ODA3 INSTITUTE

Practitioner Cheat Sheet

AI Monitoring Architecture

DocID: ODA3-2026-07-CHT-SEC-002
Classification: Public

1. Executive Overview

Monitoring AI systems is not an extension of traditional application performance monitoring or SIEM practice — it is a different problem. Conventional telemetry was architected around deterministic requests: latency, error codes, resource utilization. LLM-based systems introduce failure modes that produce no error code at all: a model that confidently returns wrong content, an agent that takes an unintended but syntactically valid action, retrieved context that silently steers behavior, or quality that degrades gradually as inputs drift [T2]. None of these surface in infrastructure metrics; all of them matter operationally, contractually, and increasingly, regulatorily.

Effective AI monitoring therefore spans four signal classes simultaneously — security (injection attempts, tool abuse, probing), quality (drift, hallucination, evaluation scores), operational (latency, token consumption, cost), and compliance (policy decisions, human approvals, audit evidence) — collected across every layer of the inference pipeline, not just the model call [T2]. This cheat sheet provides the reference architecture, telemetry requirements, detection patterns, and assurance criteria for building that capability. Technical sections carry inline evidence tags; leadership-relevant framing and action items are called out separately throughout.

> **Key Takeaway**
>
> You cannot secure, assure, or improve what you cannot see. Most AI incidents are reconstructable only if the fully assembled context, tool decisions, and policy outcomes were logged at the time — monitoring architecture is a deployment-day decision, not a post-incident retrofit.

Methodology Note

This document synthesizes publicly available evidence from standards organizations, open-source specification projects, regulatory publications, academic research, and vendor advisories. No ODA3 proprietary datasets, client telemetry, or unpublished operational intelligence are referenced, consistent with ODA3 Institute's founding in March 2026. Evidence is tiered as follows:

Tier	Description
T1	Primary Verified — standards, specifications, regulatory documents, vendor advisories, documented incidents
T2	Secondary Verified — credible reporting corroborated by ≥ 1 independent source
T3	Controlled Simulation / Academic Proxy — lab-replicated or peer-reviewed, not field-observed
T4	Anecdotal / Unverified — practitioner reports, unconfirmed claims

Where sources disagree on prevalence, exploitability, or defensive effectiveness, differing viewpoints are identified rather than resolved into an unsupported conclusion.

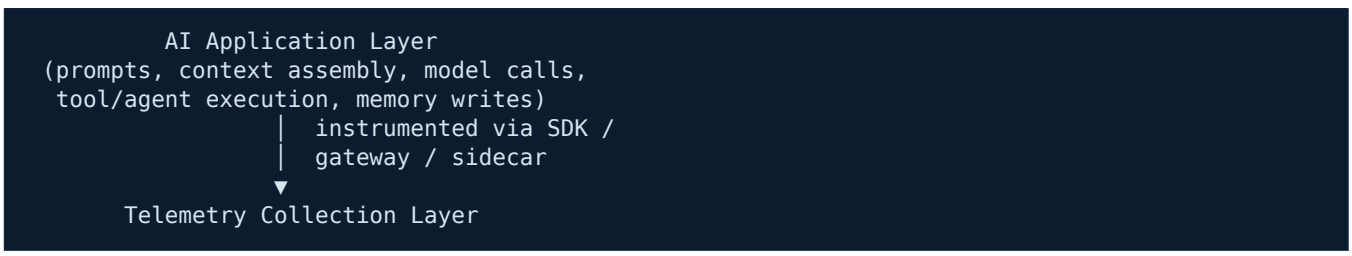
2. Key Concepts & Glossary

Term	Definition
AI Observability	The ability to reconstruct and explain what an AI system did and why, from collected telemetry — distinct from availability monitoring
Trace / Span	A trace is the end-to-end record of one request through the system; spans are its constituent steps (model call, retrieval, tool execution), linked by a shared trace ID
OpenTelemetry (OTel) GenAI Semantic Conventions	An open specification standardizing telemetry attribute names, spans, metrics, and events for GenAI clients, agents, and MCP tool calls; in Development (experimental) status as of this writing [T1]
Assembled-Context Logging	Recording the complete prompt as actually sent to the model — system instructions, retrieved chunks, history, tool outputs — not just the user's message
Context Hash / Prompt Hash	A cryptographic digest of assembled context enabling integrity verification and deduplication

