COST & FINOPS FOR AGENTIC AI

Reliability is half the question. Cost is the other half.

Why agentic AI bills explode even as tokens get cheaper, where the money actually goes, and how to govern performance against cost as a first-class discipline, not a line item you discover on the invoice.

FOR

CFO, CIO, Head of AI, FinOps, procurement

COMPANION BRIEFINGS

01 Operating model, 02 Verification, 03 Agent governance

GROUNDED IN

The LLM gateway, the verification paper, real engagements

DIRECT LINE

teams@cohorte.co

02 THE PROBLEM

The bill almost nobody forecasts.

The price of a token has fallen by roughly three quarters, which sounds like relief. Then the first agentic system ships, and the invoice arrives many times larger than anyone modelled. Both facts are true at once, and the reason is the whole subject of this briefing.

> Tokens got cheaper. Agents got hungrier. The bill went up anyway.
>
> THE CHEAPER-PER-TOKEN ERA IS THE MORE-EXPENSIVE-PER-TASK ERA

A chatbot makes one model call per answer. An agent does not. It plans, calls a tool, reads the result, reasons again, retries when it fails, pulls in context, checks its own work, and only then responds. A single user request can become hundreds of model calls. **Multiply a token that is 75% cheaper by hundreds of times more tokens, and the bill still climbs an order of magnitude or more.**

This is why teams that budgeted for a modest increase open invoices fifty or sixty times larger than last year, while usage "barely moved." It did not barely move. Each request quietly turned into a small workload. And because the cost is invisible in the demo, where one clean question runs once, nobody prices it until it is in production at volume.

The uncomfortable end of the curve: some organisations are discovering the AI cost more than the people it was meant to help. Not because AI is expensive, but because nobody decided, per workflow, how much reliability was worth how much money.

03 THE THESIS

Not "does it work?" but "what reliability, at what cost?"

A system that answers beautifully and costs a dollar a request will break the budget the week it succeeds. Reliability in the abstract is not the goal. The goal is the right reliability for a given workflow, bought at a price the business can carry, with the trade-off decided on purpose rather than discovered after the fact.

Cost is a governance question, not an accounting one. The same architecture that decides what an agent may do, and proves how reliable it is, is where its cost has to be metered, attributed, and capped. Reliability and cost are two readings on the same workflow, and they are decided together: the reliability bar from Briefing 02, the cost ceiling from this one.

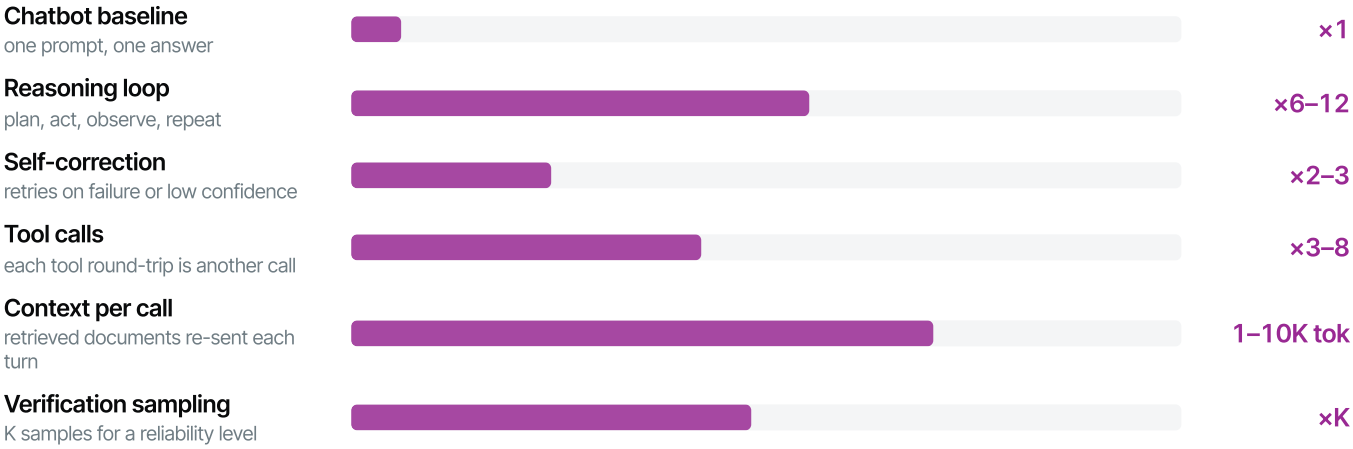
The discipline is concrete. Every workflow gets a target reliability and a cost ceiling before it ships. Every call is metered and attributed to an owner. The cheapest configuration that clears the reliability bar is the one that goes to production, not the most capable model available, and not the cheapest model that quietly fails the bar.

