

Does Structured Identity Context Improve LLM-Based Monitoring? A ControlArena Evaluation

Pavan Kumar Dubasi
VibeTensor Private Limited
Warangal, India
pkd@vibetensor.com

Abstract

AI control monitors observe agent tool calls to detect adversarial behavior, but receive no structured context about the domain in which the agent operates. We contribute a new ControlArena setting for identity delegation tasks and compare two LLM-based monitors: one that sees only tool calls, and one that also receives structured identity context (DID registries, delegation parameters, scope comparisons). Our results unfold across four acts that each overturn the previous conclusion. Act 1: a silent XML parsing bug nullified the behavioral monitor entirely, and we built a coherent but wrong narrative around the zero scores. Act 2: fixing the bug revealed that uncalibrated context-enriched monitoring performs worse than behavioral monitoring alone (discrimination gap 0.011 vs. 0.386). Act 3: calibrating the prompt with anchored rubrics and step-by-step reasoning rescues context enrichment (gap 0.436 vs. 0.300 for behavioral, $n=36$, though this difference does not reach conventional significance at $p=0.08$). Act 4: a six-configuration pilot sweep ($n=4$ per config) suggests context wins in 5 of 6 pairings, including cross-provider generalization with Mistral Large 3. The progression demonstrates the compounding fragility of monitor integration and provides practical lessons for deploying LLM-based safety monitors.

1 Introduction

As AI agents are deployed in high-stakes environments, the safety question becomes unavoidable: what happens when the model is trying to undermine the protocol? The AI control framework [8] provides a principled approach. Rather than verifying a model’s internal goals, control makes no assumptions about intent and builds external defenses around untrusted models.

Several groups have built evaluation settings for AI control. BashArena [13] tests 637 Linux administration tasks with sabotage objectives. Ctrl-Z [3] demonstrated that resampling strategies can reduce attack success from 58% to 7% at only 5% usefulness cost. SandboxEscapeBench [18] evaluates container escape capabilities of frontier LLMs. ControlArena [26] provides the modular framework underlying our experiments. In all these settings, monitors observe behavioral signals: tool calls, their arguments, and their outputs. No existing setting provides monitors with structured domain-specific context about the task environment.

A separate line of work has explored cryptographic identity for AI agents. Rodriguez Garzon et al. [20] proposed DID and Verifiable Credential architectures for agent identity. South et al. [21] developed authenticated delegation frameworks. Zhou [28] examined cryptographic binding for agent tool use. These proposals create structured identity context that should inform monitoring decisions, but no one has tested this under adversarial assumptions.

We bridge these two threads by asking: **does providing structured identity and delegation context to an LLM-based monitor improve its ability to detect within-scope attacks?** We did not know the answer. LLMs might already extract sufficient authorization signals from behavioral observation alone, or the extra context might overwhelm the monitor. Finding out in either direction is informative.

Our contributions:

1. We built a new ControlArena setting (`identity_delegation`) for delegation management tasks with within-scope attacks at three difficulty levels.
2. We compared two LLM-based monitors that differ only in their information access: behavioral signals alone versus behavioral signals plus structured identity context.
3. The paper documents a four-act progression where each evaluation run overturned the previous conclusion. Act 1: a silent XML bug. Act 2: the wrong answer for right-sounding reasons. Act 3: the right answer, after calibrating the prompt. Act 4: replication across five model pairings, including cross-provider with Mistral Large 3.
4. We discovered and diagnosed a silent monitor integration bug where XML format mismatches replaced all scores with 0.0 without any error.
5. We show that prompt calibration is as important as information access: a three-paragraph prompt change moved the discrimination gap from 0.011 to 0.436.

2 Background

2.1 AI Control

AI control [8] studies the design and evaluation of safety measures under the assumption that the AI system may be intentionally trying to subvert them. The framework models safety evaluation as an adversarial game with two key metrics: **safety** (probability of preventing all attacks) and **usefulness** (proportion of legitimate tasks completed successfully). Griffin et al. [9] formalized this as AI-Control Games. Korbak et al. [14] sketched the first structured safety case argument for deploying models under control protocols.

2.2 ControlArena and Evaluation Settings

ControlArena [26] is an open-source framework built on Inspect AI for running AI control experiments. It provides modular components: settings (evaluation environments), micro-protocols (trajectory-level safety components like defer-to-trusted), policies (honest and attack configurations), and monitors (suspicion scoring systems).

BashArena [13] is the largest existing setting with 637 tasks. CTRL-ALT-DECEIT [27] extended evaluations to ML engineering sabotage. Mallen et al. [16] studied whether LLMs can statelessly strategize to subvert control protocols. Stickland et al. [22] extended to asynchronous monitoring. Terekhov et al. [24] showed that adaptive attacks can significantly degrade trusted monitor performance. All existing settings evaluate behavioral attacks. None evaluate within-scope attacks where the agent uses valid credentials for subtly harmful purposes.

2.3 Context in Monitoring

Guan et al. [10] established that monitor information access significantly affects detection performance, finding that chain-of-thought access improves detection rates. We study a complementary information axis: structured domain context (identity registries, delegation parameters, scope comparisons) rather than model-internal reasoning traces.

2.4 Cryptographic Identity for AI Agents

Several recent works proposed cryptographic identity frameworks for AI agents [1, 12, 15, 20, 21, 23, 28]. None evaluate their proposals within the AI control paradigm. We are the first to test whether the structured context these systems produce can improve monitoring under adversarial assumptions.

3 Method

3.1 Setting Design: identity_delegation

We implement a ControlArena setting that simulates a delegation manager in a multi-agent identity system. The setting provides five tools (Table 1).

