MERIT

# The Data Layer for Agents: How to Build Pipelines That Can Actually Feed Agentic AI Without Breaking Under the Load

Most agentic AI projects will fail in 2027. Not because the agents are wrong, but because the data layer underneath was never built to feed them.

**Authored by** Tharun Mathew
Head of Data & AI Solutions, Merit Data and Technology

CONTENTS

# Table of Contents

# EXECUTIVE SUMMARY

**SITUATION**

Agentic AI has stopped being a research category. By the end of 2026, Gartner expects 40% of enterprise applications to embed task-specific AI agents, up from less than 5% the year before. Roughly four in five enterprises have agents in production somewhere in the business, even if leadership cannot always say where. The boards have approved budgets. The pilots are running. The agents are out there, making decisions.

**COMPLICATION**

And yet Gartner also forecasts that more than 40% of agentic AI projects will be cancelled by the end of 2027. The framing in most boardroom conversations puts that failure at the door of the agent itself, or the model, or the use case. Twenty years of building data infrastructure for production AI workloads tells a different story. Agents do not fail because the model is weak. They fail because the data layer underneath them was built for a different consumer of data, and nobody noticed until the agent had been operating on stale, ungoverned, or incomplete context for long enough to do real damage.

**RESOLUTION**

The agentic deployments that will still be running in 2028 are not the ones with cleverer agents. They are the ones whose data layer can keep up. The shift from stateless request-response systems like copilots, where each interaction starts fresh and ends clean, to stateful continuous reasoning like agents, where context accumulates across sessions, decisions compound, and memory has to persist reliably between actions, is what makes the data layer the decisive variable. Real-time freshness, structured memory, semantic context, automated governance, observability for autonomous systems, and a cost engineering discipline that survives contact with production. Five things, all under the surface, none of which sell themselves as exciting on a slide. All of which determine whether the agentic AI strategy delivers anything in the next eighteen months.

## KEY FINDINGS

1. **The 40% cancellation forecast is a data infrastructure forecast.** Gartner cites cost, value, and risk as the reasons agentic projects will be cancelled by 2027. All three trace back to the same place: data layers that cannot deliver fresh, governed, contextually rich data fast enough or reliably enough to keep autonomous systems on track. Switching frameworks will not fix it. Switching models definitely will not.

2. **Agents do not consume data. They run on it.** Generative AI copilots wait for a human to type a question. Agents do not wait for anything. They run continuously, retrieve context autonomously, take actions across systems, and write back to memory while doing it. That is a fundamentally different relationship with the data layer, and most enterprises are still treating it as if it were the same as the last one.

3. **Batch is not a freshness problem. It is a correctness problem.** When a human analyst sees stale data on a dashboard, they notice and they ignore it. When an agent receives stale data, it acts on it. Inventory shown as available gets sold. Tickets shown as open get touched. Customer entitlements that have been revoked get honoured. Stale context becomes wrong actions, and the wrong actions accumulate before anyone has a reason to look.

4. **Memory has to be infrastructure. It cannot stay an afterthought.** The default approach to agent memory in most organisations is whatever the framework happens to provide. That works in a demo. It fails at production volume in ways that look like model degradation but are not. Production agents need episodic, semantic, and state memory layers, designed deliberately and governed properly. Bain has been making this point in print for the past year and most enterprises have still not acted on it.

5. **Governance for autonomous systems is a different discipline to governance for humans.** Deloitte's 2026 research found that only one in five companies has a mature governance model for autonomous AI agents. That gap is not closing on its own. Traditional governance assumes deliberate, sequential, reviewable activity. Agents move at machine speed across many systems at once. The audit trail traditional tools were designed to capture is not the audit trail an agentic deployment needs.

## KEY STATISTICS

**40%+**
oof agentic AI projects will be cancelled by end of 2027
(Gartner)

**40%**
of enterprise applications will embed AI agents by end of 2026
(Gartner)

**1 in 5**
companies has a mature governance model for autonomous AI agents
(Deloitte, 2026)

Section 1

# The Agentic AI Reality Check: Why Most Pilots Will Not Survive 2027

There is a version of the agentic AI story being told in earnings calls and analyst notes that does not match what is happening on the ground. The version in the headlines is about velocity. By the end of 2026, Gartner forecasts that 40% of enterprise applications will embed task-specific AI agents, up from less than 5% in 2025. Adoption surveys put roughly four in five enterprises somewhere on the agentic curve. Eighty-eight percent of executives are planning to increase AI budgets specifically because of agentic initiatives.[1][2][3] The version that gets discussed less often is the one Gartner published in the same season: more than 40% of those agentic projects will be cancelled by the end of 2027. The cited reasons are cost, value, and risk. The underlying reason, in most of the deployments I have seen, is something the cancellation note does not quite say out loud.

The deployments that get cancelled are not the ones where the agent failed to do anything useful. They are the ones where the agent did things, kept doing them, and then somebody started asking what exactly it had been doing for the last six months. McKinsey's 2026 trust research framed this as the difference between a system that says the wrong thing and a system that does the wrong thing. The first is annoying. The second is expensive, sometimes catastrophically so, and the audit trail is usually not there to reconstruct what went wrong.[4]

Deloitte's 2026 State of AI in the Enterprise survey, which covered 3,235 leaders across 24 countries, found only one in five companies with a mature governance model for autonomous agents. That number on its own is striking. Read alongside the four-in-five adoption figure, it describes a market that has deployed faster than it can govern. Two thirds of enterprises are running systems they cannot yet hold accountable.[5]

> **"Most agentic AI projects right now are early stage experiments or proof of concepts that are mostly driven by hype and are often misapplied, blinding organisations to the real cost and complexity of deploying AI agents at scale."**
>
> Gartner, 2025

## Agentic Failure Looks Different to What Came Before

The pilot-to-production gap that frustrated generative AI through 2024 and 2025 had one thing going for it. It was visible. Pilots underdelivered. Boards lost patience. Projects shut down. The cost was real but the failure was at least loud enough to act on.

