as well as tests of specific features.

For dissemination, replicability and support towards professional users within and beyond the scope of Reschool, we provide the codebase, extensive documentation, as well as a forum.

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101096490

1.2 Contribution of partners

This document has been elaborated with the collaboration of all the RESCHOOL partners under the coordination of the project coordinator. The following table shows the responsibilities and roles taken by the partners in the delivery of this task.

Partner	Contribution
OpenRemote	Coordination and elaboration of the Community Energy Management System (CEMS), functionalities, implementation within pilots, and documentation.
Bamboo	Feedback on the Community Energy Management System (CEMS), functionalities, implementation within pilots, and documentation and implementation in the Spanish pilot.
RISE	Feedback on the Community Energy Management System (CEMS), functionalities, implementation within pilots, and documentation.
UdG	Feedback on the Community Energy Management System (CEMS), functionalities, implementation within pilots, and documentation and implementation in the Spanish pilot.
ElectriCITY	Feedback on the Community Energy Management System (CEMS), functionalities, implementation within pilots, and documentation and implementation in the Stockholm pilot.
COEN	Feedback on the Community Energy Management System (CEMS), functionalities, implementation within pilots, and documentation and implementation in the Athens pilot.

1.3 Report Structure

Chapter 2 starts with an overview of the CEMS elements applied in each pilot. In case of (partial) implementation via other software tools (Bamboo, Brikks) than the developed open source CEMS, this is also indicated.

The different asset types, forecasting services, and optimisation routine, are described in chapter 3, together with integration options and visualisation functionality.

Finally, all relevant documentation to support dissemination towards other interested users, within and beyond Reschool is introduced in chapter 4.

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101096490

2 Pilot requirements

The open source Community Energy Management System (CEMS) is built as an extension to the OpenRemote open source IoT device management platform. Within OpenRemote the device models are represented as asset types. For the CEMS functionality asset types are defined and added to OpenRemote, as representations of the energy data models as defined as part of D3.1: section 5.

The CEMS is applied in three of the four pilots. The table indicates which energy data models are represented by which asset types (for definition and reference see 3.1). Moreover, it's indicated which models are relevant for which pilot and whether the CEMS is used for implementation or any of the alternatively used technical frameworks.

Table 1 the mapping of data models on asset types in the CEMS as well as whether they are applied in the pilots. If they are applied, it's indicated whether it's through the CEMS or another technical framework.

Data model	CEMS asset type	Amsterdam	Girona	Stockholm	Athens
PV System	electricityProducerSolar	CEMS	Bamboo/CEMS	Brikks	CEMS
	electricityProducerWind	-	-	-	-
Battery	electricityBattery	-	Bamboo/CEMS	Brikks	-
Space Heater	*t.b.d.*	-	-	Brikks	-
EV	electricityVehicle electricityVehicleFleet	CEMS	-	-	-
EV Charger	electricityCharger	CEMS	Bamboo/CEMS	Brikks	-
Energy Prosumer	*Tree structure*	CEMS	Bamboo/CEMS	Brikks	-
Energy Consumption	electricityConsumer	CEMS	Bamboo/CEMS	Brikks	CEMS
Energy Production	electricitySupplier	CEMS	Bamboo/CEMS	Brikks	CEMS
Weather forecast	weather	CEMS	Bamboo/CEMS	-	CEMS
Forecasting					
Weather	Weather forecast	CEMS	Bamboo	-	CEMS
PV Power Production	Solar forecast	CEMS	Bamboo	Brikks	CEMS
Wind Power Production	Wind forecast	-	-	-	-
Energy Demand	Weighted averaging	CEMS	Bamboo	Brikks	CEMS
Optimisation					
Flexibility management	Tree structure	CEMS	Bamboo	-	CEMS
EV	Optimisation	CEMS	-	-	-
Battery	Optimisation	CEMS	Bamboo	Brikks	-
Space Heater	*to be added*	-	-	-	-
EV Charger	Optimisation	CEMS	-	-	-
Trading					
Dynamic pricing		CEMS	-		-

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101096490

3 Community Energy Management System

3.1 Data models and asset types

Within the CEMS the data models are represented as asset types. An asset type is represented by an asset type name and attributes with value types. These asset types are identical to the data models as presented in task 3.1 (see table 1 for the mapping). The attributes represent the real time data (e.g. powerSetpoint for a battery) but also additional metadata to describe the asset characteristics (e.g. powerMaxSetpoint to indicate the maximum allowed charging power, or location).

