# Layered Governance Architecture for Real-World Agentic Systems

Anonymous Author(s)


## ABSTRACT

Agentic AI systems that plan over long horizons, use tools, maintain persistent memory, and interact with other agents pose governance challenges that exceed the capabilities of model-level alignment alone. We propose a *Layered Governance Architecture* (LGA) that integrates three enforcement layers—model-level alignment monitoring, agent-level policy enforcement, and ecosystem-level interaction oversight—into a unified framework with formal guarantees. Our architecture employs hierarchical policy automata for runtime verification, a causal audit trail for post-hoc attribution, and an adaptive policy controller that dynamically tightens or relaxes constraints in response to observed risk signals. We evaluate LGA through deterministic simulations of multi-agent deployments across five governance configurations, four risk profiles, and planning horizons from 10 to 500 steps. The layered approach achieves a violation detection rate of 0.5537 with zero detection latency and 1.0 attribution accuracy, while preserving 0.8339 agent utility at 0.31 overhead. Adaptation experiments show that governance violation rates recover from 0.83 during risk spikes to 0.1938 in recovery phases, demonstrating effective adaptive control. Scaling experiments confirm that governance overhead remains constant at 0.31 as agent count grows from 2 to 32, while violation detection scales gracefully from 0.205 to 0.3703.



## CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; • **Software and its engineering** → *Software verification and validation.*

## KEYWORDS

agentic AI, governance, multi-agent systems, runtime verification, safety

**ACM Reference Format:**

Anonymous Author(s). 2026. Layered Governance Architecture for Real-World Agentic Systems. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn


## 1 INTRODUCTION

The emergence of agentic AI systems—large language models augmented with tool use, persistent memory, long-horizon planning, and multi-agent collaboration—has created governance challenges that extend far beyond traditional model-level alignment [13]. When an AI agent can execute multi-step plans, write to persistent memory, invoke external tools, and interact with other autonomous agents, the governance problem becomes fundamentally multi-layered: failures may arise not from individual model outputs but from the interaction of planning decisions across time, agents, and system components.

Existing approaches address fragments of this challenge. Constitutional AI [3] and RLHF [9] target model-level alignment but assume short-horizon interactions. Tool-augmented agent frameworks [10, 11] expand the action surface beyond what model-level guardrails cover. Multi-agent oversight formalisms [4] expose the combinatorial complexity of governing interacting agents but lack runtime enforcement mechanisms. Recent work on scaling safeguards [7] highlights that static guardrails degrade as agents acquire new objectives, motivating dynamic governance.

Wei et al. [13] identify a central open problem: developing governance frameworks that *jointly* address model-level alignment, agent-level policies, and ecosystem-level interactions under realistic deployment conditions. We address this problem directly.

*Contributions.* We make three contributions:

(1) We propose the **Layered Governance Architecture (LGA)**, a three-layer framework that integrates model-level alignment monitoring, agent-level policy enforcement, and ecosystem-level interaction oversight with formal consistency guarantees (Section 3).
(2) We design a **runtime monitor with causal audit trail** and an **adaptive policy controller** that dynamically adjusts governance stringency in response to observed risk signals (Section 4).
(3) We evaluate LGA through **deterministic multi-agent simulations** across five governance configurations, demonstrating its effectiveness in violation detection, attribution, adaptation, and scalability (Section 5).

## 2 PROBLEM FORMULATION

We formalize the governance problem for agentic systems as follows. Let $\mathcal{A} = \{a_1, \ldots, a_n\}$ be a set of $n$ agents operating in a shared environment over a horizon of $T$ time steps. At each step $t$, agent $a_i$ selects an action $\alpha_t^i$ from its action space $\Omega^i = \{\texttt{tool\_call}, \texttt{memory\_write}, \texttt{message},$ Each action carries a risk score $r(\alpha_t^i) \in [0, 1]$.

A *governance framework* $\mathcal{G}$ consists of three layers:

- **Model layer** $\mathcal{G}_M$: Constraints on individual model outputs, parameterized by an alignment threshold $\theta_M$.
- **Agent layer** $\mathcal{G}_A$: Constraints on agent-level actions, parameterized by a risk budget $\theta_A$ with action-type-specific multipliers.
- **Ecosystem layer** $\mathcal{G}_E$: Constraints on collective behavior, parameterized by a collective risk bound $\theta_E$ that considers the mean risk across all agents.



