

Demo: Interactive and Flexible Power Grid Modeling with GridForge

Lucca Ruhland, Lennart Bader, Jan Bauer, Martin Serror

Cyber Analysis and Defense, Fraunhofer FKIE, Germany

{firstname}.{lastname}@fkie.fraunhofer.de

Abstract—The ongoing digitization of power grids expands their attack surface, making experimental cybersecurity evaluations essential. Since experiments on productive grids are impractical, researchers rely on co-simulation platforms such as WATTSON, which require detailed scenarios. While automated scenario derivation tools such as POWEROWL address this challenge, they offer limited flexibility for direct modeling and customization. We present *GridForge*, an interactive tool for visually modeling coupled power grid and communication network scenarios. *GridForge* combines a graphical user interface with a shared graph abstraction and a transactional edit model with snapshot-based validation to aid users in creating flexible scenarios from scratch as well as on top of derived models, while preventing inconsistent states. It directly integrates with POWEROWL and creates WATTSON-compatible scenarios, while its modular design supports adaptation of other co-simulation backends. We demonstrate that *GridForge* lowers the barrier to practical scenario engineering for cybersecurity research on power grids.

Index Terms—scenario engineering, cyber-physical systems, co-simulation, power grid modeling, communication network modeling, cybersecurity of power grids

I. INTRODUCTION & BACKGROUND

The ongoing digitization of power grids is reshaping their operation. Legacy serial communication is increasingly being replaced or complemented by IP-based and other standard networking technologies, while the integration of distributed and renewable energy resources increases the need for distributed real-time monitoring and control [1]–[4]. At the same time, this transition of the underlying Industrial Control System (ICS) network enlarges the attack surface. While power grid operation has become more dependent on communication infrastructures, many industrial communication protocols were not originally designed with security in mind, and long device lifecycles complicate patching and replacement [1], [2], [5]. The coordinated December 2025 campaign against the Polish energy sector, for instance, destroyed remote terminal units and human-machine interfaces across more than 30 facilities, leading to a loss of visibility and controllability [6]. Such incidents highlight the need to study attack impacts and countermeasures in controlled environments. Conducting these studies on productive grids is impractical due to safety concerns and strict availability requirements [1], [5]. Security research therefore requires safe cyber-physical environments that capture both the electrical infrastructure and the underlying communication network [4], [5], [7].

A common way to realize such environments is *co-simulation*, i.e., the combination of simulators or emulators for multiple domains such as power systems and communication networks [7], [8]. Existing approaches primarily differ in how they represent these domains. On the communication side, discrete-event approaches scale well but abstract from real protocol stacks and timing behavior. Representative examples include EPOCHS [9], GECO [10], FNCS [11], mosaik [12], and HELICS [7]. Emulation-based approaches instead execute real network stacks and unmodified attack tools at the cost of scalability, as demonstrated by DSSnet [4], MiniCPS [13], and WATTSON [14]. On the power side, steady-state solvers such as pandapower [15] suffice for many cyberattack studies, whereas transient simulators offer finer detail but require proprietary hardware. Security-oriented platforms have been proposed, for example, to study data attacks against energy management systems [16], to generate attack data [5], or to support cybersecurity training [17].

Among these, WATTSON [14] provides an open-source co-simulation platform that combines power flow simulation with network emulation and supports real-world power-grid protocols such as IEC 60870-5-104 (IEC 104). WATTSON has been validated against a physical testbed and scales to realistic power grid sizes [14]. Because real applications and attack tools can be executed within the emulated network, the platform is well suited for realistic cybersecurity experiments [14]. To execute experiments in WATTSON, the setup must be specified as a *scenario*, i.e., a cyber-physical model comprising the power-grid topology, the corresponding network, and the OT communication configuration [18]. Since manually creating suitable scenarios is challenging, POWEROWL was introduced as a framework for deterministic and configurable scenario derivation for WATTSON [18]. Starting from a given power-grid topology, it derives additional information required for execution, such as IT network topology, and exports the result in a WATTSON-compatible format.

