POLITECNICO DI TORINO
Repository ISTITUZIONALE

Facilitating Smart Grids Integration through a Hybrid Multi-Model Co-Simulation Framework

Original
Facilitating Smart Grids Integration through a Hybrid Multi-Model Co-Simulation Framework / Barbierato, L., Schiera, D.S., Orlando, M., Lanzini, A., Pons, E., Bottaccioli, L., Patti, E.. - In: IEEE ACCESS. - ISSN 2169-3536. - 12:(2024), pp. 104878-104897. [10.1109/access.2024.3435336]

Availability:
This version is available at: 11583/2991367 since: 2024-08-12T10:56:19Z

Publisher:
IEEE

Published
DOI:10.1109/access.2024.3435336

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

25 November 2026

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 6 June 2024, accepted 20 July 2024, date of publication 29 July 2024, date of current version 7 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3435336

RESEARCH ARTICLE

Facilitating Smart Grids Integration Through a Hybrid Multi-Model Co-Simulation Framework

LUCA BARBIERATO[1,4], (Member, IEEE), DANIELE SALVATORE SCHIERA[2,4], (Student Member, IEEE), MATTEO ORLANDO[3,4], (Student Member, IEEE), ANDREA LANZINI[2,4], (Member, IEEE), ENRICO PONS[2,4], (Senior Member, IEEE), LORENZO BOTTACCIOLI[1,4], (Member, IEEE), AND EDOARDO PATTI[3,4], (Member, IEEE)

[1]Department of Control and Computer Engineering, Politecnico di Torino, 10129 Turin, Italy
[2]Department of Energy, Politecnico di Torino, 10129 Turin, Italy
[3]Interuniversity Department of Regional and Urban Studies and Planning, Politecnico di Torino, 10125 Turin, Italy
[4]Energy Center Laboratory, Politecnico di Torino, 10138 Turin, Italy

Corresponding authors: Luca Barbierato (luca.barbierato@polito.it) and Daniele Salvatore Schiera (daniele.schiera@polito.it)

The work of Daniele Salvatore Schiera was supported by the "Network 4 Energy Sustainable Transition—NEST," through the Project code PE0000021, Concession Decree No. 1561 adopted by the Ministero dell'Universit e della Ricerca (MUR), under Grant CUP E13C22001890001; and in part by the National Recovery and Resilience Plan (NRRP), Mission 4 Component 2 Investment 1.3–Call for tender No. 341 of MUR; funded by the European Union–NextGenerationEU. The work of Edoardo Patti was supported by the project NODES through the MUR–M4C2 1.5 of PNRR founded by the European Union–NextGenerationEU under Grant ECS00000036.

ABSTRACT Smart Grid integration plays a crucial role in transitioning towards a climate-neutral future by enabling advanced monitoring, management, and control of renewable energy sources, energy systems, and networks. However, several barriers related to technological, economic, regulatory, and social aspects hinder the integration of these innovative resources and strategies into current power systems due to the inherent complexity of heterogeneous technologies, entities, and actors. For example, interoperability issues, cybersecurity concerns and data management are relevant to the integration and digitalization of smart grids. To address these challenges, this paper proposes a hybrid multi-model co-simulation infrastructure to simulate innovative Smart Grid scenarios. The infrastructure enables the interconnection of heterogeneous software simulators with real-time hardware simulators within a shared and distributed co-simulation environment, facilitating Hardware-In-the-Loop (HIL) applications through a semi-automated scenario configuration procedure. The proposed infrastructure's capabilities and performance are assessed through a smart grid scenario, focusing on implementing a distribution voltage regulation service provided by distributed resources installed on a building premise. Specifically, the scenario includes a physical smart meter device interconnected in HIL with the simulated building energy management system to test the integrated functionality and interoperability for the ancillary service. The scenario results demonstrate the facilitated scenario design process, and the promising performance and low co-simulation latencies of the co-simulation infrastructure when coupling software and hardware simulators with HIL applications. Overall, the infrastructure has the potential to assist researchers, system operators, and energy stakeholders in evaluating Smart Grid scenarios and designing, developing, and testing new systems, technologies, and business models.

INDEX TERMS Co-simulation infrastructure, digital real-time simulator, energy systems integration, hardware-in-the-loop, software simulator.

ACRONYMS

AC Alternate Current.
AIO Analog Input/Output.
API Application Programming Interface.
BAU Business As Usual.
BEMS Building Energy Management System.
COE Co-simulation Orchestrator Engine.
DSO Distribution System Operator.
DRTS Digital Real-Time Simulator.

The associate editor coordinating the review of this manuscript and approving it for publication was Ahmed A. Zaki Diab.

© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

DB	Database.
DC	Direct Current.
DER	Distributed Energy Resource.
DIO	Digital Input/Output.
DSPL	Domain-Specific Programming Language.
EMT	Electromagnetic Transients.
EHP	Electric Heat Pump.
FMI	Functional Mock-up Interface.
GPS	Global Positioning System.
GHI	Global Horizontal Irradiance.
GIS	Geographic Information System.
GPPL	General-Purpose Programming Language.
GUI	Graphical User Interface.
HIL	Hardware-In-the-Loop.
IoT	Internet-of-Things.
IPR	Intellectual Property Right.
LV	Low Voltage.
MQTT	Message Queuing Telemetry Transport.
MV	Medium Voltage.
PTP	Precision Time Protocol.
PHIL	Power Hardware-In-the-Loop.
PID	Proportional Integral Derivative.
POD	Point of Delivery.
PV	Photovoltaic.
RMS	Root Mean Square.
RTOS	Real-Time Operative System.
SFP	Small-form Factor Pluggable.
SGAM	Smart Grid Architectural Model.
SOC	State Of Charge.
SoS	System-of-Systems.
TCP	Transmission Control Protocol.
UDP	User Datagram Protocol.
VCS	Voltage Control System.

I. INTRODUCTION

The Smart Grid concept employs digital and advanced technologies to effectively monitor and control the transmission and distribution of electricity from diverse generation sources, ensuring the fulfilment of the fluctuating electricity demand of consumers. However, integrating Smart Grids within broader multi-energy systems is the key strategy in steering the energy transition towards a low-carbon future by envisioning an economy that absorbs as much greenhouse gas as it emits, drastically reducing carbon emissions. This involves coordinating operational and planning strategies to evolve from centralized to distributed power systems seamlessly with the end-use sector and interconnecting them across different energy vectors, infrastructures, and other sectors. This approach aims to optimise the overall complex energy system to maximize efficiency and reduce overall system costs [1]. Moreover, the Smart Grid streamlines the integration of renewable energy sources, distributed multi-generation, energy conversion systems, and energy storage technologies through cross-sector coupling reducing global dependence on fossil fuels. The vision entails a shift away from conventional vertical and centralized energy systems towards horizontal, interconnected, and distributed Smart Grid promoting the adoption of advanced control strategies. In this perspective, leveraging modelling and simulation-based assessments for the development, testing, and evaluation of Smart Grid integration scenarios can provide valuable insights to explore system concepts, evaluating requirements, and significantly cutting down the time and costs associated with prototyping and deploying innovative technologies. Additionally, they offer crucial data for prediction and validation, enhancing the efficacy of planning and operating these complex energy systems [2].

A. THE CO-SIMULATION APPROACH FOR MANAGING ENERGY SYSTEMS COMPLEXITY

Regarding the modelling and simulation-based assessments for Smart Grids integration, in the last decades, the research community focused on developing standalone and domain-specific simulation tools to cope with the needs of analysing these complex systems [3]. For instance, notable examples of electrical simulation software include Neplan, Siemens PSS, and Digsilent Powerfactory. These are all commercially available and capable of performing both Root Mean Square (RMS) and transient analysis. For Electromagnetic Transients (EMT) analysis, an open-sorce option is ATP, while commercial solutions include EMTP and PSCAD, among others. However, the standalone and domain-specific solutions have different drawbacks, as follows: *i)* generally, they focus on simulating specific aspects and characteristics of the systems linked with the primary domain accurately but simplifying elements and parts of the systems correlated with the primary domain [4]; *ii)* some of the existing tools can hardly be integrated to perform complex multi-disciplinary simulations due to the usage of widely different languages, modelling approaches, standards, and communication protocols [5]; *iii)* the inherent constraints of proprietary software could lead to its inability to be easily modified as desired, thereby limiting users in customizing it to align with specific needs or preferences for multi-domain purposes; *iv)* it is tough to integrate physical devices into a real-time simulation environment to perform Hardware-In-the-Loop (HIL) simulation or even integrate electrical devices, such as motors, inverters, and generators, through Power Hardware-In-the-Loop (PHIL) simulation [6]. For instance, some of the mentioned electrical simulation software can be interfaced with other software, including custom solutions, even if they lack real-time capabilities and synchronization, making them suitable primarily for offline analysis. Therefore, standalone and domain-specific simulation may fall short in capturing the multi-disciplinary, multi-domain, and multi-model aspects essential for depicting the integrated vision of smart grids with the required spatiotemporal scalability.

On the other hand, literature has also proposed various solutions to integrate different simulators, reducing the need for custom solutions which may be too specific for a particular domain, such as EnergyPlan [7], HOMER [8],

DER-CAM [9], or even through a multi-domain approach [5]. However, those tools do not provide real-time simulation for the electrical grid and do not support the integration of external simulators or HIL. Instead, other tools such as IDEAS [10], BCVTB [11] are devoted to coupling multi-domain simulators. However, linking a real-time simulator is not completely supported by these tools alone, and synchronization is not guaranteed.

In literature, co-simulation techniques have been proposed to interconnect different Domain-Specific Programming Language (DSPL), General-Purpose Programming Language (GPPL), and hardware simulators, exploiting each of the respective language peculiarities and solvers in a shared and distributed simulation environment capable of analysing concurrent aspects of a complex system [12]. However, co-simulation is challenging when dealing with heterogeneous simulation tools, e.g., communication among different simulation entities and their time regulation and synchronisation [13]. Many researchers tried to find solutions to overcome these challenges by developing custom tools to integrate different simulators. For example, a well-known solution in the literature is the Functional Mock-up Interface (FMI) [14] that allows the co-simulation of models between different domain-specific modelling and simulation tools [15].

Other researchers tried different solutions to include Digital Real-Time Simulator (DRTS) in their simulation. In [13], the authors attempted to analyse the interaction between the electrical grid and the communication networks. These works interface real-time simulators and communication network simulators, which have two different behaviours (i.e. continuous vs discrete simulation). Most of these works, however, take only into consideration the interaction among component and communication aspects of a power system and do not include other layers and aspects of the Smart Grid Architectural Model (SGAM) [16] which could have further different timing requirements (e.g. Functional and Business layers). Other researchers proposed co-simulation infrastructure to enable HIL and PHIL tests instead. The works presented in [17] can integrate real-time simulators with HIL when using low-cost hardware. However, in most of those studies, the HIL is used only with the real-time simulator, and there is no interaction between HIL and software simulators, such as an advanced control system simulator. This kind of interaction also requires particular attention to maintain the overall synchronisation of the co-simulation. Moreover, the interconnection among software, GPPL, and hardware simulators raise complex issues for time regulation and synchronisation [18].

B. RESEARCH CHALLENGES

Building upon what was previously introduced, four broad categories of research challenges—referred to as Research Challenges (RC)—are proposed as follows:

- **RC1 - Time Synchronization, Regulation, and Data Exchange.** These represent common hurdles encountered in a co-simulation infrastructure. Indeed, it must guarantee precise time synchronization and regulation across multiple distributed simulators [6]. Still, potential challenges may arise in dealing with simulators requiring very small time step durations. This is especially relevant when managing time regulation for hardware and software simulation layers, ensuring real-time constraints are met, as the difficulty in maintaining real-time constraints is due to model computation time and minimum time step duration limitations [19].
- **RC2 - Automated Composability.** It refers to the capability of software simulation models or hardware systems to automatically integrate and interact with each other within the co-simulation infrastructure without requiring manual intervention [20]. In the software simulation layers, automated composability offers flexibility and ease of automation compared to hardware components. Software simulators are scripted and configured to automatically perform tasks such as creation, initialization, and connection. This automation streamlines processes and reduces the need for manual intervention. Essentially, automated composability allows for efficient deployment, configuration, and management of simulator instances within the infrastructure. However, while automated composability simplifies the integration process, it does not eliminate complexity entirely. Integration of multiple software simulation models, particularly with real-time models, can still be challenging, especially for complex scenarios. Users must carefully consider interconnections, dependencies, and synchronization aspects between the models to ensure accurate and reliable simulation results. Additionally, as the number of integrated models increases, computational requirements and performance may be affected, necessitating careful consideration of hardware limitations and resource allocation.
- **RC3 - Automated Deployment.** It refers to automatically deploying the co-simulation infrastructure on a cluster without manual intervention [21]. The concept aims to streamline and simplify the deployment process by automating tasks such as defining models, their parameters, inputs/outputs, connections, and initiating simulators on designated cluster nodes. However, the deployment process of co-simulation infrastructure can be quite complex, requiring different expertise, often necessitating the involvement of a co-simulation expert to assist the model developer, hindering adoption by users without specialized knowledge. Moreover, Assigning simulators to appropriate cluster nodes based on their computational requirements poses challenges, as each simulator may have different demands, requiring careful resource allocation to ensure optimal performance. Ensuring horizontal and vertical scalability management of cluster nodes to meet the computational demands of simulators is also a challenge without automated mechanisms, potentially hindering

scalability and resource utilization. Overcoming these challenges is crucial for improving the efficiency and usability of the co-simulation approaches.

