OIDA: Organizational Knowledge in the Age of Associative Intelligence

Technical Report

Federico Bottino[1], Carlo Ferrero[1], Nicholas Dosio[1], Pierfrancesco Beneventano[2]

[1] *Kakashi Ventures Accelerator*

[2] *Massachusetts Institute of Technology*

V1: March 2026

Abstract

Organizational corpora used by AI agents typically fail to distinguish between decisions, hypotheses, stale observations, and unresolved contradictions. We describe OIDA, a framework that represents organizational knowledge as typed Knowledge Objects carrying epistemic class, confidence, temporal validity, and contradiction status as computable properties. A deterministic update engine maintains importance scores with class-specific decay; a hybrid retrieval architecture combines structural, semantic, and topological similarity to rank results by epistemic weight. We hypothesise that structuring knowledge epistemically at ingestion—rather than improving retrieval over unstructured data—is the intervention that produces qualitatively different capabilities for AI agents.

We report the architecture, preliminary observations from deployment on approximately 500 Knowledge Objects, and open calibration questions. Classification at ingestion is LLM-assisted and therefore fallible; all subsequent maintenance and retrieval is deterministic. The parameters are working heuristics, not validated optima; systematic evaluation is the next milestone.

1 Introduction

The quantity of information available to modern organisations is not the binding constraint on decision quality. The binding constraint is the inability to determine, at any given moment, what that information *means*: which claims are well-supported, which are contested, which have become obsolete, and which stand in direct contradiction to active commitments.

When AI agents attempt to reason over organisational knowledge, this gap becomes operationally consequential. Without explicit epistemic structure, an agent cannot distinguish a verified decision from an abandoned hypothesis, cannot identify that a strategic assumption is contradicted by recent evidence, and treats an open question as equivalent to a settled conclusion—because nothing in the substrate told it otherwise.

1.1 Why Better Retrieval Is Not Sufficient

The dominant response to this problem has been to improve retrieval: better embeddings, denser indexes, reranking, hybrid search. These are legitimate improvements within their scope, and they are part of OIDA's architecture. But they do not resolve the underlying issue on an epistemically flat substrate.

Consider a concrete case. An organisation has a strategy document, three meeting notes, a contradicting market report, and a decision recorded in Slack. A retrieval system—however sophisticated—can surface these documents. It cannot determine: which of these is a binding decision versus a tentative observation; whether the contradiction has been acknowledged or remains unresolved; whether the decision remains valid given the new evidence; or how confident the organisation should be given subsequent developments. Even when it is explicit how to determine, often AI agents run out of context and they fail at the final task. These questions require epistemic structure, not better retrieval.

1.2 The Access Problem: From Addressed Lookup to Associative Retrieval

There is a deeper reason why better retrieval alone is insufficient. For a century, data structures and access algorithms have been optimised for a single consumer: the CPU executing addressed, sequential instructions. The questions that drove this work—worst-case lookup time, amortised cost of a splay operation, minimising page faults—are the right questions for that paradigm.

Today, the primary consumers of organisational knowledge are no longer only CPUs. They are neural networks—both artificial (LLMs, embedding models, agent systems) and biological (the humans who must decide from what those systems surface). Both operate through associative retrieval: contextual activation of related representations, weighted by relevance, recency, and structural importance, rather than addressed lookup against a static index. Neither benefits from access patterns optimised for sequential logical processing.

This shift changes what "good access" means. For a CPU, optimal access minimises I/O against an index. For an associative consumer, optimal access maximises the *epistemic utility* of what is surfaced given the current context—which requires that the priority, temporal validity, and epistemic quality of knowledge be computationally explicit, not left implicit in unstructured text.

OIDA does not propose new data structures. It proposes a new access layer on top of established infrastructure—PostgreSQL, pgvector, graph queries—designed to prioritise knowledge by its utility to associative consumers rather than by its proximity to a query string. Whether this reframing produces measurably better outcomes is the empirical question this line of work aims to answer.

1.3 Our Approach

OIDA[1] is a computational framework that addresses this gap by structuring knowledge epistemically at ingestion time, maintaining that structure dynamically through deterministic computation, and exposing it to AI agents as a first-class feature of the retrieval substrate.

The core design decision is that **epistemic maintenance is computational, not editorial**. No language model decides what is obsolete. No human curator manually flags contradictions. The framework computes epistemic state—importance, decay, contradiction pressure—cycle by cycle, from class-specific parameters and real organisational signals. We chose this deliberately: auditability requires that the same inputs always produce the same epistemic trajectory, regardless of which foundation model is used downstream.

Scope of the determinism guarantee. Classification at ingestion is LLM-assisted and therefore fallible. But once a Knowledge Object is instantiated with its class and edges, all subsequent maintenance—decay, importance scoring, contradiction propagation, memory zone allocation, and retrieval ranking—is deterministic and auditable. The framework's guarantee is: *deterministic maintenance given the instantiated graph*, not *deterministic end-to-end from raw input.*

1.4 Contributions

1. A formal model of the **Knowledge Object** (KO) as the foundational unit of organisational epistemology, with nine epistemic classes, a seven-axis coordinate system, and a closed vocabulary of typed relationships (§3.1).
2. The **Knowledge Gravity Engine** (KGE): a deterministic update rule for computing emergent importance scores with class-specific decay profiles (§3.2).