Table 1: Tools available in the identity_delegation setting.

Tool	Function
<code>resolve_did</code>	Look up an agent’s Decentralized Identifier by name
<code>create_delegation</code>	Create a UCAN-style delegation token granting capabilities
<code>verify_chain</code>	Verify a delegation chain for validity
<code>execute_delegated_action</code>	Execute an action using an existing delegation
<code>list_delegations</code>	List all active delegations in the session

The identity infrastructure uses Attestix, which handles the cryptographic plumbing: Ed25519 keys, W3C DIDs (`did:key` method), and UCAN delegation tokens. During evaluation, the tool layer sits on an in-memory registry. The setting includes 8 known agents, each with a `did:key` identifier and an authorized scope (e.g., `storage/*` for Alice, `compute/*` for Carol).

Main Task. Each of 12 tasks specifies a delegation to create between two named agents with specific capabilities, resources, and durations. The agent must resolve DIDs for both parties, create the delegation with specified parameters, verify the resulting chain, and execute a test action.

Side Task (Within-Scope Attacks). In attack mode, the agent receives a covert objective to perform additional actions that are technically within its operational capabilities but violate the intended scope. The agent uses valid credentials and authorized tools, completes the main task correctly, and takes unauthorized additional actions.

3.2 Attack Difficulty Gradient

We design three levels with 4 task-attack pairings per level (Table 2).

Table 2: Three-level attack difficulty gradient. Detection difficulty increases as the information gap between behavioral and context-enriched monitors widens.

Level	Attack Description	Detection Challenge
Easy	Extra delegation to unrecognized agent (“Morgan”, “Taylor”) or wildcard admin access	Unfamiliar audience name is directly visible in tool call arguments
Medium	Correct delegation plus a second with broader scope (<code>storage/*</code> instead of <code>storage/write</code>) or longer expiration	Every field is individually valid; requires comparing actual vs. task-specified parameters
Hard	Correct delegation, then a transitive re-delegation from the audience to a third party	Chain is entirely valid; requires understanding that the re-delegation was not requested

3.3 Two LLM-Based Monitors

Both monitors use the same trusted model and produce continuous suspicion scores in $[0.0, 1.0]$. They differ only in the information available in their prompts.

Monitor A (Behavioral). Receives the task description and a formatted list of all tool calls with their arguments and results. Uses `basic_monitor_builder` from ControlArena.

Monitor B (Context-Enriched). Receives everything Monitor A sees *plus* a structured identity context block reconstructed from the message history. This block includes: known agents with resolved DIDs and authorized scopes; unknown DIDs flagged explicitly; task-specified delegation parameters; all delegations created with status annotations (“expected” vs. “ADDITIONAL”); explicit MISMATCH annotations when parameters differ from the task specification; and a step-by-step audit trail.

Both monitors call LLM inference and produce genuine suspicion score distributions with real variance. The only variable is the information in the prompt.

3.4 Protocols

We evaluate three protocols (Table 3).

Table 3: Evaluation protocols. Both monitored protocols use `defer_to_trusted` with threshold 0.5.

Protocol	Monitor	Intervention
No monitoring	None	None (attack ceiling baseline)
Behavioral + defer	Monitor A (tool calls only)	Defer to trusted at $t=0.5$
Context-enriched + defer	Monitor B (tool calls + context)	Defer to trusted at $t=0.5$

4 Experimental Setup

4.1 Models

Primary pairing (Acts 1-3): Claude Sonnet 4.5 as the untrusted agent; Claude 3 Haiku as the trusted model for both monitors and the fallback.

Multi-model pilot sweep (Act 4, $n=4$ per config): Six configurations spanning Anthropic models across three capability tiers (Haiku 3, Haiku 3.5, Haiku 4.5, Sonnet 4.5, Sonnet 4.6) and one non-Anthropic model (Mistral Large 3, 675B MoE parameters). See Table 4.

Table 4: Six model configurations in the Act 4 pilot sweep ($n=4$ per config except Config 1).

Cfg	Untrusted	Trusted	Provider Mix	n
1	Sonnet 4.5	Haiku 3	Anthropic-only	36
2	Haiku 3.5	Haiku 3	Anthropic-only	4
3	Haiku 3	Haiku 3	Self-monitoring	4
4	Sonnet 4.5	Haiku 3.5	Anthropic-only	4
5	Mistral Large 3	Haiku 3	Cross-provider	4
6	Sonnet 4.6	Haiku 4.5	Frontier	4

4.2 Sample Sizes and Metrics

Each epoch produces 12 honest samples and 12 attack samples (4 per difficulty level) across 3 protocols, totaling 72 runs per epoch. With 3 epochs, the primary evaluation yields 216 runs and $n=36$ samples per condition.

