WSO2CON AFRICA

SEPT 22 – 24, 2026 | NAIROBI, KENYA

AI Cost Attribution & Optimization for LLM Consumption

Anusha Jayasundara

Senior Technical Lead

Adoption is settled. Cost control is not.

AI adoption is past the experimental threshold, and spend is scaling faster than cost governance.

88%

of organizations regularly use AI

↗ up from 78% in 2024

3.2x

growth in enterprise GenAI spend

↗ $11.5B → $37B in one year

$2.59T

forecast AI spending in 2026

↗ +47% YoY

"Enterprise AI budgets are coming under greater scrutiny, with increased focus on usage efficiency, cost control and measurable outcomes."

Gartner, July 2026

Sources: McKinsey State of AI 2025 (regularly use AI in at least one business function); Menlo Ventures 2025; Gartner, May 2026 and July 2026.

One request in. Many billable events.

One request in, one response out. The same surface, very different internals.

TRADITIONAL REST API

Client sends HTTP request
GET /products?id=42

Route + auth check
Validate token, match route handler

Single DB query
SELECT * FROM products WHERE id = 42

Return JSON response
Predictable. Fixed compute. ~$0.000

4 steps · LLM calls: 0 · Cost: ~fractions of a cent
Cost is fixed and deterministic per request

AI AGENT REQUEST

Setup: load prompt + RAG retrieval — $0.034
Embedding call + planning LLM

Reasoning LLM call (18K tokens) — $0.270
Full codebase in context, 73% of total cost

Tool calls (read / write file)
No LLM cost, but it expands the context window for the next call

Safety + verify + retry if needed — $0.066
Each retry = another full LLM call

4 internal events · 3 LLM calls · Cost: $0.37 (~100x more)
Cost is variable, scales with context size & steps

REST cost = compute. AI agent cost = compute + tokens + context size + steps + retries. Request-count throttling misses all of that.

Throttling on the wrong metric

WHAT HAPPENED NEXT

JUN 2025

Pricing redesigned mid-stream

Pro moved from 500 fast requests to $20 of usage billed at API rates. Cost finally entered the throttle, but users were caught off guard.

JUL 2025

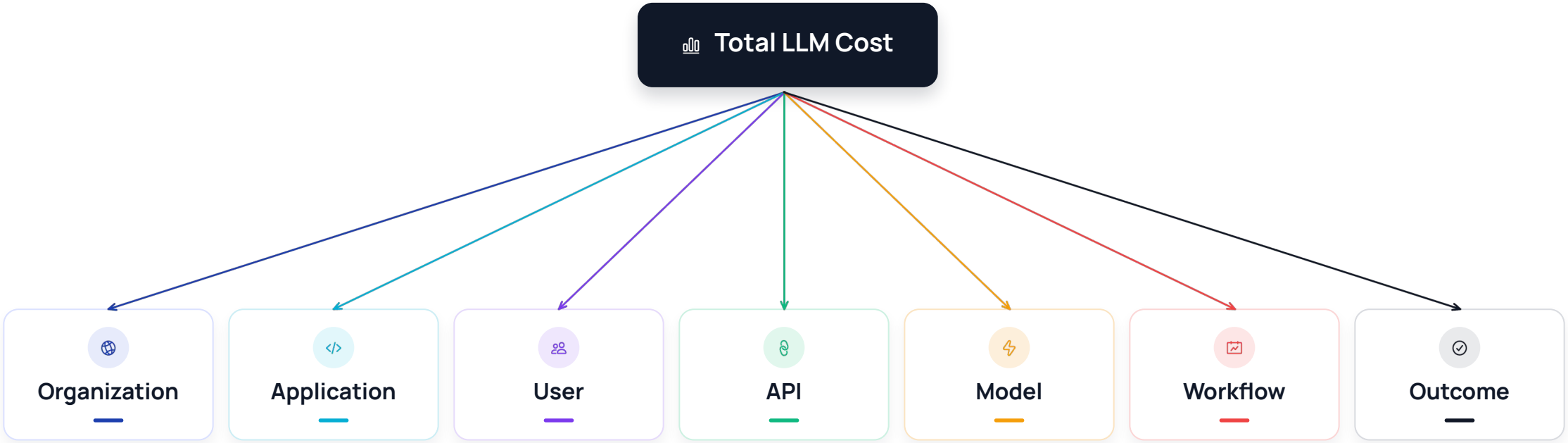
CEO issued public apology

“We work hard to build tools you can trust, and these changes hurt that trust.” — Michael Truell, 4 July 2025.