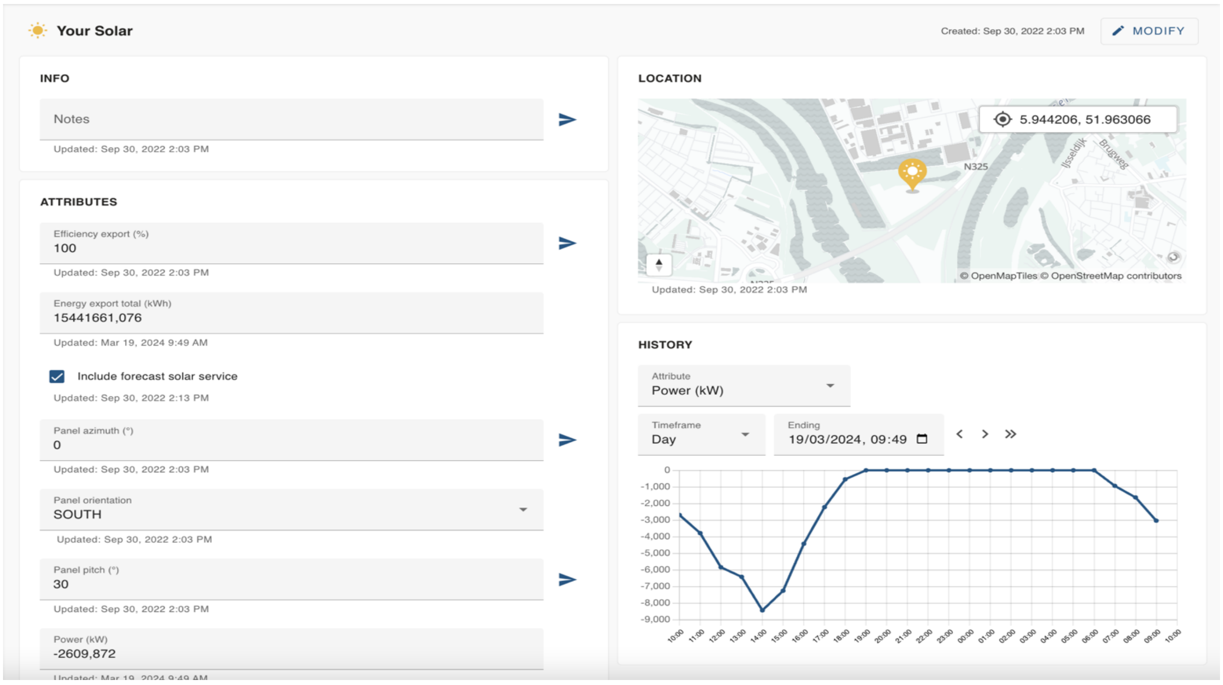
The relevance of using asset types is securing that any logic or services will always be able to read and control all assets of these types (e.g. a 'power' attribute always has the same unit). Asset types can be adjusted by adding additional optional attributes (still recognised by services) or ad-hoc attributes. Latter ones can be added by users, e.g. to include extra information, for example to capture an additional Key Performance Indicator (KPI). For more information on how assets and attributes are used and can be configured, we refer to the documentation (OpenRemote, User Guide: Assets, Agents and Attributes).

For a complete description of the asset types specifically used for the CEMS, we refer to the documentation (OpenRemote, User Guide: Energy Management, Electricity Assets and required Agents). Asset types and code are part of the OpenRemote GitHub repository (OpenRemote, GitHub code reference for energy asset models).

3.2 Energy forecasting services

The CEMS includes three forecasting services, for solar power, wind power and generic time series forecasting for demand.

The **solar power forecast** relies on the forecast service offered by Forecast Solar (Forecast Solar) and can directly be coupled to the electricityProducerSolar asset. Once all attributes, including location, are filled in correctly, the solar power is forecasted as shown in figure 2. For configuration of the solar power forecast service OpenRemote offers an online user guide (OpenRemote, User Guide: Energy Management, configuration of solar production forecast).



