Privacy Protection in Collaborative Geographically Distributed Co-Simulation of Multimodal Energy Systems

Alexander Kocher
Karlsruhe Institute of Technology
Germany
alexander.kocher@kit.edu

Jovana Kovačević
Karlsruhe Institute of Technology
Germany
jovana.kovacevic@kit.edu

Anne-Christin Süß
Karlsruhe Institute of Technology
Germany
anne-christin.suess@kit.edu

Can Berk Saner
National University of Singapore
Singapore
sanerc@u.nus.edu

Burak Dindar
Karlsruhe Institute of Technology
Germany
burak.dindar@kit.edu

Md Amimul Ehsan
UNC Charlotte
USA
mehsan@charlotte.edu

Hüseyin K. Çakmak
Karlsruhe Institute of Technology
Germany
hueseyin.cakmak@kit.edu

Veit Hagenmeyer
Karlsruhe Institute of Technology
Germany
veit.hagenmeyer@kit.edu

ABSTRACT

Considering the complexity of the design and operation of decentralized energy systems, for which cooperation of experts in different fields is required, and the need for cooperation regardless of the used technology for multi-physics modeling, the necessity of distributed co-simulation emerges. In the context of real-time geographical co-simulation across the globe, various challenges arise, including communication issues and data privacy concerns. This paper introduces a novel collaboration method for globally distributed, real-time co-simulation with increased model privacy that does not require any further simulation model exchange between the participants. With the introduced modularization and the simplified local setup process for the client software, any Functional Mock-up Unit (FMU), e.g. from a physical model, can be locally integrated into the real-time geographical co-simulation. The simplicity of using the proposed co-simulation framework is demonstrated by involving non-experts into the experiments. To test the efficiency and applicability of the new collaboration approach, the proposed method is investigated under the aspects of performance, stability and result accuracy. Study cases are carried out under real conditions, with a distributed setup spread across Europe and a long-distance test spanning 14 world time zones. The outcomes reveal a high level of accuracy of co-simulation results under diverse operational scenarios, all while prioritizing data privacy and adhering to standard interfaces.

E-Energy '24, June 04–07, 2024, Singapore, Singapore

ACM ISBN 979-8-4007-0480-2/24/06
https://doi.org/10.1145/3632775.3661990

CCS CONCEPTS

• **Security and privacy → Domain-specific security and privacy architectures**; • **Computing methodologies → Distributed computing methodologies**.

KEYWORDS

collaborative co-simulation, model privacy, data privacy, distributed co-simulation, energy system integration.

ACM Reference Format:
Alexander Kocher, Jovana Kovačević, Anne-Christin Süß, Can Berk Saner, Burak Dindar, Md Amimul Ehsan, Hüseyin K. Çakmak, and Veit Hagenmeyer. 2024. Privacy Protection in Collaborative Geographically Distributed Co-Simulation of Multimodal Energy Systems. In *The 15th ACM International Conference on Future and Sustainable Energy Systems (E-Energy '24), June 04–07, 2024, Singapore, Singapore.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3632775.3661990

1 INTRODUCTION

For achieving Net Zero Emissions by 2050, deployment of renewable energy plays an important role. The incorporation of renewable heat and transport fuel can significantly curtail emissions in various sectors, including buildings, industry, and transportation [17]. Rapid integration of these technologies across different energy domains is crucial. When coupling different energy domains, the flexibility of the energy system and trustworthiness of the energy source increases and simultaneously the impact of energy consumption on the environment decreases [16]. In order to design and develop sustainable and efficient energy systems, the planning of operation strategies is required [15].

Multi-domain energy systems, which involve aspects like energy supply, consumption, and communication control introduce complexity, emphasizing the necessity for simulation planning and strategic approaches [25]. Multi-domain energy systems require expertise from diverse domains, posing challenges in development within a single software environment. Co-simulation addresses this issue by integrating different tools [23].

In multi-domain energy systems, simulation platforms usually offer good solutions for one energy subdomain, whereas components of other subdomains are simplified. To simulate a multi-domain energy system, the necessity for co-simulation exists, where each expert will develop models in their own domain with suitable software, whereas the coupling of subdomains will be handled by a co-simulation platform. Palensky et al. [22, 23] pointed out challenges in the multi-domain energy systems, where the fundamentals and applications of the co-simulation are discussed. A review study of Alfalouji et al. [2] discussed taxonomic review on co-simulation applications, tools, reproducibility, and validation of simulations. Gomes et al. [14] discussed continuous-time, discrete-event and hybrid, as the combination of the former two co-simulation approaches. They noted that the main challenges of co-simulation are modular coupling, stability and accuracy of simulations. Furthermore, Gomes et al. [13] classified features of co-simulation into non-functional requirements, simulation unit requirements, and framework requirements.

The coupling of different modules within a co-simulation approach introduces challenges, including interface definition, scalability, data exchange, and model privacy. So far, most simulations are generally collected as Functional Mock-Up Unit (FMU)[1] and calculated centrally. Using a centralized co-simulation environment can lead to the privacy problem, since models are integrated with the information of their internal structures. For instance, in the electrical power system, although the cooperation and data exchange between transmission system operator (TSO) and distribution system operator (DSO) has been continuously strengthened due to the increasing amount of renewable generation in the distribution grid [1, 3], data privacy remains a challenge to be addressed [1]. To further increase the cooperation and data exchange regarding data privacy, a decentralized co-simulation environment could help by only using the necessary data to share between TSOs and DSOs, and keeping models and underlying data at their respective operators. However, real-world challenges, such as communication issues, necessitate testing decentralized co-simulation systems over long distances with real-time applications with regard to system stability.

