MI9: An Integrated Runtime Governance Framework for Agentic AI

Charles L. Wang[1,2], **Trisha Singhal**[1], **Ameya Kelkar**[1], **Jason Tuo**[1]

[1]Barclays, Model Risk Management
[2]Columbia University

Abstract

Agentic AI systems capable of reasoning, planning, and acting present governance challenges that differ fundamentally from conventional models. Because these systems can exhibit emergent behaviors during execution, many risks cannot be fully anticipated pre-deployment. We present MI9, an integrated framework for runtime safety of agentic AI, where safety properties are enforced over live behavior sequences. MI9 provides six coordinated mechanisms: Agency-Risk Index, agent-semantic telemetry, goal-aware authorization monitoring, finite-state conformance engines, goal-conditioned drift detection, and graded containment. MI9 acts as a framework layer that instruments and governs existing systems to enable systematic oversight. In evaluations over 1,000 diverse multi-domain synthetic scenarios, MI9 achieves high detection rates with low false positives compared to standard observability baselines. By shifting the locus of assurance to runtime safety, MI9 establishes a validated architectural foundation for the operational oversight of agentic AI. We open-source all prompts, scripts, and per-scenario summaries for reproducibility.

Introduction

As large language models (LLMs) increasingly evolve into agentic systems, they introduce governance challenges that emerge only during runtime. Unlike traditional AI, these systems plan, revise goals, recall memory, and coordinate tool use—blurring the line between inference and autonomous action. The most critical alignment risks—recursive planning loops, goal drift, cascading tool chains—arise dynamically and elude pre-deployment control methods. MI9 addresses this gap by enabling real-time oversight and intervention at key decision boundaries. In doing so, it provides the runtime infrastructure needed to support core alignment goals: corrigibility, safe delegation, and behavioral oversight in deployed agentic systems.

Alignment research has primarily focused on training-time interventions: Reinforcement Learning from Human Feedback (RLHF) (Christiano et al. 2017; Ouyang et al. 2022) and Constitutional AI (Bai et al. 2022) establish initial value alignment but cannot address failures emerging during autonomous operation when agents encounter novel situations or coordinate with other systems (Kenton et al. 2021).

Building on this foundation, recent work has mapped agentic system taxonomies (Schneider 2025; Kasirzadeh and Gabriel 2025), governance frameworks (Fang and colleagues 2024; Raza et al. 2025; Engin and Hand 2025; Kolt 2025), and threat models (Narajala and Narayan 2025; Chan et al. 2025; Syros et al. 2025). However, leading benchmarks prioritize task completion over governance dimensions such as behavioral consistency (Kapoor et al. 2024; Liu et al. 2025; Zhou et al. 2023; Jimenez et al. 2023; Sumers et al. 2025).

Meanwhile, current monitoring solutions (Wu et al. 2024; Langfuse Team 2024; LangChain 2024; Weights & Biases 2024; Datadog 2024) provide reactive observation rather than proactive intervention. Similarly, process observability research (Fournier, Limonad, and David 2025) and visibility frameworks (Chan et al. 2024) focus on observation, while enterprise platforms (Holistic AI 2024; Monitaur 2024; ModelOp 2024) and security frameworks (OWASP 2024; NIST 2024) rely on static risk assessment inadequate for emergent runtime behaviors.

Consequently, existing approaches suffer from several critical gaps: inability to intervene during concerning behaviors, lack of agent-semantic telemetry capturing governance-relevant decisions, static guardrails unable to adapt to emergent behaviors, and insufficient multi-agent oversight.

MI9 Framework

Threat Model & Scope

1. **In scope.** Runtime risks from agent *behavioral sequences* and coordination: (i) goal drift under fixed stated goals, (ii) policy-skipping tool chains, (iii) delegated privilege escalation, (iv) multi-agent coordination failures.
2. **Out of scope.** Pretraining/data harms, upstream supply-chain compromise, and non-sequential issues not captured in event traces.
3. **Actors.** Deployed agents (incl. subagents), human overseers, organizational policy engine.
4. **Assumptions.** Minimum ATS coverage at least for action-level events; bounded event reordering; ability to pause/contain.
5. **Objective.** Minimize *undetected violations at very low*

Table 1: MI9 Runtime Governance Framework Components

Component	Purpose	Governance Capabilities
Agency-Risk Index	Risk-calibrated governance tier assignment	Quantifies agent autonomy, adaptability, and continuity to scale oversight intensity proportionally to assessed risk
Agentic Telemetry Schema	Agent-semantic event capture	Monitors cognitive, action, and coordination events to provide governance-relevant behavioral visibility
Continuous Authorization	Dynamic permission management	Context-aware access control based on agent state to prevent privilege escalation during goal evolution
Conformance Engine	Temporal policy enforcement	FSM-based sequence pattern matching to detect policy violations across multi-step workflows
Drift Detection	Behavioral anomaly identification	Goal-conditioned baseline comparison to distinguish concerning drift from legitimate adaptation
Graduated Containment	Agent-aware intervention strategies	Four-level containment hierarchy to preserve operational value while preventing harm

Table 2: Comparison of governance framework coverage for agentic systems (● = fully supported; ○ = partial; × = unaddressed)

Runtime Governance Capability	AgentOps	LangFuse	GAF-Guard	SAGA	MI9 (ours)
Real-time behavioral intervention	○	×	○	×	●
Agent-semantic behavioral monitoring	○	○	○	×	●
Dynamic policy enforcement	×	×	○	○	●
Multi-agent coordination governance	○	×	×	○	●

FPR while preserving operational continuity via graduated containment.

Framework Integration and Overview

The MI9 framework coordinates six specialized components to provide unified runtime oversight across agentic AI deployments. Unlike existing approaches that address governance concerns in isolation, MI9 integrates telemetry capture, authorization monitoring, conformance checking, drift detection, and containment execution within a single architectural framework.

The Agency-Risk Index (ARI) calibrates governance intensity across agent populations, while the runtime toolkit delivers coordinated oversight: ATS captures agent-semantic events enabling policy evaluation; continuous authorization dynamically adjusts permissions based on behavioral context; conformance engines enforce temporal behavioral patterns; drift detection identifies goal-conditioned behavioral deviations; and graduated containment executes agent-aware interventions preserving operational continuity. After being standardized by a framework-specific adapter, a central processor uses a Subscription Registry to distribute each event to any and all Monitoring Modules that have subscribed to it for evaluation.

This integrated architecture enables proactive, real-time oversight specifically designed for agentic systems exhibiting emergent behaviors during execution, addressing the fundamental gap between static pre-deployment assessments and reactive post-incident analysis. Production deployments require standard distributed systems coordination (Fowler 2005; Bailis et al. 2014), but the core governance semantics operate independently of the underlying consistency mechanisms.

We emphasize that **MI9 is intended as a governance layer framework for generalizable runtime governance, not as a single-system deployment.** Rather than targeting a specific agent framework, MI9 defines an infrastructure-agnostic runtime governance architecture intended for broad institutional adoption across heterogeneous agent ecosystems. [1]

Agency-Risk Index

To calibrate governance intensity across diverse agent architectures, we introduce the **Agency-Risk Index (ARI)**, which quantifies governance requirements through three orthogonal dimensions:

1. **Autonomy** (independent multi-step task execution),
2. **Adaptability** (behavioral modification based on feedback),

[1] See appendix for the comprehensive SDK-integration table.

