Grant number: 769016
Project duration: Sept 2018 - Aug 2021
Project Coordinator: Joe Gorman, SINTEF

HORIZON 2020: Mobility for Growth
MG-4.2-2017
Supporting Smart Electric Mobility in Cities
Project Type: Innovation Action

GREEN CHARGE

greencharge2020.eu

GreenCharge Project Deliverable: D5.2

Simulation and Visualisation Tools (initial version)

Authors: Rocco Aversa (SUN), Salvatore Venticinque (SUN), Dario Branco (SUN), Svein Hallsteinsen (SINTEF)

2020 CiViTAS
Cleaner and better transport in cities

www.civitas.eu

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016

About GreenCharge

GreenCharge takes us a few important steps closer to achieving one of the dreams of modern cities: a zero-emission transport system based on electric vehicles running on green energy, with traffic jams and parking problems becoming things of the past. The project promotes:

Power to the people! The GreenCharge dream can only be achieved if people feel confident that they can access charging infrastructure as and when they need it. So GreenCharge is developing a smart charging system that lets people book charging in advance, so that they can easily access the power they need.

The delicate balance of power If lots of people try to charge their vehicles around the same time (e.g. on returning home from work), public electricity suppliers may struggle to cope with the peaks in demand. So we are developing software for automatic energy management in local areas to balance demand with available supplies. This balancing act combines public supplies and locally produced reusable energy, using local storage as a buffer and staggering the times at which vehicles get charged.

Getting the financial incentives right Electric motors may make the wheels go round, but money makes the world go round. So we are devising and testing business models that encourage use of electric vehicles and sharing of energy resources, allowing all those involved to cooperate in an economically viable way.

Showing how it works in practice GreenCharge is testing all of these innovations in practical trials in Barcelona, Bremen and Oslo. Together, these trials cover a wide variety of factors: *vehicle type* (scooters, cars, buses), *ownership model* (private, shared individual use, public transport), *charging locations* (private residences, workplaces, public spaces, transport hubs), energy *management* (using solar power, load balancing at one charging station or within a neighbourhood, battery swapping), and *charging support* (booking, priority charging).

To help cities and municipalities make the transition to zero emission/sustainable mobility, the project is producing three main sets of results: (1) *innovative business models*; (2) *technological support;* and (3) *guidelines* for cost efficient and successful deployment and operation of charging infrastructure for Electric Vehicles (EVs).

The *innovative business models* are inspired by ideas from the sharing economy, meaning they will show how to use and share the excess capacity of private renewable energy sources (RES), private charging facilities and the batteries of parked EVs in ways that benefit all involved, financially and otherwise.

The *technological support* will coordinate the power demand of charging with other local demand and local RES, leveraging load flexibility and storage capacity of local stationary batteries and parked EVs. It will also provide user friendly charge planning, booking and billing services for EV users. This will reduce the need for grid investments, address range/charge anxiety and enable sharing of already existing charging facilities for EV fleets.

The guidelines will integrate the experience from the trials and simulations and provide advice on localisation of charging points, grid investment reductions, and policy and public communication measures for accelerating uptake of electromobility.

For more information

Project Coordinator: Joe Gorman, joe.gorman@sintef.no
Dissemination Manager: Arno Schoevaars, arno.schoevaars@pnoconsultants.com

Executive Summary

The main purpose of this document is to describe the requirements and initial design and implementation of the GreenCharge simulation and visualisation tools. The purpose of these tools is to enable investigation of the possible impacts of the GreenCharge concepts beyond what is possible with the pilots realised in the project alone. To achieve this, we need to simulate scenarios involving larger and more diverse energy smart neighbourhoods than realised in the pilots, based partly on data collected by the pilots and partly on data collected in other contexts.

The GreenCharge simulator builds on a simulation tool developed by a previous FP7 project (CoSSMic). However, to satisfy the requirements of GreenCharge it had to be extended in several aspects:

- support for simulating additional energy consuming devices (charging EVs, heating/cooling devices) and electric energy storage devices.
- support for simulating business models for the fair sharing of benefits among stakeholders.
- support for visualising and analysing the results have to be adapted for the set of KPIs defined for the evaluation of the GreenCharge concept.
- support for better insight into the working of the optimisation algorithm to build confidence in its correctness and enable improvements

In this initial step we have focused on capturing the requirements, developing an overall design and implementing necessary changes to the simulation framework to accommodate the extended capabilities listed above.

The deliverable, after defining the role of simulation in the project evaluation methodology and after having described the baseline of the CoSSMic project, illustrates in detail the software requirements of the simulator, the software architecture and the technologies on which its implementation is based.

Finally, the first test simulation scenarios have been defined using some of the Use Cases foreseen in the Project Pilots as a reference model.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

Table of Contents

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

Table of Figures

List of Tables

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

1 About this Deliverable

1.1 Why would I want to read this deliverable?

This core content of this deliverable is software: the initial prototype version of the simulation and visualisation tools will complement the other evaluation activities and tools envisaged in the project. This accompanying document contains the description of the requirements, design, software architecture and initial implementation of the software. Furthermore, through the description of simple usage scenarios and the illustration of the intuitive developed GUI it also serves as a quick user guide for using the tool.

1.2 Intended readership/users

This deliverable is essentially aimed at project partners who are involved in the evaluation activities and want to use simulation to complement the analysis of the data recorded in the pilots. This document and the available software made may also be of interest to users who intend to deepen their understanding of the conceptual model underlying the innovative technologies introduced by the GreenCharge project.

1.3 Other project deliverables that may be of interest

- D5.1&D6.1 Evaluation Design / Stakeholder Acceptance Evaluation Methodology and Plan for the design of the GreenCharge evaluation methodology.
- Deliverable 4.1: Initial Architecture Design and Interoperability Specification to frame the simulation tool into the GreenCharge reference architecture for smart and green charging.
- Deliverable D2.1: Initial Strategic Plan for Pilots to identify and design the possible simulation scenarios inspired by the use cases developed in the Pilots

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

2 Evaluation methodologies

As described in Deliverable D5.1 one of the main objectives of the GreenCharge project is the evaluation of the impact on e-mobility that the specific technology can provide when it is used in a certain pilot.

In general, three evaluation methods will be used in the GreenCharge project.

- *Evaluation of stakeholder acceptance based on manual data collection methods, such as surveys and interviews.* Workshops will be organized in each Pilot. Surveys will be distributed to participants and to volunteers who are available to provide their feedbacks manually or using on-line tools
- *Evaluation of technology with analysis based on automatic data collection*. In GreenCharge Pilots users behaviour and energy utilization in charging stations will be monitored. Data will be collected automatically, e.g., via system logs, user Apps, while different business models and supporting innovative technological solution are applied. Data analytics and numerical models will be used to evaluate KPIs.
- *Evaluation of technology based on simulations*.

The use of simulation for evaluation purposes arises from the need to overcome the intrinsic limitations of Pilots in terms of scalability, configuration, data availability, regulation, and time and effort constraints. The objective of this document, in particular, is to define on one hand the role of the simulation in the project evaluation activities and on the other hand to identify the software requirements, the kind of scenarios to simulate potential limitations of the simulation approach.

2.1 Assessment of objectives and key performance indicators

The GreenCharge evaluation methodology will be based on CIVITAS Evaluation Framework. CIVITAS is a network of cities dedicated to cleaner, better transport in Europe and beyond. Since it was launched by the European Commission in 2002, the CIVITAS Initiative has tested and implemented over 800 measures and urban transport solutions as part of demonstration projects in more than 80 Living Lab cities Europe-wide.

GreenCharge has adopted and customized the CIVITAS Evaluation Framework, focusing on e-mobility, in order to exploit the valuable results of that project in terms of methodology and procedures, but also to contribute to the work of the CIVITAS network.

The main concept of CIVITAS Evaluation Framework are *Measures* and *Key Performance Indicators (KPIs)*. A *Measure* defines one or more aggregated activities which are implemented to have a positive impact on city transport. A *KPI* corresponds to one or more aggregated parameters which allow for a quantitative evaluation of the impact produced by a Measure on a specific aspect of transport. The adoption of the evaluation framework has consisted in the specialization of Measures and KPIs related to the e-mobility context, with a special attention to technology evaluation and to the assessment of stakeholder acceptance. As it is shown in the evaluation loop of Figure 1, aggregated technologies and business models will be operated by Pilots and evaluated in different WPs. Data will be collected at different milestones of the project. Intermediate and final KPIs evaluations will be scheduled. Evaluation of results consists of a quantitative and a qualitative analysis for a subset of defined KPIs, which will be selected in different Pilots according to their relevance in the specific context. As it is shown in Figure 1, when feasible, the evaluation of KPIs should highlight comparison to baseline values, for better identifying the impact related to the innovation introduced by the GreenCharge technology.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

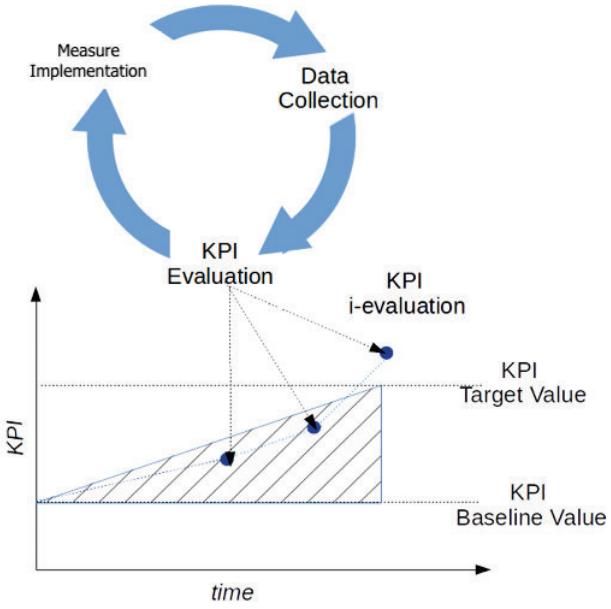

Figure 1 GreenCharge Evaluation Loop

Simulation will be used in the *GreenCharge Evaluation Loop* to operate the measure in a virtual environment where the Pilots can be extended overcoming real limitations and the measure can be easily complemented with missing functionalities.

This kind of approach will also allow the evaluation of KPIs at design time, to predict the return of investment or to optimize the dimensioning and positioning of new charging stations, as well as the acquisition of new EVs.

2.2 GreenCharge scenarios and simulation requirements

The GreenCharge project has defined some general scenarios which can be implemented integrating innovative technologies according to the GreenCharge reference architecture sketched in Figure 2.

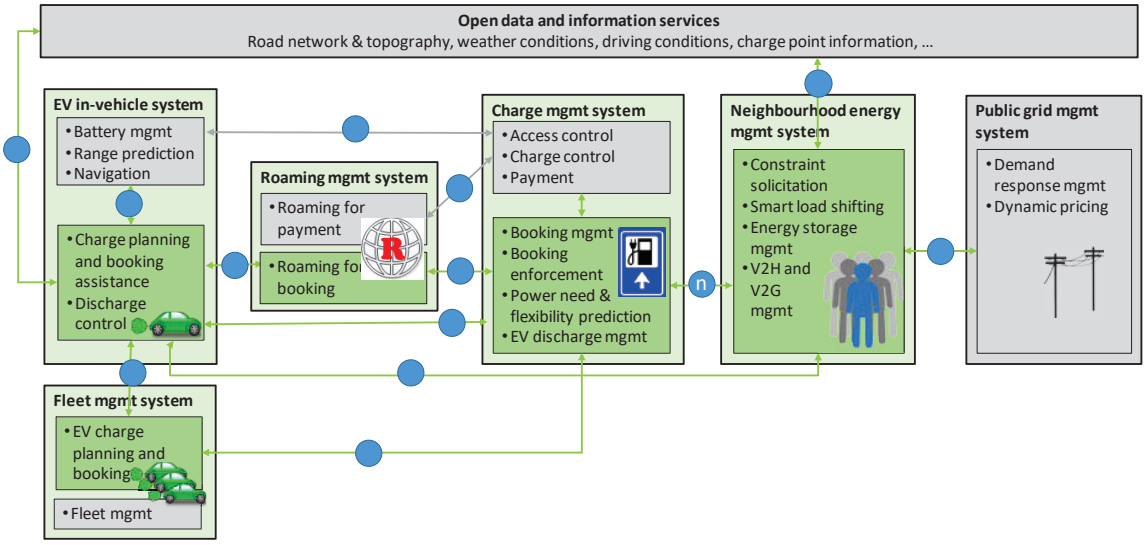

Figure 2 GreenCharge System Architecture (from DoW)

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

The GreenCharge simulator has extended the CoSSMic simulator [1], that represents the software baseline of the project to support the simulation of specific functionalities of selected scenarios.

Whereas the CoSSMic simulator allowed for the estimation of energy related performance indicators, such as self-consumption from Renewable Energy Source (RES) in a neighbourhood, one of the objectives here is to evaluate the impact of Energy Smart Neighbourhood technologies with respect to innovative business models for e-mobility.

In order to introduce the simulation requirements, which lead the design of GreenCharge simulator, we briefly recap the project scenarios defining the main subsystem roles, their responsibilities and their main interactions (from DoW).

In the next subsections, we summarize the scenarios described in the Description of Work (DOW) of the project. Afterwards, we analyse a possibly simulation coverage of the predefined scenarios, defining for each of them the first needed requirements to adapt or extend the CoSSMic simulator. Finally, the necessary extensions will be selected and implemented according to the *what-if scenarios* defined in D5.1, by which the Pilots identified how the simulation can support them to overcome the limitations of project size and implementation performing evaluations which could not be possible in other ways.