Moreover, a distributed co-simulation framework should be easily usable and applicable to large-scale scenarios without sacrificing speed. Balancing simulation speedup and accuracy is a crucial consideration that cannot be compromised [9]. For example, Mosaik [26], a Python-based co-simulation platform, is suitable for small and medium-scale simulations supporting distributed co-simulation and capable of simulating continuous and discrete event simulations [21]. However, scalability becomes a significant concern for large-scale simulations [4]. Another widely used co-simulation platform is HELICS [24], tailored for large-scale power grid scenarios. The software platform is compatible for parallel computing and scalable [5]. Nevertheless, both platforms are code-based, leading to time-consuming simulation setup processes and programming knowledge is required to be able to co-simulate. Furthermore, we set model and data privacy as highly important in a co-simulation environment, since large-scale complex systems require input and knowledge from various expert domains to be coupled, while at the same time respecting intellectual property. This aspect is often not pointed out in literature when it comes to co-simulation.

Given the aforementioned challenges, there is a crucial need for an easy-to-use framework capable of stable long distance distributed co-simulation while prioritizing data privacy. Therefore, we introduce a new method for the collaboration of experts in different domains to enhance complex system co-simulation. This is accomplished with the Energy System Co-Simulation (eCoSim) module within the Energy system Analysis, Simulation, Modeling, Optimization and Visualization (eASiMOV) software framework, as detailed in [7, 9, 10]. The eCoSim co-simulation software facilitates easy configuration, allowing interactive setup procedures in a distributed environment while ensuring data privacy of the modules. These features are further improved by the presented approach. Model privacy plays an important role, thus in the presented approach, experts do not need to share their own model with others to participate in the collaborative complex system co-simulation. Moreover, an easy setup procedure is important, so that participating experts in the co-simulation do not require any further programming knowledge. To test the ease of use of the underlying co-simulation framework and of the proposed collaboration approach, three non-experts in the field of co-simulation were invited to participate in the experiments in the context of this study. For this, we provide real-time long distance distributed co-simulation study cases, conducted using eCoSim for coupling energy system related simulation modules distributed over Germany, Serbia, Singapore, Turkey, and the USA. Further, the framework's capabilities regarding the stability of the system and correctness of the results is assessed. With its easy use and modular structure, the framework not only facilitates real-time distributed co-simulations but also ensures data privacy since the model encapsulation in FMUs exists and keeps the simulation models at the participants, respectively. Consequently, it enhances interoperability among experts and institutions in diverse domains, such as the energy and automobile sectors, which utilize distinct software solutions.

The remainder of this paper is structured as follows: In Section 2, the methodology, with an emphasis on distributed computing and collaborative co-simulation considering the proposed method, is described. To evaluate the viability of the proposed method, different use cases are presented in Section 3 before discussing the results in Section 4. Finally, the results are concluded and a brief outlook on our future work is given in Section 5.

2 METHODOLOGY

In this section, the basics of co-simulation and related standards and protocols are introduced. The new concept for collaborative co-simulation is presented as a new method for the implemented distributed co-simulation environment.

2.1 Co-Simulation Basics

Fundamentally, co-simulation consists of multiple simulation modules and a master, whereas modules represent models and corresponding solvers. The master manages and coordinates the communication between the modules.

To perform a co-simulation, orchestration of multiple models and solvers grouped in modules is needed. That is done by a simulation

[1] https://fmi-standard.org/

master, which acts as a superordinate instance and manages the order of the simulation modules. The simulation master defines the coupling timing, also called macro-step, which describes a fixed point in time where the data exchange between the modules is done. At this point in time, the simulation master receives all the results of the previous time-step from the modules and sends the data for the calculation of the next time-step to the modules. This coupling timing must be respected by all modules, and it depends on the module that simulates the longest. The simulation that is executed within each module still has its own step-size. This is called micro-step and is set individually for each module.

The orchestration of the modules is not only important to organize the modules for solving the given simulation but also to solve a possible algebraic loop, which can appear when modules need information from each other at the same time-step. This challenge can be handled in a parallel or sequential manner, described in [23]. For errors that appear due to these procedures, the authors describe an iterative approach where each module is solved multiple times during the same time-step.

For the purpose of data exchange between modules, a common standard has been established, known as Functional Mock-Up Interface (FMI) [6, 20]. It enables encapsulation of a model as a black-box model called FMU and provides functions to access its input and output values. Even though the FMU model can be regarded as a black-box, there are still reasons to limit the access to it, like restricting access to the FMU itself or its contents [8].

2.2 Distributed Computing Environment

For the distributed computing environment, the eCoSim software module [7] of the eASiMOV framework is used. The co-simulation framework is programmed in Java and can be executed with the Java Virtual Machine (JVM) on any operating system (OS). The simulation master coordinates the simulation by sending and receiving messages from the simulation modules. The simulation modules are standalone simulators implemented according to the FMI standard. Therefore, the models are exported as *.fmu* files, and implemented in eCoSim. The communication between the simulation modules is done through the master and thus have no direct connection to each other. Thus, a simulation module has no direct access to the data or structure of the other simulation modules. Together with the black-box properties of FMUs, this feature further preserves possible privacy concerns for decentralized co-simulations.

Additionally, each simulation module opens a connection to a database to store simulation results and metadata like calculation time, used memory and others. The database can be deployed on a computer independent of the computers running the master or the simulation modules. This leads to a three-level-architecture consisting of client, master, and database. The communication protocol used in eCoSim, as described in [9], is based on TCP for its built-in mechanisms for data re-transmission. For the simulation, the simulation master acts as a TCP-Server that the simulation modules connect to as TCP-Clients.

