Memory-Governed Decision Systems: A Formal Framework for AI Governance Across Heterogeneous Execution Surfaces

Michael Lord Beckrest
Graduate Student, Master of Business Creation
David Eccles School of Business, University of Utah
Founder, GENYS AI Inc.

March 2026

Abstract

Governance in artificial intelligence systems is commonly framed as constraint enforcement, compliance alignment, or oversight instrumentation. Yet deployed AI systems continue to exhibit persistent failure modes: context loss across interactions, cross-platform inconsistency, policy drift, non-reproducibility of decisions, and incomplete audit trails. We argue that these are not primarily model-capability failures but *governance-structure failures* rooted in the inadequate treatment of memory as a first-class system primitive. We introduce **Memory-Governed Decision Systems (MGDS)**: a formal paradigm in which memory is versioned, policy-bound, auditable, and constitutive of downstream action authorization. We formalize MGDS as a transition system $(\mathcal{M}, \mathcal{P}, \mathcal{D}, \mathcal{A})$, ground governance across three interdependent operational tiers (technical, institutional, enterprise) with an explicit integration principle, define seven measurable constructs for empirical evaluation, present a taxonomy and reference architecture, and specify an evaluation protocol with falsifiable hypotheses and baselines. This paper presents a conceptual framework; empirical validation via the proposed benchmark suite is identified as the primary avenue for future work. The central claim is falsifiable: if deployed governance failures can be resolved without first-class memory governance, the paradigm is unnecessary.

Keywords: AI governance; memory systems; auditability; decision systems; omnichannel governance; heterogeneous execution; governance-structure failures

1 Introduction

1.1 Governance in Artificial Intelligence

Governance in AI systems is typically defined along three axes:

Constraint Governance. Governance as enforcement of constraints over actions. Systems are governed when their outputs satisfy formal, regulatory, or policy constraints.

Accountability Governance. Governance as traceability and auditability. Systems are governed when decisions are reproducible, attributable, and explainable post hoc.

Lifecycle Governance. Governance as control over state transitions. Systems are governed when changes to system state occur under explicit authorization and version control.

Across all three definitions, a shared structural requirement emerges: *governance presupposes persistent, inspectable state*. Constraint enforcement requires evaluating state; accountability requires recording state transitions; lifecycle control requires versioned state history.

In practice, many deployed AI systems treat memory as auxiliary infrastructure rather than as the substrate of governance itself. The consequence is a class of governance-structure failures that are neither detected nor resolved by improvements to model capability alone.

1.2 Observed Deployment Failures

Documented deployment patterns consistent with governance-structure failures include loss of user preferences or constraints across sessions; divergent actions across heterogeneous platforms; inability to reconstruct why a decision changed or a policy was violated; policy violations discovered only after external rejection; and drift under distribution shift without state provenance.

These patterns are consistent with insufficiently governed memory, rather than insufficient model capability. Whether memory governance is the *primary* driver is an empirical question addressed by the hypotheses and benchmark protocol in Sections 1.5 and 12.

1.3 From Governance to Memory Governance

If governance requires persistent state, and AI systems operate through iterative state transitions, then memory is the substrate through which governance must operate.

Memory Governance is the binding of policy, authorization, and audit semantics directly to system memory state and its transitions.

This leads to the central construct of this paper: Memory-Governed Decision Systems (MGDS).

1.4 Contributions

1. **Governance framing.** A three-axis definition of AI governance (Constraint, Accountability, Lifecycle) establishing that governance presupposes persistent, inspectable state.
2. **Formal framework.** Formalization of MGDS as a transition system $(\mathcal{M}, \mathcal{P}, \mathcal{D}, \mathcal{A})$ with transition semantics and formal proofs of replay-auditability and non-repudiation.
3. **Three-tier governance.** An operational definition spanning technical, institutional, and enterprise levels, with an integration principle closing the chain from policy to measured outcomes.
4. **Measurable constructs.** Seven operationalized governance constructs with observable indicators enabling empirical evaluation.
5. **Research design.** Explicit analytical framework, research gap statement, and separation of normative from empirical claims.
6. **Taxonomy.** A four-level taxonomy of memory substrates, lifecycle stages, governance layers, and execution surfaces.
7. **Autonomy and continuity.** A formal treatment of decision continuity as a property of governed memory transition, not agent identity.
8. **Reference architecture.** An implementable four-plane architecture with threat model.
9. **Neuromorphic connection.** A structural mapping from neuromorphic principles to governance mechanisms.
10. **Distinctness argument.** Failure-mode analysis showing MGDS is not subsumed by adjacent categories.
11. **Benchmark suite.** Seven metrics, six task families, falsifiable hypotheses, and experimental methodology.
12. **Case study.** Omnichannel advertising governance for SMB merchants, with hypothetical structured metrics.

1.5 Falsifiable Hypotheses

H1. Memory governance reduces policy violation under distribution shift, as measured by Policy Violation Rate (PVR) and Constraint Satisfaction Rate (CSR).

H2. Memory governance increases audit completeness and replay fidelity, as measured by Audit Completeness (AC) and Replay Fidelity (RF).

H3. Event-triggered governance reduces mean time to detection and mean time to correction of policy violations, as measured by Governance Latency (GL) and Time to Repair (TTR).

These hypotheses are empirically testable and may be falsified. If any hypothesis is systematically refuted under controlled conditions satisfying the methodology in Section 12, the paradigm requires revision.

2 Background and Related Work

2.1 Memory in Cognitive and Neuroscience

The scientific study of memory provides the conceptual foundation for our framework. Tulving [48] distinguished episodic memory (personally experienced events bound to spatiotemporal context) from semantic memory (general world knowledge abstracted from

episodes). Squire [43, 44] proposed a taxonomy of declarative and non-declarative memory systems, supported by double dissociations in amnestic patients. Working memory [5] provides a capacity-limited buffer for active manipulation of information. Biological memory is a *system of systems* [27] with distinct substrates, consolidation dynamics, and retrieval mechanisms.

2.2 Neuromorphic Computing

Neuromorphic engineering [28] builds computing systems where memory and computation are co-located. Spiking neural networks [25] process information via discrete temporal events, enabling event-driven, asynchronous computation. Hardware implementations include IBM's TrueNorth [29], Intel's Loihi [12], and BrainScaleS [37]. Schuman et al. [39] identify memory-compute co-location as the defining architectural principle. We draw on neuromorphic principles as a *structural analogy*, not a claim of physical equivalence.

2.3 Retrieval-Augmented Generation

Lewis et al. [24] introduced RAG, combining parametric language models with non-parametric retrieval over document stores. RAG treats memory as a read-only retrieval target without versioning, policy constraints, or governance semantics.