2.2.1 The GreenCharge Scenarios as described in the DoW

Scenario 1: Charge planning and booking

1. An EV driver needs to take a trip that exceeds the range of the vehicle (with its current state of charge - SOC), and thus needs to fast-charge underway. EV in-vehicle system gets destination from user (a) and plans the route.
2. EV in-vehicle system gets data from Open data and information services (f) regarding weather, driving conditions and charging opportunities along the route, computes expected progress and selects possible charging stations.
3. EV in-vehicle system sends charge request to Charge management system of selected charging stations via Roaming management system (b,i).
4. Charge management system sends power request to Neighbourhood energy management system (n).
5. Neighbourhood energy management system allocates power and returns offer via Roaming management system (i,b).
6. EV in-vehicle system evaluates offers, selects best fit and returns accept/reject to offer providers (c/b,i); The selection criteria should include suitable time, local RES, short detour, suitable surroundings (for spending waiting time, see below), etc.
7. Charge management system cancels power requests for rejected offers (n).
8. During the trip the EV in-vehicle system monitors progress and if the deviation from the predicted progress exceeds the threshold, the EV in-vehicle system re-computes expected progress and re-plans and rebooks the charging stop.

The planning may take place when the driver gets into the car to start the trip or earlier. If it takes place in advance, already planned trips before or after the one being planned will also be considered (e.g. for estimating SOC at trip start, and possibly preparing for a following trip). Furthermore, the planning may consider other tasks on the driver's agenda, e.g. eating, shopping or having a hair-cut, that may be carried out while charging, to the extent such information is available.

Scenario 2: Charging at booked Charging station

1. EV approaches booked charging station and sends an approaching message to the Charge management system.
2. Charge management system sends guiding info to the EV in-vehicle system who displays it to driver to assist the navigation to the booked charging point. This could include a detailed map of the charging

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

facility, with indication of route to the booked charging post, the location of the EV and audio directions, like the navigation system normally works. Ideally this functionality would be seamlessly integrated with the navigation system.
3. The EV parks at the booked charging post and the EV in-vehicle system authenticates the EV (g) to the Charge management system.
4. The Charge management system controls the charging, making sure to obey the constraints provided in the booking, and in collaboration with the Neighborhood energy management system leveraging any flexibility.

Scenario 3: Booking Enforcement

An inherent practical problem of implementing booking of charging spots is that other cars may park at and block the allocated charging spot. Physical obstacles that can be controlled remotely exists but are expensive. Other solutions may be:

1. Charge management system instructs Charging post to display clearly the availability/non-availability for drop-in customers and blocks charging for other EVs in booked time slots.
2. Drop-in customers must also indicate the time-slot they will be parked by the selected charging post, and the Charge management system may enforce restrictions in busy periods with many bookings.
3. Parking at a booked spot, or leaving the vehicle by a charging post longer than agreed may cause punishment, e.g. a fine or higher price or blacklisting.

A good strategy to avoid practical problems with booking, while still ensuring good utilization of the charging equipment, may be to have more parking spots with connectors (cheap) than chargers (expensive). The final assignment of charging post for booking customers could be postponed until arrival time, leaving more flexibility to sell free slots in between bookings to drop-in customers.

Scenario 4: Home charging in older (groups of) residential or working buildings with common internal grid and parking facilities, or at work in (groups of) buildings with similar limitations

1. In the afternoon, many people return home and connects their EV to their home charging point.
2. Those who need the car soon again indicate that to the charge planning assistant.
3. Rather than starting the charging of all the cars immediately after connection, the ESN management system, possibly in collaboration with a local charge management system, schedules the charging of the different vehicles according to their expected future use and SOC, so as to exploit as far as possible locally produced electric energy while also considering other tasks that need electric power in the neighborhoods.

The internal electricity distribution grid in older (groups of) buildings often have limitations that cause problems when inhabitants want to charge EVs at home

Installing a neighbourhood's energy management system for the (group of) buildings and a Charge management system supporting booking for the charging facilities, would avoid overloading and ensure optimal use of the available capacity, and if desirable, take care of the distribution of cost among the users. It would also open the possibility to sell excess capacity to outsiders, which if the facility is conveniently located, could recover the investment.

Scenario 5: V2G

On a cloudy afternoon, the ESN does not have enough local energy production from PV panels to cover the need. However, during the sunny morning, several connected EVs with still some time left before their agreed deadline were already charged above the required charge level. In this situation, the excess energy stored in the car batteries are fed back into the neighbourhood.

During the morning when the sun is shining (as forecasted) and excess local production is foreseen, the Neighbourhood energy management system instructs the Charge management system to "overcharge" vehicles, preferably ones with planned connection times stretching into the afternoon when the forecast predicts little local production.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

1. In the afternoon, a household in ESN schedules a load, e.g. the running of a washing machine or a water heater on-cycle.
2. The Neighbourhood energy management system finds that local RES has been exhausted in the allowable period for the load. Also the possibilities to delay other loads, e.g. charging EVs has been exhausted. But there are "over"-charged EVs with planned connection time overlapping the allowable period for the load.
3. The Neighbourhood energy management system schedules the load in the overlapping period and instructs the Charge management system to discharge the battery of the selected EVs in accordance with this schedule.
4. The Charge management system controls the discharging of excess power from the selected EV batteries accordingly, and if necessary the recharging to comply with the constraints.
5. The Charge management system in collaboration with the Neighborhood energy management system ensures that the owners of the involved EVs and local PV systems are compensated for the additional discharge/charge cycle and for the provided excess energy.

Scenario 6: Reacting to Demand Response (DR) request

1. Public grid requests demand reduction or feed-in (q)
2. For acute requests, the Neighbourhood energy management system reschedules already scheduled flexible loads and/or exploits charging EV batteries to satisfy the request.

Scenario 7: eMobility in innovative 'mobility as a service' (MaaS)

Car sharing as a fleet-based service is used as innovative element of SUMPs to reduce the consumption of space for parked cars – and to introduce electric vehicles. Car sharing can widely replace car ownership – makes more efficient use of transport / parking infrastructure. Users have access to the cars of the fleet via electronic reservation and access. There are different business cases involved: for the cities, for the users/citizens, for car sharing operators, for recharging infrastructure operators, for housing companies.

It needs proper communication between cars and the fleet management system e.g. about the state of the battery in order to optimise the charging of the cars in relation to the next reservations of cars at the very car sharing station. For satisfied customers, it is necessary to provide cars with sufficient battery charge for the planned trip.

With the reservation, the user tells that he wants to go a certain distance (e.g.100 kilometres) with the e-car.

1. The reservation system checks whether the available cars have a sufficient state of battery at the pick-up time.
2. During the trip, the car communicates the state of battery to allow the management system planning the charging for the follow-up reservation

2.2.2 General requirements

The following table sums up the first needed requirements to adapt or extend the CoSSMic simulator in order to cover a single scenario or a group of scenarios described above.

Table 1 Simulation scenarios requirements

Subsystem role:	Neighbourhood energy management system
Current functions:	Autonomic ICT based system, coordinating the energy usage and storage and the exchange with the public grid of clusters of collaborating buildings with local RES. The system optimizes the consumption to the local energy production in the cluster through coordinated load shifting according to constraints set by the user preferences.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

Needed adaption and extension:	**(scenario 1) (scenario 2) (scenario 4)** - It needs to extend the configuration model of the simulator with the contributions of the pilots (e.g., data collected from pilot). The configuration model of a typical simulation session requires both a static configuration and a dynamic configuration. The static configuration includes the definition of the different energy actors: producers like PVs, consumers like appliances, prosumers like batteries) that make up the neighbourhood system to simulate. The dynamic configuration otherwise describes the specific loads that should be managed and optimized by the Load Scheduler of the simulator. Each load is characterized by its constraints and its flexibility. - It needs to implement a component wrapper that import data (in some predefined formats) from the GreenCharge pilots to simulate the scenarios according to real and significative input. - Support for battery models. The profile should consider the possibility to modulate power to obtain the desired level of charging in a fixed time interval. - The scheduler should handle the power peek in a group of producers/consumers - In GreenCharge it should be introduced an interactive simulation mode that can be paused and resumed configuring breaks in advance. It allows to pause, add events and restart the simulation session from a breakpoint. **(scenario 5)** - V2G. The EV battery must define also the capability to discharge. The scheduler must be able to handle these profiles and to coordinate the producer and the consumer behaviour. **(scenario 6)** - The scheduler should consider dynamic pricing and other DR signals and constraints on the power grid. It needs to define the timing and the format of the updates.

2.3 Role of the simulation for the KPIs evaluation

According to the CIVITAS evaluation methodology, the result of impact evaluation is provided as a quantitative estimation of selected Key Performance Indicators (KPIs).

In this section, we describe the specific role of the simulation methodology in the evaluation of the KPIs defined in the deliverable D5.1, specifying when the simulator should assume a side-by-side or a replacement function of the other proposed methodologies (especially data collected in the Pilots).

The potential role of data collected in the pilots and those computed during the simulation sessions together with the associated KPIs are analysed in Table 2. Some indicators measure the performance of the GC technology and can be investigated further to what is revealed in the pilots by simulations (marked with a D in the kind column), for example energy mix and self-consumption. Other KPIs are characteristics of the context in which the GC solution is deployed (marked with V), for example the number of EVs and the number of parking spaces with charging, but we expect that the availability and knowledge of the benefits of GC technology will accelerate the transition to e-mobility and thus impact these indicators.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

Table 2 KPIs evaluation

Indicator	Kind	Expected effect observed in pilots	Simulation opportunities
Transport system			
GC5.1 Number of EVs	V	Expected to increase during the lifetime of the project in all pilots but influence of the project difficult/impossible to assess.	Scale up in simulations together with the traffic at charging stations generated by the EVs. We can do that by replicating the charging stations included in the pilots for which we recorded data and possibly modify the recorded traffic on each station. Alternatively, we may build a charge traffic generator built on the recorded data to scale up traffic.
GC5.2 Number of parking spaces with charging	V	Additional parking spaces with charging expected at Røverkollen (Oslo) and Barcelona, not sure about Bremen.	
GC5.3 Utilization of charging points	D	Measured by operator. Some existing private charging points will be opened to external users both in Bremen and Oslo, which is likely to increase utilisation.	Compute in simulations for different combinations of varying parameters
GC5.5 Charging availability –(avg waiting time, fraction of demand actually charged,rejection rate)	D	Measured by operator. Could become worse at charging stations included in the pilot due to increased utilisation and energy demand when private CPs are opened to external users but could increase for the external users getting access to more charging facilities.	
Energy supply			
GC5.13 Charging Flexibility	V	Will be asked for and recorded in charging apps, although V2G probably not implemented in pilot. Could change during project as users become more aware of their driving patterns.	Vary in simulations (for example by increasing or decreasing recorded flexibility by a percentage)
GC5.4 Share of battery capacity for V2G	V		
GC5.9 Energy mix	D	Computed by NEMS. Expected to improve in charging stations with dedicated PV plant with stationary battery (Oslo, Bremen) and also in Manresa if adequate NEMS is installed.	Compute in simulations for varying local PV and battery capacity and charging flexibility and amount and flexibility of other neighbourhood demand.
GC5.10 Peak to average ratio	D		
GC5.14 Self-consumption	D		
Economy			
GC5.6 Average operating cost	D	Computed and recorded by operator for pilot charging installations.	Compute the energy cost for varying business models and combinations of other varying parameters. The effect on the economy KPIs will be specific to each charging station and could be estimated by the operator in each case based on the computed energy cost.
GC5.8 Average operating revenue	D		
GC5.7 Capital investment cost	D		
GC5.11 Savings	D		
Environment			
GC5.12 CO2 emission			Compute for different combinations of varying parameters

2.3.1 What-if simulation scenarios

Each GreenCharge Pilot will implement and operate a sub-set of the GreenCharge scenarios, according to the technologies and the business models they are going to implement.

In D5.1 each Pilot has identified those what-if simulation scenarios that allow them for overcoming some limitations affecting the implementation.

For sake of readability we report here *Table 10 of D5.1*, that identifies which KPIs have been identified as a desired simulation output by Pilots.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

The last column of this table defines additional data collection requirements for Pilots which are needed by simulation.

Other data already collected by Pilots have been already revised by the analysis of the initial implementation plan of the three Pilots. Those data will be imported into the simulator to model the devices and the users' behaviours. How the raw data will be used to configure the simulation is part of the extension of the CoSSMic simulator and will be described in Section 4.

