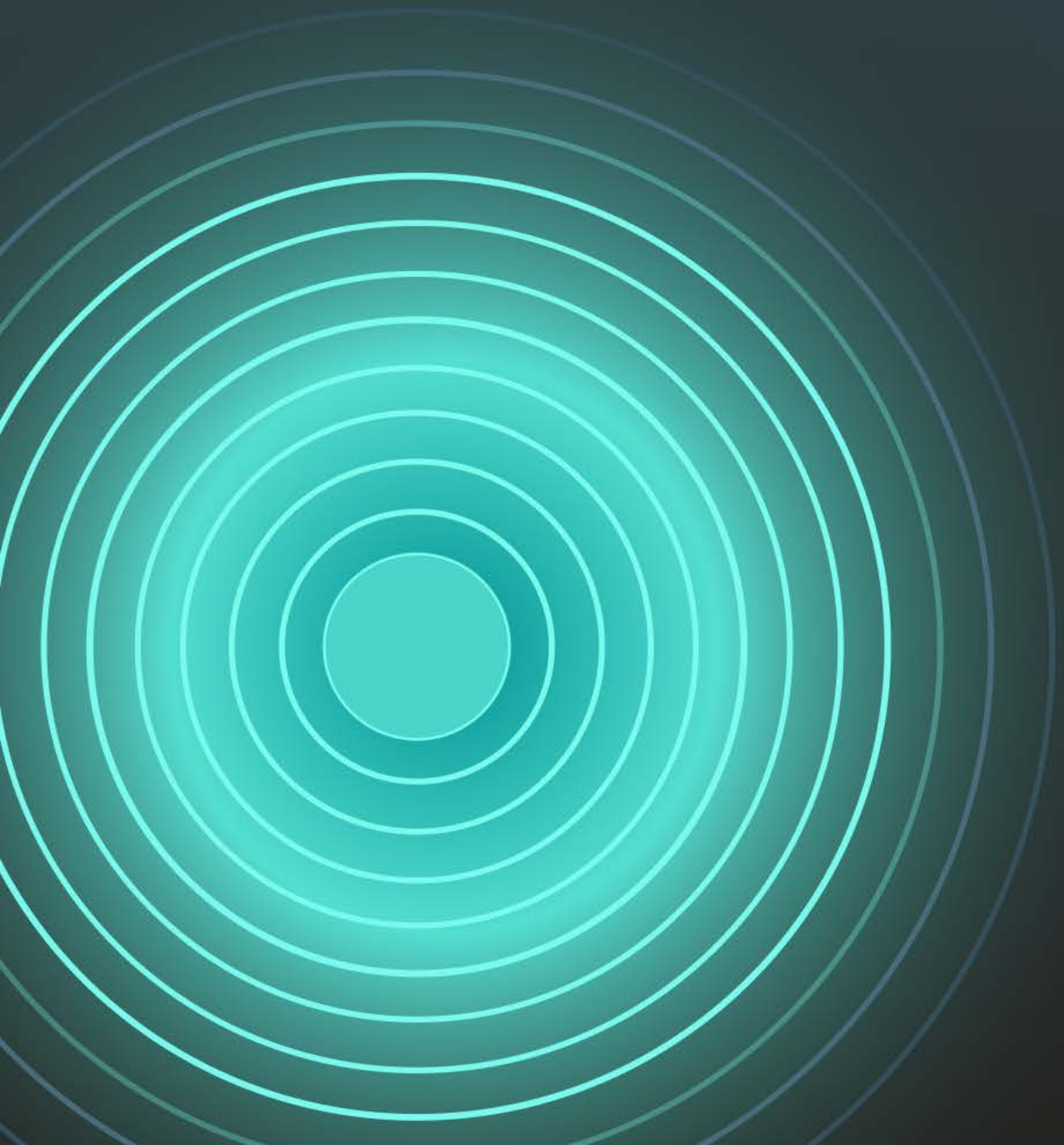
Modern WhitePaper

# How Data Determines the Cost of AI

## A guide to reducing token costs through data



# Table of Contents

# Executive Summary



AI models keep getting cheaper while enterprise AI keeps getting more expensive. It sounds contradictory, but the reason is simple. Ignore the fixed costs for a moment, such as infrastructure, integration, talent, and data. The cost of completing an AI task comes down to this:

Cost per task = Price per token × Number of tokens used

**Token prices have fallen roughly a thousandfold in three years, but the number of tokens required to complete a task has exploded.** The second variable is rising faster than the first is falling.

