software AG
A SOFTWARE GMBH BRAND

WHITE PAPER

The Canonical Truth Problem

Why Enterprise AI Can't Reach the Data That Actually Runs Your Business

Most "why enterprise AI fails" analysis gets the diagnosis right — it's the data, not the models — and the prescription incomplete. The reason: "enterprise data" is not one thing.

Three claims anchor this article:

Contextual vs. canonical. AI today runs on contextual data (documents, emails, chat transcripts). The hard problem is canonical data — the authoritative system-of-record answer to "what is this customer's balance, right now?" — and most AI architectures can't reach it.

Three timescales, three architectures. Classical ML runs on yesterday's data (T-1). RAG runs on near-live data (T-ε). Agentic AI taking consequential actions needs the state of the world this second (T-0) — and that isn't delivered by faster pipelines.

Mainframe access is the diagnostic. If your AI can't answer a canonical question end-to-end, within a single reasoning step, against a system of record in a regulated workload — you've solved the contextual problem and left the canonical one alone.

A concrete scenario

A claims-adjudication agent built on a modern LLM stack reviews a submission. It pulls the customer's policy history from a vector store, checks the claim against an embedded summary of the policy document, and auto-approves the payout. The approval is wrong. The policy was cancelled for non-payment eleven minutes before the agent ran. The vector index, refreshed every ten minutes by change-data-capture, happened to have completed its last sync twelve minutes before the agent's reasoning step — so the cancellation wasn't in the retrieved context. The system of record would have said otherwise, if the agent had asked it. Nobody did.

This is not a model problem. The model did exactly what it was asked to do: reason over the data it was given. The problem is that the data it was given was, by design, not the authoritative answer. This is the canonical truth problem, and the argument of article is that a major subset of high-stakes enterprise AI failures are canonical-truth failures — AI reasoning against the wrong copy, not because the model is stupid but because the authoritative answer was never reachable from the reasoning step. Even a well-engineered T-ε pipeline doesn't save this agent: the failure window is inside the refresh interval, not outside it.

The diagnosis is converging. The prescription isn't yet.

That enterprise AI is stalling at scale is no longer in dispute. S&P Global Market Intelligence's Voice of the Enterprise: AI & Machine Learning, Use Cases 2025, surveying 1,006 organizations with fieldwork October 21–November 25, 2024, found the share of companies abandoning most of their AI initiatives jumped from 17% the prior year to 42%, with the average organization scrapping 46% of proofs of concept before production.[1] MIT Project NANDA's July 2025 preliminary report The GenAI Divide: State of AI in Business 2025 argues that only 5% of integrated AI pilots are extracting millions in value, while 95% of organizations report no measurable return.[2] RAND, in its 2024 Root Causes of Failure for AI Projects, synthesizes prior estimates that place AI project failure above 80% — roughly twice the rate of non-AI IT projects.[3] IBM's 2025 survey of 1,700 senior data and analytics leaders, including CDOs and related roles, found only 26% confident their data can support AI-enabled revenue streams.[4]

A methodological caveat is warranted here. These numbers don't share a denominator, a population, or a definition of "failure." S&P measures organizations abandoning most of their initiatives before production; MIT measures organizations with zero measurable return alongside a much smaller group of integrated pilots extracting millions; Informatica measures obstacles to pilots reaching production; Gartner forecasts abandonment among projects without AI-ready data; IBM measures leaders' confidence in data readiness; RAND is synthesizing other people's estimates rather than presenting its own direct measurement. They are not interchangeable. What is striking is that each cut of the question — from production engineers, CDOs, IT leaders, and analysts — points in the same direction.

The diagnosis has converged remarkably fast: it's not the models, it's the data. Informatica's CDO Insights 2025 (n=600) finds data (43%) tied with technical maturity (43%) as the top obstacles preventing GenAI initiatives from reaching production, with employee skills and data literacy at 35%.[5] MuleSoft's connectivity benchmark reports that 95% of enterprise IT leaders struggle to integrate data across systems, with 80% citing data integration as a major challenge for AI adoption.[6] Gartner predicts that through 2026, organizations will abandon 60% of AI projects unsupported by AI-ready data.[7]

Everyone agrees on the headline. Where the genre falls short is in the prescription — because most blog posts and articles treat "enterprise data" as a single undifferentiated mass. It isn't. And the distinction that matters most is not between structured and unstructured, on-prem and cloud, or clean and dirty. It's between **contextual** and **canonical**.

