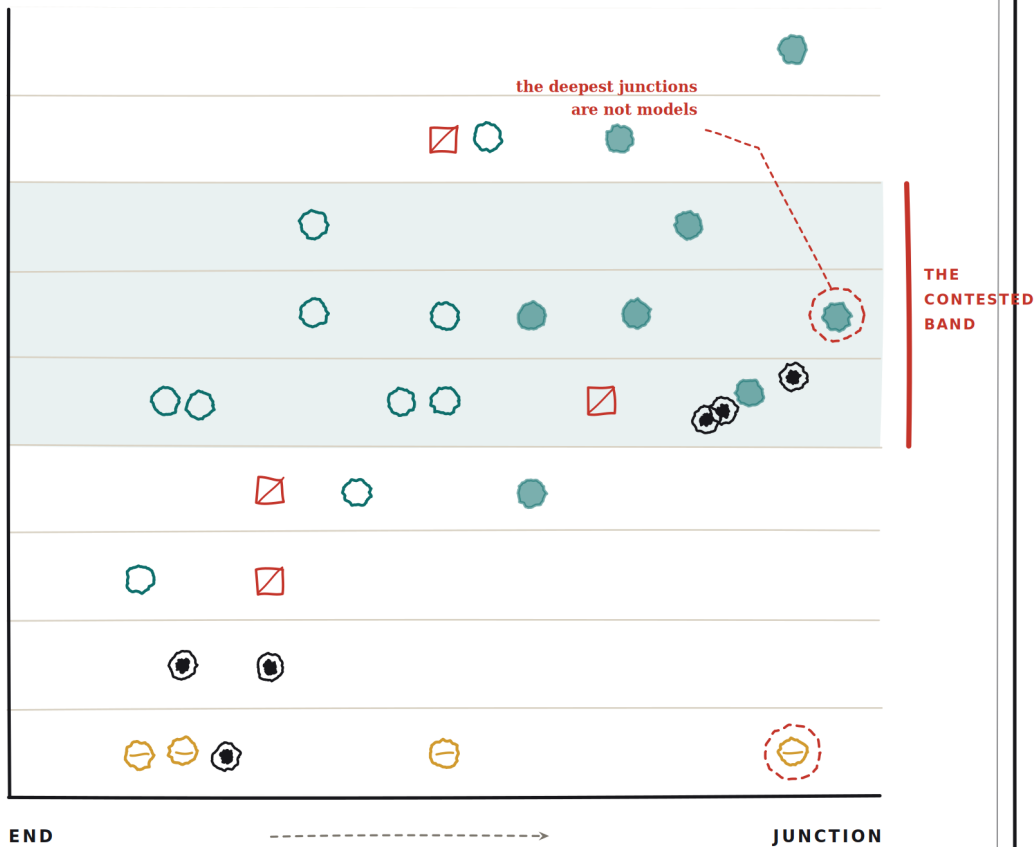


# The Enterprise AI Coordinate System

Where value pools, where you get locked in —  
a buyer's map of the whole stack, from judgment to silicon.



**Gennaro Cuofano**

*The Business Engineer*

2026

[businessengineer.ai](https://businessengineer.ai)

The Enterprise AI Coordinate System

An analysis by Gennaro Cuofano — The Business Engineer

Published August 2026 · [businessengineer.ai](https://businessengineer.ai)

Text, figures and the thirty-seven plates are original work of the author.

Junction-depth figures are editorial estimates; market-colour statistics are directional. See the note on rigor in chapter IV.

**C O N T E N T S**

|            |                                                     |    |
|------------|-----------------------------------------------------|----|
|            | <b>Introduction</b>                                 | 5  |
| <b>I</b>   | <b>The category is not the map</b>                  | 7  |
| <b>II</b>  | <b>The upper layers, where the work accumulates</b> | 15 |
| <b>III</b> | <b>The lower layers, where the machinery runs</b>   | 23 |
| <b>IV</b>  | <b>The teardown, where the war is decided</b>       | 33 |
| <b>V</b>   | <b>The playbook</b>                                 | 41 |
| <b>VI</b>  | <b>What's next</b>                                  | 49 |
| <b>VII</b> | <b>The models that carry it</b>                     | 55 |
|            | <b>Conclusion</b>                                   | 59 |
|            | <b>Addendum — August 2026</b>                       | 61 |
| <b>A</b>   | <b>Appendix — Methodology</b>                       | 67 |



---

# Introduction

As I've been arguing for the past few years, Enterprise AI has become the defining force shaping this phase of AI adoption.