2.4 Agent Frameworks and Tool Use

LLM-based agents [50, 38, 34] introduce tool use, planning, and multi-step execution. Park et al. [32] demonstrated generative agents with memory streams, but these memories are unversioned, ungoverned, and lack policy enforcement. Frameworks such as LangChain [8] treat memory as a convenience layer rather than a governed substrate.

2.5 AI Governance, Safety, and Policy

The NIST AI Risk Management Framework [30] calls for governance mechanisms throughout the AI lifecycle. The EU AI Act [15] imposes requirements including transparency, human oversight, and record-keeping for high-risk systems. Russell [36] argues for AI systems deferential to human oversight. Amodei et al. [2] enumerate concrete safety problems including distributional shift and reward hacking. Open Policy Agent [45] provides request-time policy decisions without integrated memory semantics.

2.6 Event Sourcing and Distributed Systems

Lamport [23] established logical clocks for ordering events in distributed systems. Brewer's CAP theorem [7, 18] constrains the simultaneous achievement of consistency, availability, and partition tolerance. Event sourcing [16] stores state as an append-only log, enabling replay, audit, and temporal queries.

2.7 Feature Stores and MLOps

Feature stores [46] provide versioned, point-in-time-correct feature serving but address feature engineering, not decision governance. MLOps [40] treats the model as the central artifact rather than the memory substrate. Neither provides policy enforcement on memory state or audit semantics for governed actions.

2.8 Enterprise Architecture and Organizational Governance

TOGAF [47] organizes enterprise architecture across business, data, application, and technology layers. TOGAF's information architecture layer addresses data governance structurally but does not specify mechanisms for governing AI decision records with policy-bound lifecycle semantics. MGDS can be positioned as a governance substrate instantiating TOGAF's information layer for AI decision memory specifically.

COBIT 2019 [21] establishes IT governance objectives spanning processes, structures, and information. COBIT's scope is enterprise IT governance, not system-level decision execution governance. The three-tier governance model (Definition 3) is a COBIT-compatible instantiation for AI decision systems.

Organizational decision theory. Simon's bounded rationality [42] and Cyert and March's behavioral theory of the firm [10] establish that organizational decision-making is distributed, history-dependent, and reliant on memory and satisficing. MGDS operationalizes these insights architecturally: the Memory Plane encodes decision history; the Governance Plane encodes constraint structure; the Feedback Plane closes the organizational learning loop.

Platform economics. Rochet and Tirole [35] and Parker et al. [33] characterize multi-sided platforms as coordinating heterogeneous participants across distinct network sides. Omnichannel execution surfaces are structurally analogous: value creation depends on cross-surface coordination that requires governed memory consistency to sustain.

Data governance. DAMA-DMBOK [11] defines data governance over passive data assets. MGDS extends this to active decision records that are policy-enforcing at retrieval time and constitutive of action authorization.

Design science methodology. This paper follows Hevner et al. [19], classifying MGDS as a *model artifact*: a prescriptive conceptual framework evaluated through utility in the target environment (the benchmark suite, Section 12) and rigor of theoretical grounding.

3 Formal Definitions

Definition 1 (Memory in AI Systems)**.** *Let $\mathcal{M}$ be a memory system comprising five components:*

- ***Working memory*** *$\mathcal{M}_w$: a capacity-bounded buffer of actively maintained state.*
- ***Episodic memory*** *$\mathcal{M}_e$: a temporally indexed store of interaction events, each tagged with provenance (source, timestamp, context id).*
- ***Semantic memory*** *$\mathcal{M}_s$: a structured or embedded store of domain knowledge and factual assertions, decoupled from specific episodes.*
- ***Procedural memory*** *$\mathcal{M}_p$: encoded action patterns, tool-use schemas, and learned operational routines.*
- ***Decision memory*** *$\mathcal{M}_d$: an append-only ledger of decision records $\langle state, action, justification, policy\ version, outcome\rangle$ constituting the audit trail.*

Formally,

$$\mathcal{M} = (\mathcal{M}_w, \mathcal{M}_e, \mathcal{M}_s, \mathcal{M}_p, \mathcal{M}_d).$$

Definition 2 (Governance in AI Systems)**.** *Governance G is a tuple $(C, E, A, Auth)$ where:*

- *C: constraint predicates over memory states and proposed actions.*
- *E: enforcement function returning $\{allow, deny, modify\}$.*
- *A: audit recording function $A : \mathcal{D} \to \mathcal{A}$ mapping each decision to an immutable record.*
- *$Auth$: authorization scope binding identities and roles to permitted memory and action scopes.*

Definition 3 (Three-Tier AI Governance (Operational))**.** *AI governance operates across three interdependent tiers corresponding to the three axes of Section 1:*

- ***Technical (Constraint Governance)****: Controls, logging, and enforcement layers ensuring decisions are traceable and controllable. Instantiated in* MGDS *by the Governance Plane ($\mathcal{P}$), cryptographically chained Decision Log ($\mathcal{A}$), Constraint Solver, and event-triggered enforcement mechanisms (Section 6.3).*
- ***Institutional (Accountability Governance)****: Accountability and documentation structures required by regulatory frameworks and internal oversight bodies. Instantiated in* MGDS *by versioned policy specifications with provenance metadata, human-in-the-loop Approval Gates, Attest operations for non-repudiation, and Forget operations satisfying GDPR [49], EU AI Act [15], and NIST AI RMF [30] obligations.*
- ***Enterprise (Lifecycle Governance)****: Operational policies aligning automated decisions with organizational objectives, risk tolerance thresholds, and supervisory review. Instantiated in* MGDS *by first-class policy parameters in $\mathcal{P}$ and Approval Gates routing decisions above risk threshold θ_{esc} to designated human supervisors.*

Remark 1 (Governance Integration Principle)**.** *The three tiers form a closed, ordered governance chain:*

$$\begin{aligned} policy &\longrightarrow technical\ controls \\ &\longrightarrow audit\ trail \\ &\longrightarrow measured\ outcomes \end{aligned} \tag{1}$$

The transition function τ (Definition 5) directly instantiates this chain. Any governance framework omitting a link does not satisfy the accountability and lifecycle requirements of either the institutional or enterprise tier.