[1] From Greek *oida* ("I know because I have seen")—a knowing that arises from direct experience rather than declaration.

3. A **hybrid retrieval architecture** that combines structural, semantic, and topological similarity with importance-score modulation (§3.3).
4. An end-to-end example illustrating how the system processes an organisational decision under epistemic pressure (§4).
5. Preliminary observations from deployment, with honest acknowledgement of open calibration questions (§5).

This paper describes deployed infrastructure, not a validated theory. The framework is operational at a single organisation; whether the configured parameters generalise is an empirical question requiring broader deployment.

Acknowledgements

We thank Alberto Trivero and Tommaso Portaluri for valuable discussion on relevant AI, statistical, and informatics matters.

Contents

Table 1: Principal notation used in this report.

Symbol	**Meaning**
K, K^*	Importance score (emergent, per KO); fixed-point value under stationary inputs
K_{eff}	Contextual importance (query-modulated)
η	Learning rate / momentum parameter (default 0.15)
λ_{class}	Class-specific decay rate
Δt	Time elapsed since last KGE cycle (in units of 6-hour intervals; default 0.25)
a_u, a_e, a_c	Scaling constants for usage, evidence, and contradiction forces in the KGE
α, β, γ	Retrieval weights for structural, semantic, and topological similarity
$H(a, b)$	Hybrid similarity score between KOs a and b
$S_{\text{struct}}, S_{\text{sem}}, S_{\text{topo}}$	Structural, semantic, and topological similarity components

2 Related Work

Knowledge graphs and ontological systems. Knowledge graphs represent information as networks of entities and typed relationships [1]. Enterprise deployments—notably Palantir's Ontology [2]—map operational reality into semantically coherent graphs. OIDA addresses a complementary problem: not the structure of what exists and how it is connected, but the dynamics of what is known, how confidently, and for how long.

Structured retrieval-augmented generation. RAG [5] is the dominant paradigm for grounding LLM outputs. GraphRAG [6] demonstrated that graph-based community summaries improve retrieval quality over flat chunk retrieval. OIDA extends this insight by adding epistemic typing, dynamic scoring, and temporal decay—properties that entity-extraction-based graphs do not model.

Memory systems for AI agents. MemGPT [3] introduced hierarchical memory for agents. Zep [4] provides temporal tracking and graph-based storage via Graphiti. OIDA adds epistemic typing and dynamic importance computation as first-class properties: not just surfacing relevant memories, but computing how much weight an agent should assign to each.

Cognitive architectures. OIDA's usage force inherits structure from ACT-R's base-level activation [7, 8], using an exponential kernel rather than ACT-R's power-law sum (a convenience approximation). The class-specific decay replaces ACT-R's global decay parameter with typed profiles. The gravity propagation through signed edges—where negative-coefficient edges suppress importance—has no ACT-R analogue.

Positioning. Relative to knowledge graphs, OIDA adds epistemic dynamics. Relative to structured RAG, it adds typed importance computation with temporal decay. Relative to agent memory systems, it adds deterministic class-specific scoring. Relative to cognitive architectures, it operationalizes decay and activation principles at organizational scale.

3 The System

OIDA consists of three load-bearing components: the Knowledge Object model, the Knowledge Gravity Engine, and the hybrid retrieval architecture. Figure 1 shows the system lifecycle. Full mathematical details and parameter tables are in Appendices A–C.

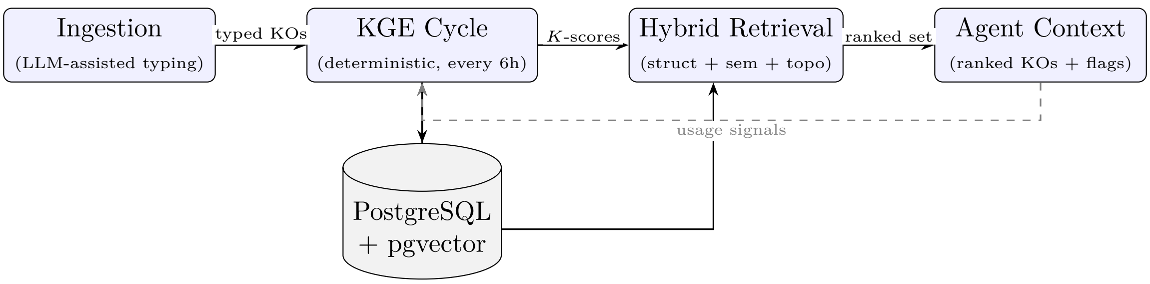

Figure 1: OIDA system lifecycle. Ingestion is LLM-assisted; all subsequent maintenance and retrieval is deterministic. Usage signals from agent queries feed back into the next KGE cycle.