- **RC4 - Spatiotemporal Scalability.** It refers to the ability of a system or infrastructure to handle increasing complexity and size over both space and time [22]. This means the system can effectively scale up to accommodate larger and more complex scenarios while maintaining performance and efficiency. Regarding co-simulation for Smart Grid scenarios, spatiotemporal scalability is crucial for simulating diverse and interconnected systems with varying spatial and temporal characteristics. Limitations and drawbacks in achieving spatiotemporal scalability must be thoroughly examined to address potential challenges. For example, ensuring that the co-simulation infrastructure can scale up to handle larger and more complex Smart Grid scenarios without compromising performance or accuracy, identifying and mitigating potential bottlenecks or limitations that may arise as the system scales, such as computational resource constraints or communication overhead, and addressing the trade-offs between scalability and other system characteristics, such as simulation accuracy or runtime efficiency.

C. RESEARCH CONTRIBUTIONS

This paper presents an innovative Hybrid Multi-Model Co-simulation Infrastructure that allows the interconnection among multi-model simulation software based on DSPL (e.g., MATLAB Simulink, Modelica, EnergyPlus) and GPPL (e.g., Python, C++, Java) with different spatiotemporal scales, as well as the integration of HIL with commercial Digital Real-Time Simulator (DRTS) (e.g. OPAL-RT) in a distributed and shared co-simulation environment among different computers and servers [23], [24]. These diverse simulation strategies enable the integration of models from different multidisciplinary Smart Grid contexts within the infrastructure to address the multidisciplinary requirements necessary to tackle Smart Grid Architectural Model (SGAM) complexity. With respect to our previous works [25], [26], the infrastructure exploits a soft real-time approach where the pure software co-simulation environment runs at the wall-clock time, mimicking a real-world scenario to couple pure software co-simulation with the hard real-time constraints of DRTS. The soft real-time approach is not obliged to respect the real-time constraints precisely and must allow the possibility to run slightly in overrun since a normal software co-simulation environment and its behaviour (e.g., building thermal dynamic) does not impact the fast transient of a power grid. The soft and hard real-time environment communication is ensured by VILLASnode [27], a near real-time middleware, to ensure the correct data exchange among simulation models and platform layers. By employing this novel strategy, the Hybrid Multi-Model Co-simulation Infrastructure ensures the correct wall-clock time evolution of the co-simulated Smart Grid scenario and respects the real-time constraints of the interconnected DRTS, allowing to include fast time-stepped simulations of a power grid model. Moreover, the DRTS capabilities enable HIL and PHIL tests of real-world hardware, creating a powerful testbed for innovative power grid technologies and components. The software-only co-simulation environment has been stressed and accelerated to reach a low time step duration to cope with these strict real-time constraints, around one hundred milliseconds, to deploy a realistic Smart Grid scenario where a building can offer demand response and ancillary services (e.g., voltage regulation) to the power grid.

To summarize, the proposed platform pushes the boundaries of the current state of the art by integrating literature and novel technologies to develop a design and simulation framework for analysing innovative Smart Grid systems, trying to tackle some of the research challenges previously discussed. In particular, the novelties are as follows:

- A multi-spatiotemporal approach enhancing composability of integrated Smart Grid scenarios based on heterogeneous and multi-domain models and exploiting multiple data sources (RC2, RC4);
- A versatile distributed software co-simulation infrastructure designed to integrate DSPL and GPPL simulators across various energy applications (RC1, RC3, RC4);
- An effective interface for plugging real-time simulators (i.e.DRTS) to enable HIL and PHIL applications (RC1, RC2);
- A middleware between the software co-simulation infrastructure and the hardware real-time simulators to couple software and real-time simulators, hardware, and real devices through Internet-of-Things (IoT) protocols (RC1, RC3);
- A comprehensive time synchronization, alignment, and regulation of the orchestration of the different infrastructure nodes exploiting IEEE1588 Precision Time Protocol (PTP) (RC1).
- A scenario design process to foster the communication, collaboration, and knowledge integration of the domain experts in a multidisciplinary research community (RC2);
- The development of the Scenario Design and the Scenario Builder modules to support the user definition of complex Smart Grid scenarios through a high-level comprehensive automated tool (RC2, RC3);

D. ROADMAP OF THE MANUSCRIPT

The structure of this paper is as follows. Section II delves the Hybrid Multi-Model Co-simulation Infrastructure. Section III presents the scenario exploited for testing the proposed infrastructure. Section IV proposed the experimental results of co-simulating the proposed scenario. Section V discusses the infrastructure challenges, trade-offs and limitations. Section VI highlights some useful applications using the co-simulation infrastructure and the

potential users and stakeholders. Finally, Section VII reports concluding remarks and future works.

II. CO-SIMULATION INFRASTRUCTURE

The proposed Hybrid Multi-Model Co-Simulation Infrastructure makes it possible to simulate complex Smart Grid systems, offering designers a comprehensive tool to easily interconnect heterogeneous models in the infrastructure in a plug-and-play fashion. The infrastructure leverages a multi-model view in which the designer can choose among several interchangeable versions of the same simulated model from different engines (i.e., GPPL, software simulators, hardware simulators) depending on the required spatiotemporal scalability. This vision ensures the ability to simulate the fast-to-slow temporal evolution of a Smart Grid scenario in a single co-simulation infrastructure, freeing up the potential to scale the scenario under analysis by choosing the right combination of models in a distributed infrastructure. The infrastructure also provides the flexibility to develop innovative component models following the appropriate interfaces for data exchange, unlocking simulation capabilities for large-scale scenarios. The designer can choose from simulation software based on DSPL (e.g., MATLAB Simulink, Modelica, EnergyPlus), GPPL models (e.g., Python, C++, Java), and commercial DRTS models (e.g., OPAL-RT). Finally, the infrastructure offers the possibility of replacing simulation modules with the equivalent HIL or PHIL device for laboratory testing. In addition, real-world applications and services can be integrated to assess their functionalities in a co-simulation environment.

The overall infrastructure is presented in Fig. 1, consisting of three main vertical layers: *i)* the Data Source Layer, *ii)* the Co-simulation Layer, and *iii)* the Application Layer. The rest of this section will describe each layer in depth.

A. DATA SOURCE LAYER

The Data Source Layer contains the sources of information needed to describe a scenario, offering standard interfaces to access, query, and retrieve data from other infrastructure layers. It includes several Database (DB) modules, which are: *i)* the Geographic Information System (GIS) data sources to provide georeferenced data for models of the co-simulation environments that make up a specific scenario (e.g. Census Data, Cadastral Maps, Digital Elevation Models, or Weather Data, *ii)* the Models data sources to provide basic elements for model definition, such as Building Archetypes & Data, Energy Networks (e.g., district heating, power network), Markets Data and Technologies Data of various energy systems, and *iii)* the Third-party Sources to support the integration of additional third-party data sources.

This layer aims to tackle the RC2 and RC4 (see Section I-B) through the spatiotemporal and scalable model in the Models data sources and their linked georeferenced data coming from the GIS data sources.

B. CO-SIMULATION LAYER

The Co-simulation Layer is the core of the Hybrid Multi-Model Co-simulation Infrastructure. It sets up the co-simulation environment, retrieving all the required information from the Application Layer (see Section II-C). It is composed of three main horizontal layers, namely *i)* the Middleware Layer, *ii)* the Software Simulation Layer, and *iii)* the Hardware Simulation Layer.

1) MIDDLEWARE LAYER

It is the central component of the Co-simulation Layer that provides the communication between the soft real-time co-simulation of the Software Simulation Layer (see Section II-B2) and hard real-time simulation environments of the Hardware Simulation Layer (see Section II-B3). It enables this communication by exploiting VILLASnode [28], a toolset for local and geographically distributed real-time co-simulations. It consists of several components called Nodes that allow flexible interconnections among different technological software (e.g. GPPL) and hardware (e.g. RTDS, OPAL-RT) components exploiting various communication protocols (e.g. Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Message Queuing Telemetry Transport (MQTT)) and data format (e.g. JSON, CSV). VILLASnode acts as an adaptation layer accelerating the redirection of messages among different Nodes to ensure negligible communication latency, thus not impacting the near-real-time approach implemented to manage the time regulation among the Software Simulation Layer and the Hardware Simulation Layer. To ensure the proper synchronisation with the infrastructure, VILLASnode resides in a server synchronised by means of IEEE1588 PTP. PTP ensures the internal clock synchronisation of each interconnected cluster node with a precision up to tens of nanoseconds. The Co-simulation Orchestrator Engine (COE) and DRTS Nodes permit the exchange of information between the Software and the Hardware Simulation Layers. Moreover, VILLASnode allows the interconnection of PHIL and HIL Nodes by exploiting common Internet communication protocols. The Middleware Layer includes a Scenario/Simulation DB based on a time-series database (i.e. InfluxDB) to store information about the scenario and collect all results coming from the overall co-simulation environment via the DB Node. These results will be visualised via the Application Layer modules (see Section II-C). Finally, VILLASnode can support different Interface Nodes that could connect real-world applications and services that want to feed the co-simulation environment. For instance, a real-world service for consumer aggregation could be interconnected to the co-simulation infrastructure to test its performance in a protected environment.

This layer aims to face both RC1 and RC3 (see Section I-B) by offering a flexible data exchange management synchronized with the overall Hybrid Multi-Model Co-simulation

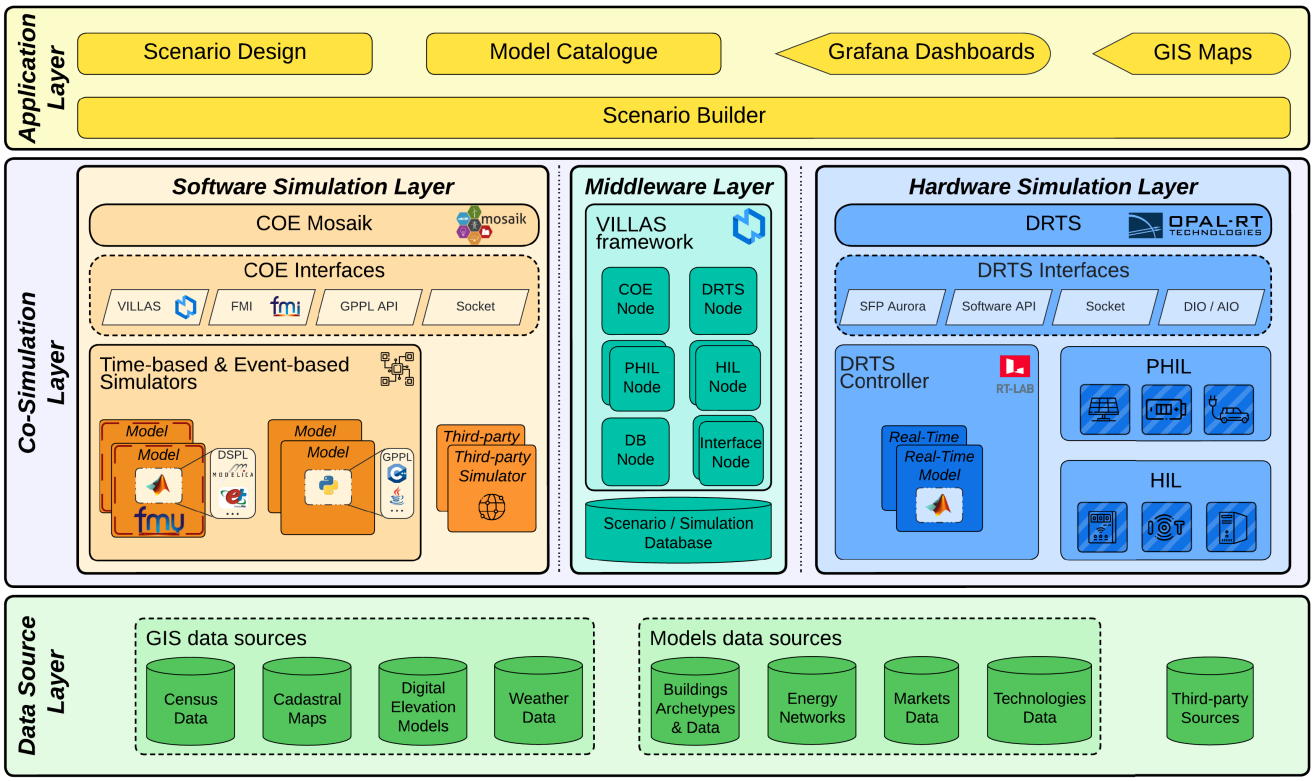


FIGURE 1. Layered architecture schema of the Hybrid Multi-Model Co-simulation Infrastructure.