Definition 4 (Omnichannel Execution Surface)**.** *An omnichannel execution surface $\mathcal{E} = \{e_1, \ldots, e_n\}$ is a set of heterogeneous execution targets where each e_i has a local constraint set $C_i \subseteq C$, a local schema S_i, a feedback function $f_i : outcomes_i \to \mathcal{F}$, and no guarantee of unified state without explicit synchronization.*

Definition 5 (Memory-Governed Decision System (MGDS))**.** *A Memory-Governed Decision System is a four-tuple $\Sigma = (\mathcal{M}, \mathcal{P}, \mathcal{D}, \mathcal{A})$ with transition function τ. Given request r and memory state $\mathcal{M}_t$:*

1. ***Retrieve**: $m \leftarrow retrieve(\mathcal{M}_t, r)$.*
2. ***Evaluate**: $v \leftarrow \mathcal{P}(m, r)$, returning verdict $v \in \{allow(a), deny(reason), modify(a')\}$ and justification trace j.*
3. ***Execute**: Dispatch a (or a') to $e_i \in \mathcal{E}$ if $v \neq deny$.*
4. ***Audit**: Record $d = \langle r, m, v, j, a, t, policy\_version \rangle$ in $\mathcal{A}$ (append-only).*
5. ***Update**: $\mathcal{M}_{t+1} \leftarrow update(\mathcal{M}_t, d, f_i(outcome))$ subject to governance constraints.*

The transition $\tau : (\mathcal{M}_t, r) \rightarrow (\mathcal{M}_{t+1}, d)$ is deterministic given fixed $\mathcal{P}$.

4 Research Design and Methodology

4.1 Research Gap Statement

We identify three specific gaps that MGDS is designed to address.

Gap 1: No measurable construct for cross-surface governance fragmentation. No published framework defines a measurable construct for the degree of governance divergence across heterogeneous AI execution surfaces, making fragmentation a qualitative concern rather than a manageable variable.

Gap 2: Memory as ungoverned infrastructure. Existing AI system categories (RAG, vector databases, agent frameworks) treat memory as retrieval infrastructure or ephemeral context. No framework treats memory as a governed first-class primitive with its own policy surface, lifecycle state machine, versioning semantics, and audit trail.

Gap 3: No unified architecture spanning all three governance tiers. Existing frameworks address individual tiers: technical enforcement (OPA [45]), institutional compliance (NIST [30], EU AI Act [15]), and enterprise alignment (TOGAF [47], COBIT [21]). No single architecture closes the integration chain (Eq. **??**) across all three tiers simultaneously.

4.2 Analytical Framework

This paper follows the design science research paradigm [19]. MGDS is a *model artifact*. The analytical methods are:

Formal systems modeling: mathematical formalization with defined transition semantics and formal proofs.

Comparative failure-mode analysis: systematic evaluation of adjacent categories against a common failure-mode taxonomy.

Illustrative case study: structured instantiation demonstrating applicability to a concrete domain.

Benchmark design: specification of the experimental methodology for empirically testing H1–H3.

4.3 Scope and Claim Types

Formal claims are mathematically provable from stated definitions and hold conditionally on stated assumptions.

Normative claims assert what governance should look like. They reflect a design position grounded in the governance literature and do not carry empirical force until instantiated and measured.

Empirical claims (H1–H3) assert observable facts about system behavior under specified conditions. At this stage they are hypotheses: motivated by structural analysis but not yet tested.

The paper's architectural contribution (formal and normative claims) does not depend on the truth of H1–H3. What changes if H1–H3 are refuted is the relative priority of MGDS *compared to complementary approaches, not the validity of the architecture itself.*

4.4 Measurable Constructs

We define seven measurable constructs that operationalize the governance properties of MGDS. These form the measurement backbone of the benchmark suite (Section 12).

Definition 6 (Policy Violation Rate (PVR))**.**

$$PVR = \frac{|\{a \in A : \exists\, c \in C,\ c(a) = false\}|}{|A|} \tag{2}$$

Observable indicator*: proportion of executed actions violating at least one constraint predicate, measured over a defined observation window.*

Definition 7 (Constraint Satisfaction Rate (CSR))**.**

$$CSR = 1 - PVR \tag{3}$$

Observable indicator*: proportion of executed actions satisfying all applicable constraints. Reported alongside PVR as the positive complement.*

Definition 8 (Cross-Surface Consistency Error (CSCE))**.** *Let $S = \{s_1, \ldots, s_n\}$ be the set of execution surfaces and $D_t(s_i)$ the decision-relevant memory state at surface s_i at time t:*

$$CSCE(t) = \frac{|\{(i,j) : D_t(s_i) \not\sim_C D_t(s_j),\ i \neq j\}|}{\binom{n}{2}} \tag{4}$$

where $\not\sim_C$ denotes inconsistency under global cross-surface constraints. Observable indicator*: proportion of surface pairs in inconsistent decision state per observation period. Lower CSCE indicates better cross-surface consistency.*

Definition 9 (Audit Completeness (AC))**.**

$$AC = \frac{|\{d \in \mathcal{A} : d \textit{ fully logged with } policy\_version\}|}{|\mathcal{A}|} \tag{5}$$

Observable indicator*: proportion of decisions for which all required audit fields are present and linked to a versioned policy.*

Definition 10 (Replay Fidelity (RF))**.** *Let $replay(d)$ denote reconstruction of decision d from logged memory and policy versions:*

$$RF = \frac{|\{d \in \mathcal{A} : replay(d) = verified\}|}{|\mathcal{A}|} \tag{6}$$

Observable indicator*: proportion of logged decisions for which memory state, policy version, and verdict can be reconstructed and independently verified. RF $\geq$ AC since replay requires completeness plus version availability.*

Definition 11 (Governance Latency (GL))**.** *Let $t_{eval}(e)$ be the elapsed time for the Governance Plane to evaluate decision event e:*

$$GL = \mathbb{E}_e[t_{eval}(e)] \tag{7}$$

Observable indicator*: mean and p99 governance evaluation latency in milliseconds, measured separately for fast-path (cached) and full-evaluation paths.*

Definition 12 (Policy Violation Drift Rate (PVDR))**.** *Let $p_v(t)$ be the PVR at time t. The PVDR over interval Δt is:*

$$PVDR(\Delta t) = \frac{p_v(t + \Delta t) - p_v(t)}{\Delta t} \tag{8}$$

Observable indicator*: signed change in PVR following a defined system event (model update, distribution shift, policy version change). Non-zero PVDR under distribution shift is the primary indicator for H1.*

Table 1: Measurable constructs, hypothesis mapping, and benchmark task linkage.

Construct	Observable Indicator	Hyp.	Task
PVR	Action constraint violation rate	H1	T2, T5
CSR	Constraint satisfaction rate (1−PVR)	H1	T2, T5
CSCE	Cross-surface state inconsistency rate	H1	T2
AC	Audit field completeness rate	H2	T4
RF	Decision replay verification rate	H2	T4
GL	Governance evaluation latency (ms)	H3	T6
PVDR	PVR change rate after system events	H1, H3	T3

Figure 2: Taxonomy of Memory-Governed Decision Systems

Level 1: Memory Substrates
Symbolic | Vector | Relational | Ledger | Neuromorphic | Hybrid

Level 2: Memory Lifecycle
Ingest → Normalize → Store → Retrieve → Update → Forget → Version → Attest

Level 3: Governance Layers
Permissioning | Constraints | Consistency | Rate Limits | Safety
Compliance (Institutional) | Enterprise Alignment | Provenance

Level 4: Execution Surfaces & Sync
Agents | APIs | Platforms | Channels
Sync: Eager | Lazy | Eventual | CRDT

Figure 1: Four-level taxonomy of MGDS.

