

A GUIDE FOR CIOs & ENTERPRISE DATA LEADERS

Why Your Contract AI Is Getting More Expensive — Not Less

Falling build costs, rising operating costs, and the architecture decision that bends the curve back down.

Naresh Kommula · Lead IT Architect — Enterprise CLM / Master Data · 2026

The Promise Was Lower Cost. Many Enterprises Got the Opposite.

Artificial intelligence was supposed to lower the cost of contract intelligence. Less manual review, faster extraction, better visibility. In the pilot, that is exactly what it does. Across the enterprise, many organizations are finding something different: **the cost of building any single AI capability is falling, while the cost of operating the resulting ecosystem is climbing**.

The reason is not a bad model or a bad vendor. It is that AI makes it dramatically easier to build a local solution than an enterprise one. Legal stands up an AI assistant for contract review. Procurement builds supplier analysis. Finance builds obligation-driven pricing and invoicing intelligence. Operations automates monitoring. Each initiative is rational, and each delivers value within its own domain.

Collectively, however, they create a growing landscape of disconnected capabilities, each with its own integrations, governance controls, security model, support process, and maintenance burden. The marginal cost of a single AI solution may be low. The cumulative cost of maintaining dozens of independent solutions is not.

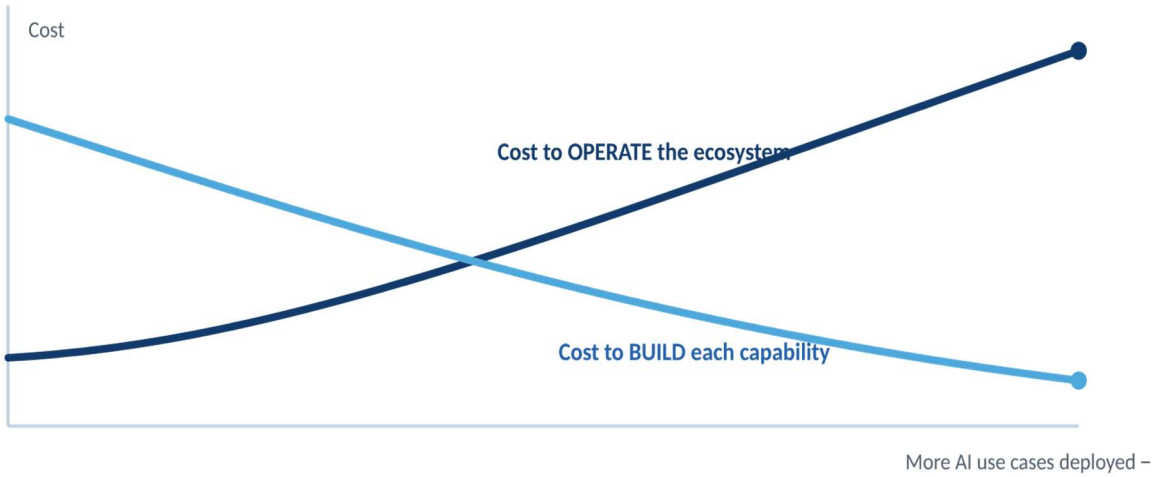


Figure 1 — As AI adoption accelerates, per-capability build cost falls while ecosystem operating cost rises.

AI lowers the cost of building contract intelligence while raising the cost of managing fragmented contract intelligence.

THREE FORCES BEHIND THE RISING COST

AI Doesn't Remove Fragmentation. It Amplifies It.

1 **It is easier to build local than enterprise.** Every function can stand up its own AI capability, and each adds its own integrations, governance, and maintenance. Individually cheap; collectively expensive.

2 **AI acts on the data it is given.** When the same obligation is represented differently across functions and systems, AI faithfully learns and operates against those differences at scale. A data-reconciliation problem quietly becomes an AI-governance problem.

3 **Agentic AI changes the nature of risk.** Traditional systems supported visibility. Agentic systems recommend, escalate, and increasingly execute. Inconsistent obligation definitions no longer produce only reporting discrepancies, they produce inconsistent actions across business processes.