Consumer AI will become increasingly important over the next couple of decades. But the harder and more consequential challenge is happening inside large enterprises: embedding AI deeply enough to transform the organisation without destroying the very capabilities that made it competitive in the first place.

“Transformation” is an overused and loaded term. Here, it has a very specific meaning: what happens to an enterprise when AI becomes part of its core operating system?

In previous pieces, I've explored how enterprise alliances — the coalitions of clouds, labs, data incumbents and integrators now forming around every layer of the stack — are shaping the broader AI ecosystem. But as these alliances deepen, a more fundamental strategic problem is emerging.

For a large enterprise, and increasingly for mid-sized organisations as well, the primary risk is no longer failing to adopt AI. The risk is adopting it incorrectly.

An enterprise can deploy AI in ways that gradually erode its competitive advantage, weaken its organisational knowledge, or transfer control of critical capabilities to external AI-native players. This can happen because the organisation fails to adapt its operating model, because it embeds the wrong components of the AI stack into its core, or because it becomes structurally dependent on vendors whose incentives are not aligned with its own.

The alternative is to build what I think of as an agnostic enterprise AI harness: an architectural and organisational layer that makes AI adoption controlled, governed, modular, and adaptable.

Such a layer should allow the enterprise to integrate external models, platforms, and capabilities without surrendering control over its data, workflows, knowledge, economics, or strategic differentiation. It can be built internally, through external agnostic partners, or through a combination of both.

But the technical architecture is only part of the problem.

The AI system also has to be aligned with the organisational core and understood by the people operating it: employees, executive teams, global functions, local business units, and the stakeholders responsible for turning AI capabilities into actual business outcomes.

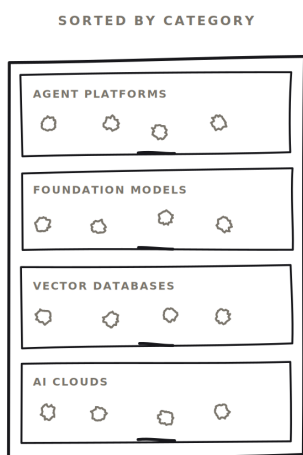
These dynamics deserve far more attention, because much of the enterprise AI landscape over the next three to five years can ultimately be reduced to one strategic question:

*How does an enterprise adopt AI, preserve and strengthen its competitive advantage, and improve its economics — without losing control of its profitability, margins, organisational core, and strategic optionality to AI-native players that could eventually lock it into their platforms?*

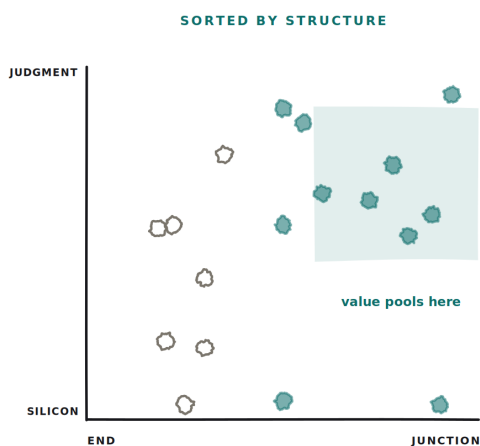
That is the problem this guide sets out to address.

# I

## The category is not the map



*tells you what a thing is called*



*tells you where value pools  
and where you get locked in*



**PLATE 1** — Structural position is the only organizing principle that survives a buying decision.

Every enterprise AI map you have seen sorts by category. Agent platforms. Vector databases. Foundation models. AI clouds. Then it asks who leads each box, and hands you a ranked list.

The list is useless. Not because the data is wrong. Because the organising principle is wrong.

A category tells you what a thing is called. It cannot tell you the two things a purchase turns on:

┃ *Where does value pool, and where do I get locked in.*

Those are structural questions, not naming questions. Where does the component sit in the dependency stack? And is it a junction — something everything routes through and leaving costs a fortune — or an end, something you rent, consume and swap without pain?

