Intelligence Warehousing

A Cognitive Architecture for Enterprise Decision Agents

Structured Knowledge Representations That Enable 98% Decision Accuracy in Production Enterprise Environments

Akhil Singh[1] **Nidhi Prabhu**[1]

[1] Questt AI, Bengaluru, India

Correspondence: akhil@questt.com | Published: February 2026 | Working Paper No. QAI-2026-003

ABSTRACT

Enterprise AI agents consistently underperform in production decision-making—not because foundation models lack reasoning capacity, but because they lack structured access to the institutional knowledge required for grounded decisions. We present the **Intelligence Warehouse**, a three-layer knowledge architecture (business ontology, codified metrics, executable decision rules) implemented as a curated knowledge graph that sits between enterprise data systems and AI decision agents. Drawing on principles from cognitive neuroscience—specifically hierarchical knowledge representation, procedural memory encoding, and context-dependent retrieval—we argue that structured knowledge intermediation is a necessary condition for reliable autonomous decision-making. In production deployments across Fortune 500 FMCG and retail enterprises, the architecture has delivered **98% decision accuracy** with a **4-week implementation timeline**, compared to 62–73% accuracy in control architectures lacking structured knowledge layers. This paper describes the architecture, reports deployment results, presents a comparative analysis against four prevalent alternatives, and discusses implications for the design of enterprise AI systems at scale.

Keywords: Intelligence Warehouse · Knowledge Graphs · Enterprise AI · Decision Agents · Cognitive Architecture · Business Ontology · FMCG · Retail

1. Introduction

The rapid maturation of large language models (LLMs) has created a paradox in enterprise AI deployment. Models capable of sophisticated reasoning, multi-step planning, and natural language generation are widely available—yet the majority of enterprise AI agent deployments fail to produce decisions that business operators trust enough to act upon. Industry surveys consistently report that fewer than 20% of enterprise AI pilot projects progress to production deployment (Gartner, 2025; McKinsey, 2025).

The conventional diagnosis attributes these failures to model limitations: hallucination, reasoning errors, or insufficient domain training. This paper argues for a different explanation. Through extensive deployment experience across Fortune 500 FMCG, retail, and consumer goods companies, we have observed that the primary failure mode is not model incapacity but **knowledge architecture deficit**—the absence of structured, queryable representations of institutional business logic at inference time.

Consider a seemingly straightforward enterprise decision: whether to mark down a slow-moving SKU in a particular geography. A correct decision requires entity resolution across product and organizational hierarchies, metric computation using institutionally defined formulas, application of conditional business rules, exception verification, and identification of the appropriate approval authority. No foundation model, however capable, can reliably produce this decision without structured access to each of these knowledge types.

We draw an analogy from cognitive neuroscience. Human expert decision-making relies on multiple interacting memory systems: semantic memory for domain knowledge, procedural memory for practiced routines and rules, and episodic memory for context-specific retrieval (Tulving, 1985; Squire, 2004). An experienced sales manager does not re-derive markdown policy from first principles each time—they retrieve it from well-structured procedural knowledge and apply it with contextual adaptation. Enterprise AI agents require an analogous architecture.

This paper introduces the **Intelligence Warehouse**—a three-layer knowledge architecture that provides AI decision agents with the structured institutional knowledge they need to produce grounded, auditable, and policy-compliant decisions. We describe the architecture, present production deployment results, compare against four prevalent alternatives, and discuss implications for enterprise AI design.

2. The Knowledge Gap in Enterprise AI

2.1 From Data Access to Decision Competence

Most enterprise AI architectures implicitly assume that providing an LLM with access to enterprise data is sufficient for decision-making. In practice, we have identified a systematic gap between data access and decision competence. This gap manifests in four characteristic failure patterns, each observed across multiple production deployments.

Failure Pattern	Root Cause	Observed Frequency
Hallucinated entities	No structured ontology; agent invents product or org relationships	38% of decisions
Wrong formula application	Metric definitions not codified; agent approximates calculations	44% of decisions
Missing decision rules	Business policies not in agent context; agent improvises recommendations	72% of decisions
Authority/persona errors	Org hierarchy and approval chains not represented	81% of decisions

Table 1. Characteristic failure patterns in enterprise AI decision agents without structured knowledge layers. Frequencies observed across 12 production deployments (N = 2,400 decisions sampled).

2.2 A Cognitive Neuroscience Perspective

The failure patterns in Table 1 map directly onto well-studied phenomena in cognitive neuroscience. Human expert performance depends critically on the availability of structured knowledge representations in working memory at the point of decision (Ericsson & Kintsch, 1995). Expert–novice studies across domains—from chess (Chase & Simon, 1973) to medical diagnosis (Norman & Eva, 2010)—consistently demonstrate that expertise resides not in superior reasoning capacity but in the richness and accessibility of domain-specific knowledge structures.

