KOTO: A Kubernetes-Based Platform for Secure and Scalable OT Orchestration in Industry 5.0

Nicole Giulianelli
nicole.giulianelli@studio.unibo.it
University of Bologna
Bologna, Italy

Andrea Sabbioni
andrea.sabbioni5@unibo.it
University of Bologna
Bologna, Italy

Sofia Montebugnoli
sofia.montebugnoli3@unibo.it
University of Bologna
Bologna, Italy

Armir Bujari
armir.bujari@unibo.it
University of Bologna
Bologna, Italy

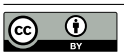
Antonio Corradi
antonio.corradi@unibo.it
University of Bologna
Bologna, Italy

Abstract

The progressive digitization of Operation Technology (OT), thanks also to the integration of Industrial Internet of Things (IIoT) devices, is reshaping the landscape of modern industrial systems. This introduces additional complexity and amplifies the heterogeneity across hardware and software ecosystems, thus exacerbating the challenges of orchestration and lifecycle management, which become particularly intricate and demanding. To meet these challenges, modern DevOps practices, such as Continuous Integration and Continuous Deployment (CI/CD), runtime observability, and fine-grained access control, are essential to shorten time-to-market, ensure service quality, manage operational complexity, while providing security guarantees. Moving a step toward seamless integration of OT into cloud-native ecosystems, we propose Kubernetes-based OT Orchestrator (KOTO), an orchestration platform that extends Kubernetes to enable comprehensive lifecycle management of OT devices in industrial environments. KOTO bridges the gap between IT and OT by seamlessly integrating with existing CI/CD pipelines and implementing Role-Based Access Control (RBAC) mechanisms tailored to device-level permissions. It abstracts the management of heterogeneous, multi-vendor hardware through a unified abstraction, enabling scalable and secure operations across diverse industrial setups. Without loss of generality, we demonstrate the use of the platform in managing Programmable Logic Controllers (PLCs). We then perform extensive evaluations in a realistic deployment environment, validating KOTO's effectiveness in enhancing operational resilience and responsiveness.

CCS Concepts

• **Computer systems organization** → **Distributed architectures**; **Sensors and actuators**; • **Security and privacy** → **Distributed systems security**; • **Applied computing** → **Command and control**; • **Software and its engineering** → *Interoperability*.



UCC '25, France, France

ACM ISBN 979-8-4007-2285-1/2025/12
https://doi.org/10.1145/3773274.3774700

Keywords

Smart manufacturing, Industry 5.0, IT/OT, Orchestration, Observability, eBPF, PLC

ACM Reference Format:
Nicole Giulianelli, Andrea Sabbioni, Sofia Montebugnoli, Armir Bujari, and Antonio Corradi. 2025. KOTO: A Kubernetes-Based Platform for Secure and Scalable OT Orchestration in Industry 5.0. In *2025 IEEE/ACM 18th International Conference on Utility and Cloud Computing (UCC '25), December 1–4, 2025, France, France.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3773274.3774700

1 Introduction

The advent of Industry 5.0 marks a paradigm shift in industrial automation, driven by the convergence of Information Technology (IT) and Operational Technology (OT). This integration embodies the promise to deliver improved reliability, scalability, and process efficiency, at the expense of introducing greater complexity in the coordination of heterogeneous system architectures [4]. The proliferation of Industrial Internet of Things (IIoT) devices and the digitization of OT environments further amplify these challenges, as diverse hardware and software ecosystems must interoperate within safety-critical dynamic infrastructures. Unlike earlier automation paradigms, Industry 5.0 introduces a human-centric dimension: advanced systems are designed to increase human decision-making, leverage operator expertise in complex scenarios, and enable adaptive control loops where human insight remains integral to safety, resilience, and long-term system sustainability [4].

To manage the complexity, modern industrial systems increasingly adopt Development and Operations (DevOps) practices, which have proven effective in bridging development and operations in IT domains [2]. Techniques such as Continuous Integration and Continuous Deployment (CI/CD) , runtime observability, and fine-grained access control are essential to accelerate time-to-market, ensure service quality, and maintain security in distributed environments. However, extending these practices to OT introduces unique constraints, including deterministic behavior, low latency requirements, and stringent safety guarantees, which traditional IT-oriented solutions are not designed to meet [3].

