ARTICLE

Lost in Translation:

Why AI Agents Cannot Scale Without Semantic Foundations

June 2026

By Andreas Rindler, Kaj Burchardi, Tim Pieters, Darshana Thakker, Veronika Devedzhieva, Andre Pereira Correa, Ronni Gallo, Gaetan De Backer

BCG PLATINION

WHO SHOULD READ THIS - AND WHY

For the CEO / COO: AI agents are being sold as a productivity revolution. They can be - but scaling them across your enterprise requires infrastructure most organizations have not yet built. This article explains why initiatives stall after the pilot, and what structural investment separates the enterprises that will compound advantage from those that will keep rebuilding from scratch.

For the CIO / CTO: The bottleneck is not the model. It is a missing architectural layer - a governed, machine-readable representation of your business's own meaning. Sections 02, 03, and 07 map the gap precisely and lay out a maturity journey you can act on now.

For the CDO / Chief Data Officer: Your data platform is necessary but not sufficient. The semantic layer described in Sections 03 and 07 is the bridge between what you have built and what AI agents actually need to reason reliably.

For the Chief Risk / Compliance Officer: Sections 04 and 05 address the auditability problem directly. As agents move from answering questions to taking actions, the ability to trace outputs to governed data and explicit logic is not a nice-to-have - it is a regulatory and operational requirement.

Enterprises are deploying AI at pace. Models and technology are capable. Yet scaling stalls - repeatedly and expensively. The constraint is not intelligence. It is meaning.
Most organizations have not given their AI systems a consistent, machine-readable understanding of their own business language and knowledge. The semantics exist - embedded in dashboards, pipelines, and the intuition of domain experts - but they have never been externalized, formalized, or made authoritative. Until these are established, every new agent rediscovers and overcomes the same fragmentation from scratch.

What this article covers

THE PROBLEM
Why AI Scaling keeps stalling

01 AI doesn't stall at the model. It stalls at the architecture.
Business definitions are implicit. Every new use case rebuilds the same interpretive work from scratch.

02 Three layers matured. One did not.
Data platforms, integration layers, systems of record — all mature. Shared business semantics: absent.

THE SOLUTION
What a semantic layer changes

03 Meaning, made machine-readable.
Definitions declared once, referenced everywhere. Ontologies encode relationships. Knowledge graphs operationalize them.

04 Retrieval is not enough for agents that must reason.
RAG cannot guarantee consistency, support multi-hop reasoning, or produce auditable outputs. Semantic grounding is the prerequisite.

WHAT TO DO
Building and scaling the layer

05 The pattern holds across industries.
Insurance, regulatory intelligence, manufacturing — semantic structure compounds. Each new dataset enriches what came before.

06 Agents need a model of the enterprise - not just memory.
None of this can be derived from a document corpus. The semantic layer is what makes reliable agentic reasoning possible.

07 This is a business commitment, expressed in architecture.
The semantic layer is not built in one step. Each stage unlocks a new class of AI capability — and compounds the value of everything that came before.

AI doesn't stall at the model. It stalls at the architecture.

Across sectors, the pattern is consistent. An AI use case is piloted, results impress, and the enterprise moves to scale. Then things slow down. The summarization tool that worked for one team produces different outputs for another. The procurement agent approves a supplier, missing a financial distress signal that any analyst would catch. The customer service assistant resolves a ticket but does not update the billing system, because the customer records don't match between systems.

The cost is not just operational friction — it is value destroyed. Studies consistently show that the majority of AI delivery effort goes to reconstructing meaning and reconciling definitions rather than building new capability, a dynamic that compounds with every new initiative. Organisations that have addressed it report measurable gains — in some cases tripling AI accuracy on the same data, or compressing months of manual work to minutes.

The problem is not the model. It is that the enterprise has not given the model a coherent picture of itself. Business definitions are buried in systems and processes rather than made explicit: what a customer means in one system versus another, how a product links to the regulation it falls under. Each new AI initiative starts by redoing this interpretive work from scratch - spending the majority of effort reconstructing meaning rather than building new capability.

The exhibit below maps this divide. Above the waterline sits what organizations invest in and what boards see: models, platforms, deployed use cases. Below it lies what actually blocks scaling - business definitions recreated from scratch without shared logic, integration complexity that grows with every new project. The visible investment is real. The constraint is what it conceals.

The Semantic Debt Iceberg

Boards see the AI investment, what blocks scaling lies beneath the surface

Visible – what boards see

- AI pilots that impressed in the demo
- Investment in cloud, platforms and models
- Deployed use cases across functions

VISIBLE
AI INVESTMENT

HIDDEN
SEMANTIC DEBT

Hidden – the true constraint on scaling

- AI gives different answers to the same question, eroding trust
- Every new use case costs as much as the first
- Initiatives stall after pilot and never reach scale
- Months of effort spent rebuilding meaning

The question for business and technology leaders is not whether AI is mature enough to scale. It is whether the enterprise architecture is structured well enough to support it.

