Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Department of the Environment,
Transport, Energy and Communication DETEC

Swiss Federal Office of Energy SFOE
Energy Research and Cleantech

Final report

Market designs and control models for the activation of flexibility at the local level

Summary of results of the NEMoGrid, MuLDeR and LIC projects

NEMoGrid

MULDER

LIC LUGAGGIA
INNOVATION COMMUNITY

University of Applied Sciences and Arts
of Southern Switzerland

SUPSI

Date: 01/12/2021

Place: Bern

Publisher:
Swiss Federal Office of Energy SFOE
Research Programme Grids
CH-3003 Bern
www.bfe.admin.ch
energieforschung@bfe.admin.ch

Author:
Vasco Medici, ISAAC, SUPSI (vasco.medici@supsi.ch)
Lorenzo Nespoli, ISAAC, SUPSI (lorenzo.nespoli@supsi.ch)
Davide Strepparava, ISAAC, SUPSI (davide.strepparava@supsi.ch)
Federico Rosato, ISAAC, SUPSI (federico.rosato@supsi.ch)
Davide Rivola, ISAAC, SUPSI (davide.rivola@supsi.ch)
Barbara Antonioli Mantegazzini, DEASS, SUPSI (barbara.antonioli@supsi.ch)
Roman Rudel, ISAAC, SUPSI (roman.rudel@supsi.ch)

SFOE head of domain: Dr. Michael Moser, michael.moser@bfe.admin.ch
SFOE programme manager: Dr. Michael Moser, michael.moser@bfe.admin.ch
SFOE contract number: SI/501705, SI/501499, SI/501840

The author of this report bears the entire responsibility for the content and for the conclusions drawn therefrom.

Summary

This report summarizes the work done in two research projects, the NEMoGrid and the MuLDeR. In these projects, we developed solutions for energy communities that allow them to maximize their self-consumption while at the same time taking into account the needs of the distribution grid. The NEMoGrid project focused more on developing market designs that would favor the integration of renewables at a local level, mainly concentrating on designing blockchain-based markets for energy communities. The MuLDeR project focused on developing distributed control strategies for the optimal actuation of flexible resources within such markets and their extension to a multi-level optimization setting. The solutions designed in the research projects are being applied in the currently running pilot and demonstrator project LIC, of which we report preliminary results. Within the LIC project the solutions developed in NEMoGrid and MuLDeR are compared with an alternative centralized solution developed by SUPSI in the Innosuisse project OptiFlex. Simulation research coupled with real-world testing allowed us to evaluate the effectiveness of the various centralized and decentralized control strategies and the attractiveness of the different market designs. This report summarises our results, discusses our findings, lessons learned, and provides suggestions for designing flexible energy communities based on local markets, which can help foster the integration of distributed renewable energy resources in the distribution grid, as called for in the Swiss Energy Strategy 2050.

Sommario

Questo rapporto riassume il lavoro svolto in due progetti di ricerca, il progetto NEMoGrid e il progetto MuLDeR. In questi progetti, abbiamo sviluppato soluzioni per le comunità energetiche che permettono loro di massimizzare l’autoconsumo, tenendo conto allo stesso tempo delle esigenze della rete di distribuzione. Il progetto NEMoGrid si è concentrato maggiormente sullo sviluppo di progetti di mercato che favorissero l’integrazione delle energie rinnovabili a livello locale, concentrandosi principalmente sulla progettazione di mercati basati su blockchain per le comunità energetiche. Il progetto MuLDeR si è concentrato sullo sviluppo di strategie di controllo distribuito per l’attuazione ottimale delle risorse flessibili all’interno di tali mercati e la loro estensione a un framework di ottimizzazione multilivello. Le soluzioni sviluppate nei progetti di ricerca vengono applicate nel progetto pilota e dimostrativo LIC attualmente in corso, e del quale riportiamo i risultati preliminari. All’interno del progetto LIC le soluzioni sviluppati in NEMoGrid e MuLDeR vengono confrontate con una soluzione alternativa centralizzata sviluppata dalla SUPSI nel progetto Innosuisse OptiFlex. La ricerca in simulazione abbinata ai test nel mondo reale ci hanno permesso di valutare l’efficacia delle varie strategie di controllo centralizzate e decentralizzate e l’attrattiva dei diversi design di mercato. Questo rapporto riassume i nostri risultati, discute le nostre scoperte e gli insegnamenti tratti, e fornisce suggerimenti per progettare comunità energetiche flessibili basate su mercati locali, che possano effettivamente aiutare a promuovere l’integrazione delle risorse energetiche rinnovabili distribuite nella rete di distribuzione, come richiesto dalla Strategia Energetica Svizzera 2050.

Contents

List of abbreviations

ADMM	Alternating Direction Method of Multipliers
AMM	Automated Market Making
AT	Auditable Tariff app
BAU	Business As Usual
D-OPF	Distributed Optimal Power Flow
DSM	Demand Side Management
DSO	Distribution System Operator
EC	Energy Community
EPNE	Ex-Post Nash Equilibrium
GBM	Gradient Boosted Models
GDI	Grid Dependence Index
GHI	Global Horizontal Irradiance
HW	Holt Winters
IBC	Inter-Blockchain Communication
IC	Incentive Compatible
IR	Individual Rationality
ISO	Independent System Operator
KPI	Key Performance Indicator
LGB	LightGBM
LIC	Lugaggia Innovation Community
LV	Low Voltage
MAE	Mean Absolute Error
MPC	Model Predictive Control
NE	Nash Equilibrium
NILM	Non-Intrusive Load Monitoring
NWP	Numerical Weather Predictions

OPF	Optimal Power Flow
P2P	Peer to Peer
PCC	Point of Common Coupling
PDF	Probability Density Function
pFB	preconditioned FOrward Backward
PII	Personal Identifiable Information
PoA	Proof-of-Authority
POD	Point Of Delivery
PoS	Proof-of-Stake
PoW	Proof-of-Work
PV	Photovoltaic
RES	Renewable Energy Sources
SCC	Self Consumption Community
SOC	State Of Charge
SPB	Sharing Problem Based
VCG	Vickrey-Clarke-Groves
VNE	Variational Nash Equilibrium
VSC	Voltage Sensitivity Coefficients

1 Introduction

1.1 Projects synergies

In recent years, we have developed a series of research and pilot projects with the goal of investigating and demonstrating how centralized and decentralized control strategies can enhance demand flexibility. The contributions from the various projects have allowed us to build a set of software and hardware tools to test our design both in simulation and in the real world. The goal of this report is to provide a comprehensive picture of the results and synergies achieved by these projects.

Our activities started with the NEMoGrid project, where we focused on the definition and evaluation of innovative business models and market designs, starting from a business-as-usual approach, gradually introducing innovation until the definition of a peer-to-peer strategy based on blockchain technology.

We then started to work on a industrialization project, the OptiFlex Innosuisse project, in collaboration with Optimatik AG, a Swiss provider of smart metering solutions and Azienda Elettrica di Massagno SA (AEM), a local DSO. AEM is the first DSO in Ticino with a full rollout of smart metering and was interested to use the modern smart meters at the maximum of their capabilities. In OptiFlex the aim is to use the smart metering solution as a demand side management system, controlling the heat pumps and electrical water heaters with an intelligent algorithm, and couple this flexibility with a local hydropower plant in order to reduce peaks at the DSO's coupling point.

The two research and industrial projects naturally brought life to the P&D project called LIC - Lugaggia Innovation Community. In the LIC project, AEM did setup a self-consumption community, interconnecting 18 residential households with a kindergarten and installing a community battery. In this project, AEM explored how a DSO could take the community manager role by keeping full compliance to the current Swiss grid regulatory framework. The pilot acted as a perfect testbed to compare two technologies, the centralized strategy of OptiFlex and a peer-to-peer blockchain system developed by SUPSI and Hive Power, a SUPSI spin-off company.

Finally we did an additional step forward with the MuLDeR research project, where we explored a multi-level approach, taking in account the flexibility actuation effect at all grid levels. The aim is to develop grid-aware mechanisms for the activation of demand response.

1.2 NEMoGrid

The NEMoGrid Project is mainly focused on the definition of innovative business models that could ease the penetration of renewables into the distribution grid, with a particular emphasis on the definition of a peer-to-peer strategy based on a blockchain technology. The new business models encourage the active participation of citizens and the assumption of their new role of prosumers, by allowing them to enter new markets as players. Among the tested scenarios, the most innovative one are based on a peer-to-peer market. In this case, new decentralized platforms based on blockchain technology allow zero marginal cost transactions. In order to test the new business models effectiveness, a simulation framework has been developed. Technical developments within NEMoGrid is supported with user research, gathering empirical data on prosumers decisions and interactions. The results has been used to develop an adoption model and to continuously refine the simulations. Results on the 3 layers of technology, marketplace and stakeholders / adoption were discussed regularly within the ERA-Net Knowledge Community to contribute to a broad European knowledge base.

1.3 OptiFlex

The OptiFlex project aims to develop a new algorithm for the management of flexible loads integrating the production of energy from renewable sources such as solar and hydropower. This is achieved by a series of subgoals, consisting in the design, implementation and validation of:

- Forecasting algorithms for demand prediction.
- Algorithms for the active estimation of the sensitivity coefficients of the grid, which make use of the load flexibility to improve the quality of the estimates.
- Algorithms for the calculation of the power flow based on sensitivity coefficient of the grid.
- Disaggregation algorithms for the identification of the controllable loads.
- Machine learning algorithms to cluster loads with similar behaviour.
- Model predictive control scheme for the load dispatching.

OptiFlex uses the installed Smart Metering base as measurement, communication and switching basis, reducing substantially the need for hardware investments for Demand Side Management use cases.

1.4 LIC - Lugaggia Innovation Community

The municipality of Capriasca installed a 30 kWp PV plant in the village of Lugaggia on the roof of the local kindergarten. The building is located on the edge of a residential area, mainly consisting of single-family houses. The self-consumption potential of the kindergarten is limited because most of the production takes place during school summer holidays when the local consumption is low. AEM, the DSO serving the area, intends, therefore, to promote the creation of a SCC named Lugaggia Innovation Community (LIC), connecting together the kindergarten and ten nearby houses. To further increase the flexibility in the SCC, AEM is installing a district-level storage system. Differently from other research projects, such as “Quartierstrom”, the energy exchange inside the community will be compliant with existing laws regulating the Self Consumption Communities. By creating the SCC, AEM aims at testing and verifying its capability to provide new energy services to its customers, by leveraging on two novel technical solutions provided by the Swiss companies Optimatik and Hive Power:

- The first solution consists of a centralized energy management platform, which uses the existing smart meter infrastructure for sensing and actuation.
- The second solution implements a decentralized control approach secured by blockchain technology and requires the installation of computing and controlling unit, connected to the smart meters via DLMS interface.

The project aims to:

1. Evaluate the needs and requirements to the realization of LIC in a real environment. The project aims to provide recommendations how to allow and facilitate the replicability and scalability of peer-to-peer self-consumption communities. In particular, with respect to the needs of the public interest, to ensure fair treatment of all stakeholders (especially in areal situations) and implement measures for the correct use of energy resources (avoid sporadic tips, excessive consumption, control of equipment with the mandatory announcement, etc.).
2. Assess blockchain as a decentralized billing management method introduced by the utility.
3. Compare centralized vs decentralized load management methods from the DSO point of view (grid costs), energy consumption and economic point of view.

4. Help to assess the local flexibility potential and the different ways in which it could be exploited from a technical point of view.
5. Evaluate the degree of knowledge or acceptance among the community stakeholders to be willing to participate in these new self-consumption communities; a living lab to test users' acceptance will be set up.

1.5 MuLDeR

A comprehensive approach towards the actuation of flexibility, taking into account its effects at all grid levels and for all actors involved, is crucial for a successful implementation of demand response (DR). The goal of the MuLDeR project is to develop grid-aware mechanisms for the activation of demand response. Two methods are investigated: a multi-level hierarchical distributed control scheme making use of trusted aggregators at different levels of the grid and single level aggregators. In both cases, we will design mechanisms that allow offering DR services to all actors involved and at all grid levels while making sure that the control actions are not violating grid constraints throughout the entire network and that power quality is guaranteed. The applicability of the proposed solutions is being evaluated in LIC in collaboration with our industrial partners.

2 Market designs

The NEMoGrid 1.2, MuLDeR 1.5 and LIC 1.4 projects share some common coordination mechanisms and market design formulations, based on the treated business case and market actors. In particular, we have adopted an explicit market mechanism formulation, in which dynamic prices emerge from the decomposition of an optimization problem of interest, related to the particular business case considered (e.g. peak shaving, welfare maximization of a pool of agents, non cooperative coordination in an energy community, ...). Other approaches exist to design market mechanisms. Here we briefly review them and highlight their weakness in the case in which we are dealing with the trading of users' consumption and production flexibility:

- **Double auction markets**. This kind of market is best suited to deal with a high number of participants. It requires the collection of flexibility request and availability and the computation of the equilibrium cost for the traded flexibility. Unfortunately, the definition of flexibility itself is non-trivial, mining the foundation of this method. In fact, the current definition of flexibility relies on the use of forecasters to assess the change in end users' demand. If market participants have to provide their own energy forecasts (which is reasonable, since they can use private information to increases the forecasts' accuracy), the flexibility computation would be prone to exploitations from the market participants' side, as well as collusion. For example, one participant can place a bid with a very low marginal price (cts/kWh) for reducing its energy consumption, that will be almost surely accepted after the market is cleared. Then he/she must reduce its own consumption in the next step, w.r.t. the one that he/she has previously declared/forecasted. There is no way for the DSO or aggregator to verify that the declaration was truthful, besides running its own forecasts on the energy produced/consumed by this particular agent.
- **Iterative price discovery mechanism**. The DSO, or an aggregator, progressively apply a discount (in cts/kWh) to energy consumption, or a bonus to energy production, in order to steer its aggregated power profile. This approach relies on the definition of flexibility and on the initial declaration of market participants on their energy production/consumption, and is prone to collusion. E.g.: market participants can collude to initially falsely declare two times the consumption that they forecast for the next 24 hours, and wait for the DSO to increase remuneration for flexibility

(for reducing the consumption) up to the business-as-usual energy price. They then accept the bid, consume half of the declared energy (which was exactly what they foresaw they would have consumed in the first place) and, as a result, they zero their energy costs.

To avoid these kind of issues, we have based all the market mechanism designs on distributed control theory, as explained in subsection 2.2.

2.1 Actors

The market mechanisms used in the MuLDeR and NEMoGrid rely on the interplay of different actors, which we introduce in the following. Figure 1 graphically depicts the usual scale of the different actors.

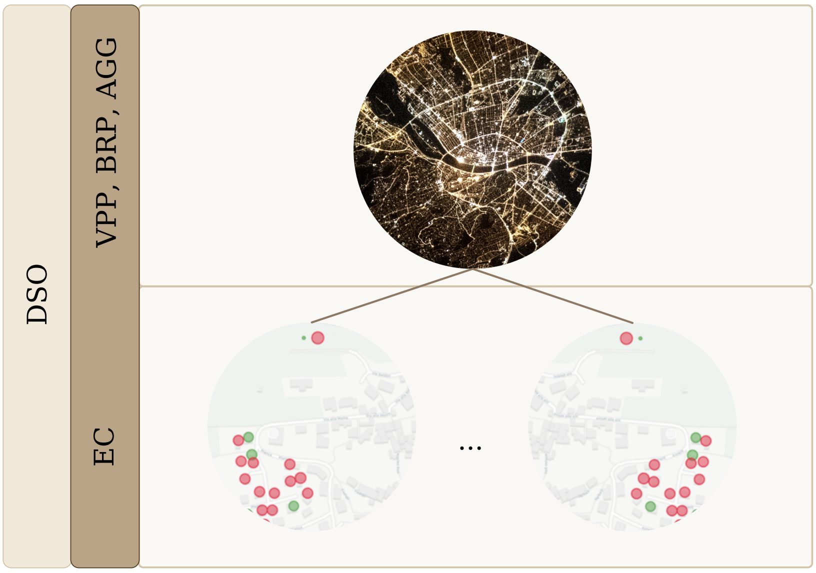


Figure 1: Scale of action of the various actors. Red and green dots represent single prosumers.

Prosumers The main actors within an EC are the end-users, some of which are also prosumers. A “prosumer” refers to “an energy consumer who, in his/her local area produces renewable energy and either saves surplus energy for future use, or trades within the local market to interested energy customers, enabling flexibility” [1]. A prosumer can thus benefit from redistributing its surplus energy to other users [2, 3]. A group of prosumers interested in energy provisioning consists of “a network of prosumers with reasonably common energy-sharing activities and profiles, that aim to accomplish a common purpose and participate collectively in the energy market” [4]. By using energy storage devices, prosumers may either store excess energy for potential usage, or transfer surplus energy to local consumers [5]. By controlling their energy demand and supply patterns and having shared storage capacity [6], prosumers will further engage in fostering distributed stability in the energy markets. The transition from conventional to future scalable energy systems [7] is also supported by numerous prosumer roles. Such roles include acting as an early market player, evaluating flexibility and strategic advantages, measuring economic benefits, validating business models and delivering technological input [7].

The demand in the selected electricity marketplaces is essentially represented by consumers and prosumers. The latter play at the same time the role of supplier, in a co-creation value process based on a close interaction between resources and audiences. Prosumers are thus designated to gain a much more relevant role in the next future, helping to change the electricity market organization. As already mentioned, this prospected scenario is coherent with the EU vision of an energy transition “consumer-centred” (EU Commission 2016) to be realized with the integration of prosumers and active consumers (individually or through aggregators) within the electricity system. In particular, the article 21 of the EU-Directive 2018/2001 on the promotion of the use of energy from renewable sources states that the Member States have to ensure that active customers are entitled to “generate, store, consume and self-generate electricity in all organized markets either individually or through aggregators without being subject to disproportionally burdensome procedures and charges that are not cost-reflective”. Consumers can participate in the market through local initiatives like the energy communities. Nevertheless, the value created may be delivered to additional qualified internal and external stakeholders within the selected market. Therefore, in a broader sense, a relevant role might be played by the DSO, the TSO, the aggregator, the retailer and, for the P2P trading, by the citizens of the local community, thanks to a mutually beneficial exchange. The presence of several stakeholders interacting within a multidimensional value chain instead of a unidirectional one allows for the inclusion of the P2P trading model into the category of network-based business models (Lund and Nielsen, 2014; Lazzarini et al., 2001).

Energy Community Energy communities (EC), also knwon as self-consumption communities, consist of a set of prosumers that exchange energy to maximize the group’s self-consumption. Electrical generators and consumers in an EC must be connected to low-voltage feeders on the same substation. They can internally optimize the synchronization of their energy production/consumption by exploiting electric storage and demand-side management. An example of an EC is a condominium where the solar energy produced on the rooftop is consumed by the tenants. Other members of the community, living in the same geographical district, also consume the solar energy produced from their own rooftops. When the solar system happens not to produce enough (e.g. at night or during cloudy days), the necessary electricity can be bought from the DSO grid. Similarly, the community can also sell excess solar power to the national grid (e.g. during summer days) and receive financial remuneration. Storage batteries can be used to optimize the self-consumption capabilities of the EC. The clear economic advantage of an EC comes from the large gap between purchase prices of energy (around 0.20 CHF/kWh) and the selling price of energy (around 0.06 CHF/kWh) from the national grid. These prices pushes distributed energy resources owners to self-consume the energy they produce. However, usually is not possible for residential prosumers to fully self-consume the energy produced by PV power plants, since the production is focused on few hours. The aggregation of flexible consumers and prosumers in ECs, increases the chances to increment the total self-consumption and decrease the total billing costs.

DSOs Distribution System Operators are in charge of monitoring the distribution gird, both for billing the costumers and for operating it. They are in charge of controlling the power quality at the end-users buses and to refurbish the grid accordingly. From the economic point of view, the rising number of prosumers and increase in self-generation tends to reduce the turnover of the DSO, considering that the grid component of the electricity tariff is usually a function of the consumed energy. On the other hand, due to the above-mentioned technical problems, the investments in the network infrastructure are expected to increase. As a consequence, in Switzerland grid tariffs are likely to increase and with them the social disparities between people who can afford a PV plant and those who do not. To restore fairness, one solution would be to redesign the local energy and grid tariffs.

Aggregators, BRP, VPP Aggregators, balance responsible parties (BRP) and virtual power plants (VPPs) are middle actors managing and aggregating the flexibility of pools of prosumers and end-users to the end of selling it on different kind of energy markets, like day-ahead, intra-day, primary secondary or tertiary regulation.

Legislators and energy regulators Legislators and Energy regulators, such as the Swiss Federal Electricity Commission (ElCom) regulate prices and tariffs in the electricity sector and "may prohibit unjustified electricity price increases or retroactively reduce excessively high tariffs"[1]. The market mechanisms that we present in 2.2 generate costs which depend on the consumption of all the energy community participants. That is, it is not directly proportional to the energy consumed by the end user, and it

[1] https://www.elcom.admin.ch/elcom/en/home.html

gets lower when the self consumption inside the energy community increases. Switzerland is embracing a causal principle on the price formation for end users, as stated in the recent modification to the Federal Electricity Supply Act [8]. This means that the electrical bills "should reflect costs caused by end users". However, the Electricity Supply Ordinance [9] states that DSO must guarantee to the end users that at least 70$\%$ of their bills are directly proportional to their energy use; at the same time they can offer opt-in tariffs in which this percentage is reduced. Under these constraints, the tariff proposed in the presented projects can be potentially applied in Switzerland.

2.2 Market mechanisms

The choices of the market mechanisms has been driven by the following desirable properties:

- The market must be cleared in pseudo-real time. Since RE power production is stochastic in nature, in order to alleviate the issues they pose on the electrical grid, the market must be cleared in pseudo-real time (e.g. every 15 minutes). Thus, mechanisms allowing fast algorithms for price formation computation, coordination, and for clearing the market are necessary.
- The energy market must be useful for several business cases and usable by different actors (e.g. Energy Communities managers, BRPs, energy retailers, etc...).
- The price formation mechanism and market clearing must be robust with respect to loss of communication of (and between) market participants.
- The energy prices formed by the market mechanism must be fair (i.e. more flexible agents must be remunerated more) and guarantee the Individually Rational (IR) property, that is, no one should pay more than what he/she would have paid with the business-as-usual tariffs.
- The design has to be easy to understand: complexity could pose unnecessary challenges to get regulatory approval and lower adoption rates among end users.

2.2.1 Single level EC - surplus repartition

The final market formulation we developed for the EC case is based on the the following settings:

- A group of end users can control some deferrable loads (e.g. electric boilers, heat pumps, etc...) or an electrochemical storage, which can be used in order to shift production/consumption patterns
- An Independent System Operator (ISO), a DSO or an aggregator, exploits a defined business model, which uses end users' flexibility (e.g. committing to a predefined day ahead power profile as a service to a balance responsible party, providing regulation energy reserve, etc...) and redistributes part of the capital gain to flexible end users as a reward to their flexibility.

Under these assumptions, the problem can be mathematically formulated as a variant of the so called sharing problem [10]:

$$\begin{aligned} &\underset{u \in \mathcal{U}}{\text{argmin}}\, e(S_{\emptyset} u) + \sum_{i=1}^{N} c(u_i) \\ &s.t. : A_{\lambda} u \leq b \end{aligned} \tag{1}$$

where $u_i \in \mathrm{IR}^{\mathrm{T}}$ are the actions associated with the agent i in the control horizon T, $\mathcal{U} = \prod_{i=1}^{N} \mathcal{U}_i$ is the Cartesian product of the flexible users feasible sets, $e(u)$ is a system level objective which defines the business model, $c(u_i)$ are the costs of each flexible user in the business as usual case, and $u = [u_1^T, ..u_N^T] = [u_i]_{i=1}^N$ is the vector of the concatenated actions of all the flexible users. Here A_{λ} is a constraint matrix, taking into account the linear influence of the end users' powers on the problem constraints, and b encodes box constraints limits on the power and voltages at specified grid's nodes.

A notable business model is the one of energy communities, or self consumption community (SCC), for which the billing is done at the point of common coupling (PCC) with the main grid. This means that users inside the energy community always pay a lower energy bill if there is a heterogeneous mix of generation and production inside the energy community. In the case of energy communities, the function $e(u)$ is the surplus that the agent community has in paying the energy at the point of common coupling with the electrical grid:

$$e(x) = c\left(\sum_{i=1}^{N} u_i\right) - \sum_{i=1}^{N} c(u_i) \tag{2}$$

where $c(\cdot)$ is the energy cost function defined as:

$$c(p_t) = \begin{cases} p_{b,t} p_t, & \text{if} \quad p_t \geq 0 \\ p_{s,t} p_t, & \text{otherwise} \end{cases} \tag{3}$$

where $p_{b,t}$ and $p_{s,t}$ are the buying and selling tariffs, respectively, at time t and p is the power at the households' electric main. In order to induce agents to follow the proposed mechanism, we must ensure that the energy tariff they pay participating in the market is always lower than the one they pay in the base case. This is always true when we are not taking into account grid constraints, since $e(u)$ as defined in (2) is always negative, when $p_{b,t} \geq p_{s,t}$, as usual in energy tariffs. However, if the agents are located in a grid with big voltage oscillations, the Lagrangian dual variables (which we can interpret as punishment prices) could be such that the cost paid by the agents is higher than their reward from the redistribution of $e(u)$.

In [11] we proved the existence and uniqueness of a generalized variational equilibrium (G-VNE) for this class of sharing problems. We prove that decomposing (1) using different repartition weights for the surplus, induces a game with unique G-VNE, which can be reached jointly minimizing the utility function of the agents, given by:

$$c_{tot,i} = c(u_i) + \alpha_i e(u) + \lambda_i^T u_i \tag{4}$$

$$= \alpha_i c\left(\sum_{i=1}^{N} u_i\right) + (1-\alpha_i)c(u_i) + \lambda_i^T u_i \tag{5}$$

where α_i is a repartition coefficient for prosumer i, $\lambda_i \in \mathbb{R}^T$ is a vector of Lagrangian multipliers associated with the i_{th} agent. The α_i coefficients are computed using moving averages of the change in the surplus due to the action of the i_{th} agent. In other words, called $e(x_{-i})$ the surplus computed without the action of the i_{th} agent, the α_i are defined as:

$$\alpha_{i,t} = a_{i,t} \frac{\sum_{k=t-\tau}^{t} \tilde{\alpha}_{i,k}}{\sum_{k=t-\tau}^{t} \sum_{i=1}^{N} \tilde{\alpha}_{i,k}} \tag{6}$$

$$\tilde{\alpha}_i = e(x_i) - e(x_{-i}) \tag{7}$$

2.2.2 Single level EC - Automated Market Making Mechanism

In the LIC community, the market mechanism is not based on the redistribution of the surplus (2), but is based on dynamic prices with functional dependence on the instantaneously produced or consumed energy inside the local grid. This kind of price formation mechanisms is also known as automated market making mechanism (AMM). Although solving problem (1) is way more flexible in terms of objective function which can be tackled, a fixed dynamic price scheme is way easier to explain and could be more easily adopted. We show in the following that the proposed dynamic tariffs promote self consumption and reduce the overall power profile variance. We defined the price formation mechanism using extremely simple and interpretable rules:

- The energy consumed from the external grid shall be paid for as if the consumer were not part of the EC

- The energy consumed from inside the EC is paid for at a total price lower than the standard tariff of the energy supplier and DSO, with a discount proportional to the ratio of the total produced and consumed energy
- The energy injected into the external grid shall be remunerated as if the consumer were not part of the EC
- The energy injected, which is consumed inside the EC is remunerated at a price higher than the standard tariff of the energy supplier, with a discount proportional to the ratio of the total consumed and produced energy

This simple set of rules push flexible users to increase the overall self-consumption. The energy price is calculated using an AMM mechanism, which follows the following principles:

- The self-consumed energy is equally split among the EC participants proportionally to their consumption and production
- The instantaneous buying and selling prices are dynamic, but for a given time slot they are the same for everyone

This generates the following dynamic prices:

$$\begin{aligned} p_b &= \left(E_c p_b^{BAU} - \min(E_c, E_p)(p_b^{BAU} - p_b^{P2P})\right)/E_c \\ p_s &= \left(E_p p_s^{BAU} - \min(E_c, E_p)(p_s^{P2P} - p_s^{BAU})\right)/E_p \end{aligned} \tag{8}$$

where p_b and p_s are the buying and selling prices generated by the AMM, E_c and E_p are the sum of the energy consumed and produced inside the energy community, while p_b^{BAU}, p_s^{BAU}, p_b^{P2P} and p_s^{P2P} are the buying and selling prices in the Business as Usual (BAU) case and inside the energy community. In such a pricing configuration, peers clearly profit from the difference in price between BAU and P2P, but the community administrator also earns money when energy is self-consumed inside the community. It is important to notice that the P2P tariff is applied only to the energy produced by the members of the community. As a consequence, it is also in the administrator interest to maximize self-consumption (no conflicting interests between peers and community admin). It can be shown that these prices generate convex costs as a function of the agents' actions, and thus are amenable to be jointly optimized in a distributed way, as stated in [11]. Besides study the convexity of the total costs, it is important to understand the implications of such defined prices. We can develop some intuition on how these prices reduce the variance of the aggregated power profile and maximize self-consumption plotting them as a function of the grid dependence index, defined as:

$$GDI = (E_p - E_c)/(E_p + E_c) \tag{9}$$

The GDI defines how much the EC is dependent on the main grid, which provide the reservoir of negative or positive energy. When the GDI is equal to 1, no one is consuming energy within the EC, while a GDI of -1 indicates that all the agents are acting as consumers. As can be seen in Figure 4, the selling price for a net energy producer increases as the GDI moves from 1 to 0, and then reaching a plateau. The same is seen for the buying price for a net consumer, decreasing while the GDI shifts from –1 to 0, and getting constant thereafter. This means that the EC maximizes its welfare when the GDI is 0, that is when the buying price is minimized while the selling price is maximized for the agents. This means that the EC maximizes its welfare when the self-consumption is maximized. For this market mechanism, we tested two control strategies. Since costs are minimized when self-consumption is maximized, instead of directly minimizing the prices in equations (8), we decided to minimize the surplus function $e(u)$ previously defined in (2), using the same control mechanism described in 2.2.1, without considering system level constraints, which are tackled with power based tariffs. We tested also a different control strategy, called "implicit", that doesn't need communication between users, and is further described in section 3.1.4.

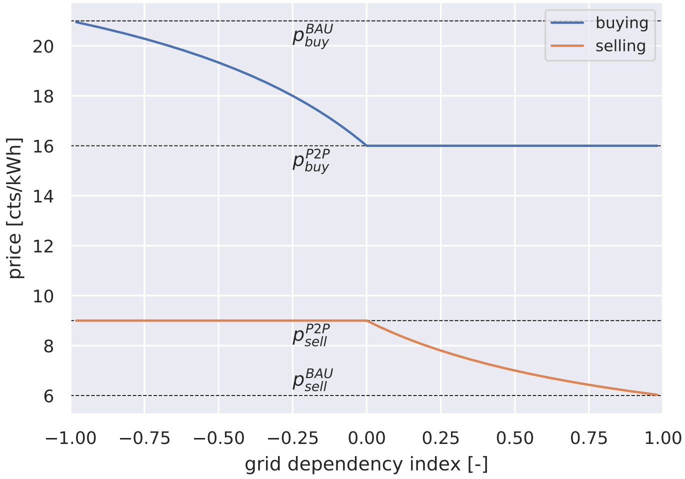


Figure 2: Buying and selling prices defined by the AMM in (8), as a function of the GDI (9)

2.2.3 Power based tariffs

Power based tariffs are an alternative to explicit grid constraints considered by the EC problem formulation in (1); they are an additional cost that can be sum to the usual energy tariff, and can be seen as a grid tariff. The power-based price can be expressed as:

$$p_{pt} = \beta E \tag{10}$$

where E is the total energy produced/consumed by all the market participants. The cost for user i then becomes:

$$c_{pt,i} = \beta E e_i \tag{11}$$

which can be thought as a quadratic cost on the ith user's energy consumption/production, since $E = \sum_i e_i$. In practice, the mechanism introduces a bonus malus component to the energy tariff, depending on the sign of the ith user power profile: if the ith user help in flattening the overall poer profile (i.e., the sign of e_i is different from the sign of E), he/she will receive a bonus; a malus applies for the converse. Depending on the signs of E and ei , the cLFM cost can be positive or negative, representing a punishment or a reward in helping to flatten the power profile. Despite being a reward for some users, we can show that this mechanism always produces a net surplus, which can be managed by the LFM operator:

$$\text{Total pt cost} = \sum_i (p_{pt} e_i) = \sum_i (\beta E e_i) \tag{12}$$

$$= \beta E \sum_i (e_j) = \beta \left[\sum_i (e_j) \right]^2 \geq 0 \tag{13}$$

2.2.4 Multilevel EC

The multilevel mechanism is an extension of the single level EC market mechanism to a hierarchically organized structure. This allows to obtain a market resilient to loss of communication of participants subgroups, and preserving privacy of the end users, since information is communicated only by means of aggregated power profiles. The multilevel algorithm was published in [12] and described in section 3.2.

2.2.5 DSO planned

In the DSO planned market mechanism the DSO can directly control deferrable loads and electric batteries owned by the end users, in change for a discount on the final electric bill.

2.2.6 Two actors with one-shot setup

This mechanism allows a flexibility provider, or aggregator, to negotiate ex-ante a compensation for the reduction of the energy peak of a DSO.

$$c = \begin{cases} S_{DSO} & \text{if} \quad P_{peak} < P^{low}_{max} \\ S_{DSO} - \alpha(P_{peak} - P^{low}_{max}) & \text{if} \quad P^{low}_{max} \leq P_{peak} \leq P^{high}_{max} \\ \text{maximum}(S_{AGG}, \beta(P^{high}_{max} - P_{peak}) & \text{if} \quad P_{peak} > P^{high}_{max} \end{cases} \tag{14}$$

where S_{DSO} and S_{AGG} are the DSO's and aggregator's stakes, respectively. If the the power's monthly peak P_{peak} stays below P^{low}_{max}, the aggregator receives all the DSO's stake. If P_{peak} falls between P^{low}_{max} and P^{high}_{max}, the final compensation for the aggregator is linearly reduced with the coefficient $\alpha\,[kCHF/MW]$. On the other hand, if the aggregator didn't manage to keep P_{peak} below P^{high}_{max}, he will ends up paying a penalty to the DSO, up to the aggregators' total stake S_{AGG}. Also in this case, this penalty is linear with coefficient $\beta\,[kCHF/MW]$. The coefficients S_{DSO}, S_{AGG}, α, β, P^{low}_{max}, P^{high}_{max} are negotiated one-shot between the DSO and the aggregator; the last two can be subject to negotiation once per month, having the maximum expected peak per month a yearly seasonality. The blockchain implementation for the NEMoGrid project for this mechanism is described in section 4.1.1.

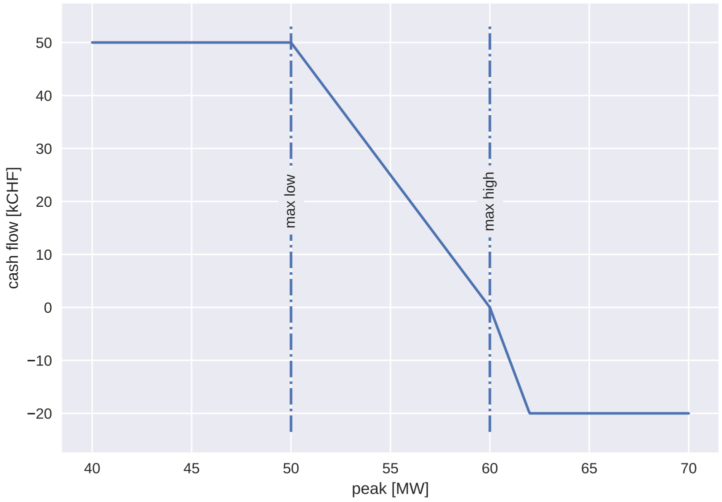


Figure 3: Cash flow example for the two actors one-shot setup mechanism

3 Algorithms

3.1 Single-level algorithms

In the following sections, the algorithms used to control the distributed batteries and electric boilers at single level are described. For sake of simplicity we describe the algorithms in the case in which

Project	Market formulations	Coordination	Actors	Tests
NEMoGrid	single level EC surplus/AMM	GNE pFB	users, community manager	LIC:simulation
	power based tariff	none/GNE pFB	users, DSO	LIC:simulation
	DSO planned	centralized	users, DSO	LIC:simulation
	two actors	One-shot setup	aggregator, DSO	Sweden: blockchain simulation
MuLDeR	single level EC surplus	GNE pFB implicit	users, community manager	LIC:simulation +preliminary results
	multilevel EC surplus/AMM	GNE multilevel ADMM	users, community manager	LIC:simulation
OPTIFLEX	DSO planned	centralized	users, DSO	tested

Table 2: Type of market formulation and coordiantion adopted by each project

the objective function is the economic cost in the business as usual. This formulation is also used in the implicit coordination case, where the prices are dynamically changed and each agent uses its own device to reduce its overall costs, without any kind of communication. Batteries and boilers are controlled through a model predictive control (MPC) approach: at each timestep of the simulation, the controller solves an optimization problem using consumption and production forecasts for the next day-ahead. Once the optimal solution has been found, the algorithms actuate only the first control action, and the procedure is repeated. We later extend it to the sharing problem with coupling constraints case presented in [11].

3.1.1 Battery control algorithm

The battery controller is supposed to be interfaced with the battery energy management system, returning an estimation of the battery's state of charge and injected and withdrawn power, into and from the battery. In this setting, the battery can be considered as a one state fully observed system and applying the MPC is straightforward. The formulation of the battery control algorithm is based on the work published in [11], and has been further improved to decrease the overall computational time, exploiting a new formulation for enforcing mutual exclusivity in charging and discharging operations. We report it in the following. Called $u = [p_{ch}^T, p_{ds}^T]^T \in \mathbb{R}^{2T}$ the vector of concatenated decision variables for the control horizon T, where p_{ch} and p_{ds} are the battery charging and discharging power, respectively, $\tilde{u} = [p_{ch}, p_{ds}] \in \mathbb{R}^{T \times 2}$ being the same vector reshaped in a 2 columns matrix, $\hat{p} \in \mathbb{R}^T$ being the forecasted power at household's main for the next contro horizon, $y \in \mathbb{R}^T$, $s_{ch} \in \mathbb{R}^T$, $s_{ds} \in \mathbb{R}^T$ being three auxiliary variables, we seek to solve the following problem:

$$u^*, y^* = \underset{u,y}{\operatorname{argmin}} \sum_{t}^{T} y_t + \|s_{ch}\|^2 + \|s_{ds}\|^2 \tag{15}$$

$$x_{t+1} = Ax_t + B\tilde{u}^T \tag{16}$$

$$y \succcurlyeq p_b \left(\tilde{u}[1,-1]^T + \hat{p}\right) \tag{17}$$

$$y \succcurlyeq p_s \left(\tilde{u}[1,-1]^T + \hat{p}\right) \tag{18}$$

$$x \in [x_{\min}, x_{\max}] \qquad u \in [u_{\min}, u_{\max}] \tag{19}$$

$$s_{ch}, s_{ds} \succcurlyeq 0 \tag{20}$$

$$s_{ch} \succcurlyeq -\hat{p} \qquad s_{ds} \succcurlyeq \hat{p} \tag{21}$$

$$u \preccurlyeq [s_{ch}, s_{ds}] \tag{22}$$

where $\succcurlyeq$ stands for $\succcurlyeq_{R_+}$, indicating element-wise inequalities, $p_b \in \mathbb{R}^T$ and $p_s \in \mathbb{R}^T$ are the business as usual buying and selling prices. We start analyzing the objective function (15) term-wise. The first summation in (15) represents the total cost of the agent in the business as usual case. For prosumers, the cost function can be either positive or negative, depending on the overall power at their household's main and can be expressed as in equation (3). The cost can be thought of as the maximum over two

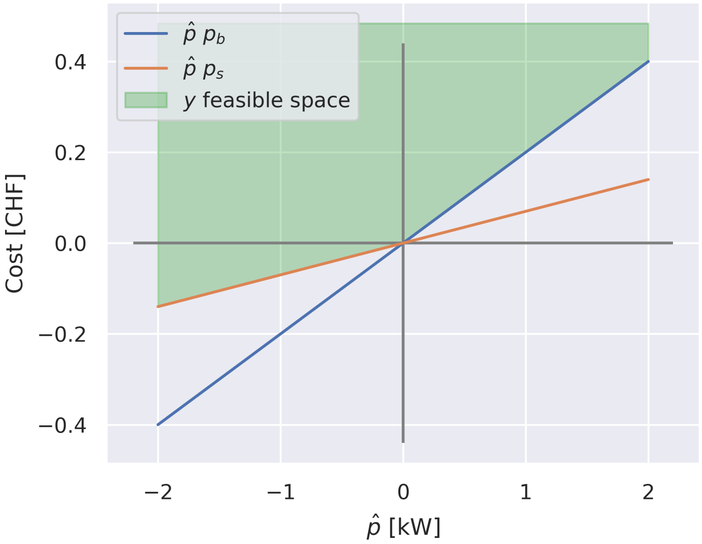


Figure 4: Visual explanation of the scope of the y variable. When linearly penalized, y is pushed to its feasible space's lower borders, collapsing on the cost function $c(p)$ in (3)

affine functions (the first and second line of equation (3), respectively). Equations (17),(18) constraint y to live in the epigraph of the maximum of these two affine functions. Minimizing y then guarantees that its value at the optimum, y^*, will lie on the epigraph's lower boundary (and will thus represents the prosumer's total costs), as shown in figure 4. Equation (16) describes the battery's dynamics. $A \in \mathbb{R}_+$ and $B \in \mathbb{R}_+^{1\times 2}$ are the discrete dynamics matrices obtained by the continuous one through exact discretization [13]:

$$\begin{aligned} A &= e^{A_c dt} \\ B &= A_c^{-1}\left(A_d - I\right) B_c \end{aligned} \tag{23}$$

where $A_c = \frac{1}{\eta_{sd}}$ and $B_c = [\eta_{ch}, \frac{1}{\eta_{ds}}]$, and η_{sd}, η_{ch} and η_{ds} are the caracteristic self-discharge constant, charge and discharge efficiencies, respectively. Since B_c defines an asymmetric behaviour in charging and discharging (even with equal charging/discharging coefficients), solving the battery scheduling requires to use two different variables for the charging and discharging powers, p_{ch} and p_{ds}. When considering grid constraints, the battery can try to dissipate energy through round-trip efficiency to help respect negative grid constraints (when there is an excessive PV generation), so that in this case we need explicit binary complementary constraints for enforcing mutual exclusivity (the battery cannot charge and discharge at the same time). This can be obtained in three ways: explicitly modeling the bi-linear constraint $p_{ch}p_{dc} = 0$, introducing a binary variable and model it through big M formulation, or trying to restrict their feasible space. The first way will make the problem non-linear, while the second will turn it into a MIQP introducing a binary variable; as both options will increase the computational time, we introduced a new formulation exploiting the third way. Charging and discharging powers are effectively separated using the auxiliary variables s_{ch} and s_{ds}. The feasible space of s_{ds} is constrained to be the epigraph of the maximum between 0 and the forecasted power at the main. As shown in figure 5 for the case of s_{ds}, the equations (20) and (21) constraint these auxiliary variables to live in the positive half-plane and to be higher than the power profile at main (or its negative value for s_{ch}). When s_{ch} is quadratically punished, it will shrink on the lower boundary of the epigraph, (orange line in the second panel of figure 5). Its optimal value can then be used to define the feasible regions of the battery charging power, as done by equation (22). The same reasoning done in figure 5 for the discharging power can be applied to define the feasible regions for the battery's charging power; this will result in two disjoint feasible sets for the charging and discharging powers.

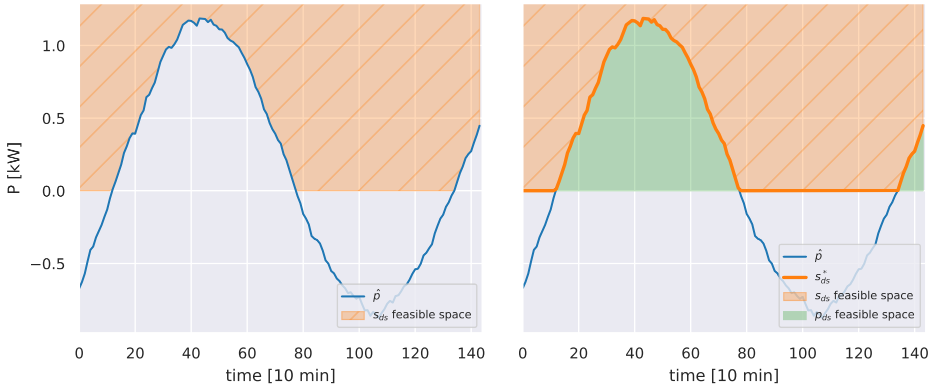


Figure 5: Visual explanation of the change in the feasible space for the discharging power.

3.1.2 Boiler algorithm

For the electric boilers, we cannot realistically assume them to be a fully observable systems. In fact, this assumption will require to have several sensors indicating their internal temperatures at different heights of the boilers. In a realistic setting, existing electric boilers has no more than two temperature sensors, used by their internal hysteresis controllers, and this information cannot typically be read from an external controller. Furthermore, consider the following simplistic one state model for the boiler's thermal dynamics:

$$cM\frac{\partial T}{\partial t} = c\Gamma\left(T_{i,t} - T_{o,t}\right) - U\left(T - T_{ext}\right) + P_{el,t} \tag{24}$$

Despite its simplicity, this model requires to know the incoming/outgoing water flux Γ, which means that a fluximeter must be installed. This is not possible but in pilot projects, since installation costs of these sensors will completely cancel out the economic benefit of an avoided grid refurbishment.

