Governed Enterprise AI Memory Beyond RAG: From Vector Retrieval to Permissioned Knowledge Graphs

Zhirong (Mark) Huang
ORCID: 0009-0007-4326-2042

2026-07-17

Abstract

Enterprises are adopting multiple AI tools at the same time: chat assistants, coding agents, office copilots, search systems, and workflow automations. These systems can produce useful work when the necessary background information is available in the prompt, context window, connector, or retrieval layer. The failure mode is that organizational background knowledge is often missing, stale, fragmented across teams, or inaccessible to the AI client performing the task. In that case, the model may generate plausible output from pretrained knowledge and local prompt clues rather than from the current state of the organization.

Retrieval-augmented generation (RAG) is the common response to this problem and has become a central technique for grounding large language models in external knowledge [2]. RAG helps, but vector retrieval alone does not inherently define which fact is current, which source is authoritative, which older decision has been superseded, or which actor is allowed to read or mutate knowledge. This article proposes governed enterprise AI memory as a layer above ordinary retrieval: a permissioned, provenance-preserving knowledge graph that can connect multiple AI clients while maintaining team isolation, read/write scopes, evidence lineage, conflict handling, and fact supersession.

The paper uses Dense-Mem as an example implementation and evaluation test bench; the architecture does not depend on it. The package includes the initial 100-scenario synthetic comparison and two public-corpus retrieval studies. The first study covers 795 cases across three axes. Relationship-heavy retrieval slightly improved top-k recall, but it substantially reduced early-rank quality and judged relevance and increased latency. An evidence-ID approach recovered much of the loss but still trailed the chunk-first control. The second study uses a 1,000-document, 206-case seed across six task families. It evaluates an evidence-first read path in which exact, vector, and text branches rank evidence while relationships feed a separate discovery path. On the matched seed, the evidence-first candidate improved Recall@10 from 0.843 to 0.886, MRR from 0.807 to 0.856, nDCG@10 from 0.787 to 0.846, and the rank-1 rate from 0.762 to 0.830 over the retained chunk-first control. That result still comes from one smoke run: the current judge cannot replay the control, and the two runs used different transports for latency. In both studies, direct evidence was the better initial RAG retrieval unit. The design keeps relationships for governance, trace, and discovery, but neither study tests whether following a relationship path improves a later answer.

1. Introduction

Large language models are useful because they can condition generation on a prompt and its surrounding context. The Transformer architecture introduced a sequence-transduction model built around self-attention that underlies many modern LLM systems [1], and retrieval-augmented generation later showed a practical way to combine parametric model behavior with external documents [2]. The enterprise problem is not only whether a model is capable of reasoning over text. The problem is whether the correct organizational background information is available, current, permitted, and traceable when the model is asked to answer.

Modern companies rarely use a single AI surface. A developer may use Claude Code or another coding agent inside a repository. A product manager may use ChatGPT or a workplace copilot to summarize customer requests. A support team may use an AI workflow that reads tickets. A business stakeholder may use a chat assistant to draft a roadmap update. Each surface can be valuable, but each surface often has a partial view of the organization.

The result is a fragmented-context failure mode:

- Business context exists but is not visible to the developer's coding agent.
- Engineering constraints exist but are not visible to a business user's AI assistant.
- A prior decision has been superseded, but an older document remains easier to retrieve.
- A model can retrieve semantically similar text, but not determine whether that text is current, authoritative, or allowed for the current actor.
- Automation can act with confidence while using incomplete or stale background information.

Industry reports indicate that organizational AI adoption is widespread, while scaled impact remains harder than adoption alone. McKinsey's 2025 global survey reported broad AI use, early agentic AI experimentation, and a gap between deployment and enterprise-level financial impact [16]. Stanford HAI's 2026 AI Index reported that organizational AI adoption continued to rise in 2025 [17]. These adoption signals make the context problem more important: the more AI tools an enterprise deploys, the more important shared, governed organizational memory becomes.

1.1 Evidence Map And Claim Boundary

The evidence base is uneven, so the article separates motivation, prior art, and implementation evidence instead of treating them as one proof.

Article claim	Current support	Strength	Boundary
Enterprise AI adoption is broad, while scaled impact remains uneven.	Stanford reports 88% organizational AI adoption in 2025, and McKinsey reports 23% of respondents scaling agentic AI systems and 39% reporting enterprise-level EBIT impact [17, 16].	Public survey evidence.	These surveys motivate the problem; they do not prove that governed memory is the only solution.
Larger context windows do not remove the need for explicit memory governance.	Lost-in-the-Middle, RULER, and Context Rot report reliability limits for long-context model use [3, 5, 15].	Academic and technical evaluation evidence.	These studies support caution about long context; they do not show that every model fails on every long-context task.
Enterprise RAG evaluation is moving toward company-internal workflows.	EKRAG, WixQA, and EnterpriseRAG-Bench evaluate enterprise or support-style retrieval and question answering [8, 9, 14].	Benchmark and dataset evidence.	These benchmarks cover retrieval and answering more than write governance, supersession, team isolation, and multi-client memory continuity.
Graph-backed or temporal memory is active prior art.	GraphRAG, knowledge graph-guided RAG, Zep, Governed Memory, Context Objects, and Stateless Decision Memory all overlap with this design space [4, 7, 10, 11, 12, 13].	Strong prior-art evidence.	This narrows the novelty claim. The article's focus is the enterprise context-loss framing and the benchmarkable governance cases.
Dense-Mem can be used as one example and test bench for governed shared memory.	The package includes a deterministic graph baseline, vector and RAG answer-generation baselines, a Dense-Mem Docker scope-enforcement smoke test, a Dense-Mem fragment-recall answer-generation run, and a 100-scenario RAG-versus-knowledge-graph comparison.	Preliminary local evidence.	The 100-scenario comparison uses curated structured facts, not automatic fact extraction. It is not yet a full typed-fact Dense-Mem governance benchmark because the Dense-Mem recall run stores source fragments, not promoted enterprise facts with first-class supersession.
More granular relationship retrieval does not automatically improve RAG quality.	A retained 1,000-document, 795-case evaluation compares a chunk-first control, a relationship-heavy approach, and an evidence-ID approach on SciFact, HotpotQA, and MS MARCO cases [20, 21, 22, 23].	Direct retrieval and AI-judge evidence.	The qrels primarily reward direct source-document retrieval. They do not measure permissions, supersession, conflict resolution, or the full value of relationship lineage.

Continued on next page

Article claim	Current support	Strength	Boundary
Evidence-first ranking with relationships reserved for discovery can outperform chunk-first retrieval on a broader task mix.	A matched 1,000-document, 206-case comparison covers SciFact, MS MARCO, HotpotQA, MuSiQue, QASPER, and LongMemEval-derived oracle cases [21, 23, 22, 24, 25, 26].	Direct retrieval evidence from one validated six-axis smoke comparison.	The case allocation is uneven; QASPER Recall@10 declined, the control has no comparable judge replay, and the latency transports differ.

The article argues that governed shared memory warrants evaluation as an architectural layer. These runs do not establish production superiority over RAG systems in general. They show where graph-oriented retrieval imposed costs and where an evidence-first redesign improved retrieval without removing relationships from governed discovery.

2. Problem Statement

This article focuses on a specific problem:

> Enterprises need a governed memory layer because LLM-powered tools answer from incomplete context when organizational background knowledge is missing, stale, isolated, or inaccessible.

The problem is not that RAG is useless. RAG is useful. The problem is that many enterprise failures happen outside the narrow question "did retrieval return a semantically similar chunk?"

2.1 Example: Business Context Missing From Developer AI

A product team decides that a billing feature is delayed until Q4 because a partner integration is not contractually approved. The decision is recorded in a planning chat and a meeting summary. A developer later asks a coding agent which billing work should be prioritized. If the coding agent only sees repository files and tickets, it may suggest implementing work that the business already deprioritized.

A memory layer should let the developer's AI client read the approved business decision without giving it permission to mutate business records.

2.2 Example: Engineering Constraints Missing From Business AI

An engineering team records that a customer-facing endpoint cannot be exposed until authorization rules are redesigned. A business user later asks an AI assistant whether the company can promise that endpoint to customers. If the assistant only sees sales notes and public documentation, it may draft a commitment that engineering cannot safely deliver.

A memory layer should let a business AI client read the current technical constraint, cite the supporting engineering source, and indicate that older commitments are superseded.

2.3 Example: Automation Without Current Context

An automated agent closes support tickets when it finds a relevant product answer in an internal knowledge base. If the agent retrieves an older article whose answer has been replaced by a security policy, it may take an action that is semantically relevant but operationally wrong.

A memory layer should distinguish similarity from current truth.

3. What RAG Solves

RAG retrieves external information and makes it available to the model at generation time. This addresses a real limitation: the model's pretrained parameters do not contain all private, current, or organization-specific knowledge. Lewis et al. framed RAG as combining parametric memory with a non-parametric memory accessed through retrieval [2].

