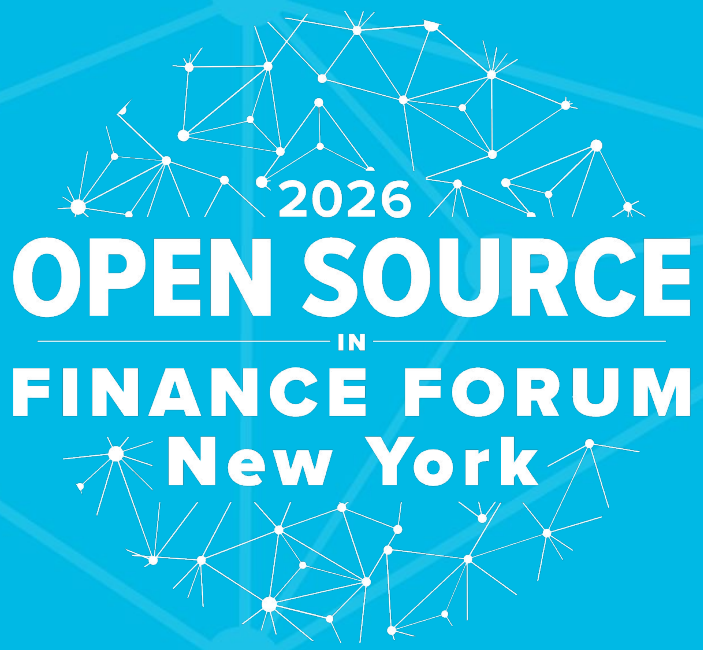


Operationalizing Security for AI Systems in Financial Services

Monica Mock-Sipos



MONICA MOCK-SIPOS

- Worked in Financial Services for over a decade
- Lead Red Hat's OpenShift specialist org for North American Financial Services
- Specialize in hybrid operations, security and control-plane design, and getting agentic workloads onto Kubernetes without breaking governance, audit, or operational standards.
- Background in cybersecurity, enterprise AI architecture, data, integration, and software development
- Active CISSP

