EMERGE DIGITAL - INCIDENT TEARDOWN

The A$2,000 AI-Spend Runaway

How a bot-driven crawl trap turned one Google Cloud API into 90% of our bill - and the four-layer fix that ended it in a day.

A founder-led teardown of a real incident on our own production ecommerce store. Every number is ours: the SKU-level bill, the one query that found the cause, the fix, and the guardrail stack that now watches every workload we run.

Rami Alcheikh - Founder, Emerge Digital

July 2026 - emergedigital.com - Dubai + Sydney

TL;DR

~A$300	~90%	~971,762	~A$9,000
per day, flat, 24/7 - one API	of the monthly GCP bill	search API calls, mostly bots	run-rate avoided per month

We run a real ecommerce store on Google Cloud - our own brand, our own bill. In July 2026 a routine cost check found one line item, the Retail Search API, burning roughly A$300 every single day, around the clock. No traffic spike. No launch. No human behavior pattern at all - which was exactly the clue.

The cause: search-engine and AI crawlers walking our faceted navigation - every filter combination a unique URL, every URL a paid API call. A crawl trap, priced per query.

The fix took one day once the bill was legible: a four-layer bot-gate that serves humans the paid AI search and serves crawlers a free fallback. The deeper fix is the part most teams skip - a guardrail stack (budgets, SKU-level billing export, a daily anomaly scan) that makes the next runaway a same-day incident instead of a month-end surprise.

This report is the full teardown: anatomy, detection, remediation, and the 10-point self-check we now run on every usage-billed AI workload. If you run Vertex AI, Retail Search, Maps, or any per-call/per-token API on a public surface, this is the failure class to design against.

1. The incident

The store is a production Medusa + Next.js build on Cloud Run, with Google's Retail Search API powering AI commerce search over 5,935 products. It had been live for weeks when the daily bill settled into a suspicious shape: flat. Human traffic breathes - it peaks in the evening, dips overnight, spikes on promotions. Our Retail Search spend did none of that. It drew a straight line at roughly A$300 a day, which annualizes to over A$100,000 for a store still in its growth phase.

The tell: machines don't sleep

A flat 24/7 usage curve on a consumer surface is close to diagnostic on its own. Humans sleep; crawlers don't. When we pulled the caller pattern apart, the search volume tracked bot user-agents and datacenter IP ranges, not sessions or carts. Nearly a million search calls had accumulated - the overwhelming majority never rendered for a human eye.

Why faceted navigation is a paid-API trap

Ecommerce category pages expose filters: brand, size, price band, availability. Each combination is a crawlable URL. A category with 8 filter dimensions explodes into thousands of unique URLs, and a crawler that finds them will dutifully visit every one. When each of those page loads fires a paid search-API call, your SEO surface becomes a metered faucet that bots hold open. Nothing is 'broken' - every system does exactly what it was configured to do. That is what makes this failure class so quiet.

2. Detection: the one query that found it

The single most valuable FinOps control we enabled is also the least glamorous: the billing export to

BigQuery. Console dashboards aggregate; the export itemizes. One query - group cost by service and SKU, order descending, compare day over day - turned a mystery bill into a named culprit in minutes: retail.googleapis.com, ':search' SKU, ~90% of total spend.

- **The rule we took from it: If you cannot name your top five SKUs from yesterday, you do not have cost visibility - you have cost archaeology.**

We now run that query automatically every morning as an anomaly scan: any SKU whose day-over-day cost jumps beyond threshold pages the owner the same day. The scan has run daily since 10 July 2026.

3. The fix: four layers, one day

We disabled the API first - spend went to approximately zero within the hour, confirming the diagnosis - then rebuilt the front door before re-enabling it:

- **Layer 1 - Facet canonicalization: collapse filter permutations into canonical URLs so the crawlable surface shrinks by orders of magnitude.**
- **Layer 2 - robots.txt: declare the faceted space off-limits to compliant crawlers; most major bots respect it, and the ones that don't get caught below.**
- **Layer 3 - UA/behavior gate: a backend search-guard classifies the caller before the paid call is made. Humans get the paid AI search; crawlers get the free built-in engine. The page still renders for everyone - only the metered call is gated.**
- **Layer 4 - Caching: repeated identical queries answer from cache instead of re-billing the API.**

Result: the API went back on, humans kept the AI-quality search experience, and the line item dropped from ~A$300/day to effectively noise. Verified in the billing export, not the dashboard.

