

Your AI Supply Chain Is a Production Dependency. Audit It Like One.

Julien Laurenceau | Published 2026-09-12 | Updated 2026-09-30

<https://pepitedata.com/ai-production-control-supply-chain/>

PepiteData | LinkedIn: Julien Laurenceau | X: @Julien_Lau



Two movements are happening at the same time, and they meet in the same place.

The first: AI systems are being given production responsibilities. Agents read tickets, call tools, edit code, open pull requests, and in some shops touch live infrastructure. The second: production data is flowing through AI. Pipelines call models to clean, classify, extract and enrich records, in batch and in streaming, at data platform scale.

Put together, a critical part of your production path is now a component you do not own, cannot inspect at will, and most likely have not audited. The model, the router in front of it, the provider behind it, the terms attached to the route, and the behavior of all three over time are part of your reliability and security surface.

September 2026 was the month this stopped being theoretical.

From assistant to orchestrator

Enterprise AI has moved from assistant to orchestrator: the pattern now is agentic, with the model given tools, credentials and goals. For three years, the AI in an enterprise was a chat window with a knowledge cutoff. Anthropic's September threat intelligence report describes the same shift on the offensive side, in its own words: "AI's role in cyber operations has become increasingly autonomous."

Operations use multi-agent frameworks that execute reconnaissance, exploitation and data exfiltration, while humans set targets and review results. One case tracked as GTG-20006, which the report links to

public reporting on Midnight Blizzard, targeted more than 20 organizations and rebuilt its malware autonomously each time a security product detected it.

"Sophisticated attacks no longer require sophisticated attackers."

Anthropic, Detecting and countering misuse of AI: September 2026

The defensive economy moved. A new detection signature used to buy defenders time. When the tool rebuilds itself on detection, that time is gone.

The agent that opens your tickets has the same shape as the agent that runs an intrusion: tools, credentials, memory, persistence. The difference is governance, and governance is what an audit produces.

The pipeline now calls the model

LLM calls are moving into the data path itself. The second movement is quieter and larger. Instead of an analyst reading unstructured data, the pipeline does: classify support tickets, extract entities from contracts, normalize addresses, clean product catalogs, flag anomalies in logs, enrich records with summaries.

This is shipping in the platforms teams already use. Snowflake offers [Cortex AI Functions](#) to run classification, extraction, summarization and sentiment in SQL. BigQuery ships [AI.GENERATE](#). Researchers have formalized the pattern as [semantic operators](#), LLM-based map, filter and reduce operations with an optimizer that batches calls, applies model cascades and plans the whole pipeline for cost. [LOTUS](#) is the open-source implementation. A 2026 [practitioner guide from Data Engineer Academy](#) lists the practices that matter: keep raw source data, prompts, model outputs and version history separate and traceable, validate output before load, and define evaluation rules and fallback logic before production.

"The safest production setup keeps raw source data, prompts, model outputs, and version history separate and traceable."

Data Engineer Academy, "Integrating LLMs into ELT Pipelines"

