

WHITE PAPER

The Signal Behind the Noise:

How AI Hype is Obscuring the Risks Enterprises Cannot Afford to Ignore

08 March 2026 | Enterprise Risk & Technology Governance | By SK Joshi

Executive Summary

The race to deploy Artificial Intelligence across enterprise functions has accelerated dramatically. AI laboratories are announcing capabilities at a pace that outstrips the ability of most organizations to evaluate them. Boards are asking about AI strategy. Vendors are promising transformative productivity gains. And employees are already using consumer AI tools, regardless of whether IT has approved them.

In this environment, a dangerous phenomenon is taking hold: hype is doing the work that governance should be doing. Organizations are adopting AI not because they have assessed the technology rigorously, but because the pressure to not be left behind feels more urgent than the discipline to ask hard questions.

This paper argues that the most complex, consequential risks of enterprise AI -data exposure, accuracy failure, agentic autonomy, shadow usage, and accountability gaps -are being systematically obscured by a narrative manufactured by labs with strong commercial incentives to accelerate adoption. The enterprises that will navigate this era well are not the ones that moved fastest. They are the ones that moved deliberately.

Core Thesis

AI is not uniquely dangerous compared to other enterprise technologies. It is being adopted with uniquely low governance rigor relative to its actual risk profile and much faster pace. That gap - between capability claims and verified reliability - is where enterprises get hurt.

1. The Hype Machine: How We Got Here

1.1 Commercial Incentives Shape the Narrative

AI laboratories are among the most richly valued companies in history. OpenAI, Anthropic, Google DeepMind, and Meta AI collectively represent hundreds of billions of dollars in investment and market capitalization. These organizations are in a race -not merely for technical supremacy, but for enterprise adoption, which translates directly into revenue, data, and strategic positioning.

This context matters because it shapes how capability claims are framed. Benchmarks are selected to showcase strengths. Failure modes are minimized in marketing materials. The narrative of "AI will do most white-collar work" serves a clear commercial purpose: it creates urgency that overrides careful evaluation. Enterprises that pause to assess risk feel like they are falling behind.

1.2 The Benchmark Problem

Labs publish impressive scores on standardized tests- reasoning benchmarks, mathematical olympiad problems, legal bar exam simulations, medical licensing tests. These numbers are real. They are also largely irrelevant to most enterprise use cases.

A benchmark is a controlled, clean problem with a known correct answer, designed by researchers to measure a specific capability. Enterprise work is none of these things. Your finance team's spreadsheets are not clean. Your customer data is not standardized. Your business logic is embedded in years of institutional knowledge that no model has seen. The gap between benchmark performance and real-world reliability is vast and almost never discussed in sales conversations.

The Benchmark Fallacy

Scoring 90% on a legal reasoning benchmark does not mean an AI will be correct 90% of the time on your legal documents. It means it was correct 90% of the time on a curated dataset of legal problems designed by researchers. These are fundamentally different claims.

1.3 The "Jobs Will Be Replaced" Narrative

Claims that AI will replace most white-collar workers within years serve multiple purposes for labs: they create investor excitement, they pressure enterprises to adopt quickly, and they position the labs as historically significant actors rather than software vendors. The reality is considerably more nuanced.

AI will restructure white-collar work more than replace it - at least for any work where the output carries real consequence. The analyst who uses AI well will outperform the one who doesn't. But the work of judgment, verification, contextual reasoning, and accountability does not disappear. It shifts. And in many cases, the cognitive burden of validating AI outputs is significant enough to erode the productivity gains (with high data privacy risk) that justified adoption in the first place.

2. The Accuracy Illusion

2.1 Stochastic Systems in Deterministic Roles

Enterprise production systems rest on a foundational assumption, given the same input, you get the same output. This is what makes systems auditable, debuggable, and trustworthy. It is what compliance frameworks are built on. It is what makes a spreadsheet formula reliable.

Large language models violate this assumption at the architecture level. They are probabilistic systems. Even at the lowest temperature settings, the same prompt can produce different reasoning chains across model versions, context window states, and inference conditions. Enterprises are being asked to integrate a fundamentally stochastic technology into deterministic trust frameworks, and the gap between those two paradigms is where risk lives.

2.2 A Case Study: The Finance Copilot

Consider a practical scenario: an AI copilot integrated into a finance team's Excel workflow. The lab's pitch is compelling - faster analysis, automated formula generation, natural language queries over financial data. The productivity case is real.