Term	Definition
	without storing full content everywhere
Semantic Monitoring	Evaluating the meaning and intent of inputs/outputs (via classifiers or judge models) rather than pattern-matching on keywords
Behavioral Baseline	A statistical profile of normal system behavior (tool-call mix, output length distribution, refusal rate) against which anomalies are scored
Data Drift / Concept Drift	Change in input distribution over time / change in the relationship between inputs and correct outputs — both degrade quality without any error being thrown
Hallucination Rate	The measured frequency of factually unsupported output on an evaluation set; meaningful only relative to a defined golden dataset and scoring method
Golden Dataset	A curated, versioned set of inputs with known-correct outputs used for regression evaluation across model or prompt changes
Canary Prompt	A scheduled, known-answer probe run against production to detect silent degradation, drift, or guardrail failure between releases
LLM-as-a-Judge Evaluation	Using a secondary model to score primary outputs for quality, safety, or intent alignment; judge scores are themselves telemetry
Guardrail Telemetry	Logs of every guardrail decision — what was evaluated, the score, and the allow/block outcome — including low-confidence allows
Policy Decision Log	The record of each authorization decision made by an independent policy engine for tool/action requests, with the rule and identity involved
HITL Audit Trail	The record of every human-in-the-loop approval or rejection: action requested, decision, decider, and rationale
Session-Level Intent Tracking	Correlating actions across a session/identity to detect distributed patterns (e.g., repeated probing) invisible in any single event
Detection Latency	Time from a monitored event occurring to an actionable alert; a first-class metric for monitoring architecture itself
Evidence-Grade Retention	Storage with integrity guarantees (immutability/WORM, access logging) sufficient for forensic or regulatory use
Model Version Pinning	Recording the exact model, embedding, prompt, and policy versions in effect for every decision, enabling post-incident attribution
Kill Switch	A deterministic, externally triggered mechanism to halt an AI system or agent and revoke its tool access, independent of the model itself
Token / Cost Telemetry	Per-request input/output token counts and derived cost, used for both budget control and unbounded-consumption abuse detection
Content Capture Modes	Configurable levels of prompt/response recording; the OTel GenAI conventions default to NOT capturing message content, requiring explicit opt-in, as a PII-protection design decision [T1]

3. Technical Deep Dive

3.1 Reference Architecture



Three properties distinguish this from conventional observability stacks. First, the pipeline must treat **security-relevant events as never-sampled**: cost pressure legitimately drives sampling of high-volume operational telemetry, but a sampled-away injection attempt is an undetectable one. Second, the architecture requires a **dual storage path**: hot storage optimized for real-time detection, and a separate evidence-grade store with integrity guarantees, because post-incident forensics and regulatory response impose requirements (immutability, chain of custody, retention) that detection storage does not [T2]. Third, monitoring must remain **logically independent of the system it observes**: a compromised or misbehaving inference pipeline must not be able to compromise, suppress, or rewrite its own monitoring evidence — and the monitoring stack itself requires monitoring, since a silently failed telemetry pipeline is indistinguishable from a perfectly healthy system [T2].

3.2 Telemetry Layers — What to Instrument

Layer	What to Capture	Why It Matters
Input	User message, session ID, authenticated identity, channel	Attribution and session correlation
Context Assembly	Retrieved chunk IDs, source/trust tags, context hash, embedding model version, vector collection ID	The layer where indirect injection enters; unlogged assembly makes forensics impossible
Model Inference	Model + version, parameters, token counts, latency, finish reason	Attribution across model updates; cost/abuse signals
Guardrails	Every evaluation: score, threshold, allow/block outcome — including low-confidence allows	Guardrail decisions are evidence; silent allows are the blind spot
Tool / Action	Tool name, parameters, authorization decision, policy version, result status	The highest-consequence layer; every action needs a decision record
Output	Validation results, judge scores, redaction events, delivery status	Quality and leakage detection
Memory	Memory writes: type, content hash, writing session, memory version	Poisoned memory persists; writes must be auditable as privileged actions
Infrastructure	Standard APM metrics, egress connections from agent sandboxes	Egress anomalies are high-confidence exfiltration indicators