As such, we assume that we can only exploit the electric power measurements for controlling electric boilers. Furthermore, we expect to be able to only turn off the boiler through a relay, and not forcing it on (due to safety reasons, since we do not any feedback). Given these constraints, the electric boiler's nominal power and energy needs are estimated using historical data of their power consumption. Then, the algorithm decides when to force off the boiler such that the boiler can always satisfy its energy needs inside 3-hours slots. We based our algorithm on the work published in [14]. The algorithm is summarized in the following points:

- The nominal power of the boiler, P_{nom} is estimated from historical power data.
- The energy needs of the boiler are forecasted using a LightGBM[2] model taking as input past data of the boiler's power profile, as well as weather predictions for the next 24 hours. Furthermore,

[2] https://lightgbm.readthedocs.io/

forcing the boiler off could result in an energy rebound effect. This can be corrected by passing to the forecaster also historical values of the control action as a categorical binary variable (since we want to forecast the energy needs of the uncontrolled boiler, this approximately counteracts our action on the system).

- The algorithm decides when to force off the boiler such that the boiler can always satisfy its energy needs inside 3-hours slots. For example, if a consumption of 2 kWh is forecasted between 18h-21h, and the estimated nominal power is of 4 kW, the boiler can be forced off at most 2h30min during this period.

Even if the boiler cannot actively be forced on, the internal control of the boiler, which is usually an hysteresis based on one or two temperature sensors, will automatically turn it on if its internal temperature is too low. The mathematical formulation is the following:

$$u^*, y^* = \underset{u,y}{\operatorname{argmin}} \sum_t^T y_t - \sum_t^T \min(\gamma, 0) \tag{25}$$

$$\text{s.t.} \quad y \succcurlyeq p_b \left(\hat{p}_b(1-u) + \hat{p}\right) \tag{26}$$

$$y \succcurlyeq p_s \left(\hat{p}_b(1-u) + \hat{p}\right) \tag{27}$$

$$S\left[(1-u)p_{nom} - \hat{p}_b\right] \succcurlyeq \gamma \tag{28}$$

$$\sum_{t=1}^{T-1} |\Delta u| \succcurlyeq n_{ch} \tag{29}$$

where y has the same role as in the battery optimization problem, representing the total costs for the prosumer, γ is a slack variable which relax the energy invariance constraint (28). Here S is a summation matrix which sum the energy in the pre-defined time slots (3 hours). Equation (29) further prevents the boiler for being turned on and off more than n_{ch} times in a control horizon.

3.1.3 Explicit optimization of EC surplus repartition

The presented problem formulations for the battery and the boiler, (15) and (25) respectively, minimize the end users' business as usual costs. These can be adapted to solve the decomposed sharing problem (1) simply modifying the part of the objective function representing the end users' costs and altering the feasible space for the charging and discharging powers. In particular, using the expression presented in section 2.2.1 for the total costs for the agent, the economic cost of the agent becomes:

$$c_{tot,i} = c(u_i) + \alpha_i e(u) \tag{30}$$

$$= \alpha_i c\left(\sum_{i=1}^N u_i\right) + (1-\alpha_i)c(u_i) \tag{31}$$

This cost function must be augmented with the Lagrangian multipliers coming from the decompostion of problem (1):

$$c_{tot,i} = \alpha_i c\left(\sum_{i=1}^N u_i\right) + (1-\alpha_i)c(u_i) + \lambda_i^T u \tag{32}$$

$\lambda_i = [\lambda_{i,ch}^T, \lambda_{i,ds}^T]^T \in \mathbb{R}^{2T}$ being the vector of Lagrangian multipliers associated to the i_{th} agent. Briefly speaking λ_i is a filtration of the overall λ induced by A_λ in (1). More details about this filtration process can be found in [11, 12]. In order to allow batteries to charge and discharge when system-level constraints are violated, we must further modify the feasible space of p_{ch} and p_{ds} in (15). In particular, equations (21) and (22) becomes:

$$s_{ch} \succcurlyeq -\hat{p} + M\lambda_{i,ch} \tag{33}$$

$$s_{ds} \succcurlyeq -\hat{p} + M\lambda_{i,ds} \tag{34}$$

$$\tag{35}$$

Here M is a big constant, that we set to $1e6$ in the simulations. An example of change in the feasible space of p_{ds} is shown in figure 6.

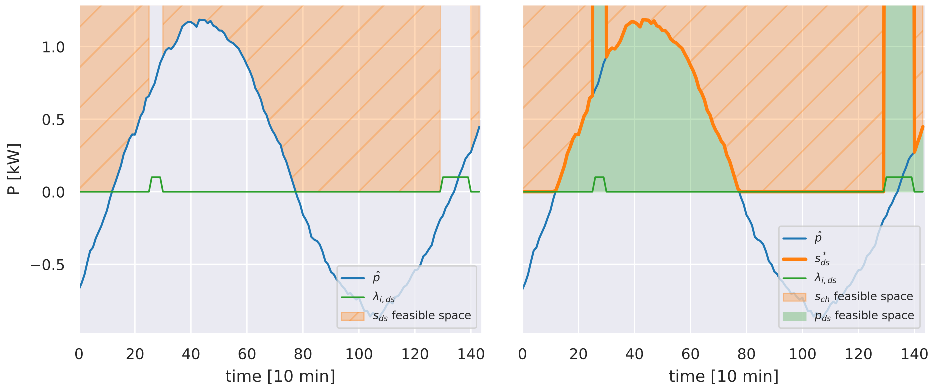


Figure 6: Visual explanation of the change in the feasible space for the discharging power, when s_{ds} is modified taking into account the Lagrangian associated to the battery's charging power $\lambda_{i,ds}$.

Decomposing (1) using different repartition weights for the surplus, induces a game with unique generalized variational equilibrium [11], which can be reached using the preconditioned forward backward (pFB) formulation [15]. At each communication round k, agents can share their plans in order to jointly minimize their objective functions. The original pFB method can be written as reported in algorithm 1. Here $\mathcal{F}(x^k)$ is the gradient of the overall objective function for the end user (4). The $\mathbf{\Pi}_Z$ in 1 stands for

Algorithm 1 Preconditioned forward-backward (pFB)

```
$u^{k+1} = \mathbf{\Pi}_{\mathcal{U}_i}\left[u^k - \alpha(\mathcal{F}(u^k) + A^T\lambda^k)\right]$
$\lambda^{k+1} = \mathbf{\Pi}_{\mathbb{R}^+}\left[\lambda^k + \beta(2A_\lambda u^{k+1} - A_\lambda u^k - b)\right]$
```

the projection onto the convex set Z. Since projecting on the EVs' set of constraints (16)-(18), $\mathcal{U}_i$, is hard (it includes binary constraints), we reformulated the gradient descent step (the first line of algorithm (1)) as the minimization of the linearization of the system level cost around the previous state, plus a quadratic punishment on the action at the previous iteration; more details on this equivalence can be found in [11]. Replacing the agent cost with the auxiliary variable y as in (15) and (25), the final objective function (for the battery) then becomes:

$$\alpha_i \nabla c \left(\sum_{i=1}^{N} u_{i,pre}\right)^T u_i + (1-\alpha_i)\sum_{i=1}^{T} y + \lambda_i^T u + \rho_d \|u - u_{pre}\|^2 + \|s_{ch}\|^2 + \|s_{ds}\|^2 \tag{36}$$

where $u_{i,pre}$ are the agents actions at the previous iteration. This formulation can be readily used to include the power based tariff presented in section 2.2.3 which has the purpose of promoting peak shaving:

$$\alpha_i \nabla c \left(\sum_{i=1}^{N} u_{i,pre}\right)^T u_i + (1-\alpha_i)\sum_{i=1}^{T} y + \lambda_i^T u + \rho_d \|u - u_{pre}\|^2 + \|s_{ds}\|^2 + \beta P_{PCC,pre} u \tag{37}$$

where β is a tunable parameter in $[CHF/kWh^2]$.

3.1.4 Implicit coordination for EC AMM prices

In this variant, batteries try to indirectly coordinate to optimize the AMM EC prices presented in section 2.2.2. The previously described explicit coordination mechanism, using the pFB algorithm, could be used also to optimize an EC using the AMM prices, as also in this case the welfare of the EC is maximized at increasing rates of self-consumption. However, an alternative coordination mechanism is possible without the need for communication between users. The agents can indirectly coordinate themselves by forecasting the dynamic prices and optimizing for their own costs. That is, the batteries solves problem (15) where p_b and p_s are forecasts of the prices which generates through the AMM equations (8). This method is expected to reach suboptimal solutions with respect to the explicit coordination presented in the previous section 3.1.3; on the other hand it doesn't require several coordination rounds, but just one.

3.2 Multi-level algorithms

The flexibility of prosumers can be exploited providing multiple services to the electrical grid, for example secondary control and energy management. Due to the electrical grid interconnections, in order to provide these services safely, an optimization method taking into account their effects on multiple voltage levels should be used. Moreover, coordinating groups of prosumers is always better than having prosumers to follow individual policies, in terms of aggregated effect. For example, let us consider the case in which we want to minimize a given objective function in terms of the aggregated power profile of a group of prosumers connected at the same LV transformer. Additionally, suppose that we require the power at the PCC to be symmetrically bounded, that is, $Sx \in \left[-\overline{P}, \overline{P}\right]$, where with $S = \mathbb{1}_N^T \otimes I_T$ we denoted the time summation matrix of all the prosumers' actions. In order to guarantee that this constraint is never violated using individual policies, we must replace it with the set of prosumers' constraints $u_i \in \left[-\overline{P}/N, \overline{P}/N\right]$. On the other hand, the feasible space of prosumers' actions would increase when allowing communication, thus leading to a better solution, as shown in figure 7. Formally:

Lemma 3.1. *The n-dimensional hypercube intersecting each axis in $\{-\overline{P}/N, \overline{P}/N\}$ is always included in the convex hull defined by the points $\left[e_i(-\overline{P}, \overline{P})\right]_{i=1}^N$, where e_i is the versor of the ith dimension.*

Proof. The maximum distance of the center from the n-dimensional hypercube defined in 3.1, equal to half of its diagonal, is $\overline{P}/\sqrt{N}$, which equal to the distance of the hyperplanes defined in 3.1 form the center, in each orthant. □

This means that constraining the aggregate instead of the single users will enlarge the feasible set $\mathcal{X}$, always resulting in a (not strictly) better solution.

In the literature, different methods have been proposed to coordinate groups of prosumers by means of an aggregator. Most of them refer to a single level hierarchy, as in [16, 17]. Both [18, 19] consider distributed MPC for a single level hierarchy, without grid constraints. In [20] a three level hierarchy has been proposed for the control of reactive power from PV inverters. The adopted sequential strategy does not explicitly take into account grid constraints and is solved through particle swarm optimization heuristic.

In paper [21], we present a multilevel hierarchical algorithm for the coordination of prosumers located in different voltage levels of the electrical grid. The hierarchical structure of the grid is described by means of a rooted tree. An example of rooted node structure is shown in figure 8. The coordination of prosumers is made possible by aggregators located at branching nodes of the tree. It is not necessary that the levels of the hierarchical structure coincide with the grid voltage levels. For instance, the first level of aggregation could be done at building level, while the second level could be placed at the LV cabinets. Here we briefly introduce the some notations needed to describe the hierarchical structure,

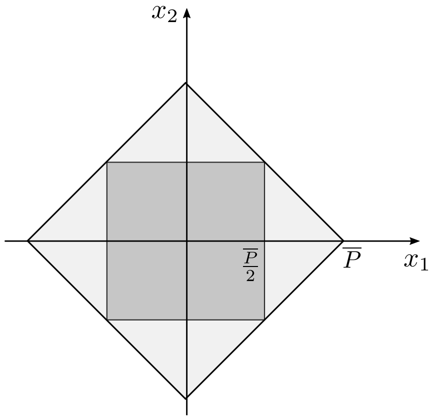


Figure 7: Feasible sets for the space of two prosumers' actions, for the constraint $|x_1 + x_2| < \overline{P}$ in the case of individual policies (dark gray) and in the case of communication (light gray). Communication enlarges the feasible set, thus potentially improving the solution.

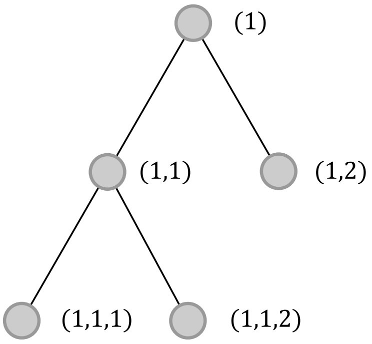


Figure 8: Example of rooted tree hierarchical structure

which are useful to explain the coordination mechanism.

Definition 3.1 (Node sets)**.**

1. Descendants of node $A = (d_1, ..d_L)$. $\mathcal{D}(A) = \{A_j = (j_1, ..j_{L_j}) : L_j > L, (j_1, .., j_L) = A\}$
2. Leaf node $\mathcal{L}(\tau) = \{A \in \tau : \mathcal{D}(A) = \emptyset\}$
3. Nodes in level l. $\mathcal{N}_l(\tau) = \{A \in \tau : L = l\}$
4. Branching nodes. $\mathcal{B}(\tau) = \tau \setminus \mathcal{L}(\tau)$
5. Anchestors of node A. $\mathcal{A}(A) = \{A_j \in \tau : A \in \mathcal{D}(A_j)\}$

Formally, given N users with a controllable loads, located in a hierarchical structure, we solve the hierarchical counterpart of problem 1:

$$\begin{aligned} &\operatorname*{argmin}_{u \in \mathcal{U}} e(S_\emptyset u) + \sum_{i=1}^{N} c(u_i) \\ &s.t. : S_B u \leq v_B \quad \forall\, B \in \mathcal{B}(\tau) \end{aligned} \tag{38}$$

where $S_\emptyset = \mathbb{1}_N^T \otimes \mathbb{I}_T$ is a summation matrix summing all the users' actions u. Here S_B are similar summation matrices, each of which encodes a constraint in the set of branching points of the hierarchical structure, $\mathcal{B}(\tau)$. They are defined as:

$$S_B = \left[M_{B,A_j}\right], \quad M_{B,A_j} = \begin{cases} a_{B,j}\mathbb{I}_\mathrm{T}, & \text{if} \quad A_j \in \mathcal{D}(B) \\ 0_T, & \text{otherwise} \end{cases} \tag{39}$$

where $\mathbf{0}_T$ and $\mathbb{I}_\mathrm{T}$ are the zero and identity matrix of size T respectively, $a_{B,j}$ is a weight associated with the j_{th} descendant of branching node B. For instance, for constraining power at a give point of the grid, neglecting power loss, the coefficients $a_{B,j}$ are equal to 1 for the nodes directly connected to branch B and 0 for the other ones.

Despite having a simple formulation, problem (38) is very flexible, allowing having multiple objectives in different branches, or appliances providing possibly competing services to the grid, like primary regulation and dispatchable operations. In paper [21] we showed that, applying a standard ADMM decomposition strategy, the problem (38) can be solved using a forward-backward communication protocol, with an iterative round of message passing spanning the hierarchical structure from the leaf node to the root node and vice versa. The mathematical formulation of the problem is the following:

$$u_i^{k+1} = \underset{u_i}{\text{argmin}}\, f_{c,i}(u_i) + \sum_{B\in\mathcal{A}(A_i)} \frac{1}{2\rho}\| \tag{40}$$

$$= (S_B x^k - y_B^k)/N_B - a_{B,i}u_i^k + u_i + \lambda_B^k\|_2^2 \tag{41}$$

$$y_\emptyset^{k+1} = \mathbf{\Pi}_{\mathcal{Y}_\emptyset}(\mathbf{prox}_{\rho e}(S_\emptyset x^{k+1}\lambda_\emptyset^k)) \tag{42}$$

$$y_B^{k+1} = \mathbf{\Pi}_{\mathcal{Y}_B}(S_B x^{k+1} + \lambda_B^k) \tag{43}$$

$$\lambda_B^{k+1} = \lambda_B^k + \frac{\rho}{N_B}(S_B x^{k+1} - y_B^{k+1}) \tag{44}$$

where N_B is the number of descendants of branch B, y are auxiliary variables used to decompose the problem and to treat the coumpling constraints, $\mathbf{\Pi}_{\mathcal{Y}_B}$ stands for the projection operator over the box constrain set defined by $\mathcal{Y}_B$ and $\mathbf{prox}_{\rho e}$ is the proximal operator for function e, with parameter ρ and $\mathcal{A}(A_i)$ are the ancestors of node A_i. Since the root node update involves the minimization of system-level objective function e, equation 42 projects its proximal minimization into the root node constraint set $\mathcal{Y}_\emptyset$, similarly to proximal gradient methods, as the forward-backward splitting algorithm [22]. As can be seen from equation (41), each agent needs some information from its ancestors in order to solve its optimization problem. This information can be provided by their respective parent nodes. This allows the algorithm to be solved in a forward-backward passage. In the forward passage each branch B sends its reference signal r_B and the one received by its parent to his children, which propagates it downwards through the hierarchy. At the same time, prosumers in leaf nodes solve their optimization problem as soon as they receive their overall reference signal r_i. In the backward passage agents send their solutions to their parents, which collect them and send the aggregated solution upward. Note that r_B contains only aggregated information from branch B, which ensures privacy among prosumers.

4 Blockchain for energy management applications

ECs have an intrinsic decentralized structure, being roughly constituted by a collection of smart meters, able to acquire, process and store data about consumption/production and by the related prosumers. For that reason, a decentralized EC management based on blockchain technology constitutes an interesting solution. Unfortunately, the usage of blockchain technology in the energy sector, and in general in IoT applications, still encounters barriers that can limit its adoption. Among them, currently the main challenges are related to scalability issues and privacy management.

Regarding the former, typically related to IoT applications, the usage of the main public platforms such as Ethereum [23] network, called 1st layer solutions, results in significant requirements in terms of CPUs, memory and disk space for the meters currently available on the market. Moreover, the number of transactions needed to manage an EC is too high and, consequently, the related costs are not economically sustainable. In addition to the economical aspects also the ecological dimension has to be considered: many blockchain solutions uses Proof-of-Work consensus mechanism to validate their blocks, which adds a significant energy footprint to every blockchain transaction, reducing the suitability for energy applications.

Besides these issues, privacy management, i.e. basically the proper handling of private data (e.g. energy consumption) on an immutable ledger like a blockchain, is also a remarkable aspect that has to be taken into account carefully. For example, GDPR [24] regulation, adopted in the EU, poses significant barriers to store private data on blockchain networks.
This chapter describes the developed applications related to specific use cases of flexibility management and energy communities managements. Section 4.1 investigates if and how 1st solutions can be used, typically when a high level of scalability is not required. Section 4.2 focuses on the analysis and the implementation of innovative solutions, called 2nd layer, which can reasonably be used in IoT applications. More in details, section 4.2 describes the scalability aspects, the technological solutions currently available on the market, the selected one and the developed applications based on it. Finally, Section 4.3 describes how privacy, which affects both 1st and 2nd approaches, has been properly handled in the applications. All the developed 2nd layer applications were deployed and tested on real embedded devices installed in the LIC demonstrator, please see Section 8.6.2 for more details.

4.1 First layer applications

4.1.1 NEMoGrid energy market

The first example of first layer application was developed within the NEMoGrid project exploiting data acquired from a Swedish demonstrator, please refer to Section 1.2 for more details.
A one-time flexibility market interacting was designed where a local flexibility service provider commits to offer a peak shaving service for the DSO using a collection of smart contracts, which run on an Ethereum blockchain. In the pilot, the project partners NGENIC[3] and Upplands Energi[4] are respectively the flexibility provider and the DSO.
SUPSI completed the development and the tests of the smart contracts (SCs) and released them as open-source code on the Gitlab platform[5]. Besides, exhaustive documentation about the SCs was published on Readthedocs[6].
Basically, interacting with the SCs, two actors, in the following named *DSO* (the energy provider) and *Player* (the flexibility provider), can play an energy market for a specific period (e.g. a day, a month, a year, etc.), defining together its parameters, among them two thresholds, in the following called *Lower* and *Upper*, related to the maximum peak of consumption for the area in which the player controls the flexibility. According to the reached maximum peak during the period, the actor will receive revenues or penalties. Refer to section 2.2.6 for a detailed description of the considered market mechanism.

Transactions sequence: *DSO* and *Player* can run the energy market related to a given period interacting with the smart contracts with a specific sequence of transactions. Figure 9 shows the sequence for a daily market.

Figure 9 depicts how the sequence is basically divided into two main steps, each of therm constituted by two transactions. In the former before the beginning of the period, *DSO* and *Player* agree about

[3] https://ngenic.se/
[4] https://www.upplandsenergi.se
[5] https://gitlab.com/supsi-dacd-isaac/NEMoGrid-smart-contracts
[6] https://NEMoGrid-smart-contracts.readthedocs.io/en/latest/

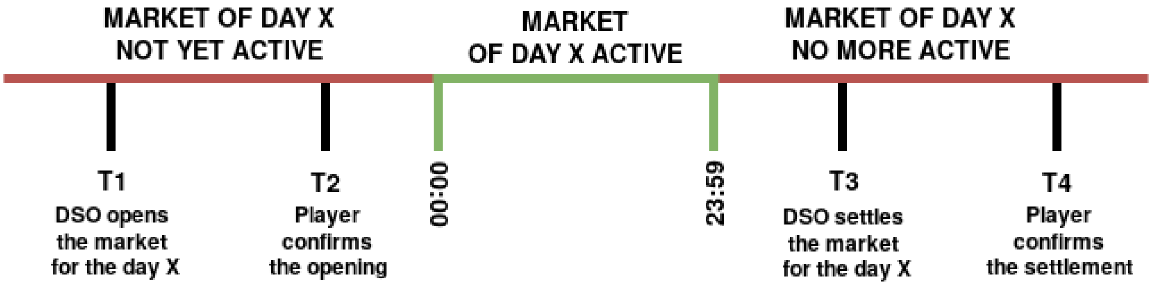

MARKET OF DAY X NOT YET ACTIVE
MARKET OF DAY X ACTIVE
MARKET OF DAY X NO MORE ACTIVE
T1 DSO opens the market for the day X
T2 Player confirms the opening
00:00
23:59
T3 DSO settles the market for the day X
T4 Player confirms the settlement

Figure 9: Transactions sequence for a daily energy market

the playing of an energy market with parameters specific of the considered period. Then, after the end of the period the market is closed by the two actors and the revenues or the penalties are applied. In the market closing transactions, *DSO* save on the blockchain the power peak of *Player*, which confirms the peak with the last transaction (T4). In case the actors do not agree about the market conclusion, i.e. *Player* do not confirm *DSO* settlement, a security mechanism, based on the intervention of a third trusted actor, named *Referee*, has been implemented in order to avoid market manipulations. All the details about the SCs behaviour are entirely described in the online documentation.

Market solving: When the market is correctly settled with the last transaction, as shown in Figure 9, the smart contracts automatically solve the market and distribute the tokens related to revenues or penalties. Figure 10 reports examples of the three different cases that can occur in the management of the flexibility markets. The picture shows the daily power peaks for three different monthly markets (blue, orange and red lines). The blue line show a *Player* able not to reach the *Lower* limit (green horizontal line). Consequently, it will receive the entire amount of revenue. Instead, the red line regards a *Player* that will be punished, having exceeded the *Upper* daily power limit. Instead, the orange line represents the case of a *Player* that will receive only a part of the revenues, being among the two thresholds. Please refer to figure 3 and section 2.2.6 to a more detailed explanation of the mechanism and the lower and upper limits.

Simulator: The developed smart contracts were deployed on the Rinkeby network, a testnet Ethereum blockchain[7] based on Proof-of-Authority consensus algorithm [25]. To interact with the smart contracts performing the transactions mentioned above, an open-source Python script, called *Market Simulator*, was developed. Basically, the software performs transactions with the deployed smart contracts simulating the behaviour of the actors, as shown in Figure 9. The simulator runs daily markets for approximately 8 months, in the period between May 2020 and January 2021 to have a significant number of played energy markets. At the same time, Ngenic developed a REST API to download the consumption data of a set of flexibility providers with a resolution of up to 1 minute to provide real power peaks data usable by *Market Simulator*. Figure 11 shows the power profiles of two providers.

Data available on the Ngenic interface are continuously downloaded and stored in an InfluxDB instance by a Python script, called *Data Collector*. The database engine is installed on a SUPSI application server, together with the aforementioned scripts and the *geth* application [8], which runs the Ethereum node providing the connection to the blockchain. Figure 12 depicts the main interactions between reports the main data flow between *Market Simulator* and *Data Collector* elements.

The list of all the transactions performed by *Market Simulator* are available at the following link: `https://rinkeby.etherscan.io/address/0xb557072d17D9433829cdF9B22f03A2B90402102f`.

[7] https://rinkeby.etherscan.io/
[8] https://github.com/ethereum/go-ethereum

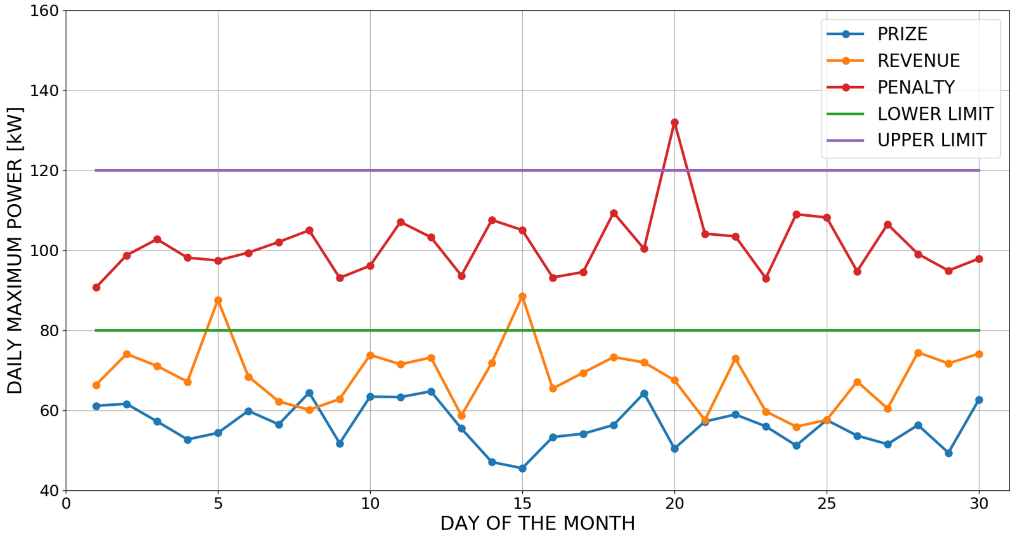


Figure 10: Example of flexibility markets for the Swedish pilot project

4.1.2 Asset Manager (*AM*)

The usage of blockchain technology in application related to assets certification is recognized to be both useful and not affected by heavy transaction costs. A generic example could be a smart contract able to manage the property of an asset. This intrinsic sustainability is mainly due to the fact that the number of transactions in these type of application is typically low. For this reason a collection of smart contracts, in the following called *AM*, were developed and tested aiming to properly manage a shared asset in an energy community. Example of shared assets could be big PV plants or batteries owned by some prosumers of the community. Indeed, some assets related to energy production and storage are too expensive to be owned by a single user and can be maintained by the entire community defining a governance approach. *AM* implements the automatic management of an assets collection for a community, taking into account both the asset governance and its operations (e.g. the revenues generation and distribution). Practically, *AM* provides the community governance using Aragon platform [26]. Aragon develops an open-source platform, named *AragonOS* [27], where different Ethereum applications can work together, similarly to processes in an operating system. Besides, using *AragonOS* a significant collection of applications already developed and tested by Aragon is available. Among them, the *Voting* application [28] provides the functionalities needed to manage the governance via a collection of votings, which can be configured and customized (e.g. majority to reach, quorum, etc.). Thus, the most meaningful operations related to an asset are handled by the governance community using *Voting* and reported in the following list:

- *Asset creation*: a new asset has to be bought and installed
- *Shares initial distribution*: after an asset creation, the shares related to the owners have to be distributed
- *Asset deactivation*: an asset has a problem and is not able to generate revenues
- *Asset activation*: an asset can generate revenues again (e.g. it has been repaired)

Fig. 13 shows the interactions between AM and *Voting* applications via *AragonOS*. It is important to

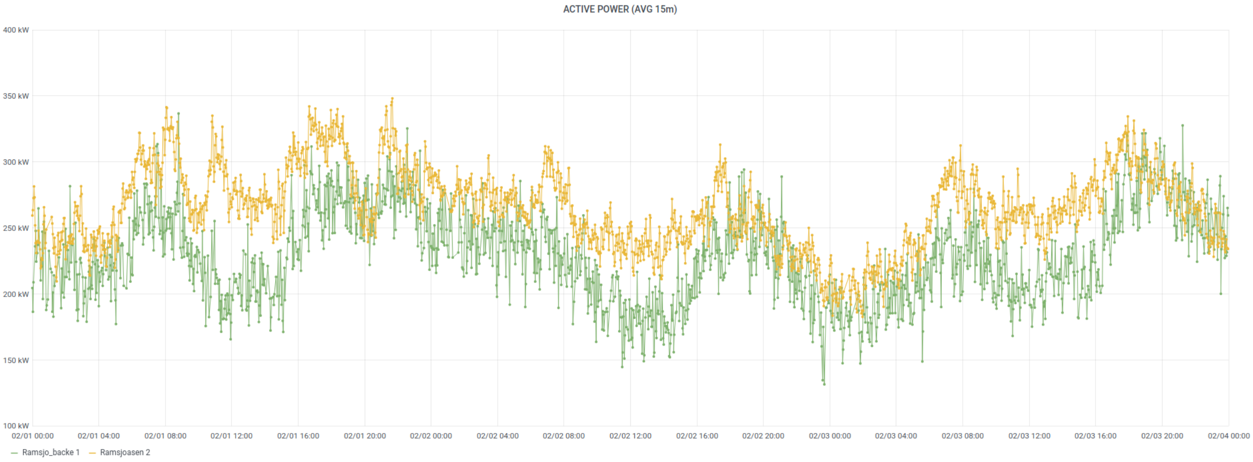


Figure 11: Data acquired in the Swedish pilot related to two *Players*

remark how the unique AM smart contract that works with *AragonOS* is *AssetManager*, described in the next paragraph.

***AM* smart contracts:** *AM* is open-source software, released under MIT license [29] and accessible on Gitlab platform [9]. Besides, a guide describing the code of the smart contracts on Readthedocs [10]. More specifically, *AM* is constituted by the following smart contracts, whose functionalities will be exhaustively described in the next paragraphs:

- *AssetManager*, which manages the operations related to all the assets of a community
- *Asset*, which manages the operations a specific asset
- *SHT*, the acronym stands for *SHare Token* and manages the processes related to the asset ownership using a sharing approach. *SHT* is a token based on ERC20 standard [30].
- *RevenueClaiming*, which operates a revenue distribution
- *RVT*, the acronym stands for *REvenue Token*, an ERC20 token [30] used for the revenue distribution

An exhaustive collection of tests has been developed within the code to ensure a meaningful level of robustness and security of the smart contracts.

Asset management: When community votes in favour of asset creation, a new instance of *Asset* and *SHT* are deployed on the blockchain. In details, an asset is defined in the smart contracts with the following features:

- a numerical *id*, used to identify the asset
- a *name* and a *description*, useful to briefly describe the asset
- a trusted *oracle*, practically a wallet allowed to perform significant operations such as the distribution of the revenues

[9] https://gitlab.com/supsi-dacd-isaac/asset-manager

[10] https://asset-manager.readthedocs.io

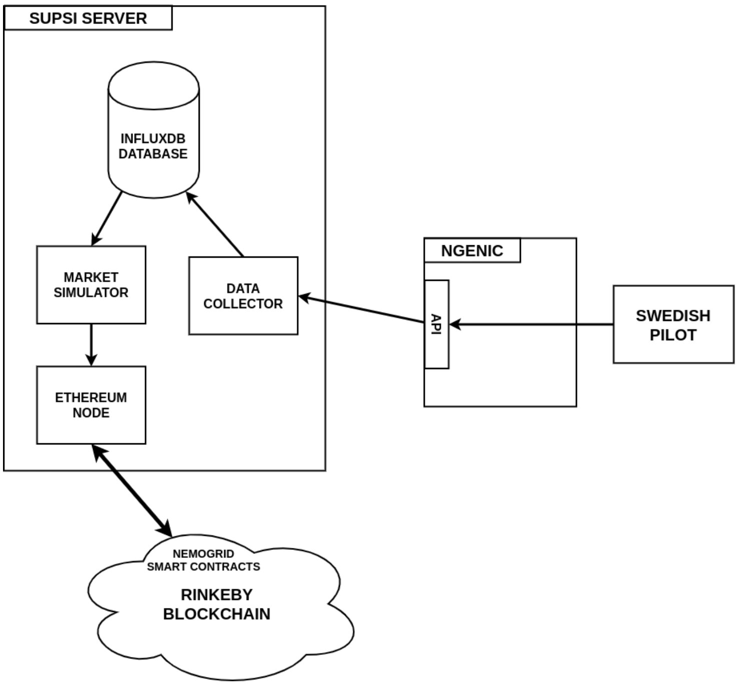


Figure 12: Energy markets playing schema

- the address of its *SHT* contract for the management of the shares, just deployed during the asset creation.
- the address of *RVT* token, which will be used to distribute the revenues.

In order to correctly implement all the operations, such as the revenue distribution, during its life an asset can be in one of the following states:

- `ACTIVE`: when shares can be exchanged, and the shareowners can not claim any revenues
- `FROZEN`: when shares cannot be exchanged, and the shareowners can claim revenues
- `INACTIVE`: when shares cannot be exchanged, and the shareowners can not claim revenues

After the creation, an asset is in `ACTIVE` state, i.e. it is working and creating revenues (e.g. a PV plant is producing energy).

Shares: When an asset is created, the related *SHT* is deployed. Besides, 1000 SHT are minted and assigned to the asset. Thus initially, all shares are owned by the *SHT* smart contract. To complete the asset setup, the community has to vote to decide how the shares will be distributed among the users. The community can determine the shares amounts of every user. Once a user has received the number of its shares, it can exchange them according to ERC-20 standard [30] when the asset is `ACTIVE`.

State transitions: As aforementioned described, three states are available for an asset. Figure 14 depicts the transitions with the related Solidity functions. It is important to remark that `activateAsset()` and `deactivateAsset()`, pertinent to `ACTIVE <-> INACTIVE` transitions, have to be performed by the community with a vote. Instead, `setRevenue()` and `defrostAsset()` can be performed only by the trusted oracle described in Chapter 4.1.2.

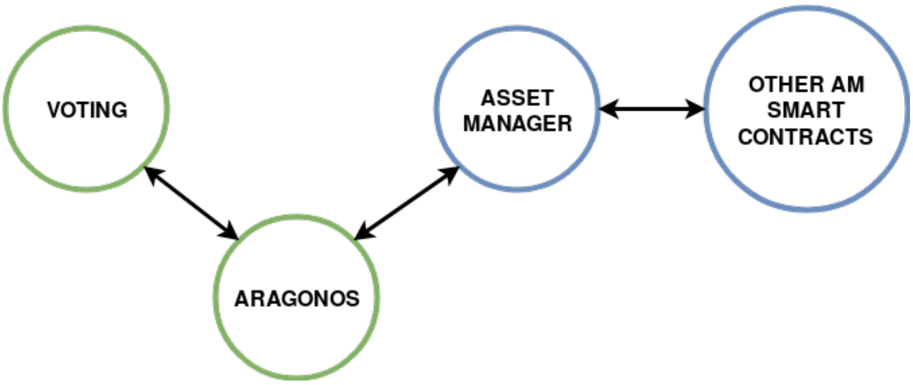


Figure 13: Interactions between *Voting* and *AssetManager* using *AragonOS*. The green circles refer to smart contracts provided by Aragon.

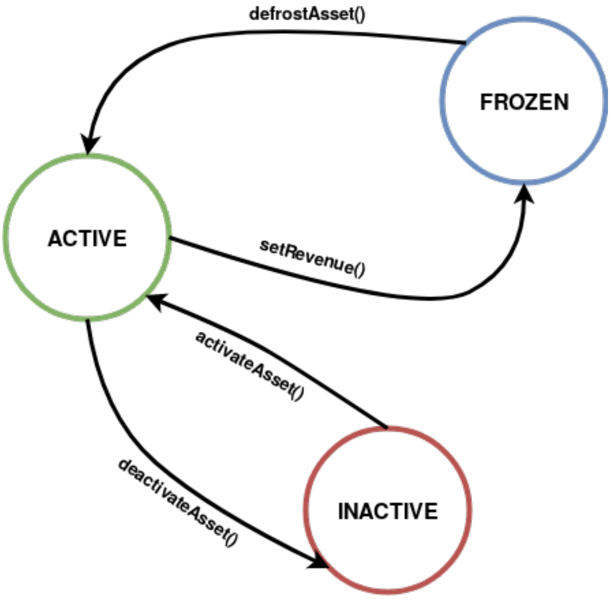


Figure 14: State transitions of an asset

When the asset state is *ACTIVE* a revenue can be set by the asset trusted oracle using the `setRevenue()` functions. After this transaction, the asset state becomes `FROZEN`, and a certain amount of *RVT* tokens are staked in the *Asset* smart contract. Besides, a new *RevenueClaiming* contract is deployed on the blockchain. It contains all the information related to the revenue (e.g. the *RVT* amount, the shareowners having already claimed the income, etc.). When the state is `FROZEN`, no shares can be transferred to avoid cheatings due to possible double-spending. Besides, each shareowner is allowed to claim its revenue portion, i.e. a part of RVT staking proportional to owned shares, using the `claim()` function. After a certain amount of time, reasonably long to assure an easy claim to the shareowners, the asset state becomes `ACTIVE` again, thanks to `defrostAsset()` function performed by the trusted oracle. Figure 15 shows the timeline of a revenue life: the initial setting, the claims when the asset is `FROZEN`, and the defrosting. Besides, when an asset is `FROZEN` no new revenues can be created. Consequently, the life of revenue currently claimable has to end with a `defrostAsset()` transaction before having a new one.

If an asset is unable to create new revenues (e.g. a PV plant has an electrical problem and needs to be repaired to produce energy again), the community has to vote to set the state to `INACTIVE` using the function `deactivateAsset()`. When an asset is `INACTIVE`, its shares are locked, and no new revenues can be generated. If and when the asset will be able again to generate new revenues (i.e. a PV plant

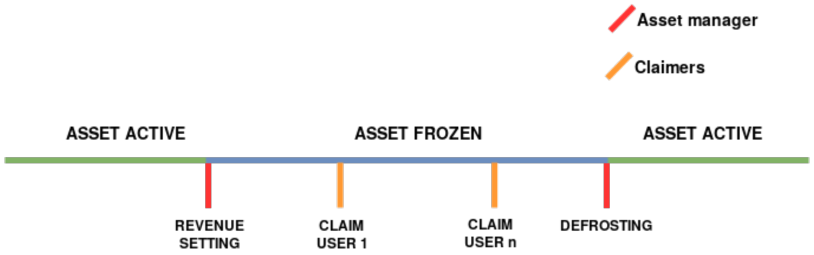


Figure 15: Timeline of a revenue life

has been repaired), the community can vote to reactivate it using the function `activateAsset()`.

4.2 Second layer applications

The solutions presented in Section 4.1 are fully based on smart contracts. After their analysis, we concluded that running an energy market purely based on smart contracts is not practical with current blockchain technology. In general, this aspect strongly depends on the application to develop. For example, *AM*, described in Section 4.1.2, typically requires few transactions to operate. Consequently, it can also be used for the management of assets belonging to an energy community. On the other hand, storing energy data on the blockchain as in the application described in Section 4.1.1 can be very expensive, depending on the value of *ether*, the Ethereum token. In addition, in applications where transactions have to be performed frequently the scalability is currently a significant barrier in adopting blockchain. This is not the case of the application described in Section 4.1.1, where few periodic transactions are needed. Still, in other use cases it is an aspect to take carefully into account. In the last years, different solutions that allow combining blockchain safety with fast and cheap transactions are raised to front this scalability issue. The next sections describe in details the scalability barrier and the developed solutions.

4.2.1 Blockchain scalability issues

The acceptance of cryptocurrencies is a major driver of the demand for blockchain technology. The technology allows people to safely transact and carry out various deals while remaining anonymous. This makes blockchains a fast-growing financial technology but at the same time its adoption entails various problems. Among them, scalability is certainly one of the most significant challenges, especially for the usage of blockchain within IoT applications.
Scalability on a blockchain is determined by the number of transactions it can complete in a second [31]. To scale up effectively, a blockchain would need to perform thousands of transactions per second (TPS). Yet, the Bitcoin blockchain can only handle about 5 to 7 TPS while the Ethereum can handle 10 to 25 TPS [32]. This is a sharp contrast to systems such as VISA that handle 1500 to 2000 TPS [33]. However, straightforward scaling up of the blockchain interferes with its integral security and decentralization attributes. In simple terms, scaling up the blockchain means enabling it to process transactions faster. To achieve this, the same system would either need to simplify its security measures or decentralization capacity.
Scalability, security and decentralization are the three main pillars that affect the mass adoption of blockchain technology. Decentralization and security are its key selling points. Yet, the market cannot meet its current and future demand without sustainable scalability. A strong structural correlation exists between these three pillars. This situation is commonly referred to as the ‘Blockchain Scalability Trilemma’ as described by Vitalik Buterin, the founder of Ethereum [34] [35]. According to Buterin, you cannot improve all the pillars of an individual blockchain system together. The system limits you to improve only one or two pillars at the expense of the third. Developers found that altering the parameters

of one pillar would in turn compromise the other two pillars of the system. The three scenarios below illustrate the blockchain scalability trilemma:

- Improving Decentralization and Security protocols reduces Scalability options
- Improving Scalability and Security protocols reduces Decentralization options
- Improving Scalability and Decentralization protocols reduces Security options

Several solutions have been proposed to increase the solutions scalability in order to have transactions both faster to be performed and more economically sustainable. This fact would improve adoption of blockchain in a wide variety of applications, among them the data management in the energy sector.
A promising approach to front the scalability problem in a blockchain is basically to move the issue to a second layer, the main blockchain itself being the first one. Currently, the available 2nd layer solutions are essentially based on two technologies: state channels and sidechains. In both the cases, the most of the user interactions is moved out of the blockchain, on the second layer, where fast, safe and cheap transactions between participants can take place.

4.2.2 State channels:

State channels technology provides secure P2P channels that can be used by two nodes connected to a blockchain to exchange data avoiding to do transactions on the main network. The only transactions performed on-chain are the opening and closing transactions, which are needed to establishes and to settle the state channel. The main process downside is that it requires the full availability of the two involved participants; otherwise the security of the exchanged data, typically tokens, can be compromised. As a consequence, state channels are substantially useful in cases where nodes exchange many state updates over a long period, to mitigate the costs of creating and settling a channel. For this reason, probably the payment channel is the typical use case where this technology is mostly used. The following list reports promising projects that provide state channels implementation:

- Lightning Network [36]
- Raiden Network [37]
- Liquidity [38]
- Celer [39]

The first three projects are among the first to provide *state channels* platforms. More precisely, they are projected to provide payment channels in order to transfer tokens in the channels. Instead, Celer project aims to manage exchanging of any type of data, not only tokens.
State channels is undoubtedly a meaningful 2nd layer technology, but it appears not to be functional for the needed usage. Indeed, in an energy community it would be advantageous to provide a light and cheap blockchain usable by all the users instead of having a huge quantity of channels connecting couples of nodes. For example, the information related to nodes consumption and production could be accessible by all the nodes with different level of allowances, not only by the two nodes creating a channel. Indeed, for what concerns our discussion, the unique use case where a *state channels* solution like the one provided by Celer project [39] could be exploited, is an energy community composed only by two nodes.
For the reasons mentioned above, *state channels* technology was not considered as suitable for the management of 2nd layer blockchain solutions in an energy community.

4.2.3 Sidechains:

Sidechains are separate blockchain networks, able to interact with the main chain. They have their own consensus mechanism, level of security, and tokens and can be public or private. When sidechain security is compromised, the damage does not affect the other connected networks. Moreover, two sidechains connected together can transfer any data. For example, tokens can be exchanged at a predetermined rate between the main chain and the sidechain. Basically, the main chain should provide the security of the entire ecosystem, while the transactions outsourced to the sidechain can sacrifice decentralization in return for scalability and velocity.

As opposed to state channels, transactions that occur on a sidechain are not private between the participants of a transaction. They are published on the sidechain network and thus visible to anyone who has access to the ledger. Moreover, the chain nodes do not need to be always available.

The main drawback of this technology is that setting up a sidechain is a significant effort, as it means to build the entire platform from scratch.

This solution appears to be extremely suitable to be used in the data management of energy communities. Indeed, custom chains could also be tailored for embedded devices such as meters and used to store data about nodes and interact with other main blockchains.

For these reasons, the *sidechains* approach was chosen to be implemented. The project chosen for the effective implementation is Cosmos [40], one of the most notable platforms currently available. Cosmos was chosen because it provides significant warranties in terms of documentation, application to develop sidechains and strategy to create networks of chains.

