Informatics

Scheduling and Controlling Smart Micro Grids / Energy Communities as Kubernetes First-Class Citizens

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Software Engineering & Internet Computing

eingereicht von

Stephan Lukas Podlipnig, BSc
Matrikelnummer 01125621

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ. Prof. Dr. Schahram Dustdar
Mitwirkung: Dr. Andrea Morichetta

Wien, 22. Jänner 2025

Stephan Lukas Podlipnig

Schahram Dustdar

Informatics

Scheduling and Controlling Smart Micro Grids / Energy Communities as Kubernetes First-Class Citizens

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Software Engineering & Internet Computing

by

Stephan Lukas Podlipnig, BSc
Registration Number 01125621

to the Faculty of Informatics

at the TU Wien

Advisor: Univ. Prof. Dr. Schahram Dustdar
Assistance: Dr. Andrea Morichetta

Vienna, 22nd January, 2025

Stephan Lukas Podlipnig

Schahram Dustdar

Erklärung zur Verfassung der Arbeit

Stephan Lukas Podlipnig, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 22. Jänner 2025

Stephan Lukas Podlipnig

Kurzfassung

Die Zukunft des Stromnetzes ist dezentralisiert: der Zusammenschluss einer immer größer werdenden Zahl von Haushalten und Heimkraftwerken mit Eigenstromerzeugung für den eigenen wie auch gemeinschaftlichen Nutzen - auf Basis erneuerbarer Energien.

Unser Denken und Verhalten bezüglich der Nutzung von Energie ist gerade dabei, sich langfristig und grundlegend zu verändern: Erneuerbare Energiegemeinschaften (EEG) ermöglichen den Austausch von Energie in der Nachbarschaft und kommen europaweit immer stärker zum Einsatz. Erzeuger und Verbraucher in örtlicher Nähe bilden dabei Energiegemeinschaften, um ihren jeweiligen Energiebedarf und Energieüberschuss über das bestehende, öffentliche Stromnetz zu teilen und so auszugleichen. Diese lokale Energieumverteilung führt zu einer größeren Unabhängigkeit von zentralen Kraftwerken, reduziert Energieschwankungen auf der Hochspannungsnetzebene, und führt in weiterer Folge zu stabilieren Stromnetzen und geringeren Kosten.

Der Einsatz von Informations- und Kommunikationstechnologie (IKT) für den Echtzeit-Informationsaustausch zur aktuellen Energienutzung und zu etwaigen Energieüberschüssen der Teilnehmer einer Energiegemeinschaft, macht die EEG nicht nur intelligent, sondern ist entscheidend, um Effizienz weitere positive Effekte hinsichtlich Kosten und Umwelt zu maximieren.

Diese Arbeit schlägt eine Lösung vor, um Produzenten und Verbraucher im Heimbereich zu sog. "Microgrids" zu vernetzen und sie als EEGs zu modellieren, sodass die Teilnehmer ihre Energie nicht nur zufällig, sondern intelligent miteinander teilen. Dies wird durch ein Framework ermöglicht, das mit einem cloud-basierten Energiemanagementsystem (EMS) alle Teilnehmer als "first-class citizen" in Kubernetes miteinander vernetzt, und bestehende IKT für die Kommunikation zwischen Cloud und Teilnehmern herstellerunabhängig unterstützt.

Die Ergebnisse dieser Arbeit zeigen die Effektivität von intelligenten Energiegemeinschaften mit einem zentralen EMS, unter Verwendung realer und simulierter Energiedaten österreichischer Erzeuger- und Verbraucherhaushalte, die sich zu EEGs zusammenschließen, dadurch als virtuelle Mikrokraftwerke innerhalb des Stromnetzes agieren, um Energie miteinander intelligent zu teilen.

Abstract

The future of the power grid is decentralized: the interconnection of a vast amount of residential sites and micro power plants, for self-sufficiency, individual and community benefit, all powered by renewable energies.

Energy communities are facilitating the sharing of energy within neighborhoods, which is becoming state-of-the-art across Europe and is about to shape how we live and think energy usage forever: Prosumers and consumers in proximity are establishing energy communities to collaborate on distributing both their energy demand and excess via the existing public power grid. This localized energy redistribution leads to greater independence from central power plants, reduces energy fluctuations at the high-voltage grid level, and ultimately results in more stable power grids and lower costs.

The use of Information and Communication Technology (ICT) for real-time exchange of current energy demand and excess data among participants in an energy community not only makes them intelligent but is also crucial for maximizing the positive environmental and ecological effects of energy sharing.

This thesis introduces a solution that connects residential prosumers and consumers into smart microgrids by conceptualizing them as energy communities, s.t. they not only share energy randomly but interact intelligently. This is achieved through a cloud-based Energy Management System (EMS), all seamlessly integrated as first-class citizens within Kubernetes, and leveraging existing ICT for cloud-to-member communication in a vendor-agnostic manner.

The results will showcase the effectiveness of intelligent energy communities with a central EMS coordinator, using real-world and simulated energy data of Austrian prosumer and consumer households, that are grouped together, acting as virtual micro power plants in central grid's perspective to share energy intelligently with each other.

Contents

CHAPTER 1

Introduction

1.1 Motivation

Decentralizing the energy realm implies shifting away from centralized power suppliers, towards a multitude of small and interconnected microgrids (MG), all physically embedded in the public electric grid but also interconnected by ICT, to build so-called energy communities.

An energy community (EC) is the organizational grouping of several energy consumers or producers connected to the power grid that are (primarily) within the same low voltage power grid. Participating sites, ranging from residential to commercial and industrial sites (including local power plants), are able to share their produced energy and to cover their consumed energy, by releasing or withdrawing from and to each member within the EC, through the public power grid.

In Austria and Germany, the EC participant's energy suppliers measure and obtain all their smart meter power values with a 15-minute sampling rate in order to virtually allocate grid feed-ins from a participant that occurred within the same 15-minute interval among the other consuming participants. Those amounts of shared energies by participants are not billed by the central energy suppliers but within the EC [ene]. Furthermore, [ene] states that:

> The public power grid is only partially used, discounted network charges (local network tariffs) are applied.

Sharing energy locally implies that ECs require at least one member with an energy production facility and/or energy storage resource (ESR) that is able to act as local energy provider within the EC, by feeding its energy into the grid at a specific point in time, when needed by another member. If other EC members consume energy within the same 15-minute interval, this amount of energy is then considered to be shared locally within the EC. The consumers and producers involved are then cleared by the community. Remaining excess energy, or demand not covered by such distributed energy resources (DERs), is billed by tariff of the energy supplier as usual. In the best case, all required energy is provided by the community; in the worst case, the energy supplier's tariff is used for every kWh of not-community-allocatable energy.

The legal basis for creation of ECs was established by *Directive (EU) 2018/2001 of the European Parliament and of the Council of 11 December 2018 on the promotion of the use of energy from renewable sources* [DSIS$^+$21]. In Austria, this was turned into domestic law by "Erneuerbaren-Ausbau-Gesetz" (EAG) in 2021 [ris] due to an EU directive [eud].

However, the benefits of ECs are not only economical. Decentralized producers, especially from renewable energy, introduce a huge environmental benefit due to stabilizing effects on the public power grid (and reducing the risk of blackouts) by reducing demand and dependence on centralized power plants, often over long distances.

1.1.1 Towards smart (optimized, autonomous) energy communities

Smart energy communities can further increase this positive ecological and environmental effect, by communicating the energy demand of the EC and forwarding this information to members with energy flexibility (e.g. energy storage (ESR) or members with dynamic load shifting capabilities).

Ecologically optimal energy communities operate closely to the so-called "island mode", making them independent of the central grid for both feed-in and demand, appearing as isolated and transparent in its perspective. Island mode is achieved by constantly evaluating and re-balancing current energy demands of all members s.t. an equilibrium is reached.

To achieve a high level of independence (equilibrium), prosuming DERs can be (financially) encouraged to release energy to the grid within the same 15-minute interval, for the benefit of the community and the central grid. On the other hand, consumers can be encouraged to adapt their momentary consumption behavior based on the current energy balance of the community by shifting loads, for example charging their electric vehicles during times of excess energy within the EC.

The member count, mixture, flexibility, and energy storage sizes of all members of the community determine the efficiency of the EC. The algorithmic search for optimal compositions of communities, out of a set of members, is beyond the scope of this thesis. This thesis focuses on optimizing ECs and compares the performance of different community compositions and central EMS strategies against each other, by providing and using a cloud-based framework.

1.2 Research Questions

The research carried out and the implementation provided by this thesis is based on the following research questions. The answers to those questions are provided as part of the conclusion section 7.

- **RQ.1: What are the concrete challenges of energy communities?**

  The introduction of energy communities in Europe is accompanied by some organizational and technical challenges (e.g. smart meter installation, counting point registration, community searching/matching). Once these physical requirements are met and a participant is able to become part of an EC, the question arises: What other problems do participants and ECs face and should be addressed in order to further grow acceptance and adoption rate of ECs?

- **RQ.2: How can smart energy communities be modeled, managed, controlled and optimized through service-oriented architectures?**

  This question connects the two independent areas of energy communities (ECs) and cloud-based service-oriented architecture (SOA) and aims at the main contribution of this thesis: whether/how ICT (specifically SOA) can be used to model and optimize ECs to bring real benefits to the participants.

- **RQ.3: Is a central EMS and data exchange on momentary energy demand and excess between participants required for optimal energy sharing?**

  RQ.3 addresses the question of whether and to what extent ICT is necessary to optimize energy sharing among EC participants, and whether the effort of such a digitalization is worthwhile.

1.3 Solution proposal

To make energy communities smart, the following building blocks are proposed:

1. information gathering: knowledge and information exchange about the current energy demand and flexibility of every member in "real-time" (updates within seconds)
2. energy flexibility: encouraging members with respective capabilities to share their ESR and/or perform consumption load shifting
3. controllability: mechanism to proactively control all or a subset of the members and their flexibility on-demand by a central EMS instance

This thesis proposes a service-oriented framework for modeling, controlling, optimizing, simulating, and evaluating energy communities. The framework provides a template for building real-time energy data exchange between the individual's HEMS (home energy management system) and the framework's central EMS. The central EMS's control loop provides a mechanism to control the member's local energy flexibility, as calculated by the obtained community data and its chosen strategy, and communicated to the members' HEMS.

1.4 Methodology

To evaluate the results and contribution of this thesis, the following research methods are applied:

1.4.1 Literature review and gap analysis

- Reviewing state-of-the-art literature, regulations and existing solutions in the domains of the smart grids, energy communities, energy management systems and their algorithms
- Analyzing and discussing the gaps and overlaps between state-of-the-art and the proposed framework.

1.4.2 Framework design and implementation

- Implementing a generic framework to build energy communities that are deployed to Kubernetes, including an operator with different custom resource definitions to model and control energy communities and agents
- Designing a generic API bridge to map HEMS to agents in the framework.
- Implementing an EMS algorithm based on MC sampling.
- To simulate whole energy communities, a set of tools is provided that allows automated setups of the environment for simulation execution, creating energy communities with a multitude of members and different behaviors, fast forwarding time, and generation of reporting data and graphs.

1.4.3 Agent behavior modeling

- Since it is unfeasible to execute and validate EMS actions derived by the community EMS on real members of a smart grid (agents), a model of prosumers has to be designed and implemented. This allows to simulate actions, e.g. charging processes of ESRs, taking into account physical limits and further effected measurements (e.g. momentary grid meter measurements).

1.4.4 Simulation

- Multiple scenarios (energy community compositions) are created and simulated for performance evaluation.

- A mechanism to simulate ICT fault-inject scenarios, i.e. connection interrupts, delays, unavailability of nodes or agents, is provided.

1.4.5 Quantitative performance evaluation

- KPIs for performance evaluation are defined and the underlying formulas provided

- The results of the simulations of different ECs compositions compared against each other using the KPIs.

- The performances of different EMS strategies are compared against each other.

1.4.6 Qualitative evaluation

- The nonfunctional aspects of the proposed framework are analyzed, challenged, and compared with existing state-of-the-art literature.

- The quality and origin of the energy test data provided by consumers and consumers is evaluated.

- The results of all EC simulations and case studies are discussed and generalized statements are derived.

1.5 Nomenclature

Sets and indices	
EC	Energy community as set of members
m	Index of member of an EC
N	Number of members in the EC
T	Length of time horizon
t	Index of time

Energy prices (for member m, in time step t)	
$c_{im}^{t,m}$	Cost [€] of 1 kWh imported energy from central grid
$c_{ex}^{t,m}$	Cost [€] of 1 kWh exported energy into central grid
$\underline{c}_{im}^{t,m}$	Cost [€] of 1 kWh imported energy from community
$\underline{c}_{ex}^{t,m}$	Cost [€] of 1 kWh exported energy into the community

Member momentary power rates	
$p_G^{m,t}$	Grid power [W] (positive=import) of member m, in time step t, on average
$p_{PV}^{m,t}$	PV production power [W] of member m, in time step t, on average
$p_C^{m,t}$	Consumed power [W] of member m, in time step t, on average
$p_S^{m,t}$	Charging power [W] into ESR of member m, in time step t, on average
$\Delta p_{target}^{m,t}$	Target power delta [W], by central EMS for member m, in time step t

Accumulated energies per member over time (sum from $t = 0..T$)	
E_{Gim}^{m}	Grid imported energy [Wh] of member m
E_{Gex}^{m}	Grid exported energy [Wh] of member m
E_{PV}^{m}	PV produced energy [Wh] of member m
E_{C}^{m}	Consumed energy [Wh] of member m

Member physical power limits

$\widehat{E}_S^m$	Total energy capacity [Wh] of ESR of member m
$\widehat{p}_S^m$	Max. charging/discharging power [W] of ESR of member m
$\widehat{p}_{Gex}^m$	Max. grid export power [W] of member m
$\widehat{p}_{Gim}^m$	Max. grid import power [W] of member m

Virtual community physical power limits (as sum of all member rates)

$\underline{p}_G^t$	Total EC grid power [W] (positive=import) in time step t, on average
$\underline{p}_{PV}^t$	Total EC PV production power [W] in time step t, on average
$\underline{p}_C^t$	Total EC consumption power [W] in time step t, on average

Key performance indicators (defined in 4.3)

A^m	Autarky [%] of member m, over whole time horizon T
$\underline{A}$	Autarky [%] of the EC, over whole time horizon T
R^m	Reward [€] of member m, over whole time horizon T
$\underline{R}$	Reward [€] of the EC, over whole time horizon T
$\underline{Q}^t$	Equilibrium [%] of the EC, in time step t
$\underline{Q}$	Equilibrium [%] of the EC, over whole time horizon T

CHAPTER 2

State-of-the-art

The concept of smart micro-grids, which involve a network of interconnected smaller producers and consumers linked through communication systems, is not a new one. In fact, it has been the subject of extensive research and discussion for several decades.

2.1 Power grid and communication networks

"While the grid has benefited from many innovations and improvements over the past century, in some respects its basic design has changed little from the days of Edison and Tesla in the 1880s." [Arn11]. The authors describe the challenges and disruptive opportunities of transitioning our grid to the future of energy supply, as the grid as we know it is a "one-way transmission and distribution system to consuming devices that have no information about the cost of electricity or whether the grid is overloaded.".

Opposed to the known grid, Gungor et al. discuss in [GSK+11] the smart grid from several points of view, by using the definition:

> Modern electric power grid infrastructure for improved efficiency and reliability through automated control, high-power converters, modern communications infrastructure, sensing and metering technologies, and modern energy management techniques based on the optimization of demand, energy, and network availability, and so on.

This and many other authors address the communication network strategy as the key component of the smart grid infrastructure [Arn11] [GSK+11] [RPD+14] [KS19] [KNHH14], i.e.

> The essential concept of the smart grid is the integration of advanced information technology, digital communications, sensing, measurement, and control technologies into the power system. [Arn11]

[RPD+14] intended to build such a Java-based ICT for energy sharing in Australia as future work, but was not found at the time of writing this thesis.

2.2 Micro Grids and their limitations

[Hat14] defines micro grids as an "Integration platform for supply-side, storage units and demand resources (controllable loads)" and its ability "to minimize green house gases, help the power grid with load balancing and voltage control and assist power markets". Another definition within this book states that MGs are defined as "distribution systems with distributed energy sources, storage devices and controllable loads ..." and that "Coordination and control of DER is the key feature that distinguishes Microgrids from simple distribution feeders with DER". Another fundamental claim of this book, that this thesis builds upon, is that "effective energy management within Microgrids is the key to achieving vital efficiency benefits by optimizing production and consumption of energy".

Balancing between production and consumption within a micro-grid using an EMS, turning it into a smart grid, is a central aspect of this thesis.

In 2006, the European Technology Platform defines in the "Vision and Strategy for Europe's Electricity Networks of the Future" [BBB+06] that a smart grid is an "electricity network that can intelligently integrate the actions of all users connected to it".

[ZEB18] defines a micro-grid as a "low-voltage distribution network of interconnected DERs" and states its limitations: high investment costs, control issues, lack of system protection, customer privacy, optimal use of energy sources.

[Hat14] stated in 2014 that "cost, policy and technology barriers have largely restrained the wide deployment of microgrids". As of the writing of this thesis (2023), all of those barriers have been largely addressed:

- DER efficiency and adoption lead to drastically lower prices for PV and ESR in the past 10 years.
- The EU "solar rooftop initiative" will require installing PV on all new public, commercial and residential buildings by 2029.
- The global energy crisis due to the war in Ukraine led to unprecedented high prices in the electrical energy market.
- Heavily ongoing and past research on micro grids, energy management, distributed systems, zero-touch networks, etc.

[**MAHS22**] did research using Germany's standardized "FNN control box" that allows grid operators to remotely control load shedding and feed-in limitation for residential and commercial sites for another purpose: [MAHS22] evaluated whether the control box could be also used to "decide by itself, based on local measurements, if the grid is overloaded" to derive further local actions. The box was therefore locally connected to the site's e.g. charging controllers to decide optimal charging time windows for grid protection.

Their conclusion: This standalone and isolated approach does not take into account customer preferences or economic factors but only focuses on the physical stability of the grid. This can result in a negative user experience or rejection because of unpredictable charging process times, while at the same time, some customers might be willing to pay a higher energy price but have no option to do so. The "local decision approach" was therefore discarded, and the FNN control box should be used as intended; by being remotely controlled by grid operators only.

This thesis follows the same hybrid approach and addresses remote controlled load shifting in case study 2 5.

[**KFK22a**] and [KFK22b] define a "Framework for the Design and Automatic Deployment of Smart Grid Applications", that focuses on the lower level and grid operational aspects of the SG, to ensure their operational safety while providing measurement and control capabilities in the industrial scale.

This thesis can be seen as orthogonal to [KFK22a], since it relies on this operational safety of the existing power grid (e.g. the power limit of CPP per member).

2.3 Smart Grids and the Blockchain

[**CSET21**] presents a concept on how energy produced by a community PV system could be billed among members can be legally realized in Germany using blockchain in a GDPR compliant manner. This concept proposes rewarding individuals if they proactively adjust their consumption to a collective tariff, without coordination as opposed by this paper, but that

> "allows participating consumers, including tenants, to have an influence over the electricity price which gives them an incentive to adjust their load to support local load balancing, which leads to an optimization of the local energy system" [CSET21].

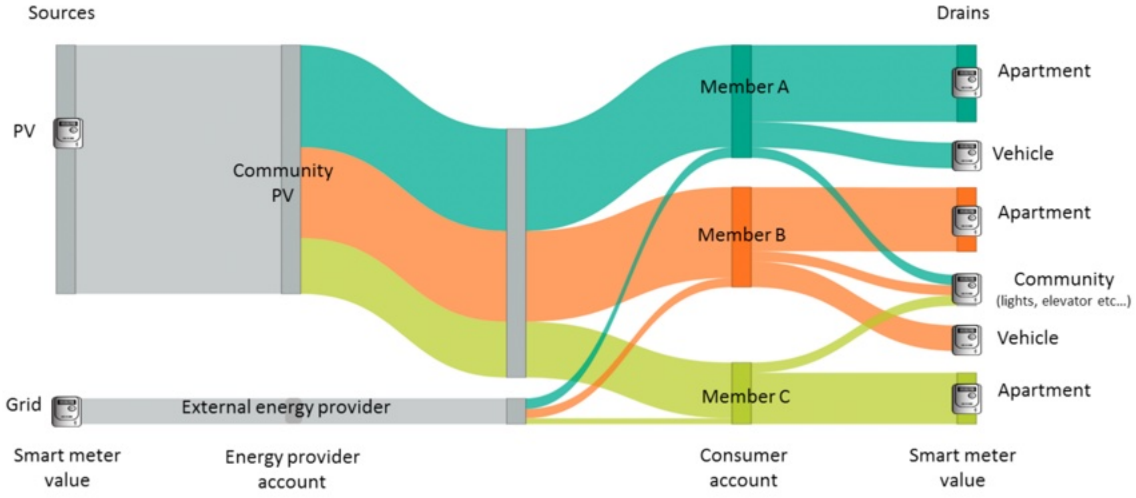


Figure 2.1: Members of a community consume energy from a shared community PV and external energy provider [CSET21].

With project **"Blockchain Grid"**, the AIT Austrian Institute of Technology published an article on "self-consumption optimization and peer-to-peer energy trading within the community" using simulation of peer-to-peer energy trading, in 2020. "First simulation models showed a potential of 10% total cost savings on average." [SZC$^+$20]

Project **"E-Cube"** [GTG$^+$19] describes a model for energy exchange across EC members with a central blockchain-based "energy bank" and a central optimization algorithm that maximized self-consumption.

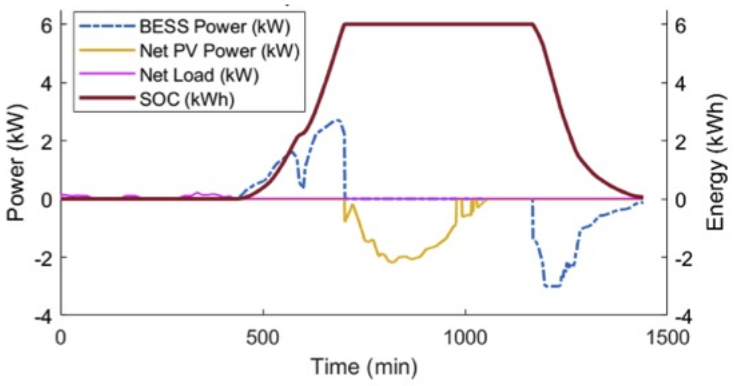


Figure 2.2: Graph shows how BESS power (ESR) is utilized to charge and discharge the ESR of a prosumer by the EMS algorithm in order to gain rewards [GTG$^+$19].

[SO21] claims to provide a "high-level layered framework that addresses the most important topics for the establishment of the SG", focusing on energy trading, as part of the so-called "Cooperative Energy Trading System (CENTS)". The article briefly explains and discusses certain high level components, e.g. "Prosumer", "Security", "Blockchain", "Load" in the context of SGs and Blockchain, without providing further details or references to their "framework".

2.4 EMS optimization algorithms

In "Microgrids EMS: A critical review on methods, solution and prospects" [ZEB18], the authors qualitatively reviewed a vast amount of different SG EMS algorithms and approaches, grouped into the following categories. The vast majority of the algorithms mentioned use the *centralized supervisor control architecture*, which is the same approach as proposed and used in this thesis.

EMS based on classical methods:

1. Linear and nonlinear programming methods (e.g. energy resource/cost optimization using linear programming (MILP)):
2. Dynamic programming and rule-based methods (using e.g. approximation of cost function, bellman equations, lookup tables)

EMS based on meta-heuristic approaches:

1. Genetic and swarm optimization (multiple objectives to be min/maximized like energy cost, GHG emissions, energy trade profit, battery aging etc. particle swarm optimization (PSO).)
2. Other metaheuristic approaches (EMS using algorithms like ant/bee colony optimization, gravitational search, bacterial foraging, adaptive firefly, crow search. Most of them with the objective of minimizing the operational cost of DERs or SG.)

EMS based on artificial intelligent methods:

1. Fuzzy logic and neural networks (multiple decision inputs e.g. SoC, grid frequency, energy/emission/maintenance cost, carbon emissions. use of reinforcement/Q learning with a goal of max. battery utilization)
2. Multi-agent systems (all the DERs are agents with their objectives. coordination between them in centralized or decentralized ways and possibly multiple hierarchies)
3. Other AI methods (e.g. uncertainty quantification, markov decision process (MDP), game theory approaches with leader-follower objectives, stackelberg equilibrium, pareto optimality, etc.)

Optimizing the efficiency of specifically energy communities (ECs) via an EMS is also a trending topic [uRFK+19] [WHK20] [RGH+20] [HBLN12]. [NZL18] among many others goes even further by also classifying consumer behaviors and the caused loads (which may or may not be timely shifted (e.g. EV charging)), in combination of weather data and energy market prices. All of those works basically train their EMS algorithms based on recorded datasets of multiple real sites in order to optimize for certain community strategies, e.g. cost or CO2 optimization.

[**RPD+14**] suggests in "Goal-Oriented Prosumer Community Groups for the Smart Grid" a goal programming model where "Prosumer Community-Groups" are formed and assigned different goal and subgoals. This thesis refers to those goals as service level objectives (SLOs). It furthermore lists a variety of such SLOs: sustainability, demand, cost, participation, resource, income. For example, the "demand SLO" is described as "a demand objective that fulfills at least the energy shortage of its own members".

[**SLK+15**] proposed the concept of "optimal electric energy management of a cooperative multi-microgrid community with sequentially coordinated operation" with the focus of energy trading with variable tariffs (hourly changing). Their architecture is very similar to this thesis, where a central EMS interacts with all local agent EMS's cooperatively to optimize the energy of the whole community (see figure 2.3), optimizing for cost savings. [SLK+15] does not provide the control framework itself, but rather a mathematical model. Furthermore, the central EMS notifies their members about "the global optimal electric energy information" nor does it further consider member behavior but evaluates the performance of the proposed/derived actions in theory.

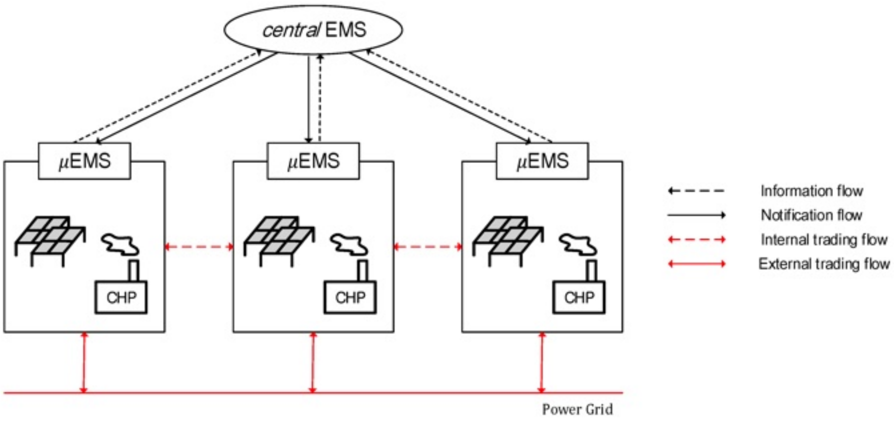


Figure 2.3: Information and energy flows in cooperative multi-microgrid community. EMS: energy management system; and CHP: combined heat and power. [SLK+15].