2.3 Collaborative Co-Simulation

The collaborative workflow is based on the communication protocol developed in [9]. In that previous study, there are two options to start a collaborative co-simulation. In the first option, every FMU containing a simulation module is sent to the co-simulation administrator to configure the simulation and run it locally on one computer. Although no underlying physical models are exchanged this way, the owner of the model still needs to hand over the standalone FMU, which can be reverse engineered or used in applications outside the co-simulation scope afterward, thus increasing the vulnerability of the intellectual property. Additionally, this approach leads to a big organizational overhead when considering larger-scale co-simulations with a high number of contributing participants. In the second option, every participant needs the entire code base installed on their computer to register their simulation module. This option allows a distributed co-simulation, but has the disadvantages of source code sharing and requires knowledge of the relevant parts of the source code from the participants. To mitigate these issues, we present a new method for improved collaboration that additionally increases model privacy.

The new simulation setup is divided into a standalone configuration phase and a simulation phase. In the configuration phase, the simulation administrator configures the simulation in a graphical user interface. Instead of sending their FMU files, the participants send the information for the required input and output interfaces of their simulation module to the simulation administrator. This increases the data privacy, as only the interface information and data resulting from the simulations have to be exchanged. Afterward, the simulation administrator sends an executable file and the information about the simulation setup to the participants, which is used for the simulation phase. Thus, the simulation setup involvement for the participants is reduced to sharing their interface information.

In the simulation phase, the master signals the start of the simulation to the participants. To prepare for this, the participants need their simulation module ready as a FMU and the simulation setup sent by the simulation administrator. Inside the simulation setup, each participant needs to change the network parameters of their own simulation to connect to the master module and update the path of the FMU on their computer, so that the received executable can access the interfaces of the model defined in the FMU. Afterward, the simulation module is ready to participate in the collaborative co-simulation by running an executable. When the simulation terminates, the simulation administrator can then extract the simulation results from the database and share them with the participants for further analysis.

Compared to the previous approach, the technical know-how required for a participant is reduced to a minimum. Knowledge about the eCoSim software code or programming in general is no longer required to participate in a co-simulation. Neither is there a need to share source code to the participants, which further eases the workflow for a collaborative co-simulation. With this approach, each participant can configure and setup their models for a collaborative co-simulation in their preferred OS and modeling environment. The participants do not need to share those resulting simulation models in order to participate in the co-simulation. This ensures model and data privacy for the participants. To show the correctness using this collaborative and distributed approach, the framework is evaluated with non-experts in the co-simulation domain in the following case studies.

To validate the new approach from a stability and data protection perspective, case studies are presented in the following section, including collaborative co-simulations with inexperienced participants from different parts of the world using the method described above to further test the usability of the approach and correctness of the software.

3 CASE STUDIES

In this section a simple multi-physics model of a building comprising models for building envelope, ventilation, heating, and the weather data module is introduced. The co-simulation model is exported as four single FMUs and used for real-world experiments over long distances. The simulation period for each study case is 31 days, with a resolution of one minute per time-step.

3.1 Simple Multi-Physics Model

Since the energy system models are not the focus of this study, only a simple model is utilized to evaluate the collaborative co-simulation approach. Extensive analysis of coupled energy systems with co-simulation techniques are presented in [11] using the eASiMOV-eCoSim co-simulation software for a comparative analysis of district multi-energy systems with 3840 variants.

For experimenting with geographically distributed co-simulation, a minimalistic modular energy-building model is developed that is derived from [19]. An overview of the Dymola model can be found in the appendix Figure 4. The system consists of four parts: building model, weather data, heating control and ventilation control. The first two parts are modeled in Dymola using the IDEAS library [18], and the latter two in MATLAB SIMULINK. The building model represents the physical behavior of a building in its entirety. It consists of the resistance-capacitance thermal building envelope [12], a respective representation of the window, and utilizes a pump, radiator and heat exchanger for heating. Furthermore, for ventilation purposes, a fan, heat exchanger and damper are employed. The control signals for heating and ventilation are represented by a hysteresis. The control signal for heating is set when the temperature in the room reaches 21 °C and heating is switched off at a temperature of 23 °C. Accordingly, the control signal for ventilation is set when the temperature in the room reaches 25 °C and switched off when the temperature drops to 23 °C. Input signals for the building model are damper and pressure control signals for ventilation, pump, and heat exchanger control signal for heating, and weather data as the outside temperature and solar irradiation. The output of the building model is measured temperature in the room. This signal is used as an input signal for both controllers.

3.2 Local Co-Simulation Setup as Reference

To compare simulation results and runtime behavior of geographically distributed co-simulations, we conducted a co-simulation in a local setup (LS). To set up LS, the simulation modules are exported as FMUs and organized as a co-simulation with the interactive modeling tool and simulated as a co-simulation in eCoSim on a singular computer. The master orchestrates the co-simulation of the modules (house, heating, weather, ventilation) locally. The simulation results will serve as a reference for comparison to the case studies.

3.3 Case Studies: Long Distance Tests

The system behavior of the co-simulation is investigated with a geographically long distance (LD) connection setup. For this, two cases are defined with variations in their locations (LD-1 and LD-2). For classification, we represent test cases as a set of pairs defined by *Location* and *Simulation-Modules*, representing the FMUs for heating, ventilation, house, weather, and the master as introduced in Section 3.1.

The locations of the simulators are selected to combine local and global sites, whereas all data communication is routed via Virtual Private Network (VPN) over the KIT network to enable secure and stable data exchange. In detail the *global sites* are Germany (DEU), USA (Charlotte), Singapore (SGP), Turkey (TUR), Serbia (SRB). The *local sites* in Germany are within of a 15 km radius of KIT (Karlsruhe Institute of Technology), namely the cities Stutensee, Bruchsal and Karlsruhe. The nominal speed of each participant in the geographically long distance co-simulation testing is indicated in Table 1 whereas Figure 1 illustrates the locations of the participants for both long distance tests LD-1 and LD-2. The assignment of simulation modules to locations is shown in Table 2.