3.3 The Four Signal Classes

Class	Representative Signals	Primary Consumer
Security	Injection-pattern hits, repeated probing, unexpected tool calls, egress anomalies, guardrail bypass indicators	SOC / AI Security
Quality	Drift scores, hallucination rate on golden datasets, judge evaluation trends, canary prompt regressions	ML/AI Engineering
Operational	Latency, token consumption, cost per request, error and timeout rates, unbounded-consumption spikes	Platform / FinOps
Compliance	Policy decisions, HITL approvals/overrides, retention adherence, model/prompt version records	Governance / Audit

These classes share a pipeline but diverge in retention, access control, and alerting. A common architecture mistake is building for one class (usually operational) and assuming the others come for free — they do not: security requires never-sampled capture, quality requires versioned evaluation datasets, and compliance requires evidence-grade retention that operational tooling rarely provides [T2].

3.4 Standardization: OTel GenAI Semantic Conventions

The OpenTelemetry GenAI semantic conventions define standard attribute names, span structures, metrics, and events for GenAI client calls, agent operations, and MCP tool executions, and have become the de facto direction for vendor-neutral AI telemetry — with commercial observability platforms adding native support [T1/T2]. Two adoption caveats matter for architecture decisions:

- **The conventions are in Development (experimental) status** — attribute names and structures have changed across recent releases, and the specification has moved to a dedicated repository. Adopters should pin the convention version they emit and plan for migration, using the documented stability opt-in mechanism during transitions [T1]
- **Prompt/response content is not captured by default** — the conventions deliberately omit message content unless explicitly enabled, as a PII-protection measure. Security forensics, however, frequently requires content; the resolution is explicit content capture routed through the redaction pipeline into the evidence store, rather than either extreme of "log nothing" or "log everything in plaintext" [T1/T2]

3.5 Detection Patterns

Baseline + anomaly: establish a 14-day behavioral baseline (tool-call mix, output length distribution, refusal rate, per-identity request patterns) before enabling blocking-mode alerts; alert on statistically significant deviation rather than fixed thresholds alone [T2].

Session-level correlation: a single denied action is unremarkable; repeated variations of the same denied action class within a session or identity window — repeated tool denials, prompt rewriting after blocks, extraction attempts — is a materially stronger signal than any individual event.

Canary prompts: scheduled known-answer probes against production detect silent regression from model updates, prompt changes, or guardrail drift between releases — the AI equivalent of synthetic monitoring.

Judge-based evaluation: a secondary model scores sampled production outputs for quality/safety/intent alignment; score trends are a leading indicator of degradation. The judge must differ architecturally from the primary model — a shared architecture shares blind spots [T2/T3].

Drift detection: monitor input distribution shift (data drift) and evaluation-score decay on golden datasets (concept/model drift); both degrade systems that report healthy on every infrastructure metric.

3.6 Monitoring Agentic Systems

Agents require monitoring across the full control loop — planner → tool selector → tool executor → memory — because injection or drift can enter at any stage, and single-point monitoring of the model call alone misses it [T2]. Practically this means: per-step spans linked under one trace ID so a full agent run is reconstructable; cross-tool chain analysis to catch sequences that are individually benign but collectively exfiltrating (e.g., read → encode → external send); memory-write auditing as a privileged-action log; and hard limits with alerts on loop depth and

sequential tool-call counts. The OTel GenAI work-in-progress explicitly includes agent and MCP tool-execution span conventions for this purpose [T1].

The monitoring gap for tool-connected agents is now formally recognized: the OWASP MCP Top 10 (2025) designates **MCP08:2025 — Lack of Audit and Telemetry** as a distinct risk category, noting that an unmonitored agent can silently perform sensitive operations or exfiltrate data for extended periods without detection, and recommending immutable audit trails of tool invocations, context changes, and user-agent interactions, OpenTelemetry-based tracing across the MCP pipeline, PII-safe logging, and behavioral baselining with anomaly detection [T1].