2.4.1 Conclusion and distinction

Although the presented thesis shows many intersections with the related work of these sections, its main focus is not on the performance evaluation of different EMS algorithms to model individual sites or communal strategies, but in evaluating a basic community EMS strategy under real-world settings that can be further optimized.

2.5 Connected vs. standalone (islanded) micro grids

[**ZEB18**] states that

> The main objectives of the energy management system are to optimize the operation, energy scheduling, and system reliability in both is- landed and grid-connected microgrids for sustainable development. Hence, microgrid energy management system is a multi-objective topic that deals with technical, economical, and environmental issues.

[**JXG13**] concludes that islanded / standalone MGs have no effect on the central grid nor EC due to their disconnect from the underlying physical grid.

This thesis / framework supports grid-connected MGs only, that are embedded within and connected to the public grid.

2.6 Community-focused energy management systems

[**MRMDP22**] presents a study on "collective self-consumption in energy communities, where participants within a given area can exchange and trade energy among themselves" that describes a rule-based central EMS, similar to this thesis.

The conclusion section of [**MRMDP22**] confirms the results of this thesis' case study 1 4.6, stating that "*the most satisfying results were obtained with an EMS that maximizes the self-sufficiency*".

Figure 2.4 shows an architecture that is similar to the framework's architecture of this thesis 3.1, depicting community members sharing energy between each other or the community or the central grid.

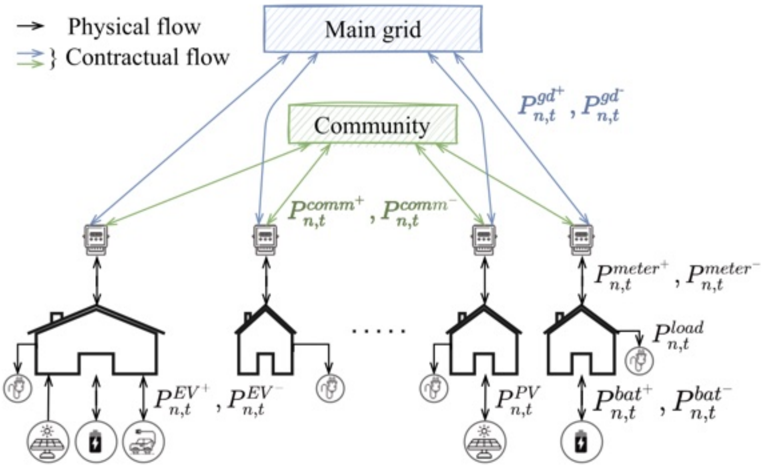


Figure 2.4: Typical energy community architecture according to [MRMDP22].

Figure 2.5 compares results (measured effect on the community level) of 4 different central EMS algorithms of a community with multiple members, very similar to this thesis.

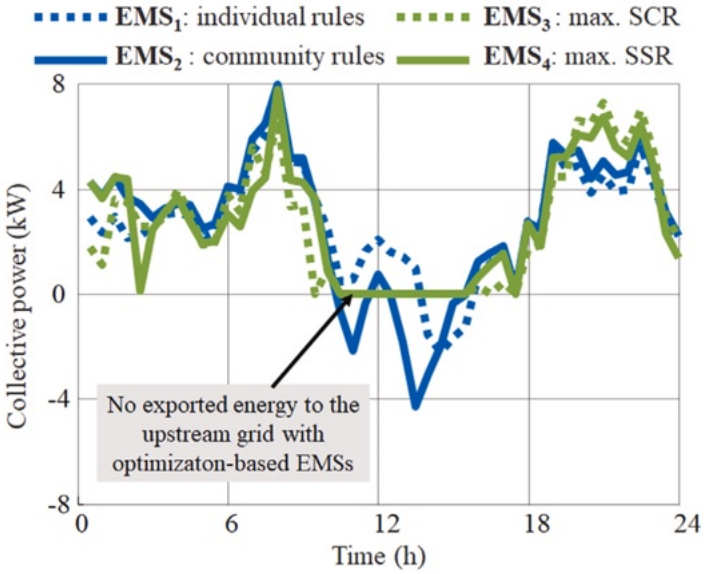


Figure 2.5: Physical power flow observed at the measurement meter at the community level at a sample day for different EMSs. [MRMDP22].

The EMS algorithm 2 "community rules" as provided by [MRMDP22] is similar to the SumZero strategy of this thesis. It takes individual member rules into account (e.g. battery power and SoC limits) and tries to balance energy excess/demand among the members. Other EMS strategies from [MRMDP22] could be implemented within the framework of this thesis to further improve the EC performance.

[**LR11**] states that "The agent based system facilitates both centralized co-ordination and local control as it operates in different levels as a hierarchy. This approach allows utilization of the strengths of both central and decentralized control systems and this is a possible candidate for future smart grid approaches."

It describes precisely what this thesis aims to achieve in practice throughout the following chapters.

[**Ju21**] proposes a "hierarchically coordinated energy management system (EMS) for a regional community comprising multiple small-scale microgrids" and "aims to minimize the total operational cost of the MG community and maximize the individual benefit of each MG simultaneously", which targets the same goal as this thesis. Figure 2.6 precisely describes the physical setup of ECs as in this thesis, except for the term "MG", which is assigned to individual sites in 2.6.

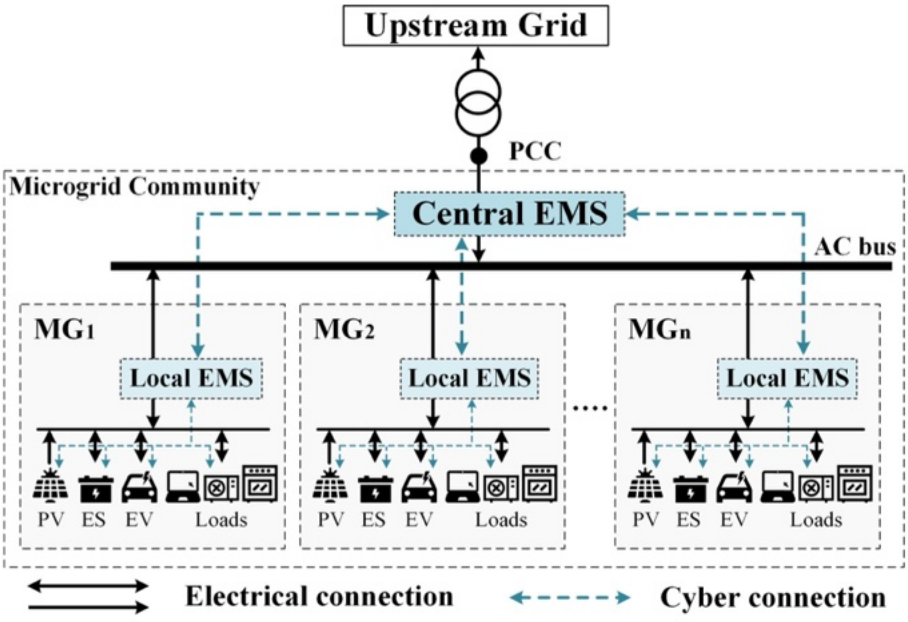


Figure 2.6: Network topology of a regional MG community. [Ju21].

The project uses the Gurobi Optimizer [gur] to first optimize an objection function given physical constraints for every site individually, and in a second stage to find the optimal EC setups and optimize the central EMS strategy (to minimize the central grid transaction cost). This could be an interesting extension of the strategy as a future work for the community EMS framework in this thesis.

This thesis differs mainly from [Ju21] by:

1. focusing on a cloud-based approach in a real-world setting, whereas [Ju21] uses MATLAB simulations (published source code: [1])

2. using a much larger set of energy data, site count, EC count, and evaluation/simulation window sizes

3. using a real-time optimization approach that iteratively collects data from individual sites and derives immediate actions for them, whereas for [Ju21] "the community-level EMS has no iterative information exchange with the HEMS" nor does it derive actions for members to control the power flow physically in order to optimize them in practice.

2.7 Further HEMS frameworks

OpenEMS [ope] founded 2018 by the OpenEMS Association e.V. has developed a Java-based EMS development platform that provides:

- a whole IoT stack (edge, backend, UI)

- integration of many energy devices via local communication protocols to read and write datapoints

- ability to use existing or develop custom rule-based EMS features for the specific site (operating on the datapoints)

- local simulation of energy devices and EMS features

OpenEMS enables site owners to model custom real-world EMS applications to control their energy devices with a vendor-agnostic approach.

---

[1] https://github.com/juchengquan/Hierarchically_Coordinated_Energy_Management_for_A_Regional_Multi-microgrid_Community

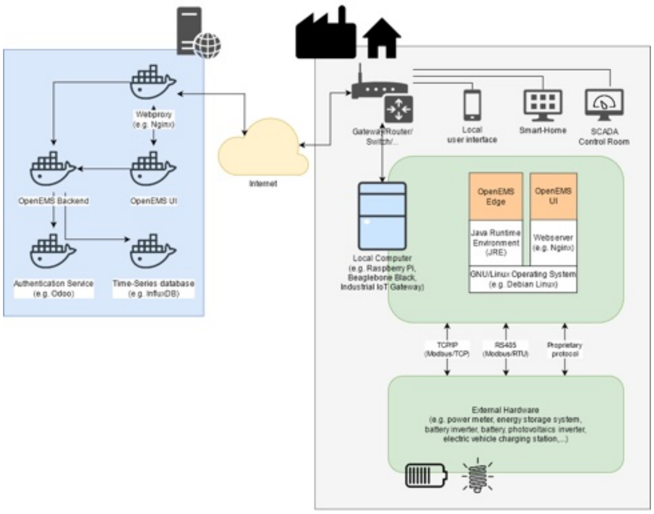


Figure 2.7: OpenEMS system architecture (for a single site) [ope].

Proprietary HEMS solutions:

Besides OpenEMS, numerous similar but proprietary HEMS frameworks are available: Examples are Reisenbauer Solutions [2], Siemens SIMATIC [3], Energis.Cloud [4], MobilityHouse ChargePilot [5].

Examples of proprietary EMS and IoE platforms:

- gridX [6] provides a HEMS using their edge gateway "gridBox" that connects to their "XENON" cloud.
- 1komma5grad [7] provides a HEMS using their edge gateway "Heartbeat" (based on gridX architecture).
- neoom [8] provides a HEMS using their edge gateway "BEAAM" that connects to their neoom CONNECT.

2.7.1 Conclusion and distinction

This thesis differs from OpenEMS and proprietary solutions by aiming to integrate and model e.g. an OpenEMS instance as an EC member and grouping multiple OpenEMS

---

[2] https://reisenbauer.solutions/

[3] https://www.siemens.com/at/de/produkte/automatisierung/industrie-software/automatisierungs-software/energiemanagement.html

[4] https://energis.cloud/en/products/platform/

[5] https://www.mobilityhouse.com/int_en/chargepilot

[6] https://de.gridx.ai/

[7] https://www.1komma5grad.com/

[8] https://www.neoom.com/

instances to ECs (as a concept and template only, not an actual bridge implementation yet). The presented framework therefore resides on a higher abstraction level, since it is capable of modeling ECs out of multiple, potentially different EMSs, as long as the requirements are met (see 3.5.1) using real-time data exchange between different third-party EMS frameworks. It moreover focuses on the practical feasibility and benefits of modeling and deploying SGs/ECs in Kubernetes as higher level ICT, that leverages existing on-site ICT on a lower level, and by providing an exemplary bridge in between those ecosystems.

At the time of writing and from the resources accessible online, none of the EMS frameworks mentioned provide such a mechanism of including 3rd party solutions out of the box. Bridging interfaces for those frameworks, based on the provided agent interface, could be implemented in future work.

CHAPTER 3

Contribution

3.1 Problem statement

As mentioned in the previous chapter in detail, research on the related work / state-of-the-art showed that the vast majority of the works were focusing either on theoretical aspects (e.g. the concept and building blocks of the smart grid, or performance of different EMS algorithms and their simulations, or on very specific, industrial smart grid problems.

In general, related works could be grouped into three categories:

1. Communication networks / ICT to provide the building blocks for SGs and EMS'.
2. Comparison of different approaches to smart grids in theory or simulation.
3. Specific EMS algorithms, either simulated or in very specific single-site smart grid settings.

At the time of writing, a gap has been identified: the lack of a framework that is not only able to build and simulate smart grids (energy communities) and evaluate their performance in theory, but that also attempts to provide implementation and interfaces that would allow realization of optimized communities in practice.

3.2 Introducing the SmartGrid Framework

This thesis proposes the **SmartGrid Framework** (figure 3.1) to model, deploy, and optimize energy communities in near real time, including software components and ICT for vendor-agnostic information exchange with physical EC members.

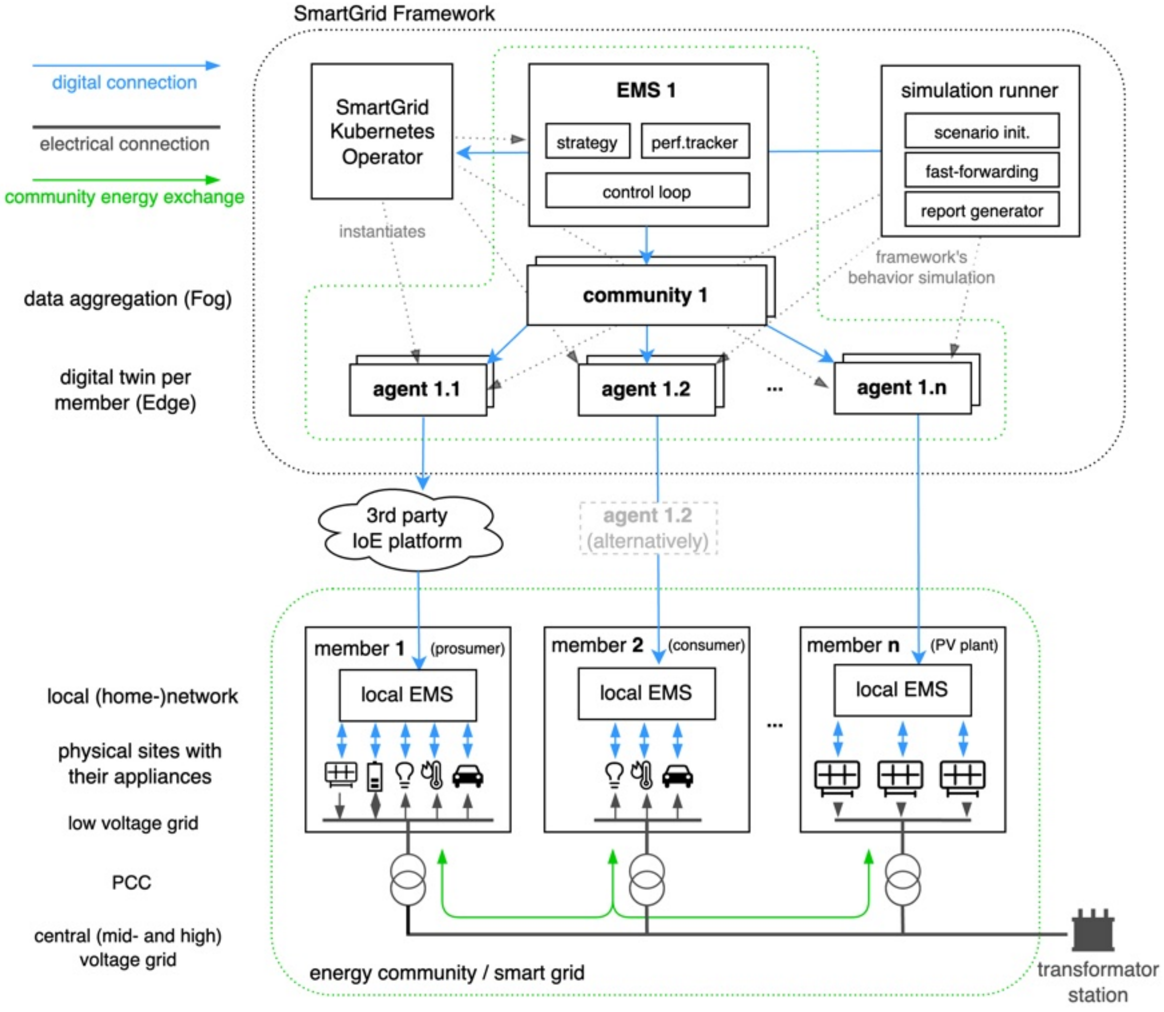


Figure 3.1: High-level framework architecture and contextual overview.

This framework is supposed to operate on a higher layer, on top of several existing physical sites. Optimizing energy in the community requires all members to allow for data exchange in near real-time (seconds to minutes):

All purely consuming members that join an EC, require a smart grid meter that provides momentary grid power data to their digital twin in the framework (read-only).

Prosumers with controllable DERs (e.g. PV strings, battery systems (ESR), EV chargers, heat pumps, etc.) furthermore require a HEMS that also provides production and consumption power, ESR capacity and state of charge, physical power limits, and optionally has the ability of being controlled, which implies a read-write data exchange between HEMS and its digital twin (agent). Such a HEMS is furthermore supposed to hide details and complexity of existing physical energy infrastructure (all devices, e.g. ESRs, multiple PV strings).

The communication channel between twin and local site can be established either through a third-party cloud API (e.g. neoom CONNECT) or by exposing an endpoint of the framework's agent that allows a HEMS / grid meter to directly connect to its digital twin. Since the agent interface is extensible, the implementation in real-world setting is up to the administrator's choice.

This mixed decentralized-centralized approach of connecting multiple HEMS to a central EMS ensures the continued and safe operation of all sites, even in the case of service interruption of central parts of the framework. Therefore, the HEMS' digital twin, as well as the digital representation of the EC state, are modeled as "Deployment" entities in Kubernetes, and are accessible through JSON/REST APIs by the central EMS deployment. This approach renders all parts of the community first-class citizens in the context of Kubernetes.

3.3 Components and deliverables

This framework as presented in 3.1 makes use of Kubernetes' concepts of operators and custom resource definitions (CRDs) for declarative creation of higher-level ICTs, as its fundamental building block to model and interconnect all virtual and physical components of a community together.

The whole framework provides or consists of:

1. A Kubernetes Operator implementation and CRDs to model all entities.
2. Vendor-agnostic scripts for automated builds and deployments of (not limited to) Minikube K8s clusters to (i) model the desired energy communities, (ii) their agents (digital twins of members), and (iii) a central energy management system (EMS) for the community.
3. Multiple predefined scenarios of EC compositions and their members.
4. Implementation of an agent that delivers data from prerecorded energy data files for their production, consumption, and grid meters in 15 minute resolution.
5. Agent implementation that models HEMS' for behavior simulation of changes that an external signal of the central EMS would cause.
6. Agent interface with the possibility to extend it for further connection to existing energy cloud platform APIs, for real-world execution.
7. Community implementation that attaches to all its agents to (i) derive the EC's momentary state, (ii) distribute actions, and thus (iii) aggregate all participating agents into a VPP / smart grid that is accessible to a central EMS.

8. Implementation of a central EMS with extensible strategies that runs the main control loop, evaluates the state of the VPP, derives the next actions, and tracks the state and KPIs over all time steps.
9. Implementation of EMS strategies: NoOp and SumZero.
10. A simulation runner to fast forward all the predefined scenarios and to create KPIs (json) and charts for evaluation purposes.
11. Production and consumption data of 24 real households over 7 days in 15 minute sampling resolution

3.4 SmartGrid Operator

The framework leverages the Operator SDK[1] of the K8s Operator Framework, which is "an open source toolkit to manage Kubernetes native applications, called Operators, in an effective, automated, and scalable way". The provided SmartGrid Operator, as depicted in figure 3.2, instantiates all the necessary parts of a community, consisting of deployments, pods, services, etc., as defined by the CRDs, including all its configuration (e.g. the communities, the agents with their configurations, the agent-to-community mappings, the strategy of the EMS). The operator transitions the system to the desired state by deploying and instantiating the entire cloud and fog ICT of one or multiple smart grids within a generic Kubernetes cluster, according to the CRDs. It further allows modifying existing communities (e.g. adding, removing, changing member assignments, or member connection losses) without further service interruption by just applying the updated manifests that are representing the community. The central EMS will automatically take into account the recent changes of the community and agent deployments during the next cycle of its EMS control loop.

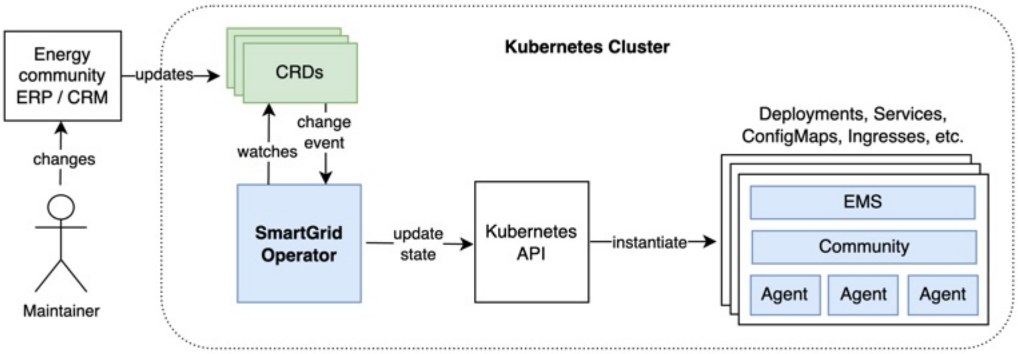


Figure 3.2: SmartGrid operator instantiating smart grid ICT in the cloud. The framework implements the blue elements.

---

[1] https://sdk.operatorframework.io

The following sections provide details about the first-class Kubernetes entities that are modeled as custom resource definitions for (i) energy communities, (ii) their agents and (iii) an energy management system (EMS). The framework does not further define how the CRDs are initially generated or maintained by the administrator, since this depends on the context the framework is used.

3.5 Agents

Power flow model

In this framework, every agent belongs to exactly one of those categories:

1. **Consumer**: member with ability to consume power from grid, no PV, no ESR.
2. **Prosumer**: Consumer with PV applicance, for self-consumption and grid export, without ESR.
3. **ESR**: Prosumer + battery storage. Optionally with controllable P_charge and $P\_discharge$.

The power flow of every member is modeled s.t. the sum of all the power is zero at every point in time, reflecting physical reality. Figure 3.3 depicts an "ESR" agent that is part of an EC.

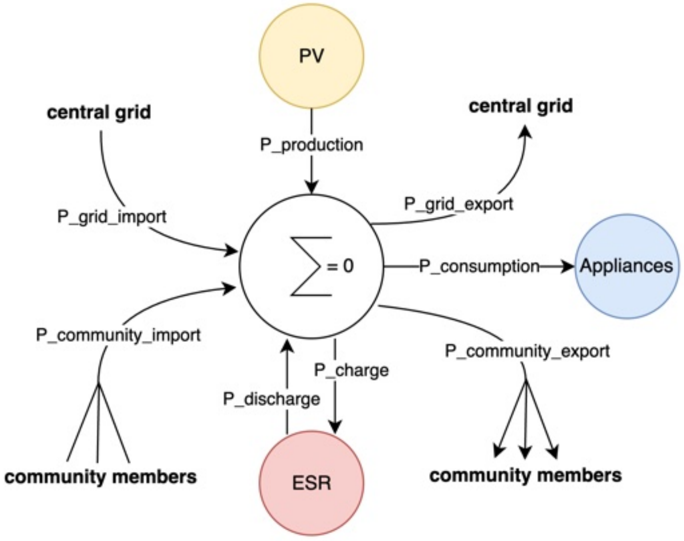


Figure 3.3: Model of a local household consisting of PV and ESR, and membership in an energy community.

The availability of certain P_* power inputs and outputs, pointing from and to the summing circle, varies depending on the agent's category and whether a membership to a community exists. All the different variants can be modeled by the framework. Additionally, further limits can be defined: ESR capacity, charging/discharging power limits, importing/exporting power limits.

Figure 3.4 depicts a "Producer" that is not part of an EC and without battery storage. This is modeled as a subset, compared to the "ESR with EC" 3.3.

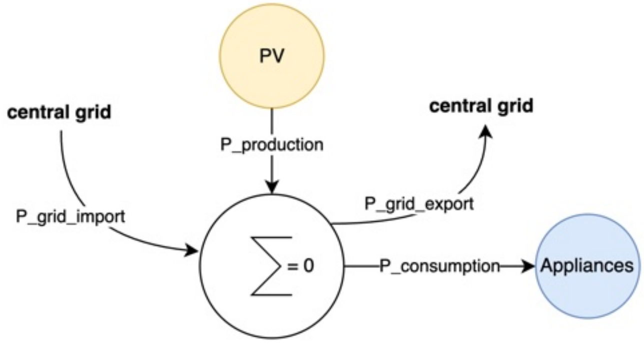


Figure 3.4: Model of a local household with PV only, but no EMS nor community membership.

3.5.1 Agent interface

Each deployed agent is required to provide the following input parameters to its attached energy community upon request.

All:

- Current grid meter import (relative energy [Wh] or average power [W])
- Current grid meter export (relative energy [Wh] or average power [W])
- Current consumption (relative energy [Wh] or average power [W])
- Grid import/export power limits [W]
- Optional: Energy price import [€]
- Optional: Energy price export [€]

Producers only:

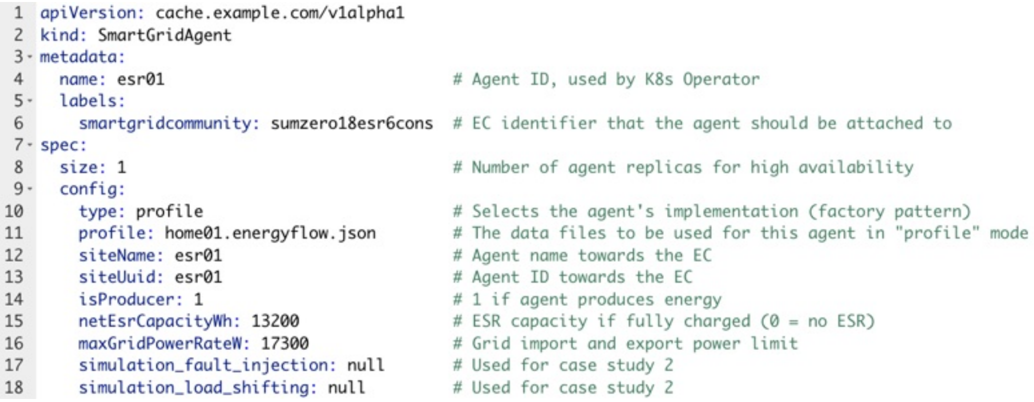
- Current PV production (relative energy [Wh] or average power [W])

ESRs only:

- Current ESR state of charge [%]
- Maximal available ESR energy [Wh]
- Optional: Minimal acceptable ESR energy [Wh]
- Optional: Charging/discharging power limits [W]
- Optional: ESR charging/discharging efficiency factors [%]

3.5.2 Agent CRD

Each member of an EC is represented by an agent CRD in the framework. All required settings for the specific agent are defined through this CRD 3.5, that is a YAML configuration file, with the config definitions deployed as ENV vars of the Agent deployment. As soon as applied to the K8s cluster, the SmartGrid operator detects the new CRD manifest and instantiates the agent deployment accordingly. Using the "smartgridcommunity" label, the operator automatically groups and registers the agent to the EC deployment (again, using ENV vars) as soon as the EC deployment exists within the same K8s namespace:



```yaml
apiVersion: cache.example.com/v1alpha1
kind: SmartGridAgent
metadata:
  name: esr01                             # Agent ID, used by K8s Operator
  labels:
    smartgridcommunity: sumzero18esr6cons # EC identifier that the agent should be attached to
spec:
  size: 1                                 # Number of agent replicas for high availability
  config:
    type: profile                         # Selects the agent's implementation (factory pattern)
    profile: home01.energyflow.json       # The data files to be used for this agent in "profile" mode
    siteName: esr01                       # Agent name towards the EC
    siteUuid: esr01                       # Agent ID towards the EC
    isProducer: 1                         # 1 if agent produces energy
    netEsrCapacityWh: 13200               # ESR capacity if fully charged (0 = no ESR)
    maxGridPowerRateW: 17300              # Grid import and export power limit
    simulation_fault_injection: null      # Used for case study 2
    simulation_load_shifting: null        # Used for case study 2
```

Figure 3.5: Example CRD to instantiate an agent called esr01, attached to EC called sumzero18esr6.