Table 1 summarizes the seven constructs with their hypothesis mapping and benchmark task linkage.

5 A Taxonomy of Memory-Governed Systems

5.1 Level 1: Memory Substrates

Symbolic: Rule-based or ontological representations. **Vector/Embedding**: Dense vector spaces for similarity retrieval [22, 26]. **Relational**: SQL or document stores with ACID semantics. **Ledger**: Append-only, immutable event logs. **Neuromorphic**: Synaptic weight matrices on co-located hardware [12]. **Hybrid**: Pro-

duction systems combine substrates.

5.2 Level 2: Memory Lifecycle

Ingest → **Normalize** → **Store** → **Retrieve** → **Update** → **Forget** (GDPR right to erasure [49]) → **Version** (DAG of memory states) → **Attest** (cryptographic signing for non-repudiation).

5.3 Level 3: Governance Layers

Permissioning: Role-based or attribute-based access control over memory and actions.

Constraints: Domain-specific predicates.

Consistency: Cross-surface invariant enforcement.

Rate limits: Temporal bounds on action frequency.

Safety: Guardrails preventing harmful or out-of-distribution outputs.

Compliance (Institutional): Three components: (i) documentation structures—versioned policy specifications with provenance metadata satisfying EU AI Act [15] and NIST AI RMF [30]; (ii) oversight integration—Approval Gates routing decisions above θ_{esc} to named human reviewers with auditable accountability chains; (iii) regulatory lifecycle management—Forget with attestation satisfying GDPR [49], and point-in-time replay satisfying post-hoc accountability requirements.

Enterprise Alignment: Organizational objectives, risk tolerance tuple $\rho = (r_{\min}, r_{\max}, \theta_{\text{esc}})$, and supervisory escalation rules encoded as first-class parameters in $\mathcal{P}$ (see COBIT governance objectives [21] and TOGAF enterprise layer [47]).

Provenance: End-to-end lineage tracking from memory ingest through decision to outcome.

5.4 Level 4: Execution Surfaces and Synchronization

Agents, **APIs**, **Platforms**, **Channels**. Synchronization patterns: Eager (synchronous constraint check), Lazy (post-hoc reconciliation), Eventual (convergence guarantees), Conflict-free (CRDT-based [41] merge semantics).

6 The Neuromorphic Connection

This section establishes a structural analogy—not physical equivalence—between neuromorphic computing principles and MGDS governance mechanisms.

6.1 Neuromorphic Principles

Neuromorphic systems are characterized by [39, 20]: (i) spiking neural networks with event-driven processing; (ii) synaptic plasticity implementing local policy via STDP [6]; (iii) memory-compute co-location eliminating the von Neumann bottleneck; (iv) local learning rules depending only on locally available signals.

6.2 Structural Mapping to MGDS

- **Spikes** ↔ **Decision events**: event-sourced [16] governance triggered by decision requests.
- **Synaptic weights** ↔ **Memory state**: $\mathcal{M}$ as the locus of system intelligence.
- **STDP** ↔ **Policy-constrained memory updates**: outcome-driven weight updates subject to governance bounds.
- **Neuromodulation** ↔ **Governance signals**: policies modulating permitted updates and their magnitude.
- **Local learning** ↔ **Decentralized enforcement**: per-surface local constraint enforcement.

6.3 Neuromorphic-Inspired Governance Mechanism

Definition 13 (Event-Triggered Policy Check)**.**

$$\mathit{sal}(e) = \|x_e - \bar{x}_{\mathit{recent}}\|_2 + \lambda \cdot \mathit{risk}(e) > \theta \quad (9)$$

Full policy evaluation $\mathcal{P}(m, r)$ is triggered only when $\mathit{sal}(e) \geq \theta$; otherwise a cached verdict is returned. This directly drives Governance Latency (GL, Definition 11).

Definition 14 (Plasticity-Like Memory Update Rule)**.**

$$w_i^{t+1} = \mathit{clip}\Big(w_i^t + \eta \cdot \Delta(m_i, d, o) \cdot g(\mathcal{P}, m_i),\ w_{\min},\ w_{\max}\Big) \quad (10)$$

where $g(\mathcal{P}, m_i) \in [0, 1]$ is a governance gating function and all weight changes are logged in $\mathcal{A}$.

7 Autonomy and Continuity

Autonomous systems require continuity of state across time. Continuity is not simply persistence; it is *governed* persistence.

An autonomous system without governed memory is effectively stateless across decision boundaries: each action is executed without systematic reference to the

Figure 3: Event-Triggered Governance Loop

1. Event e arrives
2. Compute sal(e) via Eq. 9
3a. sal(e) $<$ θ: fast-path (cached verdict)
3b. sal(e) $\geq$ θ: full policy eval $\mathcal{P}(m, r)$
4. Execute action a on surface e_i
5. Observe outcome o
6. Update weights via Eq. 10
7. Log $(e, v, a, o, \Delta w)$ to $\mathcal{A}$
8. If drift detected: lower θ (increase vigilance)

Figure 2: Neuromorphic-inspired event-triggered governance loop.

history of authorizations, constraints satisfied, and outcomes observed that preceded it. This is a governance-structure failure independent of model capability.

Definition 15 (Decision Continuity)**.** *A system Σ is* decision-continuous *if for every decision $d_t \in \mathcal{A}$, the memory state $\mathcal{M}_{v(t)}$ that governed d_t is recoverable and causally linked to prior decisions $\{d_j : j < t\}$ via the versioned memory DAG. Decision continuity is a necessary condition for replay-auditability (Proposition 1) but weaker than full state replayability.*

Decision continuity implies a constraint on system design: the governance substrate must be shared across execution surfaces, not siloed within each agent or platform adapter. Without a shared governance substrate, the CSCE (Definition 8) will increase over time as surface-local memory states diverge.

This framing clarifies the distinction between MGDS and agent memory frameworks: agent frameworks provide per-agent memory achieving local continuity but not cross-agent or cross-surface governance continuity. MGDS provides the governance substrate making continuity a system-level, auditable property rather than an agent-level, unverifiable one.

Continuity also has institutional implications. Regulatory frameworks requiring post-hoc accountability (EU AI Act [15]) implicitly require decision continuity: if memory states are not versioned and linked, regulatory replay requirements cannot be satisfied. MGDS formalizes decision continuity as an architectural invariant rather than an audit-time aspiration.