Table 1: Nominal internet connection speeds of participants

Location-Global	SGP/Singapore	USA/Charlotte	TUR/Istanbul	SRB/Belgrade
Internet Speed	1 Gbit/s	300 Mbit/s	16 Mbit/s	65 Mbit/s
Locations (DEU)	**Karlsruhe**	**Bruchsal**	**Stutensee**	**KIT (Egg-Leo)**
Internet Speed	250 Mbit/s	100 Mbit/s	1 Gbit/s	400 Gbit/s

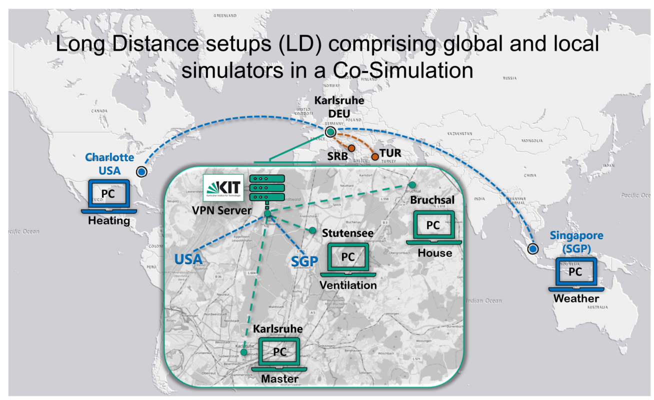


Figure 1: Co-Simulation setup of the globally and locally distributed simulators for long distance experiments LD-1 (red dashed lines) and LD-2 (blue dashed lines).

Table 2: Cases for long distance co-simulation testing

Modules	House	Ventilation	Weather	Heating	Master
Case LD-1	TUR-Istanbul	SRB-Belgrade	DEU-Stutensee	SRB-Belgrade	DEU-Karlsruhe
Case LD-2	DEU-Bruchsal	DEU-Stutensee	SGP-Singapore	USA-Charlotte	DEU-Karlsruhe

The test cases can be characterized as follows: Case LD-1 serves as a European experiment connecting three European countries, where one participant has a slow internet connection. Case LD-2 is an intercontinental long distance test, spanning over 14 world time zones, and integrates global and local testing sites.

4 RESULTS AND DISCUSSION

In this section, first, the co-simulation results for all case studies are analyzed and compared to the local reference. To analyze the geographically distributed aspects, the behavior of the system is then examined with regard to data transfer times and runtime of the reference configuration, introduced as the local setup (LS), and the long distance tests (LD). The strengths and weaknesses of the proposed approach are discussed. The case study for LS is carried out on a standard laptop computer with `Intel® Core™ i7-1255U CPU @ 1.70GHz` and `16GB` installed RAM. This is also the laptop where the master is located for the LD-1 and LD-2 cases.

4.1 Comparison of the Simulation Results

To validate the simulation results of the distributed energy system simulation in eCoSim, we compare the time series of the target value *controlled room temperature* for the long distance tests with the local co-simulation results.

Figure 2 shows a comparison of room temperature for the local setup LS and the long distance tests LD-1, LD-2. The tests are examined for 31 simulation days, whereas Figure 2 shows an excerpt of 7 days. All simulation results for the target value are identical for all simulation runs of all configurations up to the 12th decimal number. Thus, the sum of the absolute differences of the study case time series compared to the reference system time series is zero. This is a strong indicator that the eCoSim co-simulation software delivers correct simulation results and thus proves to be suitable for long distance collaborative real-time co-simulation with emphasis on data privacy.

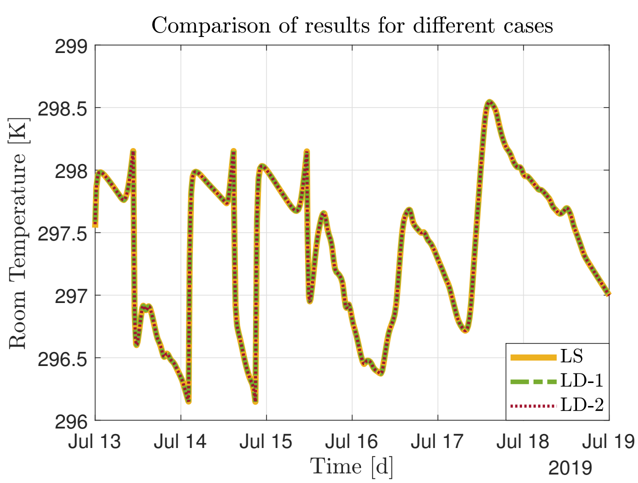


Figure 2: Simulation results for the controlled room temperature across global long-distance tests (LD) and local co-simulation (LS) for 7 days.

4.2 Runtime Analysis of Local Simulation

Interestingly, we register random swapping of the simulation durations for each simulation, which can be seen recorded with a Java profiler in the appendix Figure 5. A possible explanation is the behavior of the OS and the garbage collection of the JVM. Looking into the histogram for the computation time of the master and the simulation modules for each time step, reveals more behavior of the master module (see appendix Figure 6). The simulation modules are all close to each other in regard to the frequency for the duration of a time-step. Only the master shows outliers in the range of 4000-6000 ms. These outliers are caused by the aforementioned garbage collection of the JVM. This is also supported when looking at the CPU usage and used heap memory of LS as seen in Figure 3. Spikes in CPU usage coincide with the clearing of the used heap memory space by the garbage collection of the JVM. The duration for the whole simulation is about 26 minutes with a total of 44 640 time-steps. The duration for one simulation step is the same as the duration of the master, as it coordinates the simulation and can only finish with the current time-step after each simulation module has finished its calculation for that time-step.