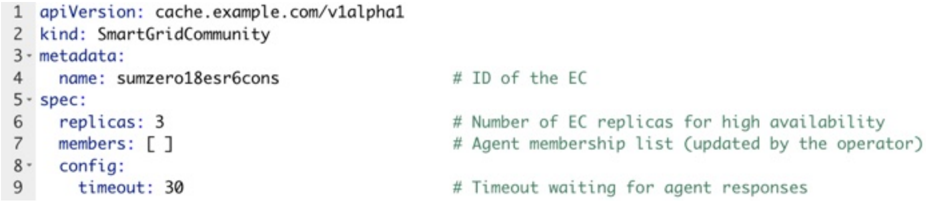
The agent is supposed to hide all complexity of the attached physical participant and to expose the generic JSON/REST API towards the community deployment. Furthermore, it acts as a proxy between the HEMS, translating all messages between the two systems, thus either caching or loading the local state on demand. On creation/restart, the agent as follower is required to obtain the latest state of the HEMS (leader) again.

Framework administrators can extend the Python implementation and CRD with additional agent types and settings as required.

3.6 Energy Communities

3.6.1 Community CRD

The community is the stateless aggregator and distributor of all data between agents and the EMS. It maintains a list of all agents under its control (updated by the SmartGrid Operator) and is only engaged by incoming API requests from a higher-level instance (EMS). Any changes recreate the community using the K8s rolling update mechanism with multiple replicas, allowing for zero-downtime changes during operation. Figure 3.6 depicts the EC YAML definition, named "sumzero18esr16", that will be deployed as 3 community pods (for HA and load distribution) by the operator, with zero members (agents) initially. The agents will register themselves to the EC, as soon as deployed.



```yaml
apiVersion: cache.example.com/v1alpha1
kind: SmartGridCommunity
metadata:
  name: sumzero18esr6cons                 # ID of the EC
spec:
  replicas: 3                             # Number of EC replicas for high availability
  members: [ ]                            # Agent membership list (updated by the operator)
  config:
    timeout: 30                           # Timeout waiting for agent responses
```

Figure 3.6: Example CRD to instantiate a community called sumzero18esr6.

3.6.2 Community API

Every invocation, triggered by the EMS, is blocking and invokes N further blocking calls, one for every member. As soon as all sub-requests are received (or timed out), the members states are aggregated into a single state representing a VPP, using map-reduce.

For example:

- Collective momentary PV power
- Collective momentary grid imports and export
- Collective momentary consumption power
- Collective average grid frequency
- Collective available ESR state of charge
- Collective max. ESR charge/discharge power
- Availability ratio of the members

Its JSON/REST API provides the following two routes:

- GET / Obtains the most resent state from all members and returns the aggregated response

- PUT / Distributes actions given by the EMS to the respective members for active target power control 4.4.

3.6.3 Large multi-level communities

To support large amounts (thousands) of agents participating in a single EC, the architectural design enables hierarchical chaining and multilevel aggregation of ECs. Although this has not yet been verified or tested, a concept is proposed below.

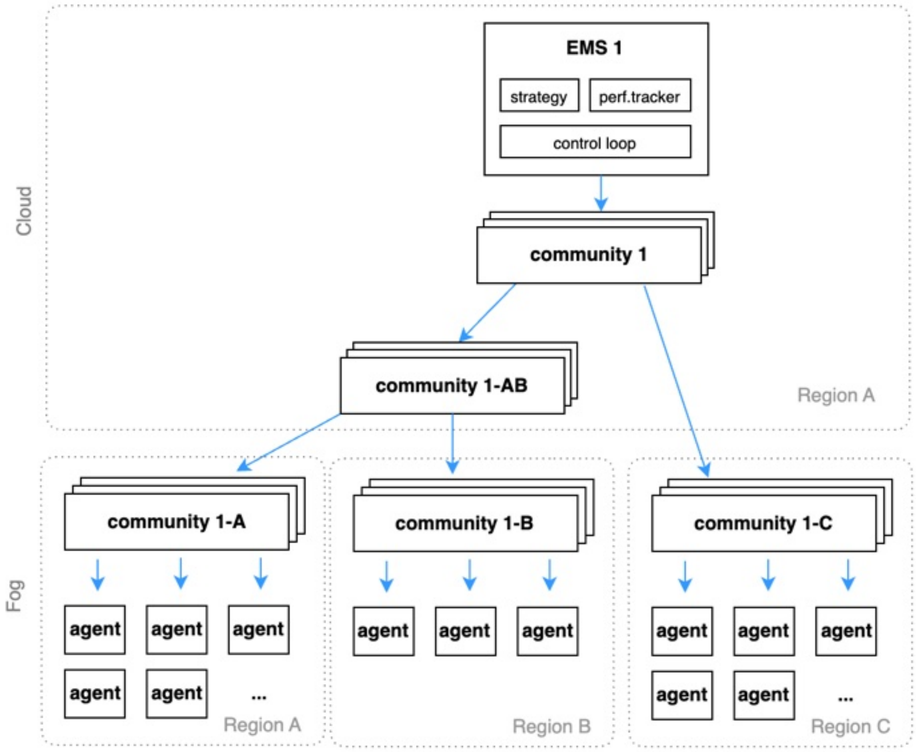


Figure 3.7: Hierarchical multi-level energy communities spanning over 3 regions.

Figure 3.7 exemplary shows a large scale deployment rolled out by the operator, spanning over multiple regions, on a multi-node/region kubernetes cluster, e.g. using K8s labels.

Regions are intended to group geographically closer agents to reduce network latency between subsets of communities and their agents. Each additional level in the community hierarchy increases the expected maximum latency. A default timeout of 30 seconds for all API requests would in this case (n=4) result in a maximum execution time of 120 seconds for every action triggered by the EMS. The summing up of latency due to the timeouts is the main bottleneck, thus hierarchy should be kept as flat as possible for large-sized communities.

3.7 Central EMS

Every EC is monitored and controlled by a central EMS instance, which triggers measurements and derives actions for their assigned community, using a (configurable) 15-minute control loop cycle time. The central EMS instance runs a control loop to "facilitate load/generation shedding within the microgrid to meet the net import/export power in grid connected mode.", as also described in [LR11]. An EMS instance is usually stateful and will in that case be the single point of failure of the framework, due to its KPI tracking and the support for further, often stateful EMS strategies that might be operating on historical and prospective data.

3.7.1 EMS Control Loop

For every time step (15-minute interval), the following four operations are executed as depicted in Figure 3.8, completing one iteration of the control loop:

1. Gathering the momentary state

A GET request is sent from the EMS to its assigned community as defined in the EMS CRD. The community processes the request, distributes it to all its registered members to obtain their latest state, and returns all available information for the current state per member and as community aggregate.

2. Execution of the strategy

The selected EMS strategy (according to the CRD) is executed (factory pattern) on the state information obtained from the whole community and its agents. The SumZero strategy is explained in detail in the following sections.

3. Propagating the derived actions

The derived actions (if any) returned after strategy execution are propagated to the community as POST request. Similarly to step 1, the community further splits the actions and notifies every affected agent individually. This invocation also returns the post-action state of every agent and the whole community back to the EMS.

4. Tracking results and updating KPIs

The immediate post-agent state of the whole community is tracked by the EMS for this time step. The KPIs are updated as a result of all preceding time steps.

3.7.2 Exemplary EMS CRD

Figure 3.9 depicts the definition of an EMS instance with $strategy = SumZero$. Due to $type = simulation$, the EMS disables its local control loop interval timer but operates

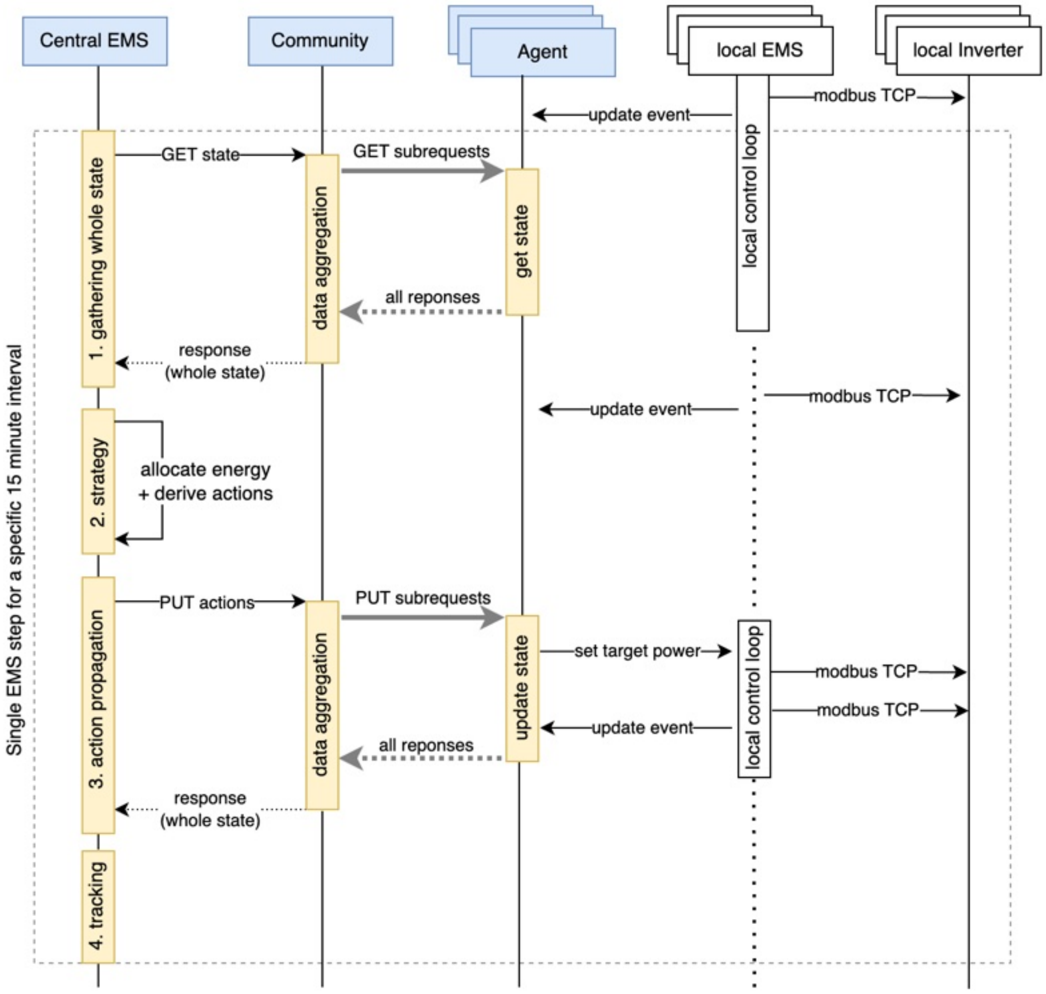


Figure 3.8: Sequence diagram of all the actions initiated in a single cycle of the EMS' control loop of one EC.

on external triggers by the Simulation Runner only. This allows to fast-forward through a whole week of data, by passing the time slot via API parameter.

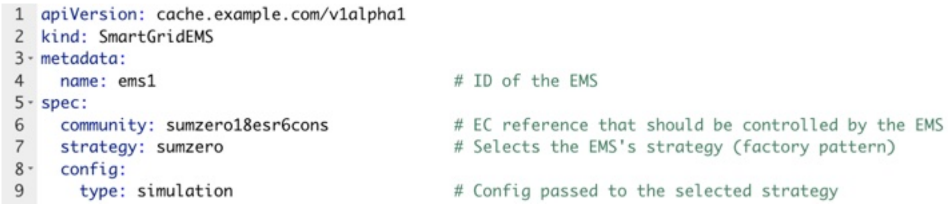

```yaml
apiVersion: cache.example.com/v1alpha1
kind: SmartGridEMS
metadata:
  name: ems1                          # ID of the EMS
spec:
  community: sumzero18esr6cons        # EC reference that should be controlled by the EMS
  strategy: sumzero                   # Selects the EMS's strategy (factory pattern)
  config:
    type: simulation                  # Config passed to the selected strategy
```

Figure 3.9: Example CRD to instantiate the central EMS with SumZero strategy, for the EC sumzero18esr6cons.

3.7.3 EMS strategy: SumZero

This algorithm / strategy is one of the central deliveries of this thesis and is used and evaluated throughout the 3 case studies. It is explained in detail in figure 3.10.

For SumZero (as well as for NoOp), the EMS instance first gathers current power demand within the community as well as the potential supply powers of the controllable ESRs with in the EC. The difference between demand and supply is calculated at each step. The resulting excess or energy deficit is split up and assigned to all available and controllable ESR members.

Every step within this strategy is executed in 15-min intervals and implemented by applying Monte Carlo sampling in up to 1000 rounds per time step. The member's SoC is chosen as a probability function for the assignment of 1/1000 parts of total power difference that has to be covered. On successful assignment of the actions for the chosen agent, its change in SoC is simulated too, by integrating the power over the duration of the time step. At the end of the round, the probabilities are updated due to the simulated change in SoC, to ensure that the total energy difference is fairly distributed over the eligible agents according to their SoC by respecting their ESR constraints (e.g. power limits).

3.7.4 Customizing and implementing further EMS strategies

The framework provides the ability to implement further strategies and easily assign/replace them to an EC. An EC can only have one EMS, as multiple EMS would interfere with each other. Many EMS strategies have already been proposed for smart grids; some of them have been referenced in Section 2.4 and could be implemented using this framework (limited by the information provided by the EC members and the set of actions).

[PBSM15], [SWL13] and [AAPG⁺17] describe further EMS algorithms that make use of state of charge (SoC) as an input variable in their EMS control loop.

A future improvement of the SumZero EMS strategy could be to incorporate past observations and decisions for the next actions, since the EMS tracks the state over time anyway. This could be done by using NNs, for example, as proposed in [AAPG⁺17], to maintain the SoC level close to 50%, optimizing the ESR lifetime.

Figure 3.10: Algorithm of SumZero EMS strategy

```
Input: Aggregated community data $\mathbf{C}$. Individual member data $\mathbf{C}[m]$.
Output: Actions per member $\mathbf{A}[m]$.

$remainingSteps = 1000$                                  ▷ Limit execution time
$p_{missing} = \mathbf{C}.gridPowerRate$                 ▷ Amount of missing power in the community
$p_{step} = 2 * \frac{p_{missing}}{remainingSteps}$      ▷ Power allocated per step among the members
$\mathbf{A}[] = \{0\}$                                   ▷ Initialize empty list of actions

// Choose next ESR member to cover $p_{missing}$ using Monte Carlo sampling
$\mathbf{S}$ = subset of $\mathbf{C}$ where $\mathbf{C}$[m] is and eligible ESR able to cover $p_{missing}$
$\overrightarrow{S} = normalize(\mathbf{S}[m].stateOfCharge \ \forall m \in \mathbf{S})$

while $p_{missing} < 0$ and $remainingStep-- < 0$ do
    $m'$ = take random sample out of $\overrightarrow{S}$

    // Ensure power assignment to $m'$ would not exceed its power limits
    if
      $abs(\mathbf{C}[m'].batterPowerRate) + p_{step} < \mathbf{C}[m'].maxBatteryDischargeRate$ &
      $abs(\mathbf{C}[m'].gridPowerRate) + p_{step} < \mathbf{C}[m'].maxGridExportPowerRate$
      then
        // Success - apply inverse target power to compensate $p_{step}$
        $\mathbf{A}[m'] = \mathbf{A}[m'] - p_{step}$
        $p_{missing} = p_{missing} - p_{step}$

        // Update m', assuming action gets actually applied
        $\mathbf{C}[m'].batterPowerRate = \mathbf{C}[m'].batterPowerRate - p_{step}$
        $\mathbf{C}[m'].gridPowerRate = \mathbf{C}[m'].gridPowerRate - p_{step}$
    end
    else
        $p_{step} = p_{step} * 0.95$          ▷ Reduce amount of assigned power for future steps
        $\mathbf{S}[m'] = \mathbf{S}[m'] * 0.5$      ▷ Halve probability of chosing m' in future steps
    end
end
return A
```

3.8 Exemplary complete CRD

An large number of communities with their agents and it's EMS can be deployed within the framework, running in parallel. The examplary figure 3.11 shows the SmartGrid-Community *2esr1con* with 3 SmartGridAgents to be initially deployed by the framework operator and continuously controlled by SmartGridEMS (SumZero strategy).

```yaml
### Community + EMS ###
kind: SmartGridCommunity
apiVersion: cache.example.com/v1alpha1
metadata:
  name: 2esr1con
---
kind: SmartGridEMS
apiVersion: cache.example.com/v1alpha1
metadata:
  name: myEMS
spec:
  community: 2esr1con
  strategy: sumzero
---
### The 3 agents (members) ###
kind: SmartGridAgent
apiVersion: cache.example.com/v1alpha1
metadata:
  name: esr1
  labels:
    smartgridcommunity: 2esr1con
spec:
  config:
    type: profile
    profile: home01.energyflow.json
    isProducer: 1
    netEsrCapacityWh: 13200
    maxGridPowerRateW: 17300
---
kind: SmartGridAgent
apiVersion: cache.example.com/v1alpha1
metadata:
  name: esr2
  labels:
    smartgridcommunity: 2esr1con
spec:
  config:
    type: profile
    profile: home02.energyflow.json
    isProducer: 1
    netEsrCapacityWh: 18000
    maxGridPowerRateW: 20000
---
kind: SmartGridAgent
apiVersion: cache.example.com/v1alpha1
metadata:
  name: cons3
  labels:
    smartgridcommunity: 2esr1con
spec:
  config:
    type: profile
    profile: home03.energyflow.json
    isProducer: 0
    netEsrCapacityWh: 0
    maxGridPowerRateW: 16000
```

Figure 3.11: The full YAML custom resource definitions, declaring all entities to model all components of the EC.

3.9 Framework and simulation setup

3.9.1 Local setup

To set up a local development and simulation environment (in this case, MacOS) from the provided source [9], the following steps are required.

- $ brew install operator-sdk go kind ansible jq
- Install kind as described at https://kind.sigs.k8s.io
- $ ./create-kind-cluster.sh

3.9.2 Local simulation execution

The run.sh script in the main folder automates the building, deployment, execution, and data collection of all existing scenarios (using the run-scenario.sh script in the scenario folder), which iterates over all scenarios and triggers the following steps:

Build steps:

1. The smart grid operator is build and deployed into the kind cluster
2. The 3 smart grid docker images are built and pushed to the local registry

Deploy steps:

1. Existing deployments in the kind cluster are deleted
2. A scenario (energy community) is deployed to the kind cluster
3. Waiting loop until scenario is ready

Run steps:

1. Python app is executed locally, which iterates over all of the given scenario to fast-forward the EMS and track every step's result
2. KPIs and detailed results are obtained from the JSON response in total and per step and stored in-memory
3. Charts are plotted and PNGs and final KPIs are written into a JSON file

Evaluation steps:

1. The KPIs of the different scenarios are converted to a CSV and imported into Excel for further comparison and visualization.

3.10 Simulation vs. real-world differences

For simulation purposes, which are further evaluated in the respective section, the whole framework is incorporated as if it were run in a real-world setting, but with the following differences and limitations.

3.10.1 Agent

For simulation purposes, every agent is deployed with mode=profile. On agent deployment, the "profile" interface implementation is selected: this mode loads the agents' dedicated JSON file that contains the agent's consumption and (if any) PV production values in watts, for every time slot (15min interval throughout the evaluation of this thesis). All other values of the agent power flow 3.3 are simulated using the behavior-simulating implementation provided for this mode. The battery (if any) charging behavior is simulated as well, the SoC state is updated, and physical limits are considered.

The agent component supports the implementation of further bridges/connectors using the factory pattern. In practice, another mode/implementation would be used that loads a thin connector implementation s.t. API calls from community to the agent are forwarded to a third-party "internet of energy" (IoE) platform, e.g. neoom CONNECT.

3.10.2 Community

The community component acts as a bridge between the EMS and all agents and does not introduce further differences between simulation and real-world execution.

3.10.3 Simulation runner

For a fast-forwardable simulation, a standalone python app called "Simulation runner" was created that communicates directly with the EMS. For simulation and evaluation purposes, it executes every single the time step individually by sending it as a request to the EMS, instead of using the EMS' internal control loop interval timer. The EMS then executes the single time step (control loop run) as usual and communicates with the community, and the community with every of its agent. Due to mode=profile, all agents update their consumption and production values from their JSON file in the given time slot and simulate their battery behavior and power flow based on those values. All JSON files are provided with 15-minute sample measurements; thus, only discrete 15min values are used (xx:00, xx:15, xx:30, xx:45).

In every EMS step, $2 * (1 + membercount)$ HTTP requests are sent in total, resulting in $1440 * 2 * (1 + 100) = \ 300k$ requests per day, given a 1min sample interval. Two requests are sent in every EMS step (community query and update) each with a specific round-trip time (EMS ⇔ community ⇔ members ⇔ HEMS ⇔ physical inverters and meters). In a real-world situation, these delays will increase the error/offset in the SumZero EMS strategy, resulting in potential performance reduction. This is further examined in case study 2.

CHAPTER 4

Case study 1: Performance comparison of energy communities

This case study places 24 households in different compositions of energy communities and simulates their behavior as if they were forming a physical smart grid. As a result, energy is (virtually) transferred between households on top of the physical power grid (simulated). This affects the individual's ESRs and energy bills, and is tracked by the framework as well. In the real world, typical ECs consist of consumers and producers/prosumers, with their energy produced being (in the best case) consumed within the neighborhood, thus reducing the load on the central grid and having economic benefits for all members. The SmartGrid Framework simulates and fast-forwards all the data exchanges between all household's HEMSes and the framework itself, and presents the performance of the whole community as well as for each member (agent) individually, for every executed scenario.

The results of all scenarios (compositions of energy communities) and EMS strategies will be compared against each other using defined KPIs. Finally, the results will be discussed and conclusions and recommendations for EC sizes and compositions will be presented throughout the following case study.

All of the following case studies have been executed and evaluated using the implemented SmartGrid framework and its capabilities to schedule real-world energy communities, as well as fast-forwarding scenarios and agent simulations on Kubernetes.

4.1 Energy data of households

This and the following case studies and their simulations have been conducted using real-world energy data from 24 households, measured and provided by neoom [1] using their internet of energy platform "neoom CONNECT".

---

[1] https://www.neoom.com/

lists the 24 households with PV appliances and ESR and their 7-day data set in detail, containing individual power data for production and consumption in 15-minute averages.

Based on this data set, multiple different EC compositions were simulated over the same 7-day time span starting 2022-08-29 00:00:00 to 2022-09-04 24:00:00. The time zone throughout this thesis is UTC. The local time zone of all measurements is CEST (+02:00).

For simplicity, pure consumers are derived from the same set of sites, but stripped of their PV and ESR capabilities during the simulation, s.t. only their household consumption profile is taken into account, that is covered solely by the grid and thus equal to it.

4.2 Real world challenges omitted in case study 1

This section briefly discusses the main differences between all simulations executed by this framework and their limitations compared to real-world execution runs.

The evaluation of the case studies was completed using real-world consumption and production data, but the EMS actions on the different EC compositions, including performance measurements, were simulated.

The simulation setup is not affected by interferences that usually occur in real-world distributed systems and moreover systems with edge devices.

The following simplifications and assumptions during the simulation might therefore affect real-world performance and results.

- The time gap between querying the community and sending the update is zero (in real-world: 2-10 sec)
- The update applies the desired actions immediately to all member's EMS and physical devices (in real-world: 5-30 sec)
- The results are immediately propagated back as result of the actions, by all members (in real-world: 5-30 sec)
- The wear-out cost of charging/discharging ESRs resulting from central EMS actions, is not taken into account
- Hidden power flows, that cancel each other and are thus not measured, might happen during the sample interval.
- For all ESR sites, the netEsrCapacityWh has been defined as high as 90% and is thus available to be controlled by the central EMS with the SumZero strategy.

4.2.1 Exemplary single consumer 24h profile

The following chart is used for introductory purposes and is, like all the time series based charts, generated by the thesis's simulation runner (EMS strategy="individual").

The household depicted in figure 4.1 covers all its energy from the central grid (overlapping lines). All dashed lines are at 0% because there is no ESR nor PV production.

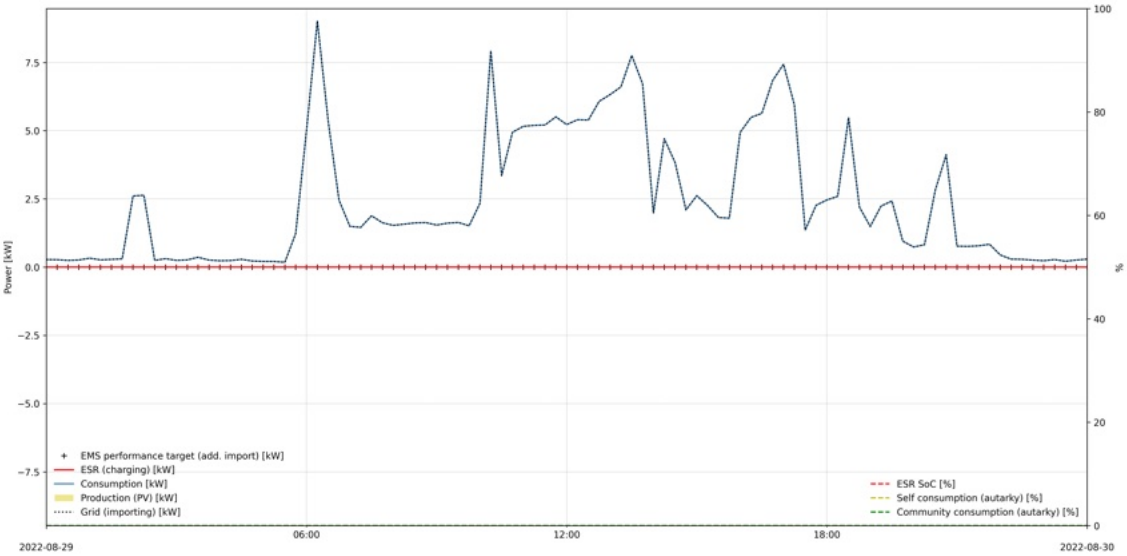


Figure 4.1: Power consumption in 24h interval of a pure consumer

4.2.2 Exemplary single ESR producer 24h profile (1)

At this site in figure 4.2, the sunrise can be observed at 5:00 AM and the sunset at 5:00 PM. From 6:00 AM, production is higher than consumption, so the battery is fully charged at 8:30 AM. The excess is then fed into the grid (dotted line). Exports into the power grid are defined and depicted as negative values. At 1:00 PM, the peak in consumption is fully covered by the PV production and the grid export is reduced accordingly. At 5:30 PM, after sundown, the battery fully covers a short consumption peak, as depicted by the mirrored red charging line. The PV production is a standard distribution with its center at 11:00 AM, which indicates a south siding PV installation. At this day, only a very low amount of energy was imported from the central grid, resulting in a high autarky of about 95%.