Your Solar

Created: Sep 30, 2022 2:03 PM

INFO

Notes
Updated: Sep 30, 2022 2:03 PM

ATTRIBUTES

Efficiency export (%): 100
Updated: Sep 30, 2022 2:03 PM

Energy export total (kWh): 15441661,076
Updated: Mar 19, 2024 9:49 AM

☑ Include forecast solar service
Updated: Sep 30, 2022 2:13 PM

Panel azimuth (°): 0
Updated: Sep 30, 2022 2:03 PM

Panel orientation: SOUTH
Updated: Sep 30, 2022 2:03 PM

Panel pitch (°): 30
Updated: Sep 30, 2022 2:03 PM

Power (kW): -2609,872

LOCATION

5.944206, 51.963066
Updated: Sep 30, 2022 2:03 PM

HISTORY

Attribute: Power (kW)
Timeframe: Day
Ending: 19/03/2024, 09:49

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101096490

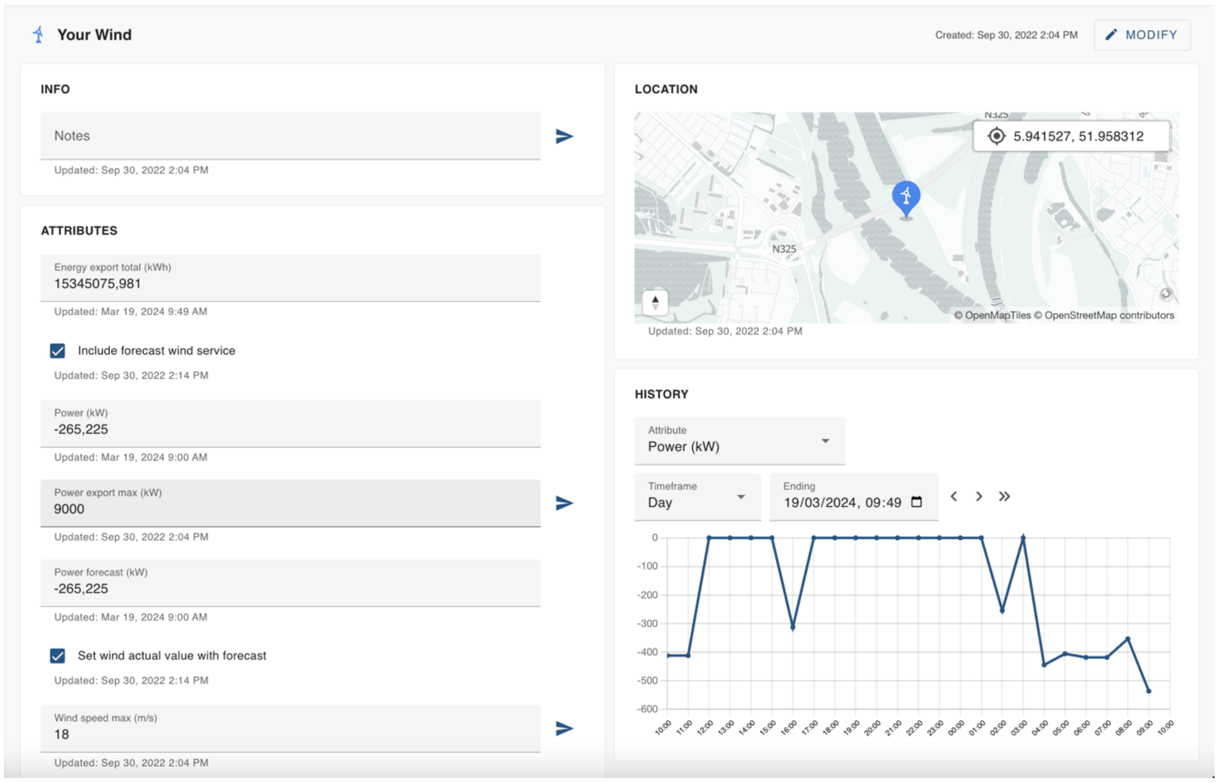
Figure 2 the asset view for an asset of the type 'electricityProducerSolar' with its attributes (left) and the 'Include solar forecast' attribute turned on. On the right side one of the selected attributes is shown in a chart, as well as the location (also an attribute).

The **wind power forecast** is based on the generic characteristics of wind turbines and the forecasted wind speed. The turbines are specified with a minimum (cut-in), reference (rated speed), and maximum wind speed (cut-off or cut-out) characteristics. The power reference speed is specified (power export max) and the power is assumed exponential with wind speed, using formula 1:

$$P = 0.5\, Cp \cdot \rho \cdot R^2 \cdot V^3 = C_{manufacturer} \cdot V^3 \; (1)$$

where
Cp: coefficient of performance (efficiency factor, in percent)
ρ: air density (in kg/m3)
R: the blade length (in meters)
V: the wind speed (in meters per second)
Cmanufacturer: wind turbine characteristic derived from model and brand

The forecast of the wind speed requires a connection to openWeather to retrieve the forecast at the location of the turbine. For configuration of the wind power forecast service OpenRemote offers an online user guide (OpenRemote, User Guide: Energy Management, configuration of wind turbine production forecast).


Your Wind
Created: Sep 30, 2022 2:04 PM

INFO
Notes
Updated: Sep 30, 2022 2:04 PM

ATTRIBUTES
Energy export total (kWh)
15345075,981
Updated: Mar 19, 2024 9:49 AM

☑ Include forecast wind service
Updated: Sep 30, 2022 2:14 PM

Power (kW)
-265,225
Updated: Mar 19, 2024 9:00 AM

Power export max (kW)
9000
Updated: Sep 30, 2022 2:04 PM

Power forecast (kW)
-265,225
Updated: Mar 19, 2024 9:00 AM

☑ Set wind actual value with forecast
Updated: Sep 30, 2022 2:14 PM

Wind speed max (m/s)
18
Updated: Sep 30, 2022 2:04 PM

LOCATION
5.941527, 51.958312
Updated: Sep 30, 2022 2:04 PM

HISTORY
Attribute
Power (kW)
Timeframe
Day
Ending
19/03/2024, 09:49

Figure 3 the asset view for an asset of the type 'electricityProducerWind' with its attributes (left). On the right side one of the selected attributes is shown in a chart, as well as the location (also an attribute).