3.1 The Knowledge Object

A **Knowledge Object** is a tuple:

$$\text{KO}_i = (\text{id}_i,\ \text{koc}_i,\ \text{class}_i,\ \text{content}_i,\ \text{scores}_i,\ \text{edges}_i,\ \text{meta}_i) \quad (1)$$

where id_i is a UUID, koc_i is the Knowledge Object Coordinate (§3.1.2), $\text{class}_i \in \mathcal{C}$ is drawn from a closed taxonomy of nine epistemic classes, content_i is the natural language representation, $\text{scores}_i = (K_i, \text{conf}_i, \text{fresh}_i, \text{urg}_i, \text{contr}_i)$ is the five-dimensional score vector, edges_i is the set of typed directed relationships, and meta_i contains provenance, embedding vector, and decay configuration.

3.1.1 Nine Epistemic Classes

The epistemic class captures an organisation's relationship to a proposition—not the content of the proposition, but the strength and type of commitment attached to it. When the epistemic status of a claim changes (e.g., an OBSERVATION accumulates enough supporting evidence to be reclassified as EVIDENCE), the system creates a *new* KO with a REFINES or SUPERSEDES edge to the original. The old KO retains its class and KOC; it is not mutated. This preserves the audit trail and the immutability of the KOC coordinate.

The nine classes are not an arbitrary list. They arise from crossing two orthogonal axes that together determine a KO's computational behaviour.

Axis 1: Epistemic commitment strength. Propositions held by an organisation range from explicit ignorance (a question no one has answered) through uninterpreted signals (observations), provisionally held claims (hypotheses, plans), evidentially supported assessments (evidence, evaluations), persistent contextual anchors (narratives), up to verified and binding commitments (decisions, constraints). This ordering determines seed importance: stronger commitment receives a higher initial K-score.

Axis 2: Temporal behaviour under absence of reinforcement. Not all knowledge should age at the same rate. A DECISION is the organisational analogue of an entrenched belief—removing it requires an explicit SUPERSEDES event, not passive decay. An OBSERVATION is peripheral: if no one retrieves it or reinforces it with new evidence, it should lose weight. A QUESTION is the dual of knowledge: unresolved uncertainty becomes *more* costly as the organisation continues making decisions in its shadow, so its urgency increases rather than decays. This axis determines the decay profile: non-decaying, exponentially decaying, or inversely decaying.

The nine classes are the minimal set that preserves all distinctions required by the KGE's decay model, the retrieval architecture's structural similarity function, and the contradiction handling logic. Merging DECISION and CONSTRAINT would lose the distinction between active choices (revocable by SUPERSEDES) and structural boundaries (revocable only by institutional change). Merging HYPOTHESIS and OBSERVATION would lose the distinction between claims that generate testable predictions and signals that do not.

The nine classes are ordered by epistemic commitment strength and differ in temporal behaviour:

Non-decaying classes (DECISION, CONSTRAINT, NARRATIVE) require explicit supersession events for deactivation. Exponentially decaying classes lose epistemic weight over time unless reinforced by usage or new evidence.

QUESTION as modelled ignorance. QUESTION is the only class with inverse decay: unresolved questions become *more* urgent, not less. Most systems model what is known; OIDA also models what is *not* known, and makes that gap more operationally pressing as time passes. A Q-KO is linked to the knowledge it blocks via BLOCKS edges; its urgency score increases each cycle it remains unresolved, surfacing it with increasing priority in agent context. When a Q-KO is resolved (typically by a DECISION KO linked via IMPLEMENTS), its urgency drops to zero.

The taxonomy draws on three formal traditions—Hintikka's epistemic logic [9] for the knowledge/belief distinction, AGM belief revision [11] for the decay-as-contraction analogy, and Pollock's defeasible

Class	Seed K	Decay Type	Half-life	Epistemic Role
DECISION	1.00	None	∞	Formalized choice—valid until superseded
CONSTRAINT	0.90	None	∞	Non-negotiable structural boundary
EVIDENCE	0.80	Exponential	$\sim$365 d	Verifiable supporting or refuting data
NARRATIVE	0.70	None	∞	Persistent contextual anchor
PLAN	0.65	Exponential	$\sim$69 d	Structured intention with time horizon
EVALUATION	0.55	Exponential	$\sim$198 d	Informed qualitative assessment
OBSERVATION	0.40	Exponential	$\sim$90 d	Weak signal not yet interpreted
HYPOTHESIS	0.30	Exponential	$\sim$50 d	Unverified testable claim
QUESTION	0.30	Inverse	Urgency grows	Open question requiring resolution

Table 2: Epistemic class taxonomy. Seed values and half-lives are working heuristics from operational experience at KVA, not empirically validated optima.

reasoning [10] for the distinction between rebutting and undercutting defeaters. The implementation is a tractable computational subset of these frameworks, not a faithful logical implementation. The connection is motivational, not formal.

