Policy-Driven Syscall-Level Runtime Mitigation for Ephemeral Serverless Workloads on Kubernetes

Pedro Escaleira escaleira@ua.pt
Carlos Ferreira carlos.mig@ua.pt
Vitor A. Cunha vitorcunha@ua.pt
Rui L. Aguiar ruilaa@ua.pt

universidade de aveiro

30 YEARS instituto de telecomunicações

ATNOG

EURO-3C

Problem Statement

FADO Detects Intrusions from Syscall Analysis

Previous work: Function-level Anomaly Detection Observer (FADO)

eBPF captures host-level syscall activity without changing tenant code.



Each HMM learns benign syscall sequences for its function and flags deviations.

However, the original FADO framework only detected intrusions

It did not provide a lifecycle-aware response control plane

The Gap Was Operational

This work reuses FADO's detector and adds the missing **runtime control plane**

FADO	This Work
Manual target attachment	**Kubernetes** lifecycle discovery
Single-container pipeline	**Per-Pod** inference isolation
Classifier output only	**Policy-directed** local action

Proposal

Lifecycle-Aware Attachment

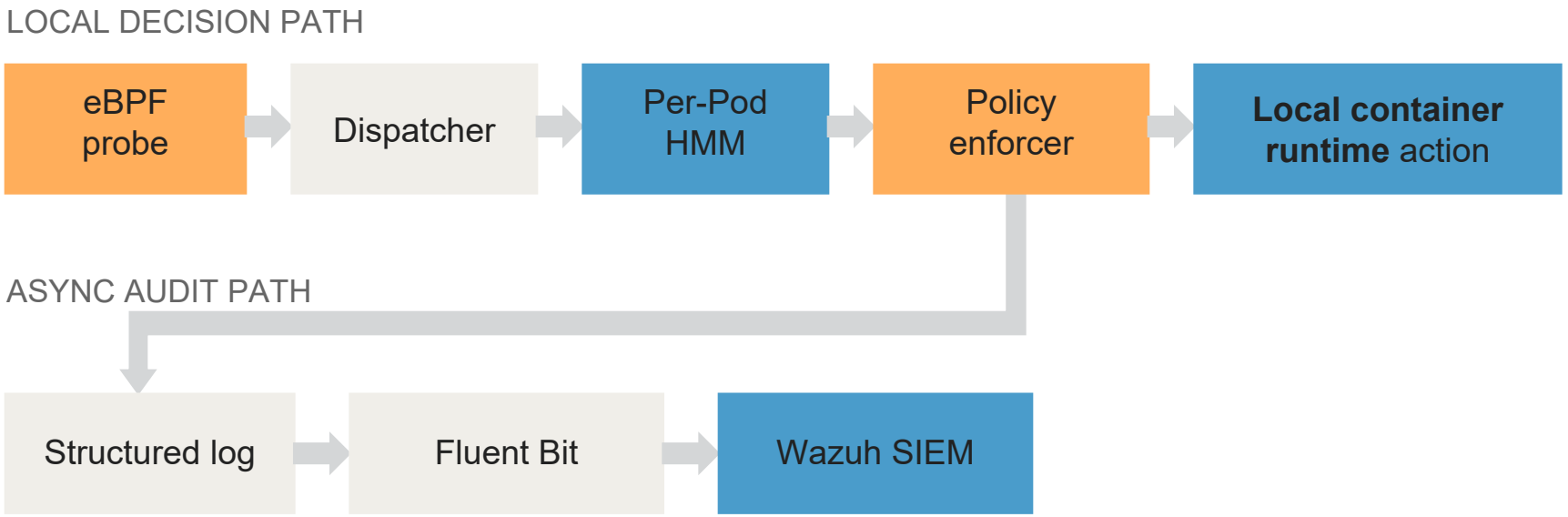
Node agents discover new function Pods and attach monitoring automatically



Host-side tracing is transparent to tenant code

Local Action, Async Audit

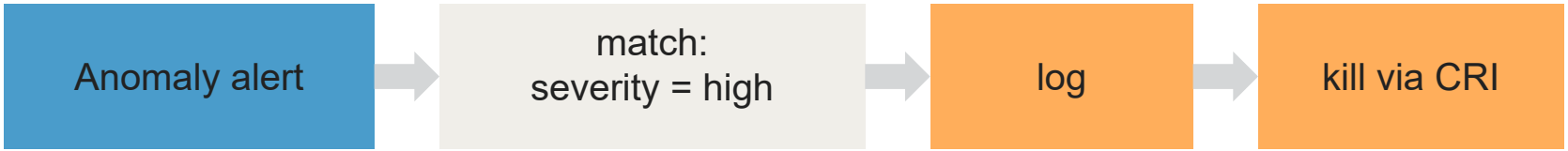
The architecture separates the response path from external logging