The reason has little to do with the model and everything to do with the data underneath it. Much of an agent's work goes to rediscovering things humans already know, such as what a column means, which numbers can be trusted, and who is allowed to see them, because nothing about the data's meaning travels with the data itself.

As Modern's CEO, Saurabh Gupta noted in "Notes from a Data Leader: Why AI Costs More Than It Should", data without that meaning, in effect, ships like a warehouse of unlabeled boxes: every worker must open, inspect, and interpret each box before any real work begins, every single time.

While stronger models and cheaper tokens can help, the highest-leverage investment is a data foundation that gives AI the context it needs without forcing it to rediscover that context every time. That foundation can include governed data products that carry their own definitions, quality signals, lineage, and access rules, built once and reused on every call. This approach has significantly cut token consumption compared to working from undocumented schemas. Over time, those unnecessary tokens are the difference between an AI program that scales economically and one that quietly becomes too expensive.

*This paper examines where AI work actually goes and why enterprise data architecture creates so much of it. It then explores an architectural shift that changes the cost structure of every model built on top of that data: moving work from inference time into data design. Modern's DataOS is the implementation of that shift: an AI-native data operating system built to carry this context so agents inherit it instead of reconstructing it.*

# Why AI Costs Keep Rising



Token prices have fallen sharply. The cost per token has dropped roughly a thousandfold since 2023, and a frontier model in 2026 costs a fraction of what the same tier cost then.

Despite this, AI spending is rising. Gartner forecasts worldwide generative AI spending will grow 76.4% each year, even as the unit price of running these models decreases. More capable models do not resolve this discrepancy, because model quality was never the primary cost driver. **The determining factor is the amount of work an AI system must perform to produce a single answer.**

This marks a major change from the chatbot usage era. Research shows that agentic tasks consume roughly 1,000 times more tokens than a standard chat interaction. A typical 2023-era exchange consisted of a prompt and a response, consuming only a few thousand tokens. A 2026-era agentic workflow might decompose a task, invoke tools, validate its outputs, and coordinate with other agents which consumes significantly more tokens regardless of how inexpensive each token might cost.

This dynamic isn't new. In 1865, an economist named William Stanley Jevons observed that more efficient coal engines didn't reduce coal consumption, but they increased it. This made coal economical for uses that had previously been too costly to justify. Token usage is following the same curve: inference costs have fallen one thousand-fold while demands rise ten thousand-fold, and enterprise AI spend rose an estimated 320% in 2025. ***By 2030, we are expecting to see a 24-fold increase in token consumption.***

# Where AI Spends Its Time (and Your Money)



AI agents don’t just answer questions anymore; they plan, call tools, split work with other agents, validate outputs, and retry failures.

When AI costs are organized around token price, that framing clouds the more useful question of what the AI is actually doing and how much of that work is necessary.

There are many steps an agent must take before it produces a useful answer: it must find the relevant data, interpret an unfamiliar schema, reconstruct business context that the data does not carry with it, validate whether the data can be trusted, coordinate with other agents performing parallel work, retry tasks that fail, select the appropriate tool for a given step, and reason over information it may not ultimately need. Each step consumes compute and adds to the total cost, yet none of it is the business value itself. It is the overhead required to get there.

The magnitude of this overhead is substantial. Anthropic’s engineering team reported that multi-agent systems use approximately fifteen times as many tokens as simple chat interaction, even before an agent produces any output of business value.

