

RESEARCH

Open Access



Lessons learned from CPES co-simulation with distributed, heterogeneous systems

Cornelius Steinbrink^{1*}, Christian Köhler², Marius Siemonsmeier³ and Thorsten van Ellen⁴

From The 7th DACH+ Conference on Energy Informatics
Oldenburg, Germany. 11-12 October 2018

* Correspondence: cornelius.steinbrink@offis.de

steinbrink@offis.de

¹R&D Division Energy, OFFIS –
Institute for Information
Technology, Oldenburg, Germany
Full list of author information is
available at the end of the article

Abstract

The increasing integration of distributed renewable energy resources into the power grid calls for employment of information and communication technology, transforming the grid into a cyber-physical energy system with new options for stable and optimized control. In order to evaluate and validate new control technologies, test systems are necessary. When the future extensibility of an approach is to be tested, laboratory and field tests reach their limits. Instead, simulation-based testing is required, like co-simulation, which allows the reuse of pre-existing simulation components. However, some co-simulation approaches designed for generic applicability tend to ignore certain setup characteristics like the need for remote coupling or exchange of complex data. This paper presents a co-simulation case study with distributed, heterogeneous simulation components. Challenges are discussed and it is shown how the framework MOSAIK helps to bridge the gap between special interfacing requirements and high system usability.

Keywords: Co-simulation, CPES, Distributed simulation, Heterogeneous systems

Introduction

The integration of an increasing amount of information and communication technology (ICT) into the classical power grid transforms it into a cyber-physical energy system (CPES) with novel monitoring and control capabilities. Such an extension is necessary to exploit formerly untapped flexibilities in the system and thus compensate for newly arising fluctuations in the energy supply due to renewable energy generation. Various approaches have already been proposed and tested to utilize flexibilities in energy generation or consumption for long-term planning or congestion handling in the energy supply (e.g. (Strbac, 2008; Ruiz et al., 2009)). An alternative approach is the prediction and subsequent avoidance of congestions. This is the goal of the research project *Proactive Distribution Grid* (German: “Proaktives Verteilnetz”, PAVN).

The basic idea of the PAVN project is to employ state estimation and prognosis algorithms (Coster et al., 2017) in the distribution grid operation. Furthermore, the grid operator acquires flexibilities via contracts with owners of distributed energy resources (DER) in their system. Consequently, DER flexibilities may be employed to avoid predicted congestions before they actually occur.

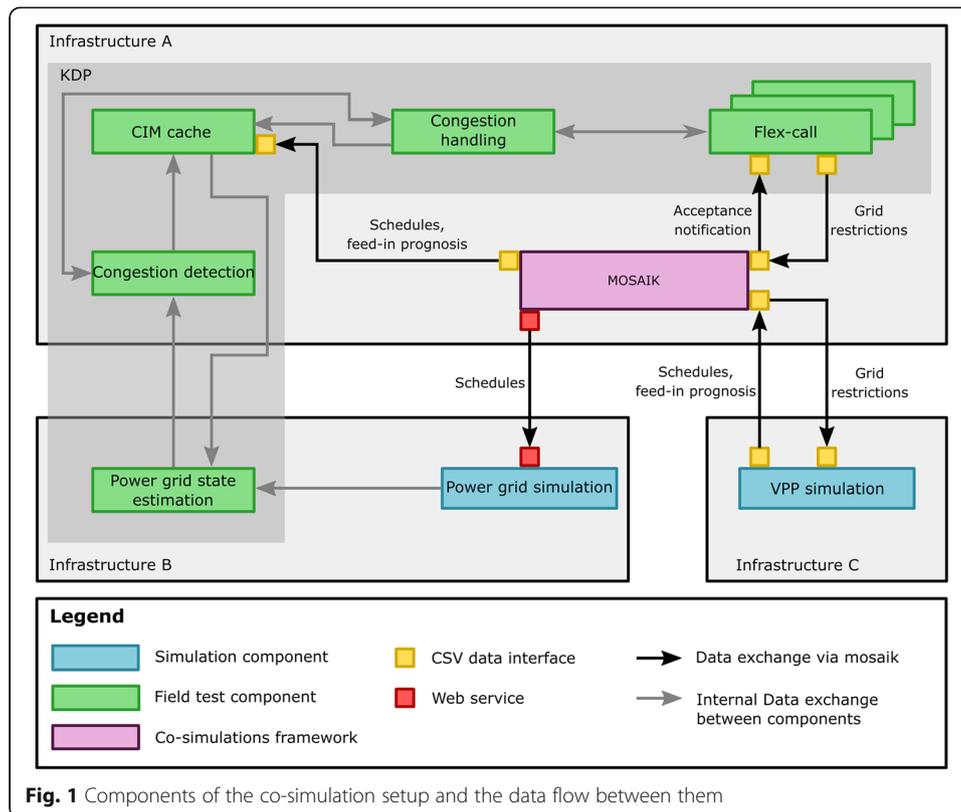
The project concept has been evaluated with a field test, on the one hand. On the other hand, a simulation setup has been established to test the usability of the approach in future systems with strongly increasing numbers of DER and new types of consumers and storages. Since several simulation systems have already been present in the project consortium, the approach of co-simulation has been chosen. This means that independently developed software tools simulating parts of the system under test are coupled via data exchange during runtime to establish a coordinated simulation of the complete system. The big advantage of co-simulation lies in the fact that additional modeling effort is avoided and existing expertise is reused in the form of validated simulators. In the PAVN project a dedicated co-simulation framework, MOSAIK (Schütte & Sonnenschein, 2012; Büscher et al., 2014), has been employed to utilize its generic algorithms for data exchange coordination. The employed simulation components, however, have been highly heterogeneous in structure which may lead to problems for generic co-simulation approaches (e.g. (Palensky et al., 2017)). In such a case, direct coupling of components (ad-hoc coupling) may be the easier solution. Nevertheless, the PAVN project attempts to demonstrate that generic co-simulation may also work for heterogeneous application cases, given the employed framework provides enough flexibility.