Agentic failure does not behave like that. The agent keeps running. It keeps making decisions. It keeps writing back to memory and acting on what it finds there. The problem is not that the system stops working. It is that nobody can quite tell whether it is working or not, because the decisions are happening too fast, in too many places, against context nobody is auditing in real time. By the time someone asks a question that surfaces the issue, the agent has been acting on the wrong assumptions for weeks. Sometimes for months. The cost is already on the books. The exposure is already in production.

This is what makes the data layer the variable that actually matters. With copilots, a poor data foundation produces unhelpful answers that humans either fix or ignore. With agents, the same poor foundation produces autonomous actions that compound. A pricing agent does not provide recommendation on inventory. It will provide wrong pricing recommendation based on inventory levels or wrong data. It produces wrong prices, on real transactions, with real customers, while the team that would have caught it is busy elsewhere. A compliance agent operating on outdated policy data does not flag a question for a human reviewer. It approves decisions it should have escalated, and the only way to find out is to go looking, which most organisations do not do until something has already gone wrong.

## The Anatomy of the Gap

- **Stale context, treated as a freshness problem when it is a correctness problem.** When a human analyst sees stale inventory data on a dashboard, they pause, question it, and decide whether to act. When an agent receives the same stale data, it acts immediately and autonomously, and the wrong action has already been executed before any human has the chance to intervene. Inventory shown as available gets sold. Entitlements that have been revoked get honoured. The damage is already on the books by the time anyone looks. Most enterprise data layers were built for analytics workloads where a few hours of latency is fine. For agentic workloads, that latency is not a freshness problem. It is a correctness problem, and the cost of the wrong action compounds with every cycle the agent runs before the error is caught.

- **Memory left to whichever framework the developer happened to pick.** Bain's platform research is direct on this point: memory management has to be treated as a first-class infrastructure concern, not an application afterthought. Most enterprises have not made that shift. They have a vector database somewhere, agents writing to it, and no architectural view of how memory should be structured, governed, or expired. It works at small scale. The degradation as deployment grows is predictable and well-documented, and at the point it becomes visible the cost of fixing it is much higher than the cost of having designed for it would have been.[7]

- **Governance frameworks built for human pace, applied to machine pace.** Most enterprise governance was designed around the assumption that someone is making a deliberate decision, that someone else is reviewing it, and that there is time for both of those steps. Agents collapse that timeline to milliseconds. The governance contract has not been redesigned and the gap is showing up in audit findings and internal incidents, even where it has not yet shown up in regulatory enforcement.[5]

- **Cost curves that look fine in pilot and ugly in production.** Agents consume tokens, retrievals, and tool calls continuously. A single agent task can trigger dozens of LLM invocations and hundreds of vector lookups. At pilot scale, the bill is small enough that nobody pays much attention. At production scale, agentic workloads have a habit of becoming the largest line on the cloud bill, and finance teams who were not part of the early conversation suddenly are. Gartner's 2027 cancellation forecast cites cost escalation as one of the three primary reasons. It is not a small share of the cancellations.[1]

## The Compounding Problem with Pilot Drift

Every week an agentic deployment runs on inadequate infrastructure, two things happen. The first is that the cost of fixing the foundation goes up, because there is more in production now and more dependencies on top of it. The second is that the exposure accumulates quietly. Decisions get taken that nobody fully audits. Audit trails that should exist do not. Trust in the system, once it starts to slip across the business, takes a long time to rebuild. None of this shows up cleanly on a single line item, which is part of why it tends to keep going for longer than it should.

This is the trap the cancellation forecast actually describes. The investment is real. The agents are running. The cost is climbing. The returns are not arriving in any way that survives a serious return-on-investment review, because the architecture beneath the agents was never going to deliver them. By the time that becomes obvious, the organisation has spent eighteen months on the wrong problem.

Section 2

# What Agents Actually Demand from Data Infrastructure

"Agent-ready data" has become a vendor category in 2026, which is usually a reliable sign that the term is being used to mean too many different things at once. In my experience, the phrase is most useful when it is reduced to four specific questions an autonomous system needs the data layer to answer reliably. Three of them generative AI raised. The fourth is what changes when the consumer of data is no longer a model answering a person but an agent acting on its own.

## The Four Questions Agents Need Answered

**What is true right now?** An agent acting in real time has to reason on data that reflects the current state of the business. Inventory at this moment, not last night's snapshot. Open tickets right now, not the overnight extract. Customer entitlements as they currently stand. For a human reading a dashboard, a few hours of latency is rarely a problem. For an autonomous system making a decision on a live transaction, the gap between yesterday's state and now is exactly where the failure happens.

**What did the agent know when it made this decision?** When something goes wrong, the question that matters is not the current state. It is what the agent saw, was told, and could access at the exact millisecond of the action. Not approximately. Not within the same batch window. At the precise moment the decision was taken, because that is the only reconstruction that is legally and operationally defensible. That is a time-travel query, and it is expensive to retrofit if the data layer was not designed for it. McKinsey's 2026 trust work makes the point that with agentic systems, the audit problem is no longer about wrong outputs but about reconstructing the exact world state that produced them. Most enterprises do not yet have the lineage discipline to answer that question reliably.

