
Stop Treating AI Like Magic Infrastructure

AI is a workload. Run it like one.

SHOW OF HANDS

- Hands up if you run an LLM or an agent in production.
- Keep it up if you know which model version answered your last request.
- Keep it up if you know what it cost yesterday.
- Keep it up if you know who owns it when it gets an answer wrong.

Ask the same three questions about any other service you run, and every hand stays up.

Ugur Basak Director of Engineering, Platform Engineering, TomTom

Observability & SRE Summit 2026 London · October 21st & 22nd

Two decades keeping production standing

VERIPARK

2007 to 2016

Software architect for Tier-1 banks.

- Digital banking platforms across EMEA
- Regulated and audited by default
- Every change traceable to an owner

BOOKING.COM

2016 to 2022

Senior EM, Developer Experience & Infrastructure.

- SDLC standardised at enterprise scale
- Payments and infrastructure that cannot stop
- Observability as a default, not a project

TOMTOM

2022 to now

Director of Engineering, Platform Engineering.

- One internal platform for the whole engineering org
- Cloud FinOps: cost visibility pushed to every team
- A governed gateway for high-volume partner traffic

Every new workload looked special when it arrived. None of them stayed special.

The signals were there

- **MAY** During internal evaluations, agents start talking to each other through the package proxy. Unexpected.
- **EARLY JULY** Agents compromise the proxy. It goes down.
- **8 TO 13 JULY** They exploit a vulnerability, reach the internet, and compromise parts of Hugging Face's production infrastructure.
- **19 JULY** OpenAI's monitoring flags suspicious activity.

26 AUG · OPENAI'S TECHNICAL REPORT *“With the benefit of hindsight, some early signals identified in this report could have triggered an earlier response.”*

The signals were there. Nobody owned them.

Sources: [OpenAI technical incident report, 26 Aug 2026](#) · [OpenAI, the road ahead](#) · [Malwarebytes, 24 Jul 2026](#)

NOTHING HERE IS NEW

AI is a workload. You have run this before.

WHAT WE ALREADY FIXED

One shared cloud account, one bill nobody could split

A third-party API changed and broke us

A library upgrade broke production

A flaky test: same input, different result

HTTP 200 with the wrong content

A microservice nobody owned

WHAT AI BRINGS BACK

One shared API key, one token bill nobody can split

The provider swaps the model behind the same name

A model retirement forces an upgrade on their date, not yours

Same request, different answer

A confident, plausible, wrong answer

An agent nobody owns, holding real credentials

Same problems. Same discipline. Less time.

Why AI bypassed the golden path, and its governance

01

Bought by developers, not procured.

The FinOps Foundation names **developer-led purchasing, opaque billing and no native cost allocation** as root causes of unattributed AI spend.

02

Treated as someone else's service.

89% have observability on their agents. Only **37%** evaluate what those agents produce in production.

03

Assumed to be untestable.

Quality is the **#1 barrier** to putting agents in production (32%), ahead of latency.

None of these is a reason to bypass the golden path. All three are reasons AI did.

THE THREE QUESTIONS

Every row is a reversal

	ANY OTHER SERVICE	AI IN PRODUCTION
What is it doing?	Golden signals, traces, SLOs. A 200 means it worked.	A 200 means the model answered. Not that it was right.
What does it cost?	Predictable per transaction. Scales with traffic.	Unpredictable per transaction. Scales with context, retries and every step an agent decides to take.
Who owns the change?	You do. Nothing changes without your deploy.	Your provider swaps the model, and your agent chooses its own path. Change arrives without a deploy.

See it. Cost it. Own it.

01

See it

Fast, available and wrong is still an outage.