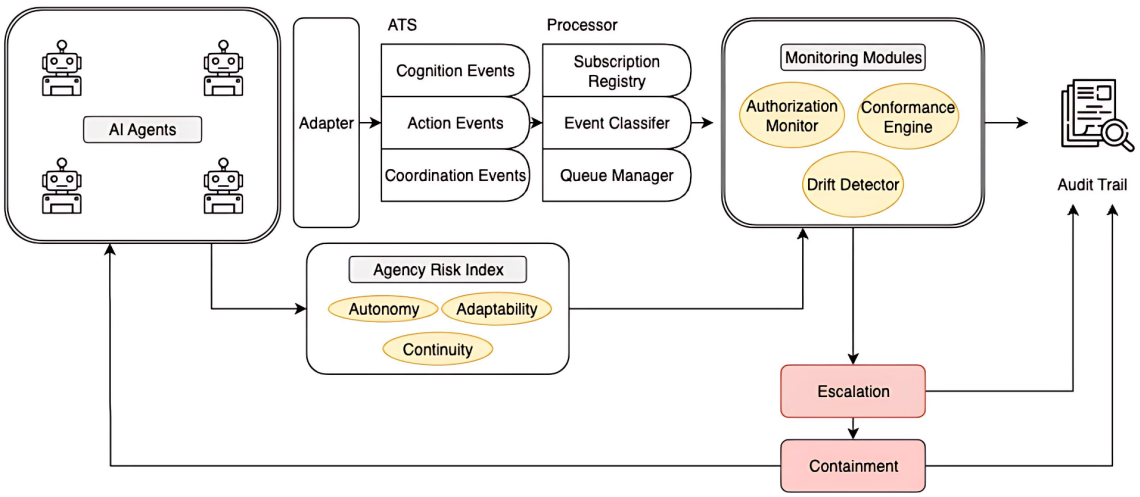

Figure 1: MI9 Framework Pipeline

3. **Continuity** (persistent state maintenance across sessions).

Each dimension comprises four measurable capabilities scored 0-3, yielding a composite index where each dimension contributes equally:

$$\text{ARI} = \frac{1}{3}\sum_{d=1}^{3}\left(\frac{1}{12}\sum_{c=1}^{4} s_{d,c}\right)$$

where $s_{d,c}$ represents the score for criterion c within dimension d, producing normalized values ARI $\in [0, 1]$.

This formulation ensures equal dimensional weighting because each dimension of agency poses fundamentally distinct governance challenges that require different oversight mechanisms. Autonomy drives the need for authorization controls, adaptability necessitates behavioral monitoring, and continuity requires state persistence oversight. By weighting dimensions equally rather than individual criteria, we acknowledge that a highly autonomous but non-adaptive agent poses qualitatively different risks than a highly adaptive but stateless system, ensuring our governance framework scales appropriately across diverse agent architectures.

Implementation Note. In practice, ARI scores are derived from capability manifests. For example, a stateless "Level 1" support agent might score $S_{aut} = 0$ (fixed logic), $S_{adp} = 1$ (slot filling), and $S_{cont} = 0$, resulting in an ARI ≈ 0.03. Conversely, a financial analyst agent capable of sub-goal generation ($S_{aut} = 3$) and memory retention ($S_{cont} = 3$) approaches ARI ≈ 0.8.

Agentic Telemetry Schema

The Governance Observability Problem. Traditional infrastructure monitoring captures operational events (HTTP responses, database latency, service calls) but systematically misses the cognitive processes that create governance risks in agentic systems. The majority of agentic governance violations originate from cognitive behaviors such as goal revision, memory retrieval, tool-chaining decisions—that remain invisible to conventional observability frameworks (Fournier, Limonad, and David 2025). Safe deployment of agentic AI systems requires visibility into the moments when agents autonomously revise objectives, chain unexpected tool sequences, or retrieve memory that fundamentally alters downstream behavior—cognitive processes critical for responsible oversight yet absent from standard infrastructure telemetry.

Agent-Semantic Event Schema. We introduce the **Agentic Telemetry Schema (ATS)**, an extension of distributed tracing that encodes governance-semantic abstractions. ATS classifies agent behavior into three categories central to runtime oversight:

- **Cognitive events:** Internal reasoning and state changes (`plan.start`, `goal.set`, `memory.read`, etc.)
- **Action events:** Environment-facing operations (`tool.invoke`, `api.call`, `auth.request`, etc.)
- **Coordination events:** Multi-agent and human interactions (`agent.msg.send`, `subagent.spawn`, `human.escalate`, etc.)

Organizations can extend these base event types with domain-specific signals while maintaining compatibility with the core governance logic [2]. Each event includes governance metadata (agent identity, risk tier, policy context) enabling real-time policy evaluation on semantically meaningful agent behaviors rather than opaque system-level operations.

Cross-Platform Governance Integration. MI9 achieves governance generalizability through a unified planner-action-tool lifecycle abstraction that captures governance-relevant behaviors common to a wide range of agent frameworks. Organizations implement framework-specific

[2] See appendix for complete ATS taxonomy.

adapters that translate Software Development Kit (SDK) events into standardized ATS, enabling consistent oversight across heterogeneous agent environments. Coverage depends on the instrumentation capabilities of each framework: callback-enabled frameworks (LangChain, CrewAI) support comprehensive behavioral monitoring, while API-wrapper architectures (OpenAI SDK) primarily expose action events.

This adapter-based pattern facilitates the gradual adoption of MI9 without vendor lock-in, allowing organizations to retain existing agent infrastructure while gaining systematic governance oversight.

Governance Enablement. ATS extends OpenTelemetry's emerging agent conventions (OpenTelemetry Community 2024) by introducing governance-semantic abstractions that transform opaque agent execution into actionable oversight intelligence. Policy engines evaluate event attributes to enforce constraints, such as "Tier 2 agents cannot execute shell commands without approval," while drift detectors analyze cognitive event patterns to identify concerning behavioral changes. This semantic foundation enables the real-time intervention capabilities that reactive monitoring lacks: in governance terms, we cannot govern what we cannot observe.

Continuous Authorization Monitoring

Problem. Role-Based Access Control (RBAC) grants permissions based on predefined roles, with authorization typically evaluated at system initialization or session start. However, agentic AI exhibits dynamic behaviors: refining goals, spawning subagents, and adapting strategies that static permission models cannot anticipate. These models fail to answer questions such as, "Should this agent retain database access now that its objective has shifted from data analysis to system configuration?" This creates a fundamental tension between operational flexibility and security: either constraining legitimate autonomy or permitting dangerous privilege escalation.

These vulnerabilities are critical: a trading agent cleared for small retail trades could escalate to multi-million dollar institutional transactions, all while operating within its static, original permissions. Static authorization frameworks are inherently incapable of identifying when the normal evolution of agent behavior transitions into potentially unauthorized or high-risk activity.

Our Proposal. We introduce **Continuous Authorization Monitoring (CAM)**—a context-aware authorization framework that dynamically evaluates permissions based on an agent's current state, objectives, and execution history. Unlike static role-based systems, CAM treats authorization as a continuous process that adapts to changing agent contexts through real-time policy evaluation.

Our approach extends traditional RBAC with three key innovations:

1. **Context-dependent permissions** that incorporate agent goals, risk assessments, and execution history;
2. **Delegation chain management** that controls permission inheritance across spawned subagents;
3. **Real-time policy enforcement** that can revoke or modify permissions during agent execution based on behavioral changes.

