

# VAC: A Software Approach to Resilient SCADA Automation

Jordan Johnson<sup>\*</sup>, Brian Weber<sup>†</sup>, Jared Smith<sup>‡</sup>, Chris Hathhorn<sup>§</sup>,  
Philip Irminger<sup>¶</sup>, Savannah Norem<sup>||</sup>, Jennifer Guerra<sup>\*\*</sup>, Joel Dawson<sup>††</sup>, Tyler McDaniel<sup>‡‡</sup>

*Oak Ridge National Laboratory*

*Oak Ridge, TN*

{\*johnsonja1, †weberb, ‡smithjm, §hathhorncr, ¶irmingerp, ||noremisa, \*\*guerrajl, ††dawsonja, ‡‡mcdanielbt}@ornl.gov

**Abstract**—To better secure critical infrastructure, especially power systems, this paper introduces a virtual SCADA automation controller. The automation controller is a gateway into a power subsystem, making it a valuable target for cyber-attacks that could cut it off from the control center and cause a loss of view and control. To prevent this, the Virtual Automation Controller (VAC) is a backup device that mirrors the capabilities of the physical controller. It can communicate via Modbus and DNP3 and is containerized so it can be deployed on a variety of platforms. Furthermore, it utilizes software-defined networking to quickly disconnect a failed automation controller and preserve its state for forensics. The VAC gives system operators time to replace the failed controller and prevents dangerous and costly damage to power systems. The VAC is compared against the SEL 3505-3 RTAC and shown to have the necessary features to act as a failover controller.

**Index Terms**—security, digital twin, software defined networking, cyber-physical systems, SCADA systems, substation protection

## I. INTRODUCTION

Modern power systems are facing unprecedented challenges as engineers seek to improve security while maintaining reliability. In recent years, several cyber-attacks such as Stuxnet [1] and CrashOverride [2], [3] have shown that improved security solutions are necessary for cyber-physical systems (CPS). This is especially true for the energy sector as it enables all other critical infrastructure sectors [4]. Additionally, smart grid devices are making it easier to manage power systems, but these network-enabled devices present new attack vectors for malicious actors and adversarial nation states [5], [6], [7].

One of the common components of a power system is the supervisory control and data acquisition (SCADA) automation controller, a device that aggregates data from a subsystem and communicates it to an upstream control center. While many of these controllers feature built-in security measures, no system is perfect, so it is important to ensure multiple layers of security are in place in the event that one fails. Currently,

there are very few solutions to recover from a successful cyber attack on an automation controller. Because the automation controller is the only entry point into a subsystem, it is a single point of failure, and it is crucial that it be replaced quickly. The subsystem monitored by an automation controller is often quite expensive, and the loss of view and control could result in damaged or destroyed components.

As such, this paper proposes a virtual automation controller (VAC), which can act as a failover device in the event of an automation controller fault. In case of such an event, and to speed up the transition to the VAC, it can be enabled via SDN to take control and prevent damage to the system. To ensure the VAC can successfully perform the function of an automation controller, an implementation is evaluated and shown to possess the key attributes needed to stabilize the system for the duration of the fault.

## II. BACKGROUND

### A. Automation Controllers

Power substations are made up of many types of intelligent electronic devices (IEDs) including circuit breakers, energy meters, and transformers. Management of these devices is handled by a SCADA system, which collects readings, events, and alarms from IEDs and sends control commands to these IEDs. Command decisions are based on system processes running in a control center that constantly monitors the state of the system as a whole. Human-machine interfaces (HMIs) present the state of the system to operators who can modify setpoints and other control values, which affect the control center processes and trigger commands sent back to the individual IEDs.

Commonly found in SCADA systems are automation controllers, which act as a gateway between the control center and a subsystem such as a power substation. The automation controller collects data from IEDs, which typically communicate using various protocols such as Modbus or DNP3. The automation controller then sends accumulated data to the control center. Likewise, commands from the control center are passed back to the automation controller, which then directs them to the corresponding IEDs.