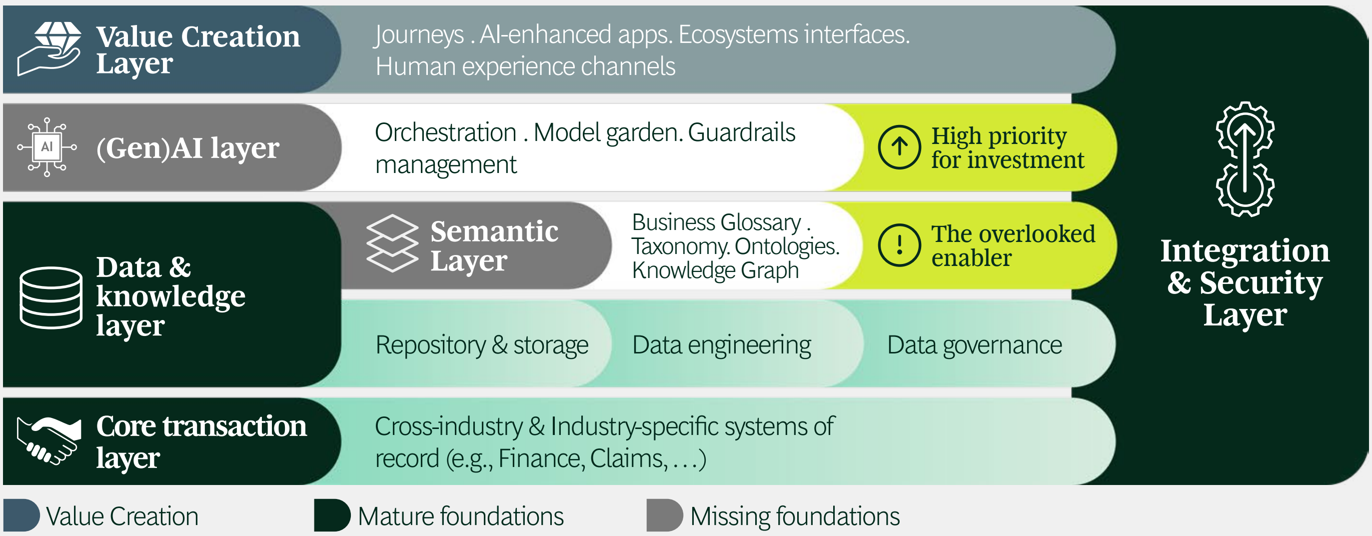
Three layers that matured. One that did not.

Most large enterprises have built out three architectural layers. Systems of record execute transactions and maintain state. Data platforms ingest, transform, store and distribute. Integration layers expose APIs and events. Each has matured considerably. What has not matured - and in most enterprises does not formally exist as a governed, machine-readable capability - is a layer that sits between these systems and the agents that consume them: a consistent, machine-readable representation of what the data actually means..

The exhibit below shows where the semantic layer sits in the enterprise stack. Three layers - core transaction systems, data and knowledge infrastructure, and the integration and security layer - have received sustained investment and have known patterns, while execution is uneven but in the right direction. The (Gen)AI layer above them is where organizations are now directing attention. What the diagram makes visible is the gap: the semantic layer, sitting between the data foundation and the AI consumers above it, remains the overlooked enabler. Without it, the layers below cannot be made legible and operable to the agents above.

In most architectures today, a core business term - revenue, customer, risk exposure - exists in multiple places simultaneously: in a reporting dashboard, a process definition, and the expertise of a domain specialist who has been with the company for fifteen years. Each definition is locally valid. None is authoritative. Human users navigate this through experience. AI systems cannot. Without explicit, consistent representations, semantic fragmentation becomes output variability - and output variability destroys trust.

Semantic layer is the overlooked enabler that essential for AI Value Creation



Three architectural layers have matured: systems of record, data platforms, and integration layers. The fourth layer - shared business semantics - remains absent in most enterprises. That absence is why AI scaling breaks.

03 - THE SEMANTIC LAYER

Meaning, made machine-readable.

The semantic layer is not a single technology — it is a capability that spans business glossaries, taxonomies, metric stores, canonical data models, ontologies, and knowledge graphs. The right depth depends on the use case. This article focuses on the higher end of that spectrum, where the complexity of enterprise AI reasoning demands formal relationship modelling.

The semantic layer is an architectural commitment: to externalize business meaning from the systems that currently embed it implicitly, and make that meaning available - consistently - to every system, application, and agent that needs it. It is also the governed access plane through which agents discover, navigate, and query enterprise data in business terms.

STANDARDIZED DEFINITIONS

Core entities - customer, product, contract, transaction - are defined once and referenced everywhere. Metrics are computed from shared logic, not reconstructed independently by each team. Metric consistency - through metric stores or SQL-first semantic layers - establishes this consistency for reporting and analytics. This is a meaningful step forward. Where it stops short is in encoding the relationships between entities, which more complex AI use cases require. This eliminates the drift that occurs when definitions evolve separately across systems, and reduces the interpretive overhead that consumes so much AI delivery effort.