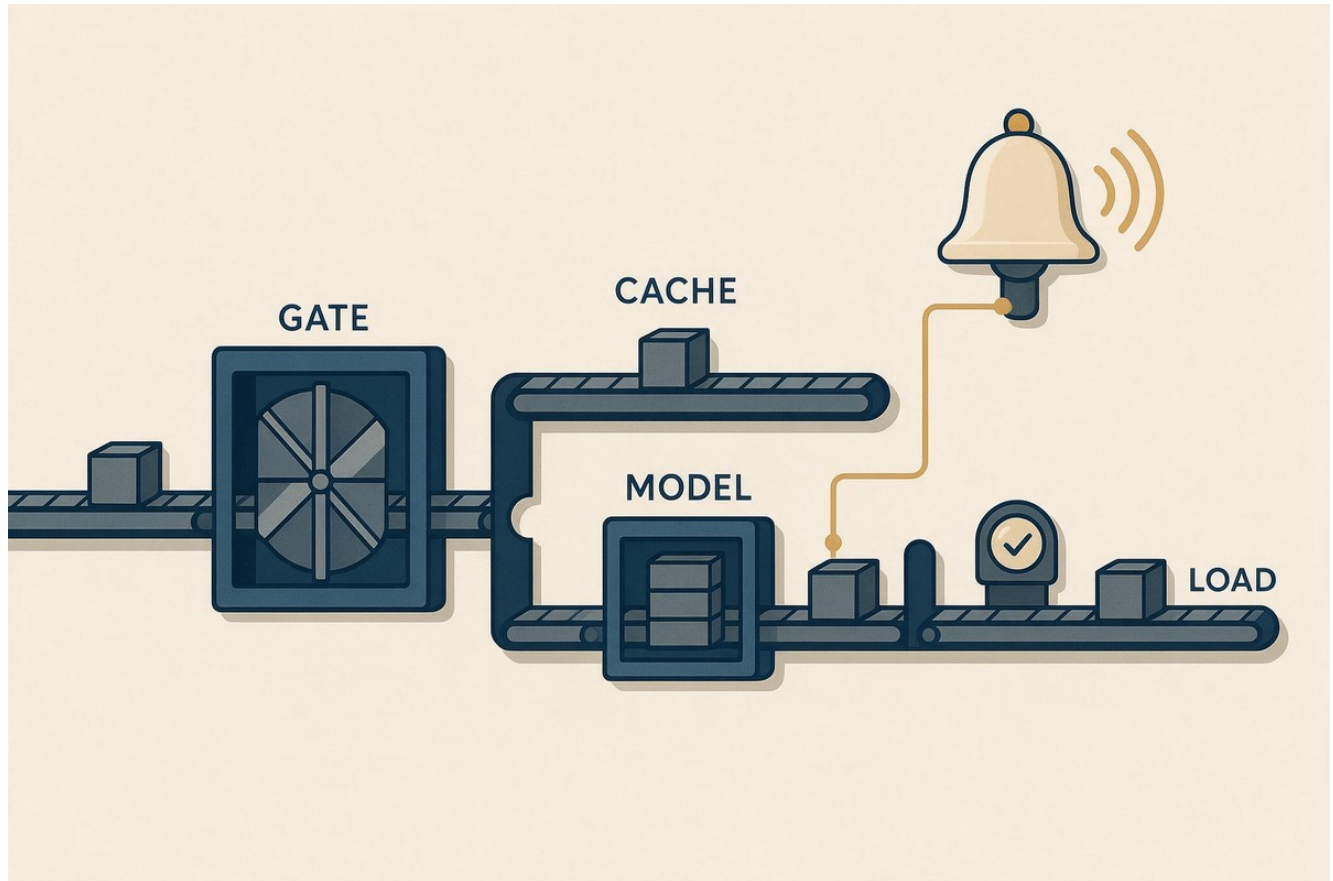
The arithmetic is what changes. In a classic ETL, adding a step changes compute and time. When the step is a model call, it changes cost per record, concurrency and determinism. A million-record backfill becomes a token budget. A streaming job becomes a rate of model calls per second with a latency budget attached to each one. A data quality rule becomes a prompt with a refusal rate. And because the output is not deterministic, reprocessing the same partition can produce a different result.

If you are designing that pipeline, these are the questions to settle before writing code:

- **Records to calls to tokens.** Records per second become model calls per second, then input and output tokens per record. The unit you buy is tokens, not rows.
- **Concurrency.** A streaming call holds a connection for its whole duration. The number that sizes the system is simultaneous calls, not daily volume.
- **Cost per million records.** This is the unit of account, not cost per call. It is also the number that survives a provider price change.
- **Cache, batching, deduplication, cascades.** What can avoid a call entirely: repeated values, shared context, a small model first with escalation only when needed.
- **Pipeline shape.** Per record, micro-batch or window; synchronous or queued; API or self-hosted. Each choice moves the bottleneck.

- **Replay.** What happens to a partition when the model changes, the provider fails, or a prompt is fixed after a week of production.
- **Observability.** Cost per record, refusal rate, schema violation rate, a drift alarm, and the resolved model version for every processed row.

None of these are model questions. They are platform questions, and they are the kind of thing I size for a living.



The LLM stage belongs inside the pipeline's controls, not beside them.

Espionage and cyber, now agentic

Espionage and cyber have gone agentic: model capabilities are extracted at industrial scale through covert campaigns, and attack pipelines run credential harvesting and autonomous vulnerability research on their own. The driver behind the urgency is documented in [Anthropic's report](#), published September 10 and covering December 2025 to August 2026. Two threads matter for anyone giving AI a role in production.