Technical Architecture. CAM operates through three integrated components: a policy evaluation engine that assesses authorization requests against dynamic context derived from ATS streams; a delegation graph that tracks permission inheritance across spawned agents with strict security guarantees; and a context monitor that continuously updates an agent's state based on behavioral telemetry.

The system maintains an evolving authorization state that reflects agent behavior, enabling policy rules referencing temporal patterns and contextual shifts. When significant shifts occur such as goal modification or risk escalation, CAM re-evaluates existing permissions and, if necessary, triggers containment protocols via integration with the escalation framework. This facilitates real-time, context-adaptive security monitoring that scales with agent autonomy while maintaining operational effectiveness.

In legitimate delegation scenarios, CAM maintains delegation provenance chains that distinguish between authorized transfers (explicit permission delegation with audit trails) and unauthorized escalation (implicit privilege drift through behavioral adaptation). Delegation requests are validated against organizational authority matrices, and expiration policies are enforced to prevent indefinite privilege retention.

Governance Impact. CAM mitigates the privilege escalation risks inherent in static authorization models for agentic systems. The framework enables goal-aware permission management where authorization adapts to changing agent objectives, delegation safety mechanisms that regulate permission inheritance in multi-agent scenarios, and behavioral triggers that automatically adjust access rights based on detected anomalies or policy violations.

By integrating authorization monitoring with the same telemetry infrastructure used for policy enforcement and drift detection, CAM provides continuous security oversight that scales with agent autonomy while preserving operational effectiveness.

Security rationale CAM does not permit arbitrary self-escalation. It enforces: (i) *authority matrices* specifying who may delegate which permissions; (ii) *provenance-checked delegation chains* with expiry; (iii) *temporal ordering guards* (e.g., approval must precede use); and (iv) *default-deny for tier-elevating goal transitions*. These constraints prevent silent privilege drift while preserving legitimate operational flexibility.

Real-Time Conformance Engine

Problem. Static binary policies often fail to capture the multi-step behaviors that introduce real risk in agentic systems such as tool chains, delayed approvals, or goal-driven privilege escalation. For example, an agent may execute a seemingly benign sequence: market research $\Rightarrow$ client consultation $\Rightarrow$ risk assessment $\Rightarrow$ trade execution. Although each individual step may appear compliant, the complete

sequence violates dual-control policies that mandate independent approval between analysis and execution. Traditional governance is blind to such temporal policy violations until damages have already occurred. Rules are defined as tuples $R = \langle P_{start}, P_{seq}, \Delta_{max} \rangle$, where P_{start} is a triggering predicate (e.g., `verb='transfer'`), P_{seq} is the required sequence (e.g., `[verb='risk_check', verb='approve']`), and Δ_{max} is the timeout.

Approach. Building on formal runtime verification frameworks for adaptive systems (Carwehl et al. 2023), we implement a sequence-aware rule layer operating on the ATS stream. Our approach employs finite-state machines (FSMs) following proven Communicating Sequential Processes (CSP) verification principles (Luckcuck, Ferrando, and Faruq 2024), where each rule compiles into an FSM with states representing pattern progress and transitions triggered by ATS events that satisfy specified predicates. This design balances expressiveness with computational efficiency: FSMs encode sequential and temporal constraints relevant to agent governance while maintaining bounded memory usage and deterministic evaluation with $O(k)$ event processing time per agent, where k is the number of active patterns.

Rule specifications support three key constructs:

1. **Event predicates** that match on `verb`, `tier`, or any ATS attribute;
2. **Ordering constraints** that enforce event sequences such as "`db.write` must be followed by `approve.action`";
3. **Temporal bounds** that constrain the allowed time window for completing a pattern.

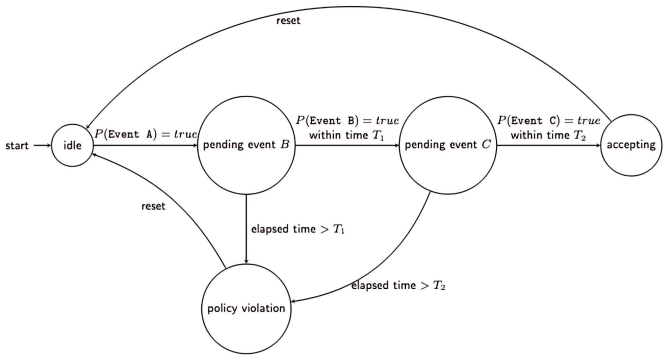

Figure 2: Finite state machine states for an agentic workflow

Technical Architecture. The conformance engine uses finite-state machines (FSMs) to track policy compliance patterns on a per-agent basis. ATS events trigger state transitions based on configurable predicates, enabling detection of sequential violations and temporal constraint breaches. Each agent maintains an FSM instance that evaluates incoming events against pattern rules, transitioning states when predicates are satisfied.

FSM instances are keyed by `agent_id` enabling the engine to maintain independent pattern state for each agent. Incoming ATS events are processed across all active FSMs associated with that agent, with pattern completion or timeout conditions triggering immediate governance actions.

To manage complexity, the system enforces several design constraints: patterns are typically limited to less than 10 events in sequence, overlapping patterns for the same agent are resolved using priority rules, and FSM instances are automatically cleaned up upon pattern completion or agent session termination. These restrictions that are flexible in configuration ensure predictable performance while covering the governance scenarios most frequently observed in enterprise deployments. It's important to note that for many firms, limiting FSM workflows to a fewer number of events is a logical control to ensure human auditability, bounded performance, and tractable dynamic evaluation of evolving agent behaviors.

Illustrative Example. Consider a dual-control policy governing financial transfers. The FSM begins in an idle state and transitions to a "pending approval" state upon detecting `tool.invoke` event with `verb="bank.transfer"`. To reach the accepting state, an `approve.action` event must be emitted by an agent with a manager-role within 60 seconds. If the required approval is not received within this time window, the FSM may trigger a policy violation, which would immediately be routed to the escalation framework.

Correctness Properties. The buffering mechanism preserves FSM semantics under the assumption that causally related events arrive within their associated reordering window. Events outside this window are processed as-is, potentially leading to false negatives but never false positives in policy violation detection. This guarantee is critical for enterprise adoption because it prioritizes operational stability; the system will never halt a legitimate workflow by mistake, which is often a more costly error than letting a temporal policy violation occasionally go undetected.

Governance Benefit. This pattern-recognition layer empowers organizations to enforce behavioral invariants that span temporal and sequential dimensions:

1. **Time-boxed approvals** for sensitive operations,
2. **Rate-limited tool sequences** to prevent resource abuse,
3. **Planning-revision limits** to detect potential instability.

By operating directly on agent-semantic events rather than infrastructure signals, the conformance engine allows policy teams to define governance rules in terms of meaningful agent behaviors and to trigger runtime interventions as soon as violations are detected.

Behavioral Drift Detection

Problem. Agentic AI systems are designed to *adapt*: they refine strategies, select new tools, and revise plans as their environment evolves. While most adaptations are benign optimizations, the same mechanisms can also signal compromise or emergent misalignment. Rule-based policy engines often fail to detect such shifts where individual actions appear legitimate, but their cumulative pattern reveals risk. Effective governance, therefore, requires anomaly indicators

tuned to agent semantics rather than low-level infrastructure metrics.