Now consider what the accuracy claims do not cover:

- A formula is syntactically correct and executes without error, but applies the right logic to the wrong data range.
- A DCF model terminal value calculation contains a subtle error that is plausible but wrong - within the range that would pass a casual review.
- Context fidelity degrades across a large spreadsheet; the model loses track of the fact that column F is an override that supersedes column D.
- Business logic embedded in years of institutional practice - seasonality adjustments, intercompany eliminations, one-time items - is invisible to the model unless explicitly provided, and even then may not be applied correctly in edge cases.
- Numerical reasoning remains architecturally weak. LLMs predict what computation looks like; they do not compute. For simple arithmetic this mostly works. For multi-step financial modeling with interdependencies, failure modes are subtle.

None of these failures produce an error message. All of them produce a plausible-looking output. The analyst reviewing the work is now in a cognitively difficult position: they are being asked to validate work that looks authoritative, was produced faster than they could do it themselves, and contains errors that are designed - by the nature of how language models work - to look reasonable.

2.3 The "Looks Right" Problem

A wrong number that is obviously wrong is far less dangerous than a wrong number that is plausible. AI outputs tend to be in the right ballpark. They have the right units, the right sign, the right order of magnitude. This property - which makes AI outputs impressive in demos - is precisely what makes them dangerous in production. It systematically defeats the casual review that time-pressured enterprise teams actually perform.

Traditional software fails loudly: a crash, an error code, an exception. AI fails quietly, producing confident, well-formatted, plausible-but-wrong outputs. There is no equivalent of a stack trace to help you find where the reasoning went wrong.

What Evals Actually Test	What Enterprise Work Actually Requires
Known problems with known correct answers	Novel problems with contextual correct answers
Clean, standardized input data	Messy, inconsistent, real-world data
Performance on anticipated scenarios	Reliability in unanticipated edge cases
Average accuracy across test set	Consistency on every individual output
Capability to perform a task type	Reliability in your specific domain context
Researcher-curated benchmark inputs	Institutional logic baked into years of practice

3. The Deprecation Trap: When the Model Expires Before You Finish Deploying It

3.1 A Concrete Example the Industry Is Not Discussing Honestly

In 2026, OpenAI retired GPT-5, GPT-4o, GPT-4.1, GPT-4.1 mini, and o4-mini from ChatGPT - models that were, in most cases, less than twelve months old. GPT-5 itself was only three to four months into production when its predecessors were being wound down. This is not an isolated product decision. It is a signal about the operating tempo of AI labs and what that tempo means for enterprises that build on their platforms.

The labs frame model deprecation as progress - the new model is better, the old one is being retired, this is how technology advances. That framing is accurate from the lab's perspective. From the enterprise architecture perspective, it describes something else entirely: a platform that changes faster than enterprise governance cycles can follow.

3.2 The Enterprise Upgrade Cycle Is Not Designed for This Tempo

When an enterprise adopts or upgrades a foundational AI model, the work involved is not trivial. A responsible upgrade cycle looks something like this:

#	Phase	Typical Duration	Reusable on Next Upgrade?
1	New model launches - awareness, initial evaluation, internal briefing	2–4 weeks	No - starts fresh each cycle
2	Fitness testing - domain-specific evals against your actual workflows and data	4–8 weeks	Partially - test harness reusable, cases need refresh
3	Upgrade plan drafted - architecture changes, integration points mapped, resourcing assessed	2–4 weeks	Partially - templates reusable, specifics change
4	Management review and budget approval - business case, risk sign-off, procurement	4–8 weeks	No - repeated each cycle
5	Integration - model swap, prompt re-engineering, API changes, tooling updates	6–12 weeks	No - model-specific work
6	Comprehensive output validation - regression testing across all production use cases	4–8 weeks	No - must be re-run in full
7	Monitoring, auditing, and logging updates - new model behaviours require new detection logic	3–6 weeks	Partially - framework reusable, rules need rebuild
8	Sanity checks and phased rollout - production validation before full deployment	2–4 weeks	No - repeated each cycle

The Governance Cycle Mismatch

End-to-end: 7–12 months for a responsible enterprise model upgrade. Average frontier model lifespan before deprecation or significant capability shift: 6–12 months. These two numbers are not compatible. By the time a responsible enterprise completes a full upgrade cycle, the model it just upgraded to may already be on the retirement list.