ONTOLOGIES: FORMALIZING RELATIONSHIPS

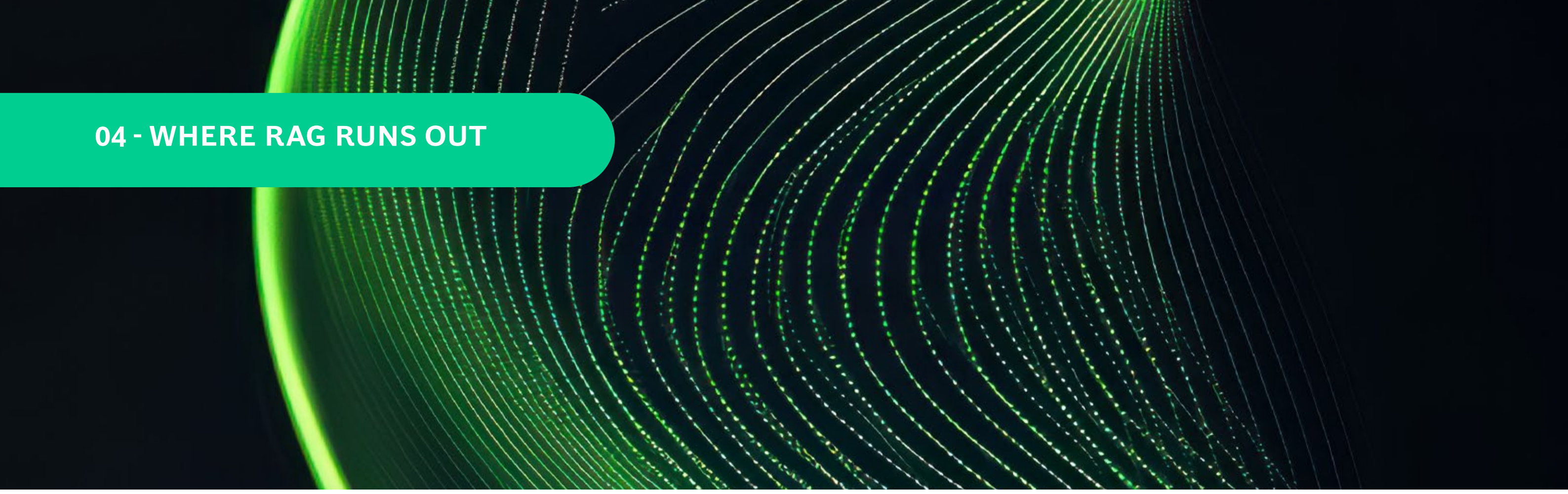
In complex domains, standardized definitions alone are insufficient. The relationships between entities must be formally encoded - not inferred through joins but explicitly modeled. An ontology defines the structure of a domain: the entities that exist, the properties that describe them, and the relationships that connect them. In a complex business domain, this means modeling not just the core entity, but all the relationships that determine its meaning: the customer and the products they hold, the obligations those products create, the risk they carry, and the regulation that governs them. The distinction from a canonical model is fundamental: master data management defines what an entity is. An ontology defines how entities relate to one another - and what can be inferred from those relationships.

KNOWLEDGE GRAPHS: DYNAMIC NAVIGATION

Ontologies are operationalized through knowledge graphs, allowing systems to traverse relationships dynamically rather than relying on predefined queries. A compliance agent can follow a regulatory obligation from its jurisdiction of origin to the products it applies to, to the processes it constrains, to the evidence that demonstrates compliance - in a single reasoning flow. Knowledge graphs do not replace relational systems. They complement them, capturing complex, many-to-many relationships that tabular structures cannot efficiently represent.

Critically, the semantic layer must also serve as the governed action plane: the interface through which agents connect to enterprise systems to execute, not merely retrieve. An agent that can reason over a customer's risk profile but cannot trigger a workflow, update a record, or initiate a process is an advisor, not an operator. The semantic layer, when connected to governed action interfaces, is what closes that gap.

Ontologies and knowledge graphs are not modeling exercises for their own sake. They are the mechanism by which the enterprise becomes legible to the agents that need to act on its behalf - precisely, reliably, and at scale.



04 - WHERE RAG RUNS OUT

Retrieval is not enough for agents that must reason.

The dominant approach to grounding AI in enterprise context today is Retrieval-Augmented Generation: retrieve relevant documents through similarity search, inject them as context, generate a response. The premise is sound: retrieve relevant passages and let the model answer from them. For simple questions, it works. For questions that require following a chain of logic - connecting a customer to their obligations, to the regulatory thresholds they must meet - it does not. The agent retrieves fragments. It cannot follow the connections between them. It produces different answers depending on what happens to surface. And in regulated environments, it cannot explain how it got there.

FOUR SPECIFIC FAILURE MODES DEFINE WHERE RAG RUNS OUT

01 Documents do not encode business logic - a policy document describes an obligation, but not the entity it applies to, the conditions that trigger it, or the process that must execute in response.