The picture below contrasts two architectures. On the left is fragmentation as it exists today: the same contract extracted, mapped, classified, and governed again for every function and every new AI use case - nothing wrong on its own, which is precisely why the cost is so hard to see. On the right is the alternative: capture each obligation once, in a single governed foundation that every function and AI consumes without reinterpretation. Getting there is not a matter of buying another tool - it takes an operating model, which is where we turn next.

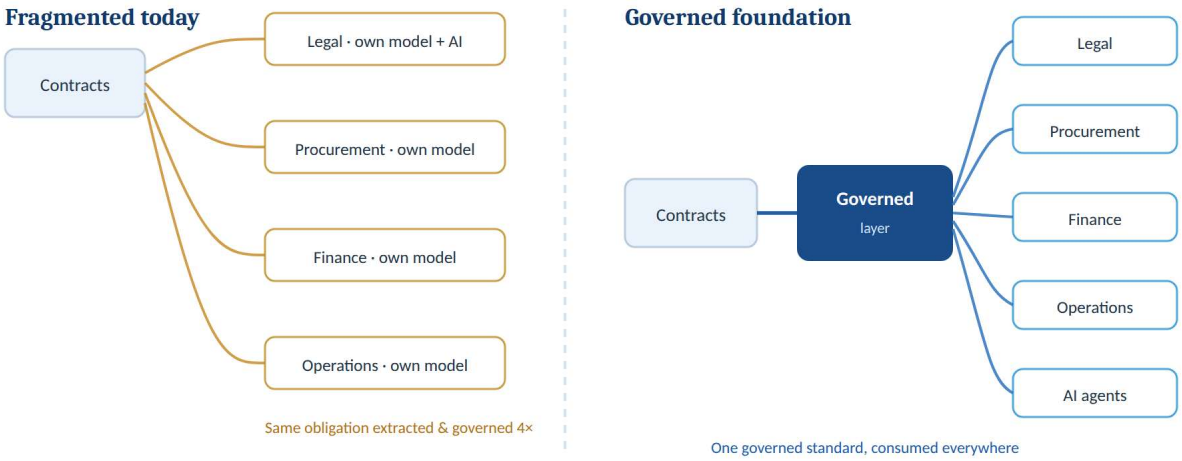


Figure 2 — Fragmented re-extraction (left) versus a single governed foundation every consumer reuses (right).

THE ROOT CAUSE

Four Patterns, One Root Cause

The AI-specific pressures described above do not exist in isolation. They are layered onto four longstanding organizational patterns that predate AI and significantly compound its impact. Together, these patterns quietly drive total cost of ownership higher, even as individual AI capabilities become less expensive to build. Most IT and data leaders will recognize them immediately.



The four recurring patterns

1. **Tool-first thinking.** Teams compare AI capabilities, features, and vendor roadmaps before aligning on how obligations should be governed, shared, and executed. Technology decisions get made before ownership, definitions, and governance responsibilities are settled.
2. **Weak obligation standardization.** The same commitment is represented differently by Legal, Finance, Procurement, and Operations. Without a canonical model, AI accelerates the processing of obligation data — but does not standardize what it means.
3. **Governance after execution.** Contracts are drafted, approved, and operationalized before decisions are made about obligation structure, stewardship, or interoperability — turning governance into reconciliation rather than alignment.
4. **Fragmented ownership.** Legal owns contract language, Procurement owns supplier commitments, Finance owns pricing, Operations owns delivery, Technology owns platforms — but no single function owns obligation intelligence end to end.

Taken together, these patterns mean the same obligation is repeatedly extracted, restructured, interpreted, and managed across different parts of the enterprise increasing total cost of ownership, governance complexity, and the difficulty of scaling AI with trust. This is the problem the COIF flywheel is built to reverse.

The COIF Framework at a Glance

So how does an enterprise get there? Not with another tool, but with an operating model. The Contract Obligation Intelligence Framework (COIF) is a flywheel that connects three reinforcing components: Contract Obligation Alignment (COA), the Governed Contract Data Layer (GCDL), and Connected Contract Intelligence (CCI), around a center axle of Continuous Governance. Each component strengthens the next, so contract intelligence becomes a reusable enterprise capability rather than a set of disconnected, per-function projects.



Figure 3 — The COIF flywheel: three reinforcing components turning on the axle of Continuous Governance.