Co-simulation system

The PAVN field test system involves an actual distribution grid, including connected consumers and DER, as well as an ICT infrastructure to realize the control approach described above. The associated co-simulation setup, on the other hand, consists of the co-simulation framework MOSAIK and three complex simulation components that are connected to it: a virtual power plant (VPP) simulator, a power system simulation tool, and a communication and service platform (German: “Kommunikations- und Dienstleistungsplattform”, KDP). The power system simulation tool again consists of two units: a power grid simulator that is connected to MOSAIK, and a grid state estimation algorithm that is employed by the KDP.

The VPP simulator contains the set of modeled DER that calculate power generation and consumption schedules. These schedules are provided to the power grid simulation, on the one hand, and to the KDP on the other hand. During each simulation step, the power grid simulator uses the current values of the DER schedules to calculate the power flow for the given simulation time. A subset of the calculated values is provided to the state estimation, representing measurement values. The KDP then provides schedules to the state estimation and activates a congestion prediction. If a congestion is about to occur, the KDP provides suggestions for flexibility activations to the VPP simulator. As a result, the VPP simulator adjusts its schedules for the next time step and the whole chain starts over again.

Figure 1 gives an overview of the co-simulation system with its different components and the data flow between them. There are several things to note for this setup. For one, only the VPP and power grid components can actually be described as simulators since they implement abstract models of real world systems. The KDP, on the other hand, is an ICT system that is functionally identical in the simulation and the field test environment with little abstraction, integrating only minor changes for better controllability in the co-simulation context. Furthermore, Fig. 1 illustrates that the co-simulation setup is distributed over three computation infrastructures. The KDP as well as the MOSAIK framework are executed on virtual machines in one laboratory environment while the power



grid and the VPP simulation run in two other geographically separated environments with remote connections to the first environment. For the overall data exchange it has to be noted that strongly different technologies are employed: the power grid simulation is connected to MOSAIK via a web service while the other components interact with the framework via the exchange of CSV files. Finally, it can be seen that the ICT system is divided into several subsystems that partially run in different infrastructures and are directly connected with each other so that MOSAIK does not manage the complete data exchange in the setup.

Components and interfaces

The CPES co-simulation framework MOSAIK has been developed especially with two major principles in mind: the integration of black box simulators and flexible configuration of simulation setups. To realize this, MOSAIK provides a set of application programming interfaces (API). The so-called *Component-API* allows the integration of new simulation components into the co-simulation environment by providing a description model for the component's variables as well as a set of interface functions. It is comparable to the *Functional Mock-up Interface* (FMI) standard (Blochwitz et al., 2011), but more simplified, specifying only the minimal set of functions a simulator needs to provide to participate in a co-simulation. For the creation of a complete co-simulation setup, MOSAIK provides the so-called *Scenario-API*. This Python library provides a slim set of commands that may be used to write a so-called scenario script that will initialize and configure the selected simulation components, connect them with one another based on the data they provide and expect, and execute the actual co-simulation process.

