

Executive Guide to Open Weight AI Models

A buyer's guide for engineering and platform leaders weighing open-weight models against hosted frontier APIs. It covers where the capability gap actually is, the cost lines most business cases miss, four deployment patterns, and the licence questions that stall procurement.

01
The decision in front of you

02
What owning the weights
actually buys you

03
Where the gap still is, and where
it is not

04
Which open-weight models are
competitive today

05
What the top of the ranking
measures

06
The real cost of ownership

07
Four deployment patterns and
when each wins

08
The operating layer is the actual
product

09
Licensing, security and
compliance questions that stall
deals

10
Evaluation checklist

Evaluation worksheet

For most of the last three years, an enterprise that wanted frontier model quality had exactly one route: send the prompt to somebody else's API and pay per token. That default is now being renegotiated in procurement meetings, architecture reviews and board decks. Open weights did not catch up everywhere. They caught up in the places where most enterprise volume sits, and the remaining gap is now narrow enough that the choice depends on operations rather than on technology.

This guide is written for the people who have to sign the architecture decision record: engineering leaders, platform owners and the executives who fund them. It does not claim that open weights always win. The first question to settle is how much of your own stack you want to own, and four facts settle it in about a week: where the data has to live, how much sustained volume you actually run, how much of the stack you are willing to operate, and what the licence allows you to do.

SECTION 01

The decision in front of you

The default is still the hosted frontier API, and for good reasons. It is fast to adopt, it needs no capacity planning, and for low and unpredictable volume it is genuinely cheaper than anything you could build. What has changed is the size of the penalty for staying there.

The strongest public measurement of that penalty comes from Epoch AI's Capabilities Index, a composite that folds scores from more than fifty benchmarks into a single capability scale. Their 2026 analysis found that since January 2026 the most capable open-weight models have lagged frontier closed models by [an average of four months, or 8 ECI points](#), with a 90 percent confidence interval of 7 to 11 points. That is a material narrowing. An earlier insight covering January 2023 to October 2025 put the same lag at around three months, but the models at each end of it were further apart in absolute quality. The practical consequence today is that a four month lag lands inside most enterprises' own release cadence.

Four months is an average. The lag is close to zero on knowledge recall, mathematics and instruction following, and it is widest on long-horizon agentic work. Workload mix should drive the decision, and the next three sections set out where the lag lands and which models are worth shortlisting.

Before commissioning a comparison, establish which of three questions you are answering. Each one has its own evidence.

1. **Permission.** Whether the licence and the regulation allow the use you intend. Settle it by reading documents.
2. **Economics.** How much sustained volume you run and what that costs to carry. Settle it with arithmetic about duty cycle.
3. **Equivalence.** Whether your own tasks pass on an open model. Settle it by running those tasks against both options.

Most stalled programmes skip the ordering and start with the third, which is the slowest and most expensive way to discover that they had a licensing constraint all along.

SECTION 02

What owning the weights actually buys you

An open-weight model is a file you can copy, which changes four things enterprises care about. We list them in rough order of importance.

The first is data control. When the weights sit on your infrastructure, the prompt never leaves your perimeter, and neither does the completion. For regulated workloads this removes an entire category of review, because there is no third party processing your data and no sub-processor to add to the register. This is why the strongest enterprise position on open weights usually comes from the security and data governance function rather than from engineering. Whitecap Resources, an oil and gas producer, put the reasoning plainly: "We're an on-prem company because we didn't want our data exposed to laws that we had no control over," said James Wakelin, Director of Business Intelligence at Whitecap. "Put it in the cloud and it may be subject to a different country's legal framework and exposure to being opened at any time." That is the argument for owning the artefact, and it is [documented as a published case study](#).

"We're an on-prem company because we didn't want our data exposed to laws that we had no control over. Put it in the cloud and it may be subject to a different country's legal framework and exposure to being opened at any time. The company made a decision to keep as much on-prem as possible, because then we had control over where that information ended up."

James Wakelin

Director of Business Intelligence @ Whitecap

The second is price stability. Metered per-token pricing moves with a vendor's model lineup, and a depreciation notice can change your unit economics without any change on your side. A model you host has a fixed cost per hour of capacity, and that number does not move when a vendor ships a successor. Whether the fixed number is cheaper is a separate question, addressed later, but the difference between the two shapes matters to anyone forecasting a multi-year budget.

The third is the freedom to specialise. Weights you own can be fine-tuned, quantised, distilled or merged without a negotiated contract amendment, and a quantised variant can cut the memory footprint substantially at a small quality cost. Vendor APIs increasingly allow fine-tuning, but you do not own the resulting artefact, and you cannot move it to another vendor.

The fourth is exit. A model pinned on your own infrastructure keeps working when a vendor changes terms, raises prices or retires the endpoint. In practice this is the benefit executives underweight and engineers value most, because the likelier failure is a healthy vendor deciding that a model you depend on is no longer worth serving.

01

The decision in front of you

02

What owning the weights actually buys you

03

Where the gap still is, and where it is not

04

Which open-weight models are competitive today

05

What the top of the ranking measures

06

The real cost of ownership

07

Four deployment patterns and when each wins

08

The operating layer is the actual product

09

Licensing, security and compliance questions that stall deals

10

Evaluation checklist

Evaluation worksheet

01	The decision in front of you
02	What owning the weights actually buys you
03	Where the gap still is, and where it is not
04	Which open-weight models are competitive today
05	What the top of the ranking measures
06	The real cost of ownership
07	Four deployment patterns and when each wins
08	The operating layer is the actual product
09	Licensing, security and compliance questions that stall deals
10	Evaluation checklist
	Evaluation worksheet

DIMENSION	HOSTED FRONTIER API	OPEN WEIGHTS ON YOUR INFRASTRUCTURE
Where the prompt goes	Third party sub-processor	Stays inside your perimeter
Unit economics	Metered per token, changes with the lineup	Fixed cost per hour of capacity
Deprecation risk	Vendor sets the retirement date	You set the retirement date
Fine-tuning output	Owned by the vendor, not portable	An artefact you own and can move
Time to first token in production	Days	Weeks, and it needs a platform team
Failure mode when usage spikes	Your bill rises	You queue, unless you over-provisioned

Table 2.1 · What changes when you hold the weights rather than rent them.