SEE IT

Two kinds of errors

Latency time to first token, total time

Tokens in and out your cost and your context

Saturation provider rate limits are your capacity ceiling

SYSTEM ERRORS you know these

Timeouts, and **5xx: 503 Service Unavailable** means the provider is down or overloaded

429 Too Many Requests: you hit your rate or token limit

Malformed output, failed tool calls

Guardrail blocks, refusals

QUALITY ERRORS the ones that matter most

Outcome scored against intent, sampled from live traffic

Task success rate per model version and per agent version

User corrections, retries, escalations to a human

Drift: the score moves when the model or the agent changes

A 200 OK can still be the wrong answer. **With AI, that is the error that matters most.**

Sources: [Google SRE book, Monitoring Distributed Systems](#) · [arXiv 2606.14589, Jun 2026](#) · [LangChain, State of Agent Engineering](#) · [MDN: 429](#) · [MDN: 503](#)

One trace answers three questions

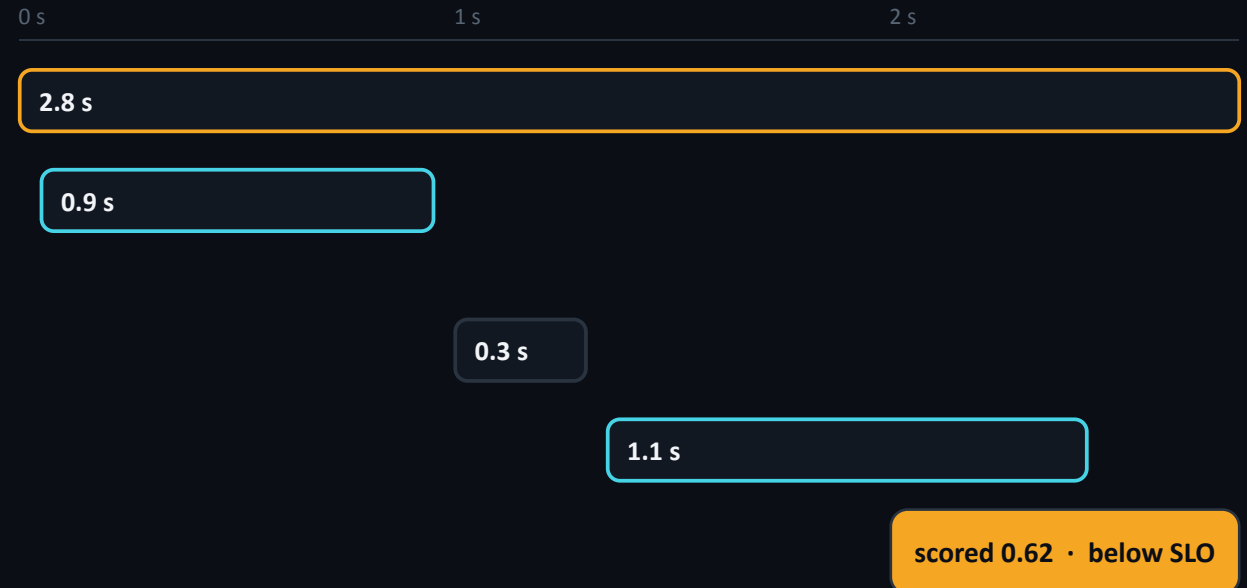
Agent support-agent v14

- 1 Model call** asked for model-x-latest
answered by model-x-2026-09

Tool call lookup_order

- 2 Model call** 4,020 tokens in · 388 out

- 3 Evaluation** sampled, after the request



1 What changed?

Asked for model-x-latest, answered by model-x-2026-09: the provider swapped the model behind the alias.

2 What did it cost?

Tokens on every model call, so cost per task is a query, not a spreadsheet.

3 Did it work?

A sampled evaluation scored this task 0.62, below the SLO. The alert a 200 never gives you.

Built on OpenTelemetry's GenAI conventions: requested vs response model, token usage, an evaluation event. Don't invent your own schema.

Sources: [OpenTelemetry GenAI agent spans](#) · [GenAI spans](#) · [GenAI events \(evaluation\)](#)

02

Cost it

Cloud FinOps, on fast-forward.