4.2.3 Exemplary single ESR producer 24h profile (2)

The site in figure 4.3 differs from figure 4.2 in the following ways:

1. From 4 to 6 AM, a consumption peak of 11 kW (very likely an EV charging process) cannot be covered by the battery (probably due to a defined blackout reserve of

Figure 4.2: Power flow in 24h interval of a household with ESR (with PV)

20%) and is thus fully covered by the grid. This is an excellent use case for an energy community with EMS, that can be precisely covered by another sufficiently charged member of the community, to prevent any central grid demand in this case.

2. The PV production peaks at 8:00 AM and then decreases drastically until 2:00 PM. This indicates an east-facing PV installation.
3. The autarky (cumulative average) reaches only 38% at the end of the day, due to the large amount of EV energy covered by the grid, compared to the remaining 62% of energy consumption and battery charging, which were both self-consumed.

4.3 Key Performance Indicators

The following three KPIs have been chosen as main indicators to measure the performance of energy communities. Every EC consists of a different ESR-to-consumer ratio, from the dataset of the 24 households, over the whole time horizon. All variables used in the following formulas are defined in section 1.5.

4.3.1 Autarky (self-sufficiency) ratio

Autarky is the total amount of energy consumed by all members of the community, which is covered by the member's production or ESR itself, or covered by any other community member(s) sharing energy in the same 15-minute time window.

The autarky (self-sufficiency ratio) A^m of a single member over the whole time horizon is defined as:

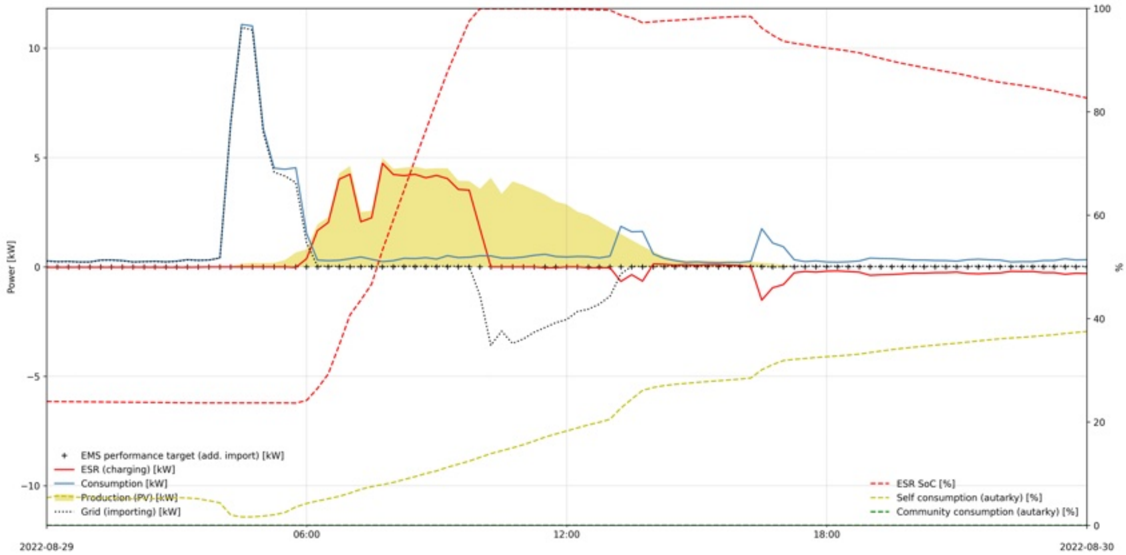


Figure 4.3: Power flow in 24h interval of another household with ESR (with PV)

$$A^m = 1 - \frac{E^m_{Gim}}{E^m_C} \tag{4.1}$$

The autarky $\underline{A}$ of the whole EC over the whole time horizon is defined as:

$$\underline{A} = 1 - \frac{\sum^{m \in EC} E^m_{Gim}}{\sum^{m \in EC} E^m_C} \tag{4.2}$$

Explanation:

- On $\underline{A} = 100\%$, no energy is consumed from the central grid; therefore, the community is not affected by the changes in the import price of the grid. This implies at least equilibrium / energy coverage or excess energy in the EC.
- On $\underline{A} = 0\%$, all members are pure consumers and therefore all energy demands are solely covered by the central grid.
- A high autarky implies low central grid imports.
- Community-to-grid excess energy E^m_{Gex} does not affect $\underline{A}$, as autarky refers only to energy imports / consumption.

4.3.2 Equilibrium ratio

Equilibrium describes the independence of the community from imports **and** exports of energy between the central grid, for every $t \in T$. Therefore, the grid power rate of each member $m \in EC$ is merged (summed up) s.t. the whole EC is considered a virtual power plant (VPP).

The (momentary) equilibrium ratio $\underline{Q}^t$ of the EC in time step t is defined as:

$$\underline{Q}^t = \left\{ \begin{array}{ll} \frac{\underline{p}^t_{Gim} - \underline{p}^t_{G}}{\underline{p}^t_{Gim}}, & \text{for } \underline{p}^t_{Gim} > \underline{p}^t_{Gex} \\ \frac{\underline{p}^t_{Gex} + \underline{p}^t_{G}}{\underline{p}^t_{Gex}}, & \text{for } \underline{p}^t_{Gim} < \underline{p}^t_{Gex} \\ 1, & \text{for } \underline{p}^t_{Gim} = \underline{p}^t_{Gex} \end{array} \right\} \tag{4.3}$$

The (average) equilibrium ratio $\underline{Q}$ of the EC over the whole horizon is defined as:

$$\underline{Q} = \frac{1}{T} \sum_{t \in T} \underline{Q}^t \tag{4.4}$$

Explanation:

- On $\underline{Q} = 100\%$, the central grid is not affected by the EC, so imports are equal to exports for all members $m \in EC$ and for all $t \in T$. The community can thus be seen as physically disconnected from the central grid and is operating in full island mode.
- On $\underline{Q} = 0\%$, all members of the community operate mutually exclusive, thus no energy is transferred between any of the members at all, in none of the time steps.

4.3.3 Cost/Reward

Cost is the total sum of all energy costs, occurring in all members $m \in EC$ and all $t \in T$, based on the prices of energy import and export for exchanges between the central grid $c^{t,m}_{im,ex}$ and the community $\underline{c}^{t,m}_{im,ex}$.

Throughout all simulations and performance evaluations, the following constant energy tariffs were used:

- $c_{ex} = -10 \frac{cent}{kWh}$ cost for the energy sold by any member into central grid.
- $c_{im} = 30 \frac{cent}{kWh}$ cost for the energy consumed by any member from the central grid.
- $\underline{c}_{ex} = -18 \frac{cent}{kWh}$ for exports and $\underline{c}_{im} = 20 \frac{cent}{kWh}$ for imports of members to/from their energy community (applicable only to the NoOp and SumZero strategies).

The total reward R^m for member m over the whole time horizon is defined as: (4.5)

$$
\begin{aligned}
R^m = \sum_{t \in T} \Big[\;& \underline{c}^{t,m}_{im} * p^{t,m}_{Gim} * Q^t_{im} \\
& + c^{t,m}_{im} * p^{t,m}_{Gim} * (1 - Q^t_{im}) \\
& + \underline{c}^{t,m}_{ex} * p^{t,m}_{Gex} * Q^t_{ex} \\
& + c^{t,m}_{ex} * p^{t,m}_{Gex} * (1 - Q^t_{ex}) \Big]
\end{aligned}
$$

where Q^t_{im} is the ratio of community-to-grid coverage, considering imports only:

$$
Q^t_{im} = \left\{ \begin{matrix} 1, & \text{for } \underline{p}^t_G \leq 0 \\ \frac{\underline{p}^t_{Gim} - \underline{p}^t_G}{\underline{p}^t_{Gim}}, & \text{otherwise} \end{matrix} \right\} \tag{4.6}
$$

and Q^t_{ex} the ratio of community-to-grid coverage, considering exports only:

$$
Q^t_{ex} = \left\{ \begin{matrix} 1, & \text{for } \underline{p}^t_G \geq 0 \\ \frac{\underline{p}^t_{Gex} + \underline{p}^t_G}{\underline{p}^t_{Gex}}, & \text{otherwise} \end{matrix} \right\} \tag{4.7}
$$

Examples:

- In time steps with $Q^t_{im} = 100\%$, no energy is imported from the central grid. Therefore, all imports by members (if any) are provided fully by other members of the community. This implies VPP equilibrium or overproduction.
- In time steps with $Q^t_{im} = 0\%$, all imports by members are fully covered by the central grid. This implies zero exports within the VPP.
- In time steps with $Q^t_{ex} = 100\%$, no energy is exported to the central grid. Therefore, all exports of members (if any) are fully consumed by other members. This further implies VPP equilibrium or underproduction.
- In time steps with $Q^t_{ex} = 0\%$, all exports of members are fully consumed by the central grid. This implies that none of the members imports any energy.
- On a ratio of 50%, imports/exports are equally proportional to the central grid and the community.

4.4 Active target power control

The "active target power control" feature of this framework is the mechanism to build smart communities where controllable ESRs can work together to intelligently equalize the community's excess or demand. It enables the central EMS to control the charging/discharging powers of individual (controllable) member ESRs, for the sake of importing/exporting from/to the grid during the time step or interval $[t, t+1]$.

Controllable ESRs allow their energy imports/exports to be controlled by the framework's EMS, e.g. by the SumZero strategy. This is implemented in the framework using an additional power parameter value $\Delta p_{target}^{m,t}$ [W], that is to be added to the agent's current import power (s.t. the battery is charged $\Delta p_{target}^{m,t} > 0$) or export power (s.t. battery is discharged on $\Delta p_{target}^{m,t} < 0$) by the local HEMS, with the respective power amount, in order to match and cover the central EMS' desired import/export target.

It is required that the agent's local HEMS as well as the local inverter provide this capability of setting/overriding the momentary AC power to be either exported into the grid from ESR/PV or imported from grid into ESR, at a defined target power rate: Changing $\Delta p_{target}^{m,t}$ is supposed to directly affect the agent's $p_G^{m,t}$ and thus also $p_S^{m,t}$. The production and consumption powers $p_{PV}^{m,t}$ and $p_C^{m,t}$ are not affected. It is up to the agent's HEMS to cooperate with the central EMS for optimal usage (in the community's perspective) of its excess energy.

The SumZero strategy of the central EMS calculates the active target power $\Delta p_{target}^{m,t}$ using its algorithm 3.10, per ESR member, and asks the member's HEMS to adapt the inverter target power settings in either a positive and negative direction, by applying $\Delta p_{target}^{m,t}$ as an additional offset in the local HEMS.

This mechanism therefore makes use of the "Controllable agent implementation" on the member side, as described in 3.3.

As an example, figure 4.4 shows all power values of a single EC member over a day. The plus symbols represent the $\Delta p_{target}^{m,t}$ in every time step, as calculated and desired by the central EMS and transmitted to this specific member. At $\Delta p_{target}^{m,t} > 0$, marked (A), the EMS asks this member to import (store or consume) this amount of energy from the grid due to an overall excess in the EC. At $\Delta p_{target}^{m,t} < 0$, marked (B), the ESR is asked to discharge its ESR to deliver this amount of energy to the grid.

The black dotted line expresses the member's measured grid power import (positive), as a result (remainder) of the sum of all its production, consumption, and $\Delta p_{target}^{m,t}$.

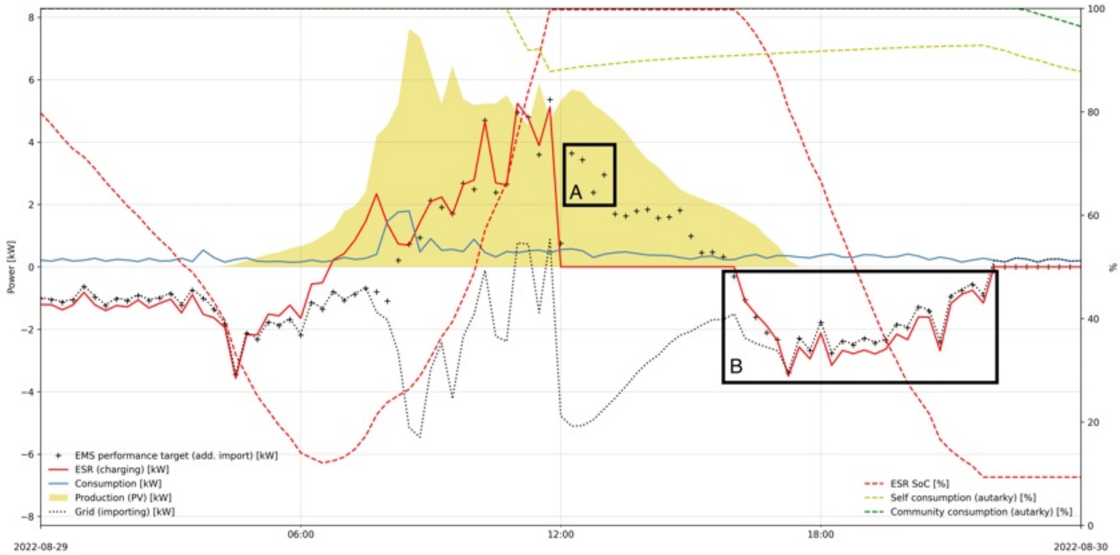


Figure 4.4: Daily overview of a single ESR member and its non-obedience (A) and obedience (B) of active target power assignments

Situation (A) shows $\Delta p_{target}^{m,t}$ set to $\sim 3kW$ for this member by the EMS, due to an excess of energy in the whole EC. The member is asked to import energy at a rate of 3 kW (0.75 kWh at every 15 minute time step). Due to the fact that the member's ESR's SoC reaches 100% at 11:45, and additionally low consumption and high production rates occur, the member does not and cannot obey it, but even feeds its excess into the grid. This amount of energy is therefore sold to the central grid.

Situation (B) shows $\Delta p_{target}^{m,t}$ set to between $-0.1kW$ to $-3.8kW$ for this member by the EMS, due to a lack of energy by other agents in the community. This member fully obeys the prompt by fully emptying its ESR. Since the ESR is empty, the central EMS no longer asks the member for more energy.

$p_{drift}^{m,t}$ quantifies the member's power delta (deviation) of central EMS expectations about the member and its actual behavior:

$$p_{drift}^{m,t} =\mid \Delta p_{target}^{m,t} - p_G^{m,t} \mid \tag{4.8}$$

If grid power and $\Delta p_{target}^{m,t}$ overlap ($p_{drift}^{m,t} = 0$), then the member fully obeys $\Delta p_{target}^{m,t}$.

At 12:15, for example, $p_{drift}^{m,t}$ reaches its maximum for this day ($\sim 8.7kW$). In this case, the member does not properly make use of its excess energy. A low integral value of $p_{drift}^{m,t}$ would increase the equilibrium of the member, which stabilizes the central grid.

$\Delta p_{target}^{m,t}$ can also be used as a trigger by a HEMS to activate additional devices consuming energy locally, on demand: Instead of selling this excess energy (usually at a low price) into the central grid, the member could enable heating rods or start EV charging. This is called "load shifting", which will be further addressed in case study 3.

4.5 Evaluation and results

As part of this case study, a multitude of simulations using multiple different EC compositions will be performed, operating with 3 different EMS strategies. The results will then be evaluated using the KPIs as just explained.

The presented charts and benchmarks were generated from data from a sunny summer week, where good photovoltaic performance was achieved across all producing members. For a holistic evaluation of the performance of the EC, a whole year of energy data would be required, which was not within the scope of this thesis.

All time series energy profile charts of ECs or households depicted below and throughout the thesis, were generated by the framework itself, by instantiating whole EC through the framework's simulation runner.

All results gathered by the "Individual" and "NoOp" EMS strategies were solely based on real-world data generated by all members, using the "readonly agent implementation".

"SumZero" EMS strategy makes furthermore use of the "active target power control" mechanism of the SmartGrid framework and thus involves modeling the HEMS' behavior: Keeping all member's production and consumption behavior unchanged, SumZero incorporates simulating the additionally effects of $p_{drift}^{m,t}$ for all ESR members of the EC using the "controllable agent implementation".

This enables the evaluation of the effects of an intelligent central EMS within energy communities.

4.5.1 Simulation of different EMS strategies and compositions

The following graphs show the test results of 10 different EC compositions, each of which contains 24 real households (ESRs), using the proposed framework and real-world data. For every EC composition, the NoOp strategy (all figures on the left side) is compared with SumZero (all figures on the right side). Using the framework's simulation runner, all compositions and strategies with data timespan of a week are fast-forwarded in a fully automated way. The generated graphs and their KPIs are then discussed throughout this section.

EC naming convention:

"XesrYcons" (e.g., 4esr20cons) is defined as follows: "XesrYcons" is the name of the energy community, containing X members that are ESRs and Y consumers as members ("ESR" and "consumer" are defined in the chapter 3.5).

The load profiles (consumption behaviors) of the agents do never change, regardless of the agent's configured category during the simulation run. This allows for comparability between the EC compositions, since energy consumption is always the same at any point in time for all members. Therefore, during the simulations, all the Y consumers are simply stripped of their PV and ESR capabilities.

Colored chart areas:

The following charts show the power measurements (sum over all grid meter measurements of all EC members, separately for excess and feed-in) and further KPIs over the period of time and use 3 different colors to depict the "energy flow", as follows:

- **Gray areas:** Amount of energy shared between members of the community. Positive energy (power values above 0 kW) is the total feed-in energy by all members at the given point in time (15min buckets). The mirrored negative gray area (power values above 0 kW) is the consumed energy.
- **Green areas**: Excess energy (too much energy within the community that could not be consumed by the community and had to be sold to the central grid). Typically seen during PV excess, after all ERS are fully charged.
- **Red areas**: Lack of energy within the community that had to be bought from the central grid to satisfy some of the EC's consumers. Typically seen at night when the EC's "virtual battery" gets empty.

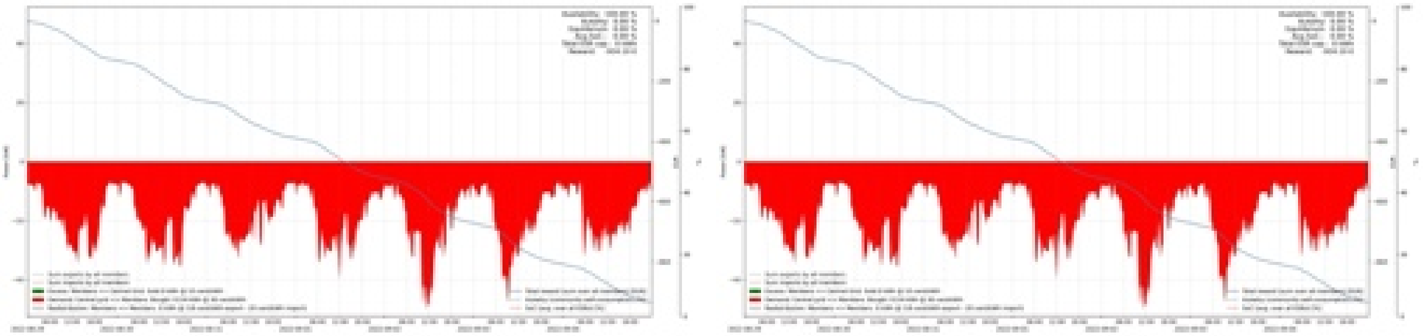


Figure 4.5: EC 0esr24cons (0 ESRs, 24 consumers), with NoOp strategy (left) and SumZero (right). Both charts and all KPIs are equal, because of the lack of any ESR. Therefore, the total energy demand of 3114 kWh of all 24 purely consuming members were imported from central grid. It is equal to the Individual strategy and thus obsoleting the EMS.

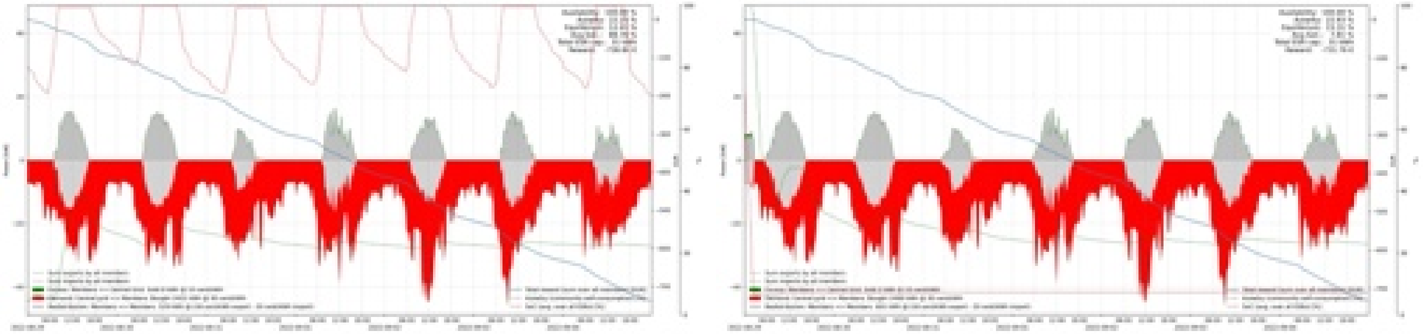


Figure 4.6: EC 2esr22cons. In this composition, 2 ESR prosumers with a total ESR cap of 31 kWh and a weekly production of 693 kWh, are opposed to 22 pure consumers. They are far from being able to cover the total demand (3114 kWh), but 601 kWh of energy is transferred from those members to the remaining 22, decreasing the EC's total energy bill. With SumZero, the central EMS almost immediately (in the first hour) empties all stored energy (ESR is considered empty by the EMS if $SoC \leq 10\%$), resulting in a low avg. SoC of about 8%, which slightly increases EC autarky and equilibrium. The central EMS has negligible benefits for such ECs.

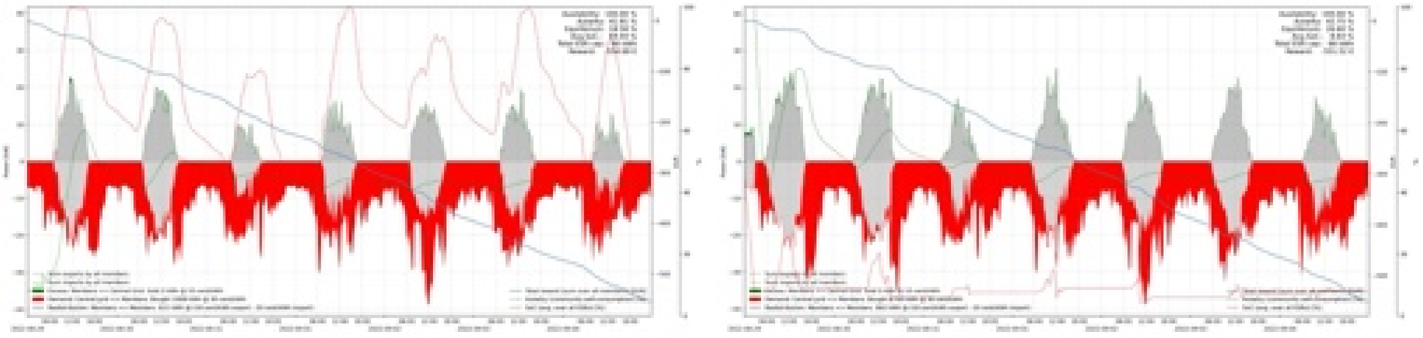


Figure 4.7: EC 4esr20cons. The behavior is very similar to the previous 2esr22cons. The EMS has negligible impact over total autarky (42%) and reward, but a noticeable impact on the EC's equilibrium (18.6% increased to 26.6%), and thus an already recognizable, positive impact on the central grid.

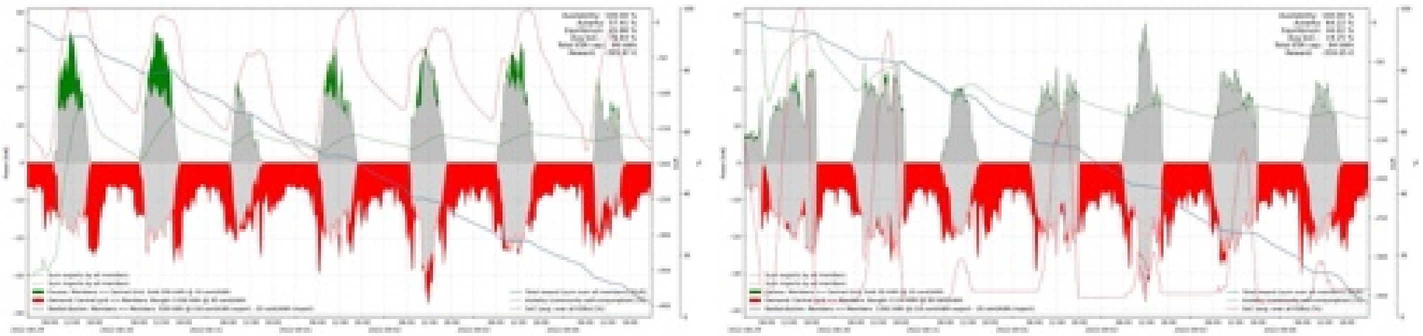


Figure 4.8: EC 6esr18cons. The benefits of the EMS and SumZero strategy are clearly visible for the first time: SumZero reallocates all the PV excess energy of the 6 prosumers uniformly to the remaining members, instead of selling it into the central grid first, only to import it again a few hours later. The benefits are visible through all performance indicators (as seen on the top right): Autarky and equilibrium (25.9% increased to 44%) show significant improvements. As already seen in all the previous charts, also here the ESR's state of charge changed drastically from 76.6% to 19.6%, indicating a heavy use of the ESR's energies, within a still imbalanced EC composition (empty and thus over-sized ESRs at the most of points in time).

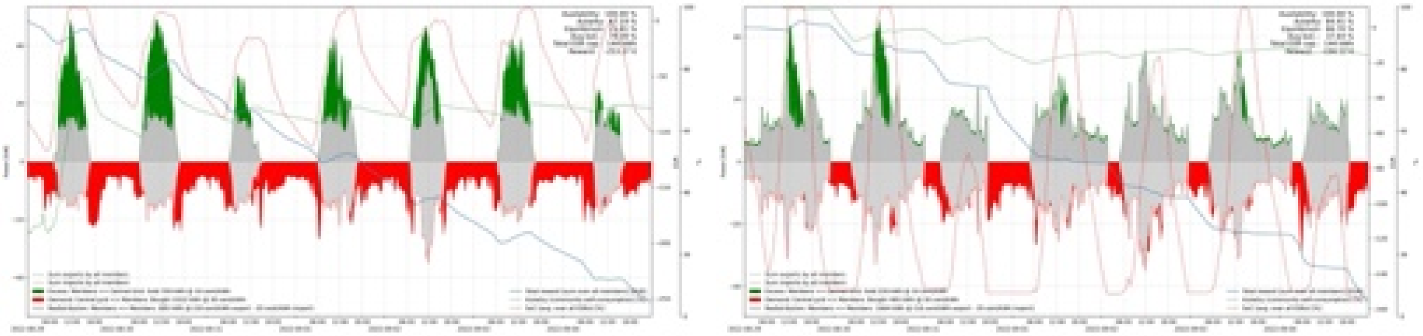


Figure 4.9: EC 9esr15cons. The previous effects for 6esr18cons are amplified and benefits of SumZero over NoOp are clearly visible. 9 prosumers, introducing a total ESR capacity of 144 kWh and daily average production of 402 kWh into the community, are largely covering the 449 kWh of avg. daily demand (of all 24 members, incl. themselves). For SumZero, the overall SoC line depicts a very good utilization of the overall ESR capacity: one full charging cycle daily.

