International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(17s) 2026 pp. 895-903
https://doi.org/10.70917/ijcisim-2026-4808
Article

Memory Governance For AI Agents: Defending Against Cognitive State Traps

Ayush Jain

Independent researcher, India, Email:ayush.jain7070@gmail.com

Abstract: Long-horizon AI agents increasingly depend on persistent memory for planning and decision-making, creating an attack surface that existing defenses leave largely unaddressed. Prompt filtering and output validation protect individual interactions but offer no protection once adversarial information enters long-term storage. This paper introduces Memory Governance, a security-oriented framework that treats agent memory as a governed asset subject to continuous evaluation rather than passive storage. The framework combines provenance tracking, a weighted trust score with explicitly constrained weights, exponential confidence decay, and three-state quarantine containment to reduce the long-term influence of adversarial information. A Trust-Decay Memory Evaluation algorithm classifies each memory object as Trusted, Review Required, or Quarantined based on source reliability, validation history, and time-elapsed confidence. A discrete-time contamination propagation model, adapted from epidemiological dynamics, derives the condition $\mu > \beta$ under which governance controls drive contamination density to zero at steady state. Together, these mechanisms establish memory governance as an architectural security control rather than an interaction-level filter.
Keywords: Agentic AI, Cognitive State Traps, Memory Governance, Memory Poisoning, Provenance Tracking, Trust Management, AI Safety, Long-Horizon Agents

1. INTRODUCTION

LLM-based agents are evolving from conversational assistants into autonomous systems capable of planning, tool execution, and task completion across extended time horizons. Unlike single-turn systems, these agents depend on persistent memory to maintain contextual awareness, accumulate experience, and adapt behavior across interactions [1][2].

Persistent memory creates a novel attack surface. Information stored in memory continues to influence future decisions long after its original acquisition, allowing adversarial inputs to persist and propagate throughout the agent lifecycle. The AI Agent Traps framework [1] identifies Cognitive State Traps as attacks that exploit exactly this persistence: by injecting misleading observations, fabricating task completion records, or poisoning retrieved documents, an adversary can bias the agent's planning process without triggering any interaction-level defense.

Existing defenses focus on the boundaries of individual interactions - prompt filtering [4], tool restriction [6], and output validation [7]. These mechanisms provide limited protection once information has entered long-term storage. A sanitized retrieval pipeline offers no defense against a memory object committed during a prior compromised session.

Memory Governance fills this gap with a dedicated protection layer whose design is described in three original contributions. First, we formalize memory governance as a continuous classification problem over memory objects, where each object is assigned a trust state based on source provenance, validation history, and time-elapsed confidence decay - a combination absent from existing agent memory architectures. Second, we introduce a Trust-Decay Memory Evaluation algorithm with three outcome states and weight constraints that make governance thresholds directly interpretable as trust levels in $[0, 1]$. Third, we adapt a discrete-time epidemiological propagation model to agent memory contamination and derive the steady-state condition $\mu > \beta$ under which governance eliminates contamination.


Copyright: © 2026 by the authors

Section II surveys related work across five research areas. Section III defines the threat model. Section IV presents the governance architecture and algorithm. Section V derives the contamination propagation model. Section VI discusses limitations and open problems, and Section VII concludes.

2. RELATED WORK

2.1 Agent Memory Architectures

MemGPT [10] introduced hierarchical memory for LLM agents that pages context between main and external storage. Generative Agents [11] demonstrated persistent memory for social simulation through reflection and retrieval. Neither system evaluates the trustworthiness of stored information; both assume that committed memory is reliable. Voyager [12] accumulates a skill library as persistent memory but applies no provenance tracking. The proposed framework extends these architectures with a governance layer that continuously evaluates and classifies memory objects, adding a security dimension that prior agent memory systems do not address.

2.2 Retrieval-Augmented Generation and Memory Poisoning

Retrieval-augmented generation [9] grounds LLM responses in externally retrieved documents. Corpus poisoning attacks [13] show that adversarially injected documents can skew retrieved context while evading similarity-based filters. PoisonedRAG [14] extends these attacks to persistent structured knowledge stores. Memory Governance complements retrieval sanitization: rather than filtering at query time, it governs what is stored and at what trust level, catching poisoned inputs before they enter the active memory pool.