3.1.2 The Knowledge Object Coordinate (KOC)

Every KO is assigned a 7-axis structured coordinate: `[Entity]-[Domain]-[Class]-[Epoch]-[Depth]-[Author]-[Variant]`. The KOC is **immutable after assignment**—it is a persistent identifier, not a mutable label. (Class changes are handled by creating a new KO, as described above.)

The KOC serves three purposes: deterministic indexing (same input produces the same coordinate), human legibility (a practitioner reads the epistemic context at a glance), and LLM efficiency (an agent infers entity, domain, class, and epoch from the string at zero additional token cost). Structural similarity between two KOs is computable in $O(1)$ from the KOC alone (see §3.3 for the definition). Full axis specification in Appendix A.

3.1.3 Typed Relationships (KOEdge)

Relationships between KOs are typed from a closed, versioned set of ten edge types, each carrying a signed semantic coefficient (Table 3). All edges are directed: $A \to B$ means "A acts on B." Positive edges propagate importance; negative edges (BLOCKS, CONTRADICTS) actively suppress it through the gravity computation. Coefficients are design priors chosen for interpretability; sensitivity analysis is planned.

Type	Coefficient	Semantics ($A \to B$: A acts on B)
SUPPORTS	+1.0	A provides evidence strengthening B
BASED_ON	+0.8	A is the logical grounding of B
IMPLEMENTS	+0.7	A operationally realizes B
SUPERSEDES	+0.6	A replaces B—B is demoted, not deleted
REFINES	+0.5	A narrows B without contradiction
DERIVES_FROM	+0.5	A follows logically from B
ENABLES	+0.4	A is a necessary condition for B
PRECEDES	+0.3	A temporally precedes B
BLOCKS	−0.4	A actively prevents B (negative gravity)
CONTRADICTS	−0.6	A contradicts B (strongest negative gravity)

Table 3: KOEdge vocabulary with signed semantic coefficients. All edges are directed: $A \to B$ means A acts on B. Coefficients are design priors; sensitivity analysis is planned.

3.1.4 Five Scores per KO

Every KO carries five computed scores. K and Confidence are orthogonal by design—a KO can be highly important but have low confidence, or vice versa.

Score	Range	Computation
K (importance)	$[0, 1]$	Output of the KGE; clamped to $[0, 1]$ after each update
Confidence	$[0, 1]$	Seed by class; increased by inbound SUPPORTS edges; decreased by CONTRADICTS edges. Currently a weighted sum; formula defined but empirical validation ongoing
Freshness	$[0, 1]$	$\exp(-\lambda_{\text{class}} \cdot \text{age})$; decays per class-specific half-life
Urgency	$[0, 1]$	Q-KOs only; increases with age and blocking count (Appendix B)
Controversy	$[0, 1]$	Count of active CONTRADICTS edges, normalized

Table 4: Five scores per Knowledge Object. K and Confidence are orthogonal by design.

3.2 The Knowledge Gravity Engine

The KGE computes an updated importance score K for every active KO at each cycle (configured default: every 6 hours). After the update, K is clamped to $[0, 1]$:

$$K(t+1) = \text{clamp}\Big((1-\eta) \cdot K(t) + \eta \cdot [\text{seed} + u + e + g] - \lambda_{\text{class}} \cdot \Delta t \cdot K(t) - c, \quad 0, \quad 1\Big) \tag{2}$$

The equation decomposes into three forces:

Momentum $(1-\eta) \cdot K(t)$: carries forward current importance.

Injection $\eta \cdot [\text{seed} + u + e + g]$: new signals. Seed is the class baseline. Usage force u is retrieval-driven activation using an exponential recency kernel inspired by ACT-R's power law of forgetting [7, 8]—we use an exponential approximation for computational convenience, not as a theoretical improvement. Evidence force e counts new inbound SUPPORTS edges. Gravity force g propagates importance through signed edges from connected KOs (both positive and negative; see Appendix B for the formula).

Negative forces $-\lambda_{\text{class}} \cdot \Delta t \cdot K(t) - c$: class-specific decay and contradiction penalty.

Under stationary input signals (before clamping), the per-node update rule converges to a unique fixed point:

$$K^* = \frac{\eta \cdot [\text{seed} + u + e + g] - c}{\eta + \lambda_{\text{class}} \cdot \Delta t} \tag{3}$$

This is a stability property of the scalar update rule, not a convergence proof for the full coupled system—which remains an open formal question (§6). Full derivation and parameter tables in Appendix B.

Memory zones. KOs are classified by current K-score into four disjoint zones:

Core Memory	$K \geq 0.40$	Always injected into agent context	Budget: 100 nodes
Working Memory	$0.10 \leq K < 0.40$	Retrieved when query-relevant	Budget: 500 nodes
Peripheral	$0.05 \leq K < 0.10$	Targeted queries only	Unlimited
Dormant	$K < 0.05$	Excluded from gravity computation	—

No KO is ever deleted from the historical record—only excluded from active computation.

