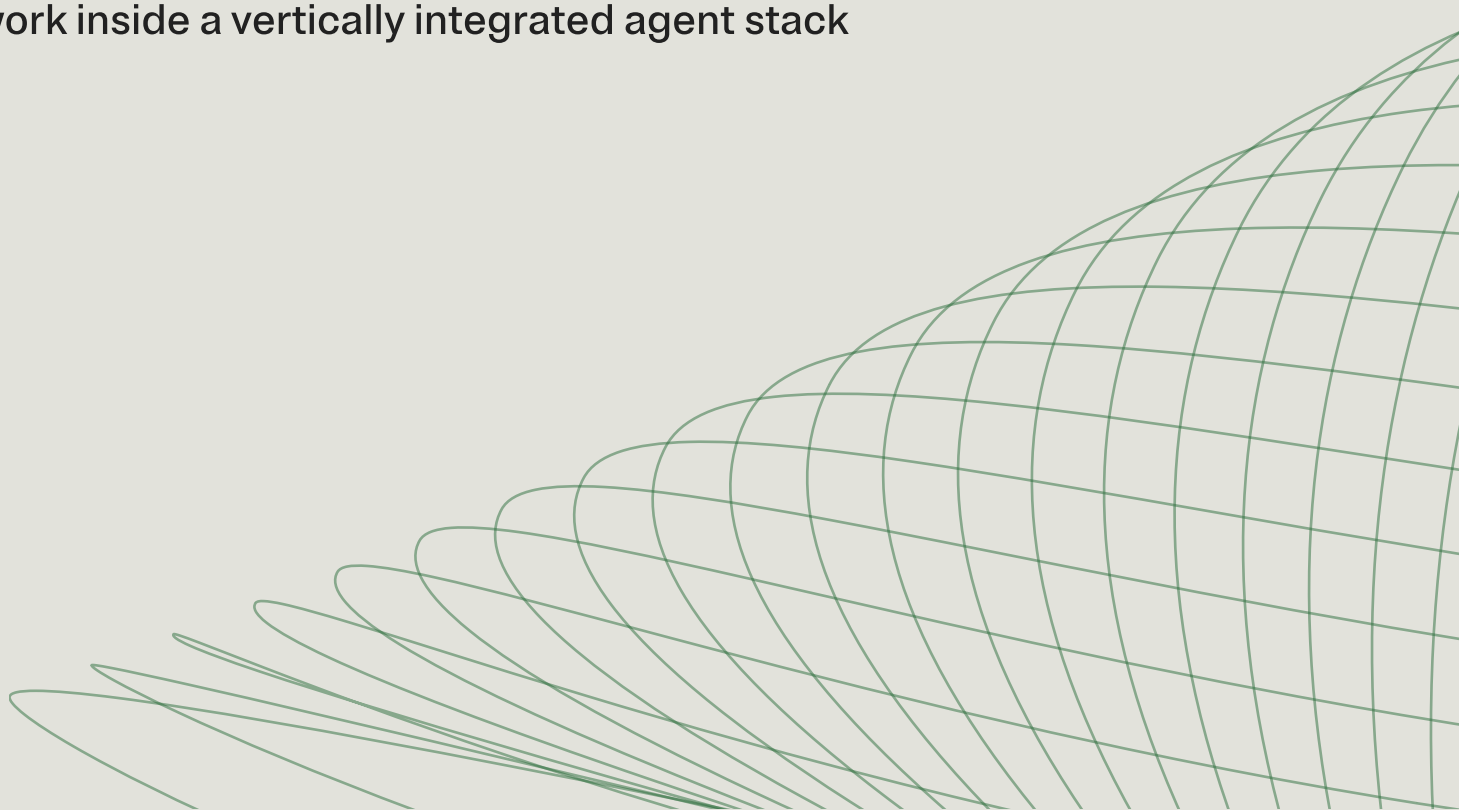
STACKLOK

Contain Your AI Costs: From Tokenmaxxing to Tokenomics

The five pillars of sustainable AI economics, and why none of them work inside a vertically integrated agent stack



Contents

For the first phase of enterprise AI, consumption was the metric. Boards asked how many tokens the organization was pushing and vendors published usage curves as proof of momentum. The implicit strategy was tokenmaxxing: consume more intelligence, faster. That phase made sense when the goal was learning what the technology could do.