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101096490

For **energy demand forecast** the CEMS offers a generic time series forecasting solution based on weighted exponential averaging. This method is used for energy demand forecasting, by looking back at historical values for the same time and day in the previous weeks. For details we refer to the user guide (OpenRemote, User Guide: Time series forecasting).

3.3 Optimisation services

The Community EMS includes a built-in optimisation service which controls flexible assets with the objective to maximise self-consumption and/or minimise costs. In section 3.3.1 we will first elaborate on the configuration of the optimisation, before addressing the actual optimisation routine for an energy community member (section 3.3.2) and the energy community as a group of members with shared assets (section 3.3.3)

3.3.1 Configuration of assets and optimisation

The optimisation service is represented as an asset type 'Optimisation'. The optimisation service will consider all assets which are listed as children of an optimisation asset. An example of the combined asset tree of the CEMS you have built must look like the one in figure 4. The following assets must be part of the asset tree:

- Electricity consuming assets, e.g. linked to an electricity meter.
- Electricity producing devices, e.g. solar panels, or wind turbines.
- Flexible assets, which can be controlled via a 'Power setpoint', e.g. batteries, chargers, electric vehicles connected to a charger, or electric heaters.
- Electricity supplier, as maximum power import and export values need to be set, but also dynamic import and export tariffs for the energy and/or network are required.

The optimisation can either target minimising energy costs, taking into account dynamic tariffs, or maximising self-consumption

To set the optimisation to target minimum energy costs the 'Financial weighting' has to be set to 100%. Furthermore the 'Electricity supplier' has to include import- and export tariffs, representing the trading tariffs. (this addresses the High Level Use Case 2 - HLUC2, as specified in D1.2).

The optimisation can also be configured to focus on maximising self-consumption (HLUC1) by setting the 'Financial weighting' to 0% and the supplier asset attributes 'Carbon import' and 'Carbon export' values (kg/kWh). Set the Carbon export values at '0' (as your generated energy is carbon free) and the Carbon import values corresponding with the actual values of the generated electricity on the main grid.

In the next section we will elaborate on the actual optimisation strategy, using both options for both individual energy community members as well as the energy community as a whole.

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101096490

3.3.2 Energy community member optimisation

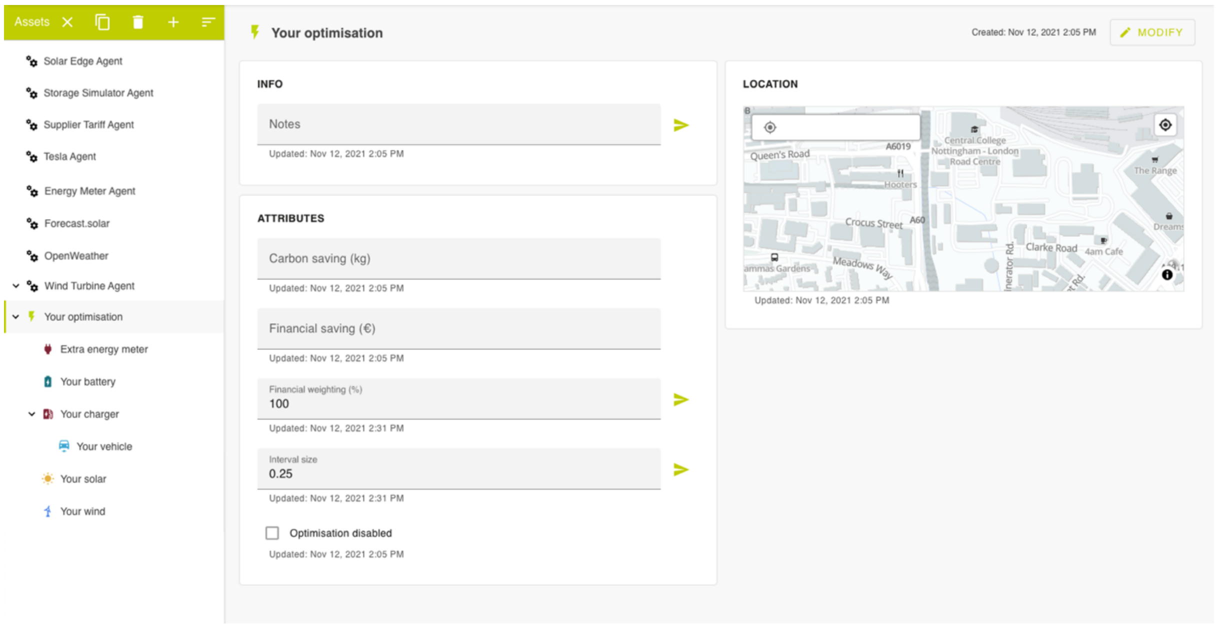
In case of optimising behind a (shared) meter with a single energy bill for this meter, the optimisation routine is also represented as an asset and can be configured in by a single optimisation asset, structuring all assets as children, as indicated in 3.3.1 (also see figure 4):