02 Retrieval does not guarantee consistency - two agents handling related tasks may receive different chunks, leading to contradictory outputs from the same model.

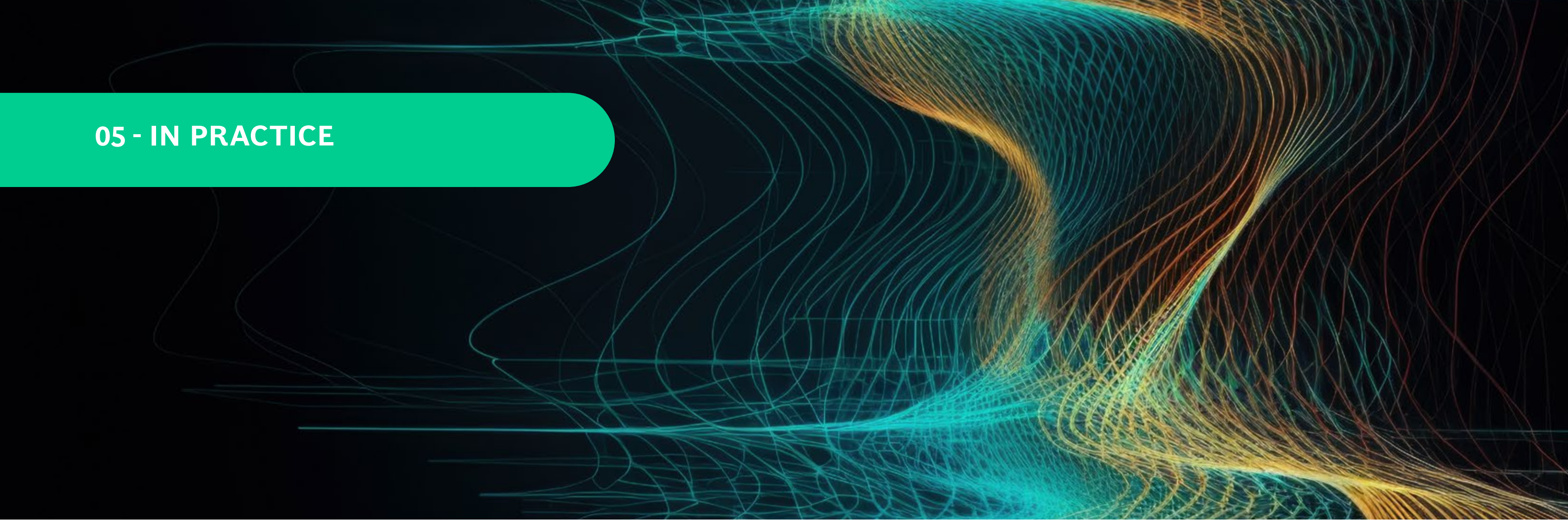
03 Multi-hop reasoning is not naturally supported - linking a customer to a product to a risk category to a regulatory threshold requires structured traversal, not document similarity.

04 Outputs are difficult to audit - with RAG, the reasoning chain passes through unstructured document chunks with no traceable path back to governed, authoritative data. In regulated industries, that is not a sufficient basis for consequential decisions.

WHAT SEMANTIC GROUNDING CHANGES

When an agent is grounded in an explicit ontology, these failure modes are structurally addressed. Tasks decompose into structured steps using domain relationships. Inferences are constrained - the agent cannot hallucinate relationships that have not been encoded. Outputs trace back to specific entities, relationships, and source data, because each inference step maps to an explicit, governed relationship. And agents can act, not just answer, because the semantic layer connects to the governed action interfaces of the enterprise. This does not displace RAG - unstructured content still benefits from retrieval - but the semantic layer is what enables the agent to know which type of source each question demands, and to apply the right approach to each.

RAG is a powerful approach for knowledge retrieval. It is not a foundation for enterprise reasoning. For agents that must decompose tasks, traverse relationships, and produce auditable outputs, semantic grounding is the prerequisite - not a supplement.