An automation controller is a secure entry point into a SCADA subsystem. If disrupted, whether due to mechanical

This manuscript has been authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the US DOE. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (<https://www.energy.gov/downloads/doe-public-access-plan>).

failure or cyber-attack, it causes a loss of view and control of the connected subsystem. As such, the automation controller is an attractive attack target. In the event of a disruption, a failover device can be used in place of the original automation controller to maintain system reliability until the controller issue is resolved.

### *B. Digital Twins*

Although originally used in manufacturing, digital twins have become increasingly powerful tools for monitoring CPS for anomalous and adverse behavior [8], [9]. In the context of CPS, a digital twin is a real-time system that takes in the same inputs as its physical counterpart and utilizes a specialized model to determine what the outputs of the system should be. The digital twin outputs are compared to the physical component's outputs to determine if it is behaving as intended. The digital twin's model can take many different forms; physics-based models and machine learning models such as neural networks are the most common [10].

In the case of an automation controller, it is possible to define the expected behavior of the system by reproducing its configuration. This consists primarily of the devices it is communicating with, the data points it is collecting from each device, and the logic programming typically defined using the IEC 61131-3 standard. By gathering these parameters, it is possible to create a digital twin without the need to rely on physics-based modeling or machine learning. This simplifies the implementation and therefore increases both flexibility and sustainability of the virtual system.

### *C. Software-Defined Networking*

Software-defined networking (SDN) centralizes management of the network control plane, allowing packet routing decisions to be made based on a view of the entire network. This centralized manager is known as an SDN controller and utilizes a protocol such as OpenFlow to communicate packet routing rules to network components. Since network connections can be defined programmatically and remotely, SDN enables fast rerouting of connections without the need to move physical network cables.

In an industrial control system (ICS) network, where inter-device communications are well defined, an SDN controller can be programmed with specific rules that describe what devices can communicate with one another and over what protocols. This severely limits unwanted network traffic that could lead to various attacks such as denial of service or man-in-the-middle. Modifying the connections is easily done with a web interface or API when changes to the system are needed. This feature allows the VAC to be enabled remotely and within seconds of a fault being detected.

### *D. Related Work*

A body of work currently exists that shows the smart grid and other cyber-physical systems would greatly benefit from the presence of secure digital twins. There are digital twins in various levels of development for cyber-physical systems,

ranging from Pileggi et al. [11], who make the case that only with enough expert knowledge will a digital twin be a candidate for system control, to Eckhart and Ekelhart [12], who present a proof of concept for a framework that automatically generates a virtual environment for digital twins based on specifications. There is more literature that goes in detail about why they are, and will become even more, necessary to protect the smart-grid in the years to come [13]. The VAC fits in a space where there is a known need with few solutions being actively developed.

The introduction of SDN enables new solutions for resiliency in SCADA systems. Others have demonstrated that SDN can be used in a variety of ways to better harden a system [14] and SCADA is no exception [15], [16]. This work adds to the mix by showcasing the agility provided by SDN. Connections to the automation controller can be swapped to the VAC near-instantaneously while the automation controller is sandboxed from the rest of the network to prevent further tampering. This capability would traditionally require physically shuffling network cables, which requires an operator to be on-site, leading to delays that could be destructive during an attack.

## III. THE VIRTUAL AUTOMATION CONTROLLER

### *A. Overview*

Traditionally, redundant automation controllers can be put in place to take over if the primary controller fails. While this may help in the event of a mechanical issue, this is unlikely to deter a cyber attack, as the redundant controller is most likely vulnerable to the same attack. As such, an alternative controller is necessary for proper defense in depth. The VAC is an implementation of a digital twin for an automation controller that can act as a redundant controller. If the primary controller stops functioning, the VAC can be enabled to temporarily command the system. To enable the quick failover capability and better secure the network, SDN is integrated with strict controls.