4.2.4 Cosmos and Tendermint

Cosmos [40] provides a complete environment to create custom side chains, which can be used as meaningful solutions for decentralized management of energy communities. Under a generic point of view, a Cosmos network is a collection of independent and interconnected custom blockchains, which run in parallel. Each blockchain is powered by Tendermint consensus algorithm, which provides a byzantine-fault-tolerant mechanism [41].

Theoretically, the creation of a blockchain is based on the development of its three main layers: *Networking*, *Consensus* and *Application*. In a blockchain ecosystem based on smart-contracts like Ethereum, an application layer is provided, named Ethereum Virtual Machine (EVM), and is used to deploy on the chain smart contracts with custom functionalities. This approach has many advantages, but it is based on a monolithic platform where the three layers cannot work separately.

The solution proposed by Cosmos is more modular: it is a solution that packages the networking and consensus layers of a blockchain into a generic engine, allowing developers to focus on application development. The communication with the *Application* layer is provided by the ABCI protocol. Figure 16 shows the structure of a typical Tendermint application.

As explained above, Tendermint is substantially a consensus/networking algorithm that can be easily configured and be used together with Cosmos SDK to create a custom side chain to manage a EC. Moreover, the custom sidechain can be integrated inside the Cosmos Network using the IBC protocol and interacting with other blockchains, typically exchanging data, as shown in Figure 17. The next paragraphs describe in details all the main elements that constitute the Cosmos framework.

It is important to remark how the decentralized approach of Cosmos can be beneficial in the management of energy communities. Indeed, some chains can be demanded for specific purposes, such as the consumption and production measurements, which are implicitly customized for the single community. Instead, other chains can be used for more general tasks (e.g. tokens management, community governance, etc.).

***Consensus* layer:** *Consensus* layer is the part of a Tendermint application to whom the management of the consensus in the blockchain is demanded. Currently, two mechanisms are implemented and usable in Tendermint: *Proof-of-Authority* (PoA) [25] and *Proof-of-Stake (PoS) [42]*. These approaches are

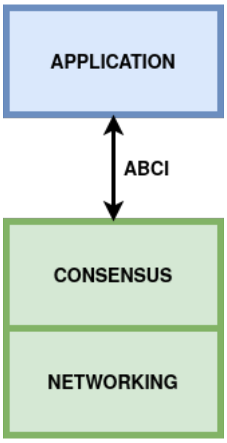


Figure 16: Tendermint application structure

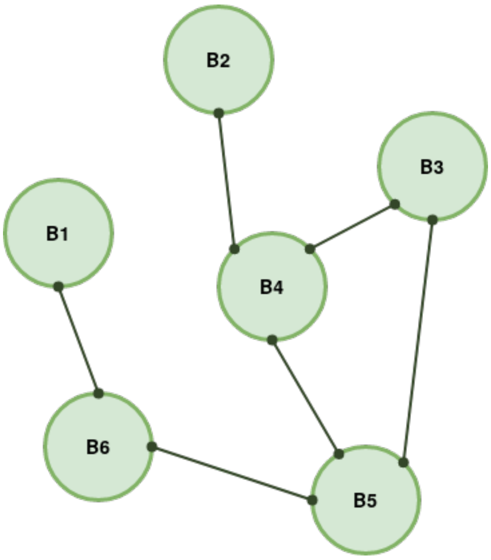


Figure 17: Tendermint blockchains in Cosmos network connected via IBC protocol

very sustainable under the computational point of view, unlike mechanisms based on Proof-of-Work[5], still broadly used in blockchain-based solutions.

It is remarkable to note that the sustainability of PoA and PoS has two meaningful consequences. The former is that embedded devices such as the smart meters have sufficient resources to run the Tendermint applications. Moreover, the energy consumed by these two mechanisms is negligible if compared to PoW.

Both PoA and PoS were positively tested on the LIC pilot (see Section 8.6.2 for details), which is basically an energy community composed of 20 nodes. Thus, on the pilot a sidechain of 20 nodes was deployed. In a first step a chain based on PoA, simpler to configure than PoS, was implemented and deployed since June 2020 until the end of the year. Instead, since the beginning of 2021 PoS was positively tested in a new sidechain, which is still running in the pilot nodes and will be used for new developments. It is important to remark how the consensus mechanism choice entails no consequences for the integration in the Cosmos platform. Indeed, the blockchains shown in Fig. 17 can communicate via IBC also if they use different consensus algorithms.

The consensus in a Tendermint network is effectively managed by a subset of the nodes, named *validators*. They are the unique nodes allowed both to create new blocks and to decide the transactions to insert in a block. Consequently, no new blocks can be created if all the validators are not connected to the network (e.g. for a power outage). In order to minimize the probability that this event occurs, the validators have to be chosen carefully depending on the installation of the nodes. For example, a good candidate to be a validator should have both a stable Internet connection and a safe power supply system.

***Networking* layer:** *Networking* layer manages the connections between the nodes of Tendermint application taking care to properly transfer all over the network the information stored in the sidechain via transactions.
The links between the nodes are managed by a P2P protocol, which is secured by the usage of private and public keys for the identification. The protocol used is TCP. Upon a nodes establishes a successful TCP connection with a peer, two handshakes are performed to empower the chain security: the former for the encrypted authentication, the latter in order to check the Tendermint versioning. Thus, this layer provides a fast, secure and configurable platform that manages the network connections in a Tendermint application.
It is important to note how Tendermint provides the feature to configure persistent peers. These are intended to be trusted peers that can be very helpful to define a stable P2P structure. To empower the stability in the community side chain, in the pilot where a side chain has been deployed (see Paragraph 4.2.5 for details) it was decided to configure a collection of persistent peers able to connect all the nodes in the network safely. This decision was taken considering how meters are usually installed in cabinets where the data connectivity can be affected by local interferences. Consequently, a P2P structure taking into account the different installations can be useful to have a more stable network.

***Application* layer** The previous paragraphs describe how the consensus and networking tasks, mandatory in any blockchain-based solution, are already provided by Tendermint code. It is possible to configure the software (e.g. setting PoA or PoS, defining the persistent peers, etc.), but no changes to the code are needed. Instead, the *Application* layer is the custom application part that has to be developed. Regarding the used technologies, all the code was developed in Go language [43] because Cosmos and Tendermint platforms are based on this programming language. Consequently, all the official documentation and the available examples are based on this programming language and so the development of custom Go applications was strongly facilitated.
The main aims of a custom *Application* layer are substantially two: the former is to provide communication with the other layers via ABCI, the latter is to interact with off-chain applications that try to perform transactions and queries via a REST API. Figure 18 shows the interactions between the aforementioned components in a chain of two nodes.

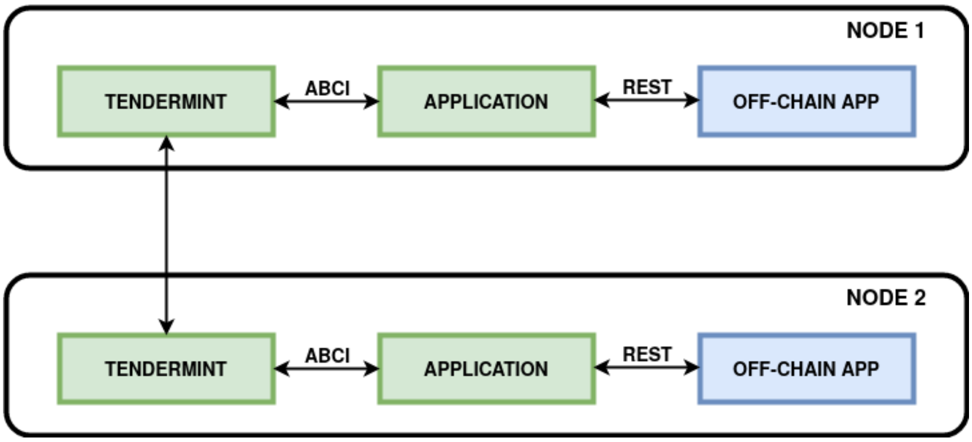


Figure 18: *Application* layer interactions in a chain constituted by two nodes

Figure 18 shows how ABCI manages the interactions between the custom applications with the other layers. Basically, it consists of a collection of Go methods that are suitable to properly operates the chain. For example, *CheckTx()* function is used to check if a transaction has to be rejected or not, performing the filtering feature mentioned above.

The second aim of the *Application* layer is to provide an interface for off-chain software that has to exchange information with the sidechain. This interaction is provided by a REST interface that allows the performing of query and transactions. The usage of a REST API has the significant consequence that the off-chain applications can be developed in any programming language, not necessarily in Go. Acting this way, specific functionalities, which can be more easily developed in other languages such as Python, are included in the off-chain part, demanding to the *Application* layer only the functionalities directly related to the chain interactions. An example can be the data gathering from a smart meter that has to be periodically stored in the chain and can be easily performed by dedicated Python modules provided by the manufacturer of the meters. The REST provides two API services: the former for the transactions, which can result in a status change of the chain, the latter related to queries that are used to read data from the chain.

Under a logical point of view, it is simple to note how the *Application* layer can act as a custom filter and an adapter between the off-chain parts and the side chain, managed by Tendermint consensus/networking layers. The first task is fundamentally used to check the nodes allowance when they attempt to access to the chain with a transaction or a query. Typically, the nodes in a chain can have different access rights, which can be managed by the *Application* layer. In addition to the filtering task, this layer can be used to transform the information to store in the blockchain when custom transactions occur. Figure 19 shows an example of how filtering and adapting functionalities can operate.

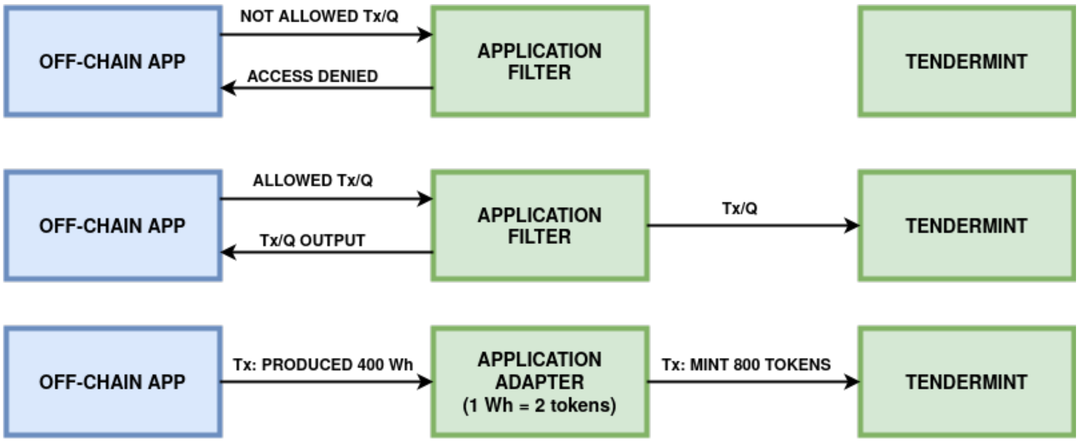


Figure 19: *Application* layer filtering and adapting functionalities

Cosmos SDK: Cosmos layers provide an exhaustive platform that comprehends all needed functionalities needed to build a custom blockchain. Anyway, many essential aspects related to blockchain-solutions are still missing. For example, a custom Tendermint application is not able to interact with other Cosmos blockchains. Moreover, fundamental features such as the accounts and tokens managements must be included in the code of *Application* layer.

Cosmos SDK [11] is the platform that provides all these missing functionalities. Its main goal is to define an ecosystem where developers can easily create custom blockchains natively able to interoperate with other Cosmos chain. The potential inside this solution is extremely powerful, where it is possible to develop a structure based on multiple networks working in parallel and able to communicate together, each of them demanded to specific operations. Fig. 20 shows a simple example of this networked approach; there are three PoA blockchains, named BA1, BA2 and BA3. Each network manages a single energy independent community where tokens are burnt according to the consumed energy. Moreover, there is a PoS network (BS), which is demanded to the token minting and exchange between BS and the other PoA chains.

[11] https://github.com/cosmos/cosmos-sdk

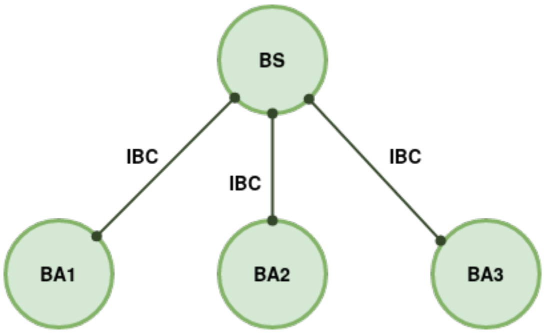


Figure 20: Cosmos blockchains connected via IBC

Figure 20 shows how the data exchanging among the nodes is guaranteed by IBC, which allows Cosmos blockchains to transfer tokens and data to each other. Consequently, side chains related to different applications are interoperable. The typical usage is the tokens exchanging, where IBC allows public and private blockchains to transfer tokens to each other as shown in Figure 20.
The integration of the developed applications inside the Cosmos network via IBC is not strictly correlated to the analysis of 2nd layer blockchain solutions. However, it is currently under development an integration of the applications described in Sections 4.2.6, 4.2.7 and 4.2.8 based on Cosmos SDK and IBC. It will be reasonably deployed on the LIC demonstrator by the end of the year.

4.2.5 Sidechain deployed in LIC demonstrator

Various Cosmos sidechains were deployed and are continuously tested on the embedded devices installed on the LIC pilot (see Section 8.6.2 for details). Figure 21 shows the structure of one deployed network. Each node corresponds to a Strato, with the exception of *nuc01*, which is the NUC board (please see Sections 6.1.2 and 6.1.3 for details). In order to enhance the security, the communication among the nodes is guaranteed by a VPN provided by SUPSI (for more details please refer to Section 7.4.2).
In Figure 21 the light blue circles represent nodes without specific permissions, instead the orange ones are the chain validators (see Paragraph 4.2.4 for details), i.e. the unique elements in the network allowed to create new blocks. These nodes have a relevant role inside the network. Indeed, if all the validators are offline (e.g. for a power outage, or a connectivity problem), then no block will be created anymore. To minimize these events, a collection of 4 nodes have been configured as validators. Moreover, two validators, i.e. *nuc01* and *53932800*, are installed in the Lugaggia kindergarten where power outages and connectivity failures should never occur.
Regarding the peering (see Paragraph 4.2.4 for details), Figure 21 shows all the defined connections. More precisely, the arrow *39189325-39189321* means that node *39189325* is a peer of *39189321*. To enhance the robustness of the network and consequently minimize the probability that a node has no available peering, each node always has three peers, with the exceptions of nodes in the corners of Figure 21, which have only two peers.

The applications described in the following Sections 4.2.6, 4.2.7 and 4.2.8 use the sidechain reported in Figure 21 as a distributed ledger to store on it the information needed by its operations.

4.2.6 *Metering*

The *Metering* application was the first to be developed and, as a consequence, its functionalities are quite simple. Basically, every quarter of hour it saves data about average consumption and production of the last 15 minutes. The datasets are saved in plain text and each meter is allowed to read data related to other nodes. Consequently, every 15 minutes the instances of *Metering* running in parallel on the nodes shown in Figure 21 perform 20 new transactions and save the related data on the chain.

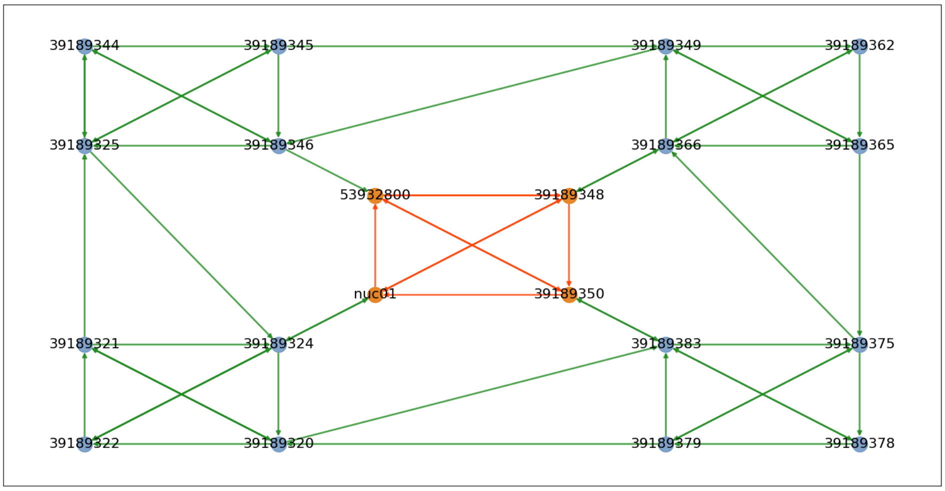


Figure 21: Example of sidechain deployed in LIC pilot

Figure 22 shows the energy consumption and production stored in the LIC sidechain. The green nodes correspond to producer, whereas the red ones to the consumers.

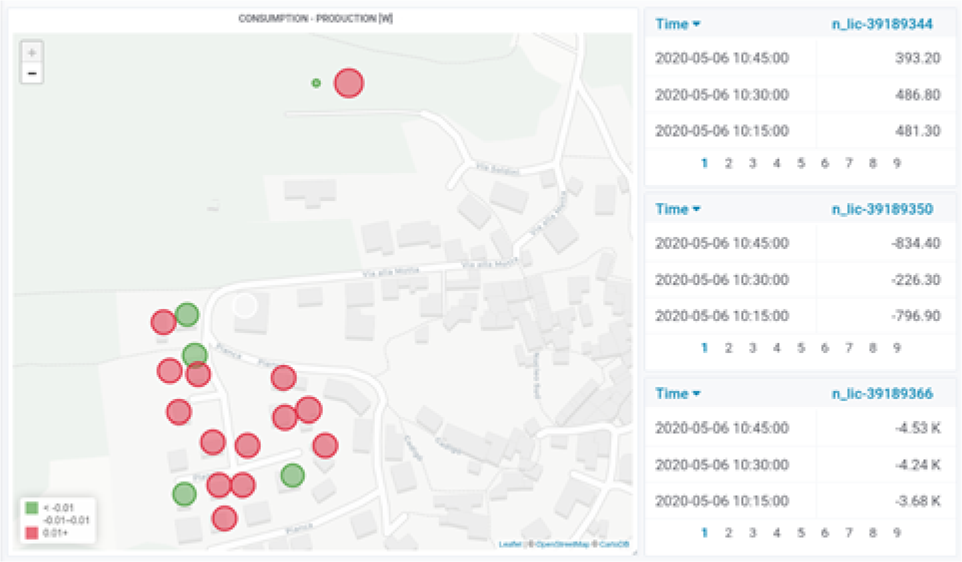
4.2.7 *Auditable Tariffs (AT)*

AT application provides an efficient approach to make the payment verifiable and the aggregator auditable by the users, preserving their privacy. In *AT* each node saves encrypted data about consumption and production on the Lugaggia chain. The data encoding is based on homomorphic encryption, which is locally provided by the *TrustedSum* Go application running on the Strato device (see Sections 6.1.2, 7.4.1 and 6.2 for details). *TrustedSum* code is released as open-source on the Gitlab platform[12]. Thanks to this type of data encoding, a node is not able to decrypt the information related to the other meters, i.e. privacy is preserved. Moreover, it can calculate the consumption/production of the entire community, and consequently verifies the payment validity, which is function of the community production.
AT algorithm is based on the trusted voting system described in [44]. Figure 23 shows the problem: Alice, Bob, Carol, Dave and Eve can vote in an election, but keep their votes secret. Besides, none of them trust Trent to count the votes. In [45] the authors shows how, with a proper blockchain-based approach, there is no need for a trust infrastructure and voters privacy is preserved. In addition, a zero-knowledge proof approach based on the Schnorr proof [46] and the Fiat-Shamir heuristic [47] can effectively perform a decentralized voting system. Figure 24 reports the operations sequence related to the trusted voting algorithm. As first step, each voters i registers its voting key with the following equation:

$$key_i = g^x \pmod{p} \tag{45}$$

where g and p are prime numbers with a size of 1024 bits chosen equal for all the participants, while x is a secret value chosen at random by the ith agent. Once the voters are registered, their voting keys are published on the sidechain. Next each voter calculates a y_i value as follows:

[12] https://gitlab.com/supsi-dacd-isaac/trusted-sum



Time ▾	n_lic-39189344
2020-05-06 10:45:00	393.20
2020-05-06 10:30:00	486.80
2020-05-06 10:15:00	481.30

1 2 3 4 5 6 7 8 9

Time ▾	n_lic-39189350
2020-05-06 10:45:00	-834.40
2020-05-06 10:30:00	-226.30
2020-05-06 10:15:00	-796.90

1 2 3 4 5 6 7 8 9

Time ▾	n_lic-39189366
2020-05-06 10:45:00	-4.53 K
2020-05-06 10:30:00	-4.24 K
2020-05-06 10:15:00	-3.68 K

1 2 3 4 5 6 7 8 9

Figure 22: LIC data saved on sidechain by *Metering* application

$$y_i = \prod_{j=1}^{i-1} g^{x_j} / \prod_{j=i+1}^{n} g^{x_j} \tag{46}$$

Then, each voter can compute and share a vote v_i and its associated auditable hash:

$$\begin{aligned} v_i &= g^{x_i y_i} g^{v_i} \pmod{p} \\ hash_i &= H(g^{x_i y_i} g^{v_i}) \pmod{p} \end{aligned} \tag{47}$$

Now, it can be shown that if we define y_i as in (46), the following equivalence holds [48]:

$$\prod_{i=1}^{n} g^{x_i y_i} \pmod{p} = 1 \tag{48}$$

Thanks to (48), all the voters can retrieve the sum of all the votes without knowing the other's voters values with:

$$\begin{aligned} tally &= \prod_{i=1}^{n} g^{x_i y_i} g^{v_i} \pmod{p} \\ &= g^{\sum v_i} \pmod{p} \end{aligned} \tag{49}$$

Known $g^{\sum v_i} \pmod{p}$, it is finally possible for the voters to retrieve $\sum v_i$ through a grid search.

It is easy to note how the aforementioned voting system shown in Figures 23 and 24 can be easily applied in the *AT* case. Indeed, *AT* can be seen as a "weighted" voting system, where the weight of each user depends on its energy consumption. Thus, the voting system can be used to implement a trusted certification algorithm of the community energy consumption and, as a consequence, of the applied tariffs. Under an operational point of view, every 15 minute *AT* performs the sequence described in algorithm 2 interacting with *TrustedSum* application on each Strato device installed in LIC pilot (see Sections 6.1.2, 7.4.1 and 6.2 for details).

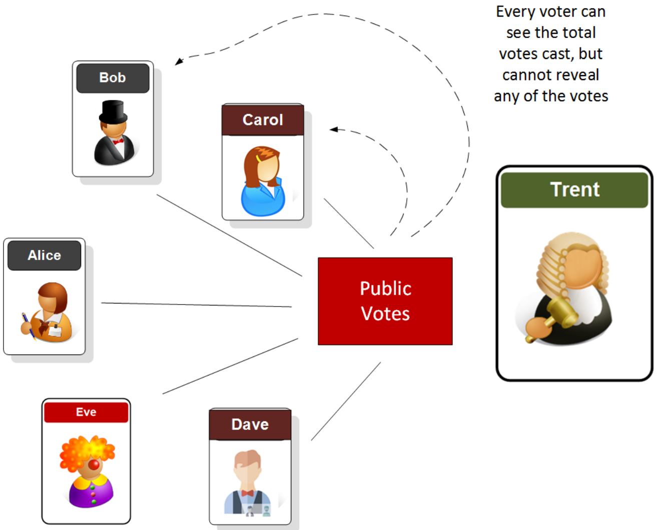

Every voter can see the total votes cast, but cannot reveal any of the votes
Bob
Carol
Alice
Public Votes
Trent
Eve
Dave

Figure 23: Trusted vote approach

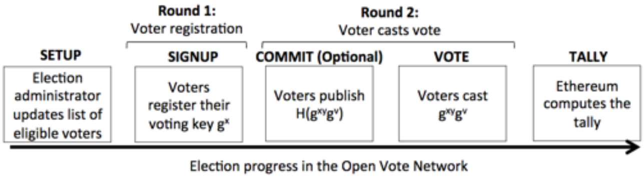

Round 1:
Voter registration
Round 2:
Voter casts vote
SETUP
SIGNUP
COMMIT (Optional)
VOTE
TALLY
Election administrator updates list of eligible voters
Voters register their voting key g^x
Voters publish $H(g^{xy}g^v)$
Voters cast $g^{xy}g^v$
Ethereum computes the tally
Election progress in the Open Vote Network

Figure 24: Trusted vote sequence

Algorithm 2 AT operations sequence

minute T=0,15,30,45
T+1: Registration
Each node saves on-chain the registration string and off-chain locally the related key
T+2: Encoding
Using the registrations saved by all the nodes at minute *T+1* and its key (saved off-chain), each node encodes the average consumption and production of the previous quarter of hour (e.g. T=15: data about [00:00-14:59] will be encoded). The encoded value is saved on-chain by all the nodes.
≥T+3: Sum and decoding
Using the encoded values saved on chain at minute *T+2* each node can calculate the encrypted sum of the entire community and then, applying the homomorphic decoding, obtains the plaint text value.

Figure 25 reports the data flow showing how *AT* works in a simplified chain of three nodes. They save on the chain encoded values, as explained in 2 (orange arrows). Then a generic node (*N* can be *A*, *B* or *C*) is able to sum the encoded values and then decode the results obtaining the plain text total consumption or production (light blue arrow).

The data flow shown in Figure 25 requires significant resources in terms of computational power and memory on the Strato boards. This is mainly due to the mathematical operations used by the homomorphic encryption and described in [44]. For this reason, a custom Go application was developed to run locally on the embedded devices the operations explained in Algorithm 2. On a Strato *AT* needs less than a minute to decode and find the plain test solution, instead on the much more powerful NUC it converges to the solution approximately in 5 seconds. It is meaningful to note how this amount of time is independent of the nodes number, being the most of the resources used for the final decryption

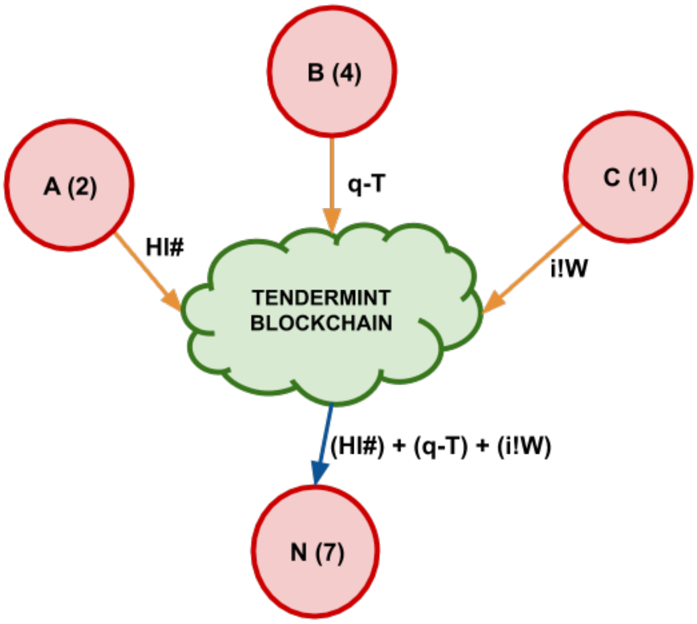


Figure 25: AT data flow

of the encoded sum. Consequently, the application scalability is guaranteed towards the community dimension.

4.2.8 *Pre-paid scenario (PS)*

Similarly to *Metering*, *PS* saves information about production and consumption on the chain. *PS* is released as open-source on the Github platform[13] [14]. The main difference compared to *Metering* is that the energy data are completely tokenized using the official library provided by Cosmos (See Section 4.2.4 for details). According to energy produced or consumed by a meter named *M* in a defined period, *M* mints or burns a corresponding amount of tokens.
To have *PS* properly functioning, periodically (e.g. monthly) the community administrator saves on the chain the settings that will regulate the application. They are reported in table 3.

convfactor_cons	Consumed energy [Wh] to tokens to burn
convfactor_prod	Produced energy [Wh] to tokens to mint

Table 3: Main parameters used in PS application

Every 15 minutes, *PS* tokenizes the consumed and produced energies related to the last quarter of an hour. Consequently, the energy values are transformed in tokens amoelated both to the energy consumption/productions and the power peak. Algorithm 3 reports in details how the energy is tokenized by PS.

[13] https://github.com/supsi-dacd-isaac/cosmos-apps
[14] https://github.com/supsi-dacd-isaac/cosmos-apps-handlers

Algorithm 3 PS operations sequence on a generic meter *M*

```
Get data about energy produced (Ep) and consumed Ec in the last 15 minutes
QUERY: Read PS parameters
QUERY: Read M tokens balance
newBalance = balance
bonus = convfactor_prod * Ep
penalty = convfactor_cons * Ec
newBalance = newBalance + bonus - penalty
TRANSACTION: Set the new token balance of M
```

In Section 4.2.10 the roadmap for the applications integration into the Cosmos Hub network [49] is explained. It is important to remark how *PS* is probably the more interesting example to be integrated. Indeed, its tokenization allows the exchange of tokens related to energy production and consumption among different blockchains.

4.2.9 Applications sustainability

The applications presented in Sections 4.2.6, 4.2.7 and 4.2.8 run in the Strato devices, which compose the Cosmos sidechain in the Lugaggia pilot. To analyze their sustainability, it has been properly tested both in a laboratory and on the demonstrator. More precisely, on Lugaggia Strato a 20-nodes sidechain was continuously tested during the period between June and September 2020 managing the transactions performed by *Metering* and *Auditable tariffs* every 15 minutes. Table 4 summarizes the information about the actuated transactions on the chain and the results in terms of resources usage.

Nodes composing the chain	20
Time resolution	15 minutes
Transactions/15m (*Metering*)	1
Transactions/15m (*Auditable tariffs*)	3
Transactions/1d for a node	384
Transactions/1d	7680
Load average (15 minutes)	0.3
CPU usage	5%
Memory usage	77 MB
Disk usage per day (community)	55 MB
Disk usage per day (single node)	2.7 MB
Disk usage per year (community)	20 GB
Network usage per day (download)	118 MB
Maximum reached data speed (download)	20.5 KB/s
Network usage per day (upload)	125 MB
Maximum reached data speed (upload)	19.7 KB/s

Table 4: Summary about tests performed on LIC nodes

The results reported in Table 4 shows how the sustainability of the Cosmos chain was always guaranteed during the testing period. As regards the computational power and memory usage, Strato devices can easily manage the applications. Moreover, the difference between validator and not-validator nodes resulted in being negligible. An aspect that has to be taken into account with special importance is disk usage. Indeed, 4 MB of space is daily used in the chain to store the data related to one node. Thus, the network cardinality has to be considered to assure safe data storing, especially if the number of transactions is much higher than the ones involved in *Metering* and *Auditable tariffs*. However, the availability of solutions with more and more capacious disks continues to increase. Consequently, also this hardware constraint can be reasonably managed with a proper preliminary analysis.

Regarding the network usage, *Networking* layer of Tendermint daily needs approximately 125 MB of data for the data download, 121 MB for the upload. Furthermore, the data speed of the applications never overcomes 20 KB/s. These network requirements can be easily provided by standard connectivity services (e.g. wired connections, wi-fi). Moreover, also typical mobile subscriptions available on the market can be used for the data exchanging in the network as in Lugaggia.
Currently, different sidechain configurations are continuing to run the developed applications on Lugaggia embedded devices. Nevertheless, now we are focusing our development not on testing robustness and sustainability but on implementing new features, such as the integration in the Cosmos network, described in Section 4.2.10. If new developments entail possible problems in terms of sustainability, additional long-term test like the one presented in this section will be performed.

4.2.10 Applications integration

The applications presented in Sections 4.2.6, 4.2.7 and 4.2.8 show examples of EC managements based on the sidechain shown in Figure 21, constituted of LIC nodes. The performed tests have shown how this solution is sustainable and scalable but still needs to be integrated into a greater framework to be effectively usable. For example, *Pre-paid scenario*, described in Section 4.2.8, performs a real energy tokenization, but it needs to be connected with other blockchains to be able to exchange the tokens on the Cosmos framework. Thus, in the next months a new version of the application running on LIC sidechain will be integrated into the Cosmos Hub [49], the framework of the Cosmos project aiming to create a working network of interconnected blockchains and sidechains. The integration will be based on IBC protocol, which first stable version has been released in the second part of March 2021.

4.3 Privacy aspects

Blockchain poses significant issues in terms of privacy users. This section analyzes these aspects taking into account the GDPR [24] legal framework as reference. Not being a member of EU, currently Switzerland does not follow GDPR, but it is expected that in the next years a similar regulations will be approved also in the Swiss territory. The privacy issues related to the usage of blockchain are essentially due to the following implicit features of the technology: immutability, distribution, decentralization and permanence.
Under a generic point of view, a private data (PII) has to be carefully managed with a blockchain technology to be GDPR compliant. Of course, the data type to store strongly depends on the applications. In the energy field, typical examples of PII are identifiers such as the POD or the serial number of the smart meters.
The solutions described in the following Sections 4.3.6 are implemented in the second-layer applications explained in Section 4.2. However, their approaches are still valid also for first-layer applications described in Section 4.1.

4.3.1 Data immutability

Blockchain entries are immutable, i.e. immutable data cannot be altered without creating a new block of data. This is true for the blockchain because each block is intricately linked to the previous one to form a chain. The information in any block is an extension of the previous block and flows into the subsequent one. You would have to alter an entire blockchain channel to manipulate information that has been committed to one block. While blockchain immutability is considered a strong DLT security feature, it creates a challenge for GDPR compliance. The GDPR policy requires citizens to have the authority to edit or update their personal data held on digital platforms. This is necessary in cases when the data submitted is no longer valid or found to be erroneous. Personal data can become invalid if the owner changes details such as names, addresses, marital status, bank accounts and others.

4.3.2 Data distribution

Before data can be added to the blockchain, it is disseminated through a network of validators for approval. After approval, the data is visible to all the participants who have access to the blockchain. This promotes blockchain transparency and authenticity. Also, the validators hold exact duplicates of the ledger. This is why the blockchain is referred to as a distributed ledger. This feature enhances the blockchain security significantly. To successfully alter any entry on a blockchain, the same entry must be altered on the duplicate ledgers simultaneously. GDPR protocols restrict data collectors from sharing their customers' personal data with unauthorized parties. Also, these organizations need the data owner's consent for third parties to access or process data containing their PII. Blockchain networks work with variable numbers and sets of validators at a time. These validators are free to participate in the data verification process at their discretion. As such, requiring consent for each participating validator complicates the entire data verification process.

4.3.3 Data decentralization

Conventional transactions and agreements are facilitated by an authorized intermediary. This could be the regional authority, a lawyer, financial institution, policy regulator or a trusted witness. The intermediary authenticates the transaction by performing due diligence and vetting the participants' information. This can be a time consuming and costly process resulting in delays in transaction approvals. Yet, on the blockchain, transactions are verified by a decentralized network of validators. The blockchain itself acts as the intermediary for the transactions. This is because validators check whether the information currently presented agrees with the data on the previous block. This ensures that there is continuity in data management on the blockchain and reduces the risk of fraud. The blockchain's decentralized structure also means that it is free of jurisdictional limitations. The participants of the blockchain are free to select the governing law or jurisdictional authority that will preside over their transactions or disputes. The agreement is then carried out in compliance with the predetermined governing law. Selecting the governing law to preside over smart contracts administered via blockchain technology requires careful consideration. These arrangements are valid whether or not the participants reside or originate from the selected governing authority. The GDPR is relevant to organizations and sites that operate in the EU region. Organizations that are registered outside the EU are also subject to the GDPR requirements if they attract customers living in the EU. The blockchain's geopolitical autonomy makes it attractive for transactions that are not subject to the regional authority's control. This significantly reduces the cost and bureaucracy involved with sensitive transactions. Yet, this form of decentralization makes it challenging for the GDPR to track and impose penalties for non-compliance.

4.3.4 Data permanence

Permanence means that you cannot erase any data that has been committed to the blockchain. This is due to the interconnectivity of blocks on the chain. Deleting one block would render all the subsequent blocks irrelevant. Also, Blockchain data is duplicated, verified and stored in a distributed network of nodes, or validators. The use of byzantine fault-tolerant (BFT) systems on the blockchain ensure that the failure or compromise of a validation node does not result in the loss of data. The data remains in the system as long as two-thirds of the validator points are fully functional. Blockchain permanence and immutability apply similar limitations to the GDPR protocols. The GDPR requires users' personal data to be deleted after use. This can be after the fulfilment of the intended purpose for submission. The data can also be held until the expiry of a period defined by the transacting parties. Deletion of unused personal data reduces the risk of privacy infringements. The data collector should also be able to delete personal data when requested by the owner of the data.

4.3.5 Viable Solutions for GDPR compliance using blockchain

GDPR compliance on the blockchain needs the alignment of the three core elements. These are the GDPR protocols, the blockchain framework and the personal data in question. The blockchain features identified as limitations to GDPR compliance are deeply set in the DLT's structure. They are fundamental characteristics that cannot be manipulated without compromising the value of the blockchain. In the same breath, the GDPR policy is quite comprehensive with little room for manoeuvrability. It is a detailed policy that consists of 99 articles that elaborate its specific applications. Its protocols are designed for application on a wide variety of platforms, both online and offline. Therefore, the GDPR requirements cannot be altered to suit the blockchain limitations. The only element left to facilitate the integration of GDPR protocols on the blockchain is the personal data submitted for processing. The GDPR policy is only relevant for data that can be used to identify a natural person. As such, to achieve compliance, the data should be fashioned such that it cannot identify the data subject. Data that cannot identify the data subject directly or indirectly doesn't pose any risk of data privacy infractions. In such a state, the data beyond is the scope of the GDPR policy and can exist freely on the blockchain. The basic idea of this approach to data management is to separate the data subject's PII from the submitted data set. Once the PII has been removed, the dataset becomes objective. It cannot be attributed to any person either directly or indirectly. This can be achieved through any one of the four methods discussed below.

1. Personal Data Avoidance
2. Personal Data Encryption
3. Personal Data Anonymization
4. Personal Data Pseudonymization

It is important to note that these approaches do not manipulate or infringe any of the GDPR or blockchain protocols. They are all legitimate data management procedures that can be applied safely to various sets of personal data. Data encryption, anonymization and pseudonymization are widely acceptable data masking procedures that have been given special consideration within the GDPR protocols.

Personal Data Avoidance: The easiest way to achieve GDPR compliance on the blockchain is to avoid processing personal data. This requires the data subjects, and controllers to be clear on what the GDPR considers personal data. Yet, the omission of all personal data from the blockchain will significantly limit the scope of its operations. A blockchain that exclusively deals with non-personal data may not be practical in today's data economy.

Personal Data Encryption: It is possible to encrypt personal data, as shown in Figure 26, before storing it on the blockchain. The access, amendment and deletion of the personal data can be managed through the data subject's unique encryption keys. However, the blockchain limitations of data permanence and immutability still pose a challenge to this approach. This is because the personal data stored on the blockchain, though encrypted, cannot be altered or deleted. If there is a breach that compromises the data subject's encryption keys, the data can be exploited. Also, as technology advances, the most versatile encryption keys today may become obsolete in a few years. This means the encrypted personal data stored on the blockchain is always at risk of exposure to some degree.

Personal Data Anonymization: Data anonymization is a process that encrypts or removes PII from data sets, it is reported in Figure 27, It aims to ensure that the data subjects associated with the data sets remain anonymous. Anonymization is a data masking method that is specifically supported by the GDPR policy. However, GDPR-compliant data anonymization is considerably high. The policy requires that any attempt to anonymize personal data should eliminate the risk of re-identification. This means the protocol should go beyond the anonymization of obvious personal data such as names and addresses. It should also anonymize indirect identifiers such as Internet Protocol (IP) addresses. Yet, eliminating

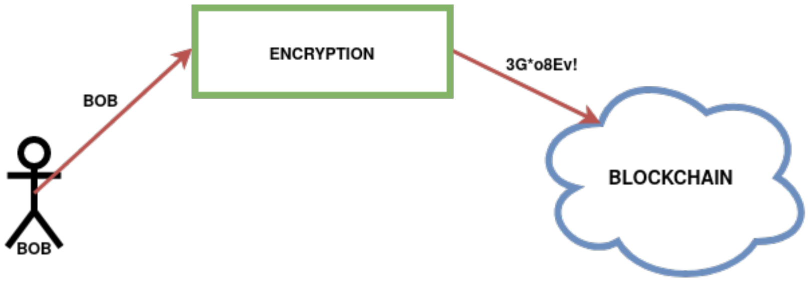


Figure 26: Private data managed via encryption strategy

the risk of data subject re-identification through anonymization can be challenging on the blockchain. To achieve this, the blockchain controller needs to screen all possible data points that can be linked back to an individual. Incomplete anonymization creates still leaves the underlying personal data at risk of exposure. Anonymization on the blockchain is permanent. As such, the data can never be linked back to an individual. While this is a quintessential advantage for GDPR compliance, it creates a challenge for blockchain transparency and validation. The lack of identifiers increases the complexity of data verification and transaction authentication.

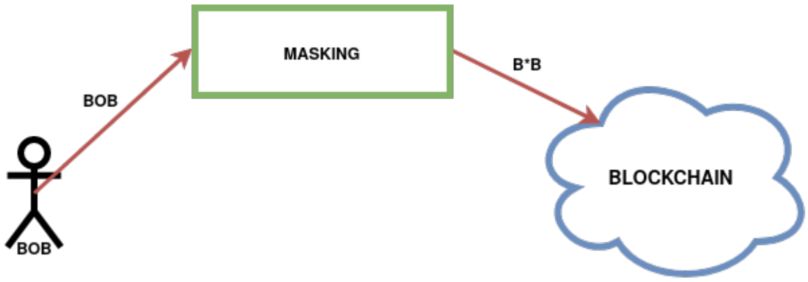


Figure 27: Private data managed via anonymization strategy

Personal Data Pseudonymization: Pseudonymization is another effective method of masking personal data on the blockchain. It involves assigning pseudonyms to personal identifiers. Figure 28 shows a schema depicting how this strategy operates. This prevents the data from being linked to an identifiable person without using additional information. Through pseudonymization, GDPR policy requires any information that can be attributed to an identifiable person to be kept separate from the data sets being processed. It authorizes the controller to apply technical measures within to ensure that the PII cannot be linked to the data subject. The GDPR pseudonymization guidelines apply to all types of organizations that process personal data. Yet, it is uniquely suited for blockchain data masking. This is because the data subject's PII can be maintained off the blockchain while processing personal data. The blockchain controllers can use various mechanisms to create links between the processed data and the personal identifiers where necessary. This approach allows data subjects to access their data on the blockchain through the links provided by the controller without being directly identifiable. GDPR compliance is thus achievable since the personal identifiers and not subject to blockchain limitations. The data subject can modify or erase their personal data without creating new blocks on the chain. Although the personal identifiers are replaced with pseudonyms, the GDPR policy still considers it to be personal data. Yet, this method enables the controller to prevent the disclosure and distribution of personal data to blockchain validators. Severing the links between PII and processed data minimizes the risk of the data being attributed to an identifiable person. Yet, the data pseudonymization is still reversible even when the additional information on personal identifiers is held off the chain. The blockchain controller can delete the personal data identifiers at the data subject's request. Once the identifiers are erased, the GDPR protocol treats it as anonymized data. At this point, the pseudonymization is irreversible and the controller is no longer obligated to comply with GDPR requirements.

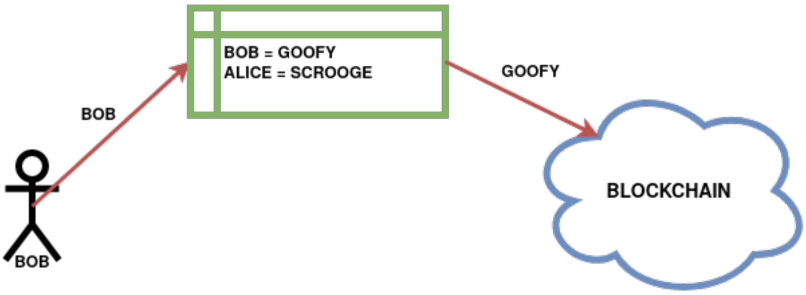


Figure 28: Private data managed via pseudonymization strategy

4.3.6 Strategies implemented in LIC pilot

Section 4.2 describes the 2nd-level blockchain applications that are developed and deployed in the LIC pilot. This section explains how the privacy of the LIC actors has been preserved in the applications. Under a generic point of view, it is important to remark how some strategies described in Section 4.3.5 can be reasonably always used in a blockchain application, a typical example is the pseudonymization. Indeed, the usage of an off-chain trusted storage strongly simplifies the privacy issues related to blockchain. On the other hand, there are others cases that are both fully decentralized and able to preserve the privacy, such as the encryption, whose use depends on the specific application. In the following the usage of both the approaches in the developed applications are described.

Pseudonymization In *Metering* and *PS* applications (see Sections 4.2.6 and 4.2.8 for details), each node stores data about its consumption and production in plain text, i.e. the datasets are not encoded. The privacy preserving strategy chosen for these two applications was the pseudonymization. Basically, the identifier of a meter N (i.e. its serial number) saved on the chain is known only by itself and the *Admin* account, which is supposed to be trusted and owns an off-chain table containing the correspondences between the pseudonyms and the real number identifiers. Once the trusted *Admin* deletes this list, nobody will be able to know the identities of the real nodes on the chain.