8 Reference Architecture

8.1 Memory Plane

Versioned Memory Store: append-only storage with content-addressable hashing (Merkle DAG), enabling temporal queries and rollback. **Embedding Index**: HNSW [26] over semantic memory $\mathcal{M}_s$ with metadata filtering. **Structured Store**: relational database for structured decision memory. **Decision Log**: append-only cryptographically chained log; each entry signed and referencing memory and policy version. **Lifecycle Manager**: orchestrates the full lifecycle (Section 5, Level 2).

API surface: `retrieve(query, filters, policy_ctx)` $\rightarrow$ `MemoryResult`; `write(entry, metadata, gov_token)` $\rightarrow$ `VersionID`; `version(id)` $\rightarrow$ `VersionDAG`; `forget(id, justification)` $\rightarrow$ `AttestReceipt`.

8.2 Governance Plane

Policy Compiler: Rego [45] or Cedar [1] to executable constraint predicates. **Constraint Solver**: SMT solving via Z3 [13] for complex constraint sets. **Verification Engine**: schema validation, budget feasibility, safety checks, cross-surface consistency. **Approval Gates**: routes decisions above θ_{esc} to human review; records reviewer identity and rationale for institutional accountability. **Policy Linter**: static analysis detecting policy conflicts and completeness gaps.

API surface: `evaluate(action, mem, policies)` $\rightarrow$ `Verdict`; `compile(spec)` $\rightarrow$ `Policy`; `lint(spec)` $\rightarrow$ `[Warning]`; `audit(id)` $\rightarrow$ `AuditRecord`.

8.3 Execution Plane

Channel Adapters: connectors translating MGDS actions to platform-specific API calls. **Agent Runtime**: agents request actions through the Governance Plane only—never directly to external systems. **API Gateway**: rate limiting and authentication. **Sandbox**: isolated environment for policy change testing.

8.4 Feedback Plane

Attribution Engine: attributes outcome signals to specific decisions and memory states. **Telemetry Collector**: aggregates latency, error rates, and resource met-

Figure 1: MGDS Reference Architecture

Memory Plane	Governance Plane	Execution Plane	Feedback Plane
Versioned Memory Store	Policy Compiler	Channel Adapters	Attribution Engine
Embedding Index	Constraint Solver	Agent Runtime	Telemetry Collector
Structured Store (SQL)	Verification Engine	API Gateway	Drift Detector
Decision Log (append)	Approval Gates	Sandbox	Evaluation Pipeline
Lifecycle Manager	Policy Linter		Anomaly Monitor

Data flows: Request → Memory Plane (retrieve) → Governance Plane (evaluate) → Execution Plane (dispatch) → Feedback Plane (observe) → Memory Plane (update). All transitions → Decision Log (audit).

Tier mapping: Technical ≡ Governance + Memory planes. Institutional ≡ Policy Compiler, Approval Gates, Attest/Forget, Decision Log. Enterprise ≡ $\mathcal{P}$ parameters, θ_{esc}, supervisory routing.

Figure 3: MGDS reference architecture spanning all three governance tiers (Definition 3).

rics. **Drift Detector**: monitors PVR, CSCE, and PVDR using Page-Hinkley [31] and Kolmogorov-Smirnov tests. **Evaluation Pipeline**: computes the seven constructs (Section 4.4) on schedule. **Anomaly Monitor**: detects security threat patterns.

8.5 Threat Model

9 Formal Properties

Proposition 1 (Replay-Auditability and Non-Repudiation)**.** *Under (a) monotonic policy constraints, (b) append-only cryptographically chained $\mathcal{A}$, and (c) content-addressed retained memory versions: (i) every governed action is **replay-auditable** and (ii) governed actions are **non-repudiable**.*

Proof sketch. (i) Each d_t references memory version $v(t)$ and policy version $p(t)$. Since both are content-addressed and versioned, re-evaluating $\mathcal{P}_{p(t)}(\mathcal{M}_{v(t)}, r_t)$ reproduces the verdict deterministically (τ is deterministic given fixed $\mathcal{P}$, Definition 5).

(ii) To repudiate action a in d_t with $h_t = H(d_t \| h_{t-1})$, a party must produce $d_t' \neq d_t$ with equal hash (contradicting collision resistance) or replace d_t and all subsequent entries (detectable by any party holding $h_T, T \geq t$). □ □

Corollary 1 (Policy Non-Regression)**.** *Under monotonic constraints, previously denied actions remain denied under any policy superset, and cannot be retroactively authorized without a new auditable policy version.*

Connection to decision continuity. Proposition 1 establishes replay-auditability as a formal property. Definition 15 establishes decision continuity as the architectural precondition: replay-auditability requires that each d_t can reference $\mathcal{M}_{v(t)}$, which requires that memory versions be governed and preserved—i.e., that the system is decision-continuous.

10 Distinctness from Adjacent Categories

Each adjacent category addresses a component-level concern. **RAG** provides grounded generation without versioning or policy enforcement. **Vector databases** provide similarity search without governance semantics. **Feature stores** version features but do not govern decisions. **MLOps** monitors models but not memory-decision consistency. **Policy engines** enforce at request time without memory integration. **Agent frameworks** provide per-agent memory without cross-surface governance continuity.

MGDS addresses the system-level concern: memory, governance, execution, and feedback forming a closed, auditable loop that is decision-continuous across all three governance tiers.

11 Economics of Memory Governance

11.1 Cost Model

$$C(a) = c_{\text{ret}} + c_{\text{eval}} + c_{\text{exec}} + c_{\text{audit}} + c_{\text{upd}} \quad (11)$$

$$C_{\text{total}} = Q \cdot \left[\alpha \cdot C_{\text{full}} + (1-\alpha) \cdot C_{\text{fast}}\right] + C_{\text{fixed}} \quad (12)$$

where α is the fraction of actions exceeding the salience threshold requiring full evaluation. The salience-gated architecture directly reduces α, controlling governance overhead.

Table 2: Threat model: vectors and mitigations.

