THE TOKEN ECONOMY
VOL. 1 · NO. 1 · ENTERPRISE AI ECONOMICS

The Token Economy

A First-Principles Playbook for Governing Claude in Production

AUDIENCE

CFOs, CTOs, Heads of AI Platform, and the Staff+ Engineers Who Implement Their Decisions

FRAME

Industry view · vendor neutral · operational

PUBLISHED

May 2026 · 35 min read

C O N T E N T S

FROM THE AUTHOR

What follows is a composite incident.

The specifics are invented. The failure mode is not. This paper is built around a scene that has played out, in some form, at enough enterprises in 2025 and 2026 that the post-mortems are converging on a single root cause: token strategy was treated as a deployment-time concern rather than a planning-time discipline. The malformed JSON was the trigger. The vacuum was the disease.

Production inference cost is governed by token discipline, not price-per-token. The firms that compound margin in the AI era will not be those that win the procurement negotiation; they will be those that institutionalize what enters the context window, what those tokens earn while resident, and when they are evicted. This paper specifies that discipline — four levers, one planning artifact, one role, one cadence — in the vocabulary the financial and executive functions of the firm already use.

Where this paper references Claude specifically, it does so because the analytical work is sharpest against a named system. The frameworks apply across model families. The numbers, the break-evens, and the operational recommendations are framed for Anthropic's Claude tier structure (Haiku, Sonnet, Opus) and feature set (prompt caching, extended thinking, the Batch API, tool use, MCP, Skills), and translate, with the appropriate substitutions, to deployments built on AWS Bedrock, Google Vertex AI, or any analogous frontier-model stack.

— May 2026

Opening scene

At 11:47 PM on a Thursday in March, a Series C fintech's on-call engineer acknowledged a billing alert from Anthropic. It was an email, not a page. That was the first mistake - and it had been made eleven months earlier.

The threshold that triggered the alert had been set at three times the prior month's spend, and it had never fired before. It fired now because the company had spent its entire prior-month inference budget in the preceding four hours.

The cause was a sub-agent. An agentic orchestration framework had been wired eleven days earlier to decompose customer onboarding questions into research sub-tasks, each of which could spawn workers, each of which could spawn workers. Recursion depth capped at five. Fan-out per level capped at nothing. A malformed tool response - a JSON parse error the supervisor agent interpreted as a clarification request - triggered the cascade at 7:42 PM. At depth five the system was running 243 concurrent agents. Each one streamed frontier-tier reasoning tokens against a 14,000-token system prompt that was not cached, because the prompt carried a per-request timestamp that invalidated the cache on every call.

The bill for that night was $312,000. The monthly run rate, reconstructed afterward, had crossed eleven times baseline nine days before the alert fired. No dashboard surfaced it. No engineer was looking. The post-mortem named the malformed tool response as the root cause and recommended better input validation.

That was not the root cause. The root cause was that a four-million-dollar-a-year inference line item had no owner, no envelope per workflow, no eviction policy, no kill switch, and — most expensively - no theory of what the workflow was supposed to be earning in the first place. The malformed JSON was the trigger. The vacuum was the disease, and the disease has four names and one origin: **the model the team chose, the workflow they wrapped around it, the data work they did not do upstream, and the prompt literacy they never built - and beneath all four, a Charter never written, a runaway scenario never projected, a business owner never named.**

This paper is about the four names. It is also about the planning discipline that, applied before a single line of code is written, makes the bill at 11:47 PM impossible to produce.

"Reactive token strategy is the strategy of explaining last quarter's invoice. Proactive token strategy is the strategy of being able to commit to next quarter's."

1. Why Claude burns tokens

Claude is the most capable production model family available today, and that capability is precisely why it burns tokens. Understanding why - mechanically, by design - is the precondition for deciding whether you are buying capability you are using or capability you are not.

1.1 Long context is an invitation, not an instruction

Claude's 200,000-token context window is a capability the field has, on average, badly misused. The window is an upper bound on what can be in scope for a single call. It is not a recommendation for what should be. The modal enterprise integration treats the window as a filing cabinet: stuff in the policy document, the user's history, the retrieved chunks, the tool descriptions, the conversation transcript, and let the model sort it out.

Every token in the context window has a carrying cost - it is read on every generation, contributes to latency, and competes with other tokens for the model's attention. The model does not, in any meaningful sense, know which tokens you intended to be load-bearing. It infers, from position and recency and relevance, what to weight. When the window is full of material that is not earning its place, two things happen simultaneously: the bill goes up, and the output quality goes down. The second effect is the more expensive one and the harder one to see.

The discipline the field needs is to treat the context window as a ***balance sheet*** rather than a filing cabinet. Every token is either an asset, a liability, or an expense. Most production stacks have never made the classification, and their bills reflect it.

1.2 Extended thinking trades tokens for correctness

Claude's extended thinking - the model's ability to allocate internal reasoning tokens before producing a user-visible answer - is one of the most economically misunderstood features in the current generation of frontier models. The reflex among cost-conscious teams is to disable it. This is sometimes correct and often expensive in ways that do not appear on the inference invoice.

The mechanical trade is direct: extended thinking spends tokens you pay for in exchange for output quality you also pay for, somewhere else in the system. If the alternative to ten thousand reasoning tokens is a human reviewer correcting a wrong answer, the reasoning

tokens are nearly always cheaper. If the alternative is the same answer reached without the reasoning budget - because the task did not require it - the reasoning tokens are pure waste.

The decision is not "thinking on or off." The decision is workload-by-workload: does additional reasoning budget improve the output enough to displace cost elsewhere in the value chain? For legal reasoning, complex financial analysis, multi-step debugging, and structured planning, the answer is reliably yes. For classification, extraction, summarization, and most conversational turns, the answer is reliably no.

1.3 Tool use and agentic patterns multiply calls

Tool use is, for the right workloads, the cheapest possible architecture. The alternative is stuffing the tool's output into the system prompt on every call regardless of whether it is needed, or shipping a dumber workflow that depends on the user to provide the missing information. Tool use is targeted retrieval: pay for the data only when the model determines it is needed.

Where it goes wrong is the unbounded case. A model that can call tools without a budget, without a depth limit, and without a circuit breaker is a model that can — given the right adversarial input or the wrong malformed response - call them forever. The Opening Scene is the limit case.

Agentic patterns multiply calls again. Each agent in a multi-agent system is a full Claude invocation, with its own system prompt, its own context, its own reasoning budget. A "team of five agents" is five times the per-call cost of a single agent, before accounting for the orchestrator's own reasoning and the inter-agent message passing. For some workloads this is the correct architecture. For the modal enterprise workload it is decoration, and the decoration is billed.