2.3 Trust and Reputation Management

EigenTrust [15] and PeerTrust [16] compute node reputation in peer-to-peer systems from historical interaction records. The Beta reputation system [17] provides a Bayesian framework for updating trust given binary feedback. These models are not designed for agent memory, where the trust object is a stored memory item rather than a peer node, and where time-elapsed decay is a critical security factor. The trust scoring component adapts weighted multi-source trust to this context with an explicit normalization constraint.

2.4 Forgetting, Decay, and Information Lifecycle

Ebbinghaus [18] established the empirical exponential forgetting curve, showing that memory retention decays without reinforcement. Catastrophic forgetting [19] describes unintended memory loss in neural networks during sequential training. Memory Governance adapts exponential decay as an intentional security mechanism: rather than modeling unintended forgetting, it reduces the influence of unverified memories over time, ensuring that unrefuted adversarial inputs lose their planning influence even when they cannot be identified and removed immediately.

2.5 Information Provenance

The W3C PROV standard [20] specifies a data model for tracking information origin and transformation history. Scientific workflow provenance systems [21] apply lineage tracking to computational processes. These systems record origin but do not evaluate trustworthiness or apply access controls. Memory Governance uses provenance metadata as the primary input to trust assessment, giving provenance a direct operational role in access classification rather than treating it as audit-only metadata.

Table I summarizes these five research areas against the proposed framework across eight capability dimensions. The gap is consistent: no prior system provides trust assessment, confidence decay, memory quarantine, and continuous governance policies in combination.

Capability	RAG [9]	MemGPT [10]	Prov. Sys.	GenAgents [11]	This Work
Source tracking	Partial	Partial		Limited	

Trust assessment			Partial	Partial	
Confidence decay					
Memory quarantine					
Contam. detection				Limited	
Governance policies			Partial	Limited	
Cognitive state prot.					

TABLE I:COMPARISON OF MEMORY APPROACHES

3. THREAT MODEL

We consider an autonomous AI agent with persistent memory, retrieval mechanisms, tool execution, and long-horizon planning. The adversary aims to influence future behavior by introducing misleading information into memory stores through fabricated observations, false task completion records, manipulated preferences, poisoned retrieved documents, or embedded instructions [1][4][5]. The objective is to alter planning decisions that manifest gradually rather than cause immediate observable compromise.

The adversary can inject information through any interface reaching the memory ingestion pipeline but cannot directly modify the governance layer or audit log, which are treated as isolated trusted components. Source reliability scores S are themselves susceptible to manipulation: a patient adversary may establish a history of reliable behavior before injecting a poisoned memory. The trust formula counters this by weighting S at only 0.3 and requiring independent corroboration through V and H channels that a source-identity attack cannot trivially forge. Prompt injection [5] represents a related attack surface; Memory Governance complements prompt-level defenses by governing what injected content, once stored, can persist and influence.

Fig. 3 contrasts the attack lifecycle with and without governance, marking the two interception points: provenance verification at ingestion and trust-decay evaluation during continuous re-assessment.

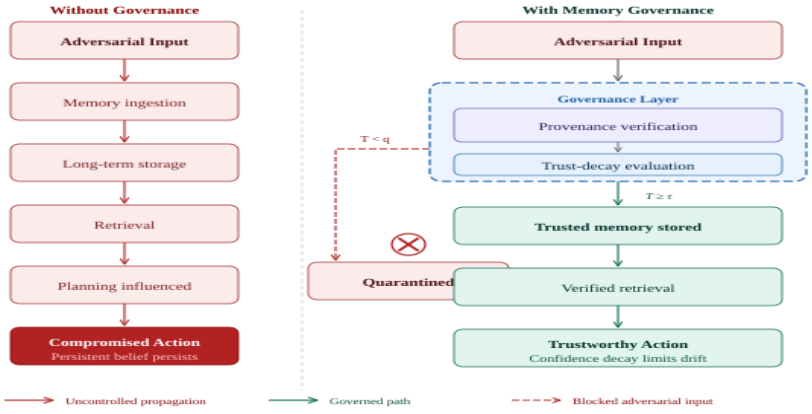

Fig. 3. Cognitive State Trap lifecycle without governance (left) and with Memory Governance (right). Without governance, adversarial inputs propagate unconstrained from ingestion through planning. With governance, provenance verification and trust-decay evaluation intercept low-trust inputs, routing them to quarantine while the governed path proceeds to trusted storage and verified retrieval.