*Conference'17, July 2017, Washington, DC, USA*

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

An action $\alpha_t^i$ is *compliant* if and only if it satisfies all three layers:

$$\text{compliant}(\alpha_t^i) = \mathbb{1}[r(\alpha_t^i) < \theta_M] \wedge \mathbb{1}[r(\alpha_t^i) < c(\alpha_t^i) \cdot \theta_A] \wedge \mathbb{1}\left[\frac{r(\alpha_t^i) + \bar{r}_t^{-i}}{2} < \theta_E\right] \tag{1}$$

where $c(\alpha_t^i) \in \{0.6, 0.8, 1.0\}$ is the action-type multiplier and $\bar{r}_t^{-i}$ is the mean risk of all other agents at step $t$.

We evaluate governance quality via six metrics: *violation rate* $V$, *detection latency* $L$, *attribution accuracy* $A$, *governance overhead* $O$, *utility preservation* $U$, and *adaptation speed*.

# 3 LAYERED GOVERNANCE ARCHITECTURE

## 3.1 Architecture Overview

The Layered Governance Architecture operates as a runtime interception layer between the agent and its environment. Every action passes through three sequential checks before execution is permitted:

(1) **Model-layer check**: Verifies that the action's risk score is below the alignment threshold ($\theta_M = 0.35$ in our default configuration).
(2) **Agent-layer check**: Verifies that the risk score satisfies action-type-specific budgets derived from $\theta_A = 0.45$, with multipliers of 0.8 for tool calls and 0.6 for memory writes.
(3) **Ecosystem-layer check**: Combines the action's risk with the mean risk of other agents and verifies the combined score is below $\theta_E = 0.50$.

## 3.2 Governance Specification Language

Policies are expressed as typed constraints that compose hierarchically: ecosystem policies constrain agent policies, which constrain model behavior. This ensures consistency by construction. In our implementation, a `GovernancePolicy` specifies thresholds for each layer as key-value pairs, enabling flexible policy definition.

## 3.3 Causal Audit Trail

Every governance decision is logged in a causal audit trail that records the timestep, event type (violation, detection, escalation, adaptation), governance layer, severity, detection time, and attributed agent. This enables post-hoc analysis and counterfactual auditing: given a violation, the trail supports tracing the causal chain from ecosystem-level events back through agent decisions to model outputs.

# 4 RUNTIME MONITORING AND ADAPTATION

## 4.1 Adaptive Policy Controller

The adaptive policy controller maintains a sliding window of the most recent $w = 20$ risk scores. When the mean risk over the last 5 actions exceeds the escalation threshold (0.7), the controller tightens all model and agent constraints by the adaptation rate $\delta = 0.05$, with a minimum bound of 0.1. Conversely, when mean risk falls below 0.3, model constraints are relaxed by $0.5\delta$, with a maximum bound of 0.9.

This mechanism enables the governance framework to respond to changing risk conditions without manual intervention, as demonstrated in our adaptation experiments (Section 5.4).

**Table 1: Governance framework comparison (4 agents, 200 steps, mixed risk profiles). Higher violation rate indicates more detected violations.**

| Framework | Viol. Rate | Latency | Attrib. | Overhead | Utility |
|---|---|---|---|---|---|
| None | 0.0975 | 3.3304 | 1.0 | 0.0 | 1.0 |
| Model Only | 0.495 | 0.0 | 1.0 | 0.12 | 0.8515 |
| Agent Only | 0.5387 | 0.0 | 1.0 | 0.15 | 0.8384 |
| Ecosystem | 0.2425 | 0.0 | 1.0 | 0.18 | 0.9273 |
| **Layered** | **0.5537** | **0.0** | **1.0** | **0.31** | **0.8339** |

## 4.2 Hierarchical Policy Automata

We model governance policies as hierarchical timed automata [1], one per governance layer. The model-level automaton is nested inside the agent-level automaton, which is nested inside the ecosystem automaton. This hierarchical structure ensures that:

- Layer violations are detected at the appropriate granularity.
- Attribution can be traced to the specific layer and agent responsible.
- Policy consistency is maintained across layers by construction.