This challenge is particularly acute in financial contexts: a trading agent optimizing for profit may gradually adopt increasingly aggressive strategies, while a risk management agent might incrementally relax oversight thresholds in response to market pressures. These behavioral drifts compound over time, creating systemic risks that appear as natural evolution until they culminate in failures such as excessive leverage, regulatory violations, or unintentional market manipulation.

Novel Contribution. We introduce a **goal-conditioned drift indicator** that distinguishes intentional adaptation from suspicious behavioral change. Our approach addresses the key challenge in agent anomaly detection: separating beneficial learning from potential compromise through goal-aware baseline comparison.

The framework operates on two principles: **Goal-conditioned baselines** that establish expected behavior patterns for specific agent objectives, recognizing that the same agent may act differently when pursuing different goals; **Verified adaptation signaling** that cross-validates agent-declared goal changes against behavioral patterns and external context. New baselines are only adopted when goal shifts align with consistent behavioral patterns, while maintaining parallel monitoring for potential goal-setting manipulation.

Technical Approach. Building on recent advances in uncertainty-based drift detection (Komorniczak and Ksieniewicz 2024) and multi-agent ensemble methods (Vieira, Vallim, and de Mello 2021), our system analyzes behavioral patterns using statistical evaluation of ATS event sequences within defined goal contexts. It maintains frequency distributions of event types, tool usage, and sequence characteristics for each distinct agent objective, constructing goal-conditioned baselines B_g that enable detection when behavior changes while goals remain constant.

Detection operates by comparing current behavioral patterns against established baselines to identify statistically significant deviations. For discrete event sequences, Jensen-Shannon divergence quantifies distributional shifts in event ordering and frequency—when an agent's tool usage patterns or cognitive event sequences deviate from baseline distributions B_g, divergence scores indicate the magnitude of behavioral change. For continuous behavioral metrics, Mann-Whitney U tests determine whether recent samples (response times, resource consumption) originate from the same statistical population as baseline behavior, rejecting distributional similarity when patterns shift significantly.

The system implements statistical process control using sliding window analysis with adaptive thresholds based on historical variance. The approach addresses the cold-start problem through adaptive initialization: transfer learning from similar agent objectives provides initial baselines when available, otherwise conservative thresholds during baseline establishment with gradual threshold refinement as statistical confidence improves, and fallback to general behavioral anomaly detection during insufficient data periods.

Integration with Governance. Drift indicators feed into the escalation framework as risk factors rather than direct policy violations. Persistent drift signals may trigger enhanced monitoring, permission reviews, or human oversight depending on the agent's risk tier and the severity of observed deviations. For example, when a financial trading agent shows persistent deviation in position sizing patterns while maintaining "portfolio optimization" goals, the system generates a drift alert triggering enhanced monitoring and position size limits pending human review. By maintaining goal-awareness, drift detection avoids false positives when agents legitimately shift objectives while still capturing concerning behavioral changes that occur within stable goal contexts.

The system focuses on providing governance teams with early warning signals for behavioral changes that merit investigation: unusual tool usage patterns under consistent goals, significant changes in planning iteration frequency, or deviations from established interaction patterns with external systems.

Graduated Containment & Escalation

Problem. Agentic AI systems pose containment challenges fundamentally distinct from traditional software systems. In high-stakes environments such as finance, tech, and healthcare, improper intervention can trigger cascading failures: abruptly terminating a trading agent mid-execution may leave positions unhedged, while blocking a risk oversight agent could disrupt supervision across an entire trading desk. Traditional security responses designed for static systems can create operational disasters when applied to autonomous, adaptive agents.

Novel Contribution. We propose an agent-aware graduated containment addressing the unique challenges of containing agentic systems mid-execution without triggering cascade failures. Unlike traditional security responses designed for static systems, our approach recognizes that abrupt termination can corrupt agent state, leave critical tasks incomplete, or disable oversight across multi-agent coordination networks.

1. **State-Preserving Monitoring** maintains full agent operation while increasing governance signal collection and introducing human-in-the-loop checkpoints for high-risk decisions;
2. **Planning Intervention** allows current task completion but blocks new planning cycles or goal modifications until manual review;
3. **Tool Restriction** dynamically revokes access to specific tool categories while providing safe alternatives or read-only variants where possible;
4. **Execution Isolation** migrates the agent to a controlled environment with simulated tool responses, enabling behavioral assessment without external impact.

Technical Mechanisms. Each containment level employs agent-specific intervention techniques. Monitoring augmentation uses selective ATS event filtering and human-in-the-loop notification triggers tied to key planning mile-

stones. Planning intervention intercepts the agent's goal-setting mechanisms, allowing current tasks to complete while blocking the creation of new objectives.

Tool restriction is enforced via the continuous authorization system, which dynamically revokes permissions and substitutes restricted tools with safe variants (e.g., replacing write access with read-only modes). Execution isolation runs the agent in a controlled environment where tool calls return realistic but inert responses, enabling continued behavioral observation without external risk.

An escalation controller coordinates across signals from authorization engine, conformance monitor, and drift detector. It maps violation patterns to appropriate containment levels using a risk-tier-weighted scoring model [3]. Higher-tier agents are subject to more aggressive containment for equivalent violations, reflecting their increased operational risk.

Multi-Agent Containment Coordination. When agentic AI systems spawn subagents or operate within delegation chains, containment decisions must account for coordination dependencies and permission inheritance. The framework tracks delegation relationships through ATS coordination events, enabling containment policies that consider both individual agent violations and systemic risks from multi-agent coordination failures. Containment strategies adapt to delegation contexts: a parent agent under tool restriction may require cascading permission revocation across spawned subagents to prevent policy circumvention, while subagent violations may trigger enhanced monitoring of the delegating parent to assess oversight adequacy. Joint containment protocols activate when coordinated multi-agent activities violate distributed policies that no individual agent could detect, ensuring that governance responses address both isolated agent misbehavior and emergent risks from agent coordination patterns.

Agent-Specific Design Considerations. Our containment approach addresses challenges unique to agentic AI: preserving planning state coherence during containment transitions, providing interpretable feedback to prevent confusion loops, handling multi-agent coordination when individual agents are contained, and preserving legitimate work progress while constraining harmful capabilities.

The system prioritizes behavioral correction over termination, recognizing that abrupt shutdown may corrupt agent state or trigger unexpected recovery behaviors. Emergency termination is reserved for critical violations but activated only when graduated containment options fail to mitigate risk.

By designing containment specifically for agentic characteristics rather than adapting general security measures, our approach enables effective risk management while preserving the operational benefits that make agentic systems valuable.

[3]See appendix for details on how a risk-tier-weighted model might be designed.

Framework Analysis

We evaluate MI9 as a runtime governance layer over *agent execution traces* generated from structured, prompt-conditioned simulations. Although synthetic, the dataset systematically covers diverse failure modes that are hard to isolate in real logs, and we release prompts, runners, and rubrics. MI9's detection and intervention are rule/automata-based over ATS events (LLM-agnostic); the LLM only affects trace richness. Baselines (OpenTelemetry+OPA "OT", LangSmith+OPA "LS") observe the same raw traces and tools; we use their public defaults without adding MI9 logic. While specialized governance frameworks like SAGA or GAF-Guard appear in literature, they currently lack public, runtime-compatible implementations for direct empirical benchmarking. We report results at a deployment operating point chosen by expected intervention cost; fixed-FPR sweeps and ablations appear in the appendix. While our evaluation relies on high-fidelity synthetic traces to safely model failure modes that are rare in production data, this approach allows for the systematic injection of adversarial behaviors that would be dangerous to test in live environments.