First, illicit distillation at industrial scale, which the report defines as:

"an industrial-scale, covert campaign to extract a model's capabilities and replicate them in another model without authorization"

Anthropic, Detecting and countering misuse of AI: September 2026

Legitimate distillation, where a smaller model is trained on the outputs of a larger one, is standard practice, and I covered [how big models teach small models](#). What Anthropic describes is the unauthorized, industrial-scale version of the same technique.

Anthropic alleges that seven China-based labs ran campaigns to extract its models' reasoning traces. It says Alibaba's campaign peaked near 3 million exchanges per day from more than 3,500 fraudulent accounts, with over 151 million exchanges observed between May and July, and that the harvested transcripts helped train its Qwen models. It says Moonshot relayed almost 300,000 customer requests to Claude in a ten-day window through a network of 5,380 fraudulent accounts, mostly in Singapore and Japan, and saved the exchanges for training. It says DeepSeek rerouted users of coding harnesses like Claude Code and OpenCode, 12.1 million exchanges in 14 days, and that Zhipu rotated 273 fraudulent accounts in ten days for its own extraction pipeline. In several cases, the relayed data included sensitive customer material: surveillance data on an individual, live credentials for a Russian government database, internal documents from a technology company. These are Anthropic's allegations from its own telemetry, and the report does not include responses from the named companies.

The accessible summary is the [International Cyber Digest thread](#). The [r/LocalLLaMA discussion](#) drew the practical conclusion faster than most vendors will: providers log exchanges with detail, and any relay in the path can end up doing the same.

Second, cyber operations. A cluster tied to ShinyHunters affiliates ran a credential pipeline across 10 AWS EC2 workers, downloaded and decompiled 1.8 million Android apps looking for secrets, and dumped more than 2,100 Azure AD token sets across 40 corporate tenants in about 34 hours. Another group ran autonomous vulnerability research against security appliances and produced working exploits. The report's framing is the one to keep: the AI supply chain became a target, loot and attack compute at the same time. A stolen API key is worth money, runs the attacker's workloads at your expense, and carries your name in the logs.

"The operators treated the AI supply chain itself as both a target and a resource."

Anthropic, Detecting and countering misuse of AI: September 2026

One detail I keep coming back to: attribution itself has become unstable. While the report circulated, an unverified claim spread that a research team had disappeared after leaking national secrets to a model. The team publicly denied it, and the person who posted the rumor said so himself. The claim and the correction both traveled at machine speed. Decisions on this terrain need first-party evidence, not screenshots.

The router in the middle

Between your application and the model there is often a router: a service that accepts one API format, picks a provider, fails over, and returns the answer. It is convenient, it is now common, and it has full plaintext access to every request and response.

"These routers operate as application-layer proxies with full plaintext access to every in-flight JSON payload, yet no provider enforces cryptographic integrity between client and upstream model."

Shou et al., "Your Agent Is Mine", ACM CCS 2026

A paper presented at ACM CCS in October 2026, "[Your Agent Is Mine](#)", measured what that access means. The authors bought 28 paid routers and collected 400 free ones. One paid router and eight free routers were actively injecting malicious code into returned tool calls. Two used adaptive evasion, delivering payloads only after a warm-up or only when the client ran in an autonomous mode. Seventeen routers touched the researchers' AWS canary credentials, and one drained ETH from a private key. Weakly configured decoys, fed with a leaked key, processed 2.1 billion tokens, exposed 99 credentials across 440 Codex sessions, and 401 of those sessions were already running with tool execution auto-approved. One of the authors had [disclosed the early findings in April](#), with larger claims: 26 routers injecting, a client wallet drained, and the assertion that hundreds of hosts could be taken over within hours. The paper keeps the measured subset. The strongest deployable defense it evaluates is a

fail-closed policy gate at the tool-call boundary: check the action before it executes, not the content after.

Then there is the data deal. In September, a security researcher [claimed to have bought a 6 TB dataset](#) from a router: prompt logs containing SSH keys, VPN configurations, cloud keys and GitLab tokens. Treat that as a claim, not a finding. But the economics it describes are visible in the free tiers. When I opened a free [OrcaRouter](#) account to check, the free GLM 5.3 Flash route comes without zero data retention: working without ZDR is the condition of the free tier, and you can verify it yourself in the options of a free account. I wrote earlier about [testing a free model before production](#): the same evaluation gate applies here, and the data terms come first. The pattern is old. When the product is free, the data is the product.