Throttling by request count is blind to 100x cost variance per request. You cannot price, route, or govern what you cannot attribute.

Cost is shaped by behavior, not just traffic volume.

Who caused the cost, and why?



Cost attribution connects technical usage to business ownership.

Engineering asks which workflow. Product asks which feature. Finance asks which business unit. Platform asks which model.

The attribution record

Emit this on every LLM call and every question on the previous slide becomes answerable later.

IDENTITY

- org_id
- tenant_id
- app_id
- api_id
- subscription_plan
- user_id / agent_id
- cost_center

EXECUTION

- **trace_id**
- parent_span_id
- step_index
- call_type
- model
- provider
- region

CONSUMPTION

- input_tokens
- output_tokens
- **cached_input_tokens**
- **reasoning_tokens**
- context_length
- tool_calls
- retry_count
- cache_hit

OUTCOME

- status
- latency_ms
- result
- **outcome_id**

trace_id is what turns four rows back into one user intent.

Where the cost number actually comes from

Three sources. Three different jobs. Trouble starts when one is asked to do another's.

SOURCE	AVAILABLE	ATTRIBUTABLE	USE IT FOR
Gateway token counting	Real time	Fully, per call	Enforcement, alerts, quotas
Provider usage API	Hours to a day	Per key only	Trend checks
Provider invoice	Monthly	Not at all	Finance truth

Snapshot the unit price at call time

Never recompute last quarter's spend with today's price table. Every price cut silently rewrites your history.

Track drift against the invoice

The gateway number is an estimate. Set a threshold where the gap becomes a bug, and alert on it.

Two planes, one loop

Neither plane gets you there alone. The value is in the loop between them.

ENFORCEMENT PLANE

In the request path

AI Gateway

Decides · Routes · Caches
Limits · Blocks · Budgets

Synchronous, and in the way. The only component that can prevent spend rather than report it.

EVIDENCE PLANE

Behind the request path

Analytics

Attributes · Aggregates · Meters
Alerts · Bills

Not in the way, which is why it can afford to be rich, historical and cross cutting.

HOW THE LOOP CLOSES

Gateway emits enriched events → Analytics attributes and finds the leak → The finding becomes a policy → Gateway enforces it in the request path

The gateway is the only place you can stop a cost. Analytics is the only place you can explain one.