> **Documented Incident Reference — What Missing Visibility Looks Like**
>
> EchoLeak (CVE-2025-32711, CVSS 9.3), disclosed by Aim Security in June 2025, was a zero-click indirect prompt injection in Microsoft 365 Copilot: a single crafted email, requiring no user interaction, could cause the assistant to retrieve internal content and exfiltrate it externally, chaining bypasses of the injection classifier, link redaction, and content security policy [T1, vendor advisory and CVE record; T2, independent corroboration]. Microsoft patched it server-side, and no in-the-wild exploitation has been confirmed.
>
> Its monitoring lesson is direct: detecting this attack class requires exactly the telemetry this document specifies — assembled-context capture (the hidden instructions entered via retrieved email content), egress monitoring (the exfiltration channel), and session-level correlation — none of which exists in infrastructure-level logging.

3.7 Monitoring Domains View

The layer model (Section 3.2) describes *where* to instrument; the domain view below describes *what operational question* each area answers. Both views should reconcile — a domain with no corresponding instrumented layer is a blind spot.

Domain	Operational Question	Representative Signals
Infrastructure	Is the platform available and performant?	Compute/GPU saturation, latency, availability
Model Behavior	Is output quality stable?	Response consistency, refusal rate, drift indicators, evaluation scores
Prompt Pipeline	What is entering the system?	Volume, classification, injection/probing attempts, sensitive-data submission
Retrieval	Is retrieved knowledge trustworthy?	Trust score distribution, source diversity, malicious-content hits, citation coverage
Guardrails	Are policy controls functioning?	Block/allow decisions, false positive/negative trends, override events, guardrail availability
Tool Execution	Are autonomous actions authorized?	Invocation rate, authorization failures, unexpected tool selection, human overrides
Governance	Can compliance be evidenced?	Policy exceptions, approvals, risk-acceptance decisions, audit findings, retention adherence

3.8 Architecture Anti-Patterns

- **Logging only the user message** — without assembled context, indirect injection forensics is impossible
- **Sampling security events** — cost-driven sampling policies applied uniformly silently discard the events that matter most
- **Judge and primary sharing one architecture** — a bypass that fools the primary likely fools the judge
- **Plaintext PII in telemetry** — monitoring must not become the leak; redaction belongs in the pipeline, before storage
- **Alert-only design** — detection without a wired response path (kill switch, HITL escalation, SOAR) converts incidents into post-mortems

- **No version pinning in events** — without model/prompt/policy versions per decision, you cannot distinguish a control failure from a control change
- **Single retention policy for all signals** — operational telemetry and audit evidence have incompatible retention and integrity requirements
- **Unmonitored monitoring** — no health checks, coverage reports, or alerting on the telemetry pipeline itself; a dead pipeline looks identical to a quiet system

4. Practitioner Decision Framework

4a. Monitoring Depth by System Risk

Risk Factor	Low	Medium	High	Critical
System capability	Read-only Q&A	RAG over internal docs	Tool-using assistant	Autonomous agent with external actions
Data sensitivity	Public	Internal	Confidential	Regulated / Critical
Monitoring depth required	Operational + basic guardrail logs	+ assembled-context logging, retrieval telemetry	+ tool/policy decision logs, session correlation, judge evaluation	+ full trace-per-run, memory auditing, evidence-grade retention, kill-switch integration

4b. Decision Tree

```
Does the system use tools or take actions?
  NO  → Does it retrieve external/untrusted content?
        NO  → Operational telemetry + guardrail decision logs.
        YES → Add assembled-context logging + retrieval
              telemetry (chunk IDs, trust tags, context hash).
  YES → Are any actions irreversible or externally visible?
        NO  → Add tool invocation + policy decision logging,
              session-level correlation.
        YES → Full stack: per-step traces, memory-write audit,
              HITL audit trail, evidence-grade retention,
              wired kill switch. Deploy monitoring BEFORE the
              capability, not after.
```

4c. Prioritization

Scenario	Risk	Priority	Timeframe
Agent in production with no tool-decision logging	Critical	Highest	Immediate
RAG system logging user messages only	High	Immediate	30 days
No behavioral baseline before alerting	Medium	High	30 days (14-day baseline + tuning)
Detection with no wired response path	High	Immediate	30 days
No evaluation/canary regression suite	Medium	Medium	60 days