**What does this data actually mean here?** Data without semantic context is data the agent will misuse. Not because the data is wrong but because the agent cannot see the relationships, constraints, or business rules that would have shaped a sensible action. This is the layer where knowledge graphs, ontologies, and structured business semantics earn their keep. They are not academic constructs. They are how an autonomous system gets the surrounding context it needs to behave like it knows what it is doing.

**What is this agent allowed to see and do?** Agents cross system boundaries that human users typically do not. They retrieve from data domains that would normally require multiple separate access requests. They take actions that, taken together, can amount to something nobody approved as a coherent action. The data layer has to enforce policy at the call level, automatically and continuously, because there is no human checkpoint between the agent and the data it is reading. Manual governance does not scale to agent workloads. It does not even degrade gracefully. It just stops working at a certain volume and the failures are silent.

## Why Existing Infrastructure Cannot Answer These Questions Well

Most enterprise data infrastructure was designed around a different consumer. The end user was a human analyst building a report or running a query. That architecture is well-understood: batch warehouses, tabular structures, BI interfaces, and a governance layer that depends on humans following processes correctly.

Generative AI deployments stretched that architecture. Copilots needed unstructured data and retrieval interfaces that warehouses did not provide. Most organisations responded by adding things on top: a vector database here, a retrieval layer there, a metadata catalogue alongside. The architecture beneath stayed broadly the same, and for copilot-grade workloads it was good enough.

Agentic AI breaks that approach in a way that is not obvious until the deployment scales. Agents do not query the data layer once and reason. They query it continuously, in feedback loops, across sessions, with persistent state and accumulating context. They depend on streaming freshness, structured memory, governed retrieval, and traceable provenance simultaneously, and they depend on all four of those things working together at the speed of an autonomous decision loop. That is not a layer to add. It is an architecture to redesign, and the organisations that have understood this are pulling away from the ones that have not.

### KEY STATISTICS

**79%**
of organisations face challenges adopting AI, double-digit increase from 2025 (Writer, 2026)

**23%**
of companies see significant ROI from AI agents (Gartner)

**3,235**
leaders surveyed across 24 countries in Deloitte's 2026 State of AI report

## The Tooling Is Not the Problem

There is no shortage of tools that solve individual pieces of this problem. Streaming platforms for real-time movement. Vector databases for retrieval. Knowledge graph engines for context. Memory frameworks for persistence. Governance and policy engines. Observability stacks. The market is full, and most of the major components are commercially mature.

What enterprises are missing is not the tools. It is the architectural integration that takes those tools and makes them behave as a coherent data layer that an autonomous system can actually rely on. The integration is the value. The tools are the components. Most procurement processes I have watched in the last twelve months get this the wrong way round, which is usually how organisations end up with sophisticated tooling that does not fit the data estate it was supposed to serve.

"Agentic systems require infrastructure that supports adaptive, multi-turn interactions in which agents dynamically discover capabilities, share context, and hand off work as tasks evolve."

Bain & Company, 2026

Section 3

# Why Pipeline Architecture Cannot Carry Agentic Workloads

The first whitepaper in this series argued that organisations stuck in pipeline thinking would not get AI into production at scale, and that platform thinking was the architectural shift that made the difference. That argument still holds. What this whitepaper has to add is that even early platform thinking is not quite enough for agentic workloads. Platforms were designed to feed applications, models, and analytical workloads. Agents are something else again, and three things about them break the assumptions even mature platform architectures depend on.

## Continuous Consumption, Not Scheduled Demand

A reporting workload is bursty. A user opens a dashboard, the warehouse runs the query, the result lands, the system goes idle until the next request. Even copilot workloads, for all their freshness requirements, are still essentially request-response. Someone types, something replies.

An agentic workload is none of those things. A single deployed agent runs reasoning loops continuously, retrieving from data sources, calling tools, writing back to memory, sometimes spawning sub-agents to handle parallel sub-problems. And this is where traditional data warehouses encounter a failure mode they were never designed for: the agentic thundering herd. A single complex task spawns multiple sub-agents simultaneously, each issuing concurrent, low-latency data requests that arrive without warning and do not wait for a batch window to open. A warehouse built for scheduled demand, sized for predictable query patterns, can be overwhelmed in seconds by a workload that looks nothing like the one it was provisioned for. The data layer that feeds agentic workloads cannot afford the latency, the batch windows, or the scheduling assumptions that worked for analytics. Bain's 2026 architecture research calls this out specifically: legacy IT was built to route predictable, stateless transactions, and that is not what agents do.

## Memory as Infrastructure, Not as Framework Default

Without persistent memory, agents forget everything between sessions. They cannot personalise. They cannot accumulate learning. They cannot reliably complete multi-step tasks that span more than a few minutes. The default response to this is to throw whatever the framework provides at the problem and hope it scales. It does not. Production agent memory is a layered architecture and the layers do different things.

McKinsey's agentic mesh research argues for layered decoupling: logic, memory, orchestration, and interface functions separated to maximise modularity. In practice that means episodic memory for interaction history, semantic memory for stable facts and extracted knowledge, and state memory for live operational data that changes as agents act. Each has different freshness requirements, different retrieval patterns, and different governance constraints. Bain has been making the same argument in different language. None of this is novel any more.[7][11][12]

What is harder is the lifecycle question. What gets stored. What gets consolidated. What gets updated. What gets discarded. What happens when the layers contradict each other, which they will. These are not application questions to be solved per agent. They are infrastructure questions to be solved once and inherited by everything the organisation deploys. The organisations that have not done this engineering are running agents that look fine until they do not, and the failure mode looks like model drift but is actually memory drift, which is a much harder problem to fix retrospectively.