Table 3 What-if simulation scenarios (*Table 10 of D5.1)*

What-if scenario	Pilots	Target KPIs	Additional data collection requirements
Baseline with no smart management	Oslo	***GC5.3, GC5.13, GC5.9, GC5.5***	No The simulator must switch of the smart energy management. Different grid capacities can be simulated.
Comprehensive neighbourhood	Oslo	***GC 5.10, GC 5.14, GC 5.9***	External data repository or analytical model of devices, users'behaviour, environmental conditions or energy sources.
V2G	Oslo	***GC 5.9***	V2G specifications (discharging power, available percentage of storage) and users' availability to share their EV battery.
Scale ups #EVs	Oslo, Bremen	***GC 5.3, GC 5.13, GC 5.10***	Number of EV into the extended Pilot, predicted or planned number of EVs in the next future, specification of future EVs (battery capacity, charging power, energy requirements).
Scale up #CP	Oslo, Bremen	***GC 5.3, GC 5.13, GC 5.10***	Number of Charging point into the extended Pilot, predicted or planned number of CP to be installed in the next future, specification of future CPs (charging power).
Scale up #EVs in Eurecat premises	Barcelona	***GC5.9, GC5.10, GC5.12, GC5.13, GC5.14***	No
Scale up #CPs in Eurecat premises combined with an increase of EVs	Barcelona	***GC5.9, GC5.10, GC5.12, GC5.13, GC5.14***	No

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

Local RES to feed battery hub (MOTIT)	Barcelona, Bremen	***GC5.6, GC5.7, GC5.8, GC5.9, GC5.10, GC5.11, GC5.12, GC5.14***	Solar irradiation for the location of the battery hub premises, or time-series of electric energy obtained from PV plants.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

3 Preliminary Design of the GreenCharge Simulator

3.1 Overview of the baseline

In this section we briefly describe the functionality, architecture and status of the CoSSMic simulator, that is the baseline which GreenCharge simulator has built upon.

The CoSSMic project designed and prototyped a neighbourhood energy management system resembling in many ways a neighbourhood energy management system as envisaged in GreenCharge. To evaluate the system and find out more about how the collective and individual benefits of the approach depend on the size and configuration of the neighbourhood, the accuracy of weather forecasts, the price models of the energy providers, and so on the CoSSMic simulator was developed, enabling the execution of the developed system in a simulated environment. In this way we could replicate the user and device behaviours observed in the pilots, varying a number of other factors, such as the configuration of the neighbourhood, the number of PVs, the capacity of the storage systems, the frequency of weather forecasts updates, the price models of the public grid, etc. In addition, using this simulation approach, it was possible to investigate how the CoSSMic distributed architecture scaled with increasing number of households and devices in a neighbourhood.

Since the main goal of the simulator is to generate more data for the evaluation stage of the project, the key design constraint of the tool has been to obtain a model that exactly reproduces (i. e. reusing the developed software components) the main steps of the real prototype software such as: the creation of the shiftable loads; the negotiation and optimization stage using a distributed algorithm; the production of the scheduled loads.

The system is able to simulate a micro-grid and, using an optimizer that utilizes weather forecasts to evaluate the amount of energy produced by solar panel, it is able to calculate the better time to switch on passive devices or to charge batteries. The simulator is able not only to simulate a single micro-grid, but also multiple houses modelling an entire neighbourhood. A micro-grid is typically confined to a smart home or an office building, and embeds local generation and storage of solar power, and a number of power consuming devices. So, using the simulator it is possible to configure a micro-grid, indicating which are the energy consuming devices (washing machine, dishwasher, etc.) and the energy producing or storing devices (solar panel and battery). For some passive devices, that we call pilotable devices, such as washing machine, dishwasher, clothes dryer, the user must set the load, i.e. a time interval that indicates when he wants that a device must be switched on. In other words, the energy demand (LOAD) of a pilotable device is modelled as a cumulative energy timeseries, submitted to the system at a certain time (Creation Time) by the user, with an Earliest Start Time (EST) and a Latest Start Time (LST). A local optimizer, calculates what is the better time to turn on each device, on the basis of the expected energy production of the panels. At the end, using the simulator, the user can see how much self-produced energy she/he is able to utilize following the suggestions of the optimizer.

The simulator is based on the discrete-event simulation (DES) model where the system appears as a discrete sequence of events in time.

Each event has marked with a timestamp (UTC time standard) and produces a change of state in the system. In one of its operating modes the simulation tool was used in the project to acquire the event streams observed during the trials, representing household behaviour patterns together with the weather data, and to reproduce the same flow of events in a simulated environment and in simulated time, where it would be possible to vary parameters such as the storage and production capacity of households etc.

In this way, it is possible to evaluate the impact of the optimization proposed by the project in slightly different situations or increase the size of the problem to assess the scalability of the approach.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

3.2 Description of the main Software Components of CoSSMic Simulator

In this section we describe more in detail the main software components, their functionalities and the technologies used. In figure 3, CoSSMic Simulator main software components are sketched, each of them with the indication, by side, of the specific technology used to develop it:

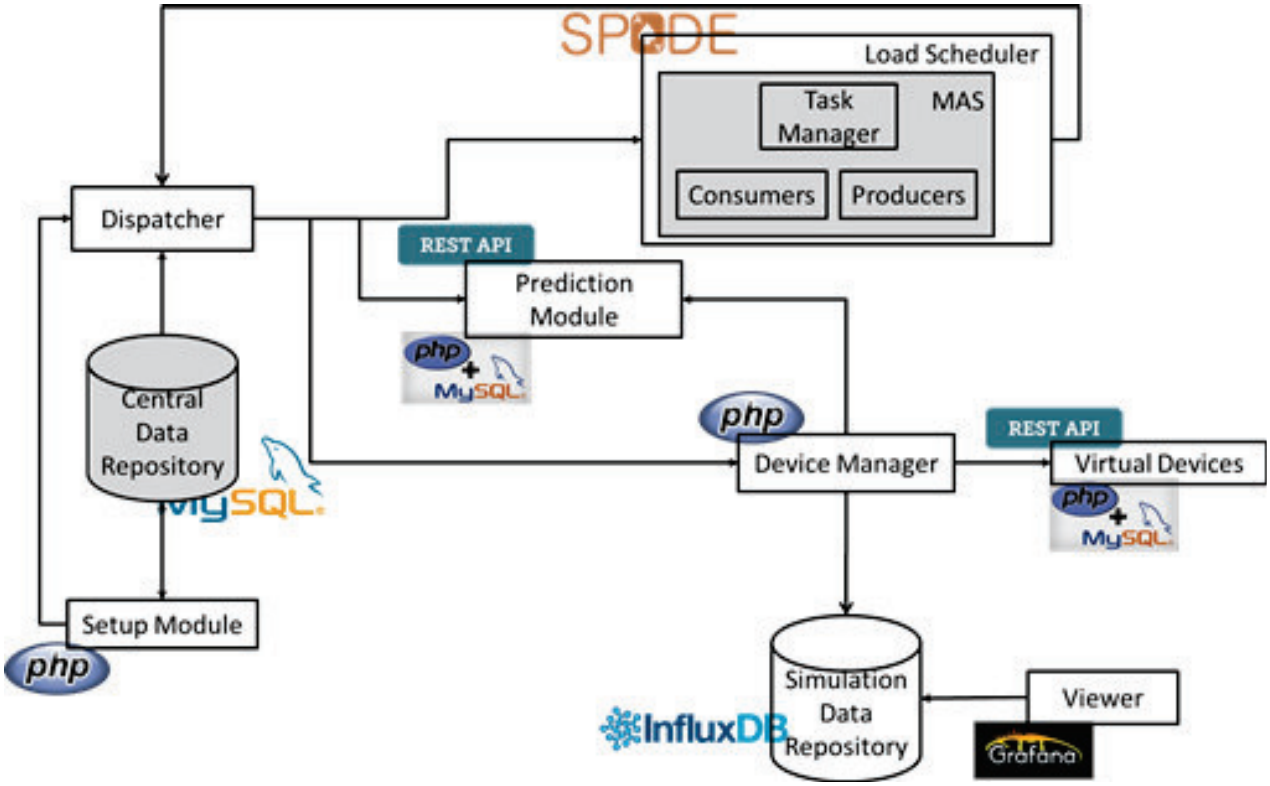

Figure 3 CoSSMic Simulator software components and used technologies

In the **Central Data Repository,** the inputs for the simulation are contained. The inputs are mostly represented by historical data coming from the trials together with the configuration of the test neighbourhoods and of each household that compose them.

The **Setup Module** stores the configuration in the **Central Data Repository** and gives the start signal of the simulation to the **Dispatcher**. It provides a GUI to set configurations and to manage simulations. It is composed of a number of PHP modules.

The **Dispatcher** reads data from the **Central Data Repository** and submits tasks for the households' devices to the Load Scheduler: this one can embed the existing MAS platform and produces the scheduled loads. Once it has received the ASTs from the Load Scheduler, it can give the start signal to the Device Manager. It is a Python module.

The **Device Manager** reads the consumption profile of the devices and stores the relative time series in the Simulation Data Repository. It is composed of a number of PHP modules

The **Simulation Data Repository** contains all simulation results and is based on InfluxDB and is accessed by the Viewer, a web application that enables to analyse the time series graphically.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

The **Viewer** is a web application that accesses the data in the **Simulation Data Repository** and provides a series of graphs to plot and analyse the results of different simulation sessions. It is realized in PHP and Javascript.

The **Prediction Module,** using hourly weather prediction data, predicts the corresponding Photovoltaic (PV) energy. It's a C module.

The **Load Scheduler** implements the **Multi-Actor Load Scheduler** (MALS), responsible of the optimized scheduling of the devices loads generated by all the houses of the neighbourhood. Regarding the distributed negotiation phase used by the optimization algorithm, the **Task Manager**, **Consumer** and **Producer** agents in the simulation tool consist of one Task Manager per household, one Consumer Agent per consumer device and one Producer Agent per producer device.

Setup Module, Dispatcher, Device Manager and Viewer were developed specially from scratch for the simulation tool, whereas **Prediction Module** and **Load Scheduler** were developed for the real CoSSMic prototype installed in the households of the trial sites. The only difference is that the **Load Scheduler** used in the simulation tool works in simulation mode, so that it takes the time not from the system clock but from the simulation engine (the **Dispatcher**) and the **Prediction Module** uses historical data and not the real weather forecast data.

3.3 GreenCharge Simulator: Requirement analysis and Software Design

In the following sections, we illustrate the requirement analysis, the architecture design and the main software components of the initial version of the GreenCharge Simulator.

The main challenges to be addressed in the tool and the major changes to the CoSSMic simulator can be summarized in the following points:

- GreenCharge simulator will extend the CoSSMic simulator to include charging station as an actor of the neighbourhood. Charging station can be defined as a collection of charging points. Besides, in addition to the dishwasher/washing machine/dryer type of device supported in the CoSSMic simulator we need to model and introduces a list of new appliances which will be supported; this list must include EVs/EV chargers including V2G, heating/cooling devices, stationary batteries and non-controlled ("background") loads.
- Finally, new functionalities should be implemented to allow the generation of simulation data input coming from different external sources (e.g. data collected in the Pilots) and to provide output that allow to estimate some GreenCharge KPIs. Of course, the data collected could be used for replicating or for scaling up scenarios recorded in the pilots.

Each of these design aspects of the simulation software tool are addressed in more detail in the next subsections defining the main use cases and software requirements. Requirements analysis will utilize **Unified Modeling Language** (**UML**) modelling language and some of the diagrams provided by the standard like Use Case and Sequence diagrams.

3.3.1 Charging stations as microgrids

According to this model, a charging station can be considered/modelled in the same way as a household with EVs and other devices (e.g., in Oslo pilot, the garage has PV, stationary battery, heating cables and charging points for EVs). Obviously the most remarkable event to record and model in a charging station is the arrival (or the booking) of an electric vehicle that will occupy one of the charging points for a certain period of time. This event surely implies a request for energy (a new Load) with certain characteristics depending on the model of the car, the current and the target charging state of the battery, the expected time of departure, etc. (the parameters of this specific event will be illustrated in the next chapter). Moreover, in GreenCharge the EV can behave also as an Energy Producer, if it supports the V2G technology and the owner allows for a flexible

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

utilization of her battery. In this case, (figure 4) it needs to extend the EV charging model and eventually the simulator, both in terms of accepted input and in terms of scheduling behaviour. In fact, this dual role of an e-car will certainly require on one hand the definition of a more accurate battery charging and discharging model and, on the other hand, proper adaptations in the scheduler to consider this additional degree of freedom in order to optimize load distribution.

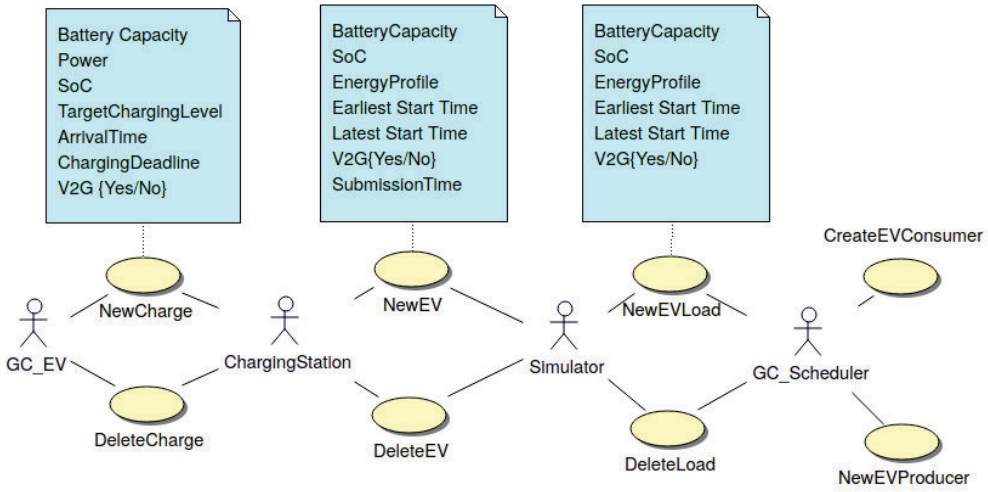

Figure 4. Use Case of EV load in Greencharge.