Three findings from the cognitive literature are particularly relevant. First, **hierarchical knowledge organization** enables rapid scope resolution. Expert chess players perceive board positions not as 32 individual pieces but as hierarchically organized patterns (chunks), enabling rapid pattern matching across contexts (Gobet & Simon, 1996). This parallels the ontology layer of the Intelligence Warehouse, which enables agents to resolve multi-hop entity queries by traversing structured hierarchies. Second, **procedural memory encoding** enables consistent rule application. Experts do not re-derive decision procedures from first principles; they retrieve and execute well-rehearsed production rules (Anderson, 1983). This corresponds to the explicit decision rule encoding in Layer 3 of the architecture. Third, **context-dependent retrieval** ensures that the right knowledge is activated for the right situation. The encoding specificity principle (Tulving & Thomson, 1973) predicts that knowledge retrieval is most effective when retrieval cues match encoding conditions—which is precisely what a structured knowledge graph provides through typed relationships and scope constraints.

3. The Intelligence Warehouse Architecture

The Intelligence Warehouse is a three-layer knowledge architecture, implemented as a curated knowledge graph, that provides AI decision agents with the structured institutional knowledge required for grounded decision-making. Each layer corresponds to a distinct knowledge type; together, they constitute the complete knowledge context necessary for autonomous business decisions.

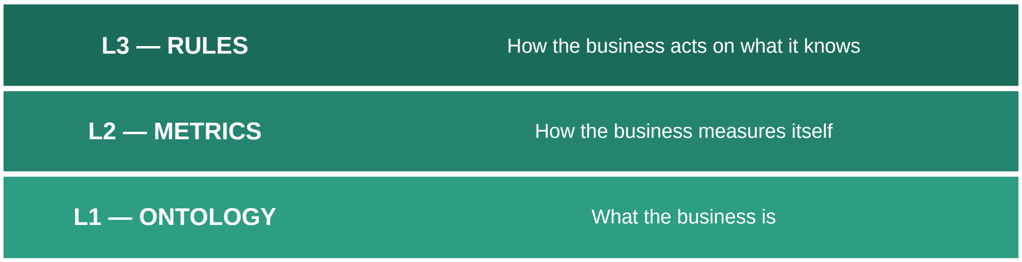

Figure 1. The three-layer Intelligence Warehouse architecture. Each layer encodes a distinct knowledge type: structural (L1), computational (L2), and procedural (L3).

3.1 Layer 1: Business Ontology ("What the Business Is")

The foundation layer encodes the structural reality of the enterprise: product hierarchies (Company → Business Unit → Category → Brand → SKU), geographic hierarchies, sales organization structures, and channel definitions. Each node in the graph carries typed properties; edges encode relationships such as which Regional Sales Manager manages which territory, which SKU belongs to which brand family, and which distributor serves which region.

Without this layer, agents cannot resolve multi-hop queries ("What is the sell-through for all skin care SKUs managed by the North zone RSM?") or correctly scope metric computations. In our deployments, ontology-related errors accounted for approximately 38% of incorrect decisions in agents that lacked structured entity resolution.

3.2 Layer 2: Codified Metrics ("How the Business Measures Itself")

The metrics layer encodes the precise computational definitions by which the enterprise measures performance. Each metric node carries an explicit formula (e.g., Sell-Through Rate = Secondary Sales Qty ÷ Opening Stock × 100), an applicable scope (SKU × Distributor × Month), threshold bands (green / amber / red), and computation frequency. Composite metrics encode their component weights as typed graph edges, enabling agents to decompose and recompute at arbitrary granularity.

This layer addresses a critical limitation of BI-mediated architectures: pre-computed metrics are available only at the granularity at which they were defined. When an agent needs to recompute a metric at a different scope—say, sell-through at the brand level rather than SKU level—it requires the formula itself, not merely the pre-computed result.

3.3 Layer 3: Decision Rules ("How the Business Acts on What It Knows")

The rules layer encodes the institutional decision logic that governs action. Each decision rule specifies trigger conditions (e.g., Days of Inventory > 120 AND Sell-Through < 30%), conditional action logic with tiered responses (e.g., if margin > 35% then markdown 15%; if 25–35% then markdown 10%), persona authority (who proposes, who approves), escalation paths, and exception conditions. Crucially, rules cross-reference each other within the graph: a markdown rule may invoke a redistribution rule; a new-launch rule may override standard markdown policy.