Even where a router does offer zero retention, [its own documentation](#) is explicit that the guarantee covers the router layer: the request still reaches the upstream provider under that provider's retention policy. And free access attracts abuse, which is why the same service now requires a linked GitHub account at least 30 days old before granting it. None of this makes routers illegitimate. It makes them a dependency with a data path, a threat model and a price that is not always on the invoice. It is the same question object storage operators faced when [MinIO put its open source project into maintenance mode](#).

Your sessions are training data

Your sessions are training data, a cleaner lesson than the surrounding drama. The second supply chain story from September involves mathematics: Tristan Buckmaster and Levent Alpöge worked for most of a year on finite-time blowup problems around the Navier-Stokes Millennium Prize, using several models, and put [every draft of the project](#) into Codex. On September 8, OpenAI [announced](#) that an internal next-generation model had produced a proof of the forced Navier-Stokes problem.

Buckmaster's public statement describes what followed: a call with an OpenAI researcher, proposals to coordinate releases, a request to drop his co-author because that co-author works at Anthropic, and two lines he quotes: "Why would you ruin your career?" and "If you don't want me to be nice, then I don't have to be nice." OpenAI's first response called the allegations false and inflammatory and said no specific user data was accessed; the researcher later called the career remark ill-chosen and retracted it, according to [the write-up on dev.to](#). [Haseeb Qureshi's summary](#) is the fast version of the story.

"I am not accusing anyone of anything. I am stating what I was told, when, and what was proposed to me."

Tristan Buckmaster, public statement, September 8, 2026

The company's [public statement](#) on the result is careful in a way worth reading: "While unlikely, we cannot rule out that de-identified data derived from their usage of our products helped improve our models." Access is not retrieval, retrieval is not visibility, and none of them answer the training question. That sentence is the honest limit of data provenance at every AI provider, not just one.

For a company, the lesson is not about mathematics. You do not have a Millennium Prize problem in your Codex history; you have an unreleased product, an architecture decision, a migration plan and incident data. If your sessions are training data by default, and your provider cannot rule out that de-identified derivatives entered a future model, then your unpublished work has a provenance problem even when everyone behaves in good faith. The [r/academia thread](#) put it plainly: do not put research you need to own into a system whose data flow you cannot trace.

Same model name, different model

A stable model name is not an identity: different weights, quantizations and refusal behavior can hide behind it, depending on who serves it. The last layer is the one that will bite operations teams first,

because it does not require an attacker. It only requires time.

Model providers update models behind stable endpoint names. Weights change, safety filters change, system prompts change, and the changelog does not. [DriftBench](#) runs a fixed battery of tests daily and charts the results. Its own documentation states the premise plainly:

"Model providers routinely update weights, safety filters, and system prompts without changelog entries."

DriftBench

It raises an alert when any score deviates more than 15 points from the seven-day rolling average. This is not a curiosity for AI products. It is the mechanical reason a launch benchmark has a shelf life.

For agents, drift means behavior changes without a deploy: the prompt that worked stops working, tool-call formatting regresses, refusals appear or disappear. For pipelines, it is worse, because the ELT contract assumes determinism. Same input, same output is the foundation every data quality rule is built on. A model call at temperature zero reduces variance; it does not create determinism, and a provider-side update can shift the output distribution of a stage the pipeline trusts.

Variants make it more complicated. [OrcaRouter shipped an uncensored build of GLM 5.3 Flash](#) three days after the official weights, with refusal rates on JailbreakBench dropping from 0.930 to 0.120 while MMLU barely moved. Those numbers are self-reported, but the pattern is the point. The same model name can describe different weights, different quantizations and different refusal behavior depending on who serves it. I covered the security-tooling side of that split in [After Coldcard, AI found 1,029 bugs](#). Add the cross-session replay techniques documented in the Anthropic report, where a model is manipulated into reconstructing its own reasoning traces, and the conclusion is hard to avoid: model output is not a stable contract.

What production requires is not a better benchmark. It is version discipline: log the resolved model for every request and every processed row, pin where the provider allows it, re-run a fixed evaluation on a schedule, alarm on behavioral change, and treat a model upgrade as a migration with a replay plan.



Same label, different contents: the model name is not an identity.

What an AI dependency review checks

An AI dependency review checks inventory and provenance, the data path and its terms, keys and access, tool-call control, output contracts, behavior verification, failure paths and sizing. This is the checklist I apply before an AI gets a production responsibility. It is cross-layer by design, from the application to the provider contract.