Cause 1: Contextual data vs. canonical data

Contextual data is what's generated around the business: emails, Slack threads, call transcripts, meeting recordings, contracts in PDF, wiki pages, Confluence docs, support tickets, product documentation. There's a lot of it — the commonly cited 80% directional estimate for unstructured enterprise data[8] lives mostly here. And despite the volume, the engineering problem is well-understood: ingest it, chunk it, embed it, store it in a vector database, retrieve on demand. The industry has spent two years building exactly this pipeline, and it mostly works. Worth saying plainly: most of the successful enterprise AI deployed today — customer-support copilots, internal knowledge assistants, contract review tools, coding assistants, document summarization — runs on contextual data, and that success is real. The "5% that works" in MIT's study is not mythical. It is largely contextual-data AI delivering measurable value.

Canonical data is different. Canonical data is the authoritative answer to a question the business has to get right: *what is this customer's current account balance? Is this claim approved? Is this policy active? Who actually owns this position? What is the committed delivery date?* Much of it is transactional in origin — balances, claims, trades, orders — but the category also includes the master records, policy definitions, entitlements, and reference data that AI has to reason over correctly. What unites them is not how they were produced but their status as the authoritative record. A document stating a balance is not the balance — it is a projection of the balance as of when it was written. Canonical access means reaching the balance itself, as of now. Canonical data lives in *systems of record* — core banking platforms, claims systems, ERP, policy administration, trading systems. A substantial share of those systems, in regulated industries, still runs on mainframes on IMS, Db2, VSAM, or Adabas. The prevalence is non-trivial: IBM reports that its mainframe is used by two-thirds of the Fortune 100, 45 of the top 50 banks, and 8 of the top 10 insurers for mission-critical workloads.

A note on lineage: the canonical/contextual distinction is a close cousin of "system of record" in enterprise architecture and of master data management in the data governance tradition. What's new here is framing it around AI's relationship to authority, not around storage or entity harmonization. And this is a spectrum, not a binary — a CRM holds canonical account status alongside contextual call notes, a warehouse holds derived views of canonical data at varying distance from the source. The distinction is useful as a diagnostic, not as a taxonomy.

Three things make canonical data fundamentally harder to feed into AI:

It has to be correct by definition, not by approximation. A contextual-data RAG system that returns a slightly wrong snippet from a policy document is annoying. A canonical-data agent that answers "your account balance is $12,400" when it's actually $1,240 is a regulatory incident. The tolerance for approximation collapses to zero. This doesn't mean canonical sources are themselves infallible — systems of record have bugs, reconciliation lag, and inconsistencies between redundant copies. But those imperfections are *known, bounded, and auditable*. AI reasoning over canonical data has to inherit that bounded imperfection, not add a new, unbounded layer of its own.

It has structure that token-based AI doesn't natively preserve. Adabas rotated arrays (periodic groups that encode multi-row relationships inside a single record), IMS hierarchies (tree-structured parent-child segments), VSAM nested records (fixed-format transactional data with internal structure), Db2 relational semantics with referential integrity — these structures are the information. Flattening them into tokens loses exactly the semantics that make them authoritative. Most RAG-style ingestion pipelines are built for prose; they are actively destructive to canonical structure.