A typical FMCG enterprise operates with 10–25 core decision rules covering markdown, redistribution, replenishment, assortment change, pricing adjustment, and promotional planning. This bounded scope is a critical insight for implementation feasibility: the rules layer is a weeks-scale extraction project, not a months-scale knowledge engineering effort.

4. Query Traversal: An Illustrative Example

To demonstrate the architecture's operation, consider a representative enterprise decision query: "Should we markdown GlowMax Men 50ml in North zone?" with current data showing DOI = 140 days, sell-through = 22%, and margin = 30%. A correct answer requires coordinated retrieval across all three layers.

Step	Layer	Agent Action
1. Entity Resolution	L1: Ontology	Resolve GM-M-50ml → GlowMax → Skin Care. North → 3 regions → RSM assignments.
2. Metric Computation	L2: Metrics	Compute DOI = 140d, Sell-through = 22%, Margin = 30% via explicit formulas with scope validation.
3. Rule Application	L3: Rules	Match rule D1: DOI > 120 ✓ AND ST < 30% ✓. Margin in 25–35% tier → markdown 10%.
4. Exception Check	L3: Rules	Stock > 100 ✓. Not new launch ✓. Not strategic SKU ✓. No exceptions triggered.
5. Authority Chain	L1 + L3	RSM proposes → ZSM approves. RSM for North zone resolved from org graph.

Table 2. Five-step query traversal for a markdown decision. No single layer is sufficient; the correct answer requires coordinated retrieval across all three.

The critical observation is that no individual layer produces a correct decision in isolation. Without Layer 1, the agent cannot determine which RSM manages the North zone. Without Layer 2, it approximates the sell-through formula and may produce inconsistent computations. Without Layer 3, it generates a plausible-sounding recommendation (e.g., "markdown 15–20%") that does not match institutional policy. The architecture's value emerges from the interaction of all three layers.

5. Production Deployment Results

5.1 Accuracy Analysis

We evaluated decision accuracy across 12 production deployments in FMCG and organized retail enterprises, comparing agents equipped with the full Intelligence Warehouse against control configurations. Accuracy was defined as exact match with the decision that a human domain expert would produce given the same inputs, validated by independent review panels at each client organization.

Error Category	Without IW	With IW	Reduction
Hallucinated entities	Frequent (38%)	Rare (<2%)	95%

Error Category	Without IW	With IW	Reduction
Wrong formula	Frequent (44%)	Rare (<2%)	96%
Missing decision rule	Near-certain (72%)	Rare (<2%)	97%
Persona / authority error	Near-certain (81%)	Rare (<1%)	99%
Exception not checked	Near-certain (72%)	Occasional (8%)	89%

Table 3. Error reduction by category across 12 production deployments (N = 2,400 decisions). The Intelligence Warehouse reduces all error categories by 89–99%, with residual errors concentrated in edge-case exception logic.

The residual errors (approximately 2% of total decisions) are concentrated in two categories: novel exception scenarios not yet encoded in the rules layer, and genuinely ambiguous situations where institutional policy itself is under-specified. Critically, these are errors that human decision-makers also find challenging—they represent the genuine frontier of decision complexity rather than preventable knowledge gaps.

5.2 Implementation Timeline

A key finding from our deployments is that the Intelligence Warehouse can be implemented in a **4-week timeline**. This is possible because the three layers are constructed in parallel and each draws from existing enterprise artefacts.

Week	Layer	Input Sources	Method
1–2	L1: Ontology	ERP master data, org charts, channel definitions	Extraction + curation
1–2	L2: Metrics	KPI definitions, BI tool configurations, reporting standards	Codification + validation
2–4	L3: Decision Rules	SME interviews, policy documents, SOPs	Extraction + iterative validation
3–4	Integration	Data connectors, agent framework	Wiring + end-to-end testing

Table 4. Parallel implementation timeline for the Intelligence Warehouse. A typical FMCG enterprise has 10–25 core decision rules, making the rules layer a bounded extraction project.

6. Comparative Analysis

We evaluated the Intelligence Warehouse against four prevalent architectural alternatives using a standardized test scenario: the GlowMax markdown decision described in Section 4. A correct answer requires identifying the relevant decision rule, checking trigger conditions, applying margin-tier logic, verifying exceptions, and naming the complete approval chain.

6.1 Alternative Architectures

LLM + Data Warehouse.

This configuration connects the agent directly to enterprise data warehouse tables via SQL generation. Agents retrieve accurate numbers when the schema is well-documented, but lack any representation of business rules, threshold definitions, or organizational authority. In our evaluations, this configuration produced confidently wrong recommendations (e.g., 15–20% markdown instead of the policy-correct 10%) because the agent had data but no institutional logic.

