

Towards a Simulation-as-a-Service Architecture for Energy Research

Corinna Seiwert^{1,*} , Mohamed Ibrahim¹ , Nurbek Halikulov¹ , and Jan Sören Schwarz^{2,3} 

¹Friedrich-Alexander-Universität Erlangen-Nürnberg, [ROR](#), Chair of Computer Science 7 (Computer Networks and Communication Systems), Erlangen, Germany

²Carl von Ossietzky Universität Oldenburg, Department of Computing Science, [ROR](#), Oldenburg, Germany

³OFFIS, Institute for Information Technology, [ROR](#), Oldenburg, Germany

*Correspondence: Corinna Seiwert, corinna.seiwert@fau.de

Abstract. Simulations play a crucial role in energy systems research, yet their execution is often complex and computationally intensive. To address these challenges, this work presents the infrastructure for a Simulation-as-a-Service (SimaaS) hub that enables on-demand execution of simulations while also providing complementary services to support their use. The proposed architecture is structured into three layers Service, Communication, and Infrastructure, which together offer scalable access to simulation tools, standardized interfaces, and reliable execution environments.

Keywords: Co-Simulation, Simulation-as-a-Service, Infrastructure

1. Introduction

Energy system research is often based on simulation, which enables the analysis of complex systems and the testing of new concepts under controlled conditions [1]. In particular, studies on the integration of renewable energies rely on simulations to evaluate system behavior, assess novel technologies, and provide evidence-based insights for political and regulatory decisions. However, performing complex simulations often requires substantial IT resources, which can be cost-prohibitive and underutilized outside peak research periods [2].

To address these challenges, the NFDI4Energy consortium aims to simplify access to simulation tools by developing a Simulation-as-a-Service (SimaaS) approach. SimaaS allows researchers to run complex simulations without investing in dedicated infrastructure, providing flexible scalability and easy access to powerful computational tools. By lowering the technical and financial barriers, SimaaS accelerates research processes, supports scenario analysis for renewable energy integration, and contributes to evidence-based strategies for the energy transition [1]. This work presents an overview of the infrastructure underlying the SimaaS hub, currently being developed as part of NFDI4Energy.

2. Overall Architecture

In previous work [3], present an initial mock-up of the SimaaS hub. The core services of the hub comprise the software registry and the SimaaS component itself. The software registry is designed to provide an overview of existing energy research software and models by cataloging their associated metadata. The SimaaS component enables users to execute simulations with mosaik [4], DaceDS [5], and the VILLASframework [6] directly over the internet. These three co-simulation frameworks serve different purposes within the energy domain. mosaik [4] is widely adopted in the energy research community and supports co-simulation through established domain-specific simulation tools. DaceDS [5] employs a publish–subscribe mechanism that enables a very loose coupling of simulation components and additionally incorporates simulators from the traffic domain. The VILLASframework [6] enables real-time co-simulation with hardware-in-the-loop setups. To implement the SimaaS hub, we are developing a prototype based on the infrastructure shown in Figure 1. The architecture is structured into three layers: the Service Layer, the Communication Layer, and the Infrastructure Layer.

A related work for the SimaaS approach is the OpenPlan tool, which provides a similar service for the use with oemof a open-source framework for the optimization of energy supply systems. It allows to create scenarios directly in the browser and execute them on a server infrastructure. The scenario management has different requirements for our co-simulation frameworks, but the infrastructure part of OpenPlan is comparable and will be used as starting point for our SimaaS approach.

Service	NFDI4Energy Platform Frontend			
	User Management			
	Scenario Management			
Communication	Scenario Ontology			ERSMeta
	Simulation-Server-API			Software Registry API
Infrastructure	Simulation-Server-Tasks			Software Registry
	DaceDS-Container	mosaik-Container	VILLASframework - Container	

Figure 1. Architecture overview of SimaaS Hub

2.1 Service Layer

The Service Layer includes the frontend, user management, and scenario management components. The frontend provides access to the available services, enabling users to configure and execute simulations as well as to add and browse registry entries. The user management component ensures that created artifacts and resources can be assigned to individual users, allowing, for example, subsequent edits to entries in the software registry. Scenario management stores simulation scenarios and enables users to share them with others. By ensuring that scenarios can be re-executed consistently, this layer contributes to scientific reproducibility.

2.2 Communication Layer

The Communication Layer acts as the connector between the frontend and the backend. It comprises two APIs that interface with the software registry and the co-simulation frameworks. For the connection between the scenario management and the execution in the co-simulation frameworks a ontology-based data schema will be implemented. It is based on the NFDI4Energy scenario ontology, which defines the concepts contained in a simulation scenario as well as the relationships between them [7]. Since all three co-simulation frameworks rely on the scenario ontology as input, it also functions as a common schema described by one of the APIs. Using supporting tools such as pydantic [8], the validity of a scenario file can be automatically verified against this ontology. The OpenPlan tool provides an existing API implementation for oemof [9]. Its lightweight design allows for straightforward adaptation, which we leveraged in our prototype. For the software registry, however, we will develop a dedicated API because the registry is based on the ERS-Meta [10] schema, which differs significantly from the scenario ontology [7]. ERS-Meta [10] is a metadata schema developed within NFDI4Energy by Ferenz to describe energy research software in a structured and interoperable manner. Through the API of the software registry, its contents can be accessed by external services.