What the gateway can actually enforce

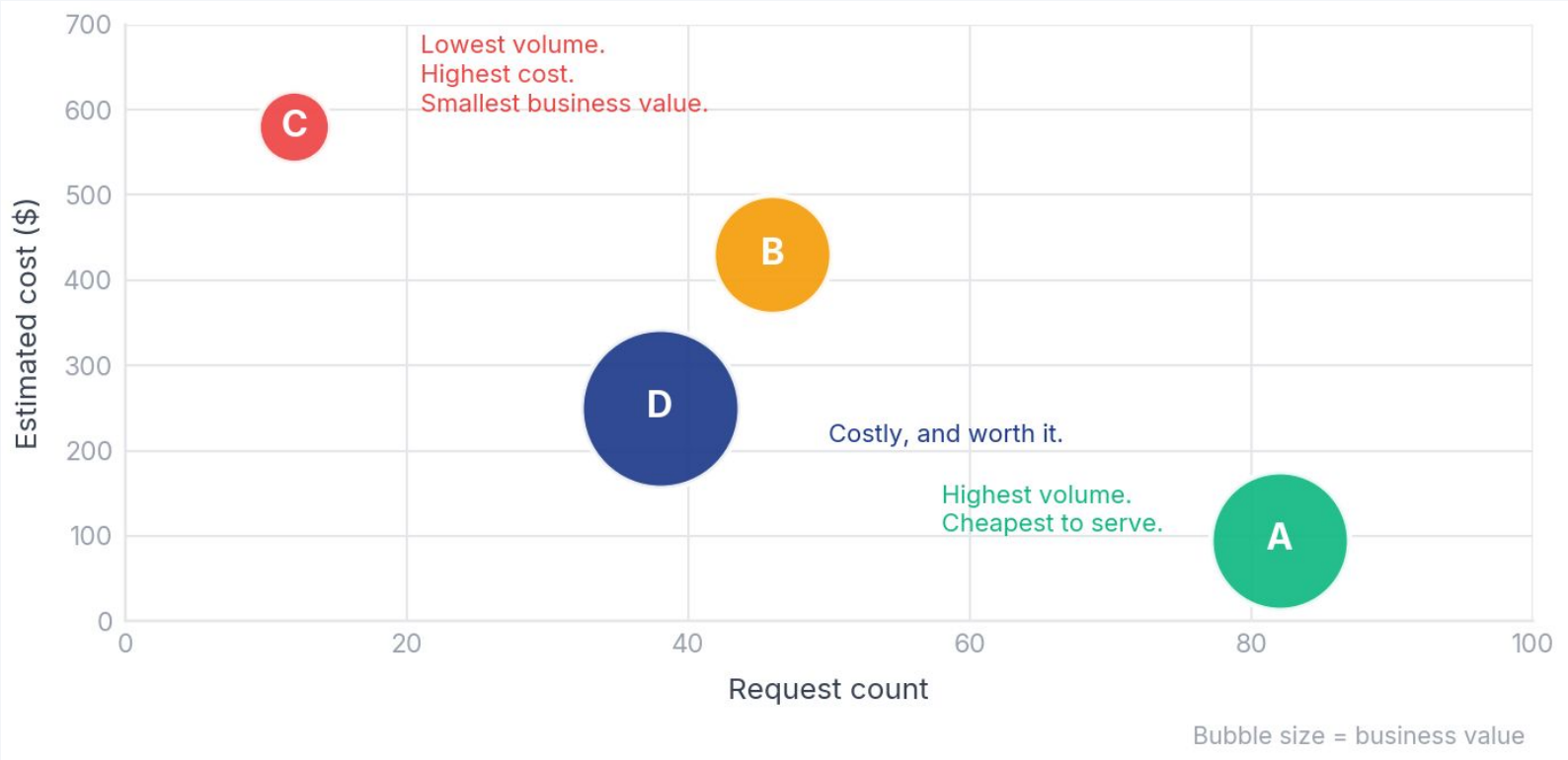
Read the right column first. Every driver on it appeared in the anatomy slide.

AI GATEWAY CONTROL	COST DRIVER IT REMOVES
Token based rate limiting	Runaway context and output length
Multi model routing and failover	Premium model doing a cheap task
Semantic caching	Repeated and near identical prompts
Prompt management and versioning	Prompt bloat duplicated across teams
AI guardrails from Policy Hub	Paying for calls that were going to be rejected
MCP governance	Tool call fan out from agents
Token budgets per team or workload	Unbounded spend with no owner

They are not seven problems. They are the same list, and each one has a control.

What analytics can answer

Request volume points at the wrong customer. Cost per outcome points at the right one.



- Which customer, team or agent drove this month's increase
- Which workflow has the worst cost per successful outcome
- Who is near their token budget right now
- What do we bill or charge back, and can the customer verify it

The same data, metered three ways: **internal chargeback, customer quota, usage based billing.**

One leak, end to end

The shape of every real cost investigation. Nothing here is switched off.

1 · SYMPTOM

LLM spend up 40% month over month while request count stays flat.

2 · ATTRIBUTE

62% of the increase traces to one internal app, one workflow, one model.