In this context, asset orchestration and lifecycle management emerge as critical enablers for IT/OT convergence. Orchestration provides automated configuration and coordination of resources, while observability ensures real-time visibility into system health,

enabling proactive maintenance and anomaly detection. However, existing orchestration frameworks often lack the capabilities required for industrial scenarios, particularly when managing Programmable Logic Controllers (PLCs) and their virtualized/software counterparts (vPLC and softPLC), which are fundamental for industrial control systems [8].

To overcome these limitations, we propose a holistic framework named Kubernetes-based OT Orchestrator (Kubernetes-based OT Orchestrator (KOTO)), a cloud-native platform that extends Kubernetes to support the comprehensive lifecycle management of OT resources. This is achieved by introducing a unified interface for managing heterogeneous multivendor devices that integrate seamlessly with existing DevOps tools, also supporting essential security features, such as permissioned access to resources and auditing of operations. Furthermore, KOTO advances state-of-the-art orchestration platforms by leveraging a novel event-driven control loop and evolved Berkeley Packet Filter (eBPF)-based observability to deliver low-latency, high-resolution monitoring without compromising performance. By bridging the gap between IT and OT, KOTO enables scalable, secure, and resilient operations, positioning itself as a key enabler for next-generation industrial automation.

We demonstrate the capabilities of KOTO through comprehensive case studies and performance evaluations, illustrating its practical benefits in real-world scenarios. In particular, we show how KOTO addresses the limitations of state-of-the-art orchestration platforms by reliably and efficiently managing heterogeneous OT assets under diverse workload conditions.

The remainder of the paper is organized as follows: Section 2 provides the necessary background, establishing the conceptual and operational foundations of our solution and distinguishing it from previous work. Section 3 introduces the proposed solution model and describes in detail its design and functional behavior. Section 4 reports the experimental evaluation and performance analysis. Finally, Section 5 draws the conclusions and outlines future research directions.

2 Background and Related Work

The ongoing digitization of OT is reshaping industrial systems by enabling real-time monitoring, control, and automation across diverse domains [2, 15]. This evolution aligns with the broader transition toward Cyber-Physical Systems (CPS), promoting industrial solutions that are human-centric, adaptive, and resilient [5].

Historically, industrial automation relied on hardware-based control systems, with PLCs serving as the backbone for reliable and deterministic operations on factory floors [8]. However, the increasing demand for flexibility, scalability, and remote management has driven the softwarization of these controllers, opening the road to software-defined control and seamless integration with IT-driven orchestration frameworks [11]. This evolution aligns with the adoption of enabling technologies such as Time-Sensitive Networking (TSN), private 5G, and edge computing, which collectively reduce latency and improve system resilience [7, 8, 11].

It is becoming clear that the convergence of OT and IT is no longer optional but a strategic necessity, as it enables data-driven insights, improves responsiveness and reduces operational silos [1]. However, this integration presents significant challenges in terms

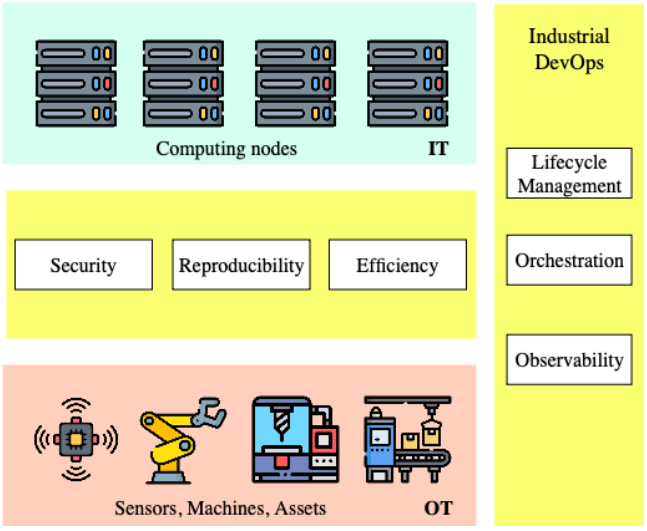


Figure 1: DevOps tools to enhance security, reproducibility, and efficiency in IT/OT convergence

of interoperability, lifecycle expectations, and security. While IT systems prioritize agility and scalability, OT environments demand deterministic behavior and long-term reliability [5]. To address these concerns, the series of international standards IEC 62443 [3] provide a comprehensive framework to secure industrial automation and control systems. IEC 62443 defines security requirements across multiple layers, ranging from component level hardening to system-wide risk management, ensuring that orchestration platforms comply with stringent cybersecurity and safety constraints.