3.3 The Hidden Cost Structure

Labs and vendors discuss model upgrades as if the cost is primarily technical - swap the API endpoint, update the prompts, done. The true cost structure is considerably more complex, and most of it is governance overhead that does not scale down regardless of how mature your AI infrastructure becomes.

Infrastructure work - monitoring frameworks, logging pipelines, audit trail architecture - is partially reusable across upgrade cycles, though it always requires recalibration as new models exhibit new behavioral patterns. The more substantial cost is the work that cannot be reused: domain-specific validation, regression testing against production use cases, approval workflows, and the management overhead of justifying the same investment again on a shorter cycle than anyone planned for.

For enterprises running AI across multiple business functions - finance, legal, HR, customer operations, engineering - this overhead multiplies. Each function has its own validation requirements, its own stakeholders, its own approval chain. A single model deprecation event can trigger parallel upgrade cycles across dozens of teams, each starting from scratch on the work that cannot be templated.

3.4 The Strategic Dependency Risk

Beyond the upgrade cycle cost, there is a strategic risk that receives almost no attention: enterprises that build deeply on a specific model's capabilities are accepting dependency on a vendor roadmap they cannot influence and cannot predict. When GPT-4o was deprecated, organizations that had fine-tuned workflows around its specific behavioral characteristics - its particular approach to formatting, its response patterns, its specific capability profile - found that the "better" replacement model was better on benchmarks but different in ways that broke their production workflows.

"Better" and "compatible" are not the same thing. A model that scores higher on reasoning benchmarks but responds differently to your specific prompts, handles your specific edge cases differently, or formats outputs differently than its predecessor is not an upgrade for your organization - it is a migration project.

3.5 What This Means for Enterprise AI Architecture

The deprecation cycle is not an argument against using AI. It is an argument for a specific kind of architectural discipline that the hype cycle actively discourages. The enterprises that manage this well share several characteristics:

- They treat model selection as a long-term architectural decision, not a best-available-today decision. Newer is not automatically better when the upgrade cost is high.

- They invest in abstraction layers between their applications and the underlying model, so that model swaps do not cascade into full application re-engineering.
- They maintain realistic model lifecycle assumptions in their technology roadmaps - treating 12–18 months as a planning horizon rather than assuming indefinite model availability.
- They resist vendor pressure to adopt the latest model immediately, recognizing that being on the leading edge of the deprecation curve maximizes upgrade frequency and cost.
- They distinguish between experimental deployments - where following the latest model makes sense - and production deployments, where stability and predictability outweigh marginal capability gains.

The hype cycle moves fast. Enterprise architecture should not be required to keep up. An organization that redesigns production governance every nine months to chase model releases is not innovating - it is running on a treadmill the vendor controls.

4. The Agentic AI Problem

4.1 Agency Requires Trust. Trust Requires Reliability.

The next major wave being pushed by AI labs is agentic AI - systems that do not merely generate text but take actions: querying databases, sending communications, executing code, making purchases, interacting with external services. The commercial case is compelling. The technical readiness is not.

Agency amplifies every reliability problem that exists in generative AI. A language model that is wrong 0.1% of the time is an acceptable writing assistant. The same model, given the ability to take irreversible actions, is wrong 0.1% of the time in ways that have real-world consequences. At enterprise scale - 100,000 daily operations - that is 100 consequential failures per day.

4.2 The Cascading Error Problem

Agentic systems do not make single decisions. They execute multi-step workflows where each step depends on the output of the previous one. A subtly wrong assumption in step 2 of a 15-step process compounds through the remaining steps. By step 15, the accumulated error can

be significant - and the chain of reasoning is often opaque enough that reconstructing what went wrong is not straightforward.

Unlike a software bug, which can be traced through a deterministic execution path, an LLM reasoning chain does not have a reliable audit trail. The model does not "know" why it made a particular decision in step 4. It generated the most probable next token given its context. Debugging that is a fundamentally different challenge than debugging conventional software.

4.3 Prompt Injection: The Underappreciated Attack Vector

If an agent reads external content - emails, web pages, documents, database results - and acts on that content, a new attack surface opens that has no equivalent in conventional software security. Malicious actors can embed instructions within content that the agent will read, hijacking the agent's behavior.

An agent that reads customer emails and takes action on them can be weaponized by any customer who knows how to craft a prompt injection payload. An agent that browses the web can be manipulated by any website it visits. There is currently no robust, general-purpose defense against this class of attack. It is not a theoretical concern - it has been demonstrated repeatedly in research and in the wild.