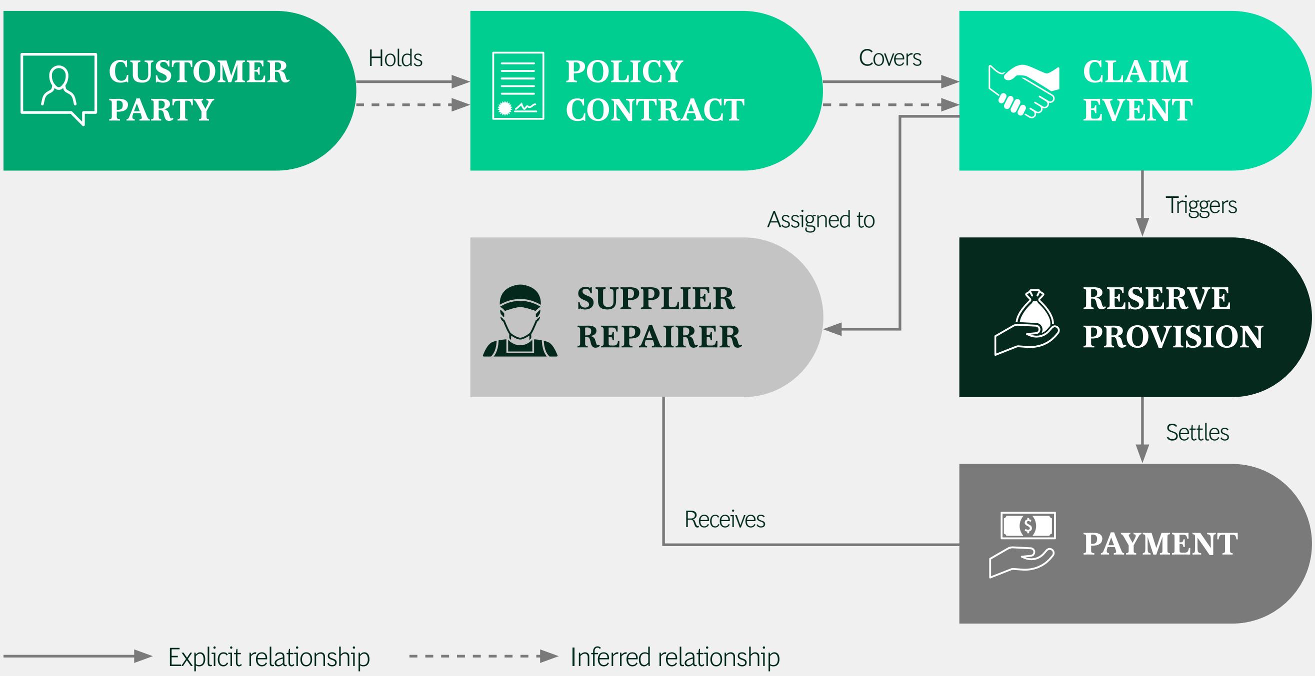
The pattern holds across industries.

Insurance: From Claims to Connected Risk

The word ‘claim’ means different things in different systems - gross versus net, across product lines, at different processing stages. For AI systems, that ambiguity propagates. When claims, policies, parties, and reserves are modeled in an explicit ontology, a claims assistant answers consistently regardless of which system it queries. A fraud detection agent identifies networks of connected parties - shared addresses, bank accounts, or repair histories - invisible to any model operating on isolated records.

The value a knowledge graph brings is measurable. A peer-reviewed benchmark using an enterprise insurance database found that GPT-4 without a knowledge graph answered only 16% of business questions correctly. Adding a knowledge graph representation of the same data raised accuracy to 54% - a threefold improvement on the same model, the same data, and the same questions.

Insurance knowledge graph: entities, explicit relationships, and inferred connections



INFORMATION SERVICES: WHERE STRUCTURE IS THE PRODUCT

For businesses whose product is organised knowledge - regulatory intelligence, consumer insights, market data, scientific content - the semantic layer is not just enabling infrastructure. It is the moat. The raw inputs (regulatory texts, survey responses, panel data, public filings) are often commoditised or publicly available. The defensible value sits in how those inputs are structured, related, and made queryable. As clients increasingly expect to consume this content through AI agents rather than dashboards or PDFs, the depth of the underlying ontology becomes the differentiator.

Regulatory intelligence: Most regulatory content is publicly available. The difficulty is structuring and relating it at scale: obligations, conditions, applicability criteria, effective dates, amendment history. Without this, downstream use cases - applicability assessment, gap analysis, control mapping - require extensive manual intervention. **In one global regulatory intelligence provider, applicability screening for a new client facility traditionally took months and ~300 manual screening questions; modelling the same logic on an ontology with typed entity relationships compresses that to a 10-minute conversation, with applicability inferred dynamically rather than reconstructed by hand.** Organizations that have built regulatory ontologies report a compounding effect: as new regulations are ingested, they are automatically enriched, linked to existing obligations, and immediately available to all downstream use cases.

Consumer and brand intelligence. The same pattern holds for businesses built on survey, panel, and behavioural data. Brand equity, advertising effectiveness, and shopper studies generate enormous volumes of structured and unstructured signal - across markets, categories, time periods, and channels - but the business logic, metric definitions, and proprietary IP that give that data meaning are never codified. They live in analysts' heads, in one-off scripts, and in project documentation that doesn't survive to the next study. We have seen fundamental metrics existing in materially different forms across studies: different scale types, different base populations, different calculation logic - **contributing to roughly half of total project delivery time being spent repeating work already done on the last study.** Once the connective tissue is encoded once, every new study, every new market, and every new agent benefits from it - turning a project-based research engine into a compounding data asset.