In fact, bridging the OT/IT divide requires the adoption of modern software engineering practices. DevOps practices, originally conceived for IT domains to promote CI/CD, are now being extended to industrial contexts [9, 13]. Industrial DevOps fosters collaboration between IT and OT teams by embedding automation, low-code orchestration, and data-driven decision-making across industrial layers [1, 14]. Recent studies demonstrate that applying DevOps principles to IIoT and CPS infrastructures improves time-to-deployment, fault detection, and overall system agility [2, 5].

In this context, applying DevOps to industrial control systems (Fig. 1), particularly PLC, and software counterparts, offers substantial benefits:

- **Security**: Standardized deployment pipelines reduce misconfiguration risks and unauthorized changes.
- **Reproducibility**: Development and test environments can be precisely replicated, improving validation and quality assurance.
- **Efficiency**: CI/CD pipelines enable faster rollouts, help minimize downtime, and optimize resource usage.

The key enablers of industrial DevOps include observability, orchestration, and lifecycle management. Observability ensures real-time visibility into system health, enabling proactive maintenance and anomaly detection [10]. Orchestration automates the configuration and coordination of heterogeneous resources, while lifecycle management encompasses version control, automated updates, and fault recovery [11, 12]. These capabilities are essential for

managing heterogeneity, ensuring reproducibility, and maintaining compliance with standards such as IEC 62443.

This rapidly evolving landscape requires new solutions to address the complexity of integrating modern DevOps techniques while securely managing heterogeneous OT environments. Our solution, KOTO, addresses these challenges by introducing the novel adoption of Kubernetes for OT management, extending and tailoring it to meet the specific requirements of industrial contexts.

3 KOTO: Kubernetes-based OT Orchestrator

To address the complexity of managing the increasing number of heterogeneous OT devices and enable DevOps-driven automation, we present KOTO, a platform supporting automation and DevOps practices in industrial environments, providing a single point of control for the entire lifecycle of OT assets. KOTO introduces a unified interface to define, configure, and query heterogeneous resources, abstract vendor-specific heterogeneity, and ensure consistency between deployments.

By logically centralizing management, KOTO simplifies integration with automated pipelines and enforces security through fine-grained access control. This approach reduces operational overhead, minimizes configuration errors, and supports smarter, more secure orchestration of industrial systems, paving the way for scalable and resilient automation.

KOTO, shown in Figure 2, provides a unified and secure framework to manage heterogeneous assets OT. Its architecture is composed of five core components: i) an Application Programming Interface (API) Server that serves as the primary interface for user interactions, ii) a Configuration Registry that ensures persistent and consistent storage of resource definitions, iii) a Controller that automates lifecycle management through a reconciliation loop, iv) an Authentication, Authorization, and Auditing (AAA) service enforcing authentication, authorization, and auditing policies, and v) an Adapter module that abstracts vendor-specific details to enable interoperability across diverse industrial environments.

At the entry point, the *API Server* acts as the main interface for user interactions, allowing operators to define, update, or remove OT resources following standar cloud resource representations [6]. This simplifies integration with existing processes and state of the art DevOps tools, ensuring a consistent and standardized approach to resource management.

The *Configuration Registry* provides persistent storage for the definitions of the OT resources and related metadata, ensuring a clear separation between the specification of the resource and the execution. This design guarantees consistency, facilitates recovery in the event of failure, and maintains historical versions of the configurations for audit and rollback purposes.