4.4 The Authorization Creep Problem

Agents need permissions to act. In practice, organizations tend to over-provision those permissions because restricting them breaks the workflow. A useful agent that needs read-write access to customer records, the ability to send emails, and access to financial systems has - if compromised or manipulated - the blast radius of a senior employee with poor judgment and no accountability structure.

The Practical Test

Before deploying any agentic system, ask: Would I give a very capable, very confident intern with no common sense, no institutional memory, and no personal accountability this level of access to production systems? If the answer is no, the agent should not have that access either.

4.5 The Accountability Vacuum

When an agent makes a consequential error - sends incorrect customer communications, deletes the wrong records, approves an unauthorized transaction - who is accountable? The AI vendor? The enterprise that deployed it? The employee who configured the workflow? Legal and compliance frameworks are years behind the deployment curve. Enterprises adopting agentic AI today are accepting legal exposure that has not yet been clearly defined by regulators or tested in courts.

5. The Shadow AI Crisis

5.1 The Data Has Already Left

For most enterprises, the question of whether employee data is flowing to external AI systems is no longer hypothetical. It is happening. Employees are using ChatGPT, Claude, Gemini, and Perplexity on personal accounts, on free tiers, on mobile devices connected to corporate networks. They are doing this not maliciously but because the tools are genuinely useful and the friction to use them is near zero.

What is flowing through those channels includes customer PII, internal financial models, proprietary source code, legal contracts, M&A documents, and HR data. The uncomfortable truth for most enterprises is that the breach they are worried about preventing has, in many cases, already occurred. They simply have not measured it.

5.2 Why Restrictive Policies Make This Worse

The instinct to respond to shadow AI with restrictive policies is understandable but counterproductive in isolation. If an enterprise bans AI usage without providing good alternatives, employees do not stop using AI. They continue using it on personal accounts, with zero corporate oversight, through channels with far weaker data handling commitments than enterprise-grade agreements would provide. Restriction without provision accelerates the shadow behavior it is intended to prevent.

5.3 The Enterprise Agreement Illusion

Enterprise agreements with AI vendors - committing to no training on customer data, providing data residency guarantees, offering audit logs - are meaningful but frequently misunderstood. They address contractual risk, not technical risk. There is no cryptographic mechanism preventing a vendor from accessing your data; there is a legal agreement creating liability if they do. These are not equivalent.

Beyond the training data question, enterprise agreements typically say little about: data in transit vulnerabilities, prompt logging for abuse detection, inference infrastructure security, supply chain risk through the vendor's own cloud providers, or insider threat at the AI lab. Each of these represents a residual risk that the contract does not eliminate.

What Enterprise Agreements Provide	What They Do Not Provide
No training on your data (contractual)	Technical guarantee data won't be accessed
Data residency commitments	Air-gap isolation from internet exposure
SOC 2 Type II certification	Visibility into vendor's actual security controls
Audit logs of your usage	Protection against inference-time exposure
Legal liability if breached	Prevention of the breach itself
DPA compliance documentation	Supply chain security through vendor's providers

6. A Framework for Responsible Enterprise AI

6.1 Reject the Binary

The choice is not between adopting AI broadly and rejecting it entirely. That framing is a false binary encouraged by the commercial interests of AI labs. The productive question is: for which specific use cases, with which specific data, under which specific governance conditions, does AI offer value that justifies its risk profile?

6.2 Data Sensitivity Tiering

Not all enterprise data warrants the same protection, and not all AI deployments carry the same risk. A tiered architecture that matches deployment model to data sensitivity is the foundation of a defensible AI governance strategy:

Tier	Data Type	Recommended Deployment	Risk Level
Tier 1	Public, non-sensitive work. Marketing copy, public research, general coding on non-proprietary problems.	Approved SaaS Enterprise plan (Claude Enterprise, ChatGPT Enterprise)	Low
Tier 2	Internal, non-regulated. Internal documents, business analysis, code on non-core IP, general productivity.	Private cloud VPC deployment. AWS Bedrock, Azure OpenAI with your own network controls.	Medium
Tier 3	Regulated, sensitive, IP-critical. PII, financials, M&A activity, core algorithms, legal contracts.	Air-gapped self-hosted open-source models, or no AI usage. Full on-premise deployment.	High - requires maximum control