4. MEMORY GOVERNANCE FRAMEWORK

A. Memory Object Schema

Each memory object m carries five typed fields:

m = {

source_id : string

timestamp : datetime

C_0 : float ∈ [0,1]

verif_history: list[event]

gov_state : {Trusted, Review, Quarantine}

}

C_0 is the initial confidence score assigned at ingestion from source context. verif_history accumulates validation events that update S, V, and H over the object's lifetime. gov_state records the current governance classification. Current confidence is computed dynamically from C_0 and elapsed time (Section IV-D) rather than stored as a separate mutable field, avoiding redundancy with the immutable C_0.

B. Provenance Verification

Before any memory object enters the repository, its source is verified against a provenance record carrying: source identifier, creation timestamp, prior validation events, and a digital signature over the source chain. Objects that fail signature verification are rejected at ingestion and logged. Objects with unknown source identifiers receive a low initial S score. High S scores require a verifiable chain of prior validated interactions, which counters source identity fabrication: an adversary cannot claim a high-reliability identity without a corresponding signature history that the governance layer can verify.

C. Trust Assessment

Trust is computed as a weighted combination of four evidence sources:

$$T = \alpha S + \beta V + \gamma H + \delta \cdot conf$$

where S is source reliability, V is the most recent validation score, H is historical accuracy across prior interactions, and conf is the current confidence value (Section IV-D). All inputs are normalized to $[0, 1]$. The weights satisfy $\alpha + \beta + \gamma + \delta = 1$ with $\alpha, \beta, \gamma, \delta \geq 0$, ensuring $T \in [0, 1]$ and making the thresholds τ and q directly interpretable as probability-like trust levels. Default weights ($\alpha = 0.3$, $\beta = 0.3$, $\gamma = 0.2$, $\delta = 0.2$) may be tuned per deployment context.

D. Confidence Decay

Memory reliability decreases over time without reinforcement. The time-elapsed confidence for object m at evaluation time t is:

$$conf(m, t) = C_0 \cdot e^{-\lambda(t - t_0)}$$

where $\lambda > 0$ is the decay rate (default $\lambda = 0.03$ per day) and $(t - t_0)$ is the object's age in days. This exponential formulation is consistent with Ebbinghaus's empirical forgetting function [18] and ensures that unverified memories approach zero influence as age increases. When corroborating evidence arrives, C_0 is updated to the current conf value plus a reinforcement increment Δ, resetting the effective decay clock.

E. Contamination Containment

The framework classifies each memory object into one of three governance states using thresholds $0 < q \leq \tau \leq 1$. An object with $T < q$ enters the Quarantine store: retrieval is blocked and the object is isolated from the planning context. An object with $q \leq T < \tau$ enters the Review Queue: it may be retrieved with a flagged confidence annotation but cannot influence planning without operator review. An object with $T \geq \tau$ is admitted to Trusted Memory with full retrieval access. All three states are reversible: re-evaluation on new evidence can promote a quarantined object or demote a trusted one.