It stops making sense when the CFO opens the invoice. The successor discipline is tokenomics: treating tokens as a governed unit of production cost that is attributed and negotiated like any other input the business depends on.

Note: the word has an older life in crypto; here it means the economics of AI inference, nothing more.

This paper does two things. First, it lays out the five pillars of an enterprise tokenomics practice, each grounded in current data. Then it makes an important argument: that these pillars cannot be built inside the closed agent stacks of frontier labs and hyperscalers, and that an open agent platform is a prerequisite for real AI cost control.

The paradox that makes tokenomics necessary

The strangest fact in enterprise AI economics is that the product keeps getting cheaper while the bill keeps getting bigger. Deloitte analysis cited in Forbes found per-token inference costs have fallen roughly 280-fold over two years. Over the same period, the FinOps Foundation's State of FinOps 2026 report, covering 1,192 organizations and 83 billion dollars in cloud spend, found AI workloads climbing to 18 percent of cloud spend at AI-forward enterprises, up from 4 percent in 2023.

Agents are the multiplier. A single user action now triggers planning, retrieval, tool calls, verification, formatting, and retry loops. Each output consumes input tokens and output tokens. Background inference compounds it; you're now burning tokens whether or not you asked for anything. Gartner analysts caution that projected unit-cost declines, as much as 90 percent by 2030, will not reach enterprise invoices intact, because agentic demand scales faster than prices fall.

Meanwhile the return side is under real scrutiny. PwC's 29th Global CEO Survey of 4,454 chief executives found 56 percent report AI has produced neither increased revenue nor decreased costs. Ramprakash Ramamoorthy, Director of AI Research at Zoho, points to the structural reason finance teams are struggling: "inference behaves unlike every cost line that preceded it." Falling unit prices will not save the budget. Discipline will.

Part One: The Five Pillars of Enterprise Tokenomics

Pillar 1: Meter and attribute every token

You cannot manage what you cannot decompose. The foundational capability is attribution: every token must be traced to a named identity, a workflow, and a business outcome, so the AI bill breaks down to an agent, a team, and a dollar amount.

Most enterprises are nowhere near this. Instead, they have to make sense of a monthly invoice from one or two providers, decomposable only to the API-key level, which tells you which application spent the money but not which workflow, which agent, or which decision. Boston Consulting Group's 2026 guidance on AI token costs draws the accounting conclusion: tokens consumed inside a product or customer interaction are 'cost of goods sold' and belong in gross-margin management, which is impossible when consumption cannot be attributed to the product feature that incurred it.

Attribution is also the prerequisite for every other pillar. Routing decisions, outcome measurement, and consumption engineering all depend on knowing who spent what, on which task, with which model. The FinOps Foundation reports that AI cost management is now the single most sought-after skill among its practitioners, with 58 percent prioritizing it for the next twelve months. The skill starts with metering.

Pillar 2: Measure cost per outcome, not cost per token

Per-token price is the wrong unit of analysis, and optimizing it can produce perverse results. A cheap model that fails a task and triggers three retries, a human review, and an escalation costs more than an expensive model that succeeds in one attempt. The unit that matters is cost per successful task.

This requires two things most organizations lack. The first is a definition of success per workflow: what a resolved ticket, a correct extraction, or an approved draft actually looks like. The second is an evaluation practice that measures models against that definition continuously, because model behavior drifts with every version update.

Evals, in other words, are the pricing mechanism of a tokenomics practice. They define the bar a cheaper model must clear before it earns the traffic.

Pillar 3: Route work to the cheapest model that clears the bar

Once outcomes are measurable, routing becomes the largest single lever. Reserve frontier models for high-stakes reasoning, and route volume work such as summarization, classification, extraction, and formatting to small or open-weight models. Many enterprises are beginning to keep a fallback warm for outages and deprecations.