- Financial weighting (between 0 and 100%). This indicates whether you are optimising on costs (HLUC2) and or carbon. As for now we are assuming financial optimisation here (HLUC2, optimising on trading tariffs) set this value at 100%. In a similar manner, using carbon tariffs (an optional attribute of the Supplier asset) you can optimise on carbon by setting the 'Financial weighting' at '0' (HLUC1, maximise self-consumption).
- Interval size (the interval at which the optimisation runs again, in hours)
- Optimisation disabled (an on/off). This attribute can only be changed in the 'MODIFY' mode, to (temporarily) disable optimisation.

Optimisation will make the following considerations:

- It considers the net resulting power injected into the grid (export) or taken from the grid (import) as the difference between export and import tariffs influence the resulting costs.
- It takes into account the forecasted power, both for electricity consumers and producers, to forecast when there is a net surplus towards the grid or a net shortage of electricity.
- The future export tariffs are taken into account, as it is preferred to charge batteries or vehicles at the time of lowest export tariffs. At that time, you would be earning the least by selling, so better store energy.
- The future import tariffs are considered, as it is preferred to discharge batteries or vehicles at the time of highest import tariffs. You will pay the highest tariffs if you buy electricity, so better use your stored energy.
- It considers the Energy level schedule (state of charge) of your vehicle into considerations as well as their Energy level percentage min, and always take care that the energy levels are met in time. The energy levels are set by the vehicle owner who decides on the available minimum mileage for the vehicle.
- It prioritises the different batteries, in this case your static battery and your vehicle, based on the so called Levelized Cost of Storage (LCOS). LCOS is the additional costs of charging or discharging a battery. It reflects the costs of your battery divided by the capacity times the maximum number of charging cycles, so representing an amortisation. This can be implemented by adding the optional attributes 'Tariff import' and 'Tariff export' to your Electric vehicle asset and your Battery. Note that you should keep the 'Tariff import' value for your vehicle at '0' as charging your vehicle is anyhow required to be able to drive, and the optimisation will not introduce extra charge cycles. We haven't included this in our tutorial and recommend to read-up expert articles before applying. An interesting reference to compare different storing technologies and future development is published by Science Direct (ScienceDirect, 2019).

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101096490



Assets

- Solar Edge Agent
- Storage Simulator Agent
- Supplier Tariff Agent
- Tesla Agent
- Energy Meter Agent
- Forecast.solar
- OpenWeather
- Wind Turbine Agent
- Your optimisation
  - Extra energy meter
  - Your battery
  - Your charger
    - Your vehicle
  - Your solar
  - Your wind

Your optimisation

Created: Nov 12, 2021 2:05 PM MODIFY

INFO

Notes
Updated: Nov 12, 2021 2:05 PM

LOCATION

Updated: Nov 12, 2021 2:05 PM

ATTRIBUTES

Carbon saving (kg)
Updated: Nov 12, 2021 2:05 PM

Financial saving (€)
Updated: Nov 12, 2021 2:05 PM

Financial weighting (%)
100
Updated: Nov 12, 2021 2:31 PM

Interval size
0.25
Updated: Nov 12, 2021 2:31 PM

Optimisation disabled
Updated: Nov 12, 2021 2:05 PM

Figure 4. The asset tree of your EMS on the left, with the Optimisation asset selected. All assets, considered by the optimisation need to be added as children. The right side reflects the configuration options the optimisation offers.

The last item to explain are the two attributes 'Financial saving' and 'Carbon saving'. These two values are reflecting the EXTRA savings you are achieving due to the optimisation using the static battery in a smart way and schedule the charging of your vehicle at the best time. To know your electricity bill, you should of course add up the total imported energy times the import tariff, and the total exported energy times the export tariff.

3.3.3 Energy Community optimisation

In the case of optimising an energy community, we assume that energy sharing between different households is allowed but households still pay an individual bill. In addition, we assume shared assets, e.g. a community solar field, a shared battery, shared chargers, or shared consumption.

