PROVISIONAL PATENT APPLICATION

Title: Neural-Backed Memory Fabric with Enterprise Digital DNA (NBMF-eDNA)

Inventor: Masoud Masoori — Richmond Hill, Ontario, Canada

Brief Description of the Drawings

- FIG. 1 illustrates a three-tier NBMF memory governed by eDNA with promotion/eviction routing.
- FIG. 2 shows tier thresholds and validation for safe promotions.
- FIG. 3 depicts L2 quarantine with consensus/divergence scoring and outcomes.
- FIG. 4 shows dual-mode encoding converging into NBMF bytecode.
- FIG. 5 depicts Merkle-notarized lineage for auditable history.
- FIG. 6 illustrates Genome, Epigenome, Lineage, and Immune components.
- FIG. 7 shows detect–quarantine–rollback defense loop.
- FIG. 8 shows dynamic CPU/GPU/TPU routing via a tensor router.
- FIG. 9 shows cross-tenant isolation with sanitized artifacts.

Detailed Description

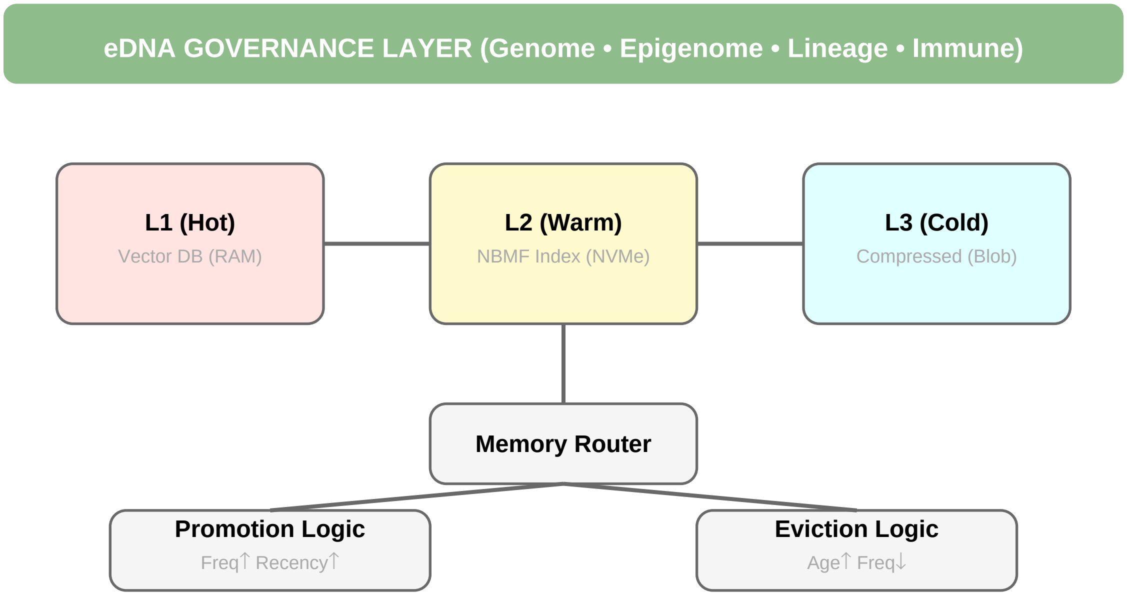

FIG. 1 — NBMF System Overview.

The eDNA banner governs a three-tier memory fabric. A policy-aware router mediates movement of memories between tiers while promotion and eviction logic apply frequency, recency, and age signals.