6.3 The Five Governance Prerequisites

Before deploying AI in any consequential enterprise workflow, five conditions should be met:

- **You cannot govern what you cannot see. Logging, monitoring, and auditing of AI usage must be in place before scale, not after. This includes detecting wrong-but-plausible outputs, not just errors and exceptions.** Observability first.
- **If an AI action cannot be undone, a human must approve it before execution. Irreversibility is the line between AI as a tool and AI as an actor.** Reversibility as a hard requirement.
- **Define acceptable error rates before deployment, not after the first incident. If your domain cannot tolerate a 0.1% error rate on consequential decisions, that is a deployment blocker, not a post-launch optimization target.** Explicit failure budgets.
- **If your team cannot reliably catch AI errors in a given domain, you should not be using AI autonomously in that domain. The capacity to verify output is a deployment condition, not an afterthought.** Verification capability as a prerequisite.
- **The DPA must be reviewed by legal. Data classification must be led by compliance, not by the business units who are incentivized to move fast. These gates exist for a reason.** Legal and compliance sign-off on data flows.

6.4 Human-in-the-Loop Is Not a Limitation

The AI industry often frames human oversight as a temporary limitation to be engineered away. From an enterprise governance perspective, human oversight is a feature - particularly for consequential decisions. The goal should not be to remove humans from AI workflows as fast as possible. It should be to identify precisely which decisions are safe to automate, and to maintain meaningful human oversight everywhere else.

Claude Code works precisely because the human is the production system. The agent proposes; the developer validates; the developer deploys. The intelligence of the AI is captured without ceding accountability. This architecture - AI as amplifier, human as decision-maker - is not a stepping stone to full autonomy. For many enterprise use cases, it is the correct end state.

7. The Economics of Doing It Right: When Does the Cost Justify Itself?

Every governance conversation in this paper has a price tag attached.

- Monitoring infrastructure.
- Eval suites.
- Human review queues.
- Compliance sign-off cycles.
- Legal review of vendor agreements.
- Tiered deployment architecture.
- Security tooling for shadow AI detection.

The question every CFO will eventually ask - is: what does doing this responsibly actually cost, and when does that cost justify itself?

The answer is not comfortable, but it is knowable. The enterprises that are getting this right are the ones that have run the full cost model - not just the model inference bill.

7.1 The Three Cost Calculations Most Enterprises Only Run One Of

There are three distinct economic questions that need separate answers, and they cascade. Most enterprises run only the first one before making deployment decisions.

The first is unit economics: what does it cost to produce one AI-assisted output versus one human-produced output? This is the calculation vendors make easy. Token cost divided by outputs, compared against hourly labor rate. The productivity case usually looks compelling here, and for narrow, well-defined tasks, it often is.

The second is risk-adjusted economics: what is the expected cost of failures, and how does that change the unit math? A 0.5% error rate on customer credit decisions is not a rounding error - it is an actuarial exposure. A wrong compliance output that triggers a regulatory review has a cost that can be orders of magnitude larger than the token savings that year. Risk-adjusted ROI requires estimating failure probability, consequence severity, and detection lag. Almost no enterprise does this before deployment.

The third is total system economics: what is the full cost of building, governing, validating, maintaining, and upgrading the AI deployment over its operational life? This is where the real number lives - and it is almost always larger than anticipated, for the reasons this section will explore.

The Calculation Most miss

ROI = (Token cost per task) vs. (Human cost per task)

Real ROI = (Token cost + Governance cost + Failure cost + Upgrade cost + Opportunity cost of diverted engineering) vs. (Human cost per task + Productivity uplift value)

These are not the same number. The gap between them is where most enterprise AI business cases quietly fall apart.

7.2 The Governance Cost Multiplier

The single most underestimated cost in enterprise AI is the governance layer itself. Every robust mechanism that makes AI deployment responsible - evals, monitoring, human-in-the-loop gates, audit logging, compliance review - has a price. And critically, many of these costs scale with usage volume in ways that erode the unit economics case as you grow.