The gains are documented and large. One engineering team that audited token usage and routed simpler subtasks to cheaper models cut monthly API costs from 40,000 to 24,000 dollars with no product changes. The practice is spreading: in a16z's 2026 enterprise survey, 81 percent of respondents reported orchestrating three or more model families in production, up from 68 percent a year earlier, and Menlo Ventures found 37 percent of enterprises running five or more models. A model portfolio is the mechanism by which price competition among model providers actually reaches your invoice.

Pillar 4: Engineer the consumption itself

The least-discussed pillar is the demand side: most enterprise token spend is not the model thinking, it is the system feeding the model things it does not need.

Tool definitions are the clearest example. Agents connected to multiple tool servers commonly load every tool schema into the context window at the start of every conversation. Stacklok's analysis found GitHub's official MCP server alone consumes 17,600 tokens of tool definitions per request. If you connect a handful of servers and an agent, you can burn tens of thousands of tokens of metadata before you blink. The fix is on-demand tool discovery: load a lightweight search interface, fetch individual tool schemas only when needed.

Caching is the second lever, but it only works on workloads structured for it: stable content at the front of the prompt, volatile content at the tail, and deliberately placed cache boundaries. Add retry budgets, context-window hygiene, and the decommissioning of background inference nobody asked for, and consumption engineering routinely rivals routing as a source of savings.

Note what all these levers have in common: none of them is a model setting. They are properties of the harness, the runtime that assembles context, loads tools, and structures prompts. A recent arXiv study titled "The Harness Effect" makes the point empirically: orchestration design sets the token economics of enterprise agentic AI, with prompt structure alone determining whether 99 percent or none of a session's input tokens are served from cache. The bill is written where the harness is designed.

Pillar 5: Maintain portfolio leverage

The final pillar is commercial rather than technical. Model prices, in the end, are negotiated, and negotiating position is a function of credible exit. A vendor facing a customer who can reroute traffic next week prices differently than a vendor facing a customer with two years of unexportable workflow state.

Portfolio leverage requires three assets: (i) your own evaluation suites (so any candidate model can be judged against your bar rather than the vendor's benchmarks), (ii) at least one production-proven alternative per workload class, and (iii) switching costs low enough that the alternative is believable.

Part Two: Why Tokenomics Requires an Open Agent Platform

Read the five pillars again, and notice what each one quietly assumes: that the enterprise controls the layer where the pillar is implemented. Metering assumes you can see every call. Outcome measurement assumes your evals travel with you. Routing assumes the set of reachable models is yours to define. Consumption engineering assumes you can modify the harness. Leverage assumes you can leave.

Inside the closed agent stacks of frontier labs and hyperscalers, every one of those assumptions fails. Let’s walk the pillars a second time.

Attribution fails because the meter belongs to the vendor. A closed stack reports the usage it chooses to report, at the granularity it chooses, in a console it operates. Decomposing spend to a named identity and workflow requires an independent enforcement point that every model call and tool call passes through, and a closed stack is precisely the architecture that prevents one. You cannot audit a bill whose measurement instrument is owned by the party that sends it.

Outcome measurement fails because the evals are captive. When evaluation suites, traces, and historical results live in one vendor's console, in one vendor's format, they can judge that vendor's models and nothing else. A market where every vendor grades its own homework in its own gradebook is not a market, and an eval suite that cannot score a competitor's model cannot set a bar for it. Pillar two collapses into a dashboard.

Routing fails by construction. A closed stack routes among the models its owner offers, on the terms its owner sets. The 280-fold collapse in inference prices was driven by competition across providers; a single-stack enterprise is structurally excluded from most of it. Whatever savings the vendor captures from cheaper inference reach the customer only at the vendor's discretion, which is exactly what Gartner analysts predict: efficiency gains that do not arrive on the invoice intact.

