Governed Intelligence Architecture for Institutional AI

May 2026

Witold Reichhart and Arnaud Gelas

Five-Paper Programme: Enterprise AI and Organizational Intelligence

This paper is part of a five-paper programme examining why enterprise AI fails in regulated environments, what architecture resolves it, and what emerges when that architecture operates at depth.

Paper A — Dynamics Blindness (Reichhart and Gelas) Diagnosis. Names the architectural failure mechanism: LLMs process tokens without tracing causal chains through organizational dependencies. Chain-of-thought, RAG, tool use, multi-agent systems, and scaling do not add the missing causal infrastructure.

Paper B — The Predictive Organization (Gelas and Reichhart) Architecture. Specifies the resolution: a tripartite structure — Map (state), Physics (dynamics), Player (agency) — coupling neural perception with symbolic reasoning, operating on claims-based knowledge with prevalence weighting.

Paper C — Build the Medium (Reichhart and Gelas) Theory. Ten independent theoretical traditions converge on what organizational intelligence requires. Introduces the capability/fertility distinction and the autonomy-to-initiative transition as the real measure of AI maturity.

Paper D — Governed Intelligence Architecture (Gelas and Reichhart) ← *this paper* Methodology. The practitioner companion: five-stage Governed Intelligence Lifecycle (Ingest, Consolidate, Curate, Expand, Apply) with governance requirements, epistemic immunity, and the path dependency argument for knowledge infrastructure investment.

Paper E — From Autonomy to Initiative (Reichhart and Gelas) Capstone. Three conditions for governed initiative. Graduated immersion systems as institutional analogue. Governance relocation mechanism. Six computational enrichments. Active inference as normative model. The domain graph as missing middle layer.

Causal spine: Enterprise AI fails because of dynamics blindness (A) → the resolution is architectural (B) → the architecture works because ten traditions converge on what living systems require (C) → the practitioner methodology is a governed intelligence lifecycle with epistemic immunity (D) → when the architecture runs at sufficient depth, it produces governed initiative — agents that perceive what matters through immersion, not instruction (E).

Abstract

AI adoption has outpaced the epistemic infrastructure required to make institutional AI reliable. In regulated organizations, AI systems do not act on knowledge in the abstract — they act on claims whose provenance, scope, freshness, coherence, and authority determine whether outputs are usable, auditable, and safe. This paper introduces epistemic immunity: a governed defense architecture for protecting institutional knowledge substrates from six systemic failure modes — pollution, staleness, fragmentation, amnesia, cascade failure, and structural distortion.

The paper distinguishes knowledge from intelligence. Knowledge is validated, contextual, and relational understanding distributed across people, systems, records, and practices. Intelligence is the governed capacity to mobilize that knowledge into reli-

able judgment and action under changing conditions. Governed intelligence therefore requires more than AI capability: it requires knowledge infrastructure that can admit, validate, scope, demote, preserve, and renew claims over time.

Epistemic immunity is operationalized through the Governed Intelligence Lifecycle — Ingest, Consolidate, Curate, Expand, and Apply — and specified through a three-scope capability architecture covering claim-level controls, graph-level integrity, and delivery-level access, explanation, and agent action. The framework draws on distributed cognition, sensemaking, autopoiesis, dissipative structures, and the adjacent possible to argue that institutional knowledge is not a static asset but a continuously regenerated socio-technical system.

Applied to financial services, the paper shows how regulated institutions can move beyond retrieval-oriented AI toward governed intelligence systems that preserve provenance, manage decay, surface contradictions, constrain agent action, and compound learning without accumulating epistemic debt. The paper develops a conceptual architecture grounded in distributed cognition and knowledge governance traditions, operationalized as an implementation methodology, and applied to the regulatory and operational context of financial services.

This is Paper D of a five-paper programme: Paper A (Reichhart and Gelas, 2026a) diagnoses dynamics blindness as the failure mechanism; Paper B (Gelas and Reichhart, 2026b) specifies the architectural resolution — a tripartite structure of state, dynamics, and agency layers; Paper C (Reichhart and Gelas, 2026c) provides theoretical foundations from ten independent traditions; this paper operationalizes the methodology; Paper E (Reichhart and Gelas, 2026e) identifies the conditions under which the architecture produces governed initiative rather than mere autonomy.

Keywords: AI governance, epistemic immunity, knowledge graphs, institutional intelligence, epistemic operational risk, path dependency, neuro-symbolic reasoning, regulated industries

---

1. Introduction: The Problem and the Architecture

Consider a composite case from settlement operations. A global bank loses its head of settlement operations after 27 years. She knows things no document captures: why the system breaks at quarter-end (a manual reconciliation procedure introduced in 2009 that nobody officially acknowledges), which client configurations fail under peak load, what the regulator actually prioritizes versus what the written guid-

ance says. She spends her final three months documenting what she knows. The documents are thorough, 200 pages, careful. They go into SharePoint. Six months later, nobody can find them. A year later, nobody knows they exist. A junior consultant onboards, configures the system correctly by every documented standard. At quarter-end, it breaks.

This is a knowledge architecture failure, and it happens thousands of times daily across regulated industries. It is not a technology problem. The institution had SharePoint, documentation standards, a knowledge management initiative, and eventually Confluence and an AI-powered search layer. Each improved the search; none changed what the search found. A better search on a graveyard is still a graveyard.

Recent McKinsey research frames the urgency: AI adoption is widespread — the majority of organizations now use AI in at least one business function — yet enterprise-level financial impact remains limited, with only a small minority reporting significant bottom-line gains. The diagnosis is structural. These organizations are not behind on capability. Many are at the frontier of technical implementation — sophisticated model selection, production-grade MLOps, well-designed agent architectures. They are ahead of their knowledge foundation. They have built powerful tools with nothing deep to run on.

Agentic AI turns this from a productivity problem into a control problem. Before agents, bad knowledge meant inefficiency — a practitioner wasted time, asked a colleague, found the right answer eventually. With agents, bad knowledge means automated action on defective institutional memory. The same incomplete domain model that a human practitioner might partially compensate for through experience is now operating through fifty AI-assisted delivery workflows simultaneously. Once AI agents act on institutional knowledge, knowledge quality becomes an operational control surface.

This paper introduces a new risk category for regulated industries: **epistemic operational risk** — the risk of institutional harm caused by acting on knowledge that is stale, polluted, fragmented, decontextualized, mis-scoped, or structurally distorted. Existing governance frameworks manage models, data, privacy, security, and compliance, but they do not manage the integrity of the institutional knowledge substrate on which AI systems act.

Definitions

This paper uses the following terms with specific meaning.