**"Memory management must be treated as a first-class infrastructure concern rather than an application afterthought."**

Bain & Company, 2026

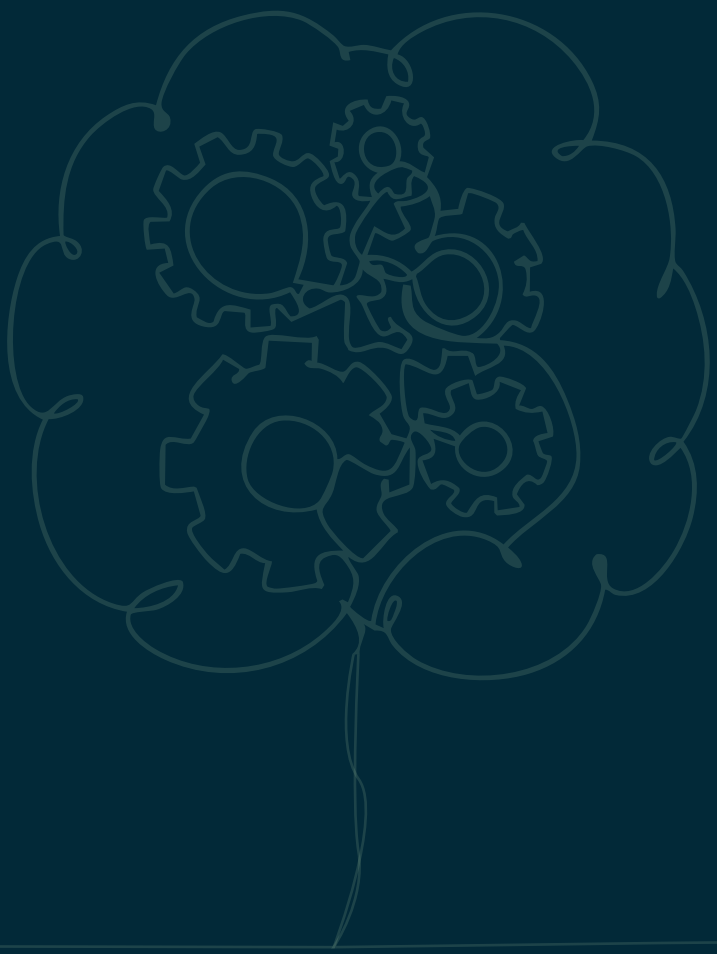
## The Data Layer Is the Decision Layer

This is the point that gets least airtime and matters most. Pipelines and platforms were built on the assumption that data would be delivered to a consumer who would then decide what to do with it. The consumer was downstream of the data layer. Agents collapse that distinction. They are the consumer and the actor in the same loop. A wrong piece of context does not produce a wrong report that a human can override. It produces a wrong action, taken before anyone has the chance to override anything. The instinctive response is to insert human-in-the-loop controls as a safeguard. But at the speed and volume agents operate, human-in-the-loop either becomes a bottleneck that defeats the purpose of deploying agents in the first place, or it becomes a rubber-stamp that humans wave through because the decisions are arriving faster than they can meaningfully review them. Neither outcome is oversight. Both outcomes trace back to the same root cause: a data layer that was not built to make autonomous decisions defensible before they are taken.

## Why More Engineers Will Not Save This

The instinctive management response to a data layer that is not keeping up with agent demand is to add engineering capacity. More data engineers, more pipelines, more tools. The first whitepaper in this series called this the hiring trap and the same trap applies here, with the wrinkle that the failure mode for agentic workloads is harder to see than the failure mode for analytics workloads.

Adding engineers to a data layer that was not designed for agentic consumption produces a more complex pipeline environment. It does not produce an agent-ready data layer. The integration points multiply, the failure modes multiply, the cost multiplies, and the agent deployment continues to operate on infrastructure that was never going to support it properly. The organisations solving this in 2026 are not the ones with the largest data engineering teams. They are the ones who recognised the architectural shift early enough to redesign rather than extend, and who treated the redesign as a strategic infrastructure decision rather than a tooling upgrade.



Section 4

# Five Pillars of an Agent-Ready Data Layer

An agent-ready data layer is not a single technology and it is not a single product. It is five interdependent capabilities that have to be designed to work as a whole. They are not phases. They are not ranked priorities. They are simultaneously active architectural concerns, and the dependencies between them are why partial implementations tend to disappoint. Build streaming foundations without governance and the agent acts faster on data it should not have seen. Build memory without observability and the agent's mistakes become impossible to audit. Build governance without cost engineering and the data layer collapses under its own bill before it has a chance to deliver value. The pillars constrain each other, which is why architecture has to be holistic from the first design decision.

## Pillar 1 - Real-Time Streaming Foundations

Agents make autonomous decisions in real time. The data layer that feeds them has to operate on the same clock. That means moving the foundation away from batch and micro-batch ingestion, which was sufficient for analytics and adequate for retrieval-augmented copilots, to event-driven streaming as the default mode of data movement. Not as a side capability for specific use cases. As the centre of gravity.

In practical terms, that means an event bus at the centre of the data architecture, capturing changes as they happen across the operational systems agents have to reason about. It means stream processing for transformation, enrichment, and quality enforcement on the fly, rather than overnight reconciliation. It means a data contract layer that lets agents subscribe to the streams they need with confidence in what each event means and what governance rules apply to it.