1.4 When the consumption is the right trade

Five tests separate the workloads where Claude's consumption profile is correct from the workloads where it is overkill. A workload earns frontier-tier consumption when the output quality is itself the product. A workload earns extended thinking when the alternative is human review at a higher fully-loaded rate. A workload earns long context when the relevant information genuinely cannot be retrieved in chunks. A workload earns tool use when the data needed varies by query in ways a static prompt cannot anticipate. A workload earns agentic decomposition when the sub-problems are genuinely parallel and individually substantial.

Workloads that fail all five tests should not be running on frontier Claude. The teams that compound margin in this market are the teams that make the classification explicit and revisit it quarterly. The teams that don't are the teams whose CFOs eventually ask, in a tone the platform lead remembers for years, what exactly are we getting for four million dollars.

2. The Pre-Mortem: planning before the build

The dominant posture toward token economics in current enterprise AI practice is reactive. A workload ships. Its bill arrives. Someone is surprised. The posture this paper recommends is the inverse: token strategy is a planning-time discipline, signed before engineering begins.

2.1 The Token Strategy Charter

The artifact is the Token Strategy Charter. One to three pages. Produced once per AI-enabled workload, before any code commits. Owned by a named business stakeholder. Co-signed by the platform team and the finance partner. For workloads in regulated domains, co-signed by the risk officer. Revised on a quarterly cadence. Surfaced to senior management in a format they read.

The Charter is not an engineering artifact. It is a commitment artifact - the analog of the documents senior management already uses to approve cloud infrastructure budgets, capital projects, and headcount plans. Its purpose is to make the economics, the risk envelope, and the accountability of an AI-enabled workload legible to the financial and executive functions of the firm, in the vocabulary those functions already use, before the workload is built.

2.2 The six elements

Purpose and alpha position

What the workload is for, in two paragraphs a board member could read. The business outcome it produces, the baseline it improves against, and the alpha-half-life classification - structural (24+ months), operational (9–18 months), commodity (under 6 months), or expired. A workload whose alpha position cannot be stated has not yet earned its Charter.

Target TRAM cell

The Token Risk Appetite Matrix cell the workload will operate in - cost variance tolerance crossed with output variance tolerance. The cell determines which models are permitted, what agentic depth is authorized, whether extended thinking is enabled, what review topology is required, and whether the Batch API is in or out of scope. The cell is chosen by the business owner and the risk officer jointly, not by engineering.

The three scenarios

Expected. Steady-state projection. Traffic the workload is built for, token consumption per call as architected, monthly cost at planned volumes. This is the number the finance partner takes back into the budget.

Stretch. The success case. Traffic at two to three times expected. The Charter commits to the operational changes - caching posture, model tier, batch eligibility - the workload will execute when it crosses into stretch territory.

Runaway. The failure case. Traffic or token consumption at ten to twenty times expected. The runaway scenario is not a worst case. It is a named case, projected explicitly, with a specified kill-switch threshold and a specified escalation path.

> **The runaway is the cheapest insurance**
>
> A Series C with the Opening Scene's incident on its books, having projected a runaway threshold at five times expected monthly spend with an automatic circuit breaker at that threshold, would have absorbed a $20,000 incident at 7:42 PM and a post-mortem the next morning. Instead it absorbed $312,000 and a board call. The cost of projecting the runaway is one meeting. The cost of not projecting it is whatever the next bill turns out to be, multiplied by however many ungoverned workloads the firm has in production.

Review topology

Drawn from the Fifth Lever in Section 5: who reviews this workload's outputs, what they review, when they review, and with what authority. The Review Topology is determined by the TRAM cell and recorded in the Charter so that it cannot be silently downgraded by an engineering team optimizing for latency. The risk officer signs this element specifically.

Kill criteria

The specific, measurable conditions under which the workload is paused, rolled back, or retired - economic, quality, and strategic triggers, automated where possible, escalated where automation is unsafe. The Opening Scene's incident had no automated economic trigger. The fix is not a smarter alert. The fix is a Charter that, at the runaway threshold, halts new requests automatically until a human authorizes their resumption.

Transparency cadence

How the Charter's promises are reported back to senior management, in what format, on what schedule. The default cadence is quarterly. The default format is one page per workload: what was planned (the Charter), what was spent and produced (the realized economics), what was

learned, and what changes to the Charter the platform team and business owner jointly recommend.

This is the element most enterprises will be tempted to cut. They should not. Without scheduled transparency to senior management, the Charter decays from a commitment artifact to an internal engineering document within two quarters. The discipline is sustained by the visibility.

2.3 Who signs

Business owner. The named individual whose unit will be billed for the workload's tokens and whose business outcomes the workload is intended to improve. Not a generic product manager. The executive accountable for the workload's P&L contribution. If no such individual can be named, the workload should not be built.

Platform team lead. The engineering accountable party. Signs for architectural feasibility, cost projection defensibility, and kill-criteria enforceability.

Finance partner. The FP&A or business-finance counterpart who maintains the inference cost line in the budget. Confirms the expected and stretch scenarios fit the envelope and the runaway threshold is consistent with risk policy.

Risk officer. Signs when the workload is in a regulated domain or its TRAM cell crosses a threshold the firm's risk policy defines. Functionally mandatory for financial services workloads of any consequence, and mandatory for healthcare.

> **Bottom line**
>
> A firm that approves an AI workload without a Charter is approving a workload without a budget, an owner, or a kill switch - and will, with mathematical certainty, eventually pay the bill for that absence. The Charter specifies the envelope. The four levers that follow are the decisions that fit the workload inside the envelope.

3. Lever One - Model selection

The first concrete decision the Charter produces is which model runs the workload. The TRAM cell constrains the choice. The alpha position narrows it. The projected envelope across the three scenarios determines whether the chosen tier is sustainable at expected volumes and survivable at runaway volumes.

Model selection is not the largest source of waste - workflow architecture is, by a factor of three to five. But every downstream lever's economics are conditioned on the model. A correctly disciplined workflow built around the wrong tier produces a bill no amount of upstream filtering can rescue. A perfectly tuned prompt running on Opus when Haiku would have sufficed is a five-cent mistake compounded across every call. Model selection is the gating lever, and it is the lever the Charter commits to first.

3.1 The three tiers as economic instruments

Claude is offered in three tiers, and the tiers are not "good, better, best." They are three distinct economic instruments with three distinct use profiles. Treating them as a quality ladder produces the wrong answer roughly half the time.

Haiku 4.5 - the distilled tier. Roughly an order of magnitude cheaper per token than Sonnet, fast, capable of substantially more than most teams credit it for. Not weak. Specialized. Correct first choice for classification, extraction from structured input, routing in a cascade, the first pass of retrieval re-ranking, short-form generation against tight templates, and the conversational turns that punctuate longer interactions. For these workloads, Haiku produces output indistinguishable from Sonnet's, at one-tenth the cost.