Threat	Description	Mitigation
Replay attack	Replayed decision events bypass nonce checks	Nonce windows, signed event timestamps, append-only ledger
Memory poisoning	Malicious writes corrupt decision state	Signed writes, anomaly detection, governance-gated ingress
Audit tampering	Modification of decision logs	Append-only with cryptographic chaining, external attestation
Cross-surface desync	Surface states diverge without detection	CSCE monitoring, eager synchronization for high-risk constraints
Policy bypass	Direct execution surface access circumvents governance	No path from Memory to Execution bypasses Governance Plane
Memory exfiltration	Unauthorized extraction of memory state	Encryption at rest/transit, RBAC, read audit logging
Connector compromise	Adapter executes unauthorized actions	Least-privilege, signed events, adapter sandboxing

11.2 Compounding Returns Hypothesis

The economic value of MGDS is hypothesized to compound through: (1) error reduction ($S_{\text{err}} = Q \cdot (p_v - p'_v) \cdot c_{\text{viol}}$); (2) decision quality improvement ΔR per action; (3) audit cost reduction from $O(N)$ manual review to $O(1)$ targeted replay; (4) consistency premium from improved cross-surface CSR. These are structural predictions requiring empirical quantification.

$$V(T) = \sum_{t=1}^{T} \frac{S_{\text{err}}(t) + Q \cdot \Delta R(t) + S_{\text{audit}}(t)}{(1+r)^t} - C_{\text{total}}(t) \tag{13}$$

Positive NPV requires non-trivial violation rates ($p_v > 5\%$) and per-violation costs exceeding marginal governance overhead—parameter ranges that are empirically testable.

Figure 4: Benchmark Suite Overview

Task Families (construct measured):
T1: Multi-turn consistency (PVDR)
T2: Cross-surface constraints (PVR, CSR, CSCE)
T3: Non-regression under updates (PVDR)
T4: Audit fidelity (AC, RF)
T5: Policy adherence, adversarial (PVR, CSR)
T6: Governance latency/cost (GL, CPGA)

Domains: Advertising, Procurement, Healthcare, Finance

Baselines: B1: RAG-only; B2: Vector DB memory; B3: Tool-only agent; B4: MLOps pipeline; B5: Rule engine

Figure 4: Benchmark suite. Task labels map to constructs in Section 4.4 and hypotheses H1–H3.

12 Evaluation Framework and Benchmark Suite

12.1 Experimental Methodology

(1) Instantiate each baseline (B1–B5) and MGDS on the same LLM and tool set. (2) Generate evaluation datasets with ground-truth annotations for each task. (3) Run each system, recording all seven constructs. (4) Statistical analysis: Mann-Whitney U tests, bootstrap confidence intervals, Bonferroni correction for multiple comparisons. (5) Ablation: remove individual MGDS planes to isolate contributions. (6) Pre-register the protocol on OSF prior to data collection. (7) Report all results including null and negative findings.

13 Case Study: Omnichannel Advertising Governance

We develop an illustrative case study applying MGDS to omnichannel advertising governance for SMB merchants. **The advertising domain is illustrative, not foundational**: the framework generalizes to autonomous agents, multi-API orchestration, distributed robotics, and enterprise workflow automation. The case study is conceptual and hypothetical; it demonstrates structural applicability and provides a template for empirical measurement. No deployment data is reported.

Table 3: Distinction of MGDS from adjacent categories. ● = native, ○ = partial/add-on, – = absent.

Capability	RAG	Vec. DB	Feat. Store	MLOps	Policy Eng.	Agent Fwk	MGDS
Versioned memory	–	–	●	○	–	–	●
Policy-bound retrieval	–	–	–	–	–	–	●
Decision audit log	–	–	–	○	○	–	●
Cross-surface consistency (CSCE)	–	–	–	–	○	–	●
Governance-gated actions	–	–	–	–	●	○	●
Memory lifecycle mgmt	–	○	○	–	–	○	●
Replay fidelity (RF)	–	–	–	–	○	–	●
Decision continuity	–	–	–	–	–	–	●
Three-tier governance	–	–	–	–	○	–	●
Measurable constructs (H1–H3)	–	–	–	○	–	–	●

Table 4: Benchmark tasks, target constructs, metrics, and datasets.

Task	Description	Constructs	Datasets
T1: Multi-turn consistency	Consistent decisions over k turns with evolving context	PVDR, PVR	Synthetic: 1,000 dialogs. Real: support ticket sequences
T2: Cross-surface constraints	Global constraints across n surfaces simultaneously	PVR, CSR, CSCE	Synthetic: multi-platform ad campaigns. Real: anonymized marketing data
T3: Non-regression	Correct decisions maintained after system updates	PVDR, regression rate	Synthetic: policy update injection. Real: model update logs
T4: Audit fidelity	All decisions replayable with complete audit trail	AC, RF	Synthetic: replay 10K decisions, verify reconstruction
T5: Adversarial policy	Governance maintained under bypass attempts	PVR, CSR, false negative rate	Adversarial prompt injection, policy-bypass attacks
T6: Latency/cost	Governance overhead bounded and predictable	GL, CPGA	Load testing at varying volumes and salience distributions

13.1 Problem Setting

A Shopify merchant operates across Google Ads, Meta Ads, TikTok Ads, and email marketing with a global daily budget of $200, brand standards, and claim substantiation requirements. Each platform imposes distinct constraints. Without MGDS, governance-structure failures emerge: budget over-allocation (CSCE elevation), unsubstantiated health claims (PVR increase), non-reproducible bid changes (RF collapse), and attribution fragmentation (AC degradation).

13.2 MGDS Application

Memory Plane: $\mathcal{M}_s$ stores certifications, brand guidelines, and policy summaries; $\mathcal{M}_e$ stores interaction history; $\mathcal{M}_d$ records every governed action, fully versioned.

Governance Plane (all three tiers): *Enterprise*—budget cap $200/day, $\rho = (0, 0.15, 0.08)$. *Institutional*—policy versions with authorship metadata, Approval Gates logging reviewer identity and rationale. *Technical*—health claim verification against $\mathcal{M}_s$, platform-specific constraints at dispatch, rate limits.

Feedback Plane: Attribution Engine detects cross-surface inconsistencies; Drift Detector monitors PVDR and CSCE; elevated PVDR triggers salience sensitivity increase.

Continuity: The case study requires decision continuity (Definition 15): budget allocation at one surface must be visible to governance at all others, requiring shared governed memory rather than per-adapter local state.

Table 5: Hypothetical construct values for the SMB merchant case. Illustrative only.

Metric	Period	No MGDS	With MGDS
PVR	30 days	0.18	0.03
CSR	30 days	0.82	0.97
CSCE	30 days	0.42	0.06
AC	1K decisions	0.23	0.99
RF	1K decisions	0.19	0.97
GL (ms)	30 days	n/a	12 (p50) / 38 (p99)
PVDR	Post-update	+0.12/day	+0.01/day

13.3 Hypothetical Structured Metrics

Table 5 presents the construct values that would be collected in a real deployment. Values are hypothetical and illustrate the measurement framework.

13.4 The Joseph Grain Principle

We note a structural parallel in the Genesis account (Gen. 41): Joseph implements what is architecturally a memory-governed resource allocation system—observed signals are stored, a policy is derived, and the policy governs distribution across a multi-region execution surface. The insight is architectural: governance that is *constitutive* of distribution, not *advisory* to it, with memory of past conditions determining present action.

