

Semantic Layers Were Built for Dashboards. AI Agents Need Something Else.

Why the AI era requires continuous semantic intelligence, and where Colrows, ThoughtSpot, Timbr.ai, and Databricks Genie fit

Prepared by Colrows (Alterbasics Technologies Pvt. Ltd.) · July 2026

Two eras, two sets of requirements

The semantic layer is not a new idea. It was invented for business intelligence, and for that job it was designed well. A few hundred analysts asked a fairly stable set of questions. Dashboards changed slowly. Business definitions were revisited quarterly, not weekly. In that world, a team of engineers could author the semantic model by hand and keep it current, because the rate of change was human-scale.

THE BI ERA

Semantic layers built for analysts

- Hundreds of analysts
- Dashboards change slowly
- Definitions change rarely
- Manual curation is affordable

THE AI AGENT ERA

Semantic layers serving agents

- Thousands of agents
- Agents generate dynamic, contextual questions beyond predefined paths
- Definitions change constantly
- Manual curation cannot keep up

The operating assumptions behind semantic layers changed completely when AI agents became the primary consumer of enterprise data.

AI changed the arithmetic. A semantic layer no longer serves hundreds of analysts clicking through curated dashboards. It serves thousands of AI agents that generate queries continuously, reason over enterprise knowledge, plan multi-step workflows, and act on what they conclude. Agents do not ask the same question twice, and they do not stay inside the paths someone modeled in advance.

AI did not simply increase demand for semantic layers. It changed the requirements for them. Earlier generations delivered semantic models. The AI era requires continuous semantic intelligence.

The evidence for why this matters is now independent of any vendor. On Spider 2.0, a benchmark built from realistic enterprise databases, frontier models answered as few as 6 to 17 percent of questions correctly when pointed straight at the warehouse. Working through a well-modeled semantic layer, the same class of models reached 98 to 100 percent (dbt Labs, April 2026). Gartner reached the same conclusion in May 2026: without semantic context, AI agents are far more likely to hallucinate and produce unreliable results.

The real problem is not building the model

Most discussions of semantic layers focus on the build: how expressive the modeling language is, how elegantly metrics are defined. Building a semantic model is a substantial engineering effort. Keeping it continuously aligned with reality is the larger and more enduring operational challenge.

The greatest challenge in enterprise semantics is not creating a semantic model. It is keeping that model continuously synchronized with an organization that changes every day.

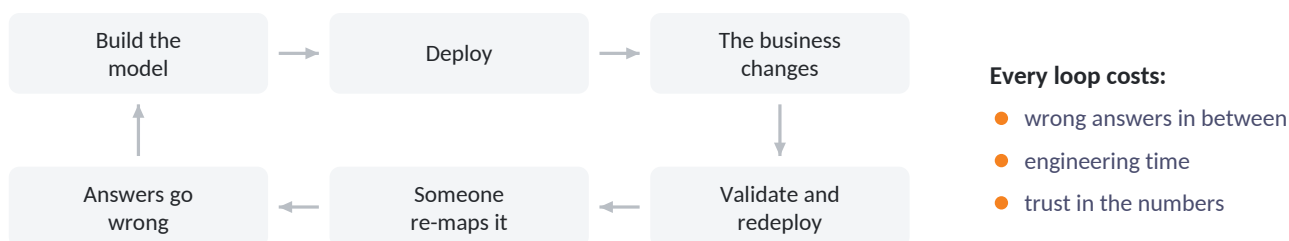
An enterprise changes constantly, and every change is a small tax on the semantic layer: schemas evolve, tables appear, columns are renamed, business definitions shift, acquisitions bring entire estates with their own vocabulary, new applications land, dashboards multiply, dbt projects are refactored, user behaviour moves, and governance policies tighten. Each one is minor on its own. Together, in a company of any size, they arrive faster than any team can absorb them.

This is why semantic maintenance grows harder as organizations grow, not easier. The model that was accurate at launch decays quietly, and the larger the enterprise, the faster it decays.

Why every earlier semantic layer becomes static

This is not a criticism of any product. It is the lifecycle their architecture assumes:

The lifecycle every hand-maintained semantic layer repeats



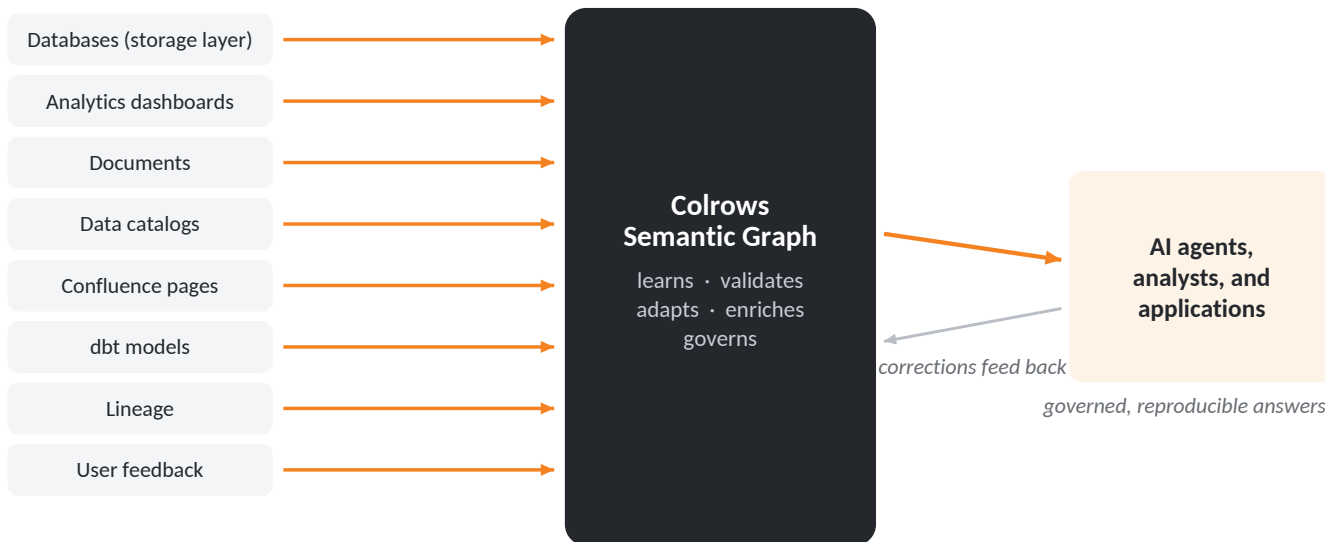
In the BI era, this loop was perfectly acceptable. A dashboard was wrong for a week, someone raised a ticket, an engineer re-mapped the model, and the business moved on. Applied to enterprise AI, the same loop becomes operationally unsustainable: agents run continuously, the window between a change and its correction is a window of confident wrong answers at machine speed, and the queue of changes grows faster than the team clearing it. The architecture is not failing. It is doing exactly what it was designed to do, under conditions it was never designed for.

Traditional semantic layers model the enterprise once. Colrows continuously observes the enterprise becoming something new.

A different architecture: semantics that emerge from how the business works

Colrows starts from a different premise. Enterprise semantics should not be a document that people write and then maintain. They should emerge continuously from how the organization actually uses its data, and they should be governed and provable at the moment of execution.

OPERATIONAL SIGNALS

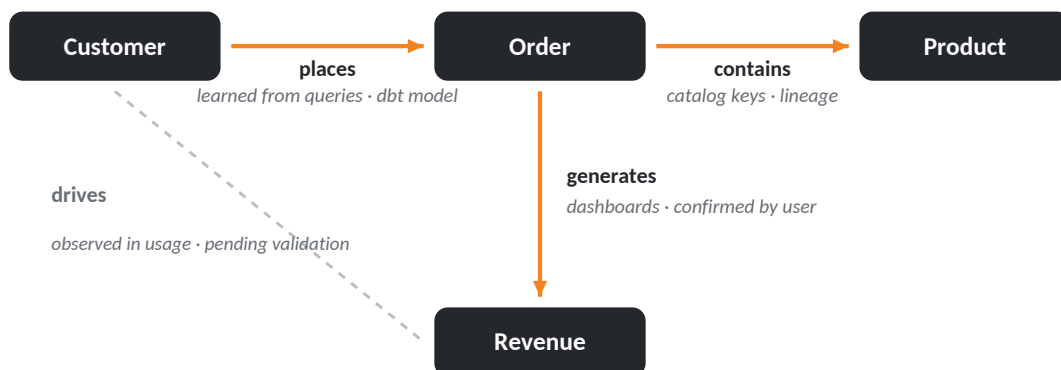


Colrows acquires semantic knowledge from operational signals continuously, then learns, validates, adapts, enriches, and governs its semantic graph as the business changes.

Colrows continuously acquires semantic knowledge from the sources an enterprise already has: **databases** provide the structural foundation of enterprise data; **analytics dashboards** show which definitions and metrics the business relies on; **documents** and **Confluence pages** carry the meaning written for humans; **data catalogs** supply governed metadata; **dbt models** and **lineage** encode the transformations the data team has already agreed on; and **user feedback** settles ambiguity. Manually authored metadata is one input among many, not the only source of truth.

Enterprise semantics should be continuously discovered and validated, not manually documented once and maintained forever.

What this creates over time is the real asset: an **enterprise semantic memory**. Not metadata, and not a model someone wrote, but a durable record of how the business actually thinks: how it measures, joins, filters, and governs information, with every definition carrying the evidence it was learned from and evidence of where each definition was learned from. A fragment of that memory looks like this:



A fragment of the Colrows semantic memory: entities and relationships carry the evidence they were learned from.

The result is not a static artifact. The Colrows semantic graph is a living system that learns from new signals, validates definitions against real usage, adapts as schemas and meanings move, enriches itself with relationships and join paths as it

observes them, and governs every answer it produces. This is what an Autonomous Semantic Layer means in practice, and it is why schema drift is a non-event rather than a project: drift is detected from the signals themselves and the graph repairs itself, consulting a human only where a genuine business decision is required.