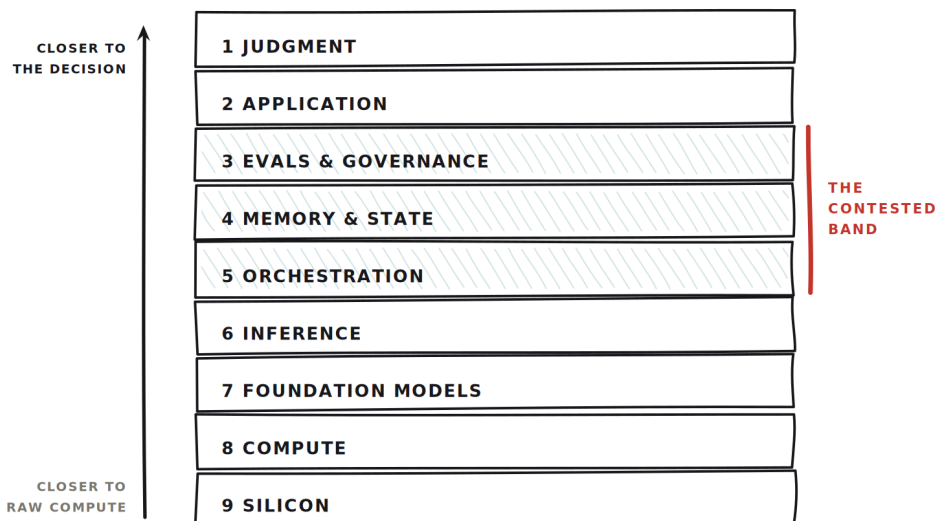
A foundation model and a data ontology can share the same category slide and sit at opposite poles.

Consider the two clearest cases on the market right now. The model layer is the most crowded, most commoditised layer on the board — more than half of the priced catalogue is free or nearly free, with the median flagship going for less than the price of a coffee per million tokens. Meanwhile Palantir sits on a decision surface so deep it has no export path, and grows near triple digits on that fact alone.

Same map if you sort by category. Opposite universes if you sort by structure.

Structural position is the only organising principle that survives a buying decision. So drop the categories, and build two axes.

## The vertical axis — nine layers, from decision down to silicon



### Read it as gravity.

Value pools where switching costs accumulate.  
It drains where substitutes are one API call away.

---

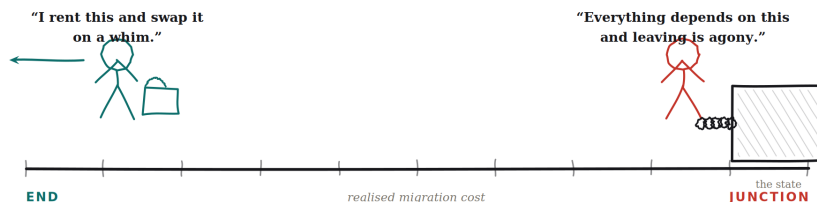
**PLATE 2 —** *The higher the layer, the harder it is to rip out.*

Ordered by how close each sits to the business decision. From the top: Judgment, Application, Evals and governance, Memory and state, Orchestration, Inference, Foundation models, Compute, Silicon.

Read it as gravity. The higher a layer, the closer it sits to the decision a human commits, and the harder it is to rip out. The lower a layer, the closer it sits to raw, fungible compute.

Value does not spread evenly up this stack. It pools where switching costs accumulate. It drains from where substitutes are one API call away.

## The horizontal axis — junction to end



**Scored from the buyer's seat.  
Never from the vendor's pitch deck.**

A vendor spends a decade building a junction,  
then markets it as an end — because frictionless sells.

**PLATE 3 —** *The vendor's crown jewel is the buyer's commodity.*

From “everything depends on this and leaving is agony” to “I rent this and swap it on a whim.”

Here is the rule that makes the whole map honest, and the rule every category map breaks:

*Junction-versus-end is scored from the **buyer's seat**, as realised migration cost. Never from the vendor's pitch deck.*

That changes everything downstream, because **the vendor's crown jewel is the buyer's commodity**.