Knowledge is validated, contextual, and relational understanding distributed across people, systems, records, and practices. **Intelligence** is the governed capacity of an institution to mobilize that knowledge into reliable judgment and action under changing conditions. This usage draws on the intelligence tradition associated with Sherman Kent, in which intelligence is not raw information but processed, validated knowledge made usable for decision. The present paper extends that tradition from national-security analysis to institutional AI: intelligence is treated not merely as an analytical product but as an institutional capacity sustained by governed knowledge infrastructure.

AI capability is the technical capacity of models and tools to perform tasks. AI capability is not equivalent to intelligence: AI provides execution and inference capacity, but governed intelligence depends on the integrity, currency, coherence, and usability of the knowledge substrate on which AI acts.

Governed means that knowledge claims and their use are subject to explicit rules of provenance, authority, confidence, scope, access, validation, decay, and accountability. Governance is not a policy layer applied after the fact — it is an architectural property of the system.

Governed intelligence is what emerges when AI capability operates over knowledge that is current, contextual, connected, validated, scoped, auditable, and capable of learning from use.

Epistemic debt is the accumulated liability created when organizations act on knowledge whose provenance, freshness, scope, coherence, or validity has degraded. Like technical debt, epistemic debt compounds silently and becomes expensive to service once structural.

Contributions

The paper's central contribution is the **epistemic immunity framework**, which defines how governed intelligence systems prevent, detect, and recover from six systemic knowledge failures. The framework is operationalized through the **Governed Intelligence Lifecycle** — a five-stage implementation methodology — and specified technically through a **three-scope capability architecture**. The theoretical foundations for these contributions are established in the companion papers of this programme (Papers A, B, and C). This paper provides the operational implementation methodology and applies the framework to the regulatory and operational context of financial services, drawing on public regulatory guidance, knowledge graph standards, and observed implementation patterns.

The framework does not claim that provenance, temporal validity, lineage, contradiction management, ontology governance, or access control are individually new. Its contribution is to organize these mechanisms as an integrated defense architecture for institutional AI systems that reason and act over organizational knowledge.

This is not a document management system, not a search layer, not a generic knowledge graph, not an MLOps framework, and not a replacement for human expertise. It is a governed epistemic infrastructure that determines what institutional AI systems may know, how confidently they may know it, and what they may do with it. The goal is not a totalizing single source of truth. It is a system that is safely incomplete rather than falsely complete. RAG retrieves content for generation. Governed intelligence infrastructure controls the validity, scope, provenance, decay, contradiction status, and actionability of the claims on which generation or agentic action depends.

2. Epistemic Immunity: Engineering the Defenses

A governed intelligence system is exposed to six distinct failure modes. This section specifies each threat and its engineered defense.

2.1 Failure Mode One: Knowledge Pollution

The threat. Bad claims entering the system. A fabricated source, an LLM hallucination ingested as fact, a misattributed regulatory citation, a confidently wrong operational claim from a junior practitioner. Decay monitoring does not catch pollution — the claim is not stale; it was never true.

The defense. Ingestion-boundary control. Nothing enters the graph without verifiable provenance — a traceable chain from claim to source. Source reliability scoring establishes tiers: regulatory publications carry inherent authority; internal documents carry medium authority; AI-extracted claims carry provisional status requiring validation. Confidence floors prevent low-provenance claims from entering active use. The immune system guards the perimeter.

A specific AI-era pollution vector requires separate defense: **synthetic knowledge contamination**. AI agents do not only consume the graph; their outputs may re-enter it. An LLM hallucinates. The output is saved to a document. The ingestion pipeline extracts it. The graph treats it as a claim. A future agent cites it. The hallucination becomes institutional memory. Defense: AI-generated content must be labeled at source; outputs cannot be promoted without independent validation; provenance tracks synthetic origin; and critically, **corroboration requires independent origin, not repeated occurrence** — multiple copies of the same LLM-generated falsehood across different documents do not constitute corroboration.

The lifecycle mapping. The defense operates primarily at Ingest and Consolidate, with Quality Assessment in Curate providing a secondary check.

2.2 Failure Mode Two: Knowledge Staleness

The threat. Claims that were true but no longer are. A regulatory requirement that has been superseded, an operational procedure that has changed, a technology constraint that has been removed. This is the failure mode that every knowledge management system eventually succumbs to — and the one that most directly generates epistemic debt. Maintenance and decay have been treated as afterthoughts in every major knowledge management approach since 1991. This architecture treats them as first-order design concerns.

The defense. Decay tracking with type-specific schedules. Regulatory claims carry long decay windows (annual revalidation). Operational claims carry short windows (quarterly or monthly). External intelligence carries the shortest (monthly or on-event). When a claim approaches its decay window, the system triggers revalidation. Auto-revalidation is defensible when the source signal is clear — if the source regulation hasn't been amended, the clock resets automatically. Human review is triggered when signals are ambiguous.

The lifecycle mapping. The defense operates within Curate, with Scouting in the Expand stage providing external change detection that triggers early revalidation.

2.3 Failure Mode Three: Knowledge Fragmentation

The threat. The graph losing coherence over time. Entity resolution handles initial coherence, but six months later a new team adds nodes using different vocabulary, a system migration introduces duplicate entities that bypass original resolution, or ontological conventions drift as usage patterns change. The graph develops islands — clusters of nodes that should be connected but aren't, because the connecting concepts are represented by different terms.

The defense. Ongoing ontology drift detection. The system monitors for new terms that semantically overlap existing entities, for node clusters with unexpectedly low cross-connectivity, and for vocabulary divergence between teams or domains. When drift is detected, the system triggers re-resolution — surfacing candidate merges for domain expert review.

The lifecycle mapping. The defense is an ongoing Consolidate function that operates continuously, not just during initial ingestion.

2.4 Failure Mode Four: Knowledge Amnesia

The threat. The system forgetting what it used to know. Claims superseded without the supersession chain being tracked. Edges deleted without recording why. Decisions made based on knowledge that is no longer visible in the graph — making it impossible to reconstruct the reasoning that led to a past action.

The defense. The governed disposition principle. Claims are demoted, not deleted — unless governed disposition requires otherwise. Supersession chains are maintained — every claim that replaces another carries a reference to what it replaced and why. Bitemporal tracking maintains two independent time dimensions: valid-time (when the claim was true in the world) and transaction-time (when the system recorded the claim). This enables precise reconstruction of past knowledge states — critical for regulatory inquiries that routinely ask "what did you know at time T?"

Event-driven architecture provides the implementation substrate. Every change — claim added, confidence updated, edge created, claim superseded — is recorded as an immutable event. The event log can be replayed to reconstruct any historical state.

Historical reconstructability must be balanced against legal retention, privacy, privilege, and client-confidentiality obligations. Epistemic memory is therefore governed memory, not indiscriminate permanence. The system supports retain, demote, archive, seal, anonymize, and dispose — each as a governed action with audit trail.

