Thesis Proposal

Comprehensive Study of a Security Testing Framework for Cloud-Native Real-Time Applications Using Local GitLab CI/CD on Bare-Metal Kubernetes.

Author: Md Hazrat Ali

Technology
Arts Sciences
TH Köln

Formal Information

1.1 Course

Study course / Studiengang	MaCSN
Course / Modul	Thesis (30 ECTS)

1.2 Student

First name and surname	Md Hazrat Ali
E-Mail address	md._hazrat.ali@smail.th-koeln.de md.hazrat.ali@fraunhofer.ipt.de
Student ID No.	11151237

1.3 External company

Company name	Fraunhofer Institute for Production Technology (IPT)
Company address	Steinbachstraße 17, 52074 Aachen, Germany
Name(s) of Supervisor(s)	Talib Sankal Maximilian Ortmann
Phone / mobile. no.	+49 241 8904-322 +49 241 8904-231
E-mail addresses	talib.sankal@ipt.fraunhofer.de maximilian.ortmann@ipt.fraunhofer.de
2nd examiner's (one of the supervisors) highest academic grade	M Sc. M Sc.

1.4 Organizational Information / Organisatorische Angaben

Planned start date	July - 2025
Proposed end date	November - 2025
Weekly workload	40 hours

2. Introduction

Cloud-native applications, with their microservice architectures and containerized workloads, are at the heart of modern industrial IT. Real-time systems must meet strict demands for scalability, low latency, and continuous security. However, traditional security methods such as manual penetration testing or external scanning often fall short, especially in on-premises environments.

As industrial systems adopt faster, automated software-delivery pipelines, security too often remains an afterthought. This can lead to overlooked vulnerabilities with potentially serious consequences in settings where reliability and safety are crucial. Although DevSecOps tools are gaining traction in the cloud, there is little research on their effectiveness in on-premises, real-time, microservice-based setups particularly in real industrial scenarios.

This thesis proposes a security-testing framework that embeds both static and dynamic threat analysis directly into self-hosted GitLab CI/CD pipelines, deploying to a bare-metal Kubernetes cluster. The framework will automate compliance gates, vulnerability scanning, runtime monitoring, and certification-ready reporting all within local industrial contexts. Conducted at Fraunhofer IPT as part of the Factory Cloud project, this work aims to deliver practical insights into how DevSecOps practices affect both security and operational performance in real-time cloud-native applications.

3. Problem Statement

Most research and case studies focus on DevSecOps in cloud environments or discuss best practices only in theory. However, there is still a lack of empirical, systematic research on the real-world effects of integrating DevSecOps tools end-to-end in a local, microservice-based pipeline, especially under real-time industrial workloads.

Although numerous open-source tools exist for individual phases of security testing, there is a lack of a unified, on-premises, and automated framework that:

- Seamlessly integrates into local CI/CD pipelines.
- Performs both static (code/image) and dynamic (runtime) security analysis.
- Ensures standards-aligned compliance.
- Operates reliably in real-time and on-premises environments.
- Automates security reporting, enforcement, and monitoring.

Without such a framework and real-world evaluation, not only are there security risks, but organizations like Fraunhofer IPT may also face increased barriers to certification and compliance - both crucial for maintaining trust and a competitive edge in digital manufacturing.

4. Objectives

The main objectives of this thesis are:

- ❖ Design a modular security testing framework for cloud-native real-time applications.
- ❖ Integrate static, dynamic, and hybrid threat analysis methods into a local GitLab CI/CD pipeline.
- ❖ Implement and thoroughly evaluate the framework using realistic test environments that meet conformity and industry standards.
- ❖ Measure the framework's security effectiveness, reproducibility, and performance across different strategies, while ensuring alignment with conformity assessment and industry certification requirements.

5. Research Questions

This thesis is guided by the following research questions:

- ❖ How does integrating both static and dynamic security analysis into a local CI/CD pipeline influence the detection of vulnerabilities, compliance outcomes, and operational efficiency in real-time cloud-native applications?
- ❖ What practical differences in security, speed, and resource usage can be expected when comparing static-only, dynamic-only, and combined (hybrid) security pipelines?

- How much does automating policy enforcement with tools like Kyverno or OPA improve deployment reliability and reduce compliance issues, compared to manual or no enforcement?
- What real-world challenges or limits arise when implementing a comprehensive, on-premises DevSecOps framework, especially as organizations scale or adapt it to different environments?
- In which conditions do the proposed integrated security framework deliver the most significant improvements over traditional (non-automated) pipelines, and where might organizations need to balance between security, performance, and the effort needed for certification?

6. Research Contribution and Methodology

This section presents the methodological framework and architectural design used to implement and, crucially, experimentally evaluate a comprehensive security testing pipeline for cloud-native real-time applications. The approach combines both static and dynamic threat analysis within a local CI/CD environment, ensuring full pipeline automation and reproducibility. All experiments are performed on a bare-metal Kubernetes infrastructure to simulate production-like conditions while maintaining control over performance.