In a typical enterprise RAG system:

1. Documents are collected from a knowledge base.
2. Documents are split into chunks.
3. Chunks are embedded into vectors.
4. A query is embedded into the same vector space.
5. Similar chunks are retrieved.
6. Retrieved chunks are added to the prompt.
7. The LLM generates an answer using the retrieved context.

This pattern is useful for many knowledge-intensive tasks. Enterprise RAG benchmarks such as EKRAG and WixQA evaluate RAG systems against corporate or support-style question answering tasks [8, 9]. EnterpriseRAG-Bench moves closer to internal-company settings by generating a large synthetic corpus across enterprise source types and testing retrieval and reasoning capabilities, including conflict resolution and absence recognition [14].

4. What Vector Retrieval Does Not Inherently Solve

Vector retrieval ranks text by embedding similarity. This is powerful, but similarity is not the same as truth, authority, recency, permission, or organizational state.

The following table summarizes the gap.

Requirement	Vector RAG helps?	Remaining issue
Find semantically related text	Yes	Related text may not answer the actual operational question.
Use private documents	Yes	Access control must still be enforced outside the vector score.
Prefer current facts	Partly	Recency and supersession need explicit metadata or logic.
Detect conflicts	Partly	Similar chunks can conflict without the retriever knowing which is active.
Explain source authority	Partly	Retrieved chunks need provenance, ownership, and authority metadata.
Share memory across AI tools	Partly	Shared access requires identity, scopes, and isolation.
Prevent unauthorized mutation	No	Retrieval is not a write-governance model.

Longer context windows do not remove the need for these controls. Work on long-context evaluation shows that models can struggle with where relevant information appears in long inputs [3] and that effective context size can be smaller than advertised context length under more demanding tasks [5]. Chroma's 2025 technical report likewise argues that input length can produce nonuniform model performance, especially when ambiguity and distractors are present [15]. These findings support a narrow claim: larger context can help, but larger context alone should not be treated as governed enterprise memory.

5. From Retrieval To Governed Memory

A graph-backed memory layer changes the question from "which chunk is similar?" to "what is the current, permitted, source-backed organizational fact?"

The memory layer should represent at least the following entities. The model is implementation-neutral: products may use different storage engines or names, but they still need an explicit path from scoped evidence to governed memory.

In this model:

- A `SourceFragment` preserves original evidence and provenance.
- A `Claim` represents a typed candidate assertion derived from evidence.
- A `Fact` represents a promoted active memory.
- A superseded fact remains available for audit instead of being overwritten.
- A team boundary controls which knowledge base is visible.
- A profile or API key controls which actor or AI client is reading or writing.
- A readonly key can recall knowledge without mutating it.

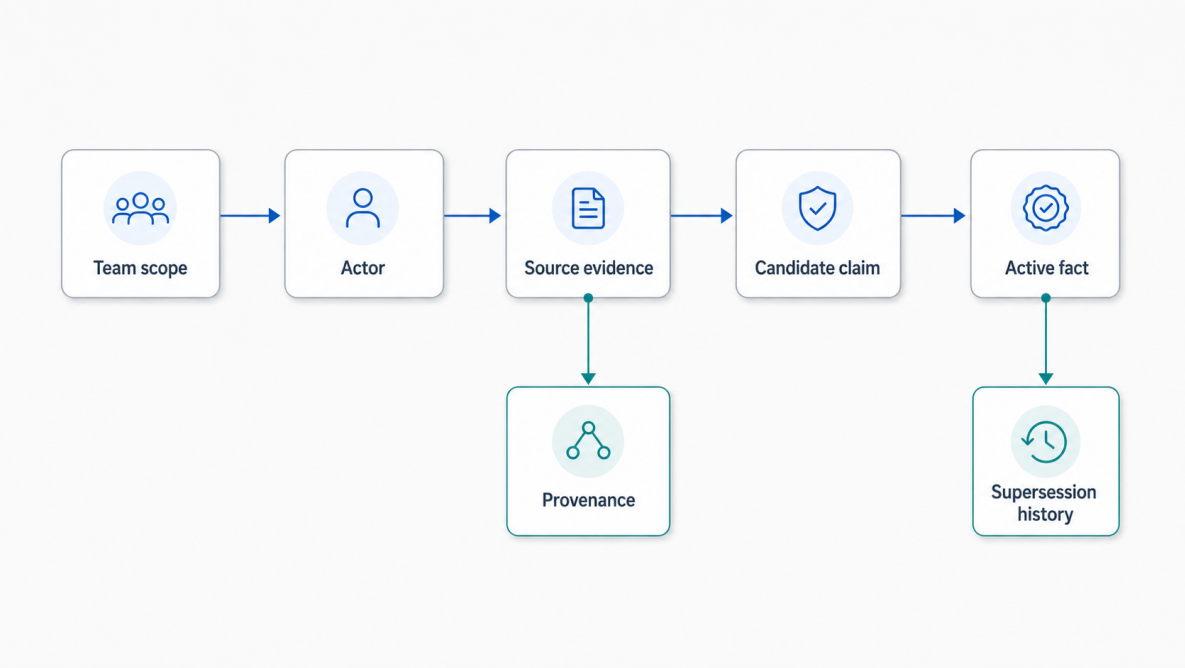

Figure 1: Implementation-neutral governed-memory entity model. A team and actor scope access to source evidence; candidate claims retain support links; active facts preserve lifecycle state, including supersession.

– A write key can submit evidence and candidate claims, subject to governance.

This does not replace RAG. It narrows RAG's job. Vector search remains useful for candidate discovery, but the graph handles provenance, relationships, authority, status, and access policy.

Relationship types are a core benefit of the graph representation. A memory service can model that one source supports a fact, one fact supersedes another, two claims conflict, or a decision depends on a security or contract condition. When these relationships are maintained, retrieval can prioritize active, supported, higher-authority facts while preserving older superseded facts for audit. This is different from ranking chunks by similarity alone: a stale document can still be semantically close to the query, but an explicit supersession edge can mark it as no longer governing the answer.

6. Dense-Mem As An Example Implementation

This paper uses Dense-Mem as an example and evaluation test bench, not as the definition of governed memory. Dense-Mem stores source fragments, typed claims, promoted facts, evidence links, verification metadata, and supersession history. Its MCP and REST surfaces let different clients connect to one memory service. On the current normal-write path, the host submits evidence from the conversation. Dense-Mem handles claim extraction, verification, placement, promotion gates, persistence, audit metadata, recall, and access control [19]. Historical import is a separate migration path that may carry explicit claims and request promotion.

The important implementation properties for this article are:

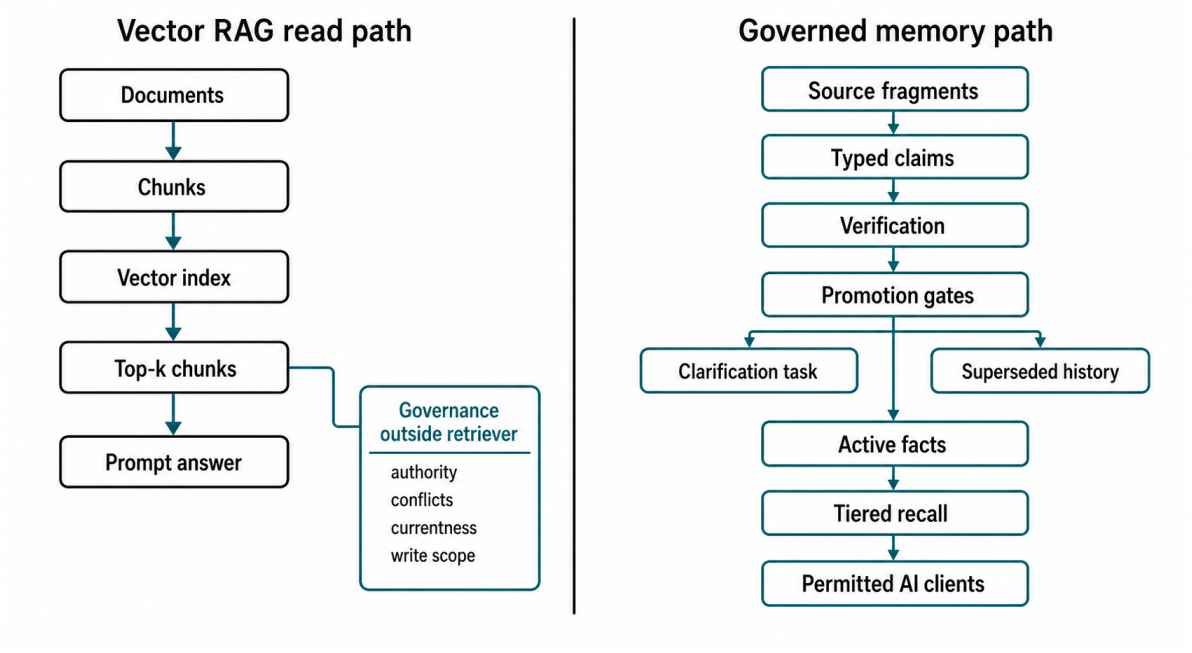

Figure 2: Vector RAG optimizes the read path from documents to retrieved chunks, while governed memory adds evidence storage, claim verification, promotion, permissions, conflicts, and supersession as explicit process stages.