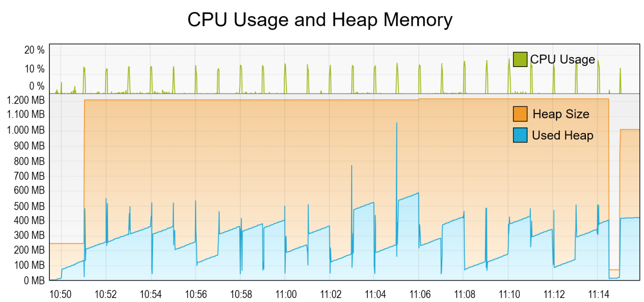


Figure 3: The CPU and heap memory usage of the local setup (LS) tracked with the Java profiler VisualVM for the master. The top part shows the CPU usage, while the bottom part shows the heap memory space.

The duration of a time-step is dependent on the underlying model and its simulation configuration. For our simple multi-physics model, used for the case studies, the average time-step needs 30 ms. Looking further into the data of LS and the configuration of the whole co-simulation, it can be deduced that the lower bound for a time-step consists of the time the house module needs plus the next longest lasting module. The house module needs the input from the three other simulation modules for its calculation. Because of this, the simulation pipeline order calculates the three other modules in parallel before the house module is calculated. Therefore, the slowest module of weather, ventilation, and heating plus the following calculation of the house module determines the duration of a time-step. The lower bound of LS also determines the lower bound for the LD test cases as, excluding round-trip times for network communication, the same simulation configuration is used.

4.3 Runtime Analysis of Long Distance Tests

Besides the accuracy of the results, run time is also considered. For LD-1 (see appendix Figure 7) the histograms show higher duration times for the master and modules due to the roundtrip time over the network. Especially for the house module, there are multiple duration times over the 2000 ms second mark. These are most likely caused by the slow network connection of the participant of the

Table 3: Statistical indicators for the run times of the simulation modules in the local (LS) and the two long-distance co-simulation tests LD-1 and LD-2.

Modules	S.I.	House	Ventilation	Weather	Heating	Master
Case LS	Mean [ms]	12.24	12.02	13.71	13.89	33.44
	Median [ms]	13.00	12.00	13.00	13.00	31.00
Case LD-1	Mean [ms]	129.01	70.83	53.22	68.20	208.41
	Median [ms]	119.00	60.00	44.00	58.00	188.00
Case LD-2	Mean [ms]	53.43	57.82	332.76	138.31	387.61
	Median [ms]	52.00	41.00	280.00	137.00	333.00

house module, as these coincide at the same location in duration in the master module. The impact of the JVM is also significantly reduced in this case compared to the master in the LS case. Due to the distributed configuration, the memory load in the computer the master runs on, is reduced and the JVM interferes less.

The long distance co-simulation test LD-2 shows similar results when looking at the histograms (see appendix Figure 8). The weather module located in Singapore has two big outliers with a duration of 38 seconds and 60 seconds respectively, which can also be seen in the histogram of the master module. Looking into the data, these two values are consecutive time-steps which leads to the assumption, that this was caused by a congestion in that network. This shows that eCoSim can handle these without stopping the simulation or having a reduced quality in the results.

The mean and median runtime values of each module for both long-distance co-simulation setups are shown in Table 3.

4.4 Discussion

To evaluate our collaborative co-simulation approach, we interconnected models and simulations over 14 world time zones, namely located in Germany, Serbia, Singapore, Turkey, and the USA as physical locations for long distance (LD) test cases. Since the LD-1 and LD-2 are evaluated on a long distance, an easy setup procedure is necessary to efficiently organize and evaluate co-simulations with all participants. Nevertheless, using our approach, participants do not need to have any knowledge about programming to contribute to the co-simulation setup. The correctness of the simulation results could be verified in the LD tests with three participating non-experts in the field of co-simulation from the USA, Turkey, and Singapore. They were able to set up the assigned FMU modules and take part in the co-simulation successfully, for which they only had to enter the internet connection data of the eCoSim master and start the client software. The models are all located at their respective participant, and thus no model information other than the simulation data previously defined by the interfaces was shared.

The results from the LD cases show that eCoSim is able to operate and deliver correct co-simulation results under high fluctuations of data transfer delays and big spatial distances of the participating simulation modules. Poor internet connections in LD-1 from Turkey and network congestion in LD-2 from Singapore have also been handled successfully by eCoSim. However, the LD-2 case needed a few restarts during testing to get a full simulation run of 31 days because of disconnecting modules. A disconnected module leads to an error state in the co-simulation necessitating a restart, as a failure recovery feature is currently under development at the time of this study. The only negatively impacted technical aspect is the higher runtime for each time-step and therefore increased duration of the overall simulation. Though this is expected with the higher roundtrip time between the simulation modules in the LD-cases compared to the local reference LS. Moreover, it can be concluded that the simulation is stable since the simulation results of these tests are equivalent up to the 12^{th} decimal number compared to the reference LS-case. The advantage of the presented approach is the high degree of model and data privacy, so that no model or FMU has to be exchanged, and the participants can take part in the co-simulation directly from their own local modeling and computing environment.