Cost Layer	One-Time Build Cost	Recurring Cost	Reusable After Upgrade?
Model inference (production)	None	Scales directly with usage volume	Yes - same billing
Eval suite development	6–12 weeks senior engineering	Rebuild/refresh each model upgrade	Partially - harness reusable
Monitoring & alerting infra	4–8 weeks engineering + tooling	Storage, compute, ongoing tuning	Mostly - rules need recalibration
Human review queues (HITL)	Process design, tooling	Scales with flagged output volume	Yes - process reusable
Audit logging & compliance	Legal review + infra build	Storage + periodic audit labor	Mostly reusable
Governance overhead per call	None	3–5x inference cost for robust monitoring	No - paid per call
Shadow AI detection tooling	4–6 weeks security engineering	Ongoing tuning + response	Yes
Model upgrade cycles	None	7–12 months engineering every 12–18 months	Partially
Failure remediation	None	Variable - actuarial exposure per domain	No - domain-specific

The governance cost multiplier - the ratio of total governance spend to raw inference spend - is the number almost no enterprise has calculated before deployment. For a responsibly governed production workflow with meaningful accuracy requirements, a multiplier of 3x to 6x over raw inference cost is a realistic planning figure. At high usage volume, this is the dominant cost, not the model bill.

7.3 The Eval Paradox

Evals are the primary mechanism labs recommend for verifying AI output quality. They are necessary. They are also expensive in ways that compound over time, and they create a subtle paradox: the more rigorous your eval suite, the more it costs to run - but running it costs tokens, which means your quality assurance mechanism is itself an AI inference cost center.

The deeper problem is that evals go stale. As your production data distribution shifts - new products, new customer segments, regulatory changes, market events - your eval suite stops

reflecting reality. Maintaining eval coverage is not a one-time investment. It is a continuous operational cost that scales with how fast your business changes.

7.4 The Human-in-the-Loop Tax

Every HITL gate that is genuinely meaningful adds two costs: latency and labor. The latency cost is often acceptable. The labor cost is where the productivity math gets uncomfortable.

If AI generates a first draft in 30 seconds that previously took 2 hours, the time saving looks dramatic. But if a qualified human must spend 45 minutes reviewing and validating that draft - because the consequences of error are material and the AI's plausible-but-wrong failure mode makes casual review insufficient - the net saving is 75 minutes, not 2 hours. And the reviewer is doing a cognitively harder task than they would have done creating from scratch, because auditing someone else's work under time pressure requires sustained scepticism rather than generative thinking.

The HITL tax compounds at scale. An enterprise processing 10,000 AI-assisted outputs per day with a 5% human review rate requires 500 human reviews per day. At 20 minutes per review, that is 167 person-hours of review labor daily. This headcount requirement does not appear in the vendor's cost model. It appears in the staffing plan.

7.5 The Upgrade Cost as a Recurring Budget Line

Every model upgrade cycle is not just an engineering project - it is a budget event. The question finance leadership needs to answer is: have we provisioned for this as a recurring capital cost, or are we treating it as a one-time integration expense?

The honest answer for most enterprises is the latter. The initial AI integration is budgeted as a project. The ongoing upgrade cycles are absorbed as operational overhead or funded through emergency budget requests. Neither approach is appropriate for a technology with a 12-to-18 month model lifecycle. The sustainable model is to treat AI infrastructure maintenance - including upgrade cycles - as a recurring budget line, sized at roughly 40 to 60 percent of the original integration cost per cycle.

7.6 When the Economics Actually Work

None of this means enterprise AI economics are unfavourable. They are favourable in specific, identifiable conditions. The honest framing is that the conditions under which AI pays for itself - including governance - are narrower than the hype suggests, but they are real and they are worth building toward.

The economics work clearly when:

- **The governance infrastructure cost amortizes across millions of calls, not thousands. Document classification, first-pass triage, structured data extraction - high volume, well-defined, auditable. The per-unit economics look good and the eval suite is maintainable because the problem space is stable.** Volume is high and tasks are structurally repetitive.
- **The HITL tax and failure cost are minimized when errors are catchable before they cause harm. Drafting assistance, internal summarization, non-customer-facing analysis - the human is downstream of the AI, not dependent on it. Error recovery is cheap.** Failure consequences are bounded and reversible.
- **AI does not need to be perfect to win the economics - it needs to be faster or cheaper than the alternative, after accounting for governance. In domains where qualified human labor is scarce and expensive, even a high governance cost can justify the investment.** The human baseline is genuinely slow or expensive.
- **If the problem space is well-defined and changes slowly - standard contract clauses, known regulatory frameworks, established data formats - the eval maintenance cost is manageable. If the problem space shifts frequently, eval churn becomes a significant cost driver.** The use case is stable enough that evals don't go stale quickly.
- **Organizations that have provisioned the upgrade cost as a recurring budget line, and have built abstraction layers that reduce migration effort, can sustain the economics through model transitions. Those that haven't will face periodic budget crises every time a major model is deprecated.** The deployment is sized to absorb upgrade cycles.