- Evidence is stored before promoted facts.
- Normal clients submit evidence rather than pre-extracted claims or promotion hints.
- Facts are not silently overwritten; corrections supersede older comparable facts.
- Comparable conflicts can return clarification tasks rather than letting the system choose silently.
- Knowledge is team-scoped.
- API keys can be read-scoped or write-scoped.
- The same memory service can serve multiple AI clients.
- Readonly clients can inspect permitted knowledge without mutating it.
- Operator and user portals are intentionally narrower than raw database access.

This enables cross-tool context sharing without collapsing all teams into one unrestricted memory space. A business user can connect a chat assistant with a readonly key to technical context. A developer can connect a coding agent with read access to approved business decisions. A separate write key can add business decisions, engineering constraints, or support incident facts into the same governed memory service.

6.1 Hosted Demo Scope

A hosted Dense-Mem demo is available at https://demo-dense-mem.markhuang.ai (https://demo-dense-mem.markhuang.ai). The demo creates a temporary isolated team and lets users test memory writes, recall, claims, verification, and fact promotion against a live reference instance before self-hosting. It should be treated as a disposable test environment, not as a place for secrets, personal data, credentials, production notes, or durable organizational records.

This matters for the article because the implementation is no longer only a local code artifact. Readers can inspect the behavior boundary directly: temporary team isolation, scoped API keys, MCP client configuration, and the difference between reading memory and mutating memory.

6.2 Process Difference From RAG

Dense-Mem differs from ordinary RAG most clearly at write time. RAG normally optimizes the read path: collect documents, split them into chunks, embed them, and retrieve similar chunks for a prompt. Governed memory adds a write-side lifecycle so the system can later explain why a fact is active, disputed, superseded, or unavailable.

Process question	**Vector RAG process**	**Governed memory process**
What is stored first?	Text chunks optimized for later similarity search.	Source fragments stored as evidence with provenance, source quality, labels, and scope.
What is extracted?	Extraction is optional and often left to the prompt or application layer.	On normal writes, the memory service extracts typed candidate claims from stored evidence. A separate historical-import path may accept explicit claims.
What is verified?	A retrieved chunk may be relevant, but relevance is not a claim-verification state.	A verifier evaluates whether a claim is supported by its evidence before it can become a durable fact.
What becomes current truth?	Currentness is usually encoded through metadata, recency filters, prompt rules, or application logic.	Promotion gates decide whether a validated claim can become an active fact.
What happens to conflicts?	Conflicting chunks can be retrieved together, but the retriever does not decide the active organizational fact.	Comparable conflicts become disputed claims and clarification tasks rather than silent overwrites.

Continued on next page

Process question	Vector RAG process	Governed memory process
What is recalled?	Usually top-ranked chunks.	Active facts, validated claims, and source fragments are returned as distinct authority tiers.
What controls writes?	Write control belongs to the surrounding application or database.	API scopes, team/profile isolation, promotion gates, and audit metadata are part of the memory service contract.

This distinction is why governed memory should not be evaluated only by retrieval accuracy. A system that retrieves a relevant stale document but cannot mark it superseded has solved a different problem from a system that can return the active fact and preserve the old document as evidence.

6.3 Promotion Logic

The high-level memory path stores evidence before it stores facts. A normal memory write creates source fragments and queues server-owned placement. The service extracts candidate claims, verifies each claim against supporting evidence, applies the promotion policy, and records the placement outcome. The client polls that asynchronous run instead of supplying claims or promotion hints. Bulk historical imports use a separate path and default to review first, so they can record evidence and explicit claims without automatically turning every imported assertion into an active fact.

Promotion is deliberately stricter than "store the latest text." A claim must be validated, its predicate must be governed by an explicit promotion policy, and it must pass the predicate's gates. The gates combine extraction confidence, resolution confidence, modality, entailment, and support. Support is not a pure document-count rule: a claim can pass when it has enough supporting fragments or when a high-quality source satisfies the source-quality threshold.

Promotion policy	Process behavior	Example governance meaning
Single-current	Keep one active fact per subject and predicate. A stronger conflicting claim supersedes older facts; a comparable one becomes disputed; a weaker one is rejected.	A person's employer, a current policy owner, or a current project decision.
Multi-valued	Allow multiple active facts for the same subject and predicate.	Skills, tools used, known contacts, or preferences where several values can be true.

Continued on next page

Promotion policy	Process behavior	Example governance meaning
Versioned	Create a new fact version and close the older active version without treating the difference as an error.	Time-evolving organizational facts where history matters.
Append-only	Add immutable history without replacing prior facts.	Corrections, audit notes, or event-like memory.

For single-current facts, the contradiction path is the core governance difference. If a new validated claim has the same object as an existing active fact, it confirms the existing fact. If it conflicts and is stronger, it supersedes the older active fact while preserving the history. If it conflicts and is comparable, Dense-Mem marks the claim as disputed and returns a clarification task for the host to ask the user. If it is weaker, it is rejected instead of becoming active memory.

Promotion is also serialized per claim. This avoids duplicate fact creation when two clients try to promote the same claim at the same time. The design therefore treats promotion as a state transition, not as a loose insertion into a vector index.

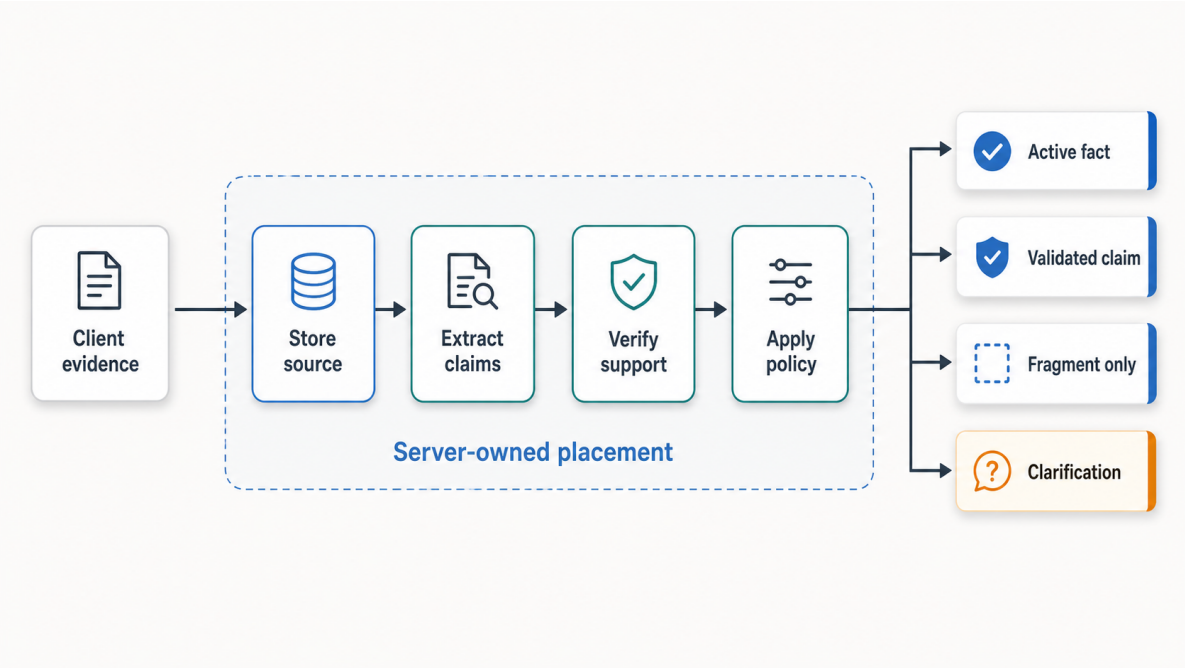

Figure 3: Current Dense-Mem normal-write lifecycle. A client submits evidence; the service stores it, performs asynchronous extraction and verification, applies placement and promotion policy, and returns governed outcomes for recall or clarification.

6.4 Recall, Export/Import, And Memory Portability

Active facts, validated claims, and supporting evidence differ in authority and lifecycle state. Public recall returns bounded evidence contexts in `results[]`, compact relationship context in `discovery_paths[]`, and guidance for a focused follow-up. Section 8.5 evaluates an evidence-first path in which exact, evidence-vector, and evidence-text branches rank the initial evidence set. Relationship-text, relationship-vector, and adjacency branches feed a separate discovery frontier. `trace_memory` can then expand a selected item into its support, contradictions, and supersession history. Similarity and graph structure both inform retrieval, but neither determines authority.