5 CONCLUSION AND OUTLOOK

To effectively simulate complex multi-domain systems, the necessity for a suitable co-simulation software platform is identified. This platform enables modules modeled in different environments to be coupled and simulated together. Additionally, data privacy concerns are addressed in this platform, allowing each model to contribute to a complex system co-simulation without revealing internal model topology. To tackle these challenges, the eCoSim co-simulation environment and a new collaborative workflow are introduced for distributed co-simulations. The preservation of data privacy of each module is ensured by only sharing necessary interface specification and no other models or files. Additionally, the master is the only one knowing the model coupling structure of the complex multi-domain system, without comprehending the internal structure of each module. In this way, each expert can develop its own models in familiar environments and couple them, without sharing its own models, to contribute to a complex system co-simulation. Furthermore, there is no required programming knowledge for model developers, which enables an easy setup procedure in the suitable operating system for each participant. To evaluate that the collaborative approach delivers correct simulation results, we built a co-simulation setup that is used for multiple case studies. A simple multi-physics model is described as a base model for the case studies, consisting of a local setup (LS) and long distance (LD). The new collaboration approach with eCoSim is tested successfully under these case studies, which are compared to a local setup used as a reference case. In the long distance tests, participants from the USA, Germany, Serbia, Turkey, and Singapore simulated together, including three non-experts in the field of co-simulation. The correctness of all simulation results show the performance of eCoSim as a framework for distributed collaborative co-simulations with data privacy.

For future work, development on the co-simulation framework will be continued to further lowering the barrier to participate in a co-simulation. Other planned features include more flexibility of the modules by testing their setup without the need for the other participants, and failure recovery for disconnecting modules. In addition, the simple multi-physics model will be replaced by a highly scaled sector coupling simulation setup involving heating, gas and electrical networks in residential neighborhoods and urban areas, including complex building models and corresponding controls. The co-simulation framework will serve as a basis for the optimization and analysis of coupled energy systems in collaboration with project partners.

ACKNOWLEDGMENTS

This work was conducted within the framework of the Helmholtz Program Energy System Design (ESD) and is partially funded under the project "Helmholtz platform for the design of robust energy systems and their supply chains" (RESUR).

REFERENCES

[1] Mohammed Al-Saadi, Rui Pestana, Ricardo Pastor, Gonçalo Glória, Aleksandr Egorov, Francisco Reis, and Tiago Simão. 2019. Survey Analysis on Existing Tools and Services for Grid and Market Stakeholders and Requirements to Improve TSO/DSO Coordination. 1–7. https://doi.org/10.1109/ISSE46696.2019.8984489

[2] Qamar Alfalouji, Thomas Schranz, Basak Falay, Sandra Wilfling, Johannes Exenberger, Thorsten Mattausch, Claudio Gomes, and Gerald Schweiger. 2023. Co-simulation for buildings and smart energy systems—A taxonomic review. *Simulation Modelling Practice and Theory* (2023), 102770.

[3] Rui Alves, Francisco Reis1, and Caihao Liang. 2015. TSOs and DSOs Collaboration: The Need for Data Exchange. In *Deregulated Electricity Market Issues in South Eastern Europe*. Trivent, 55–62. https://doi.org/10.22618/TP.EI.20151.192009

[4] Luca Barbierato, Pietro Rando Mazzarino, Marco Montarolo, Alberto Macii, Edoardo Patti, and Lorenzo Bottaccioli. 2022. A comparison study of co-simulation frameworks for multi-energy systems: the scalability problem. *Energy Informatics* 5, 4 (2022), 1–26.

[5] Alok Kumar Bharati and Venkataramana Ajjarapu. 2021. SMTD co-simulation framework with HELICS for future-grid analysis and synthetic measurement-data generation. *IEEE Transactions on Industry Applications* 58, 1 (2021), 131–141.

[6] Torsten Blochwitz, Martin Otter, Martin Arnold, Constanze Bausch, Christoph Clauß, Hilding Elmqvist, Andreas Junghanns, Jakob Mauss, Manuel Monteiro, Thomas Neidhold, et al. 2011. The functional mockup interface for tool independent exchange of simulation models. In *Proceedings of the 8th international Modelica conference*. Linköping University Press, 105–114.

[7] Hüseyin Çakmak, Anselm Erdmann, Michael Kyesswa, Uwe Kühnapfel, and Veit Hagenmeyer. 2019. A new distributed co-simulation architecture for multiphysics based energy systems integration: Analysis of multimodal energy systems. *at-Automatisierungstechnik* 67, 11 (2019), 972–983.

[8] Eric Durling, Elias Palmkvist, and Maria Henningsson. 2017. FMI and IP Protection of Models: A Survey of Use Cases and Support in the Standard. In *Proceedings of the 12th International Modelica Conference*. Modelica Association.

[9] Anselm Erdmann, Huseyin K. Cakmak, Uwe Kuhnapfel, and Veit Hagenmeyer. 2019. A New Communication Concept for Efficient Configuration of Energy Systems Integration Co-Simulation. In *2019 IEEE/ACM 23rd International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*. IEEE, 1–8. https://doi.org/10.1109/DS-RT47707.2019.8958665

[10] Anselm Erdmann, Anna Marcellan, Dominik Hering, Michael Suriyah, Carolin Ulbrich, Martin Henke, André Xhonneux, Dirk Müller, Rutger Schlatmann, and Veit Hagenmeyer. 2020. On verification of designed energy systems using distributed co-simulations. In *2020 IEEE/ACM 24th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*. IEEE, 1–8.

[11] Anselm Erdmann, Anna Marcellan, Jan Stock, David Neuroth, Christian Utama, Michael Suriyah, Sina Steinle, Felicitas Müller, Dominik Hering, Henning Francke, et al. 2023. A comparative co-simulation analysis to improve the sustainability of cogeneration-based district multi-energy systems using photovoltaics, power-to-heat, and heat storage. *Energy Conversion and Management* 291 (2023), 117226.

[12] Moritz Frahm, Felix Langner, Philipp Zwickel, Jörg Matthes, Ralf Mikut, and Veit Hagenmeyer. 2022. How to Derive and Implement a Minimalistic RC Model from Thermodynamics for the Control of Thermal Parameters for Assuring Thermal Comfort in Buildings. In *2022 Open Source Modelling and Simulation of Energy Systems (OSMSES)*. IEEE, 1–6.