PERFORMANCE × COST	A model is not "good" or "bad". It is **good enough for this task at this price.** The same model can be the right choice for triage and the wrong choice for a board memo.
COST PER OUTCOME, NOT PER TOKEN	Token price is a distraction. **The number that matters is cost per completed, verified outcome**, including the retries and the verification samples it took to get there.
DECIDED UP FRONT	The reliability bar and the cost ceiling are set in the scoping brief and enforced at the gateway. **Not negotiated with the invoice three months later.**

04 COST ANATOMY

Where one request becomes hundreds of calls.

A useful mental model: a single user request is not one model call, it is a workload. Each multiplier below stacks on the others. The chatbot you priced is the leftmost bar; the agent you shipped is all of them, multiplied together.



The compounding is the point. These do not add, they multiply. A six-step loop, each step re-sending 4K of context, two retries, five tool calls, with verification on the final answer, is not "a few cents". It is the difference between a system that pays for itself and one that does not. You cannot manage what you have not decomposed.

05 THE CONTROL PLANE

Meter, attribute, enforce. At the gateway.

Cost control is not a spreadsheet exercise after the month closes. It is the same LLM gateway from Briefing 01, doing three jobs on every single call before the spend happens. Nothing reaches a model without passing through it, so nothing escapes the meter.

1	**Meter**	Record tokens in and out, the model, the latency and the computed cost of every call, the instant it happens.	PER-CALL LEDGER
2	**Attribute**	Tag each call to an owner, a workflow and an agent. Cost stops being one opaque bill and becomes a line per team.	OWNER · WORKFLOW
3	**Enforce**	Check spend against the workflow's budget. Within budget runs; near it alerts; over it throttles or routes to a cheaper model.	ALLOW / ALERT / CAP

The shift this enables is from **"the AI bill was €X this month"** to **"this workflow costs €Y per outcome, owned by this team, against a €Z ceiling."** One of those sentences can be managed. The other can only be panicked about.

Because enforcement lives at the gateway, the response to a runaway agent is automatic, not a Monday-morning discovery. A budget breach can throttle, downgrade the model, or hold the workflow for review, the same way a reliability breach holds it at the verification gate.

WHAT GETS RECORDED PER CALL

Tokens in / out, the **model** and provider, **latency**, and computed **cost**.

Owner, workflow, agent, the three keys that turn a bill into a budget.

Outcome link, tying the spend to whether the task actually completed and passed verification.

WHERE IT RUNS

The LLM gateway service in the open-source stack. Metering and budget enforcement are built in, not bolted on.

06 MODEL ROUTING

Don't pay frontier prices for triage work.

The single biggest lever. Most requests are easy; a few are hard. Sending every request to the most expensive model is like flying every parcel first-class. Route by difficulty, and let the reliability level decide when to escalate, so cost follows need instead of habit.

MOST TRAFFIC
Small / cheap model
Classification, extraction, routing, simple drafting. Verified to clear the bar on these tasks. Cents, not euros.

WHEN IT'S HARD
Frontier model
Ambiguous reasoning, high-stakes generation, anything the small model abstains on or fails verification for.

THE DECIDER
The reliability level
Escalation is not a guess. If the cheap model's certified reliability clears the workflow's bar, it ships. If not, route up.

The trap to avoid is routing on vibes ("this feels hard"). Route on evidence. Each model carries a reliability level per task class from the verification layer. The router picks the cheapest model whose certified reliability clears the bar for that task, and escalates only the residue. **The expensive model earns its cost on the requests that need it, and nowhere else.**

Done well, this is where the order-of-magnitude savings live: a large share of traffic moves to a model that costs a fraction of frontier pricing, with no loss of guarantee, because the guarantee is measured, not assumed.

And the router is only half of it. Left to themselves, engineers reach for the most capable model to be safe, the way you would copy the most senior colleague's answer. The other half of the saving is **change management**: training the team that the best model is rarely the right one for the task, and that picking a cheaper model which clears the bar is the professional call, not a corner cut. FinOps is a habit before it is a tool.

07 THE COST LEVERS

Six levers, in order of impact.

None of these trades away reliability. Each one removes spend the system did not need to make in the first place. The savings stack, and the gateway is where most of them are pulled.

LEVER	HOW IT WORKS	TYPICAL EFFECT
Model routing	Cheap model by default, frontier only on hard or failed-verification cases.	**The largest lever.** Moves the bulk of traffic to a fraction of the price.
Certified sequential stopping	Stop sampling for a reliability level as soon as the answer is certain, instead of running a fixed budget.	**45–52% fewer samples** with no loss of guarantee (verification paper).
Prompt & result caching	Reuse responses and cached prompt prefixes for repeated or near-identical calls.	Large on workloads with repetition; near-free on cache hits.
Context budgeting	Send only the context the task needs, ranked and truncated, not the whole corpus every turn.	Cuts the per-call token bill, which compounds across the loop.
Abstention	Let the system decline low-confidence requests to a human instead of burning retries on them.	Removes the most expensive calls: the ones that fail anyway.
Batching & off-peak	Group non-urgent work; use batch and lower-priority tiers where latency allows.	Discounted rates on work that does not need to be instant.

> Most AI overspend is not the price of intelligence. It is the price of asking for more of it than the task required.