14 Discussion

14.1 Governance-Structure Failures vs. Capability Failures

The reframing from "memory failures" to *governance-structure failures* is precise: it does not claim that memory is categorically deficient, but that the governance structures binding policy, authorization, and audit to memory state are absent. This distinction matters empirically—it identifies the *structural* property whose presence or absence determines whether failures occur, independent of model capability.

14.2 The Governance-as-Routing Principle

In MGDS, governance is the *routing layer* for intelligence: every action passes through the Governance Plane, which determines which decisions become actions. This is analogous to network routers enforcing routing policies. Crucially, governance routing operates across all three tiers simultaneously: enterprise intent sets the routing policy, institutional accountability records the routing decision, and technical controls enforce it at execution time.

14.3 Implications for Autonomy

The Autonomy and Continuity section (Section 7) establishes a structural consequence: autonomous systems that are not decision-continuous are epistemically isolated across decision boundaries. Their actions are ungoverned not because governance was not attempted, but because the memory substrate required for governance to be effective does not exist. MGDS addresses this by making decision continuity an architectural invariant.

14.4 Normative vs. Empirical Claims

Normative claims—that memory should be a first-class governed primitive; that governance should span three tiers; that the integration chain should be closed—are design positions grounded in the governance literature. They do not depend on the truth of H1–H3.

Empirical claims (H1–H3) are hypotheses requiring benchmark execution. The paper's architectural contribution stands independently of whether H1–H3 are confirmed; what changes upon refutation is the relative priority of MGDS as the primary intervention, not its validity as an architecture.

14.5 Relationship to Compute Scaling

MGDS is complementary to compute scaling. Larger models may serve as more capable decision engines within the Execution Plane, but the Memory, Governance, and Feedback Planes provide the substrate converting capability into governed, consistent, auditable behavior. This is a normative claim; whether it holds empirically depends on the magnitude of capability-driven vs. governance-driven improvement, which H1 addresses.

15 Limitations

No empirical validation. The benchmark suite is proposed but not executed. H1–H3 and all seven constructs remain unmeasured in real deployments.

Governance overhead. Full policy evaluation at high throughput may introduce unacceptable GL. The salience threshold mitigates this but requires tuning and empirical characterization.

Policy specification complexity. Incorrect policies may create false governance. The Policy Linter reduces but does not eliminate this risk.

Distributed consistency. CAP theorem [7] constraints apply; MGDS provides detection and reconciliation but does not resolve the fundamental impossibility.

Neuromorphic analogy limits. The mapping is structural, not physical. Software MGDS does not achieve neuromorphic energy efficiency or hardware-level memory-compute co-location.

Single-investigator design with commercial affiliation. See Research Integrity Statement (Section 17).

Tier boundary ambiguity. Enterprise objectives and institutional compliance requirements may conflict. Formal inter-tier conflict resolution is deferred to future work.

15.1 Alternative Interpretations

Compute-centric: Larger, better fine-tuned models may partially resolve context loss and policy drift without architectural changes. H1–H3 are testable under conditions that hold model capability constant.

Tooling recombination: Existing tools (OPA, event-sourced databases, vector stores) may be combinable to achieve equivalent benchmark scores without a new paradigm. The distinctness argument (Section 10) addresses this; it remains challengeable empirically.

Organizational-process: Governance failures may be primarily human and processual—inadequate review cycles, unclear accountability—rather than architectural. MGDS formalizes organizational processes as architectural constraints, but the causal priority is an open empirical question.

16 Future Work

Benchmark execution (highest priority): pre-register, execute, and publish benchmark results including null and negative findings.

Neuromorphic hardware deployment: implement the governance loop (Section 6.3) on Loihi 2 [12] or SpiNNaker 2 [17] and characterize GL improvements.

Standard interfaces: extend the Model Context Protocol [3] with governance primitives compatible with TOGAF [47] and COBIT [21].

Formal verification: SMT-based verification [13] of policy specifications prior to deployment.

Inter-tier conflict resolution: formal frameworks for resolving enterprise-institutional policy conflicts, drawing on social choice theory [4].

Memory compression and forgetting: rate-distortion [9] and differential privacy [14] formalizations of the Forget operation.

Multi-investigator replication: independent research groups instantiating the framework, executing the benchmark, and publishing results—including contradictory findings.

Enterprise governance parameterization: formalizing ρ as a typed policy object aligned with COBIT governance objectives [21].

17 Research Integrity Statement

17.1 Conflict of Interest

The author is affiliated with GENYS AI Inc., a company with commercial interest in products related to the MGDS architecture described in this paper. This affiliation constitutes a potential conflict of interest. The theoretical framework and formal definitions are intended to be evaluable independently of any commercial implementation. The benchmark suite is designed to be executable by independent researchers without access to any proprietary system. No proprietary implementation data informs any claim in this paper.

17.2 Separation of Theoretical and Commercial Contributions

The MGDS paradigm is a theoretical and architectural contribution. Any commercial product instantiating aspects of this architecture should be evaluated on its own merits, not as validation of the theoretical claims. Theoretical claims should be evaluated on formal rigor, benchmark results when available, and peer review—not on commercial adoption.

17.3 Data Transparency Plan

This paper reports no primary empirical data. Upon executing the benchmark study, the authors commit to: (1) pre-registration on OSF prior to data collection; (2) public release of datasets, code, and anonymization procedures under an open license; (3) reporting all results including null and negative findings; (4) making the complete dataset available for independent replication.

17.4 Collaboration Framework

Independent researchers are invited to propose modifications to the formal definitions, implement alternative instantiations, execute the benchmark suite, and report results—including contradictory findings. Academic access to any commercial MGDS-adjacent system for research purposes will be facilitated under standard data sharing agreements.

17.5 Limitations of This Statement

A research integrity statement authored by the investigator with the potential conflict of interest provides limited independent assurance. External peer review, independent benchmark execution, and multi-investigator replication are the structural safeguards this statement supplements but does not replace.

18 Conclusion

Governance presupposes persistent, inspectable, authorized state. When AI systems treat memory as auxiliary infrastructure rather than as the substrate of governance itself, governance-structure failures emerge: context loss, cross-surface inconsistency, policy drift, non-reproducibility, and incomplete audit trails. These are not primarily model-capability failures; they are architectural failures in the binding of policy, authorization, and audit semantics to system memory state and its transitions.

We introduced Memory-Governed Decision Systems (MGDS) as a formal paradigm in which memory is versioned, policy-bound, auditable, and constitutive of downstream action authorization. We established that governance presupposes persistent inspectable state; formalized MGDS as a transition system $(\mathcal{M}, \mathcal{P}, \mathcal{D}, \mathcal{A})$; grounded governance across technical, institutional, and enterprise tiers with an explicit integration principle; defined seven measurable constructs; established decision continuity as a formal property of governed memory transition; presented a four-level taxonomy, reference architecture, structural neuromorphic analogy, and distinctness argument; specified a benchmark protocol with three falsifiable hypotheses; and provided an illustrative case study with hypothetical structured metrics.

