

Causal Telemetry and Guarded AI Agents for Self Healing Object Storage

Sneha Gullapalli

Principal Software Engineer | Distributed Storage AI and Observability

Abstract

Self-healing storage is often described as an anomaly detector connected to an automation tool. That design is incomplete. An anomaly establishes that behavior changed; it does not establish why the change occurred, whether intervention is safer than waiting, or whether the proposed action will preserve durability and availability. This paper presents a reference architecture for self-healing object storage built on causal telemetry and guarded AI agents. The system constructs a time-aligned evidence graph from metrics, logs, traces, topology, and change events; evaluates competing hypotheses; and emits a typed remediation contract rather than an administrative command. A deterministic authority layer rechecks preconditions, policy, blast radius, and rollback before an idempotent executor acts. Verification telemetry then determines whether the intended outcome occurred and whether secondary harm emerged. The approach supports progressive autonomy while retaining explicit human oversight for high-impact changes. It is an architectural proposal, not a report of production deployment or measured performance.

Keywords enterprise AI, object storage, causal telemetry, self-healing systems, observability, guarded agents

1 Why Detection Is Not Diagnosis

Distributed object storage produces symptoms far from their causes. A rise in tail latency may begin with a slow device, metadata contention, network loss, uneven request placement, replica repair, or a software rollout. These mechanisms can produce similar dashboards. A model trained to associate a pattern with a prior incident may select the familiar explanation even when the current topology or workload differs. Automated remediation amplifies that uncertainty because restarting a gateway, throttling repair, or moving traffic changes the system that is being diagnosed.

The design target is therefore not an agent that always acts. It is a closed loop that can collect sufficient evidence, distinguish correlation from a plausible causal mechanism, abstain when evidence conflicts, and execute only bounded operations. OpenTelemetry describes traces, metrics, logs, and contextual baggage as complementary signals [1]. Storage diagnosis also requires topology generation, software version, object-size cohort, placement state, capacity pressure, and change history. Without those dimensions, telemetry can show coincidence while hiding the affected population.

2 Closed Loop Architecture

Closed loop self healing with causal evidence and bounded authority

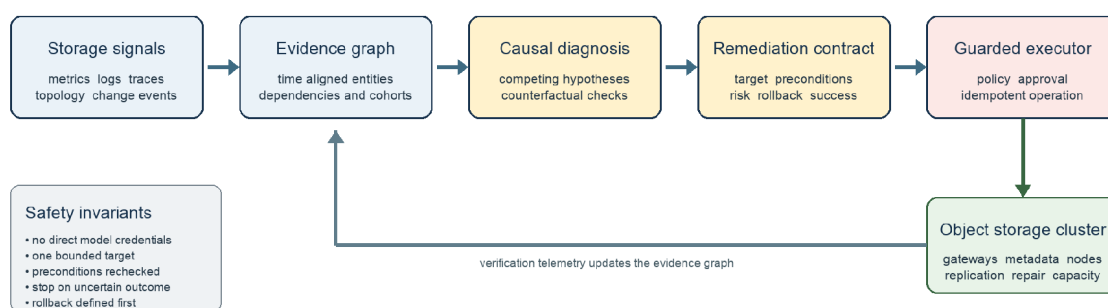


Figure 1 Causal diagnosis and guarded remediation form a verified control loop

Figure 1 separates reasoning from authority. AI agents may retrieve approved evidence, form hypotheses, request additional observations, and propose a remediation contract. They do not hold general storage credentials. The executor accepts a contract only after deterministic policy and state checks. This prevents model fluency from being mistaken for operational authority and limits the effect of prompt injection or a compromised peer agent. OWASP identifies excessive functionality, permissions, and autonomy as principal causes of excessive agency [2].

3 Constructing Causal Evidence

The evidence graph links observations to stable entities and time windows: client request, gateway process, metadata shard, storage node, device, replica set, background job, deployment, and operator change. Trace identifiers connect request paths; resource attributes connect telemetry to nodes and services; topology versions prevent evidence from different cluster states from being merged. Every observation records collection time and freshness. Missing or sampled data remains visible as uncertainty instead of being silently imputed as normal.

Diagnosis begins with at least two competing hypotheses. For example, if repair traffic causes write latency, elevated device queues should overlap repair activity on affected nodes, similar nodes without repair should provide a comparison cohort, and reducing repair concurrency should improve the target signal without increasing durability risk. If gateway saturation is causal, those predictions change. A verifier agent searches for observations that would contradict each hypothesis. The result is not a narrative alone but an evidence ledger containing supporting facts, counterevidence, unresolved gaps, and a confidence level tied to explicit tests.

4 Remediation as a Contract

A remediation proposal must be machine-checkable. Free-form explanation helps an operator, but only typed fields reach the authority layer.