5. Monitoring Controls Matrix

ID	Control	Effect.	Complexity	NIST AI RMF	ISO 42001	GAISSF™ Domain	UAIF™ Category
M-01	Assembled-context logging (full prompt as sent, with hashes)	High	Low	Measure	Cl. 9.1	Evidence Collection	Event Logging
M-02	Tool invocation + policy decision logging	High	Low	Govern	Cl. 9.1	Evidence Collection	Event Logging
M-03	Memory-write auditing (privileged-action log)	High	Med	Govern	Cl. 8	Evidence Collection	Knowledge Poisoning
M-04	Guardrail decision telemetry (incl. low-confidence allows)	High	Low	Measure	Cl. 9.1	Runtime Controls	Incident Detection
M-05	Behavioral baselining + anomaly detection	High	Med	Measure	Cl. 9.1	Continuous Monitoring	Incident Detection
M-06	Data/concept drift detection on golden datasets	Med–High	Med	Measure	Cl. 9.1	Continuous Monitoring	Output Manipulation
M-07	Canary prompt monitoring	Med–High	Low	Measure	Cl. 9.1	Continuous Monitoring	Incident Detection
M-08	Judge-based output evaluation (architecturally distinct judge)	Med–High	High	Measure	Cl. 9.1	Response Validation	Output Manipulation
M-09	Session-level intent correlation	High	Med	Measure	Cl. 9.1	Continuous Monitoring	Incident Detection
M-10	Evidence-grade retention (immutable store, dual path)	High	Med	Govern	Cl. 7.5/9.1	Evidence Collection	Event Logging
M-11	Kill switch + response-path integration (SOAR/HITL)	High	Med	Manage	Cl. 8	Human Oversight	High Severity
M-12	Model/prompt/policy version pinning in every event	High	Low	Govern	Cl. 7.5	Evidence Collection	Event Logging
M-13	OTel GenAI-aligned instrumentation (version-pinned)	Med	Med	Measure	Cl. 9.1	Runtime Controls	Event Logging
M-14	Egress monitoring on agent execution environments	High	Med	Measure	Cl. 8	Continuous Monitoring	Privilege Abuse

ISO/IEC 42001:2023 references indicate the closest management-system clause (Cl. 8 Operation, Cl. 9.1 Monitoring/measurement/analysis/evaluation, Cl. 7.5 Documented information); organizations should map to their own Statement of Applicability. NIST AI RMF references indicate the primary function (Govern/Map/Measure/Manage). GAISSF™ and UAIF™ mappings reference GAISSF™ v1.0 and UAIF™ v1.0 respectively, as factual conformance-alignment statements under GEL v1.0 Section 10.3. OWASP LLM Top 10 and MITRE ATLAS technique-level mappings are addressed in the companion prompt-injection cheat sheet (ODA3-2026-07-CHT-SEC-001) and are not repeated per-control here.

6. Detection Rules — Reference Set

```
Rule: Low-Confidence Guardrail Allow
Logic: guardrail decision == ALLOW AND confidence < tuned threshold
→ alert (Medium). A spike in low-confidence allows indicates active
boundary-testing of the classifier.
```

```
False Positive Risk: Medium — baseline before enforcing.

Rule: Tool Call Outside Workflow Profile
Logic: invoked tool ∉ baseline tool-set for this workflow/identity
→ alert (High) → require policy review before re-enable.
False Positive Risk: Medium — legitimate new workflows exist;
route through change management, not silence.

Rule: Read→Transform→Egress Chain
Logic: within one session: sensitive-read tool call, followed by
encode/transform, followed by external-send or non-allowlisted
egress → alert (Critical), consider automatic session halt.
False Positive Risk: Low in most enterprise workflows.

Rule: Canary Regression
Logic: scheduled known-answer probe deviates from expected output
class → alert (High) → block pending release review; correlate
with model/prompt version change via pinned versions (M-12).
False Positive Risk: Low with well-designed canaries.

Rule: Evaluation Score Decay
Logic: rolling judge-score mean on sampled production output
declines beyond control band over N days → quality incident
(not security) → route to AI Engineering with drift report.
False Positive Risk: Medium — seasonality in inputs is real.
```

Telemetry Retention Schedule (Reference Baseline)

Category	Recommended Minimum Retention	Storage Path
Security events & alerts	12 months	Hot + evidence store
Audit evidence (policy decisions, HITL records, admin changes)	24 months	Evidence store (immutable)
Model performance / evaluation records	12 months	Hot + evidence store
Operational metrics	180 days	Hot storage
Trace data (full spans)	30–90 days	Hot storage (hashes retained longer in evidence store)

Retention baselines above are a starting point, not a compliance determination — align final periods with applicable regulatory, contractual, and sector obligations, which frequently exceed these minimums.