SECTION 03

Where the gap still is, and where it is not

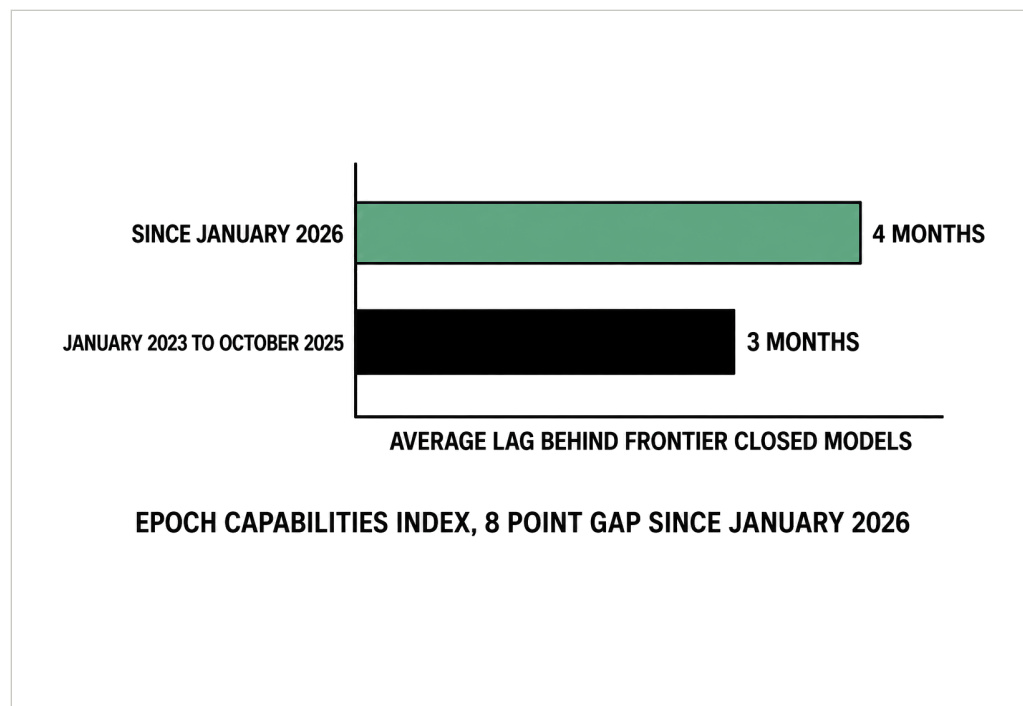
A capability gap of a few ECI points is not evenly distributed, and treating it as one number is the most common analytical mistake in this decision. Open weights are at parity for a large class of work and measurably behind for another class, and the split matters more than the average.

At parity, and this covers a large share of enterprise token volume: classification and routing, extraction into a schema, summarisation, retrieval-augmented question answering, drafting against a fixed style guide, translation, and code completion inside a known repository. These tasks have bounded outputs and are supported by examples, which is the regime where four months of frontier progress buys little.

Still behind: long-horizon agentic execution where a model must chain dozens of tool calls without losing the thread, multimodal reasoning over mixed documents and images, and reliable recall across very large contexts. The clearest independent measure here is METR's 50 percent time horizon, which measures the length of task a model can finish half the time. The best open-weight entry in the published dataset sits at about 54 minutes against about 1,045 minutes for the leading closed preview, [a difference of roughly nineteen times](#). That comparison carries its own caveat, because the dataset holds only four open-weight entries and the most recent dates to November 2025, so it is not a like-for-like reading of 2026.

The pattern is consistent across families. Open weights converge first on capabilities that can be trained from public material and graded by a bounded answer, and last on capabilities that depend on scale and long reinforcement learning on interactive tasks.

01	The decision in front of you
02	What owning the weights actually buys you
03	Where the gap still is, and where it is not
04	Which open-weight models are competitive today
05	What the top of the ranking measures
06	The real cost of ownership
07	Four deployment patterns and when each wins
08	The operating layer is the actual product
09	Licensing, security and compliance questions that stall deals
10	Evaluation checklist
	Evaluation worksheet



The direction of travel in the most recent window is worth noting. Epoch's newest measurement puts the average lag at 4 months since January 2026, slightly larger than the 3 months it measured across the longer January 2023 to October 2025 window, with a 90 percent confidence interval of 7 to 11 ECI points. What changed is what sits inside the gap. A four month old open model in 2026 is a materially more capable artefact than a four month old open model was in 2024, and six independent families now maintain that cadence rather than one or two.

One caveat belongs beside that number, and it comes from the same source. Epoch notes that open-weight models tend to optimise against benchmarks more aggressively than proprietary ones, which means the measured gap is more likely to be understated than overstated. A separate self-published analysis, which ships a reproduction repository, reaches a less comfortable conclusion. It puts the gap at 8 to 10 months on private, contamination-resistant benchmarks against 4 to 6 months on public ones, and finds the gap was narrowest around DeepSeek R1 in January 2025 and has widened since. We would not treat a self-published study as settled, but it points the same way as the caveat above.

The practical consequence is that no source publishes a single percentage-scale gap, and the available measures do not convert cleanly into one another. Epoch's own score file puts the widest spread at 9.88 index points on a logit-like scale that is not a percentage, the Artificial Analysis Intelligence Index puts it at 12 points across 199 models, and LMArena puts it at 45 Elo on text and 31 on vision. An enterprise will do better to cap its exposure by workload class than to argue about which of those numbers is the true one.