Infrastructure and the possibility to set the automated deployment of the VILLASnode required in a particular Smart Grid scenario.

2) SOFTWARE SIMULATION LAYER

The Software Simulation Layer exploits a pure software co-simulation framework leveraging upon Mosaik [29], a Python-based framework initially designed to co-simulate Smart Grid scenarios, which is easily extensible to cope with different energy domains. Its main core is the COE Mosaik, which handles the initialisation, the data exchange management, and the time regulation and synchronisation of simulators and their model instances.

During the initialisation phase, the COE Mosaik passes to all interconnected simulators their parameters (e.g. the number of model instances, time step duration, start date, end date) and model instances input/output relationships with other simulators' model instances. This important task is managed through the Scenario API that offers a common setup procedure for setting the scenario. Moreover, COE Mosaik manages the time regulation and synchronisation of both Time-based and Event-based Simulators. Time-based Simulators evolve their model instances with a constant time stepped evolution. Vice versa, Event-based Simulators wait for specific asynchronous events to trigger their model instances' internal state changes and then forward their outputs to other simulators as events.

Instead, data exchange management is achieved by exploiting the COE Interfaces that allow forwarding the initialisation information, the time regulation and synchronisation commands, and the model instances inputs/outputs to each interconnected simulator. Moreover, the COE Interfaces allow the distribution of simulators and their models on different computer clusters, enhancing their vertical and horizontal scalability. Finally, these interfaces could enable the interconnection of a Third-party Simulator to prevent Intellectual Property Right issues by exploiting the Simulator API. Thus, third-party companies can plug their models into a broader scenario without sharing their engines.

Mosaik offers two main COE Interfaces by design: *i)* the GPPL Interfaces that allow the interconnection of models designed with common programming languages (i.e. Python, C++, Java) to exploit their simulation libraries (e.g. pandapower [30], pandapipes [31]), and *ii)* the Socket Interfaces that permits to interconnect distributed simulators via TCP or UDP protocol.

In our previous work [26], Mosaik has been extended by integrating an FMI Interface for FMI standard [25] that permits the coupling of the models even if they are based on different DSPL and their simulation software (e.g. Modelica, EnergyPlus). In this paper, we present a second extension of our previous work in [26] by integrating the VILLAS Interface to exploit the Middleware Layer in exchanging co-simulation results with the Hardware Simulation Layer.

To that purpose, the Software Simulation Layer employs a soft real-time regulation and synchronisation that executes all models considering a common wall-clock time evolution synchronised with the Hardware Simulation Layer by means of the IEEE1588 PTP standard [32]. The cluster nodes, where the Software Simulation Layer resides, are equipped with a proper PTP board that receives the proper synchronisation packets from a Global Positioning System (GPS) follower clock.

This layer deals with RC1, RC3, and RC4 (see Section I-B) by managing data exchange, time synchronization aligned with the overall Hybrid Multi-Model Co-simulation Infrastructure, and time regulation among spatiotemporal scalable software simulators in a horizontally scalable infrastructure, enabling automated deployment of the simulators required by the Smart Grid scenario.

3) HARDWARE SIMULATION LAYER

It allows the interconnection of commercial Digital Real-Time Simulators (e.g. OPAL-RT or RTDS Technologies) to the proposed co-simulation infrastructure. This layer can run specific models of components that require a hard real-time execution (e.g. power grid, electric vehicle charging system). Moreover, it allows Hardware-In-the-Loop (HIL) and Power Hardware-In-the-Loop (PHIL) to test and validate a real device in a protected virtual environment, avoiding huge deployment costs and the associated risk of deploying the component in a real-world environment.

It is worth noting the parallelism between the COE Mosaik and the DRTS hardware, which is the core of this layer. The DRTS exploits the common digital real-time capabilities of self-regulating the time evolution with small time step durations of around tens of microseconds. So, the Hardware Simulation Layer exploits the real-time capabilities of the DRTS, resulting in an uncontrollable simulation system that evolves independently. The near-real-time approach slows down the Software Simulation Layer to keep up with the execution of the Hardware Simulation Layer. This approach enhances the vertical scalability of the number of models but increases the execution time of the simulation. However, this trade-off can be seen as the price of having a comprehensive digital counterpart of a real scenario in the digital twin approach.

To ensure the proper synchronisation with the overall infrastructure, the DRTS is equipped with an IEEE1588 PTP board that receives the PTP synchronisation packets from the external GPS follower clock. It offers by design different DRTS Interfaces to communicate with external entities. The most important interface is the Software API that allows receiving the compiled Real-Time Models from the DRTS Controller (e.g. RT-LAB or RSCAD). From the DRTS Controller application, the user can design and develop one or more Real-Time Models exploiting its Graphical User Interface (GUI), and compile, load and execute them on top of the DRTS hardware. The Software API is also capable of receiving commands from external entities. In fact, it serves as the main interface with the Scenario Builder of the Application Layer (see Section II-C) that is capable of loading and running a Real-Time Model already compiled by the DRTS Controller and saved in the Model Catalogue (see Section II-C).

The DRTS simulator offers other DRTS Interfaces that manage the interconnection with real devices unlocking HIL or PHIL simulations providing *i)* the Analog Input/Output (AIO) and the Digital Input/Output (DIO) Interfaces to directly plug in and integrate devices via their analogue and digital hardware interfaces; *ii)* the Small-form Factor Pluggable (SFP) Interfaces to enable a fast communication protocol based on Xilinx Aurora (2 Gbps bandwidth) with external hardware; *iii)* the Socket Interface to integrate protocols such as TCP, UDP, IEC61850 and GOOSE. The Socket Interface is the main interface that allows communication with the Software Simulation Layer via the VILLASnode DRTS Node in the Middleware Layer. Finally, HIL and PHIL devices already plugged into the DRTS can also have a second interconnection with the Software Simulation Layer via the VILLASnode HIL and PHIL Nodes, if such devices already embed socket capabilities.

This layer faces RC1 (see Section I-B) by exploiting DRTS data exchange and time regulation strategies synchronized with the overall Hybrid Multi-Model Co-simulation Infrastructure. Moreover, it tackles RC2, allowing the composability of real-time models into the platform.

C. APPLICATION LAYER

The Application Layer manages the modules describing and composing a complex Smart Grid scenario into a co-simulation environment by selecting the proper models and interconnecting them in a user-friendly, plug-and-play fashion.

The Model Catalogue collects all models from already deployed scenarios. Moreover, it gathers information about the configuration setup of each model, such as its time step duration, initial conditions, required inputs, provided outputs, and available data flows with other models. The Model Catalogue offers users the possibility of extending the collection with new models by choosing one of the different possible simulation environments (i.e. simulation software, GPPL, and DRTS) with the proper co-simulation configuration required to interconnect the model to the Co-simulation Layer.

The Scenario Design module instead supports the user definition of a scenario for developing an innovative Smart Grid system by offering a comprehensive automated tool to interconnect models contained in the Model Catalogue in a plug-and-play fashion. This module offers a standard YAML configuration template to set up the scenario by exploiting a grey-box modelling approach with minimum effort and cost [26]. By exploiting each model configuration and setup information in the Model Catalogue, the YAML template

guides the platform user in designing the scenario, preventing the manual configuration and interconnection of models, which can be error-prone. Section II-D details the Scenario Design process throughout the scenario configuration procedure.

Finally, the Scenario Builder compiles the resulting scenario from the Scenario Design module and communicates its configuration to the Co-simulation Layer, which will instantiate and configure each individual model to execute the co-simulation environment concretely. The Scenario Builder will validate the proposed scenario by exploiting simulator-specific knowledge (e.g. model configuration and setup information) from the Model Catalogue. This will enable the Scenario Builder to identify suitable interconnection and to detect possible inconsistencies in the defined scenario concerning typical input/output relationships captured by the model configuration and setup information, trying to address the issues delved in RC2 (see Section I-B). Moreover, the Scenario Builder will set up each individual simulator software and DRTS for their physical interconnections and network communication to the Co-simulation Layer.

The Application Layer also includes *i)* the Grafana Dashboard to present co-simulation results by retrieving information from the Scenario/Simulation Database in the Middleware Layer, and *ii)* the GIS Maps module instead exploits the georeferenced information from the Data Source Layer and present them into maps to better describe co-simulation results in a spatial scale.

D. MULTI-DISCIPLINARY SCENARIO DESIGN PROCESS

The Scenario Design module contributes to improving the usability and robustness of the Hybrid Multi-Model Co-simulation Infrastructureby facilitating the communication and knowledge integration of the domain experts in a multidisciplinary perspective. In Fig. 2, the process begins with a multidisciplinary team composed of domain experts who collaborate to identify and define the complex Smart Grid scenario. Each domain expert contributes their simulator tool and relevant models to the Model Catalogue. This diverse expertise ensures that the scenario encompasses various aspects of the Smart Grid, such as power generation, transmission, distribution, and control, as well as complex residential, commercial, and flexible industrial buildings, HVAC systems and Distributed Energy Resource (DER) technologies.

To assess the robustness of co-simulation, it is crucial to rely on the individual experts who deploy their models within the platform. The validation of co-simulation robustness is entrusted to these experts, who have in-depth knowledge of their specific models and simulators. This approach allows for thorough testing and verification of the individual models and their interactions in the co-simulation scenario.

During the scenario design phase, the multidisciplinary team, led by the co-simulation expert, establishes the cross-domain configuration procedure. The co-simulation expert validates the connections among the different models and

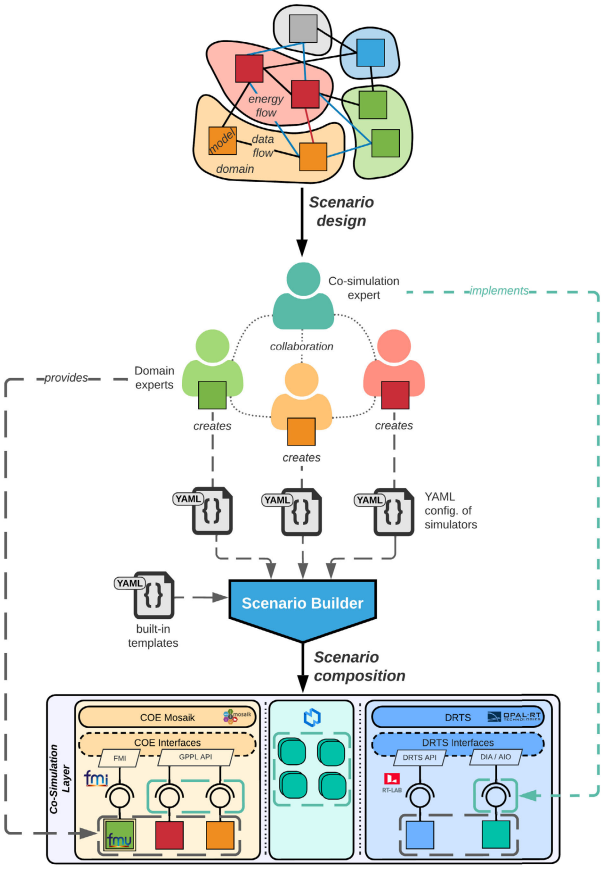


FIGURE 2. Multi-disciplinary scenario design process schema of the Hybrid Multi-Model Co-simulation Infrastructure.

ensures their accuracy and reliability. It is important to consider that each model may have its assumptions, parameters, and limitations, which need to be carefully considered in the validation and calibration process. It is worth noting that the multi-disciplinary scenario design offers a standardized process for integrating simulators and models within the Model Catalogue. This ensures consistency and enables the Scenario Builder to compile the resulting scenario based on the validated connections and configurations established by the experts.