The VPP simulation is part of a methodology which has been developed at the Institute of Power Systems and Power Economics (IAEW) to determine the costs of flexibility provision for congestion management in distribution grids' flexibility options, e. g. generation, demand and storage units. This methodology involves simulation of power generation dispatch, marketing of flexibility options, calculation of nodal generation and load situations, as well as consideration of grid restrictions. Within the PAVN co-simulation setup, however, only the power generation dispatch simulation is used to determine the power generation and consumption schedules in a rolling configuration. The VPP simulation thus provides DER schedules as output data and expects grid restriction information as input, which is used to recalculate the schedules for the next time step. Both, input as well as output data sets, are multi-dimensional. Therefore, the simulator has been implemented to accept and provide the data in the form of CSV files. The communication between MOSAIK and the VPP simulator uses the cloud computing service *Sciebo*. For each simulation time step, the VPP simulator uploads a file containing the schedules onto the Sciebo cloud and downloads a file with grid restrictions from it by means of the client-server software ownCloud (The ownCloud developers, 2018). The file name of each exchanged file consists of specifiers indicating the associated data (schedules or grid restrictions) and the currently simulated point in time as a string following a pre-defined time stamp format. This way, both MOSAIK and the VPP simulator can regularly check the Sciebo cloud for file names containing their expected data specifiers and time stamps. In other words, temporal synchronization of the remote VPP simulation with MOSAIK is established with synchronous file exchange via a shared folder on a web server.

The power system simulation is based on the Venios Energy Platform (VEP). It is based on the latest Microsoft technologies in the field of public- and private-cloud. Data management is achieved via a powerful NoSQL database. The VEP modules employed in the co-simulation provide the basic power flow calculation capability, which consists of a Newton Raphson module, the grid structure module as well as the models of attached consumers and providers. For more information on the concept of cloud-based power flow calculation see (Albrecht, 2014). The MOSAIK coupling has been implemented via a simple simulation state machine. State transitions and data transfer are realized via a JSON data structure that can be exchanged through a web socket channel. In contrast to the other presented interfaces, the MOSAIK-VEP coupling exchanges mostly scalar data values. However, more complex prognosis data is also integrated into the JSON exchange structures encoded as byte strings. VEP not only calculates the grid state for a triggered time step, but also runs the grids state estimation and flexibility calculation for the next several simulated hours. Triggering the simulation step by MOSAIK calls this module too. Afterwards the data is stored within the NoSQL database. The KDP access the stored data by calling a REST API. The same REST API is used within the field test setup of PAVN.

The communication and service platform, KDP, addresses the needs of intercompany cooperation in the energy transition context (e.g. (Expert Group 3 - Regular Recommendations for Smart Grids Deployment, 2013)). The transition to renewable energies requires the use of a plethora of new technologies and specialists that can bring the new technologies into operations and business. As a consequence, many specialists may be cooperating in different cooperation contexts and offer their services to each other. However, due to possibly daily

changing partner requirements, interoperability of such services may be hard to achieve in planning and operation. The KDP is intended to solve the described problem. The platform can be installed on premise or used in the cloud, respectively. It is designed to integrate co-operation processes, communication, security, documentation and so forth in a reusable way and be extensible with arbitrary services for specific cooperation cases. Accordingly, the KDP consists of a basic platform as well as a number of services and adapters for the integration of companies' internal systems. In Fig. 1 the KDP services in the given study are illustrated. The services employed for this application case are partly used for gathering and storing of information via the so-called *CIM cache* module. The *Congestion detection* module can be seen as an adapter to the previously discussed state estimation algorithm that is interfaced by the KDP system. The modules called here *Congestion handling* and *Flex call* represent services that distribute the calculated grid restrictions among participating DER and establish communication between them and the rest of the platform.