The sequence diagram of GreenCharge simulator becomes that shown in figure 5.

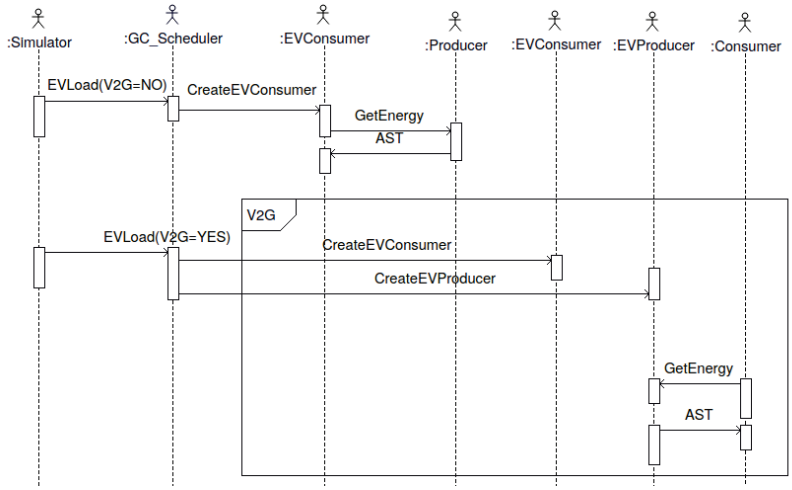

Figure 5. Sequence diagram of Green Charge simulation.

3.3.2 Data Requirements for evaluation based on Simulation

The simulation can be used to evaluate a scenario observed in the past changing or scaling some configuration parameters, which could be the available energy produced by PV panel, the number of EVs, the user flexibility, the scale-up of charging stations. In order to configure a simulation scenario a number of inputs are needed to

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

reproduce the observed reality. Input can be extracted from automatically measured data or from surveys, or can be generated using techniques based on statistical models (figure 6).

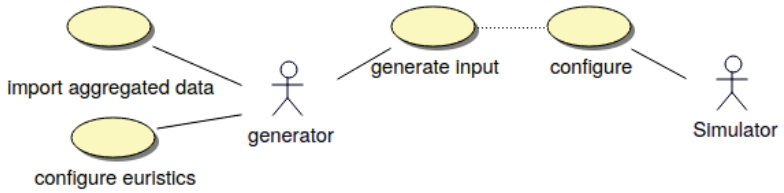

Figure 6 Simulation input

Input can be classified in static input, such as the number of households and appliances or charging points monitored in the charging station, and dynamic input, which are energy demand and the green energy production. Static input will be collected from the analysis of the Pilots or extracting by the surveys prepared for the stakeholders. Dynamic input should be automatically collected, metering energy consumption and production in the Pilots.

Just to give an example, a general simulation scenario, that includes a neighbourhood composed of micro-grids like households and/or charging stations, is characterized by these typologies of energy consumers or producers:

- Appliances with their energy demand;
- PV panels, able to produce green energy;
- Energy storage, which, in turn, produce or consume energy;
- Electric vehicles, which can be modelled as energy storage, but with their own energy demand.

Besides, once defined a simulation scenario it would be useful to have a graphical interface able to:

- select a period to be simulated and the simulator would automatically select appropriate recorded data;
- or for scaling up, specify that a described device or household or entire neighbourhood is to be replicated a given number of times in a simulation.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

3.4 GreenCharge Simulator Software Architecture

In this section, we illustrate a first sketch of the new software architecture with the main components and their interactions, focusing on the following three main software components, developed specifically for this new version of the simulator prototype: **Setup Module; Simulation Input Data Repository**, **External Source** and **Dispatcher**.

Green Charge Simulator architecture is showed in figure 7:

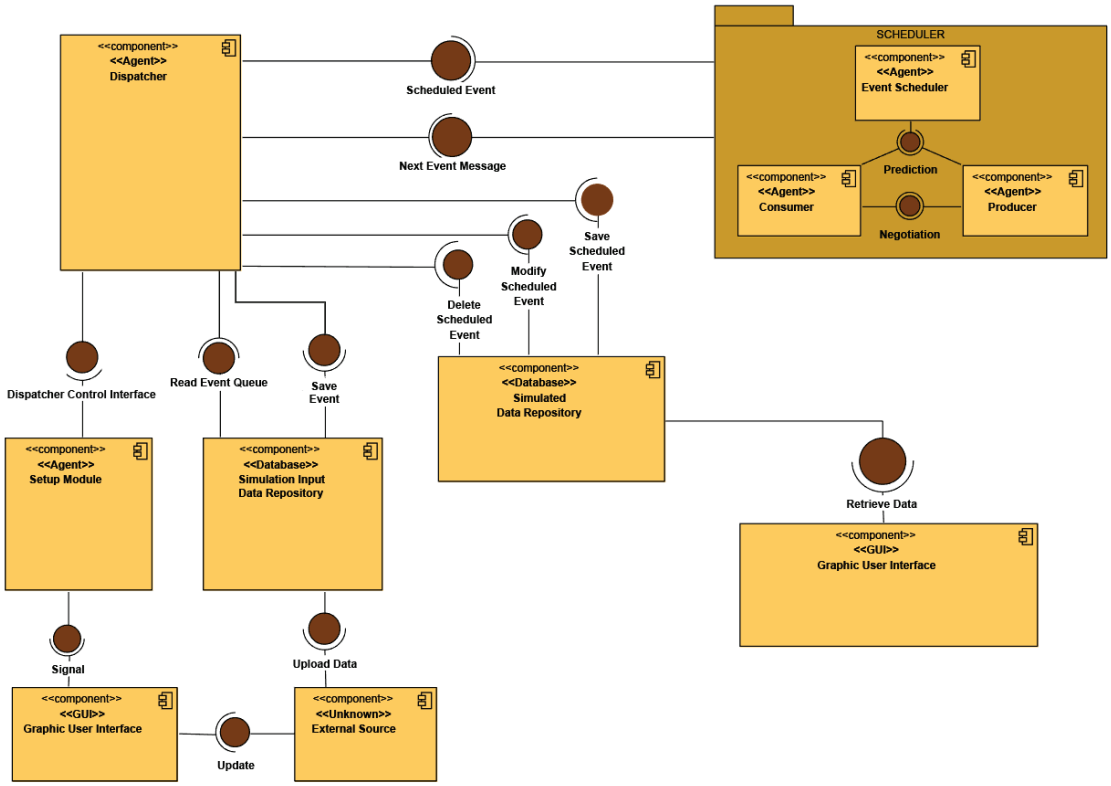

Figure 7 GreenCharge Simulator Architecture Overview

The central idea is to decouple the simulator engine, the **Dispatcher** component, that processes the events and move forward the simulated time, from the components that fills the event queue with the input data feeding the specific simulation session. Such input data, like PV panels characteristics, household's devices, energy loads etc. should be stored in a database: **Simulation Input Data Repository.** The component that has the task of filling the database is called **External Source** and can acquire the input data from different sources:

1. The GUI interface as shown in the figure above;
2. Procedures that generate a synthetic workload according to a specific statistical distribution model;
3. Directly from ad-hoc software wrappers developed in the pilots.

The ability to acquire input data in these different ways is a key aspect of the design, and offers considerable scope in setting up and experimenting with different types of simulations.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

Of course, the component **External Source** that collects the data and fills the **Simulation Input Data Repository**, and the **Dispatcher** component that extracts the events and processes them, should be synchronized and supervised by another component that we call **Setup Module**. Another task assigned to this component is the possibility of stopping, pausing or restarting the simulation session.

Below we will describe the main functions of these components and their interactions, while in the next section we will give some more details on the technologies used to develop them and their implementation.

3.5 GreenCharge Simulator Components

The **Setup Module** is a SPADE3.7 agent that gives the start signal of a simulation session to the **Dispatcher**. Besides, the **Setup Module** catches the control events from the GUI (a user could decide to stop a simulation session pause it or restart the simulation from a new Current Simulation Time) and sends the appropriate messages to the other SPADE Agents involved in the simulation.

For example, the **Setup Module** could stop the simulation, waits for the propagation of some possible changes in the configuration to the event queue, and finally sends a resume signal to the dispatcher with a new restarting time of the simulation. The simulation can start again from the point where it was stopped or from another instant, in case the user wants to go back or carry forward the current simulation time. All these signals are exchanged among the Agents using specific XMPP messages.

In the **Simulation Input Data Repository** are stored all necessary information about entities that are involved in simulation process: energy producers, consumers and prosumers.

There's a python priority queue in which the priority is given by the event's creation time. At each get action from the queue returns the event object with the lowest value of the creation time. Different types of events are foreseen such as, for example, the request for energy from a device (**LOAD**), the update of the energy production of a panel (**UPDATE**), and so on. Most of the events processed by the simulator both for the initial configuration of the scenario and for the scheduling of energy loads will be obtained from two xml files (**neighbourhood.xml** and **load.xml)** that will be described in detail in the next chapter. There will be described the structure of some significant events that feed a simulation session, together with its specific parameters.

The **External Source** has the task of filling the **Simulation Input Data Repository** with inputs coming from different sources of data: predefined configuration files, Graphical User Interface, external data repository or third-party software API. At the end of every changes in the event queue, the component notifies the **Setup Module** that the upload is terminated in order to maintain data consistency and integrity.

In this initial version of the prototype, the **External Source** is a SPADE agent with a single behaviour. Basically, at the start of a simulation session, it reads the neighbourhood structure and all devices in the neighbourhood, then it creates the event and associates them with the previously loaded devices, finally puts them in the priority queue.

In order to understand the methods of the **External Source** agent, the file and directory structure of the GreenCharge simulator is provided (fig.8).

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

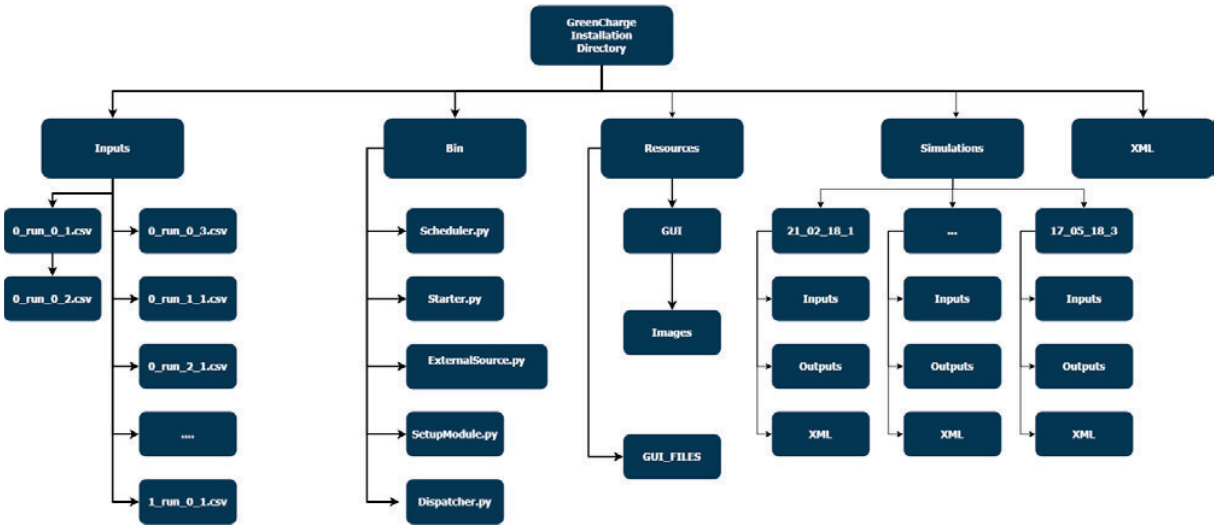

Figure 8 GreenCharge Files and Directory Tree

There are 4 main folders:

- **Inputs:** all the profiles in csv format are saved in the **Inputs** folder. All files are saved according to the format: Houseid_DeviceID_ProfileID.csv;
- **Bin:** the **Bin** folder contains all the python files needed for the simulation, including agents.
- **Resources**: all the files connected to the graphic interface are saved in the **Resources** folder, such as XML files and images present in the GUI.
- **XML**: in the **XML** folder the temporary files **neighbourhood.xml** and **loads.xml** that are built using the **GUI** are saved. When the simulation starts they are copied to the XML subfolder of the related simulation specific folder.
- **Simulations**: in the **Simulations** folder a directory is saved for each simulation performed. The name for each folder is: **Day_Month_Year_SimulationNumberInThatDay**. In each folder are created three subfolders: in **Inputs** the csv profiles related to that simulation are saved; in **XML** are copied the xml files related to that simulation; in the subfolder **Outputs** will find the simulation results.

At the moment **External Source** executes sequentially the following four methods:

- ***makeNewSimulation***():

  this method creates the necessary folders for the single simulation, copies the XML files from the temporary folder to the simulation subfolder and copies the profiles necessary for the simulation to the inputs subfolder.
- ***createDevicesList***():

  this method reads the **neighbourhood.xml** file and creates the list of devices that will be part of the simulation.
- ***createEventList***():

  This method reads the **loads.xml** file and creates a list of events.
- ***uploadInInputRepository***():

  This method updates the priority queue by entering the events not yet processed. It is called at the beginning of the simulation and every time the user makes some changes to the runtime configuration.

The **Dispatcher** is the real Simulation engine since it controls and manages the time's flow of simulation.

