Empowering Operational Technology Cybersecurity with the Asset Administration Shell

Empowering Operational Technology Cybersecurity with the Asset Administration Shell

Riccardo Bacca[2], Andrea Melis[1], Lorenzo Rinieri[1], Roberto Girau[1],
Franco Callegati[1,2], Marco Prandini[1,2]
[1] *Department of Computer Science and Engineering - University of Bologna, Bologna, Italy*
[2] *Centre for Industrial Research for Information and Communication Technologies, Bologna, Italy*
{name.surname}@unibo.it

***Abstract*—Communication between assets and systems is one of the foundational principles of the Industry 4.0 paradigm, enabling increased automation, data exchange, and real-time decision-making across industrial environments. However, the growing integration between Operational Technology (OT) networks and core Information Technology (IT) infrastructures — and their progressive exposure to the Internet — introduces a broad spectrum of cybersecurity vulnerabilities. These threats range from unauthorized access and data exfiltration to lateral movement attacks and system-level disruptions, which can significantly impact safety, production continuity, and system integrity.**

Traditional security models often fall short in this context due to the many OT components' unique constraints and legacy nature. This paper explores how Software-Defined Networking (SDN), with its centralized control and programmable architecture, offers a flexible and robust solution to improve the security posture of OT networks. By decoupling the control and data planes, SDN enables fine-grained traffic monitoring, dynamic policy enforcement, and rapid threat mitigation — essential in protecting heterogeneous industrial systems. The paper highlights the key advantages of SDN for OT-IT convergence. It discusses concrete use cases where SDN principles help detect, isolate, and respond to cybersecurity incidents in modern industrial environments.

***Index Terms*—Industrial Security, Modbus, P4, In-Network, SDN, Asset Administration Shell.**

I. Introduction

In modern manufacturing plants, the Operational Technology (OT) network connects OT devices, while the Information Technology (IT) network links standard IT equipment, including servers, computers, and mobile devices. OT is defined as 'hardware and software that detects or causes changes by directly monitoring and/or controlling physical devices, processes, and events in asset-centric industries, particularly in production and operations' [1]. In traditional setups, OT and IT networks are generally kept separate due to the limited need for data exchange, ensuring greater security for OT. However, in an Industry 4.0 context, it is essential to integrate OT and IT to enable seamless communication between production and management processes. However, this interconnection introduces new risks that could severely disrupt the OT environment and production capabilities. This paper proposes a solution to fully integrate a programmable network environment with OT systems, based on the Asset Administration Shell. The final goal is to manage in synergy both the OT components and the OT network, ensuring the necessary security levels for continuous operations and adhering to international standards and guidelines, such as ISA/IEC 62443.

The paper is organized as follows: in section II, we will review the state-of-the-art of the current technologies. In section III, we will introduce the challenges and the motivation behind this. In section IV-A, we will show the use case we developed and deployed to test the effectiveness of our proposed solutions. The experimental results and discussion are then presented in sections V and V-A.

II. State of the Art

In the context of Industry 4.0, the concept of the *Asset Administration Shell* (AAS) has emerged as a cornerstone of interoperability and standardization for cyber-physical systems (CPS). The Platform Industrie 4.0 initiative defines the AAS as a digital representation of a physical asset, providing structured and semantically rich interfaces for data access and control [2]. The AAS is designed to enable modular, decentralized, and flexible automation systems essential in highly dynamic industrial environments. The AAS consists of sub-models, which are collections of specific data and functions related to the asset. These sub-models can be either static, containing fixed data (such as lists of available services, how to access them, the physical properties

of the asset, manuals, etc.), or dynamic, containing real-time data feeds (e.g., retrieved from an OPC-UA server), enabling the execution of active actions [3]. Security, however, remains a critical challenge in industrial systems, especially as the number of connected assets grows. Traditional IT security models often fall short in addressing the specific requirements of Operational Technology (OT), such as real-time constraints, legacy systems, and limited computational resources [4]. As industrial infrastructures become more software-defined and data-driven, incorporating security-by-design mechanisms becomes increasingly essential. Recent studies have explored the role of AAS in enhancing security and trust within industrial OT networks. One promising direction involves integrating security descriptors within the AAS sub-models, enabling assets to declare their security capabilities, configurations, and compliance levels [5]. This allows for automated validation, auditing, and orchestration of security policies at runtime. Another significant advancement is the integration of AAS with Identity and Access Management (IAM) systems. For instance, assets represented via AAS can be assigned digital identities and access rights, enabling fine-grained authorization and dynamic policy enforcement [6]. This is particularly useful in collaborative manufacturing environments where assets from different vendors and domains must interact securely.