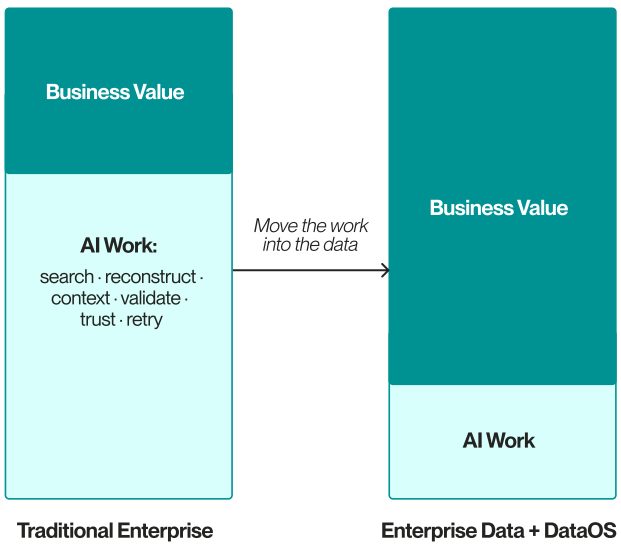
The extra cost comes from repeatedly passing the same context between agents as they work together. For example, a task that takes one agent about 10,000 tokens could take three agents around 29,000 tokens. In a larger setup with five agents and 50 steps, this repeated sharing of context can create more than two million extra tokens.

Retries create another extra cost on top of coordination. An agent that gets the right result 80% of the time may look great in a demo, but real-world systems often require closer to 95% reliability. Reaching that level usually means running more checks, retries, and extra steps. So more tokens have to be spent to make up for the uncertainty that comes with AI systems, unlike traditional systems that follow fixed and predictable rules.

DataOS is a platform that gives AI agents trusted, ready-to-use data context, so they spend tokens on results instead of re-verifying information. That changes where this work happens. Instead of each agent searching for data, rebuilding context, and checking whether the data can be trusted, DataOS handles that work once at the data layer. Agents start with trusted, ready-to-use context instead of recreating it every time. As the diagram shows, this means fewer tokens are spent finding and validating information, and more are spent producing the business outcome the AI was asked for.

## Where AI Spends Its Effort

Same task. Same model. Different data architecture




With DataOS, the AI work doesn't disappear.
It moves to where the data already lives.

# Why Enterprise Data Creates So Much AI Work



**Enterprise data architecture was not designed for AI.** It was built to store, process, and report data, with the assumption that people would fill in the gaps. Humans can rely on asking what a column means, or clarifying ambiguous information with a colleague. AI cannot do that automatically. When the business context is missing from the data, each interaction has to reconstruct that context from scratch.

**Each reconstruction carries a recurring cost.** Often, multiple systems define the same metric differently, with no single source of truth. Individual agents must re-derive the meaning of a given field rather than inheriting a definition that already exists. Access and trust need to be reverified as well. The same business logic is rebuilt from raw tables repeatedly throughout the day, by agents that do not share their work with one another.

This is a longstanding problem that AI has made urgent. Estimates of the proportion of time data professionals spend finding, cleaning, and reconciling data before analyzing it range from approximately 45% to as high as 80% of total working time. Separately, Gartner estimates that poor data quality costs the average organization $12.9 million annually. AI systems have to pay this tax on every call, at machine speed, without the human tolerance for resulting ambiguity.

**Another cost that is easy to overlook comes from the agent's tools.** Every tool has to be explained in the system prompt so the agent knows when and how to use it. As the number of tools grows, the prompt becomes longer, and the context window overflows, resulting in poorer agent output. In some production systems, this means spending tens of thousands of tokens before the user request is even processed. This is yet another reason why giving agents access to a smaller number of well-governed data products can be more efficient than using multiple raw tables.

# The Shift: Move Work from Inference Time to Data Design



A carefully engineered prompt or a larger context window is not a magic solution. To prevent AI from reconstructing context with every call, the context should be built once and used for all future calls. There are five elements that will allow an agent to query directly, rather than rely on documentation it must interpret on its own:

**1) Data products**

**2) Semantic context**

**3) Governance**

**4) Lineage**

**5) Quality as infrastructure**

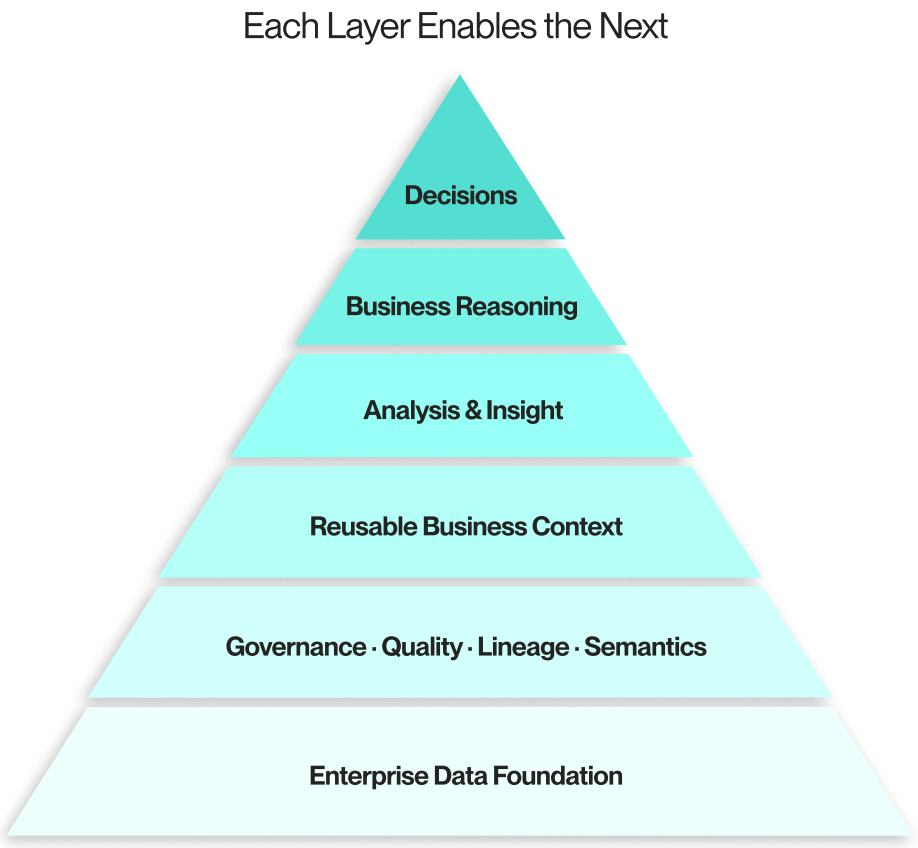
This concept is not the same as "AI-ready data," which uses clean data for a model to consume. The difference is whether the AI interprets the data once at the data layer, or repeats that work with every request.

Early evidence supports this approach. Enterprise deployments using pre-defined semantic context layers have reduced input token consumption by as much as 91%. The benefit is not just lower cost, but also better context quality. Long-context models can become less accurate when the information they need is buried inside a large amount of irrelevant material. In other words, giving a model more context is not always better. What matters is giving it the right context.

# The Enterprise Data Value Pyramid

A pyramid framework is coherent only if each layer enables the layer above it. For AI, that means searching, rebuilding context, and retrying are extra work it must do when the supporting layers are missing.

Reading the diagram below from the bottom up, each layer supports the one above it. AI cannot reason without analysis, it cannot analyze without trusted business context, and it cannot build trusted context without a strong data foundation.




Reading from the bottom up, the pyramid builds each layer once, so AI inherits it instead of rebuilding it.

As AI agents begin to build their own memory, a new kind of cost appears: memory debt. **The more information an agent stores, the harder it becomes to understand what it remembers, why it remembers it, and whether the information is still correct.** More memory does not always make an agent smarter. In Jorge Luis Borges's short story "Funes the Memorious," a young man is cursed with perfect, total recall: he can describe every leaf on every tree he's ever seen, but he can't grasp general ideas or think in abstractions, because his mind is buried under an infinite pile of undifferentiated detail. AI agents risk the same trap. Without governance, agent memory can become another data swamp inside the agent itself. The fix is to be intentional about what should be remembered, updated, explained, and forgotten.

Decisions sit at the top because they are the outcome the business ultimately wants. They become faster and less expensive when reasoning, analysis, and context are already supported by clear governance, reliable data, known lineage, and consistent definitions. Without reusable context, the AI must rebuild this understanding for every request.

This is the hidden cost the pyramid shows and it’s the problem DataOS solves. DataOS builds a data foundation once and makes it reusable across every agent: instead of each agent figuring out definitions, lineage, and data quality from scratch, DataOS provides that context upfront. Definitions stay consistent, lineage is already traceable, and quality checks happen at the source. That means less memory debt and less repeated work. Agents can spend less time rebuilding context and more time doing the reasoning the business actually needs.