The economics break down - or require careful scrutiny - when the failure consequence is high and hard to detect, when volume is low and the governance amortization math doesn't work, when the human reviewer spends nearly as long validating as they would have spent creating, or when the problem space changes fast enough that eval maintenance becomes a dominant cost.

7.7 The Cost Question Every Board Should Ask

Before approving any significant AI deployment, boards and executive leadership should require answers to five economic questions that almost no vendor pitch addresses:

- **What is the ratio of total governance spend - evals, monitoring, HITL labor, compliance, security - to raw inference cost? If the answer is unknown, the business case is incomplete.** What is our governance cost multiplier?
- **For this specific use case, what is the expected annual cost of AI errors - including remediation, regulatory exposure, customer impact, and reputational risk? This number belongs in the denominator of every ROI calculation.** What is our risk-adjusted failure cost?
- **Have we provisioned for model deprecation and upgrade as a recurring cost? At what frequency and at what capital cost per cycle?** What is our upgrade cycle budget?
- **What volume of human review does our deployment require, and have we staffed and budgeted for it? Or are we assuming review happens in the margins of existing roles?** What is our HITL labor cost?
- **Inference, governance, failures, upgrades, and HITL labor - summed over a realistic operating horizon with conservative assumptions about model lifecycle and error rates. Does the productivity value exceed that number?** What is the total system cost over three years?

The AI labs have made it easy to calculate what AI costs per token. The harder calculation - what responsible AI costs per outcome - is the one that determines

whether the investment makes sense. Enterprises that run only the first calculation are not saving money. They are deferring costs they have not yet seen.

7.8 The Infrastructure Cost Stack: What Actually Sits Around the Model

Pay-per-use inference pricing is the number AI vendors put in their decks. It is also the smallest and simplest part of the true infrastructure cost. Around every model call is a layer of engineering that the vendor does not provide and that does not appear in any ROI calculator.

Cloud providers - AWS Bedrock, Azure OpenAI, Google Vertex AI - provide the model endpoint. They add their own margin on top of the lab's price. What they do not provide (or partially provide) is the surrounding infrastructure that makes enterprise deployment viable: API gateway and rate limiting, authentication and authorisation per team and per workflow, prompt versioning and management, response caching to avoid paying twice for identical queries, retry logic and fallback handling when the API returns errors, and cost allocation tooling to know which team or workflow is spending what. Every one of these is an engineering build, a maintenance burden, and in some cases a separate vendor contract. Most importantly to really understand all these areas, you need AI expertise which many organizations are missing.

For enterprises choosing self-hosted open-source models to keep data inside the perimeter, the infrastructure cost is more visible but no smaller. GPU cluster provisioning, model serving frameworks, auto-scaling infrastructure, health monitoring, and the specialised ML engineering talent to operate it - none of this is covered by the model weights being free to download. The hosting cost of a frontier-class open-source model at enterprise scale can comfortably exceed the API cost of a comparable hosted model. The data residency benefit is real; the cost neutrality claim is not.

Voice-based systems stack the cost further. Speech-to-text transcription, LLM inference, and text-to-speech synthesis are three separate billed operations for every utterance. Latency requirements for voice interactions force higher-tier model choices - the cheaper, slower model that works acceptably for asynchronous document analysis creates an unacceptable user

experience when someone is waiting for a spoken response. Multimodal inputs - images, scanned documents, PDFs passed as context - carry their own token economics, where a single image or document page can consume thousands of tokens of context window. The pricing page shows cost per token. The actual cost per business transaction, across a voice or multimodal workflow, is a different number that few teams calculate before building.

RAG systems - now near-universal in enterprise AI deployments - introduce an infrastructure layer that is almost never fully costed at the outset. A RAG deployment requires a vector database, which has its own storage and query pricing that scales with corpus size. Every document in the retrieval corpus must be chunked and embedded - a process that requires either embedding API calls or a locally hosted embedding model, both of which carry cost. A corpus of one million document chunks can cost thousands of dollars to embed initially, then ongoing re-embedding cost as documents are updated or added. The retrieval pipeline itself requires engineering to maintain, and retrieval quality requires its own evaluation infrastructure separate from the generation quality evals. And critically: every RAG retrieval stuffs chunks into the model's context window, which means retrieval quality directly drives inference cost. A poorly tuned retrieval system that returns irrelevant chunks wastes tokens on every call. A proper functioning RAG requires extensive research, and engineering expertise which many organization are not taking seriously.