1. **Inventory and provenance.** Every model, router and provider in the production path, with the resolved version per request. If you cannot answer which model processed a given record, nothing downstream is auditable.
2. **Data path.** What leaves, to whom, under which terms, at every hop. Zero retention end to end, not just at the router. Training terms for every provider, including the ones a router adds without asking.
3. **Keys and access.** Least privilege, rotation, exposure scanning. The report's app-mining pipeline is a reminder that secrets leak through mobile apps, CI logs and container images.
4. **Tool-call control.** A fail-closed gate before execution for anything with side effects, an audit log, and explicit approval for destructive actions. Content filtering does not replace action policy.
5. **Output contracts.** Schemas, confidence thresholds, validation before load, refusal handling. Model output is an input to the next stage, not the source of truth.
6. **Behavior verification.** A fixed holdout with deterministic scoring, run on a schedule, with a canary before rollout. Alarms on drift, not dashboards nobody watches.
7. **Failure paths.** A tested fallback route, replay and idempotence, cost per record monitoring, and a named exit option for every critical dependency.
8. **Sizing.** The pipeline arithmetic from earlier: records to calls to tokens, concurrency, cost per million records, and what caching and batching remove from the bill.

Most of these are the same questions I ask about storage, databases and schedulers. A model is a component with a vendor, a version, a failure mode and a bill. It deserves the same severity of review.

Where this fits in my work

The AI supply chain is one more layer in the map I audit: data and AI infrastructure, from source code to disks. It is where I currently see the widest gap between what teams assume and what they can prove.

- If you are designing or sizing a pipeline that calls models, **Capacity Planning and Platform Sizing** is the engagement: workload to calls to tokens, concurrency, cost per million records, batch and streaming design.
- If the question spans systems, a **Data Platform Audit** covers performance, cost and reliability across the platform, including the AI dependencies now inside it.
- For one decision, such as a router, a provider or a pinning strategy, a **Flash Architecture Review** is the bounded format.
- If the risk is continuity, a **Resilience and Disaster Recovery Assessment** tests fallback, replay and exit paths.
- For transactions, **Technical Due Diligence** now includes the AI supply chain: what the target depends on, and what happens when a provider changes or dies.
- For a bounded exposure check, my **Expert Call** books directly online.

The triggers I see in practice: an agent about to get a production role, production data about to flow through a new provider or router, an enterprise AI contract about to be signed, an acquisition of an AI-heavy target, or a silent model change discovered after the fact.

FAQ

What is an AI supply chain audit?

An AI supply chain audit maps every model, router and provider in your production path, traces what data reaches each of them, verifies model versions and behavior over time, and tests the failure and exit paths. It treats the AI components as critical dependencies, with the same evidence standard as a database or a storage layer.

How do I size a pipeline that calls an LLM for every record?

Start from records per second, then convert to model calls per second and to input and output tokens per record, because the unit you buy is tokens, not rows. Then size concurrency, since a streaming call stays open for its full duration, design batching, caching and model cascades to remove calls entirely, measure cost per million records rather than per call, and budget the replay capacity for when a partition has to be reprocessed. I do this sizing against the real workload, not a vendor's per-call price.

How do I keep an LLM stage from silently changing my data?

Log the resolved model version for every request, pin versions where the provider allows it, validate output against schemas and confidence thresholds before load, and run a fixed evaluation on a schedule so a behavior change is detected as a change instead of discovered as a data incident. Treat every model upgrade as a migration with a replay plan.

Can you review a single decision, like choosing a router or a provider?

Yes. The Flash Architecture Review format is built for one decision, one bottleneck, one component: a router choice, a provider contract, a pinning and evaluation strategy, or a synchronous against asynchronous pipeline design.

Why does free AI access matter to a business?

Free access usually moves the cost from money to data or conditions. In the free account I checked, the free GLM 5.3 Flash route comes without zero data retention, and free routers are where credential harvesting and abuse concentrate. Free is a legitimate way to evaluate. It is not a production contract.

Book a 15-min intro call

[Book a free 15-min intro call](#) and we will look at your platform and which service fits. If your platform is about to give a model a production role, or your pipelines are starting to call LLMs at scale, that is exactly the kind of question I work on. If you want a second pair of eyes on your exposure, [my Expert Call](#) books directly online.

