A Cyber Secure Communication Architecture for Multi-Site Hardware-in-the-Loop Co-Simulation of DER Control

Sri Nikhil Gupta Gourisetti[1], Jacob Hansen[1], William Hofer[1], David Manz[1], Karanjit Kalsi[1], Jason Fuller[1], Shwetha Niddodi[1], Holger Kley[2], Christopher Clarke[3],Keunmo Kang[4], Hayden Reeve[4], Massimiliano Chiodo[4], Jesse Bishopric[4]
[1]Pacific Northwest National Laboratory, [2]Spirae, [3]Southern California Edison, [4]UTRC
{srinikhil.gourisetti, jacob.hansen, karanjit.kalsi}@pnnl.gov

***Abstract*— Existing approaches coordinating distributed energy resources (DERs) for grid services do not adequately evaluate the performance of such DER integration. Most studies are based on a single type of DER used for a single type of service, rather than the real-world requirements of coordinating a heterogeneous mix of DERs to provide multiple different grid services at different time-scales. Facilities also often face cybersecurity and interoperability challenges to experimenting and testing methodologies in this area. To overcome all of these challenges, Pacific Northwest National Laboratory, United Technologies Research Center, Southern California Edison, and Spirae coordinated to develop a federation between their organizations. This federation implements a cybersecure connection that facilitates near real-time communication between the four different physical sites. This not only enables control of DERs at different physical locations but also lets the software and hardware objects perform control experiments in a cybersecure environment at different time-scales. The hardware systems can consist of microgrids, building management systems, and emulated power systems objects. This paper provides a detailed overview of the federation setup and describes what this federation can be used for.**

Keywords— ***hardware-in-the-loop, distributed co-simulation, federation, cybersecurity, distributed energy resources, DERs***

I. INTRODUCTION

The rapid growth of low-inertia renewable energy resources represents an immense opportunity for the U.S. to minimize its carbon footprint but presents a challenge for system operators as traditional "spinning" generation resources are displaced [1-2]. There is growing recognition that distributed energy resources (DERs—loads, distributed generation, storage, electric vehicles, etc.) represent great potential for performing this function, but operators have concerns about the controllability and dependability of DERs, especially when they are not under the direct control of operators. Although aggregation and control of DERs for various grid services have been extensively studied in the literature, and various modeling and control techniques have been proposed [3-8], most of the current studies are limited to modeling, aggregation, and control of a single type of DER and for a particular type of service. They ignore the control and coordination of a heterogeneous mix of flexible resources to provide multiple ancillary services occurring at different time-scales to the power grid. Moreover, some of the control strategies rely on centralized direct load control methods [1-4], which are not very scalable, to manage hundreds of millions of smart-grid assets due to the stringent sensing, communication, and computation requirements of such approaches.

To address these challenges, the authors have been developing a holistic system called Network Optimized Distributed Energy Systems (NODES). It will be a generalized, incentive-based control and coordination framework for a heterogeneous class of DERs, such as residential and commercial buildings, electric vehicles, energy storage, and distributed generation. The final outcome will be a distributed hierarchical control framework that allows DERs to be integrated seamlessly into operation of the traditional grid infrastructure, and coordinated to produce the smooth, stable, and predictable response required by grid operators. The performance of the resulting system will be tested in a co-simulation environment spanning transmission, distribution, ancillary markets, and communication systems. Various classes of actual DERs, (e.g., residential and commercial heating, ventilation, and air-conditioning [HVAC] systems, smart appliances, electric vehicles, etc.) and their control systems will be used to perform hardware-in-the-loop (HIL) verification of the proposed incentive-based control approach. The first step is to establish a network architecture that enables such experimentation. This network architecture needs to be capable of handling a fully functional HIL co-simulation between the control system (simulation framework) environment at Pacific Northwest National Laboratory (PNNL) and the remote DER locations at United Technologies Research Center (UTRC), Spirae, and Southern California Edison (SCE). These facilities need to be connected in a cybersecure fashion. Therefore, for the NODES experiment, the research team decided to take a federated approach in which the remote locations can be connected over point-to-site, site-to-site, or across multi-vendor VPN architectures.

Testbed federation is of international interest. Recently, the National University of Singapore identified several issues with

This study was conducted at the Pacific Northwest National Laboratory, which is operated for the U. S. Department of Energy by Battelle Memorial Institute under Contract DE-AC05-75RL01830.

The information, data, or work presented herein was funded in part by the Advanced Research Projects Agency-Energy (ARPA-E), U.S. Department of Energy, under Award Number DE-AR0000700. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

federation testbeds including intra-domain challenges [9]. Previous work demonstrated federations using testbeds across the nation [10 – 12]. PNNL, the University of Illinois at Urbana – Champaign (UIUC), and University of Southern California – Information Sciences Institute (USC ISI) conducted a wide-area federated experiment using resources at all three institutions. The work culminated in a live demonstration conducting a situation awareness attack on two substations and a control center. Each facility was geographically distributed and was federated with each other. This work ultimately resulted in a shared definition of testbeds and federation among the participating organizations [13]. Additionally, Iowa State University has conducted research into remotely accessible testbeds for power systems applications [14]. The (federation) solution used in this paper leverages the previous experience to develop a lightweight and interoperable approach based on OpenVPN technology [15].