This pattern matches the distribution of effort described in Section 2. In a traditional enterprise architecture, an AI agent spends most of its time finding information, rebuilding context, checking whether the data can be trusted, and retrying failed steps. Only the remaining effort creates business value. In an architecture designed for AI, the opposite is true. Because context, governance, and data quality are already in place, the agent can spend most of its effort on useful business work. The greatest cost savings come from reducing the need to search, rebuild context, and retry.

# The Four Sources of AI Efficiency



Having a strong data foundation doesn't just prevent memory debt. It's also where AI efficiency gets unlocked. Once governance, quality, and semantics are handled, agents stop re-deriving the same context, re-checking the same data, and over-relying on expensive models to compensate for uncertainty. Four specific sources of waste disappear as a result. None of them require a better model. All of them require better data.

**Reusable Context**

Once business meaning is defined, it should not be reconstructed on every call. The same definition should be available to every agent that needs it, not re-derived one query at a time.

**Trusted Data by Default**

Building trust into the data from the start reduces the need to check it and retry each time. Provenance and quality checks happen once, upstream, instead of being repeated by every agent that touches the data.

**Intelligent Model Routing**

Computationally expensive reasoning should be used for problems that need it, not as a default for every task to a dense, frontier model. In early 2025, roughly 73% of enterprise token volume was being sent to the two most expensive model tiers, even though 85% of enterprise queries can be handled by lower-cost models without a measurable drop in quality. What most don't realize is how enormous the cost difference can be. Some budget models cost around $0.04 per million tokens while frontier reasoning models can exceed $180 per million tokens. Without routing, a simple FAQ can be priced as a legal analysis. Routing alone can reduce AI spend by 60-80% without changing the user experience.

**Governed Data Products**

Giving an agent one clear, well-organized data product instead of several raw tables makes it much easier to find and use the right information.

These four principles share a common characteristic: none describe a property of the model. Each describes a property of the data the model depends on, which is precisely why they remain applicable as the underlying model generation changes.

Another consideration is that more agent steps do not always cause better results. Not every task needs an agent, not every agent needs multi-tasking sub-agents, and not every workflow needs validation passes, retry loops, and a reflection cycle. Each step or handoff adds more context bloat, tokens, and opportunities to fail. As we explain in "Why Cheaper AI Tokens Are Increasing Enterprise AI Costs," an orchestration layer that costs more than the work it orchestrates is an architecture defect, and it should be modeled before deployment, not discovered in the invoice." These costs should be understood before deployment rather than discovered later in the bill.

# Measuring AI Efficiency



Most metrics used today focus on the cost per token and model cost, both of which involve the price of an input. Neither metric considers whether a given task actually needs that amount of input.

More telling measures might include: AI work per task, tokens per completed workflow rather than per individual call, retries per workflow, context retrieved per task, duplicate retrieval rate, the proportion of context that is reusable versus reconstructed, and most significantly the cost per business outcome. Cost per business outcome is one of the strongest figures for executive-level reporting, as it is the only measure that directly indicates what proportion of total spend was used for work the business did not request.

There is also an important difference between measuring cost after it happens and governing it before it happens. Cost per model tells you what was spent, while cost per workflow helps explain where that money went and if it was worth it. Many organizations realize too late that their expensive frontier models were not needed. A better approach is to place a governance step before the spend, requiring a clear reason for routing a task to a more expensive model when a lower-cost option could do the job.

Measurement also needs to be continuous. Microsoft has reported as much as a 30× difference in resource use across repeated runs of the same task, showing how much AI costs can vary even when the work itself does not change. A periodic audit can easily miss that discrepancy. Tracking efficiency at the workflow level over time makes it easier to identify waste, understand why it is happening, and correct it before it becomes a recurring cost.

# Building a Cost-Efficient Data Architecture for AI



Instead of replacing an existing data stack, implementation requires identifying the reality of where AI work is currently occurring versus what documented architecture might outline.

Once identified, the following steps apply:

**1. Identify the context most frequently reconstructed and convert it into a governed data product, built once and reused across every call.**

**2. Embed governance and quality checks upstream, such that trust becomes a property of the data rather than a task repeated by the agent.**

**3. Measure AI work on a continuous basis, rather than through periodic audit.**

**4. Treat inference cost as an outcome of architectural decisions, rather than as a fixed characteristic of whichever model is currently deployed.**

Inference cost is an architectural outcome, not a model characteristic. DataOS operationalizes each of these steps: it converts the context agents reconstruct most often into governed data products, embeds governance and quality checks upstream of any agent call, and measures AI work continuously rather than through periodic audit.