Questions the Infrastructure Cost Raises

When the API endpoint is the starting point, not the ending point - who owns the build cost of everything around it? When voice, RAG, multimodal, and agentic pipelines each multiply the per-transaction cost - has anyone modelled what a single business workflow actually costs end-to-end? When the vendor's pricing page shows cost per token and the real unit is cost per outcome - what is the gap, and who is measuring it?

7.9 Organisational Proliferation: When Every Team Wants Their Own Stack

The infrastructure cost of a single AI deployment is a solvable problem. The infrastructure cost of AI proliferating across an entire organisation - which is the direction every enterprise that has successfully deployed AI in one function is heading - is a different order of problem entirely.

AI does not stay contained. Once one team has a working RAG system over their documents, every adjacent team wants one. Once the customer-facing product has a voice interface, the internal tools roadmap fills with voice requests. Once the engineering team has a coding assistant, the finance team wants a financial analysis assistant, the legal team wants a contract

review assistant, the HR team wants an onboarding assistant. Each team's individual business case is defensible in isolation. The aggregate infrastructure cost is invisible until it isn't.

Each independent deployment brings its own vector database instance, its own embedding pipeline, its own prompt management system, its own monitoring stack, its own token budget with no visibility into what other teams are spending. The duplication is not just financial - it is operational. The same model upgrade that requires one team to re-validate their eval suite requires every team to re-validate theirs, in parallel, with no shared infrastructure to reduce the effort. The same security vulnerability in a prompt injection attack surface affects every deployment independently, requiring independent remediation across every team.

There is also a subtler proliferation cost that does not appear on any infrastructure bill: the engineering time diverted from product work into AI infrastructure maintenance. Every team that owns its own AI stack owns the on-call rotation for it. Every outage, every unexpected cost spike, every model behaviour change that breaks a workflow - these become the operational burden of teams whose primary job is something else. At scale, this is not a small number of engineering hours.

The token budget question compounds this. In a proliferated environment, there is often no single owner of AI spend visibility. Individual teams track their own usage against their own estimates. The aggregate organisational spend on AI inference, embedding, vector storage, voice APIs, and surrounding infrastructure is frequently unknown to the finance function until a cloud bill arrives that is materially larger than anyone anticipated. This is not a hypothetical scenario - it is the pattern that has already played out in cloud compute spending, in SaaS subscription sprawl, and it is beginning to repeat in AI infrastructure with the additional complexity of usage-based pricing that scales non-linearly with workflow sophistication.

Questions Organisational Proliferation Raises

When ten teams each build their own RAG system - is that ten times the value, or ten times the cost with a fraction of the coordination? When every function wants token allocation - who owns the organisational AI budget, and who has visibility across all of it? When AI infrastructure maintenance diverts engineering hours from product work - has that opportunity cost been counted? When the cloud bill arrives - will anyone have predicted it?

8. Conclusion: Deliberate Beats Fast

The enterprises that will emerge from the current AI wave in the strongest position are not the ones that moved fastest. They are the ones that asked the questions that the hype was designed to discourage: How do we verify this? Who is accountable when it fails? What data are we exposing and to whom? What is our actual error tolerance for this use case? And - the question this paper has added - what does doing this properly actually cost, and have we provisioned for it?

These are not anti-AI questions. They are the questions that any serious technology governance framework demands - of ERP systems, of cloud migrations, of any high-capability technology with meaningful failure consequences. AI deserves no special exemption from that rigor. The fact that labs are unusually skilled at manufacturing urgency is a reason for more discipline, not less.

The risks documented in this paper are genuine but manageable. Data exposure, accuracy failure, agentic overreach, shadow usage, accountability gaps, deprecation cycles, and the true economics of governance all have mitigations. None of those mitigations are available to an organization that adopted under pressure without asking the hard questions first.

The question has never been whether to use AI. It has always been whether you understand what you are using, what it will do when it fails, who is accountable when it does, and whether the total cost of doing it right - not just the token bill - has been honestly calculated.

By
SK Joshi