2.3 Infrastructure Layer

The Infrastructure Layer contains the software registry database and the three co-simulation frameworks. To avoid interference between frameworks, we deploy each in its own container. To address the scalability and interoperability challenges inherent in heterogeneous simulation environments, we leverage a cloud-native architecture based on Docker [11] and Kubernetes [12].

The OpenPlan simulation server offers not only an API but also a task-distribution mechanism that receives API requests and forwards them to the simulation framework. We extended it to a generalized Simulation Server that routes incoming requests and their associated input files to any implemented worker via a message queue, rather than being coupled directly to a single simulation tool. In this architecture, each simulation framework contains a self-contained, long-running core service that executes simulations. The core service consumes task messages from the message queue and translates them into an ephemeral job specification that it submits to the Kubernetes API. The Kubernetes API schedules the jobs dynamically and executes them in isolation. This separation means that how we manage tasks does not depend on the specific setup we use, allowing us to easily switch from a local computer to a larger distributed system in the future without changing the code.

While Kubernetes manages container resource-level scheduling, the coordination between the OpenPlan API and the simulation workers requires an additional application-level messaging mechanism. To this end, we use a message broker (RabbitMQ) and an in-memory data store (Redis). RabbitMQ implements the AMQP protocol, enabling language-agnostic communication between the API and the simulation workers. When the API receives a request, it publishes a task message to a durable RabbitMQ queue rather than invoking a simulation process directly. The queue ensures the API remains responsive under load by buffering incoming requests and dispatching them to available workers at their own pace. Redis complements this mechanism by providing real-time tracking of task states. Each simulation task is assigned a universally unique identifier, and its status is maintained in Redis. When the worker finishes its task, it saves links to the created result files in Redis. This lets the API quickly respond to status requests without having to check the simulation containers or the storage layer directly.

3. Conclusion

The paper provides an overview of the SimaaS hub prototype, developed within the NFDI4Energy consortium, to help energy researchers use simulations for their projects.

The SimaaS hub features a well-structured architecture organized into three layers:

- Service Layer: Enables user interaction and scenario management.
- Communication Layer: Connects frontend with backend services and simulation frameworks.
- Infrastructure Layer: Houses databases and co-simulation frameworks.

Utilizing a cloud-native design with Docker and Kubernetes, SimaaS enhances scalability and interoperability while ensuring scientific reproducibility. The paper highlights SimaaS's potential to enhance energy research processes and support the integration of renewable energy by providing flexible and efficient simulation capabilities.

Data availability statement

This contribution does not rely on empirical datasets. The work presented here focuses on the architecture of the SimaaS Hub. Underlying and related materials are listed in the section below.

Underlying and related material

The underlying and related materials for this contribution are listed here:

- NFDI4Energy simulation server: <https://github.com/NFDI4Energy/nfdi4energy-simulation-server>. "Fork of the simulation server used by open-plan-tool/gui to reuse for SimaaS Hub."
- OpenPlan simulation-server repository: github.com/open-plan-tool/simulation-server. "The simulation server used by open-plan-tool/gui to run simulation of energy system models."
- Scenario Ontology repository. <https://github.com/NFDI4Energy/simulation-scenario-ontology>. "GitHub repository for collecting other relevant ontologies and other resources related to the ontology."
- OEO repository. <https://github.com/OpenEnergyPlatform/ontology>. "Repository of the OEO, in which the scenario ontology is integrated in the oeo-model module."
- ERSMeta repository: <https://github.com/NFDI4Energy/ERSmeta>. "Repository with meta-data schema to describe energy research software"

Author contributions

Conceptualization: C.S., M.I., N.H., J.S.S.; Methodology: C.S., M.I., N.H., J.S.S.; Project administration: C.S.; Writing – original draft: C.S., M.I.; Writing – review & editing: C.S., M.I., N.H., J.S.S.;

Competing interests

The authors declare that they have no competing interests.

Funding

The authors would like to thank the German Federal Government, the German State Governments, and the Joint Science Conference (GWK) for their funding and support as part of the NFDI4Energy consortium (www.nfdi4energy.org). The work was funded by the German Research Foundation (DFG, German Research Foundation) under the National Research Data Infrastructure – NFDI4Energy – 501865131.