Sonnet 4.6 - the workhorse. Per-token pricing roughly an order of magnitude above Haiku, capabilities sufficient for the vast majority of enterprise workloads when the prompts are disciplined, and - this is the underappreciated point - capabilities that exceed what most enterprise workloads actually require. Sonnet is the correct default for enterprises that have not yet built the discipline to know which workloads belong on Haiku. It is also the model that runs more workloads than it should.

Opus 4.7 - the frontier instrument. Substantially more expensive than Sonnet, substantially more capable on workloads where capability is the binding constraint. Earns its cost on complex legal reasoning, multi-hop financial analysis requiring auditable inference chains, novel research synthesis, and the production of outputs where a domain expert will

sign their name to what the model produced. Running ordinary workloads on Opus is the most expensive form of cargo-culting in current enterprise AI practice.

3.2 The cascade - math, not theology

The orthodox advice is to cascade: route through Haiku first, escalate to Sonnet when Haiku's confidence is low, escalate to Opus when Sonnet's confidence is low. This is presented as obviously correct. It is not obviously correct, and a non-trivial fraction of enterprise cascades currently in production are losing money relative to flat-Sonnet.

The expected cost of a cascade per query, in its simplest form:

$$\text{“}E[C] = C_{Haiku} + P(esc_S) \cdot C_{Sonnet} + P(esc_S) \cdot P(esc_O \mid S) \cdot C_{Opus} + C_{rework}\text{”}$$

A worked example. Illustratively, Haiku costs $1 per million input tokens and Sonnet costs $3. A typical query consumes 4,000 input tokens and 500 output tokens at the Sonnet stage. Assume the cascade escalates to Sonnet 30% of the time, escalation to Opus occurs in 5% of Sonnet calls, and the misroute rate - queries that should have escalated but didn't, requiring rework - is 4%, with an average rework cost equivalent to two full Sonnet calls.

Flat-Sonnet cost: roughly $0.014 per query. Cascade cost with the misroute term: roughly $0.009 per query - a 36% saving. The cascade wins, but the margin is smaller than the headline 90% Haiku savings would suggest, and the result is sensitive to the escalation and misroute rates in ways the orthodox advice rarely discusses.

Now perturb. Raise the escalation rate to 60%. Raise the misroute rate to 8%. The cascade cost rises to roughly $0.014, identical to flat-Sonnet. At higher escalation or misroute rates, the cascade loses.

> **Contrarian - the cascade is often theater**
>
> For a non-trivial share of enterprise workloads - particularly those with high complexity variance, drifting input distributions, or weak eval coverage - the cascade is theater. It looks like cost optimization. It is, on the math, neutral or negative. The teams that win on this lever measure their escalation and misroute rates monthly and revisit the cascade decision quarterly. The teams that ship a router and forget it are the teams paying for sophistication they are not receiving.

3.3 Sourcing: Anthropic, Bedrock, Vertex

Anthropic first-party API. Most current feature set - new models appear here first, capabilities like prompt caching, the Batch API, extended thinking, and the Files API arrive here before anywhere else. Reference pricing other paths benchmark against. Default for most enterprise deployments unless a specific procurement, residency, or commitment requirement points elsewhere.

Amazon Bedrock. Hosts several Claude models within the AWS ecosystem. For enterprises with significant AWS commitments or specific data residency requirements AWS satisfies more cleanly, this path is rational. Feature parity lags Anthropic's first-party API by weeks to months for new capabilities.

Google Vertex AI. Similarly offers Claude in the GCP ecosystem, with analogous trade-offs for GCP-committed enterprises.

The economic logic of the four levers in this paper is identical across sourcing paths. Only the price points and feature availability shift. The lever you pull is the lever you pull regardless of where the tokens come from.

> **Summary for the CFO**
>
> Claude is sold in three tiers that are not a quality ladder but three economic instruments. Most enterprises run too many workloads on the middle tier. Cascading between tiers is a sophisticated instrument that wins when the inputs to its cost equation are measured and loses when they are not - and the median enterprise cascade today is not measured. The single governance practice with the largest return is the quarterly review of each workload's tier against its competitive position, recorded in the Model Bill of Materials and reconciled against the Charter.

4. Lever Two - Workflow architecture

Model selection sets the price per token. Workflow architecture determines how many tokens you spend per task. The first lever caps the unit cost. The second governs the quantity - and quantity, not unit cost, is where the modal enterprise Claude bill is built and broken.

This is the section where the Opening Scene's 243 concurrent agents come home. The recursion was a workflow failure, not a model failure. No tier choice would have rescued it. The malformed JSON, the supervisor's misinterpretation, the uncapped fan-out - these are workflow primitives, and they are the primitives that, ungoverned, scale linearly with traffic and exponentially with architecture. The teams that win on this lever are not the teams with the smartest agents. They are the teams with the fewest unnecessary calls.

4.1 The Orchestration Multiplier

The Orchestration Multiplier (OM) is the ratio of total tokens consumed per user-visible action to the tokens that would have been consumed by a single, well-prompted Claude call accomplishing the same task with native tool use.

A single tool-using call has an OM of 1.0 by definition. A two-agent supervisor-worker pattern with one round of delegation typically lands at 2.5–3.5×. A three-tier agentic decomposition with parallel workers and a synthesis step lands at 6×–10×. The Opening Scene's incident, at peak recursion, had an OM somewhere between 200 and 300.

In our experience, the modal enterprise Claude deployment has an aggregate OM of 4×–6×, and an OM of 1.5×–2× is achievable on the same workload portfolio with no loss of capability. That delta - three turns of the multiplier - is the largest single source of recoverable waste in current enterprise stacks. It is also invisible to every cost dashboard that reports tokens per call rather than tokens per user action.

4.2 Most agentic workloads should be a single call

Native tool use accomplishes most of what enterprise teams build multi-agent systems to accomplish, at one-fifth to one-tenth the token cost and with measurably higher reliability. A “research agent” that delegates to a “retrieval agent” that calls a vector store is, for the majority of production cases, a single Claude call with a tool that wraps the vector store. The agents don't add reasoning. They add ceremony, and the ceremony is billed.

The three conditions under which multi-agent decomposition genuinely earns its OM:

Parallel and substantial sub-problems. Three workers analyzing three different documents and reporting independently is real parallelism. Three workers each calling the same tool with slightly different parameters is parallelism in name only.

Synthesis the workers cannot do. If the supervisor's role is to aggregate worker outputs requiring reasoning over them as a set, the supervisor earns its cost. If the supervisor is just routing, it is overhead.

Stable decomposition. Workloads where the right decomposition varies query-by-query produce the runaway-recursion risk profile the Opening Scene illustrated.

> **Contrarian - roughly 70% of agentic workflows fail this test**
>
> Apply the three conditions to your current agentic workflows. In our auditing experience, roughly 70 percent of production agentic workflows fail at least two of them. This is the recoverable waste pool, and it is large. The TRAM cell named in the Charter determines the agentic depth the workload is authorized to consume; agentic depth above what the cell permits is an exception requiring re-authorization, not a deployment-time choice the engineering team is free to make.