SIEM delivery is not a prerequisite for local action

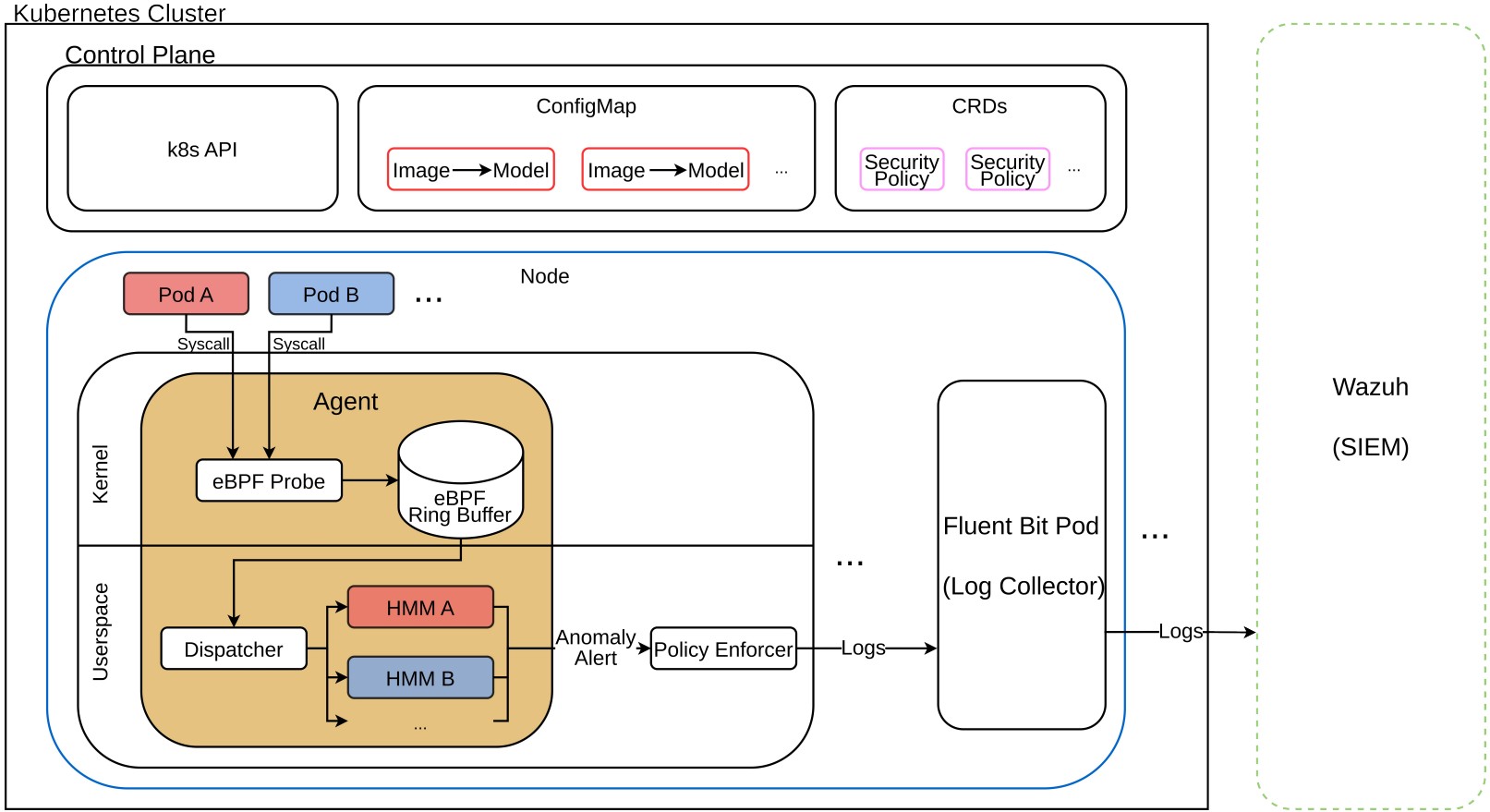
Severity-Driven Policies

A Kubernetes SecurityPolicy maps alert severity to local actions



Current prototype: match only alert_severity; actions are **nothing**, **log**, and **kill**

Architecture





Evaluation

Testbed

KUBERNETES

3 VMs

1 control plane
2 workers

STACK

K8s 1.35.2

Knative 1.21
Ubuntu 24.04.4

MONITORING

4th VM

Wazuh outside cluster

200 independent runs against a exploited Knative function

eBPF Attachment Latency

Mean time from container start until monitoring is attached



TOTAL ATTACHMENT

1162.41 ± 25.43 ms

92.76% of the mean is Kubernetes propagation
7.24% is agent work

Detection Costs More

FADO · SINGLE CONTAINER

32.46 ms

± 1.82 ms

NODE AGENT · PER-POD

152.77 ms

± 10.30 ms

4.71× the FADO baseline

Added dispatcher queues and per-Pod process isolation are part of the trade-off

Alert Latency

After detection, the Policy Enforcer will emit an alert

159.87 ± 10.29 ms

mean policy-ready alert

7.09 ms

mean detection-to-alert interval

Wazuh Is Asynchronous

Manager-side alert time includes both the local path and external log delivery

661.14 ± 24.83 ms

from benchmark timestamp to matching Wazuh alerts.json entry

About 160 ms is the agent path (~24%)

Local container runtime actions do not wait for SIEM ingestion

Takeaway

FADO supplies the detector

This work adds the runtime control plane

Future work will try to improve performance, mainly for **startup agent attachment**

Thank you for your attention



github.com/ATNoG/fado