At the core of the system, the *Controller* implements the IIoT orchestration logic following a reconciliation loop model, enacting necessary corrective actions to reach the desired state of the OT asset/process as defined in the registry. It automates the entire resource lifecycle, from provisioning and configuration to monitoring and secure decommissioning, ensuring state consistency and reducing the risk of human error through repetitive and controlled processes. As part of the Controller, the *Observability* system performs pervasive gathering and aggregation of metrics, providing

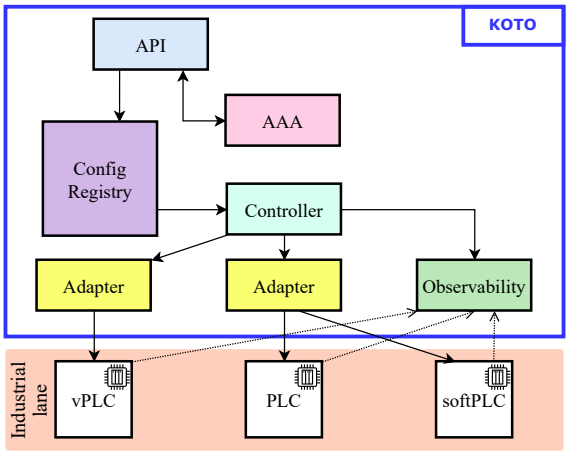


Figure 2: High-level architecture of KOTO featuring essential elements to enable a holistic management of IIoT-enabled OT assets

the controller with the current state of the managed resources. To this end, we adopt a lightweight agent-based architecture, used for continuous monitoring of the behavior and status of OT resources, providing a comprehensive view of the system's health. The acquisition of performance indicators and the detection of real-time anomalies facilitate troubleshooting and ensure that KOTO can maintain consistency and reliability throughout the OT infrastructure.

Throughout KOTO, Security is enforced through the *AAA* service, which integrates authentication, authorization, and auditing. Authentication ensures that only legitimate users or systems can access resources, authorization applies fine-grained permissions to control operations, and auditing records all actions for traceability and compliance. This approach aligns with the security requirements of IEC 62443, particularly those related to identity management, accountability, and secure access control. Before propagating new configurations or programs to IIoT devices, the Controller verifies their origin and integrity through mechanisms, such as digital signatures and certificate-based validation, ensuring that only trusted and verified artifacts are deployed. By enforcing role-based permissions, validating resource authenticity, and maintaining detailed audit trails, the platform supports the implementation of zones and conduits as recommended by the standard, allowing segmentation of assets and controlled communication paths. Finally, the *Adapter* module abstracts vendor-specific heterogeneity by translating generic vendor-independent resource definitions into manufacturer-specific commands and configurations, such as those required by Siemens or Codesys devices. This capability enables multi-vendor orchestration, reduces lock-in, and provides a flexible, scalable foundation for industrial automation. By shielding users from low-level technical differences, the adapter feature ensures that resource management remains consistent and efficient in heterogeneous environments.

To onboard a new OT asset into the system or make any changes to an existing one, the user interacts with the system through the

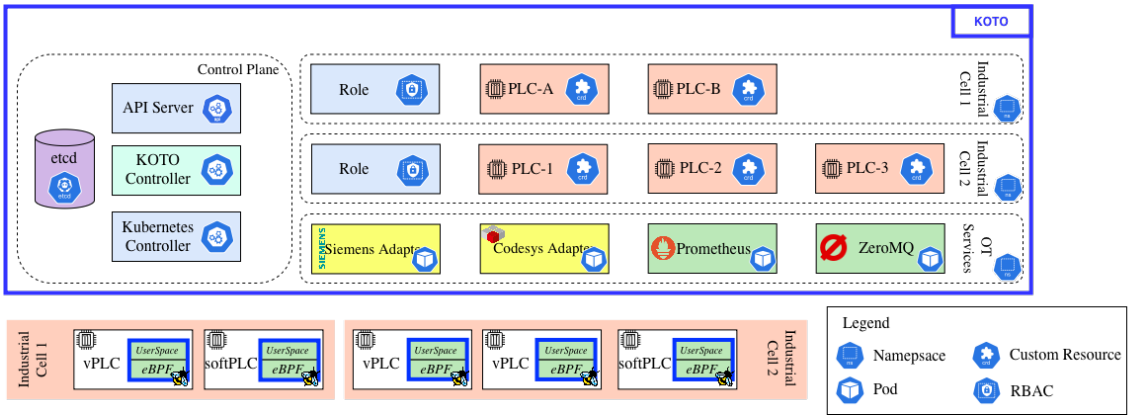


Figure 3: KOTO Kubernets-based implementation, supporting PLC management through namespace-confined custom resources and a specialized Kubernetes controller