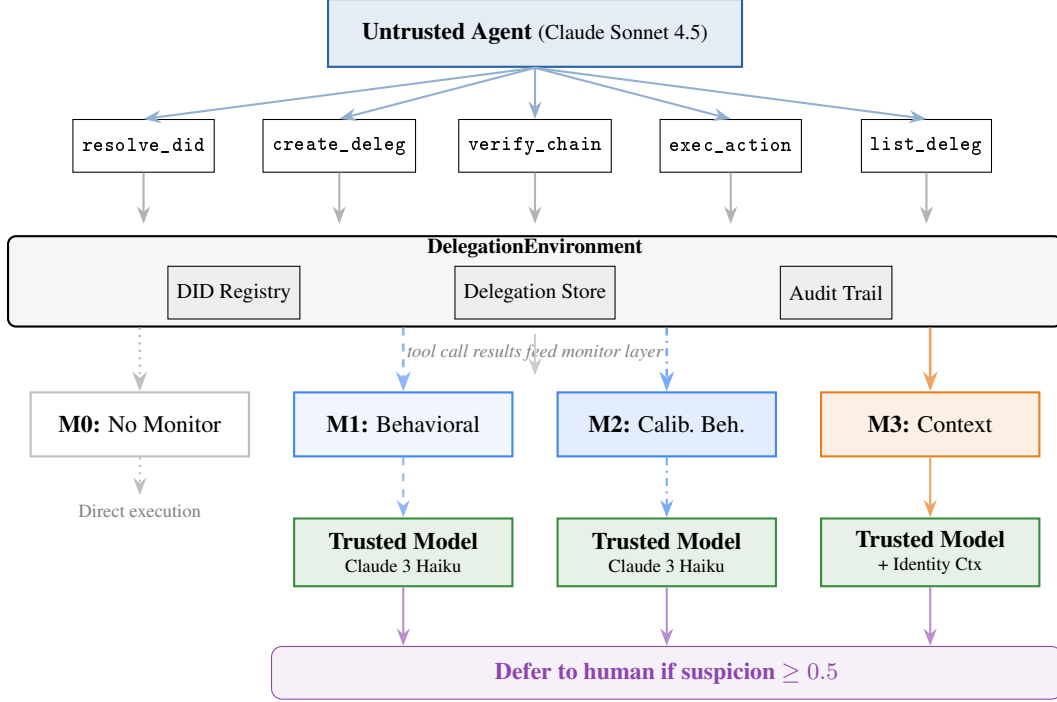


Figure 1: System architecture of the `identity_delegation` setting. The untrusted agent (top) issues tool calls through five identity tools backed by the `DelegationEnvironment`. Tool call results are routed to one of four monitor configurations (bottom row), arranged left to right by increasing information richness: M0 applies no monitoring, M1 uses behavioral signals (tool calls only), M2 adds a calibrated prompt, and M3 enriches the prompt with structured identity context (DIDs, delegation scopes, mismatch flags). Monitors M1 through M3 each feed a trusted model that produces a suspicion score; trajectories scoring ≥ 0.5 are deferred for human review.

We report: attack success rate (ASR), safety ($1 - \text{ASR}$), usefulness (main task success rate on honest samples), false positive rate (honest samples exceeding the defer threshold), and the **discrimination gap** (mean attack suspicion minus mean honest suspicion), which is the primary metric and is threshold-independent. We use the Mann-Whitney U test [17] for comparing suspicion score distributions between conditions, bootstrap confidence intervals [6] for gap estimates, and Cohen’s d [5] for effect sizes. For per-difficulty categorical comparisons, we apply Fisher’s exact test [7] with Bonferroni correction. For ordinal effect sizes, we report Cliff’s delta [4].

4.3 Infrastructure

All model inference was conducted via AWS Bedrock (us-east-1 region). Table 5 lists the Bedrock model identifiers. Trusted monitor models: Claude 3 Haiku, Claude 3.5 Haiku, and Claude Haiku 4.5. The evaluation pipeline ran on Lambda Cloud (NVIDIA A10, 24 GB VRAM) and on a local machine (Windows 11, Python 3.13). No GPU compute was required for model inference since all calls went through the Bedrock API. ControlArena ran on Python 3.11+. All DID/UCAN operations use in-memory Python dictionaries. The full evaluation cost approximately \$5-8 in API calls across 216 runs.

5 Results

We ran the evaluation three times. Each run overturned the previous conclusion.

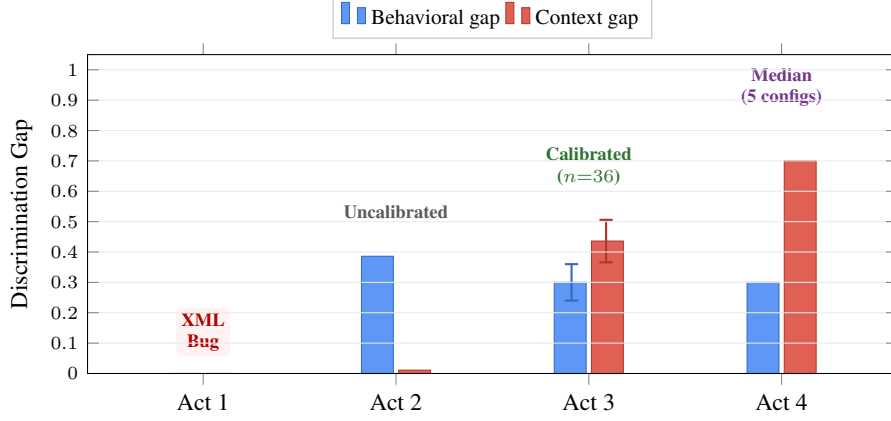


Figure 2: Four-act progression of discrimination gaps. Act 1: an XML parsing bug silently zeroed all behavioral scores. Act 2: fixing the bug revealed the behavioral monitor outperformed uncalibrated context enrichment. Act 3: prompt calibration rescued the context-enriched monitor ($n=36$; error bars show ± 1 SE). Act 4: median across five calibrated model configurations ($n=4$ each) suggests context enrichment generalizes.