Furthermore, the combination of AAS with cryptographic mechanisms and trusted execution environments (TEEs) has been proposed to enable secure bootstrapping, data integrity verification, and secure firmware updates [7]. The research community has also focused on runtime monitoring and anomaly detection by leveraging data exposed through AAS interfaces. By enabling structured access to real-time operational data, AAS can facilitate machine learning-based intrusion detection systems (IDS) tailored to specific asset behaviors [8].

Despite the growing interest, challenges remain regarding the scalability, standardization, and interoperability of security-related AAS submodels. The lack of universally accepted taxonomies and ontologies for security properties hampers seamless integration and reuse. Moreover, ensuring compliance with international security standards such as IEC 62443 within the AAS framework is an ongoing research goal [9].

This is particularly true when considering networking, as the network is often the primary vector initially exploited to attack a manufacturing system. A tighter integration of the networking properties with the operational requirements and goals is key to making the OT environment more secure. Nonetheless, to date, network devices behave according to principles usually set once and for all by network managers, without considering specific operational conditions and with minimal capabilities to adapt in real-time to react to particular challenges and/or threats. We first attempted to overcome this limitation in [10], where we proposed a solution based on integrating Software Defined Networking (SDN) with the AAS paradigm.

In its various forms (centralized control plane, programmable data plane, etc.), which will be further discussed later in this manuscript, SDN enables the real-time programming of network behavior according to application requirements. Moreover, it unifies the interaction with network components, thus opening up the possibility of integrating the network control into sub-models of the AAS.

III. Empowering OT with Software-Defined Networking

From a Cybersecurity point of view, the convergence of IT and OT systems introduces significant cybersecurity challenges due to the heterogeneous nature of IT and OT environments [4], [11]. While IT systems are typically governed by established security frameworks and frequently updated, OT systems are often legacy-driven, with limited support for modern cryptographic protocols, patching, or access control mechanisms [12].

One of the critical challenges in securing OT-IT interaction is the lack of unified visibility into OT assets [13]. Unlike IT environments, where endpoint discovery and management are standardized, OT networks are often opaque, with undocumented devices, proprietary protocols, and minimal telemetry. This hinders comprehensive threat detection and makes real-time inference on system-wide security states difficult [14].

Moreover, the absence of centralized identity and policy management across OT components limits the ability to enforce consistent access controls, making them vulnerable to lateral movement and privilege escalation attacks in the event of a breach. As cyber-physical systems become increasingly complex, the need for holistic, asset-aware cybersecurity inference becomes increasingly urgent. Solutions must be capable of mapping asset topologies, analyzing interactions, and identifying anomalies across IT and OT layers in real time [15].

Addressing these challenges requires developing interoperable, scalable, context-aware security models that bridge the IT and OT visibility gap. Such models should incorporate asset profiling, behavioral analysis, and dynamic policy enforcement to enable proactive defense across the converged environment.

A. Proposal Motivations

Our proposal involves implementing and extending the OT Network's capabilities with SDN programmable

switches and AASs elements for each factory asset, logical or physical. In our specific case, we refer to logical assets, such as Network Infrastructure or Network Control Plane. These, respectively, included the physical switches and the controller.

These assets will interconnect Level 2 and 3 of the proposed architecture with factory machines and sensors. Each switch, Factory Machine, and Network Controller will have its own Asset Administration Shell integrated into the factory's overall management control system, reachable through a high-level and more user-friendly Web User Interface.

The Operational Network will no longer be a static system; instead, it will dynamically adjust its behaviors and programmed network routing rules based on current working conditions or requests. Starting from the System Admin or real-time data analysis incoming from each AAS, network controllers will set up specific network rules to achieve the requirements at each time. Thanks to all factory-connected AAS, these operations are executed rapidly, user-controlled, or automatically to prevent or stop incoming network attacks.