Better obligation alignment (COA) creates a common language at the source. Better governed data (GCDL) lets every function consume the same trusted foundation. Better connected intelligence (CCI) compares commitments against actual outcomes and those insights feed back into how future obligations are drafted and governed.

In a fragmented model, each new AI capability adds complexity. Inside the flywheel, each new AI capability compounds value.

How Each Component Answers a Pattern

How COIF answers each pattern

- **Tool-first thinking** is addressed by COA, which establishes shared obligation meaning before platform decisions dominate execution.
- **Weak standardization** is addressed by GCDL, which creates a vendor-agnostic enterprise standard for contract meaning.
- **Governance after execution** is addressed by COA and Continuous Governance, which move governance earlier and sustain it.
- **Fragmented ownership** is addressed by Continuous Governance, which defines stewardship, decision rights, and accountability.

Continuous Governance: four triggers that keep it active

Continuous Governance is not a one-time setup, it activates on: contract amendments that change terms or obligations; threshold breaches where actual performance deviates from commitments; regulatory changes affecting how obligations must be interpreted; and audit findings that surface gaps between what contracts require and what systems have recorded. Each trigger propagates changes across every function and system that acts on the affected obligation.



Contract Obligation Intelligence Framework (COIF) and the Governed Contract Data Layer (GCDL) for AI-Ready Contract Data

THE ARCHITECTURAL FIX

The Governed Contract Data Layer

Of the three components, the GCDL is where the cost curve actually bends, so it is worth examining closely. As the foundation of the COIF flywheel, its principle is simple to state and hard to enforce: capture each contract obligation once, in a standardized, vendor-agnostic form, with lineage back to source language, and make it reusable by every downstream function and system that depends on it.

What GCDL is, and is not

GCDL is not a CLM feature, a repository, or a product you buy. It is the enterprise standard for how obligation data is structured, governed, validated, traced, and reused across departments. The enterprise defines the standard first; platforms are configured to consume or conform to it.

In too many enterprises, the platform becomes the data standard. Obligations get structured according to a tool's data model, governed through that tool's workflow, and exposed through that tool's integrations. Add a second platform in a second function and the enterprise now owns two "standards" and two interpretations of contract meaning. GCDL reverses that relationship.

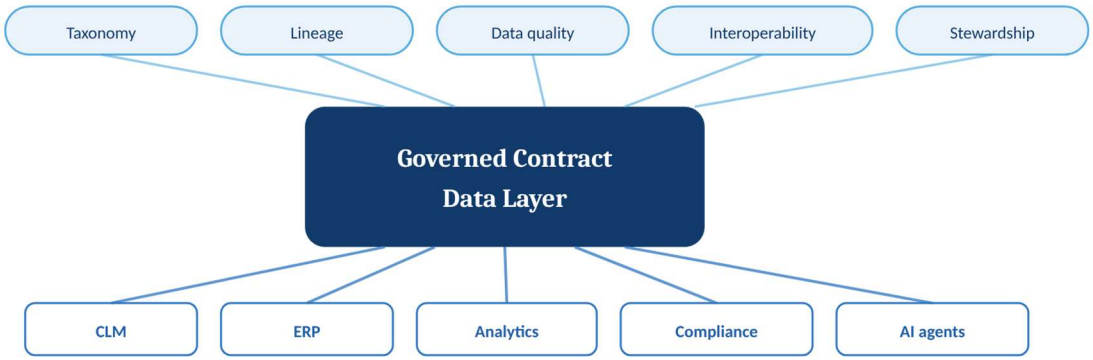


Figure 4 — Capture once, govern everywhere: one standard defines the attributes; every system consumes it without reinterpretation.

Where it lives is deliberately flexible

GCDL can be a governed schema inside an existing CLM or enterprise data platform or MDM environment, a dedicated obligation store that aggregates from multiple source systems, or a design standard enforced through integration and data-quality controls. What matters is not the storage location, but that a single governed representation of obligation meaning exists. One that every function, system, and AI agent consumes without reinterpretation.

THE ECONOMICS

Bending the Cost Curve

This is the structural change that bends the curve. In a fragmented model, every new AI use case needs new extraction, mapping, integration, governance, and reconciliation, so each capability adds complexity. On a governed foundation, new AI and agentic capabilities reuse the same obligation standard, so each capability compounds value.