API Server, which is the entry point to the platform, allowing one to define and submit OT resource definitions. Before any operation is authorized and executed, the AAA service first records a log of the requested action for future auditing purposes. Next, it checks whether the user identity that submits the OT resource definition within the specified namespace has the required permissions to perform the action. Once authorized, the submitted definitions are stored in the Configuration Registry, which serves as the persistent and structured source of truth for all OT resources. This registry is the persistent and consistent store that the Controller references, monitoring its state to ensure that the managed OT resources align with the desired configuration. Whenever the Controller observes a deviation in the desired state and the observed state, it plans a series of actions to achieve the desired state, supported by the Adapter module, to which it delegates low-level configuration tasks.

3.1 Implementation

We now present the first prototype implementation focused on the management of IIoT-enabled PLCs. The increasing heterogeneity of technologies and vendors, combined with the recent rise of virtual PLCs, makes modern management of these assets a primary concern, allowing secure operations and fostering the adoption of DevOps practices in industrial contexts.

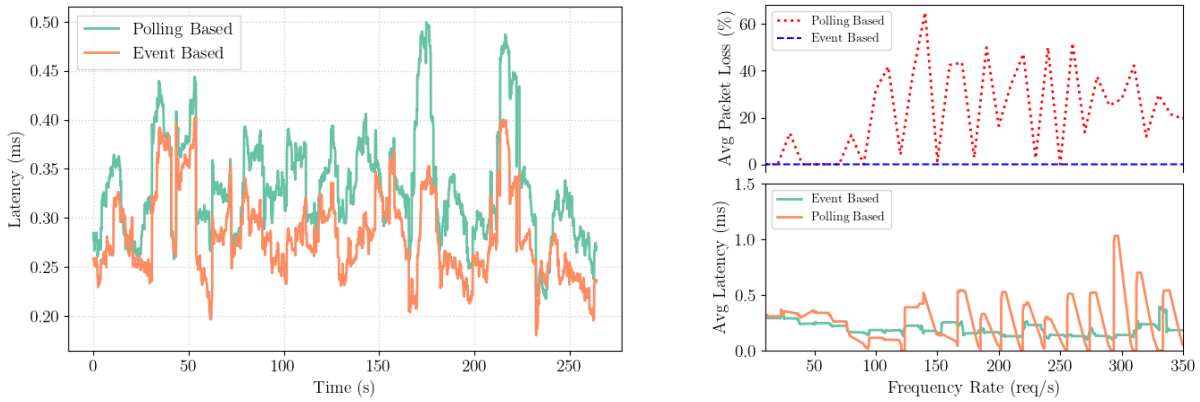
The implementation architecture, illustrated in Figure 3, builds upon Kubernetes, a widely adopted open-source platform used to automate the deployment, scaling, and management of containerized applications. KOTO extends Kubernetes to support the control and lifecycle management of industrial PLCs through the Operator Pattern [6]. This pattern leverages Custom Resource Defintions (CRDs) and controllers to encode domain-specific operational logic into software, enabling Kubernetes to manage complex applications beyond its native capabilities. The KOTO Controller is implemented as a Kubernetes Operator and its associated CRD, extending the control plane to define and manage the deployment, configuration, and full life cycle of industrial control systems, including PLCs, vPLC, and softPLC.

To define, onboard, and deploy a PLC in KOTO, users provide a standard Kubernetes manifest that describes the corresponding CRD, which includes the characteristics of the object and the configuration metadata. Once submitted, the manifest is persisted in *etcd* via the Kube-API Server. The KOTO Controller continuously monitors these definitions, compares them with the observed state, and, in case of discrepancies, plans and executes the necessary actions to reconcile the system with the desired state. Unlike the traditional Kubernetes reconciliation loop, which relies on periodic polling, the KOTO Controller adopts an event-driven model, directly subscribing to changes streamed by the Observability system, enabling faster reactions to state changes, reducing latency, and improving synchronization with real-world industrial processes.