A vendor spends a decade building a junction and then markets it as an end. Open. Portable. Swap anytime. A frictionless-sounding product sells better than a moat.

The buyer's job is to invert the pitch:

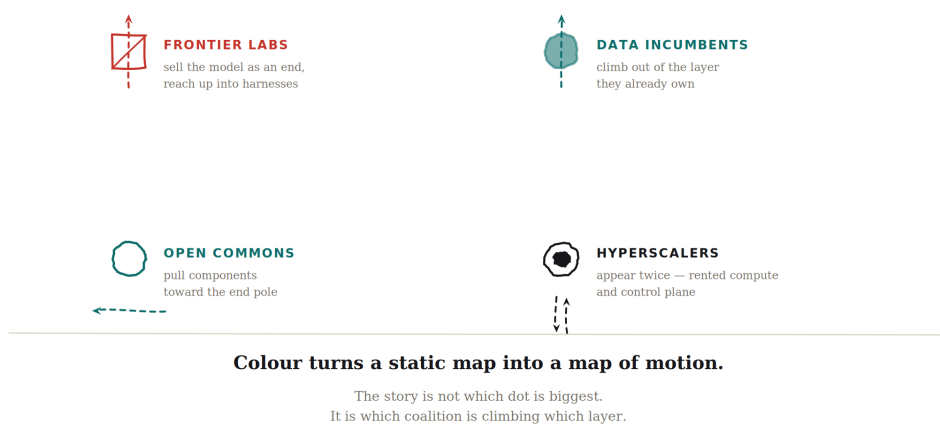
- When a lab prices a model as a cheap, interchangeable end — believe it. That layer is a commodity. Rent it lightly.

- When the same lab, or the data incumbent one floor up, hands you a “free and open” harness that quietly threads through its identity, governance and catalog — the openness is the disguise. The control plane underneath is the junction.

This is why the same open component can score two opposite ways, and why the map always resolves toward realised cost. vLLM is buyer-owned open infrastructure and the right thing to run — but its migration cost is genuinely low, so it sits nearer the end pole than its dominance suggests.

Popularity is not lock-in. What locks you in is what you cannot take with you when you walk. The typed ontology. The identity graph. The compiled kernel. The packaging line in Taiwan.

## Colour — who is doing the locking



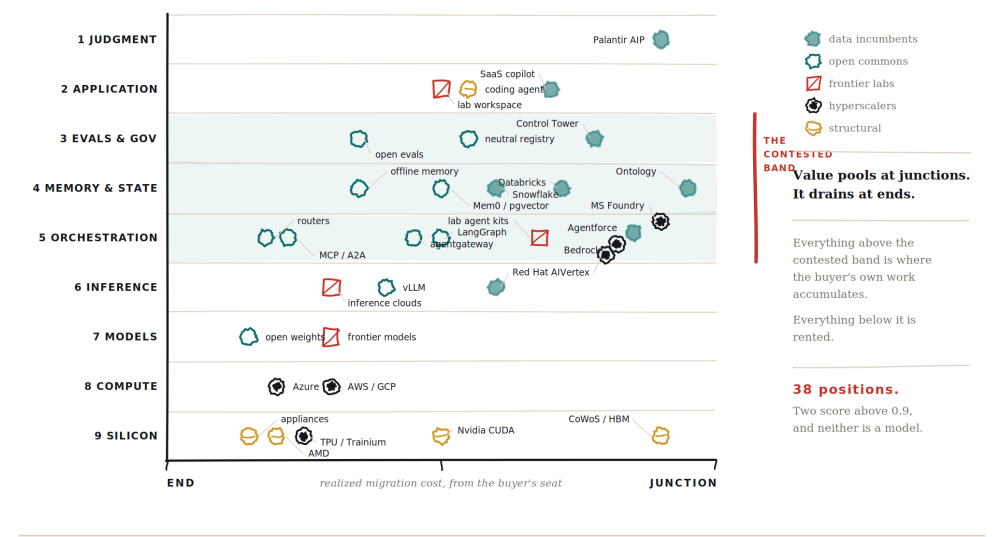
**PLATE 4 —** *The climb is concentrated in three contested middle floors.*

One thing the two axes cannot show: who is doing the locking. That is not a position. It is an allegiance. So it becomes colour, not coordinate.