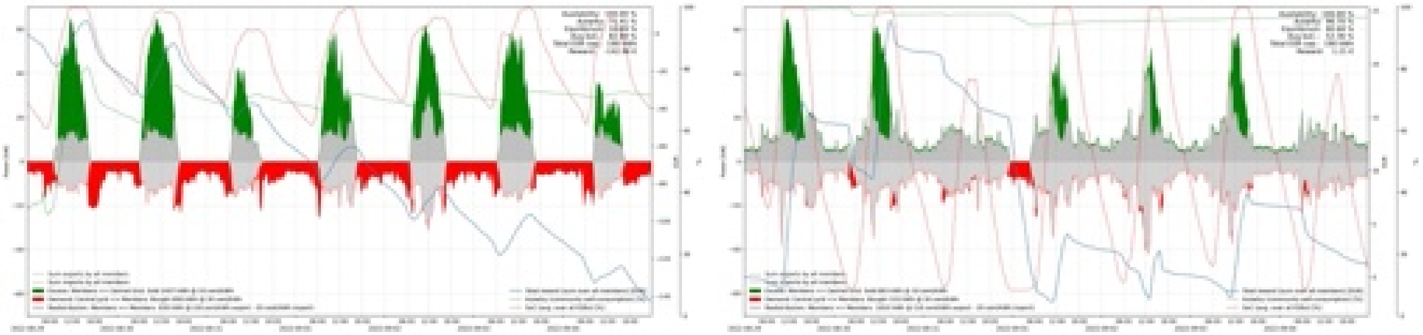


Figure 4.10: EC 12esr12cons. The NoOp strategy leaves plenty of ESR capabilities unused, resulting in an even lower equilibrium compared to 9esr15cons, resulting in a high dependency on the central grid for exports and imports. With SumZero, the average SoC of the VPP's distributed battery of 52% indicates an almost perfectly balanced EC composition, resulting in a great community autarky of 96.7% and high equilibrium of 80% (100% is full independence). In this impressive case, 12 prosumers with ESRs greatly increase the efficiency of 12 other consumers, due to the work of a central EMS.

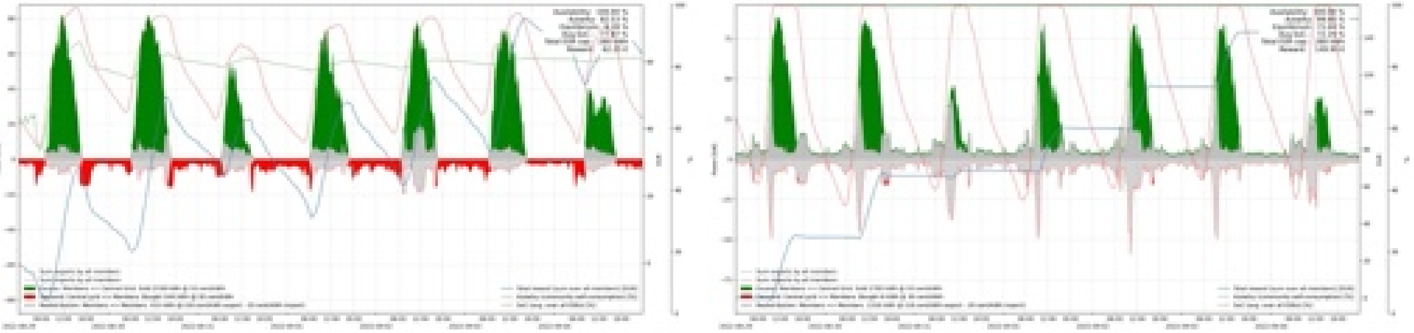


Figure 4.11: EC 18esr6cons. In this EC, all PV and ESR sizes, which have been precisely optimized for every of its 18 prosumers, are easily capable of fully covering the energy demand of 6 further consumer households (at during a sunny summer week). 99.8% of the consumed energy is produced by the EC itself. More plain consumers should join the EC for optimal operation, since some ESR capacity is left unused, as can be seen by the high avg. SoC.

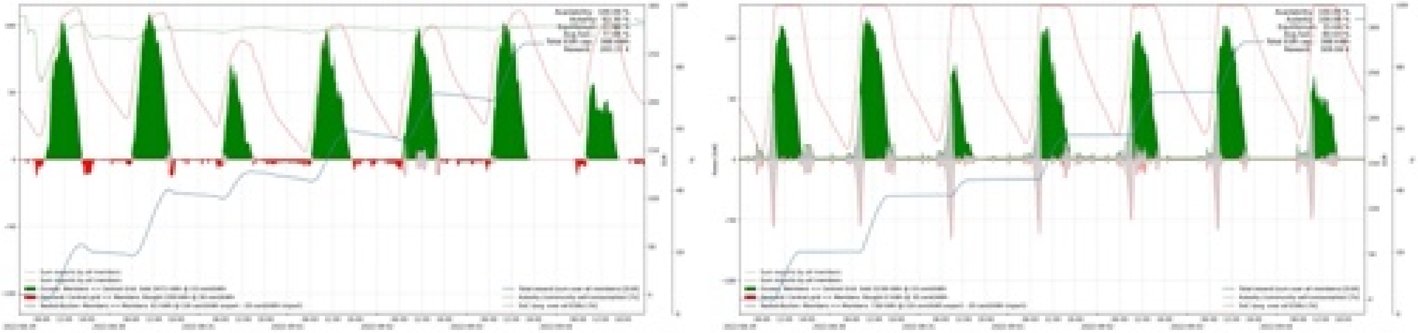


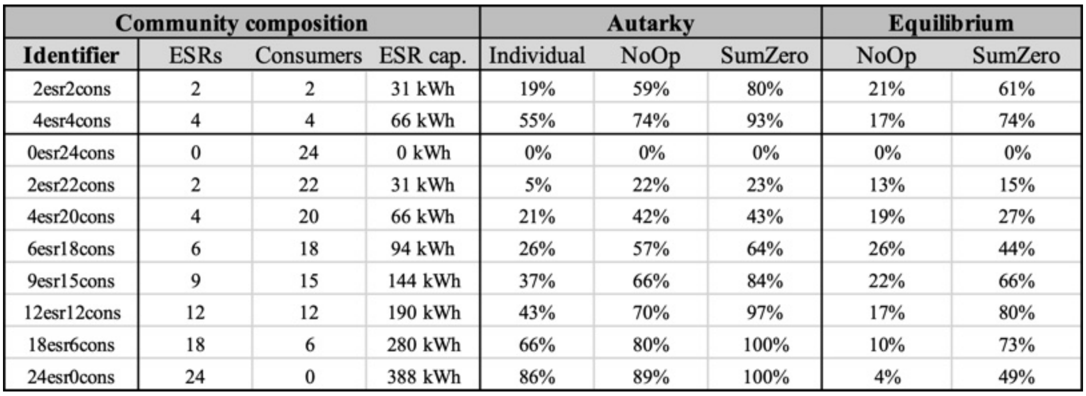
Figure 4.12: EC 24esr0cons. For completeness, this EC composition consists solely of similar-sized, prosumer households with ESR. Although the EC accomplished a very high reward from selling excess energy, the transformer station is unnecessarily stressed by the excess (assuming no further consumers are attached that would consume the excess at the same time) and community ESR capacity left largely unused (again, too high avg. SoC). Such EC compositions must be avoided due to economical and ecological reasons.

4.5.2 Performance comparison and evaluation

This section summarizes and compares the key performance indicators from the previous qualitative discussions of the 10 EC compositions and 3 strategies.

Table 4.1 shows the autarky and equilibrium, over 1 week of data of different EC compositions (as discussed in the previous section). The capacity of the community ESR is the total amount of energy that the central EMS can control. Individual autarky is (per definition and simulation) always zero and was therefore omitted.

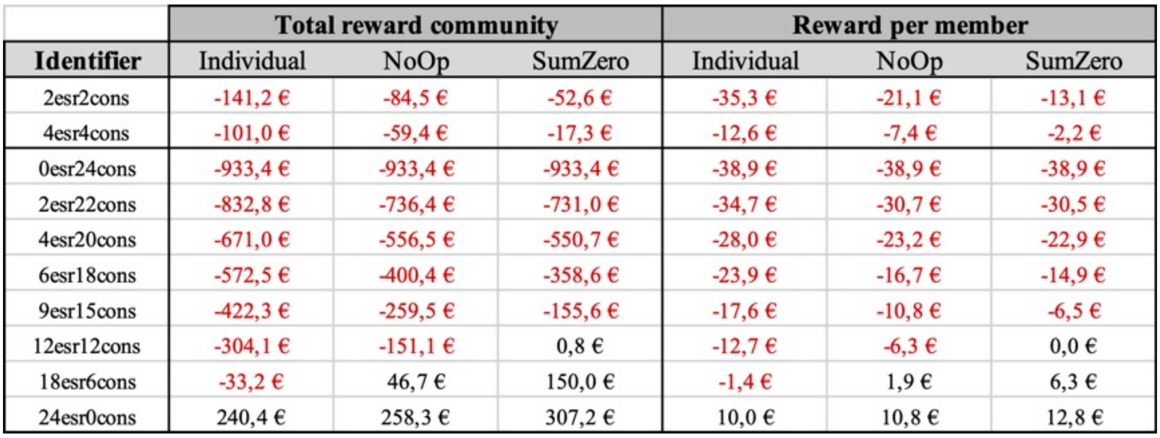
The highest equilibrium is obtained in the simulation on EC 12esr12cons @ SumZero, which is also considered the most efficient of those 10 EC compositions, from a CapEx perspective. The smaller the EC, the weaker their performance in general (comparing 2esr2cons with 4esr4cons and 12esr12cons):



Community composition				**Autarky**			**Equilibrium**	
Identifier	ESRs	Consumers	ESR cap.	Individual	NoOp	SumZero	NoOp	SumZero
2esr2cons	2	2	31 kWh	19%	59%	80%	21%	61%
4esr4cons	4	4	66 kWh	55%	74%	93%	17%	74%
0esr24cons	0	24	0 kWh	0%	0%	0%	0%	0%
2esr22cons	2	22	31 kWh	5%	22%	23%	13%	15%
4esr20cons	4	20	66 kWh	21%	42%	43%	19%	27%
6esr18cons	6	18	94 kWh	26%	57%	64%	26%	44%
9esr15cons	9	15	144 kWh	37%	66%	84%	22%	66%
12esr12cons	12	12	190 kWh	43%	70%	97%	17%	80%
18esr6cons	18	6	280 kWh	66%	80%	100%	10%	73%
24esr0cons	24	0	388 kWh	86%	89%	100%	4%	49%

Table 4.1: Autarky and equilibrium, over 1 week of data of different EC compositions.

Table 4.2 extends the previous table and summarizes the total rewards of the 10 different ECs, each individually simulated using 3 different EMS strategies:



	Total reward community			**Reward per member**		
Identifier	Individual	NoOp	SumZero	Individual	NoOp	SumZero
2esr2cons	-141,2 €	-84,5 €	-52,6 €	-35,3 €	-21,1 €	-13,1 €
4esr4cons	-101,0 €	-59,4 €	-17,3 €	-12,6 €	-7,4 €	-2,2 €
0esr24cons	-933,4 €	-933,4 €	-933,4 €	-38,9 €	-38,9 €	-38,9 €
2esr22cons	-832,8 €	-736,4 €	-731,0 €	-34,7 €	-30,7 €	-30,5 €
4esr20cons	-671,0 €	-556,5 €	-550,7 €	-28,0 €	-23,2 €	-22,9 €
6esr18cons	-572,5 €	-400,4 €	-358,6 €	-23,9 €	-16,7 €	-14,9 €
9esr15cons	-422,3 €	-259,5 €	-155,6 €	-17,6 €	-10,8 €	-6,5 €
12esr12cons	-304,1 €	-151,1 €	0,8 €	-12,7 €	-6,3 €	0,0 €
18esr6cons	-33,2 €	46,7 €	150,0 €	-1,4 €	1,9 €	6,3 €
24esr0cons	240,4 €	258,3 €	307,2 €	10,0 €	10,8 €	12,8 €

Table 4.2: Total community and per-member reward as a result of 1 week simulation of different EC compositions given fixed energy prices.

Figure 4.13 compares the avg. reward per member, when placed in different EC compositions (x-axis), or more precisely: the same consumption behavior, but with its agent configured as ESR or consumer (as defined in 3.5). Each point used in the chart was (as before) determined by using this framework's simulation capability based on real energy data and the time horizon of 1 week.

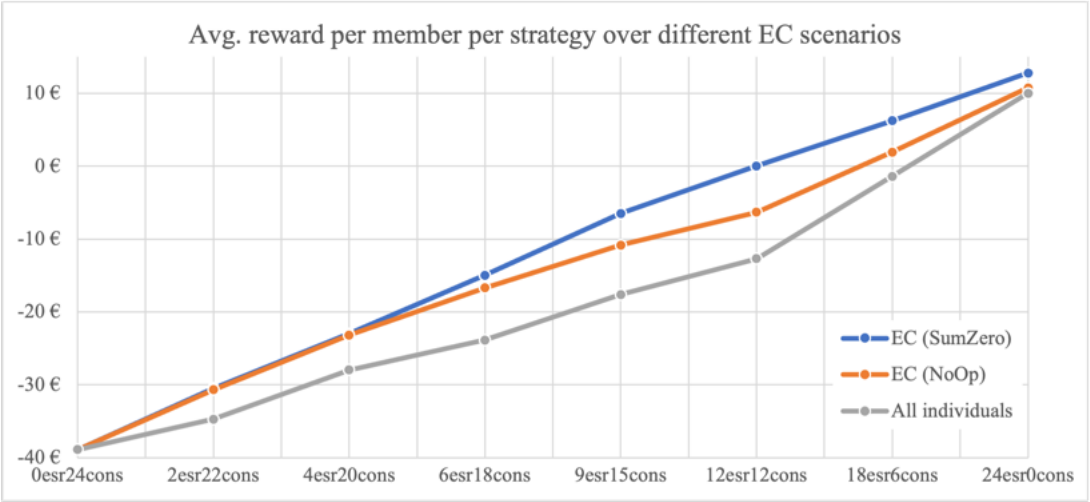


Figure 4.13: Avg. reward per member, per each strategy and different EC compositions.

An interesting finding in figure 4.13: The SumZero strategy (with active power control) has almost no impact on ECs containing only a few ESRs but performs almost the same as NoOp. In other words: Figure 4.13 shows that the concept of ECs in general already pays off, even without any coordination using a central EMS (which is the NoOp strategy). However, the SumZero strategy is able to further increase the average autarky of all members by up to 28% (12esr12cons).

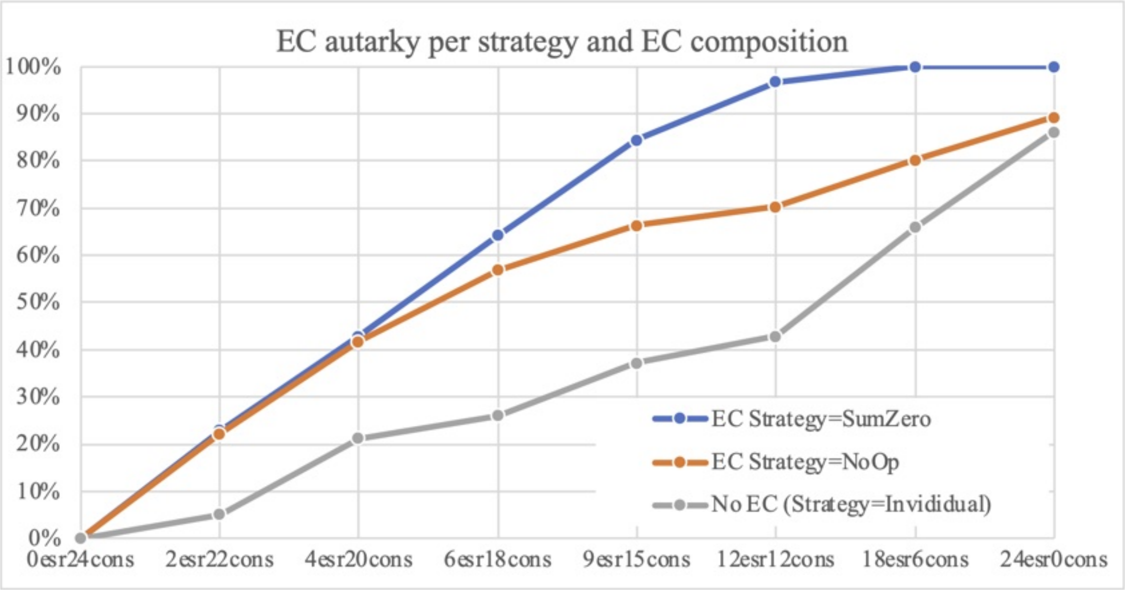


Figure 4.14: Autarky level of the whole EC, per strategy and different EC compositions, over 1 week.

Figure 4.15 shows all simulations (all EC compositions and all strategies) in a single chart with their resulting rewards and autarky levels.

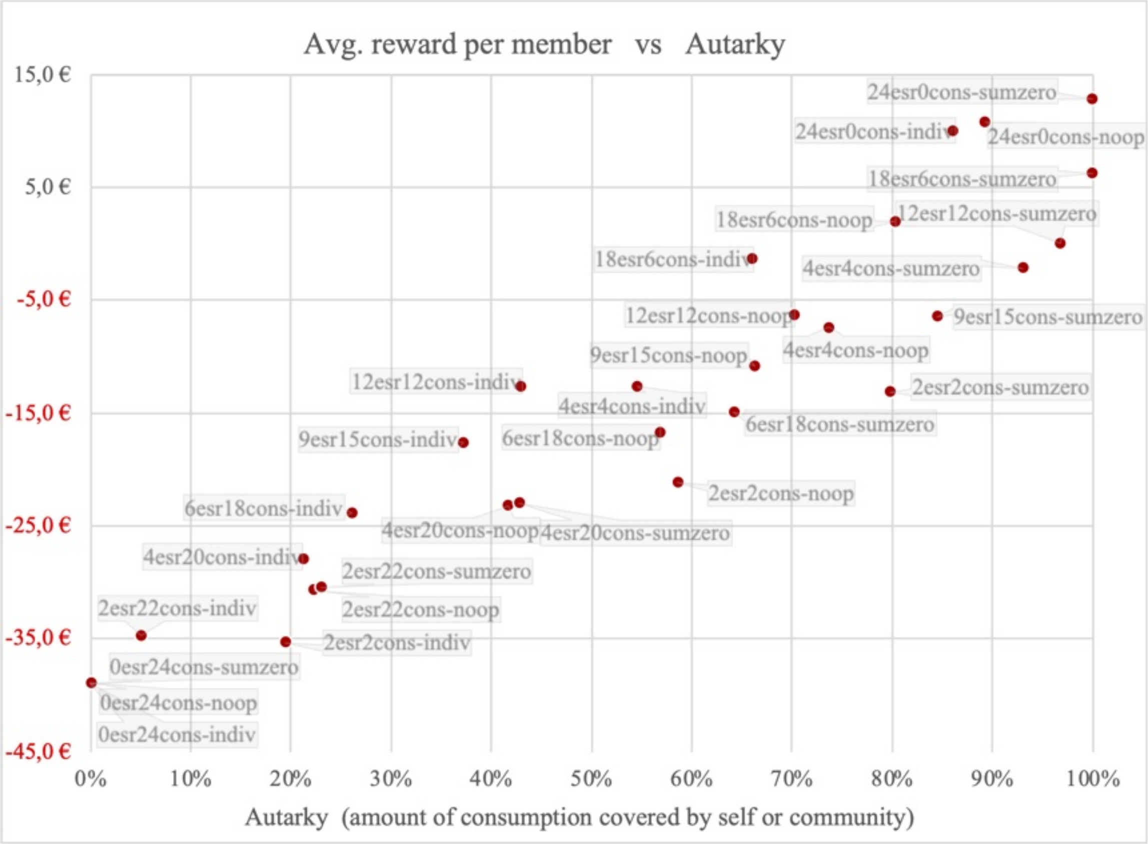


Figure 4.15: xy-chart of all simulations based on reward and autarkies.

Interpreting all tables and figures within this section, the following statements can be derived:

1. Energy communities running with SumZero strategy and consisting of ESRs behave as intelligent virtual power plants (VPPs). These ECs produce (and store) energy during the day and are able to transfer energy between members by compensating the EC's excess and demand, as simulated, controlled, and measured by the EMS of this framework.
2. Energy communities outperform individuals (see 4.1) in all simulated compositions, also for the NoOp strategy: ECs generate a higher total reward and autarky for each individual (see 4.2), given the defined, realistic energy tariffs. Only in the edge case *0esr24cons* (EC with consumers only; no exports, and imports are fully covered by the central grid), the EC itself does not bring any benefit in any strategy.
3. In general, bigger ECs perform better than smaller ECs with the same prosumer-ratio. This is due to the increased stability of the central EMS algorithm, caused

by a larger distribution of $\Delta p^{m,t}_{target}$ over a larger number of members (the law of large numbers applies).

4. The NoOp strategy is good enough for ECs with low ESR count (see 4.13). Already for *2esr22cons* (only 8% of the EC members are producers with ESR), the superiority of ECs (given the defined pricing settings) is clearly visible (7.8% higher reward). SumZero strategy would be negligibly better (8.2%). The highest rewards from EC scenarios that use an individual strategy as baseline are: *6esr18cons* with 13.9% higher rewards when using NoOp, and *18esr6cons* with 24.6% using SumZero.

5. To maximize autarky (and thus high reward, given community prices are better than grid prices), the efficient usage of the ESR capacity of the community is critical. An outstanding example of the performance difference of these 2 strategies is *12esr12cons* in figures 4.1 and 4.2: For the same EC composition, NoOp achieves autarky=70%, community reward=-151€, whereas SumZero achieves autarky=97% and community reward=1€. The average SoC in all ESRs and time steps is 85% in NoOp and 51% on SumZero. On NoOp, the ESRs are only minimally and thus inefficiently utilized by the community, which is also illustrated by the SoC line in the graphs.

6. The self-sufficiency ratio (autarky) is directly proportional to reward (figure 4.15, as long as central grid energy consumption prices are higher than community prices, and as long as the community pays a better price for excess energy than the grid. This is typically the case in practice and is caused by higher taxes and fees on any energy transfers to/from the central grid energy (and has been applied to all simulations as well (see 4.3.3)).

4.6 Discussion

The results, especially figure 4.14 demonstrates that the SmartGrid Framework is capable of scheduling, controlling, and optimizing energy communities.

The SumZero strategy of this thesis' EMS was implemented to maximize self-sufficiency (algorithm 3.10). SumZero provides, as the simulation results show, the best self-sufficiency performance and rewards across all EC mixes. This confirms the conclusion of an article on community-focused energy management systems during state-of-the-art research 2.6.

The superiority of energy communities with a central EMS coordinator and active power control to advise HEMSes, over non-smart ECs (NoOp), can already be achieved and was demonstrated for ECs with a prosumer ratio of at least 20% (e.g. 5esr19cons). This applies to reward (economical benefit) as well as to autarky and equilibrium (ecological benefit).

The flexibility of those prosumer households with ESRs is positively impacting all the remaining, purely consuming households in the EC, increasing their individual reward and autarky.

EC *12esr12cons*, with a prosumer ratio of 50%, shows the biggest relative improvements for the whole EC: The avg. autarky could be increased from 70% to 97%. This could be done without any change in member's consumption behavior, but by utilizing the ESR in a smart and coordinated way.

The following case study 3 can be seen as an extension. It is based on the same EC compositions and will analyze the positive impact of members willing to provide additional behavioral flexibility to the EC.

It is important to note that all these values and results are based on 24 randomly chosen Austrian households, and over a time window of one week during summer and good weather (see 8).

CHAPTER 5

Case study 2: The framework as a distributed system

5.1 Introduction

This case study analyses different aspects of the framework in the context of distributed systems, its hybrid approach, architecture, and limitations, using qualitative and quantitative methods.

In section 5.2 a comparison between the presented thesis and an interesting article during the research is further analyzed and claims addressed. Section 5.3 qualitatively examines the non-functional characteristics of the framework. Section 5.4 quantitatively simulates random fault injections into the framework and Kubernetes, to simulate connection losses of individual members and their impact on EC performance. The final section summarizes the findings of this case study.

5.2 Qualitative analysis of [CDC18]: Centralized vs. decentralized MGMS

[CDC18] discusses micro grid management systems (MGMS) and highlights the benefits of pure distributed MGMS approaches, which act without a central controller. It states that „The distributed MGMS framework delivers not only the same control functions as the centralized MGMS but also greater scalability, reliability, and resiliency.“. It further lists the following statements on the superiority of distributed MGMS over a MGMS with a centralized MGCC, which are addressed and objected as follows.

> "Distributed control systems have greater controller redundancy and are robust to a single point of failure. Controller failures will not cause system

> blackout."

Objection: The presented thesis uses a hybrid approach of decentralized HEMS, which is controlled by a centralized cloud-based EMS. Case Study 2 will further analyze the effects and impacts of such failures.

> "System maintenance and upgrade can be done without shutting down the entire system."

Objection: The presented framework is based on Kubernetes and uses the rolling update mechanism for all stateless components, which allows zero-downtime maintenance, upgrades, and MG topology changes. The stateful EMS exists per EC and is the only single point of failure that requires a coordinated shutdown. A consensus mechanism can be a future improvement to allow coordinated hot-swapping of the EMS deployment. Case Study 2 will further address this topic.

> "Local decision making reduces network use, relaxing the communication bandwidth requirement." and "The distributed control framework is more flexible and scalable for future modifications and expansions."

Objection: Future modifications and expansions increase the complexity. In a P2P framework as proposed by [CDC18], interoperability among all participants must be established, which significantly increases complexity and the amount of data. For an optimally performing smart grid, its EMS requires full knowledge of the momentary state of all the agents with fast updates, in order to make the best decisions (minimal uncertainty). Gossip-based communication is prone to delays, which increases with every hop. Furthermore, such network topologies introduce redundancy, which increases bandwidth and network traffic, compared to a centralized approach. In a P2P/gossip-based network, a fully connected topology requires n(n-1)/2 communication paths between decentralized controllers (worst case), which drastically increases network traffic on scaling.

> "Historical user energy information is stored locally, which protects user privacy."

Objection: This is correct, but the hybrid approach, as presented, requires the members/agents to provide only information that is required to optimize overall performance. Agents with a narrow view can never outperform central systems/agents that have at least the same and even more information available.

The main difference of [CDC18] is that the approach presented here is based on grouping of multiple, isolated, distributed sites to a large community, which is then controlled by a central MGCC. With this hybrid approach, outage of the central MGCC/EMS does not lead to a global outage, but results in a reduced efficiency of the community in worst-case; as if the members were not grouped at all.