Despite the composite structure of the KDP, consisting of different modules, coupling with MOSAIK is realized via a single interface. Via this interface MOSAIK provides various data sets to the KDP. In field application, the platform would acquire this data from several services and interfaces in the energy system. In the co-simulation, however, the data is provided by simulators and partly given in different formats than required by the KDP. Therefore, some data mapping and formatting is implemented in the interface. This aspect is discussed in more detail in the next section. The overall data sets are strongly multi-dimensional so that data exchange is realized via CSV files that are placed in a shared folder structure, similar to the interfacing of the VPP simulator. However, since MOSAIK and the KDP are executed in the same computation infrastructure, no web server is needed in this case. As seen in Fig. 1, the KDP provides grid restriction data to MOSAIK while requiring DER schedules and power feed-in prognosis data from renewable energy resources. Furthermore, the platform requires acceptance notifications as an answer to the grid restrictions. This acceptance would be iteratively negotiated with the VPP in the field, but in the given co-simulation setup this is not possible since the employed VPP simulator does not support it. Therefore, a scenario has been established in which the provided grid restrictions are always accepted. The associated notification is directly constructed via the MOSAIK interface. One of the most crucial aspects of interfacing MOSAIK and the KDP is the time synchronization. The KDP has been developed for operation in real-time systems. Thus, it has been implemented to obtain time information from the system clock of its computation environment. This would lead to a setup that cannot be simulated faster or slower than real-time. In order to establish more temporal flexibility, the MOSAIK interface implements its own *Network Time Protocol* (NTP) server that provides the simulated point in time as a UTC-compliant time stamp. The KDP has been configured to synchronize with this server instead of its local system clock.

Challenges, solutions and lessons learned

The presented co-simulation setup has been shown to contain heterogeneous components that require different forms of interfacing and data exchange, run in geographically distributed computation infrastructures, and partly display composite structures with interconnected subsystems. Such a setup presents some challenges to the idea of generic co-simulation since this approach typically follows the idea of somewhat similar simulator types. Differences between simulation components are

mostly discussed in terms of different modeling paradigms, i.e. simulators displaying continuous time, discrete time or discrete event dynamics (e.g. (Ptolemaeus, 2014; Camus et al., 2016)). However, especially the interfacing of simulation components is subjected to considerable standardization effort with the FMI standard gaining increasing popularity (e.g. (Awais et al., 2013; Müller & Widl, 2015)). Since this standard expects the wrapping of a simulation component within a so-called Functional Mock-up Unit, it can quickly lead to interfacing hurdles or overhead when dealing with remote connections (e.g. for the VPP simulation) or composite, modular component structures (e.g. for the KDP). Overall, the CPES domain is quite diverse and contains few standardized simulation tools. Complex simulation components, on the other hand, representing in-house solutions with their own computation environments are likely to come up – as is presented in this study. Therefore, FMI is rather unlikely to cover all co-simulation interfacing needs in this domain in the near future. The MOSAIK Component-API used in this project possesses the advantage of directly supporting several programming languages and being interoperable with MOSAIK without requiring any additional wrapping of the simulation components. This allows for greater versatility when simulators require interfacing via web services or shared folders.

An aspect of the presented setup that is typically not considered in CPES co-simulation is the complex structure of input data for all the components. While simulators are often expected to accept and provide data in the form of scalar values, the VPP simulation as well as the KDP illustrate that complex simulation tools will often exchange data consisting of various vectors, partly associated with temporal or grid topology data. One of the easiest and most common ways to represent such data sets is via CSV files. Since the employed simulation components have been pre-existing before the creation of the co-simulation setup, every component has its own structuring of data in the CSV format. The re-formatting of the data files is done in the component interface implementations. Accordingly, a document has been established before the implementation process that illustrates the mapping of data between the components. This implementation procedure partly disagrees with the idea of generic co-simulation since component interfaces should ideally be design to be agnostic to other employed simulation components. This shows that the discussed co-simulation represents a mixture between generic co-simulation and ad-hoc tool coupling. After all, a co-simulation framework is used that manages the scheduling of the data exchange while on the other hand interfaces had to be tailored specifically to the component interactions given in this setup.