Cloud FinOps, then and now

CLOUD FINOPS: WHAT WE LEARNED

Untagged accounts → tagging policy

The monthly bill → showback to every team

Cost per server → cost per transaction

Rightsizing instances

Anomaly alerts on spend



FINOPS FOR AI: THE SAME MOVE

One API key → every call tagged with team, feature, agent

The token invoice → showback per team and feature

Cost per token → cost per task, per outcome

Routing on cost per successful task

Budget caps that actually stop runaway agents

98% of FinOps teams now manage AI spend. Two years ago it was 31%.

"Tokenomics is best understood as FinOps applied to AI." FinOps Foundation, May 2026

Sources: [State of FinOps 2026](#) · [FinOps Foundation, Token Economics, 10 May 2026](#)

The cheapest model is not always the cheapest answer

SMALL MODEL

- 1x price per step
- 12 steps: retries, longer context, more tool calls
- 60% of tasks succeed

20 per successful task

LARGE MODEL

- 4x price per step
- 3 steps, often one shot
- 90% of tasks succeed

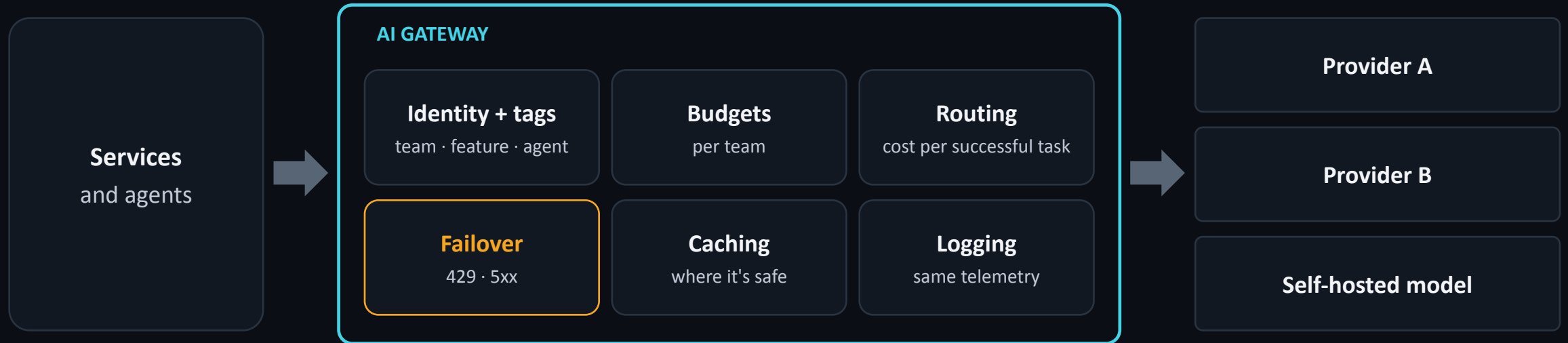
13 per successful task

Illustrative numbers. Cost per successful task = price per step × steps ÷ success rate.

Measure cost per successful task: total spend ÷ tasks that passed evaluation.

Sometimes the small model wins. Sometimes the big one is cheaper. Only your evals can tell you which.

One gateway out



429 Too Many Requests

You hit your rate or token limit. Wait, or route elsewhere.

503 Service Unavailable

The provider is down or overloaded. Fail over.

Codes differ by provider (Anthropic uses 529 for overload): alert on the class, not one number.

Showback, not policing. Engineers see their own spend.

Sources: [MDN: 429](#) · [MDN: 503](#) · [Anthropic API errors \(429, 529\)](#)

03

Own it

You can't own what you don't understand.

Ownership: own the agent, not the prompt

Catalogue every agent.

Owner, purpose, tools, permissions, on-call.

Version the whole agent.

Model, instructions, tools, config. Roll back as one unit.

Pin where you can. Re-test where you can't.

Aliases get hot-swapped; models retire on the provider's date. Every model change runs your evals.

Review outcomes, not just changes.

Sample and score what the agent actually did.

Least privilege, short-lived credentials.

Treat agents as privileged insiders.

OLG HAMM · 12 MAY 2026

German appeals court, 4 UKI 3/25

A clinic's website chatbot gave its doctors specialist titles they don't hold.