The scenario configuration procedures use a YAML configuration file schema developed to standardise all the information needed to set up the entire co-simulation environment and the individual integrated models. YAML is a human-readable standard for serialising data based on nested objects as ‘‘key: value’’ structures and is commonly used as a configuration file. Fig. 3 depicts the adopted schema of a YAML file describing the structure of the scenario template and its main contents. The schema is based on five root objects, which are described below:

- Scenario Configuration contains the most important settings for the co-simulation and the scenario, such as the name of the scenario, the start date of the simulation to set the time-series data to the selected start date, the

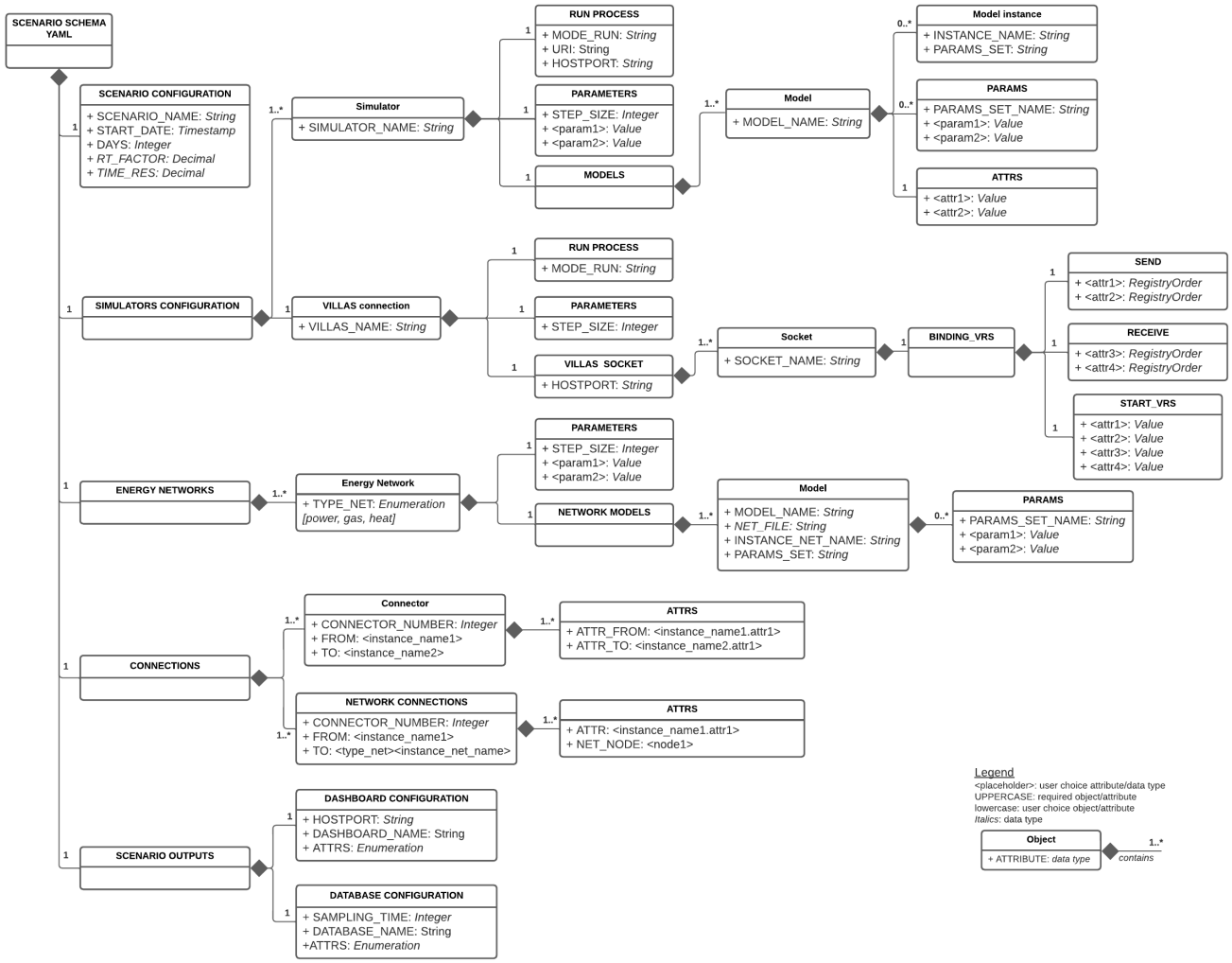


FIGURE 3. YAML schema for scenario configuration of the Hybrid Multi-Model Co-simulation Infrastructure.

number of days to simulate, the real-time factor, which must be equal to one if the simulation is combined with real-time simulators and the time resolution to set the default time step resolution of the software simulators;

- Simulators Configuration contains the information about the instances of the models with their simulation settings and initial conditions. In particular, the Simulator object shall contain the information about the running process (i.e. the simulator configuration as required by the COE of the Software Simulation Layer) and the simulator parameter settings, including the simulator step size. The Model object may contain specific parameter sets to characterise the model instances. Still, it must also specify the attributes used as inputs/outputs for coupling models and any attributes that the modeller wishes to save or display in the dashboard. Another type of Simulator object is the VILLAS connection, which is specifically provided for setting up communication and data exchange with the Middleware Layer;
- Energy Networks provides the settings for the integrated energy network simulators using the Python libraries pandapower and pandapipes;
- Connections contains the information topology of the connections between the instances of the models and the energy networks, exploiting a list of the runtime data exchanged for each connection;
- Scenario outputs contain the necessary settings to display the desired variables via the dashboard or save them using databases.

The use of YAML by the platform users makes it possible to simplify the composition of scenarios by using the default settings of the objects and focusing only on changing the desired parameters in a plug-and-play fashion. Combining different YAML files with template objects that use the same schema into a single YAML configuration file is possible. For example, a modeller can focus on defining the parameters and connections of a significant model instance in coordination with the other experts and the requirements of the scenario while later adding integrated YAML templates

to set up the interface of the connection protocol and the common interface of the shared simulation environment interface. When the platform user explicitly requests the output of the desired variables under the YAML root object scenario outputs, the Scenario Builder automatically adds and connects backend components required to provide the functionality of the co-simulation infrastructure layers. It is worth noting that the semi-automated YAML configuration scheme serves as a bridge towards a fully automated process, i.e. from the high-level description of the scenario to the implementation and execution in the co-simulation environment.

Finally, the Scenario Builder plays a crucial role in validating the proposed scenario by leveraging simulator-specific knowledge from the Model Catalogue. The Scenario Builder can identify suitable interconnections and detect any inconsistencies in the defined scenario by exploring the model configuration and setup information. This validation process enhances the robustness of the scenario by ensuring that the input/output relationships captured by the model configurations align with the desired simulation outcomes.

By following this multidisciplinary approach, involving domain experts, co-simulation experts, and the Scenario Design and Builder modules, the proposed infrastructure aims to provide a robust and reliable platform for generating Smart Grid scenarios, trying to tackle the issues as stated in RC2 and RC3 in Section I-B. The comprehensive validation, calibration, and verification processes help ensure the fidelity of the overall simulation and enhance the infrastructure's robustness under different conditions and uncertainties commonly encountered in Smart Grid scenarios.

III. DEMONSTRATIVE SCENARIO AND MODELS DESCRIPTION

This section describes the Smart Grid scenario and simulator models exploited to test both the capabilities and performance of the proposed Hybrid Multi-Model Co-Simulation Infrastructure. The scenario proposes the provision of flexibility and effective ancillary services to the power distribution grid via the growing penetration of DERs installed on the building site and the deployment of smart Building Energy Management System (BEMS) and control strategies [33]. In particular, the service is provided directly by the BEMS of a building equipped with a Photovoltaic (PV) and a battery system coordinated with a Voltage Control System (VCS).

The proposed scenario represents a complex and realistic application of Smart Grid technologies, highlighting the challenges and complexities inherent in real-world deployments. The scenario focuses on enhancing flexibility and providing effective ancillary services to the power distribution grid by leveraging DERs installed on building sites and deploying BEMS and control strategies. Integrating DERs, such as photovoltaic systems and battery storage, with advanced control strategies presents several challenges. These include managing intermittent renewable energy generation, optimizing energy storage usage, and coordinating distributed energy

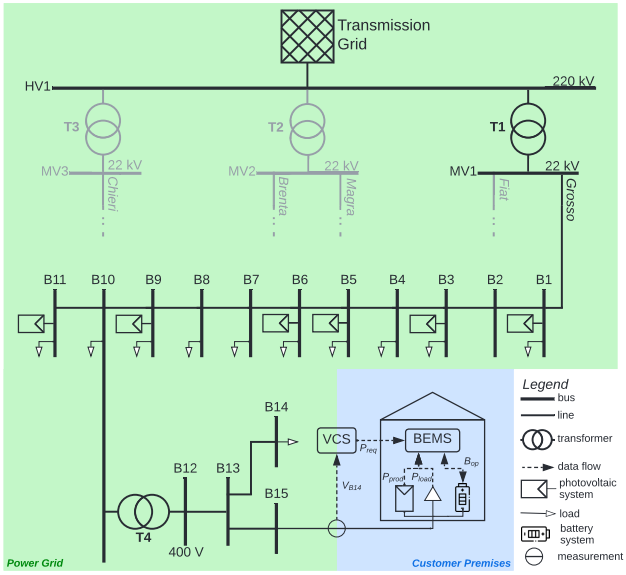


FIGURE 4. Schema of the MV/LV power grid and the low voltage regulation system coupled with the building energy management system.

resources to provide grid support services. By simulating this scenario using the proposed Hybrid Multi-Model Co-Simulation Infrastructure, researchers can assess the capabilities and performance of the system under various conditions. This includes evaluating the effectiveness of control strategies in maximizing energy flexibility, assessing the impact of DER penetration on grid stability, and optimizing the coordination of DERs to provide ancillary services. Overall, the chosen scenario encapsulates the complexity of real-world Smart Grid applications, offering a comprehensive testbed to evaluate the proposed co-simulation infrastructure's effectiveness in addressing the challenges posed by integrating diverse energy resources and advanced control strategies into the grid.

Fig. 4 shows the overall energy scenario, which consists of an area of interest of a real Italian MV/LV network with real demand and generation profiles obtained from an Italian Distribution System Operator (DSO), and customer premises connected to the distribution grid. The grid was taken from [34], and it consists of five main MV feeders, of which Grosso feeder in Fig. 4 was considered to perform the scenario simulation. The test feeder consists of 11 buses with 10 equivalent loads and 6 equivalent power injections from PV plants. These injections and withdrawals represent equivalent power system models of the external areas of the grid. The $B10$ bus extends towards the $B13$ bus the LV network where an equivalent LV load is present in bus $B14$ and the building energy system with the LV point of measure and the VCS in bus $B15$. The grid was placed under severe energy demand by reasonably increasing the loads, causing a decrease of the buses' voltage towards the lower voltage limit to demonstrate the capability of the VCS and test the platform. Indeed, the VCS measures the voltage

across the LV bus $B15$ and sends the power requests to the BEMS only if the measured voltage exceeds the tolerance of 10% around the nominal voltage value in both directions, i.e. 0.9 $p.u.$ and 1.1 $p.u.$, in compliance with the European standard EN 50160 CENELEC [35]. After receiving the power request, the BEMS evaluates the availability of the internal energy resources (i.e., PV production and battery capacity) based on the building load characteristics to decide if the building can provide the total amount of the power requested or a part of it and for how long. The VCS is based on a fuzzy logic Proportional Integral Derivative (PID) controller that can operate at different sample times. For example, it could be deployed directly on the control system of the distribution grid, operating at very short sample times. In this scenario, the VCS is considered a cloud service over the Internet, taking the real voltage measures and communicating the power requests to BEMS. Finally, the customer premises encompass the building with all the interacting elements, both physical systems such as the PV system, battery, appliances and the building envelope, and cyber-physical components such as BEMS and human activities.

Each energy scenario element was modelled and implemented into the Hybrid Multi-Model Co-simulation Infrastructure. Fig. 5 depicts the scheme of simulation blocks and their connections among them building up the scenario. In particular, the blocks represent the cyber-physical components containing the standalone models related to the customer premises and the power grid. The simulation blocks interact with the shared simulation environment, linking them through different kinds of connection based on the simulator typologies, i.e., real-time hardware (blue blocks) and pure software (orange blocks) simulators are connected throughout the Mosaik COE and DRTS Nodes. In addition, the blue striped block represents the HIL component, i.e., the Physical Smart Meter. To build up the energy scenario, the simulation blocks are added, parameterised and connected through the YAML configuration files provided by the domain experts through the Scenario Design and finally built by the Scenario Builder that instances the models, composes the overall scenario automatically, and executes the co-simulation.

The following subsections describe the characteristics of the models implemented for the scenario under analysis, which are divided into *i)* Power Grid and *ii)* Customer Premises.

A. POWER GRID

In this scenario, the environment is modelled by the Weather module that provides meteorological data to the system, retrieving the information from third-party data sources, e.g., dry-bulb temperature, dew-point temperature, relative humidity, barometric pressure, direct normal radiation, diffuse horizontal radiation, total and opaque sky cover, wind direction, and wind speed. The DRTS simulator has been used to execute the simulation of the power grid running the MV/LV Power Grid model. The aggregated MV/LV Load Profiles and PV Systems generation profiles have been used for all the buses with the only exception of the bus $B15$, which is connected with the high-detailed model of the customer premises. A Physical Smart Meter device is connected between the grid and the simulated household employing an analogue output block in the DRTS model [36]. Therefore, the analogue voltage and current measurements from the analogue output board are collected from the DRTS. Moreover, the smart meter communicates the RMS values to the other component of the infrastructure that requires the measurement as input via the HIL Node of VILLASnode. A more detailed description of the above-mentioned simulators and the device can be found in our previous works [26], [34], [36], respectively.

The LV Voltage Controller implements the VCS, which is based on fuzzy logic PID controller monitoring the voltage V_{B15} and sending the request capacity P_{req} to the BEMS if the voltage exceeds the tolerance. The controller calculates the error value as the difference between the desired min/max tolerated voltage and the measured value V_{B15} and applies a correction based on proportional, integral and derivative terms on the P_{req} that will be requested to the BEMS, thus minimising the error over time. The model is developed in Simulink using the Fuzzy Logic and Control System Toolbox, and integrated via the FMI interface of the COE Mosaik.