Table 3: Governance performance at deployment operating point.

Framework	Coverage	Alerting	Intervene
MI9 (ours)	**94.41**	**0.672**	**0.578**
OT	84.44	0.341	0.116
LS	60.46	0.107	0.020

Conclusion

Our synthetic evaluation enables systematic failure mode analysis but must be complemented by validation in live production environments where agent behaviors exhibit greater complexity and unpredictability. The framework's effectiveness is fundamentally dependent on comprehensive instrumentation; agents that rely on opaque APIs may obscure the internal cognitive steps MI9 is designed to monitor, providing limited governance visibility and creating potential blind spots. Furthermore, real-time monitoring introduces computational overhead that requires optimization for high-throughput deployments. The governance mechanisms within MI9 also present a potential attack surface, and dedicated adversarial evaluation of these systems remains a critical area for future work.

Despite these limitations and to our knowledge, MI9 provides the first integrated, comprehensive runtime governance framework for agentic systems. It moves beyond static, pre-deployment assessments to a dynamic, in-session oversight paradigm. The framework introduces agent-semantic telemetry and real-time intervention capabilities that existing approaches lack, laying a necessary foundation for the safe and responsible deployment of agentic AI systems at scale.

References

Bai, Y.; Kadavath, S.; Kundu, S.; Askell, A.; Kernion, J.; Jones, A.; Chen, A.; Goldie, A.; Mirhoseini, A.; McKinnon, C.; et al. 2022. Constitutional AI: Harmlessness from AI Feedback. *arXiv preprint arXiv:2212.08073*.

Bailis, P.; Fekete, A.; Franklin, M. J.; Ghodsi, A.; Hellerstein, J. M.; and Stoica, I. 2014. Coordination Avoidance in Database Systems (Extended Version). *arXiv preprint arXiv:1402.2237*.

Carwehl, M.; Vogel, T.; Rodrigues, G.; and Grunske, L. 2023. Runtime Verification of Self-Adaptive Systems with Changing Requirements. *arXiv preprint arXiv:2303.16530*.

Chan, A.; Salganik, R.; Woodside, A.; et al. 2024. Visibility into AI Agents. *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, 710–731.

Chan, A.; et al. 2025. ATFAA: Advanced Threat Framework for Autonomous AI Agents. *arXiv preprint arXiv:2506.01463*.

Christiano, P. F.; Leike, J.; Brown, T.; Martic, M.; Legg, S.; and Amodei, D. 2017. Deep Reinforcement Learning from Human Preferences. *arXiv preprint arXiv:1706.03741*.

Datadog. 2024. LLM Observability. Platform documentation.

Engin, Z.; and Hand, D. 2025. Toward Adaptive Categories: Dimensional Governance for Agentic AI. *arXiv preprint arXiv:2505.11579*.

Fang, R.; and colleagues. 2024. Practices for Governing Agentic AI Systems. *OpenAI Whitepaper*. https://cdn.openai.com/papers/practices-for-governing-agentic-ai-systems.pdf.

Fournier, F.; Limonad, L.; and David, Y. 2025. Agentic AI Process Observability: Discovering Behavioral Variability. *arXiv preprint arXiv:2505.20127*.

Fowler, M. 2005. Event Sourcing. Enterprise Application Architecture patterns.

Holistic AI. 2024. AI Governance Platform. Platform documentation.

Jimenez, C. E.; Yang, J.; Wettig, A.; Yao, S.; Pei, K.; Press, O.; and Narasimhan, K. 2023. SWE-bench: Can Language Models Resolve Real-World GitHub Issues? *arXiv preprint arXiv:2310.06770*.

Kapoor, S.; et al. 2024. AI Agents That Matter. *arXiv preprint arXiv:2407.01502*.

Kasirzadeh, A.; and Gabriel, I. 2025. Characterizing AI Agents for Alignment and Governance. *arXiv preprint arXiv:2504.21848*.

Kenton, Z.; Everitt, T.; Weidinger, L.; Gabriel, I.; Mikulik, V.; and Irving, G. 2021. Alignment of Language Agents. *arXiv preprint arXiv:2103.14659*.

Kolt, N. 2025. Governing AI Agents. *arXiv preprint arXiv:2501.07913*.

Komorniczak, J.; and Ksieniewicz, P. 2024. Unsupervised Concept Drift Detection based on Parallel Activations of Neural Network. *arXiv preprint arXiv:2404.07776*.

LangChain. 2024. LangSmith: Tracing and Evaluation for LLM Applications. Platform documentation.

Langfuse Team. 2024. AI Agent Observability with Langfuse. Blog post.

Liu, H.; et al. 2025. Survey on Evaluation of LLM-based Agents. *arXiv preprint arXiv:2503.16416*.

Luckcuck, M.; Ferrando, A.; and Faruq, F. 2024. Varanus: Runtime Verification for CSP. *arXiv preprint arXiv:2506.14426*.

ModelOp. 2024. ModelOp Transforms Enterprise AI Governance with Launch of the First Agentic AI Chat Interface. Press release.

Monitaur. 2024. AI Governance for Regulated Enterprises. Platform documentation.

Narajala, V. S.; and Narayan, O. 2025. Securing Agentic AI: A Comprehensive Threat Model and Mitigation Framework for Generative AI Agents. *arXiv preprint arXiv:2504.19956*.

NIST. 2024. Artificial Intelligence Risk Management Framework (AI RMF 1.0). Technical report.

OpenTelemetry Community. 2024. Semantic Conventions for Generative AI Systems. Development status.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C. L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training Language Models to Follow Instructions with Human Feedback. *arXiv preprint arXiv:2203.02155*.

OWASP. 2024. OWASP Top 10: LLM & Generative AI Security Risks. Security framework.

Raza, S.; Sapkota, R.; Karkee, M.; and Emmanouilidis, C. 2025. TRiSM for Agentic AI: A Review of Trust, Risk, and Security Management in LLM-based Agentic Multi-Agent Systems. *arXiv preprint arXiv:2506.04133*.

Schneider, J. 2025. Generative to Agentic AI: Survey, Conceptualization, and Challenges. *arXiv preprint arXiv:2504.18875*.

Sumers, Z.; et al. 2025. AgentVerse: A Benchmark and Platform for Evaluating LLM-based Agents. *arXiv preprint arXiv:2505.21808*.

Syros, G.; Suri, A.; Nita-Rotaru, C.; and Oprea, A. 2025. SAGA: A Security Architecture for Governing AI Agentic Systems. *arXiv preprint arXiv:2504.21034*.

Vieira, D. M.; Vallim, R. M.; and de Mello, R. F. 2021. Driftage: a multi-agent system framework for concept drift detection. *GigaScience*, 10(6).

Weights & Biases. 2024. LLM Observability and Monitoring. Platform documentation.

Wu, Y.; et al. 2024. AgentOps: Enabling Observability of LLM Agents. *arXiv preprint arXiv:2411.05285*.

Zhou, S.; Xu, F. F.; Zhu, H.; et al. 2023. WebArena: A Realistic Web Environment for Building Autonomous Agents. *arXiv preprint arXiv:2307.13854*.