Thus, the presented approach combines the benefits of centralized and distributed approaches into a hybrid agent-based framework.

5.3 Framework non-functional characteristics

This section lists and discusses non-functional characteristics of the presented framework with its hybrid HEMS + centralized EMS approach (see also the framework's architecture 3.1).

5.3.1 Timing and communication

- The HEMS running on-site is able to react in ms area by analyzing and controlling the local devices within the LAN.
- The HEMS updates its digital twin agent regularly in an interval of 1-60s pull or push based.
- The centralized EMS control loop also gathers the communities' data in a 1-60s interval to derive and propagate actions back to the twins.
- A hierarchical star topology is used throughout the framework to gather information and distribute actions.
- No redundant information is exchanged, leading to reduced bandwidth usage compared to decentralized gossiping protocols.
- Due to the star topology, communication outages or delays of the members are locally isolated and do not further affect other members. The overall performance is proportional to the amount of outage (see 5.6).

5.3.2 Performance and scalability

- The majority of the computation takes place in the EMS of the local site. The local (and locally centralized) EMS has full knowledge and direct connection to all energy devices on-site.
- Changes at a site only affect the HEMS, since it abstracts the details of the site from the perspective of the framework. This further increases privacy.
- The framework is therefore not affected regarding performance concerns by growing an existing site in terms of adding additional energy devices like inverters, EV chargers or batteries.

- Since every member is assigned to exactly 1 community, every community agent is responsible to gather the information and distribute actions from and to every of its members. This is done efficiently, without further data processing.

- Since every community is assigned to exactly one central EMS / MGCC, the number of connections remains unchanged, but the amount of data exchanged and to be processed increases linearly.

- The performance and scalability limits of the MGCC are defined by the strategy / implementation of the MGCC itself: The MGCC runs as a control loop that obtains the whole community state incl. that of every member, processes it, derives the next actions, and pushes it back to the community, which further propagates it to the twins. Therefore, large communities required efficient implementation of the strategy.

- Due to the locality restrictions given by law, members of the energy community are located within a limited geolocation area. This introduces a natural boundary to the maximum size of a community and its MGCC.

- Multiple energy communities in the current framework implementation are isolated from each other. A future approach, further discussed in this thesis, is to extend the existing hierarchy to form communities of communities. This allows building large-scale VPPs composed of thousands or millions of members but introduces an additional delay for every hop in the star topology.

- The amount of computing resources and the number of deployments and services within the Kubernetes clusters, created and maintained by the framework, grow linearly with the number of members. This can be reduced to the number of pods and nodes the Kubernetes cluster can handle. The current best practices are state [https://kubernetes.io/docs/setup/best-practices/cluster-large]: Max. 110 pods per node, 5000 nodes, and 150000 total pods. This introduces an upper limit of max. 140k members that can be handled by one instance of the framework.

- Changes to the community topology (changing community membership) are handled by the SG Operator of the framework and are currently processed sequentially by a single controller. This bottleneck can be overcome by introducing a shared, global locking s.t. multiple controllers can run in parallel with distributed load. Since a topology change (i.e., switching the community) is a seldom operation in the lifetime of a member, this thesis does not further address this issue.

5.3.3 Resilience

- Kubernetes provides a lot of mechanisms to keep deployments available, even in case of failures. The scheduler always tries to keep at least the minimum defined amount of replicas running and automatically reschedules faulty entities.

- QoS settings, Autoscalers, Pod-disruption-budgets, and Anti-affinities can be added to the SmartGrid Kubernetes Operator to increase resilience against sudden node failures, especially for the potentially large amount of agent and community pods.
- The framework uses timeouts for all requests to gracefully handle disruptions. This leads to performance degradation, but prevents starvation.

5.3.4 Privacy and Security

- Since the framework itself runs on Kubernetes, operational security and privacy are fully up to the operating party. The framework can be deployed in a private cloud or even on a single machine.
- The central EMS deployment is the only stateful entity that collects, tracks, and keeps all the information from all members. The community deployments forward all individual per-member, as well as the aggregated data representing the community, towards the EMS upon its request. This request is stateless and transient. Therefore, all EMS and community entities should be deployed on private compute resources.
- All existing external connections are the TCP connections from the agent entity (as a digital twin) to its physical counterpart (the member with its HEMS). Every single agent is an isolated deployment, with a per-member bridge implementation, an isolated communication path, and unique credentials. It can also be deployed closer to the edge.

5.4 Fault injections

As is commonly known, all distributed systems are prone to certain failures. The "Eight Fallacies of Distributed Computing" [fal] as summarized by Peter Deutsch:

> Essentially everyone, when they first build a distributed application, makes the following eight assumptions. All prove to be false in the long run and all cause big trouble and painful learning experiences.
>
> 1. The network is reliable
> 2. Latency is zero
> 3. Bandwidth is infinite
> 4. The network is secure
> 5. Topology doesn't change
> 6. There is one administrator
> 7. Transport cost is zero
> 8. The network is homogeneous

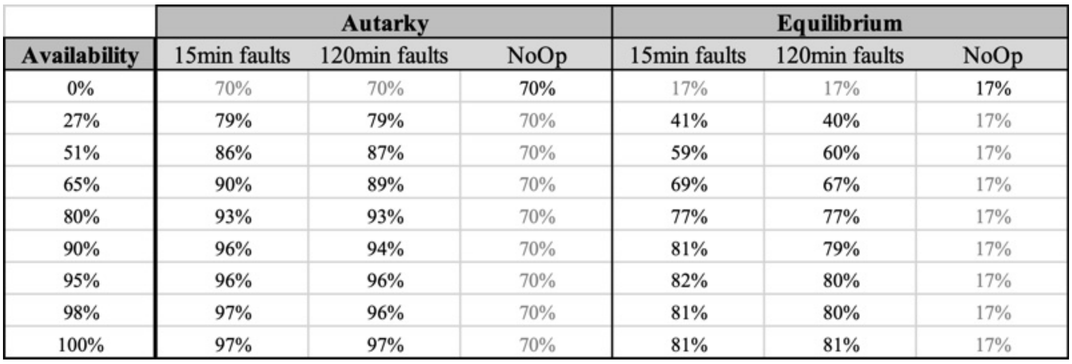
Since the presented framework is also not spared either, this section analyses the effects of failures towards the EC performance. These sections will largely cover the failures 1, 2 and 5, but using a rather simple all-or-nothing outage simulation per member. Therefore, a random failure table has been generated that decides for every single member for an a) 15-minute interval and b) 120-minute interval, if the member should be available to the framework and thus provide a data update, or not (timeout). The framework has a configurable default timeout of 30 seconds, which is "fast-forwarded" to accelerate the simulation.

5.5 Evaluation and results

Based on a randomly generated but static "failure table" for every single member over the whole time horizon of 1 week, and a customized "Controllable agent implementation" (see 3.3) that simulates request timeouts on reads and writes to the agent instances, the following results were gained. All the following simulations and evaluations were performed using the EC composition "12esr12cons".

Table 5.1 shows the resulting, degraded autarky and equilibrium values towards the community, when simulating sporadic network outages. 8 times 2 simulations have been executed using 12esr12cons: 8 different "failure tables" that lead up to the total availabilities (0 to 100%, rows in the table), with the 2 different kinds of failures (short vs. long outages) and depict their effect on the EC's autarky and equilibrium.

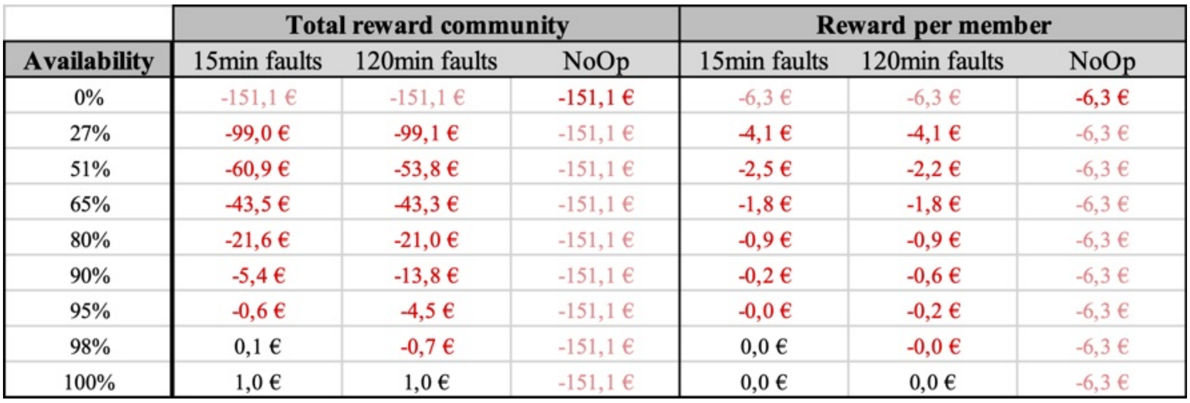
The strategy "NoOp" is resistant to network failures, because it does not carry out any actions toward agents and is equal to an availability of 0%.



	Autarky			Equilibrium		
Availability	15min faults	120min faults	NoOp	15min faults	120min faults	NoOp
0%	70%	70%	70%	17%	17%	17%
27%	79%	79%	70%	41%	40%	17%
51%	86%	87%	70%	59%	60%	17%
65%	90%	89%	70%	69%	67%	17%
80%	93%	93%	70%	77%	77%	17%
90%	96%	94%	70%	81%	79%	17%
95%	96%	96%	70%	82%	80%	17%
98%	97%	96%	70%	81%	80%	17%
100%	97%	97%	70%	81%	81%	17%

Table 5.1: Autarky and equilibrium for EC 12esr12cons (as already presented in case study 1), but with additional fault injections.

Table 5.2 extends the previous table 5.1 to show the effect of failure injection on the rewards (total and per member):



	Total reward community			Reward per member		
Availability	15min faults	120min faults	NoOp	15min faults	120min faults	NoOp
0%	-151,1 €	-151,1 €	-151,1 €	-6,3 €	-6,3 €	-6,3 €
27%	-99,0 €	-99,1 €	-151,1 €	-4,1 €	-4,1 €	-6,3 €
51%	-60,9 €	-53,8 €	-151,1 €	-2,5 €	-2,2 €	-6,3 €
65%	-43,5 €	-43,3 €	-151,1 €	-1,8 €	-1,8 €	-6,3 €
80%	-21,6 €	-21,0 €	-151,1 €	-0,9 €	-0,9 €	-6,3 €
90%	-5,4 €	-13,8 €	-151,1 €	-0,2 €	-0,6 €	-6,3 €
95%	-0,6 €	-4,5 €	-151,1 €	-0,0 €	-0,2 €	-6,3 €
98%	0,1 €	-0,7 €	-151,1 €	0,0 €	-0,0 €	-6,3 €
100%	1,0 €	1,0 €	-151,1 €	0,0 €	0,0 €	-6,3 €

Table 5.2: Total community and per-member reward for EC 12esr12cons (as already presented in case study 1), but with additional fault injections.

Figures 5.1 and 5.2 are visual representations of the tables discussed above. The y-axis shows the respective KPI, the x-axis shows the 8 different total failure durations, and the two blue lines show the 2 types of failures.

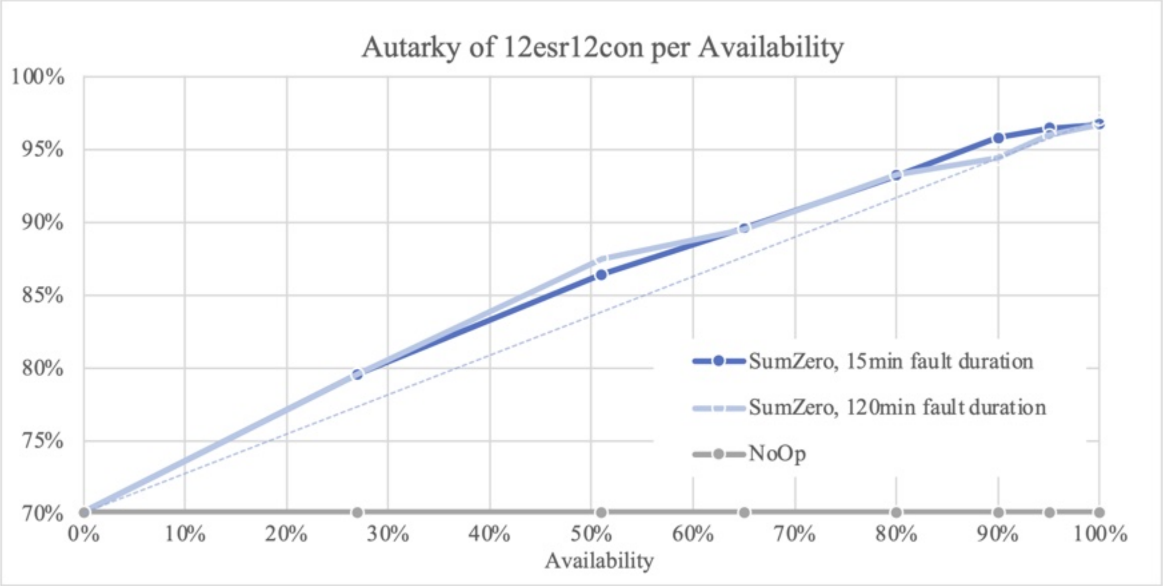


Figure 5.1: Autarky level of the whole EC 12esr12cons (from case study 1), but simulated with different fault scenarios in effect.

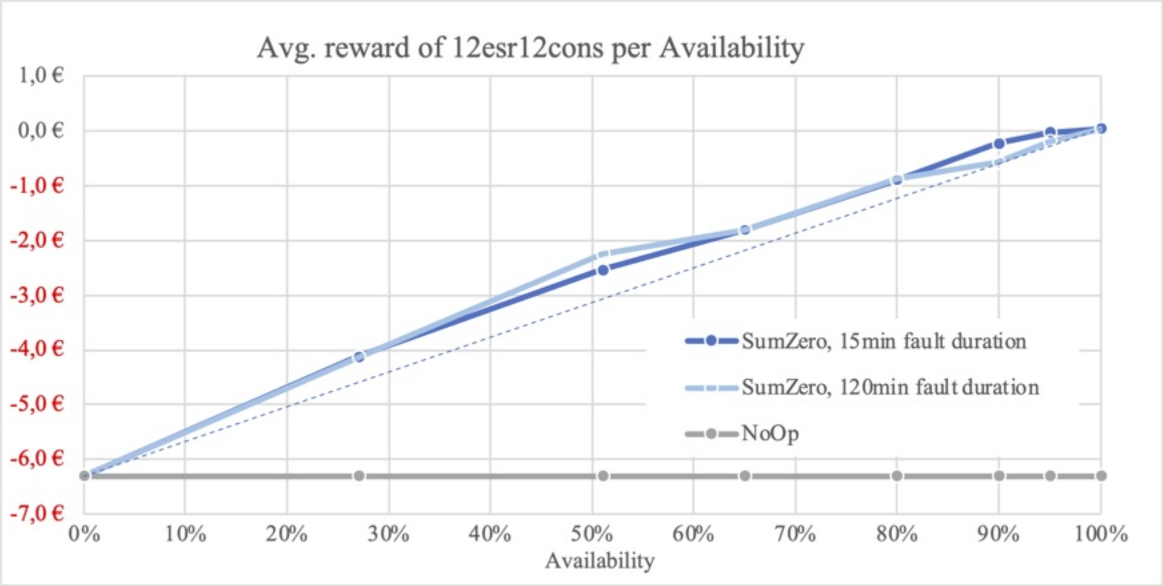


Figure 5.2: Avg. reward per member for EC 12esr12cons (from case study 1), but simulated with different fault scenarios in effect.

5.6 Discussion

The performance of all different community settings is highly correlated with the availability of members. Both total reward and autarky of the community under test (12esr12cons), show a behavior directly proportional to the measured availability (request success rate). The simulation results for other community compositions other than 12esr12cons show the same pattern.

As can be clearly seen in figure 5.1, the kind of failure (comprised of multiple small outages, or fewer but longer outages) has almost no impact on small communities (24 members in this case) and is therefore negligible for even larger ones.

A full outage of the central EMS controller (e.g. due to redeployment or temporary node loss) has the same effect as losing connection to all community members at the same time. This was not simulated, but has the same effect as applying the "NoOp" strategy for the respective amount of time.

It can be stated that random outages across all members of the community for about 5% (approx. 1 hour) per day on average do not have a noticeable impact on community performance.

Another positive side effect of the hybrid approach of this framework's smart grid architecture, which requires HEMS instances for each member: the worst-case effect to members and the community is equal to falling back to the "NoOp" strategy during the time of outages.

Although Kubernetes provides a lot of flexibility and benefits for running distributed applications, it also adds complexity and comes with its own limitations. For smaller sized communities, similar results could possibly be achieved by using a monolithic software architecture, since high availability turned out to be of negligible importance.

CHAPTER 6

Case study 3: Consumption load-shifting

6.1 Motivation

While most EC members with PV and ESR are already optimizing their local energy budget for self-consumption, by utilizing PV excess energy as much as possible in predefined order (for example: first cover momentary consumption, then charge connected EVs, then enable the heat pump, then charge ESR, then feed the remainder into the grid): Members that are pure consumers were not further controlled by this framework so far, but treated as read-only agents; up until now.

Making use of dynamic load shifting brings an enormous benefit to the whole community, since energy flexibility is not further restricted to ESR members, but extended to the consumption side. Therefore, all members of the community, also pure consumers, can participate in the further optimization of EC KPIs.

The goal of this case study is to extend the EMS and implement a simple load-shifting mechanism that (slightly) changes the consumption behavior of all consumers (non-ESR members) and to quantify this effect on the community, by simulating the load shifts, while keeping the total energy balance of the consumer constant, for comparability.

6.1.1 Load shifting for residential sites

In Austria, aWATTar [1] provides flexible energy tariffs that reflect the EPEX spot market with energy prices in 15-minute resolution for at least 24 hours in advance.

---

[1] https://www.awattar.at

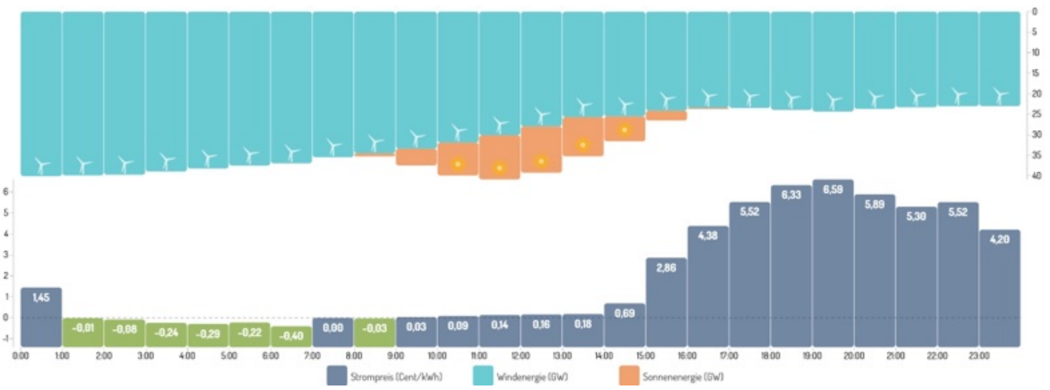


Figure 6.1: aWATTar day-ahead energy price visualized

Every household with such an energy tariff and intelligent HEMS that obtains those datasets via API has the ability to dynamically shift loads to increase economical rewards. Common use cases of the dynamic scheduling using a HEMS are:

- EV charging processes
- Enabling heating rods or heat pumps (thermal energy can be seen as one-way ESR)
- ESR charging to cover nightly shortages (especially in communities with only a few ESRs)
- Smart-home devices like washing machines

Companies like neoom [2], enpal [3] or gridX [4] are already providing such a HEMS, that is connected to the household's energy and smart home devices.

Flexible customers can benefit from dynamic energy tariffs from energy providers. Such tariffs are, in general, dearer than static ones, but flexibility pays off. Especially consumers with electric vehicles benefit a lot from dynamic energy tariffs because of their flexibility.

Most new EV chargers provide API access to services such as aWATTar to find optimal charging windows. However, such solutions only take into account the energy demand for the single device, without having the big picture of the site's or even EC's energy budget.

Ultimately, the benefits of dynamic tariffs are closely coupled to the end user's behavior, preference, convenience, demand, and flexibility.

---

[2] `https://www.neoom.com`
[3] `https://www.enpal.com`
[4] `https://www.gridx.ai`

6.2 EMS strategies and time windows

Since the central EMS already has the overall information on the communities' energy budget, this information can be forwarded to all members to provide suggestions for adjusting their individual's consumption behavior; in the same way, the active target power control has already been used to control the ESRs.

To achieve this, two new EMS strategies are introduced. For simplicity, these strategies extend SumZero by a fixed load-shifting schedule for all pure consumers of an EC and advise them to shift their loads accordingly.

During the simulation of the following scenarios to measure their performance, all consumers are wired (hardcoded, by overwriting their consumption profiles) to fully honor the EMS's load-shifting suggestions by adapting the "Controllable agent implementation" (described in 3.3).

EMS strategy: SumZeroWithConsShift50perc2h

Each pure consumer is wired to shift 50% of its energy consumption that would occur between 17:00–19:00, to 13:00–15:00 instead.

EMS strategy: SumZeroWithConsShift50perc4h

Each pure consumer is wired to shift 50% of its energy consumption which would occur between 16:00–20:00 to 12:00–16:00 instead.

Both load shifting windows were considered in a way where the direct usage of excess energy (and thus equilibrium) is increased within the community. This time window is known to be high demanding for the public grid, since production drops to zero while residential consumption is increased.

6.3 Evaluation

As a baseline for comparison, figure 6.2 shows the simulation results for EC *12esr12cons* when using the SumZero EMS strategy, as already discussed in case study 1. The colors of the following charts are also explained in chapter 4.5.1.

6.3.1 1-day Performance

Figure 6.2 shows a day of EC 12esr12cons that ends with an already excellent EC's avg. self-consumption of 99.65% and a reward of €14.40 due to a sunny summer day with a lot of excess energy sold to the central grid. By the end of the day, the EC's avg. SoC of the 12 ESRs is only 27.7%, because all the consumer could be satisfied by the EC's ESRs, between 17:00 and 24:00.

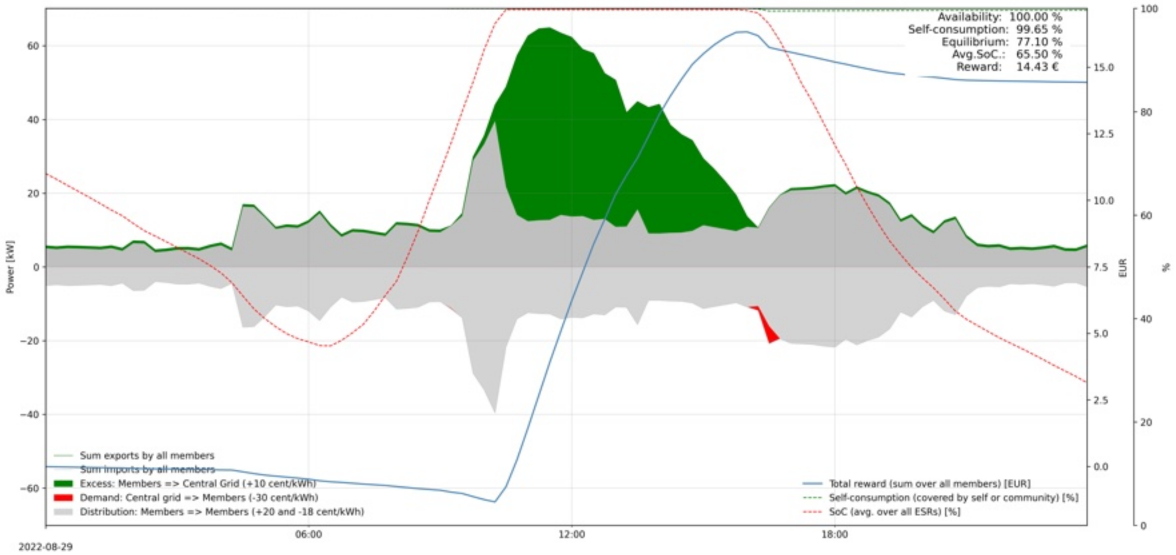


Figure 6.2: 1-day historical energy budget of EC *12esr12cons* with EMS strategy SumZero (without load shifting), used as baseline.

Figure 6.3 shows the same situation, but with simulated consumer load shift using SumZeroWithConsShift50perc2h: At the end of the day, less reward was achieved (€12.27) because about 10% of the excess energy produced (20 kWh) was not sold to the central grid. It was instead redistributed due to the load shifting strategy, resulting in about 20 kWh more energy available in the EC's ESRs, with 39.0% SoC by midnight. In addition to the higher SoC that will positively affect the self-consumption of EC in the following hours, in the future; another benefit is the slightly increased EC equilibrium (from 77.0% to 78.8%) making the EC more independent of central energy providers.

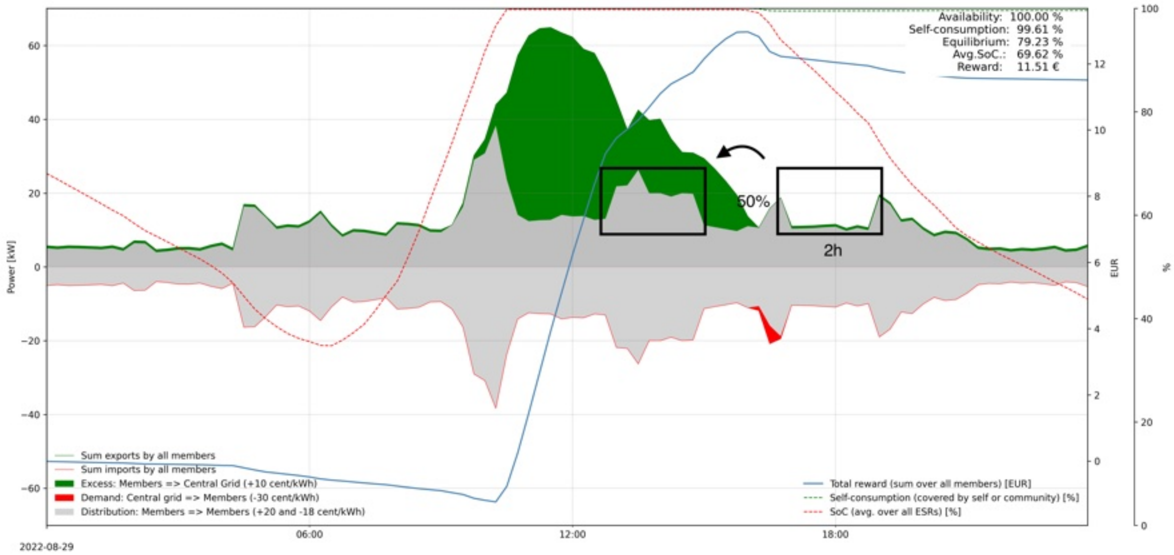


Figure 6.3: 1-day historical energy budget of EC *12esr12cons* with EMS strategy SumZeroWithConsShift50perc2h.

Figure 6.4 with SumZeroWithConsShift50perc4h further increases all positive effects: At the end of the day, the daily reward temporarily reduced to €11.21, but SoC reached 46.1% and equilibrium 79.9%, due to 34 kWh not sold but redistributed within the EC and thus also still available within the ESR.