Its usual operation cycle can be summarized in the following steps:

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

- reads next event from **Simulation Input Data Repository** relying on a current simulation time (CST);
- deliver a specific message to the **Load Scheduler** (on the basis of the next event to process);
- wait for a reply message;
- store scheduled events in **Simulated Data Repository** that contains the complete trace of the simulated session.

It's SPADE agent with two behaviours. First behaviour waits and handles **Setup Module** XMPP messages about start and stop signal. Second behaviour becomes active when a start signal is notified and it gets and consumes the next event in the priority queue. It generates different XMPP messages (depending on the specific event type) to send to the **Load Scheduler**.

In addition, the Agent **Dispatcher**, on the basis of the scheduling answer for the specific load calculated by the **Load Scheduler**, creates and inserts in the priority queue a **DELETE** load that represents the completion of the specific energy consumption period.
The detailed description of the protocol between the Simulation Engine component (the **Dispatcher**) and the **Load Scheduler,** together with the specification of the exchanged messages payload will be illustrate in the next section.

3.6 Protocol between dispatcher and scheduler

In this section, we illustrate, with the help of some specific use cases (UC), the current implementation of the protocol between the two main components of the Simulation tool: the Dispatcher and the Load Scheduler. The punctual description of this protocol is also essential to allow, if possible, to integrate other optimizers/schedulers in the subsequent versions of the tool.

Since, as said, all interactions between the components of the simulator take place through an exchange of XMPP messages between Actors/Agents first of all we define the Actors involved. Actor means precisely a component with an XMPP identifier (XMPP ID) represented as **username@domain/resource**.

Actors/Agents
TaskManager (Dispatcher):
It's the simulation engine and runs as a XMPP client which sends and receives asynchronous XMPP Messages.
XMPP Server:
Software that allows for the communication channel (e.g. ejabberd, prosody).
ActorManager (Load Scheduler):
This is a component that optimizes the scheduling of the loads and exchanges asynchronous XMPP Messages with the other Actors (TaskManager; Producerss; Consumers) through its own XMPP connection.
Producers:
These are different instances of the same Component which produce energy and exchange asynchronous XMPP Messages with the other Actors (TaskManager; ActorManager; Producers; Consumers) through their own XMPP connections.
Consumers:
These are different instances of the same Component which consume energy and exchange asynchronous XMPP Messages with the other Actors (TaskManager; ActorManager; Producers; Consumers) through their own XMPP connections.

UC1: Initial Connection

The TaskManager and the ActorManager must be connected before to start the simulation by a specific XMPP IDs:
TaskManager: taskmanager@[domain]/taskmanager
ActorManager: taskscheduler@[domain]/actormanager

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

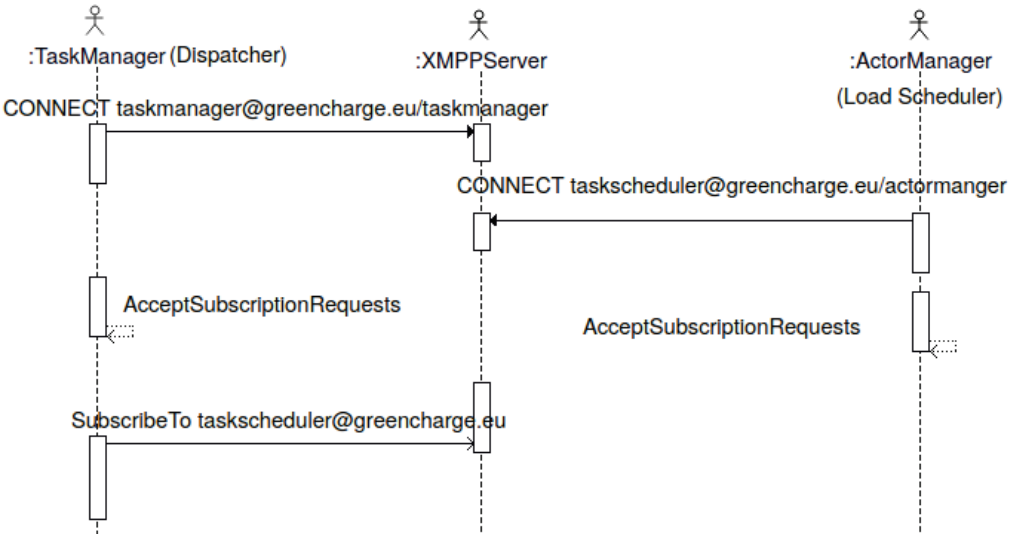

Figure 9 UC1: Initial Connection

UC2: Create Producer

The TaskManager asks the ActorManager to create a new Producer (typically a PV panel) with a specific production profile (described in the file profile.csv). In the current version a producer is identified by a **household_id** and by a **producer_id.**
Producer taskmanager@[domain]/taskmanager/pv_producer[houdehold_id][producer_id]

The message used to create a new Producer appears like this:

Create Producer Message Example:

From: taskmanager@[domain]/taskmanager
To: taskscheduler@[domain]/actormanager
Body: CREATE_PRODUCER PV [0]:[1] profile.csv
Profile.csv is a file that contains the forecasted timeseries of cumulative energy production
E.g:

Time	*Energy (KWh)*
1568115549	1
1568118549	9

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

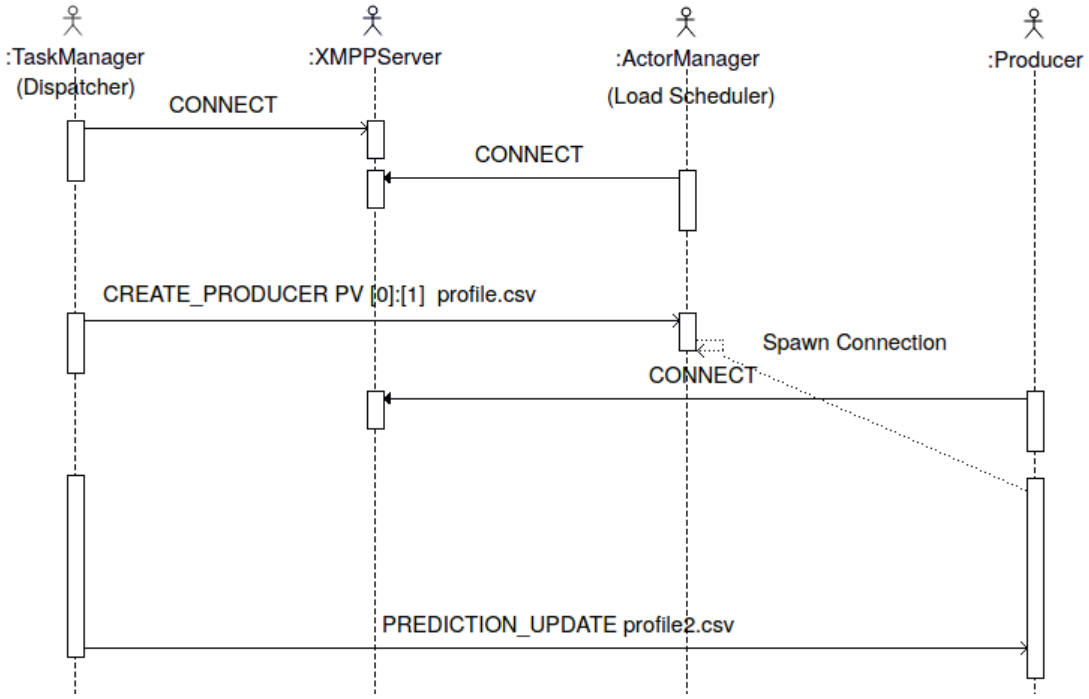

Figure 10 UC2: Create Producer

An update of the predicted production is notified directly to the producer by a message from the TaskManager that appears like this:
Prediction Update Message Example:

From: taskmanager@[domain]/taskmanager
To: taskscheduler@[domain]/ pv_producer[0]:[1]
Body: PREDICTION_UPDATE profile2.csv

UC3: New Load

The schedule of an energy load is composed, normally, by a sequence of three message:

- Load request: asks for a schedule
- Assigned start time: assign the schedule
- Delete Load: discard or confirm the execution of the load

Load Message Example:

From: taskmanager@[domain]/taskmanager
To: taskscheduler@[domain]/actormanager
Body: LOAD ID [0]:[2]:[1] SEQUENCE 1 EST 1568119348 LST 1568129348 PROFILE c1.csv

- The ID is composed of [household_id]:[device_id]:[program_id]. The program_id represents a different consuming profile for the same device (low energy, eco, high energy) and its used by the scheduler to learn for the future the best energy producer for this load

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

- The SEQUENCE is an incremental number that specify that this is the first, second … run for this device
- The EST is the earliest start time for the load (Unix time)
- The LST is the latest start time for the load
- C1.csv contain a time series with cumulative energy for this load (relative time)

C1.csv is a file that contains the timeseries of cumulative energy consumption.
E.g:
[time (sec)] [cumulative_energy (KWh)]
0 0
180 0.01
360 0.03
...

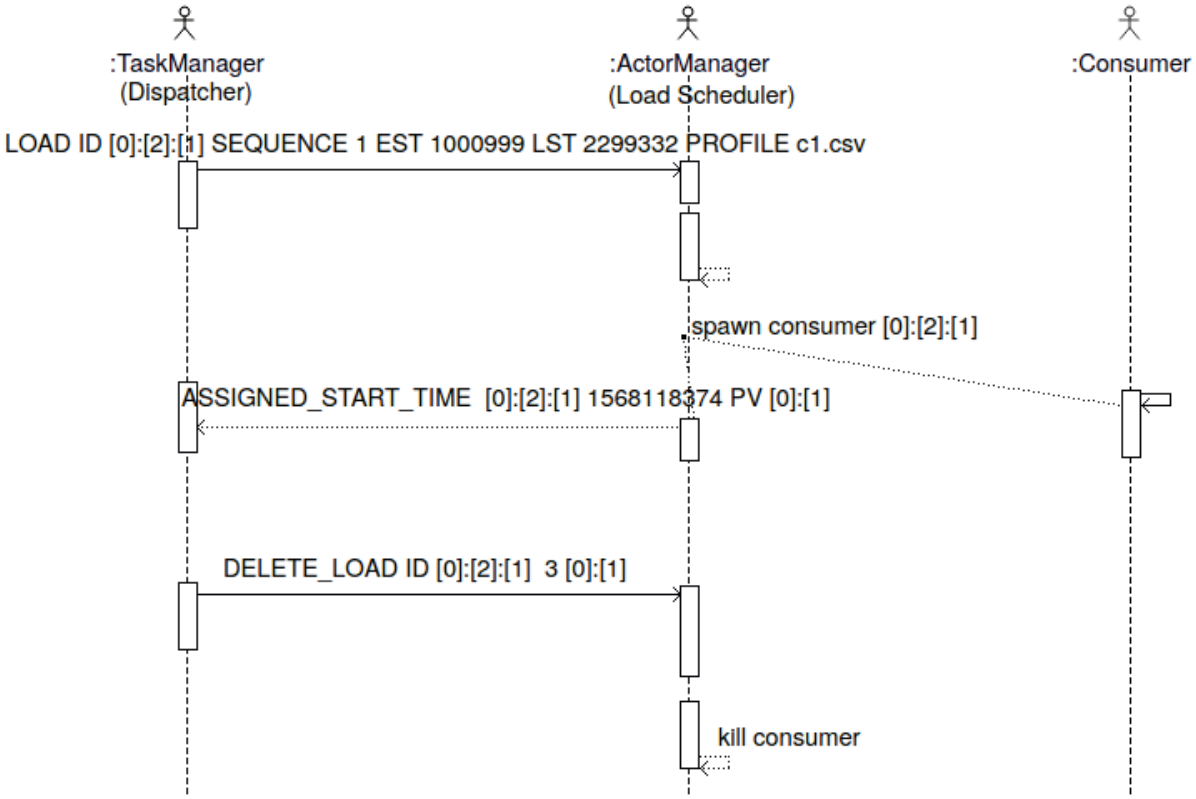

Figure 11 UC3: New Load

Assigned Start Time Message Example:

The ASSIGNED_START_TIME messages include the load id, the scheduled time for the load, and the producer is supposed to provide energy for the load.
E.g.
From: taskmanager@[domain]/actormanager

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

To: taskscheduler@[domain]/taskmanager
Body: ASSIGNED_START_TIME [0]:[2]:[1] 1568115549 PV [0]:[1]

Delete Load Message Example:

This message is used by the TaskManager to discard a planned load or to communicate the completion of the load. In fact, when the load is terminated the TaskManager communicate to the scheduler the actual amount of energy consumed (the producer is assumed equal to the one assigned by the scheduler).