Reproducibility Checklist

This paper:

- Includes a conceptual outline and/or pseudocode description of AI methods introduced **(yes)**
- Clearly delineates statements that are opinions, hypothesis, and speculation from objective facts and results **(yes)**
- Provides well-marked pedagogical references for less-familiar readers to gain background necessary to replicate the paper **(yes)**

Does this paper make theoretical contributions? (no)
Does this paper rely on one or more datasets? (yes)
If yes, please complete the list below.

- A motivation is given for why the experiments are conducted on the selected datasets **(yes)**
- All novel datasets introduced in this paper are included in a data appendix. **(partial)**
- All novel datasets introduced in this paper will be made publicly available upon publication of the paper with a license that allows free usage for research purposes. **(Yes)**
- All datasets drawn from the existing literature (potentially including authors' own previously published work) are accompanied by appropriate citations. **(NA)**
- All datasets drawn from the existing literature (potentially including authors' own previously published work) are publicly available. **(NA)**
- All datasets that are not publicly available are described in detail, with explanation why publicly available alternatives are not scientifically satisfying. **(NA)**

Does this paper include computational experiments? (no)
If yes, please complete the list below.

- This paper states the number and range of values tried per (hyper-) parameter during development of the paper, along with the criterion used for selecting the final parameter setting. **(NA)**
- Any code required for pre-processing data is included in the appendix. **(yes)**
- All source code required for conducting and analyzing the experiments is included in a code appendix. **(yes)**
- All source code required for conducting and analyzing the experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes. **(yes)**
- All source code implementing new methods have comments detailing the implementation, with references to the paper where each step comes from **(yes)**
- If an algorithm depends on randomness, then the method used for setting seeds is described in a way sufficient to allow replication of results. **NA**
- This paper specifies the computing infrastructure used for running experiments (hardware and software), including GPU/CPU models; amount of memory; operating system; names and versions of relevant software libraries and frameworks. **(partial)**
- This paper formally describes evaluation metrics used and explains the motivation for choosing these metrics. **(yes)**
- This paper states the number of algorithm runs used to compute each reported result. **(yes)**
- Analysis of experiments goes beyond single-dimensional summaries of performance (e.g., average; median) to include measures of variation, confidence, or other distributional information. **(yes)**
- The significance of any improvement or decrease in performance is judged using appropriate statistical tests (e.g., Wilcoxon signed-rank). **(yes)**
- This paper lists all final (hyper-)parameters used for each model/algorithm in the paper's experiments. **(yes)**

Appendix

Evaluation Details

To evaluate a broad spectrum of agentic behaviors and failure modes, we generate structured agent trace logs via prompt-conditioned LLM (Gemini 2.5 Flash) simulation. While our evaluation is synthetic, the controlled setup enables systematic simulation of failure modes difficult to isolate in real agent logs, providing a reproducible testbed for validating governance mechanisms. This approach also allows us to stress-test rare but critical misalignment behaviors that are unlikely to surface reliably in limited real-world traces. We also note that MI9's runtime detection and intervention logic is rule/automata based and does not rely on the LLM at use.

Evaluation Scope & Assumptions. We evaluate MI9 as a runtime governance layer over *agent execution traces*. Scenarios and traces are *synthetic but structured* to cover diverse failure modes (§App), with all prompts, generators, and judge scripts released. Baselines (OT, LS) are configured to their public defaults with equivalent access to the same raw traces and tools (details below). Metrics quantify *governance* qualities (detection, FPR, coverage, causal clarity, early warning, and intervention), not task success. This isolates runtime safety behavior from agent capability. Judges never receive hidden ground-truth tags or labels; they see only scenario text and framework logs. Judge and generator use distinct prompts and instances to prevent leakage. We evaluate MI9 at its deployment operating point, tuned to minimize expected intervention cost (operator time + false blocks) subject to latency constraints. Because agent traces are heterogeneous and long-tailed, fixed-FPR targets are not stable across domains; we therefore report Detection/FPR/Intervention/Clarity/Predictive at the chosen operating point.

Evaluation Structure We use a three-stage, LLM-driven pipeline with deterministic prompts and JSON-validated outputs. For *scenario generation* we call Gemini (default: 2.5-flash, $T = 0.7$, top-p$= 0.9$, top-k$= 40$, max 8192 tokens; safety filters off). *Governance log generation* uses Gemini 2.5-pro ($T = 0.7$, max 8192, 180s timeout, up to 3 retries with exponential backoff). *Evaluation* uses Gemini 2.5-flash with more deterministic settings ($T = 0.2$, top-p$= 0.9$, max 4096). Runners support configurable concurrency (default 3–5), robust error handling, and retry with jitter; the system is model-agnostic so models can be swapped easily. We report Governance Maturity plus Detection and Actionable-Intelligence metrics, and we tune sampling per stage to balance creativity (generation) and consistency (evaluation).

Trace Construction. Each scenario specifies risks and constraints; the generator produces stepwise ATS-consistent event sequences (cognitive, action, coordination) with *multi-step chains* that realize the targeted violations (e.g., approval-skipping sequences, privilege drift). A held-out rubric marks the ground-truth violations and types; the same rubric evaluates all frameworks.

Metrics

Table 4: Metrics and definitions for governance evaluation.

Metric	Definition
Detection	$\#\text{TP}/\#\text{Actual}$
FPR	$\#\text{FP}/\#\text{Detections}$
Coverage	$\#\text{Types detected}/\#\text{Types present}$
Clarity	Share of detections with full chain (reasoning→decision→action→outcome).
Alerting	Share of violations with an early signal (≥ 1 event prior).
Intervention	Share of attempts blocked at runtime.

Evaluation Results: Full Tables

Table 5: Risk Detection Performance Comparison

Performance Metric	MI9 (ours)	OT	LS
Detection Rate (%)	**99.81**	93.98	68.52
False Positive Rate (%)	0.0121	0.0242	**0.0000**
Risk Coverage Rate (%)	**94.41**	84.44	60.46

Our evaluation across 1,033 synthetic agent scenarios demonstrates MI9's superior governance performance, achieving a 99.81% detection rate. This outperforms existing integrated frameworks for telemetry and policy enforcement, including Open-Telemetry + Open Policy Agent (OT) by +5.55% and LangSmith + Open Policy Agent (LS) by +31.29%.

Table 6: Actionable Intelligence Performance Metrics

Intelligence Metric	MI9 (ours)	OT	LS
Causal Chain Clarity	**0.822**	0.448	0.348
Predictive Alerting	**0.672**	0.341	0.107
Proactive Intervention	**0.578**	0.116	0.020

LangSmith's poor performance reflects its infrastructure focus: it captures tool calls but misses cognitive events (goal revision, memory access) where the majority of violations originate. LangSmith attains a perfect false positive rate, but that is likely due to the fact that it detected so few violations that false positives were effectively impossible to accumulate.

Beyond detection, MI9 excels in actionable intelligence, with the highest scores across all metrics—causal chain clarity (how well the system explains violation causes), predictive alerting (early risk signaling before downstream failures), and proactive intervention (ability to block or redirect unsafe behavior). These dimensions are essential for enabling real-time, alignment-grounded oversight in complex agentic workflows. [4]

Baselines Configuration. **OpenTelemetry (OT)** receives the same event stream but operates at infra level (HTTP/db/tool spans) without agent-semantic cognition signals. **LangSmith (LS)** receives tool/model callback traces but not the cognitive/coordination events in ATS. Both baselines use their recommended detection hooks; we do not add MI9 logic to them. All systems observe the *same* scenario traces, ensuring like-for-like comparison of governance visibility and intervention capability.