Homomorphic encryption Homomorphic encryption is a peculiar type of encryption, which allows an entity to perform calculations on encrypted data but not to decrypt the inputs calculations. Consequently, an entity, e.g. a node in the Lugaggia demonstrator, is allowed to calculate data related to the entire community (i.e. the total consumption, given by the sum of the nodes encrypted data), but it cannot decrypt private data of other nodes. An effective implementation of the homomorphic encryption in LIC has been used in *AT* application, please refer to Section 4.2.7 for details.

5 Simulation environment

In this section, a description of the common simulation environment used throughout the projects is provided. The simulation environment, called complexively OPTISIM, is a package written in the Python language consisting of several components.

5.1 Load flow simulation

The selected simulation engine is OpenDSS, a freely available, industry-grade distribution system simulator by EPRI. It is a fast, widely used and powerful simulator usable through a COM interface. To facilitate the interaction with the Python language, the krangpower package was created, aiming to provide several enhancements over the native OpenDSS interface. Krangpower is built over the OpenDSS-direct.py thin wrapper and offers syntactic sugar, such as operator-based insertion of elements in the circuit and retrieval of objects through simple indexing. Furthermore, krangpower offers the following capabilities:

- Items that OpenDSS returns as simple lists of floats, representing real and imaginary parts of flattened arrays of physical quantities, are returned as a numpy array with the correct shape and format.
- Structured items such as the admittance matrix are returned as Pandas DataFrames for easier manipulation and export.
- Items come, where appropriate, as quantities including information on the measurement unit, enabling easy conversions and secures against miscalculations.
- The OpenDSS text interface is checked for errors, instead of returning them as strings without raising an error in Python.

Another key additional feature is the ability of returning a graph representing the underlying grid. First of all, this enables the possibility of advanced drawing. The graph has the associated electric elements as properties of the nodes (e.g., loads) or the edges (e.g., lines). This allows, leveraging the built-in iteration modes of the networkx graph, to extract quantities of interest, such as total installed power at the nodes, node voltages and so on. Krangpower is also able to read and write specifically designed structured files in the JSON format, completely specifying all the electric elements, the connections and the settings of the circuit. In contrast to the typical solution adopted with vanilla OpenDSS and other packages of saving the circuit-building script, by design in the GUI version and typical with thin wrappers, this allows to read and write the circuit-defining structure with external tools, unlocking further visualization and computer-aided design possibilities.

5.2 Simulation architecture

5.2.1 Interaction with models

Many of the most interesting smart grid studies involve co-simulation with models of batteries, heating, houses and other appliances connected to the grid. We will call them, generically, "agents". A typical example is that of heating systems whose final user request is known (from request profile generation tools or by other means). In order to model the resulting electrical power request presented to the grid by the agent, one must take into account a thermal model of the house (in the most simple case, a RC model [50]) and a model of the heating machinery complete with its control (for example, a non-modular heating pump with a stratified water tank, controlled with an hysteresis cycle for avoiding too frequent starts). OPTISIM offers the possibility to insert software models of these kind of objects directly into the simulation. These models are configured in a dedicated structured json file, whose parameters are fed to the constructor of these objects. During the main simulation cycle, OPTISIM is able to get the computation results from the objects, obtaining active and reactive powers that are then used to configure Krangpower for the following step. Often, these models need time-series data streams as input for computing the final power (such as the aforementioned consumption profiles, irradiance profiles for the models of photovoltaic plants). In vanilla Krangpower, it is possible to use simple delimited files, leveraging the underlying capabilities of OpenDSS. In the OPTISIM framework, a more flexible solution was chosen, involving a full-fledged time-series database, as we will see shortly.

5.2.2 Interaction with algorithms

The central feature of OPTISIM is the ability to run separately algorithms for governing the behavior of the agents. In order to foster clean and modular writing of the algorithms, OPTISIM is able to communicate with the algorithm, running in separate processes, through standardised message passing through a message broker. OPTISIM formulates a request for instructions every n simulation steps, where n is a pre-configured time horizon for which the algorithms make decisions, and expects a JSON-encapsulated response with control instructions. Concretely, the core of these messages is a table whose columns represent the controlled appliances and whose rows represent the upcoming timesteps for which a decision is made available. OPTISIM then passes the appropriate instruction to the agent models and goes on with the next step of the simulation. The message broker, as we will see in section 5.2.5, is used for various communication tasks between the software components, that all run asynchronously and in parallel.

5.2.3 Multiprocessing

Since a simulation in the OPTISIM framework may involve, for each timestep, the computational evaluation of many resource-intensive models, there is an evident incentive in running them in parallel, in order to benefit from the usage of several CPU cores in terms of simulation wall-clock time. The single agents, such as house models, batteries, etc., are aggregated under a single "meter", representing a commercial point of delivery that owns the underlying appliances. From a computational infrastructure point of view, each meter acts as the collector of all the instruction messages for its appliances, that are consumed through a queue; through another queue, it returns its total power consumption as already described in Section Interaction with models. Each meter with its agent models runs asynchronously, in a separate process. Note that this architecture is similar to the pattern of load balancing servers, where queue-interfaced workers execute computationally intensive jobs, albeit in this case the workers are not interchangeable and the balancing works since the computational load is, in each case, similar.

5.2.4 Time series database

Complete and full-featured simulations of networks with a consistent number of agents typically involve several forms of time-series information. A first example regards recorded smart meter data; when smart metering is available for research, for example in pilot projects such as LIC 1.4, it is often interesting to replicate measured data in order to validate grid models or simulate the effects of operating controlled devices in that same grid. Other typical cases are irradiance profiles, needed for simulating photovoltaic installations and their effects on the grid, or heating/consumption user profiles. The volume of data stored can be significant, but the common trait is the time-series nature of the streams. A simulation needs to perform time alignment of the different data flows utilized and needs a fast, time-oriented database. To this end, the InfluxDB database was selected as the best option. As well as being fast and intrinsically capable of managing measurement units and timezone-aware timestamps, the InfluxDB ecosystem includes powerful visualization tools, such as the Grafana web application, that directly interact with the database. This possibility is very useful when dealing with several data streams, different in nature, and allows for immediate, industry-grade monitoring panel construction. Many outputs of the simulations are time-series too, and it is naturally possible to upload them on the database. Interesting results that typically need to be analyzed are powers, node voltages and currents, internal state of models, etc., that are stored as streams of datapoints in time tagged by node, line, point of delivery, etc. according to the particular characterization of the quantity. The flexible visualization of this data under different aggregations and in various types of graph, beyond the self-evident advantage in the post-simulation analysis stage, is also precious for detecting potential simulation or input errors.

5.2.5 Overview and message broker

As we have seen, the OPTISIM framework involves several processes running in parallel (the main script with the power flow simulation, the database, the externally interfaced algorithms, the agent models). In figure 29, an overview of the whole modular architecture is depicted, together with the data flows between the parts.

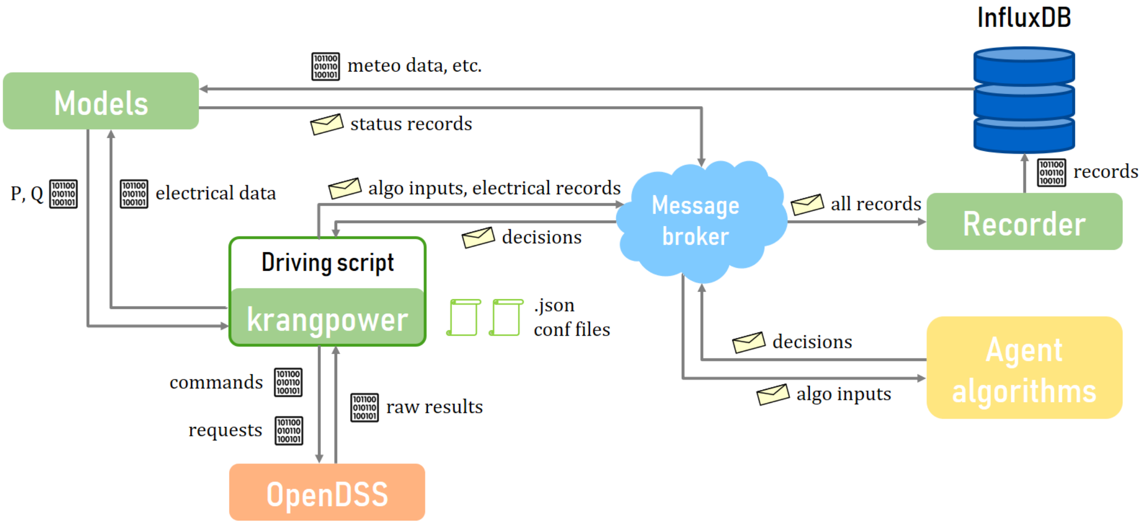


Figure 29: Scheme of the simulator

The main script governs directly krangpower, by calling functions for updating the circuit elements according to the actions taken, advancing the simulation and evaluating the electrical results of interest, both needed by the algorithms and the models (for example, a voltage-reacting battery needs information about the voltage). Krangpower, in turn, interacts directly with the OpenDSS engine as seen in the dedicated section. It is the main script that creates the processes containing the meters, and doing so it configures input and output queues for exchanging information with them. Information exchange with the algorithms and to the database happens through a message broker. The message broker chosen for OPTISIM is RabbitMQ. Both the models and the power flow simulator send all the simulation results as messages to a common recorder, that subscribes to these flows and uploads them on InfluxDB in batches. Some of the results are also sent as messages as needed by the algorithm. As explained previously, the algorithms must implement an asynchronous interface reading these messages, that also have the function of triggering the next simulation steps, and respond similarly by sending messages to the main script containing the decisions. If needed, the models can query the database directly (through a buffer, not depicted, for reducing the number of queries and increase performance).

5.3 Energy profiles generation

The simulations rely on a synthetic dataset of residential load profiles, generated using the Load Profile Generator [51]. The Load Profile Generator (LPG) uses a behavioural conditional to specific meteorological conditions to simulate a plausible consumption. Consumption profiles due to heating and cooling are modeled separately, using the models specified in [52] and only the water consumption profiles from the LPG are fed to the models that are generating them. A detailed explanation of the thermal simulation is available in the Annex.

5.4 Components modeling

5.4.1 Thermal loads

In order to obtain a representative dataset for Switzerland, we used the STASCH standard [53] and its variants as a reference for the heating system and the control logic. In particular we implemented STASCH6, which includes both space heating and domestic hot water production. The STASCH6 standard comprehends 3 main components: a heatpump (HP), a water tank used as an energy buffer, and a heating element delivering heat to the building. The control logic is explained in detail in annex A.1.

Building model We modeled the building thermal dynamics with a simple one state RC equivalent model, as done in [54]. The main reason for this choice is that it is hard to generalize RC models with higher number of states, since no values can be found in the literature for the needed parameters. Estimating an RC model from data requires different measurements of temperatures, internal and solar gains, at a resolution of at least 10 minutes. This kind of datasets are extremely hard to find, and limited to only a few, often undwelled, cases. These equivalent RC circuit parameters could, in theory, be estimated starting from first principles, however recently proposed studies show that this can give worse results then estimating a model from data [55]. The second reason is that, while a higher order model leads in general to smaller one step ahead residuals compared to a lower order model, the loss of accuracy passing from a one state model to an higher order one when considering a longer period of simulation is much lower [56]. Last, when considering RC models for buildings with a number of states higher than 3, the chances of overfitting are high, and additional measurements such as the heat fluxes between thermal zones are required to guarantee observability. Alternatively, pseudo-random binary sequences can be applied to the heating systems in order to excite the system in a wide range of frequencies [57], while being uncorrelated with other exogenous inputs. This technique induces high changes in internal temperature of the building and cannot clearly be applied to occupied buildings.
The final model is the following;

$$C\frac{\partial T_z}{\partial t} = \frac{T_{ext} - T_z}{R} + kQ_h + A_{eq}I_s \tag{50}$$

where T_{ext} is the the external temperature, R is the equivalent thermal resistance for the building, C is the thermal capacitance, k is a parameter weighting the estimated power coming from the heating system Q_h, I_s is the incoming solar radiation and A_{eq} is the estimated equivalent window area. In order to obtain representative simulations, R, C, k and A_{eq} were estimated from statistical data for Swiss households. For the simulation regarding the LIC pilot, R was directly estimated from data and C and A_{eq} where estimated from the buildings' equivalent area.
At Swiss level, in a future version the thermal characteristics of the building will be linked with the map of the demand from residential and commercial buildings provided by the Swiss Federal Office of Energy in the GEO Admin API[15], which are based on the Federal Statistical Office's 2014 Buildings and Dwellings Statistics and 2013 Statistics on Company Structures, and the Federal Office of Energy's figures in the annual report 'Energy consumption in the industry and services sector'.

Floor heating The heat distribution system that allows to transfer the required heat from the storage tanks to the building was designed from first principles. A heating system based on serpentines was modeled, the details are presented in Annex A.2.

Water tanks The water tanks connected with the floor heating, which are used as a buffer by the heat pump, and those used for domestic hot water (DHW) are modeled as N-states fully-mixed stratified tanks. Despite not being able to model buoyancy driven effects such as heat plumes and transient de-stratification, this kind of models are suitable for 1D simulations and control [58]. A detailed description of the model can be found in Annex A.3.

Heat pump model The heat pump is modeled by means of interpolated tables, in which heating and electrical power are available as a function of the evaporator and the condenser temperatures. The tables were taken from the energy simulation software Polysun[16]. When the heat pump produces heat for both the heating system and the domestic hot water, its control logic prioritizes the latter, meaning that the buffer is heated as long as the DHW tank temperature sensor reaches the upper bound of its hysteresis control.

[15] https://map.geo.admin.ch/?layers=ch.bfe.fernwaerme-nachfrage_wohn_dienstleistungsgebaeude&lang=en

[16] www.velasolaris.com

5.4.2 PV model

Residential PV power plants were modeled using the Sandia National Laboratories PV Collaborative Toolbox [59], which is based on the 1985 Grover Hughes' Engineering Astronomy course at Sandia National Laboratories. The global horizontal irradiance from the weather model is projected into the plane of the PV array and the AC power produced by it is calculated assuming standard poly-crystalline modules are used.

5.4.3 Stationary battery models

Stationary batteries have been modelled as a one state dynamic system with asymmetric charging and discharging efficiencies, self-discharge, and a minimum charging and discharging power.

6 Lab environment

Prior to field installation of the hardware needed for decentralized load control and also to begin to familiarize with the meters installed by AEM, a laboratory test environment was set up. The hardware and software of the test environment are described below. The same combination of hardware and software was then used in the pilot project.

6.1 Hardware setup

6.1.1 Landis+Gyr E450

E450, produced by Landis+Gyr[17], is an industrial and widely used device in the field of residential smart metering. It has been exhaustively tested in the laboratory due to its installation in the prosumer constituting the LIC demonstrator (see 8.6.2 for more details). E450 provides a notable collection of signals that can be accessed for monitoring. Besides, two relays of the meter can be remotely actuated. It is important to remark that E450 provides an optical serial interface that can be used to continuously read data and send command using the open DLMS protocol[18].

6.1.2 Strato

The Strato Pi CM board [19], shown in Figure 31, is based on a Raspberry CM platform[20]. The Raspberry CM combines the computational power and easiness to use of the Raspberry Pi, i.e. a complete Linux operating system based on an ARM v8 platform, with the high reliability and service continuity of an industrial PC. This is achieved, in particular, thanks to the absence of an SD card, which is substituted with a much more robust internal eMMC Flash, and thanks to the presence of a hardware watchdog. Lab tests were conducted to verify the stability of the system against sudden power outages. All the devices always restarted the operating system without problems. Table 5 summarizes the main features of the Strato devices.

[17] https://www.landisgyr.it/product/landisgyr-e450/

[18] https://www.dlms.com/

[19] https://www.sferalabs.cc/strato-pi/

[20] https://www.raspberrypi.org/products/compute-module-3

Figure 30: E450 smart meter

CPU	4 ARMv8 64-bit 1.2GHz
RAM	1 GB
DISK	32 GB
USB ports	2
Connectivity	1 Ethernet port
OS	Raspbian GNU/Linux 9 (Stretch)

Table 5: Strato main features

6.1.3 NUC

Strato are embedded devices that can be used in a wide variety of applications. However, they are based on ARM architecture [60] and, consequently, they can not manage specific applications that require too significant hardware resources. In order to manage these peculiar cases, the NUC board [21] was taken into account. It is an embedded solution provided by Intel®, more powerful and robust than the Strato. For example, in the LIC pilot (see Section 8.6.2 for details) the NUC acts like an aggregator in the community running specific control applications that cannot be managed by the Stato devices. Table 5 summarizes its main features.

CPU	8 Intel(R) Core(TM) i7-8559U CPU @ 2.70GHz
RAM	16 GB
DISK	500 GB
OS	Ubuntu 18.04.4 LTS

Table 6: NUC main features

6.2 Software setup

The present section describes the main software installed on the Strato and NUC devices. It is important to remark how these applications are not the only ones deployed on the boards. Fundamentally, they are needed by the applications described in Chapters 3 and 4 to operate correctly. Thus, they can be

[21] https://www.intel.com/content/www/us/en/products/details/nuc.html

Figure 31: Strato device

considered as a software layer between the smart meters and the developed applications, which can be installed locally on the Strato or remotely on other machines. Figure 32 reports the interactions between the main software running on a Strato, which are described in the following of this section, and a custom application, like the ones described in Chapters 3 and 4.

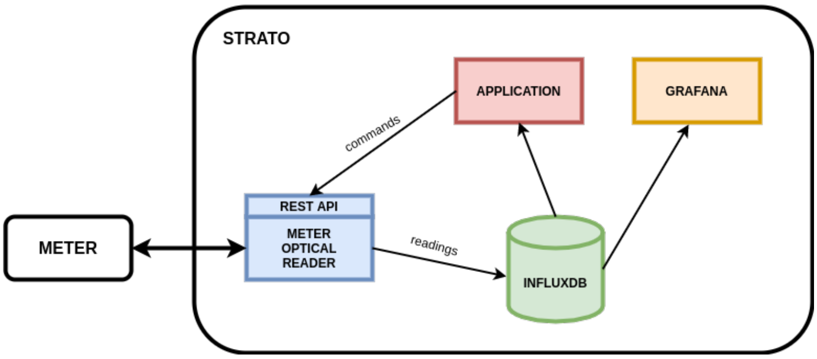


Figure 32: Main applications running on Strato

6.2.1 Meter optical reader

Meter Optical Reader is a custom Python application aiming to continuously provide a bidirectional interface with the smart meter. The hardware interface between the Strato and the meter is provided by a USB optical reader. Thank to it, the software can gather electrical signals (e.g. active and reactive power, current, voltage, etc.) from the smart meter and send command to its relays. The data acquired by *Meter Optical Reader* are saved locally in an InfluxDB database [22], which provides a local data source that can be used by the applications described in Chapters 3 and 4. In addition, *Meter Optical Reader* is configured to save the datasets also on a remote InfluxDB database server. This dual data saving is due to be able to manage both a centralized control algorithm based on a central database and a decentralized approach, please refer to Chapter 3 for details.
The software manages to collect an entire dataset, comprehensive of 27 signals, with a remarkable time resolution of approximately 5 seconds. Regarding the commands, *Meter Optical Reader* provides a REST API that can be used by authorized applications to actuate the meter relays. The data exchanging

[22] https://www.influxdata.com/

is based on the DLMS protocol [61] and exploits the open-source Gurux library [62]. Table 7 reports the signals currently collected and the commands with the related DLMS codes.

V1	’1.0.32.7.0.255’
V2	’1.0.52.7.0.255’
V3	’1.0.72.7.0.255’
I1	’1.0.31.7.0.255’
I2	’1.0.51.7.0.255’
I3	’1.0.71.7.0.255’
IN	’1.0.91.7.0.255’
ITot	’1.0.90.7.0.255’
freq	’1.0.14.7.0.255’
PImp	’1.0.1.7.0.255’
PExp	’1.0.2.7.0.255’
QImp	’1.0.3.7.0.255’
QExp	’1.0.4.7.0.255’
S	’1.0.9.7.0.255’
PF	’1.0.13.7.0.255’
AUL1UL2	’1.0.81.7.10.255’
AUL1UL3	’1.0.81.7.20.255’
AUL2UL3	’1.0.81.7.21.255’
AUL1IL1	’1.0.81.7.40.255’
AUL2IL2	’1.0.81.7.51.255’
AUL3IL3	’1.0.81.7.62.255’
EPImp	’1.1.1.8.0.255’
EPExp	’1.1.2.8.0.255’
EQImp	’1.1.3.8.0.255’
EQExp	’1.1.4.8.0.255’
Relay1	’0.1.96.3.10.255’
Relay2	’0.2.96.3.10.255’

Table 7: Signals acquired by *Meter Optical Reader*

6.2.2 InfluxDB

As explained in Section 6.2.1, *Meter Optical Reader* needs a local database server in order to locally save the collected measurements. This data management is provided by the installation of an InfluxDB server on each Strato. It is a time-series database optimized for the management of time-series datasets. Besides, InfluxDB can easily run on boards based on ARM architecture like Raspberry and, consequently, also Strato. Thus, it is used by *Meter Optical Reader* to store the collected data. However, the resource requirements of InfluxDB on Strato was exhaustively. No problem was encountered in terms of CPU, RAM and disk usage.

6.2.3 Grafana

In order to monitor the data stored in InfluDB by *Meter Optical Reader*, an installation of Grafana [23] has been installed on each Strato. Similarly to InfluxDB, its behaviour was tested to control if its usage was heavy for the operating system. As in the case of InfluxDB, no problems were encountered.

[23] https://grafana.com

7 Pilot environment

7.1 Stakeholders onboarding

We deployed a 3 steps procedure for setting up the LIC Community:

- Legal structure of the community
- Stakeholders' information
- Contracts submission to the stakeholders

7.1.1 Legal structure of the community

The legal structure has been setup as a simple partnership managed by AEM, the local DSO, which is still owning the low voltage cable supplying the district, although it has been put out of the grid solidarity. The LIC simple partnership signed a supply contract and a grid contract with AEM on one hand, and with all the end users/producers inside the district on the other one. At the end of the pilot it is under consideration to create a benevolent association ("Eingetragener Verein") among the buildings' owners and AEM.

7.1.2 Stakeholders' acknowledgment

The stakeholders' acknowledgment happened in two intermediary steps. First of all, we did send to all the LIC's end users a leaflet explaining LIC's goals, it's way of working and highlighting end users' rights and constraints. Secondly, we did organise a general meeting gathering all the stakeholders in order to have the time to present and discuss LIC activities and end users' implication. This meeting was well attended and enjoyed the support of the Municipality (which is also involved in LIC through the Kindergarten), which played an important role for boosting trust among the participants.

7.1.3 Contracts submission to the stakeholders

Finally, we did meet each candidate in a 1:1 individual meeting, which led to the contract signing. It is important to emphasize that all the LIC end users decided to opt in and sign the contracts. In the future, for setting up new "self-consumption districts" inside AEM supply territory, based on LIC experience and results dissemination among AEM's end users, we will need to simplify this procedure which has been quite time consuming (an investment which is hard to be paid back through the Community income).

Nevertheless, we have been positively surprised by the cooperative attitude of the end users (as previously said all of them signed the contracts and are participating to the pilot project) and their attitude to start thinking as a community and less as single users. For instance, a building owner decided to put its PV program on hold because he would like to be sure not to impact negatively on the self-consumption balance inside the community. Similarly, other end users offered us their rooftop, if the construction of additional PV capacity would help in increasing resilience and autonomy of the LIC.

7.2 Field configuration

The LIC pilot[24] is located in Lugaggia, a small village near Lugano. In a part of the municipality, an energy community has been created with the collaboration of AEM[25], the local DSO. The energy community consists of 18 residential houses and a kindergarten. Figure 33 shows the pilot map.

[24] https://lic.energy/

[25] http://aemsa.ch

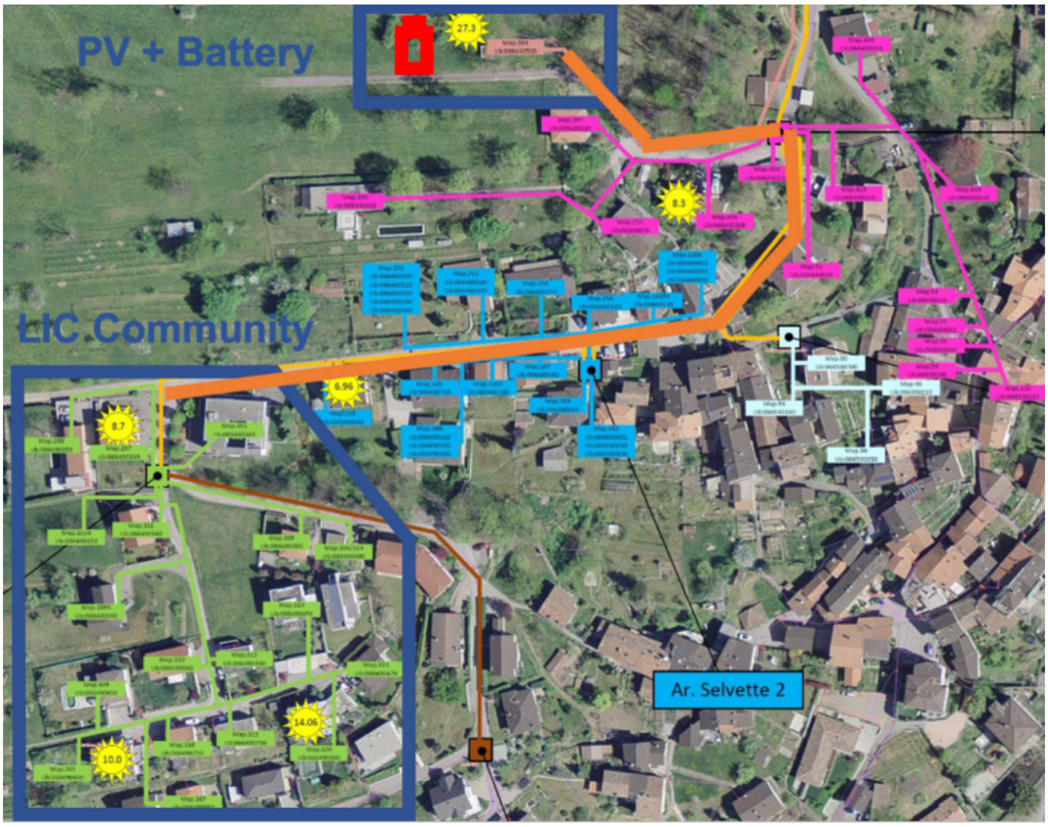


Figure 33: Lugaggia pilot

LIC community is composed by the two blue polygons reported in Figure 33, representing the kindergarten at the center top and the LIC houses at the bottom left. Three houses, represented by a sun icon in Figure 33, are equipped with photovoltaic systems installed on the roof, for a total nominal power of 33 kWp. Besides, a 27 kWp photovoltaic plant and a battery with a capacity of 60 kWh are installed in the kindergarten.

7.3 Centralized data management framework

The Optiflex project, described in Section 1.3, aims to develop a centralized control solution fully based on the existing smart metering infrastructure, without the need for additional hardware. Consequently, a solution allowing to retrieve the data from the smart meters and send commands to them was developed. Figure 34 depicts the data flow of the centralized system, which is based on two main components: the smart metering head-end system (HES) and the big data platform Kibid.
The HES provides the low-level interface with the Landis+Gyr smart meters. Normally, the smart meters are read using PLC via a data concentrator, which is represented by the light blue rectangle in Figure 34. The meters that required to be read at a higher frequency (e.g. the coupling point of the community) are directly connected to the HES using fiber.
The data that are continuously sent to the HES by the meters are stored in a time-series database within Kibid, a big data software solution by Kisters AG [26]. It is important to remark that this centralized solution was not developed only for the LIC pilot but for the entire area managed by AEM. Currently, data concentrators collect data of about 9’000 smart meters. Both HES and Kibid are installed on-premise on a set of machines to assure a high level of robustness and redundancy.
The data exchange between HES and Kibid shown in Figure 34 is provided using the IEC61968-9 pro-

[26] https://energie.kisters.de/loesungen-produkte/kibid-big-data-data-analytics/

tocol [63]. Third-party applications can access the datasets stored in Kibid using libraries developed by Kisters and based on REST APIs. Each access requires a successful login of the third-party application.

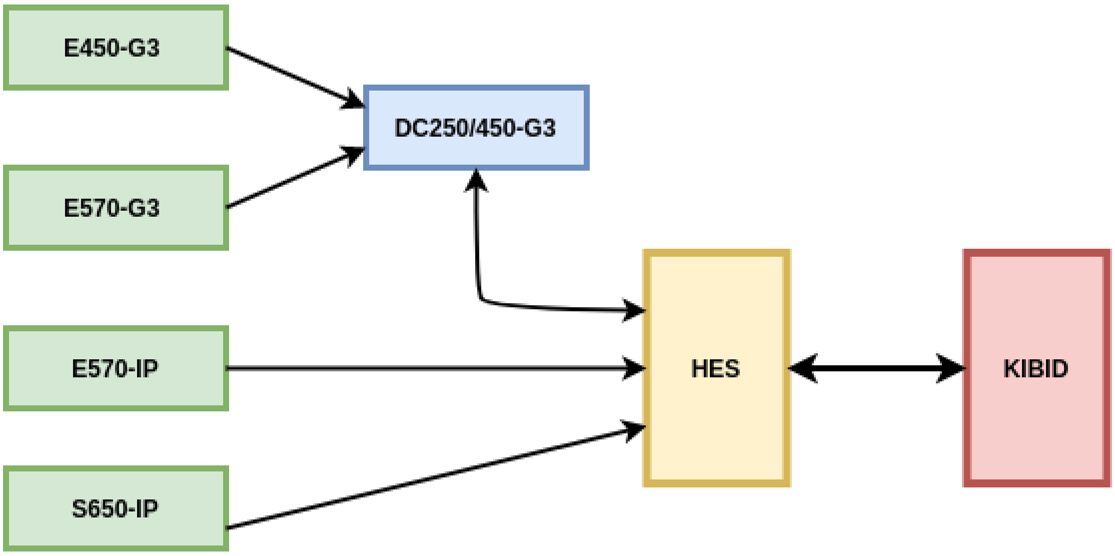


Figure 34: Data flow between HES, Kibid and the smart meters

Thank to the Kibid interface, various measurements can be downloaded for each meter. Table 8 reports the one-minute resolution signals currently downloaded using the Kibid interface and the related DLMS codes.

P_import	"0.2.3.4.1.1.12.0.0.0.0.0.0.0.0.38.0"
P_export	"0.2.3.4.19.1.12.0.0.0.0.0.0.0.0.38.0"
I_L1	"0.2.3.0.0.1.4.0.0.0.0.0.0.0.128.0.5.0"
I_L2	"0.2.3.0.0.1.4.0.0.0.0.0.0.0.64.0.5.0"
I_L3	"0.2.3.0.0.1.4.0.0.0.0.0.0.0.32.0.5.0"
PF_L1	"0.2.3.12.0.1.38.0.0.0.0.0.0.0.128.0.0.0"
PF_L2	"0.2.3.12.0.1.38.0.0.0.0.0.0.0.64.0.0.0"
PF_L3	"0.2.3.12.0.1.38.0.0.0.0.0.0.0.32.0.0.0"
Q_export	"0.2.3.4.19.1.12.0.0.0.0.0.0.0.0.0.63.0"
Q_import	"0.2.3.4.1.1.12.0.0.0.0.0.0.0.0.0.63.0"
V_L1	"0.2.3.0.0.1.54.0.0.0.0.0.0.0.128.0.29.0"
V_L2	"0.2.3.0.0.1.54.0.0.0.0.0.0.0.64.0.29.0"
V_L3	"0.2.3.0.0.1.54.0.0.0.0.0.0.0.32.0.29.0"

Table 8: Signals acquired through Kibid interface (1 minute data resolution)

7.4 Decentralized data management framework

The solution described in Section 7.3 is suitable for a centralized approach as the one proposed in Optiflex project (see Section 1.3 for details).

On the other hand, the applications described in Sections 3 and 4 require a specific decentralized setup in terms both of deployed hardware and of data collection software. An example of the needing of this peculiar setup is the fact that the framework described in Section 7.3 is not able to guarantee measurements with a time resolution smaller than a minute. Besides, the data collection is not performed in real-time and some delays can occur in the data collection. This problem can affect also the actuation of the smart meter relays, which cannot be performed directly but has to be addressed by the Kibid and

HES systems.
For the aforementioned reasons, an additional decentralized framework was implemented, comprehensive of both hardware and software components that are described in the following sections.

7.4.1 Hardware setup

In order to have a direct interface with the smart meters, the devices tested in laboratory (see Section 7.4.1 and 6.2 for details) were deployed in the cabinets of the LIC houses.
Basically, each LIC end-user corresponds to a node, which is associated to a single point of delivery and equipped with a smart meter. Thus, a Strato device, described in Section 6.1.2, is connected to each of the smart meters via an optical USB port. Besides, in the basement of the kindergarten, close to the central battery, a NUC machine (please refer to Section 6.1.3 for details) has been installed. This machine acts as the aggregator of the community and as an interface with the battery, as described in Section 6.1.3.
Figure 35 shows a Strato installation in a LIC cabinet. The Strato is in the red rectangle, and it is connected to the smart meter (green rectangle) via an optical USB reader (violet circle). The Internet connectivity is guaranteed by a USB dongle (blue rectangle), that provides a 4G data mobile connection.

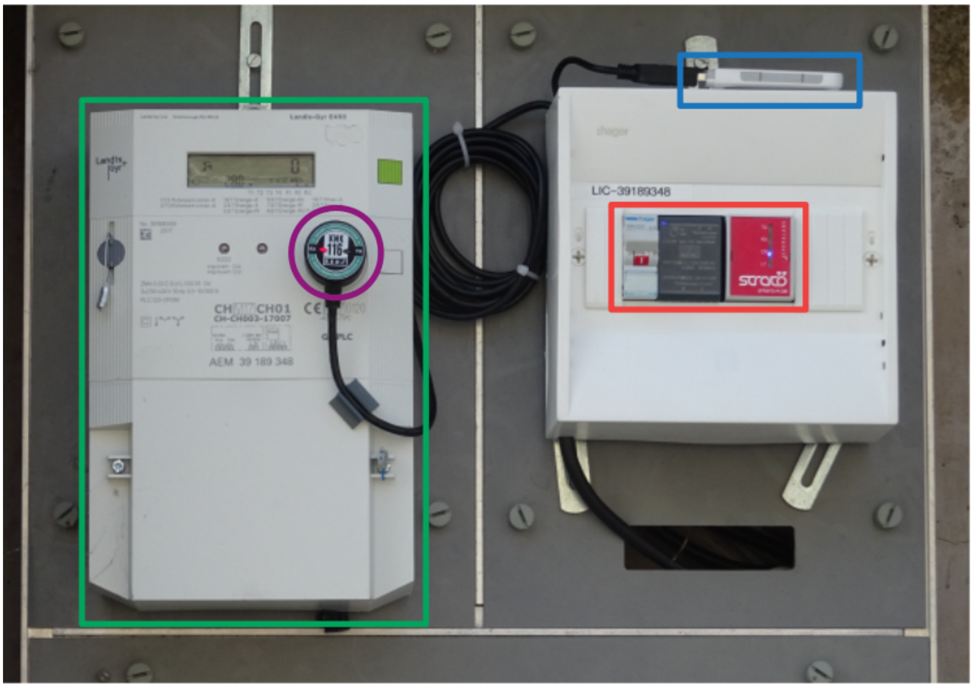


Figure 35: Strato setup in a cabinet

7.4.2 Software setup

Section 7.4.1 describes the nodes installed in the LIC pilot where the applications explained in Sections 3, 4.2 and 6.2 run. In addition to them, to have secure and easy connections among the nodes, a VPN has been deployed in the LIC pilot. Each Strato and NUC has its own VPN certificate providing a static IP address. The VPN is ruled by an OpenVPN[27] instance installed in a SUPSI server.
It is important to remark how this tool has extreme importance in the maintenance of the nodes installed in the pilot, such as the Strato and the NUC devices. Indeed, with OpenVPN each node can be easily

[27] https://https://openvpn.net

and safely accessed to perform various operations, including the upgrading of the running applications, the checking of the operating system status and the interactions between the sidechain nodes (please refer to 4.2.5 for more details).
In addition, an application acting as an interface with the community battery is continuously running on the NUC device. The interaction with the battery is guaranteed by a fast queue protocol defined by Kisters.

8 Results

8.1 KPIs

The impact of the different business models and of the consequent control actions associated with them has been evaluated both from a technical point of view (i.e. impact on the grid) and from a financial point of view. The impact on the grid was evaluated by quantifying:

- The effect of the control actions on the aggregated power of the community. In particular the reduction of peaks and valleys.
- The reduction of ohmic losses (dissipated by Joule effect) on the lines of the internal network, resulting from a better synchronization between production and local consumption and the reduction of peaks.
- The reduction of voltage excursions in the internal distribution network.

Next, we evaluated the economic impact of the different control algorithm pricing schemes, both in terms of the impact on end users and on a hypothetical community administrator.

8.2 Single level EC simulation

We will focus our analysis on the simulated month of July for the LIC simulation model. Simulations step was set to 5 minutes. The computational time required by a simulation step varies considerably depending on the coordination type, the forecasting method used and the control model. It can take from about 1.5 seconds in case of implicit coordination of batteries to about 30 seconds in the case of explicit coordination of boilers. Explicit coordination requires a much higher computational time since it is solved using an iterative approach. In this series of detailed simulations, the iterative distributed problem is solved using the pFB method, described in section 3.1.3. Parameter tuning for the pFB algorithm has proved to be essential and delicate. There is, in fact, a tradeoff between optimal conversion speed to the solution and stability. After parameter tuning, we limited the maximum number of iterations to 50, as we evaluated that, on average, the coordination converged sufficiently well after this amount of iterations.
The optimization problems are always solved using a model predictive control MPC approach over a horizon of 24 hours. In order to make the simulations as realistic as possible, we have assumed that the controller does not have perfect forecasts of any future variable. On the contrary, we have trained different types of forecasting algorithms and evaluated their performance in open and closed loop. We selected a boosted tree based model as the best performing one; see section 9.6 for more details on this evaluation.
The local DSO doesn't allow the injection of energy from batteries into the grid, as any injection into the grid is remunerated as being produced by green energy sources. In the context of an energy community, there is no law regulating the injection of energy into the community's internal grid. Nevertheless, in the case of distributed batteries, there is a possibility that the energy injected into the internal grid will flow from the PCC into the main grid, especially if the batteries do not coordinate. To tackle this problem, we have tested two variants of control algorithms: a variant in which the batteries cannot inject energy into the internal grid, except in case they have to contribute to respect of grid constraints (control model

explained in detail in section 3.1.3), and a more permissive variant, in which batteries are allowed to inject energy into the internal grid, with the risk of leaking into the main grid from the PCC. The following scenarios were tested:

1. Baseline (no batteries, water heaters always enabled)
2. Batteries:
   (a) Batteries not coordinating: In this case the batteries are operated in local (at house level) self-consumption optimization mode. They have no incentives in coordinating, nor in shaving peaks. A delayed charging strategy is in place, to preserve battery life.
   (b) Batteries in implicit coordination mode without the possibility to inject energy into the grid: In this case the batteries will try to optimize the household energy costs, using the AMM pricing scheme presented in section 2.2.2.
   (c) Batteries in implicit coordination mode, with the possibility to inject energy into the grid
   (d) Batteries in explicit coordination mode without grid constraints enforcement, without the possibility to inject energy into the grid. In this case the batteries coordinate with each other, and the surplus money derived from the fact of being members of in an energy community, with the method described in 3.1.3.
   (e) Batteries in explicit coordination mode with grid constraints enforcement, without the possibility to inject energy into the grid.
   (f) Batteries in explicit coordination mode with grid constraints enforcement, with the possibility to inject energy into the grid.
3. Water heaters:
   (a) Water heaters in implicit coordination mode.
   (b) Water heaters in explicit coordination mode with grid constraints enforcement.

8.2.1 Grid analysis

Batteries The impact of the control actions on the grid has been evaluated by quantifying the effect of the control actions on the power at the point of common coupling (PCC) of the community, the total losses in its internal grid and the voltage at its nodes. The total losses considered are the ohmic losses (dissipated for Joule's effect) over the internal grid lines, that in this case are equal to the difference between the power measured at the PCC and the sum of the powers of the meters in the grid. The time-series of the power at the PCC for the first three days of July is shown in figure 36. Figure 37 shows the probability density function (PDF) of the power at the PCC, and figure 38 allows to further analyse the effects of batteries on the active power at PCC, by displaying statistics of the total controlled power as a function of the power at the PCC.

It is clear that the mere presence of batteries in the grid helps to minimize power and voltage fluctuations. It is equally clear, however, that without proper coordination of their actions, the benefit is minimal. In general, allowing batteries to inject energy into the internal grid even when grid constraints are not violated improves performance across all KPIs. However, one can notice in figure 36 and figure 38 that implicit coordination through price signals can cause significant power fluctuations, which with excessive penetration of storage systems could lead to instability. At certain times of the day, the energy injection of batteries into the internal grid is excessive and results in an energy leak into the main grid.
On can also notice that the synchronism between battery actions is greater when the power at the PCC is negative (i.e. when the community is injecting into the main grid). While loads of individual homes are only minimally correlated with each other, production from photovoltaic systems is almost perfectly correlated. This, combined with the fact that all agents use the same methods to forecast PV production, generates a higher synchronization of the batteries when they are charging.
Explicitly enforcing the grid constraints, as expected, helps to reduce the positive and negative peaks

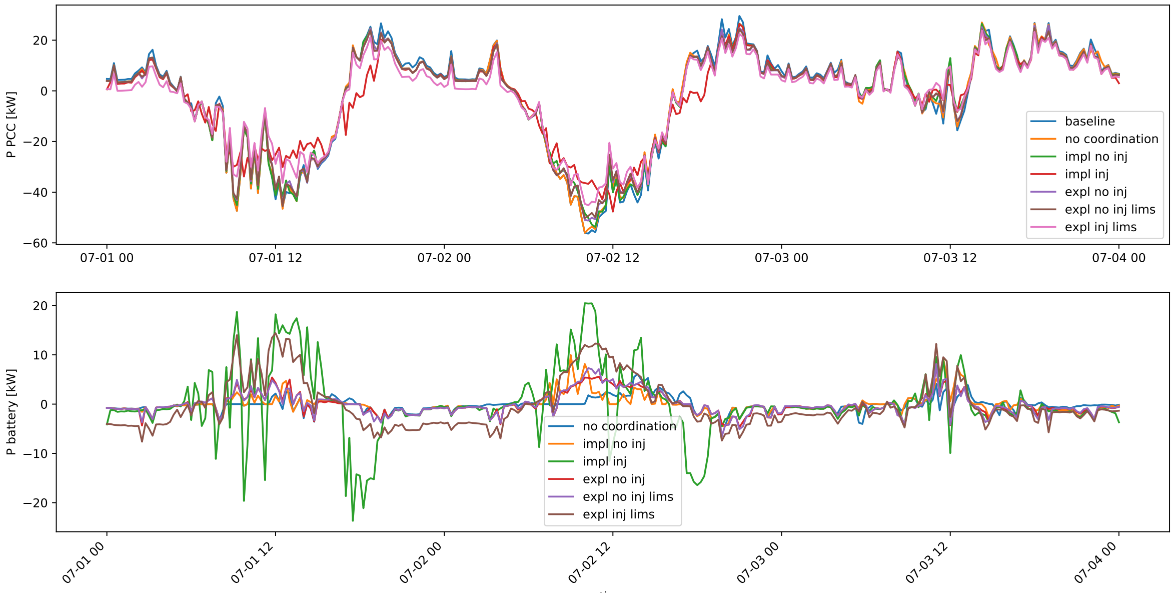


Figure 36: Time-series of the active power at the point of common coupling of the community and the total power of the distributed batteries for the first 3 days of July. Negative power corresponds to injection into the grid, positive power corresponds to consumption. Top: Active power at PCC. Bottom: active power of batteries. *baseline*: baseline simulation with no batteries, *no coordination*: batteries optimizing local self-consumption only, *impl no inj*: implicit coordination, no injection into the grid allowed, *impl inj*: implicit coordination, injection allowed, *expl no inj*: explicit coordination, no injection allowed, *expl no inj lims*: explicit coordination, no injection allowed, power limits at PCC enforced, *expl inj lims*: explicit coordination, injection allowed, power limits at PCC enforced.