The paradigm makes falsifiable predictions. If systems with first-class memory governance do not measurably outperform systems with equivalent model capability but ad-hoc memory on H1–H3 under controlled conditions, the paradigm requires revision. We advance this claim knowing the benchmark has not been executed and that the alternative interpretations presented in Section 15.1 may prove more parsimonious. The value of this contribution, at this stage, is the precision of the architectural argument, the operationalizability of the governance constructs, and the specification of the experimental conditions under which the central claims can be confirmed or refuted.

Acknowledgments

[Redacted for review.]

References

[1] Amazon Web Services. Cedar: A language for defining permissions as policies. `https://www.cedarpolicy.com/`, 2023.

[2] Dario Amodei et al. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.

[3] Anthropic. Model context protocol. `https://modelcontextprotocol.io/`, 2024.

[4] Kenneth J Arrow. A difficulty in the concept of social welfare. *J. Political Economy*, 58(4):328–346, 1950.

[5] Alan Baddeley. Working memory. *Science*, 255(5044):556–559, 1992.

[6] Guo-qiang Bi and Mu-ming Poo. Synaptic modifications in cultured hippocampal neurons. *J. Neuroscience*, 18(24):10464–10472, 1998.

[7] Eric A Brewer. Towards robust distributed systems. In *ACM PODC*, 2000.

[8] Harrison Chase. LangChain. https://github.com/langchain-ai/langchain, 2023.

[9] Thomas M Cover and Joy A Thomas. *Elements of Information Theory*. Wiley, 2nd edition, 2006.

[10] Richard M Cyert and James G March. *A Behavioral Theory of the Firm*. Prentice-Hall, 1963.

[11] DAMA International. *DAMA-DMBOK: Data Management Body of Knowledge*. Technics Publications, 2nd edition, 2017.

[12] Mike Davies et al. Loihi: A neuromorphic manycore processor. *IEEE Micro*, 38(1):82–99, 2018.

[13] Leonardo de Moura and Nikolaj Bjørner. Z3: An efficient SMT solver. In *TACAS*, pages 337–340, 2008.

[14] Cynthia Dwork et al. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284, 2006.

[15] European Parliament and Council. Regulation (EU) 2024/1689 (AI Act). Official Journal of the EU, 2024.

[16] Martin Fowler. Event sourcing. https://martinfowler.com/eaaDev/EventSourcing.html, 2005.

[17] Steve B Furber et al. The SpiNNaker project. *Proc. IEEE*, 102(5):652–665, 2014.

[18] Seth Gilbert and Nancy Lynch. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *ACM SIGACT News*, 33(2):51–59, 2002.

[19] Alan R Hevner, Salvatore T March, Jinsoo Park, and Sudha Ram. Design science in information systems research. *MIS Quarterly*, 28(1):75–105, 2004.

[20] Giacomo Indiveri et al. Neuromorphic silicon neuron circuits. *Frontiers in Neuroscience*, 5:73, 2011.

[21] ISACA. COBIT 2019 framework: Introduction and methodology. https://www.isaca.org/resources/cobit, 2019.

[22] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Trans. Big Data*, 7(3):535–547, 2019.

[23] Leslie Lamport. Time, clocks, and the ordering of events in a distributed system. *Commun. ACM*, 21(7):558–565, 1978.

[24] Patrick Lewis et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. *NeurIPS*, 33, 2020.

[25] Wolfgang Maass. Networks of spiking neurons. *Neural Networks*, 10(9):1659–1671, 1997.

[26] Yu A Malkov and D A Yashunin. Efficient and robust approximate nearest neighbor search using HNSW. *IEEE TPAMI*, 42(4):824–836, 2018.

[27] James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex. *Psychological Review*, 102:419–457, 1995.

[28] Carver Mead. Neuromorphic electronic systems. *Proc. IEEE*, 78(10):1629–1636, 1990.

[29] Paul A Merolla et al. A million spiking-neuron integrated circuit. *Science*, 345(6197):668–673, 2014.

[30] NIST. Artificial intelligence risk management framework (AI RMF 1.0). Technical Report NIST AI 100-1, U.S. Dept. of Commerce, 2023.

[31] E S Page. Continuous inspection schemes. *Biometrika*, 41:100–115, 1954.

[32] Joon Sung Park et al. Generative agents: Interactive simulacra of human behavior. In *ACM UIST*, pages 1–22, 2023.

[33] Geoffrey G Parker, Marshall W Van Alstyne, and Sangeet Paul Choudary. *Platform Revolution*. W. W. Norton & Company, 2016.

[34] Toran Bruce Richards. AutoGPT. https://github.com/Significant-Gravitas/AutoGPT, 2023.

[35] Jean-Charles Rochet and Jean Tirole. Platform competition in two-sided markets. *Journal of the European Economic Association*, 1(4):990–1029, 2003.

[36] Stuart Russell. *Human Compatible*. Viking, 2019.

[37] Johannes Schemmel et al. A wafer-scale neuromorphic hardware system. In *IEEE ISCAS*, pages 1947–1950, 2010.

[38] Timo Schick et al. Toolformer: Language models can teach themselves to use tools. *NeurIPS*, 36, 2023.

[39] Catherine D Schuman et al. Opportunities for neuromorphic computing. *Nature Computational Science*, 2(1):10–19, 2022.

[40] Shreya Shankar et al. Operationalizing machine learning: An interview study. *arXiv preprint arXiv:2209.09125*, 2022.

[41] Marc Shapiro et al. Conflict-free replicated data types. In *SSS*, pages 386–400, 2011.

[42] Herbert A Simon. *Administrative Behavior*. Macmillan, 1947.

[43] Larry R Squire. Memory and the hippocampus. *Psychological Review*, 99(2):195–231, 1992.

[44] Larry R Squire. The memory systems of the brain. *Neurobiology of Learning and Memory*, 82(3):171–177, 2004.

[45] Styra, Inc. Open policy agent. `https://www.openpolicyagent.org/`, 2023.

[46] Tecton. What is a feature store? `https://www.tecton.ai/blog/what-is-a-feature-store/`, 2021.

[47] The Open Group. *TOGAF Standard, Version 9.2*. The Open Group, 2018.

[48] Endel Tulving. Episodic and semantic memory. In *Organization of Memory*, pages 381–403. Academic Press, 1972.

[49] Paul Voigt and Axel von dem Bussche. *The EU General Data Protection Regulation (GDPR)*. Springer, 2017.

[50] Shunyu Yao et al. ReAct: Synergizing reasoning and acting. *ICLR*, 2023.