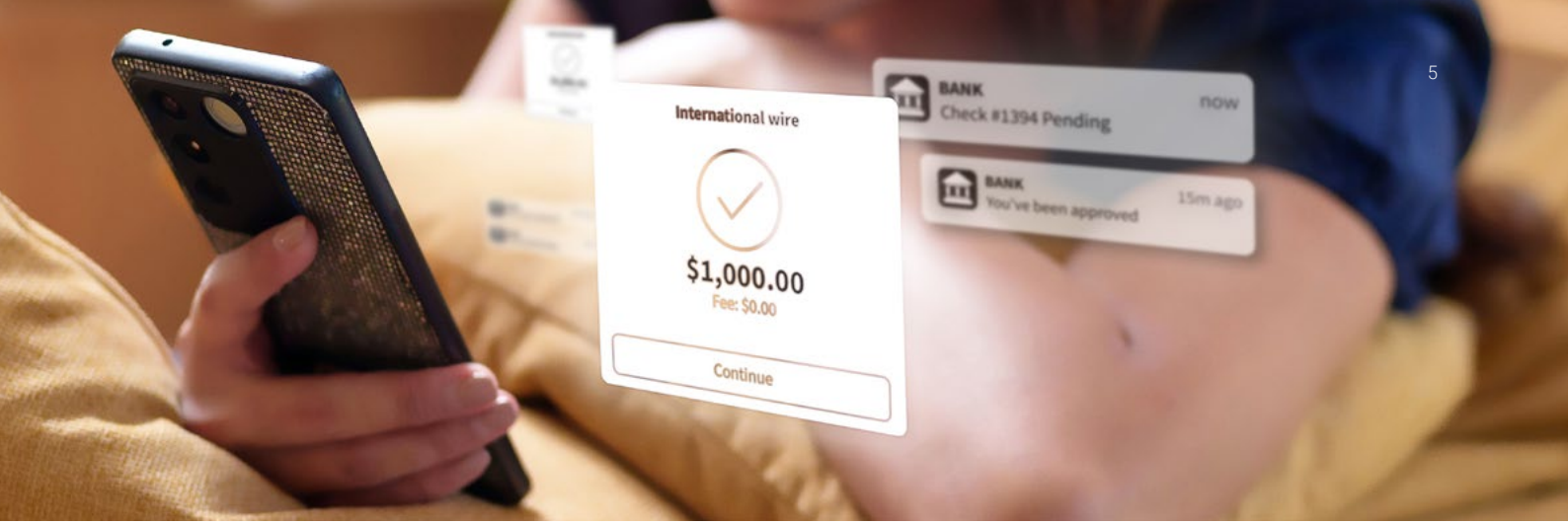
It lives behind governance boundaries that don't move. Contextual data can often be copied to a cloud vector store with modest risk. Canonical data, in regulated industries, cannot. Regulatory regimes — SOX, GDPR, HIPAA, PCI DSS, Basel III, Solvency II, GLBA — draw hard lines around systems of record. Compliance is not won by finding a clever place to stash a copy.

This is why IBM's reference to an IDC estimate — that less than 1% of unstructured enterprise data is currently being used in generative AI[9] — understates the canonical-data issue specifically. If you measured only canonical data reaching AI reasoning steps under regulated workloads, the number would be far lower, and the gap far more consequential.

Cause 2: Three timescales, three architectures

Compounding the canonical-data problem is a fact that most of the AI failure literature blurs: **each generation of AI needs data on a different timescale, and each timescale demands a different architecture.**

Generation	Timescale	Typical pipeline	Failure mode if violated
Classical ML	T-1 (yesterday)	Nightly ETL → warehouse → feature store → batch retraining	Model decay, stale features
RAG-based GenAI	T-ε (minutes)	Change-data-capture → vector store → inference-time retrieval	Hallucination against stale content
Agentic AI (consequential actions)	T-0 (now, inside the transaction boundary)	Direct query to system of record within reasoning step	Action taken on already-superseded state



A clarifying point: traditional OLTP systems (Online Transaction Processing — the category of systems built to handle transactional reads and writes inside strict isolation boundaries) have been doing T-0 for decades. A banking core that processes a wire transfer does not consult last night's snapshot; it locks the row, reads the current balance, and commits. **The novelty isn't T-0 access itself. The novelty is an AI reasoning step operating inside that same boundary,** making non-deterministic decisions on canonical state, with enough auditability to satisfy a regulator. That is the new thing.

Much of what's sold as "agentic" is really multi-step RAG and doesn't need T-0 — summarizing a thread, drafting a reply, researching a topic. Those workflows are fine on T-ε. But when an agent takes a *consequential, irreversible action against a system of record* — writing a trade, approving a claim, releasing an order, updating a policy — it needs the state of the world *as of this moment, inside the reasoning step itself*. Not this morning. Not five minutes ago. Now. Because by the next step, the agent will have acted on that state, and the reconciliation against reality will happen only after the fact.