[13] Cláudio Gomes, Casper Thule, David Broman, Peter Gorm Larsen, and Hans Vangheluwe. 2017. Co-simulation: State of the art. *arXiv preprint arXiv:1702.00686* (2017).

[14] Cláudio Gomes, Casper Thule, David Broman, Peter Gorm Larsen, and Hans Vangheluwe. 2018. Co-simulation: a survey. *ACM Computing Surveys (CSUR)* 51, 3 (2018), 1–33.

[15] Armin Grunwald and Christine Rösch. 2011. Sustainability assessment of energy technologies: towards an integrative framework. *Energy, sustainability and society* 1, 1 (2011), 1–10.

[16] Charitha Buddhika Heendeniya, Andreas Sumper, and Ursula Eicker. 2020. The multi-energy system co-planning of nearly zero-energy districts–Status-quo and future research potential. *Applied energy* 267 (2020), 114953.

[17] IEA. [n. d.]. Tracking Clean Energy Progress. https://www.iea.org/energy-system/renewables. Accessed: 21.12.2023.

[18] Filip Jorissen, Glenn Reynders, Ruben Baetens, Damien Picard, Dirk Saelens, and Lieve Helsen. [n. d.]. Implementation and Verification of the IDEAS Building Energy Simulation Library. *Journal of Building Performance Simulation* 11 ([n. d.]), 669–688. Issue 6. https://doi.org/10.1080/19401493.2018.1428361

[19] Jansen Jorissen, Heijde. [n. d.]. *__CrashCourse__*. https://github.com/open-ideas/__CrashCourse__

[20] Modelica Association. 2020. *Functional Mock-up Interface for Model Exchange and Co-Simulation Version 2.0.2.* Technical Report. Modelica Association, Linköping. https://fmi-standard.org

[21] Annika Ofenloch, Jan Sören Schwarz, Deborah Tolk, Tobias Brandt, Reef Eilers, Rebeca Ramirez, Thomas Raub, and Sebastian Lehnhoff. 2022. MOSAIK 3.0: Combining Time-Stepped and Discrete Event Simulation. In *2022 Open Source Modelling and Simulation of Energy Systems (OSMSES)*. IEEE, 1–5.

[22] Peter Palensky, Arjen van der Meer, Claudio Lopez, Arun Joseph, and Kaikai Pan. 2017. Applied cosimulation of intelligent power systems: Implementing hybrid simulators for complex power systems. *IEEE Industrial Electronics Magazine* 11, 2 (2017), 6–21.

[23] Peter Palensky, Arjen A Van Der Meer, Claudio David Lopez, Arun Joseph, and Kaikai Pan. 2017. Cosimulation of intelligent power systems: Fundamentals, software architecture, numerics, and coupling. *IEEE Industrial Electronics Magazine* 11, 1 (2017), 34–50.

[24] Bryan Palmintier, Dheepak Krishnamurthy, Philip Top, Steve Smith, Jeff Daily, and Jason Fuller. 2017. Design of the HELICS high-performance transmission-distribution-communication-market co-simulation framework. In *2017 Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES)*. IEEE, 1–6.

[25] Florian Schloegl, Sebastian Rohjans, Sebastian Lehnhoff, Jorge Velasquez, Cornelius Steinbrink, and Peter Palensky. 2015. Towards a classification scheme for co-simulation approaches in energy systems. In *2015 International Symposium on Smart Electric Distribution Systems and Technologies (EDST)*. IEEE, 516–521.

[26] Cornelius Steinbrink, Marita Blank-Babazadeh, André El-Ama, Stefanie Holly, Bengt Lüers, Marvin Nebel-Wenner, Rebeca P. Ramírez Acosta, Thomas Raub, Jan Sören Schwarz, Sanja Stark, Astrid Nieße, and Sebastian Lehnhoff. 2019. CPES Testing with mosaik: Co-Simulation Planning, Execution and Analysis. *Applied Sciences* 9, 5 (2019). https://doi.org/10.3390/app9050923

APPENDIX

Energy Building Model

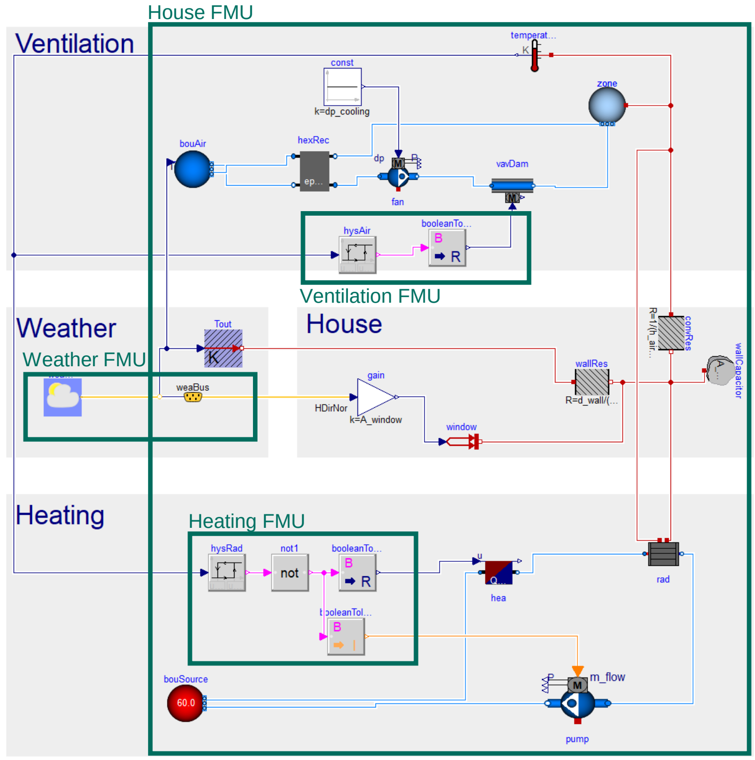


Figure 4: Modular Energy-Building Model used for the experiments for distributed simulation coupling.