To successfully act as an automation controller, the VAC needs some specific capabilities. Primarily, the VAC must be able to communicate with both IEDs and the command center. In modern power systems, this is typically done via Ethernet using protocols such as Modbus and DNP3. It will need to aggregate data from multiple protocols adopted by different IEDs and convert it to a single protocol to be sent to the command center. Next, the VAC must be able to control various functions of the IEDs based on a logic program. The most common industrial programming languages are those defined by IEC 61131-3 such as structured text and ladder logic. Additionally, the VAC would benefit from an integrated human machine interface (HMI), so an operator can view the state of the controller. Finally, the VAC should be a secure entry point into the subsystem to prevent further attacks, and it should keep the subsystem stable until an engineer is able to replace the failed controller.

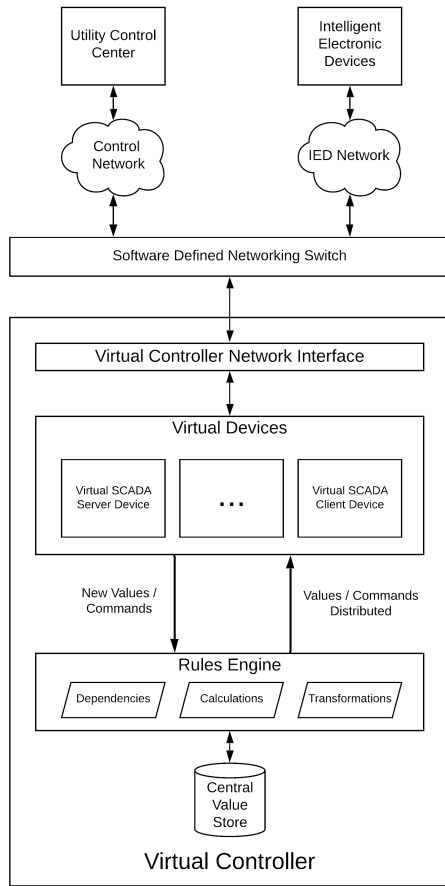


Fig. 1. High level architecture and communication networks of the VAC.

### B. Implementation

To evaluate the capabilities of a VAC, a test version was implemented using Python and Docker. The general architecture of the system can be seen in Figure 1. At the top are the control center and the IED networks. The control network can include various devices such as HMIs, historians and controllers. The IED network is typically made up of an array of devices and connections such as programmable logic controllers, remote terminal units, as well as sensors and actuators. Both networks are connected to an SDN switch along with the VAC shown in the bottom half of the figure. Each device in the IED network is monitored by a virtual component of the VAC typically called a client, while a virtual interface known as a server reports IED data to devices on the control network. These two networks are logically separate so they cannot communicate directly. Rather, all traffic must pass through the automation controller.

One of the primary roles of an automation controller is data aggregation from devices with heterogeneous communication protocols. The most common protocols are Modbus and DNP3 [17], and both are implemented in the VAC. Modbus is a client-server protocol that was originally developed by Modicon and later made open source by the Modbus Organization [18]. Distributed Network Protocol 3 (DNP3) is a more complex open standard introduced by Harris and now

maintained by the DNP3 Users Group [19]. The VAC can stand up virtual clients and servers for both protocols to communicate with devices on the control and IED networks.

The rules engine ensures data is correctly communicated between the components of the VAC as well as the IEDs and the control center. When a client receives an updated value from an IED, the new value is sent to the rules engine and stored. Based on its configuration, the rules engine distributes the update to the control center and to any other IEDs that need to know the value. It may also perform calculations or data transformations using data from the IEDs. In the same way, commands from the control center are sent to a server, which passes it to the rules engine. The command is then forwarded to the appropriate devices. All of these details are specified in the VAC's configuration file, which is parsed when the VAC starts up.

Finally, there must be a way for a failover to occur. The software-defined networking (SDN) switch is the facilitator for this feature and is controlled with a simple script. When an operator determines that a fault has occurred, they can trigger the script that will connect the VAC to the network and then disconnect the physical controller. All of this happens remotely via software, and all connections between devices are logical, so an operator can perform this action within seconds without the need to modify physical connections.