There is also a category of signal that tells you almost nothing about fitness for your workload, and it is worth naming so it stops consuming review time.

- **Single benchmark leaderboard positions**, because the top of a public leaderboard is exactly the region model developers optimise against, a limitation Epoch AI states openly about its own index.
- **Parameter counts**, because an active-parameter count in a mixture-of-experts model determines runtime cost while total parameters determine memory footprint, and neither predicts task accuracy by itself.

01	The decision in front of you
02	What owning the weights actually buys you
03	Where the gap still is, and where it is not
04	Which open-weight models are competitive today
05	What the top of the ranking measures
06	The real cost of ownership
07	Four deployment patterns and when each wins
08	The operating layer is the actual product
09	Licensing, security and compliance questions that stall deals
10	Evaluation checklist
	Evaluation worksheet

- **Vendor-reported scores**, which are not independent measurements and rarely use the same prompting or reasoning budget across models.
- **Release recency on its own**, because the practical question is whether a given model is good enough for a defined task, and that is a property of the task.
- **Community enthusiasm**, which tracks novelty rather than reliability and is a poor proxy for how a model behaves on the thousandth request.

SECTION 04

Which open-weight models are competitive today

Averages across families hide the models themselves, so it is worth naming the current leaders. Artificial Analysis publishes the Intelligence Index, which scores models on ten fixed evaluations and reports the result as one number. The figures below come from release v4.3.2.

OPEN-WEIGHT MODEL	INTELLIGENCE INDEX SCORE	LICENCE SHAPE
MiMo-V2.6-Pro	46	Open weights
GLM-5.3 (max)	45	Open weights, commercial use restricted
Kimi K3 (max)	44	Open weights, commercial use restricted
GLM-5.3-Flash	42	Open weights
DeepSeek V4.1 Flash (max)	39	Open weights
Qwen3.8 27B (xhigh)	34	Open weights
K2 Horizon 375B A23B	31	Open weights
MiniMax-M3	29	Open weights, commercial use restricted
Inkling (xhigh)	25	Open weights
Nemotron 3 Ultra	23	Open weights
Muse Glimmer (high)	17	Open weights

Table 4.1 · The leading open-weight models on the Artificial Analysis Intelligence Index v4.3.2.

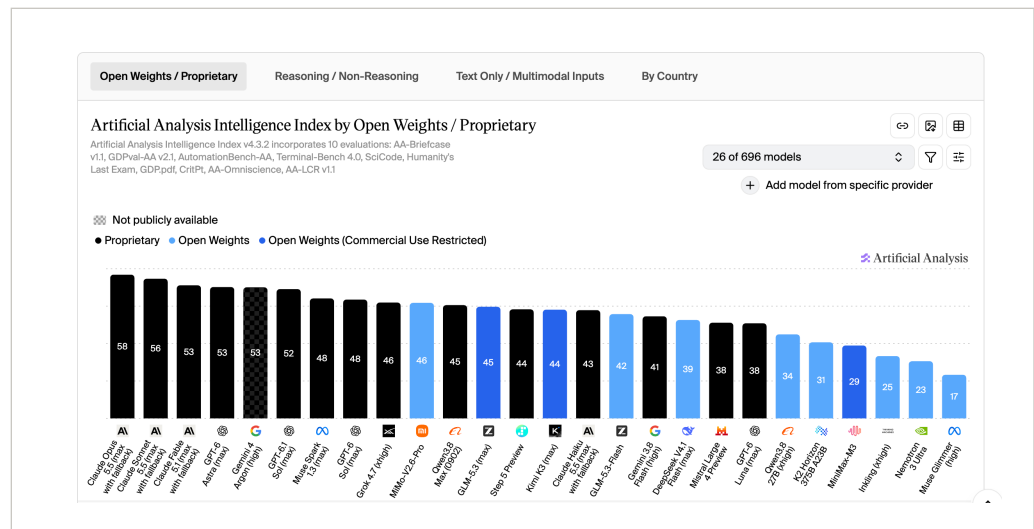
The same release puts the leading proprietary models at 58 for Claude Opus 5.5, 56 for Claude Sonnet 5.5, and 53 for Claude Haiku 5.1 and GPT-6 Astra. It scores Gemini 4 Argon, which it marks as not publicly available, at 53 as well. The best open-weight model scores 46, twelve points behind the leader.

The two groups interleave. MiMo-V2.6-Pro at 46 sits above Qwen3.8 Max at 45 and Step 5 Preview at 44, both of which are proprietary, and beside GLM-5.3 and Kimi K3 in the same band. Three of the five models scoring between 44 and 46 are open weights, and every model above 46 is closed. That is the shape of the gap today: the best open models compete with the middle of the proprietary field and trail its top.

The licence shapes in that table matter as much as the scores, and three of the eleven carry a commercial-use restriction. GLM-5.3, Kimi K3 and MiniMax-M3 are all downloadable, and all three attach conditions that a legal review needs to read before the model reaches a customer-facing product. The licence section below covers what those conditions usually say.

01	The decision in front of you
02	What owning the weights actually buys you
03	Where the gap still is, and where it is not
04	Which open-weight models are competitive today
05	What the top of the ranking measures
06	The real cost of ownership
07	Four deployment patterns and when each wins
08	The operating layer is the actual product
09	Licensing, security and compliance questions that stall deals
10	Evaluation checklist
	Evaluation worksheet

Families also mix the two licensing regimes internally. Alibaba publishes open Qwen weights under Apache 2.0, and the same release of the index scores Qwen3.8 Max on the proprietary side of the line. A single family name therefore tells you nothing about licence obligations, and the model name has to be checked on its own.



SECTION 05

What the top of the ranking measures

The names at the top of that table are genuinely ahead, and the section above describes where. It is worth knowing what they are ahead at, because the tasks that settle the top of the index are not the tasks most enterprises run.