Evaluation of Local Setup (LS)

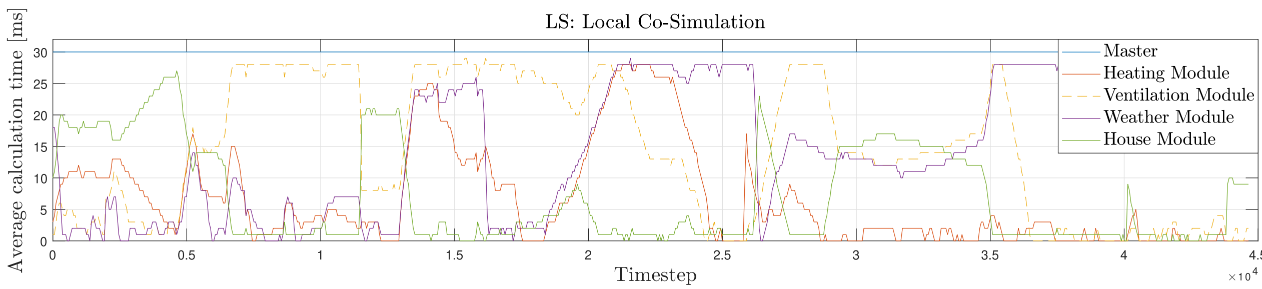
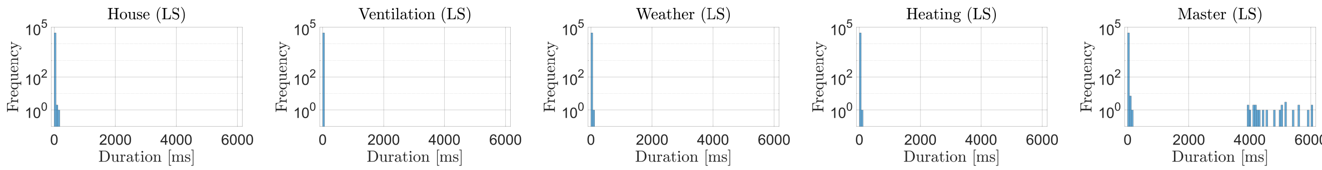


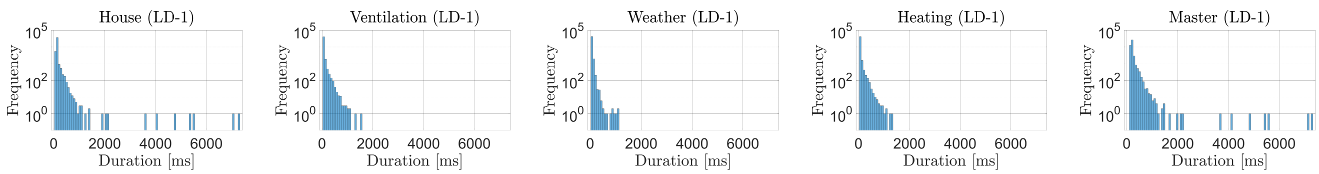
Figure 5: Evaluation of the runtimes for the local setup (LS): Average calculation times for the master and the modules for heating, ventilation, weather, and house.

Runtime Histograms



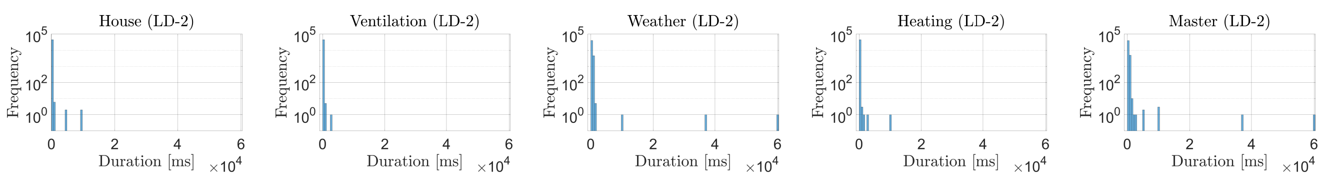
(a) House, 250 Mbit/s DEU-Karlsruhe **(b) Ventilation, 250 Mbit/s DEU-Karlsruhe** **(c) Weather, 250 Mbit/s DEU-Karlsruhe** **(d) Heating, 250 Mbit/s DEU-Karlsruhe** **(e) Master, 250 Mbit/s DEU-Karlsruhe**

Figure 6: Histograms of runtimes for each module in the local co-simulation test LS.



(a) House, 16 Mbit/s TUR-Istanbul **(b) Ventilation, 65 Mbit/s SRB-Belgrade** **(c) Weather, 1 Gbit/s DEU-Stutensee** **(d) Heating, 65 Mbit/s SRB-Belgrade** **(e) Master, 250 Mbit/s DEU-Karlsruhe**

Figure 7: Histograms of runtimes for each module in the European long distance co-simulation test LD-1.



(a) House, 100 Mbit/s DEU-Bruchsal **(b) Ventilation, 1 Gbit/s DEU-Stutensee** **(c) Weather, 1 Gbit/s Singapore** **(d) Heating, 300 Mbit/s USA-Charlotte** **(e) Master, 250 Mbit/s DEU-Karlsruhe**

Figure 8: Histograms of runtimes for each module in the world-wide long distance co-simulation test LD-2.