The lifecycle mapping. The defense is architecturally embedded — event sourcing and bitemporal tracking operate across all stages.

2.5 Failure Mode Five: Cascade Failure

The threat. A bad claim propagating through the graph via dependency edges. If one node is incorrect and ten others depend on it through "requires" or "supports" edges, the damage is not contained — it spreads through the reasoning chain. Standard maintenance does not catch this because each downstream node appears healthy individually. Only when you trace the dependency chain do you discover that the foundation is compromised.

The defense. Dependency-aware confidence propagation. When a source node's confidence drops — through contradiction, failed revalidation, or detected pollution — the system traces forward through dependency edges and flags everything built on it. Downstream claims don't automatically lose confidence, but they are tagged for review: "This claim's confidence depends on [compromised claim]. Verify independently." The immune system contains contagion before it becomes systemic.

The lifecycle mapping. The defense operates within Curate, triggered by confidence changes at any node in the graph.

2.6 Failure Mode Six: Structural Distortion

The threat. Certain nodes accumulate disproportionate connections — a concept like "Payment" links to hundreds of other nodes because everything in a financial services domain touches payment in some way. These hub nodes create three pathologies. First, they distort traversal: every query path goes through the hub, so unrelated queries return the same node, polluting subgraph extraction. Second, they create false proximity: two genuinely unrelated concepts appear connected because they both link to the same hub, suggesting relationships that do not exist. Third, they become single points of failure: if the hub node's confidence drops, the cascade propagates through an outsized portion of the graph.

This is a hub-dominance problem: high-centrality nodes distort traversal, inflate apparent proximity, and concentrate dependency risk.

The defense. Topology monitoring through centrality analysis. The system continuously computes centrality metrics — betweenness (how many shortest paths pass through a node), degree (how many connections it has), closeness (how easily it can reach all other nodes). When a node crosses a centrality threshold, the system flags it for decomposition: splitting "Payment" into more specific nodes — "Payment Authorization," "Payment Processing," "Payment Settlement," "Payment Reconciliation" — that distribute the connections more evenly. Weighted traversal algorithms can also dampen hub influence during subgraph extraction, ensuring that high-centrality nodes do not dominate query results.

The lifecycle mapping. The defense operates within Curate as a graph health function, with decomposition recommendations reviewed by the Semantic Authority (the governance role responsible for ontological decisions).

2.7 The Immune System as a Whole

These six defenses are not independent features bolted onto the graph. They form a coherent system — each defense operating continuously, each addressing a different threat vector, all maintaining the same property: the integrity of the knowledge the system claims to hold.

The biological analogy is useful but bounded. A biological immune system has both innate immunity (general defenses that operate against any threat — the perimeter) and adaptive immunity (specific defenses that learn from past encounters and respond faster next time). Epistemic immunity has a similar structure. Provenance requirements, confidence floors, and decay schedules are innate — they operate generically against all threats. Resolution patterns, decomposition strategies, and cascade containment rules are adaptive — they learn from the specific threats the system has encountered and improve their response over time.

Like biological immunity, epistemic immunity must balance defense against openness. A system that rejects all low-confidence novelty becomes sterile; a system that accepts all novelty becomes polluted. The governance problem is therefore not maximum exclusion but calibrated admission, monitoring, and escalation.

Existing knowledge graph platforms and governance frameworks address several of these concerns individually — such as lineage, access control, ontology management, or data quality — but they do not typically organize them as a unified defense architecture against systemic epistemic degradation. The contribution of epistemic immunity is therefore integrative and operational: it treats degradation as a predictable system condition rather than as a set of isolated maintenance tasks.

2.8 Meta-Risk: Epistemic Autoimmunity

A sophisticated reader will immediately ask: what happens when the immune system attacks valid novelty?

Epistemic autoimmunity occurs when the defense architecture suppresses valid but weakly institutionalized knowledge because it conflicts with authoritative sources, established ontology, or existing confidence hierarchies. Examples: a junior employee identifies a real operational truth that conflicts with authoritative documentation; an AI extraction surfaces an undocumented but valid pattern; a minority expert view contradicts the dominant ontology; a regulatory interpretation changes before official guidance catches up. A rigid immunity system might classify these as pollution or contradiction and suppress them.

The defense is quarantine rather than rejection. Contested claims remain visible in provisional status, linked to contradiction type, evidence basis, and required validation pathway. The system preserves scoped disagreement where the domain is genuinely plural, contested, or evolving — it does not force premature consensus. The goal is not total capture of institutional knowledge (that is impossible) nor maximum exclusion of uncertain claims. The goal is to make action-critical knowledge sufficiently explicit, scoped, validated, and renewable for humans and AI agents to use safely. The system should be

safely incomplete rather than falsely complete.

---

3. The Capability Architecture: Three Scopes

The immunity framework specifies what threatens the system. The capability architecture specifies what the system must be able to do — organized across three scopes that determine who benefits from each capability.

3.1 Node-Level Capabilities

Properties of individual claims — what the system knows about each piece of knowledge. Claims are the atomic unit of governance, but the graph also contains entities, sources, events, and other objects. Claims are distinguished by being the objects to which validation, confidence, decay, and use-governance attach.

Ontology provides the typed schema. Every node has a type, every edge has a type, required properties are enforced. The ontology defines what kinds of knowledge exist in the domain and what relationships are valid. Ontology construction combines bottom-up extraction from code, configuration, documents, process artifacts, and usage traces with top-down domain modeling and expert review. AST parsing is useful for software artifacts, but governed ontologies cannot be inferred from artifacts alone.

Provenance and lineage track two distinct questions. Provenance: where did this claim come from? Lineage: what transformations did it undergo between source and graph? In regulated environments, auditors require both — not just the origin of a claim but the chain of processing decisions that shaped it. Provenance alone is necessary but insufficient — governed intelligence also needs: Is it still true? Where is it valid? Who validated it? What contradicts it? What depends on it? What action may be taken from it?

Scope is a first-class property of every governed claim. A governed claim is never merely true or false; it is true or false within a declared scope. Scope metadata includes jurisdiction, entity, client, process, system, version, temporal validity, and authority boundary. Many knowledge failures in regulated industries are not truth failures but scope failures — a claim that is true in the EU but false in the US, true for standard settlement but false for exception handling, true before a system migration but false after. Scope enforcement prevents claims from being applied outside the boundary within which they are valid.

Confidence scoring provides composite trust from multiple signals: source reliability, corroboration count, recency, and expert validation. Four deterministic levels — Provisional, Emerging, Validated, Foundational — map to structural checks. Bayesian updating means new evidence changes confidence in both directions. Confidence is not a generic trust score. It is a policy-bound decision variable determining whether a claim may be used for search, recommendation, reasoning, autonomous action, or regulatory evidence.