Although POWEROWL automates scenario derivation, its workflow presumes an existing power grid model as a starting point, and it further follows a fixed derivation logic. Users who intend to customize components, to alter the communication topology, or to construct scenarios from scratch must therefore overwrite the derivation logic in code or directly edit the configuration artifacts. To date, no interactive tool exists that supports flexible engineering of power grid scenarios while ensuring the consistency of the resulting cyber-physical model.

To close this gap, we present *GridForge*, an interactive scenario-engineering tool for modeling power grids. In contrast to derivation-driven workflows, *GridForge* allows users to explicitly configure every part of a scenario, revise derived structures where needed, and construct new scenarios from scratch. While the current prototype builds on POWEROWL and directly integrates with WATTSON, *GridForge* itself is not tied to a specific backend and can be adapted to other co-simulation platforms.

II. DESIGN & IMPLEMENTATION

The design of *GridForge* is guided by three goals that follow directly from the limitations discussed above. First, flexibility (G1) requires that users can construct scenarios from scratch or modify derived ones and freely edit every part of a scenario without having to touch code or raw configuration files. Second, since G1 permits open-ended edits, consistency (G2) demands that the tool keeps the scenario and all its views valid and mutually consistent at all times. Third, backend independence (G3) requires that the editing workflow is not tied to a specific co-simulation platform, so that *GridForge* can serve as a general-purpose scenario editor rather than a WATTSON-only frontend.

GridForge realizes these goals through three main mechanisms. A shared graph representation of both domains supports uniform editing (G1). A transactional edit model with snapshot-based validation prevents invalid scenario states (G2). Finally, a client-server architecture decouples interactive editing from backend-specific model handling (G3).

In the power domain, graph nodes capture entities such as *facilities* (sites that group electrical equipment, e.g., substations or distribution substations (DSSs)), buses, transformers, loads, or generators, while edges represent electrical interconnections. In the communication domain, nodes represent devices such as hosts, switches, routers, or firewalls, while edges capture communication links. This shared abstraction allows topological editing across both domains in a uniform way while preserving domain-specific attributes and constraints.

To ensure that every edit preserves a valid scenario state (G2), all user changes are processed transactionally. Changes are expressed as high-level scenario operations, such as adding components, changing connectivity, or modifying configuration attributes. They are first validated against structural and technical constraints before being applied to a snapshot of the current scenario state. Only if all checks succeed and the operation does not result in an erroneous state, the updated snapshot is committed. Otherwise, the change is rolled back by discarding the snapshot, and the user is notified accordingly. This prevents partially applied edits and keeps the scenario and its views synchronized at all times.

The current prototype realizes this design through a middleware and a web frontend. The middleware is implemented with *FastAPI*¹ and exposes a REST API for high-level editing operations, which are translated into backend-specific

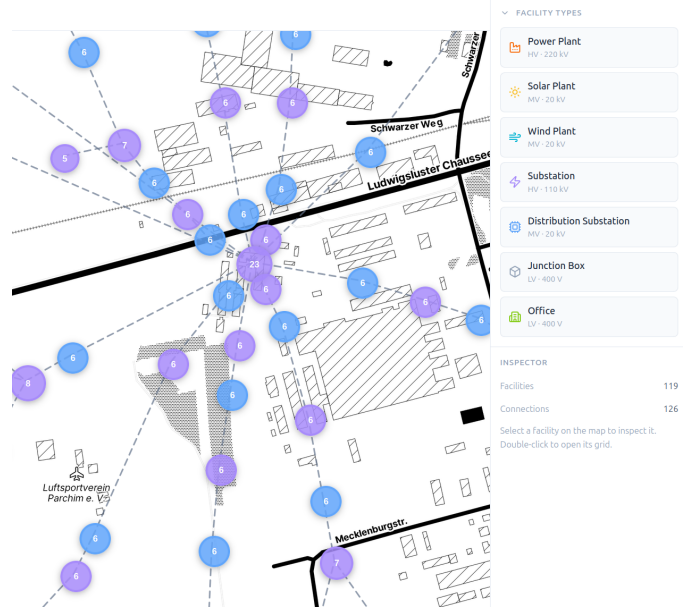


Fig. 1. Geo view of *GridForge*, showing facilities placed according to their geographic coordinates and the interconnections between them. In this example, substations and DSSs are distinguished by color, and the number shown within each facility indicates how many assets it contains.