Artificial Analysis builds the index from ten evaluations, and several of them target research-grade work. Humanity's Last Exam, SciCode, Terminal-Bench, GDPpdf and CritPt are all in that set. Epoch AI's FrontierMath points the same way. It is a set of hundreds of original problems written and vetted by working mathematicians, running from advanced undergraduate material to early-career research, and [a typical problem takes an expert mathematician several hours to solve](#). In July 2025 a frontier model reached [gold-medal standard at the International Mathematical Olympiad](#), and other systems have since matched that result on the 2025 problems.

That advance is real, and it matters for research mathematics and for the hardest reasoning work an organisation can point a model at. It also sits a long way from the work that fills an enterprise queue. The largest published study of how people use AI at work follows Microsoft 365 Copilot across more than a million organisations and finds that the [dominant pattern is assistive](#): drafting, editing, advising and explaining, with people checking and rewriting rather than delegating a whole problem. The tasks listed earlier in this paper are of that kind, with bounded outputs and clear acceptance criteria.

For this decision, one measurement is more useful than the ranking. A [2026 analysis from MIT Sloan](#) puts open models at about 90 percent of closed-model performance at the time of release, closing the remaining distance quickly, and puts closed models at around six times the cost of open ones. In one published accounting workflow, replacing the closed frontier models with open weights and re-running the same sixty cases through the same scorer produced [95.0 against](#)

01	The decision in front of you
02	What owning the weights actually buys you
03	Where the gap still is, and where it is not
04	Which open-weight models are competitive today
05	What the top of the ranking measures
06	The real cost of ownership
07	Four deployment patterns and when each wins
08	The operating layer is the actual product
09	Licensing, security and compliance questions that stall deals
10	Evaluation checklist
	Evaluation worksheet

[94.6 for the closed configuration](#), at roughly the same cost and about twice the latency.

The ranking and the volume of enterprise work describe different things. The top of the index is settled by who can do a mathematician's job. Enterprise work sits far below that ceiling, in the region where open weights have been at parity for some time. The question worth asking about any model is whether it is good enough for a named task and whether you can prove that it is. The evaluation checklist at the end of this paper is what turns that proof into a decision.

SECTION 06

The real cost of ownership

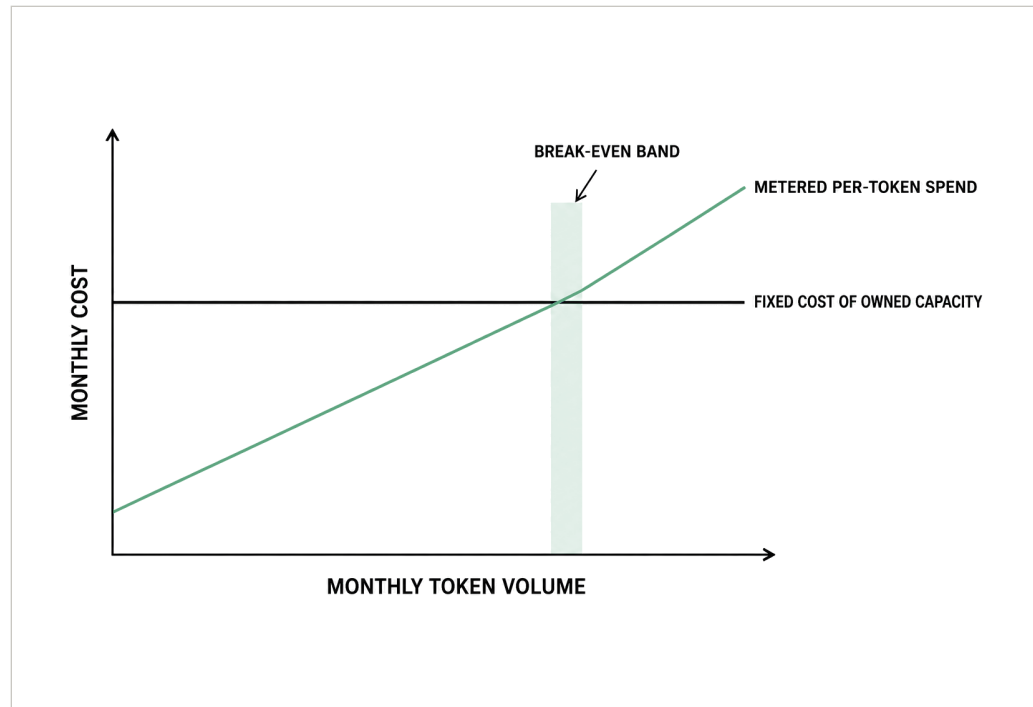
The cost comparison that kills open-weight programmes is the one in the spreadsheet, because a per-token invoice is a complete cost while a per-hour GPU price is not.

Ownership adds at least five cost lines that have no line item on an API bill, and a business case that omits them will look excellent right up to the point where the runbooks have to be written.

- **Idle capacity.** You buy for peak and pay for the trough. A cluster sized for month-end batch load runs at low utilisation for most of the month, and the fixed cost does not care.
- **Evaluation.** Every candidate model, every quantisation variant and every fine-tune has to be measured against your own task set before it goes near production, and doing that credibly is ongoing work rather than a one-off project.
- **Upgrade and regression testing.** When a new open release lands, adopting it means re-running the evaluation suite and re-qualifying guardrails. The cadence of open releases is a benefit and a maintenance liability at the same time.
- **On-call.** A self-hosted inference endpoint is a production service. It fails at three in the morning, it needs version pinning, and the engine needs patching against a moving upstream.
- **The platform itself.** Routing, caching, observability, per-workload accounting and access control are all things a hosted API gives you for free and you now provide.

Those cost lines make ownership a volume decision rather than a mistake. The mechanics are straightforward. Self-hosting beats a premium hosted API at comparatively modest sustained volume, because the alternative is expensive per token, while beating a cheap hosted API for the same open model takes far more volume than a single accelerator can typically serve. The break-even sits wherever your own duty cycle puts it, so plot your actual monthly token volume against the two cost shapes instead of arguing about which is cheaper.