Bitemporal tracking maintains two independent time dimensions: valid-time (when the claim was true in the world) and transaction-time (when the system recorded it). This enables precise reconstruction of past knowledge states.

Decay and revalidation manage claim freshness with type-specific schedules, auto-revalidation where signals are clear, and human review where they are ambiguous.

3.2 Graph-Level Capabilities

Properties of the system as a whole — what the system can do that no individual claim can do alone.

Entity resolution creates coherence across sources, unifying the same concept under canonical identifiers while preserving aliases and source provenance.

Epistemic immunity provides systemic defense against the six failure modes specified in Section 2.

Typed contradictions manage disagreement explicitly. When the graph contains conflicting claims, the system types the contradiction — jurisdictional, temporal, logical, scope, extraction — and presents it transparently. Not all contradictions require resolution; jurisdictional contradictions (US rules versus EU rules) are expected and should persist. The system's job is to detect, type, and surface them.

Failure protocol declares the boundaries of what the system knows. When the graph lacks sufficient information to answer a query reliably, it says so — explicitly declaring what it does not know rather than generating a confident answer from insufficient evidence. In regulated environments, this property is more valuable than the knowledge itself. The system should not be described as a single source of truth. It is a governed source of claims, evidence, confidence, scope, contradiction, and history.

Neuro-symbolic reasoning bridges two modes. Large language models reason probabilistically — they understand natural language, detect patterns, and generate hypotheses, but they cannot guarantee logical consistency and they hallucinate. Symbolic reasoning — the kind built into ontology languages like OWL — applies formal rules to structured data deterministically. It doesn't hallucinate, but it can't handle natural language or novel situations.

The governed intelligence architecture combines both. During Ingest, LLMs extract claims from unstructured text. During Consolidate, symbolic rules verify those claims against the ontology — are the types valid, are required properties present, do relationships make sense? During Curate, the symbolic layer enforces consistency — a new claim that contradicts a Foundational claim is flagged automatically. During Apply, the reasoning engine chains logical rules across the subgraph to derive conclusions that neither the LLM nor the graph alone could reach. The combination is what makes the architecture safer for autonomous agents: the LLM brings understanding, the symbolic layer brings rule-bound consistency checks and constraint enforcement within the limits of the formal ontology. Symbolic reasoning does not guarantee truth outside the modeled assumptions — it guarantees consistency relative to formalized rules.

3.3 Delivery Capabilities

How the graph serves agents and users — determining what knowledge reaches whom, in what form, with what explanation.

Subgraph extraction delivers relevant slices of the graph in response to queries, with confidence information threaded through every link in the reasoning chain. An agent doesn't receive the whole graph; it receives the subset relevant to its task, with explicit signals about which parts of the reasoning chain are strong and which are uncertain.

Tier scoping controls access at the node and edge level — more granular than document-level access control. Different users and agents see different subgraphs based on their role and authorization, each receiving a complete, coherent view within their scope.

Explainability provides audit-grade reasoning transparency. Confidence scoring tells you how much to trust a claim; explainability tells you why. The system generates human-readable narratives tracing provenance, lineage, corroboration records, and validation history — satisfying the regulatory requirement to demonstrate not just what was known but how the system arrived at its conclusions.

Vector-graph hybrid retrieval combines semantic discovery (vector similarity for finding relevant entry points) with structured reasoning (graph traversal for following typed relationships, checking confidence, and extracting coherent subgraphs). This is not a transition from vectors to graphs but a fusion: L2 discovery feeding L3 reasoning.

Federated architecture enables reasoning across organizational boundaries without centralizing sensitive data. Multiple graph instances — across legal entities, jurisdictions, or client organizations — can participate in queries while preserving data sovereignty. The query travels to the data, not the other way around.

3.4 Standards Alignment

The capability architecture aligns with and extends two established frameworks. The IEEE 2807 family — specifically IEEE 2807.2, published June 2024 — defines technical frameworks and implementation guidelines for knowledge graphs in financial services. The lifecycle stages map to IEEE's construction methodology; the governance authorities map to their stakeholder framework. The Enterprise Knowledge Graph Forum Maturity Model, developed under the Object Management Group, assesses enterprise knowledge graphs across four pillars — Business, Organization, Data, and Technology — with five maturity levels from Initiation through Operational Ecosystem. The capability architecture targets Level 4 (Strategic Asset) and Level 5 (Operational Ecosystem), which require continuous improvement and governance frameworks.

Neither framework includes epistemic immunity, structural distortion defense, the three-scope capability taxonomy, or scope as a first-class governed property of claims. These are original contributions of

this architecture. The contribution is not that provenance, decay, contradiction, or ontology management are individually new — many fields have addressed them, including truth maintenance systems, belief revision, temporal databases, data governance, and ontology evolution. It is that they are treated as interdependent immune functions required for institutional AI to operate safely on organizational knowledge.

---

4. The Governed Intelligence Lifecycle

The immunity framework and capability architecture specify what the system defends and what it can do. The lifecycle specifies how it operates over time.

Living knowledge is an architecture, not a metaphor. It has five concurrent processes running as a continuous loop, with feedback mechanisms at every stage. The organizing principle is simple: claims are the atomic unit of governance, and connections between claims constitute the graph's reasoning capacity. Every stage creates or maintains one or both. The loop is the life; the stages without the feedback arrows are an archive with a more complex intake process.

Stage 1: Ingest — Acquire Within Authorized Scope

Everything that might become knowledge enters here as raw, unvalidated memory. The system does not judge quality at this stage — it collects.

Three authorized acquisition channels operate in parallel. *Authorized internal archive ingestion* accesses project-by-project designated repositories — project archives, test suites, data models, architecture documents, defect logs — under explicit authorization. *Structured expert capture* extracts practitioner knowledge through AI-assisted structured interview sessions, voluntary and fully disclosed, with output structured as memory rather than verbatim transcript. *External source acquisition* collects published regulations, industry standards, vendor documentation, licensed analyst intelligence, and continuous monitoring feeds.

Everything that enters carries provenance and source type. Nothing else. No confidence score, no connections, no structure. This is potential, not knowledge.

Stage 2: Consolidate — Memory Becomes Claims

The atomic unit of a governed intelligence system is not a document, a wiki page, or a summary. It is a claim: a discrete, verifiable assertion carrying provenance, confidence score, freshness date, scope metadata, and graph connections. "CSDR imposes cash penalties on settlement fails, calculated daily at CSD-set rates" is a claim. It can be confirmed, contradicted, or updated independently of every other claim in the graph. "Settlement is complex" is not a claim. It cannot be invalidated, updated, or connected to anything. Institutional AI cannot be governed at document granularity. Agents act on claims, not

documents. Governance must therefore move from document-level access and retention to claim-level validity, scope, confidence, and actionability.