Memory portability matters because useful agent behavior is not carried only by a skill file or tool instruction. A skill file can describe how to run a workflow, but memory carries the situated experience that makes the workflow perform well: what has been tried, which assumptions held, which background conditions mattered, what use case motivated a decision, which corrections were made, and what evidence supports the current answer. A team may therefore need to teach a newly connected agent not only a procedure, but also the governed knowledge and historical context that explain when that procedure should be used.

Export/import is the portability mechanism for that memory layer. It should move approved knowledge between AI clients without reducing it to opaque prompt text or an unreviewed database dump. A portable memory package can carry selected facts, validated claims, manual knowledge, and supporting evidence so the receiving system can inspect what it is being taught. This is closer to sharing an agent's governed experience than sharing a static skill file.

Dense-Mem is one example of this pattern, not the boundary of the concept. Its available export path packages selected facts, validated claims, and manual triples into canonical JSON with a SHA-256 integrity hash; when support is included, the package can carry supporting claims and source fragments so an imported item is not detached from its evidence. That example path is useful for controlled memory transfer, but the broader architectural target is portable governed memory across decisions, policies, temporal facts, background situations, use-case context, and accumulated operating experience.

Imports are designed as a governance workflow. Review-mode import records the bundle as evidence and creates claims without trusting them as active facts. Trusted import can validate and promote imported fact items, but trusted URL imports require an expected hash and conflicts require explicit decisions before local active facts are superseded. Import batches are recorded in a ledger so rollback can be attempted later; rollback is blocked if affected graph entities changed after the import. This favors auditable transfer over silent merge.

6.5 Performance And Operational Trade-Offs

The approach deliberately moves some cost from read-time prompting into write-time governance. Fragment creation may require embeddings, claim verification may call a verifier model, and promotion may traverse the graph, evaluate gates, serialize the claim transition, and write fact relationships. That is heavier than appending a note to a file or inserting a chunk into a vector database.

The benefit is that later reads have more structure to use. A query can return an active fact instead of asking the model to infer current truth from several chunks. A stale fact can remain visible as

superseded history. A comparable conflict can become a user-facing clarification task. A readonly client can recall permitted knowledge while the storage layer rejects mutation.

Section 7.6 reports better scores for explicit graph structure on controlled conflict and multi-hop governance cases. It does not establish production performance superiority for Dense-Mem. Section 8 tests the read path directly. The relationship-heavy approach widened top-k coverage but hurt early ranking, relevance, and latency. The evidence-ID approach recovered much of that loss, yet still trailed the chunk-first control in the original three-axis study. The later six-axis comparison ranked evidence first and exposed relationships through a separate discovery path. That candidate surpassed the retained control on all four retrieval-quality aggregates, although its mean and median latency were higher. On this seed, evidence-first ranking performed better when relationships stayed out of the initial RAG result list and remained available for governance, lineage, and discovery.

7. Benchmark Design

This preprint defines a benchmark plan and includes executable seed-data checks. The current `data/results.jsonl` records local runs: a deterministic graph sanity baseline, a vector retrieval-only baseline, a metadata-filtered RAG answer-generation baseline, a Dense-Mem Docker smoke test, a Dense-Mem fragment-recall answer-generation run, and a 100-scenario RAG-versus-curated knowledge graph comparison. It does not yet report a full typed-fact Dense-Mem corpus-ingestion benchmark. The seed benchmark files in `data/` are a controlled starting point for the first experiment.

7.1 Compared Approaches

The first benchmark should compare four approaches:

1. Long-context prompting over a selected set of source documents.
2. Vector RAG using `text-embedding-3-large` with 3072-dimensional embeddings.
3. Graph and knowledge graph retrieval using explicit entity and relationship traversal.
4. Governed memory using source fragments, claims, active facts, supersession, team isolation, and read/write scopes.

The embedding model is fixed so the first study evaluates retrieval and memory governance behavior rather than embedding-provider variation.

7.2 Public Data Points

The benchmark package separates public evidence about enterprise AI adoption and existing RAG benchmarks from the synthetic enterprise task corpus. The file `data/public_data_points.jsonl` records curated data points such as Stanford AI Index and McKinsey adoption statistics, EnterpriseRAG-Bench corpus scale, WixQA dataset sizes, and Chroma's long-context evaluation scope. These data points are not benchmark results for this article. They support the motivation and prior work framing.

7.3 Scenarios

The synthetic enterprise corpus should test:

- Current business decision recall.
- Superseded decision rejection.
- Conflicting claim detection.
- Missing-evidence refusal.
- Cross-team leakage prevention.
- Readonly mutation denial.
- Multi-client continuity between business and developer tools.
- Source-backed answer generation.

7.4 Metrics

The first benchmark should report:

- Answer correctness.
- Faithfulness to cited source.
- Supersession accuracy.
- Conflict handling accuracy.
- Unauthorized access rate.
- Unauthorized mutation rate.
- Context tokens used.

Security and governance metrics are first-class in this benchmark. A system that answers correctly by leaking another team's knowledge should fail the benchmark.

7.5 Reproducibility Harness

The article package includes a local harness under `benchmark/` for validating seed data, generating the 100-scenario comparison corpus, collecting public-source retrieval evidence, scanning prior-art metadata, running a deterministic graph baseline, and running optional vector, RAG-answer, or Dense-Mem integration baselines when credentials are available. Optional runs must record unavailable dependencies as `not_run` rather than silently treating missing API keys or Docker services as successful benchmark results.

7.6 Preliminary Local Results

Six local runs were executed and recorded in `data/results.jsonl`. Dense-Mem runs were executed after resetting the benchmark Compose project and volumes.

Approach	Result	Interpretation
Deterministic governed graph sanity baseline	9/9 questions passed	The seed fact graph can represent current facts, supersession, disputes, missing evidence, access refusal, and readonly mutation refusal.
Permission-filtered vector retrieval baseline	7/9 questions passed	Vector retrieval found many supporting documents, but failed the conflict-detection case and the readonly-mutation case.
Metadata-filtered RAG answer-generation baseline	7/9 questions passed	RAG generated correct answers for most permitted-context tasks, but failed the conflict case after missing the authoritative product update and cannot enforce readonly mutation at the storage layer.
Dense-Mem Docker integration smoke test	Passed	A local Dense-Mem instance exposed read/write tools to a write-scoped key, hid write tools from a readonly key, and returned `403` for a readonly write attempt.
Dense-Mem fragment-recall answer-generation baseline	8/9 questions passed	Dense-Mem fragment recall plus answer generation handled the readonly mutation case through API scope enforcement, but still failed the conflict case because this run did not promote typed enterprise facts or model supersession as first-class Dense-Mem facts.
RAG versus curated knowledge graph comparison	RAG: 50/100; knowledge graph: 99/100	On 100 synthetic scenarios derived from the same scenario information, RAG over source documents passed simple lookup and supersession cases but failed the conflict-detection and multi-hop evidence cases under the combined retrieval-and-answer rubric. The curated knowledge graph baseline performed better because status, authority, conflict, supersession, and dependency edges were explicit.

These results should be read narrowly. The graph baseline is deterministic and uses curated seed facts. The vector baseline scores retrieval only. The RAG answer-generation and Dense-Mem recall runs use an automated scoring rubric over synthetic documents and should be manually reviewed before being treated as publication-strength empirical results. The 100-scenario comparison recorded aggregate generation latency and token counts for audit: RAG used 154,887 ms, 23,654 prompt tokens, and 5,245 completion tokens, while the curated knowledge graph baseline used 125,545 ms, 27,651 prompt tokens, and 6,118 completion tokens. These local timings should not be generalized as performance claims without repeated runs and a fixed runtime environment. The 100-scenario comparison is measured benchmark data, but it is still synthetic and curated: the knowledge graph facts are generated from scenario templates rather than extracted automatically from documents. The single knowledge graph miss was an answer-classification error: the required facts were retrieved,

but the model returned `answer_active` instead of `allowed_if_all_conditions` for one multi-hop case. The Dense-Mem recall run tests fragment recall and storage-scope enforcement, not a full typed-fact ingestion, promotion, supersession, or conflict-resolution benchmark.

8. Evaluating Retrieval Granularity In Dense-Mem

Section 7 evaluates controlled governance failures. This section asks a narrower question about the read path: does granular relationship retrieval improve RAG recall? The project log called the two retained approaches attempt 7 (relationship-heavy retrieval) and attempt 8 (evidence-ID retrieval). These are experiment numbers, not Dense-Mem release versions. A later six-axis study tested an evidence-first design that reserves relationships for discovery. Other intermediate development runs are outside the scope of the article.

The comparison asks two questions:

1. Does expanding retrieval around fine-grained semantic relationships improve coverage without reducing the rank and relevance of direct evidence?
2. If relationship expansion is too noisy, does making evidence the primary retrieval unit preserve graph context while better matching RAG consumption?