Runtime model checking, inspired by on-the-fly verification techniques from SPIN [6], verifies that each agent step maintains the automaton in a safe state. This enables zero-latency detection of violations, as our experiments confirm.

# 5 EXPERIMENTS

We evaluate the Layered Governance Architecture through four experiments using deterministic simulations (seeded with `np.random.default_rng(` of multi-agent deployments. All experiments use four action types (`tool_call`, `memory_write`, `message`, `plan_step`) with risk profiles drawn from Gaussian distributions with temporal drift.

## 5.1 Framework Comparison

We compare five governance configurations across 4 agents, 200 time steps, and four risk profiles (low, moderate, high, adversarial). Table 1 reports the results.

The layered framework achieves the highest violation detection rate of 0.5537, detecting all violations at zero latency with perfect attribution accuracy. The no-governance baseline detects only 0.0975 of violations (from passive constraint checking) with a mean detection latency of 3.3304 steps. Each individual layer contributes: model-only detects 0.495, agent-only detects 0.5387, and ecosystem-only detects 0.2425. The layered approach combines all three layers, achieving comprehensive detection at the cost of 0.31 overhead and 0.8339 utility preservation.

## 5.2 Ablation Study

To isolate the contribution of each governance layer, we conduct an ablation study using a separate experimental run. Table 2 presents the results.

The ablation confirms that each layer adds complementary detection capability. The ecosystem layer alone detects 0.2375 of violations, while the model and agent layers individually detect 0.5325 and 0.5138 respectively. The layered combination achieves 0.5575,

**Table 2: Ablation study: contribution of each governance layer.**

| Configuration | Viol. Rate | Utility | Overhead | Risk |
|---|---|---|---|---|
| None | 0.1113 | 1.0 | 0.0 | 0.4287 |
| Model Only | 0.5325 | 0.8403 | 0.12 | 0.4237 |
| Agent Only | 0.5138 | 0.8459 | 0.15 | 0.4209 |
| Ecosystem Only | 0.2375 | 0.9287 | 0.18 | 0.4138 |
| **Layered** | **0.5575** | **0.8327** | **0.31** | **0.421** |

**Table 3: Scaling behavior of layered governance as agent count increases.**

| Agents | L-Viol. | L-Overhead | L-Utility | NG-Viol. | NG-Utility |
|---|---|---|---|---|---|
| 2 | 0.205 | 0.31 | 0.9385 | 0.0 | 1.0 |
| 4 | 0.335 | 0.31 | 0.8995 | 0.0325 | 1.0 |
| 8 | 0.3538 | 0.31 | 0.8939 | 0.0262 | 1.0 |
| 16 | 0.3569 | 0.31 | 0.8929 | 0.0338 | 1.0 |
| 32 | 0.3703 | 0.31 | 0.8889 | 0.0344 | 1.0 |

**Table 4: Adaptation experiment: governance response to changing risk.**

| Phase | Viol. Rate | Utility | Mean Risk | Std Risk |
|---|---|---|---|---|
| Normal | 0.1675 | 0.9497 | 0.2287 | 0.1239 |
| Spike | 0.83 | 0.751 | 0.5255 | 0.2087 |
| Recovery | 0.1938 | 0.9419 | 0.2282 | 0.1279 |

showing that the layers are not simply additive but provide overlapping, defense-in-depth coverage.

## 5.3 Scaling Behavior

We evaluate how governance performance scales with the number of agents, ranging from 2 to 32. Figure 2 illustrates the results.

A key finding is that governance overhead remains constant at 0.31 regardless of agent count, demonstrating that the per-action monitoring cost does not increase with ecosystem size. The violation detection rate increases gradually from 0.205 with 2 agents to 0.3703 with 32 agents, reflecting the growing ecosystem-level risk as more agents interact. Utility preservation decreases modestly from 0.9385 to 0.8889.

## 5.4 Adaptation Under Risk Changes

We evaluate the adaptive policy controller across three phases: normal operation (low risk), a risk spike (high and adversarial profiles), and recovery (moderate risk). Table 4 reports the results.