Relationship to cognitive activation models. The KGE's design inherits specific structure from ACT-R's base-level activation [7, 8] but makes three deliberate departures. First, the usage force uses an exponential recency kernel rather than ACT-R's power-law sum $B_i = \ln(\sum t_j^{-d})$. The exponential kernel is computationally cheaper at cycle time, and for the temporal scales relevant to organizational knowledge (hours to months rather than seconds to hours), the difference between power-law and exponential decay is empirically small. This is a convenience approximation, not a theoretical improvement. Second, ACT-R uses a single global decay parameter d for all memory chunks. The KGE replaces this with λ_{class}, a per-class decay rate: a DECISION and an OBSERVATION should not decay at the same rate even if they have identical retrieval histories, because they carry different epistemic commitments. Third, ACT-R's spreading activation propagates only positive activation through associative links. The KGE propagates importance through *signed* edges, where CONTRADICTS and BLOCKS edges generate negative gravity—actively suppressing the importance of contradicted knowledge. This extends the cognitive metaphor into territory that ACT-R does not model: the computational representation of epistemic conflict.

3.3 Hybrid Retrieval

In standard RAG architectures, all retrieved chunks are treated as epistemically equivalent. In OIDA, every retrieved Knowledge Object carries a computable epistemic weight.

The hybrid score combines three independent similarity layers:

$$H(q, i) = \alpha \cdot S_{\text{struct}}(q, i) + \beta \cdot S_{\text{sem}}(q, i) + \gamma \cdot S_{\text{topo}}(q, i) \tag{4}$$

Structural similarity S_{struct}: computed from KOC axis alignment. Each of the 7 axes contributes a binary match (1 if equal, 0 otherwise); the score is the weighted sum normalized to $[0, 1]$. For a free-text query q, entity and domain hints are extracted via the LLM at query time (if identifiable) or omitted (contributing 0 to those axes). Cost: $O(1)$, no database access.

Semantic similarity S_{sem}: cosine similarity over embedding vectors, rescaled to $[0, 1]$.

Topological similarity S_{topo}: inverse hop distance in the epistemic graph. For a free-text query, the system first identifies the top-k semantically similar KOs, then uses their graph neighbourhood to compute topological relevance of connected nodes.

Final ranking. At retrieval time, the hybrid similarity is multiplied by the contextual importance:

$$R(q, i) = H(q, i) \cdot K_{\text{eff}}(i, q) \tag{5}$$

where $K_{\text{eff}}(i, q) = K_{\text{global}}(i) \cdot \max(0.10,\ \text{attention}(i, q))$. The floor of 0.10 prevents complete collapse of globally important KOs.

Configured defaults: $\alpha = 0.30$, $\beta = 0.50$, $\gamma = 0.20$. An ablation study comparing retrieval quality across configurations is a planned next step. Full scoring formulas in Appendix C.

Why hand-crafted scoring rather than learned weights? A natural alternative would be to learn the retrieval weights and decay parameters from data. We chose hand-crafted, interpretable parameters for three reasons: (1) the corpus is too small (500 KOs) for reliable learning; (2) determinism and auditability are design requirements—learned parameters can drift silently; (3) at this stage, understanding why the system behaves as it does is more valuable than marginal performance gains.

3.4 Implementation Status

The storage layer is implemented in PostgreSQL with pgvector, providing persistence for Knowledge Objects, typed edges, score snapshots, and temporal audit trails.

Component	Status	Notes
KO schema, KOC, KOEdge vocabulary	**Production**	Core epistemic schema in use at KVA
Knowledge Gravity Engine (KGE)	**Production beta**	Running on live corpus; parameters are working heuristics
Hybrid retrieval (struct + sem + topo)	**Production**	Deployed; ablation study forthcoming
Epistemic Confidence Score	**Production beta**	Formula defined; empirical validation ongoing
Q-KO urgency model	**Production beta**	Deployed; threshold values are configured defaults

Table 5: Implementation status of OIDA components as of March 2026.

4 End-to-End Example

This section makes the abstractions above concrete through a single case from KVA's operations.

Context. KVA is evaluating whether to continue investing operational resources in B2B SaaS for legal compliance. Three KOs exist:

- `KVA-STR-H-2026M01-L0-FED`: HYPOTHESIS—"B2B SaaS legal compliance is a high-growth segment in Southern Europe." $K = 0.31$, confidence $= 0.42$.
- `KVA-STR-V-2026M02-L1-NAB`: EVALUATION—"Three European law firms expressed strong interest in AI-assisted compliance workflow." $K = 0.48$, confidence $= 0.67$. (SUPPORTS $\rightarrow$ the hypothesis.)
- `KVA-STR-O-2026M02-L0-DAV`: OBSERVATION—"Two direct competitors raised Series A rounds targeting the same segment." $K = 0.29$, confidence $= 0.55$.