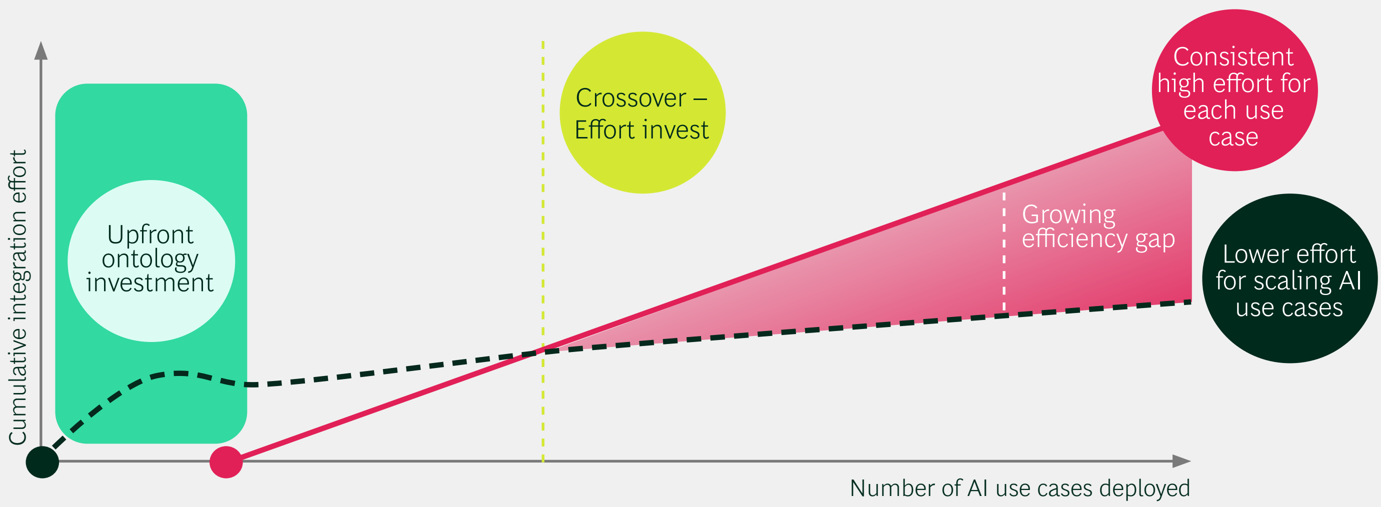
MANUFACTURING: TRACEABLE INTELLIGENCE

Quality management, production scheduling, maintenance, and supplier data sit in separate platforms. Without a semantic layer, each AI initiative rebuilds the connections between them from scratch. With one, an agent traverses the full provenance of a quality event - from affected batch to production run, raw material lot, supplier, and concurrent batches using the same input - in a single auditable reasoning chain.

> The pattern is consistent: organizations that invest in explicit semantic structure see compounding returns. Each new dataset enriches the graph. Each new use case benefits from the relationships already encoded. The architecture scales. The alternatives do not.

The Compounding Returns of a Semantic Layer

An upfront investment in ontology is required – but integration effort then flattens while the traditional approach compounds upward with every new use case



Ontology-driven – upfront investment, then effort flattens as definitions are reused
Traditional – each new use case compounds integrations effort

Agents need more than memory. They need a Model of the Enterprise.

The evolution from AI assistants to AI agents changes the requirements fundamentally. An assistant answers questions. An agent plans, decides, coordinates, and acts - across time, across systems, and where errors have real consequences. To do this reliably, an agent needs to know what entities exist, how they relate, what actions are permitted, and what the downstream consequences of those actions are. None of this can be reliably derived from a document corpus.

The semantic layer is not just one input to the agent's reasoning - it is the grounding layer that makes reliable reasoning possible. Without it, agents hallucinate relationships, produce inconsistent outputs, and cannot be audited. With it, agents operate within governed workflows and produce outputs traceable to the data and logic that generated them. The enterprises that lead the next phase of AI adoption will be those that have built the semantic infrastructure to deploy models reliably, accountably, and at scale.

A critical capability the semantic layer provides is governed orchestration: the intelligence to route each question to the right type of source. When an agent evaluates a loan applicant's creditworthiness, it must know to query a credit score database or mortgage rate API with exact arithmetic - not ask a language model to reason those numbers from unstructured market commentary. Equally, it must know when to apply language model reasoning where it genuinely excels: interpreting macroeconomic signals, assessing market sentiment, synthesising qualitative context. The semantic layer encodes this distinction - defining which questions demand deterministic precision and which benefit from probabilistic reasoning. Without it, agents default to using language models for everything, producing confident answers grounded in the wrong type of evidence.

The semantic layer maturity model: your journey from retrieval to reasoning.

This is the ‘how’ of the article. Building a semantic layer is not a greenfield undertaking, and it does not require a big-bang transformation. In most enterprises, partial semantic structure already exists - in taxonomies, data dictionaries, BI metric definitions, and MDM systems. The maturity model below maps where you are today and what the path forward looks like. Most organizations will recognize themselves in the early stages. The goal is not to jump directly to the top - it is to take the next step deliberately, and to understand what becomes possible at each level.

