CYBERAID

AI-Driven Cybersecurity for Financial Service Providers

Module 4

Integration with the CyberAId Architecture and Agents

CyberAId Learning Centre – eBPF Tutorial Course

October 2026

Co-funded by the European Union

ECCC EUROPEAN CYBERSECURITY COMPETENCE CENTRE

The project funded under Grant Agreement No 101249596 is supported by the European Cybersecurity Competence Centre

CyberAId architecture recap

Presentation + HITL

Dashboards, operations CLI, Learning Centre

LLM Orchestration

Coordinator: routes requests and alerts, human checkpoints

Agentic Analysis

Specialist agents: reasoning and controlled tool use

Integration + Messaging

Event and Alert Bus, log collector, access control, Tool Adapter Layer

Knowledge + Data Fabric

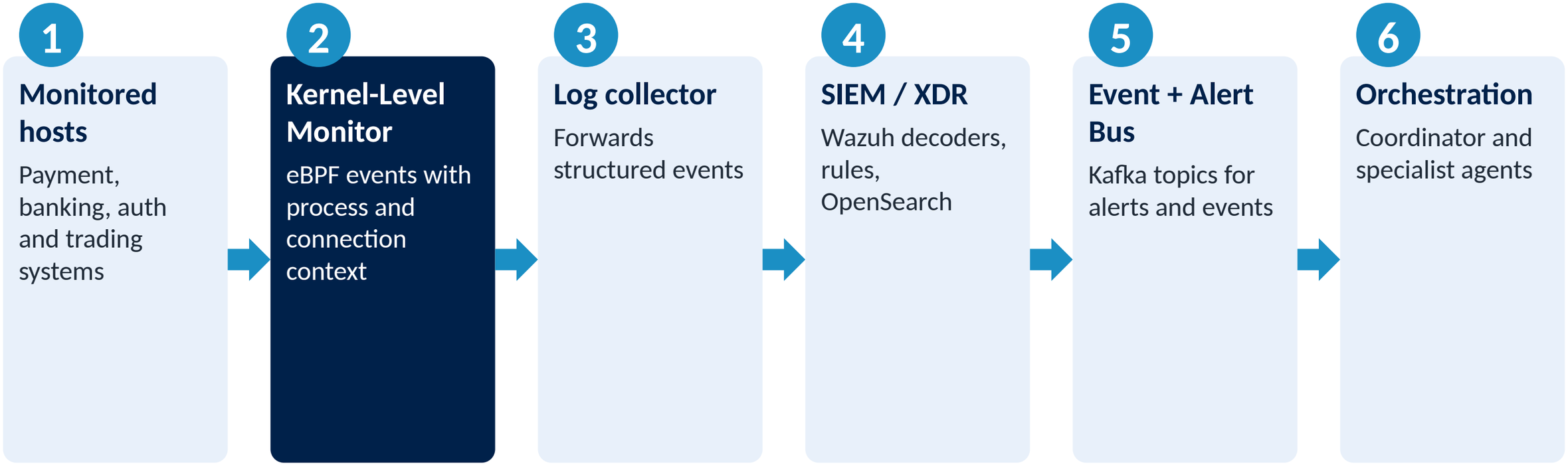
Security knowledge model, retrieval store, fraud graph, event store

Security Capability Services

SIEM/XDR, Kernel-Level Monitor, graph analytics, digital twin, risk engine, forensics

Based on the CyberAId reference architecture (D2.2) and orchestration design (D2.5).

Where the monitor sits in the data path



Simplified from D2.2 and D2.5; exact connector topology is fixed in the integration specification.

Two paths to the orchestration layer

Path A: via the SIEM

Events decoded and graded in Wazuh; high-priority alerts forwarded to the bus by a dedicated connector

Path B: direct publication

The monitor publishes kernel-level alerts as a source on the bus, next to SIEM and scanners

Parallel event forwarding

Event data also flows for contextual processing, correlation and forensics

Same principle

A detector publishes to a topic; the orchestration layer subscribes

The choice of path is to be confirmed by the alerting and correlation tasks (T4.4, T4.5).

Alignment with the Knowledge Model

1
Observation
Raw kernel facts: process, connection, file open

2
Detection
Rule or analysis linking observations; ATT&CK technique

3
Finding
Validated agent record with evidence references

4
Incident
Confirmed situation with assets, impact and response

Observation, detection, finding and incident layers follow the Security Knowledge Model (D2.4) and the finding contract (D2.5).

Agent integration: three patterns (proposed)