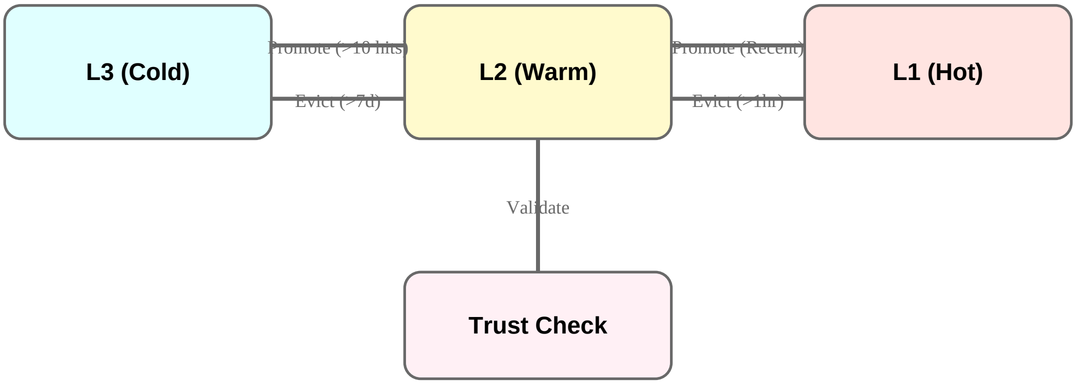

FIG. 2 — Promotion & Eviction flow with thresholds and validation.

Memories move upward when they are recent or popular and move downward when stale. A trust gate validates promotions to prevent corruption of warm and hot tiers.

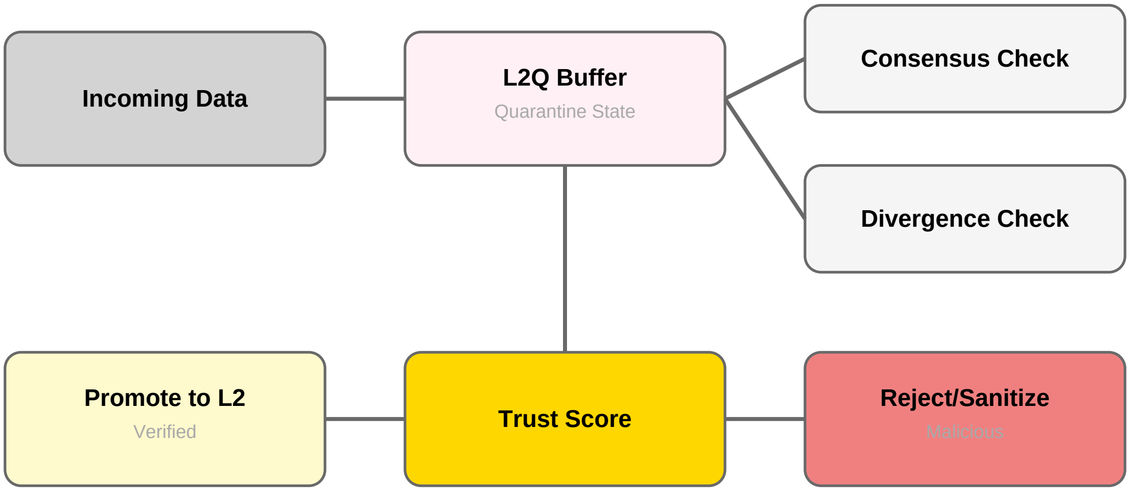

FIG. 3 — Trust & Quarantine pipeline including consensus/divergence checks.

Before persistence, new items pass through an L2 quarantine buffer and are scored by consensus and divergence. Only sufficiently trusted items are promoted; others are sanitized or rejected.

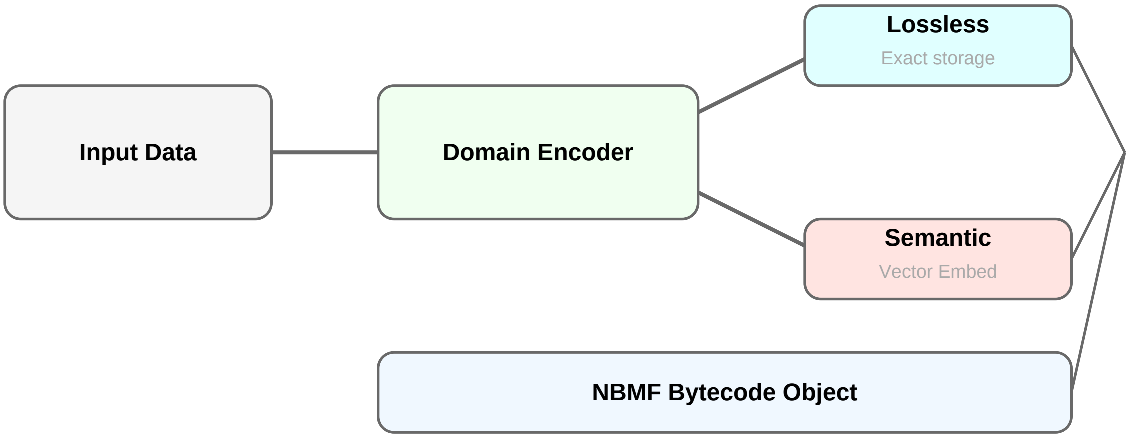

FIG. 4 — Neural encoding into lossless vs semantic NBMF bytecode.