## IV. EVALUATION

To evaluate the feasibility of the VAC as a failover device, three capabilities were considered. First, the failover time should be minimal to prevent damage to the system; ideally, it should complete in a few seconds. Furthermore, once the failover has completed, the VAC must be able to take on the role of the physical automation controller. It must be able to aggregate data from all connected devices and respond to data requests in a timely manner. Finally, the VAC may need to perform automation tasks such as sending a command to a device based on a set of conditions. In this section, failover time, general performance, and automation delay are evaluated.

To run the experiments, the test environment utilizes an SEL-2740S SDN switch for networking and an SEL-3505-3 Real-Time Automation Controller (RTAC) for the physical controller. The VAC is installed on a Revolution Pi Core 3+ and connected to the SDN switch along with the RTAC. Additionally, the tests include a set of emulated power systems IEDs. These virtual devices are connected to the network and host a Modbus server. Each has two coils representing binary values and four holding registers representing two floating point values.

### A. Failover Experiment

This first experiment showcases the SDN-enabled failover capability of the VAC. The failover delay was measured to determine how quickly the switch occurred. This was done using a script that communicated with the SEL-2740S SDN switch using its built-in API, connecting the VAC to the

|                | Delay (s) |
|----------------|-----------|
| <b>Average</b> | 1.623     |
| <b>Max</b>     | 2.502     |

Fig. 2. Average and max delay across 20 iterations for a failover to complete.

| IED Count   | SEL RTAC (ms) | VAC (ms) |
|-------------|---------------|----------|
| 2           | 0.1           | 0.1      |
| 10          | 0.1           | 0.1      |
| 30          | 0.1           | 0.1      |
| 60          | 0.1           | 0.1      |
| <b>Mean</b> | 0.1           | 0.1      |

Fig. 3. Performance results when communicating with a varying number of devices.

network and disconnecting the RTAC. The delay was measured from the time the script started to the time the virtual IED received the first request from the VAC.

To harden the network, only specific connections were allowed between the RTAC and the IEDs. The connections were defined using OpenFlow, and each connection included the IP addresses of the RTAC and IED and the IED's Modbus TCP port. When the failover script runs, it first copies all connections from the RTAC, substituting the VAC's IP address. It then sandboxes the RTAC, removing all former connections to prevent further attacks on the system.

The results are shown in Figure 2 for twenty iterations. With a max delay of 2.5 seconds, the failover is likely to complete considerably faster than the default action of sending an engineer to the site to replace the failed controller.

### B. Performance Experiment

The number of IEDs within the IED network varies depending on the system, so it is important that an automation controller be capable of maintaining reliable performance even when communicating with many different devices. To determine the efficiency of each controller, they were evaluated with a few different device counts from 2 to 60. Each controller was configured to poll data from these devices every two seconds. Meanwhile, a control device periodically requests a value from the controller and the response time is recorded. The performance of the controllers is measured over the course of two minutes and the response time is averaged. These results are shown in Figure 3.

As shown in the results, both the SEL RTAC and the VAC have nearly zero delay when responding to requests. Both controllers are able to maintain a reliable connection to the IEDs and the control center.

### C. Automation Control Experiment

To demonstrate the automation capabilities of the VAC, this experiment evaluates the Modbus communication ability as

|                | SEL RTAC (ms) | VAC (ms) |
|----------------|---------------|----------|
| <b>Average</b> | 156.5         | 48.4     |
| <b>Max</b>     | 158           | 72       |
| <b>Std Dev</b> | 0.81          | 10.37    |

Fig. 4. Delay measured from a value over the threshold being sent and the trip command being received, observed across 20 iterations.

well as the logic processor, which can be programmed to make decisions for the local subsystem. For instance, if a voltage meter detects a large enough spike, it is a good idea to open a circuit breaker to prevent damage to the system. The emulated power system device is set up with a simulated voltage reading that rises over time and a simulated circuit breaker control that can be opened, both of which communicate via Modbus. The RTAC and VAC are given an arbitrary threshold value, 490V, and programmed to send a command to open the breaker when the voltage surpasses this threshold.