- **Start where you already are**. Identify which maturity level your highest-priority AI use cases sit at today. Most enterprises have the foundations for metric consistency or a canonical model already - in their BI platforms, MDM systems, or data dictionaries. Build on these rather than starting from scratch.

- **Design the ontology in two tiers**. A shared core defines the entities and relationships common across use cases - customer, product, contract, obligation. Use-case specific tiers extend the core with the domain detail each application needs: a procurement use case adds supplier structures and compliance requirements; a quality management use case adds production runs, material lots, and failure modes.

- **Move up one level at a time.** Each level in the maturity model unlocks a new class of AI capability. Metric consistency enables reliable analytics agents. A canonical model enables cross-domain AI use cases. Semantic reasoning is what separates agents that answer from agents that act - reliably, traceably, at scale. Define which level your most critical use cases demand, and close that gap first. Not every use case needs to reach the same level - a reporting agent may operate reliably at Metric Consistency, while a compliance or fraud detection use case demands Semantic Reasoning. The goal is the right level of maturity for each use case, not a uniform standard across all of them.

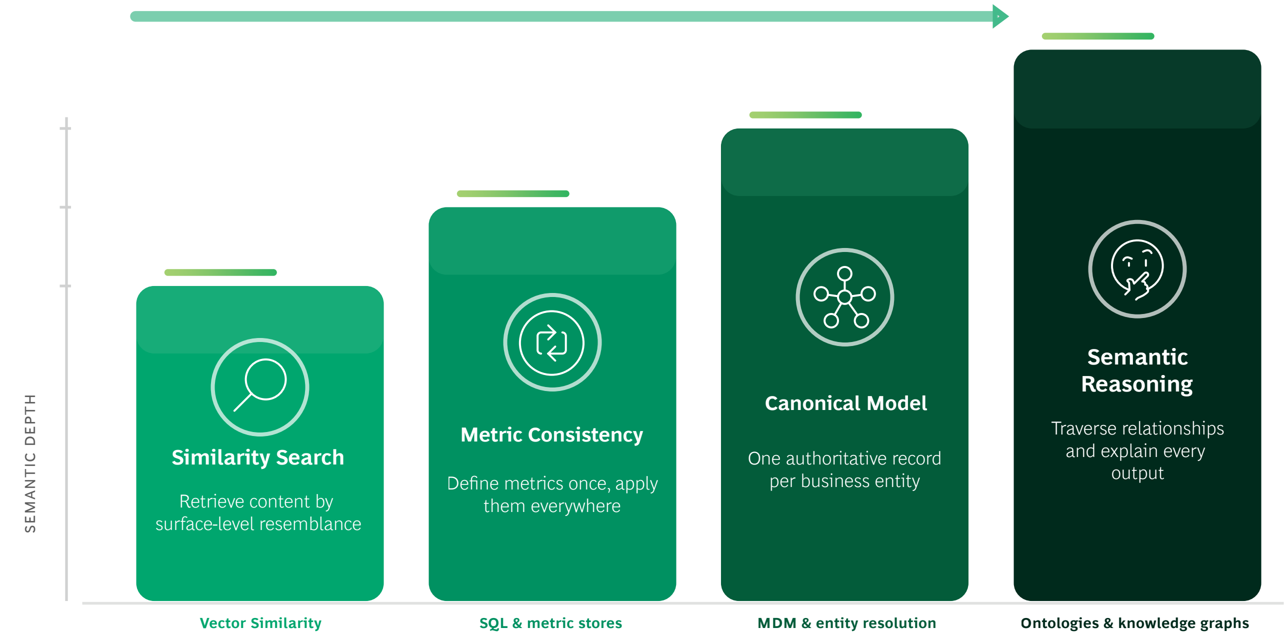
- **Make it the AI-ready data access plane.** At every maturity level, the semantic layer must be a governed, queryable interface - not a documentation artifact. It should be versioned, connected to the data platforms your agents consume, and accessible through the action interfaces that enable agents to act, not just answer. In highly distributed landscapes, a federated approach can provide this interface virtually.

- **Design for compounding returns.** The value of the semantic layer increases non-linearly as you progress. Each new entity enriches the graph. Each new relationship opens new reasoning paths. Each new use case benefits from everything encoded before it. Organizations that invest early compound that advantage. Those that wait face a competitive gap - not a stable equilibrium.

- **The organizational reality.** Ontology design is a political exercise as much as a technical one. Defining shared concepts means reconciling assumptions that different teams have held for years - what “customer” means in sales versus finance versus legal. Organisations that hand this to IT alone will produce models that business teams ignore. The right people are also genuinely scarce: knowledge engineering and ontology design remain niche disciplines, and most enterprises will need a mix of targeted hires, upskilling, and external support to get started. Begin early - scarcity is a reason for urgency, not delay.