Input is encoded via a domain encoder. Lossless mode preserves exact bytes; semantic mode stores vectorized meaning. Both converge into an NBMF bytecode object with metadata.

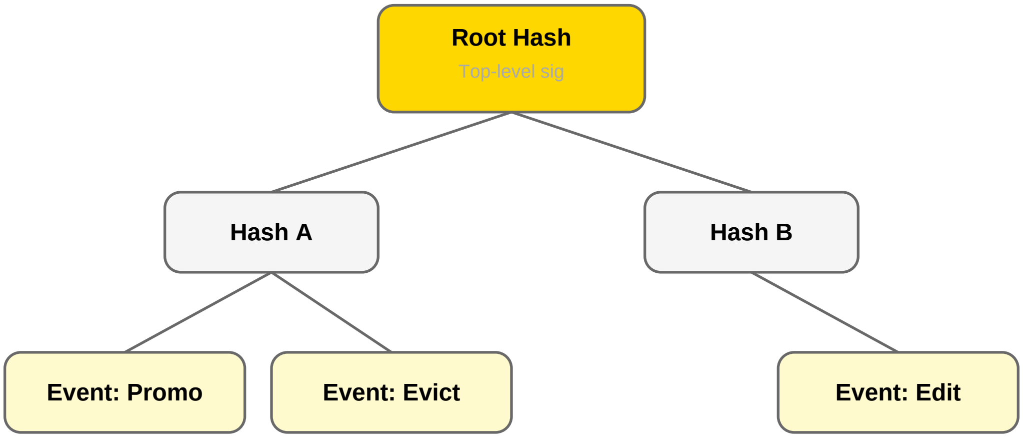

FIG. 5 — Merkle-notarized lineage tree for audit proofs.

All promotions and edits append events to a Merkle tree that yields a verifiable root for audits and rollback.