To quantify the results, the emulated power system device is set up to record the delay from when it sends a Modbus response where the voltage crosses the threshold to when it receives a Modbus command to open the breaker. As the RTAC and VAC are configured in the same way, the network delay is expected to be equal in both cases, and this scenario disregards any delay related to how often the devices poll the emulated power system device for data. This poll delay is configured by the system operator and can be set anywhere from a fraction of a second to multiple minutes between polls. Instead, this experiment focuses on the delay of the logic engine, which is the time it takes for the automation controller to make a command decision based on incoming data. The results for 20 iterations are shown in Figure 4.

Both the physical SEL RTAC and our VAC performed correctly in this scenario, triggering the breaker to trip on the emulated relay when the value reported by the emulated voltage meter breached the threshold. The VAC was much faster than the SEL RTAC, however this is due in part to the configurable automation delay within the RTAC being set to 100 milliseconds. Lowering this value to as little as 4 milliseconds would likely decrease the total delay. Furthermore, the VAC had a much higher standard deviation across tests with a maximum delay of 72 milliseconds, just under half the delay of the SEL RTAC. Nevertheless, the performance of the VAC's automation engine is on par with that of the SEL RTAC and could be further improved in future revisions of the VAC implementation.

## V. DISCUSSION

### A. Use Cases

The primary discussion throughout this paper has focused on the use of a virtual automation controller as a backup for a physical one. However, it is important to note that there are other potential use cases for the VAC. As mentioned earlier, it is crucial to better secure industrial control systems, however

most of these systems are expensive, large, and difficult to reconfigure. Development of digital twins such as the VAC allows for an inexpensive, flexible, and simple implementation to model a real system for research, leading to new security solutions in the future.

Another use for digital twins is as a training platform. When a piece of critical infrastructure goes offline, first responders must be able to bring it back online as quickly as possible. Basing the VAC on a physical automation controller such as the SEL RTAC means it is similar in function and behavior and can help first responders learn to work with automation controllers.

### B. Limitations

The VAC provides a method for recovering from a failed automation controller, however it also has some limitations. First, the VAC in its current state operates entirely via Ethernet communications. While Ethernet has become much more common in industrial control systems in recent years, serial and even lower-level protocols are still in use. It is possible to implement these in a virtual system, but this might require specialized hardware. The Revolution Pi Core 3+, as used in the evaluation of the VAC, is one example of a platform that could be utilized in the future for serial communications.

While Modbus and DNP3 are currently the most common SCADA protocols currently in use, others exist that may need to be supported. For example, Generic Object Oriented Substation Events (GOOSE) is a publisher-subscriber protocol for IED communication that is becoming more prevalent. EtherNet/IP, which makes Common Industrial Protocol (CIP) available over Ethernet [20], is another protocol that is typically found in manufacturing and automation. Supporting additional protocols would further extend the usefulness of the VAC.

Lastly, in order to withstand harsh industrial environments such as outdoor substations, hardened equipment is a necessity, which increases the cost of the VAC deployment. For example, the Revolution Pi Core 3+ used in the evaluation is one of several Raspberry Pi devices designed specifically for industrial environments, however, its price is much higher than that of a standard Raspberry Pi. Alternatively, to minimize this cost, one could integrate a wireless network tap at the substation that connects remotely to a centralized location where the VAC is located. Assuming the connection is fast and secure, it is a viable option for a single VAC to backup this subsystem as well as others. Furthermore, as the VAC is containerized, a platform such as Kubernetes could allow for a distributed VAC to scale if multiple automation controllers have failed.