manipulations of the scenario model (G3). In the current implementation, these operations are mapped to POWEROWL. Although the middleware is designed to remain backend-agnostic (G3), this concrete mapping enables a seamless export to WATTSON. As a result, edited scenarios are immediately available as ready-to-use evaluation setups without requiring any additional steps, demonstrating *GridForge*'s potential.

The frontend is implemented in *React*² using *React Flow*³ as the visualization framework. This fits the graph-oriented structure of both domains and supports a uniform interaction model for scenario editing. The current implementation provides four synchronized views: a *Geo View* for placing facilities and defining inter-facility connections (Fig. 1), a *Facility Detail View* for inspecting internal components, a *Power Grid View* for the global single-line diagram, i.e., the simplified schematic that represents the three-phase power system by single lines, which also allows editing of component properties (Fig. 2), and a *Network View* for communication topology and device configuration (Fig. 3).

III. DEMO

The demonstration will be conducted on a laptop running both *GridForge* and WATTSON locally. *GridForge* will be mirrored to a large screen visible to the attendees. They can interact with the system directly using a keyboard and mouse.

The demo highlights two aspects of *GridForge*. First, attendees can explore how to create and modify power grid scenarios through the graphical interface. Starting from a small

¹<https://fastapi.tiangolo.com/>

²<https://react.dev/>

³<https://reactflow.dev/>

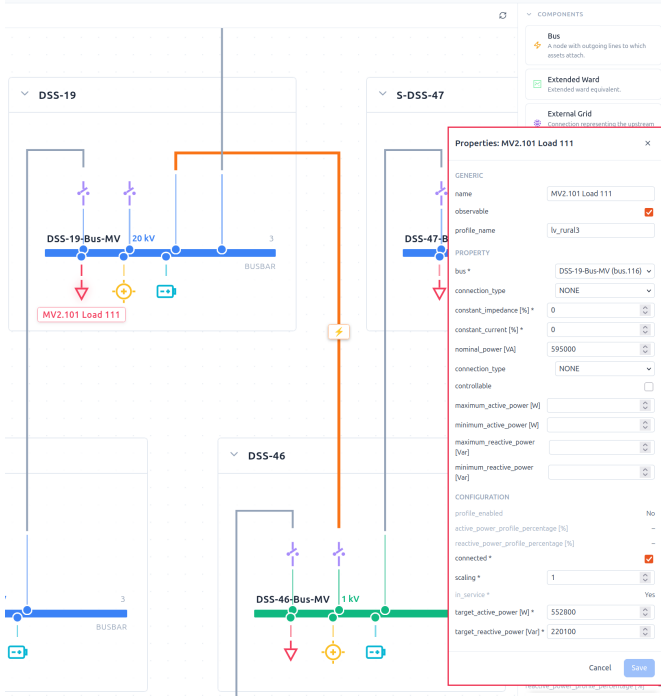


Fig. 2. Power grid view of *GridForge*, illustrating a portion of the complete single-line diagram of the modeled power system, including multiple inter-connected DSSs and the editing of component properties.