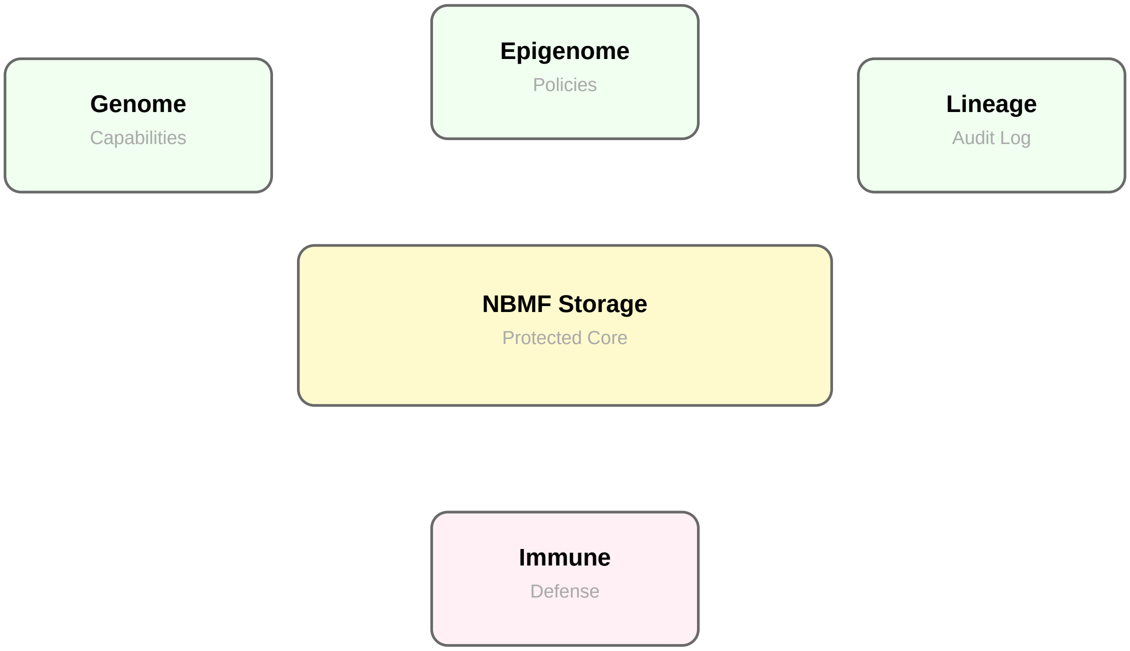

FIG. 6 — eDNA components: Genome, Epigenome, Lineage, and Immune modules.

The eDNA layer exposes capability schemas (Genome), policy (Epigenome), notarized history (Lineage), and a defensive Immune module that intervenes on risk.

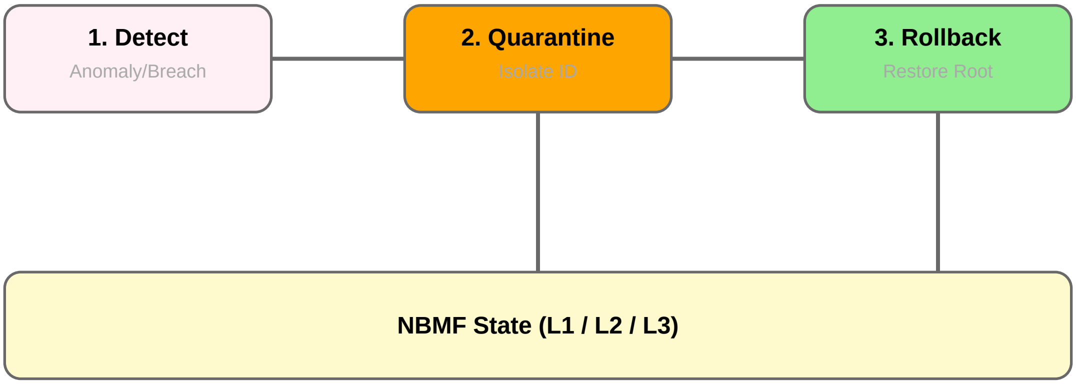

FIG. 7 — Immune workflow: detect, quarantine, and rollback.

Detections trigger quarantine; if necessary the system rolls back state to a known good Merkle root to maintain integrity.

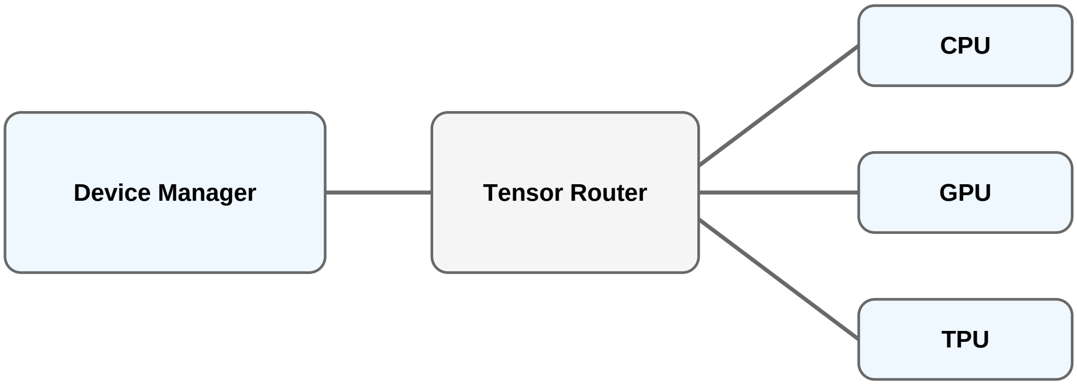

FIG. 8 — Hardware abstraction and tensor routing across CPU/GPU/TPU.

A device manager and tensor router dynamically choose CPU, GPU, or TPU targets to balance cost and latency for encoding and retrieval operations.