One sharpening worth making explicit: the hardest version of this problem is not agents *reading* canonical state in real time — it is agents *writing* to it. An agent that reads a stale balance produces a wrong answer; an agent that writes against stale state produces a wrong *action*. Consequential agentic workflows — approving a claim, releasing an order, writing a trade — are all writes, and writes require the same isolation, locking, and conflict-detection semantics that OLTP systems have offered for decades. For most such actions, the right architecture pairs T-0 canonical access with an explicit human-in-the-loop checkpoint. Fully-autonomous write-back against a system of record without any human review is a narrower use case than current agentic marketing suggests.

It is worth naming that direct T-0 querying at reasoning time is not the only architecture worth considering. Several weaker patterns can handle meaningful subsets of the problem: commit-time validation (the agent reasons on T-ε data but verifies against the system of record at the moment of commit); reservation or hold semantics (a tentative write, confirmed only after reasoning completes); optimistic concurrency control with a version-check at commit; human approval gates; reversible workflows; and idempotent writes designed to tolerate a stale read. Each of these shifts *where* the freshness check happens, but none removes the requirement. The point of this article is not that every consequential agent must query canonical state in real time — it is that *none* of these patterns work unless the AI layer can reach the authoritative record at all, under the governance that sits around it. That is the gap the industry has not yet closed.

The claims scenario at the top of this article is exactly this failure. An agent approving a claim on T-ε data can be approving against a policy that has already been cancelled, a balance that has already been encumbered by another transaction, a position that has already been closed. The consequences are not theoretical: the industry has already seen early incidents of AI agents acting on stale state in trading, order management, and claims adjudication workflows, and these will grow.

Most enterprise data infrastructure is T-1 by default and T-ε by effort. T-0 for canonical data under consequential workloads is a different problem — one that shrinking the batch or CDC interval can narrow but not close. It requires the AI layer to query the system of record directly, within transaction boundaries, with the same isolation and governance guarantees as the systems that already live there.

Agentic AI, once it starts taking real actions, doesn't gently raise the bar. It changes which architecture you need underneath.

Cause 3: The audit inversion

There is a secondary consequence of agentic AI on canonical data that the "why AI fails" literature has not caught up to yet: **the nature of audit trails has to invert.**

Traditional enterprise audit is a record of what users did to data. User U updated field F from value X to value Y at timestamp T. Thirty years of audit infrastructure — on mainframes, in databases, in SIEM systems — is organized around this schema.

AI agents acting on canonical data require a different schema. You need to log: which data was shown to which model, what the model inferred from it, which tool call that inference triggered, what state the system was in when the action completed, and which human (if any) was in the loop. Reconstructing a bad outcome — a wrong trade, a wrongly denied claim, a compliance breach — requires walking this chain backward.

To make that concrete, a minimal agentic-audit record looks closer to the following than to a traditional database audit row. This is illustrative rather than normative; production schemas will differ by industry:

```
{
  "action_id": "uuid-4b9f...",
  "action_timestamp": "2026-04-20T14:02:11.402Z",
  "agent":  { "id": "claims-adjudication-agent", "version": "3.2.1" },
  "model":  { "id": "foundation-model-name", "version": "...",
              "prompt_hash": "sha256:..." },
  "context_shown": {
    "retrieval_snapshot_id": "vec-snap-9f2c",
    "retrieval_window_age_ms": 612000,
    "sources": ["policy-doc-8821", "customer-note-441"]
  },
  "canonical_reads": [
    { "system": "adabas-policies",
      "query": "POLICY-ID=AX-992831",
      "snapshot_tx_id": "TX-2026-04-20-14020984",
      "returned_fields": { "status": "ACTIVE",
                           "premium_paid_through": "2026-05-31" } }
  ],
  "inference_trace": { "reasoning_step_ids": ["r1","r2","r3"],
                       "tool_calls": ["read_policy","compute_payout"] },
  "action": { "tool": "approve_claim",
              "parameters": { "claim_id": "CL-772", "amount": 12400.00 } },
  "system_state_at_commit": { "tx_id": "TX-2026-04-20-14021107" },
  "human_in_loop": { "user_id": "u-3381",
                     "approval_timestamp": "2026-04-20T14:02:06Z" }
}
```

The point isn't the exact field set. The point is that every one of these fields has to be emitted by a different layer of the stack — agent runtime, model runtime, retrieval layer, canonical-data access layer, system of record, approval UI — and then reconciled. A partial solution in any one tier doesn't produce a usable audit trail.