Due to the structure of CPES research and its simulation tools mentioned above (strong diversity, little standardization, complex components) co-simulation setups in this domain are likely to diverge from the idea of completely generic, standardized coupling, and shift towards ad-hoc coupling approaches. Using co-simulation frameworks still provides the merit of covering the need for scheduling of data exchange so that users do not have to establish their own scheduling algorithms. However, the interfacing to such a framework needs to be flexible and allow adjustments based on the given co-simulation setup. Some frameworks are designed with such flexibility in mind, like the Simulation Message Bus (Mosshammer et al., 2013) or systems based on the High Level Architecture standard (Dahmann & Morse, 1998). For these frameworks the middleware for data exchange typically provides rather basic functionalities and users need to establish several specifications to create a co-simulation setup. While the degree of freedom such frameworks provide

agrees well with co-simulation use cases that lean towards an ad-hoc coupling approach, the usability for inexperienced users is decreased as well. As discussed in (Steinbrink et al., 2018), usability is an important feature of co-simulation systems in interdisciplinary areas like the CPES domain. The framework MOSAIK supports usability via easily configurable co-simulation setups (Scenario-API) and data exchange scheduling that requires no additional user input. At the same time, the Component-API of MOSAIK allows for great flexibility via the support of several common programming languages. This way, special interfacing needs like remote coupling or data file formatting is supported and the gap between generic and ad-hoc co-simulation is bridged.

Conclusion

The paper at hand presents a co-simulation setup with heterogeneous components executed in distributed, geographically remote locations. Due to the special structure of the different components and the overall setup, co-simulation aspects need to be addressed that are usually not considered in ideal, generic coupling approaches.

For example, a time server module has been implemented to interact with an ICT system designed for field test applications. Furthermore, different technologies for exchange of complex data under remote coupling have been successfully employed, i.e. exchange of CSV files via shared web servers as well as transmission of byte strings via web services.

Due to the complexity of the exchanged data and different data file requirements of the components, data file formatting had to be implemented in the component interfaces. Thus, part of the interfaces have been specifically designed for the given co-simulation setup resulting in an application case deviates from the idea of fully generic co-simulation (as described above) and shift towards ad-hoc coupling. The authors argue that the interdisciplinary character of the CPES domain in combination with the structural diversity of simulation components and increasing interest in co-simulation will lead to numerous co-simulation applications that require bridging the gap between ad-hoc coupling and ideal, generic coupling. Such applications imply the need for specialized interfacing, on the one hand, tailored to the co-simulation setup in questions, and high-level scheduling modules, on the other hand, that make co-simulation accessible for users that cannot implement their own scheduling algorithms. This study shows that the framework MOSAIK is able cover these requirements rather well due to its different API.

For future co-simulation endeavors in the CPES domain a higher degree of standardization in component modeling would be beneficial. On the one hand, more standardized models lead to better support of interface standards like FMI, on the other hand they support co-simulation by easier composition of complex setups. Basic approaches for such modeling standards are given, e.g. in the form of the Open Energy Modeling Framework (Hilperta et al., 2018). However, not all types of models can be expected to comply with such standards and already existing models are unlikely to be re-implemented just to match a standard. An alternative is a standardized classification of simulation tools, their properties and the data they provide and expect. If a classification can be established that is detailed enough, it could greatly help with special interfacing needs and data mapping as seen in this paper. First approaches for such categorizations are presented by the Open Energy Modeling Initiative (Open Energy Modeling Initiative, n.d.). Optimally, the classifications will in the future also contain information about simulation tools aspects like performance or computational cost/complexity, accuracy, and data granularity.

Abbreviations

API: Application Programming Interface; CPES: Cyber-Physical Energy System; DER: Distributed Energy Resource; FMI: Functional Mock-up Interface; ICT: Information and Communication Technology; KDP: Kommunikations- und Dienstleistungsplattform (Communication and Service Platform); NTP: Network Time Protocol; PAVN: Proaktives Verteilnetz (Proactive Distribution Grid); VEP: Venios Energy Platform; VPP: Virtual Power Plant

Acknowledgements

The authors would like to thank the Federal Ministries for Economy and Energy (BMWi), and Education and Research (BMBF) for the funding of the project "Proaktives Verteilnetz".

Funding

The presented work is funded by the Federal Ministries for Economy and Energy (BMWi), and Education and Research (BMBF). Publication costs for this article were sponsored by the Smart Energy Showcases – Digital Agenda for the Energy Transition (SINTEG) programme.

Availability of data and materials

Further information about the presented work will be given in the final report of the project "Proaktives Verteilnetz". Source code and documentation of MOSAIK can be found on mosaik.offis.de.

About this supplement

This article has been published as part of *Energy Informatics* Volume 1 Supplement 1, 2018: Proceedings of the 7th DACH+ Conference on Energy Informatics. The full contents of the supplement are available online at <https://energyinformatics.springeropen.com/articles/supplements/volume-1-supplement-1>.