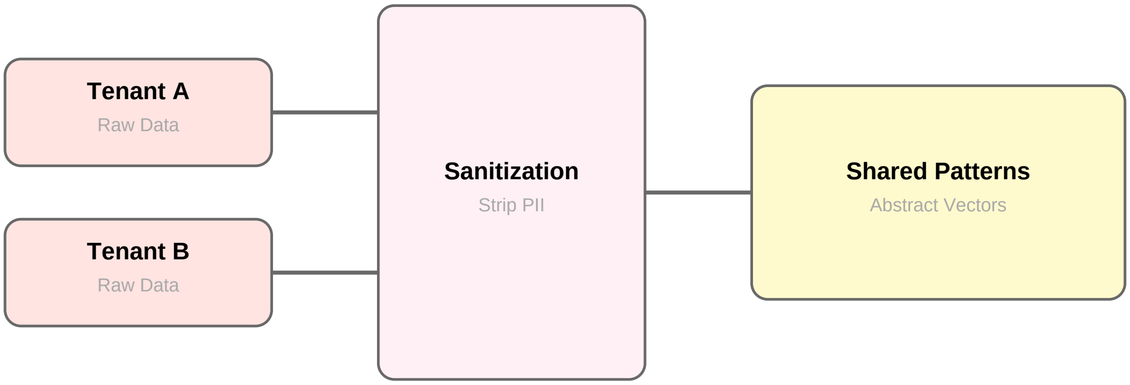

FIG. 9 — Cross-tenant isolation with sanitized, shareable artifacts.

Tenant data never leaves its boundary; only abstracted artifacts—patterns and vectors—are shared to enable cross-tenant learning without raw data leakage.

Claims

1. Three-Tier Hierarchical Memory System with Automatic Promotion/Eviction — A memory storage system for multi-agent artificial intelligence systems comprising: a) a first tier (L1) configured to store vector embeddings for fast similarity search with latency less than 25 milliseconds at the 95th percentile; b) a second tier (L2) configured to store Neural-Backed Memory Format (NBMF) encoded records with full metadata, encrypted with AES-256, with latency less than 120 milliseconds at the 95th percentile; c) a third tier (L3) configured to store compressed archives for long-term storage with on-demand decompression; d) a memory router configured to automatically route memories between tiers based on access frequency, age, and trust scores; e) a promotion controller configured to promote memories from lower tiers to higher tiers when access frequency exceeds a threshold and age is below a maximum age; f) an eviction controller configured to demote memories from higher tiers to lower tiers when age exceeds a threshold or access frequency falls below a minimum threshold.

2. Neural Bytecode Memory Format (NBMF) Encoding System — A method for encoding data into a neural bytecode format comprising: a) receiving input data from an AI agent; b) encoding the input data using a domain-specific neural encoder to produce a latent vector representation; c) compressing the latent vector representation to produce compressed bytecode; d) storing the bytecode with metadata including emotion tags, trust scores, and provenance information; e) wherein the encoding operates in either lossless mode for 100% accuracy or semantic mode for meaning preservation with 95%+ similarity.

3. Content-Addressable Storage with SimHash Deduplication — A method for deduplicating memories in a multi-agent AI system comprising: a) computing a SHA-256 hash of memory content for exact duplicate detection; b) storing memories in a content-addressable storage (CAS) system using the hash as a key; c) computing a SimHash value for near-duplicate detection by extracting features, hashing each feature, and creating a bit vector; d) detecting near-duplicates by comparing SimHash values within a similarity threshold; e) reusing existing encodings for near-duplicates to reduce storage and processing costs.

4. Trust Pipeline with Quarantine System — A method for validating memories before permanent storage comprising: a) storing new memories in a quarantine store (L2Q) before promotion to permanent storage; b) validating memories using multi-model consensus by querying multiple language models and comparing responses; c) calculating divergence scores against existing memories using vector similarity search; d) computing a trust score based on consensus score and divergence score; e) promoting validated memories to permanent storage (L2) only if the trust score exceeds a threshold and divergence score is below a threshold.