This is not a small change. Most enterprise data estates are built around batch as the centre of gravity, with streaming added as a side capability for specific use cases. Agent-ready architectures invert that. Streaming becomes the centre. Batch becomes the edge case for workloads where freshness genuinely does not matter. The shift has cost implications, governance implications, and skills implications, and they need to be planned, not stumbled into. The organisations that started this transition two years ago for other reasons have a quiet head start that the rest of the market has not yet noticed.

For C-suite leaders, the right question is not "do we have streaming?" It is whether the streaming layer is actually the foundation of the data architecture or a tactical add-on that the next generation of agents will quickly outgrow.

## Pillar 2 - Memory and Context Architecture

Memory is the most under-engineered layer of most agentic deployments I have looked at. The default is whatever the agent framework provides, which is usually a vector database doing everything: episodic context, semantic facts, operational state, all in the same store, with no architectural distinction between them. That works at demo scale. At production scale it produces context dilution: as episodic, semantic, and state memory accumulate in a single undifferentiated store, the agent's retrieval starts surfacing irrelevant historical context alongside current operational data, its reasoning gets noisier, and its error rates climb in ways that look exactly like model degradation but are not.

A production-grade memory layer treats memory as infrastructure with deliberate decoupling. Episodic memory captures interaction history: what was asked, what was retrieved, what action was taken. Semantic memory holds stable knowledge: facts about the business, the customer, the domain. State memory holds live operational data: account balances, inventory positions, current entitlements. Each has different freshness requirements, different retrieval patterns, different governance constraints. Bain and McKinsey have both made the case for layered decoupling and there is broad agreement at this point that single-store approaches do not survive contact with production.

The harder questions sit underneath the architecture. Without an infrastructure-level expiration and consolidation policy, agents eventually drown in their own historical logs. Context that should have expired keeps surfacing. Contradictions between layers go unresolved. Memory contamination spreads when one agent writes something wrong and others start reading it as fact. None of these are application concerns to solve per agent. They are infrastructure concerns to solve once, and the engineering effort sits at the data layer where it can be inherited by everything the organisation deploys.

Without that discipline, context dilution is inevitable and progressive. McKinsey's research is direct: bolting memory stores onto existing generative AI stacks is necessary but not enough. What is needed is a fundamental architectural shift from static, model-centric infrastructure to a dynamic, modular, governed environment built specifically for agent-based intelligence.

## Pillar 3 - **Semantic and Knowledge Layer**

Agents that operate on data without semantic context tend to make decisions that are technically correct and commercially wrong. They cannot see the relationships between entities, the business rules that bound acceptable actions, the context the data does not contain. They take the data at face value. This is why so many early agentic deployments produce outputs that are individually defensible but collectively make no sense to anyone who knows the business.

A semantic layer addresses this by giving the agent structured context. Which entities exist, how they relate, what rules apply, what the data does not say. This is where knowledge graphs, ontologies, and graph-based retrieval architectures earn their place in the data layer. They are not the academic exercises some technical leaders still think they are. They are the architectural safeguard that prevents agents from taking actions that are technically correct but commercially nonsensical.

In production this means combining vector retrieval, which answers what is similar, with graph retrieval, which answers what is connected. Any team can buy a vector database. The engineering required to build a domain-specific knowledge graph that encodes the business rules, entity relationships, and operational constraints of a specific industry is what actually separates agents that behave intelligently from agents that behave plausibly. The retrieval contract changes from "find similar text" to "return the structurally complete context for this question" and those produce materially different inputs to the agent. They also produce materially different actions.

The semantic layer is also where domain expertise gets encoded. An agent operating in financial services needs different context than one operating in construction or healthcare. Generic semantic models do not deliver this. Domain-specific knowledge graphs, populated and maintained by people who actually understand the business, are what give an agent the kind of operational intelligence that makes it useful. This is one of the few parts of the agentic stack where you cannot buy your way out. It has to be built, and it has to be maintained by people with domain authority. That is not a technology procurement problem. It is a capability question.

## Pillar 4 - **Governance and Observability for Autonomous Systems**

Traditional data governance was built for human workflows. A user requests access. An approver reviews. A policy gets applied. The user reads or downloads. The action is logged. The model assumes deliberate, sequential, reviewable activity that happens at human speed.

Agents break every assumption in that model. They make hundreds of data calls per task. They act across systems in parallel. Their decisions are not deliberate, they are continuous and contextual. They generate audit trails that traditional governance tools cannot reconstruct because the events happen too quickly and in too many places to capture by manual review. Deloitte's 2026 finding that only one in five companies has mature governance for autonomous agents is not surprising once you look at what mature governance actually requires.[5][13]

Governance for agents has to be embedded into the data layer itself. Access controls, privacy rules, and policy constraints enforced at the call level, automatically. Every retrieval an agent makes has to be policy-checked, logged with full lineage, and traceable back to source. Every action an agent takes has to be reconstructable, not just in terms of what happened but in terms of what the agent knew at the moment, what context it had, what alternatives it considered. None of this is exotic. It is just not what most existing governance stacks were designed to do, and retrofitting is harder than building correctly from the start.

Observability for agents is closely tied. Traditional monitoring tracks system health. Agent observability has to track decision health. Are agents reasoning over fresh data. Are their retrievals returning the right context. Are their actions consistent with policy. Is performance drifting in ways that suggest the underlying memory or semantic layer needs attention. This is a different telemetry contract, and most organisations have not made the investment yet. The ones that have are catching problems earlier, which over a 12-month deployment cycle compounds into a meaningful difference in operational reliability.