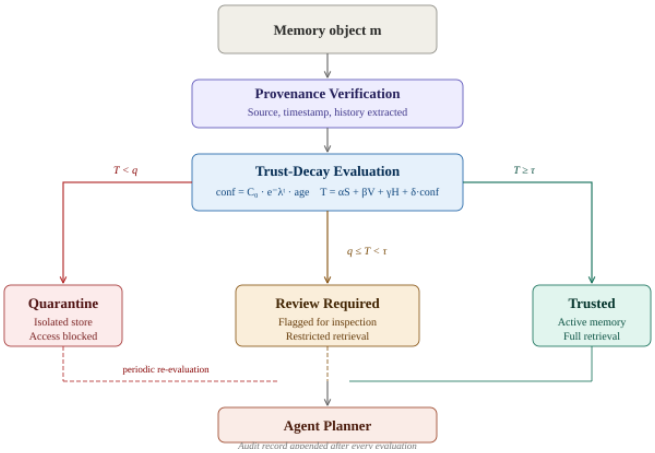

Fig. 1. Trust-Decay Memory Evaluation decision flow. Memory objects pass through provenance verification and trust-decay evaluation. Three outcome branches are separated by thresholds q and τ. Quarantined and reviewed objects undergo periodic re-evaluation; all admitted paths converge to the agent planner. An audit record is appended after every evaluation.

F. Trust-Decay Memory Evaluation Algorithm

The evaluation procedure runs over the active memory set at configurable intervals:

Input: Memory set M, thresholds τ, q, decay rate λ

for each $m \in M$ do

age ← CurrentTime − m.timestamp

conf ← $m.C_0 \times \exp(-\lambda \times \text{age})$

T ← $\alpha \cdot S(m) + \beta \cdot V(m) + \gamma \cdot H(m) + \delta \cdot \text{conf}$

if T < q then Quarantine(m)

elif T < τ then Review(m)

else Trust(m)

AuditLog.append(m.id, T, decision, age)

end for

return Updated M

Time complexity is O(N) per evaluation cycle where $N = |M|$. Space complexity is O(N) for the memory set plus $O(N \cdot E)$ for verification histories of average length E. The audit log provides a complete history of governance state transitions, supporting forensic analysis after a suspected poisoning event.

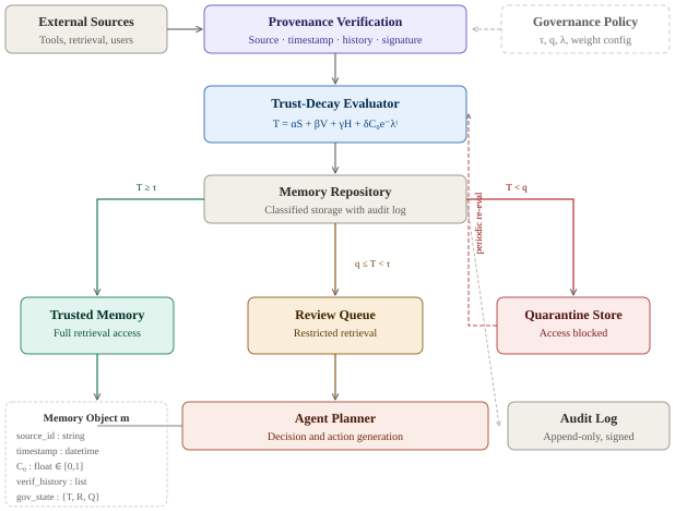

Fig. 2. Memory Governance architecture. External inputs pass through provenance verification and the Trust-Decay Evaluator before reaching the Memory Repository. The repository routes objects to Trusted Memory, Review Queue, or Quarantine Store based on thresholds τ and q. Governance Policy configures evaluator parameters. The memory object schema (bottom left) and Audit Log (bottom right) are auxiliary components.

5. CONTAMINATION PROPAGATION MODEL

To characterize the long-term effectiveness of governance controls, we model memory contamination dynamics using a discrete-time formulation adapted from compartmental epidemiological models [22]. Let M(t) denote total memory objects and C(t) denote contaminated objects at time step t. Contamination density is $\rho(t) = C(t)/M(t)$.