To enable an optimised financial situation for all participants the following set-up needs to be established

- Set up all individual households the same way as described in section 3.3.2. These can be listed next to each other in the asset tree, as the community EMS can manage multiple optimisations at the same time.
- Set up a community optimisation asset under which all shared assets are allocated. Similar as described in section 3.2.2.
- For the electricity supplier asset, as a child of the community optimisation asset, the asset should include the contractual import- and export tariffs from the energy provider and network provider.
- For the electricity supplier asset, as a child of the individual household optimisation assets, the asset should include the effective import- and export tariffs from the energy provider and network provider. Normally these internal tariffs are part of the contractual agreement the individual households agree upon within a community.

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101096490

3.4 Integration external devices and services

3.4.1 Protocol agents

To connect the CEMS to external services or devices, as a client, the EMS includes a range of generic protocol agents. These agents include configuration options to enable authentication, subscribing and publishing based on a visual interface. The Agents are also visually represented as assets.

More information about protocol agents can be found in the online OpenRemote documentation (OpenRemote, User Guide Agent Overview).

3.4.2 Manager APIs

Devices and services can also connect to the EMS as client, the EMS being the server. For that the EMS includes three types of API's, MQTT (broker), Websocket and HTTP, with proper authentication methods. More information about the manager API's can be found in the online documentation (OpenRemote, User Guide, Manager API's).

3.5 Visualisation and dashboards

3.5.1 Insights dashboard

The CEMS includes a dashboard builder, called 'Insights' with several types of widgets (see figure 5). It allows for visualising data in real time as well as add control elements to it. Moreover, these dashboards can be shared as standalone apps to individual users. It can therefore be tailored to any user, both individual and community level, of the CEMS.

Sharing the dashboards via the generic iOS and Android apps, not just allows mobile view but also using the functionality for push notifications and/or geo-fencing. More information about how 'Insights' works can be found in the OpenRemote documentation (OpenRemote, User Guide Manager UI).

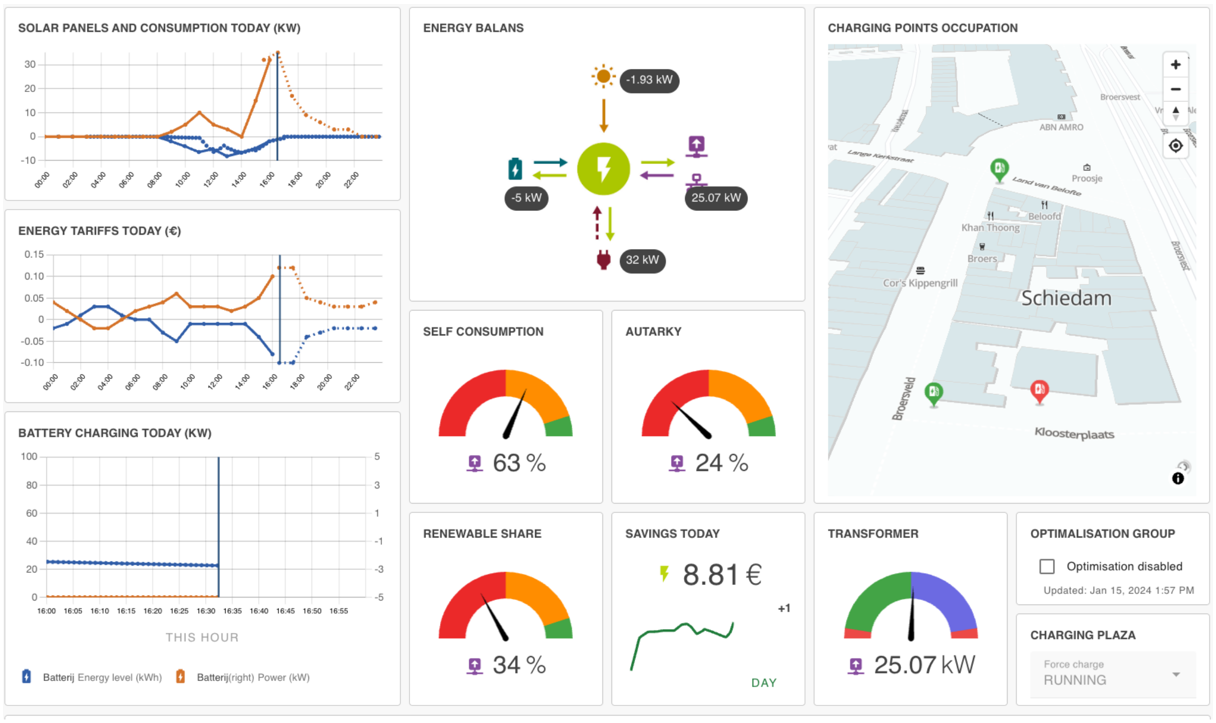


Figure 5. An example of a dashboard build for a user, within the 'Insights' functionality of the CEMS.