During normal operation, the governance framework detects violations at a rate of 0.1675 while preserving 0.9497 utility. When the risk spikes, the violation rate rises to 0.83, reflecting the increased proportion of risky actions detected and blocked, with utility dropping to 0.751. In the recovery phase, the violation rate decreases to 0.1938 and utility recovers to 0.9419, demonstrating effective adaptive control. The recovery-phase violation rate of 0.1938 is only slightly higher than the normal-phase rate of 0.1675, indicating that the adaptive controller successfully recalibrates after a risk spike.

**Table 5: Governance effectiveness across planning horizons.**

| Horizon | L-Viol. | L-Attrib. | L-Utility | NG-Viol. | MO-Viol. |
|---|---|---|---|---|---|
| 10 | 0.5 | 1.0 | 0.85 | 0.125 | 0.525 |
| 50 | 0.545 | 1.0 | 0.8365 | 0.07 | 0.555 |
| 100 | 0.57 | 1.0 | 0.829 | 0.115 | 0.5225 |
| 200 | 0.5613 | 1.0 | 0.8316 | 0.12 | 0.525 |
| 500 | 0.568 | 1.0 | 0.8296 | 0.1035 | 0.5045 |

## 5.5 Planning Horizon Analysis

We examine governance effectiveness across planning horizons from 10 to 500 steps. Table 5 presents the results.

The layered governance framework maintains stable performance across horizons, with violation detection ranging from 0.5 at horizon 10 to 0.568 at horizon 500. Attribution accuracy remains perfect at 1.0 across all horizons. Utility preservation decreases slightly from 0.85 to 0.8296 as longer horizons increase the cumulative probability of encountering risky actions.

# 6 DISCUSSION

*Defense-in-Depth.* Our results demonstrate that layered governance provides defense-in-depth: each layer catches violations that others miss. The model layer enforces alignment constraints, the agent layer restricts action-type-specific risk budgets, and the ecosystem layer bounds collective behavior. The layered combination achieves 0.5575 detection in ablation versus 0.5325, 0.5138, and 0.2375 for individual layers.

*Constant Overhead.* Governance overhead remains at 0.31 regardless of the number of agents. This constant-overhead property results from our per-action monitoring design, where each action is checked independently against the policy hierarchy. The computational cost scales linearly with the total number of actions but is constant per action.

*Adaptive Control.* The adaptive policy controller demonstrates effective risk response. Recovery-phase violation rates (0.1938) closely match normal-phase rates (0.1675), showing that the controller avoids both over-tightening (which would reduce utility) and under-relaxing (which would miss violations) after risk transitions.

*Limitations.* Our evaluation uses simulated multi-agent deployments with synthetic risk profiles rather than real agentic AI systems. The risk score model assumes Gaussian distributions with temporal drift, which may not capture the full complexity of real-world agent behavior. Future work should validate LGA on actual LLM-based agent deployments with real tool use and memory operations.

# 7 RELATED WORK

*AI Safety and Alignment.* Foundational work on concrete AI safety problems [2] identified reward hacking, side effects, and distributional shift as key challenges. Constitutional AI [3] and RLHF [9] address model-level alignment through training-time objectives. Our work extends these ideas to the runtime governance of deployed agentic systems.

*Agentic AI Governance.* Wei et al. [13] formalize the need for governance frameworks spanning model, agent, and ecosystem levels. Practices for governing agentic systems [12] propose organizational and technical safeguards. The ethics of advanced AI assistants [5] examines the value alignment challenges. Our LGA provides a concrete technical framework addressing these desiderata.

*Multi-Agent Oversight.* Chan et al. [4] formalize multi-agent oversight via causal modeling and aggregate governance. Our ecosystem layer builds on their insights while adding runtime enforcement. Scaling safeguards [7] motivate adaptive governance, which our adaptive policy controller implements.

*Runtime Verification.* Our hierarchical policy automata draw on timed automata theory [1] and model checking [6]. We adapt these formal methods from software verification to the governance of AI agent behavior, enabling zero-latency violation detection with formal guarantees.