4.3 The Cache Cliff

The Cache Cliff is the QPS threshold below which prompt caching is net-negative despite appearing in every cost-optimization checklist as a default-on optimization.

Prompt caching reduces the read cost of cached prefix tokens by approximately 90% on hit, at the cost of an approximately 25% premium on the initial cache write and a TTL after which the cache is invalidated. For caching to be net-positive, hit count must exceed roughly 0.28 × write count. Translated to QPS for a five-minute TTL: caching breaks even at roughly 3 to 5 sustained QPS on a given system prompt.

Contrarian - caching is often enabled below the break-even

A substantial fraction of enterprise workloads sit below the Cache Cliff. They have caching enabled because it looks legible to finance, not because anyone ran the break-even. Below the threshold, caching costs more than it saves. The corollary is worse: a system prompt that contains a per-request timestamp, a user identifier, or any variable element invalidates the cache on every call - infinite writes, zero hits, pure premium paid for no benefit. This anti-pattern is present in roughly one-third of enterprise Claude deployments that have caching enabled.

4.4 The Eviction Discipline

The Eviction Discipline is the explicit policy that specifies when tokens leave the working context. It is conspicuously absent from most production stacks.

Multi-turn workflows accumulate context. Each turn's input becomes part of the next turn's context. Each tool call's result joins the working set. Each retrieval result enters and, in most stacks, never leaves. By turn ten of an extended conversation, the working context can carry 40,000 to 80,000 tokens of accumulated history, retrieval, and tool output - most of which is no longer earning its place. The bill on turn ten is not proportional to the work of turn ten. It is proportional to the unmanaged accumulation that preceded it.

The Eviction Discipline borrows from cache eviction in operating systems. It specifies, per workflow: a turn cap, a relevance score, a tool-result truncation rule, and an explicit summarization trigger. A workflow that runs ten-turn conversations with no eviction discipline consumes, by turn ten, roughly five to eight times the tokens of the same workflow with eviction at turn five and summarization at turn seven. Output quality is, in most cases, better with eviction - because the model is not weighting thirty thousand tokens of stale conversational history against the current user input.

Summary for the CFO

The largest single source of recoverable waste in enterprise Claude spending is not the choice of model but the architecture of the workflow wrapped around it. Multi-agent decomposition is sold as sophistication and is, for the majority of workloads, ceremony billed at five to ten times the cost of a single well-prompted call. Prompt caching, defaulted on across most stacks, is net-negative on a substantial fraction of workloads. Multi-turn conversations accumulate context indefinitely because no one has written down the rules for when context is allowed to leave the working set. The single highest-ROI exercise a platform team can run is an Orchestration Multiplier audit of its top ten workloads by spend.

5. The Fifth Lever - Human judgment

The four levers control how much the model costs to run. The Fifth Lever controls which of the model's outputs propagate into the world without a human ratifying them. It has a measurable cost, a measurable return, and a substitution structure with the other four - and it is where control, governance, and ethics converge into a single decision.

5.1 The four questions

A workload's Review Topology answers four questions. The questions are simple. The answers are not, and the work is making them operational rather than aspirational.

Who reviews. The qualification level of the reviewer, named explicitly. A licensed clinician for clinical outputs. A credit officer with specified delegated authority for credit decisions. A senior associate with bar admission for legal drafting. “Anyone” is not an answer. “Subject-matter experts as appropriate” is not an answer. The reviewer's qualification is named, recorded in the Charter, and tied to the regulatory regime that justifies the workload's existence.

What they review. One of four patterns: every output (pre-issue review of 100%), statistical sample (post-issue review of 1–10% with documented sampling strategy), exceptions (review of model-flagged low-confidence outputs plus a stratified sample to validate flagging), or incidents (post-hoc review triggered by complaint or failure).

When they review. Pre-issue, post-issue pre-action, post-action, or post-incident only. The timing decision is the largest single driver of latency cost, and the lever engineering teams are most often tempted to silently downgrade. The Charter signature prevents the silent downgrade.

With what authority. Veto (reviewer can stop the output), override (reviewer can modify with logging), advisory (reviewer flags concerns but output proceeds), or audit (reviewer documents post-hoc without altering the historical decision).

5.2 Review economics

Review economics resolve to a simple decision rule:

$$\text{“Review earns its place when } p \cdot V > C_R + L_R\text{”}$$

Where p is the probability the model errs on this class of workload, V is the value of catching the error before it propagates, C_R is the fully-loaded reviewer cost per case, and L_R is the latency cost incurred by the review step.

> **The expensive insight that conventional HITL writing misses**
>
> p is not constant. It depends on which model the Charter authorized, how much the workload spent on data pre-work, how well-calibrated the prompt is, and how disciplined the workflow architecture is. Review economics are derivative of the other four levers' discipline. A team with weak prompt literacy and poor data pre-work needs more review than a team with strong upstream discipline - and is often less able to afford it because their tokens are already going to remediation rather than to capability. The fastest way to reduce review costs is rarely to optimize review. It is to invest upstream in the four levers that reduce p.

5.3 The four topologies, by TRAM cell

Low cost variance, low output variance - regulated claims adjudication, credit decisions, clinical recommendations: pre-issue review of every output by a qualified reviewer with veto authority. Latency is high; the workload absorbs it because the alternative is regulatory exposure or patient harm. The risk officer's name is on it.

Low cost variance, high output variance - internal research assistance, exploratory document analysis, draft generation: post-issue statistical sampling by a qualified reviewer with override authority. The model's outputs are used immediately; review happens on a sample to calibrate quality and catch drift.

High cost variance, low output variance - high-throughput classification on Haiku, extraction pipelines: exception-based review of model-flagged low-confidence outputs plus a stratified sample to validate the flagging.

High cost variance, high output variance - genuine experimental work, novel applications: post-incident review and aggressive eval coverage. The workload is permitted higher error rates because its purpose is to learn whether the application is viable.

5.4 Governance - the regulatory lens

SR 11-7 model risk management applies in spirit and increasingly in regulatory letter to LLM-driven decisions in financial services. BCBS 239 demands data lineage; in the LLM context,

the analog is context lineage - what entered the working context window for each decision, from where, authorized by whom, recorded how. The EU AI Act Article 14 requires human oversight for high-risk systems and is specific about what oversight must enable. NIST AI RMF provides the most operationally useful framework for non-regulated industries. The Review Topology recorded in the Charter is the artifact that satisfies all of them. The Charter is the governance evidence. The MBOM is its operational record. The quarterly review is its audit trail.

5.5 Ethics - the harder lens

The paper takes a position rather than describing the debate from above. Three categories of decision, treated differently.