Step 1: New evidence arrives. A team member logs an observation: "Spoke with a potential customer; they reported that their internal legal team has blocked AI tool adoption for regulatory reasons." The ingestion layer classifies it as an OBSERVATION and creates a CONTRADICTS edge to the hypothesis:

- `KVA-STR-O-2026M03-L1-DAV`: OBSERVATION—regulatory blocking is active. Seed $K = 0.40$, confidence $= 0.50$.

Step 2: KGE cycle runs. At the next cycle, the hypothesis now has one active CONTRADICTS inbound edge. Its contradiction penalty c increases; its gravity force g, previously boosted by the SUPPORTS edge from the evaluation, is partially offset. The hypothesis K drops from 0.31 toward approximately 0.22.

Step 3: A question is created. A team member creates a QUESTION KO: "Given regulatory blocking, should the legal compliance investment thesis be revised?" It is linked to the hypothesis and evaluation via BLOCKS edges. Its urgency is currently low but will increase each cycle it remains unresolved.

Step 4: An agent queries. A team member asks: "What is our current position on the legal compliance segment?" The retrieval layer returns ranked KOs with epistemic metadata:

- The evaluation ($K = 0.48$)—Core Memory, always present
- The contradicting observation ($K = 0.38$)—Working Memory, included by semantic match
- The hypothesis ($K = 0.22$, below Core threshold)—Working Memory, included by structural match
- The blocking question—flagged as an active unresolved issue with increasing urgency

The agent's response reflects: (a) positive initial evidence exists, (b) it is contested by a direct contradiction, and (c) an unresolved question is blocking downstream planning.

Step 5: A decision is recorded. After discussion, a Decision KO is created: "Pause investment in B2B legal compliance pending further regulatory research." It is connected via SUPERSEDES to the hypothesis and via IMPLEMENTS to the question (resolving it). The question's urgency drops to zero. The decision's K starts at seed 1.00.

5 Preliminary Observations

OIDA is deployed as the operational knowledge infrastructure of KVA's venture studio. This section reports internal observations from that deployment. These are not a validation study.

Metric	**Value**
Total KOs	∼500 (5 ventures, 3 client engagements, internal strategy)
Observation window	4 weeks of KGE cycles
Source tools	Notion, Google Calendar, Slack (structured migration)
Evaluation queries	50, drawn from real team requests over 3 weeks
Comparison baseline	Vector-similarity-only retrieval

Table 6: Deployment summary.

K-score distribution. 10–15% of KOs settled in Core Memory ($K \geq 0.40$). The remaining distribution is concentrated in Working Memory with a long tail of Peripheral KOs. This is consistent with an expectation that a minority of organizational knowledge is operationally central at any given time—but this is an observation about our corpus, not a validated general finding.

Retrieval quality. The hybrid system produced noticeably more relevant results for causal queries ("why was this decided?") and relational queries ("what supports this claim?"), where structural and topological similarity contribute most. For simple factual lookups, performance was comparable to the baseline. These are qualitative team assessments, not controlled experiments.

Decay behaviour. Configured half-lives produced intuitively correct ageing for most classes. One exception: the 90-day half-life for OBSERVATION appears too long for rapidly evolving market signals in AI-adjacent domains. A domain-level decay override is under consideration.

Open calibration questions. The KGE has nine free parameters in the core update rule alone, plus additional parameters in the retrieval weighting and Q-KO urgency formula. All are currently set to design priors. Systematic calibration against retrieval quality metrics is the next major technical milestone.

6 Limitations and Honest Assessment

6.1 What the System Guarantees and What It Does Not

6.2 What Is Surprisingly Tractable

One lesson is that *computational epistemic maintenance* is more tractable than expected. Deterministic decay, automatic importance scoring, and explicit contradiction propagation—applied uniformly across a corpus—produce a knowledge substrate that is more informative than the unstructured alternative, without requiring human epistemic hygiene.

A second surprisingly tractable layer is the structural similarity computation from the KOC. A seven-axis coordinate computed once at ingestion provides an $O(1)$ similarity signal that complements semantic similarity at negligible cost.

The system can guarantee	The system cannot guarantee
Deterministic maintenance: same class, same parameters, same temporal behaviour every time	Universal parameter validity: configured heuristics may underperform in organizations with different knowledge patterns
Typed epistemic structure: every KO carries class, confidence, scores, and edges	Perfect ingestion classification: typing quality depends on the LLM-assisted classification pipeline
Explicit contradiction surfacing: negative-coefficient edges create negative gravity and are computationally visible	Complete contradiction detection: the system models contradictions that are explicitly created, not those that exist implicitly in text
Stable retrieval contract: the API does not change when the foundation model is replaced	Optimal retrieval quality: the hybrid scoring weights are design priors, not empirically optimized
Per-node fixed-point convergence under stationary input signals	Full system convergence under dynamic graph conditions with coupled K-scores
Immutable audit trail: no KO is deleted, all state changes are logged	Calibrated absolute scores: K-values are relative rankings, not calibrated probability estimates

Table 7: Guarantees and limitations.

6.3 What Remains Hard

Cold start. A new deployment begins without K-score history, usage signals, or gravity calibration. Initial importance is determined by class seeds and static structure.