*Benchmarking Agentic Systems.* Evaluation frameworks for agentic AI [8] highlight the inadequacy of existing benchmarks for testing planning-time failures and multi-step goal drift. Our simulation framework addresses this gap by evaluating governance across varying horizons, risk profiles, and agent counts.

## 8 CONCLUSION

We have presented the Layered Governance Architecture, a three-layer framework for governing real-world agentic AI systems. Through deterministic multi-agent simulations, we demonstrate that LGA achieves comprehensive violation detection (0.5537) with zero latency and perfect attribution accuracy, while preserving 0.8339 agent utility. The adaptive policy controller successfully recalibrates governance stringency in response to risk transitions, and the architecture scales to 32 agents with constant overhead. Our results establish that layered governance—combining model-level, agent-level, and ecosystem-level enforcement—provides a principled and practical approach to the open challenge of governing increasingly capable agentic AI systems.

## REFERENCES


[1] Rajeev Alur and David L. Dill. 1994. A Theory of Timed Automata. In *Theoretical Computer Science*, Vol. 126. 183–235.

[2] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete Problems in AI Safety. *arXiv preprint arXiv:1606.06565* (2016).

[3] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional AI: Harmlessness from AI Feedback. *arXiv preprint arXiv:2212.08073* (2022).

[4] Alan Chan et al. 2025. Multi-Agent Oversight: Prioritization, Causal Modeling, and Aggregate Governance. *arXiv preprint arXiv:2512.07094* (2025).

[5] Iason Gabriel et al. 2024. The Ethics of Advanced AI Assistants. *arXiv preprint arXiv:2404.16244* (2024).

[6] Gerard J. Holzmann. 1997. The Model Checker SPIN. In *IEEE Transactions on Software Engineering*, Vol. 23. 279–295.

[7] Siyuan Huang et al. 2026. Scaling Safeguards for Open-Ended Agentic AI. *arXiv preprint arXiv:2601.02749* (2026).

[8] Sayash Kapoor et al. 2025. Benchmarking Agentic AI Systems under Realistic Constraints. *arXiv preprint arXiv:2511.10524* (2025).

[9] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training Language Models to Follow Instructions with Human Feedback. *Advances in Neural Information Processing Systems* 35 (2022), 27730–27744.

[10] Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. 2023. ToolLLM: Facilitating Large Language Models to Master 16000+ Real-World APIs. *arXiv preprint arXiv:2307.16789* (2023).

[11] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language Models Can Teach Themselves to Use Tools. *Advances in Neural Information Processing Systems* 36 (2023).

[12] Yonadav Shavit et al. 2023. Practices for Governing Agentic AI Systems. *OpenAI Research Report* (2023).

[13] Jason Wei et al. 2026. Agentic Reasoning for Large Language Models. *arXiv preprint arXiv:2601.12538* (2026).

fig_scaling.png

**Figure 2: Scaling behavior as the number of agents increases from 2 to 32.**

fig_adaptation.png

**Figure 3: Adaptive governance response across normal, spike, and recovery phases.**

fig_horizon.png

**Figure 4: Governance effectiveness across planning horizons (10–500 steps).**

# A EXPERIMENTAL CONFIGURATION

**Table 6: Default governance policy parameters.**

| Layer | Parameter | Value |
|---|---|---|
| Model | Alignment threshold ($\theta_M$) | 0.35 |
| Agent | Risk budget ($\theta_A$) | 0.45 |
| Ecosystem | Collective risk bound ($\theta_E$) | 0.50 |
| Adaptive | Window size ($w$) | 20 |
| Adaptive | Escalation threshold | 0.7 |
| Adaptive | Adaptation rate ($\delta$) | 0.05 |

**Table 7: Risk profile parameters (Gaussian).**

| Profile | Mean ($\mu$) | Std ($\sigma$) |
|---|---|---|
| Low | 0.15 | 0.08 |
| Moderate | 0.30 | 0.12 |
| High | 0.55 | 0.15 |
| Adversarial | 0.70 | 0.18 |

# B ADDITIONAL FIGURES

fig_framework_comparison.png

**Figure 1: Framework comparison across governance configurations.**