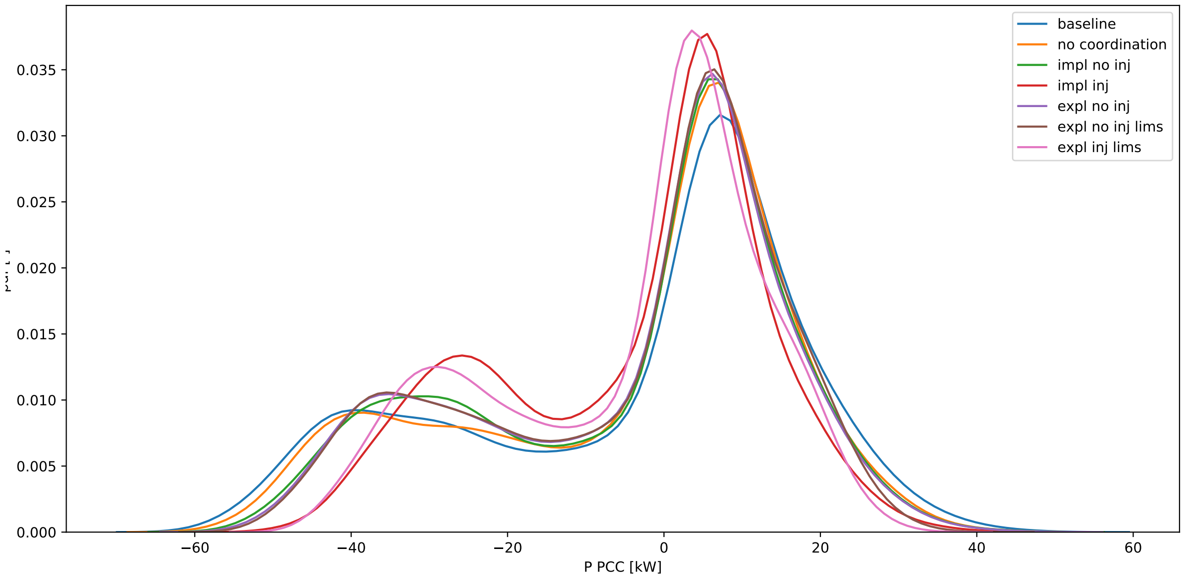


Figure 37: Probability density function (PDF) of the power at the PCC of the community.

at the PCC. Nevertheless, it can be noted that, due to the fact that the control is based on imperfect forecasts, these limits are still violated. Future work will concentrate on the development of robust and stochastic distributed control strategies that will take into account the uncertainty in the forecasts. Allowing the batteries to discharge into the grid at any time, and not only when constraints violations are expected, leads to better performance. This is because the actions of the batteries would have been beneficial even without enforcing grid constraints.

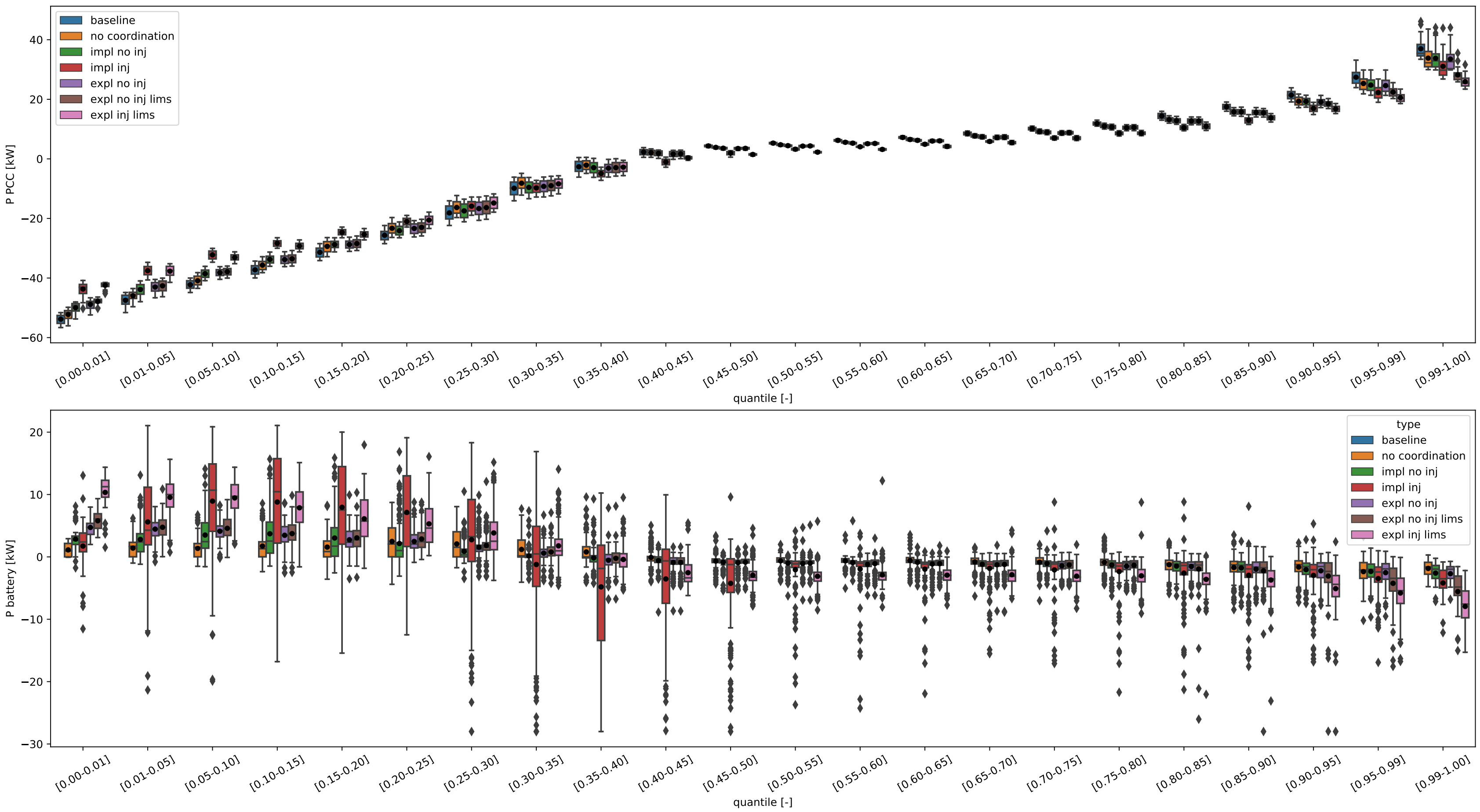


Figure 38: Effect of batteries control on the power at the point of common coupling (PCC) of the community. Top: boxplots of the power at the PCC per inter-quantile range. Bottom: boxplot of the total power of the batteries per inter-quantile range of the power at the PCC.

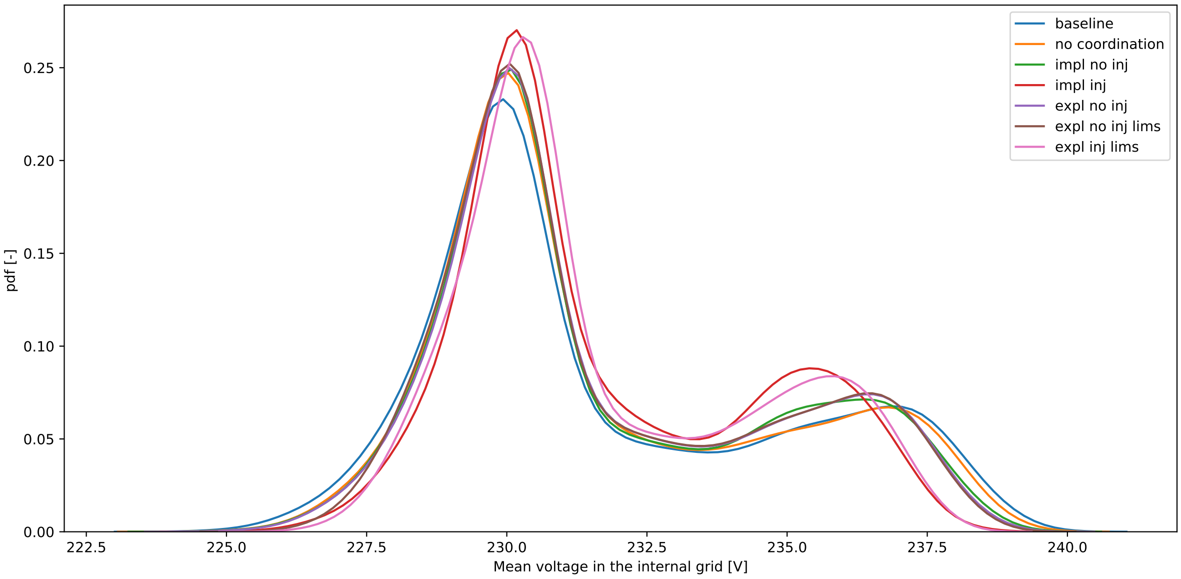


Figure 39: Probability density function (PDF) of the mean L-N voltage inside the community.

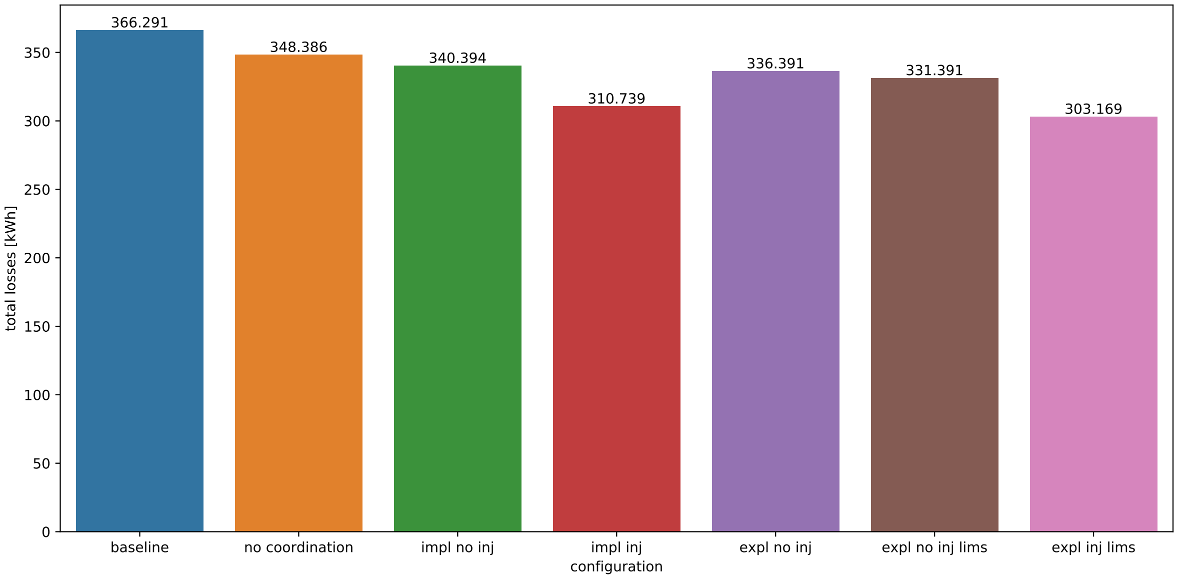


Figure 40: Total ohmic losses inside the community for the month of July.

In the case of explicit coordination, the strategy adopted has the advantage of converging towards a single Nash equilibrium, which guarantees a fair solution for all participants in the cooperative game. However, one might further increase the community surplus if one did not take into account that this might disadvantage some individual participants in the game. To assess the difference between the maximum attainable surplus, and the attainable surplus in the condition where a Nash equilibrium is reached, a simulation was run in which batteries were forced to ignore their internal costs and maximise the community surplus, only. This is achieved by setting $\alpha = 1$ in equation 5. The effect of this strategy on the grid can be seen in figures 41, 42, 43 and 44. One can notice the effect of maximizing the surplus of the community on the grid parameters is not significantly different than when single users' costs are taken into account.

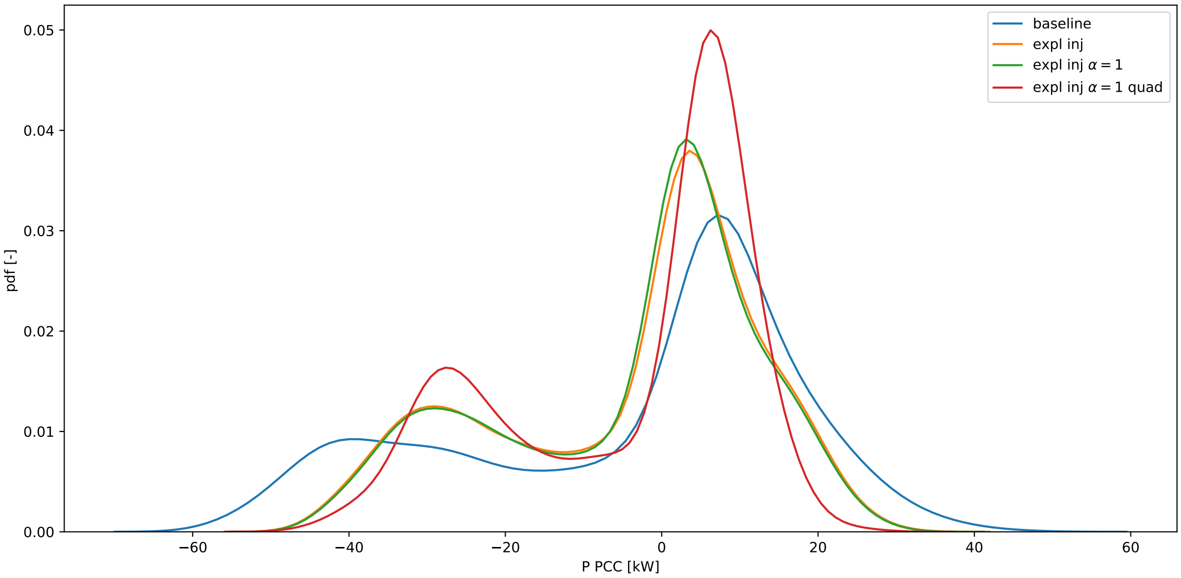


Figure 41: Probability density function (PDF) of the power at the PCC of the community. *baseline*: baseline simulation with no batteries, *expl inj*: explicit coordination, injection allowed, power limits at PCC enforced, *expl inj* $\alpha = 1$: explicit coordination, injection allowed, power limits at PCC enforced using Lagrangian dual variables, $\alpha = 1$, *expl inj* $\alpha = 1$ *quad*: explicit coordination, injection allowed, power limits at PCC enforced using quadratic punishment in the global objective function, $\alpha = 1$.

Another way to limit grid stress is to punish quadratically the excursions from a nominal value. This can be achieved by adding a quadratic punishment term to the objection function 1. Such strategy was applied to limit the power excursions at the coupling point. In this case the batteries try to flatten the profile as much as possible regardless of whether the grid constraints are violated or not. In this case, α was still set to 1, so that the batteries would only optimize against the community surplus. The effect on the grid of such strategy is shown in figures 40, 41, 42, 43 and 44. One can notice that, in this case the effect on the grid KPIs is more pronounced. The PDFs of power and mean voltage are narrower, and the losses, as expected, are lower. However, setting $\alpha = 1$ and adding a quadratic punishment in the objective function could theoretically worsen the financial balance of end-users. This is discussed in detail in section 8.2.2.

Water heaters Water heater control also leads to improved grid KPIs, although the effect is not as pronounced as with battery control. The effect of water heaters control on the power at the PCC of the community is shown in figure 45. It can be seen that the control of the boilers allows reducing the peaks of energy consumption significantly. At the same time, the effect on the reduction of power injection is minimal. This is mainly because the controllability of boilers is not as refined as in the case of batteries and presents inherent asymmetries. In fact, boilers can be forced off, but cannot be switched on on command if their internal controller based on hysteresis does not consider it necessary to charge. This, combined with the fact that it is not possible to force the boilers off for too long without risking compromising the comfort of the end-users, means that often when boilers are left free to turn on, they don't actually do so. This leads to an imperfect reduction of the negative peaks, which would require the boilers to turn on to be smoothed out. The effect the difference between implicit and explicit coordination is less pronounced. Raising the aggressiveness of the boiler control algorithms by lowering the turn-on time should help increase control performance against the grid KPIs. However, there is a risk of violating user comfort. In this case, the comfort settings are very conservative, as the same settings are used in the LIC pilot project. Additional simulations are planned with more aggressive settings.

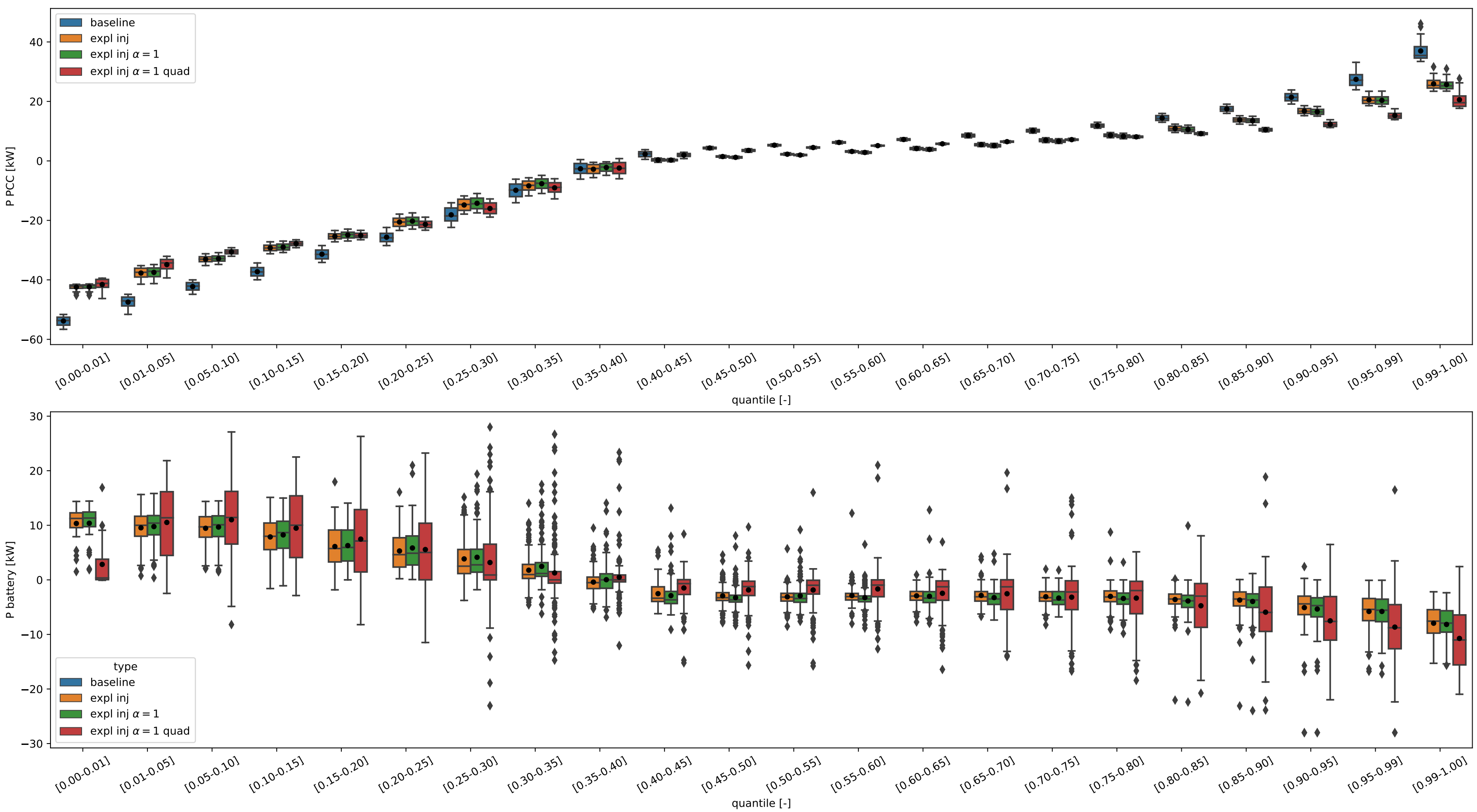


Figure 42: Effect of batteries control on the power at the point of common coupling (PCC) of the community. Top: boxplots of the power at the PCC per inter-quantile range. Bottom: boxplot of the total power of the batteries per inter-quantile range of the power at the PCC. The high variance of the batteries' power in the bottom plot starting from quantile 0.3 up to quantile 0.7 of the PCC's power, is due to the fact that the PCC's power is close to 0 in this range.

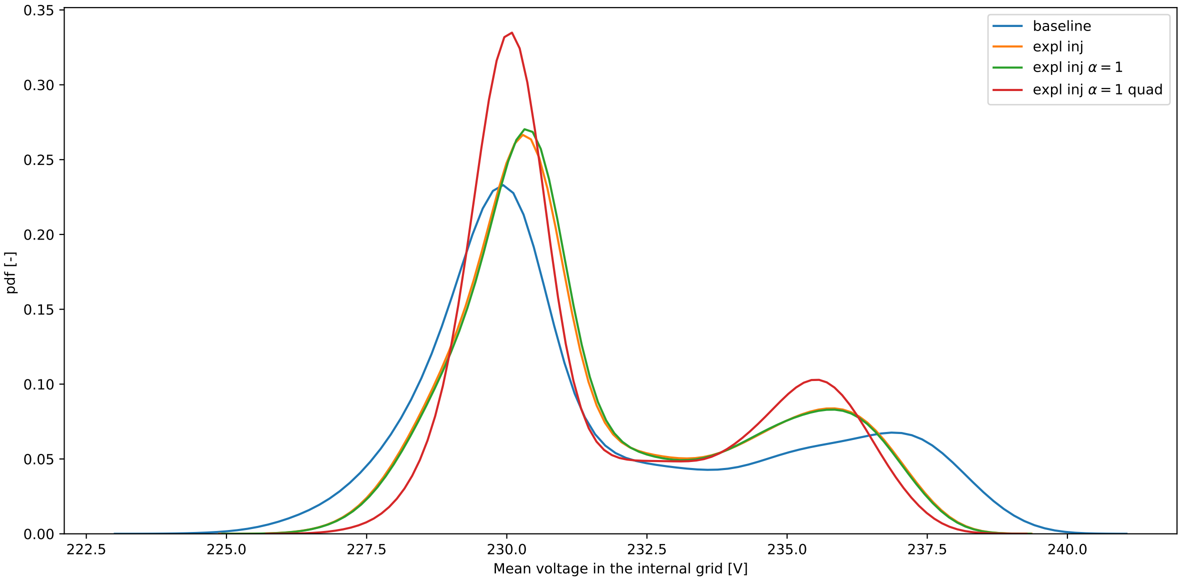


Figure 43: Probability density function (PDF) of the mean L-N voltage inside the community.

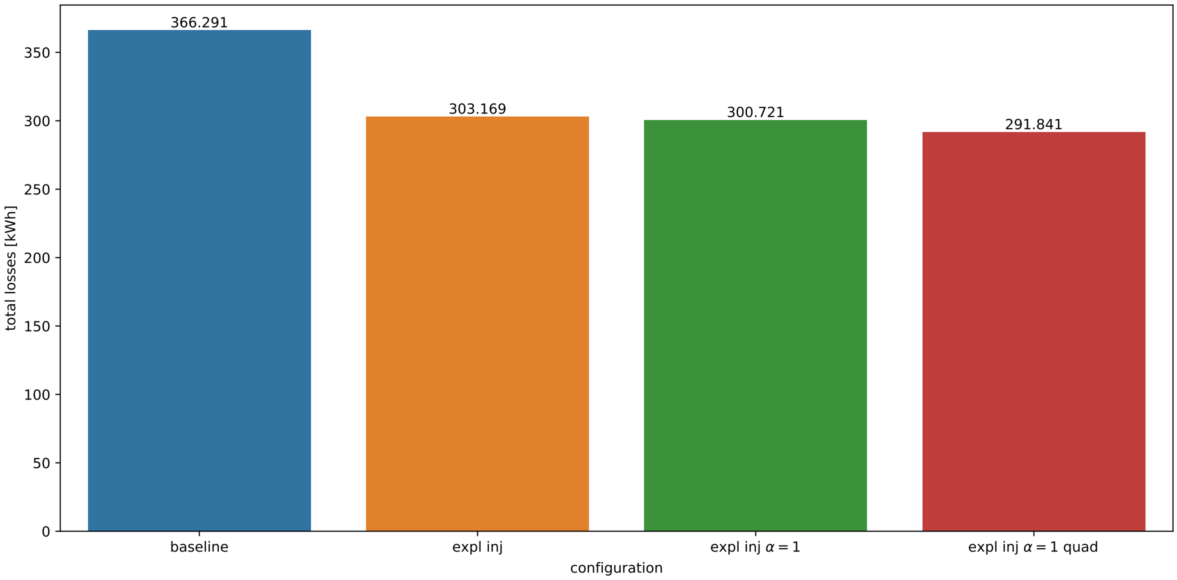


Figure 44: Total losses inside the community for the month of July.

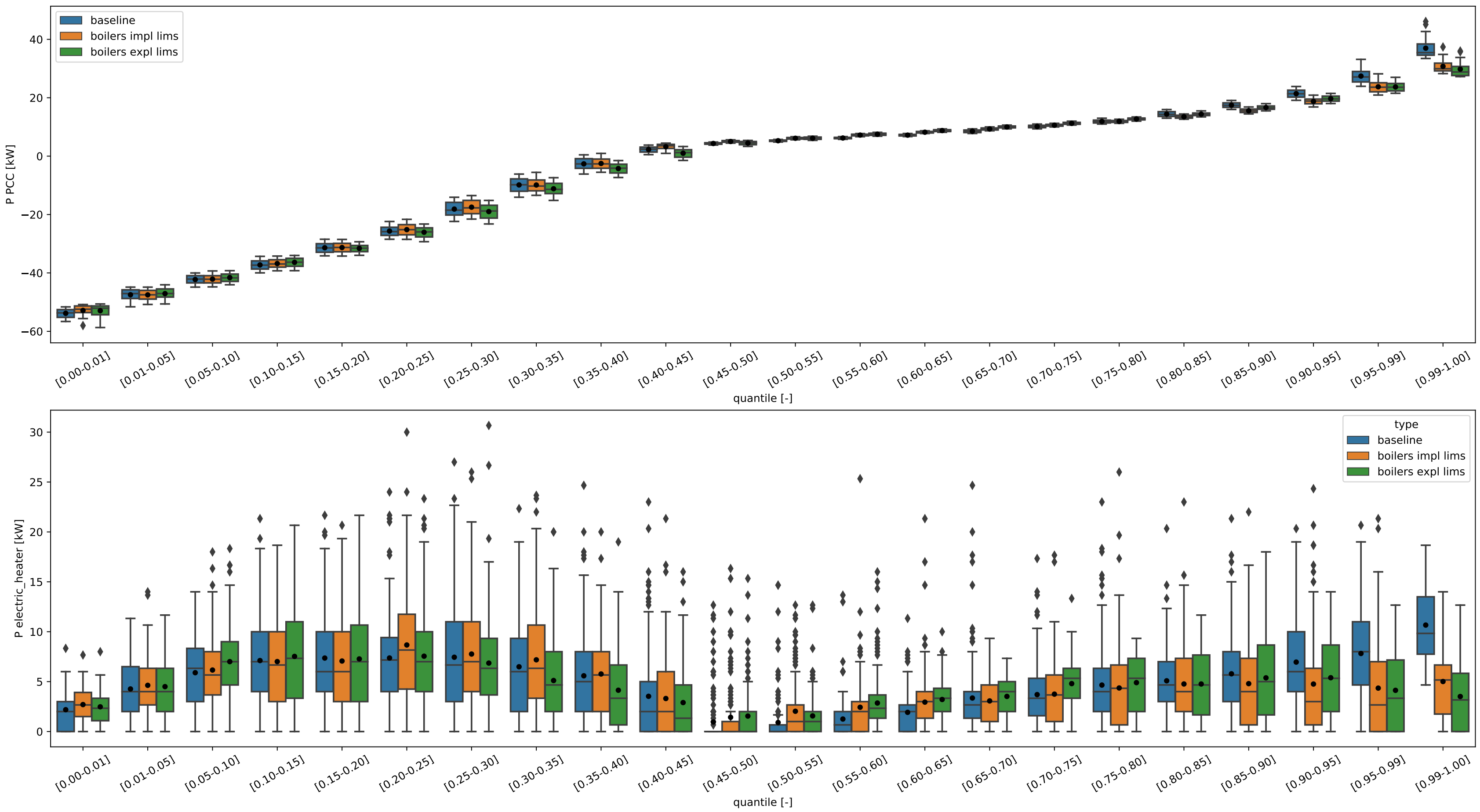


Figure 45: Effect of water heaters control on the power at the point of common coupling (PCC) of the community. Top: boxplots of the power at the PCC per inter-quantile range. Bottom: boxplot of the total power of the batteries per inter-quantile range of the power at the PCC.

8.2.2 Economic analysis

In the previous section, the different control strategies for batteries and water heaters have been evaluated against grid KPIs. We now focus on the financial performance for both the end-users and the administrator of the energy community. Two pricing schemes have been evaluated. The sharing-problem-based (SPB) pricing scheme against which the explicit control algorithms optimize, presented in section 3.1.3 and the LIC pricing scheme presented in section 2.2.2. In the first case, the community's surplus (equation (2)) is equally split among the end-users and the community manager. The half of the surplus accruing to end-users is then divided according to their contribution to the surplus according to the rule defined in equations (6)-(7). Users who contributed the most to the surplus receive more. The redistributed surplus from now on is named bonus.

In the second case, the energy prices applied inside the LIC community are used. In this case, the prices for selling to and buying from the community are fixed and summarized in table 9. The self-consumed energy is equally split among the community members proportionally to their consumption and production, which results in buying and selling prices that are dynamic, but that for a given 15min slot are the same for everyone.

	outside the community	inside the community (LIC case)
buying	21	16
selling	6	9

Table 9: Energy prices (in cts/kWh) applied in the evaluation of economic impact.

In both cases, the community administrator pays the bill at the coupling point, where the DSO's prices are applied and gets paid by the end-users according to the above-mentioned pricing schemes. The difference between the administrator costs and revenues is used to pay for the internal grid, cover the administrative costs, and ideally make some profit.

Both, the effect of batteries and water heaters control is evaluated in the next paragraphs.

Batteries The different control strategies presented in section 8.2.1 are evaluated. The total costs for the end-users and the administrator, for the SPB and LIC pricing schemes are shown in tables 10 and 11, respectively. The difference in the monthly costs with respect of the business as usual case is shown in figures 46 and 47, for the SPB and LIC pricing schemes, respectively. The business as usual pricing scheme is that in which the users are not members of an energy community. The baseline_scc case represents the case in which the users are members of a community, but no batteries are present.

It can be seen that the mere fact of being a member of a community generates considerable savings for end-users. And, as expected, the introduction of batteries, by encouraging self-consumption both locally and at the community level, further improves the financial situation, for the end-users and for the community administrator. The explicit coordination method in which agents communicate with each other is the most effective from a financial point of view. In both pricing schemes, the battery owners are among those who are profiting the most from being inside a community. This is desirable, as they have to amortise the cost of the battery and fair since their actions contribute the most to increase the community's self-consumption. At the moment, the preliminary results (one month simulations) are not enough to estimate a payback time for the installation of a BES system. Based on the simulations done for the month of July, considering the explicit no injection case reported in table 11 and the installation cost of a residential BES of 515 CHF/kWh (as per the Tesla Powerwall 2), we can esteem a lower bound for the average payback time for the end user would be of 8.9 years for a 10 kWh BES system.

In section 8.2.1, we also proposed two alternative models, in which the batteries only optimised the community's own consumption, regardless of the effect this might have on their individual costs: models *expl inj* $\alpha = 1$ and *expl inj* $\alpha = 1$ *quad*. Both of the above models are not guaranteed to reach a generalised Nash equilibrium and consequently have the potential to reduce savings (if not cause additional costs) to individual users. Their financial figures are shown in tables 12, 13 and figures 48, 49. As could already be seen by analysing performance against grid parameters (section 8.2.1), the model in which alpha is set equal to 1 differs little from that in which it is not, also with respect to financial performance.

	user costs bau	bonus	user costs tot	PCC costs	PCC balance
baseline bau	1476.3	0	1476.3	0	0
baseline scc	1476.3	-453	1023.3	600.5	-422.9
expl inj lims	1376	-519.6	856.4	360.4	-496
expl no inj	1386.3	-456.4	929.9	500.5	-429.4
expl no inj lims	1387.7	-465	922.8	484.2	-438.7
impl inj	1401.2	-519	882.2	387.2	-494.9
impl no inj	1400.5	-457.8	942.7	512.2	-430.5
no coordination	1412.2	-454.7	957.4	530.8	-426.6

Table 10: Total costs (in CHF) for the endusers and the administrator of the community using the surplus costs redistribution mechanism for the case of battery control. *baseline*: baseline simulation with no batteries, *no coordination*: batteries optimizing local self-consumption only, *impl no inj*: implicit coordination, no injection into the grid allowed, *impl inj*: implicit coordination, injection allowed, *expl no inj*: explicit coordination, no injection allowed, *expl no inj lims*: explicit coordination, no injection allowed, power limits at PCC enforced, *expl inj lims*: explicit coordination, injection allowed, power limits at PCC enforced.

As expected, the community administrator's financial figures improve when $\alpha = 1$ since the batteries only optimise for the community surplus, which directly translates to higher revenues for the administrator. One can also see that the total costs for the end-users are slightly lower in the $\alpha = 1$ case in both pricing schemes. However, the average costs for battery owners are slightly higher (figures 48 and 49), which is consistent with the fact that they are sacrificing part of their personal revenues in order to increase the general welfare of the community. These effects are even bigger in the case in which the batteries are contributing to the reduction of quadratic punishment at the coupling point. In this case, the administrator is earning more money, but this is not the case for the community members, especially the battery owners. This is because there is actually no financial incentive to mitigate power excursions at the coupling point and, therefore, the battery actions are not sufficiently rewarded. In this analysis, we did not include remuneration schemes to reward users for helping to comply with network constraints. For the case in which constraints are enforced using upwardly trimmed Lagrangian multipliers, putting a hard cap on them corresponding to the price the grid operator is willing to pay to enforce the respect of the grid constraint is a fairly straightforward way to implement such mechanism, as the actual costs are directly reflected in the optimisation function of each individual user. Such a mechanism could also be applied to perform peak shaving actively. In a big enough community, the administrator will need to pay peak costs at the coupling point. Peak mitigation could be implemented exactly in the same way in which grid constraints are enforced, using Lagrangian multipliers. This concept is easily explained using a concrete example. The current monthly community peak to date is 50kW. In this case, the administrator will set a constraint on the maximum power at the coupling point to 50kW and reward those who allow the power to remain below the set limit by placing a hard cap on Lagrangians that corresponds to a reward significantly higher than the average cost of energy. For example, the administrator could reward the end-users with 0.1CHF/kW, which for a 15-minute step would correspond to 0.4CHF/kWh. It is not trivial, however, to understand if the administrator will save money by doing so. This is due to the highly non-linear nature of the existing billing system, which is based on the maximum monthly peak. In other words, if for 15 minutes in a month something goes wrong (e.g. a battery goes offline, or the forecasts are extremely bad), all the work done by the batteries, and remunerated by the administrator, is wasted. This topic is being currently investigated and will be discussed in future reports in the context of the LIC project.

	user costs	PCC costs	PCC balance
baseline bau	1476.3	0	0
baseline scc	993.1	600.5	-392.7
expl inj lims	821.7	360.4	-461.4
expl no inj	899.5	500.4	-399.0
expl no inj lims	891.8	484.2	-407.7
impl inj	847.6	387.2	-460.3
impl no inj	912.2	512.2	-400.0
no coordination	927.1	530.8	-396.3

Table 11: Total costs (in CHF) for the endusers and the administrator of the community using the AMM costs redistribution mechanism for the case of battery control

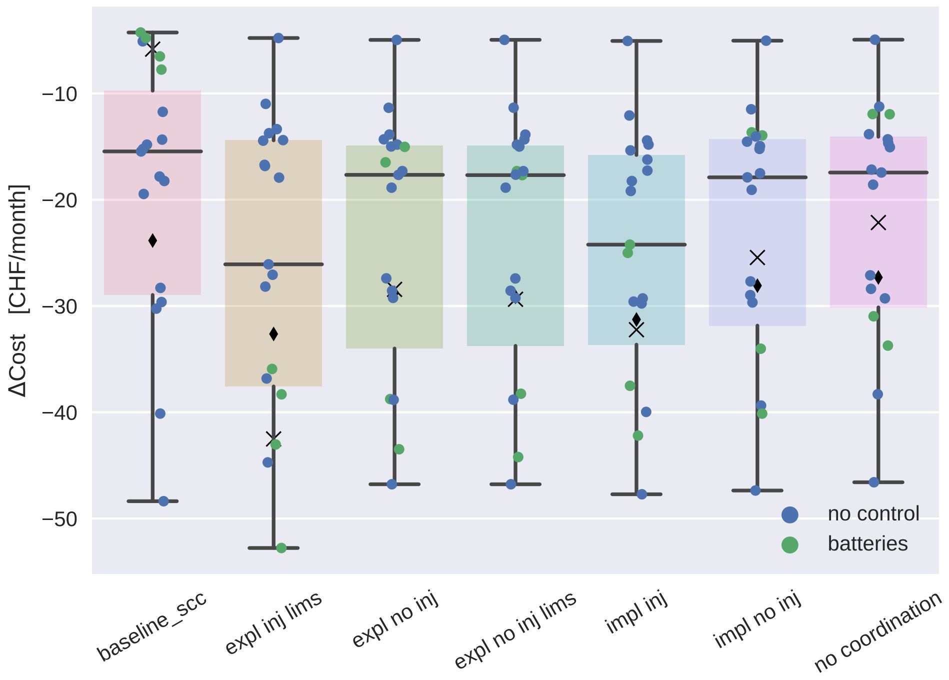


Figure 46: Boxplots of the difference in the monthly costs using SPB pricing scheme with respect of the business as usual case (in which the end-users are not part of an energy community), for the simulated month of July. The box shows the quartiles of the dataset while the whiskers extend to show the rest of the distribution, except for points that are determined to be outliers. Points that extend beyond 1.5 times the inter-quartile range, i.e. $v < q25 - 1.5(q75 - q25)$ or $v > q75 + 1.5(q75 - q25)$ are considered outliers. Black diamond: average price reduction for all the users. Black X: average price reduction for users with batteries.

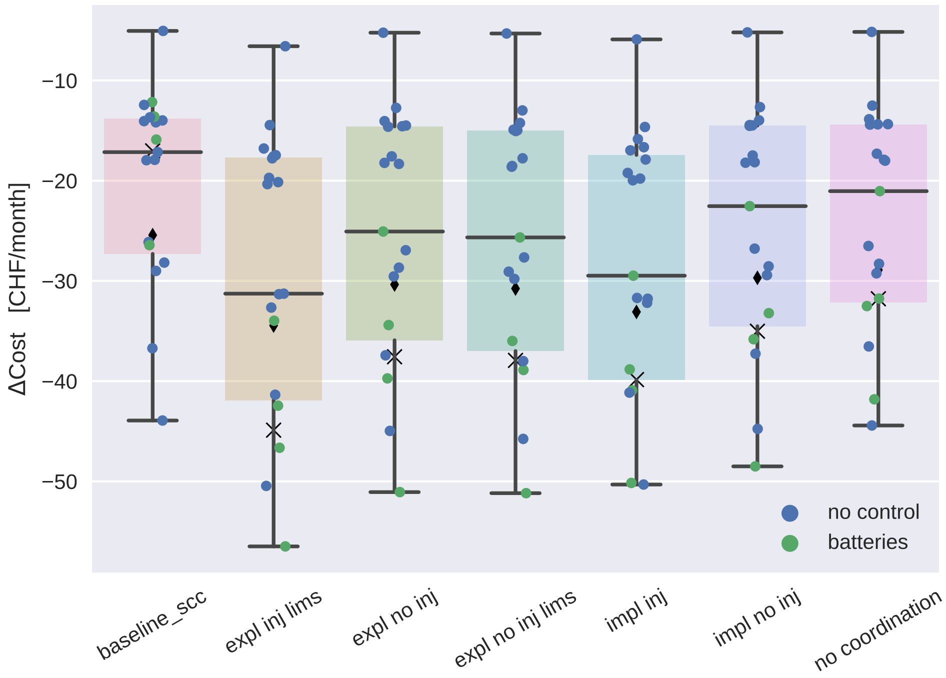


Figure 47: Boxplots of the difference in the monthly costs using LIC pricing scheme with respect of the business as usual case (in which the end-users are not part of an energy community), for the simulated month of July. Black diamonds: average price reduction for all the users. Black Xs: average price reduction for users with batteries.

	user costs bau	bonus	user costs tot	PCC costs	PCC balance
baseline bau	1476.3	0	1476.3	0	0
baseline scc	1476.3	-453.0	1023.3	600.5	-422.9
expl inj	1376.0	-519.6	856.4	360.4	-496.0
expl inj $\alpha = 1$	1391.2	-536.3	854.8	341.9	-512.9
expl inj $\alpha = 1$ quad	1438.8	-563.5	875.3	333.9	-541.4

Table 12: Total costs (in CHF) for the endusers and the administrator of the community using the explicit costs redistribution mechanism for the case of battery control, setting $\alpha = 1$ and using quadratic punishment.

	user costs	PCC costs	PCC balance
baseline bau	1476.3	0	0
baseline scc	993.1	600.5	-392.7
expl inj	821.7	360.4	-461.4
expl inj $\alpha = 1$	819.1	341.9	-477.1
expl inj $\alpha = 1$ quad	837.8	333.9	-503.9

Table 13: Total costs (in CHF) for the endusers and the administrator of the community using the implicit costs redistribution mechanism for the case of battery control, setting $\alpha = 1$ and using quadratic punishment.

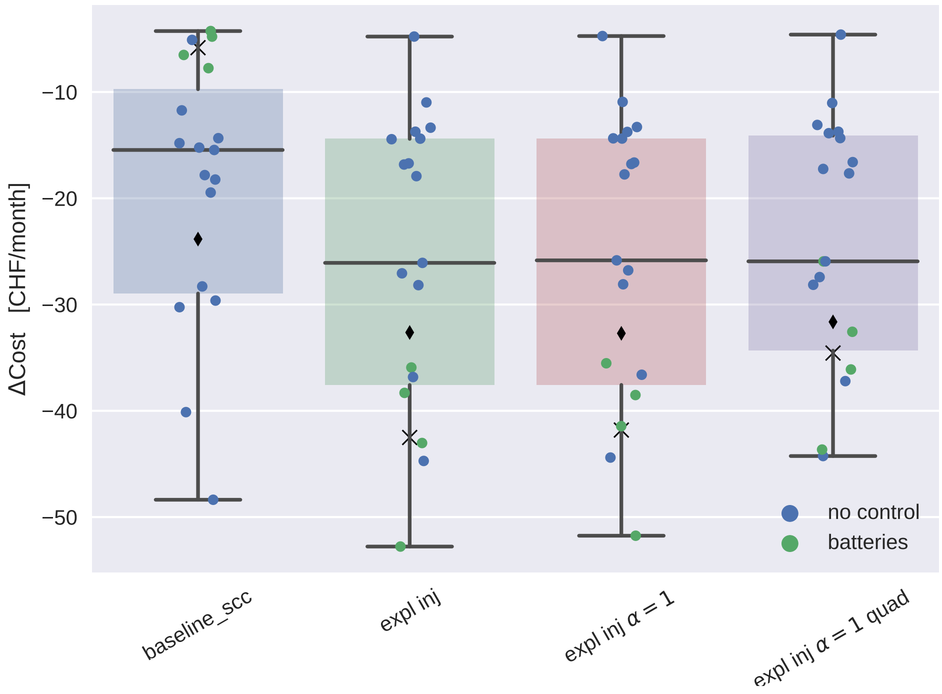


Figure 48: Difference in the monthly costs using SPB pricing scheme, with respect of the business as usual case, in which the end-users are not part of an energy community, for the simulated month of July. Batteries control case, setting $\alpha = 1$ and using quadratic punishment.

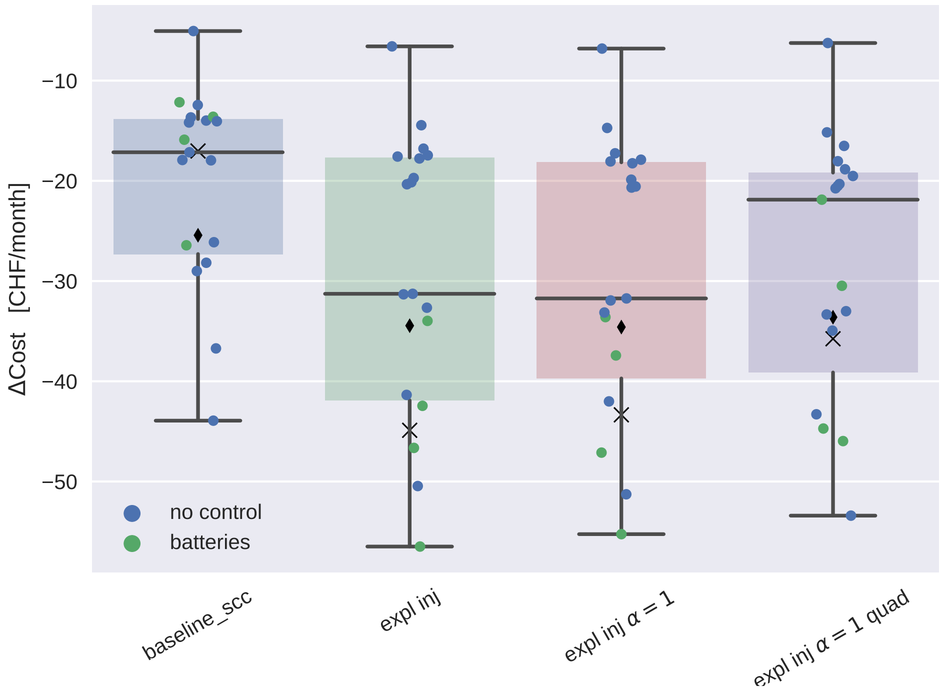


Figure 49: Difference in the monthly costs using LIC pricing scheme, with respect of the business as usual case, in which the end-users are not part of an energy community, for the simulated month of July. Batteries control case, setting $\alpha = 1$ and using quadratic punishment.