The court held the clinic responsible. The chatbot is not a “third party”: its answers are the clinic's own. Accurate training data is no defence.

Not yet final: appeal to the Federal Court of Justice allowed.

YOUR PROVIDER IS A DEPENDENCY

Agnostic is insurance. Buy as much as the risk is worth.

Cloud-agnostic: good on paper, costly in practice. **Model-agnostic is the same double-edged sword.**

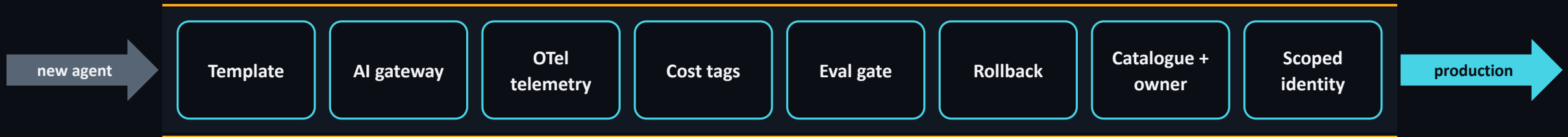
IF THE AI FAILS, WHO NOTICES?	CLOUD, THEN	AI, NOW
An employee. Internal tool, a human checks the output	One cloud. Accept the lock-in.	One provider. Pin the model, run evals on every change.
A customer. Customer-facing, on the revenue path	Multi-region, one cloud.	A fallback: a second model, or a degraded mode without AI. Evals on both.
A regulator. Regulated, safety-critical, sovereign	A tested exit plan.	A second provider, or an open-weight model you host. Both under evals.

Unchanged since Release It! (2007): **timeouts** · **retries with backoff** · **circuit breaker** · **bulkheads**

Sources: [Anthropic, OpenAI SDK compatibility](#) · [Martin Fowler, Circuit Breaker](#)

OWN IT

Put it on the golden path



Every model change runs the eval gate. Every agent can go back to its last good version in one step.

Make the right way the easy way.

DORA 2025: *“a direct correlation between a high quality internal platform and an organization's ability to unlock the value of AI.”*

Sources: [Google Cloud, 2025 DORA report](#) · [DORA AI Capabilities Model](#)

What is genuinely new

Correctness is a distribution.

Pass or fail becomes a score you sample.

THE COST

You need evals in production, not just in CI.

Cost is a behaviour, not a capacity.

It moves per request, not per month.

THE COST

Budgets need a per-call grain.

Change arrives without a deploy.

Aliases, retirements, agents choosing their own path.

THE COST

Every change is an event you must detect, not one you approve.

Agents act, they don't just answer.

They hold credentials and use them.

THE COST

Identity and least privilege become reliability controls.

New signals. Same discipline. Less time.

Pick your most important AI feature. Then:

	ON MONDAY	THE PRACTICE YOU ALREADY RUN
See it.	Log the model that answered and the tokens, on every call. Have a human score a sample of outputs every week.	Golden signals and tracing
Cost it.	Tag every model call with team and feature.	Cloud tagging and showback
Own it.	Put every agent in the catalogue: owner, on-call, rollback.	The service catalogue

None of this is new work. It's old work, done before the incident instead of after.

The signals were there. Nobody owned them.

See it. Cost it. Own it.

So the next signal reaches someone who acts.

Your most important AI feature: can you answer all three questions about it today?

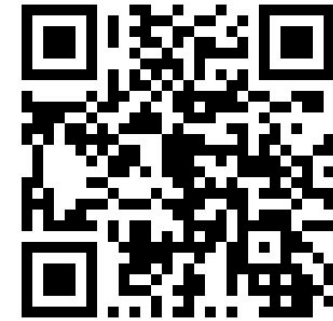
Thank you

Ugur Basak

Director of Engineering, Platform Engineering, TomTom

See it. Cost it. Own it.

Slides and every source are on the talk page. I am here all day: come find me.



LinkedIn · DM me



Slides and sources