That same architectural discipline should extend to token consumption. Just as cloud architects learned to treat latency and data transfer as first-class design variables, AI teams should budget and attribute token usage at the workflow level. That requires visibility into individual calls, clear policies for model selection, and accountability for cost across the system. AI efficiency then becomes something designed into the architecture, rather than discovered later when the invoice arrives.

# Conclusion



**The Economics of AI Are Really the Economics of Data**

**The most efficient AI programs will be built on data that requires the least amount of work for AI to understand and use.** Data products, semantic context, governance, lineage, and quality aren't supporting capabilities. They're the architectural foundation that determines the cost of AI.

This means AI economics are not determined by model pricing alone. Price per token is only one part of the equation. The other is how many tokens a task requires. When business meaning, relationships, and quality signals are missing from the data, agents have to reconstruct that context themselves which drives up both token usage and cost.

DataOS is Modern's answer to that constraint: an AI-native data operating system that builds the data products, semantic context, governance, lineage, and quality infrastructure once, so every model and agent built on top of it inherits them instead of rebuilding them.

**Want to See What This Looks Like For Your Data?**
If you'd like to learn more about DataOS or talk through how this applies to your organization, please get in touch.

# Bibliography


Anthropic. (2025). How we built our multi-agent research system. https://www.anthropic.com/engineering/multi-agent-research-system

Gartner, Inc. (2025, March 31). Gartner forecasts worldwide GenAI spending to reach $644 billion in 2025. https://www.gartner.com/en/newsroom/press-releases/2025-03-31-gartner-forecasts-worldwide-genai-spending-to-reach-644-billion-in-2025

Gartner, Inc. (n.d.). Data quality: Why it matters and how to achieve it. https://www.gartner.com/en/data-analytics/topics/data-quality

Goldman Sachs. (2026, May). AI agents forecast to boost tech cash flow as usage soars. https://www.goldmansachs.com/insights/articles/ai-agents-forecast-to-boost-tech-cash-flow-as-usage-soars

Jevons, W. S. (1865). The coal question: An inquiry concerning the progress of the nation, and the probable exhaustion of our coal mines. Yale University, Economic History resources. https://energyhistory.yale.edu/w-stanley-jevons-the-coal-question-1865/

Microsoft Research. (2026). How do AI agents spend your money? Analyzing and predicting token consumption in agentic coding tasks. https://www.microsoft.com/en-us/research/publication/how-do-ai-agents-spend-your-money-analyzing-and-predicting-token-consumption-in-agentic-coding-tasks/ (see also preprint: arXiv:2604.22750, https://arxiv.org/abs/2604.22750)

# Bibliography


Mohan, B. (2026). The token paradox: Why cheaper compute produces bigger bills. Modern Data 101. https://moderndata101.substack.com/p/the-token-paradox-of-cheaper-compute

MLOps Community. (n.d.). How I reduced AI token costs by 91% with semantic tool selection and Redis. https://mlops.community/blog/how-i-reduced-ai-token-costs-by-91percent-with-redis

Swartz, A. (2020, July 6). Data prep still dominates data scientists' time, survey finds. BigDATAwire. https://www.hpcwire.com/bigdatawire/2020/07/06/data-prep-still-dominates-data-scientists-time-survey-finds/

The Modern Data Company. (2026). DataOS: Empowering business-ready data products for faster insights. https://www.themoderndatacompany.com/dataos

The Modern Data Company. (2026, July 8). Why cheaper AI tokens are increasing enterprise AI costs. https://www.themoderndatacompany.com/blog/why-cheaper-ai-tokens-are-increasing-enterprise-ai-costs

Token coherence: Adapting MESI cache protocols to minimize synchronization overhead in multi-agent LLM systems. (2026). arXiv. https://arxiv.org/pdf/2603.15183

# Contact Us


Authors

**Brij Mohan Singh,** *The Modern Data Company,* **brijmohan.singh@tmdc.io**
**Natasha Akali,** *The Modern Data Company,* **natasha.akali@tmdc.io**


To schedule a demo or learn more about DataOS: **info@tmdc.io**

**www.themoderndatacompany.com**

Modern™