Consumption engineering fails because the harness is sealed. If orchestration design sets token economics, then whoever designs the orchestration sets your costs. In a closed stack, prompt structure, cache boundaries, tool-loading behavior, and retry policy are the vendor's engineering decisions, made for the vendor's margin across all customers, not for your interests. An enterprise that cannot restructure its own prompts cannot implement pillar four at all.

Leverage fails last and worst. Every quarter inside a closed stack deepens the well: agent definitions in a proprietary format, memory that exports only as transcripts, policies with one implementation. Exit stops being credible, and pricing follows. In a 2026 AvePoint analysis, 47 percent of enterprise executives said losing their primary AI vendor would disrupt a key business function. That number is the negotiating table, seen from the wrong side.

The conclusion is uncomfortable but hard to escape: a tokenomics program run inside a walled garden is a reporting exercise. You cannot run tokenomics from inside the store that issues the bill.

What the open alternative looks like

An open agent platform inverts each failure. The components are auditable, so the meter can be trusted because it can be read. Every model call and tool call passes through gateways the enterprise operates, so attribution to a named identity is a property of the architecture rather than a vendor report. Evals live in portable formats, so any model, frontier or open-weight, can be scored against the same bar. Routing policy is the enterprise's to write, spanning every provider willing to compete. The harness is modifiable, so cache shape, tool discovery, and context discipline are engineering decisions the enterprise makes in its own interest. And because agent definitions, memory, and policies are stored in forms more than one runtime can execute, exit stays credible and every negotiation stays honest.

None of this requires abandoning frontier models. Enterprises just need to insist on owning the layer that measures, routes, and shapes their consumption.

Where to start

The sequencing matters, because instrumentation without control produces reports, and control without instrumentation produces guesses.

1

Instrument first

Stand up metering at an enforcement point you operate. Attribute spend to identities and workflows for thirty days before optimizing anything.

2

Define success per workflow

Write the evals. A cheaper model cannot earn traffic until there is a bar for it to clear.

3

Tier the obvious workloads

Summarization, classification, extraction, and formatting rarely need frontier reasoning. Route them and measure the quality delta.

4

Attack consumption

Audit tool-schema overhead, restructure prompts for caching, set retry budgets, and kill unrequested background inference.

5

Make exit credible

Require agent definitions, memory, evals, and policies in portable formats as a procurement condition, and test the export before renewal, while you still have leverage.

The tokenmaxxing era treated consumption as proof of progress. The tokenomics era treats it as a cost to be governed, and governance requires owning the layer where cost is driven and value is created. Put the meter, the router, and the harness on your side of the wall.

Sources

- FinOps Foundation, State of FinOps 2026 (as reported by CIO.com, April 2026)
- Deloitte inference cost analysis (as cited in Forbes Technology Council, July 2026)
- VentureBeat reporting on inference unit prices and consumption growth, April 2026
- Gartner, inference cost and token demand analysis, 2026 (as reported by CIO Dive and CloudCostChefs, March 2026)
- PwC, 29th Global CEO Survey (as reported by CIO.com, April 2026)
- Ramprakash Ramamoorthy, Zoho, in The Source Code, July 2026
- Boston Consulting Group, "How Enterprises Can Control AI Token Costs," July 2026
- a16z, "AI Adoption by the Numbers," Kimberly Tan, 2026
- Menlo Ventures, "2025: The State of Generative AI in the Enterprise," December 2025
- Optimum Partners, Q1 2026 analysis of 2.4 billion enterprise API calls, May 2026
- The Source Code, "Token Costs Are Breaking Enterprise AI Business Cases," July 2026 (routing case study)
- StackOne, "MCP Token Optimization: 4 Approaches Compared," March 2026
- Scalekit benchmark of tool-interface token overhead (as reported by Apideck and Roadie, March to June 2026)
- AgentWorks, "Token Caching Strategies That Actually Work in Production," May 2026
- arXiv, "The Harness Effect: How Orchestration Design Sets the Token Economics of Enterprise Agentic AI," 2026
- AvePoint, "Avoid AI Vendor Lock-In: A Multi-Model AI Strategy," 2026