To enable event-driven notifications of state changes in observed resources, the Observability system integrates *Prometheus* and *ZeroMQ*, which act as a metrics collector and a messaging broker, respectively, for data gathered from lower OT layers. Prometheus is an open-source monitoring and alerting system designed for adaptability, reliability, and scalability, providing extensive support for metric collection from OT assets. ZeroMQ, on the other hand, is a high-performance asynchronous messaging library optimized for building scalable and distributed applications. In particular, to support time-sensitive deployments, KOTO leverages an eBPF agent, namely *bpftrace*, to perform non-intrusive traffic analysis in vPLC and softPLC, which is then streamed directly to the controller KOTO through ZeroMQ's lightweight queues.

To address the heterogeneity of PLC technologies, characterized by diverse protocols, data models, and licensing constraints, the controller relies on the Adapter module. This consists of containerized components that expose a uniform REST API to perform device-specific operations, such as program deployment or configuration updates, while abstracting vendor-specific details.

In KOTO, PLC-type resources are organized within Kubernetes namespaces, providing logical isolation and enforcing access control through Role Base Access Control (RBAC). Security is embedded across the architecture in alignment with IEC 62443 guidelines, thereby enforcing defense-in-depth and least-privilege principles. Kubernetes-native RBAC enables fine-grained authorization, while all internal communications are protected using Transport Layer Security (TLS) encryption to guarantee confidentiality and integrity.



(a) (b)

Figure 4: Latency of a traditional (poll-based) Kubernetes controller versus the KOTO Controller (event-based) in detecting mismatches between observed and desired states under two conditions: (a) constant observability metric injection, and (b) linearly increasing load from 0 to 350 requests per second, injected in increments of 10 requests/second.

The Controller validates the authenticity and integrity of the onboarded resources before propagating configuration changes to adapters and devices, preventing unauthorized or tampered configurations. Namespace-based isolation further restricts resource creation and operations to authorized users, supporting multi-tenancy and compliance with industrial cybersecurity standards.

4 Experimental Results

The objective of the experimental analysis is twofold: (i) validate and evaluate the end-to-end performance of KOTO in observing resources in industrial environments and (ii) assess the performance benefits of the eBPF-based in-kernel observability mechanisms when compared to the traditional user-space monitoring approaches. The testbed environment consists of a three-node Kubernetes cluster, each equipped with 8 CPUs and 32GB of RAM, with KOTO deployed on the master node, supplemented by three dedicated virtual machines: two hosting Soft Programmable Logic Controller (softPLC) and Virtual Programmable Logic Controller (vPLC) instances, and one dedicated to running the Codesys adapter.

In particular, in the first experiment, we assess the orchestration capabilities of KOTO in comparison to a traditional Kubebuilder-based operator, with a specific focus on the transition from a polling-based mechanism to an event-driven processing model. The assessment examines the effect of this architectural change on operator responsiveness, particularly in scenarios with high-frequency observability updates coming from softPLC and vPLC instances through the Codesys adapter. This architectural change is expected to substantially reduce latency, allowing more timely and reactive handling of control messages, an essential requirement for responsiveness in industrial environments.

In addition, a key contribution of our approach is the integration of in-kernel data plane processing through eBPF, which enables low-latency, end-to-end communication between IT and OT domains. In this context, our objective is to show that offloading critical packet handling operations to the kernel space accelerates monitoring processes and improves the overall reliability, efficiency, and security of the system. The use of eBPF-based in-kernel mechanisms significantly reduces message processing time, enabling near-real-time control over OT resources.

4.1 Event-Driven vs. Periodic Control

In the first test, we evaluated KOTO's capability to orchestrate OT resources, specifically softPLC and vPLC. As a direct metric, we measure the latency, computed as the time interval between the generation of an observability signal, reflecting the actual resource state of the resource under control, and the moment the signal is received by the controller. To demonstrate the benefits of our approach, we evaluate an end-to-end deployment of our solution under different traffic patterns. In Figure 4a, we consider a constant submission rate of observability data. In contrast, Figure 4b also evaluates a monotonically increasing rate, starting at 10 submissions per second and increasing by 10 submissions per second with each subsequent second. Due to this rapid growth, the test lasted only six minutes, as the Kubebuilder-based operator was unable to sustain higher loads without failing.