This paper explains the simulation framework, including all the federates that are part of the framework, the hardware messaging platform, and the methods of federation between the federates and hardware. As stated above, necessary architecture has been established to perform a cybersecure HIL co-simulation [13, 14] with federated partners. Prior to the NODES experiment, the federated approach was tested at prototype level using hardware (Raspberry-Pi). But the system had never been implemented on a large-scale experiment that involves both hardware and software components across four organizations. Because of the cybersecurity and reliability advantages of the federated approach, the NODES research team used it for this experiment, which led to the first multi-organization (more than two participants) field deployment of the system.

This paper is organized as follows: Section II discusses the co-simulation environment and HIL components, along with methods for connecting HIL and co-simulation. Section III presents a federated cybersecure connection between test sites located in distant geographical areas of different networks, and finally, Section IV provides concluding remarks.

II. Hardware-in-the-Loop Co-Simulation Environment

The experimental setup for NODES consists of two general components: a simulation and a HIL framework. The simulation framework is the family of simulators and required data to run the simulations, etc. As shown in Fig. 1, HIL is realized by connecting collaborators' hardware systems to the federates through VOLTTRON agents. VOLTTRON [19] is an open-source platform that enables distributed sensing, data exchange, and controls. It is a messaging bus, controller platform, and hardware interface rolled into one package.

The workflow steps of the NODES experiment are:

- Each of the collaborators sends virtual battery model (VBM) parameters representing the current state of their respective connected hardware systems to the VOLTTRON instance on their network.
- The VOLTTRON instance sends the VBM parameters to PNNL'S VOLTTRON instance.
- PNNL's VOLTTRON instance sends that information to the simulators. Following the same communication route, the simulators generate and dispatch a power profile, such as setpoints, back to the collaborator's hardware systems.

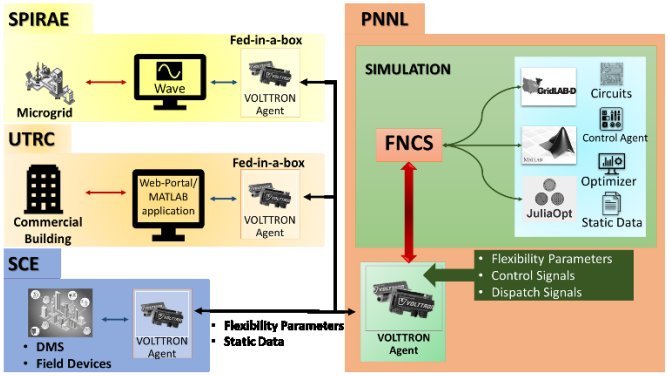

Fig. 1. Architectural Overview of the NODES Experiment

A. Simulation Environment

The simulation environment is the overarching framework that brings together all the federates that are part of the experiment. This co-simulation framework is built around FNCS—an open-source co-simulation software [16]—to integrate multiple simulators across multiple domains, ensuring interoperability across many different commercial and open-source tools. The primary role of FNCS is to transfer information/data across different simulators in a time-synchronized manner. This allows researchers to explore the interactions of normally stove-piped planning and control tools, while developing new control and optimization solutions in tools with which they are familiar. The full simulation framework is depicted in Fig.2.

Each block in the framework shown in Fig. 2 describes a federate, some of which are simulators. The different simulators interact with each other through FNCS. Note that multiple instances of each box could be used for large-scale complex simulations. The framework consists of five groups of simulators described in this section:

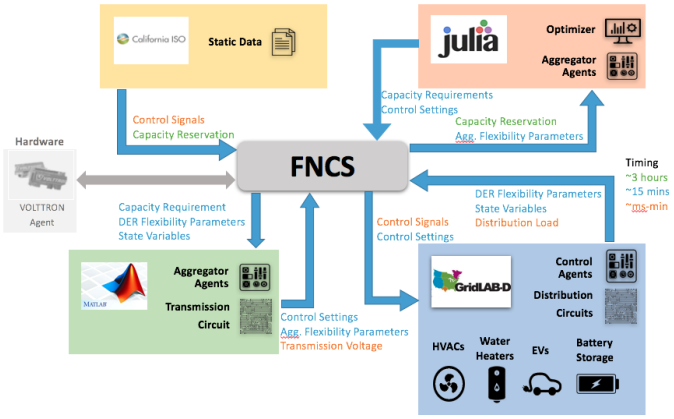

Fig. 2. The NODES Simulation Environment

1. *GridLAB-D:* GridLAB-D simulator [17] is used to build the distribution circuits. It also houses dynamic model of all types of controllable devices. This layer receives control signals broadcasted from the independent system operator (ISO), along with control settings from the Aggregator Agents. Each device in this layer provides its

DER flexibility parameters and state information back to an Aggregator Agent.