The same property makes Colrows unusually comfortable with **legacy schemas**. Because meaning is learned from usage, lineage, and documentation rather than from tidy naming, decades-old systems with cryptic column names can stay exactly as they are. No renaming programme is needed before AI becomes useful.

Requirements that only exist in the AI agent era

Colrows was designed around requirements that did not exist when earlier semantic layers were architected: **continuous semantic adaptation**, because change is constant; **autonomous semantic acquisition**, because manual authoring cannot cover an enterprise; **enterprise-scale semantic maintenance**, because the model must stay true across thousands of assets; **deterministic execution**, so an agent's answer is compiled rather than sampled; **compile-time governance**, so access rules are enforced before a query runs, not after; and **reproducible reasoning**, so the same question yields the same governed answer when an auditor asks again. In the BI era, none of these were requirements. Agents made all six unavoidable.

From a layer to an operating system

This shift also changes what the category is. During the BI era, semantic layers sat between dashboards and databases: useful, but peripheral. During the AI era, semantic intelligence becomes the operating system through which every AI agent understands enterprise knowledge. Every agent that queries, reasons, plans, or acts does so through it. That is no longer a reporting convenience. It is core enterprise architecture.

Four architectural generations

The clearest way to compare the market is not feature by feature, but by the architectural philosophy each product represents. Each is well built for the assumptions it was designed around.

Architecture	Representative product	What it assumes
AI-generated query assistant	Databricks Genie	No lasting semantic model. An LLM writes SQL at query time from catalog metadata and curated hints. Fast to start; Databricks documents that the same question can return different answers.
Human-curated semantic model	ThoughtSpot	A governed model defined and verified by people. Accurate and well governed where the data is well modeled and the model is kept current by a team.
Ontology-driven semantic layer	Timbr.ai	A SQL knowledge graph with inheritance and inference across sources. Rich modelling; the ontology and its mappings are authored and maintained by people.
Continuous semantic intelligence	Colrows	A semantic graph that acquires meaning from operational signals, adapts as the business changes, and compiles every answer deterministically under compile-time governance.

Databricks Genie is the strongest choice for an estate standardized entirely on Databricks: nothing to install, governance inherited from Unity Catalog, no separate licence. It sees only Databricks, its SQL is generated probabilistically at query time, and it stays accurate through continuous human curation of instructions and examples.

ThoughtSpot brings a mature, human-verified semantic model and a strong security layer, and was named a Leader in the 2026 Gartner Magic Quadrant for Analytics and BI. Its architecture reflects the assumptions of the business intelligence era, where semantic models were primarily authored and maintained manually: third-party analyses typically budget half to one full-time person for model upkeep, and its own documentation has historically measured accuracy near 80 percent

on small, clean models and near 60 percent on complex ones.

Timbr.ai is a thoughtful ontology product with genuine depth: inheritance, inference, and graph traversal in standard SQL, across sources, without moving data. Founded in 2018, its architecture also reflects the assumptions of the BI era, where semantic models were authored and maintained by hand. Its mapping assistance is semi-automatic, and its published documentation describes no mechanism for detecting schema drift or semantic decay on its own. That is an architectural optimization for a different set of operating conditions, not a defect.

Earlier semantic layers help enterprises build semantic models. Colrows helps enterprises keep those models continuously aligned with reality.

Enterprise AI will not be limited by model intelligence.

It will be limited by semantic intelligence.

The organizations that succeed will not simply have semantic layers. They will have semantic systems that continuously learn, adapt, and govern enterprise knowledge as the business evolves.

Colrows was designed from the ground up for that future.

As foundation models converge in capability, continuous semantic intelligence becomes the enduring competitive advantage of the enterprise.

Every Colrows capability described here can be demonstrated live against your own schemas, including your oldest ones. To arrange a working session, write to engage@colrows.com.

Selected sources

Spider 2.0 enterprise benchmark: Lei et al., ICLR 2025, arXiv:2411.07763. Accessed July 15, 2026.

dbt Labs, Semantic Layer vs. Text-to-SQL: 2026 Benchmark Update, dbt Developer Blog, April 7, 2026.

data.world knowledge graph study, arXiv:2311.07509; BEAVER enterprise benchmark, arXiv:2409.02038; Cube paired benchmark, arXiv:2604.25149.

Gartner press release, May 11, 2026: Lack of Semantics Causes Inaccurate AI Agents and Wasted Spending.

Databricks Genie documentation (overview, best practices, talk-to-genie, troubleshooting), docs.databricks.com and learn.microsoft.com. Accessed July 15, 2026.

ThoughtSpot documentation (semantic-layer, search-sage, security, consumption-pricing), docs.thoughtspot.com; 2026 Gartner Magic Quadrant announcement, GlobeNewswire, July 1, 2026. Accessed July 15, 2026.

Timbr.ai documentation (platform, intro-timbr, data-mapper), docs.timbr.ai and timbr.ai; founding year per PitchBook company profile. Accessed July 15, 2026.

Statements about ThoughtSpot, Timbr.ai, and Databricks Genie reflect vendor documentation and public sources as of July 15, 2026 and may change as those products evolve. Product names and trademarks belong to their respective owners.