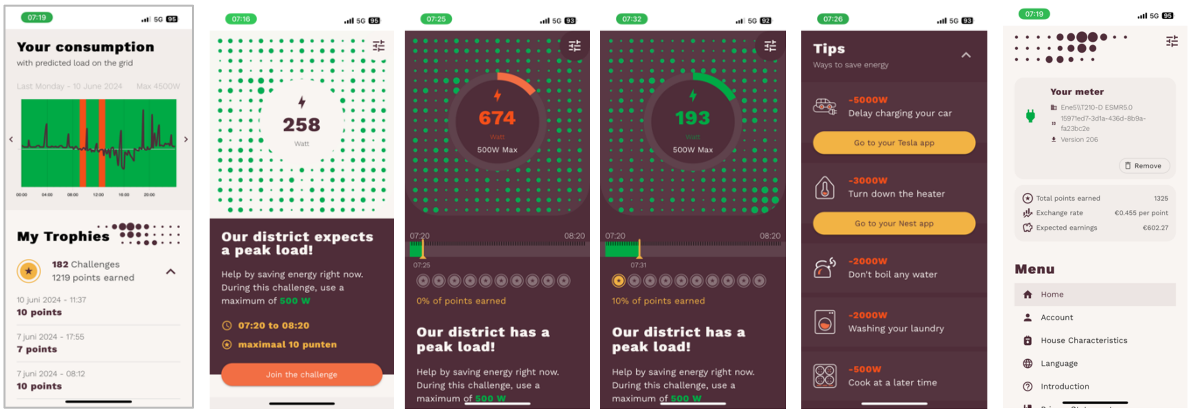
This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101096490

3.5.2 Reschool OurGrid app for community users in Amsterdam pilot

The intention of the Amsterdam pilot is to proof that within the district Sporenburg a high percentage of residents is willing to participate and change their energy consumption behaviour in such a way that grid congestion can be avoided. Therefore, the generic CEMS optimisation services are only applied to automatically control shared flexible assets (public or shared charging infrastructure). To warrant scalability today, individual users with flexible loads are participating via challenges triggered by the optimisation service. This prevents the technical complexity of integrating many different flexible loads which are today not standardised or flexible loads which even don't have a way to digitally connect. Therefore any user can participate.

For this purpose the CEMS includes an additional mobile app for community users, called OurGrid. This app (figure 6) is available for iOS and Android (see Apple Store and Google Playstore). The OurGrid app offers the following functionality to community users:

- Realtime power data to be able to evaluate your current power consumption and therefore be able to asses which devices are currently using much power
- Realtime insight in whether there is any current of forecasted critical load on the district transformer (based on solar forecast and demand forecast services)
- Receive challenges at the moment a critical load is expected (based on forecast) at the transformer level. These challenges include a personalised power target (based on forecasted individual demand) as well as personalised 'tips' on how the personalised power target can be achieved. If the set target is met for a full hour 10 points are earned.
- An overview of earned challenge as well as peak points and financial value for points earned through challenges. A peak point can be earned for every day that users use more than 75% of their energy consumption outside critical time windows for the transformer.
- Settings and configuration. Users can create an account (and remove it), add their own meter and set their house characteristics.



182 Challenges
1219 points earned

10 juni 2024 - 11:37
10 points

7 juni 2024 - 17:55
7 points

7 juni 2024 - 08:12
10 points

Our district expects a peak load!

Help by saving energy right now. During this challenge, use a maximum of 500 W

07:20 to 08:20

maximaal 10 punten

Join the challenge

0% of points earned

Our district has a peak load!

Help by saving energy right now. During this challenge, use a maximum of 500 W

10% of points earned

Our district has a peak load!

Help by saving energy right now. During this challenge, use a maximum of 500 W

Tips

Ways to save energy

-5000W
Delay charging your car

Go to your Tesla app

-3000W
Turn down the heater

Go to your Nest app

-2000W
Don't boil any water

-2000W
Washing your laundry

-500W
Cook at a later time

Your meter

Ene5\\T210-D ESMR5.0
15971ed7-3d1a-436d-8b9a-fa23bc2e
Version 206

Remove

Total points earned	1325
Exchange rate	€0.455 per point
Expected earnings	€602.27

Menu

- Home
- Account
- House Characteristics
- Language
- Introduction

Figure 6 Screenshots of the Reschool OurGrid app for users in the Amsterdam Sporenburg pilot. From left to right: real-time insight in current consumption of user as well as indication of district transformer by green/red background (1); When critical load is expected on district transformer, users receive a challenge (2-4); personalised tips are given linked to the users home situation (5); meter can be set as well as home settings, and earned points can be tracked (6).

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101096490

4 Documentation and dissemination

4.3 Reschool Community EMS

The Reschool CEMS is applied in three of the four pilot locations. For the Amsterdam pilot, the complete software solution including forecasting and optimisation services is used.

In Girona and Athens, some parts of the solution are used, see Table 1, as well as Section 4.