Acknowledgements

The authors of this article have drawn on various preparatory works from NFDI4Energy, citing them where appropriate. We gratefully acknowledge all individuals who contributed to shaping and supporting the structures and results presented here. Special thanks are extended to the additional participants of Task Area 5 within NFDI4Energy (Laura Fuentes Grau, Zhiyu Pan, Nan Liu, Philipp Schmurr, and Ramiz Qussous) for their valuable discussions and contributions to the developments described in this document. We also thank the OpenPlan developer team, particularly Pierre-François Duc, for their insightful feedback and discussions regarding the reuse of the simulation server.

References

- [1] T. Preisler, T. Dethlefs, and W. Renz, "Simulation as a service: A design approach for large-scale energy network simulations", *Proceedings of the 2015 Federated Conference on Computer Science and Information Systems, FedCSIS 2015*, vol. 5, pp. 1765–1772, 2015, ISBN: 9788360810651. DOI: [10.15439/2015F116](https://doi.org/10.15439/2015F116).
- [2] M. Gütlein and A. Djanatliev, "Modeling and simulation as a service using apache kafka:" in *Proceedings of the 10th International Conference on Simulation and Modeling Methodologies, Technologies and Applications*, Lieusaint - Paris, France: SCITEPRESS - Science and Technology Publications, 2020, pp. 171–180, ISBN: 978-989-758-444-2. DOI: [10.5220/0009780501710180](https://doi.org/10.5220/0009780501710180). Accessed: Aug. 20, 2024. [Online]. Available: <https://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0009780501710180>.
- [3] C. Seiwerth et al., "Towards a simulation-as-a-service hub for the energy domain", Mar. 25, 2025, Conference Name: 2. NFDI4Energy Conference Publisher: Zenodo. DOI: [10.5281/zenodo.17431852](https://doi.org/10.5281/zenodo.17431852). Accessed: Nov. 27, 2025. [Online]. Available: <https://zenodo.org/records/17431852>.
- [4] C. Steinbrink et al., "CPES testing with mosaik: Co-simulation planning, execution and analysis", *Applied Sciences*, vol. 9, no. 5, 2019, ISSN: 2076-3417. DOI: [10.3390/app9050923](https://doi.org/10.3390/app9050923). [Online]. Available: <https://www.mdpi.com/2076-3417/9/5/923>.
- [5] C. Seiwerth, N. Halikulov, and R. German, *D 5.3.3.1 extending a data-centric distributed simulation framework for the energy domain*, Apr. 2025. DOI: [10.5281/zenodo.15268434](https://doi.org/10.5281/zenodo.15268434). [Online]. Available: <https://doi.org/10.5281/zenodo.15268434>.
- [6] S. Vogel, N. Eiling, M. Pitz, A. Bach, M. Stevic, and P. A. Monti, "Villasnode: An open-source real-time multi-protocol gateway", *Journal of Open Source Software*, vol. 10, no. 112, p. 8401, 2025. DOI: [10.21105/joss.08401](https://doi.org/10.21105/joss.08401). [Online]. Available: <https://doi.org/10.21105/joss.08401>.
- [7] *Towards an ontology for co-simulation scenarios of energy systems*, Zenodo, Mar. 2025. DOI: [10.5281/zenodo.15061653](https://doi.org/10.5281/zenodo.15061653). [Online]. Available: <https://doi.org/10.5281/zenodo.15061653>.
- [8] S. Colvin et al., *Pydantic validation*, version v2.13.0a0+dev, original-date: 2017-05-03T21:23:58Z, Oct. 2025. Accessed: Nov. 27, 2025. [Online]. Available: <https://github.com/pydantic/pydantic>.

- [9] S. Hilpert, C. Kaldemeyer, U. Krien, S. Günther, C. Wingenbach, and G. Plessmann, "The open energy modelling framework (oemof) - a new approach to facilitate open science in energy system modelling", *Energy Strategy Reviews*, vol. 22, pp. 16–25, 2018, ISSN: 2211-467X. DOI: <https://doi.org/10.1016/j.esr.2018.07.001>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2211467X18300609>.
- [10] S. Ferez, *ERSmeta*, original-date: 2025-02-20T15:02:54Z, Oct. 28, 2025. Accessed: Nov. 27, 2025. [Online]. Available: <https://github.com/NFDI4Energy/ERSmeta>.
- [11] D. Merkel, "Docker: Lightweight linux containers for consistent development and deployment", *Linux J.*, vol. 2014, no. 239, Mar. 2014, ISSN: 1075-3583.
- [12] E. A. Brewer, "Kubernetes and the path to cloud native", in *Proceedings of the Sixth ACM Symposium on Cloud Computing*, ser. SoCC '15, Kohala Coast, Hawaii: Association for Computing Machinery, 2015, p. 167, ISBN: 9781450336512. DOI: [10.1145/2806777.2809955](https://doi.org/10.1145/2806777.2809955). [Online]. Available: <https://doi.org/10.1145/2806777.2809955>.