Contract field	Control purpose	Storage example
Target and scope	Bounds the affected resources	one gateway instance in cluster A
Preconditions	Rejects stale assumptions	two healthy peers and stable quorum
Risk budget	Caps cumulative disruption	no second node action in the same zone
Idempotency key	Makes uncertain retries safe	incident target action and generation
Success signal	Defines measurable recovery	p99 latency and error rate return to baseline
Rollback and stop	Precommits recovery behavior	restore routing and stop on durability alarm

Table 1 Minimum controls in a storage remediation contract

5 Authority and Execution

Immediately before execution, the authority layer resolves the requester, target, cluster generation, maintenance policy, concurrent operations, and current health. Approval binds to the exact contract, not to an earlier conversational summary. Low-risk read operations can run automatically. A reversible single-target routing change may require one operator. Replica reduction, permission changes, destructive data operations, and multi-zone actions require stronger separation of duties or remain prohibited. NIST recommends defined human-AI roles, deployment-like evaluation, production monitoring, override, recovery, and change-management mechanisms [3].

The executor exposes narrow, allowlisted operations rather than a shell. It writes an intent record before the external call and a result record afterward. A timeout produces an unknown outcome, not an automatic failure; authoritative state must be inspected before retry. Generation checks and idempotency keys prevent a successful but unacknowledged action from being duplicated. Kubernetes health guidance illustrates the operational distinction between removing an unready endpoint and restarting a failed process, while warning that incorrect liveness behavior can cause cascading failures [4]. Storage automation needs the same discipline: isolate before restart, preserve quorum, and never confuse overload with irrecoverable failure.

6 Verification and Progressive Autonomy

Command success is not outcome success. Verification compares post-action telemetry with a pre-action baseline and a matched unaffected cohort. It checks the intended signal, such as tail latency, together with guard signals such as error rate, replica health, repair backlog, capacity headroom, and cross-zone traffic. If the intended metric improves while a guard signal degrades, the loop must not declare success. It rolls back when safe or escalates with the exact completed effects preserved in the audit trail.

Autonomy should advance through incident replay, read-only production diagnosis, operator-reviewed recommendations, approved reversible actions, and only then narrow automatic remediation for repeatedly validated cases. Release gates should measure localization accuracy, unsupported-claim rate, abstention quality, policy rejection, false remediation, rollback success, and recurrence. Tests must include stale signals, missing traces, contradictory evidence, topology changes during approval, injected instructions in logs, executor timeout, and failed rollback. NIST's generative AI profile emphasizes risk management across the lifecycle rather than relying on a one-time model assessment [5].

7 Telemetry Governance and Reproducibility

Enterprise storage telemetry can contain tenant identifiers, object names, access paths, and operational details that reveal customer behavior. Collection must therefore begin with purpose limitation: retain only the attributes needed for diagnosis, tokenize tenant identity where possible, and enforce authorization before retrieval rather than filtering after information reaches a model. Evidence indexes should preserve tenant and environment boundaries. Model prompts must not become an alternate observability archive, and secrets or raw credentials should never enter model context.

Every diagnosis should also be reproducible. The audit record binds the evidence snapshot, query definitions, topology generation, model version, retrieval configuration, policy version, proposed contract, approval, executor result, and verification window. When a model, prompt, embedding index, or policy changes, historical incidents can be replayed to detect regressions before release. This record separates an improved model from a changed dataset and allows reviewers to determine whether a recommendation was reasonable using the information available at that time. Reproducibility turns an apparently intelligent action into an inspectable engineering decision.

8 Conclusion

A credible self-healing storage platform is not an AI model with administrative tools. It is a constrained control system in which causal evidence supports a diagnosis, a typed contract bounds the proposed change, deterministic policy controls authority, and telemetry proves the result. This structure allows enterprise teams to gain speed from AI reasoning without transferring durability, availability, or security decisions to probabilistic output. The most valuable form of autonomy is not frequent action; it is reliable judgment about when evidence is sufficient, when a bounded intervention is safe, and when the system must stop and ask an operator.

Declaration of AI Assisted Writing

OpenAI Codex [7] assisted with preparation of the initial draft, organization of the technical argument, creation of the original diagram, language editing, reference discovery, and document formatting. The author supplied the professional context, selected the technical direction, reviewed and revised the manuscript, and accepts responsibility for its accuracy, originality, interpretation, and final content.

References

- [1] OpenTelemetry Authors, "Signals," OpenTelemetry Documentation. [Online]. Available: <https://opentelemetry.io/docs/concepts/signals/>.
- [2] OWASP Foundation, "LLM06:2025 Excessive Agency," OWASP GenAI Security Project, 2025. [Online]. Available: <https://genai.owasp.org/llmrisk/llm062025-excessive-agency/>.
- [3] National Institute of Standards and Technology, "AI Risk Management Framework Core," AI Resource Center. [Online]. Available: <https://airc.nist.gov/airmf-resources/airmf/5-sec-core/>.
- [4] Kubernetes Authors, "Liveness, Readiness, and Startup Probes," Kubernetes Documentation. [Online]. Available: <https://kubernetes.io/docs/concepts/workloads/pods/probes/>.
- [5] National Institute of Standards and Technology, "Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile," NIST AI 600-1, July 2024, doi: 10.6028/NIST.AI.600-1.
- [6] OpenTelemetry Authors, "Observability Primer," OpenTelemetry Documentation. [Online]. Available: <https://opentelemetry.io/docs/concepts/observability-primer/>.
- [7] OpenAI, "Codex," OpenAI Developers. [Online]. Available: <https://developers.openai.com/learn/codex>.