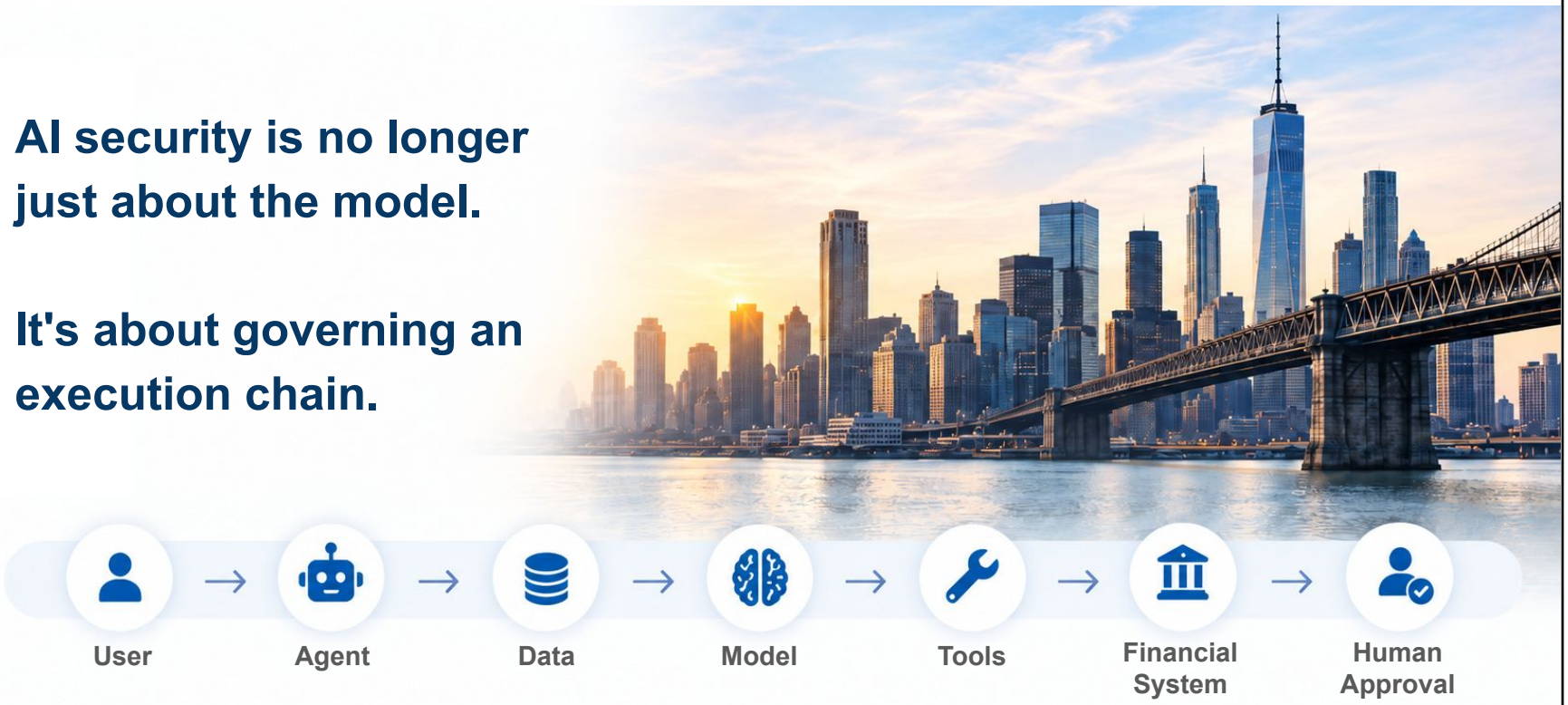
LinkedIn
Medium

Operationalizing Security for FSI Organizations

From AI governance principles to runtime control

AI security is no longer just about the model.

It's about governing an execution chain.



The Security Boundary Has Moved

The model is not the security boundary. The execution environment is.

Static Applications

Fixed boundaries and predictable execution.



- Known identity and access at login.
- Policy applied at deployment and admission.
- Security controls around a stable, bounded application.

Distributed Execution

Multi-step, agentic workflows across services, tools, and data.



- Execution spans multiple systems and security domains.
- Identity, authority, and policy must travel with the workflow.
- Trust conditions can change during execution.

Security Implication

Security must govern execution continuously, not just at the start.



- Decisions must be reevaluated as context and risk change.
- Workflows must be governable and replayable end to end.
- The security boundary now follows the execution, not the application.

The core challenge: keep trust, policy, and control consistent across a distributed workflow where context and risk can change at any point.

One FSI Workflow, Many Trust Transitions

Agent-assisted suspicious payment investigation

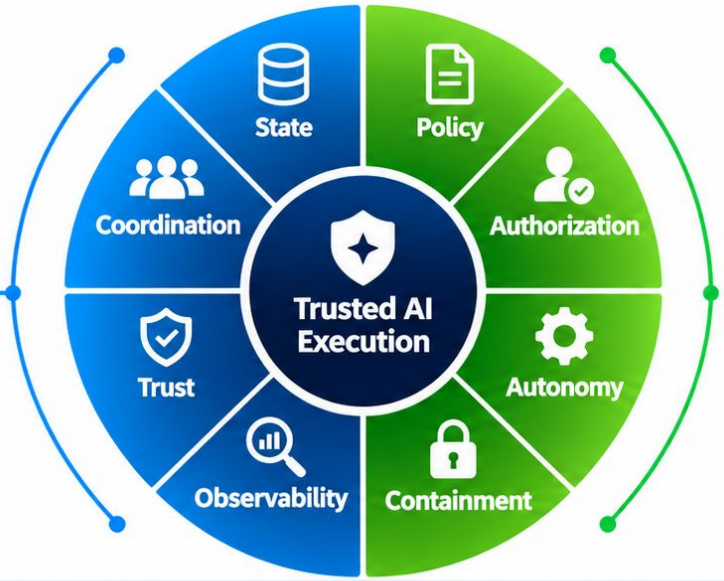


Enterprise AI Operational Model

Eight operational responsibilities for trusted AI execution.

Preserve What Execution Needs

Keep execution coherent, traceable, and understandable.



Govern What Execution May Do

Establish boundaries, rights, and control.

Two sets of responsibilities preserve trust and govern action across the execution path.

A complete, open-source operational model that preserves execution, governs behavior, and makes AI workflows secure, auditable, and replayable.

Open Source Makes This Operational

Illustrative architecture using open-source projects aligned to the Enterprise AI Operational Model.

Illustrative reference map — not an implementation sequence.

FSI Governance & Architecture

FINOS AIGF
Governance risks and controls for AI in financial services

FINOS CALM
Architecture and control relationships for AI systems

Preserve What Execution Needs
Enable trusted execution.

Govern What Execution May Do
Constrain and control execution.