Adverse decisions about identified individuals belong with accountable humans regardless of model capability. A loan denial, an employment rejection, a benefit termination, a clinical action with material consequences, a school admission decision, a parole recommendation - these are decisions where the affected individual must have an accountable human to whom appeal can be made. The model may propose, may score, may summarize the case. The decision is human, and the human is named. This is not a function of current model accuracy. It is a function of what accountability requires, and accountability requires a person, not a system.

Decisions that aggregate over populations can be model-driven with human governance of the policy. Pricing, resource allocation, fraud detection thresholds, recommendation algorithms, content moderation policies - these are decisions whose individual-level outputs do not require individual-level human review, because the ethical question is the policy the model implements, not the application of the policy in any given case.

Decisions whose error mode is erosion of trust sit in a middle zone. The right Review Topology depends on the firm's relationship with its users and what it has promised them. A wealth management firm whose clients believe they are receiving human-authored communications has made a promise the Charter must honor. A consumer chatbot whose users know they are talking to an AI assistant has made the opposite promise.

The ethical line, plainly

Explicitly naming the category each workload belongs to - in the Charter, before engineering begins - is the discipline that prevents the firm from accidentally crossing ethical lines it would not have crossed deliberately. The Opening Scene's fintech did not deliberately decide to recurse 243 times on a malformed JSON. It accidentally decided, by not deciding. The Fifth Lever's ethical discipline is the practice of deliberately deciding, in writing, in advance, with the appropriate signatures.

6. Lever Three - Data pre-work

The Fifth Lever determines which model outputs propagate into the world. This lever determines what enters the model's context window in the first place - and the firms that have built Charter discipline find, when they audit their stacks, that the largest remaining efficiency gains live here.

A token filtered upstream - at ingestion, in the index, in the retrieval ranker, in the chunker - is paid for once. A token filtered in-context, by the model itself, is paid for every time the workload runs. Across a high-traffic workload running ten thousand calls a day for a year, the difference between filtering once and filtering 3.65 million times is the difference between a one-time engineering investment and a recurring inference line item that scales with adoption.

6.1 The four upstream interventions

Ingestion-time filtering. Documents and data entering the retrieval system are filtered for relevance, quality, recency, and authority before they are indexed at all. Stale policy documents, superseded contract versions, low-quality auto-generated content, and documents outside the workload's authorized scope are excluded at the source. A document never indexed is never retrieved, never chunked, never tokenized, and never paid for.

Index-time structure. The index is structured to make filtering at query time fast and cheap. Metadata fields indexed. Vector embeddings at appropriate chunk granularity. Hierarchical structure preserved. The question is not which vector database to use, but what the index is capable of distinguishing at retrieval time.

Retrieval-time ranking and filtering. At query time, the system returns more candidates than working context will admit and ranks and filters down. This is the intervention point most “RAG optimization” discussions concentrate on. It is also the one with the least leverage relative to the two upstream of it.

Chunk shape and size. The unit of context retrieved determines how much working context the workload consumes per relevant fact. Chunks too large carry passenger tokens. Chunks too small fragment the reasoning context. Most enterprise RAG stacks have never tuned chunk shape against the actual workload they serve.

6.2 Filter once, or filter every call

Consider a workload that retrieves, on average, eight chunks per query, each averaging 400 tokens. Without ingestion-time filtering, roughly 40% of those chunks are passengers - present but not contributing. The workload runs 5,000 queries per day.

"Passenger tokens per day: 8 × 400 × 0.40 × 5,000 = 6,400,000.
Roughly 1.9 billion per year."

Every one paid for at the workload's per-token rate, every one read by Claude on every call, every one contributing to latency and to the model's attention budget. A one-time ingestion filtering project - three engineer-months of work, a documented filter policy, automated re-application as documents enter the corpus - eliminates roughly 70% of the passenger volume. The recurring saving, at Sonnet pricing, is on the order of several hundred thousand dollars per year for this single workload. The fixed cost of the project pays back inside a quarter. The saving compounds in year two, and in year three, and continues compounding as long as the workload runs.

6.3 The Reasoning–Retrieval Frontier

Reasoning tokens and retrieval tokens are partially substitutable. The right balance for a given workload is the point that minimizes total cost-per-correct-answer subject to the workload's latency and accuracy constraints.

At one extreme - pure reasoning, no retrieval - the workload depends entirely on the model's parametric knowledge. Cheap, fast, reliable on knowledge the model has and unreliable on knowledge it does not. At the other extreme - heavy retrieval, minimal reasoning - the workload depends entirely on the retrieval substrate. Expensive per call, slower, reliable on dynamic or proprietary content.

Contrarian — extended thinking is sometimes cheaper than retrieval

For a substantial class of enterprise questions - particularly internal questions about general business, technical, or domain knowledge that does not change quarter to quarter - retrieval is overhead the workload is paying for to access information the model already has. The discrimination is empirical, not theoretical. Measure the accuracy of the workload's questions answered by extended thinking alone, against the accuracy with retrieval, against the cost of each approach. In our experience, roughly 20 to 30 percent of enterprise RAG workloads are operating in this category and have not measured the alternative. The honest counter-position: for workloads against proprietary data, against rapidly changing data, or against data with regulatory provenance requirements, retrieval is not optional.

6.4 Skills and deferred-context mechanisms

Claude's Skills system, MCP servers, and analogous mechanisms share an economic property: they move context from always-resident to retrieved-on-demand. A workload whose system prompt contains detailed instructions for fifteen different document types - 12,000 tokens total - pays for fourteen of the fifteen as resident liability on most calls. Under a Skills architecture, the system prompt contains a 600-token catalog and the specific 800-token instruction set is loaded only when the call requires it. Per-call resident tokens drop from 12,000 to 1,400 - an 8:1 compression ratio. The savings compound across every call the workload runs.

6.5 When pre-work is mandatory

Regulated financial workflows produce decisions whose data lineage must be auditable. A credit decision, a trade recommendation, a regulatory filing, a Suspicious Activity Report - each must be traceable to the data that supported it, with the provenance, the version, and the authorization recorded. Retrieval architectures produce this lineage naturally; pure reasoning architectures do not. For these workloads, retrieval is mandatory not because it improves accuracy but because it produces the audit trail. The Charter signature from the risk officer is the line where this is committed.

Summary for the CFO

A token filtered upstream is paid for once; a token filtered in the model is paid for every call. The largest recoverable savings in most enterprise Claude deployments are not in clever ranking algorithms but in what enters the index at all and what shape the indexed content takes - interventions that pay back in single-digit-month timeframes and compound for the life of the workload. The discipline is upstream, the savings are downstream, and the artifact that aligns them is the Charter's data section.

7. Lever Four - Prompt literacy

The right data work upstream determines what enters the model's context window. This lever determines whether the humans on either end of the system know how to communicate with what is in that window. It is the lever that operates furthest from the platform team's direct control and closest to the workload's actual users.