7. AI Lifecycle Mapping

Phase	Monitoring Concern	Required Capability
Model Acquisition	Baseline behavior of a new/updated model unknown	Golden-dataset evaluation before production traffic
Evaluation	Regression across model/prompt versions	Versioned evaluation harness, canary suite
Deployment	Instrumentation gaps ship with the release	Telemetry coverage as a release gate
Inference	Live security/quality/operational signals	Full pipeline per Sections 3.1–3.5
Monitoring	Detection efficacy degrades silently	Periodic red-team validation of detection rules

Phase	Monitoring Concern	Required Capability
		themselves
Retirement	Evidence obligations outlive the system	Retention plan for evidence store independent of system decommission

8. Assessment & Assurance Lens

8a. Verification vs. Validation of Monitoring

Aspect	Verification	Validation
Question	Is telemetry actually being collected as designed?	Would the monitoring detect a real attack or failure?
Evidence	Pipeline configuration, coverage reports, schema conformance	Red-team exercises against detection rules; injected-fault drills; measured detection latency
Failure mode if skipped	Silent coverage gaps	Dashboards that look healthy during an incident

Monitoring is the control most prone to verification-without-validation: pipelines run, dashboards render, and nobody has ever tested whether a simulated exfiltration chain actually fires an alert. Detection rules require the same adversarial validation as any other control [T2].

8b. Assurance Questions by Role

Security Architect: Can a full agent run be reconstructed from telemetry alone — assembled context, every tool decision, every memory write? What is never sampled?

MLOps Engineer: Which convention version is instrumentation pinned to, and what is the migration plan given the spec's experimental status? How is the evaluation harness versioned?

AI Governance Lead: Can you produce, for a given decision, the model, prompt, and policy versions in effect at that moment? What is the evidence-store retention and integrity guarantee?

CISO: What is measured detection latency for the highest-consequence scenario? When was a detection rule last validated by adversarial exercise, and what failed?

8c. Residual Risk

Even a fully implemented monitoring stack does not observe everything. Residual risk includes: semantic failures that pass every configured evaluation; novel attack patterns with no matching rule or baseline deviation; provider-side behavior changes invisible below the API boundary; judge-model blind spots shared with the primary; and detection latency itself — harm that completes faster than the alert path. Monitoring reduces time-to-detection and enables response and forensics; it does not prevent the underlying events, and should be represented in risk registers accordingly.

8d. Assessment Traps

Three claims that assessors should treat as red flags when offered as evidence of AI monitoring maturity:

Claim	Why It Fails
"We have monitoring"	Infrastructure monitoring (CPU, GPU, latency) does not capture injection attempts, tool misuse, drift, or agent behavior. The evidence to request is AI-specific: assembled-context logs, tool decision records, evaluation telemetry.
"We have logs"	Tamper-evidence is a technical property, not a policy statement. A log in ordinarily writable storage does not meet evidence-grade requirements regardless of what the retention policy document says.

Claim	Why It Fails
"We have a SIEM"	Ingestion is not detection. Generic correlation rules do not fire on prompt injection, out-of-profile tool calls, or agent drift; the evidence to request is AI-specific detection rules and their last adversarial validation result.

9. Maturity Model

Level	Characteristics
1 — Ad Hoc	Provider dashboard + application logs; user messages only; no baselines
2 — Repeatable	Structured logging of model calls; basic guardrail logs; manual review
3 — Managed	Assembled-context + tool decision logging; baselines established; alerting wired
4 — Measured	Detection latency and rule efficacy measured; drift + canary + judge evaluation operating; version pinning universal
5 — Optimized	Adversarial validation of detection in CI/CD; automated response paths; OTel-aligned, vendor-portable telemetry
6 — Assured	Independent assurance demonstrates detection efficacy with objective evidence; evidence store meets forensic/regulatory standards

10. Research Insight ★

Finding: The OpenTelemetry GenAI semantic conventions — the emerging vendor-neutral standard for AI telemetry — remain in Development (experimental) status as of this writing, have relocated to a dedicated specification repository, and have changed attribute structures across successive recent releases, while simultaneously expanding to cover agent operations and MCP tool-execution spans [T1, project specification and release records; T2, independent industry corroboration].