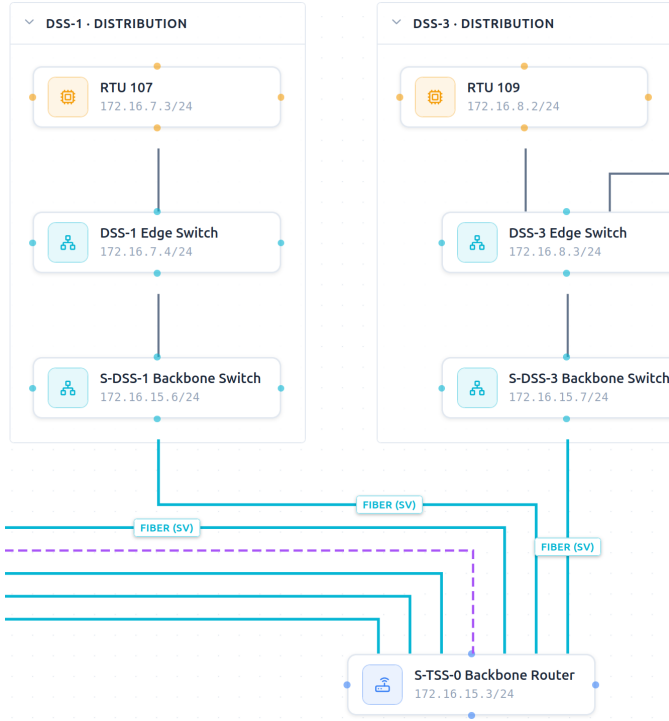


Fig. 3. Network view of *GridForge*, showing a portion of the communication network of interconnected DSSs.

demo scenario, they can inspect facilities, edit their internal components, and explore the corresponding communication topology in the network view. This is intended to demonstrate that *GridForge* makes scenario engineering accessible to attendees without deep prior knowledge of power grids.

Second, the modified scenario can be exported and simulated with WATTSON. Attendees can observe the resulting simulation artifacts such as the generated communication network and traffic between power grid components in domain-specific protocols like IEC 104. This part of the demo illustrates that *GridForge* is not a mockup editor but rather a useful tool for constructing scenarios that can be directly used in WATTSON.

To accommodate different interests, the demo supports guided exploration along a more power-grid- or network-oriented workflow. In the former, the attendees extend the power grid topology by adding facilities as well as grid elements and inspect how these changes are reflected in the single-line diagram and communication network. In the latter, the attendees focus more on the communication perspective by modifying the network topology and configuration of network devices, and observing the resulting changes. In both cases, we transfer the created *scenario* to WATTSON and demonstrate the impact of a cyberattack on it. Thus, the demo highlights that *GridForge* lowers the barrier to creating practical WATTSON scenarios that can be used in cybersecurity experimentation.

IV. CONCLUSION & OUTLOOK

GridForge demonstrates how power grid scenarios can be created, inspected, and modified through a flexible visual workflow. By explicit modeling of power grid and network domains, revising derived structures, and constructing scenarios from scratch, it complements derivation-driven approaches to scenario generation. The current prototype shows that this concept can be concretely integrated with POWEROWL and WATTSON, while the underlying design remains modular and adaptable to other co-simulation backends. We expect that this will make power grid scenario-engineering more practical for security experimentation and encourage broader engagement with energy-sector communication protocols.

For future development, we plan to improve validation so that the system assesses not only technical correctness, such as physical consistency and POWEROWL compatibility, but also operational realism by identifying configurations that may be inefficient, unusual, or potentially unsafe in practice. To support adoption and reuse within the research community, we open-sourced *GridForge* at <https://github.com/fkie-cad/gridforge>. Finally, we plan to conduct a user study with participants from both the power systems and cybersecurity communities to systematically evaluate *GridForge*'s usability and effectiveness for practical scenario engineering, and to guide the aforementioned improvements to the validation logic of our implementation.