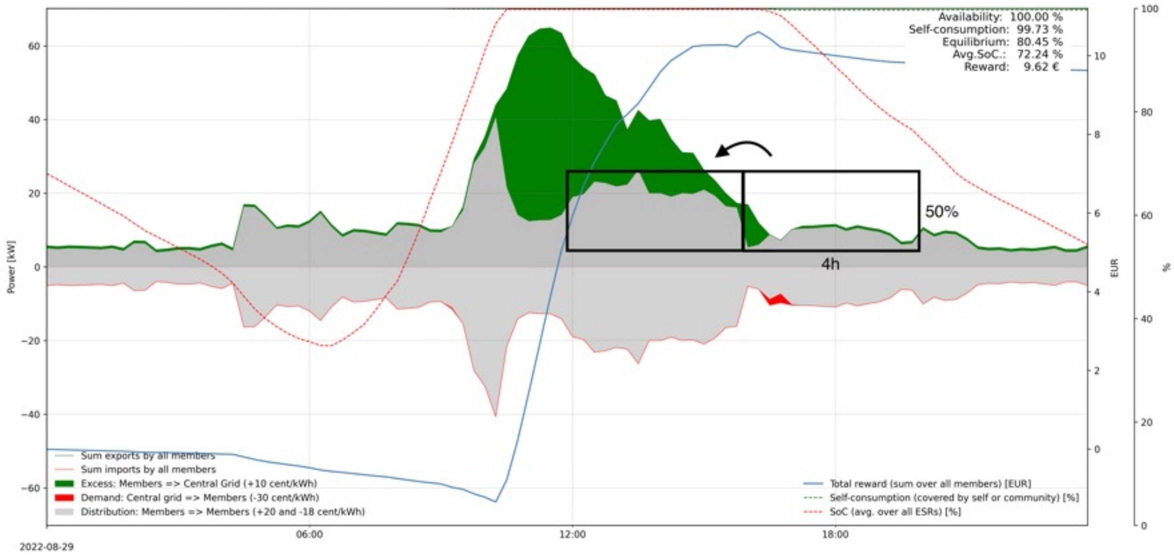


Figure 6.4: 1-day historical energy budget of EC *12esr12cons* with EMS strategy SumZeroWithConsShift50perc4h.

6.3.2 1-week Performance

When comparing the two EMS strategies SumZero (figure 6.5) with SumZeroWithConsShift50perc4h (figure 6.6) for EC *12esr12cons* in the 2 charts below, throughout the week, the positive effects of load shifting are magnified:

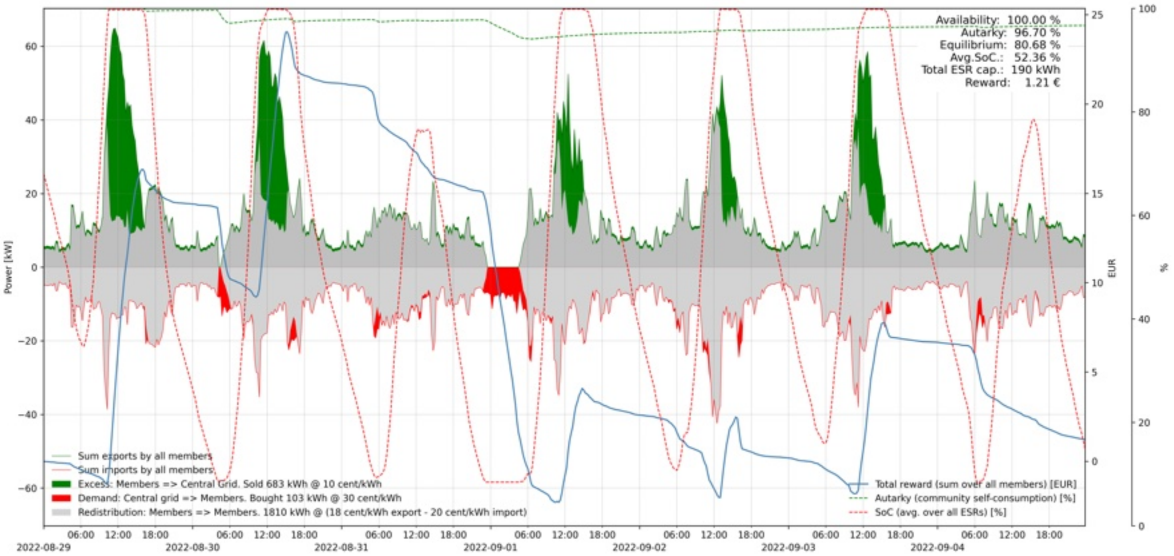


Figure 6.5: 1-week historical energy budget of EC *12esr12cons* with EMS strategy SumZero (without load shifting).

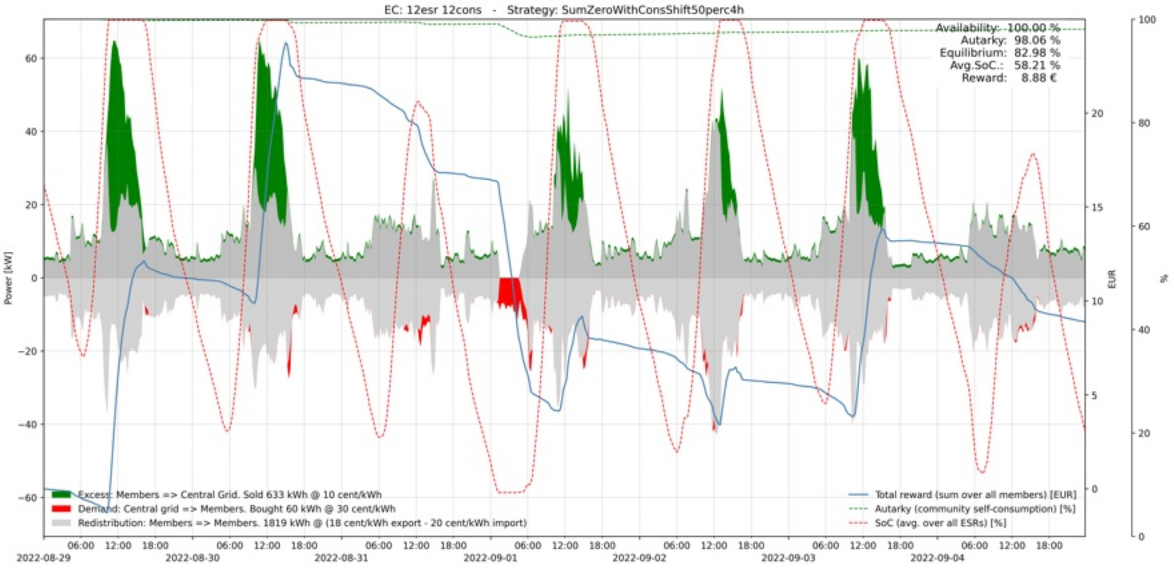


Figure 6.6: 1-week historical energy budget of EC *12esr12cons* with EMS strategy SumZeroWithConsShift50perc4h.

- Autarky increased from 96.70% to 98.06%
- Equilibrium increased from 80.68% to 82.98%
- Central grid imports decreased from 103 kWh to 60 kWh (along with decreased exports)
- Reward increased from €0.58 to €8.35

Furthermore, comparing SumZero with SumZeroWithConsShift50perc4h for *9esr15cons* (charts omitted) from the weekly perspective, results in similar (relative) performance improvements. This EC has 3 less ESRs and 3 more consumers, with a total ESR capacity of only 144 kWh (compared to 190 kWh for *12esr12cons*).

- Autarky increased from 84.41% to 87.02%
- Equilibrium increased from 66.70% to 72.84%
- Central grid imports decreased from 485 kWh to 404 kWh (along with decreased exports)
- Reward increased from €-156.12 to €-140.90

6.4 Discussion and results

As a result of this load shifting, the amount of centralized grid feed-in is reduced, since it is consumed by the community instead (reducing short-term reward but increasing long-term reward). Furthermore, the utilization of ESRs is increased, because the loads were shifted away from the evening time window, resulting in longer lasting ESR usage throughout the night until the next morning. This additionally increases the flexibility and security of supply for all individuals in the EC.

In general, EC consumers should be advised by an EMS to shift as much energy as possible (in this simulated case study: 50% in 2 specific time windows and 2 EC mixes) to periods of excess energy. This energy will then be harvested from the roof of a neighbor.

ECs and households that are open to a behavioral flexibility benefit from a higher economical and ecological effect.

The ongoing adaption of dynamic energy tariffs by society will introduce a major paradigm shift in the way energy is used and thought. Well-sized ESRs in households provide the best of both worlds: Profit from dynamic tariffs while having low-cost energy available on demand.

6.4.1 Outlook: Advanced EMS load scheduling

By implementing further, more intelligent EMS strategies that incorporate weather forecasts, individual member consumption forecasts, and dynamic price tariffs, the performance of the EC can be increased even further.

Local and central EMS can both simultaneously follow their strategies, but since the EPEX Spot [5] day-ahead prices are defined per country, this logic could also be implemented in the central EMS strategy.

It is important to mention that using and optimizing for dynamic energy tariffs (e.g. EV charging during low central grid prices) interferes with energy communities, especially ECs lacking a central EMS coordination (e.g. NoOp strategy), whereas smart energy communities (e.g. SumZero strategy) compensate for it.

Such an EMS strategy is able to outperform all individual HEMS, as a result of more data available and higher scheduling flexibility: The EMS has access to the information of the individual's EMS and at the same time the ability to distribute load shifting actions to the EC: Members could announce a prioritized list of energy tasks to their EC (e.g. charge my car until 08:00, house heating preferences) and let the central EMS strategy derive optimal actions for every member, to then start the respective task.

---

[5] `https://www.epexspot.com/en/market-data`

CHAPTER 7

Conclusion

7.1 Summary

This thesis provided a framework to demonstrate, that energy communities can be modeled as first-class citizens on distributed cloud operating systems such as Kubernetes, showing the effectiveness of smart energy communities with a central EMS over conventional (non-smart) energy communities.

With its three case studies, it has covered different domains, considering recent and ongoing research of those domains.

The first case study (Chapter 4) evaluated the performance of real households in simulated energy communities of different settings. The results showed that communities always outperform individuals and how the different community settings compare to each other, using energy data from 24 households over a time period of one week.

The second case study (Chapter 5) focused on the framework and Kubernetes itself, its benefits, and limitations. The distributed system was observed during its execution, with the conclusion that the high availability is less significant than initially expected. The results showed that the performance of the community in the case of the worst availability of members (0%) performs the same as the "individual" EMS strategy and that performance is directly proportional to availability.

The third case study (Chapter 6) used the framework to analyze the effect on the performance of the community consisting of households capable and willing to flexibly shift parts of their consumed energy into time windows that are beneficial to the community. It showed that members with higher behavioral flexibility improve their energy efficiency and cost savings even further.

In summary, this thesis aimed to combine this state-of-the-art research on the superiority energy communities, with Kubernetes as a state-of-the-art distributed cloud operating system, to fill identified gaps between those two domains: Plenty of research on different EC optimization strategies exists but is lacking publicly available software to model and control ECs in a near-real-world setting, and the openness to implement custom strategies and extensions. The framework provided by this thesis aims to provide a stepping stone for any individual to create and optimize smart energy communities that outperform communities without central EMS and thus provide benefits for all participants.

7.2 Research Questions

Following the results of the thesis, the research questions can be answered as follows.

7.2.1 RQ.1: What are the concrete challenges of energy communities?

- **Achieving high community self-sufficiency**
  ECs relief the already tense power utilization of distribution grids. The high self-sufficiency of the ECs implies high independence from the central grid and its fluctuations, made visible in volatile energy prices on the spot market. In order to achieve high self-sufficiency, a forecasting and matching algorithm is required per EC, that monitors and controls the energy usages of every member in real-time. This is important, especially during night times, bad weather, or holidays, where consumption is usually higher and can even lead to unexpected and thus unmet consumption peaks (e.g. Christmas, New Year's Eve). Energy peaks are the enemy of all grid operators, as they are costly and can lead to grid overloads and can only be addressed by reducing peak consumption (using load shedding) or by increasing power feed-in to cover them within the EC. Both methods were implemented and simulated by the framework and have been discussed as part of this thesis' case studies.

- **Finding an optimal EC member mix**
  The better the mix of producers and consumers in an EC, and their flexibility and adaptability time-wise, the higher the self-sufficiency and therefore the cost savings. Participants can also be powerhouses like wind parks or hydroelectric plants. Such non-solar producers are beneficial to cover base loads and bridge nightly energy gaps. This thesis does not further dive into the area of finding optimal EC compositions but exemplarily simulates different mixes of ECs with different consumer and producer count, in order to compare their KPIs.

- **Clearing and billing**
  The 15-minute energy values are required from all members of all their energy providers, to do the clearing (usually on a monthly basis). This is challenging because energy providers are obliged to provide their own ICT to customers and ECs to obtain historical energy data from all smart meters. Furthermore, the energy data sets are prone to later corrections and have several levels of settlement.

- **Onboarding and administration of EC and members**
  Energy communities must be registered, as well as all their members (through their counting point numbers). Memberships may change at any time, and members can participate in multiple ECs at the same time (proportionally or onion-layered with regard to residual energy not covered by the upper layer).

- **Operating a cloud-based EMS**
  Technical challenges arise for smart energy community frameworks as presented in this thesis, where energy data is required in real-time (smart meters provide delayed data only). They must be maintained and members must be registered and connected to the EMS.

7.2.2 RQ.2: How can smart energy communities be modeled, managed, controlled, and optimized through service-oriented architectures?

- **Through a SOA like the SmartGrid framework**

  This thesis demonstrates the usage of Kubernetes as SOA to model, manage, control and optimize a multitude of energy communities with a multitude of members, independently. It therefore leverages basic K8s entities, e.g. isolation and networking, to interconnect members to a central EMS, or high available deployment strategies, for updating EC structures, memberships, and even software features. It ships with an operator and custom resource definitions, to make energy community management and optimization first-class in the context of K8s. As a result, this SOA framework can be deployed on any hyperscaler with low effort (3.9.1). The architecture is explained in detail in 3.3 and the framework is deployed and used throughout all case studies to obtain the presented results.

7.2.3 RQ.3: Is a central EMS and data exchange on momentary energy demand and excess between participants required for optimal energy sharing?

- In general **no**, but for optimal energy sharing **yes**.

  As EC members will behave differently than expected/planned over time: As a matter of chance, not all amounts of energy will be covered by the community over time, and any energy demanded by the central grid (due to no or suboptimal planning of the EC) must be accounted for, since all residual energy gets invoiced by the individual's energy provider as usual.

  A central EMS is able to prevent/reduce those cases, as demonstrated in case study 1 using SumZero vs. NoOp) 4.6: The real-time cloud EMS outperforms any slower control-loop, because it regulates towards energy self-sufficiency (directly proportional to reward) compared to "doing nothing". Such a central EMS can furthermore be extended, to forecast the communities profile to buy and sell energy autonomously for the EC without requiring any further software or hardware changes on-site, but just using a pure cloud-side software update.

But practicability in the real world is another story: In larger energy communities, the creation and forecasting of static consumption load profiles might already be good enough, making a cloud-based EMS less relevant or even obsolete, compared to its maintenance and member onboarding overhead.

7.3 Future Work

As future work, the performance of an EMS based on real-time as proposed should be compared against energy communities that make use of fixed / static consumption load profiles. Such profiles can be trained and forecasted from the historical smart-meter data of the EC member mix, which could already provide results good enough for the EC. The forecast error could be used as the KPI to compare it with the performance of a real-time EMS.

For future work regarding this framework, I suggest verifying (i) the SumZero strategy with its active target power control in a real-world test bed of households willing to participate and (ii) for a wider time span.

The simulation conducted by this framework had a rather simple model of member behavior regarding active target power control (all members behave the same, always trying to obey as longs as physically possible), which limits the outcome of case studies 1 and 3. A real-world test is expected to improve the positive effects of the central cloud EMS further, since real-world members are expected to react stronger to community excess when incentivized, by shifting their loads more dynamically, and thus decreasing future central grid consumption (as discussed in case study 3). Ultimately, members can improve their local decision making using the additional information provided by the central EMS.

In addition, further EMS strategies and features can be implemented that include additional data such as weather forecast, historical community consumption behavior, and decisions, improving the SumZero strategy and the components of the framework. This will further decrease the energy demand from the central grid, providing additional benefits to the community, members, and the environment.

More and more residential and commercial sites are becoming intelligent consumers or prosumers participating in smart energy communities. This noticeably increases their individual reward as well as the overall self-sufficiency of the communities.

–

The central power grid, long a fixture of our energy infrastructure, is already undergoing an irreversible transition toward decentralization, marking a transformative shift with far-reaching implications for the future of energy systems and our society.

CHAPTER 8

Appendix A: Individual charts

The following 24 charts show all the individual's power values for PV production, their consumption, grid meter values, as well as ESR SoC and power values, throughout the one-week timespan at a 15-minute resolution. All EC compositions throughout this thesis are comprised of those 24 sites and used as-is for NoOp and Individual strategies. For ECs consisting of pure consumers, the production power and ESR capacity were set to 0 during the simulation, throughout the week, and the grid meter values were overridden to match consumption. The source measurement data points are provided in chapter 9.

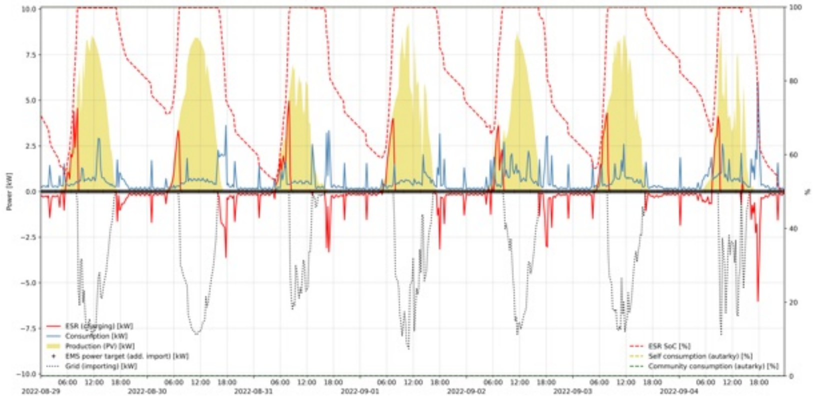


Figure 8.1: ESR01: 13.2 kWh ESR (net), 17.3 kW max grid power.

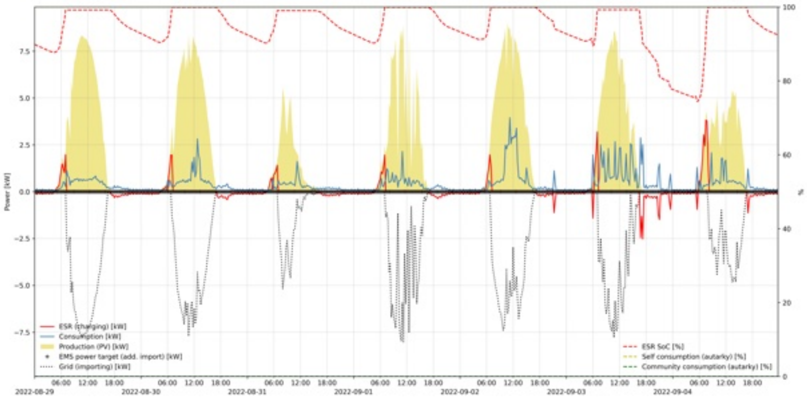


Figure 8.2: ESR02: 17.8 kWh ESR (net), 10.0 kW max grid power.

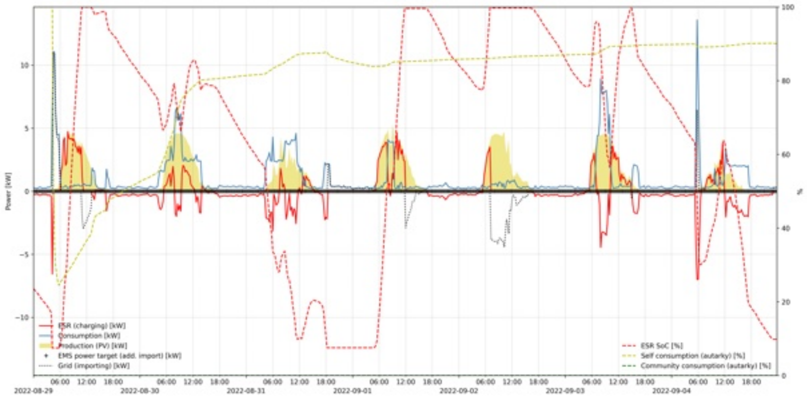


Figure 8.3: ESR03: 17.6 kWh ESR (net), 10.0 kW max grid power.

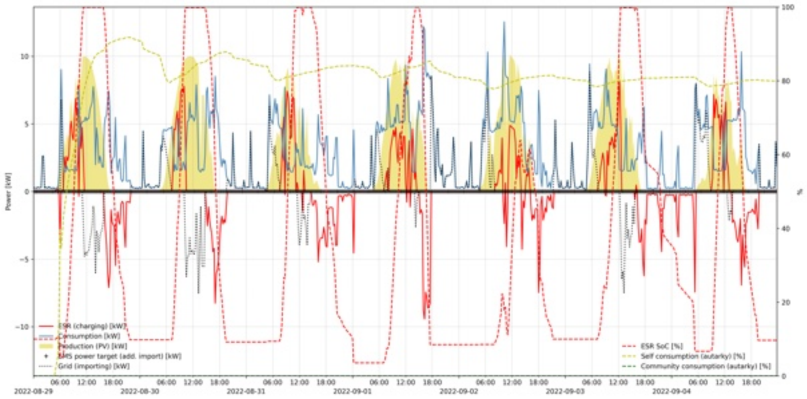


Figure 8.4: ESR04: 17.8 kWh ESR (net), 25.0 kW max grid power.

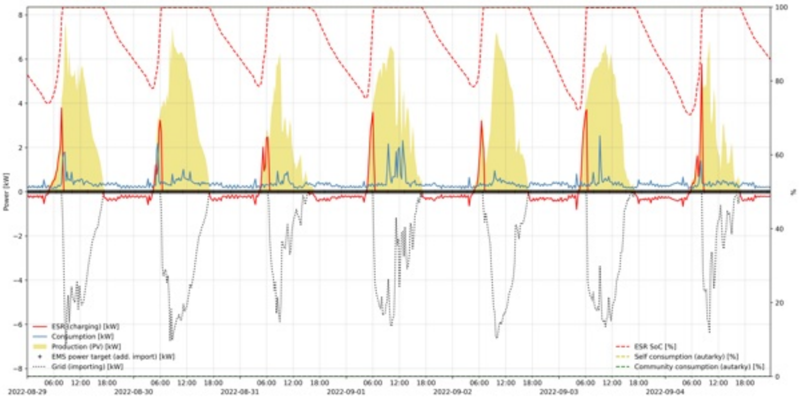


Figure 8.5: ESR05: 13.9 kWh ESR (net), 24.0 kW max grid power.

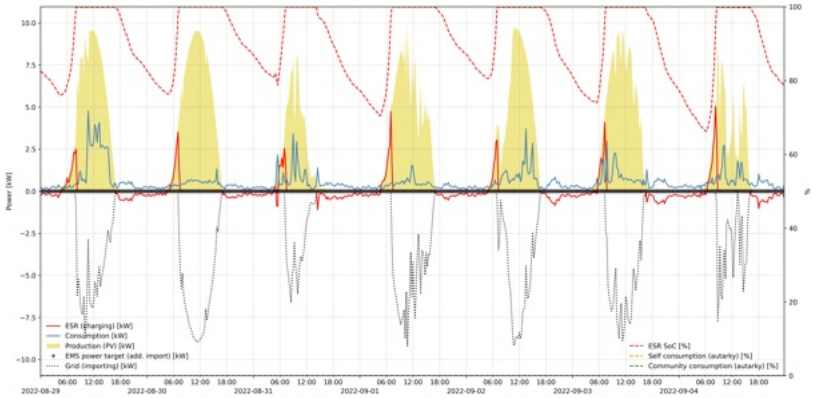


Figure 8.6: ESR06: 14.2 kWh ESR (net), 24.0 kW max grid power.

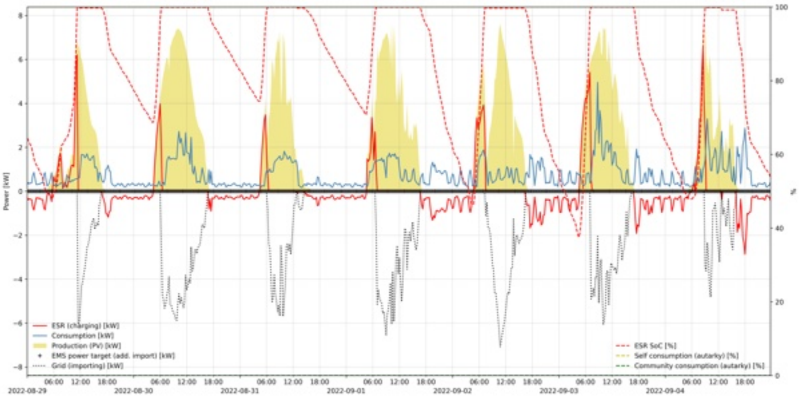


Figure 8.7: ESR07: 14.2 kWh ESR (net), 24.0 kW max grid power.

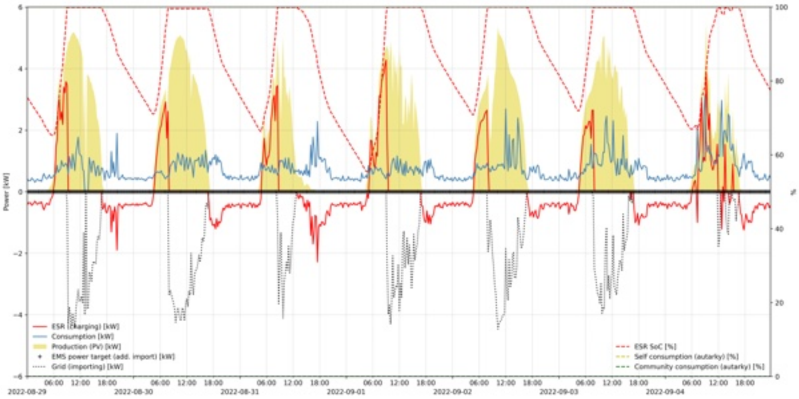


Figure 8.8: ESR08: 21.3 kWh ESR (net), 25.0 kW max grid power.

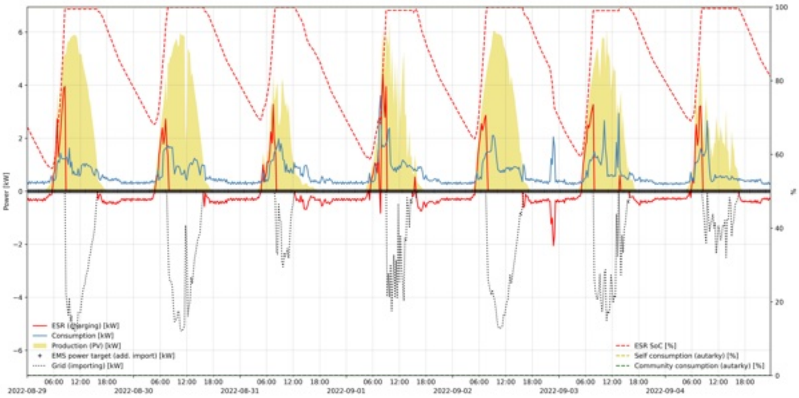


Figure 8.9: ESR09: 14.2 kWh ESR (net), 24.0 kW max grid power.