01	The decision in front of you
02	What owning the weights actually buys you
03	Where the gap still is, and where it is not
04	Which open-weight models are competitive today
05	What the top of the ranking measures
06	The real cost of ownership
07	Four deployment patterns and when each wins
08	The operating layer is the actual product
09	Licensing, security and compliance questions that stall deals
10	Evaluation checklist
	Evaluation worksheet



The two cost shapes are what matter: one scales with volume, and the other is driven by fixed capacity and utilisation. The point where they cross moves with your duty cycle.

The per-token side of that comparison is at least observable, and vendor pricing pages are the primary source for it, for example [OpenAI's published per-million-token rates](#) across its current model tiers. Reading them side by side shows the shape of the hosted market: a wide spread between a small, cheap tier and a flagship tier, which is the same spread that determines whether you are replacing an expensive model or a cheap one.

COST LINE	WHAT DRIVES IT	HOW TO PUT A NUMBER ON IT
Metered inference	Tokens in and out, model tier	Vendor pricing page, your own traffic mix
Capacity, owned or leased	Accelerators, term, utilisation	Hourly rate times hours, divided by real duty cycle
Idle capacity	Peak-to-trough ratio	Peak provisioned capacity minus measured average use
Evaluation and qualification	Candidate count, task set size	Engineer days per candidate, per quarter
Upgrade and regression	Release cadence you choose to adopt	Eval reruns plus guardrail requalification per upgrade
On-call and patching	Service criticality, engine churn	Share of a platform engineer, on rotation
Routing, caching, accounting	Whether you build or buy the layer	Either platform cost or engineer quarters

Table 6.1 · The seven cost lines, and where the honest number comes from.

Two of those lines deserve emphasis because they are the ones that get left out and then cause the programme to be cancelled. Evaluation is a standing capability, and the second model you qualify costs less than the first only if you built the harness well. Upgrade testing is the same harness pointed at a moving target, and an organisation that adopts every open release without it is running unmeasured changes in production.

SECTION 07

Four deployment patterns and when each wins

The decision that matters is how much of the stack you want to own. The four realistic patterns sit on a single axis of operational responsibility, and almost every enterprise we see adopts at least two of them at once.

At one end, you call a hosted API and own no weights at all. This remains the right answer for the workload that is genuinely open ended, for the burst that arrives four times a year, and for the team that needs a capability in a week rather than a quarter. The cost is metered and the failure modes are the vendor's problem, which is a real business benefit that procurement decks routinely omit.

Next, a managed endpoint puts the weights in your cloud account on dedicated capacity. You keep residency and network isolation, you stop sharing a noisy neighbour, and you still do not patch a serving engine. It is the pattern that satisfies most internal security review while preserving the economics of someone else operating the inference layer.

Then there is the self-hosted cluster, where the weights, the serving engine, the scheduler and the upgrade cadence are all yours. This is the only pattern that delivers complete control of the artefact, and it is the only pattern that creates the operational obligation described later in this paper.

The fourth pattern is hybrid routing, where a gateway sends each request to a different destination according to policy rather than preference. Sensitive traffic stays inside the perimeter on weights you own, and everything else goes to the endpoint that offers the best price and quality that day.

PATTERN	WHERE THE WEIGHTS RUN	WHAT YOU OPERATE	WHEN IT WINS
Hosted API	Vendor infrastructure	Nothing below your own application	Burst capacity, open-ended synthesis, fastest path to a capability
Managed endpoint	Your cloud account, dedicated capacity	Residency, network, keys, never the engine	Security review that wants isolation without a serving team
Self-hosted	Your accelerators	Everything, including upgrades	Regulated data, hard residency rules, sustained high volume
Hybrid routing	Both, chosen per request	The routing policy and its audit trail	A mixed estate, where one policy cannot fit every workload

01

The decision in front of you

02

What owning the weights actually buys you

03

Where the gap still is, and where it is not

04

Which open-weight models are competitive today

05

What the top of the ranking measures

06

The real cost of ownership

07

Four deployment patterns and when each wins

08

The operating layer is the actual product

09

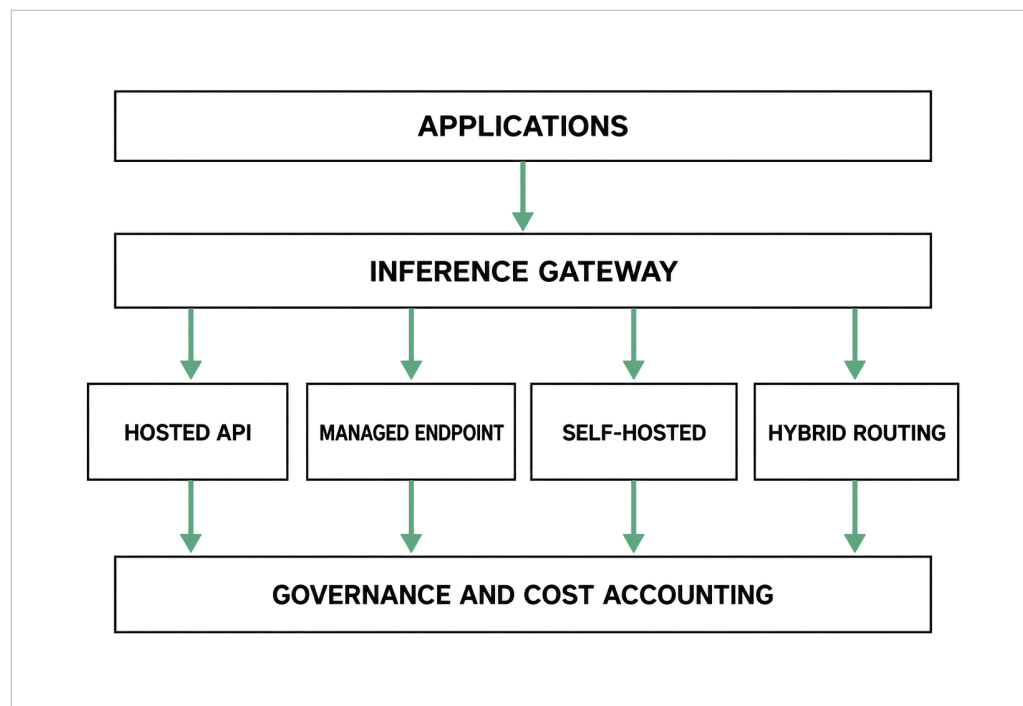
Licensing, security and compliance questions that stall deals