Figure 4 compares the average latency with which the Controller detects deviations between the current and desired states and triggers the reconciliation loop. The results contrast KOTO with a traditional Kubebuilder-based solution under the two scenarios described previously: constant load and incremental load. Both scenarios provide some evidence to the effectiveness of KOTO Controller. In particular, under critical load conditions, traditional orchestration solutions suffer from significant data loss: as the update rate increases, an increasing number of events are dropped, reaching up to nearly half of those submitted. These results also underline the importance of architectural solutions, such as KOTO, which extends existing operator-based mechanisms to provide improved robustness and responsiveness under demanding workloads.

4.2 Monitoring Efficiency

In the second test, we perform a zoom-in to evaluate the efficiency of kernel-space monitoring compared to user-space monitoring. The evaluation focused on latency, since it represents a fundamental metric in industrial environments, particularly for PLC, where predictable and timely responses are critical.

For this experiment, we submitted a linearly increasing number of requests over a duration of 35 seconds, to test both approaches under stressful load conditions. The results in Figure 5 show that kernel-space monitoring, implemented through eBPF, exhibits lower and more stable latency, with limited variance compared to the user-space approach. In contrast, the user-space monitoring shows significantly higher latencies, characterized by frequent spikes that in some cases reach an order of magnitude higher than the kernel-space. These results highlight the importance of executing monitoring operations directly in the kernel to reduce overhead and ensure more predictable performance.

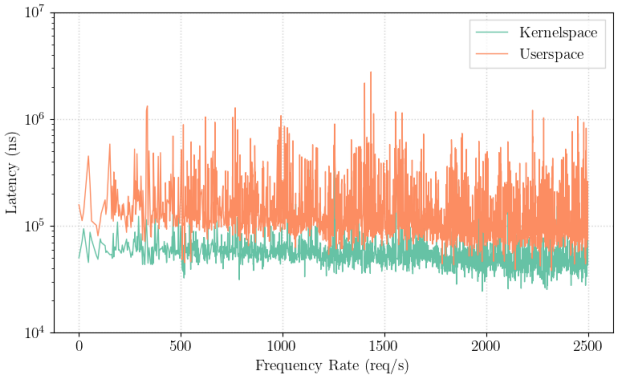


Figure 5: Comparison of traditional monitoring versus eBPF (kernel-level) monitoring performance, with a linearly increasing load from 0 to 2500 requests per second

5 Conclusion and Future Work

The adoption of DevOps tools for orchestration in Industry 5.0 scenarios is essential, as it enables continuous integration, advanced monitoring, and automation of the software lifecycle. However, existing orchestration solutions do not fully support OT requirements. In this work, we presented KOTO, a platform that brings DevOps principles for the flexible management of OT resources, addressing key limitations of existing orchestrators, particularly those concerning time-sensitive operations, polling-based semantics, and the need for pervasive observability. Our experimental results show that KOTO can efficiently orchestrate heterogeneous OT resources, improving responsiveness, reliability, and operational visibility through an event-driven approach and the use of eBPF. In future work, we plan to extend the capabilities of KOTO by supporting the onboarding & management of additional OT assets, such as industrial sensors and networks, and by improving the support for failure scenarios through failover and load-balancing mechanisms.

Acknowledgments

This work is supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, "SEcurity and RIghts In the CyberSpace" (PE00000014 - program "SERICS") CUP:J33C22002810001.

References

[1] Ch. Benmhamed *et al.* 2024. Industrial DevOps Soft: A New Framework for Enhancing Industry 4.0 Efficiency. In *Proc. of International Conference on Smart-Digital-Green Technologies and Artificial Intelligence Sciences (CSDGAIS)*. 1–7. doi:10.1109/CSDGAIS64098.2024.11064768

[2] Anurag Choudhry and Anshu Premchand. 2021. Microservices and DevOps for Optimal Benefits from IoT in Manufacturing. In *Proc. of International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications*, Vinit Kumar Gunjan and Jacek M. Zurada (Eds.). Springer Singapore, Singapore, 375–384. doi:10.1007/978-981-15-7234-0_33

[3] Ivan Cindrić, Marko Jurčević, and Tamara Hadjina. 2025. Mapping of Industrial IoT to IEC 62443 Standards. *Sensors (Basel, Switzerland)* 25, 3 (2025), 728.