7.1 The Prompt Literacy Ladder

Level One - Consumer. Types natural-language questions, reads natural-language answers. No model of the model. Workloads safe at this level: single-turn Q&A against well-engineered system prompts, narrowly-scoped conversational interfaces. Training cost to move to Level Two: roughly four hours of structured exposure with hands-on practice.

Level Two - Structured user. Understands prompts have anatomy - context, instruction, examples, constraints, format - and structures inputs deliberately. Knows when to provide examples and when not to. Workloads: drafting, analysis, summarization, moderate-complexity multi-turn workflows. Training cost to Level Three: roughly two days of structured training plus four to six weeks of practice with feedback.

Level Three - Configurator. Writes effective system prompts, defines tool descriptions, structures retrieval grounding, configures conversational surfaces to produce workloads other users will consume. Workloads: building reusable internal AI tools, configuring department-specific agents, owning the prompt layer of moderate-complexity workloads under platform-team review. Training cost to Level Four: three to six months on real production prompts with senior review.

Level Four - Prompt engineer. Designs prompt architectures for production. Understands caching invalidation patterns. Debugs prompts that pass tests and fail in production. Writes evals that catch the failure modes their prompts are most prone to. Population per enterprise: typically a handful, regardless of size.

Level Five - Prompt architect. Designs prompt strategy across the workload portfolio. Defines the patterns the organization uses. Owns the Prompt-as-Code discipline. Represents prompt design in Charter reviews. Population per enterprise: typically one to three, often co-located with the Token Architect.

7.2 Prompt-as-Code

A prompt is the full set of instructions, examples, constraints, tool definitions, and grounding sources that condition the model's generation for a given call. The configuration surface of an enterprise chat interface is itself a prompt artifact - often the largest one in the workload, and almost always the least disciplined.

System prompts grow. Tool descriptions accumulate. Grounding source lists expand as new content gets indexed. Conversational scaffolding gets added in response to specific user complaints. None of this is captured in the version control discipline most enterprises apply to their application code. The configuration drifts, the workload's behavior drifts with it, and the platform team discovers months later that the prompt powering a critical workflow has been edited fourteen times by people who do not remember what they changed.

The discipline this paper recommends is **Prompt-as-Code**: prompts and configuration surfaces are versioned, diffed, reviewed, canaried, and rolled back with the same discipline applied to application code. Eval gates run on prompt changes the same way regression tests run on code changes. This is, in our auditing experience, the single most absent discipline in enterprise AI programs.

> **Contrarian — prompt literacy training has higher ROI than platform-team hiring**
>
> The orthodox response to AI program underperformance is to hire more platform engineers, more prompt engineers, more AI specialists. For a substantial fraction of mid-sized enterprises, this is a wrong-direction investment. A $200,000 training program that moves 500 users from Level One to Level Two produces measurable per-interaction quality improvements across hundreds of thousands of interactions per quarter. The same $200,000 spent on a single engineer-year produces improvements on a handful of workloads. The math frequently favors training the population over hiring more specialists — and the Charter discipline that surfaces this trade-off is the discipline that prevents the default toward hiring.

7.3 User education that actually works

The literature on prompt training for business users ranges from condescending to engineering-only, and the gap is where most enterprises actually live. The training that works has three properties.

Anchored in the user's actual job. A salesperson learns prompt engineering as it applies to drafting client communications, summarizing call transcripts, analyzing pipeline data - using the actual systems and content they work with daily.

Paired with feedback loops. The user produces a prompt, sees the output, and receives structured feedback. Without feedback, users learn idiosyncratic and often counterproductive habits.

Explicit about when not to use AI. The most expensive prompt is the one that produces a bad answer the user trusts. Training that teaches users to recognize the workloads their AI tools are suited for - and the workloads they are not - pays back in errors prevented, not just productivity gained.

8. The operating model

A Charter signed once is a document. A Charter reviewed quarterly, against a ledger of realized outcomes, by named roles on a defined cadence, is a discipline. The difference is the operating model - the institutional machinery that converts the planning posture this paper has argued for into a sustained organizational capability.

8.1 The Token Architect

A new staff-level discipline, named explicitly because the absence of a name is part of why it has not yet emerged in most enterprises. The Token Architect owns the operating model for AI economics across the workload portfolio.

Charter review. For every new AI-enabled workload before engineering begins. The Token Architect is not the approver - the business owner, finance partner, and risk officer are - but the technical conscience of the review.

Model Bill of Materials. Every workload, its model tier, its workflow architecture, its data sources, its prompts, its review topology, the date of last review.

Governance dashboard. Surfaces exceptions: workloads whose realized economics have diverged from Charter projections, workloads whose tier has not been reviewed in two quarters, workloads whose prompts have changed outside the Prompt-as-Code discipline.

Quarterly Charter review cycle. Convenes the cross-functional reviews, produces the one-page summaries for senior management, recommends ratify/revise/retire decisions for each workload.

Portfolio-level KPI. Not “reduce inference spend,” which is the wrong KPI, but Alpha-net per dollar of inference spend, computed per workload and aggregated to the portfolio.

The seniority of the role matters. The Token Architect must have the standing to push back on business owners proposing Charters that fail technical or governance scrutiny, the technical depth to challenge platform-team architectural decisions, and the financial fluency to discuss the portfolio's economics with the CFO in the CFO's vocabulary. Staff-level in mid-sized firms. Director or VP in large enterprises. Hiring below the appropriate level is the most common implementation failure.

8.2 The Model Bill of Materials

A structured record of every AI-enabled workload. Fields: workload name, owning business unit, named business owner, alpha position, TRAM cell, model tier, workflow architecture pattern, data sources and ingestion filter version, prompt artifact references and versions, review topology, kill criteria, expected/stretch/runaway thresholds, date of last Charter review, ratify/revise/retire status from the most recent review.

The MBOM is the artifact that the regulatory frameworks - SR 11-7, BCBS 239, EU AI Act Article 14, NIST AI RMF - already implicitly require. Most enterprises build them piecemeal, one workload at a time, in different formats across different teams. The MBOM consolidates the requirement into a single ledger, owned by a single role, reviewed on a single cadence. The auditors notice. The regulators notice. The board notices.

8.3 Dashboards and cadence

Platform-team dashboard, real-time. Per-workload token consumption against Charter thresholds, Orchestration Multiplier per workload, cache hit rates and Cache Cliff status, prompt-version drift, eval pass rates, escalation rates in cascade routers, latency percentiles, and exceptions on any of the above.

Executive dashboard, monthly. Total inference spend against budget, spend by alpha category, Alpha-net per dollar across the portfolio, workloads ratified/revised/retired in the most recent review cycle, runaway-scenario incidents, the three workloads with the largest positive and negative variance to their Charters.