6.1 Research Contribution

This thesis aims to provide:

- A systematic, empirical evaluation of integrated security testing strategies within local GitLab CI/CD pipelines on bare-metal Kubernetes for real-time cloud-native applications.
- Analysis of the impact on vulnerability detection, compliance, performance, and deployment reliability across different pipeline configurations.
- Demonstration of how automated reporting and evidence collection within the CI/CD process can streamline and partially automate the certification process for cloud-native applications.
- Practical recommendations and insights for implementing DevSecOps in on-premises environments, including analysis of trade-offs and scalability.

6.2 Methodology

6.2.1 Research Design

The study will use a scenario-based experimental methodology to investigate the impact of automated security testing on application reliability, vulnerability detection, and compliance adherence. Four core CI/CD pipeline configurations will be developed and executed:

- Baseline Pipeline (no security integration)
- Static-only Pipeline
- Dynamic-only Pipeline
- Hybrid Pipeline (static + dynamic + policy enforcement)

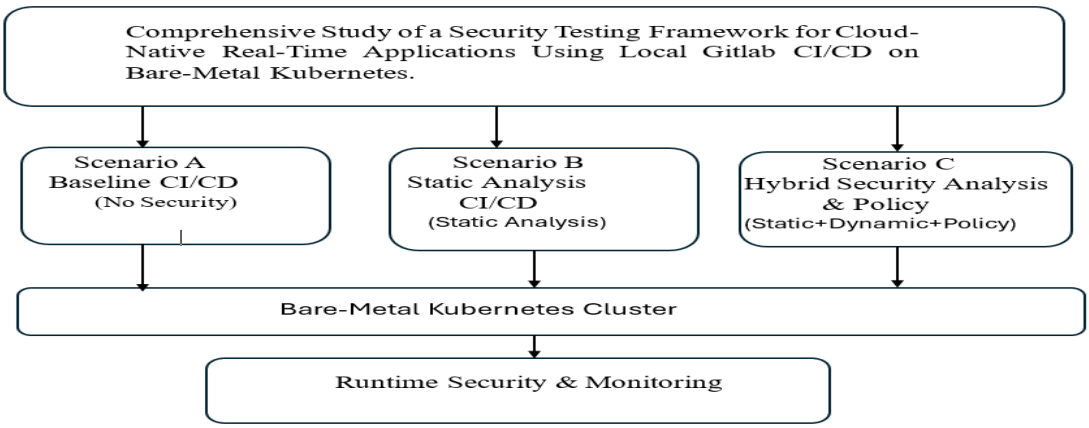

Figure 1. Comparative Pipeline Scenarios

This comparison will provide clear, evidence-based insights into the technical and operational trade-offs between different security integration strategies.

6.2.2 System Architecture Overview

The overall system architecture integrates security controls at every stage of the local GitLab CI/CD pipeline.