5. Enterprise-DNA Governance Layer — A governance system for multi-agent AI systems comprising: a) a Genome component storing capability schemas and versioned behaviors per agent and department; b) an Epigenome component storing tenant policies including Attribute-Based Access Control (ABAC) rules, retention policies, jurisdiction constraints, and service level objectives; c) a Lineage component storing Merkle-notarized promotion history with cryptographic proofs linking to memory ledger transactions; d) an Immune component detecting threats including anomalies, policy breaches, and prompt injection attempts, and triggering protective actions including quarantine, quorum requirements, and rollback.

6. Merkle-Notarized Lineage Chain — A method for creating cryptographic audit trails for memory operations comprising: a) recording a lineage entry for each memory promotion between tiers, including object identifier, source tier, destination tier, and timestamp; b) computing a Merkle root by combining parent lineage hash with current promotion data and computing a cryptographic hash; c) linking lineage entries to NBMF ledger transaction identifiers; d) constructing a lineage chain by linking parent and child lineage entries; e) enabling verification of lineage chain integrity using Merkle proofs and cryptographic hashes.

7. Hardware Abstraction for Multi-Device Tensor Operations — A hardware abstraction system for neural memory operations comprising: a) a DeviceManager detecting available compute devices including CPU, GPU, and TPU; b) a device selection logic choosing optimal device based on operation type, configuration, and availability; c) a tensor operation router routing NBMF encoding and decoding operations to selected device; d) batch size optimization for TPU operations using a configurable batch factor; e) automatic framework detection

and device capability assessment.

8. Cross-Tenant Learning via Abstract Artifacts — A method for safe cross-tenant knowledge sharing comprising: a) generating abstracted NBMF artifacts from tenant memories by removing raw tenant data and preserving semantic patterns; b) sanitizing artifacts to remove tenant-specific identifiers and sensitive information; c) extracting reusable patterns from abstracted artifacts; d) sharing abstracted artifacts across tenants without raw data leakage; e) maintaining tenant isolation for raw data while enabling pattern learning from abstracted artifacts. --- ## Dependent Claims ### Claim 9 (depends on Claim 1) The memory storage system of Claim 1, wherein the promotion controller promotes memories from L3 to L2 when access frequency exceeds 10 accesses and from L2 to L1 when recency is less than 1 hour. ### Claim 10 (depends on Claim 1) The memory storage system of Claim 1, wherein the eviction controller demotes memories from L1 to L2 when age exceeds 1 hour and from L2 to L3 when age exceeds 7 days. ### Claim 11 (depends on Claim 2) The method of Claim 2, wherein the domain-specific neural encoder is selected from a group consisting of conversation encoder, financial encoder, legal encoder, and general encoder. ### Claim 12 (depends on Claim 2) The method of Claim 2, wherein lossless mode achieves 13.30× compression ratio and semantic mode achieves 2.53× compression ratio. ### Claim 13 (depends on Claim 3) The method of Claim 3, wherein SimHash is computed as a 64-bit or 128-bit value, and near-duplicates are detected when SimHash distance is below a threshold. ### Claim 14 (depends on Claim 4) The method of Claim 4, wherein the trust score is computed as: Trust Score = (Consensus Score × 0.6) + ((1 - Divergence Score) × 0.4). ### Claim 15 (depends on Claim 4) The method of Claim 4, wherein memories are promoted to L2 if trust score >= 0.7 and divergence score < 0.5, flagged for human review if trust score >= 0.5 and divergence score < 0.7, and rejected otherwise. ### Claim 16 (depends on Claim 5) The governance system of Claim 5, wherein the Epigenome component enforces retention policies causing automatic archival to L3 cold storage based on data age and classification. ### Claim 17 (depends on Claim 5) The governance system of Claim 5, wherein the Immune component triggers automatic rollback of recent memory promotions when critical threats are detected. ### Claim 18 (depends on Claim 6) The method of Claim 6, wherein the Merkle root is computed as SHA-256(merkle_parent || promotion_data) where promotion_data includes object_id, promotion_from, promotion_to, and transaction identifier. ### Claim 19 (depends on Claim 7) The hardware abstraction system of Claim 7, wherein TPU operations use a batch factor of 128 for optimal performance. ### Claim 20 (depends on Claim 8) The method of Claim 8, wherein abstracted artifacts include NBMF bytecode, semantic patterns, and structural information, but exclude raw text, PII, and tenant-specific identifiers. --- **End of Claims**