The dashboards are not the discipline. The Charter and the review cycle are the discipline. The dashboards are the visibility that makes the discipline sustainable, because invisible disciplines decay and visible ones compound.

8.4 The rhythm

Daily. Platform team monitors the real-time dashboard, addresses exceptions.

Weekly. Token Architect reviews accumulated exceptions, escalates patterns rather than incidents.

Monthly. Executive dashboard published. Finance reviews variance to budget. Business owners review workloads' performance against Charters.

Quarterly. Full Charter review cycle. Every workload reviewed. Ratify/revise/retire decision documented. Updated Charters re-signed. Portfolio summary produced for senior management.

Annually. The operating model itself is reviewed. The meta-discipline that prevents the operating model from ossifying into bureaucracy.

This cadence is borrowed from the rhythms mature enterprises already run for capital projects, cloud spend, and regulatory compliance. It is not new machinery. It is the existing machinery, applied to AI economics, with the artifacts and roles named for the domain.

9. The Anti-Patterns Gallery, and the Frontier

Ten rogues. Each is named, with symptom, root cause, telemetry fingerprint, illustrative dollar impact, and fix. Each is also a Charter discipline failure as much as a technical failure - and the Charter is the upstream fix.

1. The Recursive Sub-Agent

Symptom. Token consumption spikes 5–50× without proportional traffic increase.

Root cause. Agentic workflow with uncapped recursion depth or fan-out, triggered by a malformed input or unexpected tool response.

Fingerprint. Per-call token count distribution develops a long right tail; concurrent-agent count exceeds workflow's projected maximum.

Impact. The Opening Scene's $312,000 night.

Fix. Depth and fan-out caps enforced in code. Circuit breaker at the Charter's runaway threshold. Automated kill that requires human authorization to resume.

2. The Cached Timestamp

Symptom. Cache write costs accumulate, cache hit rate stays at zero.

Root cause. Variable content embedded in a cached prefix, invalidating the cache on every call.

Fingerprint. Cache write count equals call count. Cache read count near zero.

Fix. Audit cached prefixes for variable content. Move variable elements out of the cached region. Measure hit rate weekly.

3. The Phantom RAG

Symptom. Retrieval-augmented workload performs no better than the same workload without retrieval.

Root cause. The workload's questions are answerable from the model's parametric knowledge; retrieval adds tokens without adding correctness.

Fingerprint. A/B test of with-retrieval versus without-retrieval shows no accuracy delta.

Fix. Run the A/B test. Migrate qualifying workloads to extended thinking without retrieval.

4. The Eternal Conversation

Symptom. Per-call token consumption grows monotonically with conversation length.

Root cause. No eviction discipline; full conversation history carried in working context indefinitely.

Fingerprint. Input token count correlates with turn number. Conversations beyond turn ten consume 5–8× the tokens of turn one.

Fix. Implement eviction policy: turn cap with summarization, relevance scoring, explicit reset triggers.

5. The Opus Habit

Symptom. Workloads running on Opus that produce outputs indistinguishable from Sonnet outputs on the same inputs.

Root cause. Tier was chosen at deployment time, never revisited; no eval comparison against cheaper tier.

Fingerprint. Workload runs on Opus, no documented eval comparison against Sonnet exists, last tier review date in MBOM is "never."

Fix. Run the eval. Downgrade where indistinguishable. Document the decision in the next Charter review.

6. The Unbounded System Prompt

Symptom. System prompt has grown to 15,000+ tokens through accumulation.

Root cause. No Prompt-as-Code discipline. Prompts edited ad-hoc without review or eval gating.

Fingerprint. Prompt size has grown without bound over multiple quarters. Prompt change history shows many small additions, no consolidations.

Fix. Implement Prompt-as-Code. Decompose accumulated instructions into Skills or analogous deferred-context mechanisms.

7. The Cascade That Doesn't

Symptom. Haiku-Sonnet-Opus cascade in production, but actual savings versus flat-Sonnet are zero or negative.

Root cause. Escalation rates higher than the cascade was designed for, or misroute costs higher than projected, or both.

Fingerprint. Escalation rate to Sonnet exceeds 50%. Misroute-driven rework rate exceeds 5%.

Fix. Measure the escalation and misroute rates. Run the break-even math. Switch to flat-Sonnet if the cascade is not earning its place.

8. The Ungoverned Tool

Symptom. Tool calls per user-action exceed projection.

Root cause. Tool definitions are unclear. The model re-fetches the same information because the prompt does not specify retention.

Fingerprint. Same tool called with the same parameters multiple times in a single workflow trace.

Fix. Clarify tool descriptions. Add prompt instructions about retention. Consider caching tool results within the conversation.

9. The Silent Downgrade

Symptom. Production workload's review topology has weakened from Charter specification.

Root cause. Engineering team facing latency or throughput pressure silently relaxed the review step.

Fingerprint. Review-step completion rate has decreased without a Charter revision authorizing it.

Fix. Governance dashboard surfaces the variance. Token Architect escalates. Either the topology is restored or the Charter is formally revised with risk officer signature.

10. The Orphaned Workload

Symptom. Production workload consuming non-trivial tokens with no named business owner, no recent review, no documented current purpose.

Root cause. Workload's original sponsor moved on. No one explicitly inherited ownership.

Fingerprint. MBOM record shows owner field blank or invalid. Last Charter review date exceeds two quarters.

Fix. Identify the workload's current consumers. Assign an owner or retire the workload.

"Each anti-pattern is a Charter discipline failure as much as a technical failure. Technical anti-patterns are downstream of governance gaps, and the Charter is the upstream fix."

9.1 The Frontier - three calibrated predictions

Prediction one. By end of 2027, the per-token price of the workhorse tier will have fallen sufficiently that more than half of currently-running cascade architectures will no longer be economically justified versus flat-mid-tier deployment. Falsification: workhorse-tier pricing in late 2027 has not fallen at the rate observed over the prior 36 months, or escalation rates in production cascades have fallen faster than pricing.

Prediction two. By end of 2027, agentic decomposition will have inverted its current economic profile in at least one direction: either sub-agents become substantially cheaper than supervisors through specialized small models, or the workhorse tier's single-call capability will have advanced sufficiently that the current 70% of agentic workloads that should be single calls becomes 90%. Falsification: neither trend dominates; the current ambiguous middle persists into 2028.

Prediction three. By end of 2028, regulatory frameworks in financial services and healthcare in at least two major jurisdictions will require some form of context lineage documentation, making the MBOM-and-Charter discipline either explicit regulatory requirement or de facto industry standard. Falsification: regulatory frameworks remain focused on model approval rather than context lineage.

These predictions are commitments to falsification criteria, made publicly so the paper's framework can be evaluated against subsequent reality. A paper that does not commit to falsifiable predictions is a paper that cannot be wrong, which means it cannot have been right.