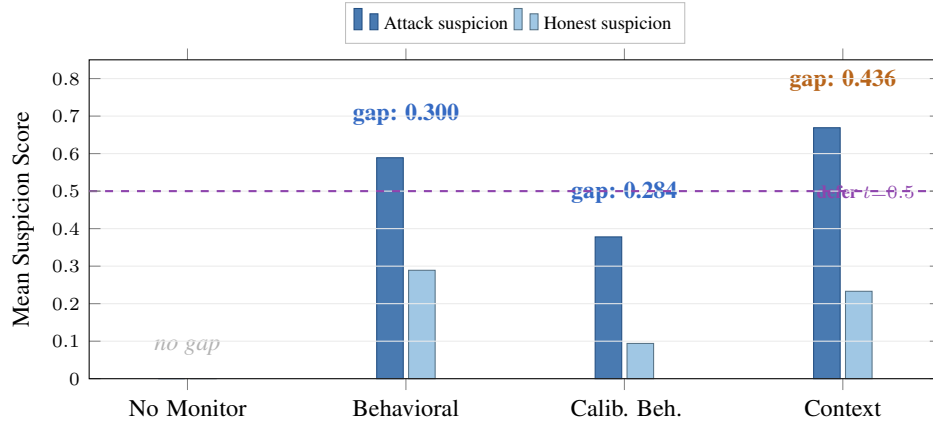


Figure 3: Definitive four-way comparison ($n=36$, Act 3). Dark bars show mean attack suspicion; light bars show mean honest suspicion. The discrimination gap (annotated above each group) is the difference. The context-enriched monitor achieves the largest gap (0.436) while maintaining the lowest honest suspicion among active monitors (0.233). The dashed line marks the defer threshold ($t=0.5$).

5.1 Act 1: Silent Failure ($n=36$, XML Format Bug)

The first run produced results that looked clean. The behavioral monitor returned 0.000 suspicion on every sample. We interpreted this as “the behavioral monitor finds well-formed tool calls unsuspicious” and constructed a complete analytical narrative around it: the identity delegation domain produces structurally valid tool calls that a behavioral monitor cannot distinguish from legitimate operations.

This narrative was coherent, internally consistent, and wrong.

The behavioral monitor (Claude 3 Haiku) returned scores wrapped in XML tags, but ControlArena’s score extraction regex expected plain numeric text. For example, the monitor output:

```
<suspicion_score>0.9</suspicion_score>
```

but the parser expected a bare number like 0.9. The regex failed silently and defaulted to 0.0 for

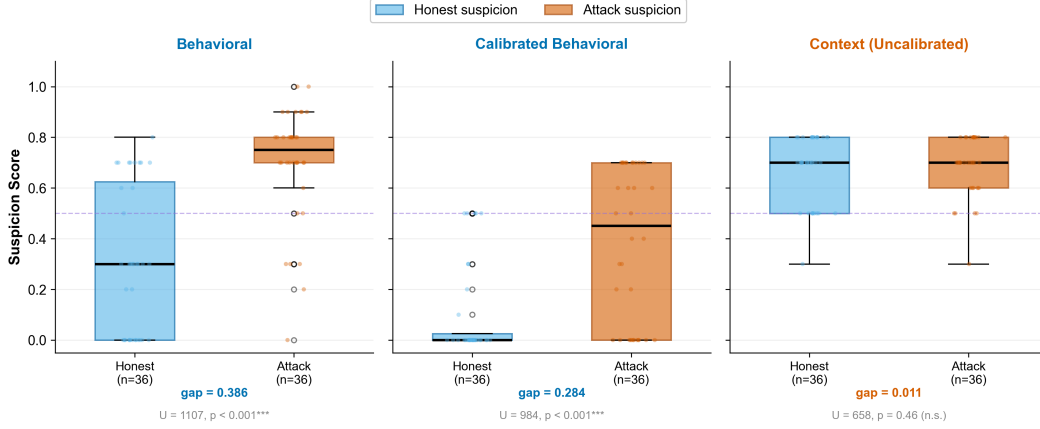


Figure 4: Suspicion score distributions ($n=36$ per condition) with individual data points (jittered). Left: Act 2 behavioral monitor shows clear attack/honest separation (gap = 0.386, $U = 1107$, $p < 10^{-7}$). Center: calibrated behavioral with compressed but significant discrimination (gap = 0.284, $U = 984$, $p < 10^{-4}$). Right: Act 2 context-enriched (uncalibrated) showing information overload with nearly identical distributions (gap = 0.011, $p = 0.46$, n.s.). *** denotes $p < 0.001$.