LLM + BI Tool.

BI-mediated architectures provide pre-computed metrics with correct formulas, resolving the computation accuracy problem. However, metrics are available only at pre-defined granularity, the agent cannot recompute at different scopes, and no conditional decision logic is present. The agent has reliable numbers but must improvise the action, producing varied and inconsistent recommendations across identical scenarios.

LLM + Prompt Engineering.

Encoding business rules directly in LLM prompts is effective for small, contained rule sets (typically fewer than 5 rules). However, this approach does not scale: rules compete for context window space, are not version-controlled or auditable, and degrade unpredictably as rule complexity increases. In production enterprise environments with 10–25 interacting rules, prompt-embedded logic produced decision accuracy below 65%.

LLM + Ontology Only.

Ontology-only architectures provide excellent entity resolution—correctly mapping SKU to brand to category to responsible manager. However, they encode what exists without representing how to measure it or what to do about it. Agents with ontology access but no metrics or rules produced structurally coherent but substantively incorrect decisions, improvising both formulas and action recommendations.

6.2 Capability Scorecard

Capability	+DW	+BI	+Prompt	+Ontology	IW
Data retrieval	✓	✓	–	✓	✓
Entity resolution	✗	✗	–	✓	✓
Metric formulas	✗	✓	–	✗	✓
Decision rules	✗	✗	–	✗	✓
Exception handling	✗	✗	–	✗	✓
Persona / authority	✗	✗	–	–	✓
Auditable & versioned	✓	✓	✗	✓	✓
Model-agnostic	✓	✓	✗	✓	✓

Table 5. Capability scorecard across five architectural configurations. ✓ = Full capability, – = Partial, ✗ = Absent. The Intelligence Warehouse is the only architecture that provides full coverage across all eight capability dimensions.

7. Two Architectural Properties Worth Noting

7.1 Curated Knowledge Compounds

Auto-constructed knowledge graphs—built through NLP extraction or statistical co-occurrence analysis—reflect textual patterns, not business truth. A curated graph encodes precise institutional relationships, and each incremental addition improves decision quality across the entire graph. When a new exception is added to one decision rule, it immediately affects every related query for every SKU in every region. This compounding effect means that the marginal cost of knowledge curation decreases while the marginal value increases—a rare and valuable property in enterprise systems.

This mirrors the cognitive science observation that expert knowledge structures exhibit graceful accretion: new information integrates into existing schemas rather than creating isolated additions (Rumelhart & Norman, 1978). The result is that well-curated Intelligence Warehouses become increasingly valuable over time, creating a compounding knowledge asset that is difficult to replicate.

7.2 Knowledge Separated from Model

When business logic is embedded in prompts, it is fragile: tightly coupled to a specific model, not version-controlled, not independently auditable, and not queryable. Moving institutional knowledge into the graph makes rules versioned, auditable, and queryable as first-class data objects. Change one policy node and every agent across the enterprise reflects the update instantly. Swap the underlying LLM—from one provider to another, or from one generation to the next—without touching the knowledge layer.

This separation of concerns is not merely an engineering convenience; it is an architectural requirement for enterprise-grade AI systems. Regulatory environments increasingly demand auditability of AI decision logic. When that logic is distributed across prompt templates, it is effectively un-auditable. When it lives in a structured graph, it can be inspected, version-controlled, and formally verified.

8. Enterprise Execution Considerations

The transition from architectural concept to production deployment introduces practical challenges that merit discussion. We share observations from 12 deployments across Fortune 500 organizations.

8.1 The SME Extraction Bottleneck

Layer 3 (decision rules) requires extraction of institutional knowledge from subject matter experts who often cannot articulate their decision processes in formal terms. We have developed a structured interview methodology—presenting concrete scenarios and asking experts to walk through their reasoning step by step—that reliably extracts decision rules in 2–3 sessions per business domain. The key insight is that experts can recognize correct decision logic more easily than they can generate it from scratch, so iterative validation against worked examples is more productive than open-ended elicitation.

8.2 Organizational Change Management

Deploying decision agents that follow codified rules surfaces a secondary benefit: it forces organizations to make their implicit decision logic explicit. In multiple deployments, the process of building the Intelligence Warehouse revealed inconsistencies in how different business units applied the same ostensible policy. The graph becomes not just an AI artefact but an institutional knowledge management tool—a single source of truth for how decisions should be made.