4. The guardrail stack that stays on

The incident fix is worth one bill cycle. The stack below is what changes the trajectory - it is now standing infrastructure on every project we run, and it is what an AI Spend Audit installs:

- **Budgets that actually page: with 50% / 90% / 100% and forecast alerts, sized per workload - ours are deliberately tight (a A$200 monthly guardrail on the account that had just burned A$300 a day).**
- **Billing export to BigQuery: SKU-level, queryable, the substrate for everything else.**
- **Daily anomaly scan: day-over-day by service and SKU; a runaway becomes a same-day incident.**
- **Pre-launch gates: quotas, bot-gating and caching configured BEFORE a usage-billed API goes live on any public surface - launch requirement, not retrofit.**
- **Hygiene sweeps: scale-to-zero on idle services, right-sized CPU, storage lifecycle policies, registry cleanup - the boring 20% that pays for the audit by itself.**

5. Why AI spend is different

Classic cloud waste is idle capacity: an oversized VM burns a predictable amount whether or not anyone uses it. Usage-billed AI APIs invert that - cost scales with invocations, and on a public surface you do not control who invokes. Search APIs, vision APIs, per-token model calls, geocoding: each is a metered endpoint that third parties (crawlers, scrapers, abusive clients, other people's AI agents) can drive without your consent.

The FinOps Foundation's State of FinOps 2026 reports that 98% of practitioners now manage AI spend - the

category has arrived faster than the tooling habits around it. Agentic patterns raise the stakes again: agents fire API calls without a human in the loop and retry on failure. A retry loop against a paid endpoint is a runaway with intent.

The discipline that follows: treat every usage-billed endpoint as a security surface AND a cost surface. Gate who can invoke it, cap how fast, observe it daily, and make someone own the number.

6. The 10-point self-check

- **1. Can you name yesterday's top 5 cost SKUs (not services - SKUs)?**
- **2. Is your billing export flowing to a queryable warehouse?**
- **3. Does any budget alert reach a human who can act, before 100%?**
- **4. Do you know which of your APIs are usage-billed and publicly reachable?**
- **5. Would a flat 24/7 usage curve on a consumer surface trigger anyone?**
- **6. Are crawlers gated away from paid API calls (not just robots.txt)?**
- **7. Are quotas set on every metered API - at the number you expect, not the default?**
- **8. Do repeated identical requests answer from cache?**
- **9. Does anything scan cost day-over-day automatically?**
- **10. If an AI agent you run entered a retry loop tonight, what would stop it - and when would you know?**

Score honestly. Seven or more confident yeses and you likely need nothing from us. Below that, the audit exists.

7. If you want this done for you

The AI Spend Guardrail Audit is a fixed-fee engagement (quoted on scope) that installs everything in section 4 on your Google Cloud estate: SKU-level visibility, budgets that page, the daily anomaly scan, pre-launch gates on every usage-billed API, and a hygiene sweep with the quick wins actioned. Delivered from our own playbook - the same stack that watches our production workloads today.

AI Spend Watch is the optional monthly layer on top: the daily scan reviewed by a human, a monthly report with trend and forecast, and a flagged-anomaly channel. A report subscription, deliberately not an on-call SLA.

Book a 20-minute scoping call: cal.com/rami-alcheikh/strategy-call - or reply to the email that brought you here.

About Emerge Digital - a Dubai + Sydney digital consultancy building governed AI systems on Google Cloud and Cloudflare: enterprise CX assistants, data platforms, and the cost-governance stack in this report. Human-reviewed AI delivery; measured claims only. Listed in the Google Cloud partner directory.