For regulated industries, fragments of this are already being required. The EU AI Act — Regulation (EU) 2024/1689, Articles 12–15 — mandates record-keeping, transparency, human oversight, and accuracy/robustness for high-risk systems.[10] GDPR Article 22 covers decisions based solely on automated processing that produce legal or similarly significant effects. Related control regimes — FINRA Rule 3110 on supervision, HIPAA 45 CFR §164.312 on technical safe guards — are useful analogues for auditing automated activity, but they predate agentic AI and do not specifically contemplate it. These requirements are fragmentary today and will consolidate — either through regulation converging on a common schema, or through enterprises being forced to invent one after the first major incident. Almost no existing enterprise data layer captures the full trace, because existing audit was designed for a different schema.

This reshapes what the protocol layer between the agent and the system of record has to do. That protocol sits on every call in the chain and is the only layer positioned to carry provenance consistently across inference, tool invocation, data access, and commit; a protocol that doesn't carry it forces audit to be rebuilt somewhere else. That is why "just put MCP in front of your mainframe" is not a real answer. The protocol has to carry the audit trail, the data layer has to emit it, and the AI layer has to consume it — and all three have to reconcile. Most of the current solutions don't.

The mainframe as diagnostic

Here is a test that cuts through most AI strategy conversations.

> *Pick a question whose answer lives in a system of record. What is this customer's current balance? Is this claim approved as of right now? What is the open position on this instrument? Ask your AI architecture to answer it, end-to-end, within a single agent reasoning step, with full audit trail, without going through a human, a batch report, or a pre-aggregated dashboard.*

If your architecture can answer that in a regulated workload, you've solved the canonical truth problem. If it can't, you haven't — and most of the AI you've deployed is working on approximations of truth, not truth.

The reason mainframe access is the sharpest version of this test is not that mainframes are special. It's that mainframes hold the most constrained version of the problem: proprietary data formats (VSAM, IMS, Adabas), MSU-based licensing economics (MSU — Million Service Unit — is the traditional IBM mainframe billing unit; query patterns show up directly on the license bill), performance isolation requirements (the same workload is running the business in real time), and compliance boundaries older and harder than anything in the cloud-native stack. Annual cost varies substantially by contract type, peak-billing model, and specialty-processor use — practitioner estimates span a wide range and should be taken as directional — but at any point in that range, naïve analytical query patterns against a mainframe data source show up on the license bill within a quarter.

Solve canonical access against a mainframe and you've solved the hardest version of the problem. Most of the solution then generalizes to other systems of record that share the governance, structure, and transactional properties — even when they don't share the licensing economics. The inverse is the more important direction: if your AI architecture only works against your cloud data lake, your "AI strategy" is a strategy for a subset of your business — usually not the subset the business actually runs on.

That's why mainframe is the diagnostic, not the case study.

What canonical-AI-ready access actually requires

Stripping the vendor marketing away, the requirements for AI access to canonical data are narrower and harder than the requirements for AI access to contextual data. They are:

Structural fidelity. The access layer has to preserve the semantics of the source — relational integrity, hierarchical nesting, array and field-level metadata — not flatten it into prose tokens.

Transactional freshness. Queries have to reflect the state of the system of record inside the same transaction boundary as the decision being made. T-0 for consequential actions, or one of the weaker patterns above applied deliberately and with the same discipline.

Governance at the source. The authorization, audit, and data classification of the system of record have to propagate unchanged through the AI layer. Compliance cannot be reimplemented in the retrieval tier.

Runtime-exploratory query economics. The access layer has to support unpredictable query patterns without creating a proportional bill on the system of record. For mainframes, that means CDC, virtualization, or specialty-processor offload — not naïve repeated reads.

Protocol alignment. A stable open standard for AI-to-system communication, carrying auth, audit, and typed capabilities per call. MCP is the current leading candidate; something in its lineage is the future.

Organizations that meet these five requirements are positioned to deploy AI that acts on the authoritative record rather than on projections of it. Organizations that don't will continue to deploy AI that acts on approximations — and will continue to contribute to the 42% abandonment figure until they understand why.