08 THE FRONTIER

Pick the cheapest config that clears the bar.

For any workflow there is a set of configurations: which model, how many verification samples, how much context. Each has a reliability and a cost. The job is not to maximise reliability or minimise cost. It is to find the cheapest configuration whose certified reliability clears the workflow's bar, and ship that one.

CONFIGURATION	CERTIFIED RELIABILITY	RELATIVE COST / OUTCOME
Frontier model, K=20 samples	98%	×30 — over-engineered for a 90% bar
Frontier model, sequential stop	97%	×15
Mid model, sequential stop	**93% — clears a 90% bar**	**×4 — the pick**
Small model, single sample	82% — below bar	×1 — cheap, but fails the requirement

Illustrative, not a price list. The numbers depend on the workflow, the bar, and the models of the day. The discipline is what travels: a reliability bar makes the cost decision objective. Without it, teams either gold-plate (pay ×30 for a ×4 requirement) or under-build (ship the ×1 config that quietly fails). The bar turns a guess into an engineering choice.

09 THE CADENCE

Unit economics, not a quarterly surprise.

Once cost is metered and attributed, it becomes a unit-economics question a CFO recognises. The metric that matters is cost per completed, verified outcome, by workflow, watched on a cadence, with the honest comparison made out loud.

Cost per outcome, by workflow. Total spend divided by completed, verified outcomes, per workflow and owner. This is the number that tells you whether a system pays for itself, and which workflows to scale, fix, or retire.

Budgets with automatic enforcement. Each workflow carries a ceiling. The gateway alerts as it approaches and throttles or downgrades when it breaches, so the failure mode is a slower workflow, not a surprise invoice.

The honest comparison. Put the cost per outcome next to the fully-loaded cost of the human process it supports. Sometimes AI wins decisively; sometimes it only wins after routing and verification discipline; sometimes the right answer is not to automate that workflow at all. Say which, in writing.

THE CADENCE

Real-time. Per-call metering and budget alerts at the gateway.

Weekly. Cost-per-outcome by workflow reviewed beside reliability drift. The two move together.

Monthly. Owner-level budgets vs actuals. Routing and caching revisited as prices and models change.

Per change. Any model swap re-runs both the reliability level and the cost-per-outcome before it ships.

THE ONE-LINE TEST

Can you state the cost per outcome of your top workflow today, by owner? If not, you are not yet doing FinOps; you are receiving an invoice.

10 COMMON MISTAKES

Four ways the bill runs away.

Every one of these has been seen in the first months of a production agentic system. They are the default, not the exception, and each has a one-line fix.

MISTAKE 01

No meter.

Spend is one monthly number with no breakdown by team, workflow or agent. Nobody can act because nobody knows where it goes.

Instead: meter and attribute every call at the gateway, from day one.

MISTAKE 02

Frontier for everything.

The most capable model handles triage, classification and drafting alike, at first-class prices for economy-class work.

Instead: route by difficulty; escalate on the reliability level, not on habit.

MISTAKE 03

No bar, so no ceiling.

Without a target reliability, there is no objective way to choose a configuration, so teams gold-plate or under-build.

Instead: set the reliability bar and the cost ceiling together, in the scoping brief.

MISTAKE 04

Burning retries on lost causes.

The system retries low-confidence requests that were never going to succeed, paying repeatedly to fail.

Instead: abstain early and escalate to a human; the cheapest failed call is the one not made.

11 REFERENCES

References. Each claim, anchored.

The methods and sources behind the claims on the previous pages. The research record lives at **teams.cohorte.co/research**.

Mouzouni, C. (2026). Black-Box Reliability Certification for AI Agents via Self-Consistency Sampling and Conformal Calibration. Preprint. The reliability level and the 45–52% saving from certified sequential stopping. Code at github.com/Cohorte-ai/trustgate.

The LLM gateway. Metering, cost attribution and budget enforcement in the open-source stack. github.com/Cohorte-ai. The control plane described on page five.

The Enterprise Agentic Platform (Cohorte, 2026). The four-layer architecture and the role of the gateway. cohorte.co/playbooks/the-enterprise-agentic-platform.

On the consumption pattern. Falling per-token prices alongside rising per-task token consumption is now widely reported across enterprise AI deployments; the multipliers on page four are typical ranges, not a fixed benchmark, and should be measured against your own gateway data.

Companion: Briefing 02, Verification & evaluation. The reliability level that this briefing routes and budgets against.

Companion: Briefing 01, The operating model. Where cost sits among scoping, verification, governance and monitoring.

BRING THE BILL

One discovery call. We model the cost before you commit.

Sixty minutes with our founder. We map one workflow's cost anatomy against your stack, set a reliability bar and a cost ceiling, and you leave with a one-page model the CFO can read. No deck.

teams@cohorte.co

Cohorte SAS · Société par actions simplifiée, registered in France · founded September 2022 · Paris & Rabat

The full briefing series, the open-source stack, and the research record at **teams.cohorte.co/trust-and-governance**