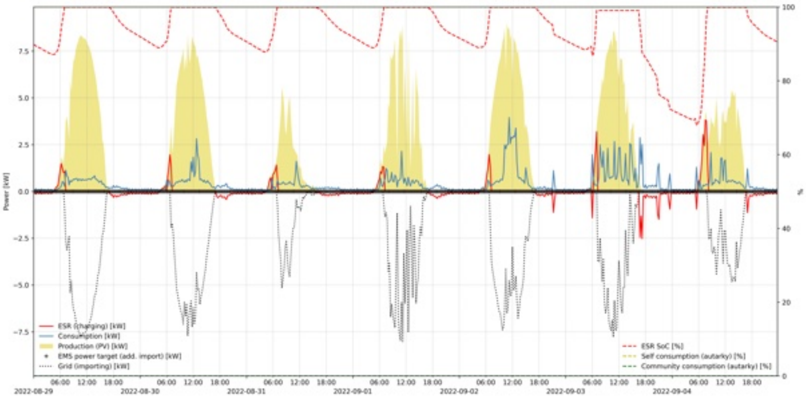


Figure 8.10: ESR10: 21.3 kWh ESR (net), 24.0 kW max grid power.

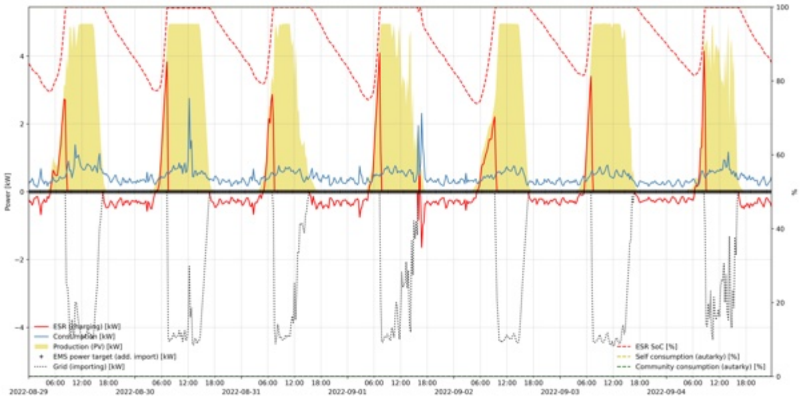


Figure 8.11: ESR11: 17.8 kWh ESR (net), 10.0 kW max grid power.

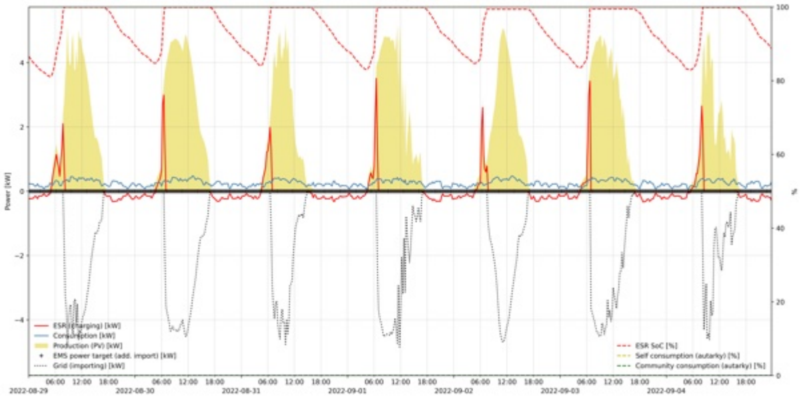


Figure 8.12: ESR12: 14.2 kWh ESR (net), 17.3 kW max grid power.

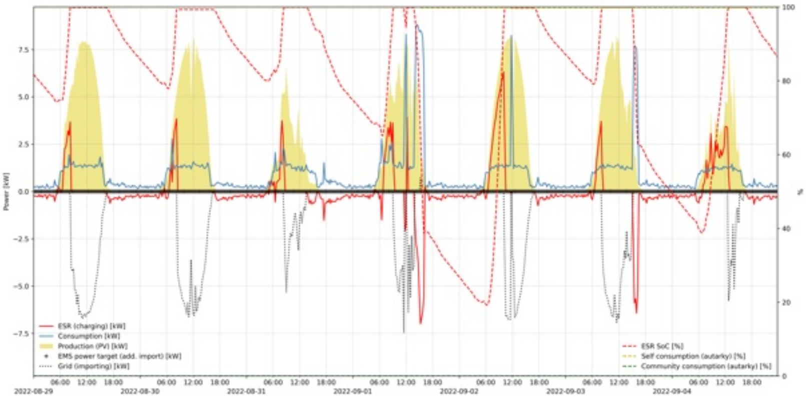


Figure 8.13: ESR13: 17.8 kWh ESR (net), 10.0 kW max grid power.

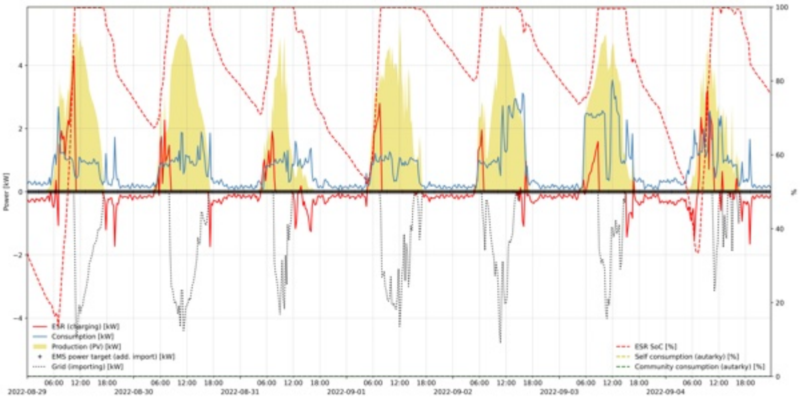


Figure 8.14: ESR14: 9.6 kWh ESR (net), 24.0 kW max grid power.

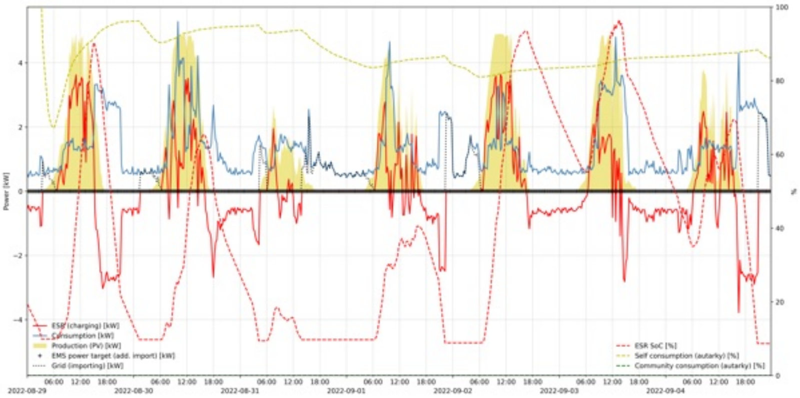


Figure 8.15: ESR15: 21.6 kWh ESR (net), 20.0 kW max grid power.

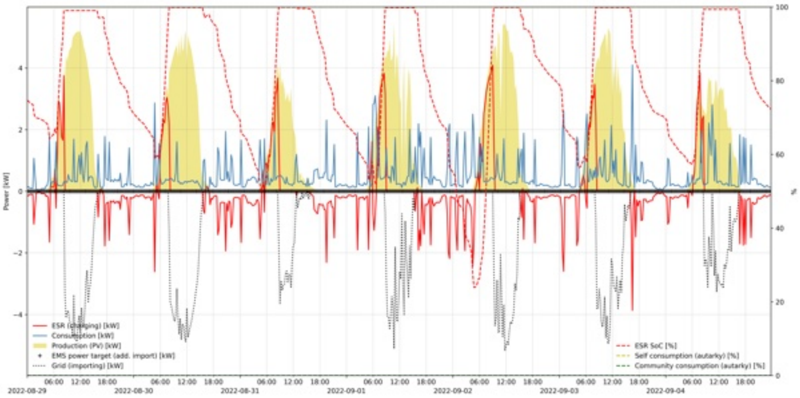


Figure 8.16: ESR16: 12.0 kWh ESR (net), 24.0 kW max grid power.

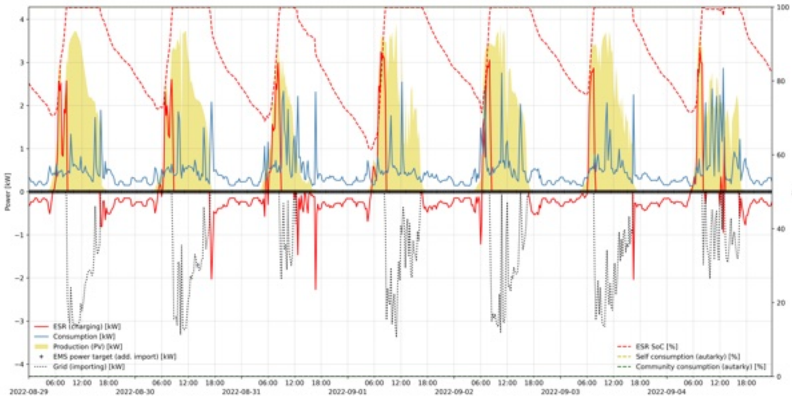


Figure 8.17: ESR17: 14.2 kWh ESR (net), 24.0 kW max grid power.

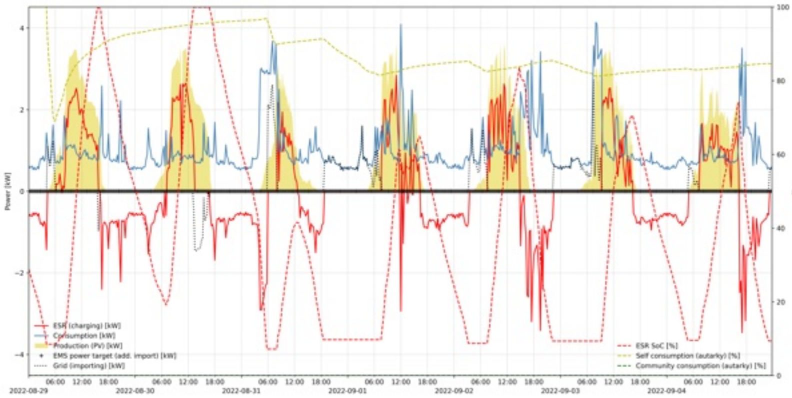


Figure 8.18: ESR18: 14.2 kWh ESR (net), 24.0 kW max grid power.

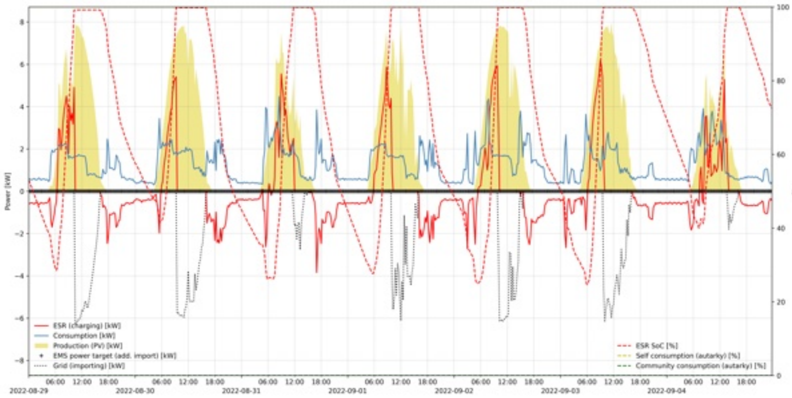


Figure 8.19: ESR19: 17.8 kWh ESR (net), 24.0 kW max grid power.

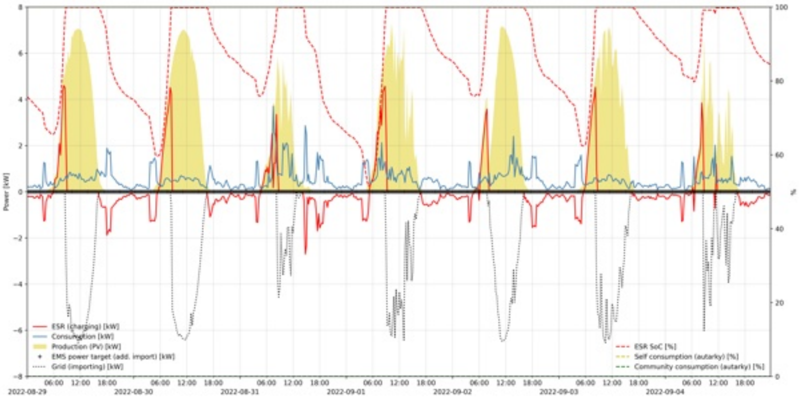


Figure 8.20: ESR20: 17.8 kWh ESR (net), 24.0 kW max grid power.

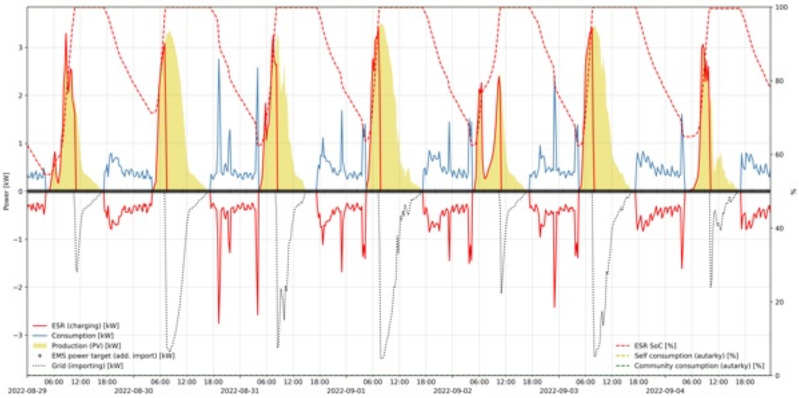


Figure 8.21: ESR21: 17.0 kWh ESR (net), 24.0 kW max grid power.

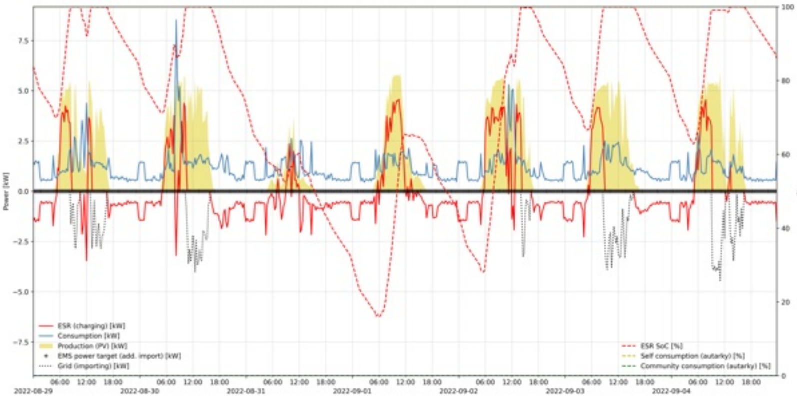


Figure 8.22: ESR22: 31.9 kWh ESR (net), 24.0 kW max grid power.

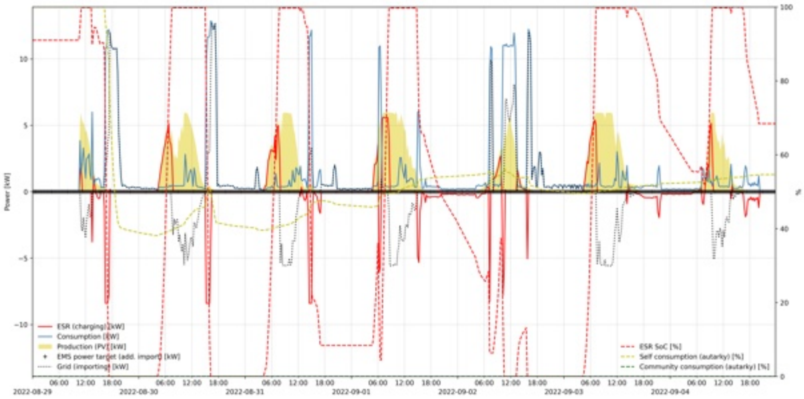


Figure 8.23: ESR23: 9.6 kWh ESR (net), 12.0 kW max grid power.

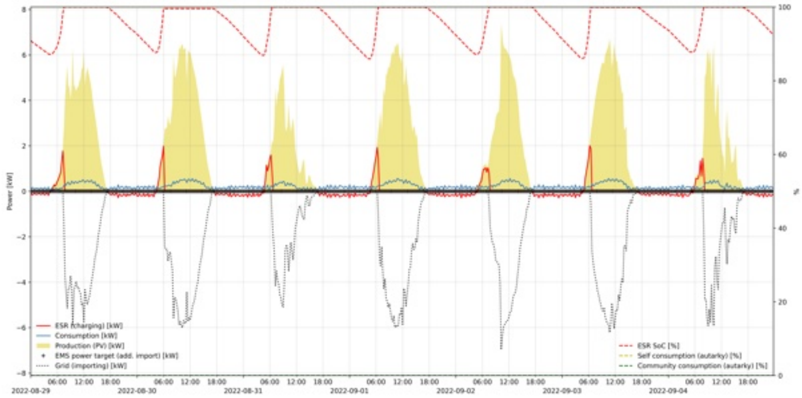


Figure 8.24: ESR24: 14.2 kWh ESR (net), 22.0 kW max grid power.

CHAPTER 9

Appendix B: Source Code

The following links refer to the source code of the framework with all its components, as well as scripts, scenario files, source measurements and data points, and simulation results.

9.0.1 SmartGrid Framework implementation and source measurements

- https://gitlab.com/ec-thesis/smartgrid

9.0.2 SmartGrid Kubernetes Operator, build and deployment scripts

- https://gitlab.com/ec-thesis/operator

9.0.3 Simulation scenario files and results

- https://gitlab.com/ec-thesis/scenario

List of Figures

List of Tables

Acronyms

CRD Custom Resource Definition.

DER Distributed energy resource.

DR Demand response.

EC Energy community.

EMS Energy management system.

ESR Energy storage resource.

EV Electric vehicle.

GA Genetic algorithm.

GHG Greenhouse gas.

HEMS Home energy management system.

ICT Information and communications technology.

IOE Internet of Energy.

K8s Kubernetes.

KPI Key performance indicator.

LC Local controller.

LP Linear programming.

MAS Multi-agent system.

MG Microgrid.

MGCC Microgrid central controller.

MILP Mixed integer linear programming.

MPC Model predictive control.

NN Neural network.

PCC Point of common coupling.

PV Photovoltaic appliance.

RER Renewable energy resource.

SG Smart grid.

SOA Service-oriented architecture.

SOC State of charge.

SSR Self-sufficiency ratio.

VPP Virtual power plant.

Bibliography

[AAPG+17] Diego Arcos-Aviles, Julio Pascual, Francesc Guinjoan, Luis Marroyo, Pablo Sanchis, and Martin P Marietta. Low complexity energy management strategy for grid profile smoothing of a residential grid-connected microgrid using generation and demand forecasting. *Applied energy*, 205:69–84, 2017.

[Arn11] George W Arnold. Challenges and opportunities in smart grid: A position article. *Proceedings of the IEEE*, 99(6):922–927, 2011.

[BBB+06] Yves Bamberger, João Baptista, Ronnie Belmans, Bern Michael Buchholz, Maher Chebbo, José Luis Del Valle Doblado, Venizelos Efthymiou, Livio Gallo, Edmund Handschin, Nickolas Hatziargyriou, et al. Vision and strategy for europe's electricity networks of the future: European technology platformsmartgrids. 2006.

[CDC18] Zheyuan Cheng, Jie Duan, and Mo-Yuen Chow. To centralize or to distribute: That is the question: A comparison of advanced microgrid management systems. *IEEE Industrial Electronics Magazine*, 12(1):6–24, 2018.

[CSET21] Stefan PM Chantrel, Arne Surmann, Thomas Erge, and Jessica Thomsen. Participative renewable energy community—how blockchain-based governance enables a german interpretation of red ii. *Electricity*, 2(4):471–486, 2021.

[DSIS+21] Maria Luisa Di Silvestre, Mariano Giuseppe Ippolito, Eleonora Riva Sanseverino, Giuseppe Sciumè, and Antony Vasile. Energy self-consumers and renewable energy communities in italy: New actors of the electric power systems. *Renewable and Sustainable Energy Reviews*, 151:111565, 2021.

[ene] Umsetzung von Erneuerbaren-Energie-Gemeinschaften (EEG) durch das Erneuerbaren-Ausbau-Gesetzespaket (EAG-Paket) Website. `https://www.e-netze.at/Strom/Energiegemeinschaften/Default.aspx`. abgerufen am 12.10.2023.

[eud] EU directive on the promotion of the use of energy from renewable sources Website. https://eur-lex.europa.eu/legal-content/DE/TXT/?uri=CELEX:32018L2001. abgerufen am 1.10.2023.

[fal] The Eight Fallacies of Distributed Computing Website. https://nighthacks.com/jag/res/Fallacies.html. abgerufen am 11.09.2023.

[GSK+11] Vehbi C Gungor, Dilan Sahin, Taskin Kocak, Salih Ergut, Concettina Buccella, Carlo Cecati, and Gerhard P Hancke. Smart grid technologies: Communication technologies and standards. *IEEE transactions on Industrial informatics*, 7(4):529–539, 2011.

[GTG+19] F Grasso, G Talluri, A Giorgi, A Luchetta, and L Paolucci. Peer-to-peer energy exchanges model to optimize the integration of renewable energy sources: The e-cube project. *Energ. Elettr. Suppl. J*, 96:1–8, 2019.

[gur] Gurobi Optimizer Reference Manual Website. https://www.gurobi.com/documentation/10.0/refman/index.html.

[Hat14] Nikos Hatziargyriou. *Microgrids: architectures and control*. John Wiley & Sons, 2014.

[HBLN12] Kazi M Moyeenull Huq, ME Baran, Srdjan Lukic, and Otsebele E Nare. An energy management system for a community energy storage system. In *2012 IEEE Energy Conversion Congress and Exposition (ECCE)*, pages 2759–2763. IEEE, 2012.

[Ju21] Chengquan Ju. Hierarchically coordinated energy management for a regional multi-microgrid community. *arXiv preprint arXiv:2102.03745*, 2021.

[JXG13] Quanyuan Jiang, Meidong Xue, and Guangchao Geng. Energy management of microgrid in grid-connected and stand-alone modes. *IEEE transactions on power systems*, 28(3):3380–3389, 2013.

[KFK22a] Felix Knorr, Thomas Frühwirth, and Wolfgang Kastner. Framework for the design and automatic deployment of smart grid applications. *Electric Power Systems Research*, 212:108531, 2022.

[KFK22b] Felix Knorr, Thomas Frühwirth, and Wolfgang Kastner. Functional smart grid application development. In *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–4. IEEE, 2022.

[KNHH14] Nipendra Kayastha, Dusit Niyato, Ekram Hossain, and Zhu Han. Smart grid sensor data collection, communication, and networking: a tutorial. *Wireless communications and mobile computing*, 14(11):1055–1087, 2014.

[KS19] Peng-Yong Kong and Yujae Song. Joint consideration of communication network and power grid topology for communications in community smart grid. *IEEE Transactions on Industrial Informatics*, 16(5):2895–2905, 2019.

[LR11] NWA Lidula and AD Rajapakse. Microgrids research: A review of experimental microgrids and test systems. *Renewable and Sustainable Energy Reviews*, 15(1):186–202, 2011.

[MAHS22] Daniel Masendorf, Raad Alsayyed, Pia Henzel, and Thorsten Schlösser. Smecon-box–secure ev charging using the "fnn steuerbox". 2022.

[MRMDP22] Alyssa Diva Mustika, Remy Rigo-Mariani, Vincent Debusschere, and Amaury Pachurka. A two-stage management strategy for the optimal operation and billing in an energy community with collective self-consumption. *Applied Energy*, 310:118484, 2022.

[NZL18] Sibo Nan, Ming Zhou, and Gengyin Li. Optimal residential community demand response scheduling in smart grid. *Applied Energy*, 210:1280–1289, 2018.

[ope] Open Source Energy Management System Website. `https://github.com/OpenEMS/openems`.

[PBSM15] Julio Pascual, Javier Barricarte, Pablo Sanchis, and Luis Marroyo. Energy management strategy for a renewable-based residential microgrid with generation and demand forecasting. *Applied Energy*, 158:12–25, 2015.

[RGH+20] Md Mamun Ur Rashid, Fabrizio Granelli, Md Hossain, Md Alam, Fahad Saleh Al-Ismail, Rakibuzzaman Shah, et al. Development of cluster-based energy management scheme for residential usages in the smart grid community. *Electronics*, 9(9):1462, 2020.

[ris] Gesamte Rechtsvorschrift für Erneuerbaren-Ausbau-Gesetz, Fassung vom 06.01.2024 Website. `https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=20011619`. abgerufen am 10.1.2024.

[RPD+14] AJ Dinusha Rathnayaka, Vidyasagar M Potdar, Tharam Dillon, Omar Hussain, and Samitha Kuruppu. Goal-oriented prosumer community groups for the smart grid. *IEEE Technology and Society Magazine*, 33(1):41–48, 2014.

[SLK+15] Nah-Oak Song, Ji-Hye Lee, Hak-Man Kim, Yong Hoon Im, and Jae Yong Lee. Optimal energy management of multi-microgrids with sequentially coordinated operations. *Energies*, 8(8):8371–8390, 2015.

[SO21] Fábio Silva and Brian O'Regan. An innovative smart grid framework for integration and trading. In *E3S web of conferences*, volume 294, pages 1–5. EDP Sciences, 2021.

[SWL13] Manuela Sechilariu, Baochao Wang, and Fabrice Locment. Building-integrated microgrid: Advanced local energy management for forthcoming smart power grid communication. *Energy and Buildings*, 59:236–243, 2013.

[SZC⁺20] Mark Stefan, Paul Zehetbauer, Stephan Cejka, Franz Zeilinger, and Gregor Taljan. Blockchain-based self-consumption optimisation and energy trading in renewable energy communities. In *CIRED 2020 Berlin Workshop (CIRED 2020)*, volume 2020, pages 371–374. IET, 2020.

[uRFK⁺19] Zia ur Rehman, Mulika Fatima, Ahmad Khan, Sajid Shah, Raja Wasim Ahmad, Ahmad Din, and Attiqa Rehman. Energy optimization in a smart community grid system using genetic algorithm. *International Journal of Communication Systems*, page e4265, 2019.

[WHK20] Martin Warneryd, Maria Håkansson, and Kersti Karltorp. Unpacking the complexity of community microgrids: A review of institutions' roles for development of microgrids. *Renewable and Sustainable Energy Reviews*, 121:109690, 2020.

[ZEB18] Muhammad Fahad Zia, Elhoussin Elbouchikhi, and Mohamed Benbouzid. Microgrids energy management systems: A critical review on methods, solutions, and prospects. *Applied energy*, 222:1033–1055, 2018.