REFERENCES

- [1] B. Green, A. Lee, R. Antrobus, U. Roedig, D. Hutchison, and A. Rashid, "Pains, Gains and PLCs: Ten Lessons from Building an Industrial Control Systems Testbed for Security Research," in *Proc. of the USENIX Workshop on Cyber Security Experimentation and Test (CSET)*, Vancouver, BC, 2017.
- [2] T. Krause, R. Ernst, B. Klaer, I. Hacker, and M. Henze, "Cybersecurity in Power Grids: Challenges and Opportunities," *Sensors*, vol. 21, no. 18, 2021.
- [3] N. E. M. Barreto and A. R. Aoki, "Cyber-Physical Power System Digital Twins—A Study on the State of the Art," *Energies*, vol. 18, no. 22, 2025.
- [4] C. Hannon, J. Yan, and D. Jin, "DSSnet: A Smart Grid Modeling Platform Combining Electrical Power Distribution System Simulation and Software Defined Networking Emulation," in *Proc. of the ACM SIGSIM Conference on Principles of Advanced Discrete Simulation (SIGSIM-PADS)*, Banff, Alberta, Canada, 2016.
- [5] O. Bel, J. Kim, W. J. Hofer, M. Maharjan, B. Hyder, S. Purohit, and S. Niddodi, "Co-Simulation Framework for Network Attack Generation and Monitoring," *IEEE Access*, vol. 12, 2024.
- [6] CERT Polska, "Energy Sector Incident Report – 29 December 2025," Naukowa i Akademicka Sieć Komputerowa (NASK), Tech. Rep., 1 2026, <https://cert.pl/en/posts/2026/01/incident-report-energy-sector-2025/>.
- [7] T. D. Hardy, B. Palmintier, P. L. Top, D. Krishnamurthy, and J. C. Fuller, "HELICS: A Co-Simulation Framework for Scalable Multi-Domain Modeling and Analysis," *IEEE Access*, vol. 12, 2024.
- [8] P. Mihal, M. Schvarcbacher, B. Rossi, and T. Pitner, "Smart grids co-simulations: Survey & research directions," *Sustainable Computing: Informatics and Systems*, vol. 35, 2022.
- [9] K. Hopkinson, X. Wang, R. Giovanini, J. Thorp, K. Birman, and D. Coury, "EPOCHS: A Platform for Agent-Based Electric Power and Communication Simulation Built From Commercial Off-the-Shelf Components," *IEEE Transactions on Power Systems*, vol. 21, no. 2, 2006.
- [10] H. Lin, S. S. Veda, S. S. Shukla, L. Mili, and J. Thorp, "GECO: Global Event-Driven Co-Simulation Framework for Interconnected Power System and Communication Network," *IEEE Transactions on Smart Grid*, vol. 3, no. 3, 2012.
- [11] S. Ciraci, J. Daily, J. Fuller, A. Fisher, L. Marinovici, and K. Agarwal, "FNCS: A Framework for Power System and Communication Networks Co-Simulation," in *Proc. of the Symposium on Theory of Modeling & Simulation – DEVS Integrative*, Tampa, FL, USA, 2014.
- [12] S. Schütte, S. Scherfke, and M. Tröschel, "Mosaik: A Framework for Modular Simulation of Active Components in Smart Grids," in *Proc. of the Workshop on Smart Grid Modeling and Simulation (SGMS)*, Brussels, Belgium, 2011.
- [13] D. Antonioli and N. O. Tippenhauer, "MiniCPS: A Toolkit for Security Research on CPS Networks," in *Proc. of the ACM Workshop on Cyber-Physical Systems-Security and/or Privacy (CPS-SPC)*, Denver, CO, USA, 2015.
- [14] L. Bader, M. Serror, O. Lamberts, Ö. Sen, D. van der Velde, I. Hacker, J. Filter, E. Padilla, and M. Henze, "Comprehensively Analyzing the Impact of Cyberattacks on Power Grids," in *Proc. of the European Symposium on Security and Privacy (EuroS&P)*, Delft, Netherlands, 2023.
- [15] 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 Transactions on Power Systems*, vol. 33, no. 6, 2018.
- [16] K. Pan, A. Teixeira, C. D. López, and P. Palensky, "Co-simulation for Cyber Security Analysis: Data Attacks against Energy Management System," in *Proc. of the IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Dresden, Germany, 2017.
- [17] T. D. Le, A. Anwar, S. W. Loke, R. Beuran, and Y. Tan, "GridAttackSim: A Cyber Attack Simulation Framework for Smart Grids," *Electronics*, vol. 9, no. 8, 2020.
- [18] L. Bader, E. Wagner, and M. Serror, "PowerRange: An immersive cyber range for power grid operators," *International Journal of Critical Infrastructure Protection*, vol. 53, p. 100845, 2026.