"In the age of agentic AI, organisations can no longer concern themselves only with AI systems saying the wrong thing. They must also contend with systems doing the wrong thing."

McKinsey, State of AI Trust 2026

## Pillar 5 - Cost Engineering for Agentic Workloads

Agentic workloads have a cost profile that does not look like anything that came before them. An analytics workload incurs cost when a user runs a query. A copilot workload incurs cost when a user types a prompt. An agentic workload incurs cost continuously and autonomously, and the cost scales with deployment because agents run whether or not anyone is actively using them. Retrieving context. Calling tools. Invoking models. Writing back to memory. The bill keeps moving.

At pilot scale this is invisible. The numbers are small, the team is enthusiastic, the business case is hypothetical. At production scale, agentic workloads have a documented tendency to become the largest line on the cloud bill, and to do so suddenly enough that finance teams find out about it from the invoice rather than the architecture review. Gartner cites cost escalation as one of the three primary reasons agentic projects get cancelled by 2027 and it is not a small share. Organisations that did not engineer for cost from the start tend to find out the hard way.[1]

Cost engineering has to be designed into the data layer, not added afterwards. That means unit economics tracking at the agent and workload level, so the organisation actually knows what each agent task costs. It means caching strategies for retrieval and memory operations to avoid redundant LLM calls and vector lookups. It means tiered storage and compute for memory layers, with the right freshness and access pattern matched to the right cost profile. And it means observability that surfaces cost alongside performance, so finance and engineering see the same picture without needing to reconcile two different reports a quarter later.

For C-suite leaders, the right question is not how much the agentic deployment costs in aggregate. Aggregate cost figures are where financial liabilities hide. The question is whether the organisation has unit economics tracking at the task level: what does each individual agentic goal actually cost in cloud compute, retrieval operations, and LLM invocations. If that number cannot be produced per task, per agent, and per workflow, the deployment is not a managed programme. It is a looming financial liability for 2027, one that will surface as a line on a cloud invoice rather than as an architectural decision the organisation made with full visibility. The CFO should not be the first person to discover the cost curve has broken. The data layer should have surfaced it months earlier.

Section 5

# What Agent-Ready Infrastructure Delivers

By 2026 the agentic AI conversation has split into two markets that do not fully see each other. The visible one is about the agents themselves, the models, the orchestration frameworks, the use cases, the vendors. That is where most of the headlines and most of the budget conversations live. The structural market, where the next two years of competitive position will actually be decided, is about the data layer underneath, and it is much quieter because the work is harder to demonstrate in a slide deck. Every enterprise has access to broadly the same agent frameworks. What separates the deployments that scale from the ones that get cancelled is whether the data layer can carry the workload.

## Agentic Initiatives That Survive Past Pilot

The most direct return on investment in an agent-ready data layer is the ability to keep agentic systems running in production once they get there. Gartner's 40% cancellation forecast describes a market where most agentic deployments will not make it. The minority that do are not the ones with cleverer agents. They are the ones whose data layer can deliver fresh, governed, contextually rich data at the speed and reliability that autonomous decisions actually require.

That is the difference between an agentic deployment that produces measurable business value and one that produces a slow accumulation of cost, exposure, and disappointment. The investment in the data layer is what determines which side of that line a deployment lands on, and the architectural decisions compound. Organisations that get this right in 2026 will have a multi-year structural advantage that competitors working agent by agent and integration by integration will find very difficult to close, because the work itself takes years and cannot be replicated in a quarter.

## Decision Quality at Machine Speed

When agents are running on fresh, governed, contextually rich data, the quality of their decisions improves in ways that show up directly in business outcomes. Pricing agents that respond to live market signals price more accurately. Risk agents that see exposure in real time intervene earlier. Customer experience agents that have current context personalise more relevantly. Operations agents that know the actual state of the business take actions the business would actually endorse if it were asked.

None of this is about model sophistication. All of it is about the data infrastructure underneath. The model is increasingly a commodity in the sense that every competitor has access to broadly the same frontier capability. The data layer is the moat, because building one takes sustained engineering investment over multiple years and there are no shortcuts that do not show up later as technical debt.

**88%**
of executives plan to increase AI budgets due to agentic initiatives

**33%**
of organisations report AI governance maturity at level 3+
(McKinsey, 2026)

**3,235**
leaders surveyed in Deloitte's 2026 State of AI report

## Cost Curves That Do Not Run Away

The agentic workloads that get cancelled for cost reasons are usually running on data layers that were never designed for them. Each retrieval is more expensive than it needs to be. Each memory operation triggers more LLM calls than it should. Each governance check happens at the wrong layer of the stack. The cost compounds, the unit economics get worse as deployment scales, and at some point finance asks an uncomfortable question that engineering cannot answer cleanly. By that point the budget conversation has already moved past architecture and into damage control.

Agent-ready data layers deliver a different curve. Caching strategies and retrieval optimisation reduce redundant operations. Tiered storage matches cost to freshness requirements. Unit economics visibility lets engineering identify the workloads that are expensive without delivering proportional value, and lets the organisation actually shut those down before they accumulate further cost. The cost of running agents stabilises and scales sub-linearly with deployment, which is the cost profile a CFO can plan around without needing weekly reassurance.