Consolidation constructs claims from ingested material, builds the connections between them, and surfaces contradictions rather than silently resolving them. Contradictions are information: an expert account that conflicts with published documentation is typically evidence that the documented process and the actual process diverge — the precise kind of operational reality that regulatory examinations surface and that delivery teams need to know.

Entity resolution is a critical consolidation activity. When the same concept appears in multiple systems with different names — "PaymentInstruction" in one system, "PaymentOrder" in another — the system must unify them into canonical entities while preserving aliases. This process uses multiple signals: string similarity, semantic embedding comparison, structural analysis of entity properties, and contextual analysis of the systems they belong to. Automated matching resolves sixty to seventy percent of entities with high confidence. Medium-confidence candidates are presented in ranked groups for human review — typically two to four days of domain expert time for initial resolution of a new domain, then incremental thereafter. Each subsequent domain benefits from resolution patterns the system has already learned.

Connection-building is where graph topology emerges. A regulatory claim connects to the operational claims it governs, which connect to the technology claims describing what enforces them, which connect to the delivery patterns documenting what breaks when implementation diverges from the standard.

Stage 3: Curate — The Quality Engine

Curation is the architectural control point — the stage that prior knowledge management systems either collapsed into structuring (making it invisible) or omitted entirely (making it dangerous). In this lifecycle, curation is not a gate that material passes through once. It is the continuous quality engine that operates across the entire graph: validating, maintaining, deduplicating, fighting decay. It absorbs what traditional architectures treat as a separate maintenance stage, because maintenance is not something that happens at the end of a pipeline — it is something that happens to every claim, continuously, from the moment it enters the graph.

Six functions execute within curation. *Legal and IP classification* reviews every claim against a three-tier architecture: client-walled material (engagement-specific, restricted to the engagement team), domain graph material (sanitized, domain-general patterns compounding across engagements), and public domain material (regulations, standards, vendor documentation, unrestricted). Classification is enforced architecturally, not through policy guidance. *Deduplication* identifies near-duplicate claims; the richer provenance is preserved and a corroboration link is built rather than creating redundant nodes. *Quality assessment* flags material that is too thin, too old, or from an insufficient source — for human review, not automatic rejection. *Expert validation for fluency claims* ensures operational reality claims are validated by a practitioner with direct experience before entering the pipeline. Automating this step produces a

graph full of plausible-sounding operational claims with no one accountable for their accuracy.

Decay monitoring is the function that makes the graph a living system rather than a repository — and the function that pays down epistemic debt rather than letting it accumulate. Claims carry freshness dates and decay categories. Domain mechanics decay slowly (years; triggered by process redesign or technology migration). Regulatory claims decay at medium speed (months; triggered by regulatory change or enforcement action). Operational reality and client ecosystem claims decay fast (weeks; triggered by staffing changes, organizational restructures, leadership transitions). The system monitors freshness, triggers re-validation before staleness becomes operational risk, and demotes claims that fail re-validation — dropping their confidence score and removing them from active use without deleting them or their history.

Delivery feedback integration closes the loop between operational use and graph quality. Three capture points are built into the delivery cadence: at engagement start (the team loads domain context and identifies gaps and stale claims), at mid-delivery (a structured session identifying what contradicts or extends the graph), and at engagement close (structured harvest of what the next team would need to know). Remove the feedback loop and the system starves. Initial quality may be high; without continuous enrichment, the graph becomes historical within 18 months.

Stage 4: Expand — Growing Beyond What Was Put In

Most knowledge systems work only on what they have already ingested. The expansion stage is the architectural property that creates compounding returns.

Two mechanisms operate in parallel. *Lateral expansion* detects the current boundary of the knowledge graph — the nodes at the periphery with few connections and low confidence — and actively probes beyond it. Agents query domain experts about identified gaps, search external sources for claims that fill frontier nodes, and run structured inference across existing claims to surface implications not explicitly captured. *Scouting* runs as a continuous monitoring function watching for external signals relevant to existing claims: a new regulatory proposal, a vendor announcement, a published case study that contradicts an operational claim.

A static knowledge base grows when someone adds to it. A living knowledge system grows through its own activity.

Stage 5: Apply — Operational Use by Teams and Agents

Validated, high-confidence claims enter operational use through three channels. Delivery teams pull domain context at engagement start, receiving not a search interface over documents but a structured account of domain mechanics, regulatory context, failure modes, and operational edge cases — the equivalent of six months of domain immersion, available on day one. The graph-reasoning layer traces consequences across connected claims ("if this regulatory requirement changes, which systems, processes, and client configurations are affected?"), turning the graph from a searchable library into operational intelligence.

AI agents operating in the domain draw on the graph as their knowledge substrate; the quality of agent output is bounded by the depth of the domain knowledge available to them.

An AI system cannot be more reliable, auditable, or context-aware than the knowledge substrate on which it acts.

The quality gate before a domain enters operational use is sustained expert interrogation: can the knowledge base hold up across the full domain — end-to-end process flow, constraints, failure modes, regulatory context, edge cases — without gaps or contradictions that a practitioner would immediately surface? Shallow coverage collapses under extended interrogation; sufficient depth does not.

---

5. Theoretical Foundations: Why This Design

The architecture described in Sections 2-4 — the immunity framework, capability taxonomy, and lifecycle — is not a technology stack. It is the operational expression of a specific theoretical position: that institutional knowledge is not an asset to manage but a living system to inhabit. This section establishes why that position is well-founded and why three decades of the alternative approach have produced persistent failure patterns.

5.1 The Structural Failure of Asset-Based Knowledge Management

This is not an unsolved problem. It is a repeatedly failed one. The distinction matters because it tells us the solution is not to try harder with better technology. The approach itself requires examination.

The pattern across every era is consistent: better technology, same failure mode. In 1991, Brown and Duguid identified that organizational learning happens in informal practitioner networks, not formal knowledge transfer. Organizations responded by creating formal communities of practice that dissolved within 18 months of losing their central funding. In 1995, Nonaka and Takeuchi showed how knowledge spirals between tacit and explicit forms through social interaction — but their later argument that the spiral requires *ba*, a shared context for emerging relationships (Nonaka and Konno, 1998), was systematically ignored in favor of documentation systems that captured the easier half of the framework. Davenport and Prusak argued in 1998 that knowledge lives in people and relationships, not systems. The organizational response was knowledge management software. Snowden's Cynefin framework reached a wide audience in 2007 with the observation that complex knowledge requires ongoing sense-making, not documentation and retrieval. The prescribed response, in practice, was better documentation.