2. *California ISO:* California ISO represents a static data federate. It contains static recorded data that can be injected into the FNCS stream. In this experiment, the California ISO data federate provides four signals. The first signal is a capacity reservation signal that is communicated to the DRC optimizer inside Julia. This signal describes the capacity the ISO is procuring from the distributed resource controller (DRC) in each category. Furthermore, the ISO broadcasts a 4-second regulation signal directly to individual asserts inside GridLAB-D that participate in the proposed control framework.
3. *Julia:* Julia is an open-source, high-performance, and dynamic programming language. It is used for building the multi-period power allocation optimization algorithm that is proposed at the DRC level. Julia was chosen because it provides an excellent framework for solving computationally heavy optimizations algorithms. The DRC receives the capacity reservation signal from the ISO, along with aggregated flexibility parameters, also known as the VBM, from each individual Aggregator Agent. The DRC uses the collected VBMs to solve the resource allocation algorithm, ensuring that the DRC aligns enough resources to fulfill the capacity requirement set forth by the ISO. Furthermore, this allocation calculates control settings for each Aggregator Agent. The control setting signal differs depending on what resource(s) the Aggregator Agent is aggregating. In the case of purely thermostatically controlled loads these settings will be temperature setpoints.
4. *MATLAB:* MATLAB is used in the simulation framework for two purposes. The first is to incorporate a transmission system solver using MATPOWER. This allows the project to couple distribution systems with a single transmission system to allow for a more detailed co-simulation that can evaluate control performance in the simulation of a truly integrated system. Coordination of devices is handled by the Aggregator Agents. This control entity is built using MATLAB. Aggregator Agents receive control settings and capacity requirements from the DRC, along with individual DER flexibility parameters and state information from each device in GridLAB-D. This allows the Aggregator Agent to construct the aggregated flexibility parameters and forward them to the DRC. It also forwards control settings to each device in GridLAB-D.

Simulations frameworks similar to the one described above have shown great potential for their flexibility and scalability. Previous efforts performed under the Control of Complex Systems Initiative at PNNL have shown that this framework can easily support thousands of distribution systems along with hundreds of thousands of controllable devices [18].

B. Hardware-in-the-Loop

The second part of the experiment setup is the HIL where the hardware systems are located at various (geographical) locations. These hardware systems are connected to send flexibility parameters (VBM) to simulators. The VBM may include any power systems and control systems parameters such as base power, energy limits, power limits, and associated scalars, etc. The simulators discussed in the previous section generate the power profiles and send them back to the hardware systems. By design, the simulation environment is independent from the HIL environment—in other words, the simulations can be run without the HIL—but the addition of the HIL adds richness and any desired complexity to the experiments. To enable such coordination between the software systems, simulators, remote hardware systems, and independent VOLTTRON instances are installed at remote hardware locations. VOLTTRON agents are deployed on those instances to exchange the data.

VOLTTRON: VOLTTRON is an innovative distributed control and sensing software platform that supports modern control strategies, including agent-based and transaction-based controls. It enables mobile and stationary software agents to perform information gathering, processing, and control actions. VOLTTRON can be used to independently manage and control a wide range of systems, such as HVAC systems, electric vehicles, distributed energy, or entire building loads, leading to improved operational efficiency and energy and cost savings.

The independent VOLTTRON agents receive state information, building control setting, asset flexibility, etc. from the hardware systems as VBMs. The VBM is then sent to the VOLTTRON instance at PNNL. Through a FNCS broker, the VBMs are sent from VOLTTRON to the simulators and control agents (see Fig. 3). A calculated power profile is dispatched to the remote hardware system through the same connection. This full-duplex connection, as shown in Fig. 3, handles: 1) exchanges the VBMs between remote VOLTTRON agents and the PNNL VOLTTRON agent in an asynchronous queue-/token-based fashion and 2) dispatches back the power profiles. As mentioned above, the optimization algorithm in Julia or MATLAB generates the power profile with control setpoints, signals and sends to the remote hardware systems through VOLTTRON.

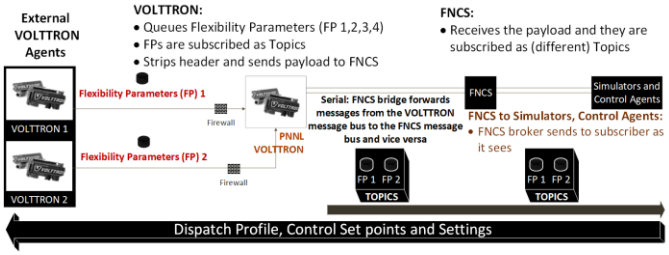

Fig. 3. Description of VOLTTRON Message Bus Configuration

The next section highlights the software tunnel between VOLTTRON and FNCS that lets a VOLTTRON agent forward the data to FNCS.

C. FNCS VOLTTRON Bridge

Once the VOLTTRON instance receives data from remote hardware systems, the data need to be forwarded to simulators, as shown in Fig. 3, and this can be done through FNCS. The FncsVolttronBridge is designed to send messages from the VOLTTRON message bus to the FNCS message bus and vice versa. As shown in Fig. 3, at the VOLTTRON stage, the VBMs are subscribed as topics. The header is stripped, and the payload

is sent to FNCS. FNCS receives the payload, which is subscribed as different topics. Then, the FNCS broker sends the data to subscribers as it sees the data/message(s).

An initial test to exchange data between VOLTTRON and FNCS was performed on an Ubuntu Linux virtual machine to test the efficacy and latency of the FncsVolttronBridge. A pre-created data file of FNCS-recognizable messages was transmitted from a VOLTTRON instance using a forwarder agent from the FNCS-installed machine to a VOLTTRON instance that is located on a different machine/Virtual Machine. The test file used has 60 data entries at 1-second intervals. Once the VOLTTRON and FNCS instances were initiated to start the data exchange, each data point in the test file was expected to move from VOLTTRON to FNCS (and vice versa) at the designated time interval (1 sec). As shown in Table. I, the data-exchange stream test was conducted at 1-sec intervals. With a latency typically less than ~100 msec, the data were received at the expected ~1-second intervals. The base data transmission rate requirements for the NODES experiment is much slower (between a few seconds and up to ~5-minute intervals). Because this stream/exchange test was conducted to transmit the data at 1-second intervals, this validated that the architecture will work seamlessly for this experiment.