Operational Responsibility	Purpose	Selected Open Source Projects
State	Preserve execution context	PostgreSQL • Redis • Apache Kafka
Coordination	Advance and recover execution	Argo Workflows • Temporal • Kubernetes Jobs
Trust	Carry identity and provenance	SPIFFE/SPIRE • Sigstore • in-toto
Observability	Preserve operational evidence	OpenTelemetry • Jaeger • Prometheus
Policy	Preserve organizational intent	Open Policy Agent • Kyverno
Authorization	Govern execution rights	OpenFGA • Keycloak
Autonomy	Bound independent action	LangGraph • Model Context Protocol (MCP) • Human approval gates
Containment & Runtime Boundary	Constrain execution and blast radius	OpenShell (agent sandbox) • Cilium • Falco • Kata Containers
Platform Foundation	Provide the execution substrate	Linux • Containers • Kubernetes • Service Mesh • Hybrid Infrastructure

Trusted Outcomes for FSI

Secure
Identity, policy, and runtime controls remain connected.

Auditable
Decisions, actions, and context can be reconstructed.

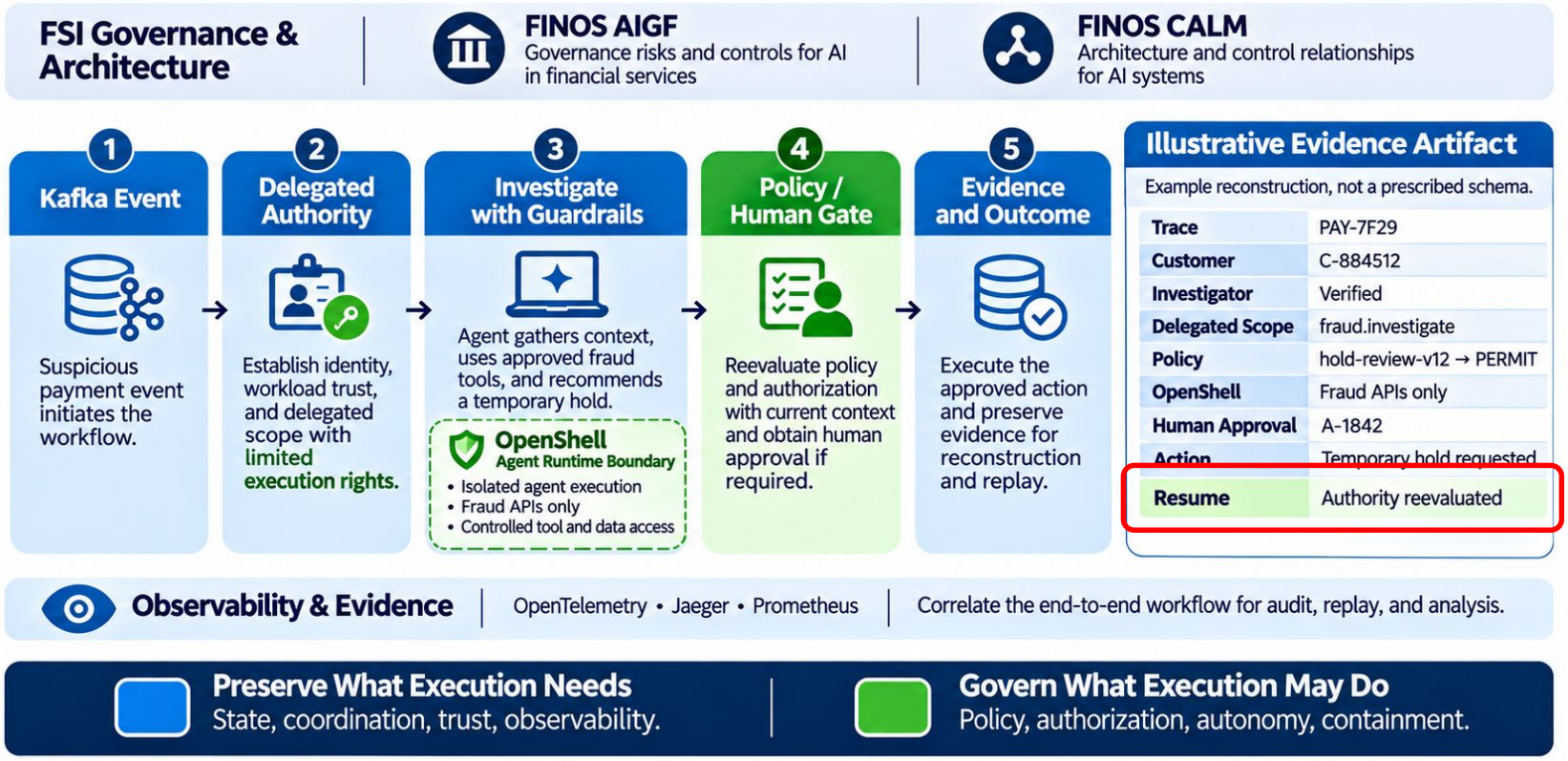
Replayable
Workflows can be validated, reviewed, and explained.