B. Hardware-In-The-Loop

To achieve further improvements, explicability and feasibility of the proposed solution, it's also possible to consider some Hardware in the Loop (HITL) components inside the proposed architecture, as shown in Figure 1. These hardware components are connected to the virtual network infrastructure and are thus fully integrated with the test bed. In the work here reported, we exploited components based on the Raspberry Pi platform, in particular Revolution Pi (RevPi) modules[1].

The RevPi enhances the capabilities and usage possibilities of a standard Raspberry Pi by offering various expansion modules. These modules enable the attachment of multiple sensors, allowing for the collection of a wide range of data, which is then accessible through the Web UI via its Asset Administration Shell. In this case, it is attached to temperature and pressure sensors for dynamic readings. Moreover, in Levels 2-3, we also exploited two Raspberry Pi5 to run the P4 Controllers (one for each Switch). All these components interact with the rest of the test bed exactly in the same way, regardless of whether they are implemented with virtual or real components. They interface with the rest of the world via their respective AASs, which are not concerned about the specific implementation of the asset.

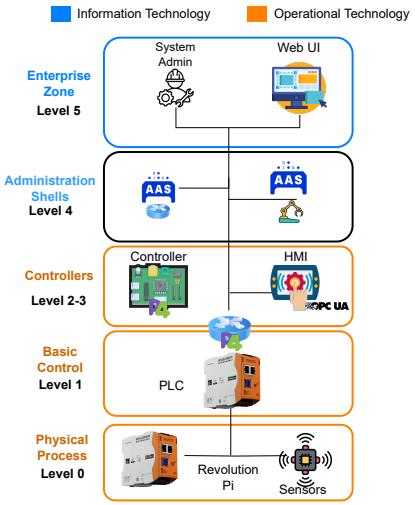

Fig. 1. Hardware in the loop customization and improvements, to the proposed solution.

IV. Testbed Set-up

The testbed used for this work aims to show the feasibility of the proposed approach.

The AAS Infrastructure is developed using Eclipse BaSyx[2] as a library and framework. In particular, we utilized BaSyx Registry and Server to establish the IT infrastructure and the AAS Web UI to provide the final user with an easy-to-use web interface. The BaSyx Java Library was then used to implement and deploy the various AASs. Each AAS is deployed using an Apache HTTP Server on the leading network and registered on the BaSyx Registry and BaSyx Server. The Web UI displays and allows interactions with information about submodels, names, and addresses of each AAS registered on the Server and Registry, directly from the user interface. This method enhances the possibility of abstracting the detailed low-level complexity, providing the user with a unified and intuitive interface.

Each Asset Administration Shell is composed of:

- one or more *Submodels*;
- a Submodel provides one or multiple Operations or Properties
- *Operations* allow the user to make changes to the system or the single factory component,
- *Properties* allow the user only to be made aware of some specific data of the component.

Finally, the AASs can communicate with one another using the provided network configuration, contributing