TABLE I. DATA EXCHANGES BETWEEN VOLTTRON AND FNCS

Item	Output
Data-sending agent	VOLTTRON
Receiving agent	FNCS
Sampling time (Data transmission interval)	1 sec
Data packets sent	60 packets
Total transmission time	1 min
Latency/delay	≤ ~100 msec
Operating system	Ubuntu Linux
Host	Virtual machine and a physical machine

D. VOLTTRON to FNCS Timing Mechanisms

Based on the tests performed, it was evident that the FncsVolttronBridge can handle messages at any speed at which VOLTTRON is configured to send messages to FNCS and vice versa. Fig. 4 demonstrates the routing process with an example. As shown, the data packet/message (denoted as M1) sent from VOLTTRON to FNCS at a simulation timestep is received by the federates (example: GridLAB-D, Julia) at the next simulation timestep. A numerical way to describe this example is if M1 is sent by the remote VOLTTRON at 05:00:00 and M2 is sent at 10/10/2017 05:00:01, the federates receive M1 at 10/10/2017 05:00:01 and M2 at 10/10/2017 05:00:02.

Finally, to enable VOLTTRON-to-VOLTTRON data exchange, a well-tested forwarder agent is activated on both the instances. The final data flow architecture enables data exchange between the remote VOLTTRON instance, local VOLTTRON instance, FNCS, and the simulators/control agents.

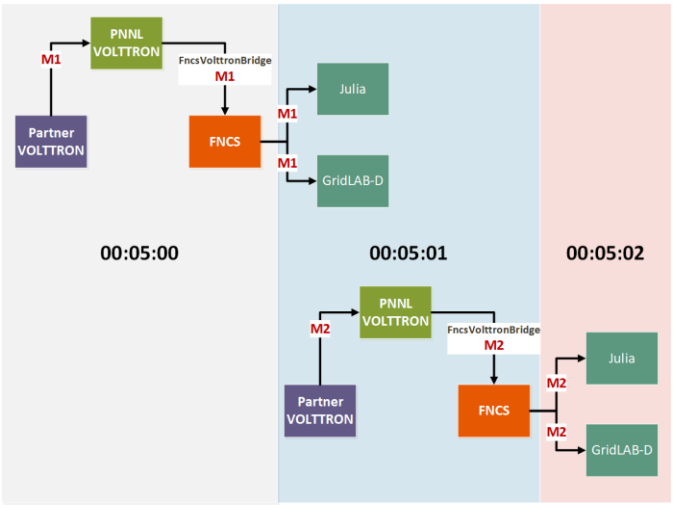

Fig. 4. VOLTTRON to FNCS – Illustrative Data Exchange over 2 Timesteps

III. SECURE FEDERATED CONNECTION

Federated connection enables cybersecure data-exchange capabilities between systems of different networks. In the case of the NODES experiment, this connection was being established in three different ways: 1) Layer-2 (Data Link Layer) site-to-site; 2) Layer-3 (Network Layer) point-to-site; 3) multi-vendor VPN connections. This section introduces the core concepts of "Fed-in-a-box"—a virtual machine (VM) with the ability to establish any of the above connections. Then, the architectural elements and the connection methods with UTRC, Spirae, and SCE are discussed.

A. Fed-in-a-box

Fed-in-a-box is a VM that has pre-scripted Open-SSH scripts that enable it to connect to a client or a server. It is called fed-in-a-box because the federated connection is established through this VM (box). As an example, if two sites are on different networks, Fed-in-a-box bridges the machines on these two sites and makes it look like they are on same network. Fed-in-a-box has two interfaces: in this experiment, one interface faces PNNL and the other faces the world-wide web. The VPN bridge allows a Layer-3 (the network layer) communication between a computer on the PNNL network and a computer on the collaborator's network. Federated architecture comes with several advantages. 1) At an organizational level, the PNNL IT-Cybersecurity office strictly prohibits data routing from an external entity. Through federation, the VPN bridge connects selected/defined external devices to an organization's computer over a private connection. This way, neither of the sites is exposed to each other (further explained in Sections III-C and III-D). Therefore, during the data exchange, both sites (external entities and PNNL) are secure from a cybersecurity perspective. 2) Laboratory-level prototype tests have been conducted using the Fed-in-a-box approach with some hardware systems (Raspberry-Pi) to evaluate its efficacy. 3) Through Fed-in-a-box, the data are already encrypted during the transfer/exchange that eliminates the possibilities of data spoofing and stealing. 4) Based on the laboratory-level tests using VMs on different networks, this architecture has proven to be efficient and fast in exchanging data at high sampling

frequencies (less than sub-second sampling time). This provides great flexibility to the NODES project to perform both fast and slow experiments.

B. Challenges of Federation

Connecting systems that are in different proprietary networks to perform experiments such as HIL co-simulation is not a trivial task. The remote networked systems landscape is completely different from connecting systems in the same network. Some of the common policy barriers associated with connecting systems between networks in two different organizations include the following: 1) A given organization's network may not allow external traffic through firewalls. 2) If firewall exceptions are added, the cybersecurity risk level increases by allowing such traffic from another organization. 3) There is risk of exposing the entire network of an organization to another organization. 4) There is risk of accidental changes to the network manager (often through administrative access) that could potentially damage the entire network.