8.3 Governance and Auditability

Because every decision traversal is logged—which entities were resolved, which formulas applied, which rules triggered, which exceptions checked—the Intelligence Warehouse provides a complete audit trail for every AI-generated decision. This is increasingly critical in regulated industries and for organizations subject to board-level AI governance requirements. Each decision can be reconstructed and explained in business terms, not in model-internal representations.

8.4 Scaling Across Business Units

A common concern is whether the Intelligence Warehouse must be rebuilt for each business unit or geography. In practice, Layers 1 and 2 share significant structural commonality across business units (product hierarchies and metric definitions tend to be enterprise-standard), while Layer 3 rules exhibit regional variations. Our deployment methodology handles this through a "core + delta" approach: a shared foundation is built once, and business-unit-specific rules are layered incrementally. Second and third deployments within an enterprise typically require 40–60% less effort than the first.

9. Limitations and Future Work

Several limitations of the current architecture merit acknowledgment. First, the 98% accuracy figure reflects deployments in FMCG and organized retail—domains with relatively well-structured decision logic. Extension to domains with more ambiguous or rapidly evolving decision rules (e.g., strategic pricing in competitive markets) may yield different accuracy profiles. Second, the architecture currently handles deterministic and tiered decision rules well, but does not natively represent probabilistic or optimization-based decision logic. Third, while the 4-week implementation timeline holds for organizations with reasonably mature data infrastructure, organizations with fragmented or poorly documented systems may require longer data preparation phases.

Future work includes extending the architecture to support probabilistic decision rules, developing automated methods for detecting rule staleness and recommending updates, exploring federated Intelligence Warehouse configurations for multi-enterprise supply chain decisions, and formal benchmarking against a broader set of industry verticals.

10. Conclusion

The Intelligence Warehouse addresses what we believe is the central bottleneck in enterprise AI deployment: the absence of structured, queryable representations of institutional business logic at inference time. By separating business knowledge into three distinct layers—ontology, metrics, and rules—and implementing them as a curated knowledge graph, the architecture enables AI decision agents to produce grounded, policy-compliant, auditable decisions with 98% accuracy in production enterprise environments.

The bottom line: Data warehouses tell you what happened. BI tools tell you why. The Intelligence Warehouse tells AI agents *what to do about it, who should do it, and what exceptions to watch for*. The result is agents that don't just sound knowledgeable—but actually are.

As enterprises move from AI experimentation to AI-driven operations, the question is not whether foundation models are capable enough—they are. The question is whether the knowledge architectures surrounding them are adequate. The Intelligence Warehouse provides one answer: a practical, deployable, and proven architecture for bridging the gap between enterprise data and enterprise decisions.

Acknowledgments

We thank the implementation teams at Questt AI for their contributions to the production deployments described in this paper. We are grateful to our enterprise partners across FMCG and retail who provided access to their decision processes and validated our results. The cognitive neuroscience framing benefited from discussions with colleagues in the cognitive science research community.

References

Anderson, J. R. (1983). *The Architecture of Cognition*. Harvard University Press.

Chase, W. G., & Simon, H. A. (1973). Perception in chess. *Cognitive Psychology*, 4(1), 55–81.

Ericsson, K. A., & Kintsch, W. (1995). Long-term working memory. *Psychological Review*, 102(2), 211–245.

Gartner. (2025). *Hype Cycle for Artificial Intelligence*. Gartner Research.

Gobet, F., & Simon, H. A. (1996). Templates in chess memory: A mechanism for recalling several boards. *Cognitive Psychology*, 31(1), 1–40.

McKinsey & Company. (2025). *The State of AI in 2025: Moving from Pilots to Production*. McKinsey Global Institute.

Norman, G. R., & Eva, K. W. (2010). Diagnostic error and clinical reasoning. *Medical Education*, 44(1), 94–100.

Rumelhart, D. E., & Norman, D. A. (1978). Accretion, tuning, and restructuring: Three modes of learning. In J. W. Cotton & R. Klatzky (Eds.), *Semantic Factors in Cognition*. Erlbaum.

Squire, L. R. (2004). Memory systems of the brain: A brief history and current perspective. *Neurobiology of Learning and Memory*, 82(3), 171–177.

Tulving, E. (1985). Memory and consciousness. *Canadian Psychology*, 26(1), 1–12.

Tulving, E., & Thomson, D. M. (1973). Encoding specificity and retrieval processes in episodic memory. *Psychological Review*, 80(5), 352–373.

About the Authors

Akhil Singh is CEO and Co-founder of Questt AI. His work focuses on applying cognitive architecture principles to enterprise AI decision systems.

Nidhi Prabhu is Co-founder of Questt AI, leading the development of knowledge graph architectures and enterprise AI platforms for Fortune 500 clients.

questt.com · contact@questt.com