Bounded
Autonomy and failures remain constrained.

No single project provides operational trust.
Trust emerges when identity, evidence, policy, authorization, execution, and containment remain connected.

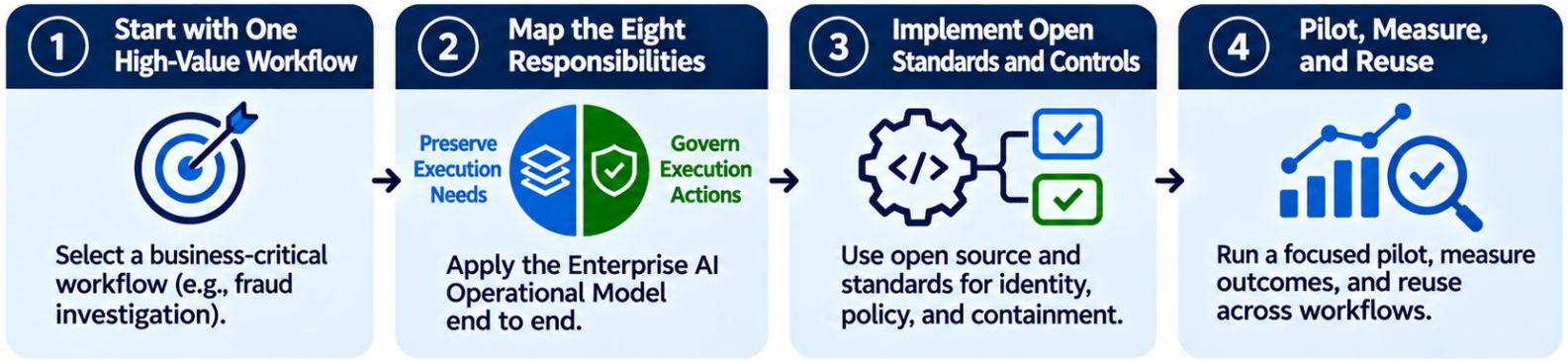
A Secure, Auditable, and Replayable Agentic Workflow

Agent-assisted suspicious payment investigation using open standards and open-source building blocks.



Operationalizing Secure AI for FSI

Four practical steps to turn the Enterprise AI Operational Model into action.



Key Takeaways

- **Build security into workflows.**
- **Use open standards and open source.**
- **Design for evidence, approvals, and containment.**

FINOS Community Ask

Advance practical security for AI in FSI through two initiatives:

- **FINOS AIGF** — Governance risks and controls.
- **FINOS CALM** — Architecture and control relationships.

Additional Resources

Notes, standards, and open-source references behind the Enterprise AI Operational Model.

FSI Governance & Architecture

- FINOS CALM: Architecture as code
- FINOS AI Governance Framework (AIGF): FSI AI risks and controls
- Enterprise AI Operational Architecture Model: Article series

Identity, Policy & Operational Trust

- SPIFFE / SPIRE: Workload identity
- Open Policy Agent / Kyverno: Policy evaluation and enforcement
- OpenFGA / Keycloak: Authorization and identity-aware access

Agent Execution, Autonomy & Containment

- OpenShell: Open-source agent sandbox/runtime boundary
- Model Context Protocol (MCP) / LangGraph: Tool integration and agent workflow
- Kata Containers / Cilium / Falco: Isolation, network controls, runtime detection

State, Coordination, Provenance & Evidence

- PostgreSQL / Redis / Apache Kafka: State and eventing
- Argo Workflows / Temporal / Kubernetes: Coordination and orchestration
- OpenTelemetry / Jaeger / Prometheus: Telemetry and operational evidence
- Sigstore / in-toto: Artifact provenance and supply-chain evidence

Use these **open-source building blocks** as a reference for secure, governable, and replayable AI workflows.

Thank You!



QR Code for Session Feedback