Authors' contributions

All authors have read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details

¹R&D Division Energy, OFFIS – Institute for Information Technology, Oldenburg, Germany. ²Venios GmbH, Frankfurt am Main, Germany. ³Institute of Power Systems and Power Economics (IAEW), RWTH Aachen University, Aachen, Germany. ⁴BTC Business Technology Consulting AG, Oldenburg, Germany.

Published: 10 October 2018

References

- Albrecht C (2014) Entwicklung einer skalierbaren Netzberechnungskomponente für ein Cloud-basiertes Monitoringsystem für Smart Grids. Ruhr Universität Bochum
- Awais MU, Palensky P, Mueller W, Widl E, Elsheikh A (2013) Distributed hybrid simulation using the hla and the functional mock-up interface. In: Annual Conference of the Industrial Electronics Society, IECON, pp 7564–7569
- Blochwitz T, Otter M, Arnold M, Bausch C, Clauß C, Elmqvist H, Junghanns A, Mauss J, Monteiro M, Neidhold T, Neumerkel D, Olsson H, Peetz JV, Wolf S (2011) The functional mockup Interface for tool independent exchange of simulation models. In: 8th International Modelica Conference, pp 173–184
- Büscher M, Claassen A, Kube M, Lehnhoff S, Piech K, Rohjans S, Scherfke S, Steinbrink C, Velasquez J (2014) Integrated smart grid simulations for generic automation architectures with RT-LAB and mosaik. In: IEEE International Conference on Smart Grid Communications (SmartGridCom)
- Camus B, Paris T, Vaubourg J, Presse Y, Bourjot C, Ciarletta L, Chevrier V (2016) Mecsyc: a multi-agent devs wrapping platform for the co-simulation of complex systems. In: PhD thesis, LORIA, UMR 7503, Université de Lorraine, CNRS, Vandoeuvre-lès-Nancy. Inria Nancy-Grand Est, Villers-lès-Nancy, France
- Coster E, Fidler H, Broekmans M, Koehler C (2017) Capacity management of low voltage grids using universal smart energy framework. In: Proceedings of Conference on Electricity Distribution (CIRED)
- Dahmann JS, Morse KL (1998) High level architecture for simulation: An update. In: 2nd International Workshop on Distributed Interactive Simulation and Real-Time Applications, pp 32–40
- Expert Group 3 - Regular Recommendations for Smart Grids Deployment (2013) Eg3 first year report: Options on handling smart grids data. Technical report, Smart Grid Task Force
- Hilperta S, Kaldemeyera C, Kriend U., Günthera S, Wingenbacha C., Plessmann G. The open energy modelling framework (oemof) – a new approach to facilitate open science in energy system modelling; 2018
- Moshammer R, Kupzog F, Faschang M, Stifter M (2013) Loose coupling architecture for co-simulation of heterogeneous components. In: Annual Conference of the Industrial Electronics Society, IECON, pp 7570–7575
- Müller W, Widl E (2015) Using fmi components in discrete event systems. In: 2015 Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES), pp 1–6
- Open Energy Modeling Initiative (n.d.) [<http://www.openmod-initiative.org/>], Accessed -30 July 2018
- Palensky P, Van Der Meer AA, López CD, Joseph A, Pan K (2017) Cosimulation of intelligent power systems: fundamentals, software architecture, Numerics, and coupling. IEEE Ind Electron Mag 11:34–50
- Ptolemaeus C (2014) System design, modeling, and simulation: using Ptolemy II. Vol. 1. Ptolemy org, Berkeley

- Ruiz N, Cobelo I, Oyarzabal J (2009) A direct load control model for virtual power plant management. *IEEE Trans Power Syst* 24:959–966
- Schütte S, Sonnenschein M (2012) Mosaik: scalable smart grid scenario specification. In: *Proceedings of the Winter Simulation Conference*, p 160
- Steinbrink C, Schlögl F, Babazadeh D, Lehnhoff S, Rohjans S, Narayan A (2018) Future perspectives of co-simulation in the smart grid domain. In: *2018 IEEE International Energy Conference (Energycon)*
- Strbac G (2008) Demand side management: Benefits and challenges. *Energy Policy* 36:4419–4426
- The ownCloud developers. *ownCloud User Manual*; 2018

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

Submit your next manuscript at ▶ [springeropen.com](https://www.springeropen.com)