10

Evaluation checklist

Evaluation worksheet

01	The decision in front of you
02	What owning the weights actually buys you
03	Where the gap still is, and where it is not
04	Which open-weight models are competitive today
05	What the top of the ranking measures
06	The real cost of ownership
07	Four deployment patterns and when each wins
08	The operating layer is the actual product
09	Licensing, security and compliance questions that stall deals
10	Evaluation checklist
	Evaluation worksheet



Every request crosses the same gateway and the same governance boundary, whichever destination the policy chooses.

The practical advice is to start where the value is and let the pattern follow the constraint. Three things change as you move down that list, and only the first of them is a technical change.

- The weights become your artefact, so versioning, provenance and rollback become your problem.
- The cost becomes dominated by fixed capacity, so utilisation becomes a first-class metric rather than an implementation detail.
- The upgrade decision becomes yours, which means the regression risk is yours as well and has to be absorbed by an evaluation harness rather than by a vendor's release process.

SECTION 08

The operating layer is the actual product

Executives usually hear this part of the argument too late. The weights themselves are a commodity that anyone can download, and the licence usually permits it. What separates a demonstration from a production system is the layer around the weights, and that layer carries real cost, requires deliberate design, and does not appear in any benchmark table.

The serving engine is the first component of that layer. Continuous batching and paged attention are what made open weights economically viable to serve, because they let a single accelerator handle many concurrent requests without wasting memory on padding, a design documented in [vLLM's own documentation](#) and in the [original paged attention write-up](#) from June 2023, which is the design's origin story rather than a current performance measurement.

Dedicated stacks such as [NVIDIA's TensorRT-LLM](#) push further with in-flight batching, quantisation schemes such as FP8, FP4 and INT4 AWQ, and speculative decoding. For a buyer, the engines are largely interchangeable, the right choice

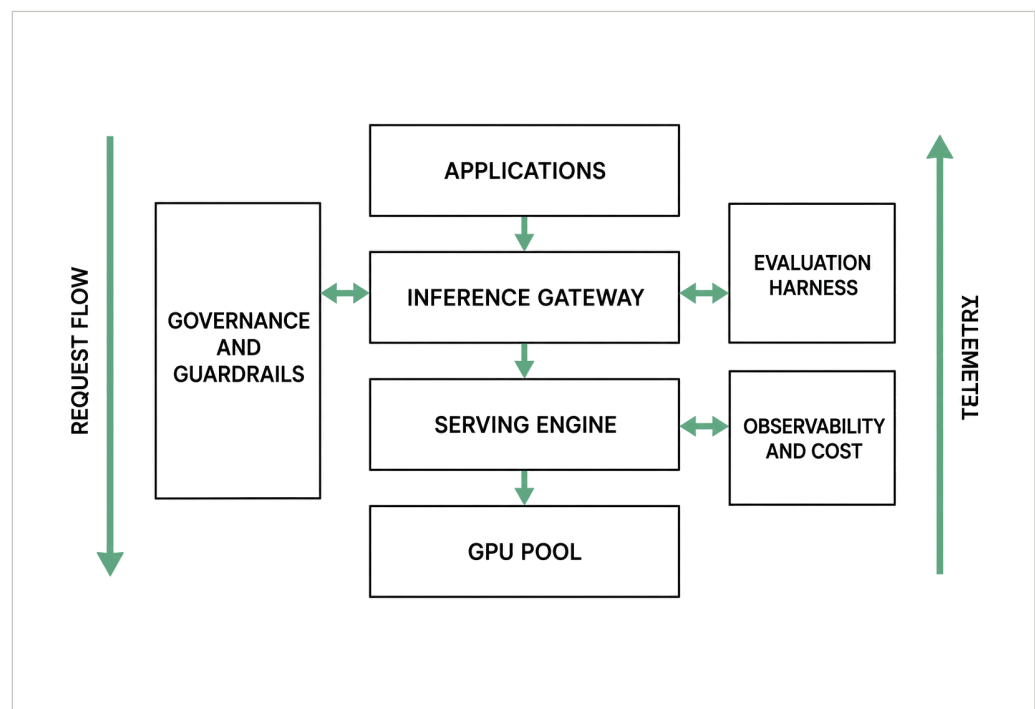
01	The decision in front of you
02	What owning the weights actually buys you
03	Where the gap still is, and where it is not
04	Which open-weight models are competitive today
05	What the top of the ranking measures
06	The real cost of ownership
07	Four deployment patterns and when each wins
08	The operating layer is the actual product
09	Licensing, security and compliance questions that stall deals
10	Evaluation checklist
	Evaluation worksheet

depends on your traffic shape, and the ability to change that choice later is worth more than any single benchmark win.

Caching is where the operating layer most often over-promises. Prefix and key value reuse are genuinely valuable when traffic repeats a common preamble, which is the norm for retrieval-augmented systems sharing one instruction block. Semantic caching is a different proposition, and the gateway vendor's own documentation is unusually direct about its limits: it suits single shot prompts and [goes badly wrong on agentic traffic](#), where two superficially similar requests imply different actions.

Beyond the engine, five things have to exist, and none of them arrives with the weights.

