TensorCost

FINOPS FOR AI

tensorcost.com
Two-week read-only pilot
Written findings report — either way

THE COST CONTROL PLANE FOR AI

The accelerators you own and the tokens you rent — *one bill you can defend.*

A training job holding a MIG slice and an inference workload spread across four providers are the same question asked twice — what did this cost, and would something cheaper have done it? TensorCost prices your own hardware properly, joins that to token spend in one attribution model, and measures every accepted change against your own spend before and after.

6	48h	1·2·3·7	$/call
Sources in one schema — five inference providers plus your own fleet	First spend snapshot, from read-only billing access	MIG slice recommendations from observed concurrency, A100 and H100	Owned hardware against vendor list price, energy basis disclosed

§ 01 The gap

Two bills, neither of them comparable

You bought GPUs and you rent tokens. **The hardware shows up as a capex line nobody amortises properly, the APIs show up as three invoices in three formats, and neither is expressed in a unit you can compare.** So the training cluster sits at thirty percent while a workload that could have run on it goes to a vendor at list price, and nobody finds out until the quarter closes. When the board asks *"what did that workload cost us, and should we have run it ourselves?"* nobody in the building can answer. That gap is what TensorCost closes.

§ 02 The stack

Eight layers, metal to decision

LAYER 01 — MEASURE

What the card is doing

NVML sampling per GPU — utilization, memory, power — labelled idle, data-loading, forward, backward, checkpoint or eval. Plus five inference adapters and an SDK.

Kubernetes, bare metal, SageMaker, Vertex, Azure. Phase labelling is v1, rule-based.

LAYER 02 — PARTITION

MIG, as it really is

Every partition, its profile and memory, what is allocated and what sits idle — then a 1/2/3/7 slice recommendation from observed concurrency, priced monthly.

Read-only: we report MIG layout, never reconfigure it. A100 and H100.

LAYER 03 — COST THE METAL

Your real $/GPU-hour

Purchase price depreciated over the life you set, straight-line or MACRS. Power at your tariff × your PUE. Rack, allocated ops and cost of capital on top.

Your assumptions, not a list price. Change one and every figure downstream moves.

LAYER 04 — ATTRIBUTE

The join

A GPU-hour on metal you own and a token on an API you rent, in one model — resolved to a team, a feature and a day, with the share that was simply wasted.

Workload-level tagging needs the SDK. You also see what stays unattributed.

LAYER 05 — COMPARE

Build against buy

Dollars per call on the card you already own versus the vendor's price. Then the actions: consolidate under-40% cards, rent out spare capacity, retire early when residual beats running cost.

A labelled estimate range, and it says whether energy was measured or nameplate TDP.

LAYER 06 — CAP

A budget that refuses

Per-run spend caps checked at the proxy before the provider is called; runaway workloads caught at 5× their own baseline. Spot interruptions caught on the 2-minute warning so training checkpoints instead of dying.

Caps cover proxied, non-streaming calls. Streamed responses pass through uncapped.

LAYER 07 — EVALUATE

No quality roulette

A three-judge panel scores quality, cost, latency and safety in both answer orders. Accepted swaps apply through the gateway you already run.

Recommendations today. Inline routing is on the roadmap, no committed date.

LAYER 08 — PROVE

A ledger that can say no

Each accepted change measured against your own spend 30 days either side, per team and model. It can come back negative. Hash-chained audit log underneath, FORCE'd RLS across 154 tables.

Before/after spend, not invoice-reconciled. SOC 2 on the roadmap, no auditor, no date.

§ 03 One integration, two buyers

Neither has to sell the other

PLATFORM & ENGINEERING

MIG-level fleet visibility, a real $/GPU-hour built from your own depreciation and power assumptions, and per-workload attribution across every provider — plus a spend cap you can put on a run before it burns the budget. We never touch your request path until you choose to put us there.

CFO & FINANCE

One number for AI spend, chargeback per team, and verified savings on a ledger you can audit. Our own pricing is flat and published — Growth and Scale at a fixed monthly price, Enterprise scoped — so you know the fee before you sign, with no share of savings baked into the bill.

§ 04 What changes

Before and after

	TODAY	WITH TENSORCOST
Owned hardware	A capex line nobody amortises, and a utilization graph	→ A real $/GPU-hour, per MIG slice, against vendor $/call
Attribution	By provider and by month	→ By workload, team, model and day — owned and rented alike
A looping agent	Found on the invoice, next month	→ Capped at the proxy, or flagged within the day
Savings claim	A number nobody can defend to an auditor	→ Before/after on your own spend, and it can read negative
Request path	—	→ Nothing in it until you choose to put us there
Our fee	—	→ Flat and published. No share of your savings

§ 05 Start with a pilot

Read-only, two weeks

2 wks
SHADOW MODE

Our sales pilot, not a separate product tab. Connect **read-only billing-API access** — no production access, no code changes — review the spend in the console, and get a written findings report in two weeks. First snapshot in 48 hours. **If the dollars don't clearly clear our fee, the report says so and we don't push the deal.**

If your AI bill has outgrown the spreadsheet — *let's find the dollars.*

Where we are: pre-seed · product shipped and running in production, fleet side and inference side · three design partners, Xeni under LOI and named with permission · zero paying customers, first paid logo targeted Q4 2026. Savings percentages anywhere in our material are modelled ranges anchored to the published methodology — never customer results.

tensorcost.com
Start a pilot →
Read the methodology →