To go into more detail, each organization's network is behind a firewall that blocks any external traffic into their systems. For illustrative purposes, the data-receiving organization is referred to as the "host entity" and the data-sending organization is referred to as "external entity". External entity traffic can reach the host entity only when the administrator of the host entity permits the external entity to cross the firewall to let the traffic in. One of the ways to do that is to open a public-facing port for use by the external entity. But doing so could risk exposure of the entire host network to an external entity. From a cybersecurity perspective, an attacker can attack the external entity and potentially gain access to the host entity's complete network. Therefore, connecting different systems from different organizations poses hard challenges both from the cybersecurity perspective and relative to strict organizational policies.

As stated in the previous section, by using the Fed-in-a-box concept, an organization would only expose a system to a system in another organization instead of exposing the entire network. PNNL is achieving this by opening a secure OpenVPN tunnel and connecting Spirae's and UTRC's systems to the NODES system at PNNL. When these systems are connected through the OpenVPN tunnel, the connection/authentication goes through a server in the CyberNET testbed, a private cloud maintained by a research team at PNNL, and complete visibility is maintained on all the systems that are coming through this connection. The external sites systems are only able to communicate with systems in an isolated virtual network created in the cloud platform. Therefore, a secure connection is established, and organizational policies are not violated.

C. Layer-2 Connection between PNNL and UTRC

The current solution employed to federate with UTRC involves a Layer-2 site-to-site VPN tunnel using an OpenVPN access server and transport layer security (TLS) encryption. The PNNL team has set up an OpenVPN access server in the CyberNET testbed environment that is reachable by the internet through Port Network Address Translation (PNAT) on a designated public IP and port. We use a port that has been set up with PNAT to be reachable by the internet.

On PNNL's end, the OpenVPN access server is deployed within an OpenStack cloud environment. It is dual-homed. The first interface is attached to a private software-defined subnet that is reachable externally via network address translation (NAT). A second interface is attached to a private software-defined subnet that houses the NODES VM. Using a Linux bridge the VPN TAP interface created by OpenVPN and the second interface on the server are bridged on Layer 2 of the Open Systems Interconnection (OSI) model.

On UTRC's end, a dedicated hardware system that is also dual-homed is deployed. Similarly, one interface is attached to a network that can reach out to the internet minimally to the port listening on the VPN server, and a second interface is connected to a private LAN segment managed by a switch. By downloading a client configuration via the web frontend of PNNL's OpenVPN access server using a pre-shared key, UTRC can use OpenVPN to connect to PNNL. The OpenVPN client system then connects to the OpenVPN access server and establishes a TLS tunnel. The client configuration file that is downloaded specifies two scripts that get triggered when the OpenVPN service starts and stops. These scripts configure the OpenVPN client system to use the same bridging strategy as the OpenVPN access server in PNNL's OpenStack cloud. The TAP interface of the OpenVPN connection and the private side interface are housed on a Linux bridge.

Upon launching the OpenVPN service, a script to turn on the bridge is executed. Upon stopping an OpenVPN service, a script to turn off the bridge is executed.

Once this site-to-site bridge is set up, systems in the virtual private LAN in PNNL's OpenStack testbed and systems in the private LAN segment at the client site can communicate over Ethernet/Layer 2 in the OSI model. Fig. 5 shows a simple network diagram of what the connection looks like and below that is the step-by-step instructions for setting up the OpenVPN client box using Ubuntu 16.04 as the operating system.

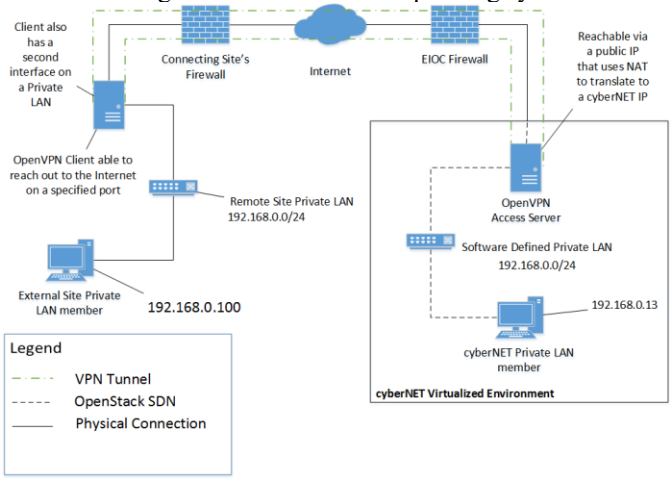

Fig. 5. Illustraive Network Diagram of Layer-2 Federation Connection

Software and Hardware Components at UTRC: On the UTRC side the architecture consists of the following:

- a VOLTTRON agent running on a RedHat Linux machine

- a Python wrapper that provides an easy-to-use application programming interface (API) to the ZeroMQ (ZMQ) [20] layer needed to exchange communication with the VOLTTRON agent
- a set of MATLAB functions that encapsulate and hide details of the above Python layer. This allows sending and receiving information with the VOLTTRON agent at a level of abstraction suitable for driving the MATLAB application.
- a MATLAB application that performs required elaboration of data exchanged with VOLTTRON agent.