From Retrieval to Reasoning

A maturity progression — each stage encodes a deeper layer of business meaning and unlocks a new class of AI capability



Governance is what separates a semantic layer that scales from one that stalls - and getting ownership right is the single most critical decision. The semantic layer is not an IT asset. Only business leaders - the COO and heads of functions - have the authority and domain knowledge to define what entities mean, how relationships should be modeled, and what constraints must hold across the organisation. Technology teams build and maintain the infrastructure; business leaders own the definitions and are accountable for their accuracy as the business evolves. Framed as an architecture problem, this work drifts into IT for its own sake. Framed as a business leadership commitment, it becomes the foundation on which every AI initiative compounds.

Semantic Layer Maturity – 4 different stages

	Similarity Search	Metric Consistency	Canonical Model	Semantic Reasoning
WHAT IT IS	Retrieves content by embedding similarity across unstructured data. No explicit business semantics-operates on surface-level resemblance rather than meaning	Encodes business meaning in queries, metrics, and views. A SQL-first semantic layer that enforces KPI consistency across reporting and analytics	Defines canonical business objects-Customer, Product, Policy-with consistent identifiers and attributes across systems via master data management.	Models entities and relationships explicitly using ontologies and knowledge graphs. Enables multi-hop reasoning, explainable outputs, and inference across complex domain logic.
BEST FOR	Copilots, document search, and prototyping AI use cases where speed to deploy matters more than precision	Reporting consistency, text-to-SQL, and conversational analytics where metric governance is the primary requirement	Cross-system AI use cases and customer-or product-centric domains where a single authoritative view of core objects is the bottleneck.	Complex reasoning tasks - fraud detection, compliance, supply chain, root cause analysis-where AI must traverse relationships and produce auditable outputs.
ATTENTION POINTS	No explicit meaning-outputs are inconsistent and cannot be audited. Often mistaken for a semantic layer; it is not	Relationships between entities remain implicit. Works well for analytics but breaks down quickly for complex AI reasoning use cases	Relationship logic stays external to the model. Significant MDM governance overhead; limited multi-hop reasoning without an ontology layer on top.	Requires active business sponsorship and governance — defining who owns each term is as important as the technology. Steeper adoption curve than earlier stages.

The architectural decision in front of every enterprise

The next phase of enterprise AI will not be defined by improvements in models. It will be defined by how well organizations represent their own business to those models. The enterprises that win will be those that have externalized their meaning - turning implicit knowledge into governed, machine-readable structure that every agent can rely on. They will compound that advantage with every new dataset, every new use case, every new capability built on top. Those that do not will keep rebuilding the same mappings for each new initiative, spending the majority of their AI budget on reconstruction rather than progress. The constraint is not the model. Building it now is not a data management best practice. It is the infrastructure decision that determines whether AI can scale at all - and whether your organization leads that transition or watches others do it.

Before your next AI investment, ask three questions.

01 Where in your enterprise does the definition of a core entity - customer, product, risk, obligation - differ between systems, and which team owns the authoritative answer?

02 How much time of your last three AI initiatives was spent reconstructing data relationships and business logic that already existed somewhere in the organisation?

03 And which of your planned agent use cases requires multi-hop reasoning - linking entities across domains, tracing a chain of logic to a governed conclusion - and does your current architecture actually support that?

If any of these questions is difficult to answer, the semantic layer is not a future investment. It is the current constraint.

About the Authors

Andreas Rindler
Platinion Managing Director
London
rindler.andreas@bcgplatinion.com

Kaj Burchardi
Platinion Managing Director
Amsterdam
Burchardi.Kaj@bcgplatinion.com

Tim Pieters
Platinion Director
Brussels
Pieters.Tim@bcgplatinion.com

Darshana Thakker
Platinion IT Architecture Director
London
Thakker.Darshana@bcgplatinion.com

Veronika Devedzhieva
Platinion Principal IT Architect
Brussels
Devedzhieva.Veronika@bcgplatinion.com

Andre Pereira Correa
Platinion Principal IT Architect
Amsterdam
PereiraCorrea.Andre@bcgplatinion.com

Ronni Gallo
Platinion Principal, Tech Advisory & Delivery
London
Gallo.Ronni@bcgplatinion.com

Gaëtan De Backer
Platinion Senior IT Architect
Brussels
DeBacker.Gaetan@bcgplatinion.com

This article was developed by BCG Platinion.
The authors draw on client work across financial services, insurance, manufacturing, and information services.

Sources

BCG: Laying the Tech Foundation for Gen AI Success

BCG: Building effective enterprise agents

BCG case experience in Insurance, Information Services & Manufacturing

Sequeda, J.F., Allemang, D. & Jacob, B. (2024).
A Benchmark to Understand the Role of Knowledge Graphs on Large Language Model's Accuracy for Question Answering on Enterprise SQL Databases.

ABOUT BCG PLATINION

As a part of Boston Consulting Group (BCG), BCG Platinion provides consulting services in IT architecture, AI platforms, and the development and implementation of advanced technology solutions that fuel critical transformation and create next-generation business models. Today, our presence spans the globe with offices in Europe, North and South America, Africa, and Asia Pacific.

BCG
PLATINION