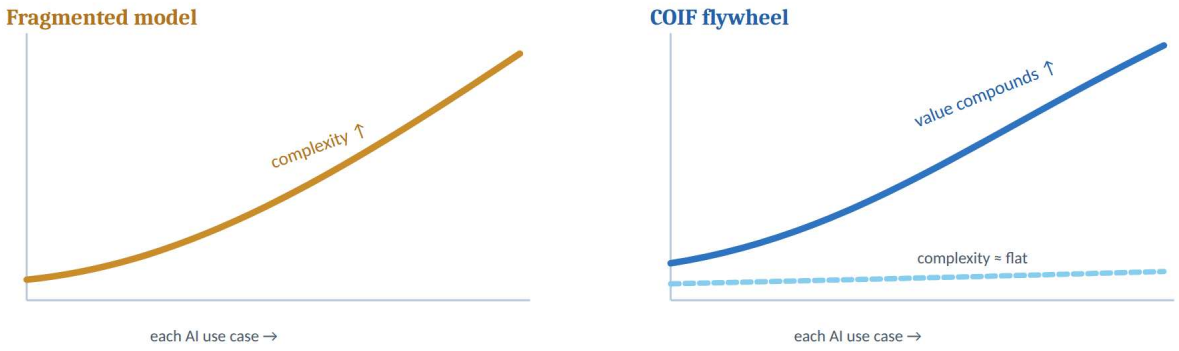


Figure 5 — In a fragmented model each capability adds complexity; inside the flywheel each capability compounds value.

When obligations are aligned at the source and standardized through GCDL, new AI use cases no longer recreate contract understanding from scratch. AI systems consume one governed standard rather than negotiating with multiple silos, which is why value compounds while governance complexity and operational overhead grow far more slowly. This is also what makes Connected Contract Intelligence (CCI) tractable rather than another isolated build: AI-assisted obligation monitoring, revenue-leakage detection that validates pricing and rebates against billing activity, supplier and customer performance management, renewal and expiration intelligence, and agentic workflows that route and escalate exceptions, all-consuming the same governed data instead of re-deriving it per function.

What this means for vendor and platform strategy

- **Define the obligation model before the tool.** When the platform becomes the standard, contract meaning is constrained by one vendor's data model, and multiple platforms breed multiple standards.
- **Evaluate platforms on governance, not just features.** Can the tool align to the enterprise taxonomy, support lineage back to source language, and let obligation data be reused by downstream systems and AI?
- **Sequence vendor selection after architecture.** When standards come first, vendor decisions get easier because the evaluation criteria are already clear.

The question changes from "Which platform should define contract intelligence?" to "What obligation standard should every platform consume?" — THE COIF PRINCIPLE

LEADERSHIP TAKEAWAY

Get the Sequence Right

The implication is not that organizations should slow down AI adoption. It is that the sequence must change. Many enterprises are trying to scale AI before establishing a shared foundation for obligation data, governance, and execution, so they automate fragmentation rather than eliminating it.

For technology leaders, the highest-leverage decision in a contract-AI roadmap is not which model or which platform. It is whether the enterprise defines the obligation standard before the tools do. Get that sequence right and AI value compounds while governance overhead grows slowly. Get it wrong and the organization spends years reconciling platform-driven definitions it never chose.

Three principles to carry into the next budget cycle

1. **Govern contract meaning before scaling AI.** AI can only be as reliable as the obligation data it consumes. Prioritize a common operational language and a governed foundation before expanding agentic workflows.

2. **Treat contract intelligence as an enterprise capability.** Contracts influence operational, financial, compliance, and customer outcomes at once, so obligation intelligence should be governed cross-functionally, not rebuilt per department.

3. **Build once, reuse everywhere.** A single governed representation of obligations, consumed across functions, systems, and AI, reduces duplication, lowers total cost of ownership, and accelerates future innovation.

The organizations that gain the most from AI will not be those that deploy the most AI tools. They will be those that establish the strongest foundation for how contractual obligations are defined, governed, connected, and executed across the enterprise.

About this framework

This guide draws on the Contract Obligation Intelligence Framework (COIF) and its foundation, the Governed Contract Data Layer (GCDL), an enterprise operating model for turning contractual obligations into governed, reusable, executable intelligence across people, systems, and AI. © 2026 Naresh Kommula.