The communication between the Python/MATLAB code and the UTRC VOLTTRON agent is handled via a ZeroMQ publish/subscribe mechanism. The information exchange is completely asynchronous; therefore, a simple application-level protocol is defined to associate each message with its corresponding response. Because the exchanges are typically very infrequent (with period measured in minutes) the possible overhead required to guarantee a non-lossy communication is negligible.

The VOLTTRON agents at PNNL and UTRC also communicate via a ZMQ-based publish/subscribe mechanism. Because the two VOLTTRON instances are running on separate machines, each instance provides a forwarder agent that uses ZMQ to send messages to the other instance—PNNL forwards to UTRC, and UTRC forwards to PNNL. Within each VOLTTRON instance, agents subscribe to information that allows them to participate in the hierarchical control framework. For the PNNL VOLTTRON agent this means subscribing to the VBM from the UTRC VOLTTRON agent, and for the UTRC VOLTTRON agent this means subscribing to the control setpoints from the PNNL VOLTTRON agent.

Fig. 6 shows one application using the communication architecture described above. Using the collected data from UTRC campus operation and simulated data from PNNL via the communication described above, a MATLAB application optimizes building operation.

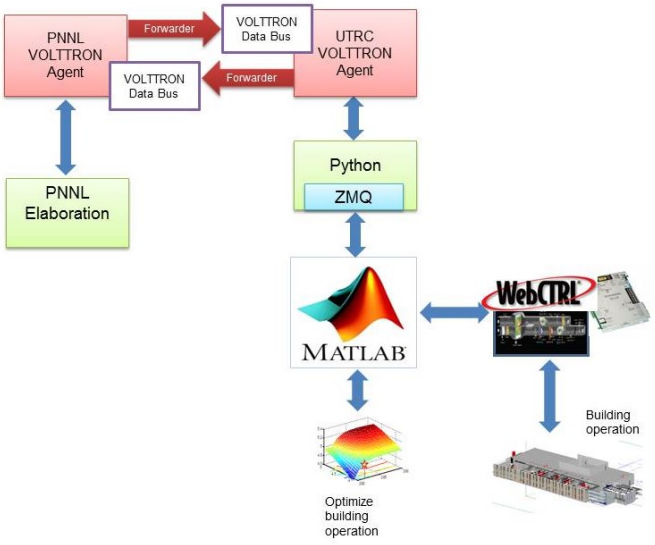

Fig. 6. Fed-in-a-box Application Connected to Building Control Application

D. Layer-3 Connection Between PNNL and Spirae

The current solution employed to federate with Spirae involves a Layer-3 point-to-site VPN tunnel using an OpenVPN access server and TLS encryption. PNNL has set up an OpenVPN access server in the CyberNET testbed environment that is reachable by the internet through PNAT on a public IP.

At PNNL's end, the OpenVPN access server is deployed within an OpenStack cloud environment. Similar to the Layer-2 connection, it is dual-homed, and the first interface is attached to a private software-defined subnet that is reachable externally via NAT. A second interface is attached to a private software-defined subnet that houses the NODES VM. By configuring static routes in the OpenVPN access server settings, clients can reach PNNL's private network in the testbed on Layer 3 of the OSI model.

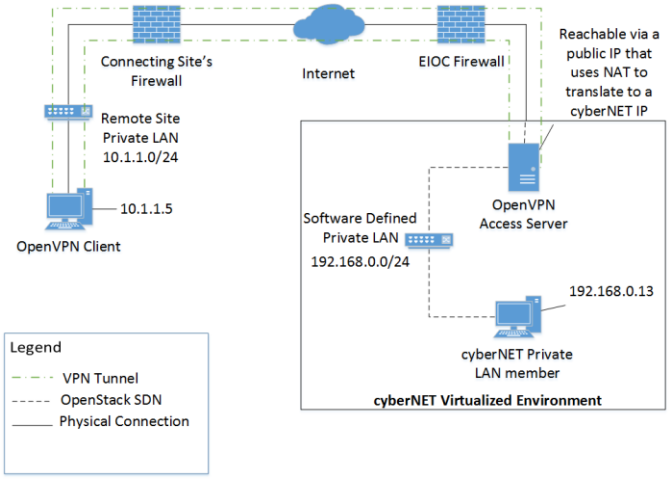

Fig. 7. Illustrative Network Diagram of the Layer-3 Federation Connection

On Spirae's end, a dedicated VM is deployed. One interface is attached to a network that can reach out to the internet minimally to the port the PNNL VPN server is listening on. By downloading a client configuration via the web frontend of the OpenVPN access server using a pre-shared key, Spirae can use OpenVPN to connect to PNNL. The OpenVPN client system then connects to the OpenVPN access server and establishes a TLS tunnel. The client configuration file that is downloaded specifies that the client can route to a private LAN behind the OpenVPN access server.

Once this point-to-site tunnel is set up (Fig. 7), systems in the virtual private LAN in PNNL's OpenStack testbed and the connected client system on the Spirae network can communicate over TCP/IP, or Layer 3 in the OSI model. Fig. 7 is a simple network diagram of what the connection looks like and below that are the step-by-step instructions for setting up the OpenVPN client box using Ubuntu 16.04 as the OS.