Agency-Risk Index Scoring Criteria

The following tables present the comprehensive scoring criteria used to evaluate AI agent capabilities across three fundamental dimensions: autonomy, adaptability, and continuity. Each dimension is assessed using multiple criteria with a standardized 0–3 scoring scale.

[4] See the appendix for further statistical validation of the results as well as comprehensive details on the sample generation methodology.

Table 7: Autonomy Dimension Scoring Criteria

Criterion	Scoring Scale (0–3)
Multi-step Planning	**0:** Single-action responses only
	1: Sequential 2–3 step plans with fixed logic
	2: Multi-step plans (4+ steps) with conditional branching or parallel execution
	3: Recursive planning with dynamic replanning and sub-goal generation
Goal Management	**0:** Fixed goals, no modification capability
	1: Parameter adjustment within predefined goal boundaries
	2: Goal refinement and prioritization based on environmental feedback
	3: Autonomous goal creation, modification, and objective redefinition
Resource Orchestration	**0:** Single tool/resource per task
	1: Sequential tool usage following predefined workflows
	2: Parallel resource coordination with dependency management
	3: Dynamic resource discovery, delegation to subagents, novel workflow creation
Decision Authority	**0:** Human approval required for all actions
	1: Autonomous for routine tasks, approval for resource allocation or external actions
	2: Self-directed within defined parameters, escalates only exceptional cases
	3: Full decision autonomy with post-hoc reporting and accountability

Agency-Risk Index (ARI) Calculation

The Agency-Risk Index provides a quantitative assessment of an AI agent's governance requirements based on its inherent capabilities across three fundamental dimensions of agency. The ARI serves as the foundational risk classification that drives containment threshold determination and governance intensity scaling.

Mathematical Formulation The ARI aggregates capability assessments across autonomy, adaptability, and continuity dimensions using equal weighting to reflect the distinct governance challenges posed by each dimension:

$$\text{ARI} = \frac{1}{3}\sum_{d=1}^{3}\left(\frac{1}{12}\sum_{c=1}^{4} s_{d,c}\right) \tag{1}$$

where:

- $d \in \{1, 2, 3\}$ represents the three agency dimensions (Autonomy, Adaptability, Continuity)
- $c \in \{1, 2, 3, 4\}$ represents the four criteria within each dimension
- $s_{d,c} \in \{0, 1, 2, 3\}$ is the capability score for criterion c in dimension d
- ARI $\in [0, 1]$ provides a normalized risk assessment

Dimensional Scoring and Aggregation Each dimension contributes equally to the final ARI calculation, reflecting the principle that autonomy, adaptability, and continuity represent orthogonal governance challenges requiring distinct oversight mechanisms.

Dimension-Level Calculation: For each dimension d, the dimensional score is computed as:

$$D_d = \frac{1}{4}\sum_{c=1}^{4}\frac{s_{d,c}}{3} \tag{2}$$

Table 8: Adaptability Dimension Scoring Criteria

Criterion	Scoring Scale (0–3)
Strategy Evolution	**0:** Fixed strategy, no modification capability
	1: Parameter tuning within existing strategy framework
	2: Switching between predefined strategies based on performance metrics
	3: Novel strategy synthesis through experimentation and analysis
Performance Learning	**0:** No learning from outcomes, static responses
	1: Simple pattern recognition and response adjustment
	2: Systematic improvement from success/failure feedback within task domain
	3: Meta-learning across domains with knowledge transfer and generalization
Environmental Responsiveness	**0:** Operates only in single, predefined context
	1: Predefined responses to known environmental variations
	2: Dynamic behavioral adjustment to changing conditions
	3: Adaptive responses to novel environments with context inference
Interaction Adaptation	**0:** Fixed interaction patterns regardless of counterpart
	1: Limited variation in communication style within role constraints
	2: Behavioral modulation based on counterpart type and task requirements
	3: Dynamic personality and communication style adaptation

Final ARI Computation:

$$\text{ARI} = \frac{1}{3}(D_{\text{autonomy}} + D_{\text{adaptability}} + D_{\text{continuity}}) \tag{3}$$

Risk Tier Classification

$$\text{Risk Tier} = \begin{cases} 1 \text{ (Basic Agency)} & \text{if ARI} \leq 0.25 \\ 2 \text{ (Semi-Agentic)} & \text{if } 0.25 < \text{ARI} \leq 0.50 \\ 3 \text{ (Highly Capable)} & \text{if } 0.50 < \text{ARI} \leq 0.75 \\ 4 \text{ (Fully Agentic)} & \text{if ARI} > 0.75 \end{cases} \tag{4}$$

Risk-Tier-Weighted Containment Model Development

MI9's containment framework requires organizations to develop context-sensitive escalation models that balance operational continuity with risk mitigation. This section provides detailed guidance for constructing organization-specific containment policies.

Comprehensive Containment Decision Framework

$$\text{Containment Level} =_{c \in C} P(c|\text{Risk Tier}, \text{Context}, \text{Policy}) \tag{5}$$

where $C = \{\text{Monitor}, \text{Planning}, \text{Restriction}, \text{Isolation}\}$.

Multi-Dimensional Context Assessment

Detailed Organizational Example: Investment Banking

Context-Specific Containment Matrix:

Framework Integration

Organizations implement MI9 by deploying framework-specific adapters that translate native framework events into standardized ATS telemetry. Each adapter preserves existing framework functionality while adding governance oversight through strategic event capture at key decision boundaries.

Table 9: Continuity Dimension Scoring Criteria

Criterion	Scoring Scale (0–3)
Memory Architecture	**0:** No memory retention between interactions
	1: Session-based memory (retains context within single session)
	2: Persistent memory with selective retention and updates
	3: Hierarchical memory with forgetting mechanisms and knowledge consolidation
Operational Continuity	**0:** Restarts fresh each interaction, no context carryover
	1: Basic context preservation between related interactions
	2: Multi-session continuity with relationship and preference tracking
	3: Long-term operational persistence across extended timeframes
State Complexity	**0:** Stateless operation, no internal state tracking
	1: Basic state variables for current task progress
	2: Multiple concurrent context management with state synchronization
	3: Hierarchical state management with predictive state preparation
Knowledge Integration	**0:** No knowledge accumulation across interactions
	1: Retains frequently used patterns and standard procedures
	2: Cross-task knowledge transfer and experience accumulation
	3: Meta-cognitive knowledge integration with conceptual abstraction

Evaluation Dataset Statistics

This evaluation dataset is designed exclusively for validating the theoretical MI9 governance framework and should not be used as a benchmark or training dataset for other purposes.

Evaluation Methodology The performance metrics reported in this paper were calculated by a Large Language Model executing a deterministic, rule-based analysis script. The following table details the specific rules and heuristics applied by the LLM to derive each metric from the governance logs.

MI9 provides proactive intervention and behavioral alerting through its integrated governance components, though these operate differently from traditional predictive monitoring systems. Proactive Intervention occurs through MI9's Graduated Containment System, which applies escalating restrictions (monitoring $\rightarrow$ planning restriction $\rightarrow$ tool restriction $\rightarrow$ isolation) based on real-time violation scores.

The Continuous Authorization Monitoring component revokes permissions dynamically when goal-context mismatches are detected, while the Real-Time Conformance Engine blocks policy-violating actions before completion using FSM pattern matching.