8.1 Evaluation Design

The fixed seed contains 1,000 public documents and 795 evaluation cases: 300 SciFact cases distributed through BEIR, 172 HotpotQA multi-hop cases, and 323 MS MARCO passage cases [20, 21, 22, 23]. All three runs use the same seed hash and return at most ten results. The chunk-first control was retained from Dense-Mem revision `4293ba4`. Each candidate was evaluated in a separate runtime after the read-path logic changed. The candidate artifacts preserve run and artifact hashes but not an exact code revision, so the paper identifies systems by their retrieval logic rather than by product version.

The corpus was ingested through the production memory interface. Evaluation then made exactly one production recall request per case. There was no second retrieval step and the evaluator did not add candidates. All systems used `text-embedding-3-large` with 3,072 dimensions. The candidate imports used `gpt-5.4-mini` for review and verification. The supporting AI judge used the exact model `gpt-5.5` and received the exact initial recall response. Judge scores supplement the retrieval metrics; they are not treated as ground truth.

The primary metrics are Recall@K, mean reciprocal rank (MRR), normalized discounted cumulative gain (nDCG), the rate at which a required reference is ranked first, and recall latency. Recall@K measures whether required source documents appear in the result set. MRR and the rank-1 rate are sensitive to whether useful evidence appears early. nDCG also rewards ordering graded evidence correctly. The package records scalar case-level metrics and source artifact hashes in `data/v3_case_metrics.jsonl` and `data/v3_retrieval_comparison.json`; it excludes corpus text, queries, generated answers, and raw traces.

Design	Primary retrieval unit	Intended advantage	Main evaluation pressure
Chunk-first control	Source documents and chunks	Direct alignment with document-level RAG targets	Less explicit semantic structure in the returned unit
Relationship-heavy approach (attempt 7)	Fine-grained relationships with surrounding evidence	Richer lineage and multi-hop context	Expanded candidates can displace direct evidence
Evidence-ID approach (attempt 8)	Evidence identities with selective semantic context	Preserve evidence lineage while reducing relationship noise	Evidence still has to rank and map cleanly to source-document targets

The target-unit distinction matters. The public qrels primarily identify relevant source documents. They provide a valid test of direct RAG retrieval, but they do not measure access control, supersession, conflict resolution, or the full value of provenance and relationship lineage.

8.2 Relationship-Heavy Semantic Retrieval

The relationship-heavy candidate made fine-grained semantic relationships and their surrounding context prominent in recall. Its strongest result was coverage: Recall@K rose from 0.959329 for the control to 0.969140. Judge answerability rose from 4.481761 to 4.542138, and completeness rose from 4.631447 to 4.642767. On the 172 HotpotQA cases with evidence-level labels, evidence Recall@K was 0.994186.

Those gains did not translate into better overall RAG retrieval. MRR fell from 0.917756 to 0.691015, nDCG fell from 0.924471 to 0.751443, and the required rank-1 rate fell from 0.886792 to 0.527044. Judge relevance fell from 4.457862 to 3.817610. Mean recall latency increased from 332 ms to 1,969 ms, with p95 latency increasing from 535 ms to 2,254 ms.

Metric	Chunk-first control	Relationship-heavy	Evidence-ID
Recall@K	0.959329	0.969140	0.957296
MRR	0.917756	0.691015	0.864721
nDCG	0.924471	0.751443	0.881535
Required reference at rank 1	0.886792	0.527044	0.802516
Mean latency	332 ms	1,969 ms	901 ms
p95 latency	535 ms	2,254 ms	1,126 ms
Judge pass rate	0.871698	0.860377	0.862893
Judge relevance / 5	4.457862	3.817610	4.038994

Continued on next page

Metric	Chunk-first control	Relationship-heavy	Evidence-ID
Judge answerability / 5	4.481761	4.542138	4.510692
Judge completeness / 5	4.631447	4.642767	4.606289

The per-case direction counts show that the aggregate ranking loss was broad. Compared with the control, relationship-heavy MRR was lower on 334 cases and higher on 51; judge relevance was lower on 482 cases and higher on 63; latency was higher on 788 of 795 cases. Recall@K was unchanged on 752 cases, higher on 26, and lower on 17.

These observations are consistent with relationship expansion widening the candidate set while allowing related but less directly responsive material to occupy early ranks. The evaluation did not instrument every internal stage, so it cannot assign the loss to one operation. It does show that fine-grained relationship representation is not, by itself, a drop-in improvement over chunk retrieval for RAG. The architecture gained a small amount of coverage but paid substantially more in rank precision, judged relevance, and latency.

8.3 Evidence-ID Semantic Retrieval

The evidence-ID candidate changed the primary retrieval target from expanded relationships to evidence identities, while preserving semantic context for lineage and multi-hop use. Relative to the relationship-heavy approach, MRR increased by 0.173707, nDCG by 0.130091, and the rank-1 rate by 0.275472. Mean latency fell by 1,068 ms and judge relevance increased by 0.221384. Per case, MRR improved on 313 cases and declined on 93; relevance improved on 290 and declined on 127; latency was lower on 792 of 795 cases.

Evidence-level HotpotQA ranking also improved, although evidence Recall@K declined slightly from 0.994186 to 0.991279. Evidence MRR was 0.954457, evidence nDCG was 0.941505, and an evidence target was ranked first in 0.912791 of cases. On this slice, evidence IDs provided a better bridge between graph structure and the context expected by a RAG consumer.

The redesign still did not beat the control. Its Recall@K was almost equal, with a difference of -0.002034, but MRR remained 0.053035 lower, nDCG 0.042936 lower, and the rank-1 rate 0.084277 lower. Mean latency remained 569 ms higher, about 2.7 times the control mean, and judge relevance remained 0.418868 lower. Across the 795 cases, Recall@K was unchanged on 748. MRR was lower on 121 and higher on 50; judge relevance was lower on 375 and higher on 76; latency was higher on 777.

Dataset and design	Recall@K	MRR	Rank 1	Mean latency
SciFact–control	0.920556	0.837319	0.783333	354 ms
SciFact–relationship-heavy	0.958222	0.746569	0.600000	2,011 ms
SciFact–evidence-ID	0.929611	0.766142	0.670000	899 ms

Continued on next page

Dataset and design	Recall@K	MRR	Rank 1	Mean latency
HotpotQA–control	0.985465	0.988372	0.976744	341 ms
HotpotQA–relationship-heavy	0.994186	0.915698	0.843023	1,929 ms
HotpotQA–evidence-ID	0.991279	0.954457	0.912791	933 ms
MS MARCO–control	0.981424	0.954863	0.934985	307 ms
MS MARCO–relationship-heavy	0.965944	0.519771	0.291022	1,951 ms
MS MARCO–evidence-ID	0.964912	0.908497	0.866873	886 ms

MS MARCO showed the largest recovery, from a relationship-heavy MRR of 0.519771. HotpotQA remained the strongest candidate slice, although this study cannot isolate whether its multi-hop structure caused that result. On SciFact, evidence-ID retrieval gave up some of the relationship-heavy coverage gain and recovered only part of its ranking loss.

8.4 Observed Associations And Lessons

Case-level associations connect the rank metrics to the judge results without establishing causation. Spearman MRR-relevance correlations were 0.567 for the control, 0.687 for the relationship-heavy approach, and 0.567 for the evidence-ID approach. Across paired cases, the change in MRR and the change in relevance had a correlation of 0.601 for relationship-heavy versus control, 0.599 for evidence-ID versus relationship-heavy, and 0.491 for evidence-ID versus control. The association was positive in each dataset, but weaker for HotpotQA; the relationship-heavy versus control interval for HotpotQA included zero. Early rank quality and judged relevance moved together, but the analysis does not show that either metric caused the other.

Exploratory association, all 795 cases	Spearman rho	Bootstrap 95% interval
Control MRR and judge relevance	0.567	0.507 to 0.617
Relationship-heavy MRR and judge relevance	0.687	0.648 to 0.723
Evidence-ID MRR and judge relevance	0.567	0.510 to 0.614
Relationship-heavy versus control: change in MRR and relevance	0.601	0.550 to 0.648
Evidence-ID versus relationship-heavy: change in MRR and relevance	0.599	0.555 to 0.647
Evidence-ID versus control: change in latency and relevance	-0.059	-0.134 to 0.015

The latency-relevance association for evidence-ID versus the control was near zero, and its bootstrap interval crossed zero. Slower recall therefore did not correspond to better judged relevance in this study. Full-cohort and dataset-stratified estimates are recorded in `data/v3_correlations.jsonl`.

The table summarizes five practical lessons from the retained results.

Lesson	**Observation**	**Implication**
Storage granularity is not retrieval quality.	Fine-grained relationships preserved semantics but were poor first-ranked units.	Store granular structure without assuming it should lead the result list.
Coverage and ranking must be reported together.	The relationship-heavy approach improved Recall@K while substantially reducing MRR and rank 1.	A coverage gain can conceal a worse RAG context.
Relationship expansion should be selective.	Broad expansion displaced direct evidence.	Use relationships after strong evidence candidates are identified.
Evidence is a better graph-to-RAG interface.	Evidence-ID retrieval recovered much of the relationship-heavy loss.	Rank and cite evidence, then expose graph context for follow-up.
Governance and retrieval require separate tests.	Added lineage, permissions, conflicts, and history did not improve direct-document qrels.	Evaluate governance value independently from retrieval accuracy.