Water heaters Water heater control also leads to improved financial figures, although the effect is not as pronounced as with battery control, as shown in tables 14, 15 and figures 50, 51. As already mentioned in section 8.2.1, the lack of controllability of the water heaters (they can be forced off, but cannot be switched on on command) and the stringent constraints imposed on user comfort are limiting the effect of boilers control. Raising the boiler control algorithms' aggressiveness by lowering the turn-on time should help increase control performance against the financial KPIs, with the risk of violating user comfort. Additional simulations are planned with more aggressive settings.

	user costs bau	bonus	user cost tot	PCC costs	PCC balance
baseline scc	1476.3	-453.0	1023.3	600.487641	-422.7
expl lims	1442.3	-441.4	1000.8	588.231841	-412.6
impl lims	1454.0	-454.0	999.9	574.382804	-425.5

Table 14: Total costs (in CHF) for the endusers and the administrator of the community using the explicit costs redistribution mechanism for the case of boiler control.

	user costs	PCC costs	PCC balance
baseline scc	993.1	600.5	-392.7
boilers expl lims	971.4	588.2	-383.2
boilers impl lims	969.7	574.4	-395.3

Table 15: Total costs (in CHF) for the endusers and the administrator of the community using the implicit costs redistribution mechanism for the case of boiler control.

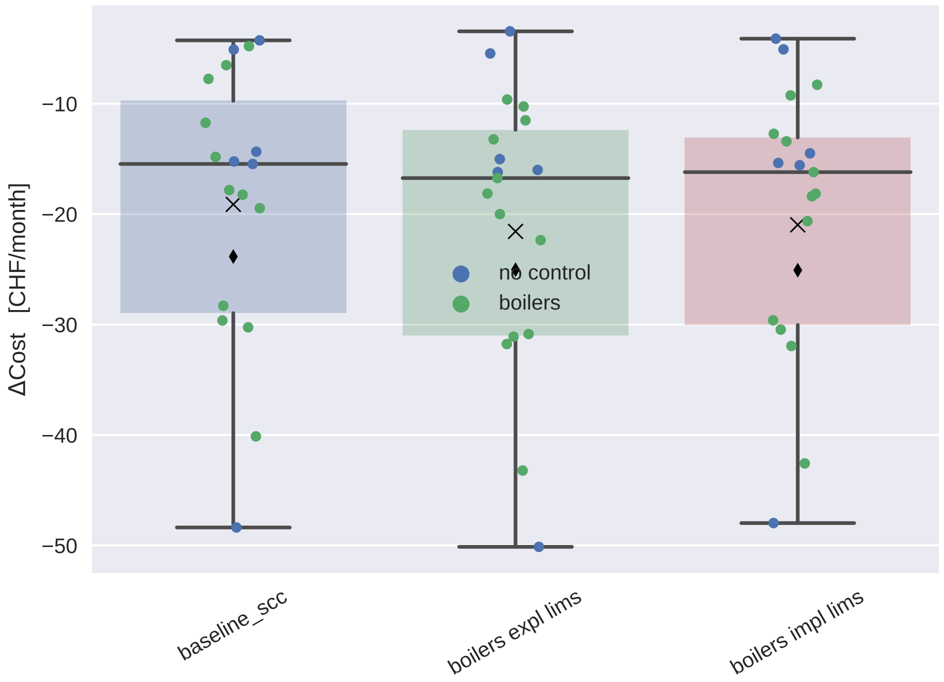


Figure 50: Difference in the monthly costs using SPB pricing scheme, with respect of the business as usual case, in which the end-users are not part of an energy community, for the simulated month of July. Water heaters control case.

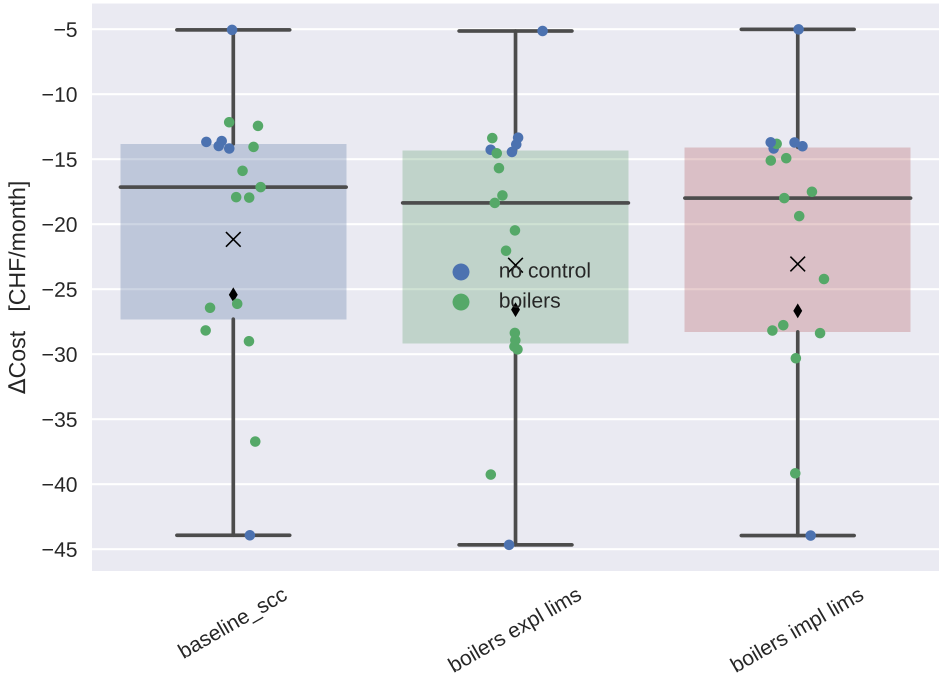


Figure 51: Difference in the monthly costs using LIC pricing scheme, with respect of the business as usual case, in which the end-users are not part of an energy community, for the simulated month of July. Water heaters control case.

8.3 Single level EC with power based tariffs simulations

We investigated the role of applying a power based tariff on top of the AMM prices generated by the single level EC market formulation. The coordination problem 1 is solved making use of the preconditioned forward backward algorithm. We compared three scenarios:

1. *Baseline*: no batteries
2. *LEM + grid constraints*: batteries coordinate to minimize the LEM prices, using (37) as objective function
3. *LEM + LFM*: batteries coordinate to minimize the LEM and LFM prices, using (36) as objective function, but without the term including λ

8.3.1 Grid analysis

For the grid constraint, the two limits for the maximum positive and negative power at the coupling point were chosen based on an estimate of how much could ideally be steered using the available batteries if one had perfect forecasts. The limits have been selected based on a baseline simulation of the energy community without batteries. The limits have been selected to be as low as possible, given that the following two criteria are respected:

- The maximum daily energy exceeding the positive and negative limits must be smaller than the total energy storable in the batteries. This assumption means that in the worst day, all the batteries should have been empty (respectively full) to fulfil negative (respectively positive) grid constraints.
- The difference between the minimum limit and the quantile 0.01 of the power at the coupling point and between the quantile 0.99 of the power at the coupling point and the maximum limit must be smaller than the maximum charge and discharge power of the batteries, respectively.

Among the PCC's power density distribution, these limits are shown in detail for the month of July in the following figure. We stress that these limits were chosen automatically based on yearly simulation in which no devices were controlled.

The purpose of these simulations is to see the LFM price structure's effectiveness over explicitly integrating grid constraints into the distributed control problem. In figure 53, the effect on the PDF of the power at the LIC's PCC is shown for the three cases. In both the controlled instances, the batteries successfully shrink the PDF towards zero. It is also interesting to see how, for the case in which grid constraints where explicitly considered, the limits set on the PCC power were not respected. This can be explained as the effect of the imperfect energy forecasts: the Lagrangian multipliers handling grid constraints are non-zero only in cases in which their violations are correctly forecasted. On the other hand, adding a quadratic punishment further shrinks the distribution towards zero; since the presence of this quadratic term is not dependent on the quality of the forecasts, batteries can manage to shrink the tails of the power distribution further.

8.3.2 Economic analysis

In this section, we evaluated the effect of the parameters of the LEM and LFM prices on the overall energy costs for end-users. Figure 54 shows the effect of changing p_b^{P2P}, p_s^{P2P} and the LFM quadratic parameter β on the final costs for the LIC end users, without controlling any device. The costs are computed using one-year real data from the pilot site for 2020. The p_b^{BAU} and p_s^{BAU} are fixed to the DSO prices for LIC, which are 6 and 21 CHF cts/kWh, respectively. In the next figure, p_b^{P2P}, p_s^{P2P} are jointly changed between 8-10 and 14-18 CHF cts, respectively, when LFM is not active. The effect on the users' yearly costs is shown in term of yearly bonus, i.e. cost reduction w.r.t. the BAU case. We recall that when only LEM is active, all the users always have a net benefit. The left panel shows the yearly

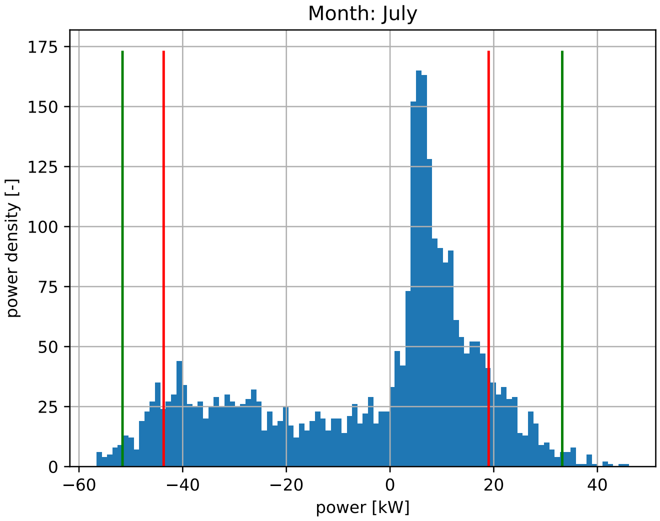


Figure 52: Power limits for the simulated month of July. Blue bars: histogram of the active power at the coupling point of the community. Green vertical lines: quantiles 0.01 and 0.99. Red bars: selected negative and positive power limits (-43.67kW and 19.02kW, respectively).

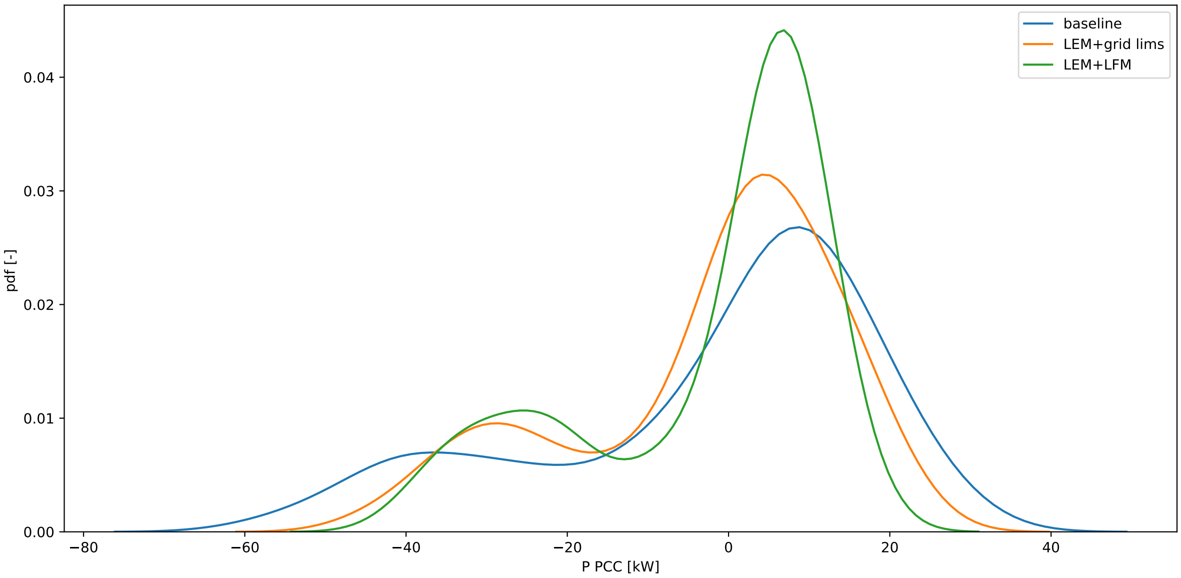


Figure 53: Probability density function (PDF) of the PCC's power for the baseline, with coordination, LEM prices and and explicit grid limits, and with LEM and LFM prices.

bonus for users without a roof-mounted PV power plant. In this case, their yearly bonus is not affected by the value of p_s^{P2P}, as we would expect. On the other, linearly decreasing the p_b^{P2P} linearly increases the bonus. For those users having a PV, the right panel shows that the effect of p_s^{P2P} is predominant over p_b^{P2P}. As most of the users don't have a PV power plant in LIC, energy production is still a scarce resource, and users who can sell energy have a higher economic bonus.

Figure 54 shows the same sensitivity analysis in terms of percentages. While the users without PV have a bonus in the range of 1-6$\%$ of their yearly consumption, the range for PV owners is substantially higher,

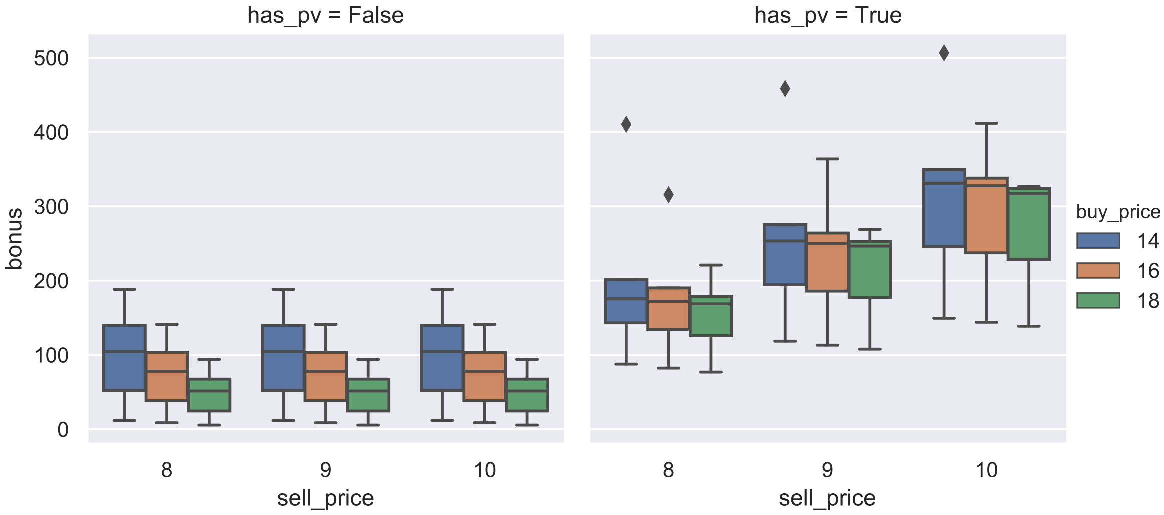


Figure 54: Effect of different p_b^{P2P}, p_s^{P2P} LEM prices on the final yearly cost for the end-users. Boxplots contain differences w.r.t. BAU. Left: users without roof-mounted PV power plants, right: with.

up to 12%. The higher outlier is the kindergarten, in which only the community PV is present.

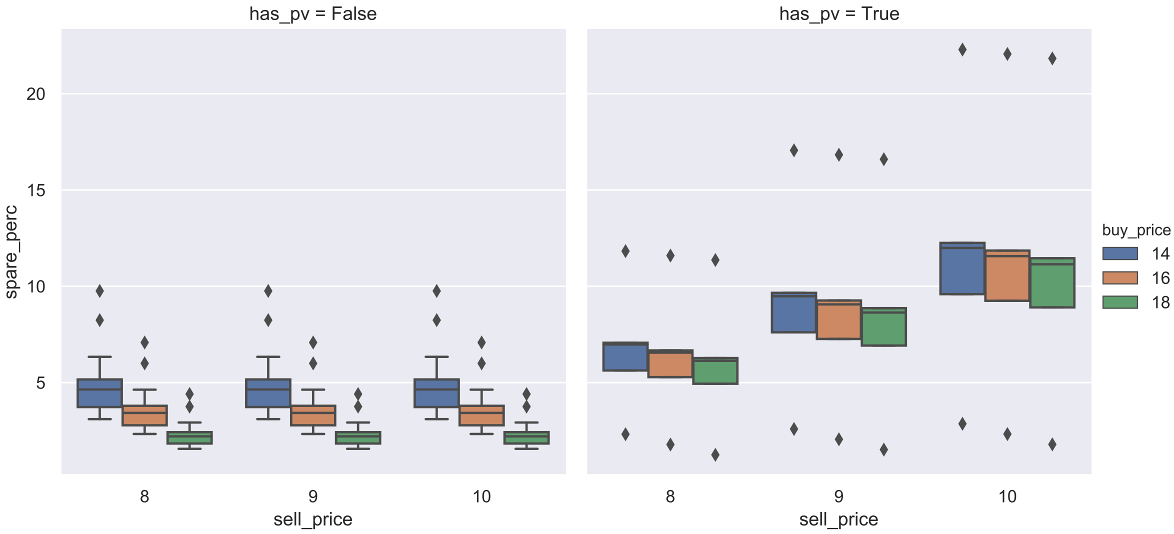


Figure 55: Effect of different p_b^{P2P}, p_s^{P2P} LEM prices on the final yearly cost for the end-users, as percentages

Figure 56 shows the effect of the LFM β [cts/kWh²] parameter on the yearly savings in terms of CHF, grouped by PV owners or net consumers. We fixed p_b^{P2P}, p_s^{P2P} to 16 and 9 cts, respectively. As previously explained, the LFM pricing scheme doesn't guarantee a cost reduction for end-users, as is the case for the LEM prices scheme. On the contrary, the LFM mechanism applies a bonus-malus scheme depending on the current contribution of the end users in shrinking the overall power profile to zero; as such, we can expect that for high values of β some users face a cost higher than the BAU case. As this is not desirable, the β parameter must be tuned for the specific market to which it is applied. This can be done in silico, based on historical production and consumption data, as in this case.

In this situation the PV owners are less penalized by the increase of the LFM β parameter, since energy production is the scarce resource inside the market. The converse is true when we look at simulation cases, in which, we recall, additional 33kWp of roof-mounted PV plants where installed. Under this condition, in the month of July we have an overproduction from PV power plants, which leads to a decrease of savings in CHF for PV users, if they don't optimize their power profile curves considering the LFM prices. In the following figure, these use cases where considered:

- No control. Users without PV nor battery (blue), users with PV+ battery (orange)
- Self-consumption optimization. Users with PV + battery (red), optimize for their self-consumption. Other users (green) take no actions.
- LEM+LFM: Users with PV + battery (violet), optimize for the LEM + LFM prices. Other users (brown) take no actions.

In the case in which the LFM is not active (β=0), the first four cases are not significantly different, while the savings increase when batteries are operated in order to directly optimize for the LEM prices. We stress that, in this case, the benefit of doing so also affects users who do not possess a PV nor control a battery (violet boxplot). When the LFM is activated, at increasing values of beta, PV owners see a reduction of their savings. This is because of the overproduction of PV in July. On the other hand, users without PV nor batteries face no significant changes w.r.t. the case in which only the LEM is active. Finally, we see that when users with a controllable battery use it to optimize the actual LEM+LFM prices, they still increase their final savings, even for high values of β.

8.4 DSO planned, centrally controlled battery

The distributed control strategy was compared to a centralized control strategy, in which a single battery is installed by the community manager and is operated in order to optimize the DSO's requirements for network stability. This corresponds to the DSO planned scenario of NEMoGrid. In this case, the DSO's needs were translated into a minimization of power excursions at the grid connection point. The central battery was operated following a lexicographic control scheme. The problem is solved by running two consecutive optimizations. The first one just optimizes for the self consumption of the EC, minimizing f_{sc}, which in our case, since the buying price is always higher than the selling price, can be reduced to a simple cost optimization function at the coupling point of the community:

$$f_{sc} = \sum_{t}^{T} c_{PCC} \tag{51}$$

The second optimization problem minimizes a peak-shaving objective function f_{ps}, which consists of the squared sum of the active power at the coupling point of the community over the optimization horizon:

$$f_{ps} = \sum_{t}^{T} (P_{PCC})^2 \tag{52}$$

subject to the following hard constraint:

$$f_{sc} \leq \eta f_{sc}^{*} \tag{53}$$

where f_{sc}^{*} is the optimal value found by the first optimization. In our case $\eta = 1$. This corresponds to the case in which helping the DSO does not reduce the EC's self consumption.

In other words, the first optimization finds the minimum costs attainable operating the battery; the second optimization try to perform peak shaving while not worsening the optimal cost found by the first optimization. Like in the previous sections, we used the LIC community as a benchmark case, and compared the effect of central battery control against the distributed batteries approach. Analogously to the real-world case, a single 60kWh / 60kW battery located in the kindergarten was used. The total capacity of the batteries in both the distributed and the central case is the same.

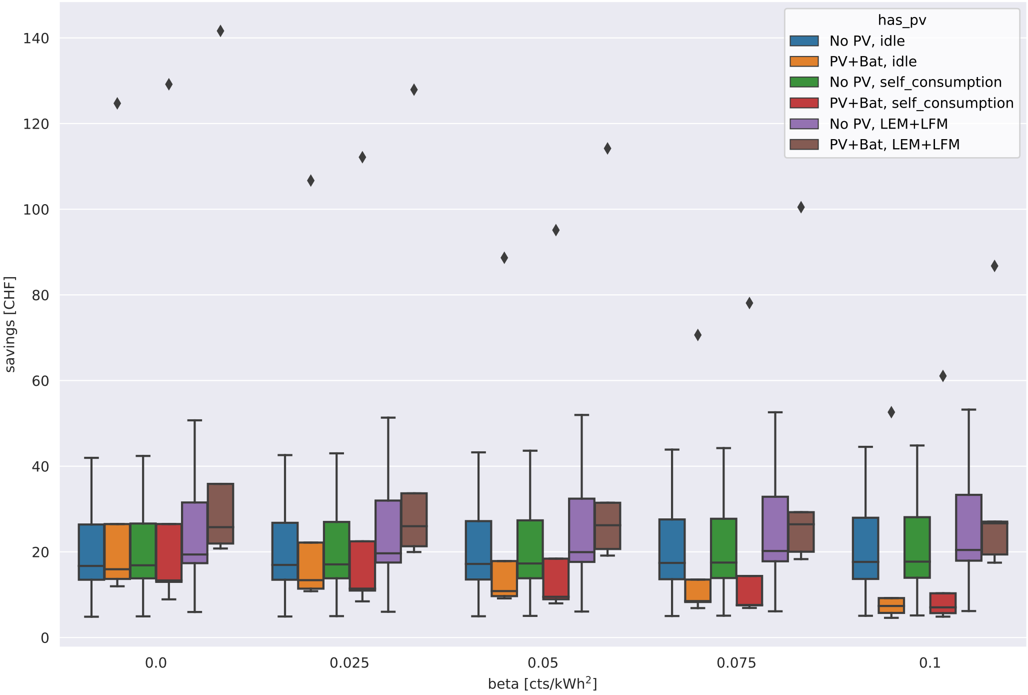


Figure 56: Boxplots of savings for the simulated month of July, for different use cases and groups of users. Blue, orange: no control, users with/without PV and battery. Green, red: self-consumption optimization, users with/without PV and battery. Violet, brown: LEM+LFM optimization, users with/without PV and battery.

8.4.1 Grid analysis

Power at the coupling point of the community. Since the centrally controlled battery maximizes the welfare of the community, the use case should correspond to the welfare maximization achieved by setting $\alpha = 1$ and adding a quadratic punishment to the power excursions in the objective function (1) when operating the batteries using the explicit distributed coordination. However, the optimization problem which is solved is different. In the central battery case, the optimization is solved lexicographically; in the distributed control case, the problem is solved iteratively using a preconditioned Forward-Backward method. Therefore, it is interesting to see if there is a difference in the performance of the two algorithms in optimizing the grid constraints, which, in our case, correspond to a soft constraint on the power at the community coupling point. Figure 57 shows the PDF of power at the simulated coupling point and figure 58 allows to further analyse the effects of batteries on the active power at PCC, by displaying statistics of the total controlled power as a function of the power at the PCC. It can actually be seen that the results of centralized battery control and the *expl inj* $\alpha = 1$ *quad* version are extremely similar, which supports the thesis that the ultimate goal of the optimization is the same.

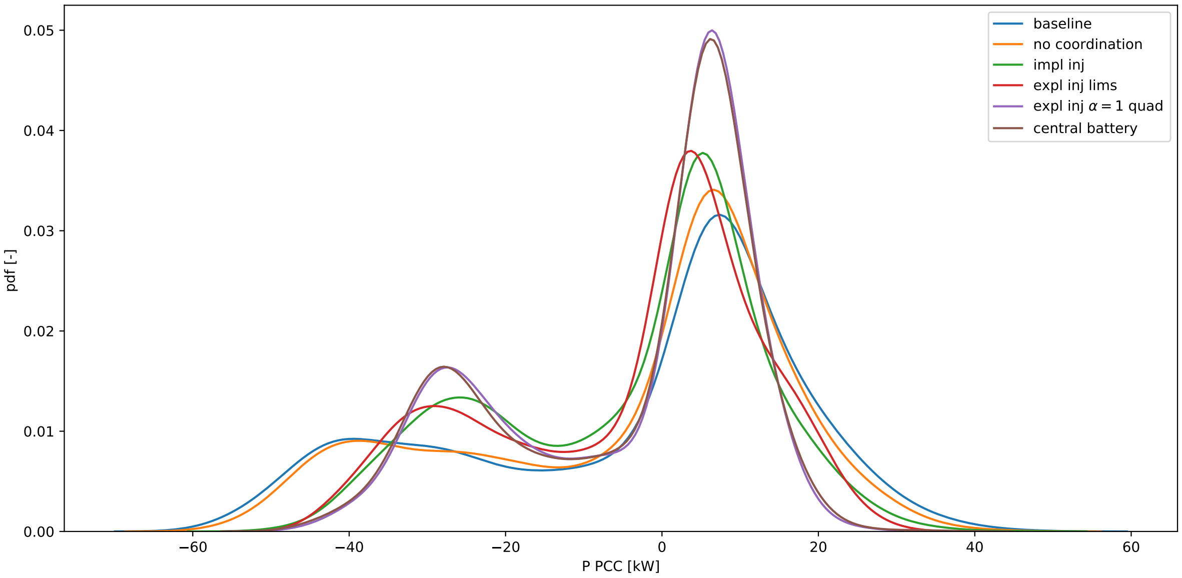


Figure 57: Probability density function (PDF) of the power at the PCC of the community. *baseline*: baseline simulation with no batteries, *no coordination*: batteries optimizing local self-consumption only, *impl inj*: implicit coordination, injection allowed, *expl inj lims*: explicit coordination, injection allowed, power limits at PCC enforced. *expl inj* $\alpha = 1$ *quad*: explicit coordination, injection allowed, power limits at PCC enforced using quadratic punishment in the global objective function, $\alpha = 1$. *central battery*: centrally operated battery.

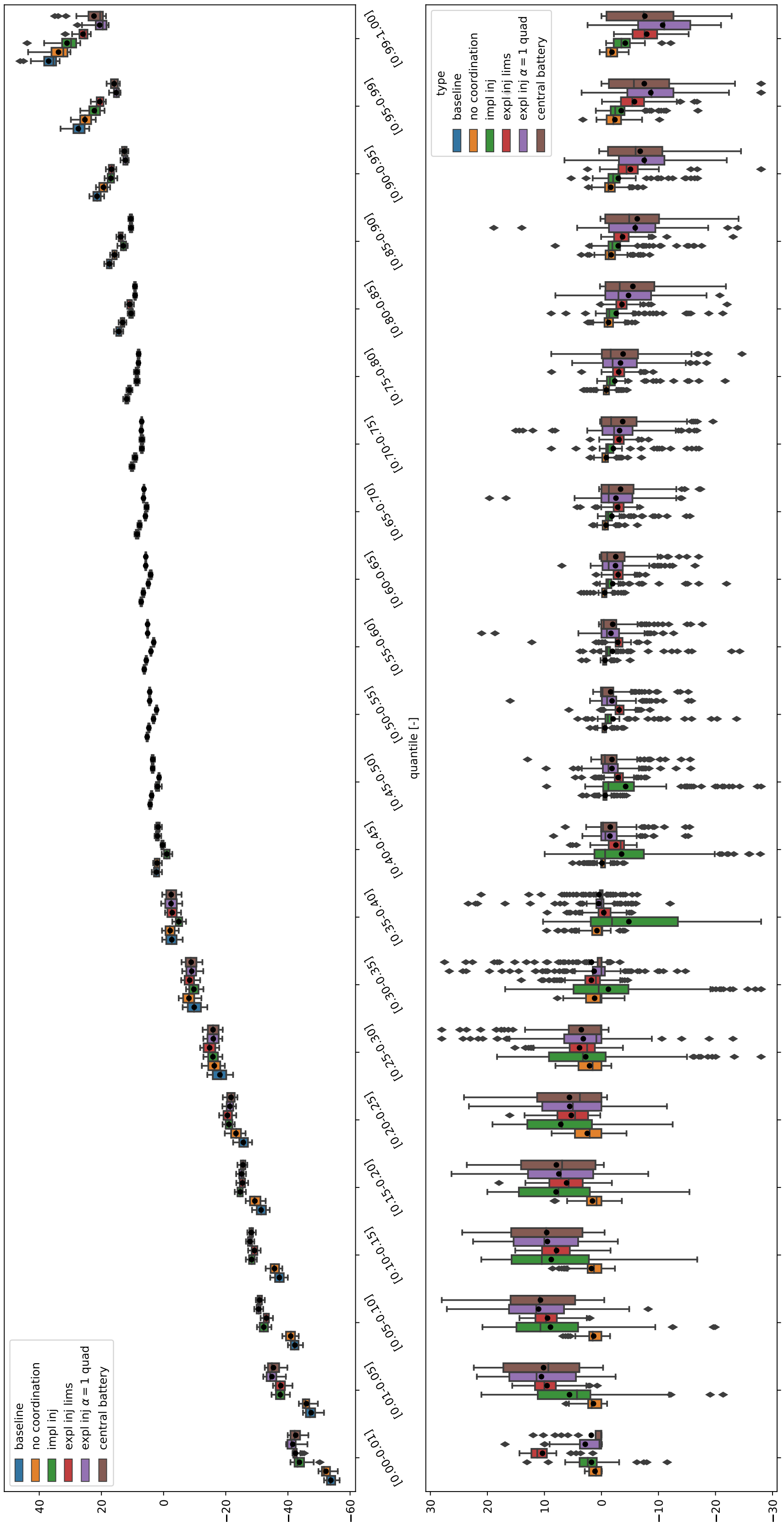


Figure 58: Effect of batteries control on the power at the point of common coupling (PCC) of the community with central and distributed batteries. Top: boxplots of the power at the PCC per inter-quantile range. Bottom: boxplot of the total power of the batteries per inter-quantile range of the power at the PCC. The high variance of the batteries' power in the bottom plot starting from quantile 0.3 up to quantile 0.7 of the PCC's power, is due to the fact that the PCC's power is close to 0 in this range.

Voltage and losses. Figure 59 shows the PDF of the L-N voltage at all the nodes and all the phases in the community. Also in this case, the difference between *expl inj* $\alpha = 1$ *quad* and the centrally controlled battery is barely noticeable. However, when concentrating on the kindergarten node, a difference should be perceivable. In fact, in LIC, the line connecting the kindergarten to the distribution box is particularly long, generating significant voltage excursions. In the distributed control case, no battery is present in the kindergarten. The batteries are instead installed in houses with PV. A battery placed at the kindergarten should help limit the voltage excursions at its node more effectively than the indirect effect of the distributed batteries. Figure 68 shows the PDF of the voltage at the kindergarten node. It is clear in this case that, as expected, the voltage is stabilized by the introduction of the battery at the node, even if the distributed batteries case still as an indirect impact. Figure 61 shows the time course of voltage at all the nodes and all the phases of the simulated LIC community for the first three days of July. One can see that the kindergarten is indeed the most problematic node of the grid and that installing the battery at its node is the optimal solution in this case.

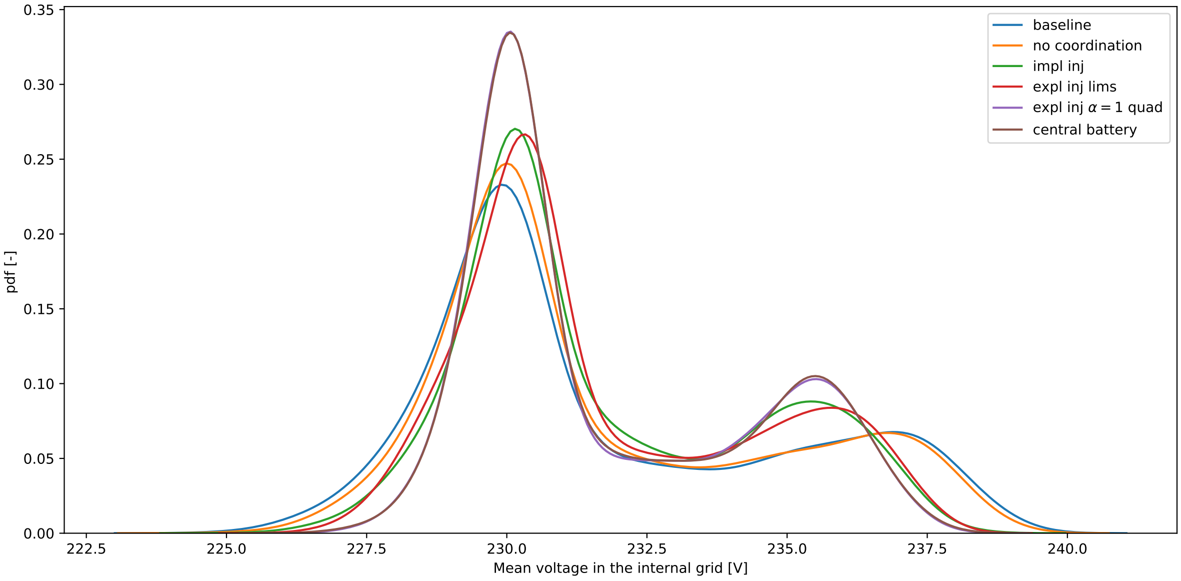


Figure 59: Probability density function (PDF) of the L-N voltage inside the community.*baseline*: baseline simulation with no batteries, *no coordination*: batteries optimizing local self-consumption only, *impl inj*: implicit coordination, injection allowed, *expl inj lims*: explicit coordination, injection allowed, power limits at PCC enforced. *expl inj* $\alpha = 1$ *quad*: explicit coordination, injection allowed, power limits at PCC enforced using quadratic punishment in the global objective function, $\alpha = 1$. *central battery*: centrally operated battery.

This improvement in the voltage KPI also reflects a substantial decrease in losses. Indeed, figure 62 shows that battery placement in the node of the kindergarten helps to decrease the losses within the community more than in the case where the batteries are distributed in the nodes of the houses with PV.

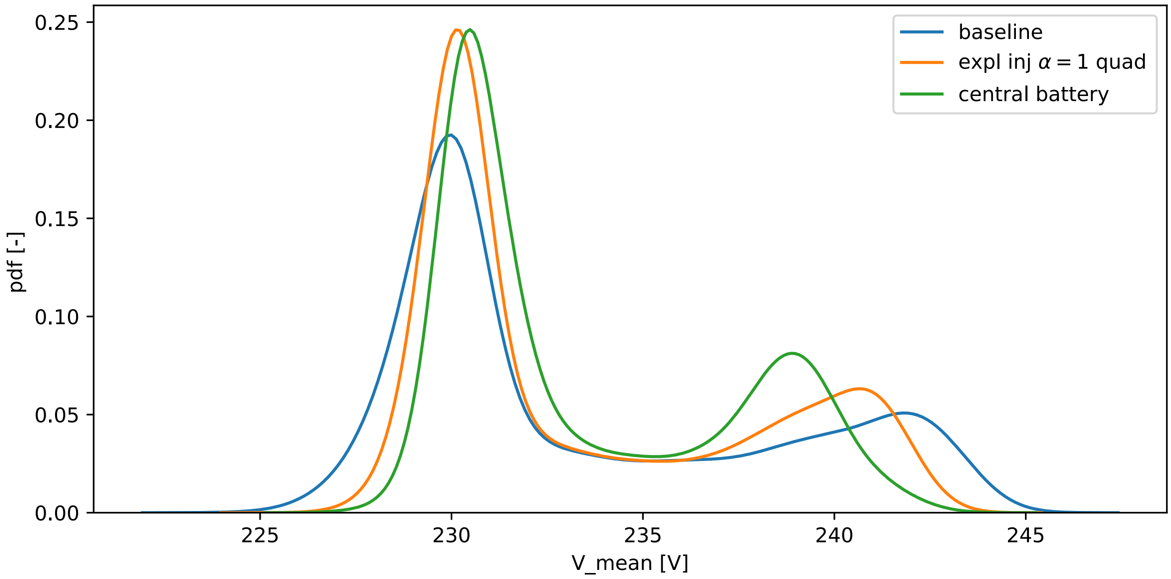


Figure 60: Probability density function (PDF) of the L-N voltage at the kindergarten node.*baseline*: baseline simulation with no batteries, *no coordination*: batteries optimizing local self-consumption only, *impl inj*: implicit coordination, injection allowed, *expl inj lims*: explicit coordination, injection allowed, power limits at PCC enforced. *expl inj* $\alpha = 1$ *quad*: explicit coordination, injection allowed, power limits at PCC enforced using quadratic punishment in the global objective function, $\alpha = 1$. *central battery*: centrally operated battery.

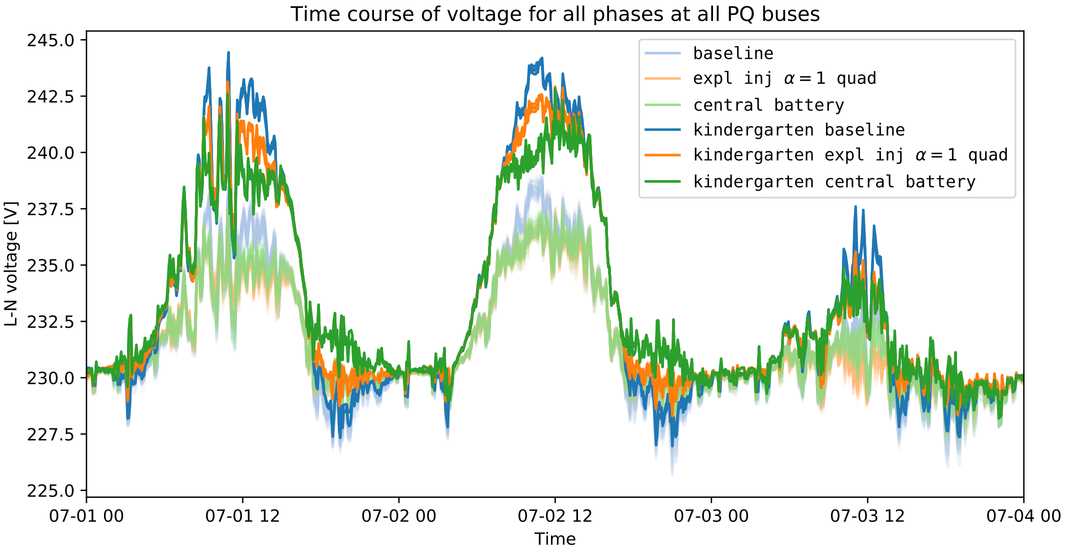


Figure 61: Time course of voltage at all nodes and phases of the community for the first three days of July. The kindergarten node is highlighted. *baseline*: baseline simulation with no batteries, *expl inj* $\alpha = 1$ *quad*: explicit coordination, injection allowed, power limits at PCC enforced using quadratic punishment in the global objective function, $\alpha = 1$. *central battery*: centrally operated battery.

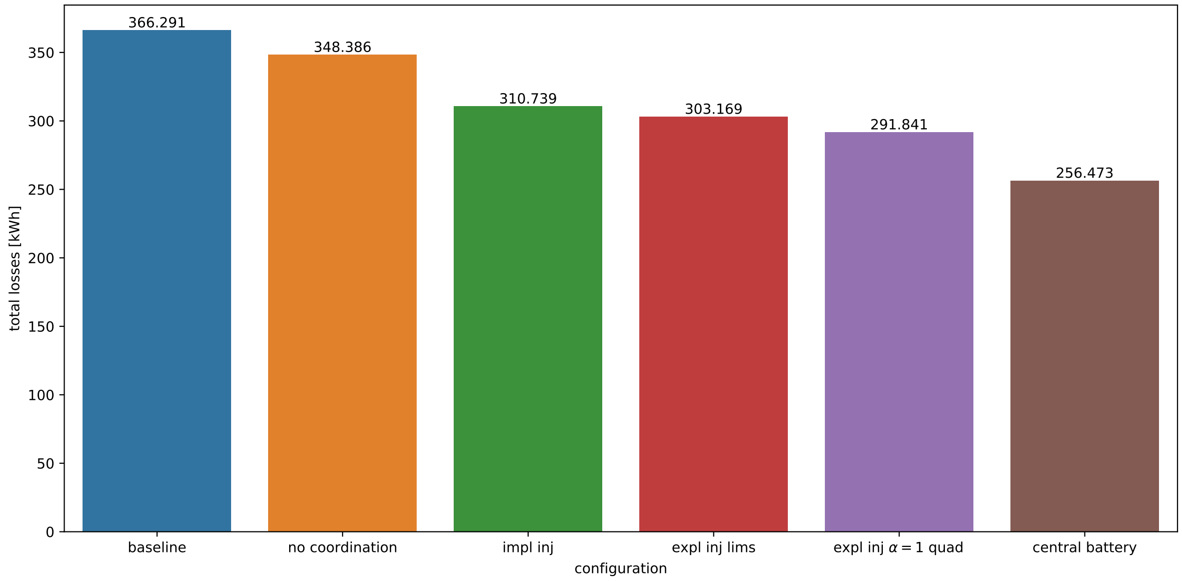


Figure 62: Losses for the 6 considered cases, computed as differences between the power at the PCC and the sum of the power at QP buses in the network.

8.4.2 Economic analysis

In evaluating the economic impact of the central battery, we focus on two different billing schemes:

1. The community administrator shares the battery with the community members. The battery meter is included in the calculation of the total injected energy E_p and total consumed energy E_c, used to calculate buying and selling prices (equation 8).
2. The administrator does not share the earnings offered by the battery with community members. The energy price is calculated without including the battery actions. In this case, the presence of the battery does not bring any benefit to the end-users, all the additional gains of the community go to the administrator.

These two cases are summarized in Table 16. One can see that in the shared battery case, the overall costs of the community members are reduced thanks to the presence of the battery. This is due to the fact that, when selling, the producers have an additional consumer inside the community to which they can sell the energy at a price higher than that of the grid, and when buying the consumers have an additional producer inside the community from which they can buy at a price lower than that of the grid. The interests of the community administrator are aligned with those of the members, and the way in which the battery is operated is beneficial to everyone. In fact, he too has an interest in maximizing self-consumption at the coupling point, as this reduces the bill he will have to pay to the utility.

Compared to the baseline scenario, total end-user costs in the shared battery scenario are reduced by 14%, while the administrator's earnings increase by 32%. In contrast, in the non-shared scenario, the administrator increases its earnings by 68%, while end-users do not gain from the presence of the battery. It is clear that this analysis does not include the investment costs required to purchase the battery. Buying a battery is not justified at current prices because it is not yet possible to recoup the investment by optimizing only the community's own consumption. In the continuation of the LIC project, it will be evaluated whether the DSO's introduction of a power tariff could make it attractive for the community to purchase a battery. It should then be possible for the administrator to take into account battery costs when calculating the internal tariffs.

	central battery shared	central battery non-shared	expl inj $\alpha = 1$ quad	expl inj lims	impl inj	no coord
increase in administrator earnings wrt baseline	32%	68%	28%	17%	17%	1%
decrease in costs for end-users wrt baseline	-14%	0%	-16%	-17%	-15%	-7%

Table 16: Difference in administrator earnings and end-user costs in different scenarios.

8.5 Multi-level control simulations

The algorithm to solve the multilevel coordination problem with grid constraints presented in 3.2 has been tested on several hierarchical configurations and distributions of smart agents represented by controllable batteries. Since the multi-level setting requires an higher amount of steps to converge with respect to the single level case, we initially only tested it without electrical simulation coupling, in order to test its correctness and characterize it in terms of computational time. Then, a detailed simulation including the load flow part in which two energy communities synchronize with each other to optimize self-consumption and peak shaving at a higher node was performed and is presented at the end of this section.

8.5.1 Simplified simulations of the multilevel control

An example of solution in terms of aggregated quantities and state of charge of controllable batteries for a 4 levels hierarchy is shown in figure 63. In order to test the algorithm also in the case of voltage violations, reasonable voltage sensitivity coefficients were assigned at random to the agents.