3 · DIAGNOSE

A retrieval change pushed average context from 4K to 19K tokens, and timeouts retried as fresh calls.

4 · ENFORCE

Context capped, summarization downgraded to a smaller model, the stable prefix cached, retries budgeted.

5 · VERIFY

Cost per successful outcome drops, volume holds, and no feature is removed.

Steps 2 and 3 are impossible without the record. Steps 4 and 5 are impossible without the gateway.

Reduce waste without blocking value

Every control trades something.

Name the tradeoff before you ship it, or engineering stops believing the programme.

Target: cost per successful outcome.

QUICK WINS

Semantic caching
trades: staleness, tune the threshold

Output token caps
trades: truncation on long answers

Cost alerts per app and tenant
trades: alert fatigue if static

Route cheap steps to smaller models
trades: quality regression, needs evals

Retry budgets
trades: lower success on flaky backends

STRATEGIC INVESTMENTS

Token budgets per team and workflow
trades: every budget needs an owner

Cache aware prompt design
trades: constraints prompt iteration

Price on outcomes, not requests
trades: a billing model change

Prompt and context hygiene
trades: coordination across teams

Agent step and tool call limits
trades: some tasks will not complete

From AI API exposure to AI API economics

1	2	3	4	5
Expose	**Measure**	**Attribute**	**Optimize**	**Operationalize**
managed AI APIs	tokens, models, workflows	users, apps, tenants, outcomes	routing, limits, caching	Dashboards, alerts, Reviews

Visibility first, then control, then continuous optimization.

YOU ARE HERE WHEN

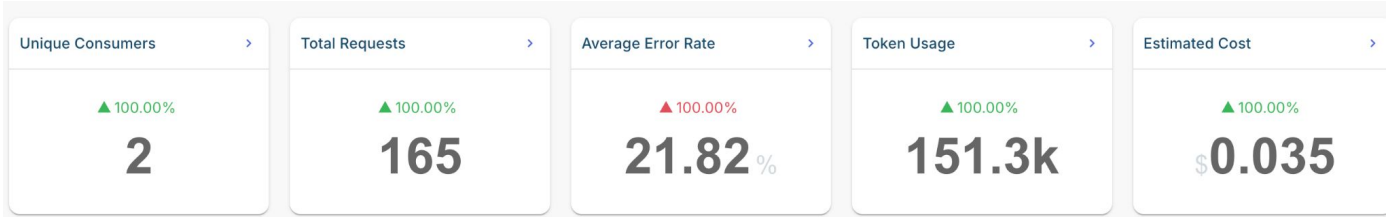
AI traffic goes through a gateway, not scattered SDK keys	Tokens, model, context and steps on every call	Every call carries an owner	Policies change because of evidence	Budget owners, alerts and a monthly review

The view that makes it real

moesif

Same pipeline, three outputs: internal chargeback, customer facing quota, usage based billing.

REQUESTS FLAT. TOKENS AND COST ARE NOT.



COST PER APPLICATION, NOT PER ACCOUNT.

AI Application Details

Application Name	Provider Name	Token Usage	Cost Estimate
(none)	mistralai	99002	0.024398299999999998
(none)	anthropic	8376	0
HotelChatbot	mistralai	42900	0.01063765
DemoApp	mistralai	621	0.0001205
DevApp	mistralai	448	0.000112

Same provider, two applications, roughly a 200x gap in cost estimate. That is the chargeback conversation, and it is unavailable from request counts.

Five things to take with you

1. One user request can be many billable events. Request count does not see them.
2. Emit the attribution record on every call: trace ID, cached tokens, reasoning tokens, outcome.
3. The gateway enforces. Analytics explains. Neither one works alone.
4. Every control trades something. Name the tradeoff before you ship it.
5. Attribution is not a cost project. It is the prerequisite for monetization.

WSO2CON AFRICA

SEPT 22 – 24, 2026 | NAIROBI, KENYA

Thank You!

Anusha Jayasundara

Senior Technical Lead