B. CUSTOMER PREMISES

The customer premises represents the building energy system with its physical components modelled by coupling diverse simulators, described in the following sections. In particular, the Building Envelope simulates the building thermal dynamic; the Household Behaviour simulates the occupancy and the use of light and appliances in terms of energy consumption derived by the action of each inhabitant in the house; the Rooftop PV System (block in orange in Fig. 5) provides a more granular PV profile than the PV systems simulator (block in blue in Fig. 5), as it uses more granular spatiotemporal data and detailed technical information of the building rooftop solar potential; finally, the Electric Heat Pump (EHP) and its EHP Controller simulate the operations of the heating system to control and maintain the desired indoor temperature. A more detailed description of the above-mentioned simulators can be found in our previous works [25], [26], while the interconnections of these models within the scenario are depicted in Fig. 5.

The Battery System model was designed in MATLAB Simulink and embedded into the DRTS to couple it with the power distribution network in real-time to describe the fast dynamics correctly. The model simulates the dynamic behaviour of a Li-Ion battery, and it can be fully parameterised using a commercial battery datasheet. The battery model was electrically connected on the behind-the-meter Point of Delivery (POD) of the household. The main state

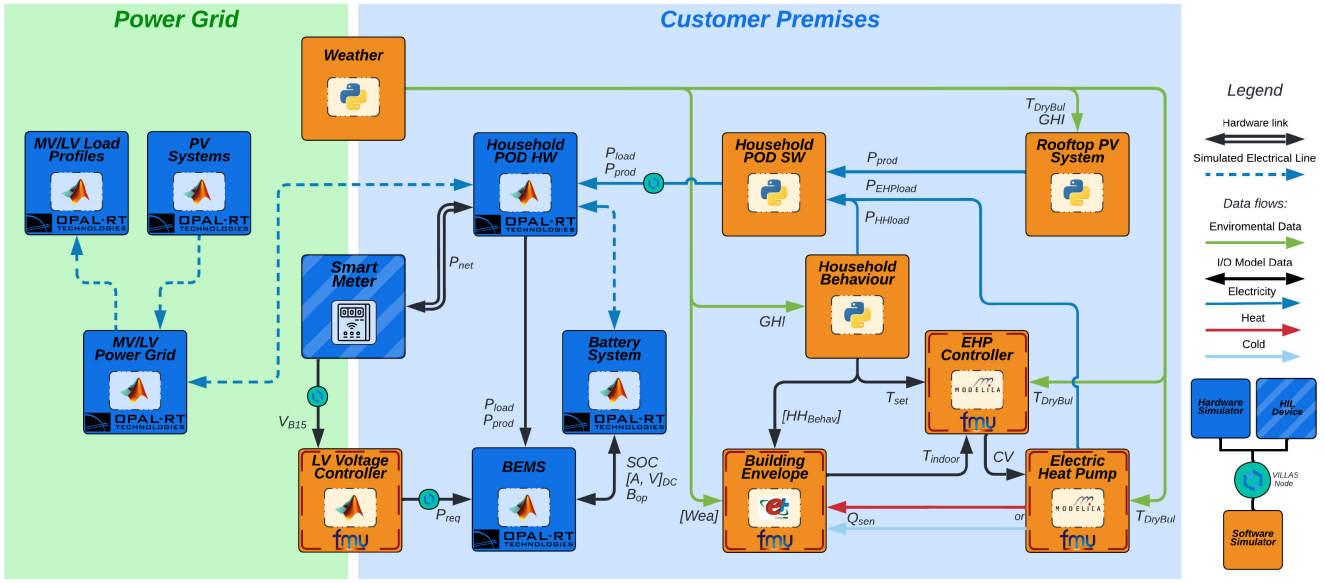


FIGURE 5. Simulator block diagram of the energy scenario designed within the Hybrid Multi-Model Co-simulation Infrastructure.

variables of the battery (i.e., *SOC*, *A* and *V* in Direct Current (DC)) are sent to BEMS that manages the energy fluxes in the building though the battery signal operations B_{op} based on the requests of the VCS. The battery model is set to simulate the charge and discharge of the battery with a resolution of 10 minutes.

The Household POD HW represents the physical point of withdrawal and/or injection of electricity into the distribution network. It was modelled in Simulink as a PQ Load element and embedded into the DRTS. It is physically connected through a hardware link with the Smart Meter to exchange the power measures and electrically connected through real-time simulated electrical lines with the behind-the-meter battery system and the front-of-the-meter power distribution grid. In particular, the Household POD receives the power-related data signals from the pure software simulators of the building through COE Node of VILLASnode and translates them into real-time simulated Alternate Current (AC) voltage and current. Moreover, It sends the power withdrawal or injection data to the BEMS.

The Household POD SW provides the data interface between the hardware Household POD and the pure software simulators of the building. It collects the PV electrical generation (P_{prod}), the aggregated household electrical load (P_{HHload}), the EHP electrical consumption ($P_{EHPload}$), and main state variables of the battery (*SOC*, *A* and *V* in DC). It was used as a data collector manager, returning the simulation results either in run-time or at the end of the simulation. The smart meter simulator can perform the simulation at any time resolution without any limitations.

The Building Energy Management System is a rule-based control algorithm that manages the charge and discharge of the battery under the depth of discharge limit, prioritising self-consumption and, eventually, charging the battery only from the surplus of PV production. However, when the BEMS receives the requests for upward or downward capacity from the VCS, it verifies if there is space for providing the requested flexibility or not by directly controlling the battery parameters for further injection/withdrawal or the PV system only for withdrawal.

IV. EXPERIMENTAL RESULTS

The scenario presented in Section III was simulated in order to test the functionalities of the presented Hybrid Multi-Model Co-simulation Infrastructure.

A. SOFTWARE, HARDWARE, AND NETWORK SETUP

The Co-simulation Layer is deployed among five servers and one OPAL-RT OP5700 (i.e. a DRTS), which are all interconnected between them via a 10 Gbps Ethernet switch minimising the latency in data exchange.

The first server hosts the COE Mosaik, its Interfaces, and all the Middleware Layer entities in Fig. 1. The other four servers instead host all the software modules in the energy scenario in Fig. 5 as follows: *i)* the second server runs the Household Behaviour, the Rooftop PV System, the Household POD SW, and the Weather modules that are software simulators in Python; *ii)* the third server hosts the Building Envelope modelled with EnergyPlus; *iii)* the fourth server executes the LV Voltage Controller developed in MATLAB Simulink; *iv)* the fifth server runs the Electric Heat Pump and the EHP Controller as OpenModelica models. All these models communicate with the COE Mosaik by implementing the FMI and GPPL API Interfaces.

The OPAL-RT OP5700 hosts the following modules of Fig. 5: *i)* MV/LV Power Grid, *ii)* MV/LV Load Profiles, *iii)* Photovoltaic System, *iv)* Household POD, *v)* Battery System, and *vi)* BEMS. To ensure the time synchronization between

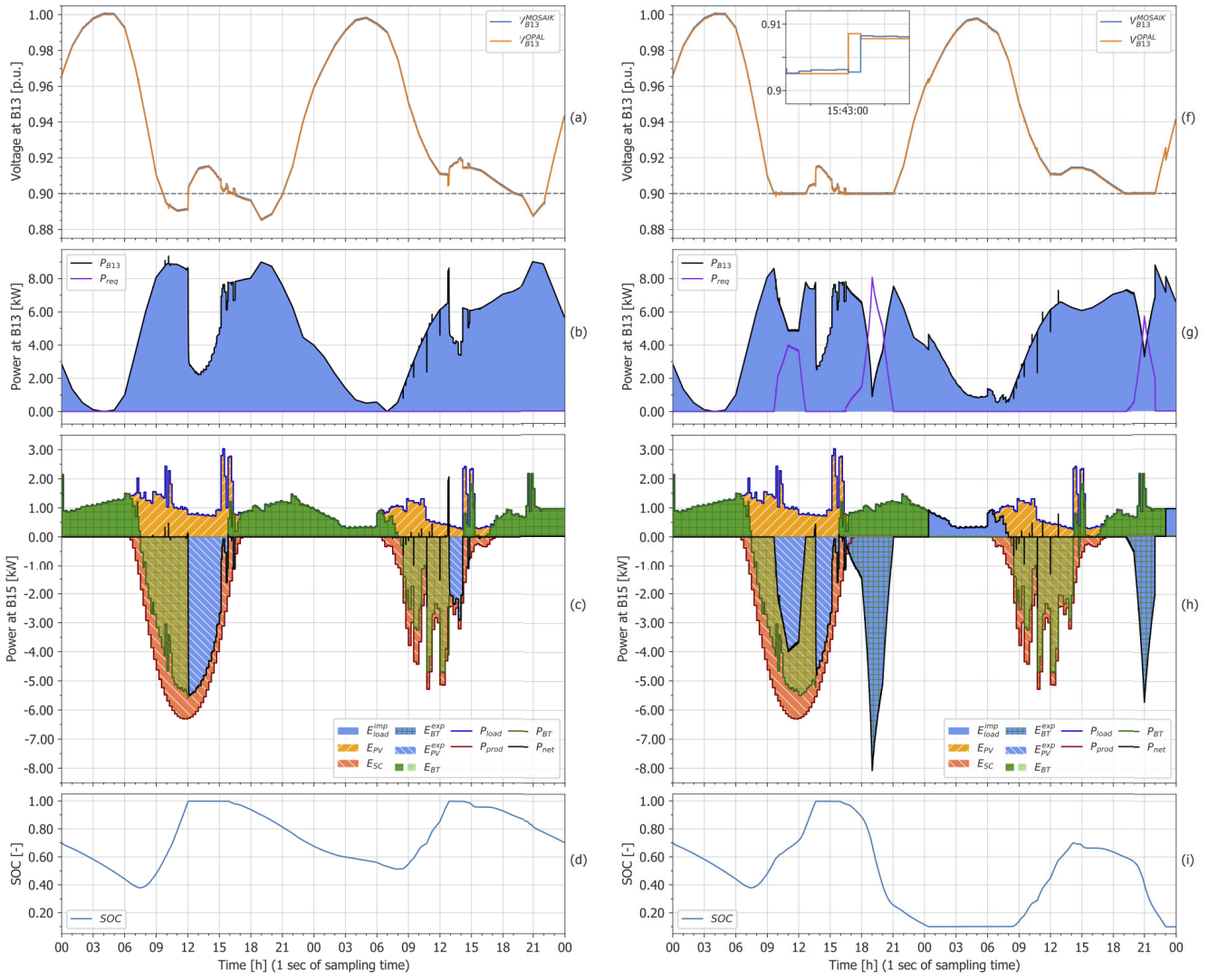


FIGURE 6. Simulation results of two consecutive days during the thermal season by considering the use cases without voltage control Business As Usual (BAU) (*on the left*), and with activated voltage control system VCS (*on the right*). In particular, plots (c) and (h) show the power and energy exchanges at bus B15 measured on the customer premises, i.e., related to the battery (P_{BT}, E_{BT} and E_{BT}^{exp}), PV (P_{prod}, E_{PV} and E_{PV}^{exp}), consumption (P_{load} and E_{load}^{imp}), net power and self-consumption (P_{net} and E_{SC}).

the server and the OPAL-RT OP5700, they are all equipped with an Oregano Syn1588 PTP synchronization board that implements IEEE1588 PTP standard.

Finally, the Physical Smart Meter is connected to the OPAL-RT OP5700 via an AIO DRTS Interface from which the meter retrieves the voltage and current of the monitored LV Bus. Then, the meter computes both RMS and phase and sends these results to the rest of the infrastructure via the HIL Node in the Middleware Layer.

B. SCENARIO SETUP

The standardized YAML configuration files were filled through the *Scenario Design* with all data and parameters required to set up the co-simulation environment, simulators and their models, as well as connections among them as depicted in Fig. 5. The relevant model settings are described in the following.

The models related to the customer premises were parameterized and set as in [26]. In particular, the household consists of four members, and the installed PV system provides 10 kW_p to the premises. The time steps were set considering the capability of the models' solvers and computational effort, as well as the needs for the real-time hardware coupling: EnergyPlus Building Envelope 10 *min*, Modelica Electric Heat Pump 5 *min*, Rooftop PV System 15 *min*, Household Behaviour 10 *min*, Matlab Simulink LV Voltage Controller 100 ms, and Household POD SW 100 ms. In particular, these last two were set with smaller time steps as they shall be coupled with the hardware simulators, as depicted in Fig. 5. The *Weather* module provides data to each simulation engine at the requested time step by interpolating the original data

sources. All the models implemented in OPAL-RT (i.e., the MV/LV Power Grid, the MV/LV Load Profiles, the PV Systems, the Battery System, the BEMS, and the Household POD HW) run with a time step duration of 50 µsec. Moreover, it was assumed that for all the connection points of the grid, the power factor is maintained above 0.95. The Battery System is composed of two battery packs in series with a rated capacity of 60 Ah and a nominal voltage of 200 V. The parameters of the discharge characteristics were derived from the built-in Li-Ion battery Simulink model.