Technical Significance: The industry is standardizing AI observability in real time. Organizations face a genuine tension: adopting now yields vendor-portable telemetry and agent/MCP coverage; the experimental status means schema migration is a certainty, not a risk.

Validation of Guidance: This directly supports Control M-13 (OTel-aligned instrumentation, version-pinned) and M-12 (version pinning in every event) — pinning is what makes an inevitable schema migration a managed change rather than a forensic discontinuity.

Operational Lesson: Align with the conventions now, pin the emitted version explicitly, and use the documented stability opt-in mechanism during transitions — and do not assume default instrumentation captures message content, because by design it does not.

11. Common Mistakes

Mistake	Correct Approach
Logging only user messages	Log the fully assembled context with hashes; indirect injection enters at assembly
Treating monitoring as an ops-only concern	Build for all four signal classes: security, quality, operational, compliance
Uniform sampling across all telemetry	Security-relevant events are never sampled; sample operational volume instead

Mistake	Correct Approach
Alerting without a baseline	Establish a 14-day behavioral baseline before enabling blocking-mode alerts
Judge model sharing the primary's architecture	Use an architecturally distinct judge; shared architecture shares blind spots
No response path wired to detection	Integrate kill switch, HITL escalation, and SOAR before go-live
PII stored in plaintext telemetry	Redaction in the pipeline, before storage; content capture is explicit and routed
One retention policy for everything	Dual path: hot detection storage + immutable evidence store with distinct retention
No version pinning in events	Record model, prompt, embedding, and policy versions on every decision
Never testing detection itself	Red-team the monitoring: injected-fault drills, measured detection latency, rule efficacy review
Assuming instrumentation captures content by default	OTel GenAI conventions omit message content unless explicitly enabled — verify what you are actually collecting

12. Notably Absent

> Current publicly available evidence does not include several capabilities that monitoring programs are often assumed to have [T1/T2/T4]:

- **No stable, ratified industry-standard telemetry schema for GenAI** — the OTel GenAI semantic conventions, the leading candidate, remain in Development/experimental status with breaking changes across recent releases [T1]
- **No vendor-neutral benchmark for detection efficacy** — published false-negative rates for semantic injection detection, drift detection, or judge-based evaluation across heterogeneous production systems do not exist; vendor efficacy claims are self-reported [T2/T4]
- **No standardized evidentiary requirements** for AI incident forensics accepted across regulators — what constitutes sufficient, admissible AI telemetry evidence remains undefined [T2]
- **No public field data on real-world detection latency** for AI-specific attacks in fully instrumented production environments — published incidents overwhelmingly involve partially instrumented systems [T2/T4]

Consequently, monitoring-architecture decisions in this domain necessarily rest on engineering judgment and controlled evaluation rather than field-validated benchmarks — a limitation this document shares, and states, rather than obscures.

13. Quick Reference

Signal-to-Layer Coverage Check

If you cannot answer this…	You are missing…
What exact context did the model see?	Assembled-context logging (M-01)
Who approved this action, under which policy version?	Policy decision + HITL audit trail (M-02)
What was written to memory, and by which session?	Memory-write auditing (M-03)
Is this identity probing the guardrails?	Session-level correlation (M-09)

If you cannot answer this…	You are missing…
Did the last model update change behavior?	Canary + golden-dataset evaluation (M-06/M-07)
Which model/prompt version made this decision?	Version pinning (M-12)
Could we halt this agent right now?	Kill switch + response path (M-11)
Would this evidence survive an audit?	Evidence-grade retention (M-10)

Implementation Checklist

Item	Yes	No	Partial
Assembled context logged with hashes	☐	☐	☐
Every tool call carries an authorization record	☐	☐	☐
Memory writes audited as privileged actions	☐	☐	☐
Security events exempt from sampling	☐	☐	☐
14-day behavioral baseline before alert enforcement	☐	☐	☐
Canary prompts scheduled against production	☐	☐	☐
PII redaction in pipeline before storage	☐	☐	☐
Dual storage: hot detection + immutable evidence	☐	☐	☐
Model/prompt/policy versions pinned per event	☐	☐	☐
Kill switch tested and wired to detection	☐	☐	☐
Detection rules adversarially validated	☐	☐	☐
Telemetry aligned to OTel GenAI conventions, version-pinned	☐	☐	☐