What to do Monday

If you read nothing else, three actions are worth taking this week:

- ✓ **Run the diagnostic.** Pick one canonical question whose answer lives in a system of record in your organization. Try to answer it with your current AI architecture, end-to-end, under realistic governance. Observe where it fails.
- ✓ **Inventory your pilots by data type.** For each AI pilot or production system, classify its primary data dependency as contextual or canonical. You will likely find your successes concentrated in the first column and your failures concentrated in the second. That is the shape of the problem.
- ✓ **Measure in authority, not volume.** Stop reporting AI readiness in gigabytes ingested or documents embedded. Start reporting it in the fraction of your canonical systems your AI can reliably, auditably act on. That is the number that predicts whether your AI program survives the next budget cycle.

A brief note on solution, and where this series goes next

At Software AG, we work on this problem through CONNX, a data access and virtualization layer built specifically to make canonical enterprise data — Adabas, VSAM, IMS, Db2, and other systems of record — reachable by modern AI workloads without replatforming the systems that run the business. CONNX exposes these sources through the Model Context Protocol natively, preserves source-side governance across hybrid environments, and is engineered to keep MSUs overhead proportional to the value of the query rather than the volume of the reads.

How CONNX compares to the alternatives above — and where an MCP-native, canonical-data-first approach changes the economics of agentic AI on regulated workloads — is the subject of the next article in this series.

References

1. S&P Global Market Intelligence, Voice of the Enterprise: AI & Machine Learning, Use Cases 2025, n=1,006 North American and European organizations, fieldwork October 21–November 25, 2024. Reported that the share of organizations abandoning most of their AI initiatives rose from 17% to 42%, and that the average organization scrapped 46% of PoCs before production.
2. MIT Project NANDA, The GenAI Divide: State of AI in Business 2025, preliminary findings, July 2025. Methodology: systematic review of 300 publicly disclosed AI initiatives, interviews with representatives from 52 organizations, and survey responses from 153 senior leaders. Headline figures — 95% of organizations reporting zero measurable return, and 5% of integrated AI pilots extracting millions in value — have been subject to methodological critique, but the directional finding is corroborated by independent surveys cited here.
3. Ryseff, De Bruhl, and Newberry, The Root Causes of Failure for Artificial Intelligence Projects and How They Can Succeed, RAND Corporation, RR-A2680-1, 2024. The ~80% / 2× figure is RAND's synthesis of prior estimates referenced in the report, not an independent RAND measurement.
4. IBM Institute for Business Value, 2025 Chief Data Officer Study: The AI Multiplier Effect, n=1,700 senior data and analytics leaders (including CDOs and related roles) across 27 geographies and 19 industries, in cooperation with Oxford Economics; published November 13, 2025.
5. Informatica, CDO Insights 2025: Racing Ahead on GenAI and Data Investments While Navigating Potential Speed Bumps, n=600 data leaders, published January 2025. Top obstacles to moving GenAI from pilot to production: data issues (43%) and lack of technical maturity (43%), with employee skills and data literacy at 35%.
6. MuleSoft / Salesforce, 2025 Connectivity Benchmark Report, n=1,050 enterprise IT leaders. 95% of respondents struggle to integrate data across systems; 80% cite data integration as a major challenge for AI adoption.
7. Gartner press release, Lack of AI-Ready Data Puts AI Projects at Risk, February 26, 2025, drawing on a July 2024 Gartner survey of 1,203 data management leaders.
8. Gartner, widely cited directional estimate that roughly 70–80% of enterprise information is unstructured. The figure is frequently reproduced but its underlying methodology is not public; treated here as directional rather than precise.
9. IBM, citing IDC: "less than 1% of unstructured enterprise data is being used in generative AI today." Referenced in IBM "Think" insight pieces and IBV materials, 2025.
10. Regulation (EU) 2024/1689 ("AI Act"), Articles 12–15 on record-keeping, transparency/information, human oversight, and accuracy/robustness/cybersecurity for high-risk systems; GDPR Article 22 on decisions based solely on automated processing producing legal or similarly significant effects. FINRA Rule 3110 (supervision) and HIPAA 45 CFR §164.312 (technical safeguards) are control regimes offered here as analogues, not as rules written for agentic AI.

Take the next step

Find out what you can do with Software AG's technology

Let's connect