E.g.
From: taskmanager@[domain]/taskmanager
To: taskscheduler@[domain]/actormanager
Body: DELETE_LOAD [0]:[2]:[1] 2.3 [0]:[1]

- [0]:[2]:[1] is the ID of the completed Load
- 2.3 (KWh) is the energy actually consumed
- [0]:[1] is the producer that provided the energy

UC4: Load rescheduled

In this use case there are not new messages. The TaskManager should eventually wait for 0 or more ASSIGNED START TIME messages (see fig.12) when an update prediction is sent. In fact, some already scheduled, not started loads, could be re-scheduled.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

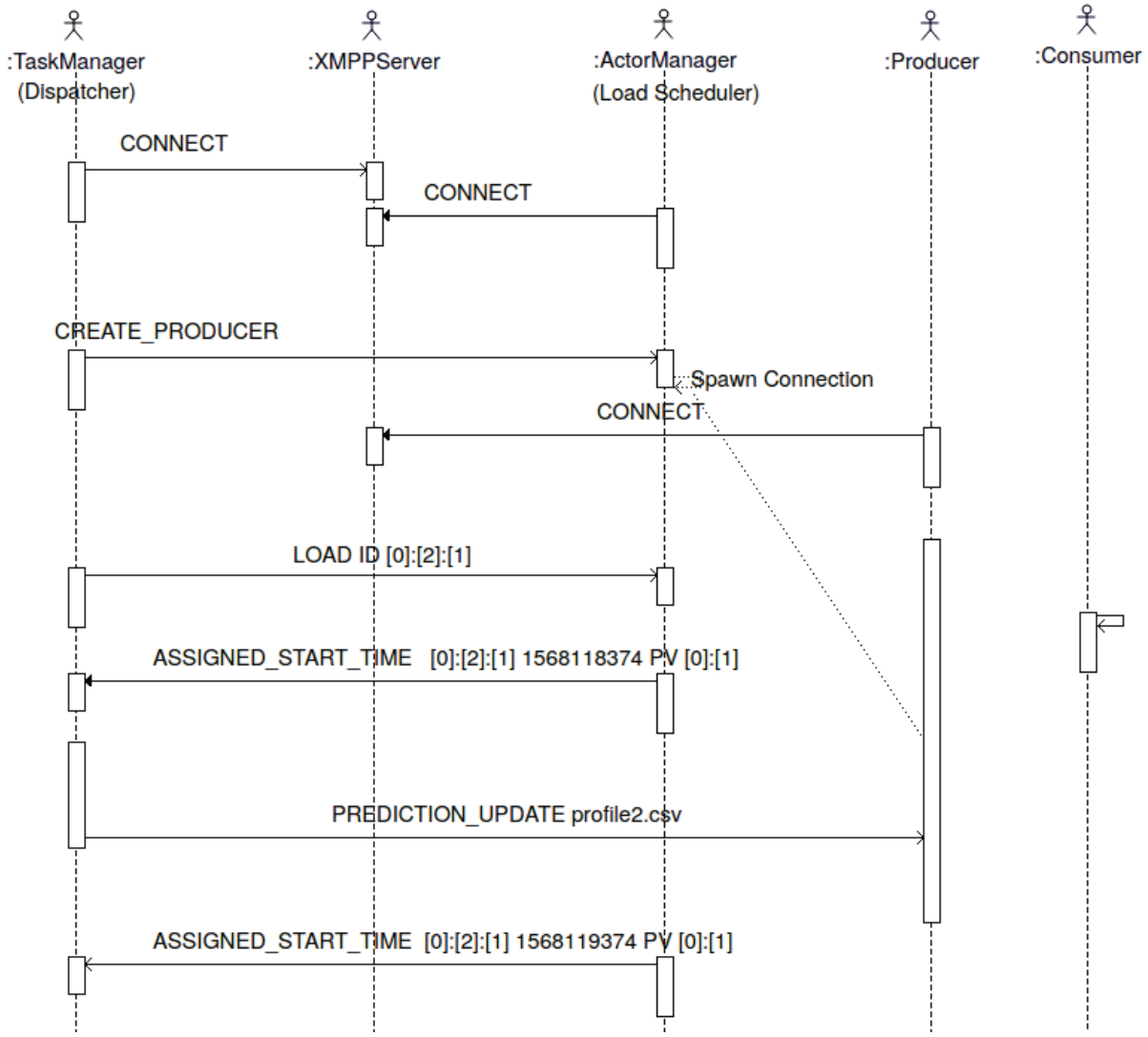

Figure 12 UC4: Load rescheduled

3.7 Main Software Technologies Used

In this section, we illustrate the main technologies used for the development of the main components together with some more details about their implementation.

GreenCharge Simulator is a Multi Agent application based on **SPADE** platform. In fact, the main simulator components described above like **Dispatcher**, **Setup Module**, **External Source** and **Load Scheduler** are all **SPADE** agents.

SPADE (Smart Python multiagent Development Environment) is a multi-agent systems platform written in Python and based on instant messaging (XMPP) (https://pypi.org/project/spade/#description). SPADE offers many features that ease the construction of MAS, such as an existing communication channel, the concepts of

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

users (agents) and servers (platforms) and an extensible communication protocol based on XML, just like FIPA-ACL. The SPADE Agent Library is a module for the Python for developing SPADE agents: it is a collection of classes, functions and tools for creating new SPADE. The Agent Model is basically composed of a connection mechanism to the platform, a message dispatcher, and a set of different behaviours that the dispatcher feeds with messages. Every agent needs an identifier called Jabber ID and a valid password to establish a connection with the XMPP server. Communications in SPADE are handled internally by means of the XMPP protocol. This protocol has a mechanism to register and authenticate users against an XMPP server. Each SPADE agent has an internal message dispatcher component. This message dispatcher acts as a mailman: when a message for an agent arrives, it places it in the correct "mailbox" associated with a behaviour. An agent can run several behaviours simultaneously. A behaviour is a task that an agent can execute using predefined repeating patterns. A behaviour has a message template attached to it and the message dispatcher uses this template to determine which behaviour the message is for, by matching.

The graphical interface, described with more details in the next chapter, was created through the **QT5 python library** which provides a wide range of Widgets and layouts with which the GreenCharge GUI was built

Finally, we used the **Docker** containers for application deployment. This technology will greatly simplify the development and integration of multiple components, providing a reliable and secure environment. Furthermore, being highly portable, they make the software work on any type of platform.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

4 Configuration of the first scenario of use

4.1 Configuration of the simulation environment with the involved energy actors

In this section, we describe, as first scenario of use to test the initial version of the prototype, a simplified simulation environment testbed related to the **Scenario 1**: **Charge planning and booking**, described in section 2.2. In this scenario are included, the first three Use Cases foreseen in the Oslo pilot, obviously with increasing levels of complexity. In the following we summarizes for each of the Oslo Use Cases a short description of the scenario together with the related challenges when implementing them in the pilot:

Oslo UC#1 Normal charging in the garage

Description:

- Individually owned parking places – typically overnight charging
- App for input of next time of use (and state-of-charge)
- System schedules charging based on other electricity consumers (other EVs, heated driveway) and on availability of solar power (on rooftop) and state of charge of stationary battery

Main Challenges for pilot implementation:

- User need to buy own charging point + establishment of business agreement
- User willing to provide accurate information to **Charge Management System** (CMS)
- Integration of CMS and **Neighbourhood Energy Management System** (NEMS)
- Procurement of solar power and stationary battery

Oslo UC#2 Long-term parking (V2G possibilities) in the garage

Description:

- Extension of UC#1 for EVs parked for longer

Main Challenges for pilot implementation:

- Availability of vehicles with V2G
- Users willing to provide their in-vehicle battery (rewards)
- Users willing to provide necessary input to the system (time of departure)

Oslo UC#3 Drop-in charging

Description:

- Making existing outdoor chargers available to externals visitors or employees at nearby school
- 4 semi-fast chargers – currently used by residents through simple booking system (Google docs spreadsheet)
- Access will be through app

Main Challenges for pilot implementation:

- Inform and attract potential users – sufficient usage required to ensure return of investment
- Establishment of business agreement
- Motivate download of app for payment

In order to simulate these scenarios, we, first of all, concentrated on the changes to be made to the configuration of the **Neighbourhood Energy Management System** so that, in addition with the management of the loads generated in the households of the neighbourhood, should take in account also the additional power requests coming from the nearby **Charge Management System**.

In order to quickly test the first extensions of the tool, we developed a simplified stand-alone version of the simulator prototype using a **Load Scheduler** Agent that at the moment simply replies to the Load requests messages coming from the **Dispatcher** Agent without any optimizations of the neighbourhood energy.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

The configuration model of a typical simulation session requires both a static configuration and a dynamic configuration.

The static configuration (the **neighbourhood.xml** file – see section 4.1.1) includes the definition of the different energy actors: producers like PVs, consumers like appliances, prosumers like batteries that make up the neighbourhood system. To each of them we could associate one or more different profiles of energy production or of energy consumption. First of all, to simulate the new GreenCharge scenarios, we, surely, need to include in the configuration of the system one or more EV chargers that can be seen as sockets or charging points of a new Energy Actor: the **Charging Station**. Each socket of the **Charging Station** can behave at different times both as consumer when needs energy to charge a plugged e-car plug and as a producer if the V2G technology is enabled and authorized by the user.

The dynamic configuration (the **load.xml** file – see section 4.1.2) otherwise describe the specific loads of the day that should be managed and optimized by the **Load Scheduler** of the simulator. Each load is characterized by its constraints and its flexibility. In this first scenario, for example, we need to characterize as a load each of EV charging requests (or bookings) accepted by the charging station. As described in the precedent chapter the dynamic input data that feed the simulation session (in particular the **LOAD** events) can be add using the **Graphical User Interface** of the tool, for example to simulate a specific scenario with a specific artificial distribution of the loads. The other way, that will be the most useful for evaluating the technologies and policies tested in the Pilots, will require the need of querying external sources through the use of appropriate APIs developed in the Pilots' sites.

In this first version of the prototype, in order to easily test the functionalities of the tool, we will limit ourselves to use a new developed GUI, described in the next section, to configurate and fill a simulation session with data input.

4.1.1 Neighbourhood Configuration

Fields:

- **Neighbourhood**: represents the entire neighbourhood to simulate. The "id" is a number that identifies the neighbourhood. It contains all the houses that you want to simulate.
    - **House**: represents the household. The "id" is a number that identifies the house within the neighbourhood. It contains a single user in the current version.
        - **User**: Currently, for each house there is only one user but for possible future user we prefer to add it. The user contains all the devices of the house**.**
            - **Device**: represents a consumer or a producer device. It is described by:
                - **id**: it identifies the device within the house
                - **name**: a mnemonic name
                - **type**: it can be consumer or producer.
    - **Charging Station**: represents the charging station associated to the neighbourhood. The "id" is a number that identifies the charging station within the neighbourhood.
        - **User**:
            - **Device**: represents a charging point:
                - **id**: it identifies the socket within the station
                - **name**: a mnemonic name
                - **type**: Prosumer

Example File:

```
<neighbourhood id="25" name="GreenCharge_Neighbourhood">
      <house id="7">
```

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

```
            <user id="1">
                  <device>
                        <id>1</id>
                        <name>Solar Panel</name>
                        <type>producer</type>
                  </device>
                  <device>
                        <id>4</id>
                        <name>Washing Machine</name>
                        <type>consumer</type>
                  </device>
            </user>
      </house>
      <cstation id="10">
            <user id="1">
                  <device>
                        <id>3</id>
                        <name>Standard socket</name>
                        <type>prosumer</type>
                  </device>
                  <device>
                        <id>4</id>
                        <name>Fast socket</name>
                        <type>prosumer</type>
                  </device>
                  <device>
                        <id>5</id>
                        <name>Fiat 500</name>
                        <type>Prosumer</type>
                  </device>
                  <device>
                        <id>6</id>
                        <name>BMW Q5</name>
                        <type>Prosumer</type>
                  </device>

            </user>
      </cstation>
</neighbourhood>
```

4.1.2 Loads Configuration

This file (**load.xml**) should contain all the energy requests that must be dynamically satisfied and optimized by the scheduler. These requests arrive at different times during the simulation session and are characterized by a submission time (**Creation Time**), a specific profile of energy required (**Profile**) and a fixed degree of flexibility (**EST** and **LFT**). All these load requests are managed as events by the simulator and are ordered on the basis of the **Creation Time** field.

The structure of each load event should be the following:

- **Device:** it indicates the device in which the load event refers to;
- **EST:** Earliest Start Time, it's a timestamp;
- **LFT:** Latest Finish Time, it's a timestamp;
- **Creation_Time:** event creation timestamp;

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

- **Profile:** describes the operating model of the device and its associated to a specific energy request file (e.g. different ways of using a washing machine consume differently).

This is the structure and the parameters used by the Load Scheduler in CoSSMic to optimize the scheduling of the energy and maximize the self-consumption rate of the neighbourhood. The initial idea is to use the same structure of the event also to characterize the energy demand (LOAD) of an e-car, modelled as a cumulative energy timeseries (Profile), submitted at plugin-time, with an Earliest Start Time (EST), that in this case is equal to the **Planned Arrival Time** and a Latest Finish Time (LFT) equal to the **Planned Departure Time**. In summary, for an electric car that needs to be recharged at a charging station most likely we could reasonably rely on information like this:

- EV model (likely to be specific constraints on charging process associated with the different models). This must be described in templates (or similar) per EV model
- Planned Arrival Time (EST)
- Planned Departure Time (LFT)
- Initial Status of Charge
- Target Status of Charge
- Actual Arrival Time
- Actual Departure Time
- V2G enabled with any limitations or constraints, to serve as standby capacity in case of unplanned need to use the car.

The objective is therefore with a pre-processing phase to generate from this initial information one or more events with a structure similar to the event specified above, probably with an energy profile dynamically calculated through a simple analytical model of the charging process.