In the end, the Scenario Builder module parses the YAML configuration files to retrieve all the required information to perform the scenario simulation and automatically distributes them to the Co-Simulation Layer exploiting the API of COE Mosaik, DRTS OPAL-RT, and middleware VILLASnode.

C. SCENARIO RESULTS

The scenario was simulated by exploiting the real-time capability of the Hybrid Multi-Model Co-simulation Infrastructure for two consecutive days during the heating season by considering the use cases without voltage control, called Business As Usual (BAU), and with the activated voltage control system, called VCS. The main results of the simulation are depicted in Fig. 6 (BAU on the left and VCS on the right). It shows the main characterizing variables for comparing the two use cases bringing out the benefits of the control strategies, as well as the delays and noises that derive from the coupling of pure software and hardware simulators. In particular, Fig. 6*(a)* and Fig. 6*(f)* show the RMS voltage in p.u. measured at bus $B13$. To highlight the round trip time latency due to the HIL and the co-simulation application, the voltage measure was collected directly from the Real-Time Model, V_{B13}^{OPAL}, and indirectly from the Physical Smart Meter, V_{B13}^{MOSAIK}, both in OPAL-RT. Fig. 6*(b)* and Fig. 6*(g)* show the power measured at bus $B13$, P_{B13}, and the power request, P_{req}, for voltage control, both collected from OPAL-RT. Fig. 6*(c)* and Fig. 6*(h)* show the power and related energy exchanges at bus $B15$ measured on the customer premises (see the figure caption for more details), and the battery State Of Charge (SOC) in Fig. 6*(d)* and Fig. 6*(i)*. In particular, the load, P_{load}, and production, P_{prod}, data were collected from Mosaik, while the data related to the battery and the net power and energy exchanges were collected from OPAL-RT. Overall, the data were collected with a sampling time of 1 s. However, it is possible to request information from each module exposing inputs/outputs and parameters or from the simulation environment directly at any sampling time.

By comparing the voltage curves in Fig. 6*(a)* and Fig. 6*(f)*, the VCS together with the BEMS succeed on maintaining the voltage above the tolerance limit of 0.9 $p.u.$ during the whole period of simulation, as opposed to BAU use case in which the voltage falls under the limit due to the high network withdrawals, especially during the peak hours as depicted in Fig. 6*(b)*. Considering the VCS use case, the Fig. 6*(g)* and Fig. 6*(h)* clarify how VCS and BEMS act to control the voltage on the distribution grid. The first capacity request P_{req} occurs from 09:30 till 12:30 of the first simulated day, amounting to approximately 4 kW at the peak. It can be seen that the request is immediately fulfilled by the PV production E_{PV}^{exp} of the building by injecting directly into the grid rather than charging the battery, as highlighted by the reduction of the stored energy E_{BT} in the request period. By analysing the second grid request of the day, from 16:00 to 21:00, of about 8 kW at the peak, the capacity is provided directly from the battery that is discharging to cover the household demand E_{BT} as well as the external request E_{BT}^{exp}. It is important to note that, unlike the BAU use case, the PV production and battery capacity are exploited when needed and if possible by the VCS causing probable inconveniences for the household. For example, as shown in Fig. 6*(h)* and Fig. 6*(i)* as opposed to Fig. 6*(c)* and Fig. 6*(d)*, the battery results completely discharged at about 00:00 and it is not able to cover the household demand till 09:00 of the day after (the SOC reaches the discharge limit of 10 %), naturally obliging withdrawal from the grid during this period. This is a cost for the family that must be adequately covered and remunerated to sustain the network ancillary service. Moreover, this withdrawal, which happens in VCS use case compared to BAU, does not affect the ancillary service because it occurs at night when the demand for electricity in the grid is low.

In the end, the voltage regulation service, provided by the coordinated action of the LV VCS with the BEMS, succeeds in limiting the voltage drop and not letting it exceed the tolerated limits, as highlighted in Fig. 7. In particular, the figure shows the area encompassing the voltage from bus $B1$ in the MV grid towards the LV grid at the final bus $B15$ in both scenarios. It can be observed that in VCS scenario, the voltage is always maintained above 0.9 p.u., while in the BAU scenario the voltage is not controlled and it drops up to 0.885 p.u.. Clearly, the voltage in MV bus $B1$ is not influenced by the LV regulation. In addition, it can be noted that the LV regulation changes the use of the grid also at times when control does not take place, e.g. from 12:00 to 14:00 of the first day and from 13:00 to about 15:00, in which the voltage is lower than the BAU scenario, but still above the tolerated limits.

D. INTRINSIC CO-SIMULATION LATENCIES

The complex interactions among the Co-simulation Layer entities could cause latencies and relative inaccuracies mainly due to two main factors: *i)* the System-of-Systems (SoS) hierarchy of the interconnected Software Simulator Layer modules, and *ii)* the interconnection among the Hardware Simulation Layer and the Software Simulation Layer via the Middleware Layer.

In the Software Simulation Layer, the co-simulation application decomposes the overall integrated energy system in its fundamental components and equipment models. This SoS vision is then applied to the COE Mosaik and imposes the same hierarchy between the simulator inputs

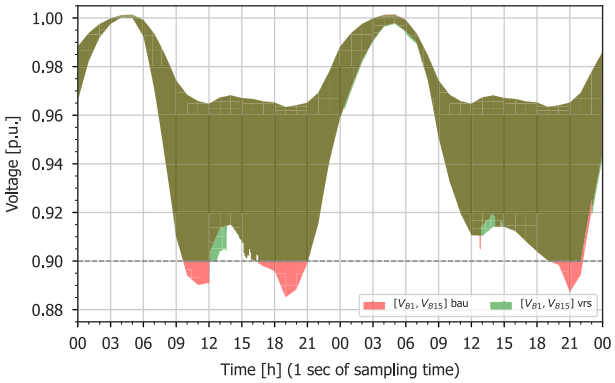


FIGURE 7. Area encompassing the range of voltages buses in the whole MV/LV grid considering both BAU (red) and VCS (green) scenarios.

and outputs, causing latencies multiple of the smallest time step duration of the modules interconnected by the COE Mosaik in the propagation of the effects related to a particular simulator, causing inaccuracies in the overall simulation results. For instance, the Rooftop PV System module receives information about the Global Horizontal Irradiance (GHI) from the Weather module as input, which will affect its output in the next time step. So the latency is exactly a one-time step duration, i.e. 100 ms. Then, the Rooftop PV System simulator generates outputs that are communicated as input to other simulators. For instance, the power production of the PV system is forwarded to the Household POD SW module and will impact its output in the next time step of co-simulation, causing two-time steps of latencies for being affected by the Weather module output. When the output value fluctuations of the simulator are negligible, the effect of this latency is negligible. When the values are strongly fluctuating instead, the effect becomes noticeable and could cause serious inaccuracies. Moreover, this is further exacerbated when the simulators present different time step durations. In Fig. 6*(c)* and Fig. 6*(g)* for instance, the power production of the PV system on the second day between 9:00 and 15:00 strongly fluctuates due to the cloudy day data received in input by the Weather module. This causes inaccuracies in the fluctuations of the Household POD SW calculation due to the latency of the above-mentioned SoS hierarchy.

Another latency contribution is caused by the interconnection among the Hardware and Software Simulation Layers via the Middleware Layer. It is worth noting that these latencies are predominant in the case of HIL and PHIL applications [32]. To calculate this effect during the co-simulation, the results of the RMS voltage have been measured directly inside the Real-Time Model and indirectly from the COE Mosaik. In Fig. 6*(f)*, the direct and indirect measurements of the RMS voltage $B13$ in p.u. are depicted in the zoomed plot. The direct measure V_{B13}^{OPAL} is calculated inside the Real-Time Model and saved in a MATLAB file. It is worth noting that V_{B13}^{OPAL} anticipates the indirect measure V_{B13}^{MOSAIK}. In fact, V_{B13}^{MOSAIK} is experiencing a round trip time latency generated by the operations as depicted in Fig. 8 with their respective timings, and described in the following:

a. the 3-phase voltage signals of bus $B15$ are calculated in the Real-Time Model for each time step (i.e. 50 µsec) and reproduced as three ±5 V signals in the analogue outputs of the OPAL-RT OP5700;

b. the Physical Smart Meter collects the ±5 V signals, calculates each 50 msec the 3-phase RMS voltage values, and sends them to the HIL Node of VILLASnode via UDP in a packet;

c. VILLASnode receives this packet and redirects it to the COE Node via UDP;

d. the COE Mosaik receives the packet and sends it in the next time step (i.e. after 100 msec) to the LV Voltage Controller via TCP that decodifies it to then use the 3-phase RMS voltage values to generate the P_{req} battery request;

e. the LV Voltage Controller codifies the packet and sends it back to the COE Mosaik via TCP in the next COE Mosaik time step (i.e. after 100 msec). The Mosaik COE redirects it to the COE Node in VILLASnode via UDP;

f. VILLASnode receives the packet, translates it to the OPAL-RT data format, and redirects it to the DRTS Node;

g. Finally, the Socket Interface of the OPAL-RT OP5700 DRTS receives the packet and save the 3-phase RMS voltages in a MATLAB file.

It is worth noting that the communication latencies among the entities in the wired network (i.e. the black arrows) are negligible with respect to operations' latencies. The resulting overall round trip time latency can be expressed as follows:

$$RTT = T_{Collect}^{SM} + T_{Process}^{SM} + T_{Redirect}^{VILLAS} + 2 * T_{Step}^{Mosaik} + T_{Redirect}^{VILLAS} \quad (1)$$

where RTT is the overall round trip time;$T_{Collect}^{SM}$ is the Physical Smart Meter collection duration of the ±5 V signals which is a constant latency of 50 ms; $T_{Process}^{SM}$ is the Physical Smart Meter processing time due to the RMS calculation and codification of RMS values in the VILLASnode data format which is a variable latency with an average value of 37.5 ms; $T_{Redirect}^{VILLAS}$ is the time spent by VILLASnode to receive a message from an input Node, process it, and redirect to the right output Node which is a variable latency with a negligible average value of 0.754 µs; and T_{Step}^{Mosaik} is the COE Mosaik time step duration which is a constant latency of 100 ms. The overall RTT results in a variable latency with an average value of 287.5 ms.

V. DISCUSSION ON INFRASTRUCTURE CHALLENGES, TRADE-OFFS, AND LIMITATIONS

The proposed co-simulation infrastructure offers a powerful tool for testing and optimizing Smart Grid systems and their emergent necessity of integrating modern technologies. However, the research challenge solutions described in the manuscript come with certain limitations and drawbacks that should be considered.

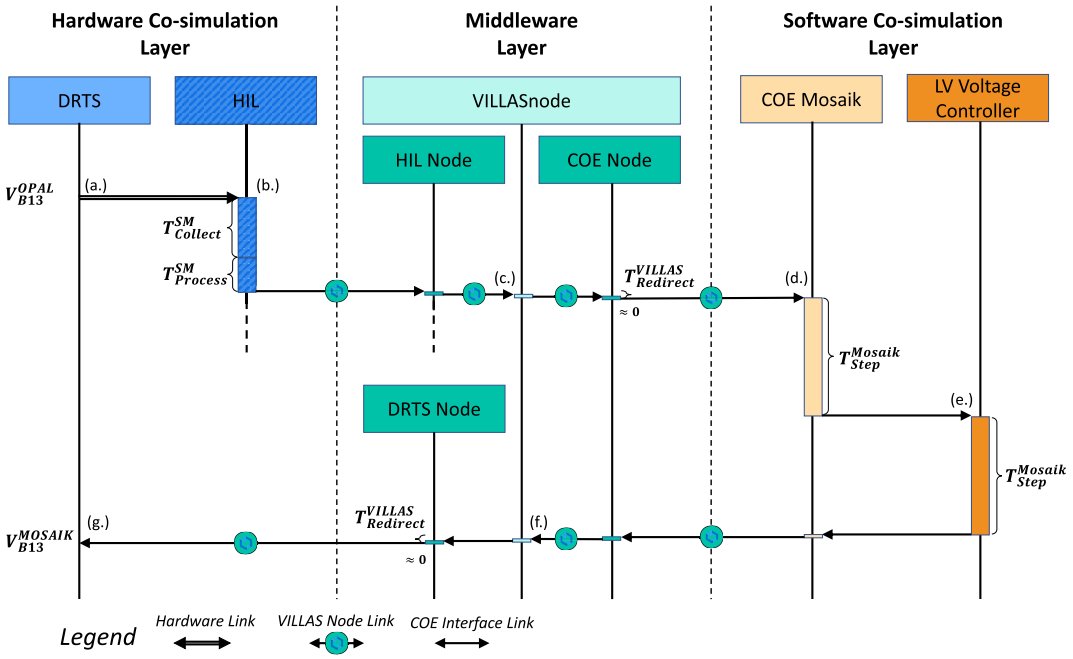


FIGURE 8. Sequence diagram of the direct and indirect measurement operations to calculate the round trip time latency of V_{B13}^{MOSAIK}.