## Regulatory Confidence as Scrutiny Tightens

Regulatory pressure on AI is tightening across every major market, and not slowly. The EU AI Act is in force. National AI legislation is moving in multiple jurisdictions. Sector regulators in financial services, healthcare, and energy are issuing increasingly specific requirements around model transparency, data provenance, and decision auditability. For agentic systems, which take actions autonomously rather than just generating outputs, the regulatory bar is higher than for any AI category that came before, and it will go on rising for the rest of this decade.

Organisations whose data layer enforces governance automatically, logs every retrieval and action with full lineage, and supports point-in-time reconstruction of what an agent knew when it acted are not just better positioned for compliance. They are positioned to deploy more aggressively in regulated sectors where competitors without that infrastructure are going to be forced to slow down. Governance, in the agentic era, has stopped being a brake on AI strategy. For organisations that built it correctly, it is the thing that lets them keep moving while everyone else is recalibrating.

## Talent That Stays

There is one more dimension that matters at the C-suite level and that boards routinely under-weight. The data engineers and AI engineers capable of building production-grade agentic infrastructure are scarce, expensive, and well aware of which organisations are doing this work properly. They do not want to spend their careers maintaining brittle pipelines or firefighting agent failures that better architecture would have prevented. They leave for organisations that are doing the work the right way, because the problems are more interesting and the systems are more reliable. In a market where engineering capacity is the binding constraint on AI execution, the data layer investment is also a talent retention investment, and over a three-year horizon that probably matters more than any single procurement decision.

Section 6

# Where to Start: A Practical Roadmap

The most common and most expensive mistake organisations make at the start of an agentic infrastructure programme is reaching for tools before they have defined the architecture. A new vector database. A new memory framework. A new orchestration platform. None of these will deliver their promised value if the data layer underneath has not been properly designed. In practice, this is how organisations end up with sophisticated agentic tooling that does not fit the data estate it was meant to serve, expensive licences for platforms that cannot be fully integrated, and an agent deployment that hits a ceiling its underlying architecture was never going to let it cross.

Architecture before tooling is a commercial discipline rather than a philosophical preference. Getting the architecture right first means understanding what the data estate actually looks like today, where the gaps are, what production agentic workloads will demand of the infrastructure in two to three years, and what sequencing of investment unlocks the most value at each stage. Only once that picture is clear does technology selection become a defensible decision. Without it, every vendor pitch sounds plausible, and most of them are.

## Phase 1 - Assess and Architect

Before any tool gets selected or any agent gets deployed at scale, the organisation needs an honest assessment of its current data layer relative to the demands agentic workloads will place on it. This is the foundation everything else is built on. It is also the stage most organisations either skip or compress into a few weeks of vendor-led workshops that produce a slide deck rather than an actionable architectural blueprint, which is a common pattern and a costly one.

A genuine architectural assessment for agentic readiness covers data freshness and streaming maturity, memory architecture and persistence patterns, semantic context coverage across the domains the organisation operates in, governance and observability gaps specifically for autonomous workloads, and the unit economics of the agent tasks the organisation is prioritising. The output is not a technology recommendation. It is an architectural design and an investment roadmap that shows what needs to be built in what order, and why. The order matters as much as the design.

## Phase 2 - Stabilise and Govern

With the architecture defined, the priority is stabilising the data foundation that agents will run on. That means addressing the most critical gaps from the assessment, not comprehensively but strategically, focused on the domains and use cases that matter most for the agentic workloads in the immediate pipeline. Streaming foundations get prioritised for the data domains that genuinely need real-time freshness. Memory architectures get designed for the agent types being deployed. Governance and observability frameworks get embedded at this stage, before any agent deployment scales, so that every workload built on top inherits compliance and traceability by default rather than requiring it to be retrofitted later under pressure.

## Phase 3 - Modernise and Scale

With a stable, governed foundation in place, the organisation can start scaling agentic deployments with some confidence that the foundations will hold. That means rolling out the streaming, memory, and semantic layers across additional data domains as use cases mature, expanding self-service access so AI engineering teams can build on the data layer without filing tickets for every new requirement, and running cost engineering as a continuous discipline rather than an annual exercise. Tooling decisions happen here, informed by the architectural blueprint defined in Phase 1, which is what makes them defensible against vendor pressure and market noise.

## Phase 4 - Optimise and Compound

The final phase is where the platform thinking applied to agentic infrastructure starts paying back the investment. New agents deployed on the data layer inherit the streaming, memory, semantic, governance, and cost engineering already embedded in the foundation. Engineering effort shifts from building bespoke plumbing for each new agent to extending a platform that gets more valuable with every workload added to it. Decision velocity rises. Costs stabilise. Governance gets stronger as the layer matures. The organisation builds an agentic capability that competitors working agent by agent will find increasingly difficult to close the gap on, because they are buying tools while the platform-grade organisation is compounding architecture.

## How Merit Approaches This

Merit's data engineering practice is built on a working belief: data infrastructure for agentic AI is a strategic asset, and treating it as a tooling decision is one of the most expensive mistakes an organisation can make in this cycle. We design agent-ready data layers from the architecture up, defining streaming, memory, semantic, governance, and cost engineering as a single system rather than separate workstreams. Governance and observability get embedded in the foundation rather than appended. Memory and semantic context are treated as first-class infrastructure. The same DataOps and platform discipline that mature data engineering teams have always applied to their critical pipelines gets applied to agentic workloads from day one.

Our clients do not inherit a brittle agentic stack from us. They inherit a data layer that scales with the agents they deploy on it, and that compounds in value with every workload added.

## Conclusion

The question facing every enterprise leader with an agentic AI agenda in 2026 is not whether to deploy agents. It is whether the data layer beneath those deployments can carry them through 2027 and beyond.