Software and Hardware Components at Spirae: Under the NODES project, Spirae is providing remote access to around 60 physical power system resources located at two sites powered by the same distribution substation. Resources include curtailable solar inverters, battery energy storage systems, small generation, and a variety of interruptible single- and three-phase loads, including a curtailable electric vehicle charging station. Access is by means of APIs exposed by

Spirae's Wave® microgrid control software, which delivers asset- and group-based monitoring and control functionality. Thus, the asset-specific interfaces are abstracted to a common secure format. This way, the larger simulation can interact with those assets, while limiting exposure of Modbus interfaces to the Wave components. For initial testing purposes, Spirae has emulated a version of the physical assets—communicating to the microgrid software via the same Modbus points—thus limiting the need to expose control of physical devices to high-value testing times. This software and hardware architecture is summarized in Fig. 8. The final experiment may involve interaction and control using real hardware systems instead of emulated software systems.

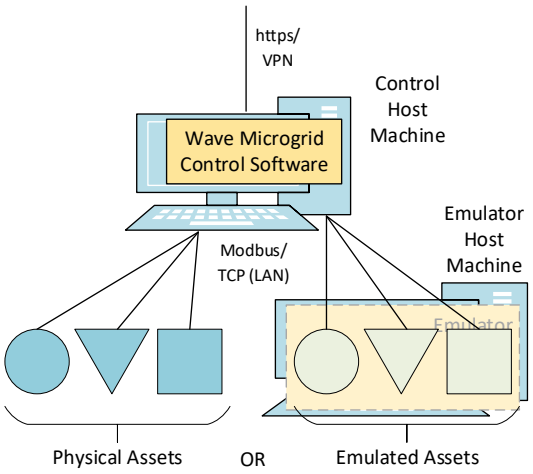

Fig. 8. Diagram of Components Hosted by Spirae

By participating in this project, Spirae will demonstrate secure, remote control of assets and groups using its microgrid software. Furthermore, because one of the sites is Spirae's InteGrid lab, NODES will serve to richly illustrate the possibilities of future distributed lab activity for that facility (including the possibility of physical/virtual hybrids).

E. Layer-2 Connection Using Existing Firewall VPN

The connection between PNNL and SCE is a work in progress. Currently, the plan is to implement a Layer-2 connection using OpenVPN as a client and connect to Southern SCE's existing VPN server to gain access to their experimental networks.

Software and Hardware Components at SCE: The core of SCE's controls testbed is a real-time power system simulator combined with a supervisory control and data acquisition (SCADA) protocol gateway. The simulator performs a three-phase unbalanced dynamic RMS simulation synchronized to the system time, and the SCADA gateway provides DNP3, Modbus, ICCP, and 61850 MMS clients and servers that can be configured to communicate with external systems. The real-time power system simulator and SCADA protocol gateway interface via OPC, and exchange measurement and control points once per second. When the power system simulator and SCADA protocol gateway are combined the resulting testbed simulates the real-time performance of a distribution feeder, including autonomous operation of DER assets, load changes over time, and closed-loop response to external control system commands.

SCE has interfaced the controls testbed with the VOLTTRON platform via a Modbus Server at every DER and telemetry monitoring point read and written to by the VOLTTRON master driver agent. In this configuration, there are over 100 Modbus sessions for simulated DERs on a single feeder. VOLTTRON agents have been developed that interact with the scheduler and actuator agents to send commands to the DERs in the simulated feeder and subscribe to feeder telemetry points. This architecture is summarized in Fig. 9.

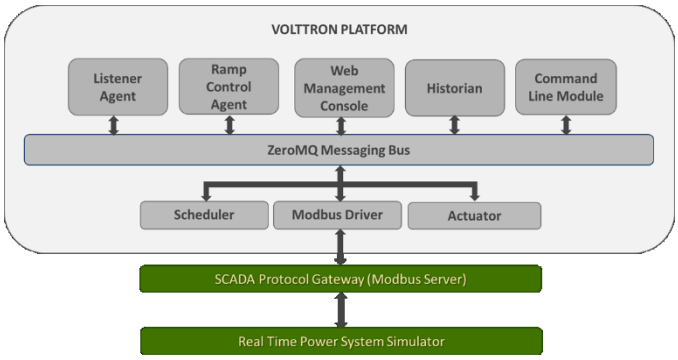

Fig. 9. Connected Architectural View at SCE

F. Validation of Federatation Network Connections

The NODES experiment system is connected to UTRC's and Spirae's systems through the defined federated approach. Each of the systems is equipped with VOLTTRON. A full-duplex data-exchange test was conducted between these systems by deploying VOLTTRON agents that transmit illustrative temperature data. A VOLTTRON listener agent was activated on the receiving systems test if the data were received. These tests concluded successfully as the data transmitted from PNNL system were being received by UTRC's and Spirae's systems and vice versa. Ongoing work is focused on developing the VOLTTRON agents that can communicate with collaborators' proprietary software/hardware and transmit the data to PNNL. Upon using the data and running the controls system experiment, PNNL will transmit the data back to the collaborators.