In this first version of the prototype, associated to a charging request from an e-car we will find in the file loads.xml these parameters:

- Planned Arrival Time
- Planned Departure Time
- E-Car Model (type of battery)
- Initial Status of Charge
- Target Status of Charge
- V2G enabled

Example File

```
<neighbourhood id="25" executionId="1">
      <house id="7">
            <user id="1">
                  <device>
                        <id>4</id>
                        <est>1462519108</est>
                        <lft>1462522707</lft>
                        <creation time>1462518108</creation time>
                        <profile>washing79.csv</profile>
                  </device>
                  <device> // same device different load
                        <id>4</id>
                        <est>1462526308</est>
```

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

```
                    <lft>1462529907</lft>
                    <creation time>1462518108</creation time>
                    <profile>washing80.csv</profile>
                    </device>

          </user>
     </house>
     <cstation id="10">
          <user id="1">
               <device>
                    <id>5</id> // ECAR
                    <est>1462526408</estt>
                    <lft>1462529980</lft>
                    <creation time>1462518108</creation time>
                    <profile>Fiat500.csv</profile>
                    <ecarModel>Fiat500</ecarModel>
                    <initialSOC>47%</initialSOC>
                    <targetSOC>90%</targetSOC>
                    <v2g>1</v2g>
               </device>
               <device>
                    <id>6</id> // ECAR
                    <est>1462526520</est>
                    <lft>1462521000</lft>
                    <creation time>1462518108</creation time>
                    <profile>BMWQ5.csv</profile>
                    <ecarModel>BMWQ5</ecarModel>
                    <initialSOC>23%</initialSOC>
                    <targetSOC>100%</targetSOC>
                    <v2g>0</v2g>
               </device>

          </user>
     </cstation>
</neighbourhood>
```

4.2 Set & Start Simulation

In this section, we describe in detail the steps to configure the simulation environment using the developed GUI. The configuration should include the different involved energy actors: Producers, Consumers, Prosumers. Each of them is characterized by a specific energy profile that, in some cases (i.e. the PVs), could be dynamically updated whenever the weather forecast is updated.

The final goal of the configuration phase is, regardless of where the data comes from (from the GUI or from the pilots) to produce the files **neighbourhood.xml** and **loads.xml** described above. At this point, to start the simulation, all that remains is to set the day and the starting and ending time of the simulation session.

In the next sections we describe the main features of the GUI and through a complete Use Case diagram how to, from scratch, configure and start a simulation session.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

4.3 Graphic User Interface

The GreenCharge Simulator Graphic User Interface aims to be simple, smart and intuitive. The top menu in the first version was designed to contain 4 tabs: **Settings**, **Control Panel**; **Show Results** and **GreenCharge Simulation Tool Info** (fig.13).

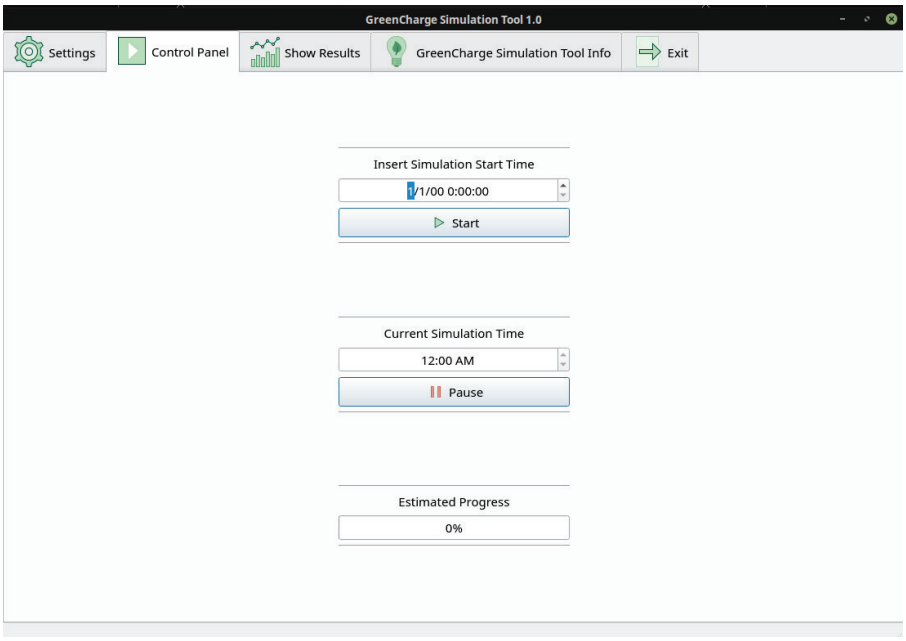

Figure 13 Greencharge Control Panel

4.3.1 Settings

The **Settings** panel is divided into two fields. The first deals with the static and topological configuration of the neighbourhood and the second with the dynamic configuration of the simulation session where we can upload the devices together with their energy demand. In the next section we illustrate more accurately these two distinct logical steps of the configuration phase.

4.3.2 Control Panel

The **Control Panel** represents a kind of dashboard of the tool to set and overlook a simulation session (fig. 14).

In this panel, after completing the configuration phase, we can fix the day and the starting time of the simulation. Pressing the start button activates all the simulation agents and starts the scheduling process. As the simulation progresses over time, the actual simulation time is updated, allowing the user to keep track of the simulation evolution. Furthermore, the user can pause and change the current simulation time, possibly skipping a few hours or going back in time. This feature could be very useful since the user could dynamically insert other devices or changing some parameters into the configuration in order to test, on the fly, new simulation conditions.

Finally, a progress bar indicates to the user the percentage of completion of the simulation.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

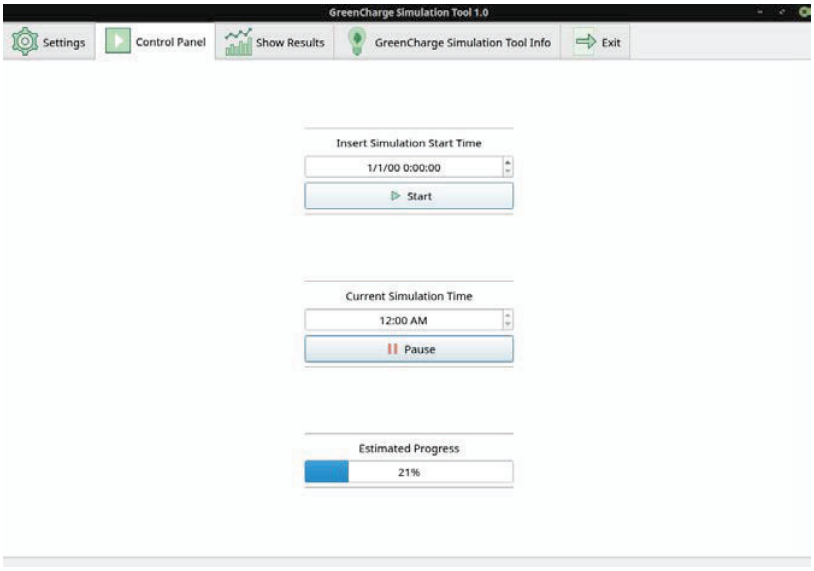

Figure 14 Simulator Dashboard

4.3.3 Show Results

In the **Show Results** panel, as expected, the user will be able to enquire the database of the output data produced by the current and past simulation sessions, select and visualizes different graphical views of results. In this panel it should also be possible to calculate and display some of the KPIs defined to quantify the impact of some measures tested in the Pilots

This part of the simulator has not yet been implemented but it has been designed the database of the output results which will be briefly illustrated in section 4.5.

4.3.4 GreenCharge Simulation Tool Info

In this panel we will found information about the project, the partners involved and the contact references.

4.4 The Configuration phase using the tool GUI

In the figure 15 is presented a complete Use Case Diagram showing how to configure and start a simulation session using the graphical features of the **Settings** panel of the developed GUI.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

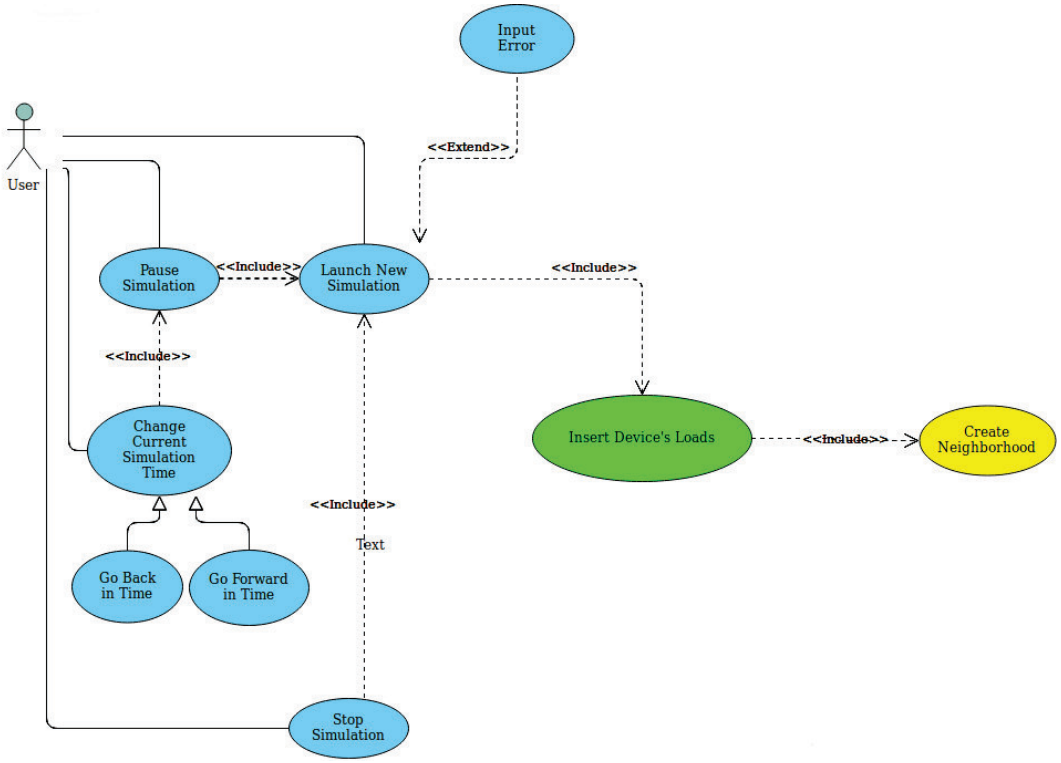

Figure 15 Setting Use Case

A user can mainly start a new simulation, pause it, stop it or change the current simulation time if it has previously paused the simulation.

However, to launch a new simulation, it is necessary to complete the configuration phase. This phase consists of two steps explained by other two use case diagrams described below: the first defines the neighbourhood composition in terms of charging stations, houses, together with the associated PV panels, batteries and devices (Create Neighbourhood yellow box in fig.16); in the second step it's needed to set the energy requests (loads) associated to the specific devices together with the constrains fixed by the user (Insert Device's Loads green box).

4.4.1 Configuration of the Neighbourhood

Th first step of the simulation configuration can be done in two ways (see the Use Case Diagram in fig. 16):

a) A user can select their own XML configuration files (neighbourhood.xml) simply by uploading an existing xml file. Once the configuration files have been selected and uploaded, the user can decide whether to modify it or leave it as it is. When a neighbourhood configuration is uploaded, also devices associated with the single neighbourhood element are loaded, so that they can be afterwards modified, added or deleted. In particular, using the GUI, it's possible to add a device already present or to create a new device from scratch.

b) A user can decide to build a neighbourhood from scratch using the features of the GUI.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

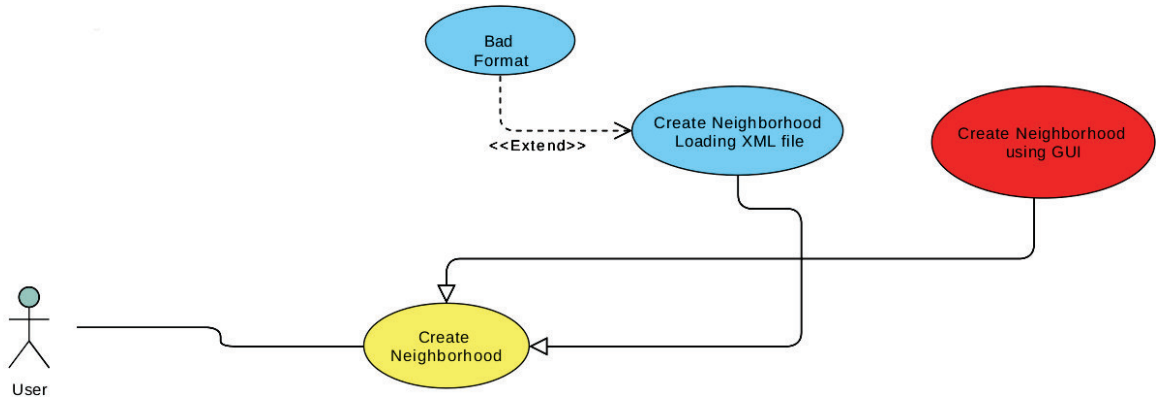

Figure 16 Create Neighbourhood Use Case

As said above, to associate a device or a solar panel to a house or, even, an e-car to a charging station, the user can select one of the devices already defined or creates a new one (see fig. 17). Once created, the new device is added to the repository so that it can be reused in successive simulations. In this first version of the prototype it's possible to create three kinds of devices: producers like PV panel; consumers (passive devices) like dishwashers or washing machines, or prosumer like batteries or e-car models. For each of them at least the following parameters must be specified:

- Name
- Type (Consumer – Producer - Prosumer)

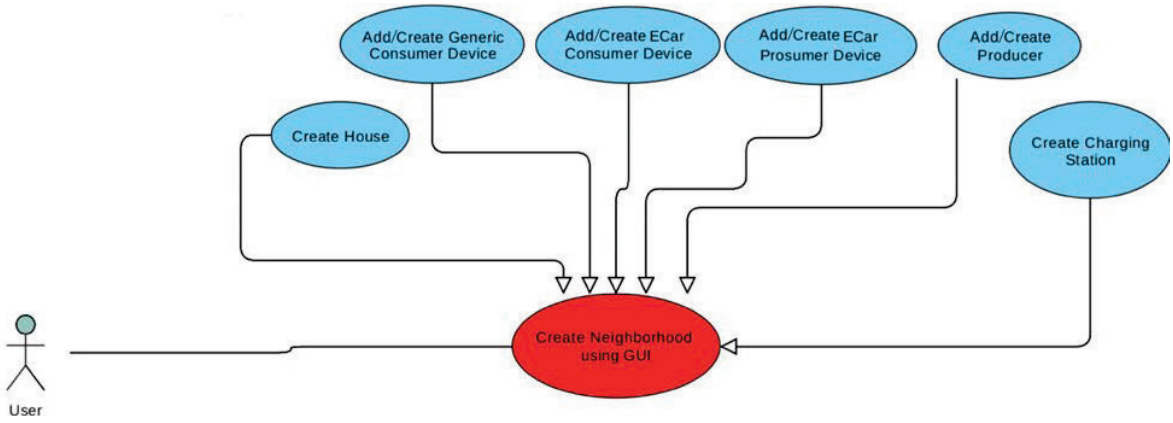

Figure 17 Create Neighbourhood using GUI

Furthermore, it is possible to modify a device present in the repository.