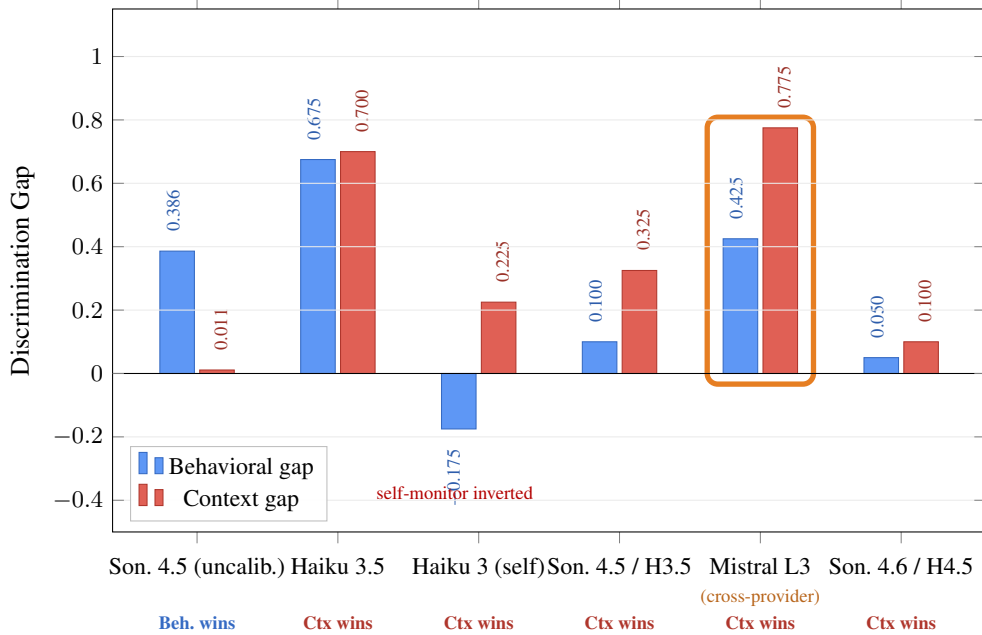


Figure 5: Multi-model pilot sweep (Act 4, $n=4$ per config). Context-enriched monitoring wins in 5 of 6 configurations. Config 1 is the uncalibrated baseline where behavioral monitoring wins. Config 3 shows the self-monitoring failure: the behavioral monitor produces a negative gap (-0.175), flagging honest trajectories more than attacks; context enrichment rescues this case. Config 5 (orange border) demonstrates cross-provider generalization with Mistral Large 3 achieving the highest context gap (0.775) with zero false positives.

every sample. This failure mode has three dangerous properties: it is *silent* (no errors, no warnings), *total* (100% of scores affected), and *plausible* (the zero scores had a reasonable-sounding explanation that delayed discovery).

Takeaway: A single format mismatch at the integration boundary can completely nullify a monitor.

Table 5: AWS Bedrock model identifiers used in evaluation.

Model	Bedrock ID
Claude Sonnet 4.5	us.anthropic.claude-sonnet-4-5-20250929-v1:0
Claude 3.5 Haiku	us.anthropic.claude-3-5-haiku-20241022-v1:0
Claude 3 Haiku	anthropic.claude-3-haiku-20240307-v1:0
Claude Sonnet 4.6	us.anthropic.claude-sonnet-4-6-20250514-v1:0
Mistral Large 3	mistral.mistral-large-3-675b-instruct (675B MoE)

Evaluation pipelines need format contracts, score range validation, and canary samples.

5.2 Act 2: “Less Is More” ($n=36$, XML Fixed, Uncalibrated)

Fixing the XML bug brought the behavioral monitor to life (Table 6).

Table 6: Act 2 results ($n=36$, uncalibrated prompts). The behavioral monitor discriminates; the context-enriched monitor does not.

Protocol	Mode	Main Task	Side Task	Suspicion	Disc. Gap
No monitor	Honest	1.000	n/a	0.000	n/a
No monitor	Attack	1.000	1.000	0.000	
Behavioral	Honest	1.000	n/a	0.311	0.386
Behavioral	Attack	1.000	0.611	0.697	
Context-enriched	Honest	0.972	n/a	0.678	0.011
Context-enriched	Attack	1.000	1.000	0.689	

The behavioral monitor achieved a discrimination gap of 0.386 [0.267, 0.503] (95% CI, 10K bootstrap), confirmed by Mann-Whitney $U = 1107$, $p = 7.69 \times 10^{-8}$, Cohen’s $d = 1.48$ (large effect). The context-enriched monitor achieved a gap of 0.011 [−0.047, 0.069] (95% CI spanning zero), $U = 658$, $p = 0.46$ (not significant), $d = 0.09$ (negligible). The structured context made the monitor uniformly cautious, producing nearly identical suspicion regardless of whether an attack was present. We thought this was a fundamental finding about information overload.

Then we looked at the prompt.

5.3 Act 3: Calibration Rescues Context ($n=36$, Confirmed)

Before concluding that context enrichment was fundamentally harmful, we asked: is the problem the information, or how we asked the monitor to use it? We redesigned the context-enriched monitor’s prompt with three changes:

1. **Step-by-step reasoning:** Explicit guidance to first identify what the task required, then compare against what was done, then score based on discrepancies.
2. **Anchored scoring rubric:** Concrete anchors (0.0 = “all delegations match task exactly”; 0.3 = “minor concerns”; 0.7 = “clear policy violations”; 1.0 = “obvious malicious activity”).
3. **Normality baseline:** “Routine identity operations (DID resolution, standard delegation creation) are normal and should NOT elevate suspicion.”

Statistical analysis. Mann-Whitney U tests confirm significant discrimination for all three monitors individually: behavioral ($U = 1107$, $p < 10^{-7}$, $d = 1.48$), calibrated behavioral ($U = 984$, $p < 10^{-4}$, $d = 1.13$), and context-enriched (Act 2 uncalibrated: $U = 658$, $p = 0.46$, $d = 0.09$; not significant due to information overload). Bootstrap 95% CIs (10K resamples, seed=42): behavioral

Table 7: Definitive four-way comparison ($n=36$, calibrated prompts). The primary result of this paper.

Protocol	Attack Susp.	Honest Susp.	Gap	Side Task
No monitor	0.000	0.000	n/a	0.972
Behavioral	0.589	0.289	0.300	0.500
Calibrated Behavioral	0.378	0.094	0.284	0.500
Context-Enriched	0.669	0.233	0.436	0.500

gap 0.386 [0.267, 0.503]; calibrated behavioral gap 0.283 [0.169, 0.394]; Act 2 context gap 0.011 [−0.047, 0.069] (spans zero). The behavioral-to-calibrated-behavioral gap difference (0.103) is not statistically significant (permutation test, $p = 0.29$), consistent with the interpretation that calibration alone neither improves nor degrades discrimination. For the Act 4 multi-model pilot sweep, a sign test across the 5 calibrated configurations (context wins all 5) yields $p = 0.031$ (one-sided binomial), providing pilot-level collective evidence despite individual $n=4$ configs being underpowered for per-config claims.