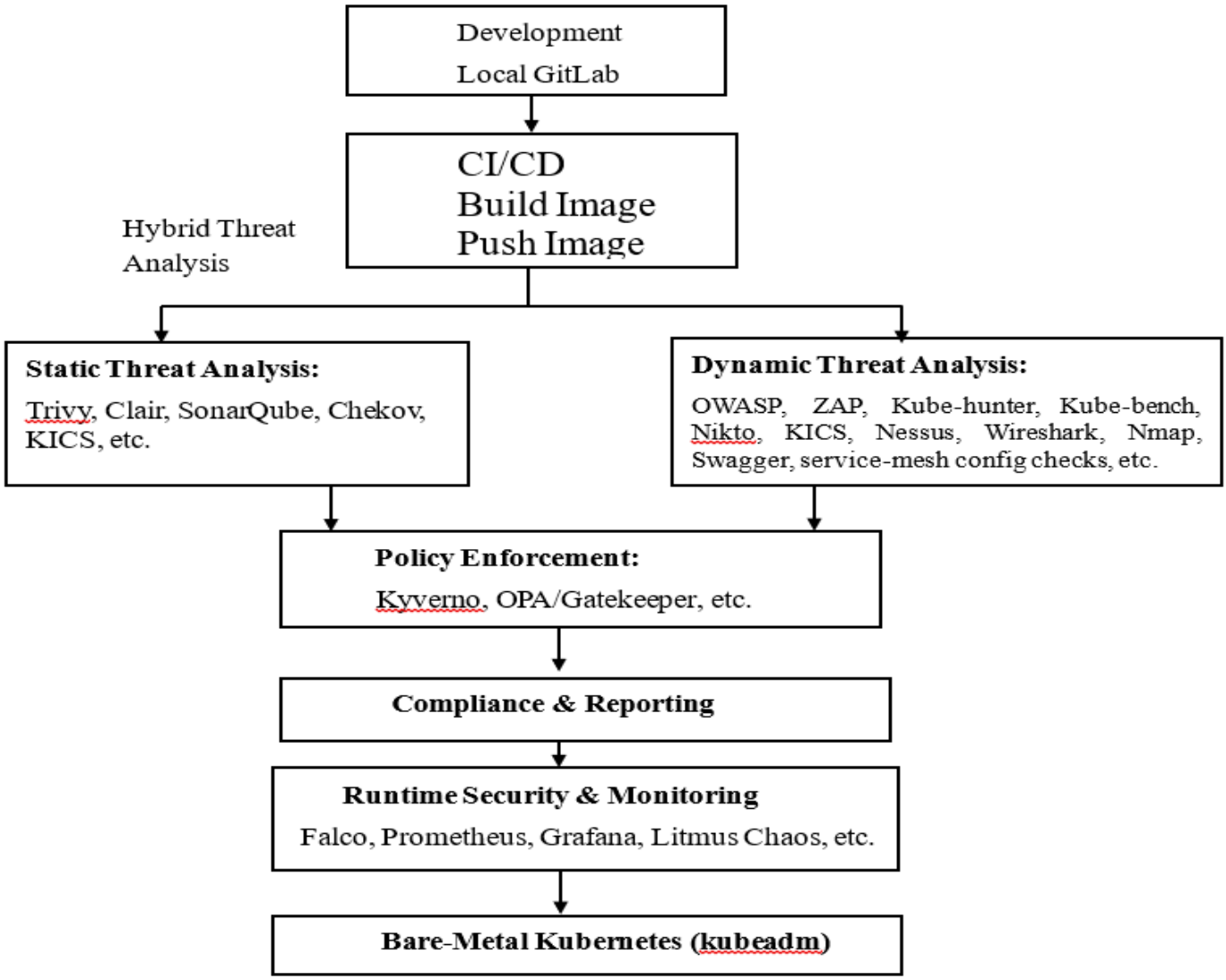

Figure 2. End-to-End Architecture of Local GitLab CI/CD with Integrated Security Stages.

The full pipeline and its test environment will include:

- **GitLab CI/CD (self-hosted):** Central automation and orchestration.
- **Bare-metal Kubernetes cluster:** Realistic, production-like deployment environment.
- **Local container registry:** Secure image storage.
- **Open-source security tools for each stage:**
  - Static analysis: Trivy, Clair, SonarQube, Checkov, KICS.
  - Dynamic analysis: OWASP ZAP, kube-hunter, kube-bench, Nessus, Nikto, nmap, Wireshark, Swagger, Istio-ctl etc.
  - Policy enforcement: Kyverno, OPA.
  - Runtime monitoring: Falco, Prometheus, Grafana, LitmusChaos etc.

All infrastructure and configuration will be managed as code to maximize reproducibility and maintainability.

7. Expected Outcomes

To evaluate the effectiveness and robustness of the proposed security testing framework, four distinct pipeline setups- Baseline, Static-only, Dynamic-only, and Hybrid—will be implemented and tested within a Kubernetes environment at Fraunhofer IPT. The evaluation approach will focus on a comprehensive set of Key Performance Indicators (KPIs) that reflect both security outcomes and operational impact.

All results will be compared across pipeline scenarios to provide a clear, data-driven assessment of how the security framework influences security posture and efficiency. Collected data will be analyzed to ensure that observed differences are significant and actionable.

The outcomes of this research are expected to include:

- A validated security testing framework ready for real-world use in local CI/CD pipelines
- Reliable, practical data on security, compliance, performance, and developer experience to inform future projects
- Detailed guides, automation scripts, and configuration templates for easy adoption and reuse
- Built-in reporting and documentation features that streamline compliance and certification efforts
- Enhanced security awareness and DevSecOps expertise within Fraunhofer IPT, directly supporting the organization's innovation and digitalization goals

By combining rigorous security evaluation with practical deliverables, this thesis will provide actionable, reusable results for real-world projects at Fraunhofer IPT and contribute new empirical knowledge to the field of cloud-native security and DevSecOps.

8. Conclusion

This thesis is not just a technical exercise; it's a real-world research project embedded within Fraunhofer IPT and the Factory Cloud initiative. The goal is to deliver measurable insights and reusable solutions for secure, efficient microservice deployment in industrial environments, all while balancing performance and developer experience.

A central outcome of this thesis will be the development and documentation of a comprehensive security testing framework for local CI/CD pipelines and microservices. This framework will integrate tools for static and dynamic code analysis, vulnerability scanning, image signing, and real-time threat detection providing a practical, step-by-step guide that can be applied directly in the Factory Cloud and beyond.

The results will help Fraunhofer IPT build a stronger, more resilient Factory Cloud platform and will provide the security testing framework that can be reused, adapted, and extended for future projects, while also advancing scientific understanding of DevSecOps in real-time, on-premises industrial environments.