4.4.2 Definition of the Neighbourhood Loads

The **second step** can be performed in two ways (see the Use Case Diagram in fig. 18):

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

a) A user can select their own XML configuration files (loads.xml) simply by selecting an existing xml file. As in the previous step, once the configuration file has been selected, the user can decide whether to modify the single loads or leave them as they are.

b) Alternatively, the user can decide to manually add load devices to neighbourhood items. In this case it will be possible for the user to insert a load, associated to a specific device present in the configuration, by specifying the following parameters

for a passive device like:

- Earliest Start Time
- Last Finish Time
- Creation Time
- Consumption Profile

for an e-car:

- Planned Arrival Time
- Planned Departure Time
- E-Car Model
- Initial Status of Charge
- Target Status of Charge
- V2G

The other necessary information for scheduling, such as maximum battery power, if e-car is enabled for fast charging, etc. are static information obtained directly from the e-car model.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

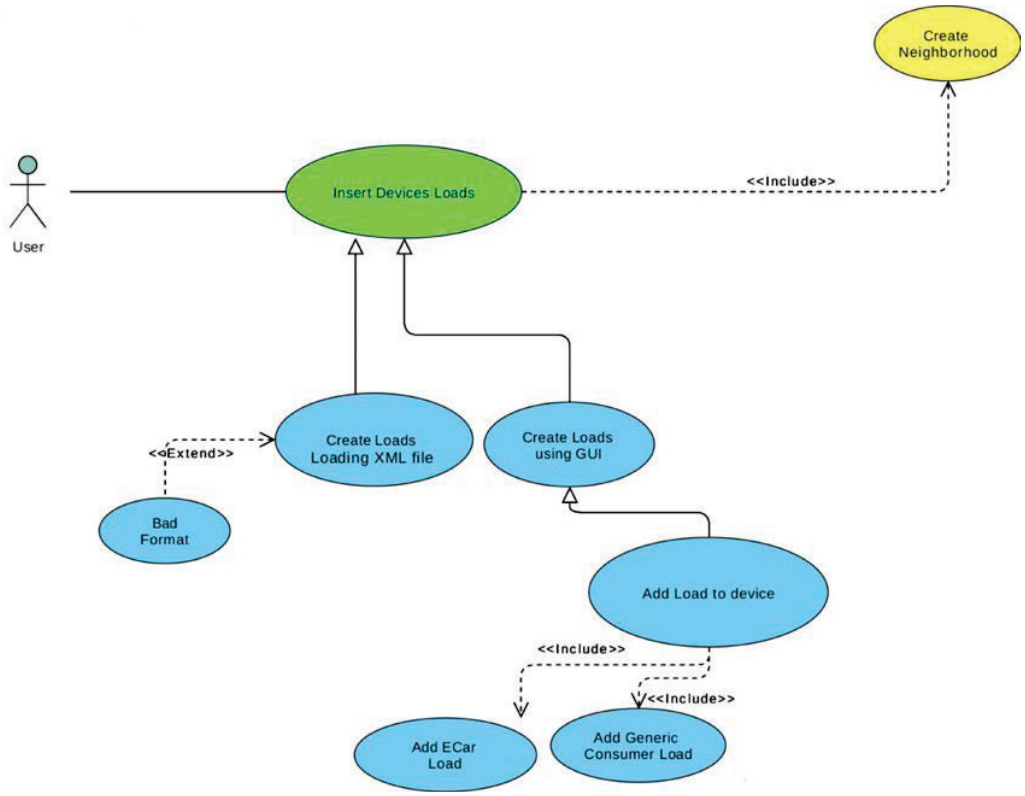

Figure 18 Insert Devices Loads Use Case

4.5 Output Data Model

In the fig. 19 is showed a first sketch of the ER diagram modelling the output data that should be produced by the simulation sessions. The outgoing data model has been designed in order to save in a structured way the essential information produced by the simulator. In this way it will be possible to carry out (even with other specific tools) different specific searches on the output database in order to analyse and generate graphs that will allow us to highlight and measure the aspects that will be essential for the evaluation phase of the project.

The following describes the Tables showed in the ER model, each of them associated to a logical component.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

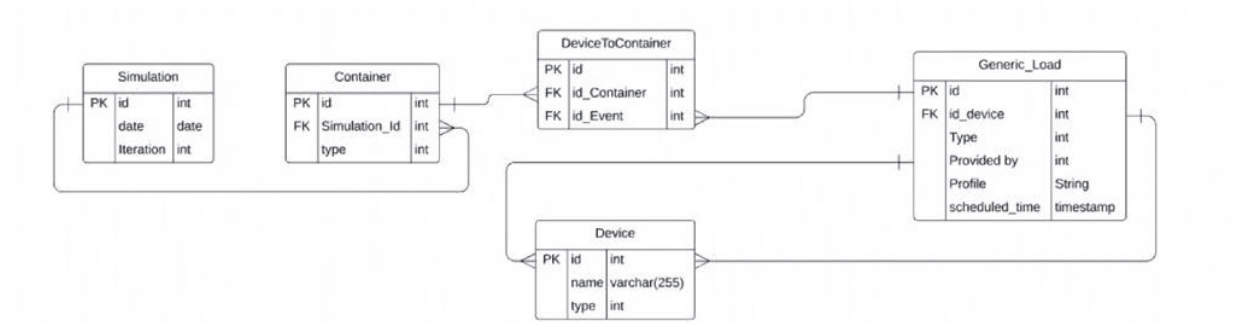

Figure 19 Output data model

4.5.1 Simulation

The simulation table will serve to distinguish the various simulation sessions. It contains information about the Date and the iteration. The date is useful when you want to do analytics based on a period of the year (e.g. evaluate the energy consumption of winter days compared to summer ones) but also to do analytics on a specific day of the year, to show how self-consumption has improved from the previous year to the next.

The iteration field, on the other hand, could be useful to discriminate the different simulation sessions made for the same day under the same static and dynamic conditions (obviously this makes sense if some component of the simulator, e.g. the scheduler, introduces some non-deterministic aspects).

4.5.2 Container

The "Container" table contains the houses and the charging stations that can be inserted into the configuration of a specific simulation session.

The foreign key "simulation_id" binds the house/charging station to the simulation session, while the type field is used to discriminate the type (if it is a house or a charging station). The latest information can be useful when you need to do analytics only on houses or only on charging stations.

4.5.3 Device To Container

This table simply serves to connect the devices events (basically loads) to the various containers through a many-to-many relationship.

4.5.4 Device

The devices table contains all the different devices, characterized by two attributes. The attribute **name** is simply the name of the device while the attribute **type** can take values "0" or "1" if the device is, respectively, a generic device or an ecar.

4.5.5 Generic Load

This table contains all the device events (loads) recorded in a simulation session. The event is characterized by various attributes:

- **id_device**: is used to link the event to a specific device;
- **Type**: identifies the type of event: "0" if it is a Consumer load, "1" if it is an energy production event by a Producer device. (N.B. the single event cannot be of prosumer type, a prosumer device will have more events associated that will be, in turn, of consumer or producer type).
- **Provided_by**: indicates, only for Consumer load events, which was the id of the producer device that supplied the energy.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

- **Profile**: indicates the path of the file that contains the consumption/production profile of the specific event.
- **scheduled_time**: is the timestamp provided by the scheduler and indicates the time set by the scheduler to start the device.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

5 Further work

Based on the requirements analysis and design activities described in this document, a first version of the prototype simulation and visualisation tools has been developed that can be used on simple scenarios. By applying these tools to some basic scenarios of the project, we will be able to identify detailed adaptations and extensions that will be needed, and plan the work to modify the tool and integrate new modules.

The following functional extensions/integrations are already planned:

- a more detailed analytical model of the charging/discharging of a battery;
- the characterization of the consuming energy profile of the heaters/coolers;
- the definition of new components to allow the integration in the simulator of different schedulers with different optimization algorithms in order to carry out comparative evaluations of some relevant KPIs;
- the inclusion of new features into the graphical interface to support the replication and/or the scaling up of simulated scenarios or to allow repeated simulations of exact same scenario to neutralize stochastic variation of the scheduling algorithm;
- the introduction of new functionalities to allow the generation of simulation data input coming from different external sources (e.g. data collected in the Pilots).

Finally, the simulator will have to be integrated with search and graphic visualization tools for the analysis of the output data generated by the simulation sessions, in order to, adequately support the evaluation phase of the innovation strategies experimented in the project.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

6 References

[1] A. Amato, R. Aversa, B. DiMartino, M. Scialdone, and S. Venticinque, “A simulation approach for the optimization of solar powered smart micro-grids,” in Complex, Intelligent, and Software Intensive Systems, L. Barolli and O. Terzo, Eds. Cham: Springer International Publishing, 2018, pp. 844–853.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

Members of the GreenCharge consortium

SINTEF	SINTEF AS (SINTEF) NO-7465 Trondheim Norway www.sintef.com	**Project Coordinator:** Joe Gorman Joe.Gorman@sintef.no **Technical Manager:** Shanshan Jiang Shanshan.Jiang@sintef.no
eSmart Systems	eSmart Systems AS (ESMART) NO-1783 Halden Norway www.esmartsystems.com	**Contact:** Frida Sund frida.sund@esmartsystems.com
HUBJECT	Hubject GmbH (HUBJ) DE-10829 Berlin Germany www.hubject.com	**Innovation Manager:** Sonja Pajkovska sonja.pajkovska@hubject.com
eurecat Centre Tecnològic de Catalunya	Fundacio Eurecat (EUT) ES-08290 Barcelona Spain www.eurecat.org	**Contact:** Regina Enrich regina.enrich@eurecat.org
ATLANTIS TRACKING YOUR WORLD	Atlantis IT S.L.U. (ATLAN) ES-08007 Barcelona Spain www.atlantis-technology.com	**Contact:** Ricard Soler rsoler@atlantis-technology.com
enchufing	Millot Energy Solutions SL (ENCH) ES-08223 Terassa Spain www.millorbattery.com	**Contact:** Gerard Barris gbarris@enchufing.com
motit www.motitworld.com	Motit World SL (MOTIT) ES-28037 Madrid Spain www.motitworld.com	**Contact:** Valentin Porta valentin.porta@goinggreen.es
Freie Hansestadt Bremen	Freie Hansestadt Bremen (BREMEN) DE-28195 Bremen Germany	**Contact:** Michael Glotz-Richter michael.glotz-richter@umwelt.bremen.de

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.

ZET GmbH (MOVA)
DE-28209 Bremen
Germany
www.zet.technology

Contact: Nils Jakubowski
nils@zet.technology

Personal Mobility Center Nordwest eG (PMC)
DE-28359 Bremen
Germany
www.pmc-nordwest.de

Contact: Bernd Günther
b.guenther@pmc-nordwest.de

Oslo kommune (OSLO)
NO-0037 Oslo
Norway
www.oslo.kommune.no

Contact: Sture Portvik
sture.portvik@bym.oslo.kommune.no

Fortum OYJ (FORTUM)
FI-02150 Espoo
Finland
www.fortum.com

Contact: Jan Ihle
jan.haugen@fortum.com

PNO Consultants BV (PNO)
NL.2289 DC Rijswijk
Netherlands
www.pnoconsultants.com

Contact: Arno Schoevaars
arno.schoevaars@pnoconsultants.com

Universita Deglo Studi Della Campania Luigi Vanvitelli (SUN)
IT-81100 Caserta
Italy
www.unicampania.it

Contact: Salvatore Venticinque
salvatore.venticinque@unina2.it

University of Oslo (UiO)
NO-0313 Oslo
Norway
www.uio.no

Contact: Geir Horn
geir.horn@mn.uio.no

ICLEI European Secretariat GmbH (ICLEI)
DE-79098 Freiburg
Germany
www.iclei-europe.org

Contact: Stefan Kuhn
stefan.kuhn@iclei.org

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 769016.