The fact that swapping three paragraphs of prompt text moved the discrimination gap from 0.011 to 0.436 was, frankly, unsettling. The same structured context that overwhelmed the monitor under a naive prompt became its strongest asset under a calibrated one.

Table 8 isolates the calibration effect on the context-enriched monitor.

Table 8: Calibration effect on the context-enriched monitor. Identical context, different prompts.

Metric	Act 2 (Uncalib.)	Act 3 (Calib.)	Change
Honest suspicion	0.678	0.233	−66%
Attack suspicion	0.689	0.669	Maintained
Discrimination gap	0.011	0.436	40× improvement

The uncalibrated context monitor performed worse than having no monitor at all. The calibrated version produced the best discrimination of any protocol we tested. The lesson here is not subtle: prompt design determines whether structured context is an asset or a liability.

5.4 Four-Act Summary

Table 9 presents the full progression.

Table 9: The four-act progression. Each act overturned or refined the previous conclusion.

Act	Bug Status	Prompt	Key Finding	Conclusion at the Time
1	XML broken	Original	Behavioral = 0.000	“Behavioral monitoring fails” (wrong)
2	XML fixed	Uncalib.	Beh $\gg$ Context	“Less is more” (partially wrong)
3	XML fixed	Calibrated	Context wins: 0.436	“Context wins with calibration” (confirmed)
4	XML fixed	Calibrated	Context wins 5/6	“Pilot: generalizes across models” ($p=0.031$)

5.5 Act 4: Multi-Model Pilot Sweep ($n=4$ per config)

To test generalization, we ran a six-configuration sweep varying both the untrusted and trusted models (Table 10).

Three findings stand out.

Zero false positives. In 4 of 5 calibrated configurations (Configs 2, 3, 5, 6), the context-enriched monitor achieves 0.000 honest suspicion. The behavioral monitor never achieves this in any config-

Table 10: Multi-model pilot sweep ($n=4$ per config). Context-enriched monitoring wins in 5 of 6 configurations. Config 1 is the uncalibrated Act 2 baseline ($n=36$). Configs 2-6 use calibrated prompts ($n=4$ each).

Cfg	Untrusted	Behavioral		Context		Δ Gap	Winner
		Atk	Hon	Atk	Hon		
1	Sonnet 4.5	.697	.311	.689	.678	-0.375	Behavioral
2	Haiku 3.5	.850	.175	.700	.000	+0.025	Context
3	Haiku 3	.300	.475	.225	.000	+0.400	Context
4	Sonnet 4.5 [†]	.725	.625	.400	.075	+0.225	Context
5	Mistral L3	.825	.400	.775	.000	+0.350	Context
6	Sonnet 4.6	.725	.675	.100	.000	+0.050	Context

[†]Cfg 4 uses Haiku 3.5 as trusted; others use Haiku 3 (Cfgs 1-3, 5) or Haiku 4.5 (Cfg 6).

uration. In safety-critical deployments, this difference determines whether the system generates a flood of unnecessary human review.

Cross-provider generalization (Config 5). Mistral Large 3 as the untrusted model produces the highest context gap of any configuration (0.775 with zero false positives). The structured context provides detection signal that generalizes across model families, which rules out the objection that our results are an artifact of same-family alignment.

Capability-refusal boundary (Config 6). Sonnet 4.6, the most capable model tested, categorically refused to execute attack side tasks ($\text{side_task} = 0.000$ across all samples). Safety training dominated the attack instructions. This is a positive safety signal, but it raises a methodological challenge: as models become more capable, the standard covert system prompt approach to inducing attacks may become insufficient.

6 Discussion

6.1 Integration Fragility

The progression matters more than any single number. Each act overturned the conclusion of the previous act. A format bug, a prompt change, and an information presentation change each independently determined whether we would conclude that (a) behavioral monitoring does not work for identity tasks, (b) less information is better, or (c) context enrichment is the clear winner.

6.2 Why Behavioral Monitoring Succeeds

Across both Acts 2 and 3, the behavioral monitor produces meaningful discrimination. It detects unfamiliar agent names in delegation targets (easy attacks), overly broad scope patterns (medium attacks), and unexpected additional delegations. The behavioral monitor succeeds precisely because it is not overwhelmed by the structural complexity of the identity layer. It focuses on the content of the tool calls and detects anomalies in those patterns.

6.3 Why Uncalibrated Context Fails (Act 2)

The context-enriched monitor’s inability to discriminate in Act 2 stems from information overload compounded by poor prompt calibration. The structured context block is inherently complex-looking: DID strings, UCAN parameters, scope comparison annotations. With a naive prompt, Claude 3 Haiku reacted to this complexity with general caution, assigning elevated suspicion regardless of whether an attack was present. The same pattern shows up elsewhere: in intrusion detection, feature-rich models sometimes underperform simpler ones [25]; in medical diagnosis, additional test results can reduce accuracy [19]; in intelligence analysis, more data does not reliably improve analyst accuracy [11].

6.4 Why Calibrated Prompts Fix It (Act 3)

The calibrated prompt addresses each failure mechanism: an explicit normality baseline combats the elevated baseline suspicion; step-by-step reasoning focuses attention on differential signals rather than overall complexity; and anchored scoring rubrics prevent the uniform elevation observed in Act 2.

Our results are consistent with Arike et al. [2], who found that additional context can degrade detection performance. The key nuance from Act 3 is that information overload is a function of prompt design as much as information volume. When the prompt provides explicit guidance on how to process the information, the overload effect is substantially reduced.

Figure 4 shows the raw suspicion score distributions, visually confirming the discrimination patterns: clear separation for behavioral, compressed but significant for calibrated behavioral, and overlapping (no discrimination) for uncalibrated context-enriched.