[1]https://revolutionpi.com/en

[2]https://github.com/eclipse-basyx

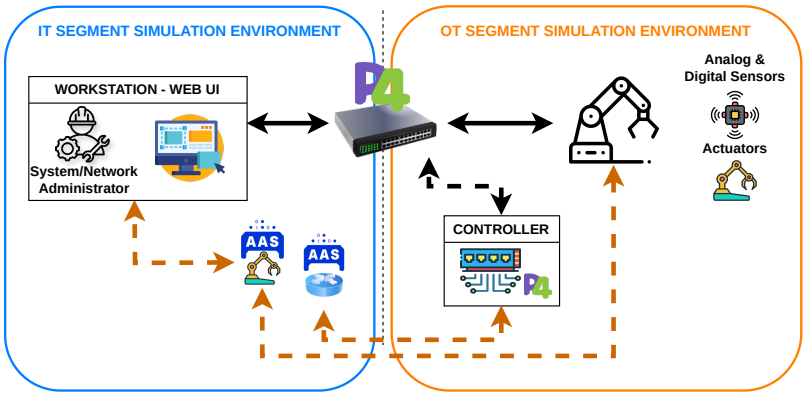

Fig. 2. Scheme showing testbed implementation

to the entire factory's coordination and security.
The Network Configuration to implement the provided Scenarios is composed as shown at a high level in Figure 2. Two P4 Switches act as a conduit between IT and OT infrastructures, configured using two external controllers written in Go. Each Network Asset is represented by an Asset Administration Shell, reachable from the Web UI in the IT Environment.
In detail, the P4 Switches are interconnected and each of them is coupled to its own Network Controller; Network controllers can collect traffic and statistics from the switches and may modify the forwarding rules installed; the Machines are connected as follows: One and Two to Switch 1, Machine Three to Switch 2, as visible in figure3; Each AAS takes information about his physical or logical asset thanks to IT-OT Interaction through each Switch; The Web UI acts as an independent Apache HTTP Server, reachable from a main workstation, and takes info from each AAS, including those devoted to the Switches and Network Controllers.

In technical terms, this network infrastructure was implemented in virtualized form, exploiting Kathará[3], which is an open-source container-based network emulation system designed for demonstrating, developing, and testing production networks in a controlled sandbox. Kathará employs Collision Domains as fundamental network nodes to exchange and interconnect network assets.

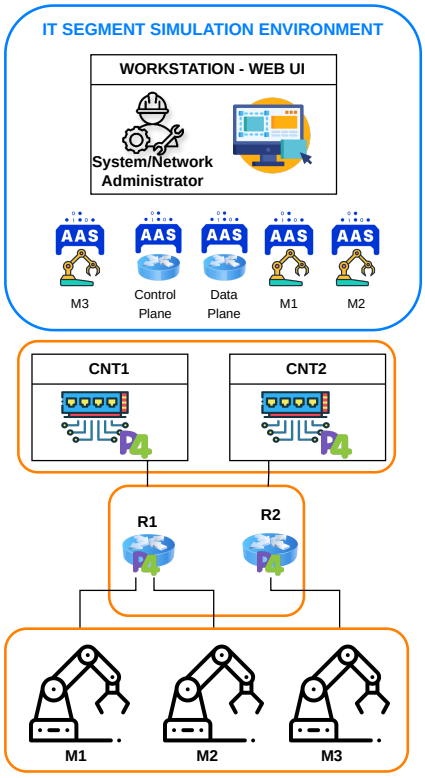

Fig. 3. The architecture considered shows some insights about the exploited use case scenario.

A. *Use case scenario*

The use case presented here is designed to demonstrate the effectiveness of the proposed approach. We refer to the architecture detailed in Figure 3. The main components are:

- three manufacturing machines;
- two SDN Switches supporting P4 based programming[4], implemented with the reference P4 software switch, the behavioral model v2 (bmv2)[5];
- two custom-made switch controllers implemented with the Go Language[6], can load on demand a

[3]https://www.kathara.org

[4]https://www.p4.org
[5]https://github.com/p4lang/behavioral-model
[6]https://www.go.dev

specific p4 program or subset of instructions into the switches, thus changing the network forwarding behavior on demand.

When approaching the AAS architecture, we focused on the functional network representation; therefore, we designed an AAS for the network control plane (specifically, the two switch controllers) and an AAS for the network data plane (specifically, the two switches). This is shown in Figure 3. With this architecture, the network control plane AAS will be responsible for the network's logical behavior; therefore, it will monitor the switches and traffic flows and take any action that requires a network modification via reprogramming. On the other hand, the network data plane AAS will be responsible for monitoring the various switches in real-time, collecting statistics, and identifying potential anomalies.

Figure 3 also clearly shows the logical separation between the IT and the OT networks. The AASs remain within the IT network, interacting to ensure the correct implementation and monitoring of plant behavior. Nonetheless, the interactions between the fundamental plant components happen in the OT network. The interaction between IT and OT happens through well-known and protected traffic flows that connect the assets with the AASs, minimizing the system's potential attack surface.