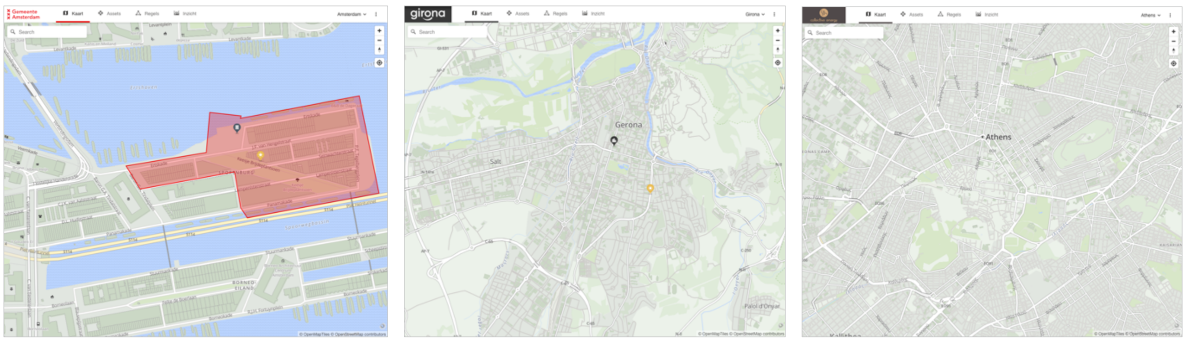


Figure 6. The map overviews in the Community EMS as present and used for three of the four pilot locations: Amsterdam (left), Girona (middle), and Athens (right).

The online environments can be found at

- Amsterdam Reschool Community EMS
- Girona Reschool Community EMS
- Athens Reschool Community EMS

4.4 GitHub code and developers documentation

All the EMS code as described is merged within the OpenRemote main repository, which can be found on the OpenRemote GitHub (OpenRemote, GitHub repository).

Reference developers documentation can be found as online documentation (OpenRemote, Online documentation).

4.5 Get started EMS users

For setting up the Community Energy Management System OpenRemote has set up a 'Get Started' User guide (OpenRemote, Tutorial: Create your Energy Management System).

4.6 Forum

The OpenRemote Forum is gathering an online community for users of both the generic OpenRemote platform as well as the Energy Management application. See OpenRemote Forum.

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101096490

5 Conclusions

Energy communities have a wide range of requirements, both functional , technical and legal. This diversity is represented within the Reschool pilots. Therefore, potentially there is a big leap to make when translating a generic approach from the architecture as described in D3.1 into an open source Community EMS as presented in this deliverable.

Based on the different contributions from partners, we have however managed to successfully define a common set of features. Moreover, we have developed, documented and implemented the open source version of the CEMS with already three of the four pilots starting to implement it, or parts of it.

Although first implementations look promising in terms of the addressed scope of features and practical usability, it's of course too early to draw conclusions. Final implementation and performance will therefore be evaluated as part of deliverable 4.

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101096490

References

Forecast Solar, restful API for solar production forecast, https://forecast.solar/

OpenRemote, Get Started Community EMS, Tutorial: Create your Energy Management System

OpenRemote, Assets Agents and Attributes, definition and models within OpenRemote. https://docs.openremote.io/docs/user-guide/assets-and-attributes/assets-agents-and-attributes/

OpenRemote, User Guide: Energy Management, Electricity Assets and required Agents, https://docs.openremote.io/docs/user-guide/domains/create-your-energy-management-system/#electricity-assets-and-required-agents

OpenRemote, GitHub code reference for energy asset models, https://github.com/openremote/openremote/tree/master/model/src/main/java/org/openremote/model/asset/impl

OpenRemote, User Guide: Energy Management, configuration of solar production forecast, https://docs.openremote.io/docs/user-guide/domains/create-your-energy-management-system/#pv-solar-asset

OpenRemote, User Guide: Energy Management, configuration of wind turbine production forecast, https://docs.openremote.io/docs/user-guide/domains/create-your-energy-management-system/#wind-turbine

OpenRemote, User Guide: Time series forecasting, https://docs.openremote.io/docs/user-guide/rules-and-forecasting/forecasting/

OpenRemote, User Guide: Agents and Protocols, https://docs.openremote.io/docs/category/agentsprotocols/

OpenRemote, User Guide: Manager APIs, https://docs.openremote.io/docs/user-guide/manager-apis/

OpenRemote, User Guide: Manager UI, using dashboards, https://docs.openremote.io/docs/user-guide/manager-ui/

OpenRemote, GitHub, https://github.com/openremote/openremote

Science Direct, Oliver Schmidt, Sylvian Melchior, Adam Hawkes, Iain Staffell (2019), Projecting the Future Levelized Cost of Electricity Storage Technologies. https://www.sciencedirect.com/science/article/pii/S254243511830583X

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101096490