The solution proposed for the time synchronization, regulation, and data exchange (RC1) presents limitations in both Software and Hardware Simulation Layers. The main limitation of the Hardware Simulation Layer is the model computation time that must respect real-time constraints to avoid overrun errors that could compromise the time self-regulation of the DRTS and, so, the overall infrastructure. The main limitation of the Software Simulation Layer instead is the minimum time step duration that cannot act faster than tens of milliseconds. It is difficult to give a clear value because it depends on the model complexity and how many models are running within a time step. Moreover, the near-real-time approach presents a minimal deviation of the time step duration due to the non-deterministic process scheduler of the COE cluster node. This drawback is limited by a Real-Time Operative System (RTOS) that enhances the priority of the process on the scheduler and avoids huge deviations.

Finally, the RC1 solution suffers limitations on the data exchange management related to the limit on the packet size exchanged among the co-simulation entities, which can affect the number of instances managed by each simulator. Furthermore, the complex relationships among simulators instantiated across different cluster nodes can also impact the aforementioned limitations imposed by the orchestrator.

In terms of automated composability (RC2), the Software Simulation Layer offers more flexibility and ease of automation compared to the Hardware Simulation Layer. Software simulators that could be integrated into the infrastructure are scripted and configured to perform the creation, initialization, and connection tasks automatically, allowing for streamlined processes and reducing the need for manual intervention. This automation capability enables efficient deployment, configuration, and management of simulator instances within the infrastructure.

The Hardware Simulation Layer has limitations and drawbacks that should be considered. A potential limitation is the compatibility and interoperability of models within the DRTS environment. In our scenario and setup, OPAL-RT offers automated composability when the model is already designed, developed, and validated with the simulator's framework (i.e. RT-LAB). Suppose a particular model has not been previously designed, developed, and validated. In that case, additional effort may be required to modify or adapt the models to ensure proper integration into the co-simulation infrastructure. Moreover, the automated composability feature simplifies the integration process when the external interconnections of the real-time model are already set. Still, it does not eliminate the complexity of model integration. Combining multiple software simulation models with a real-time model can still be challenging, especially when dealing with complex power system scenarios. Users must carefully consider the interconnections, dependencies, and synchronization aspects between the models to ensure accurate and reliable simulation results. Finally, as the number of integrated models increases, the computational requirements and performance of the DRTS can be affected. Running large-scale simulations with numerous interconnected software models may demand significant computational resources to handle the asynchronous sending and receiving of information inside the real-time simulated environment, such as processing power and memory. Users need to be aware of the hardware limitations and ensure that

the DRTS system can handle the computational demands of the integrated models.

Another limitation of automated composability is the integration of HIL, which necessarily requires manual setup and configuration. While efforts can be made to automate certain aspects of the HIL setup, there are inherent limitations to fully automating hardware interconnection, e.g. hardware components require physical installation, connections, and adjustments that cannot be automated. Therefore, manual intervention is usually necessary to set up and identify hardware nodes within the infrastructure.

Regarding automated deployment (RC3), the infrastructure currently requires manual deployment through the compilation of a YAML file. This task necessitates a co-simulation expert who assists in deploying the setup on a cluster of computer systems. Assigning specific simulators to appropriate cluster nodes based on their computational requirements is a crucial step that requires expertise, limiting the platform's usability. Conversely, the co-simulation infrastructure should be capable of automatically handling large-scale simulations and increased computational demands. It should be designed to scale resources effectively, such as processing power and memory, to accommodate varying simulation requirements. Performance monitoring and optimization techniques can help ensure the infrastructure operates efficiently, even under high loads.

To conclude, the solution to spatiotemporal scalability (RC4) is influenced by various factors that impact the overall Co-simulation Layer. In the Software Simulation Layer, the COE imposes a threshold on the packet size exchanged by the models integrated into the scenario, affecting the number of instances each simulator manages. In our previous work [22], we tested the scalability of the COE, also known as Mosaik, against various co-simulation frameworks. The results demonstrated that Mosaik 2.6 can support highly complex scenarios without exceeding tens of thousands of simulator instances. Furthermore, the latest version of Mosaik 3.0 has been updated within the current infrastructure to handle massive scenarios, surpassing the previous limitations. Indeed, our solution exploits Mosaik 3.0. However, it is important to note that several factors influence the scalability of the co-simulated scenario. Firstly, the communication protocol used by Mosaik (i.e. TCP) and the internal data exchange management strategies impose a threshold on the packet size, affecting the number of instances each simulator manages. Secondly, the complex relationships among simulators instantiated across different cluster nodes could further impact the aforementioned limit. Therefore, scalability depends on the number of simulators handling different real-world entities (e.g. buildings, appliances, power grid, renewable energy sources) and the volume of information exchanged. Moreover, the time resolution of the simulated entities and the real-time nature of the simulation execution can either decrease or increase the scenario's scalability.

In the Hardware Simulation Layer, scalability is influenced by the number of CPUs and cores licensed on the DRTS interconnected to the platform. In this manuscript, the authors integrated Opal-RT, which allows for a certain number of simulable power grid nodes per core. The complexity of the network components (e.g., transformers) could further decrease the dimension of the simulable power grid. Moreover, DRTS has generally a limited number of AIO and DIO and external communication interfaces (e.g. TCP, UDP) that could reduce the number of HIL and PHIL connections. The interconnections among different DRTS can increase the available number of power grid nodes, so the dimension of the simulable power grid. For instance, in our previous work [32], the authors used interface algorithms to split the power system under test into two DRTS, effectively reducing the latency impact on the stability and accuracy of the EMT solution. The current manuscript focuses on a single DRTS, but in future works, the authors will integrate the work in [32] in the former Hardware Simulation Layer to enhance its scalability.

Additionally, we included HIL and PHIL functionality, specifically a real smart meter, to showcase the adaptability of the overall architecture in real-world scenarios. This demonstrates that the infrastructure can test novel prototyping devices even with IoT capabilities, allowing for the reception and injection of measurements or data from the field into the simulated environment and vice versa. In the proposed manuscript, we have demonstrated this integration by including in the simulation scenario a prototype of an IoT-enabled smart meter directly connected to the Hardware Simulation Layer and the Middleware Layer.

To conclude, the Middleware Layer has been demonstrated to successfully communicate with IoT devices, as demonstrated by the integration of a real smart meter in the proposed scenario. This integration showcases the potential for interconnecting a wide range of IoT devices and enables scalable integration with the scenario. Although the technology implemented for the Middleware layer has been demonstrated to deal with a huge number of IoT devices [28], the scalability of these interconnections is affected by the limited capability of AIO, DIO, and external communication interfaces (e.g. TCP, UDP) of the DRTS used in the Hardware Simulation Layer. These limitations can reduce the number of HIL and PHIL connections. As mentioned above, interconnections among different DRTS can increase the available number of power grid nodes and, consequently, the dimension of the simulable Smart Grid and the number of external communications [37].

VI. POTENTIAL APPLICATIONS

The proposed co-simulation infrastructure enables the definition of large-scale multi-energy scenarios, which increasingly characterize the energy landscape under the ongoing energy transition process. As highlighted in [38], this co-simulation infrastructure is a valuable and needed instrument to promote and accelerate energy transition by making the simulation

of complex MES scenarios easier, more accurate, and more efficient. Ease of use will come from tools that automate and validate the definition of involved energy scenarios. Accuracy will come from the inclusion of heterogeneous simulators that capture the multifaceted nature of the scenario under analysis. Efficiency will come from distributed simulations and support for dedicated hardware.

This promises a broad impact for both academic researchers and practitioners working in the energy sector. In particular, our solution is devoted to actors such as: *i)* distribution system operators, who manage the distribution grids; *ii)* balancing service providers, who manage the resources and loads of both consumers and prosumers to generate savings and offer ancillary services to the grid; *iii)* public administrators and decision-makers, who are in charge of drawing policies to plan the energy transition; *iv)* energy managers, who manage energy flows in a building/small district; *v)* energy communities, which are a new emerging actor of the energy sector consisting of single citizens, companies, and public administration, aiming at adopting renewable energy sources and storage systems to produce and consume energy locally and share it in a peer-2-peer market.

Researchers can assess innovative ideas with wide experimental campaigns on the co-simulation platform; likewise, the aforementioned players that gravitate around the energy sector can study the impact of new solutions (new hardware, technologies, deployments, but also policies and regulations) in a safe and realistic test-bed environment. The platform will make decision-making faster and better supported by solid analysis, with time and economic savings.

VII. CONCLUSION

This paper presents the Hybrid Multi-Model Co-simulation Infrastructure to facilitate Smart Grid integration that enables the coupling of multi-model simulation software with hardware simulators and devices in a shared and distributed co-simulation environment. To couple software-only simulators with the hard real-time world, the infrastructure uses a near real-time middleware based on VILLASnode for data processing and exchange between simulation models. In this way, the correct temporal evolution of the co-simulated scenario is ensured and the real-time constraints of the DRTS are respected, allowing the inclusion of the fast time-stepped simulation of a power grid model.

A Smart Grid scenario is proposed to test both the capability and performances of the infrastructure to perform multi-domain analysis of scenarios by interconnecting different software, devices, and digital real-time simulation environments, extending the capabilities of standalone or monolithic simulations. The scenario evaluates a voltage regulation ancillary service for the power grid distribution network provided directly by the BEMS of a building equipped with a PV and a battery system coordinated with a VCS. The scenario mainly consists of software-only simulators for all the entities and systems composing the customer premises, except for the battery and BEMS simulated into DRTS together with the power grid and the Physical Smart Meter device in HIL. The scenario results showed the capability of the infrastructure to deal with coupled software and hardware simulators and HIL devices with good performances, high details, and low co-simulation latencies with an average value of 287.5 ms.

In conclusion, the proposed infrastructure allows it to be used as a virtual test bed for complex Smart Grid integration, as well as for broader energy systems like multi-energy systems that aim to integrate multiple energy vectors, infrastructure, and sectors. Indeed, the flexibility and the capability of the proposed co-simulation platform allow the integration of different domain-specific models and tools towards the analysis and simulation of integrated energy systems to assess innovative monitoring, management, and control strategies through new enabling technologies or by system and flexibility operators to design and test new business models.

The research challenges proposed in this manuscript are tackled comprehensively. However, some different trade-offs and limitations are pointed out in Section V and could be tackled in the research challenge solutions. For instance, automated composability (RC2) will be addressed in future works by implementing model composability through ontologies. This will automate the generation of the YAML file avoiding manual fulfilment and reducing this error-prone process. Moreover, improvements could be made in the composability of the Hardware Simulation Layer and the HIL deployment by automating the process of model integration of complex hardware models and real-world components.

To enhance automated deployment (RC3) of the proposed infrastructure, future works will also explore virtualization technologies like Docker or Kubernetes, which not only facilitate deployment but also enable scalability management of cluster nodes in terms of computational power. This approach will effectively meet the computational demands of the simulators. These technologies not only facilitate deployment but also enable horizontal and vertical scalability management of the cluster nodes in terms of computational power, thereby effectively meeting the computational demands of the simulators. Moreover, no automated recovery process has been developed, and this issue will surely be pointed out in future works.

ACKNOWLEDGMENT

(Luca Barbierato and Daniele Salvatore Schiera are co-first authors.)

REFERENCES

[1] *Communication From the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions Powering a Climate-Neutral Economy*, Eur. Commission, 2022. Accessed: Jul. 29, 2024. [Online]. Available: https://eurlex.europa.eu/legal-content/EN/ALL/?uri=COM:2020:299:FIN

[2] M. Vogt, F. Marten, and M. Braun, "A survey and statistical analysis of smart grid co-simulations," *Appl. Energy*, vol. 222, pp. 67–78, Jul. 2018.