Without governance, contaminated objects influence future memory creation through retrieval-conditioned generation: retrieved contaminated context biases the generation of new observations, which are then stored, spreading contamination. The propagation follows the discrete logistic growth equation [22]:

$$C_{t+1} = C_t + \beta C_t(1 - C_t/M_t)$$

where β is the contamination propagation rate. Introducing governance remediation at rate μ, representing the fraction of contaminated objects identified and quarantined per cycle:

$$C_{t+1} = C_t + \beta C_t(1 - C_t/M_t) - \mu C_t$$

A. Steady-State Analysis

At steady state, $\Delta C = 0$. Setting the governed propagation equation to zero and factoring out C_t (assuming $C_t > 0$):

$$\beta(1 - C^*/M) = \mu$$

Solving for the non-trivial steady-state contaminated count:

$$C^* = M(1 - \mu/\beta)$$

This yields three operational regimes. When $\mu > \beta$, $C^* < 0$, which is inadmissible - the only consistent interpretation is that contamination is driven to zero before a positive equilibrium can form. When $\mu = \beta$, $C^* = 0$,

representing the critical governance threshold. When $\mu < \beta$, $C^* = M(1 - \mu/\beta) > 0$, representing a persistent contamination equilibrium.

Property (Contamination Elimination): Let M be constant or grow sublinearly relative to the remediation capacity. A memory governance system with remediation rate $\mu > \beta$ drives contamination density $\rho(t) \to 0$ as $t \to \infty$. Under unbounded memory growth, the condition generalises to $\mu > \beta(1 - C_t/M_t)$ evaluated at the operating point.

Memory Governance targets $\mu > \beta$ through two mechanisms: continuous evaluation increases the rate at which contaminated objects are identified and quarantined (increasing μ), while decay-driven demotion reduces the planning influence of borderline objects before they condition new memories (effectively reducing β by interrupting the retrieval-generation-storage chain).

6. DISCUSSION

6.1 Adversarial Source Reliability

The trust formula's partial dependence on S creates a slow-burn vulnerability: an adversary who establishes a history of reliable behavior before injecting poisoned memory can arrive with a high S score. Two architectural controls bound this risk. The weight α on S is capped at 0.3 by default, so even a maximized S score contributes at most 0.3 to T; the remaining 0.7 must be earned through independent validation (V) and historical accuracy (H) channels. Additionally, provenance verification requires a cryptographic signature chain; fabricating a high-S identity requires forging that chain. Operators in high-threat environments should reduce α and increase the weight on H, which reflects long-run agent-observable accuracy rather than self-reported source identity.

6.2 Evaluation Overhead

The O(N) evaluation algorithm over the full memory set may introduce latency for agents with large stores ($N > 10^4$ objects). Three practical mitigations reduce this: prioritizing recently ingested or modified objects, batching re-evaluation of stable high-trust objects at longer intervals, and triggering immediate re-evaluation only on contradiction detection. These trade-offs can only be characterized empirically, which requires production agent deployments and evaluation infrastructure that this paper leaves to future work.

6.3 Policy Composition

When multiple governance policies apply simultaneously - a data sensitivity policy, a source-domain policy, and a temporal policy may all assign different thresholds τ and q to the same object, the framework resolves conflicts using a precedence rule: the most restrictive threshold wins. An object subject to both $q = 0.35$ and $q = 0.50$ is evaluated against $q = 0.50$. Policy conflicts are recorded in the audit log rather than silently resolved.

6.4 Limitations

Three limitations bound the current framework. The contamination propagation model treats memory objects as independent, but semantically related memories may share contamination pathways through associative retrieval, a structure the logistic model does not capture. The trust weight calibration has no automated learning mechanism; weights are operator-configured rather than learned from agent experience, requiring domain expertise to set correctly in adversarial environments. Most significantly, the framework has not been empirically validated against real agent deployments; no benchmark exists for evaluating memory governance effectiveness. Memory governance benchmarks, analogous to adversarial robustness benchmarks for classifiers, are an open research problem that this work identifies but cannot resolve.

Table II summarizes the expected behavioral differences between a conventional agent and an agent operating under Memory Governance across six representative scenarios.