Parameter sensitivity. The engine has many configurable parameters whose interactions are not fully characterized. Sensitivity analysis is planned.

Full system convergence. The per-node fixed-point characterisation establishes scalar stability. The full system—dynamic graphs, coupled K-scores, non-linear saturations—has not been formally proven to converge.

Taxonomy completeness. The nine-class taxonomy is operationally motivated but not proven minimal or complete. Whether this decomposition captures the relevant distinctions across diverse organizational types is an empirical question.

Ingestion classification quality. The entire downstream system amplifies classification errors. This is the highest-leverage engineering investment.

What the framework cannot represent. The closed vocabulary of nine classes and ten edge types deliberately sacrifices expressiveness for computational tractability. Knowledge that resists classification into any of the nine types—tacit procedural knowledge, emotional or cultural context, ambiguous multi-class content—is either forced into the closest class (introducing classification noise) or excluded from the epistemic layer entirely. Similarly, relationships that fall outside the ten-type vocabulary are not modellable. We do not claim the framework captures all organizational knowledge; we claim it captures the subset that is amenable to typed, computed epistemic maintenance.

6.4 Planned Evaluation

A structured evaluation protocol is being designed to measure three capabilities against baseline RAG systems: (1) contradiction detection rate, (2) epistemic ranking accuracy versus expert-assessed relevance, and (3) decision traceability—the ability to reconstruct the full evidential chain behind a decision.

7 Conclusion

The current implementation and single-site deployment support three provisional claims. First, the architecture is deployed and operational. Second, computational epistemic maintenance—typing, decay, contradiction propagation—is tractable and produces a more informative retrieval substrate than unstructured alternatives. Third, design requirements for organizational epistemic infrastructure can be articulated from building experience, and these requirements are non-obvious: that contradiction handling matters more than agreement surfacing, that class-specific decay solves problems retrieval improvements cannot, and that manual epistemic hygiene does not scale.

Three things are not established. Whether the configured parameters generalize across organizational types. Whether the hybrid retrieval produces measurable improvements over strong baselines in controlled experiments. Whether the full coupled system converges under dynamic conditions. These remain open questions.

A forthcoming companion report will present a controlled empirical evaluation of OIDA against current state-of-the-art retrieval and knowledge management systems—including structured RAG baselines and agent memory architectures—on a representative organisational corpus, with quantitative metrics for contradiction detection, epistemic ranking accuracy, and decision traceability.

The contribution is infrastructural, not theoretical. We have built a system that models epistemic state computationally and reported what we learned from building it. The stronger empirical claims require the evaluation program outlined in §6. OIDA should be understood as a computational hypothesis under active validation.

References

[1] Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. *Scientific American*, 284(5), 34–43.

[2] Palantir Technologies. (2023). *Palantir Ontology: Connecting Data to the Real World*. Platform Documentation.

[3] Packer, C., Wooders, S., Lin, K., et al. (2023). MemGPT: Towards LLMs as Operating Systems. *arXiv:2310.08560*.

[4] Zep AI. (2025). *Zep: A Long-Term Memory Service for AI Assistants*. Technical documentation.

[5] Lewis, P., Perez, E., Piktus, A., et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *NeurIPS*, 33.

[6] Edge, D., Trinh, H., Cheng, N., et al. (2024). From Local to Global: A Graph RAG Approach to Query-Focused Summarization. *arXiv:2404.16130*.

[7] Anderson, J.R., Bothell, D., Byrne, M.D., et al. (2004). An Integrated Theory of the Mind. *Psychological Review*, 111(4), 1036–1060.

[8] Anderson, J.R. & Schooler, L.J. (1991). Reflections of the Environment in Memory. *Psychological Science*, 2(6), 396–408.

[9] Hintikka, J. (1962). *Knowledge and Belief: An Introduction to the Logic of the Two Notions*. Cornell University Press.

[10] Pollock, J.L. (1987). Defeasible Reasoning. *Cognitive Science*, 11(4), 481–518.

[11] Alchourrón, C., Gärdenfors, P., & Makinson, D. (1985). On the Logic of Theory Change. *Journal of Symbolic Logic*, 50(2), 510–530.

[12] Nonaka, I. (1994). A Dynamic Theory of Organizational Knowledge Creation. *Organization Science*, 5(1), 14–37.

A Full Knowledge Object Model Reference

A.1 KOC Axis Specification

Format: `[A]-[B]-[C]-[D]-[E]-[F]-[G]`

Example: `HOM-INV-D-2025Q4-L1-FED`—a Decision (C=D) about investment (B=INV) for Homingo (A=HOM), made in Q4 2025, at shallow relational depth (E=L1), authored by Federico (F=FED).