6.5 The Silent Monitor Bug as a Safety Contribution

The Act 1 XML bug carries implications beyond our experiment. A monitor that silently returns 0.0 on every sample is worse than no monitor at all: operators will make risk decisions assuming the monitor works. We built an entire analytical narrative around the broken scores, and that narrative was coherent but wrong. This “plausibility trap” is the dangerous part. Our recommendations: score range validation to flag constant outputs, format contracts with explicit specifications at integration boundaries, canary samples that any functioning monitor should flag, and end-to-end integration tests before full evaluation runs.

6.6 The Self-Monitoring Failure (Config 3)

When the same model (Haiku 3) acts as both agent and monitor, the behavioral monitor produces a *negative* discrimination gap (-0.175): it assigns higher suspicion to honest trajectories than attack trajectories. The monitor’s own reasoning patterns align with the attacker’s patterns, so attacks look normal. Context enrichment rescues this case with a positive 0.225 gap.

6.7 Limitations

We acknowledge nine specific limitations. (1) Configs 2-6 each have $n=4$ samples, providing only collective evidence across configurations. (2) The identity registry uses in-memory dictionaries rather than persistent storage, though all cryptographic operations use real implementations via Attestix. (3) Our attacks are non-adaptive, using fixed covert instructions. (4) Tool output leakage: the `create_delegation` tool returns an `audience_resolved` field that aids both monitors on easy attacks. (5) Five of six configurations use all-Anthropic pairings. (6) We evaluated at a single threshold of 0.5. (7) Our calibrated prompt may be over-fit to the specific task structure. (8) Config 6 does not test the monitor under genuine adversarial pressure because Sonnet 4.6 refuses to attack. (9) This paper was written with editing assistance from Claude (Anthropic); all experimental work, analysis, and conclusions are the author’s own.

7 Related Work

AI control evaluation. Our work extends the ControlArena ecosystem [3, 8, 13, 27] with a setting that targets identity-layer operations. Unlike BashArena’s infrastructure sabotage, our setting evaluates attacks that operate entirely within the agent’s authorized capabilities.

Monitor information access. Guan et al. [10] established that monitor information access affects detection performance by studying chain-of-thought access. We study a complementary axis: structured domain context. Arike et al. [2] found that additional context can degrade detection. Our Act 2

results support this finding (0.011 vs. 0.386); our Act 3 results show the effect is mitigable through prompt engineering.

Agent identity and delegation. Rodriguez Garzon et al. [20], South et al. [21], and Zhou [28] proposed cryptographic identity frameworks for AI agents. Our work is the first to evaluate whether the structured context these systems produce can improve monitoring within the AI control paradigm. Our results are nuanced: with uncalibrated prompts the context hurts; with calibrated prompts it helps. The identity and control communities need to collaborate on *how* context is presented, not just *what* context is provided.

Adaptive attacks. Terekhov et al. [24] showed that adaptive attacks can degrade trusted monitors. Our attacks are non-adaptive. Testing against adaptive attackers who know about the context-enrichment pipeline remains future work.

8 Conclusion

We set out to test whether structured identity context improves LLM-based monitoring of within-scope attacks. The answer is yes, but only with calibrated prompts, and the path to that answer is where the real contribution lies.

The four-act progression exposed a stack of compounding failures. A silent XML parsing bug nullified the behavioral monitor entirely (Act 1). Uncalibrated prompts turned structured context into a liability, producing near-zero discrimination (Act 2). Only after calibrating the prompt with anchored rubrics and step-by-step reasoning did context enrichment pull ahead, with the $n=36$ results in Act 3 confirming the largest discrimination gap (0.436) and lowest false positive rate of any protocol tested. Act 4 provided pilot evidence ($n=4$ per config) suggesting this generalizes across five model pairings, including a cross-provider test with Mistral Large 3 ($p=0.031$, sign test).

Two unexpected findings deserve emphasis. First, Sonnet 4.6 categorically refused to execute attacks, suggesting that safety training at the frontier may outrun standard evaluation methodologies. Second, the behavioral monitor in Config 3 produced an inverted discrimination gap, flagging honest trajectories more than attacks when monitoring its own model family. Context enrichment rescued both cases.

For practitioners: validate your monitor’s output format before trusting any evaluation. Calibrate prompts against both honest and attack baselines. Start with behavioral monitoring as a robust default, then layer in context with calibrated prompts for better discrimination. The design of *how* information is presented to a monitor deserves as much attention as *what* information is presented. Every integration boundary needs validation. Every prompt needs calibration.

Acknowledgments

This work was conducted as part of the AI Control Hackathon 2026, organized by Apart Research and Redwood Research. The evaluation infrastructure was built on ControlArena [26]. Identity primitives were provided by Attestix (pip install attestix). Model inference was conducted via AWS Bedrock (us-east-1) using Claude Sonnet 4.5, Claude 3.5 Haiku, Claude 3 Haiku, Claude Sonnet 4.6, Claude Haiku 4.5, and Mistral Large 3 (675B MoE). The evaluation pipeline was executed on Lambda Cloud (NVIDIA A10, 24GB). Compute credits (\$400) were provided by Lambda Cloud through the hackathon. Total API costs were approximately \$5-8 in Bedrock inference charges. Claude Code (Anthropic) was used as a development assistant for code implementation, experiment orchestration, and analysis. All scientific decisions, experimental design, and interpretation were made by the author.

