QUANTERIOS WHITEPAPER

quanterios
cryptographic and ai security and governance

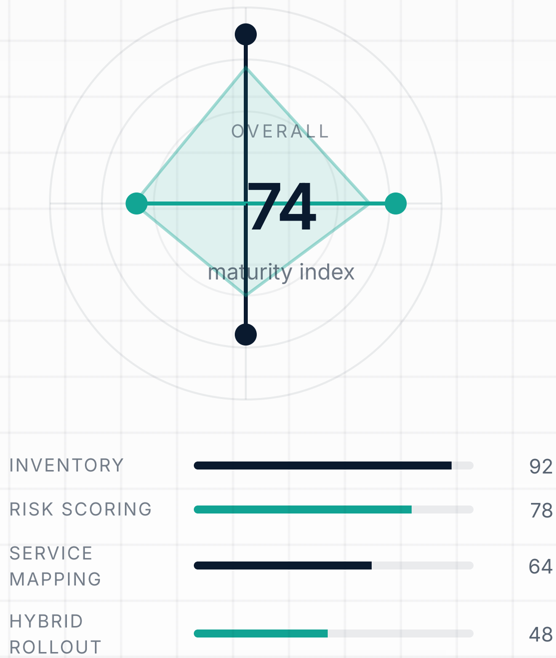
EDITION
Knowledge paper 02
02 May 2026

OFFICIAL PUBLICATION | WHITEPAPER SERIES

AI Runtime Protection for Agentic Systems

A practical control model for prompt injection, tool abuse, output validation, and human approval in live AI workflows.

PQC READINESS SYSTEM



Runtime MCP 4

Cookie preferences

Set how Quanterios handles non-essential storage and measurement.

Necessary storage supports security, language, and core site operation. Analytics and any future marketing technologies stay off until you allow them.

Review cookie settings

EDITORIAL BRIEF

Executive summary

AUDIENCE

AI Security Lead, Platform Engineer, Model Governance Lead, SOC Architect

FORMAT

Quanterios Whitepaper

LENGTH

16 pages

Agentic systems expand risk beyond model quality. Once models can invoke tools, access MCP servers, trigger workflows, and interact with customer or operational data, the security boundary shifts to runtime.

This paper explains the minimum runtime controls required for production AI systems, especially in regulated environments where governance must be visible in operation and not only in policy documents.

ISSUE MAP

Contents

WHY THIS PAPER MATTERS

Quanterios whitepapers are designed to be carried into investor meetings, buyer reviews, partner conversations, and internal strategy sessions as serious editorial assets, not marketing one-pagers.

SECTION 01

Why runtime is the real control boundary

WHY URGENCY IS COMPOUNDING

Research

Planning

Procurement

Modernization

Inventories and model cards matter, but they do not stop a system from taking the wrong action at the wrong time. Runtime is where prompts, tool calls, approvals, identities, and policies actually meet operational reality.

As systems become more agentic, static design-time reviews become less sufficient. Teams need continuous control over what the system is allowed to request, read, infer, write, or trigger.

SECTION 02

The four runtime checks

Effective runtime protection usually combines four checks: prompt evaluation, output evaluation, action validation, and approval logic. Each catches a different class of failure and none is enough alone.

Prompt checks look for injection attempts or scope shifts. Output checks prevent unsafe or non-compliant responses. Action validation ensures tool invocations and side effects remain within policy. Approval logic enforces human decision points where autonomy must

OPERATING MODEL

OPERATING CYCLE

Discover to evidence

01 Discover cryptography everywhere

02 Score posture and exposure

03 Prioritize by service consequence

04 Roll out hybrid migration waves

05 Prove progress with live evidence

PRACTICAL IMPLICATIONS

- Prompt checks detect intent-shaping attempts and hostile instruction layering.
- Output checks prevent data leakage,

SECTION 02

The four runtime checks, continued

ADDITIONAL CONSIDERATIONS

- Human approval gates keep high-risk workflows reviewable and auditable.

SECTION 03

MCP and tool-connected systems

WHERE MIGRATIONS STALL

Late discovery
Hidden certificates, appliances, and supplier code

Fragmented ownership
Security, delivery, infra, and compliance see different records

No service map
Findings are not tied to consequence or change windows

No exception logic
Temporary tolerances cannot be reviewed or expired cleanly

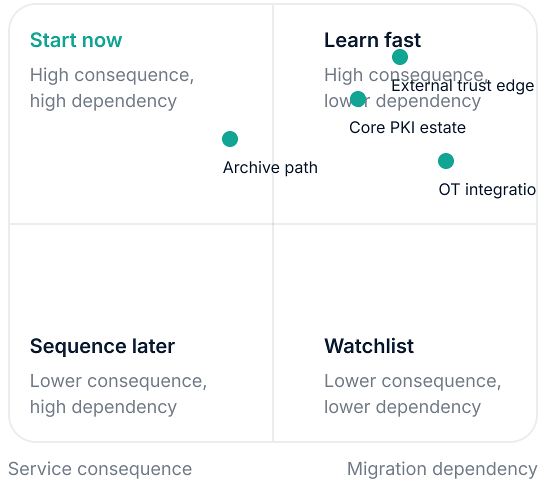
MCP-connected systems widen the blast radius because they standardize how models and agents reach tools and data sources. That interoperability is useful, but it also means an error can travel farther unless scope, identity, and approval rules are explicit.

Teams should treat MCP-connected access the same way they would treat privileged integration middleware: everything should be identity-aware, policy-bound, and evidence-producing.

SECTION 04

Production evidence for AI assurance

FIRST-WAVE PRIORITIZATION



Runtime protection becomes more valuable when it produces evidence that governance and compliance teams can actually use. Logs alone are not enough. Evidence should show what policy existed, what event occurred, how the system responded, and who approved exceptions.

That evidence path is what lets security teams, platform teams, and assurance teams speak from the same operating record.

CLOSING SPREAD

From insight to action

KEY TAKEAWAYS

- Runtime controls, not policy documents alone, define the safety boundary of agentic AI.
- Tool-connected systems need identity, scope, and approval logic as first-class controls.
- Good runtime protection should generate evidence useful to security and assurance teams alike.

WHAT STRONG TEAMS DO NEXT

NOW
Make the estate visible
Create one living control record across assets, services, owners, and deadlines.

NEXT
Run a first-wave queue
Move the highest-consequence services into a controlled hybrid rollout plan.

THEN
Prove progress
Review migration status and exceptions as an operating program, not a slide deck.

NEXT STEP

Need a production runtime model for regulated AI?

Use this whitepaper as the decision framework, then let Quanterios map it to your real agent and tooling estate.

Review AI runtime controls

Cookie preferences

Set how Quanterios handles non-essential storage and measurement.

Necessary storage supports security, language, and core site operation. Analytics and any future marketing technologies stay off until you allow them.

Review cookie settings