- A gateway that routes per request, holds the policy, and produces an audit trail for every routing decision.
- An evaluation harness that can qualify a candidate model against your own tasks before it reaches traffic.
- Observability that attributes cost and latency per workload rather than per cluster.
- Guardrails that are enforced in the serving path, not documented in a policy wiki.
- A versioning discipline covering weights, engine, quantisation and prompt template as one deployable unit.



Request flow descends through the gateway into the engine and the accelerators. Telemetry, evaluation signal and cost accounting have to travel back up, or the layer operates blind.

This is the layer that a platform team ends up building whether or not anyone decided to build it, and it is the layer where the difference between a pilot and a production estate is decided. It is also, in our experience, the reason a programme that looked cheap on paper does not stay cheap.

SECTION 09

Licensing, security and compliance questions that stall deals

A licence is a contract, and open weights are not open source. Two models can both be downloadable today and impose completely different obligations on your legal team, your product, and the way you let a partner integrate your system.

The permissive end of the market is genuinely permissive. [Open-weight releases from Alibaba's Qwen family](#) are published under Apache 2.0, and [DeepSeek's releases](#) are published under MIT, both of which permit commercial use, modification and redistribution with attribution and no revenue test.

Where the landscape has shifted is that permissive licensing is now a competitive claim in its own right, with vendors such as Z.ai advertising [an MIT licence](#) for their open releases on the explicit basis that it carries no regional limits.

The custom end is where reviews stall. Meta's [Llama 4 Community License](#) adds additional commercial terms that apply above a monthly active user threshold, and it is paired with an [acceptable use policy](#) that flows downstream to anyone you pass the model to.

Google's Gemma family is the clearest illustration of how fast this can change inside a single family. Gemma 4 is now distributed under [Apache 2.0](#), having moved off the bespoke Gemma Terms of Use that governed earlier generations, so a licence review that is a year old may already be describing a different contract.

LICENCE SHAPE	WHAT IT GRANTS	WHAT IT COSTS YOU
Apache 2.0 or MIT	Commercial use, modification, redistribution, attribution only	Nothing beyond attribution compliance and your own diligence
Community licence with a revenue or user threshold	Broad use up to a defined ceiling	A tracking obligation, and a renegotiation if you cross the ceiling
Licence with a service clause	Use, with a separate commercial route for hosted resale	A separate agreement before you embed the model in a product you sell
Acceptable use policy attached to any of the above	Use, subject to use restrictions	Inherited restrictions on your own customers, and a duty to pass them on

Regulation has its own shape, and it does not map neatly onto licence permissiveness. The European Commission's obligations for providers of general purpose AI models have applied since [2 August 2025](#). Many teams assume an open licence exempts them from the whole regime, which is the single most common compliance error we see in this area. The exemption is narrower than the assumption, and it is textual rather than interpretive. Article 53(2) lifts the training-content and copyright duties for a model released under a free and open-source licence whose weights, architecture and usage information are public, but it does not extend to models with systemic risks, and Recital 103 withholds it from any component monetised through a price, technical support or a platform while confirming that hosting on an open repository is not by itself monetisation. The Commission's [published answers](#) set out the same boundary. Read plainly, open weights you run yourself sit inside the exception, and the same weights served to your customers as a product generally do not.

Enforcement is the other half of the picture. From 2 August 2026 the AI Office can require corrective measures and issue fines against general purpose AI model providers, at a ceiling of 3 percent of worldwide annual turnover or EUR 15 million, whichever is higher, and a [2026 amendment to the Act's application dates](#)

01
The decision in front of you

02
What owning the weights actually buys you

03
Where the gap still is, and where it is not

04
Which open-weight models are competitive today

05
What the top of the ranking measures

06
The real cost of ownership

07
Four deployment patterns and when each wins

08
The operating layer is the actual product

09
Licensing, security and compliance questions that stall deals

10
Evaluation checklist

Evaluation worksheet

01	The decision in front of you
02	What owning the weights actually buys you
03	Where the gap still is, and where it is not
04	Which open-weight models are competitive today
05	What the top of the ranking measures
06	The real cost of ownership
07	Four deployment patterns and when each wins
08	The operating layer is the actual product
09	Licensing, security and compliance questions that stall deals
10	Evaluation checklist
	Evaluation worksheet

left the 2025 date for these duties untouched, and the Commission has [published guidelines](#) clarifying their scope.

The security questions are the ones that have changed least and matter most. A downloaded artefact arrives without a vendor's security process behind it, so provenance and integrity checking are your responsibility. Prompt injection and unsafe tool use remain properties of the whole system rather than of any classifier. An acceptable use policy you inherited creates obligations toward your own users that a procurement checklist will not surface on its own.

Loblaw runs a very large engineering estate and has to answer those questions in front of its own risk function rather than in a slide deck. Charu Pujari, Senior Vice President of Engineering and AI at Loblaw, describes the balance they were looking for, which is the same balance this section is about.

"We needed a platform that gives our teams the freedom to use the best tools available while maintaining the governance and oversight that a company of our scale requires. Shakudo delivers exactly that."

Charu Pujari

Senior Vice President, Engineering and AI, Loblaw

You can read how that programme is governed in the [Loblaw Digital case study](#).