Scenario	Conventional Agent	Governed Agent
Poisoned memory persistence	Indefinite	Decays to quarantine
Trust reassessment	None	Continuous per cycle
Forgery detection	None	Provenance mismatch flagged
Contamination spread	Unconstrained (β)	Bounded when $\mu > \beta$
Memory explainability	None	Full audit trail per object
Recovery from poisoning	**Manual or none**	**Automatic via re-evaluation**

TABLE II: **Illustrative Behavioral Comparison**

7. CONCLUSION

When an AI agent commits information to memory, it effectively grants that information indefinite planning influence. Existing defenses operate at interaction boundaries and cannot revoke that grant. Memory Governance addresses this structural gap by treating memory as a security asset subject to continuous oversight: each stored object carries provenance metadata, is scored against a weighted trust formula with decaying confidence, and is classified into one of three governance states that control its retrieval access.

Three design choices work in concert. Provenance verification reduces the probability that adversarial inputs enter storage with a high initial trust score. The weighted trust formula with normalized weights ensures that a single evidence channel, even one under adversarial control, cannot drive an object to the Trusted state alone. Confidence decay ensures that unverified objects lose planning influence over time regardless of whether they are identified as contaminated, creating a time-bounded window for adversarial influence even when governance is imperfect.

The framework is conceptual and requires empirical validation. Future work should investigate adaptive weight learning from agent feedback, multi-agent contamination dynamics where compromised agents introduce poisoned memory into shared pools, and benchmark design for memory governance evaluation. These directions are prerequisites for deploying memory governance in production agent systems operating under NIST AI RMF [3] requirements for high-risk autonomous systems.

REFERENCES

1. **Franklin et al.**, "AI Agent Traps: Cognitive State Manipulation in Autonomous LLM Agents," 2026. [Preprint]
2. I. Rahwan *et al.*, "Machine behaviour," *Nature*, vol. 568, pp. 477–486, 2019.
3. NIST, *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*, National Institute of Standards and Technology, 2023.
4. OWASP, "OWASP Top 10 for Large Language Model Applications," v2025, Open Worldwide Application Security Project, 2025.
5. K. Greshake *et al.*, "Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection," in *Proc. ACM AISec*, 2023.
6. OpenAI, "Safety Evaluations for Autonomous Agents," Technical Report, OpenAI, 2025.
7. Anthropic, "Agent Security Considerations," Technical Report, Anthropic, 2025.
8. P. Lewis *et al.*, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in *Proc. NeurIPS*, 2020.
9. C. Packer *et al.*, "MemGPT: Towards LLMs as Operating Systems," *arXiv:2310.08560*, 2023.
10. J. S. Park *et al.*, "Generative Agents: Interactive Simulacra of Human Behavior," in *Proc. ACM UIST*, 2023.
11. G. Wang *et al.*, "Voyager: An Open-Ended Embodied Agent with Large Language Models," *arXiv:2305.16291*, 2023.
12. J. Zhong *et al.*, "Poisoning Retrieval Corpora by Injecting Adversarial Passages," in *Proc. EMNLP*, 2023.
13. W. Zou *et al.*, "PoisonedRAG: Knowledge Poisoning Attacks to Retrieval-Augmented Generation of LLMs," *arXiv:2402.07867*, 2024.
14. S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The EigenTrust Algorithm for Reputation Management in P2P Networks," in *Proc. WWW*, 2003.
15. L. Xiong and L. Liu, "PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 7, pp. 843–857, 2004.
16. A. Jøsang and R. Ismail, "The Beta Reputation System," in *Proc. 15th Bled Elec. Commerce Conf.*, 2002.

17. H. Ebbinghaus, *Über das Gedächtnis*, Duncker & Humblot, 1885. (Trans. *Memory: A Contribution to Experimental Psychology*, 1913.)
18. M. McCloskey and N. J. Cohen, "Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem," *Psychol. Learn. Motiv.*, vol. 24, pp. 109–165, 1989.
19. W3C, "PROV-DM: The PROV Data Model," W3C Recommendation, Apr. 2013. [Online]. Available: https://www.w3.org/TR/prov-dm/
20. P. Missier, K. Belhajjame, and J. Cheney, "The W3C PROV Family of Specifications for Modelling Provenance Metadata," in *Proc. EDBT*, 2013.
21. W. O. Kermack and A. G. McKendrick, "A Contribution to the Mathematical Theory of Epidemics," *Proc. Royal Soc. London A*, vol. 115, pp. 700–721, 1927.