To demonstrate the effectiveness of this architecture, we utilized a P4-implemented SDN switch equipped with packet and byte counters, which are accessible directly to users via a provided web UI. This enables the network manager, or the autonomous system, to monitor real-time traffic between switches and specific ports, correlating with each machine's CPU usage and adjusting the network routing rules and the factory behavior based on this data. By collecting this data, it's possible, as shown in the provided use cases, to detect any OT network attacks on specific machines. To demonstrate the usefulness and effectiveness of these analyses, the workflow is as follows:

1) At system startup, everything works as it should, and all traffic is routed between machines, controllers, and SDN switches;
2) The SDN Switches could be equipped with custom forwarding rules or entirely new controllers to change network and factory behaviors;
3) At some stage, the Network Manager, using the provided submodel "Get Counters" inside Network Infrastructure AAS, notices a strange quantity of packets and bytes exchanged between two factory machines;
4) At the same time, the CPU usage of those machines is too high compared to standard usage;
5) Here, we have the main difference between the provided use cases, as shown below in Figure 4 and Figure 5.

V. Results

To test the architecture proposed, we experimented with two scenarios: An anomaly in the working conditions of a machine that produces a traffic spike that resolves independently in a short period; An attack on a machine that results in a persistent abnormal traffic pattern, and in an increased CPU workload indicative of a potential attack(Figure 4) and (Figure 5).

In the former scenario, the system detects a temporary increase in network throughput and CPU load, which later normalizes without intervention. This is plotted in Figure 4, where the CPU load (green line) and the machine traffic pattern (purple line) are plotted. The CPU and traffic loads spike higher than usual but then tend to re-normalize (see the traffic decrease at second 80).

In the latter scenario, high network activity and CPU usage go on for a much longer time, as plotted in Figure 5. We assume this may be an indication of an attack.

In both cases, the working statistics of the network and machine are collected by the corresponding AASs. The network control plane detects a higher traffic volume and assumes a possible problem, and checks CPU usage on the corresponding machine AAS.

In the former case, when CPU usage and traffic tend to decrease after some time, no action is taken, assuming this was a spike of work due to some working anomaly that has been solved. On the other hand, in the latter case, when both traffic and CPU stay at higher values than expected, it is assumed that something anomalous, possibly a cybersecurity issue, is ongoing. Therefore, the network control plane deploys a countermeasure by re-programming the switch to block the suspicious communication using the firewall capabilities of the implemented SDN network, successfully mitigating the potential harm to the manufacturing process.

These results confirm that combining SDN and AAS enhances the network's visibility and control, enabling proactive detection and response to threats. The findings underscore the effectiveness of this approach in securing OT networks against cyber threats while maintaining operational efficiency.

A. Conclusion

In this paper, we demonstrate the feasibility of empowering network managers at a manufacturing plant to supervise and manage the network's forwarding behavior in a centralized and highly flexible manner, leveraging the Asset Administration Shell (AAS). The AAS

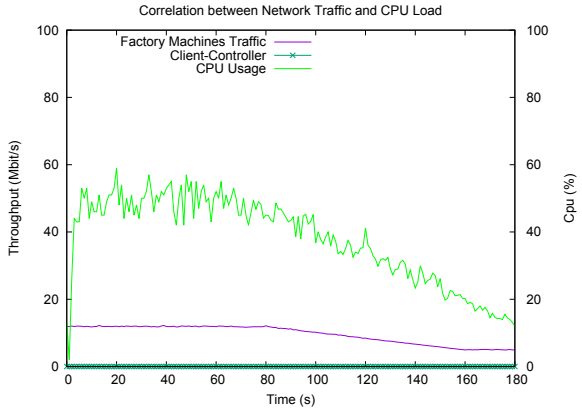

Fig. 4. Example of the OT system control, achievable using factory machines' AAS and CPU usage. Throughput and CPU are higher than normal. Soon after, the data falls quickly, so the network manager takes no action.

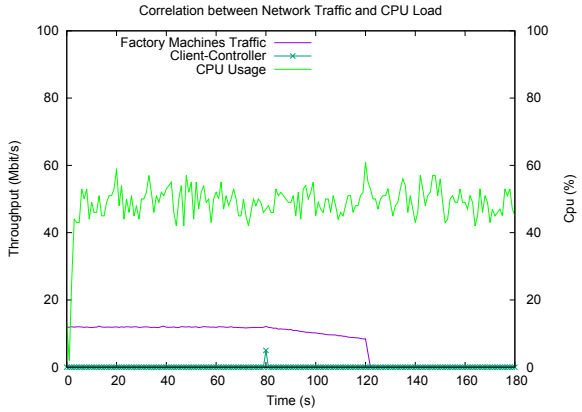

Fig. 5. Example of the OT system control, achievable using factory machines' AAS and CPU usage. Throughput and CPU it's higher than normal. Network managers take action to block possible attacks.

integration facilitates a more streamlined and dynamic network traffic control, enabling swift adjustments to meet the demands of an ever-evolving industrial landscape.