[3] H.-K. Ringkjøb, P. M. Haugan, and I. M. Solbrekke, "A review of modelling tools for energy and electricity systems with large shares of variable renewables," *Renew. Sustain. Energy Rev.*, vol. 96, pp. 440–459, Nov. 2018.
[4] J. Allegrini, K. Orehounig, G. Mavromatidis, F. Ruesch, V. Dorer, and R. Evins, "A review of modelling approaches and tools for the simulation of district-scale energy systems," *Renew. Sustain. Energy Rev.*, vol. 52, pp. 1391–1404, Dec. 2015.
[5] A. Sola, C. Corchero, J. Salom, and M. Sanmarti, "Multi-domain urban-scale energy modelling tools: A review," *Sustain. Cities Soc.*, vol. 54, Mar. 2020, Art. no. 101872.
[6] C. Gomes, C. Thule, D. Broman, P. G. Larsen, and H. Vangheluwe, "Co-simulation: A survey," *ACM Comput. Surveys*, vol. 51, no. 3, p. 49, 2018.
[7] H. Lund, "Energyplan-advanced energy systems analysis computer model," *Documentation version*, vol. 12, p. 12, Sep. 2015.
[8] P. Lilienthal, "Homer registered micropower optimization model," Nat. Renew. Energy Lab., Golden, CO, USA, Tech. Rep. NREL/CP-710-37606, 2005.
[9] Berkeley Lab. *Distributed Energy Resources Customer Adoption Model Plus (DER-CAM+)*. Accessed: Jul. 29, 2024. [Online]. Available: https://ipo.lbl.gov/lbnl2016-075/
[10] R. De Coninck, R. Baetens, B. Verbruggen, D. Saelens, J. Driesen, and L. Helsen, "Modelling and simulation of a grid connected photovoltaic heat pump system with thermal energy storage using modelica," in *Proc. Syst. Simul.*, 2010, p. 21.
[11] J. Huber and C. Nytsch-Geusen, "Development of modeling and simulation strategies for large-scale urban districts," in *Proc. Building Simul.*, vol. 49, 2011, pp. 1–20.
[12] F. Schloegl, S. Rohjans, S. Lehnhoff, J. Velasquez, C. Steinbrink, and P. Palensky, "Towards a classification scheme for co-simulation approaches in energy systems," in *Proc. Int. Symp. Smart Electric Distrib. Syst. Technol. (EDST)*, Sep. 2015, pp. 516–521.
[13] D. Bhor, K. Angappan, and K. M. Sivalingam, "Network and power-grid co-simulation framework for smart grid wide-area monitoring networks," *J. Netw. Comput. Appl.*, vol. 59, pp. 274–284, Jan. 2016.
[14] Modelica Association Project. *Functional Mockup Interface*. Accessed: Jul. 29, 2024. [Online]. Available: https://fmi-standard.org/
[15] V. Reinbold, C. Protopapadaki, J.-P. Tavella, and D. Saelens, "Assessing scalability of a low-voltage distribution grid co-simulation through functional mock-up interface," *J. Building Perform. Simul.*, vol. 12, no. 5, pp. 637–649, Sep. 2019.
[16] J. Bruinenberg et al., "CEN CENELEC ETSI smart grid coordination group on smart grid reference architecture," CEN CENELEC ETSI Coordinating Group Smart Grid, Brussels, Belgium, Tech. Rep. M/490 Mandate, 2012, pp. 98–107.
[17] V. H. Nguyen, T. L. Nguyen, Q. T. Tran, and Y. Besanger, "Synchronization conditions and real-time constraints in co-simulation and hardware-in-the-Loop techniques for cyber–physical energy system assessment," *Sustain. Energy, Grids Netw.*, vol. 20, Dec. 2019, Art. no. 100252.
[18] F. Ponci, A. Monti, and A. Benigni, "Simulation for the design of smart grid controls," in *Proc. IEEE 1st Int. Workshop Smart Grid Model. Simul. (SGMS)*, Oct. 2011, pp. 73–78.
[19] P. Mihal, M. Schvarcbacher, B. Rossi, and T. Pitner, "Smart grids co-simulations: Survey & research directions," *Sustain. Computing: Informat. Syst.*, vol. 35, Sep. 2022, Art. no. 100726.
[20] E. Kamburjan, V. N. Klungre, S. L. Tapia Tarifa, R. Schlatte, M. Giese, D. B. Cameron, and E. B. Johnsen, "Emerging challenges in compositionality and correctness for digital twins," in *CEUR Workshop Proc.*, vol. 3507, 2023, pp. 1–23.
[21] G. Schweiger, C. Gomes, G. Engel, I. Hafner, J. Schoeggl, A. Posch, and T. Nouidui, "An empirical survey on co-simulation: Promising standards, challenges and research needs," *Simul. Model. Pract. Theory*, vol. 95, pp. 148–163, Sep. 2019.
[22] L. Barbierato, P. Rando Mazzarino, M. Montarolo, A. Macii, E. Patti, and L. Bottaccioli, "A comparison study of co-simulation frameworks for multi-energy systems: The scalability problem," *Energy Informat.*, vol. 5, no. S4, p. 53, Dec. 2022.
[23] D. S. Schiera, "Towards energy systems integration: A holistic methodology, software platform, and toolset for modelling and simulation of cyber-physical-social energy systems," Ph.D. dissertation, Dept. of Energy, Politecnico di Torino, Turin, Italy, 2022.
[24] L. Barbierato, "Modelling and co-simulation of multi-energy systems: Distributed software methods and platforms," Ph.D. dissertation, Dept. of Control and Computer Engineering, Politecnico di Torino, Turin, Italy, 2022.
[25] D. S. Schiera, L. Barbierato, A. Lanzini, R. Borchiellini, E. Pons, E. F. Bompard, E. Patti, E. Macii, and L. Bottaccioli, "A distributed platform for multi-modelling co-simulations of smart building energy behaviour," in *Proc. IEEE Int. Conf. Environ. Electr. Eng. IEEE Ind. Commercial Power Syst. Eur.*, Jun. 2020, pp. 1–6.
[26] D. S. Schiera, L. Barbierato, A. Lanzini, R. Borchiellini, E. Pons, E. Bompard, E. Patti, E. Macii, and L. Bottaccioli, "A distributed multimodel platform to cosimulate multienergy systems in smart buildings," *IEEE Trans. Ind. Appl.*, vol. 57, no. 5, pp. 4428–4440, Sep. 2021.
[27] S. Vogel, V. S. Rajkumar, H. T. Nguyen, M. Stevic, R. Bhandia, K. Heussen, P. Palensky, and A. Monti, "Improvements to the co-simulation interface for geographically distributed real-time simulation," in *Proc. 45th Annu. Conf. IEEE Ind. Electron. Soc.*, vol. 1, Oct. 2019, pp. 6655–6662.
[28] A. Monti, M. Stevic, S. Vogel, R. W. De Doncker, E. Bompard, A. Estebsari, F. Profumo, R. Hovsapian, M. Mohanpurkar, J. D. Flicker, V. Gevorgian, S. Suryanarayanan, A. K. Srivastava, and A. Benigni, "A global real-time superlab: Enabling high penetration of power electronics in the electric grid," *IEEE Power Electron. Mag.*, vol. 5, no. 3, pp. 35–44, Sep. 2018.
[29] OFFIS—Institute for Information Technology. *Mosaik*. Accessed: Jul. 29, 2024. [Online]. Available: https://mosaik.offis.de/
[30] L. Thurner, A. Scheidler, F. Schäfer, J.-H. Menke, J. Dollichon, F. Meier, S. Meinecke, and M. Braun, "Pandapower—An open-source Python tool for convenient modeling, analysis, and optimization of electric power systems," *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 6510–6521, Nov. 2018.
[31] D. Lohmeier, D. Cronbach, S. R. Drauz, M. Braun, and T. M. Kneiske, "Pandapipes: An open-source piping grid calculation package for multi-energy grid simulations," *Sustainability*, vol. 12, no. 23, p. 9899, Nov. 2020.
[32] L. Barbierato, E. Pons, A. Mazza, E. F. Bompard, V. S. Rajkumar, P. Palensky, E. Macii, L. Bottaccioli, and E. Patti, "Stability and accuracy analysis of a distributed digital real-time cosimulation infrastructure," *IEEE Trans. Ind. Appl.*, vol. 58, no. 3, pp. 3193–3204, May 2022.
[33] T.-T. Ku, C.-H. Lin, C.-T. Hsu, C.-S. Chen, Z.-Y. Liao, S.-D. Wang, and F.-F. Chen, "Enhancement of power system operation by renewable ancillary service," *IEEE Trans. Ind. Appl.*, vol. 56, no. 6, pp. 6150–6157, Nov. 2020.
[34] A. Estebsari, M. Orlando, E. Pons, A. Acquaviva, and E. Patti, "A novel Internet-of-Things infrastructure to support self-healing distribution systems," in *Proc. Int. Conf. Smart Energy Syst. Technol. (SEST)*, Sep. 2018, pp. 1–6.
[35] CEN CENELEC. *CEN CENELEC EN 50160 Voltage Characteristics of Electricity Supplied by Public Distribution Systems*. Accessed: Jul. 29, 2024. [Online]. Available: https://standards.globalspec.com/std/14582389/en-50160
[36] M. Orlando, A. Estebsari, E. Pons, M. Pau, S. Quer, M. Poncino, L. Bottaccioli, and E. Patti, "A smart meter infrastructure for smart grid IoT applications," *IEEE Internet Things J.*, vol. 9, no. 14, pp. 12529–12541, Jul. 2022.
[37] E. Buraimoh, G. Ozkan, L. Timilsina, P. K. Chamarthi, B. Papari, and C. S. Edrington, "Overview of interface algorithms, interface signals, communication and delay in real-time co-simulation of distributed power systems," *IEEE Access*, vol. 11, pp. 103925–103955, 2023.
[38] P. Esposito, E. Marrasso, C. Martone, G. Pallotta, C. Roselli, M. Sasso, and M. Tufo, "A roadmap for the implementation of a renewable energy community," *Heliyon*, vol. 10, no. 7, Apr. 2024, Art. no. e28269.



LUCA BARBIERATO (Member, IEEE) received the M.Sc. degree in telecommunications engineering from the University of Brescia, Brescia, Italy, in 2013, and the Ph.D. degree (cum laude), in 2022. He is currently pursuing the Ph.D. degree in control and computer engineering with the Energy Center Laboratory, Politecnico di Torino, Turin, Italy. From 2014 to 2018, he was a Collaborator with TIM SWARM JOL, Turin. During this experience, he joined as a Research Assistant with the Department of Computer and Control Engineering, Politecnico di Torino. He is currently a Postdoctoral Researcher with the Energy Center Laboratory, Politecnico di Torino. His main research interests include the energy efficiency of smart cities, with a focus on the design of co-simulation infrastructure for multi-energy systems integration.



DANIELE SALVATORE SCHIERA (Student Member, IEEE) received the B.Sc. degree in energy engineering from Università degli Studi di Palermo, Italy, in 2016, and the M.S. degree in energy and nuclear engineering and the Ph.D. degree (cum laude) in energetics from Politecnico di Torino, Italy, in 2018 and 2022, respectively. He is currently a Research Fellow with the Energy Center Laboratory, Turin, Italy. His main research interests include the development of a holistic methodology, co-simulation infrastructure, and toolset for the deployment of sustainable integrated multi-energy systems.



MATTEO ORLANDO (Student Member, IEEE) received the B.Sc. degree in physics engineering, in 2015, the M.Sc. degree in ICT for smart societies, in 2018, and the Ph.D. degree, in 2022, with a thesis focused on designing and developing a self-configuring IoT smart meter for monitoring the power grids and enabling novel services. He spent one year as a Research Assistant. His main research interests include software architectures and solutions for smart grids and renewable energy, with a focus on PV technologies and smart meters.



ANDREA LANZINI (Member, IEEE) received the M.S. degree in energy and nuclear engineering, in 2007, and the Ph.D. degree in energetics, in 2011. He is currently a tenure-track Assistant Professor with the Energy Department, Politecnico di Torino, Italy, where he is a member of the Thermochemical and Electrochemical Systems Group. He is also a member of the Energy Center, Politecnico di Torino. He conducts research in the field of high-temperature fuel cells, carbon capture and reuse technologies, and integrated energy systems analysis. He teaches classes in the sustainable use of energy, poly generation and advanced energy systems, and thermal design and optimization.



ENRICO PONS (Senior Member, IEEE) received the master's degree in electrical engineering and the Ph.D. degree in industrial safety and risk analysis from Politecnico di Torino, Turin, Italy, in 2004 and 2008, respectively. He is currently an Associate Professor with Politecnico di Torino, where he teaches electrical installations. His research interests include complexity in energy systems, power systems security, complex network methodologies for the analysis of power systems vulnerability, real-time simulation of power systems, integration of renewable generation in distribution networks, smart metering, traction electrification systems, and electrical safety.



LORENZO BOTTACCIOLI (Member, IEEE) received the B.Sc. degree in mechanical engineering from the University of Perugia, in 2011, the M.Sc. degree in energetic and nuclear engineering from Politecnico di Torino, Turin, Italy, in 2014, and the Ph.D. degree (cum laude) in computer engineering from Politecnico di Torino, in 2018. He is an Assistant Professor at the Energy Center Laboratory of Politecnico di Torino. His main research interests include concern smart energy, cities, and communities, with a focus on software solutions for planning, analysing, and optimising smart energy systems and spatial representation of energy information.



EDOARDO PATTI (Member, IEEE) received the M.Sc. and Ph.D. degrees in computer engineering from Politecnico di Torino, Italy, in 2010 and 2014, respectively. He is currently an Assistant Professor with Politecnico di Torino. His main research interests include ubiquitous computing, the Internet of Things, and smart systems applications.

Open Access funding provided by 'Politecnico di Torino' within the CRUI CARE Agreement