In the 2010s, SharePoint, Confluence, and enterprise wikis proliferated. In one unpublished internal audit reviewed by the authors, 73% of Confluence pages at a major consultancy had not been opened in the preceding 12 months. The technology matured; the failure mode didn't change. Since 2023, the major consulting and professional services firms have gone AI-native — McKinsey's Lilli, Accenture's Knowl-

edge Assist, EY's Discover Reimagined. These are serious systems with real investment and genuine adoption. The underlying architectural assumption is unchanged: knowledge is an asset — something to be stored and served.

Every knowledge management approach since 1991 shares one assumption so embedded it was never articulated as an assumption: that knowledge is an asset. Something you have. Something you can store. Something that, once captured, stays captured.

This assumption works for a specific class of knowledge: stable, bounded, and verifiable. Aviation checklists, pharmaceutical dosage protocols, regulatory text — these are artifacts, and disciplined document management handles them adequately. The knowledge that defeats every knowledge management system is structurally different. Michael Polanyi understood this. His concept of tacit knowledge is routinely interpreted as a taxonomy of two types — explicit and tacit. This interpretation distorts his argument into its opposite. Polanyi's claim was epistemological: knowing is always a process, and the tacit dimension is a structural property of all knowing. "We know more than we can tell" does not mean some knowledge is hard to transfer. It means that knowing is an act that always involves more than what is explicitly represented.

Karl Weick's work on sensemaking reinforces this. Organizational knowledge is meaning constructed retrospectively as people make sense of situations in context. Store the file without the context in which it was meaningful, and you have stored something — but not the knowledge. Ralph Stacey's work on complex responsive processes reaches the same conclusion from a different direction: knowledge in organizations is irreducibly relational, existing in the quality of interaction between people, not in documents that represent those interactions.

There is a second, compounding failure that makes this architecture necessary: maintenance. Knowledge decays. A regulatory claim may be outdated within months of the regulation changing. An operational reality claim can shift in weeks when a team restructures. Organizations build repositories, invest in capture, and then treat the system as complete — trusting that what was true at capture remains true at retrieval. It doesn't. What remains underdeveloped is an integrated architecture that treats decay, fragmentation, contradiction, provenance failure, and cascade risk as a single class of epistemic integrity problems — and that treats this class as a first-order design concern rather than an afterthought.

5.2 The Living System Argument

Three decades of knowledge management failure point to a category error, not a technology gap. The right model for institutional knowledge is not a library. A knowledge system for the kind of knowledge that defeats every KM approach requires continuous renewal — a system that must renew itself or cease to function.

Edwin Hutchins' work on distributed cognition provides the strongest theoretical anchor. His analysis of aircraft navigation — the "cockpit that remembers" — shows that cognitive capability in complex systems

is not located in individual minds but distributed across representations in instruments, procedures, trained crews, and the physical environment. The system's intelligence is a property of the configuration, not any component. This is the property a knowledge graph creates: system-level intelligence distributed across claims, connections, and the agents that traverse them. No single person needs to hold the full domain model. The model is distributed.

Humberto Maturana and Francisco Varela's concept of autopoiesis supplies a design principle rather than a literal identity. The system is not autopoietic in the biological sense. Rather, autopoiesis provides a requirement: institutional knowledge systems must maintain identity through continuous processes of validation, boundary control, and renewal, not through static storage.

Ilya Prigogine's work on dissipative structures provides a complementary insight. The analogy is not thermodynamic identity but organizational logic: stability is maintained through throughput, not stasis. Knowledge repositories are equilibrium structures — they achieve stability through stasis, which is why they decay. A living knowledge system requires continuous metabolic flow.

Stuart Kauffman's work on the adjacent possible provides the competitive argument. The adjacent possible is the set of outcomes reachable from where you are now. An organization that builds deep knowledge infrastructure is continuously expanding the set of what is possible: new connections become visible, new combinations become available, new solutions become reachable that were previously out of range. An organization that builds execution capability without knowledge depth is optimizing within a fixed frontier. The tools get faster; the frontier doesn't move.

5.3 The Three Types of Invisible Knowledge

Not all knowledge that defeats KM systems is the same problem. Three categories exist, each requiring a distinct intervention:

Tacit knowledge is genuinely below conscious articulation — the expert's felt sense for when a process is about to break, the pattern recognition that fires before the pattern is named. It was never explicit, so it cannot be extracted. The appropriate intervention is better conditions for propagation: shared operational contexts where tacit patterns leave traces that others can absorb through practice.

Undocumented explicit knowledge is articulable but never written down — the exception-handling logic a specialist could explain in detail if asked, but nobody asks systematically. The appropriate intervention is structured capture: AI-assisted interview sessions that surface what practitioners know they know, directed toward the specific claim types the system needs.

Dark knowledge — valid institutional knowledge that exists in practice but is invisible to formal systems — is already written but structurally invisible. Buried in test suites, defect logs, architecture decision records, project post-mortems, Slack discussions, code comments, and client-specific adaptations. The appropriate intervention is systematic ingestion with intelligent extraction: treating the organization's

documentary history as an underutilized raw material rather than an archive.

Conflating these three and applying the same documentation-based approach to all of them is a primary reason knowledge management consistently fails to capture what matters most.

---

6. Path Dependency and the Fertility Trap

Building or deferring this infrastructure is a strategic commitment with asymmetric consequences — not a reversible capability choice that can be made later without cost.

6.1 Capability and Fertility

Organizations investing in AI capability are building execution power: the ability to perform specified functions faster, at lower cost, with higher consistency. This is real and valuable. It is also, by itself, insufficient for value generation at the level the market data implies is available.

The reason is captured by what we term the fertility framework. Capability and fertility are distinct organizational properties with a specific sequential relationship. Capability is the power to execute within a defined possibility space. Fertility is the capacity to expand the possibility space itself — to generate novel value, discover non-obvious solutions, perceive opportunities that were previously out of range. Kauffman's adjacent possible is the formal expression of fertility: the set of outcomes reachable from current conditions. Capability optimizes within the adjacent possible; fertility expands it. This framework is developed in full theoretical depth in Paper C (Reichhart and Gelas, 2026c).

The strategic insight is that these two properties are sequentially dependent in a way that creates path dependency. An organization that builds capability first — execution speed, AI tooling, process automation — and then attempts to build fertility is attempting to build it on top of a structure optimized for the opposite. Capability-first AI programmes harden workflows, incentives, data structures, and agent behaviors around shallow knowledge substrates. Retrofitting deep knowledge infrastructure later requires not only technical integration but organizational unlearning. The cost of re-opening the window rises sharply as capability-optimized patterns solidify.

6.2 The Current Competitive Configuration

Four competitive positions follow from this framework, and they are not symmetric.

Most organizations were in the low capability/low fertility position in 2022: neither AI execution capability nor knowledge depth. The most common current configuration — for organizations that have invested heavily in AI since 2023 — is high capability/low fertility. They are powerful tools: executing efficiently with AI assistance, producing results faster and cheaper. They are not getting smarter through operation. Every engagement produces the same quality of output, because the knowledge substrate doesn't

compound.