This is particularly valuable in Industry 4.0, where networking is growing more complex. The proposed solution provides the tools necessary to safeguard the integrity of OT networks, reduce vulnerabilities, and enhance the overall reliability of network operations.

ACKNOWLEDGMENT

This study was carried out in the framework of the projects Cri4.0 (CUP: E37G22000490007) and IGNITE5.0 (CUP: E77G22000640003) co-funded by the European funds of the Emilia-Romagna Region, ERDF Regional Programme 2021-2027. This study was also partially funded within the MOST – Sustainable Mobility National Research Center and received funding from the European Union Next-GenerationEU, CN00000023). This manuscript reflects only the authors' views and opinions.

REFERENCES

[1] Gartner, "Operational Technology Security Reviews 2022." https://www.gartner.com/reviews/market/operational-technology-security., 2022. Accessed: August 26, 2022.

[2] Plattform Industrie 4.0, "Details of the asset administration shell – part 1: The exchange of information between partners in the value chain of industrie 4.0," tech. rep., Plattform Industrie 4.0, 2020.

[3] K. L. Lueth, "State of the asset administration shell and standards in industry 4.0," *IoT Analytics*, 2019. Available: https://iot-analytics.com/asset-administration-shell-and-standards-in-industry-4-0/.

[4] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems security—a survey," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1802–1831, 2017.

[5] F. Koehler, H. Trsek, U. Epple, and J. Jasperneite, "Security aspects in the asset administration shell: Towards secure industrie 4.0 components," in *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1–8, IEEE, 2022.

[6] T. Rodrigues, L. Oliveira, N. Kumar, N. Chilamkurti, and A. J. G. A. Filho, "Identity and access management for industrial iot: A blockchain-based approach," *Journal of Network and Computer Applications*, vol. 144, p. 102459, 2020.

[7] E. Vasilomanolakis, J. Daubert, T. Eitelhuber, E. Kiesling, and A. Pretschner, "Towards trusted industrial iot: A secure device onboarding framework using tpm and aas," in *Proceedings of the 2020 IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*, pp. 1–8, IEEE, 2020.

[8] A. Mohamed, F. Zhang, and G. Wainer, "A digital twin-based anomaly detection architecture for industrial control systems," *Simulation Modelling Practice and Theory*, vol. 114, p. 102396, 2022.

[9] International Electrotechnical Commission, "Iec 62443 industrial communication networks - network and system security." https://webstore.iec.ch/publication/60225, 2018. Part 1-1 to 4-2.

[10] R. Bacca, C. Grasselli, G. Tripi, A. Melis, L. H. Bonani, and F. Callegati, "Software-defined and secure industrial networks for the industry 4.0," in *2024 24th International Conference on Transparent Optical Networks (ICTON)*, pp. 1–4, 2024.

[11] B. Group, "It-ot convergence: Building a unified security strategy," 2019. Available at: https://www.bsigroup.com/en-GB/our-services/cybersecurity/insights/it-ot-convergence/.

[12] S. Miller, D. C. Rowe, and C. McMillan, "Threat modeling industrial control systems: A process-aware perspective," *Computers & Security*, vol. 92, p. 101742, 2020.

[13] D. Berardi, F. Callegati, A. Giovine, A. Melis, M. Prandini, and L. Rinieri, "When Operation Technology Meets Information Technology: Challenges and Opportunities," *Future Internet*, vol. 15, no. 3, p. 95, 2023.

[14] E. D. Knapp and J. T. Langill, "Industrial control systems and cybersecurity: a survey of threats and risk mitigation strategies," *ISA Transactions*, vol. 50, no. 1, pp. 16–27, 2011.

[15] C. Grasselli, A. Melis, L. Rinieri, D. Berardi, G. Gori, and A. A. Sadi, "An industrial network digital twin for enhanced security of cyber-physical systems," in *2022 International Symposium on Networks, Computers and Communications (ISNCC)*, pp. 1–7, 2022.