[4] European Commission, Directorate-General for Research, Innovation, M. Breque, L. De Nul, and A. Petridis. 2021. *Industry 5.0 – Towards a sustainable, human-centric and resilient European industry.* Publications Office of the European Union. doi:doi/10.2777/308407

[5] Jürgen Dobaj, Andreas Riel, Georg Macher, and Markus Egretzberger. 2023. Towards DevOps for Cyber-Physical Systems (CPSs): Resilient Self-Adaptive Software for Sustainable Human-Centric Smart CPS Facilitated by Digital Twins. *Machines* 11, 10 (2023).

[6] Jason Dobies and Joshua Wood. 2020. *Kubernetes operators: Automating the container orchestration platform.* O'Reilly Media.

[7] Massimiliano Gaffurini, Paolo Bellagente, Alessandro Depari, Alessandra Flammini, Emiliano Sisinni, and Paolo Ferrari. 2024. Virtual PLC in Industrial Edge Platform: Performance Evaluation of Supervision and Control Communication. *IEEE Transactions on Instrumentation and Measurement* 73 (2024), 1–10. doi:10.1109/TIM.2024.3370746

[8] Massimiliano Gaffurini, Dennis Brandão, Stefano Rinaldi, Alessandra Flammini, Emiliano Sisinni, and Paolo Ferrari. 2025. Characterizing the Real-Time Communication Performance of Virtual PLC in Industrial Edge Platform. *IEEE Open Journal of Instrumentation and Measurement* 4 (2025), 1–11. doi:10.1109/OJIM.2025.3559573

[9] Wilhelm Hasselbring, Soren Henning, Bjorn Latte, Armin Mobius, Thomas Richter, Stefani Schalk, and Maik Wojcieszak. 2019. Industrial DevOps. In *Proc. of IEEE International Conference on Software Architecture Companion (ICSA-C)*. IEEE, 123–126.

[10] Ji, Zhiduo and Chen, Cailian and He, Jianping and Zhu, Shanying and Guan, Xinping. 2022. Edge Sensing and Control Co-Design for Industrial Cyber-Physical Systems: Observability Guaranteed Method. *IEEE Transactions on Cybernetics* 52, 12 (2022), 13350–13362. doi:10.1109/TCYB.2021.3079149

[11] Lorenzo Rosa, Andrea Garbugli, Lorenzo Patera, and Luca Foschini. 2023. Supporting vPLC Networking over TSN with Kubernetes in Industry 4.0. In *Proc. of the 1st Workshop on Enhanced Network Techniques and Technologies for the Industrial IoT to Cloud Continuum* (New York, NY, USA) *(IIoT-NETs '23)*. Association for Computing Machinery, New York, NY, USA, 15–21.

[12] Jorge Sasiain, David Franco, Asier Atutxa, Jasone Astorga, and Eduardo Jacob. 2024. Toward the integration and convergence between 5G and tsn technologies and architectures for industrial communications: a survey. *IEEE Communications Surveys & Tutorials* 27, 1 (2024), 259–321.

[13] Mojtaba Shahin, Muhammad Ali Babar, and Liming Zhu. 2017. Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices. *IEEE Access* 5 (2017), 3909–3943. doi:10.1109/ACCESS.2017.2685629

[14] Stephen John Warnett and Uwe Zdun. 2025. Bridging the Gap Between MLOps and RLOps: An Industry 4.0 Case Study on Architectural Design Decisions in Practice. In *2025 IEEE 22nd International Conference on Software Architecture (ICSA)*. 232–242. doi:10.1109/ICSA65012.2025.00031

[15] Michal Wisniewski, Bartlomiej Gladysz, Krzysztof Ejsmont, Andrzej Wodecki, and Tim Van Erp. 2022. Industry 4.0 Solutions Impacts on Critical Infrastructure Safety and Protection–A Systematic Literature Review. *IEEE Access* 10 (2022), 82716–82735. doi:10.1109/ACCESS.2022.3195337