Sources

1. Anthropic, "Detecting and countering misuse of AI: September 2026", September 10, 2026.
<https://www.anthropic.com/threat-intelligence-report-september-2026> (PDF: [Anthropic-Detecting-and-countering-091026.pdf](#))
2. International Cyber Digest, thread on the Anthropic report, September 10, 2026.
<https://x.com/IntCyberDigest/status/2098149671957103005>

3. r/LocalLLaMA, "Anthropic: Detecting and Addressing AI Misuse by China", September 2026.
<https://www.reddit.com/r/LocalLLaMA/comments/1wd4mro/>
4. Prakash (@8teAPi), post on the unverified Kimi rumor, September 12, 2026.
<https://x.com/8teAPi/status/2098594444425756808>
5. Chaofan Shou, Hongbo Wen, Yanju Chen, Ryan Jingyang Fang, Yu Feng, "Your Agent Is Mine: Measuring Malicious Intermediary Attacks on the LLM Supply Chain", ACM CCS 2026, arXiv:2604.08407, April 9, 2026. <https://arxiv.org/abs/2604.08407>
6. Chaofan Shou (@shouccccc), first router disclosure, April 10, 2026.
<https://x.com/shouccccc/status/2042423713019412941>
7. Chaofan Shou (@shouccccc), the 6 TB prompt-log dataset claim, September 2026.
<https://x.com/shouccccc/status/2098169782541631871>
8. OrcaRouter, zero data retention and data handling documentation.
<https://docs.orcarouter.ai/operations/zero-data-retention> and
<https://docs.orcarouter.ai/operations/data-handling>
9. OrcaRouter, free-tier model update and access requirements.
<https://x.com/OrcaRouter/status/2096881959830229345> and
<https://x.com/OrcaRouter/status/2098308927855448166>
10. explainx.ai, "OrcaRouter Bakes Refusal Removal Into GLM-5.3-Flash's Native FP8 Weights", August 30, 2026. <https://www.explainx.ai/blog/orcarouter-glm-5-3-flash-uncensored-block-fp8-august-2026>
11. DriftBench, LLM drift detector. <https://driftbench.ai/>
12. OpenAI, Navier-Stokes announcement and statement, September 8, 2026.
<https://x.com/OpenAI/status/2097374640582668336> and
<https://x.com/OpenAI/status/2097375276384567642>
13. Tristan Buckmaster, public statement, September 8, 2026.
<https://cims.nyu.edu/~tristanb/statement.pdf>
14. Haseeb Qureshi (@hosseeb), summary of the Buckmaster and Alpöge story, September 8, 2026.
<https://x.com/hosseeb/status/2097217378753106336>
15. r/academia, thread on the Navier-Stokes episode, September 2026.
<https://www.reddit.com/r/academia/comments/1waxvlt/>
16. Jonathan Murray, "OpenAI can't rule out that it stole its most recent breakthrough", dev.to, September 8, 2026. https://dev.to/jon_at_backboardio/openai-cant-rule-out-that-it-stole-its-most-recent-breakthrough-12d2
17. Snowflake, Cortex AI Functions documentation.
<https://docs.snowflake.com/en/user-guide/snowflake-cortex/aisql>
18. Google Cloud, BigQuery AI.GENERATE documentation.
<https://cloud.google.com/bigquery/docs/reference/standard-sql/bigqueryml-syntax-ai-generate>
19. Liana Patel et al., "Semantic Operators: A Declarative Model for Rich, AI-based Data Processing", arXiv:2407.11418, VLDB. <https://arxiv.org/abs/2407.11418>
20. LOTUS, LLM-powered document processing and data query engine.
<https://github.com/lotus-data/lotus>

21. Data Engineer Academy, "Integrating LLMs into ELT Pipelines", May 8, 2026.

<https://dataengineeracademy.com/blog/integrating-llms-into-elt-pipelines/>

No affiliation: the author has no relationship with Anthropic, OpenAI, OrcaRouter, Snowflake, Google or any of the companies and projects mentioned. Nothing above is an endorsement.

Update 2026-09-29. This post is now part of [AI Infrastructure and SRE: Running Models in Production](#), the field guide to running models in production.

Related posts

- [AI Infrastructure and SRE: Running Models in Production](#)
- [Ox Alpha Review: Test a Free AI Model Before Production](#)
- [Real-Time AI Streaming Is a Product Trend, Not a Benchmark](#)
- [How Big Models Teach Small Models, and Why It Looks Like School](#)
- [After Coldcard, AI Found 1,029 Bugs. Censored Models Found None.](#)
- [MinIO goes into maintenance mode](#)

PDF snapshot of the public article. The online version is authoritative; links remain clickable.