A third position — low capability/high fertility — exists in specialist advisory firms with deep knowledge and limited AI tooling. They carry domain depth that could become a Living Medium relatively quickly if they add the execution layer.

The fourth position is the Living Medium: AI agents operating on deep, continuously refreshed knowledge graphs with epistemic immunity. Every engagement enriches the substrate for the next. The quality of output improves because the knowledge available to the models deepens — and the integrity of that knowledge is maintained by the immunity framework rather than degrading through use. The graph deepens, the agents perceive more, ramp time shortens, and the gap between this organization and a competitor starting from zero widens each quarter.

6.3 The Agent Dimension

The path dependency argument sharpens when the organizational actors consuming the knowledge base are AI agents rather than human practitioners.

Human practitioners can partially compensate for thin knowledge infrastructure through informal networks, personal experience, and the tacit pattern recognition Polanyi describes. They ask colleagues, triangulate across incomplete documentation, and apply judgment built from years of domain exposure. These compensatory mechanisms are slow and unreliable, but they exist.

AI agents cannot compensate in this way. An agent configuring a settlement system without domain knowledge produces the documented answer, confidently, at scale — and the documented answer is frequently wrong in exactly the ways that matter. The gaps are systematic: divergence between documented and actual process, undocumented edge cases, the legacy reconciliation procedure that nobody acknowledges but that is the load-bearing constraint. These errors compound with scale, which is what makes thin knowledge infrastructure dangerous in AI-assisted delivery contexts.

This has a second implication beyond execution quality. The distinction between AI agents that are *autonomous* (independent in how they execute assigned objectives) and agents that have genuine *initiative* (capable of perceiving what is worth pursuing without being explicitly assigned it) depends on contextual immersion. An agent operating over a deep, governed domain substrate is better positioned to surface anomalies, detect non-obvious connections, and identify opportunities that no instruction set could have specified in advance. Initiative is not a feature of model architecture alone. It emerges from the interaction between model capability, task environment, tool access, governance, and depth of contextual knowledge substrate — and the integrity of that substrate, maintained by epistemic immunity, is what prevents the agent from acting on corrupted knowledge with the same confidence it acts on sound knowledge.

The engineering methodology for building governed agents over this substrate — including phased autonomy tiers, evidence-backed deployables, and structured governance evaluations — is the subject of

ongoing work within this programme. The methodology governs the agent; this paper governs the knowledge the agent acts on. Neither is sufficient alone.

Organizations that build governed intelligence systems will operate over richer, fresher, more coherent, and better-governed knowledge substrates than competitors' systems. That gap compounds with every engagement while epistemic immunity prevents epistemic debt from accumulating beneath the growth.

---

7. Implications

7.1 Epistemic Operational Risk

AI turns knowledge degradation from an internal productivity problem into an operational risk category. Before agents, stale knowledge meant a practitioner wasted time. With agents, stale knowledge means automated regulatory violations, cascading decision errors, and audit failures — at scale.

Epistemic operational risk — the risk of institutional harm caused by acting on degraded knowledge — should be recognized as a distinct risk category alongside model risk, operational risk, and cyber risk. The six failure modes map directly to risk consequences:

Pollution produces wrong advice, bad configurations, and regulatory breaches. Staleness produces outdated compliance and obsolete process execution. Fragmentation produces partial reasoning, duplicated work, and missed dependencies. Amnesia produces failed audits and inability to explain past decisions. Cascade failure produces systemic error propagation. Structural distortion produces biased retrieval and false proximity.

Governed intelligence infrastructure belongs in a separate strategic investment category — distinct in its governance requirements, compounding return profile, and time horizon from IT projects or knowledge management programmes. The governance implications are material: knowledge domain ownership (accountable individuals who sign off on promotion decisions and own the readiness verdict for their domain), expert registries that track who knows what at what confidence level, and data partitioning enforced architecturally rather than through policy guidance.

The sequencing principle is: start where you already have material and can close the loop. A single domain, built through the full five-stage lifecycle to the point where the feedback loop activates, proves the model. Based on comparable enterprise graph deployments and the authors' implementation experience, initial timelines are ten to fourteen weeks per domain to reach operational readiness, six to nine months to production integration, with ongoing curation cost of half to one full-time equivalent per domain. These estimates should be adjusted by domain complexity, source quality, regulatory burden, and SME availability. The compounding begins at first feedback.

7.2 For AI and Knowledge System Design

The architectural distinction between vector databases and knowledge graphs is not a technical preference. It is the difference between retrieval and reasoning.

Vector databases measure probabilistic word proximity: they find documents whose language is statistically similar to a query. For discovery, this is useful. For diagnosis, it is insufficient. The gap is causal — why a 1990 reconciliation process is the structural cause of a 2024 compliance gap — and that causal chain requires mapped relationships between concepts, not word proximity.

A knowledge graph maps the relationships. An agent operating on one can traverse causal chains, understand why two claims connect, and reason about what changes downstream when one node shifts. Combined with neuro-symbolic reasoning — where LLMs generate hypotheses and symbolic logic verifies them against graph constraints — the system achieves a safety profile that neither approach offers alone.

Enterprise AI shifts the bottleneck from information access to epistemic validity. Before LLMs, the problem was finding information. With LLMs, the problem is knowing whether found or generated information is valid for use. Search assumes the user judges relevance and validity. Agentic AI collapses that assumption because the system may retrieve, reason, and act without a human inspecting every source. Knowledge systems must therefore move from search-oriented design to action-oriented epistemic governance.

7.3 For Regulated Industries Specifically

Financial services, healthcare, and other regulated industries face a particular configuration that makes the case for governed intelligence infrastructure acute.

Regulatory environments change faster than documentation cycles. In April 2026, U.S. banking agencies issued revised model risk management guidance superseding SR 11-7. OCC guidance states that generative and agentic AI models are not within the scope of that guidance, creating a governance boundary problem for institutions deploying precisely those systems. The EU AI Act's high-risk classification imposes conformity requirements — including risk management systems, data governance, technical documentation, record keeping, transparency, human oversight, and accuracy — for which claim-level provenance, confidence tracking, and audit trails may provide a stronger implementation substrate than document-level controls alone. DORA's operational resilience requirements — covering ICT risk management, incident reporting, resilience testing, third-party risk management, and information sharing — position knowledge about systems as itself an operational resilience asset. A system that loses its institutional knowledge of critical processes has suffered a resilience failure regardless of whether any system went down.