A trivial way of decomposing the problem would consist in reaching convergence in the lower level, then sending the results to the upper level of the hierarchy. In this way, the whole number of iteration before convergence in the lower level would represent a single iteration in the upper level. This solution concept would clearly result in an exponential computational time with respect to the number of considered levels, namely:

$$N_{tot} \sim tnN_i^L \prod_{l=1}^{L} N_b^l \sim N_i^L N_b^{\frac{L+L^2}{2}} \tag{54}$$

where t is the computational time for each agent for solving its local problem, n is the number of agents per level, N_i is the number of iterations before convergence for a single level, L is the number of levels, and N_b is the number of branches per level. Instead of following this strategy, we decompose the monolithic formulation of the problem to obtain nearly linear convergence, with respect to the number of levels. A study on iterations needed for convergence is shown in figure 65, for hierarchical structures with different number of levels and controlled agents. This was obtained creating random hierarchical structure with a maximum of 4 aggregation levels and of 20 agents per terminal aggregation nodes. The near-linear convergence with the number of levels is shown in figure 64, where the computational time before convergence from 500 simulations is shown in terms of boxplots for each considered number of levels.

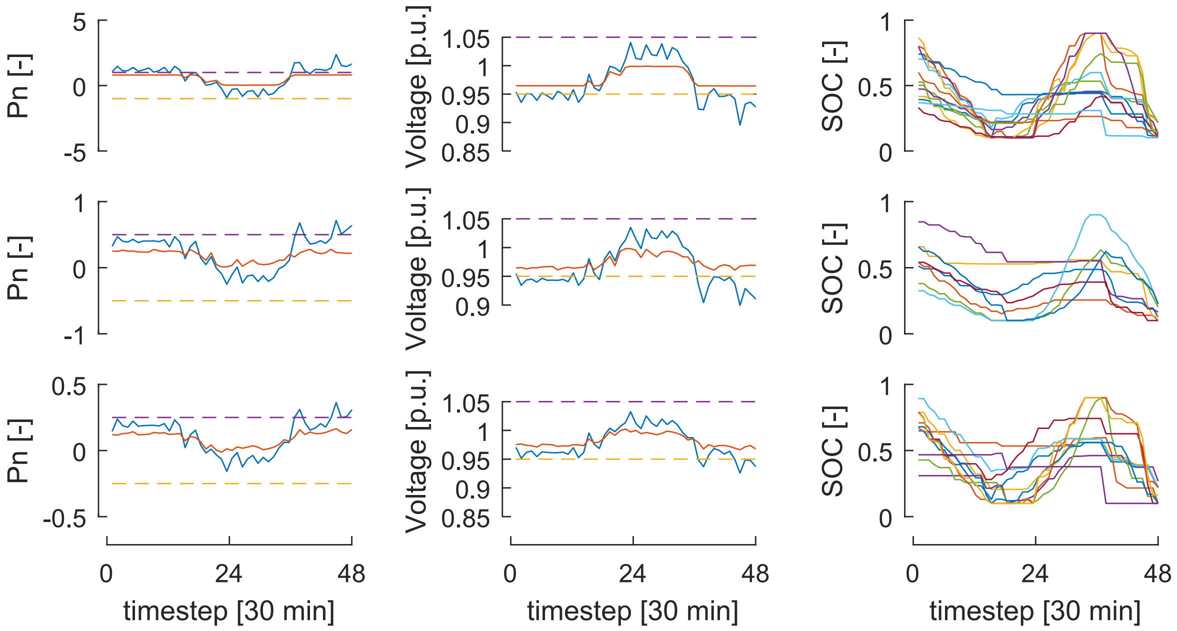


Figure 63: Example case. The considered hierarchical structure has 4 levels, and a single branching node in the first 3 levels. For the first two columns, the first row refers to the root node, second and third rows to the second and third level. First column represents the aggregated power profiles. Blue line: no battery actions. Red line: optimized power profiles. Dashed lines: power constraints. The second column represent voltage profiles. Blue line: no battery actions. Red line optimized voltage profiles. voltages and the state of charge (SOC) of the batteries are shown. The third column represents state of charge of prosumers' batteries in the second, third and fourth levels.

8.5.2 Detailed simulations of 2-level distributed control

Two energy communities were generated and connected to each other at medium voltage level. The first corresponds exactly to the LIC community simulated previously, the other is a similar version in which the house thermal characteristics and the uncontrolled loads profiles were changed to avoid perfect synchronization between the two communities, and the total amount of PV was increased by 20kW. Each of the two energy communities has four 15kWh / 7kW batteries, for a total capacity of 120kWh and power of 56kW. Each single community has power constraints at its local coupling point of $-40kW < P < 20kW$. The global coupling point has a power constraint of $-60kW < P < 30kW$. A schematic is detailed in figure 66.

Three scenarios were simulated:

1. A baseline scenario in which the batteries are idle
2. A single-level coordination scenario in which each community coordinates its batteries to achieve the objective of maximizing total self-consumption while respecting the power limits both at its local coupling point and at the global coupling point, without communicating with the other community.
3. A multi-level coordination scenario in which all batteries coordinate to maximize total self-consumption while respecting both local and global constraints, using the control algorithm presented in section 3.2.

These three scenarios allow us to test the hypothesis, introduced in section 3.2 and depicted in figure 7, that allowing communication between larger groups of controllable devices can lead to better solu-

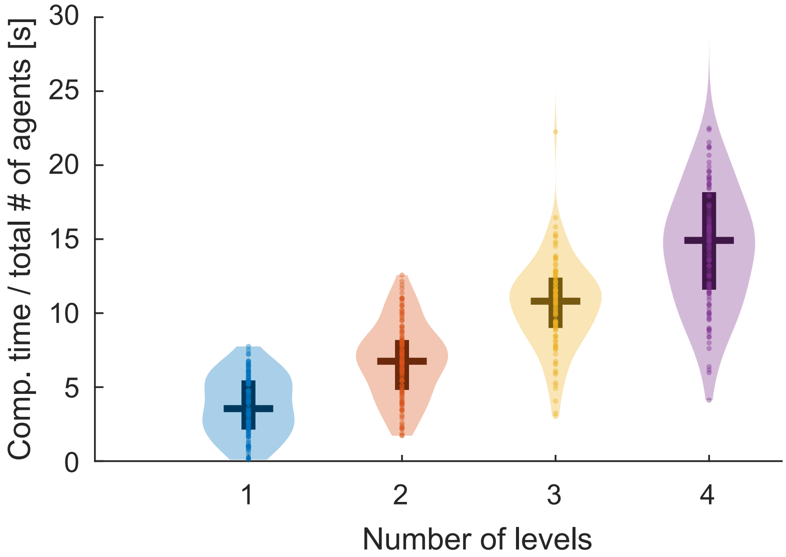


Figure 64: Estimated pdfs of the computational time divided by the total number of agents, as a function of the number of levels in the hierarchy. The vertical bar is the interquartile range, the horizontal line is the median.

tions in terms of objective function and constraint violations, enlarging the set of feasible actions for the agents. To further investigate this hypothesis, we compare the results of the coordination schemes with and without perfect forecasts. Figure 67 shows the time series for the power at the coupling point of the community (upper panel) and the total power of the distributed batteries (lower panel). We can see how all the considered strategies are able to flatten the power profile at the point of common coupling, respecting the power constraints. While a lower variance can be appreciated in the green profile, representing the multi-level coordination with perfect forecasts, it is hard to directly compare the performance of the other methods in terms of time series. An easier comparison can be done looking at figure 68, which shows the pdf of the power at the PCC for the different methods. The single level coordination and the multi-level one are very similar in terms of power pdf. On the other hand, the multi-level formulation is clearly more effective in shrinking the power pdf towards zero, compared to the other methods (even the single level with perfect forecasts). We can conclude that the quality of the forecast used for the control play a determinant role in the effectiveness of the multi-level formulation over the simpler single level coordination .

In the first panel of figure 69 the same behaviour can be better seen for the extreme quantiles of the power distribution of the point of common coupling: when the power at the coupling point is in its 1% quantile, the batteries controlled using the multilevel strategy with perfect forecasts are the best in absorbing power (shifting the power distribution to higher values and with a significant lower variance than all the other cases). The same behaviour can be observed at highest values of the PCC power distribution: the multilevel controller with perfect forecasts is the best at shrinking the pdf towards zero. On the the hand, always considering the tails of the PCC's pdf, the boxplots referring to the multilevel and to the single level cases, when not using perfect forecasts, are very similar. This confirms that the forecast's quality is determinant to exploit the control advantage that the multilevel formulation can provide.

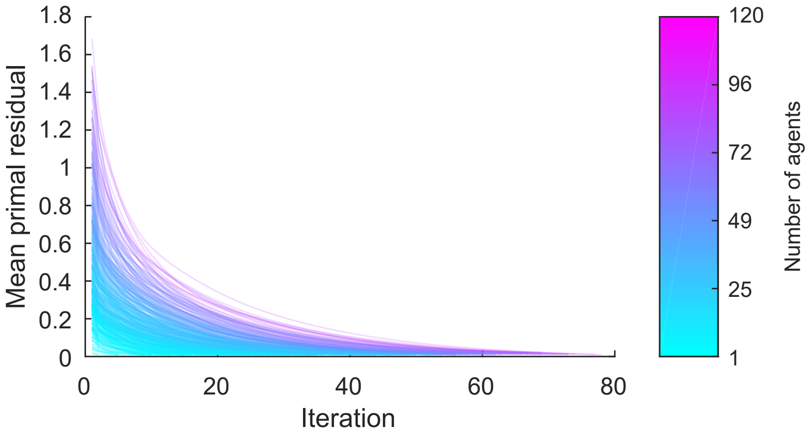


Figure 65: Mean overall primal residual for all the simulations, plotted versus the number of iterations before convergence, colored against the number of agents.

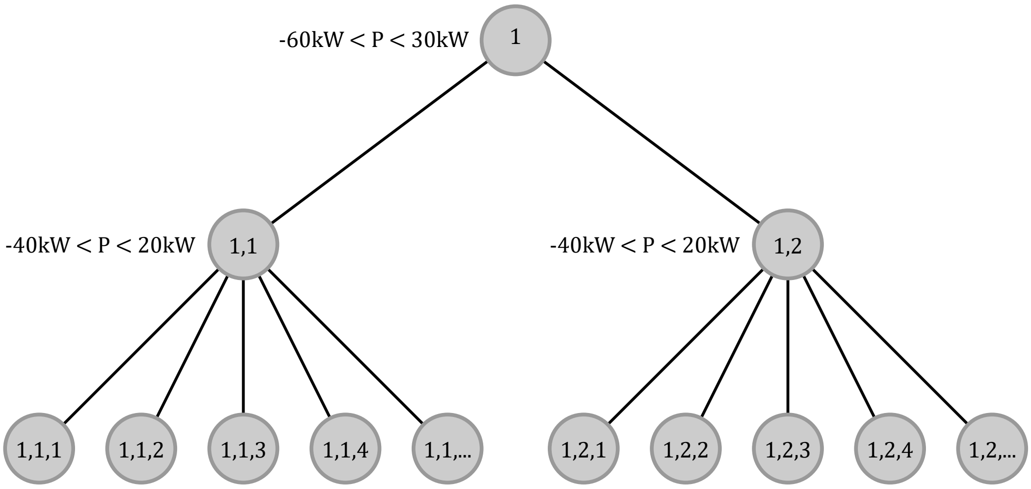


Figure 66: Hierarchical structure of detailed multi-level simulation. Top level: global coupling point, Middle level: local coupling points of the single communities. Bottom level: smart meters.

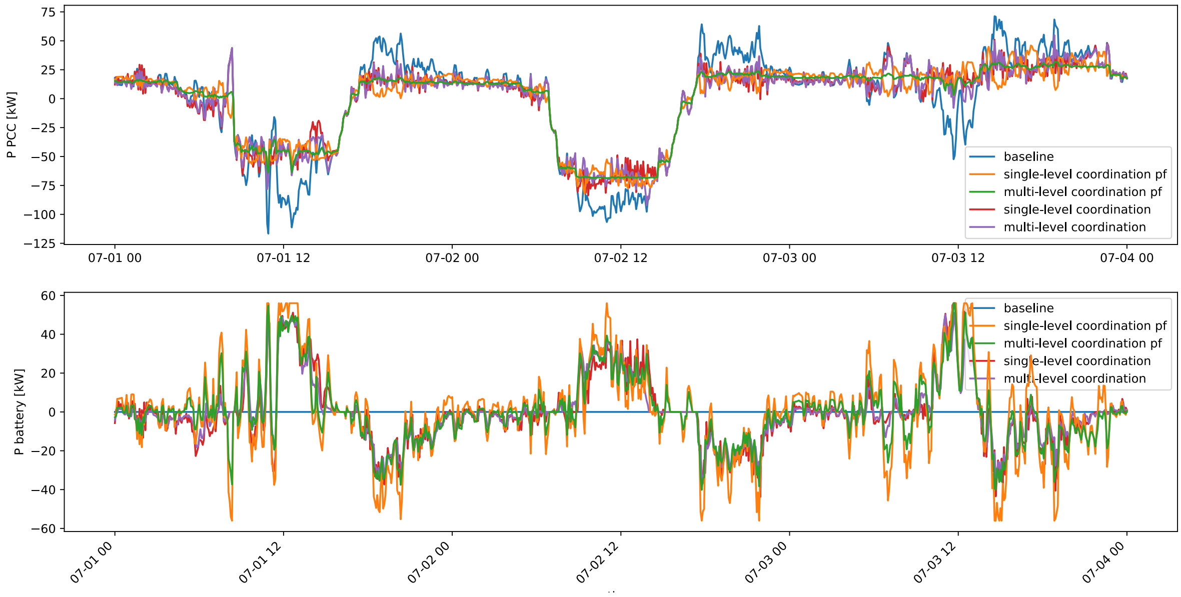


Figure 67: Time-series of the active power at the global coupling point of the community and the total power of the distributed batteries for the first 3 days of July. Perfect forecasts are denoted by *pf*

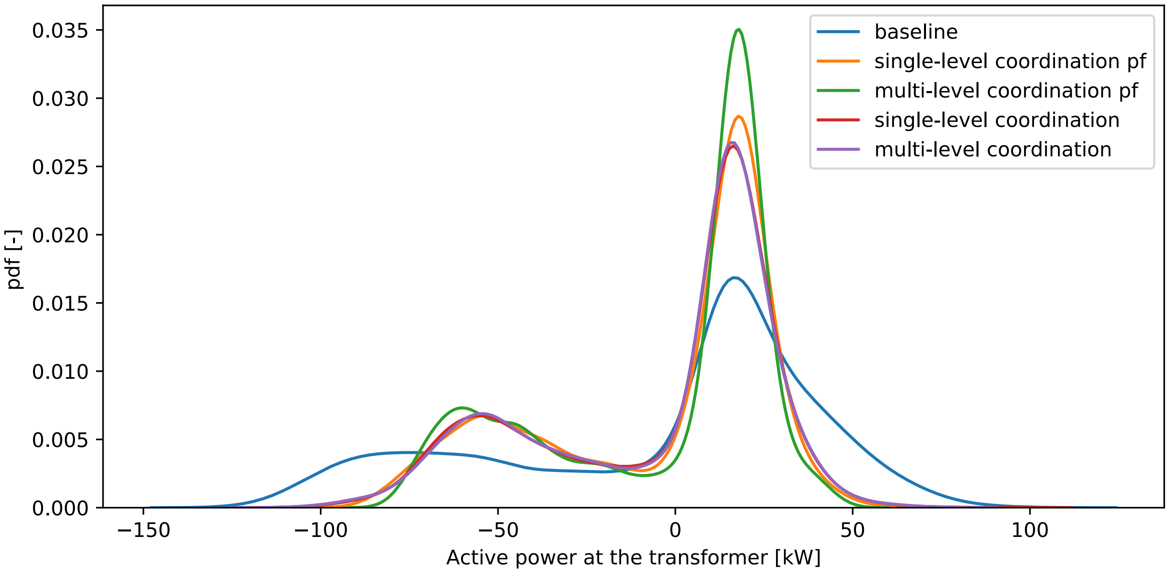


Figure 68: Probability density function (PDF) of the power at the global coupling point. Perfect forecasts are denoted by *pf*

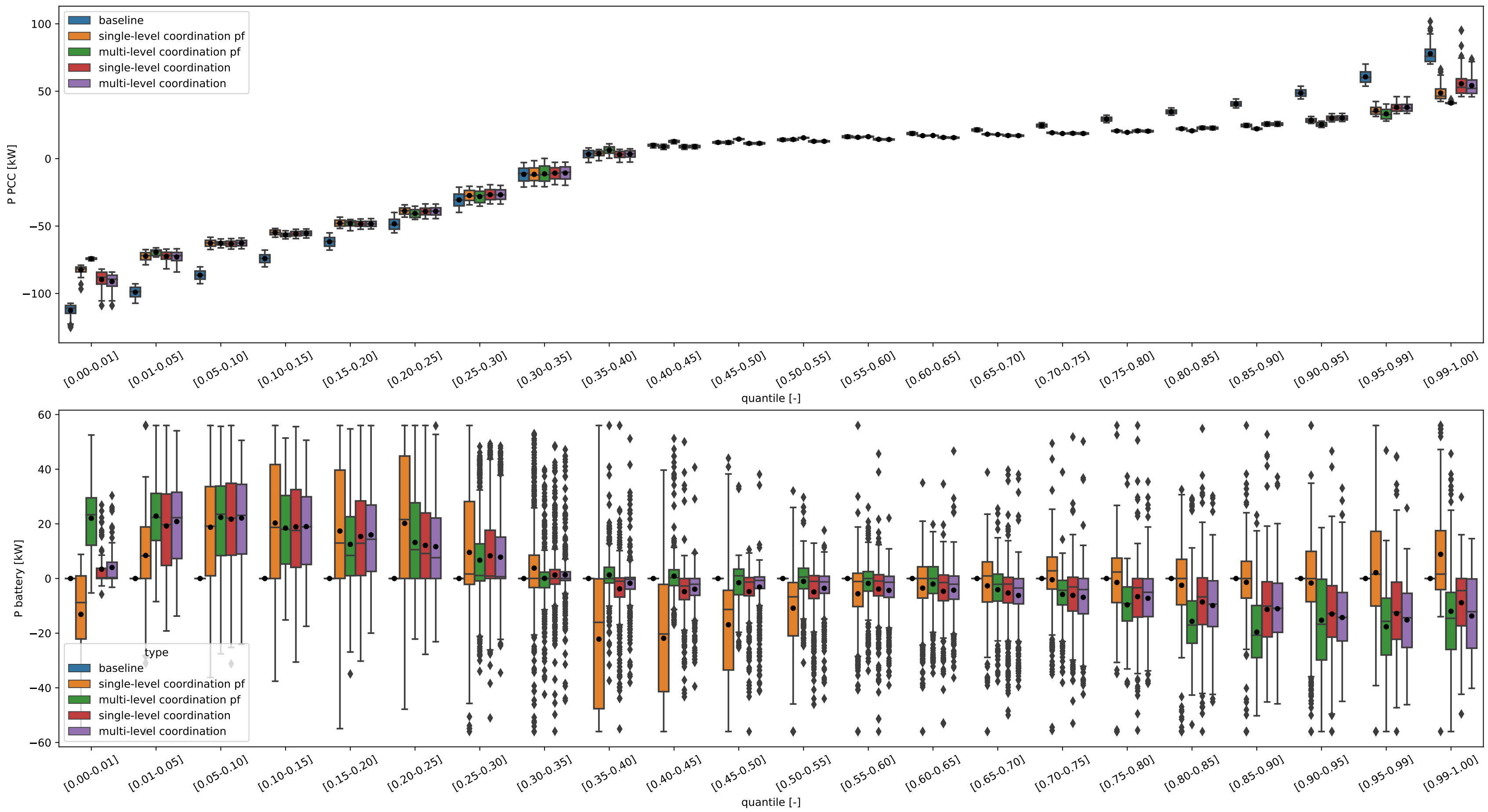


Figure 69: Effect of batteries control on the power at the point of common coupling (PCC) of the community with multi-level control. Top: boxplots of the power at the PCC per inter-quantile range. Bottom: boxplot of the total power of the batteries per inter-quantile range of the power at the PCC.

8.6 LIC Pilot

8.6.1 Tariff design analysis

Energy tariff With a year's worth of data available from the LIC community, we tested the effect of the new local tariffs on the community members. The tariff scheme in LIC for the year 2020 is shown in Table 17, and follows the automated market making mechanism presented in section 2.2.2.

	buy	sell
internal prices (cts/kWh)	16	9
utility prices (cts/kWh)	21	6

Table 17: LIC prices for the year 2020.

We evaluated the effect of systematically varying the buying the selling prices on the end-users using the prosumption profiles of the year 2020. The relative savings with respect to BAU are shown in figure 73. In general, one can notice that, given the current price scheme, the prosumers are those profiting the most from being part of the community, which could be justified by the fact that they are actually those who invested in the PV plants. Figure 71 shows the revenue split between the community members and the community administrator, defined as the percentage of the difference between the total costs to the members in the BAU case and what the administrator pays to the DSO going to the administrator. This estimate already takes into account measurement costs for the administrator. In contrast, administrative costs and battery depreciation would still need to be subtracted from the administrator's revenues to calculate any actual profits.

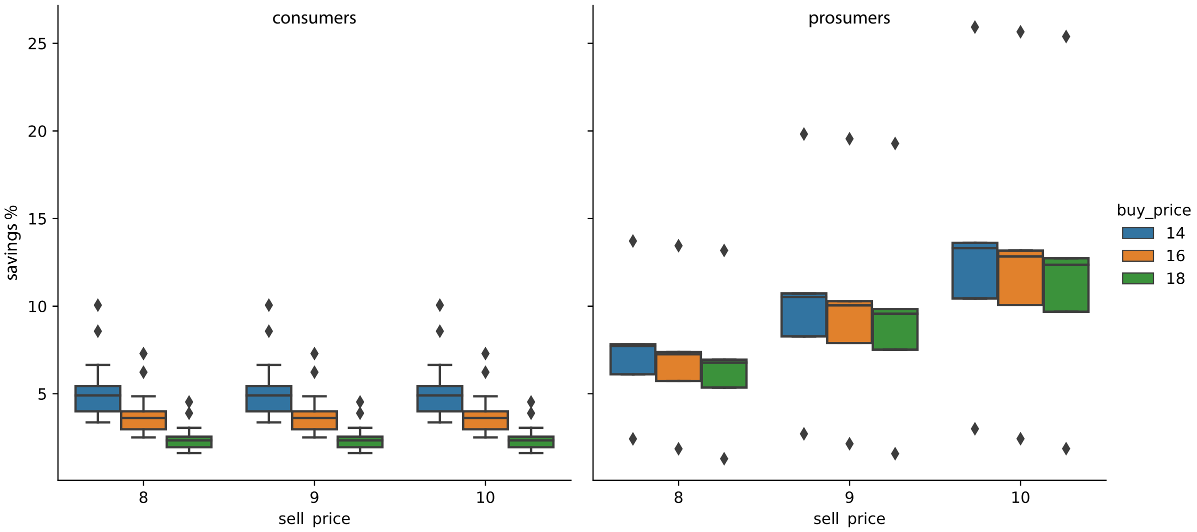


Figure 70: Relative savings of the community members w.r.t. a business as usual scenario, as a function of the buying and selling prices and user type.

It can be seen that in this case the community manager has been particularly generous to the users.

Power tariff Next, we investigated the effect of adding a power component of the tariff at the community coupling point as proposed in section 2.2.3 and passing it on to the end-users according to

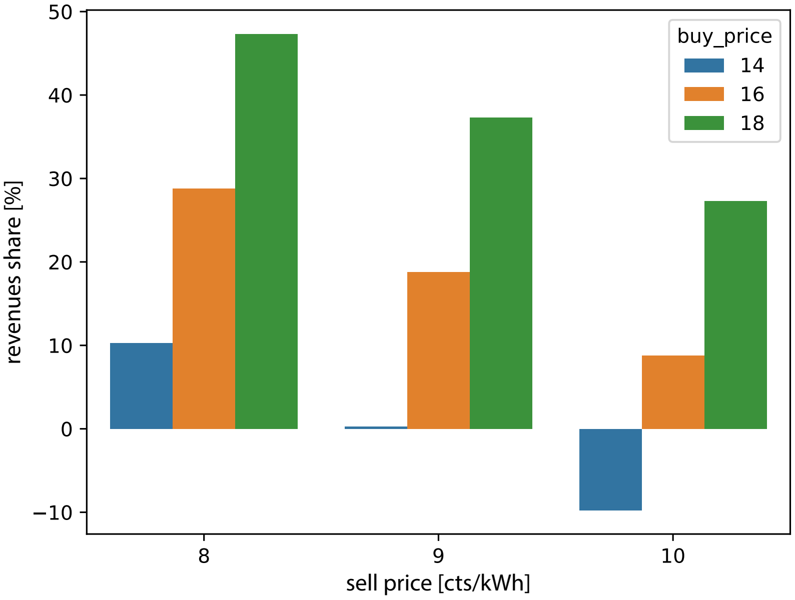


Figure 71: Revenues split between community administrator and members.

their relative (positive or negative) contribution to the total power. Figure 72 shows the effect of the β [cts/kWh^2] parameter on the relative yearly savings of the community members, grouped by prosumers and consumers. As explained in the previous sections, the LFM pricing scheme does not guarantee a cost reduction for end-users, as in the case of the LEM prices scheme. On the contrary, the LFM mechanism applies a bonus-malus scheme depending on the current contribution of the end users in shrinking the overall power profile to zero.

The administrator revenues instead increase with the increase of β. This is merely because the administrator is operating the battery, and the battery is always contributing to the reduction of peaks. Fig 73 shows the relative increase in revenue of the administrator with respect to the case in which $\beta = 0$.

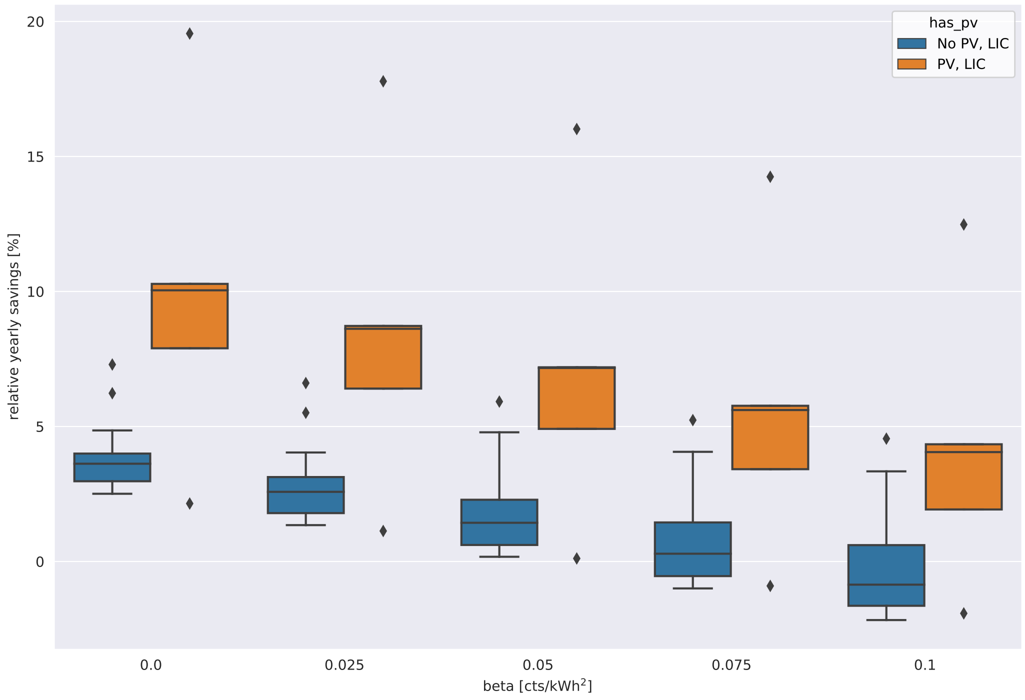


Figure 72: Effect of different values for the LFM parameter β on the final yearly cost for the end-users. Boxplots contain differences w.r.t. BAU.

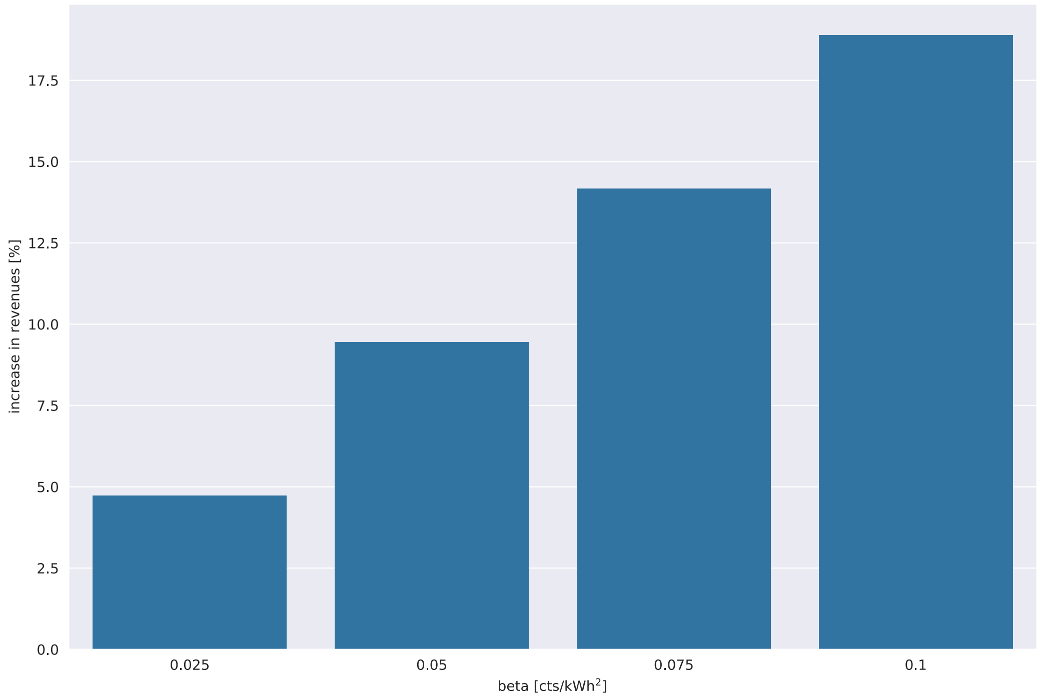


Figure 73: Relative increase in revenues of the administrator w.r.t. the case in which no power tariffs are applied (i.e. $\beta = 0$).

8.6.2 Preliminary analysis of battery effect

In this section, we will draw up a preliminary assessment of the effect of the presence of the battery on the technical and financial performance of the community. It is important to note that the battery was not consistently operated in this first year of the community's life and that the control algorithm has been modified in successive iterations. Nevertheless, we find it interesting to present this preliminary result.

First, an analysis can be made of the increase in self-consumption and autarky in the community. It is very important here to point out that we are not evaluating the total self-consumption of the energy produced by PV within the community. The energy self-consumed by the prosumers within their homes is not taken into account. In fact, the total self-consumption cannot be computed from the measured data because PV systems are not separately metered. So, in this case, with community self-consumption and community autarky, we refer to the excess PV energy that leaves the houses and is consumed within the community. Table 18 shows the increase in community self-consumption and autarky achieved thanks to the battery. We can see that self-consumption is increase to a very high value of 93.7%.

	without battery	with battery
community self-consumption	72.6 %	93.7 %
community autarchy	8.0 %	12.3 %

Table 18: Self-consumption and autarky

Figure 74 shows an example of a consumer in the community and the effect the presence of the battery has on the origin of the energy consumed and costs. Figure 75 shows the same analysis for an example of a prosumer. In addition to the effect on consumed energy, for prosumers one can also see the effect on produced energy. In general, for the end users, the presence of the battery leads to an increase of 22% in the savings with respect to the case in which the battery is not present, bringing the total savings with respect to the scenario in which they are not part of the community from 3.5% to 4.2%.

In 2020, the battery was operated to maximize self-consumption without having the ability to charge from the grid and priority in excess production is given to end-users. This means that it was not possible to limit the community's peak consumption by using the battery in the winter, as there was no surplus production from which to charge the battery. In the course of 2021, algorithms that give priority to the battery and allow it to charge from the grid to minimize the peak will be tested. This is financially justified as the community pays for the peak power to the DSO.

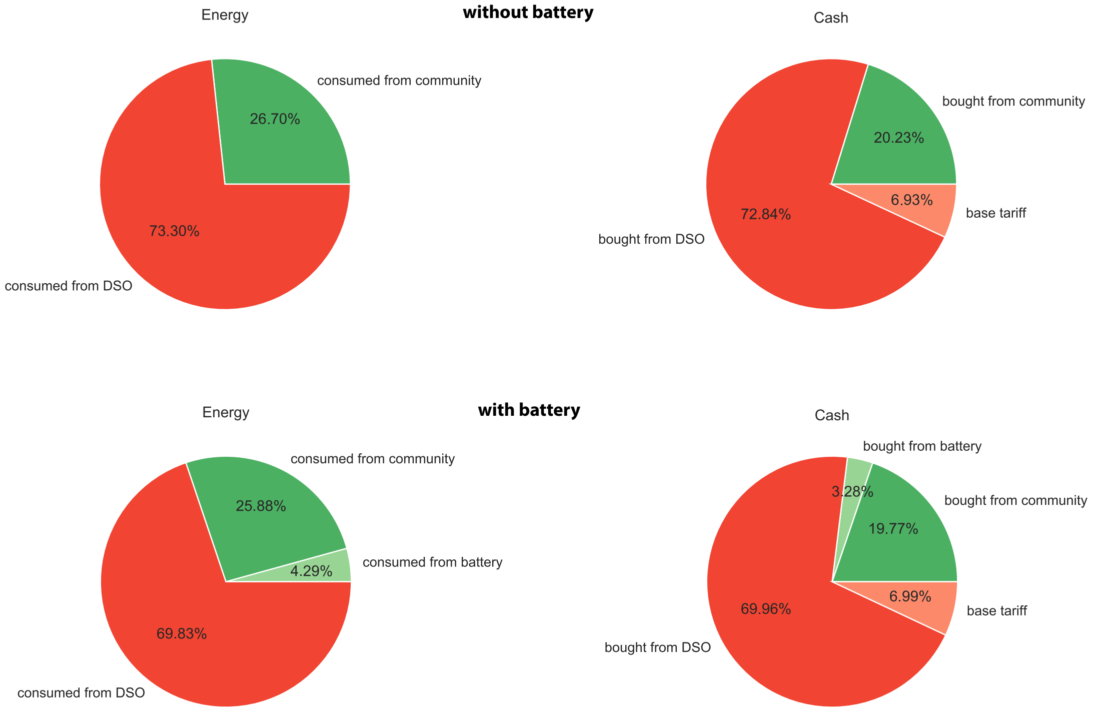


Figure 74: Contribution of battery for an example consumer.

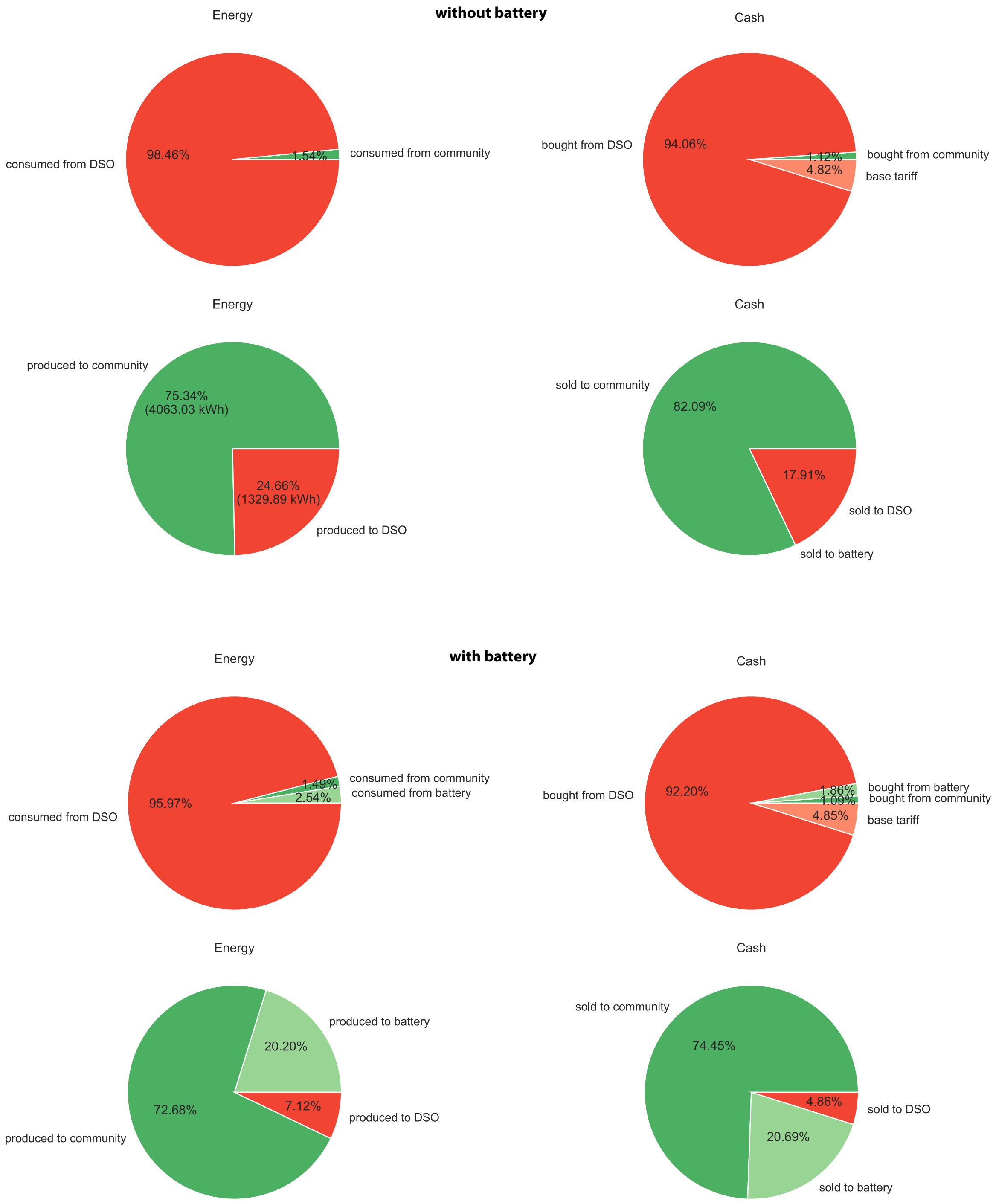


Figure 75: Contribution of battery for an example prosumer.

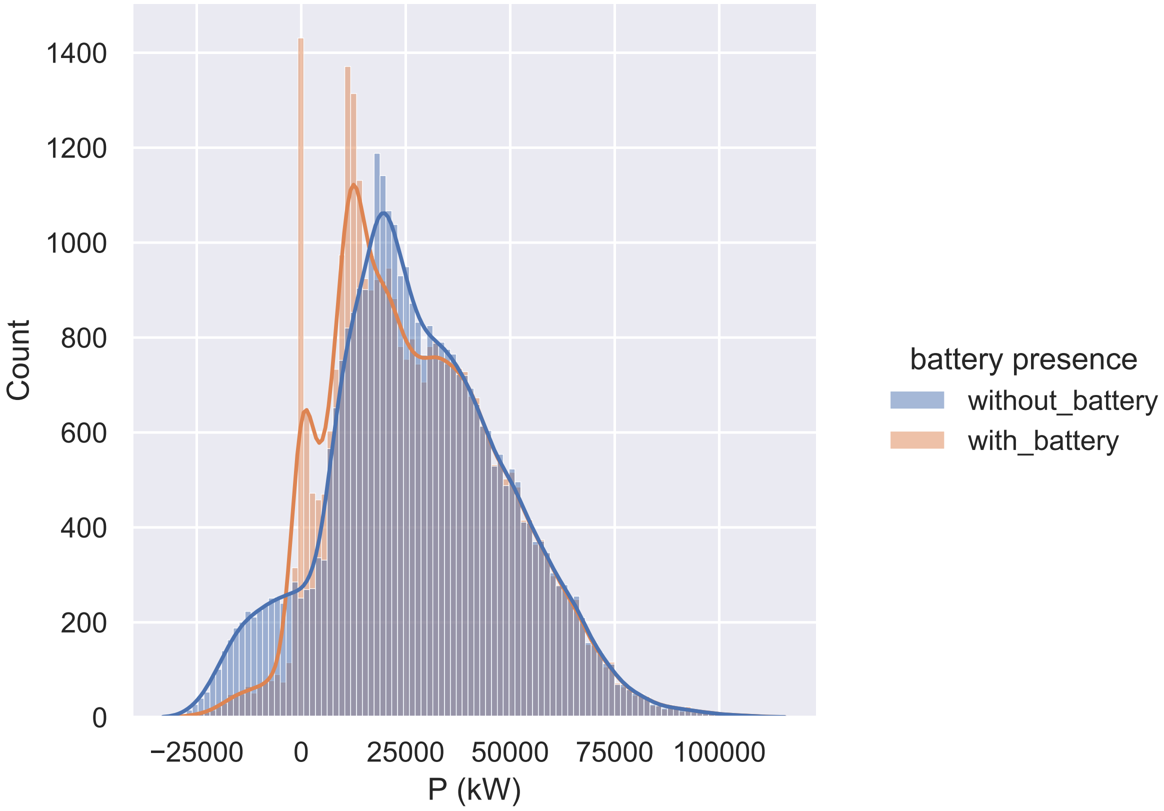


Figure 76: Distribution of active power at the virtual coupling point with and without battery.

9 Lessons learned

Research projects, in particular pilot projects, are a valuable learning opportunity. In this section we will highlight some lesson we learned during the projects execution.

9.1 Smart metering infrastructure bandwidth and latency

During the OptiFlex experimentation we pushed at the maximum limits the industrial smart meter infrastructure. The use of powerline connections introduces “best-effort” aspects due to the nature of the physical communication infrastructure. In theory the metering infrastructure can go from the standard 15-minutes acquisition interval to a 1-minute interval, but in order to achieve this, a higher density of data concentrator is needed. Even after the installation of this additional hardware and some smart meter firmware update it was still quite difficult to reach a consistent performance on all meters, because the communication distance and external noise disturbances can vary a lot. In addition to that the collected data will reach the DSO’s server with one hour delay typically. Therefore a demand-side management control strategy working on top of a powerline smart metering infrastructure has to be designed considering these infrastructure constraints.

9.2 Limited metadata about behind-the-meter flexibilities

DSOs keeps metadata about behind-the-meter flexibilities in their Enterprise Resource Planning (ERP) system. The metadata however is not a perfect picture of the real situation on the field. The process to fill the metadata rely historically on paper forms filled by installers, and is quite error prone. An automatic system controlling remotely these flexibilities has to consider these deviations from the ground truth.

9.3 Limited digitalisation of grid topology data

In the last years the DSOs started the process to digitalise the low voltage grid topology. The availability of a digital grid topology is a helpful tool to improve the actuation of demand side management to help the local grid stability. However the process is not yet finished, usually paper drawings and diagrams are converted and imported in CAD/GIS systems, with a strong focus on the civil engineering aspect. The priority for a DSO is to be able to locate the physical location of underground pipes. What was missing in our projects was a digital electrical layout, with all the relevant electrical properties of cables and component. Some manual work was needed to reconstruct this layout, combining the Autocad/GIS data with components data in the ERP.

9.4 Disaggregation challenges

The LIC’s controllable loads are not directly monitored, as is usually done in demand side management projects. This poses challenges in the control of these devices, as planning the control of the devices requires to forecast their energy needs, that is, to know their disaggregated power profiles; on the other hand, not needing to install dedicated sensors for each controlled appliance, significantly lowers the technology cost of the DSM solution, as this only relies on an already installed sensor: the smart meter.

Many non-intrusive load monitoring algorithms (NILM) have been proposed in the literature; however, many of them cannot be applied out of the box in this case, as we are considering the following setting:

- unsupervised learning: we do not possess the ground truth for the output of the disaggregation. We are in the most general case in which we do not know: number of appliances, types of appliances, their nominal power, nor their characteristic power profiles.

- we are working with 10s sampling time, but we would like to use an algorithm able to disaggregate with lower sampling times (down to 10 minutes), as smart meters usually send data with a granularity of minutes.
- the algorithm must be able to run on the Strato's ARM architecture, which is also used to solve the optimal control problem: both training and test must run in less then one minute.

We have tested different algorithms satisfying these requirements, namely an approach based on clustering derivatives of the active and reactive power profiles and an approach based on the identification of appliances' signatures. While these approaches works well for certain profiles, we have encountered several inaccuracies during the monitoring period of the LIC pilot. Furthermore, after the disaggregation process, the algorithms must be able to link a given disaggregated load to the operations of a relay. In each household two relays are installed, which can be connected or not to a controllable load. In order to associate the relay to a disaggregated load, a perturb and observe strategy must be adopted: activations of the disaggregated load must be linked to the operations of a given relay looking at correlations in the two signals.

These problems have been investigated and can be summarized in the following:

- Correlated loads. Only a load is associated with a given relay, since one relay controls at most one load. However, it can happen that a controlled load is highly correlated with another, uncontrolled load; for example, during winter, the auxiliary internal resistance of the heat pumps can highly correlate with the activation of the heat pump's compressor, resulting in a "virtual load" with a higher observed power (e.g. a 6 kW heat pump with an auxiliary internal resistance of 3 kW can result in observing a 9 kW load during winter time). This will result in a poor disaggregation during spring and summer time, when the auxiliary resistance is not turned on. To partially overcome this issue, prior knowledge on the range of active and reactive power of the controlled loads in a given household can be used, to help the algorithm associating a given disaggregated profile to a relay.
- Power offset. Some households present power offsets: for prolonged period of times (several weeks) a constant load is turned on. This situation affects the estimation of the nominal power of the disaggregated loads and results in bad performances when the offset disappears.
- Non-constant signatures with high activation variance. Some monitored loads, especially heat pumps, present a signature with a non-constant shape; in fact, the power profile of the heat pump is influenced by the thermal inertia of the heating system and by the external temperature, throught the coefficient of performance. This lead to signatures with complex shapes, changing mean power and different activation duration. These facts influence the final disaggregation results, which assumes constant nominal power for the disaggregated loads.