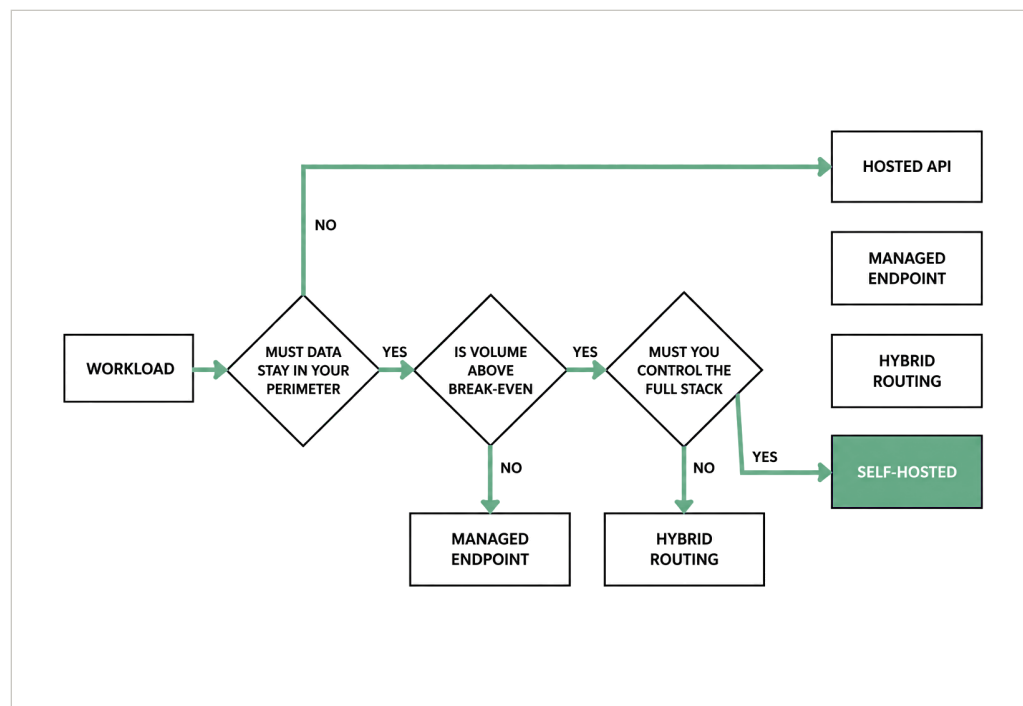
The blockers that stop a deployment are rarely technical: a licence nobody has read end to end, a residency claim that has never been tested against the architecture, an acceptable use policy discovered after product launch, and an evaluation harness that does not exist when the first upgrade arrives.

SECTION 10

Evaluation checklist

By this point the decision has a shape. Data that must stay inside the perimeter pushes you toward weights you run. Sustained volume decides whether the fixed cost is worth carrying. The appetite to own a serving stack decides whether you operate it or rent it. A gateway lets you answer differently for different workloads in the same estate.

01	The decision in front of you
02	What owning the weights actually buys you
03	Where the gap still is, and where it is not
04	Which open-weight models are competitive today
05	What the top of the ranking measures
06	The real cost of ownership
07	Four deployment patterns and when each wins
08	The operating layer is the actual product
09	Licensing, security and compliance questions that stall deals
10	Evaluation checklist
	Evaluation worksheet



The diagram resolves three questions into four destinations. Drawing it explicitly turns the answers into policy that can be reviewed, changed and audited instead of re-litigated per project.

- ☐ The workload is named, with its data classes, and the residency requirement is stated as an architectural constraint rather than an aspiration.
- ☐ The monthly token volume is estimated from measured traffic, and the break-even shape has been modelled against a named alternative.
- ☐ Every shortlisted model's licence has been read in full, including any threshold, service clause and acceptable use policy.
- ☐ The evaluation harness exists and can qualify a candidate model against your own tasks before it reaches production traffic.
- ☐ Cost and latency are attributable per workload, not only per cluster.
- ☐ The weights, engine, quantisation and prompt template deploy as one versioned unit with a tested rollback.
- ☐ A serving-engine substitution has been tested, so the engine choice is reversible.
- ☐ Routing policy is stored as configuration with an audit trail, not embedded in application code.
- ☐ Upgrade cadence is a deliberate decision, with regression testing budgeted rather than assumed.
- ☐ An exit path exists in both directions, so the workload can move between self-hosted and hosted without a rewrite.

The weights are the cheap part, so decide who operates the layer around them before you download anything.

The gap between open weights and the closed frontier is now measured in months, and it is moving faster than most procurement cycles can follow. Open weights will not be the right answer for every workload, but the default should become a decision someone owns rather than a purchase order.

WORKSHEET

Evaluation Worksheet

Score each criterion 1 to 5 for the platform under review. Weight reflects how often the criterion blocks a deployment outright. Reproduce this page freely for internal comparison.

#	CRITERION	WEIGHT	SCORE	WEIGHTED
01	Data residency enforceable at the infrastructure layer, not by policy	3		
02	Model weights version pinned and reproducible across environments	3		
03	Licence reviewed in full, including any threshold and service clause	3		
04	Evaluation harness can gate a model upgrade without a code change	3		
05	GPU utilisation and cost visible per workload, not per cluster	2		
06	A workload can move between self-hosted and hosted without a rewrite	2		
07	Runbooks published for every component you operate yourself	2		
Total				

Worksheet 1 · Weighted score out of 90.

01
The decision in front of you

02
What owning the weights
actually buys you

03
Where the gap still is, and where
it is not

04
Which open-weight models are
competitive today

05
What the top of the ranking
measures

06
The real cost of ownership

07
Four deployment patterns and
when each wins

08
The operating layer is the actual
product

09
Licensing, security and
compliance questions that stall
deals

10
Evaluation checklist

Evaluation worksheet

Run your data and AI stack on infrastructure you control

Shakudo is the operating system for data and AI. It installs into your own cloud account or on-premise hardware, integrates the tools your team already wants to use, and keeps them upgraded, secured and observable. Platform teams ship products instead of maintaining plumbing.

[Book demo](#)

[Read customer case studies](#)

WHAT SHAKUDO GIVES YOU

Your infrastructure, your data. Shakudo runs inside your cloud account, your VPC or your data centre. Data and models never leave the boundary you already trust.

Over 200 integrated data and AI tools. Best-of-breed components, installed, version-matched and upgraded for you, with no vendor lock-in on any single layer.

Governance built in, not bolted on. Access control, audit trails and per-workload cost visibility come with the platform rather than as a later project.