Closing

The fintech in the Opening Scene did not fail because of a malformed JSON. It failed because a four-million-dollar inference line item had no Charter, no owner, no runaway scenario, no kill criteria, and no quarterly review. The technical failure was downstream of the governance gap. The governance gap was downstream of a posture.

This paper has argued that the posture is wrong and has specified the discipline that corrects it. Four levers - model selection, workflow architecture, data pre-work, prompt literacy - control the workload's economics. A fifth lever - human judgment - controls which of the workload's outputs propagate into the world. A planning-time artifact - the Token Strategy Charter - commits the firm to the levers' configuration before engineering begins, signed by the business owner, the platform team, the finance partner, and the risk officer. A role - the Token Architect - owns the operating model. A ledger - the MBOM - records the portfolio. A cadence - the quarterly review - maintains the discipline. A KPI - Alpha-net per dollar of inference spend - measures whether the discipline is producing the economic value it claims.

The firms that build this discipline will compound margin against firms that do not. The compounding is not visible in any single quarter. It is visible across years, in the spread between firms whose AI programs produce economic value and firms whose AI programs produce expensive activity. The Cost Clock ticks down. The Alpha Clock ticks up and then stops. The firms that govern both clocks deliberately, in writing, with the appropriate signatures, build the operating posture that survives the next price drop, the next capability frontier, and the next regulatory framework. The firms that do not, build the conditions for the next 11:47 PM email.

> **The choice**
>
> The Charter is the artifact. The discipline is the practice. The choice is whether the practice begins this quarter or after the next incident makes it unavoidable.

Appendix A Glossary of coined terms

Alpha Half-Life Taxonomy. Classification of AI-enabled workloads by the expected decay rate of their competitive advantage: structural (24+ months), operational (9–18 months), commodity (under 6 months), expired.

Cache Cliff. QPS threshold below which prompt caching is net-negative due to write premium and TTL waste. Typically 3–5 sustained QPS on a given cached prefix.

Context Balance Sheet. Accounting frame classifying every token in working context as asset (load-bearing), liability (resident but unearning), or expense (consumed once and discharged).

Eviction Discipline. Explicit policy specifying when tokens leave working context — turn caps, summarization triggers, relevance scoring, tool-result truncation.

Inference Alpha Equation. Alpha_net = (R_AI - R_baseline) - C_inference - C_governance – C_opportunity. The full P&L view of an AI-enabled workload.

Model Bill of Materials (MBOM). Structured ledger of every AI-enabled workload - tier, architecture, data sources, prompts, review topology, Charter review history.

Orchestration Multiplier (OM). Ratio of tokens consumed per user-visible action to tokens consumed by a single well-prompted call accomplishing the same task. Modal enterprise OM: 4–6×. Achievable: 1.5–2×.

Prompt-as-Code. Doctrine treating prompts and configuration surfaces as first-class software artifacts subject to versioning, review, eval gating, and rollback.

Prompt Literacy Ladder. Five-level capability framework: Consumer, Structured User, Configurator, Prompt Engineer, Prompt Architect.

Reasoning–Retrieval Frontier. Substitution curve between extended thinking tokens and retrieval tokens. Optimal point varies by workload's content profile.

Review Topology. Specification of a workload's human review: who, what, when, with what authority.

Token Architect. Staff- or director-level role owning the operating model for AI economics across the workload portfolio.

Token Risk Appetite Matrix (TRAM). 3×3 governance instrument crossing cost variance tolerance against output variance tolerance. Each cell implies sanctioned model tier, caching posture, agentic depth, and review topology.

Token Strategy Charter. One- to three-page planning-time artifact specifying a workload's purpose, alpha position, TRAM cell, three traffic scenarios, review topology, kill criteria, transparency cadence, and named signatories.

Two-Clock Problem. Trade-off between the Cost Clock (per-token prices falling) and the Alpha Clock (competitive advantage decaying). Spend through inefficiency while alpha is rare; optimize ruthlessly once it is not.

Appendix B Core formulae

Cascade expected cost

$$\text{“}E[C] = C_{Haiku} + P(esc_S) \cdot C_{Sonnet} + P(esc_S) \cdot P(esc_O|S) \cdot C_{Opus} + C_{rework}\text{”}$$

Cache break-even

$$\text{“Caching net-positive when } N_{hits} > 0.28 \cdot N_{writes}\text{”}$$

Review economic threshold

$$\text{“Review earns its place when } p \cdot V > C_R + L_R\text{”}$$

Where p is error probability, V is the value of catching the error, C_R is reviewer cost, L_R is latency cost.

Inference Alpha

$$\text{“}Alpha_{net} = (R_{AI} - R_{baseline}) - C_{inference} - C_{governance} - C_{opportunity}\text{”}$$

Appendix C Charter template

A workload Charter contains seven structured sections.

1. Purpose and alpha position

Two-paragraph statement of the workload's business outcome, the baseline it improves against, and its Alpha Half-Life classification.

2. TRAM cell

Stated cost variance tolerance, stated output variance tolerance, derived sanctioned configuration envelope.

3. Three scenarios

Expected, stretch, and runaway projections - for monthly token consumption, monthly cost, and per-call cost - with operational thresholds at which each scenario triggers different actions.

4. Review topology

Who reviews, what they review, when they review, with what authority. Drawn from the Fifth Lever framework.

5. Kill criteria

Economic, quality, and strategic triggers. Automated where possible, escalated where automation is unsafe.

6. Transparency cadence

Format and schedule for reporting against the Charter to senior management.

7. Signatories

Business owner, platform team lead, finance partner, risk officer (where required), with date of signature and date of next scheduled review.

Appendix D The Monday Diagnostic

Eleven questions a platform team can run against its own stack this week. For each: pull the telemetry, score, identify the highest-leverage intervention.

1. What is the Orchestration Multiplier of your top ten workloads by spend?
2. What fraction of your cached prefixes contain variable content invalidating the cache on every call?
3. What is the escalation rate of your production cascade routers? The misroute rate?
4. Which of your workloads have not had their model tier reviewed in two quarters?
5. Which of your retrieval workloads have measured their accuracy against extended-thinking-without-retrieval as a baseline?
6. What fraction of your system prompts exceed 8,000 tokens? When were they last consolidated?
7. Which workloads' multi-turn conversations carry full history with no eviction discipline?
8. What is your Review Topology for each production workload? Where has it silently downgraded from its Charter specification?
9. Which workloads in production have no named business owner in your MBOM?
10. What is your firm's distribution of users across the Prompt Literacy Ladder? What is the training program to move the next cohort up a rung?
11. What is your firm's Alpha-net per dollar of inference spend, computed across your workload portfolio, this quarter compared to last?

> **Scoring**
>
> Three or fewer questions answered with documented data: a firm operating reactively. Seven or more: a firm operating with token discipline. The gap between those scores, multiplied across the firm's AI portfolio and compounded across years, is the difference this paper has been about.