14. If You Can Only Do Five Things This Week

#	Action	Impact	Effort
1	Enable assembled-context logging — record the full prompt as actually sent, with context hashes	Very High	Low
2	Log every tool invocation with its authorization decision and the policy version in effect	Very High	Low
3	Exempt security-relevant events from all sampling policies	High	Low
4	Start the 14-day behavioral baseline now so alerting can be tuned on real distributions	High	Low
5	Run one injected-fault drill: simulate a read→transform→egress chain and measure whether — and how fast — an alert fires	High	Medium

For Leadership: Questions to Ask Your Team

Can we reconstruct exactly what an AI system saw and did during a suspected incident — today?

Has our AI detection ever been tested against a simulated attack, and what was the measured detection time?

Do we know which model and policy versions were in effect for any given past decision?

If an agent misbehaves right now, who can stop it, and how fast?

15. References (Verified for This Edition)

Source	Relevance	Tier
OpenTelemetry GenAI Semantic Conventions (dedicated specification repository; Development status; spans/metrics/events for GenAI clients, agents, and MCP tool execution; content capture off by default)	Basis for Sections 3.4, 3.6, Research Insight, Notably Absent	T1
OpenTelemetry semantic-conventions release records (gen_ai.* migration to dedicated repository; stability opt-in transition mechanism)	Version-pinning and migration guidance	T1
Independent industry analyses of GenAI observability adoption and platform support for the conventions	Corroboration of adoption direction and experimental status	T2
NIST AI RMF 1.0 (Govern/Map/Measure/Manage functions)	Controls Matrix function mapping	T1
ISO/IEC 42001:2023 (Cl. 7.5, 8, 9.1)	Controls Matrix clause mapping	T1
OWASP MCP Top 10 (2025) — MCP08:2025, Lack of Audit and Telemetry	Formal risk-category basis for Section 3.6; corroborates telemetry, baselining, and immutable-audit guidance	T1
EchoLeak, CVE-2025-32711 (Aim Security disclosure, June 2025; Microsoft advisory; CVSS 9.3; no confirmed in-the-wild exploitation)	Documented incident reference in Section 3.6	T1/T2
ODA3-2026-07-CHT-SEC-001 (Prompt Injection Mitigation Controls)	Injection-specific technique/control detail referenced, not repeated	Internal

The GenAI conventions are under active development; verify the current version and stability status against the specification repository before citing in external certification materials. EU AI Act article-level mappings are intentionally omitted pending direct verification against the regulation's current consolidated text. Additional externally suggested references — specific MITRE ATLAS technique-ID mappings for monitoring, an "OWASP Agent Observability Standard," and fixed ATLAS tactic/technique counts — were reviewed for this edition and excluded: they either conflicted with independently verified sources or could not be verified, consistent with this document's evidence-fabrication safeguard.

Bottom Line

AI monitoring is the control that makes every other control auditable. Organizations that instrument fully at deployment can detect, respond, and prove; organizations that retrofit after an incident can only estimate. Treat telemetry coverage as a release gate, validate detection the way you validate any other control — adversarially — and keep evidence to a standard an assessor, not just an engineer, would accept.

Document Metadata

Field	Value
Classification	Public
Intended Audience	Primary: CISOs, Security Architects, AI Governance Leads, Compliance Officers. Secondary: CEOs, CFOs, Board Members, General Counsel, Risk Committees
Evidence Confidence	Mixed T1–T4; see inline tags
Review Cycle	Quarterly
ODA3 Framework Cross-References	GAISSF™ v1.0 (Continuous Monitoring, Evidence Collection, Runtime Controls, Human Oversight domains), UAIF™ v1.0 (Incident Detection, Event Logging, Knowledge Poisoning, Output Manipulation categories)
Related ODA3 Publications	ODA3-2026-07-CHT-SEC-001 — Prompt Injection Mitigation Controls

Field	Value
Framework Licensing	GAISSF™, UAIF™, and AI-IRF™ are frameworks of ODA3 Institute, referenced under the GAISSF Ecosystem License (GEL) v1.0. Mappings herein are factual conformance-alignment statements (GEL v1.0 §10.3) and do not constitute certification.