Pattern 1: alert producer

Kernel alerts trigger the log_correlation workflow

Owned by the Log Analysis and Event Correlation agent

No new agent: first milestone

Pattern 2: tool for agents

Read-only queries in the Tool Adapter Layer

Used on demand by correlation, anomaly and forensics

Governed, validated and audited

Pattern 3: host-triage agent

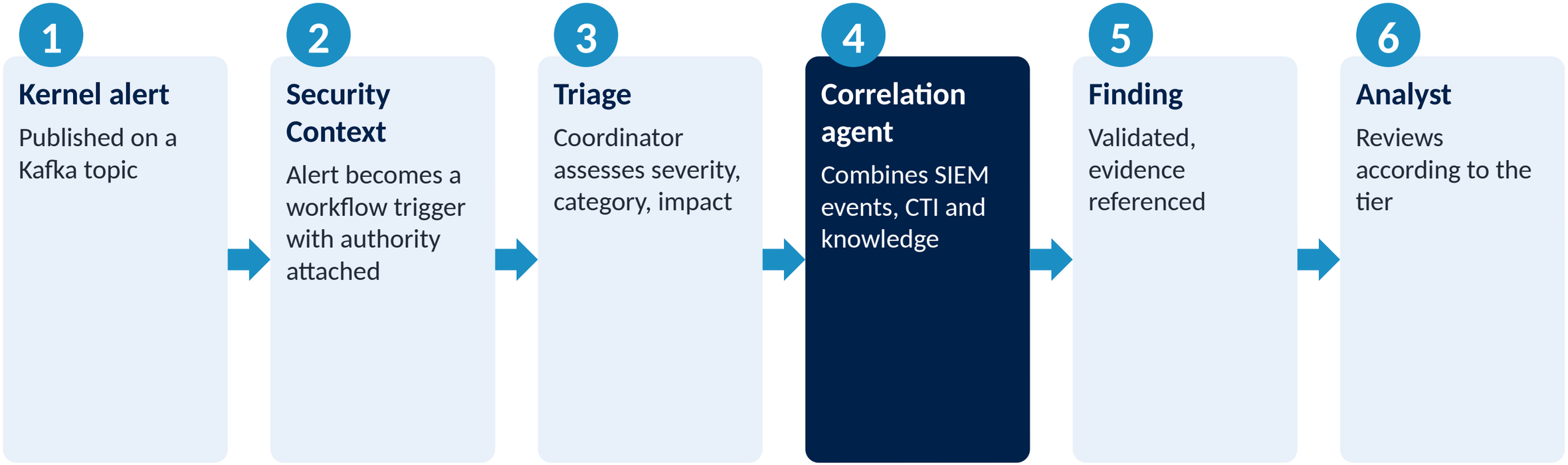
Optional service-backed wrapper around the monitor

Adds scope, evidence references, severity

Emits validated findings

Proposed patterns for discussion with the agent and runtime task teams; D2.5 leaves specialist internals to each task.

Pattern 1: alert-driven workflow



Based on D2.5 (alerting bridge, entry points, workflow catalogue); the correlation agent is delivered by T4.4.

Pattern 2: the monitor as a governed tool

Adapter types

Native typed (preferred)

Command-line wrapper

MCP (external boundary)

Every call

Check permitted tool set

Validate arguments, execute

Audit, including rejections

```
{ "tool": "ebpf.process_tree",
  "adapter": "native",
  "mode": "read-only",
  "args": {
    "host": "string",
    "pid": "integer",
    "window_s": "integer" },
  "requires": "allowed_tools",
  "audited": true }
```

Illustrative manifest only; names and fields are placeholders, not an agreed specification.

Governed actions and human-in-the-loop tiers

Tier	Illustrative eBPF-related action	Oversight
Tier 1	Enrichment: process tree, connection history, summarisation	Autonomous
Tier 2	Reversible containment: isolate a pod, temporarily block a destination	Analyst countersignature
Tier 3	Enforcement with side effects: BPF-LSM block or XDP drop on payment systems	Named approval with justification

Tiers follow D2.5; actions travel downward through adapters, never over the bus. Examples are illustrative.

The finding contract for kernel evidence

Referenced, not copied

Host, PID, event identifiers and time range point back to the source

Metadata only

No payloads, transaction content or secrets in findings

Claims must re-ground

The narrative must trace to the evidence and retrieval references

Lifecycle

Open, acknowledged, accepted, rejected, corrected; feedback improves later analyses

Agent collaboration around kernel telemetry

Agent or capability	Proposed exchange with kernel telemetry
Log Analysis and Event Correlation (T4.4)	Kernel alerts and events as correlation inputs
Anomaly Detection (T4.3)	Process and connection baselines per service
Digital Forensics (T5.4)	Timeline reconstruction from retained events
Risk and Compliance (T5.2)	Runtime evidence for control gaps under DORA and NIS2
Reporting (T5.1)	Incident narratives and regulatory notifications from findings

Proposed integration points, to be confirmed in the integration specification.

Worked example: reverse shell, end to end

1	**Observe**	execve of a shell and an outbound connect in one process tree
2	**Alert**	Rule grades CRITICAL, maps T1059/T1071; alert reaches the bus
3	**Triage**	Coordinator activates the correlation agent with a Security Context
4	**Investigate**	Agent queries the process tree; checks the destination against CTI
5	**Decide**	Finding with evidence; containment paused for approval by tier
6	**Report**	Analyst decides; Reporting agent drafts the incident narrative

Key takeaways

- The monitor is a deterministic capability; agents reach it through the integration fabric
- Telemetry flows via the SIEM or directly over the Event and Alert Bus
- Three proposed patterns: alert producer, governed tool, optional host-triage agent
- Same governance for all: Security Context, validated tool calls, audit, tiered oversight
- Findings reference evidence and carry metadata only



Thank you!

Connect With CyberAID:
LinkedIn: CyberAID

Next: Module 5 – Operations, Privacy and Compliance

CyberAID has received funding from the DIGITAL EUROPE Programme under grant agreement No. 101249596