8.5 Six-Axis Follow-Up: Evidence First, Relationships For Discovery

The follow-up separates evidence ranking from graph exploration more strictly. The relationship-heavy approach allowed granular semantic structure to compete with direct evidence. The evidence-ID approach reduced the rank loss but did not remove it. In the follow-up design, only evidence can enter the initial ranked result set. Relationships appear later as bounded discovery and trace context.

8.5.1 From The Three-Axis Seed To A Six-Axis Standard

Sections 8.1 through 8.4 cover three axes: SciFact, HotpotQA, and MS MARCO. The six-axis standard keeps those task types and adds MuSiQue, QASPER, and LongMem Oracle. MuSiQue puts more pressure on connected multi-hop evidence, QASPER tests evidence retrieval inside long research papers, and LongMem Oracle tests memory retrieval across long multi-session conversations. Later Dense-Mem evaluations will use this six-axis frame because the three-axis seed cannot expose all six task-specific failure patterns. The six labels name task families rather than metrics, and each family uses the same retrieval measures. Together they cover scientific, web-passage, multi-document, connected multi-hop, long-paper, and multi-session memory retrieval [21, 23, 22, 24, 25, 26].

Axis	**Cases**	**What it is intended to measure**
SciFact	25	Scientific claim-to-document retrieval
MS MARCO	84	Short web-query passage retrieval
HotpotQA	13	Multi-document supporting-evidence retrieval
MuSiQue	27	Connected multi-hop evidence retrieval
QASPER	48	Evidence retrieval within long papers and across answer forms
LongMem Oracle	9	Long-term chat-memory retrieval on non-abstention oracle cases
Total	206	Six retrieval task families

The evaluation uses deterministic seed `public_6axis_1k_v1`, with 1,000 corpus rows and 206 cases. Both the control and candidate completed the full seed. Validation checked the counts and hashes, source identifier uniqueness, row lengths, qrel resolution, and metadata leakage. Evidence was split rather than truncated; each generated row contains at most 999 Unicode code points. Seed hashes and compact validation records are in `data/v3_six_axis_comparison.json`.

The added task families expose failures that the earlier study could not, but the completed 1k run remains a smoke comparison. Its allocation is small and uneven. The MuSiQue slice contains only 2-hop cases, and the LongMem slice contains only knowledge-update cases. One run cannot estimate run-to-run variation or performance on production workloads.

8.5.2 Evaluated Read Path And Protocol

The candidate uses branch-priority fusion for the initial result set.

The initial ranking uses this priority order: exact evidence, evidence vector, then evidence text. Relationship-text, relationship-vector, and adjacency candidates feed a bounded discovery frontier rather than the top-k fusion. `trace_memory` expands selected knowledge into its support and lifecycle history. This experiment scores the initial evidence ranking. It does not show whether following a relationship path improves a later recall.

Both systems imported all 1,000 corpus rows, and neither recorded an import failure. They used the same seed hash, a top-k limit of ten, `text-embedding-3-large`, and 3,072 embedding dimensions. Each case made one production recall request; the evaluator added no candidates and made no second retrieval. The evidence-first candidate used `gpt-5.4-mini` for review and verification, and

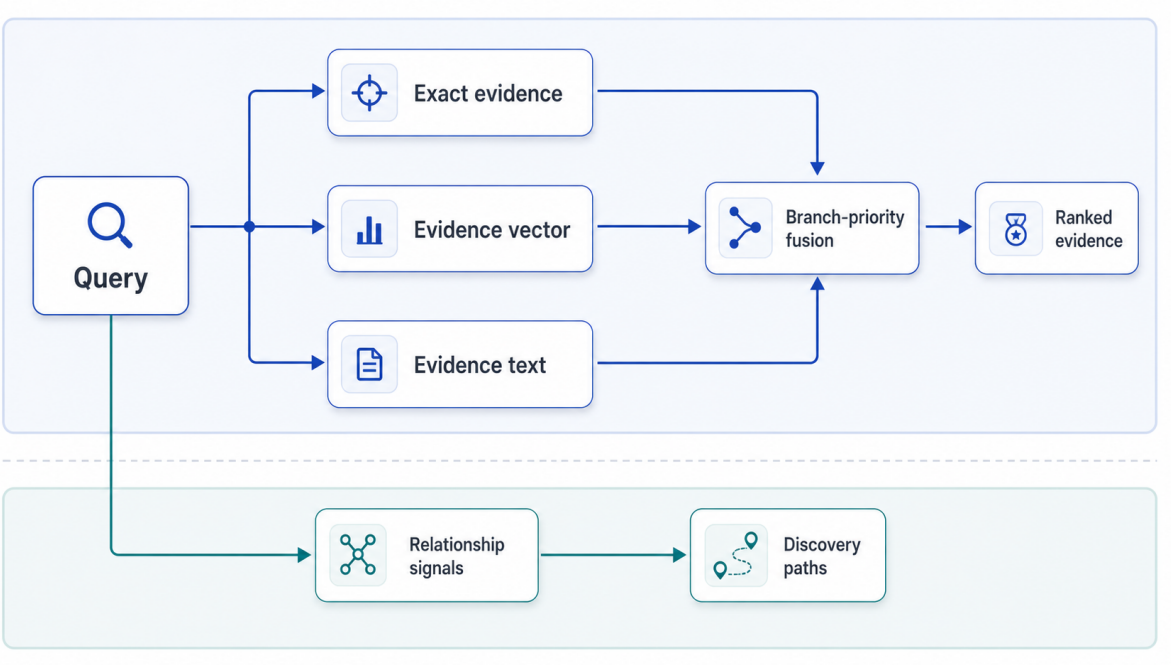

Figure 4: Evidence-first retrieval ranks exact, vector, and text evidence into the initial result set. Relationship text, relationship vectors, and adjacency feed a separate bounded discovery path for navigation and trace.

the answer-quality judge used `gpt-5.5` on the initial response.

Latency is not controlled because the retained control used REST and the evidence-first candidate used MCP. The control trace also lacks the exact initial response required by the current judge contract, so the candidate judge scores have no comparable control baseline. The run artifacts retain run, seed, score, and source-artifact hashes, but omit the exact code revision and active ranking-profile fields.

8.5.3 Results On The 1k Corpus

On the matched 206-case seed, the evidence-first candidate improved all four direct retrieval aggregates over the retained chunk-first control. The gains range from 4.3 to 6.8 percentage points within this fixed smoke comparison; they do not establish general production superiority. Mean latency increased by 196 ms, while p95 latency changed by 1 ms. Because the runs used different transports, these latency values cannot isolate the cost of the architecture.

Metric	Chunk-first control	Evidence-first candidate	Difference
Recall@10	0.842961	0.886003	+0.043042
MRR	0.806916	0.856060	+0.049145
nDCG@10	0.786505	0.845718	+0.059213

Continued on next page

Metric	Chunk-first control	Evidence-first candidate	Difference
Required reference at rank 1	0.762136	0.830097	+0.067961
Mean latency	331 ms	527 ms	+196 ms
p50 latency	276 ms	491 ms	+215 ms
p95 latency	652 ms	653 ms	+1 ms

The micro-average weights each case equally, so the 84 MS MARCO cases and 48 QASPER cases count more than the 9 LongMem cases. The equal-axis macro average weights each task family equally. Macro Recall@10 improved from 0.841297 to 0.910359, MRR from 0.828743 to 0.863231, nDCG@10 from 0.788349 to 0.851283, and rank 1 from 0.781149 to 0.831627. Most cases did not change. Recall@10 increased on 25 cases, declined on 8, and remained unchanged on 173. MRR increased on 26, declined on 16, and remained unchanged on 164.

Axis	Cases	Control R@10	Candidate R@10	Control MRR	Candidate MRR
SciFact	25	0.960000	1.000000	0.893333	0.980000
MS MARCO	84	0.964286	1.000000	0.925430	0.994048
HotpotQA	13	1.000000	1.000000	1.000000	1.000000
MuSiQue	27	0.759259	0.888889	0.981481	1.000000
QASPER	48	0.586458	0.573264	0.412955	0.448727
LongMem Oracle	9	0.777778	1.000000	0.759259	0.756614

Two task families regressed on one metric. QASPER Recall@10 declined by 0.013194 even though its MRR improved. LongMem MRR declined by 0.002646 while its recall rose to 1.0. SciFact, MS MARCO, and MuSiQue improved on both measures. The small HotpotQA slice was already saturated on Recall@10 and MRR.