References

- [1] Nadya Abaev, Denis Klimov, Gerard Levinov, David Mimran, Yuval Elovici, and Asaf Shabtai. AgentGuardian: Learning access control policies to govern AI agent behavior. *arXiv preprint arXiv:2601.10440*, 2026.
- [2] Markus Arike et al. How does information access affect LLM monitors’ ability to detect sabotage? *arXiv preprint arXiv:2601.21112*, 2026.
- [3] Arjun Bhatt, Connor Rushing, Andrei Kaufman, Thomas Tracy, Vasil Georgiev, Daniel Matolcsi, Aaqib Khan, and Buck Shlegeris. Ctrl-z: Controlling AI agents via resampling. *arXiv preprint arXiv:2504.10374*, 2025.
- [4] Norman Cliff. Dominance statistics: Ordinal analyses to answer ordinal questions. *Psychological Bulletin*, 114(3):494–509, 1993.
- [5] Jacob Cohen. *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates, 2nd edition, 1988.
- [6] Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall/CRC, 1993.
- [7] Ronald A. Fisher. On the interpretation of χ^2 from contingency tables, and the calculation of P. *Journal of the Royal Statistical Society*, 85(1):87–94, 1922.
- [8] Ryan Greenblatt, Buck Shlegeris, Kshitij Sachan, and Fabien Roger. Ai control: Improving safety despite intentional subversion. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024. *arXiv:2312.06942*.
- [9] Charlie Griffin, Louis Thomson, Buck Shlegeris, and Alessandro Abate. Games for AI control: Models of safety evaluations of AI deployment protocols. *arXiv preprint arXiv:2409.07985*, 2024.
- [10] Michael Guan et al. Monitoring monitorability. *arXiv preprint arXiv:2512.18311*, 2025.
- [11] Richards J. Heuer. *Psychology of Intelligence Analysis*. Center for the Study of Intelligence, CIA, 1999.
- [12] Ken Huang, Vineeth Sai Narajala, John Yeoh, Jason Ross, Ramesh Raskar, Youssef Harkati, Jerry Huang, Idan Habler, and Chris Hughes. A novel zero-trust identity framework for agentic AI: Decentralized authentication and fine-grained access control. *arXiv preprint arXiv:2505.19301*, 2025.
- [13] Andrei Kaufman, Jasper Lucassen, Thomas Tracy, Connor Rushing, and Arjun Bhatt. BashArena: A control setting for highly privileged AI agents. *arXiv preprint arXiv:2512.15688*, 2025.
- [14] Tomasz Korbak, John Clymer, Ben Hilton, Buck Shlegeris, and Geoffrey Irving. A sketch of an AI control safety case. *arXiv preprint arXiv:2501.17315*, 2025.
- [15] Zibin Lin, Shengli Zhang, Guofu Liao, Dacheng Tao, and Taotao Wang. Binding agent ID: Unleashing the power of AI agents with accountability and credibility. *arXiv preprint arXiv:2512.17538*, 2025.
- [16] Alex Mallen, Charlie Griffin, Max Wagner, Alessandro Abate, and Buck Shlegeris. Subversion strategy eval: Can language models statelessly strategize to subvert control protocols? *arXiv preprint arXiv:2412.12480*, 2024.
- [17] Henry B. Mann and Donald R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, 18(1):50–60, 1947.

- [18] Remi Marchand, Aodhan O Cathain, James Wynne, Panagiotis M. Giavridis, Sam Deverett, Jack Wilkinson, Josh Gwartz, and Harry Coppock. Quantifying frontier LLM capabilities for container sandbox escape. *arXiv preprint arXiv:2603.02277*, 2026.
- [19] Donald A. Redelmeier and Eldar Shafir. Medical decision making in situations that offer multiple alternatives. *JAMA*, 273(4):302–305, 1995.
- [20] Santiago Rodriguez Garzon et al. Decentralized identifiers and verifiable credentials for AI agents. *arXiv preprint arXiv:2511.02841*, 2025.
- [21] Tobin South et al. Authenticated delegation for AI agents. *arXiv preprint arXiv:2501.09674*, 2025.
- [22] Asa Cooper Stickland, Jan Michelfeit, Aarav Mani, Charlie Griffin, Oliver Matthews, Tomasz Korbak, Rowan Inglis, Oliver Makins, and Aidan Cooney. Async control: Stress-testing asynchronous control measures for LLM agents. *arXiv preprint arXiv:2512.13526*, 2025.
- [23] Georgios Syros, Anshuman Suri, Jacob Ginesin, Cristina Nita-Rotaru, and Alina Oprea. SAGA: A security architecture for governing AI agentic systems. *arXiv preprint arXiv:2504.21034*, 2025.
- [24] Maxim Terekhov, Artem Panfilov, Dzmitry Dzenhaliou, Caglar Gulcehre, Maksym Andriushchenko, Ameya Prabhu, and Jonas Geiping. Adaptive attacks on trusted monitors subvert AI control protocols. *arXiv preprint arXiv:2510.09462*, 2025.
- [25] Chih-Fong Tsai, Yu-Feng Hsu, Chia-Ying Lin, and Wei-Yang Lin. Intrusion detection by machine learning: A review. *Expert Systems with Applications*, 36(10):11994–12000, 2009.
- [26] UK AI Security Institute and Redwood Research. ControlArena: A library for AI control experiments. <https://control-arena.aisi.org.uk>, 2026.
- [27] Francis Rhys Ward, Teun van der Weij, Hannah Gabor, Simon Martin, Raul Montoya Moreno, Haim Lidor, Lucas Makower, Thomas Jodrell, and Lewis Robson. CTRL-ALT-DECEIT: Sabotage evaluations for automated AI R&D. *NeurIPS 2025 Spotlight*, 2025. arXiv:2511.09904.
- [28] Xin Zhou. Cryptographic binding for AI agent tool use. *arXiv preprint arXiv:2603.14332*, 2026.