The three-tier data architecture — client-walled, domain graph, public domain — is not only a governance requirement. It is the architectural mechanism that allows knowledge to compound across engagements

without creating client confidentiality exposure. Without this architecture, the choice is binary: either knowledge is locked inside individual engagements (no compounding) or it is shared (confidentiality risk). The three-tier design resolves this.

The epistemic immunity framework is particularly relevant in regulated contexts because the cost of knowledge failure is asymmetric. A stale claim in a consumer application produces a poor recommendation. A stale claim in a compliance system produces a regulatory violation. A cascade failure in a consumer graph produces inconsistent search results. A cascade failure in a compliance graph produces a chain of decisions built on a compromised foundation — each one a potential enforcement action. The immunity framework is not a quality-of-life improvement. It is risk management at the knowledge layer.

7.4 Limitations

This paper offers a conceptual and implementation architecture, not a completed empirical validation. The implementation estimates — timelines, staffing, entity resolution rates — are drawn from the authors' experience with comparable enterprise graph deployments and should be treated as implementation hypotheses requiring domain-specific calibration. Not every claim requires equal governance; the architecture should apply heavier controls to claims that are action-critical, regulatory, client-impacting, or agent-actionable. Future work should evaluate whether epistemic immunity reduces stale-claim use, improves reconstructability of AI-assisted decisions, reduces contradiction leakage into delivery workflows, and improves safe agent-action rates in regulated environments.

---

8. Conclusion

The organizations that built great IT departments in the 1990s had a structural advantage that eroded as IT became infrastructure. The organizations that built great software delivery capability in the 2000s had a similar edge; it is eroding now as AI makes technical delivery increasingly table stakes.

The next structural advantage is knowledge infrastructure — a living medium with the immune defenses to maintain its integrity under continuous operation. A medium is inhabited, enriched through use, and becomes more valuable the more people and agents operate within it. Every engagement deepens the substrate. The immunity framework ensures the deepening is healthy — that epistemic debt does not accumulate beneath the growth.

The broader world modelling research community is converging on the same conclusion from a different direction. Chu et al. (2026) survey over 400 systems across physical, digital, social, and scientific domains, identifying evidence-driven model revision (their L3 Evolver) as the capability frontier. Their governance discussion identifies three risks — benchmark overfitting, knowledge contamination, and misattribution cascades — that map directly onto the epistemic immunity framework's six failure modes,

albeit in ML systems rather than institutional knowledge substrates. What the ML community calls "governed validation" for model revision, this paper calls epistemic immunity for knowledge lifecycle management. The parallel is structural: any system that revises itself from evidence requires defenses against contamination, mechanisms for detecting degradation, and governance gates that distinguish valid revision from drift. The Governed Intelligence Lifecycle provides that infrastructure for the organisational regime — where the evidence is regulatory change, operational learning, and market dynamics rather than experimental measurements.

AI has raised what it costs to operate without this infrastructure. Agents at scale are bounded by the knowledge available to them. Bounded agents at scale produce bounded results at scale. The organizations compounding epistemic capital through governed knowledge systems will operate on qualitatively different substrates than competitors who deferred — and the knowledge infrastructure gap, unlike the capability gap, cannot be closed by purchasing the solution. It must be built, and building takes time that deferral consumes.

Institutional AI requires a governed knowledge substrate capable of preserving provenance, detecting degradation, maintaining context, and learning from use. Without that substrate, AI scales institutional ignorance as readily as institutional intelligence. The architecture described in this paper makes human judgment accountable, traceable, and reusable — governed intelligence operating over a protected, living knowledge substrate that compounds through the work it enables.

References

Brown, J. S., & Duguid, P. (1991). Organizational learning and communities of practice: Toward a unified view of working, learning, and innovation. *Organization Science*, 2(1), 40-57.

Chu, M., Zhang, X.B., Lin, K.Q., Kong, L., Zhang, J., Tu, T., Ma, W., et al. (2026). Agentic world modeling: Foundations, capabilities, laws, and beyond. *arXiv preprint arXiv:2604.22748*.

Davenport, T. H., & Prusak, L. (1998). *Working Knowledge: How Organizations Manage What They Know*. Harvard Business School Press.

Enterprise Knowledge Graph Forum. (2024). *EKG Maturity Model*. Object Management Group. https://maturity.ekgf.org

Hutchins, E. (1995). *Cognition in the Wild*. MIT Press.

IEEE. (2024). *IEEE 2807.2-2024: Guide for Application of Knowledge Graph for Financial Services*. IEEE Standards Association.

Kauffman, S. A. (1995). *At Home in the Universe: The Search for the Laws of Self-Organization and Complexity*. Oxford University Press.

Kauffman, S. A. (2000). *Investigations*. Oxford University Press.

Kent, S. (1949). *Strategic Intelligence for American World Policy*. Princeton University Press.

Maturana, H. R., & Varela, F. J. (1980). *Autopoiesis and Cognition: The Realization of the Living*. Reidel.

McKinsey & Company. (2026). The AI transformation manifesto: 12 themes driving growth. *McKinsey Quarterly*, April 2026.

Nonaka, I., & Konno, N. (1998). The concept of "ba": Building a foundation for knowledge creation. *California Management Review*, 40(3), 40-54.

Nonaka, I., & Takeuchi, H. (1995). *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation*. Oxford University Press.

Polanyi, M. (1966). *The Tacit Dimension*. Doubleday.

Prigogine, I., & Stengers, I. (1984). *Order Out of Chaos: Man's New Dialogue with Nature*. Bantam Books.

Reichhart, W., & Gelas, A. (2026a). *Dynamics Blindness: When AI Is Locally Correct and Globally Non-Compliant*. Working Paper.

Gelas, A., & Reichhart, W. (2026b). *The Predictive Organization: Architecture for Enterprise Intelligence*. Working Paper.

Reichhart, W., & Gelas, A. (2026c). *Build the Medium: Why Organizational Intelligence Is Mechanism, Not Metaphor*. Working Paper.

Reichhart, W., & Gelas, A. (2026e). *From Autonomy to Initiative: Enterprise AI's Real Endgame.* Working Paper.

Snowden, D. J., & Boone, M. E. (2007). A leader's framework for decision making. *Harvard Business Review*, 85(11), 68-76.

Stacey, R. D. (2001). *Complex Responsive Processes in Organizations: Learning and Knowledge Creation*. Routledge.

Tsoukas, H. (2003). Do we really understand tacit knowledge? In M. Easterby-Smith & M. A. Lyles (Eds.), *The Blackwell Handbook of Organizational Learning and Knowledge Management* (pp. 410-427). Blackwell.

Tsoukas, H., & Vladimirou, E. (2001). What is organizational knowledge? *Journal of Management Studies*, 38(7), 973-993.

Weick, K. E. (1995). *Sensemaking in Organizations*. Sage.

Correspondence: witold.reichhart@gmail.com, arnaudgelas@gmail.com