The candidate-only judge recorded a 0.839806 pass rate, 0.092233 partial rate, and 0.067961 fail rate. Its mean scores were 4.466019 for answerability, 4.004854 for relevance, 4.504854 for completeness, and 4.694175 for faithfulness. These absolute scores describe the candidate; without a replayable control response they provide no judge-based comparative evidence.

8.5.4 Pros, Cons, And Lessons

Dimension	Evidence	Interpretation
Direct retrieval	All four micro aggregates improved by 4.3 to 6.8 percentage points.	The evidence-first candidate worked better than the control on this fixed smoke comparison.

Continued on next page

Dimension	Evidence	Interpretation
Axis balance	All four equal-axis macro aggregates also improved.	The 84-case MS MARCO slice does not explain the aggregate result by itself.
Known regressions	QASPER Recall@10 fell by 0.013194, and LongMem MRR fell by 0.002646.	The candidate did not improve every task family or metric.
Latency	Mean and p50 latency were higher, while p95 differed by 1 ms.	The REST-to-MCP transport mismatch prevents architectural attribution.
Judge and provenance	The control cannot be replayed through the current judge, no repeated runs measure variance, and the candidate artifact omits its exact commit and ranking profile.	Judge-based comparison and run-to-run stability remain untested.
Relationship value	Relationships remained available in discovery and trace but were excluded from initial fusion.	The study does not show whether following a discovery path improves a later answer.

Broad relationship expansion was a poor initial ranker. Evidence-ID retrieval was better, but did not beat the control in the earlier three-axis study. On the six-axis 1k seed, evidence-first branch priority beat the retained control after relationships moved out of initial fusion. In these runs, branch role and fusion order mattered as much as the stored representation. The six-axis frame exposes more retrieval failure modes and should remain the default for later reports. Until repeated runs or broader workloads confirm the pattern, the finding applies only to this 1,000-document, 206-case smoke comparison.

9. Relationship To Existing Work

This article does not claim that graph retrieval is new. GraphRAG and knowledge graph-guided RAG are active areas. Microsoft's GraphRAG approach constructs entity graphs and community summaries for query-focused summarization over private corpora [4]. Knowledge graph-guided RAG has been evaluated for improving retrieval and answer organization [7], and the GraphRAG survey literature frames graph-augmented retrieval as a broad design space [6].

This article's focus is narrower and more operational:

- Enterprise memory must connect multiple AI clients.
- Shared memory must preserve evidence and fact history.
- Read and write permissions are not optional deployment details.
- Stale facts and conflicting facts must be represented explicitly.
- Governance failures should be benchmarked alongside answer quality.

Zep's temporal knowledge graph architecture for agent memory is especially relevant prior work because it emphasizes dynamic enterprise knowledge, conversation-derived memory, and temporal graph structure [10]. More recent enterprise-agent memory work is even closer. Taheri's Governed Memory preprint frames a shared memory and governance layer for multi-agent workflows and reports controlled experiments on governance routing, token reduction, cross-entity leakage, and governed-memory saturation [11]. Chamarty's Context Objects proposal treats temporal validity, provenance lineage, confidence, organizational weight, and feedback as first-class retrieval properties for enterprise AI agents [12]. Srinivasan's Stateless Decision Memory argues that regulated enterprise agents require deterministic replay, auditable rationale, multi-tenant isolation, and statelessness for horizontal scale [13].

These works narrow the novelty claim of this article. The contribution here is not "governed memory exists" or "temporal enterprise memory is new"; it is a pedagogical and benchmark-oriented framing of the enterprise context-loss problem, with Dense-Mem used as one example implementation for permissioned knowledge graph memory, fact supersession, conflicts, and cross-client access control.

10. Discussion

RAG moves LLM systems beyond pretrained knowledge, but RAG by itself is not a complete enterprise memory model. A vector database can retrieve similar text, but enterprises also need to know who can access the text, whether the fact is current, where the fact came from, whether a newer fact supersedes it, and whether two sources conflict.

The opportunity is to make shared memory an enterprise infrastructure layer. Instead of every AI tool starting from a cold prompt or a disconnected knowledge-base index, tools can connect to a governed memory service. This does not require every tool to use the same model or the same user interface. It requires a shared memory contract:

- Read permitted facts and cite their source evidence.
- Respect team boundaries and reject unauthorized mutation.
- Surface conflicts and preserve supersession history.

The three retrieval designs put relationships in different positions on the read path. In the first study, the relationship-heavy approach found slightly more required documents in the top ten, but ranked direct evidence later, returned context judged less relevant, and took substantially longer. The evidence-ID approach improved the result but still trailed the chunk-first control. The later

six-axis study reversed that ordering on all four retrieval aggregates after the initial ranking was limited to evidence and relationships moved to a separate discovery surface. In this preliminary comparison, the highest-scoring design ranked evidence first and used relationships for navigation, lineage, and governance.

This framing builds on the author's prior informal discussion of AI memory beyond RAG [18], but this article narrows the claim toward an enterprise-governance architecture and a benchmarkable evaluation plan.

For enterprise automation, this matters because actions depend on current context. The goal is not to make AI systems omniscient. The goal is to make the available organizational context explicit, source-backed, permissioned, and auditable.

11. Limitations

This preprint defines the architecture, benchmark plan, preliminary local baselines, and two direct public-corpus retrieval comparisons. It still does not include a full Dense-Mem governance benchmark that jointly measures promotion, supersession, conflicts, permissions, retrieval, and generated answers. The initial governance corpus remains synthetic so that stale facts, conflicts, team boundaries, and access-control cases can be controlled. Synthetic data weakens claims about real-world frequency, but strengthens evaluation control for governance failure modes.

The three-axis retrieval study contains one retained run family on one fixed seed: 1,000 documents, 795 cases, one embedding model, a top-k limit of ten, and one AI judge model. The control was retained while the two candidates ran in separate fresh environments. The latency values describe that comparison, not production performance.

The six-axis result comes from one 1,000-document, 206-case comparison. Its case allocation is uneven: it has only 9 LongMem cases and covers only 2-hop MuSiQue and knowledge-update LongMem subtypes. The control used REST while the candidate used MCP, which makes the latency difference descriptive. The control also lacks the initial response needed for a comparable judge replay. The retained configurations omit the exact candidate code revision and active ranking profile. Repeated runs under the same transport are needed to measure runtime variance, judge stability, and persistence of the retrieval gains.

The public qrels reward direct source-document retrieval. That target favors a chunk-first system when the answer already appears in one document and can undermeasure provenance, multi-hop relationships, temporal validity, and governance behavior. The HotpotQA evidence-level results partly address the target mismatch, but only for 172 cases. The reported correlations are exploratory, include tied ordinal judge scores, and do not establish causation.

The article also does not yet quantify how often context is lost across specific enterprise AI products, such as office copilots, coding agents, chat assistants, and workflow agents. It does not compare the cost of governed shared memory with custom model training. Those questions matter, but they require a different survey or deployment study than the seed benchmark included here.

The remaining evaluation work includes:

– A governance-focused benchmark that scores permissions, supersession, conflicts, temporal

validity, provenance, and answer quality together.

- A repeated matched control and candidate run on the validated six-axis 1k seed, with exact code revisions, ranking profiles, transports, and replayable judge inputs.
- Repeated public retrieval runs across seeds, embedding models, retrieval limits, and controlled runtime environments.
- A direct evaluation of whether relationship discovery paths improve focused follow-up retrieval and final answers after evidence ranking.
- Broader memory export/import coverage for enterprise decisions, policies, temporal facts, background situations, and accumulated operating experience.
- A real-world case study whose data can be shared without exposing private organizational information.

12. Conclusion

Enterprise AI systems need more than larger context windows and isolated RAG indexes. They need governed shared memory: source-backed, permissioned, team-isolated, and capable of representing fact history. Vector retrieval is a useful component, but similarity alone is not current organizational truth. A knowledge graph memory layer can make facts, sources, teams, actors, conflicts, and supersession explicit.

The retrieval results narrow the claim. In the three-axis study, fine-grained relationship expansion slightly improved top-k coverage but lost to chunk-first RAG on rank-sensitive quality, judged relevance, and latency. Evidence-ID retrieval recovered much of the loss but still trailed the control. On the six-axis 1k seed, evidence-first branch-priority ranking surpassed the retained control on Recall@10, MRR, nDCG@10, and rank-1 rate after relationships moved to discovery. Those gains, 4.3 to 6.8 percentage points, apply only to this fixed smoke comparison and do not yet establish a general performance result. The evaluated design ranks source evidence first and uses relationships to connect, trace, and govern it.

Revision History

- Version 1, 2026-05-30: Initial preprint with the governed-memory architecture, benchmark plan, seed datasets, and local baseline results.
- Version 2, 2026-06-02: Added hosted demo context, expanded the process discussion from source fragments through claims, verification, promotion, recall, and export/import, reframed memory portability as governed knowledge and experience transfer, added process-flow figures, and clarified benchmark and portability limitations.
- Version 3, 2026-07-17: Added a 795-case public-corpus comparison of attempts 7 and 8, the relationship-heavy and evidence-ID approaches, against the retained chunk-first control. Added a six-axis evidence-first smoke comparison on a 1,000-document seed, moved relationship discovery out of the initial evidence ranking, and adopted six axes as the standard reporting frame.