IV. CONCLUSION

This paper detailed the software and hardware components of the NODES experiment. The authors introduced a sophisticated methodology for connecting various software systems and power systems solvers. Then, an experimental framework was shown using the solvers and software systems that were connected to remote hardware systems. As demonstrated, such network architecture is not only cyber-secure but also widely scalable. In this paper, it was shown that a federated connection between PNNL and UTRC, Spirae, and SCE was established. The paper provided deep technical details about the Layer-2 point-to-site connection between PNNL and Spirae, Layer-3 site-to-site connection between PNNL and UTRC, and the potential multi-vendor-VPN connection between PNNL and SCE. From the demonstrations, it was clear that each of the different types of federated connections has

unique advantages. In general, encrypted communication provides more confidentiality and integrity than unencrypted communication. The choice to use OpenVPN and TLS was driven solely by cost and ease of use. To fully evaluate the security of this type of communication channel, an end to end transparency would be required. In any federated connection using VPN software and an encryption method, the security of the communications is directly related to the security of the software itself. Using cybersecure federated connection and intermediate VOLTTRON instances (and agents), the hardware and software systems were able to exchange VBMs, power profiles, and other flexibility parameters needed to perform complex control system simulations. Using and building upon this fully tested and implemented architecture, the ongoing work is focused on remote DER control through Optimizer in Julia, power systems simulation in GridLAB-D, and data exchange through VOLTTRON agents across these geographically distant facilities. Follow-on papers will present the test results acquired from a co-simulated federated HIL system with fully controllable DERs.

REFERENCES

[1] J. Smith, M. Milligan, E. DeMeo, and B. Parsons, "Utility wind integration and operating impact state of the art," IEEE Trans. Power Syst., 2007, vol. 22, no. 3, pp. 900–908

[2] Y. Makarov, C. Loutan, J. Ma, and P. de Mello, "Operational impacts of wind generation on california power systems," IEEE Trans. Power Syst., 2009, vol. 24, no. 2, pp. 1039–1050

[3] H. Hao, B. M. Sanandaji, K. Poolla, and T. L. Vincent, "Aggregate flexibility of thermostatically controlled loads," IEEE Transactions on Power Systems, 2015, vol. 30, no. 1, pp. 189–198

[4] W. Zhang, J. Lian, C. Y. Chang, and K. Kalsi, "Aggregated modeling and control of air conditioning loads for demand response," IEEE Transactions on Power Systems, 2013, vol. 28, no. 4, pp. 4655–4664

[5] H. Hao, Y. Lin, A. S. Kowli, P. Barooah, S. Meyn, "Ancillary service to the grid through control of fans in commercial building HVAC systems," IEEE Transactions on Smart Grid, 2014, vol. 5, no. 4, pp. 2066-2074,

[6] S. Meyn, P. Barooah, A. Busic, Y. Chen, and J. Ehren, "Ancillary service to the grid using intelligent deferrable loads," IEEE Transactions on Automatic Control, 2015, vol. PP, no. 99,

[7] Z. Ma, D. S. Callaway, and I. A. Hiskens, "Decentralized charging control of large populations of plug-in electric vehicles," IEEE Transactions on Control Systems Technology, 2013, vol. 21, no. 1, pp. 67–78,

[8] S. Li, W. Zhang, J. Lian, and K. Kalsi, "Market-Based Coordination of Thermostatically Controlled Loads - Part I: A Mechanism Design Formulation," IEEE Transactions on Power Systems, 2015,vol.PP, no.99.

[9] W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, "A Conceptual Framework to Federate Testbeds for Cybersecurity", Winter Simulation Conference, 2017

[10] T. Edgar, D. Manz, and T. Carroll, "Towards an experimental testbed facility for cyber-physical security research", Proceedings of the Seventh Annual Workshop on Cyber Security and Information Intelligence Research, ACM, New York, USA, Article 53, 2011

[11] T. Benzel et al., "Experience with DETER: a testbed for security research", 2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, Barcelona, 2006, pp. 10 pp.-388

[12] D. C. Bergman, D. Jin, D. M. Nicol, and T. Yardley, "The virtual power system testbed and inter-testbed integration", Proc. 2nd Workshop Cyber Security Experiment. Test, 2009, pp. 1–6

[13] A. Hussain, T. Faber, R. Braden, T. Benzel, T. Yardley, J. Jones, D. M. Nicol, W. H. Sanders, T. W. Edgar, T. E. Carroll, D. O. Manz, and L. Tinnel, "Enabling Collaborative Research for Security and Resiliency of Energy Cyber Physical Systems", IEEE International Conference on Distributed Computing in Sensor Systems, Washington, DC, USA, 2014

[14] A. Ashok, S. Krishnaswamy and M. Govindarasu, "PowerCyber: A remotely accessible testbed for Cyber Physical security of the Smart Grid," 2016 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), Minneapolis, MN, 2016, pp. 1-5

[15] OpenVPN Website. Accessed at: http://openvpn.net/

[16] S. Ciraci, J. Daily, J. Fuller, A. Fisher, L. Marinovici, K. Agarwal, "FNCS: A Framework for Power System and Communication Networks Co-simulation", Proceedings of the Symposium on Theory of Modeling and Simulation – DEVS Integrative Article No. 36, California, 2014

[17] D. P. Chassin, K. Schneider and C. Gerkensmeyer, "GridLAB-D: An open-source power systems modeling and simulation environment," 2008 IEEE/PES Transmission and Distribution Conference and Exposition, Chicago, IL, 2008, pp. 1-5.

[18] J. Hansen, T. Edgar, J. Daily and D. Wu, "Evaluating Transactive Controls of Integrated Transmission and Distribution Systems using the Framework for Network Co-Simulation," in 2017 American Control Conference, Seattle, 2017.

[19] B. Akyol, J. Haack, B. Carpenter, S. Ciraci, M. Vlachopoulou, and C. Tews, "Volttron: An agent execution platform for the electric power system", Third international workshop on agent technologies for energy systems valencia, spain. 2012.

[20] ZeroMQ Website. Accessed at: http://zeromq.org/