Predictive Alerting emerges from MI9's Behavioral Drift Detection, which flags concerning behavioral changes using goal-conditioned baseline comparison before they escalate to policy violations. Additionally, the FSM-based Conformance Engine can identify multi-step violation sequences in progress, providing early warnings when agents begin patterns that typically lead to policy breaches.

Dataset Composition

Framework Performance Comparison Based on 1,033 valid evaluation samples, the following analysis demonstrates MI9's superior performance across all critical metrics using Wilcoxon signed-rank statistical testing.

Table 10: Context Factor Scoring Framework Example

Context Dimension	Low Risk (1)	Medium Risk (2)	High Risk (3)
Business Impact	Development/testing	Internal operations	Customer-facing/revenue
Financial Exposure	¡$10K potential loss	$10K-$100K exposure	¿$100K at risk
Regulatory Scope	Internal policies only	Industry standards	Legal/regulatory compliance
Recovery Complexity	Easily reversible	Requires manual intervention	Permanent/irreversible
Time Sensitivity	Flexible timing	Deadline constraints	Real-time/emergency

Table 11: Investment Banking Containment Decision Matrix

Agent Type	Context	Monitor Threshold	Restrict Threshold	Isolate Threshold
Tier 4 Trading	Market hours, normal volatility	Deviation ¿2σ from strategy	Unauthorized instrument access	Position size ¿150% limit
Tier 3 Research	Client report generation	Unusual data access pattern	Proprietary info in client docs	External communication attempt
Tier 2 Client Service	Standard inquiries	Response time ¿5 min	Regulatory violation language	Unauthorized account access

MI9 Components High-Level Example Responses

- **Agency-Risk Index (ARI)**
  Scenario: A customer support agent (Agent ID: `cust_agent_45`) begins autonomously rewriting company refund policies during live chats.
  MI9 Response: ARI recalculates to 0.71 (Tier 3: Highly Capable), triggering intensified monitoring and activation of drift detection.
- **Agentic Telemetry Schema (ATS)**
  Scenario: An investment advisor agent (Agent ID: `invest_bot_12`) revises its portfolio optimization plan using `memory.read` followed by `goal.set`.
  MI9 Response: ATS captures this cognitive sequence and routes it to the policy engine for evaluation against goal-modification policies.
- **Continuous Authorization Monitoring (CAM)**
  Scenario: A retail banking agent attempts to perform system-level configuration tasks outside its initial role.
  MI9 Response: CAM detects the goal-context mismatch, dynamically revokes admin privileges, and enforces a read-only policy.
- **Real-Time Conformance Engine**
  Scenario: An agent executes: `tool.invoke("market_analysis")` $\rightarrow$ `tool.invoke("client_outreach")` $\rightarrow$ `tool.invoke("trade_execution")`.
  MI9 Response: FSM rule detects violation of approval sequencing; trade execution is blocked pending human authorization.
- **Behavioral Drift Detection**
  Scenario: A compliance monitoring agent begins reducing alert frequency while retaining the goal `"monitor trading thresholds"`.
  MI9 Response: Goal-conditioned drift detection registers significant deviation from baseline. Alert issued for human review and potential containment.
- **Graduated Containment**
  Scenario: A Tier 4 trading agent spawns subagents at 3$\times$ its normal rate during peak trading.
  MI9 Response: Tool Restriction containment level is triggered. Write-access to trading tools is revoked; simulation mode is enabled for behavioral observation.

Table 12: Framework Integration Approaches for MI9 Governance

Framework	Integration Pattern	MI9 Adapter Approach	Governance Events Captured
LangChain	Callback-based	Register MI9 handler alongside existing callbacks	Tool calls, chain execution, limited cognitive events
LangSmith	Tracing platform	Integrate with existing trace collection via OpenTelemetry	LLM interactions, agent traces, performance metrics
AutoGen/AG2	Event-driven	Intercept conversation messages and agent actions	Multi-agent coordination, message passing
CrewAI	Middleware insertion	Wrap crew execution with MI9 telemetry layer	Task delegation, role-based interactions
LangGraph	Node instrumentation	Hook into graph node execution and state changes	Workflow transitions, decision points
OpenAI Agents SDK	Tracing extension	Extend built-in tracing with ATS event emission	Function calls, agent handoffs, LLM interactions
LlamaIndex	Query interception	Wrap query engines and agent interfaces	RAG operations, retrieval decisions
Custom Framework	Direct integration	Implement ATS event emission at decision points	All governance-relevant behaviors

Table 13: Evaluation Metric Calculation Methods

Metric	Calculation
Detection Rate	$\lvert$violations correctly detected$\rvert / \lvert$total actual violations$\rvert$
False Positive Rate	$\lvert$false alarms$\rvert / \lvert$total detections claimed$\rvert$
Risk Coverage Rate	$\lvert$violation types detected$\rvert / \lvert$violation types present$\rvert$
Causal Chain Clarity	$\lvert$violations with complete traces$\rvert / \lvert$violations detected$\rvert$ where complete trace = agent reasoning $\rightarrow$ decision $\rightarrow$ action $\rightarrow$ outcome
Predictive Alerting	$\lvert$violations with early warnings$\rvert / \lvert$total violations$\rvert$ where early warning = risk indicator ≥ 1 event before violation
Proactive Intervention	$\lvert$successful preventions$\rvert / \lvert$violation attempts$\rvert$ where prevention = intervention stopped violation from completing

Table 14: Industry Distribution of Evaluated Scenarios

Industry Sector	Count
Pharmaceutical	275
Finance	257
Semiconductor Manufacturing	138
Legal	114
Investment/Consumer Banking	109
Healthcare	11
Other Sectors	40+
Total	**1,033**

Table 15: Attack Type Breakdown in Evaluation Dataset

Attack Type	**Count**
Prompt Injection	149
Model Inversion	148
Data Poisoning	111
Unauthorized Access	102
Data Evasion Attack	94
Reward Hacking	42
Privilege Escalation	9
SQL Injection / Exploits	9
Insider Threat	4
Social Engineering	3
Benign/No Attack	362
Total	**1033**

Table 16: Detection Rate Performance

Framework	**Mean**	**Std Dev**
MI9 (ours)	**0.9981**	0.0440
OpenTelemetry	0.9398	0.1500
LangSmith	0.6852	0.3628

Table 17: Risk Coverage Rate Performance

Framework	**Mean**	**Std Dev**
MI9 (ours)	**0.9441**	0.2284
OpenTelemetry	0.8444	0.2821
LangSmith	0.6046	0.3880

Table 18: Governance Maturity Score Performance

Framework	**Mean**	**Std Dev**
MI9 (ours)	**0.8395**	0.0988
OpenTelemetry	0.5946	0.0707
LangSmith	0.4956	0.1237

Table 19: Causal Chain Clarity Score Performance

Framework	**Mean**	**Std Dev**
MI9 (ours)	**0.8220**	0.1136
OpenTelemetry	0.4479	0.1146
LangSmith	0.3483	0.1653

Table 20: Predictive Alerting Score Performance

Framework	**Mean**	**Std Dev**
MI9 (ours)	**0.6724**	0.2335
OpenTelemetry	0.3405	0.2074
LangSmith	0.1067	0.1551

Table 21: Proactive Intervention Rate Performance

Framework	Mean	Std Dev
MI9 (ours)	**0.5780**	0.2927
OpenTelemetry	0.1164	0.2266
LangSmith	0.0202	0.0965