The 40% cancellation forecast that has put a quiet shadow over agentic AI investment across every sector is not a model problem, not a vendor problem, not a use-case problem. It is a data layer problem. One that pipeline thinking cannot solve. One that even early platform thinking will struggle with. One that requires a deliberate architectural commitment to building infrastructure for autonomous workloads, not extending what was built for human ones.

Organisations that make the right architectural decisions in 2026 will not just be ahead in 2028. They will have spent two years compounding an advantage rivals cannot close in a hurry, because agent-ready data layers are not something you can replicate in a quarter. They are built through sustained, sequenced investment, and the organisations that started early will have the production agents, the decision quality, the cost discipline, and the regulatory confidence to show for it.

The ones that did not will still be cancelling pilots in 2027, watching the budget overruns, and explaining to their boards why the agentic AI strategy did not deliver. 2026 is not the moment to plan the data layer for agents. It is the moment to build it.

## Ready to build the data layer your agents need?

Merit's data engineering teams design and build agent-ready data layers for organisations in financial services, energy, maritime, healthcare, and beyond. We work from architecture and assessment through to production deployment, ensuring the infrastructure beneath your agentic AI investment is built to scale, govern, and deliver.

meritdata-tech.com

## About Merit Data & Technology

Merit Data and Technology is part of Merit Group PLC. For over 20 years, we have been helping enterprises design and build data infrastructure that performs under real-world conditions, not just in the proof of concept but in production, at scale, and under regulatory scrutiny.

Two decades in this field means we have seen technology cycles come and go. We have worked with organisations that got the architectural decisions right early and compounded the advantage, and with organisations that deferred those decisions and paid a steep price to recover. That experience shapes how we approach every engagement: with the rigour, the realism, and the long-term perspective that high-stakes data transformation decisions deserve.

For C-suite leaders who need more than a vendor, who need a partner with the track record to be trusted with decisions that will define their organisation's AI trajectory for years to come, Merit brings 20 years of delivery credibility to the table.

## Sources


1. Gartner (2025). Press Release: Gartner Predicts Over 40% of Agentic AI Projects Will Be Canceled by End of 2027. June 25, 2025. gartner.com/en/newsroom/press-releases/2025-06-25-gartner-predicts-over-40-percent-of-agentic-ai-projects-will-be-canceled-by-end-of-2027

2. Gartner (2025). Forecast cited in press release: 40% of enterprise applications will embed task-specific AI agents by end of 2026, up from less than 5% in 2025. Ibid.

3. Writer (2026). Enterprise AI Adoption in 2026 Survey Findings. Survey of 2,400 global leaders. writer.com/blog/enterprise-ai-adoption-2026

4. McKinsey & Company (2026). State of AI Trust in 2026: Shifting to the Agentic Era. AI Trust Maturity Survey of approximately 500 organisations conducted December 2025 to January 2026. mckinsey.com/capabilities/tech-and-ai/our-insights/tech-forward/state-of-ai-trust-in-2026-shifting-to-the-agentic-era

5. Deloitte AI Institute (2026). State of AI in the Enterprise 2026. Survey of 3,235 senior leaders across 24 countries. Finding: only one in five companies has a mature governance model for autonomous AI agents. deloitte.com/us/en/what-we-do/capabilities/applied-artificial-intelligence/content/state-of-ai-in-the-enterprise.html

6. Bain & Company (2026). Building the Foundation for Agentic AI: Technology Report. Analysis of legacy batch-based system limitations and the need for real-time, API-accessible architectures. bain.com/insights/building-the-foundation-for-agentic-ai-technology-report-2025/

7. Bain & Company (2026). The Three Layers of an Agentic AI Platform. Analysis of orchestration, memory management, and governance as first-class infrastructure concerns. bain.com/insights/the-three-layers-of-an-agentic-ai-platform/

8. Gartner (2025). Press Release on agentic AI cost escalation as a primary failure cause. June 25, 2025. Ibid.

9. McKinsey & Company (2026). Trust in the Age of Agents: Agentic AI Governance for Autonomous Systems. mckinsey.com/capabilities/risk-and-resilience/our-insights/trust-in-the-age-of-agents

10. Bain & Company (2026). Why Agentic AI Demands a New Architecture. Analysis of why legacy IT architectures cannot support adaptive, multi-turn agent workflows. bain.com/insights/why-agentic-ai-demands-a-new-architecture/

11. Bain & Company (2026). From Roadmap to Reality: Phasing Agentic AI into Production. Analysis of memory governance and persistent context across sessions. bain.com/insights/from-roadmap-to-reality-phasing-agentic-ai-into-production/

12. McKinsey & Company (2025). Seizing the Agentic AI Advantage. Analysis of layered decoupling for logic, memory, orchestration, and interface functions in the agentic AI mesh. mckinsey.com/capabilities/quantumblack/our-insights/seizing-the-agentic-ai-advantage

13. McKinsey & Company (2026). State of AI Trust in 2026: governance maturity for agentic systems. Ibid.

14. McKinsey & Company (2026). Reimagining Tech Infrastructure for and with Agentic AI. Analysis of the architectural shift required for agentic operations. mckinsey.com/capabilities/mckinsey-technology/our-insights/reimagining-tech-infrastructure-for-and-with-agentic-ai

15. McKinsey & Company (2026). Building the Foundations for Agentic AI at Scale. Analysis of data transformations required for enterprise agentic AI. mckinsey.com/capabilities/mckinsey-technology/our-insights/building-the-foundations-for-agentic-ai-at-scale