References

[1] Vaswani, Ashish; Shazeer, Noam; Parmar, Niki; Uszkoreit, Jakob; Jones, Llion; Gomez, Aidan N.; Kaiser, Lukasz; Polosukhin, Illia (2017). *Attention Is All You Need*. Advances in Neural Information Processing Systems. https://arxiv.org/abs/1706.03762.

[2] Lewis, Patrick; Perez, Ethan; Piktus, Aleksandra; Petroni, Fabio; Karpukhin, Vladimir; Goyal, Naman; Kuttler, Heinrich; Lewis, Mike; Yih, Wen-tau; Rocktaschel, Tim; Riedel, Sebastian; Kiela, Douwe (2020). *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. Advances in Neural Information Processing Systems. https://arxiv.org/abs/2005.11401.

[3] Liu, Nelson F.; Lin, Kevin; Hewitt, John; Paranjape, Ashwin; Bevilacqua, Michele; Petroni, Fabio; Liang, Percy (2024). *Lost in the Middle: How Language Models Use Long Contexts*. Transactions of the Association for Computational Linguistics. https://arxiv.org/abs/2307.03172.

[4] Edge, Darren; Trinh, Ha; Cheng, Newman; Bradley, Joshua; Chao, Alex; Mody, Apurva; Truitt, Steven; Metropolitansky, Dasha; Ness, Robert Osazuwa; Larson, Jonathan (2024). *From Local to Global: A Graph RAG Approach to Query-Focused Summarization*. arXiv:2404.16130. https://arxiv.org/abs/2404.16130.

[5] Hsieh, Cheng-Ping; Sun, Simeng; Kriman, Samuel; Acharya, Shantanu; Rekesh, Dima; Jia, Fei; Zhang, Yang; Ginsburg, Boris (2024). *RULER: What's the Real Context Size of Your Long-Context Language Models?*. arXiv:2404.06654. https://arxiv.org/abs/2404.06654.

[6] Han, Haoyu; Wang, Yu; Shomer, Harry; Guo, Kai; Ding, Jiayuan; Lei, Yongjia; Halappanavar, Mahantesh; Rossi, Ryan A.; Mukherjee, Subhabrata; Tang, Xianfeng; He, Qi; Hua, Zhigang; Long, Bo; Zhao, Tong; Shah, Neil; Javari, Amin; Xia, Yinglong; Tang, Jiliang (2025). *Retrieval-Augmented Generation with Graphs (GraphRAG)*. arXiv:2501.00309. https://arxiv.org/abs/2501.00309.

[7] Zhu, Xiangrong; Xie, Yuexiang; Liu, Yi; Li, Yaliang; Hu, Wei (2025). *Knowledge Graph-Guided Retrieval Augmented Generation*. Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies. DOI: https://doi.org/10.18653/v1/2025.naacl-long.449. https://aclanthology.org/2025.naacl-long.449/.

[8] Yu, Tan; Zhou, Wenfei; Yang, Lei; Shukla, Aaditya; Madugula, Meenakshi; Gundecha, Pritam; Burnett, Nick; Xu, Anbang; Seth, Vishal; Bar, Tamar; Akkiraju, Rama; Zhang, Vivienne (2025). *EKRAG: Benchmark RAG for Enterprise Knowledge Question Answering*. Proceedings of the 4th International Workshop on Knowledge-Augmented Methods for Natural Language Processing. DOI: https://doi.org/10.18653/v1/2025.knowledgenlp-1.13. https://aclanthology.org/2025.knowledgenlp-1.13/.

[9] Cohen, Dvir; Burg, Lin; Pykhnivskyi, Sviatoslav; Gur, Hagit; Kovynov, Stanislav; Atzmon, Olga; Barkan, Gilad (2025). *WixQA: A Multi-Dataset Benchmark for Enterprise Retrieval-Augmented Generation*. arXiv:2505.08643. https://arxiv.org/abs/2505.08643.

[10] Rasmussen, Preston; Paliychuk, Pavlo; Beauvais, Travis; Ryan, Jack; Chalef, Daniel (2025). *Zep: A Temporal Knowledge Graph Architecture for Agent Memory*. arXiv:2501.13956. https://arxiv.org/abs/2501.13956.

[11] Taheri, Hamed (2026). *Governed Memory: A Production Architecture for Multi-Agent Workflows*. arXiv:2603.17787. https://arxiv.org/abs/2603.17787.

[12] Chamarty, Madhu (2026). *Context Objects: A Temporal, Provenance-Aware Memory Primitive for Enterprise AI Agents*. DOI: https://doi.org/10.2139/ssrn.6775102. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=6775102.

[13] Srinivasan, Vasundra (2026). *Stateless Decision Memory for Enterprise AI Agents*. arXiv:2604.20158. https://arxiv.org/abs/2604.20158.

[14] Sun, Yuhong; Rahmfeld, Joachim; Weaver, Chris; Chen, Weijia; Desai, Roshan; Huang, Wenxi; Butler, Mark H. (2026). *EnterpriseRAG-Bench: A RAG Benchmark for Company Internal Knowledge*. arXiv:2605.05253. https://arxiv.org/abs/2605.05253.

[15] Hong, Kelly; Troynikov, Anton; Huber, Jeff (2025). *Context Rot: How Increasing Input Tokens Impacts LLM Performance*. Chroma. https://www.trychroma.com/research/context-rot.

[16] McKinsey & Company (2025). *The State of AI in 2025: Agents, Innovation, and Transformation*. McKinsey & Company. https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai. See key findings item 3, section "Many organizations are already experimenting with AI agents," and section "AI as a catalyst for innovation."

[17] Stanford Institute for Human-Centered Artificial Intelligence (2026). *The 2026 AI Index Report*. Stanford University. https://hai.stanford.edu/ai-index/2026-ai-index-report. See Economy chapter, item 5.

[18] Huang, Zhirong (2026). *AI Memory Beyond RAG*. https://markhuang.ai/blog/ai-memory-beyond-rag. Informal blog post.

[19] Huang, Zhirong (2026). *Dense-Mem*. https://github.com/markhuangai/dense-mem. Software repository and gray literature.

[20] Thakur, Nandan; Reimers, Nils; Rücklé, Andreas; Srivastava, Abhishek; Gurevych, Iryna (2021). *BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models*. Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks. https://arxiv.org/abs/2104.08663.

[21] Wadden, David; Lin, Shanchuan; Lo, Kyle; Wang, Lucy Lu; van Zuylen, Madeleine; Cohan, Arman; Hajishirzi, Hannaneh (2020). *Fact or Fiction: Verifying Scientific Claims*. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing. DOI: https://doi.org/10.18653/v1/2020.emnlp-main.609. https://aclanthology.org/2020.emnlp-main.609/.

[22] Yang, Zhilin; Qi, Peng; Zhang, Saizheng; Bengio, Yoshua; Cohen, William W.; Salakhutdinov, Ruslan; Manning, Christopher D. (2018). *HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering*. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. DOI: https://doi.org/10.18653/v1/D18-1259. https://aclanthology.org/D18-1259/.

[23] Bajaj, Payal; Campos, Daniel; Craswell, Nick; Deng, Li; Gao, Jianfeng; Liu, Xiaodong; Majumder, Rangan; McNamara, Andrew; Mitra, Bhaskar; Nguyen, Tri; Rosenberg, Mir; Song, Xia; Stoica, Alina; Tiwary, Saurabh; Wang, Tong (2016). *MS MARCO: A Human*

Generated Machine Reading Comprehension Dataset. arXiv preprint arXiv:1611.09268. https://arxiv.org/abs/1611.09268.

[24] Trivedi, Harsh; Balasubramanian, Niranjan; Khot, Tushar; Sabharwal, Ashish (2022). *MuSiQue: Multihop Questions via Single-hop Question Composition*. Transactions of the Association for Computational Linguistics. DOI: https://doi.org/10.1162/tacl_a_00475. https://aclanthology.org/2022.tacl-1.31/.

[25] Dasigi, Pradeep; Lo, Kyle; Beltagy, Iz; Cohan, Arman; Smith, Noah A.; Gardner, Matt (2021). *A Dataset of Information-Seeking Questions and Answers Anchored in Research Papers*. Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. DOI: https://doi.org/10.18653/v1/2021.naacl-main.365. https://aclanthology.org/2021.naacl-main.365/.

[26] Wu, Di; Wang, Hongwei; Yu, Wenhao; Zhang, Yuwei; Chang, Kai-Wei; Yu, Dong (2024). *LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory*. arXiv:2410.10813. https://arxiv.org/abs/2410.10813.