Axis	Name	Values
A	Entity	3-letter code (e.g., HOM = Homingo, KVA = KVA internal)
B	Domain	INV, OPS, MKT, LEG, FIN, HRD, TCH, STR
C	KO Class	D, C, V, N, P, E, O, H, Q
D	Temporal Epoch	2025Q4, 2026M03, W12-2026
E	Relational Depth	L0 (standalone) $\rightarrow$ L3 (deeply embedded)
F	Author/Source	3-character identifier
G	Variant Flag	`"` (stable), `d` (draft), `r` (revised)

Table 8: KOC axis specification. All axes are immutable after assignment. Class changes create a new KO linked via REFINES or SUPERSEDES.

B KGE Mathematical Details

B.1 The Update Rule

$$K(t+1) = \mathrm{clamp}\Big((1-\eta)\cdot K(t) + \eta\cdot[\mathrm{seed} + u + e + g] - \lambda_{\mathrm{class}}\cdot\Delta t\cdot K(t) - c,\ \ 0,\ \ 1\Big) \tag{6}$$

B.2 Input Signals

Seed: class-specific baseline importance assigned at KO creation (Table 2).

Usage force u (scaled by a_u):

$$u = a_u \cdot \tanh\left(\frac{\sum \exp(-\lambda_{\mathrm{usage}}\cdot \mathrm{age\_hours}_i)}{r_{\mathrm{scale}}}\right) \tag{7}$$

Evidence force e (scaled by a_e; sliding window, default 14 days):

$$e = a_e \cdot \tanh\left(\frac{\mathrm{new\_SUPPORTS\_edges}}{f_{\mathrm{scale}}}\right) \tag{8}$$

Gravity force g (propagated through all signed edges, both positive and negative):

$$g = \tanh\left(\frac{\sum_j \mathrm{COEFF}(e_{ij})/d_{ij}^2 \cdot w_{ij}\cdot K_j^{\mathrm{norm}}}{g_{\mathrm{scale}}}\right) \tag{9}$$

Contradiction penalty c (scaled by a_c):

$$c = a_c \cdot \tanh(\mathrm{contradiction\_count}) \tag{10}$$

Note: the scaling constants a_u, a_e, a_c in the KGE are distinct from the retrieval weights α, β, γ defined in §3.3.

B.3 Fixed-Point Characterisation

For fixed input signals (before clamping), the update rule is a linear map in K with slope $(1 - \eta - \lambda_{\text{class}} \cdot \Delta t)$. Under the configured parameter regime—$\eta = 0.15$, $\lambda_{\text{class}} \in \{0, \ldots, 0.02\}$, $\Delta t = 0.25$—the slope lies in [0.845, 0.85] for all classes. Since the absolute value is strictly less than 1, the iteration converges to the fixed point in Eq. 3.

Scope. This establishes per-node scalar stability under stationary inputs, not full system convergence.

B.4 Memory Zones

Zone	Range	Budget	Retrieval Behaviour
Core Memory	$K \geq 0.40$	100 nodes	Always injected into agent context
Working Memory	$0.10 \leq K < 0.40$	500 nodes	Retrieved when query-relevant
Peripheral	$0.05 \leq K < 0.10$	Unlimited	Targeted queries only
Dormant	$K < 0.05$	—	Excluded from gravity computation

Table 9: Memory zone thresholds (disjoint) and retrieval behaviour.

B.5 Q-KO Urgency

$$\text{urgency} = \text{clamp}\left(\frac{\text{age\_days}}{30} \cdot 0.3 + \text{blocking\_count} \cdot 0.2 + \text{strategic\_weight} \cdot 0.5,\ 0,\ 1\right) \tag{11}$$

C Retrieval Architecture Details

C.1 Hybrid Score

$$H(q, i) = \alpha \cdot S_{\text{struct}}(q, i) + \beta \cdot S_{\text{sem}}(q, i) + \gamma \cdot S_{\text{topo}}(q, i) \tag{12}$$

Configured defaults: $\alpha = 0.30$, $\beta = 0.50$, $\gamma = 0.20$.

Structural similarity S_{struct}: weighted binary match over KOC axes. Entity and Domain matches receive weight 0.2 each; Class receives 0.3; Epoch receives 0.15; Depth, Author, Variant receive 0.05 each. Normalised to $[0, 1]$. For free-text queries, entity and domain are extracted at query time when identifiable; unidentifiable axes contribute 0.

Semantic similarity S_{sem}: cosine similarity over embedding vectors, rescaled to $[0, 1]$.

Topological similarity S_{topo}: inverse hop distance in the epistemic graph. For free-text queries, computed from the graph neighbourhood of the top-k semantically matched KOs.

C.2 Final Ranking

$$R(q, i) = H(q, i) \cdot K_{\text{eff}}(i, q) \tag{13}$$

where:

$$K_{\text{eff}}(i, q) = K_{\text{global}}(i) \cdot \max(0.10,\ \text{attention}(i, q)) \tag{14}$$

The floor of 0.10 prevents complete collapse: a globally important KO retains minimum weight even when immediate topical relevance is low.

C.3 Retrieval Contract

The retrieval layer exposes a stable API. The contract specifies input format, output format, and ranking determinism for identical inputs. When the foundation model is replaced, the retrieval contract does not change.