## VI. CONCLUSION

This paper has presented the VAC, a digital twin for a SCADA automation controller. It can prevent loss of view and control for power systems by acting as the controller and communicating with DNP3 and Modbus devices. By utilizing SDN, the VAC can take over for the automation controller when a fault is detected and prevent damage to the system

while preserving the controller's state for later forensics. The key capabilities of the automation controller are demonstrated in the example implementation and the performance is on par with the SEL-3505-3 RTAC. As such, the VAC will be useful in supporting existing architecture and mitigating critical security issues that currently exist in SCADA and ICS.

## ACKNOWLEDGEMENT

Research sponsored by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the U. S. Department of Energy.

## REFERENCES

- [1] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Security & Privacy*, 2011.
- [2] A. Greenberg, "Crash Override: The malware that took down a power grid," *Wired Magazine*, 2017.
- [3] D. Whitehead, K. Owens, D. Gammel, and J. Smith, "Ukraine cyber-induced power outage: Analysis and practical mitigation strategies," *Annual Conference for Protective Relay Engineers (CPRE)*, 2017.
- [4] B. Obama, "Presidential Policy Directive 21: Critical Infrastructure Security and Resilience," 2013.
- [5] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, "A survey on cyber security for smart grid communications," *IEEE Communications Surveys Tutorials*, 2012.
- [6] S. A. Yadav, S. R. Kumar, S. Sharma, and A. Singh, "A review of possibilities and solutions of cyber attacks in smart grids," in *Conference on Innovation and Challenges in Cyber Security*, 2016.
- [7] R. Deng, P. Zhuang, and H. Liang, "CCPA: coordinated cyber-physical attacks and countermeasures in smart grid," *IEEE Transactions on Smart Grid*, 2017.
- [8] C. Brosinsky, D. Westermann, and R. Krebs, "Recent and prospective developments in power system control centers: adapting the digital twin technology for application in power system control centers," in *ENERGYCON*, 2018.
- [9] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, "Digital twin in industry: state-of-the-art," *IEEE Transactions on Industrial Informatics*, 2019.
- [10] M. Atalay and P. Angin, "A digital twins approach to smart grid security testing and standardization," in *IEEE Workshop on Metrology for Industry 4.0 IoT*, 2020.
- [11] P. Pileggi, J. Verriet, J. Broekhuijsen, C. v. Leeuwen, W. Wijbrandi, and M. Kongsman, "A digital twin for cyber-physical energy systems," in *Workshop on Modeling and Simulation of Cyber-Physical Energy Systems*, 2019.
- [12] M. Eckhart and A. Ekelhart, "Towards security-aware virtual environments for digital twins," in *ACM Workshop on Cyber-Physical System Security*, 2018.
- [13] B. Miller and D. Rowe, "A survey of SCADA and critical infrastructure incidents," in *ACM Conference on Research in Information Technology*, 2012.
- [14] I. Ahmad, S. Namal, M. Ylianttila, and A. Gurtov, "Security in software defined networks: a survey," *IEEE Communications Surveys Tutorials*, 2015.
- [15] E. G. d. Silva, L. A. D. Knob, J. A. Wickboldt, L. P. Gaspari, L. Z. Granville, and A. Schaeffer-Filho, "Capitalizing on SDN-based SCADA systems: an anti-eavesdropping case-study," in *IEEE symposium on integrated network management*, 2015.
- [16] A. Aydeger, K. Akkaya, and A. S. Uluagac, "SDN-based resilience for smart grid communications," in *IEEE conference on network function virtualization and software defined network*, 2015.
- [17] W. Xu, Y. Tao, and X. Guan, "The landscape of industrial control systems (ICS) devices on the internet," in *Conference on cyber situational awareness, data analytics and assessment*, 2018.
- [18] Modbus FAQ. [Online]. Available: <https://modbus.org/faq.php>
- [19] Overview of the DNP3 protocol. [Online]. Available: <https://www.dnp.org/About/Overview-of-DNP3-Protocol>
- [20] ODVA. Ethernet/IP™ – cip on ethernet technology. [Online]. Available: <https://www.odva.org/wp-content/uploads/2020/05/PUB00138R6-Tech-Series-EtherNetIP.pdf>