The described issues pose a great challanges to the controllability of the flexible loads. First of all there is a risk of mis-association between a disaggregated load and a relay; secondly, wrongly predicting the nominal power of a controllable load affect the controller performance, both leading it to incorrect solutions and affecting the forecasts of the next activation period. For these reason the NILM problem under these settings must be further ivestigated.

9.5 Distributed algorithm convergence

In our distributed implementations, problem (1) is decomposed using Lagrangian duality theory (ADMM) or the preconditioned forward backward algorithm. In both cases, the following undesirable situations can verify:

- Existence of a feasible point. For these algorithms, and for other convex optimization algorithms, convergence is guaranteed only if a feasible point exists. This is not always true, e.g. if we place a too stringent power limit to the EC's transformer which cannot be satisfied only by steering the flexible devices or the installed electric batteries. In this case, we can takle the problem in two ways. The first way is to consider a virtual "slack battery", with unlimited capacity, a peak shaving objective and an operational cost much higher than the other devices. Such a battery has the scope of guarantee the existence of a feasible point; having a much higher operational cost, the slack battery is only activated if the flexible devices are not able to find a feasible solution to the problem. To obtain the real solution, at the end of the convergence the actions of the slack battery must be subtracted by the overall power profile. The second way is to dump the Lagrangian multipliers λ taking into account the coupling constraints: after a given number of iterations, the Lagrangian multipliers growth rate is constantly reduced, leading to the convergence to an infeasible point (that is, not respecting the imposed grid constraints; this can be seen as a point of "best effort").

- Oscillations. The ADMM and pFB formulations are taught to decompose problems in which the system-level cost is influenced by continuous variables. In the case in which the controlled devices can't modulate their power, e.g. electric boilers, the local problem becomes a mixed-integer linear or quadratic problem. This could lead several devices to switch decisions on their scheduling at the same iteration, especially when the number of controlled devices is low. This could lead to oscillations in the objective function and ultimately prevent the convergence of the algorithms. To avoid this, we have added a parameter dampening the change of decisions from previous iterations.

9.6 Importance of forecasting algorithms

As seen in section 8.5.2, the accuracy of users' forecasts is a key factor in determining the effectiveness of different control approaches. As a preliminar step to the simulations presented in this report, we have compared different forecasting algorithms to understand their impact on the closed-loop economic performances of different control schemes. For these analysis we have always used the LIC model, and considered four controllable batteries. We have compared a temperature-detrended Holt-Winters (HW) model with double seasonality and different flavours of forecasters based on boosted trees. Results show how the first step ahead accuracy has a higher impact on the economic results under both the implicit and explicit coordination and remuneration schemes. This can be seen comparing figure 77, where the open-loop accuracy for different forecasters is shown, with figure 78, where the closed-loop economic performance for the explicit case are plotted for all the users, green dots representing users with a controllable battery. Despite all the forecasters different from the HW models are much better at forecasting steps beyond the first one, we can see a comparable closed-loop economic performance for all the cases. On the other hand, when we relaxed the constraint for which the batteries can only charge or dis-charge from and into the households' electric main, accuracy beyond first step-ahead became more relevant. This relaxation increases the feasible space of the controller, allowing the batteries to perform arbitrage, which optimal planning requires accurate medium-term (daily) forecasts. Figure 79 shows that two households with a battery worsen their performances when using HW forecaster, w.r.t. the case in which the battery can't inject or charge in and from the grid. This can be due to the low accuracy of the HW forecaster for the higher steps-ahead. This hypothesis is strengthened by the fact that the LGB hybrid forecaster, which has a higher accuracy w.r.t. the HW model for higher steps-ahead, shows a consistent decrease of costs for the end-users with a battery.

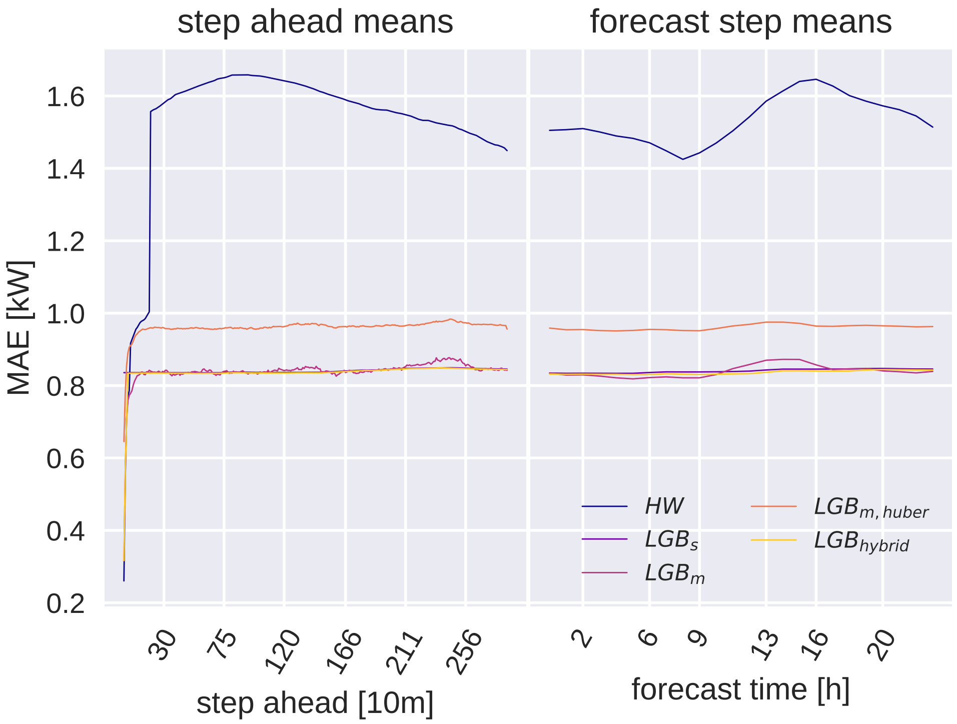


Figure 77: Mean results in terms of MAE for the households with controlled batteries. Left: results mediated on the step ahead. Right: results mediated on the time of the day on which the forecast was performed.

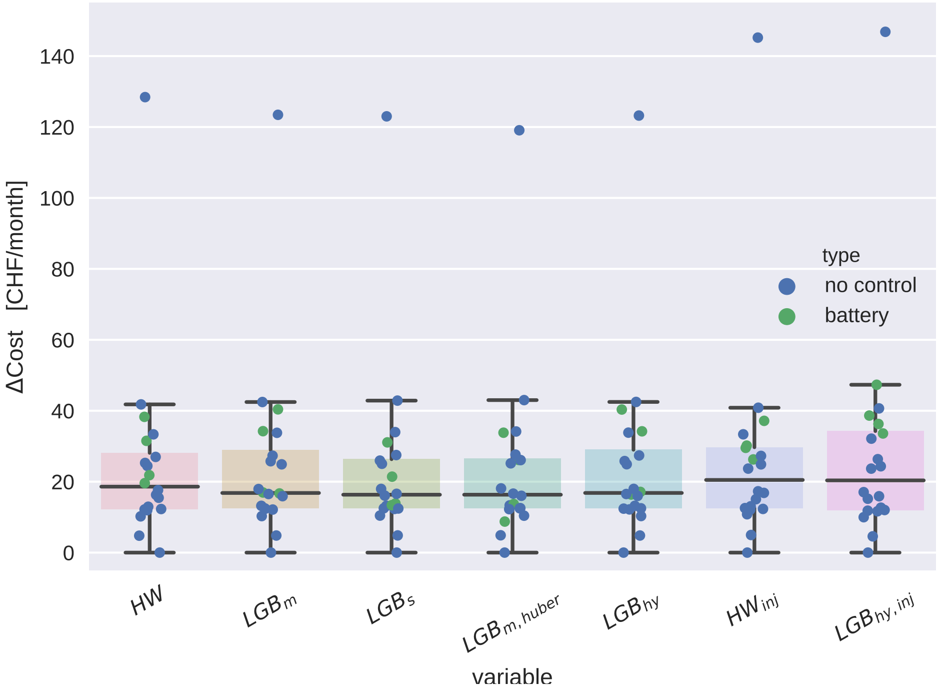


Figure 78: Difference in monthly costs w.r.t. the BaU with no allowed injections, with explicit coordination, for different forecasters.

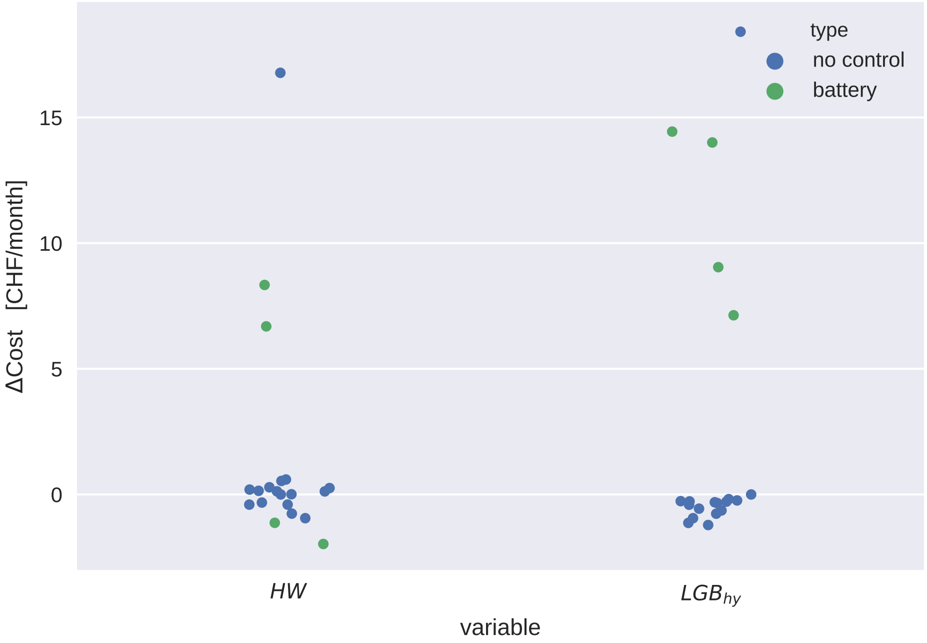


Figure 79: Difference in monthly costs w.r.t. the BaU with no allowed injections, with explicit coordination.

10 Conclusions and outlook

As distributed generation increases, the use of flexibility will become increasingly important in the future. Local flexibility has the potential to foster the integration of DERs, as local self-consumption decreases the burden on the distribution grid brought by them. There is, however, an essential set of points to consider for local flexibility control to have the desired effect, which we will discuss in the following.

The stage needs to be set to allow local flexibility markets to thrive For owners of behind-the-meter flexibility to be willing to offer it externally, there needs to be a business model that remunerates them for it. When focusing on the problems caused by renewables on local DSO grids, one of the tools with the highest potential are local flexibility markets. In LIC a self-consumption community was established to exchange energy internally in a local flexibility market, in a legally complying way. In Switzerland, to form a self-consumption community, one must be physically connected to other community members. In Lugaggia, part of the public network was unbundled for this purpose. In our opinion, this kind of solution is suboptimal because it is very complicated, and not always the DSO is willing to sell or rent part of his network to the community. This means that generally, self-consumption communities in Switzerland are limited to single buildings or newly built neighbourhoods. In other countries, the rules by which self-consumption communities can be formed are (or will be) less strict. For example, in Italy, a virtual community approach has been chosen [64]. There will be no need for a direct private physical connection. It will be enough to be under the same secondary substation. In our opinion, the virtual community model is much more streamlined and allows for faster growth in the number of communities. In the case in which communities do not limit themselves to single buildings or new neighbourhoods, another problem we see is the great difficulty accessing data that can be used to assess the potential of establishing a self-consumption community. In the various pilot projects we conducted in the past, we learned that most DSOs do not have a direct way to know which secondary cabin a meter is under. Also, the license plate data of the flexibilities are often incorrect or missing, and in any case, each DSO has its own way of saving and retrieving them. This represents an obstacle that can be overcome by pushing towards greater digitization and standardization. In this context, the project of a centralized data hub at the Swiss level promoted by the SFOE is very interesting [65]. A central data hub would be an ideal tool coupled with a virtual community model. Indeed, through the creation of a "flexibility registry" in the Datahub, recording all flexible installations in CH is envisioned. This registry could then be used to initiate matchmaking between flexibility providers and flexibility users.

Local energy markets need clear market rules and secure and efficient platforms on which to run Local markets are characterized by a small number of players and high uncertainty of the production and consumption costs. For these reasons, a classic auction-based market mechanism is not optimal. In our research work, we devised a possible market design that guarantees that all the market players are incentivized to play fairly by designing the market rules as those of a cooperative game, in which a generalized Nash equilibrium is achieved. In such market design, players have no incentive to lie because this would be to their detriment and is therefore also applicable in scenarios in which players do not forcibly trust each other or when there are several technology providers for the actuation of the flexibility. In section 8.2, we show how, with these rules of the game, the total welfare of the community is not guaranteed to be maximized, because to do so, some players would have to sacrifice themselves for the common good. There is thus a tradeoff between common welfare and the welfare of individuals.. In addition to proper market design, one needs a marketplace that users can trust, and that is transparent and auditable. In our case, we focused on blockchain technology as a way to implement such a marketplace. Blockchain is a promising enabling technology for the digitalisation of energy. In Switzerland the interest in this technology is shared by 30 stakeholders from the industry and research sectors, who launched the DLT-for-Power standardisation initiative, supported by the Schweizerische Normen-Vereinigung (SNV)[28]. However the peer-to-peer blockchain-enabled local market is not the easiest use case to tackle, blockchain value proposition is suited in use cases where multiple stakeholders wants

[28] https://www.snv.ch/en/about-us/news-portal/news-details/new-ways-in-the-electricity-market-using-block-chain-and-co.html

to interact in a trustless matter. The P2P local market use case has a limited number of stakeholders and interacts directly with a highly trusted physical layer. Smart meters are standardised and certified devices, installed by accredited technicians following a clear regulatory framework. Moreover current smart meter technology has not yet the bandwidth and the computation power necessary to run even a lightweight blockchain client. It's highly likely that the current limitations will be reduced in the future by the continuing increase of edge computing capabilities and the interconnection of use cases, ranging from services for the local grid management, to services to crowd balancing platform like Equigy[29]. Therefore is crucial to continue the experimentation in pilot projects, including also the business innovation thanks to the foreseen regulatory sandboxes.

Innovative market design is needed for the local energy market to truly support the grid The Smart Grid Switzerland Association (VSGS) recently issued a whitepaper on the quantitative evaluation of the redistribution effects of the decrease in grid (energy) usage due to self-consumption [66]. The whitepaper states that consumption peaks are decisive for the design of the distribution network and for the costs of the network, and since such peaks usually happen in winter evenings during which PV is not producing, SCCs are not helping to lower the grid costs. Additionally. since for the self-consumed energy, the users do not pay for grid costs, and overall grid costs for the non-community members will increase up to 37.60 CHF/household/year by 2050. On the other hand, forming self-consumption communities makes PV much more profitable for those who build it, which is central for the energy strategy 2050. It is also true that the sizing of the PV, in this case, is made according to the amount self-consumption of the community, which indirectly helps to optimize the grid locally, which is something that with other incentive models would be more challenging to achieve. In our research and pilot projects, we have shown that demand-side management has the potential to increase the profitability of self-consumption communities. Still, it also can help control the grid when leveraged with suitable algorithms. So, in our opinion, energy communities do also have the potential to reduce grid costs if their flexibility is properly actuated. This becomes especially true when local storage is present and, in a slightly further future, when vehicle- to-grid will be available.

However, there is a need for a way to incentivize grid friendly behaviour by communities. That way comes through appropriate market design, for example markets in which communities can offer their flexibility to the DSO. But a much simpler solution, though one with less potential, is the design of tariffs that take into account the effect on the network. The most simple grid tariff is the already in use power tariff, in which the users pay for the monthly maximum consumption point. We proposed here an alternative model, which continuously optimizes the total power of the community, instead of invoicing the monthly maximum. Our reasons for choosing this alternative model are as follows. Losses reduction is more significant with our model. The optimization problem is easier to solve, leading to a lower cost for its eventual use in production. With the current power tariff model, if by mistake the community reaches the maximum peak at the beginning of the month, it has no more incentive to limit its consumption in the following days. At the same time, at the DSO level this would still make sense, both because the community peak does not necessarily correspond with the DSO peak and because DSO's losses would be reduced. Section 8.2 also shows how, to shave consumption and production peaks, imposing a quadratic punishment at the coupling point is slightly more effective than imposing explicit constraints on the power at the coupling point because the control algorithm is less sensitive to the errors made by forecasting algorithms.

How to deal with uncertainty In our projects, we have shown how flexibility can be exploited to maximize the potential of energy communities both in terms of their profitability and the service they can offer to the grid. To this end, we have proposed several versions of control algorithms (distributed and non-distributed) based on MPC logic and have demonstrated their effectiveness in simulation. We find it useful here to point out the limitations of the algorithms we have presented. First of all, as discussed in 9.6, we noted how the quality of the forecasts has an important influence on the results. Forecasting consumption and production at such a small and localized level is a particularly challenging task. One possible solution that can mitigate this uncertainty is to extend the control algorithms to a probabilistic

[29] https://equigy.com

setting using robust or stochastic control strategies such as those presented in [52]. The extension of the distributed control framework to a stochastic setting will be the subject of future work.

Load shifting vs real-time control Another challenge is that of the speed at which control can be performed at the local level. The algorithms we investigated and the algorithms developed in the Optiflex project act on loads and storage with relatively slow dynamics. Considering the time required to reach convergence during distributed control, realistically one could reach a control action every 10s - 1min depending on the complexity of the control hierarchy. These dynamics are too slow if one wanted to use only this method to 100% guarantee that network constraints are never violated. Distribution grids have higher dynamics compared to transmission networks, leading to stringent time constraints. In our view, there is, therefore, a need to complement flexibility coordination systems with local control algorithms that can react faster with simpler control logic, especially for battery storage systems, which can offer very fast control dynamics.

Hardware costs of the optimization infrastructure Another critical issue is the cost benefits related to the adoption of local measurement and control hardware. A system that allows a high-frequency direct measurement and actuation of loads definitely gives greater observability and controllability of flexibilities. On the other hand, the costs of such devices are not always justifiable from a financial point of view. We have shown that load shifting is a useful tool for energy communities and DSOs, but this goes as far as its marginal rollout costs are kept minimal. There is a plethora of technical solutions spanning from high observability and controllability but high costs to low controllability and observability but low costs. One can go from adding additional meters to the controlled devices, to connecting to the smart meter in situ and perform disaggregation (like it is done in LIC), to using the existing smart metering infrastructure for metering and control (like it is done in the Optiflex project). Of course, there is a tradeoff between costs and the effectiveness and granularity of the control. This is investigated in the LIC project. There is also also a tradeoff between the computational complexity and the locality of the control actions. When trying to enforce grid constraints and reduce losses, a DSO could prefer to centrally compute the optimal scheduling of groups of deferrable loads (e.g. boilers, heat pumps and EVs) for a sub-portion of the grid and broadcast it using the smart meters as actuators. Such a method, of course, does not have the fine-tuning capability of a system in which every single device is accessible. On the other hand, it strongly reduces computational and communication requirements. Also, it alleviates the need to know single devices' profiles using unsupervised NILM techniques, which proved to be extremely challenging (see section 9.4).

Outlook The work carried out so far at SUPSI has first focused on the design of market models for energy communities and control algorithms that work within them. Later, we focused on how to extend these local control systems to a larger scale by linking them together in a distributed way. Meanwhile, the blockchain-based market design and some of the control algorithms are being tested and compared to other solutions within the LIC pilot project. We believe that the natural outcome of this pilot and demonstration work is the extension of our pilots to a multi-level setting, in which several communities cooperate with each other to offer services, first to the DSO and then to the TSO, as illustrated in the roadmap in figure 80.

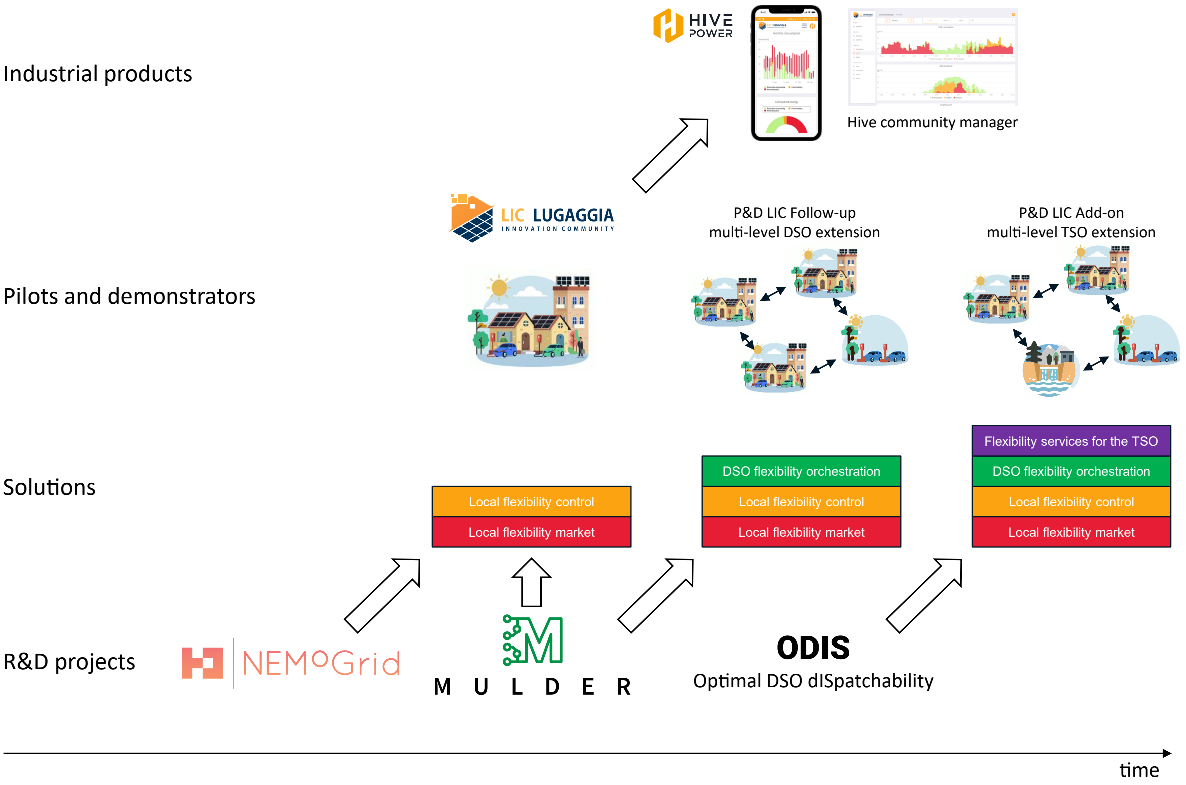

Industrial products
HIVE POWER
Hive community manager
Pilots and demonstrators
LIC LUGAGGIA INNOVATION COMMUNITY
P&D LIC Follow-up
multi-level DSO extension
P&D LIC Add-on
multi-level TSO extension
Solutions
Local flexibility control
Local flexibility market
DSO flexibility orchestration
Local flexibility control
Local flexibility market
Flexibility services for the TSO
DSO flexibility orchestration
Local flexibility control
Local flexibility market
R&D projects
NEMoGrid
MULDER
ODIS
Optimal DSO dISpatchability
time

Figure 80: Ideal future roadmap for integrating research work into pilots that result in industrial products helping the Energy Strategy 2050

A Annex: Heating system models

A.1 Control logic of heating systems

The heat pump control logic is based on two temperature sensors placed at different heights of the water tank, while the circulation pump connecting the tank with the building's heating element is controlled by an hysteresis on the temperature measure by a sensor placed inside the house.
We describe the control logic in a sequential way, following the heating components of the system. The first decision is taken by the building central controller, which decides its working mode, that is, if the building needs to be cooled or heated, based on a moving average of the historical data of the external temperature:

$$\begin{cases} wm_t & = -1 \quad \textbf{if} \quad T_{ma,t} > T_{max,ma} \\ wm_t & = \ \ 1 \quad \textbf{if} \quad T_{ma,t} < T_{min,ma} \\ wm_t & = \ \ 0 \quad otherwise \end{cases} \tag{55}$$

where the working mode wm_t is negative when the building requires to be cooled, positive when heating is required, and 0 when no actions are needed.$T_{max,ma}$ and $T_{min,ma}$ represent the maximum and minimum values of the external temperature's moving average, which is based on the past 7 days. The actual activation of the heating element is controlled by the hysteresis on the internal temperature of the building, T_z. If the working mode is positive, this is given by:

$$\begin{cases} s_{hy,t} & = 1 \quad \textbf{if} \quad (\ T_z < T_{min,hy} - \Delta T/2) \\ & \qquad \ \textbf{or} \quad (T_z < T_{min,hy} + \Delta T/2 \quad \textbf{and} \quad s_{hy,t-1}) \\ s_{hy} & = 0 \quad otherwise \end{cases} \tag{56}$$

where $s_{hy,t}$ is the state of the hysteresis at time t, 1 meaning that the circulation pump of the heating element must be activated, and ΔT was chosen to be equal to 1$^\circ C$. For completeness, we report also the control logic when the building is in cooling mode:

$$\begin{cases} s_{hy,t} & = 1 \quad \textbf{if} \quad (\ T_z > T_{max,hy} + \Delta T/2) \\ & \qquad \ \textbf{or} \quad (T_z > T_{max,hy} - \Delta T/2 \quad \textbf{and} \quad s_{hy,t-1}) \\ s_{hy} & = 0 \quad otherwise \end{cases} \tag{57}$$

The incoming water temperature in the heating element is then modulated linearly through a 3-way valve between a maximum and minimum value, based on the external temperature, both in the heating and cooling modes. When operative, the heating element requests hot or cold water to the water tank, which control logic is based on two temperature sensors located in two different layers. When the building is in heating mode, the control logic is a simple hysteresis based on the temperature of the sensor in the uppermost layer, which is identical to the one in (56). When in cooling mode, the control logic is the following:

$$\begin{cases} s_{hy,t} & = -1 \quad \textbf{if} \quad (\ T_{up} > T^c_{max} + \Delta T/2) \\ & \qquad \quad \textbf{or} \quad T_{low} > T^c_{max} + \Delta T/2 \\ s_{hy,t} & = 0 \quad \ \ \textbf{if} \quad (\ T_{low} < T^c_{min}) \ \textbf{or} \ (T_{up} < T^c_{max} - \Delta T/2) \\ s_{hy,t} & = s_{hy,t-1} \quad otherwise \end{cases} \tag{58}$$

where T_{up} and T_{low} are the temperature measured by the upper and lower sensors, respectively, and T^c_{min} and T^c_{max} are the minimum and maximum desired temperatures of the water in the tank while in cooling mode.
The value of $s_{hy,t}$ is then communicated to the HP. In the case in which the HP is also used for the domestic hot water (DHW), the DHW tank is always served with priority by the HP.

A.2 Heat distribution system

Floor heating was modeled starting from first principles. Considering a fixed and uniform temperature for the ground and the building internal temperature at each time-step and stationary conditions, we can retrieve the analytical expression of the temperature profile along the pipe, through the energy balance on an infinitesimal element of the pipe. This can be expressed as:

$$\frac{\partial c T_x}{\partial t} = \Phi_x - \Phi_{x+\partial x} + \dot{q}_{up} + \dot{q}_{down} \tag{59}$$

where c is the heat capacity in J/K, x is the distance from the pipe entrance, T_x is the temperature of the water inside the pipe at x, Φ are enthalpy flows at the entrance and exit of the considered infinitesimal volume, $\dot{q}_{up}$ and $\dot{q}_{down}$ are the heating powers from the building and from the ground. Expressing the latter through equivalent resistance taking into account convective and conductive effects, the balance in steady state can be rewritten as:

$$\frac{\dot{m} c_p}{\rho^*} \frac{\partial T_x}{\partial x} = \frac{R_{down} T_z + R_{up} T_g}{R_{down} + R_{up}} - T_x = T^a - T_x \tag{60}$$

where T^a is the asymptotic temperature and where:

$$R_{down} = \frac{1}{h_{in} w} + \frac{1}{h_{u,eq} w} + R_u \tag{61}$$

$$R_{up} = \frac{1}{h_{in} w} + R_g \tag{62}$$

$$\rho^* = \frac{R_{up} + R_{down}}{R_{up} R_{down}} \tag{63}$$

where w is the diameter of the tube, h_{in} is the internal coefficient of heat transfer, which can be retrieved using available empirical relation for fully developed flow with fixed temperature at the boundary conditions [67], $h_{u,eq}$ is the heat transfer coefficient between the floor and the building air including both the effect for natural convection and radiation. The values of $h_{u,eq}$ can be found in the literature [68],[69]. The value of the thermal resistances R_u and R_g, towards the floor and the ground, can be found in the literature as well. We can reformulate (60), making it adimensional through a change of variable:

$$\frac{\partial \Theta}{\partial \mathcal{X}} = -\Theta \tag{64}$$

from which solution we can retrieve the temperature profile of the water inside the pipe:

$$T_x = T^a + (T_0 - T^a) e^{\frac{-x\rho^*}{\dot{m} c_p}} \tag{65}$$

where T_0 is the temperature of the water at the pipe inlet. We can use (65) to retrieve the heating power flowing into the building, integrating $\dot{q}_{up}(x)$ along the pipe.

$$\dot{Q}_{up} = \int_0^L \dot{q}_{up}(x) \mathrm{d}x = \int_0^L \frac{T(x) - T_z}{R_{up}} \mathrm{d}x \tag{66}$$

where L is the length of the serpentine. Integrating, we obtain

$$\dot{Q}_{up} = \frac{(T^a - T_z) L - (T_L - T_0) \frac{\dot{m} c_p}{\rho^*}}{R_{up}} \tag{67}$$

where T_L is the temperature of the water at the outlet of the serpentine. Note that the equation (67) tends to $(T_L - T_0)\dot{m} c_p$ when R_{down} increase and R_{up} is kept fixed.

The nominal mass flow of the heating system and the length of the serpentine are found as the solution of the following optimization problem:

$$\underset{L,\dot{m}}{\text{argmin}} \left(\dot{Q}_{up}(L) - \dot{Q}_{nom}\right)^2 + 10^{-3} \left(\dot{m} - \dot{m}_{nom}\right)^2 \tag{68}$$

where $\dot{m}_{nom}$ is a reference mass flow, equal to $0.1\,[kg/s]$ and $\dot{Q}_{nom}$ is the power required to keep the building internal temperature constant under reference conditions (we used an external temperature of -4$°C$ and a desired internal temperature of 20 $°C$):

$$\dot{Q}_{nom} = \frac{\Delta T_{ref}}{R} \tag{69}$$

where R is the resistance of an equivalent RC circuit describing the heating dynamics of the building.

A.3 Boiler model

The dynamic equation describing the evolution of the temperature of the tank's layers is the following:

$$C\frac{\partial T_i}{\partial t} = \dot{Q}^u_{buo,i} + \dot{Q}^d_{buo,i} + \dot{Q}_{h,i} + \dot{Q}_{loss,i} + \dot{Q}^u_{cond,i} + \dot{Q}^d_{cond,i} + c_p\dot{m}(T_{i-1} - T_i) \tag{70}$$

where T_i is the temperature of the i_{th} layer, Q^u_{buo},Q^d_{buo},Q^u_{cond},Q^u_{cond} are the thermal powers due to buoyancy and conduction, from the lower and upper layer, respectively. The last term represents the enthalpy flow due to mass exchange, while C is the thermal capacity of the layer, in $[J/K]$ and $Q_{h,i}$ is the thermal power due to an electric resistance (for the boiler) or an heat exchange (for the heating system buffer). The expression for the above thermal power are the following:

$$\begin{align}
\dot{Q}^u_{buo,i} &= k\,\max(T_{i+1} - T_i, 0)N, \quad 0 \quad for \quad i = N \tag{71}\\
\dot{Q}^d_{buo,i} &= k\,\max(T_{i-1} - T_i, 0)N, \quad 0 \quad for \quad i = 1 \tag{72}\\
\dot{Q}^u_{cond,i} &= u_{amb}(T_{i+1} - T_i), \quad 0 \quad for \quad i = N \tag{73}\\
\dot{Q}^d_{cond,i} &= u_{amb}(T_{i-1} - T_i), \quad 0 \quad for \quad i = 1 \tag{74}\\
\dot{Q}_{loss,i} &= u_{amb}(T_{ext} - T_{i)} \tag{75}\\
\dot{Q}_{h,i} &= \dot{Q}_{tot}/n_h \quad if \quad i \in \mathcal{I} \tag{76}\\
& \tag{77}
\end{align}$$

where N is the number of layers, u_{amb} is the equivalent thermal loss coefficient with the ambient and $\mathcal{I}$ is the set of the n_h layers heated by the heat exchange (or electric resistance). The buoyancy model is the one proposed in the IDEAS library [70]. Detailed description of the parameters for the boiler model can be found in [14].

B References

[1] A. J. D. Rathnayaka, V. M. Potdar, T. Dillon, and S. Kuruppu, “Framework to manage multiple goals in community-based energy sharing network in smart grid,” *International Journal of Electrical Power & Energy Systems*, vol. 73, pp. 615–624, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0142061515002136

[2] R. Zafar, A. Mahmood, S. Razzaq, W. Ali, U. Naeem, and K. Shehzad, “Prosumer based energy management and sharing in smart grid,” *Renewable and Sustainable Energy Reviews*, vol. 82, pp. 1675–1684, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032117310894

[3] Y. Parag and B. K. Sovacool, “Electricity market design for the prosumer era,” *Nature Energy*, vol. 1, no. 4, p. 16032, 2016. [Online]. Available: http://www.nature.com/articles/nenergy201632

[4] A. J. D. Rathnayaka, V. M. Potdar, T. S. Dillon, O. K. Hussain, and E. Chang, “A Methodology to Find Influential Prosumers in Prosumer Community Groups,” *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 706–713, 2014.

[5] A. Mahmood, A. R. Butt, U. Mussadiq, R. Nawaz, R. Zafar, and S. Razzaq, “Energy sharing and management for prosumers in smart grid with integration of storage system,” in *2017 5th International Istanbul Smart Grid and Cities Congress and Fair (ICSG)*, 2017, pp. 153–156.

[6] M. Kubli, M. Loock, and R. Wüstenhagen, “The flexible prosumer: Measuring the willingness to co-create distributed flexibility,” *Energy Policy*, vol. 114, pp. 540–548, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0301421517308704

[7] K. Kotilainen, S. J. Mäkinen, P. Järvetausta, A. Rautiainen, and J. Markkula, “The role of residential prosumers initiating the energy innovation ecosystem to future flexible energy system,” in *2016 13th International Conference on the European Energy Market (EEM)*, 2016, pp. 1–5.

[8] “LAEI 734.7, https://www.admin.ch/opc/it/classified-compilation/20042411/index.html.”

[9] “OAEI 734.71, https://www.admin.ch/opc/it/classified-compilation/20071266/index.html.”

[10] M. Hong, Z. Q. Luo, and M. Razaviyayn, “Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems,” *SIAM Journal on Optimization*, vol. 26, pp. 337–364, 2016.

[11] L. Nespoli, M. Salani, and V. Medici, “A rational decentralized generalized Nash equilibrium seeking for energy markets,” in *2018 International Conference on Smart Energy Systems and Technologies, SEST 2018 - Proceedings*. Institute of Electrical and Electronics Engineers Inc., 10 2018.

[12] L. Nespoli and V. Medici, “Constrained hierarchical networked optimization for energy markets,” in *Proceedings - 2018 IEEE PES Innovative Smart Grid Technologies Conference Europe, ISGT-Europe 2018*, 2018.

[13] L. S. Shieh, H. Wang, and R. E. Yates, “Discrete-continuous model conversion,” *Topics in Catalysis*, 1980.

[14] L. Nespoli, A. Giusti, N. Vermes, M. Derboni, A. Rizzoli, L. Gambardella, and V. Medici, “Distributed demand side management using electric boilers,” *Computer Science - Research and Development*, vol. 32, no. 1-2, 2017.

[15] G. Belgioioso and S. Grammatico, “Projected-gradient algorithms for Generalized Equilibrium seeking in Aggregative Games are preconditioned Forward-Backward methods,” *arXiv*, 2018.

[16] M. Diekerhof, F. Peterssen, and A. Monti, “Hierarchical Distributed Robust Optimization for Demand Response Services,” *IEEE Transactions on Smart Grid*, vol. 3053, no. c, p. 1, 2017.

[17] P. McNamara and S. McLoone, "Hierarchical Demand Response for Peak Decomposition," *IEEE Transactions on Smart Grid*, vol. 6, no. 6, pp. 2807–2815, 2015.

[18] P. Braun, T. Faulwasser, L. Gr, C. M. Kellett, S. R. Weller, and K. Worthmann, "Hierarchical Distributed ADMM for Predictive Control with Applications in Power Networks," *IFAC Journal of Systems and Control*, vol. 3, pp. 10–22, 2018.

[19] K. Baker, J. Guo, and S. Member, "Distributed MPC for Efficient Coordination of Storage and Renewable Energy Sources Across Control Areas," *IEEE Transactions on Smart Grid*, vol. 7, no. 2, pp. 992–1001, 2016.

[20] X. Su, M. A. S. Masoum, and P. J. Wolfs, "Multi-Objective Hierarchical Control of Unbalanced Distribution Networks to Accommodate More Renewable Connections in the Smart Grid Era," *IEEE Transactions on Power Systems*, vol. 31, no. 5, pp. 3924–3936, 2016.

[21] L. Nespoli and V. Medici, "Constrained hierarchical networked optimization for energy markets," in *Proceedings - 2018 IEEE PES Innovative Smart Grid Technologies Conference Europe, ISGT-Europe 2018*, 2018.

[22] L. Stella, A. Themelis, and P. Patrinos, "Forward – backward quasi-Newton methods for nonsmooth optimization problems," *Computational Optimization and Applications*, vol. 67, no. 3, pp. 443–487, 2017.

[23] "Ethereum https://ethereum.org."

[24] "General Data Protection Regulation https://eur-lex.europa.eu/eli/reg/2016/679/oj."

[25] "Proof of authority https://en.wikipedia.org/wiki/Proof\textunderscore of\textunderscore authority."

[26] "Aragon, https://aragon.org."

[27] "AragonOS, https://github.com/aragon/aragonOS."

[28] "Voting app, https://github.com/aragon/aragon-apps/tree/master/apps/voting."

[29] "MIT license, https://en.wikipedia.org/wiki/MIT_License."

[30] "ERC20, https://github.com/ethereum/eipsi/ssues/20."

[31] "https://blockchainhub.net/blog/blog/blockchain-scalability-statchannels-sidechains/."

[32] "https://alephzero.org/blog/what-is-the-fastest-blockchain-and-why-analysis-of-43-blockchains/."

[33] "https://academy.binance.com/en/glossary/transactions-per-second-tps."

[34] "https://medium.com/@aakash\textunderscore13214/the-scalability-trilemma-in-blockchain-75fb57f646df."

[35] "https://www.bitorb.com/campus/what-is-the-scalability-trilemma."

[36] "Lightning Network https://lightning.network/."

[37] "Raiden Network https://raiden.network/."

[38] "Liquidity https://liquidity.network/."

[39] "Celer https://www.celer.network/."

[40] "Cosmos Network https://cosmos.network/."

[41] "Byzantine fault tolerance https://en.wikipedia.org/wiki/Byzantine\textunderscore fault."

[42] "Proof of stake https://en.wikipedia.org/wiki/Proof\textunderscore of\textunderscore stake."

[43] “Go programming language https://golang.org/.”

[44] “https://asecuritysite.com/encryption/go\textunderscore priv.”

[45] P. McCorry S. F. Shahandashti and F. Hao, “A smart contract for board- room voting with maximum voter privacy,” *International Conference on Financial Cryptography and Data Security*, pp. 357–375, 2017.

[46] C.-P. Schnorr, “Efficient signature generation by smart card,” *International Conference on Financial Cryptography and Data Security*, vol. 4, no. 3, pp. 161–174, 1991.

[47] A. S. A. Fiat, “How to prove yourself: Practical solutions to identification and signature problems,” *Conference on the Theory and Application of Cryptographic Techniques*, vol. 4, no. 3, pp. 186–194, 1986.

[48] R. Cramer, I. Damgård, and J. B. Nielsen, “Multiparty computation from threshold homomorphic encryption,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2045, pp. 280–300, 2001.

[49] “Cosmos Hub https://hub.cosmos.network.”

[50] H. Hens, *Applied Building Physics: Ambient Conditions, Building Performance and Material Properties: Second Edition*, 2016.

[51] N. Pflugradt and B. Platzer, “Behavior based load profile generator for domestic hot water and electricity use,” pp. 1–9, 2012.

[52] L. Nespoli, “MODEL BASED FORECASTING FOR DEMAND RESPONSE STRATEGIES,” Ph.D. dissertation, 2019.

[53] T. Afjei, U. Schonhardt, C. Wemhöner, M. Erb, H. R. Gabathuler, H. Mayer, G. Zweifel, M. Achermann, R. von Euw, and U. Stöckli, “Standardschaltungen für Kleinwärmepumpenanlagen Teil 2: Grundlagen und Computersimulationen. Schlussbericht,” Tech. Rep., 2002.

[54] L. Girardin, “A GIS-based Methodology for the Evaluation of Integrated Energy Systems in Urban Area,” p. 218, 2012. [Online]. Available: http://infoscience.epfl.ch/record/170535

[55] K. Arendt, M. Jradi, H. R. Shaker, C. T. Veje, and S. Denmark, “Comparative Analysis of white- , gray- and black-BOX models for thermal simulation of indoor environment: teaching building case study,” in *Building Performance Modeling Conference*, 2018, pp. 173–180.

[56] G. Reynders, J. Diriken, and D. Saelens, “Quality of grey-box models and identified parameters as function of the accuracy of input and observation signals,” *Energy and Buildings*, vol. 82, pp. 263–274, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.enbuild.2014.07.025

[57] P. Bacher and H. Madsen, “Identifying suitable models for the heat dynamics of buildings,” *Energy and Buildings*, vol. 43, no. 7, pp. 1511–1522, 7 2011. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0378778811000491

[58] T. Schütz, H. Harb, R. Streblow, and D. Müller, “Comparison of models for thermal energy storage units and heat pumps in mixed integer linear programming,” 2015.

[59] J. S. Stein, “The photovoltaic Performance Modeling Collaborative (PVPMC),” *Conference Record of the IEEE Photovoltaic Specialists Conference*, pp. 3048–3052, 2012.

[60] “https://en.wikipedia.org/wiki/ARM\textunderscore architecture.”

[61] “https://www.dlms.com/.”

[62] “https://github.com/Gurux/Gurux.DLMS.Python.”

[63] “https://en.wikipedia.org/wiki/IEC_61968.”

[64] GSE, “Regole tecniche per l’accesso al servizio di valorizzazione e incentivazione dell’energia elettrica condivisa,” 2020. [Online]. Available: https://www.gse.it/documenti_site/Documenti%20GSE/Servizi%20per%20te/AUTOCONSUMO/Gruppi%20di%20autoconsumatori%20e%20comunita%20di%20energia%20rinnovabile/Regole%20e%20procedure/Regole%20Tecniche%20per%20accesso%20al%20servizio%20di%20valorizzazione%20e%20incentivazione%20energia%20elettrica%20condivisa.pdf

[65] M. D. Galus, “Die Bedeutung digitaler Plattformen für den Informations- und Datenaustausch im Strommarkt 2.0 der Schweiz,” in *Realisierung Utility 4.0 Band 1*. Springer Fachmedien Wiesbaden, 2020, pp. 511–528. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-658-25332-5_30

[66] M. Bachmann, A. Hurni, and U. Meyer, “Umverteilungseffekte aufgrund reduzierter Netznutzungsbeiträge beim Eigenverbrauch – eine Quantifizierung.” [Online]. Available: https://smartgrid-schweiz.ch/wp-content/uploads/2020/08/2020_VSGS_Whitepaper_Eigenverbrauch.pdf

[67] F. P. Incropera, D. P. DeWitt, T. L. Bergman, and A. S. Lavine, *Fundamentals of Heat and Mass Transfer*, 2007.

[68] T. Cholewa, M. Rosiński, Z. Spik, M. R. Dudzińska, and A. Siuta-Olcha, “On the heat transfer coefficients between heated/cooled radiant floor and room,” *Energy and Buildings*, vol. 66, pp. 599–606, 2013.

[69] P. Wallenten, “Heat Transfer Coefficients in a Full Scale Room With and Without Furniture,” *Lund Institute of Thechnology*, pp. 1–8, 1999.

[70] R. D. Coninck, R. Baetens, D. Saelens, A. Woyte, L. Helsen, A. Mechanics, and B. P. Section, “Rule-based demand side management of domestic hot water production with heat pumps in zero energy neighbourhoods,” *Journal of Building Performance Simulation*, 2013.