- The **frontier labs** sell the model as an end while reaching up into workspaces and harnesses.
- The **data incumbents** — Palantir, Databricks, Snowflake, Salesforce, ServiceNow — annex the middle band by climbing out of the layer they already own.
- The **open commons** — LangGraph, Mem0, vLLM, MCP — pulls components toward the end pole.
- The **hyperscalers** get their own colour, because they appear on the board twice. Once as rented-away compute, near the end. Once as control-plane junctions, near the top of the score. That duplication is the point.

Colour turns a static map into a map of motion. The story of this year is not which dot is biggest. It is which coalition is climbing which layer. And the climb is concentrated in three contested middle floors: Orchestration, Memory, Evals and governance.

Now walk the board. Top to bottom, junction to end.



**PLATE 5 — The board.** Nine layers × junction↔end from the buyer's seat, coalition as shape. Thirty-eight positions.



# II

## The upper layers, where the work accumulates



**PLATE 6 —** Own the four upper layers. Rent everything below them.

The war is not fought at the model. The model is a price-competed end. The moat is built above it, in the four layers where the buyer's own work piles up: Judgment, Application, Evals and governance, Memory and state.

Read each layer as two poles. A junction pole, where migration cost collects. An end pole, where the buyer rents and walks. The through-line this year is the same everywhere:

*The data-and-workflow incumbents climb out of the layer they own to annex the layers above it.*

## Layer 1 — Judgment

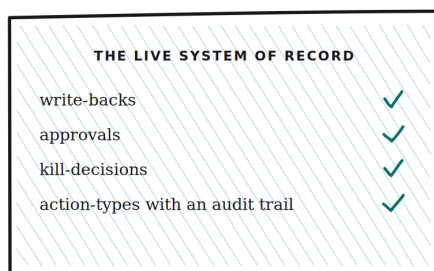
**The junction is the decision graph and the action-types.**

the decision



executes

**They have no export path.**



*fully portable — and that is the decoy*

**PLATE 7 —** *Not a chatbox that suggests. The place where the decision executes.*

### One occupant, and it is Palantir.

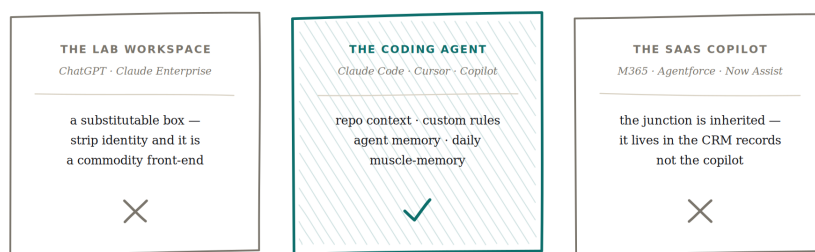
For most of this map's history the top layer was empty. Now it isn't. Palantir AIP is the surface where humans and agents commit business judgments — write-backs, approvals, kill-decisions, action-types with an audit trail. Not a chatbox that suggests. The place where the decision executes against a live system of record.

And here is the trick the whole board turns on. AIP’s architecture makes the underlying model fully portable. Swap OpenAI for Anthropic for a sovereign open-weight model — nobody cares. The model is a decoy. The junction is the decision graph and the action-types, and they have no export path.

You cannot take your AIP decision surface to a competitor.

The direction of travel hardened this year. Palantir became the first partner to run classified inference inside the most restricted government environments, and extended the same stack into regulated, air-gapped commercial deployments. The dependency arrow points straight down — Judgment sits on Ontology.

## Layer 2 — Application



**All three swap models freely.**  
One is lab-owned, one hyperscaler, one independent.

**The coalition is mixed because the model underneath is fungible  
and the workflow above it is not.**

**PLATE 8 —** *The junction is never the intelligence. It is the accumulated context around it.*

### Three cells, one that actually holds.

The obvious occupant is the frontier-lab workspace — ChatGPT Enterprise, Claude Enterprise. It is the weakest of the three. The chat surface is a substitutable box. Strip identity and access, and it is a commodity front-end.

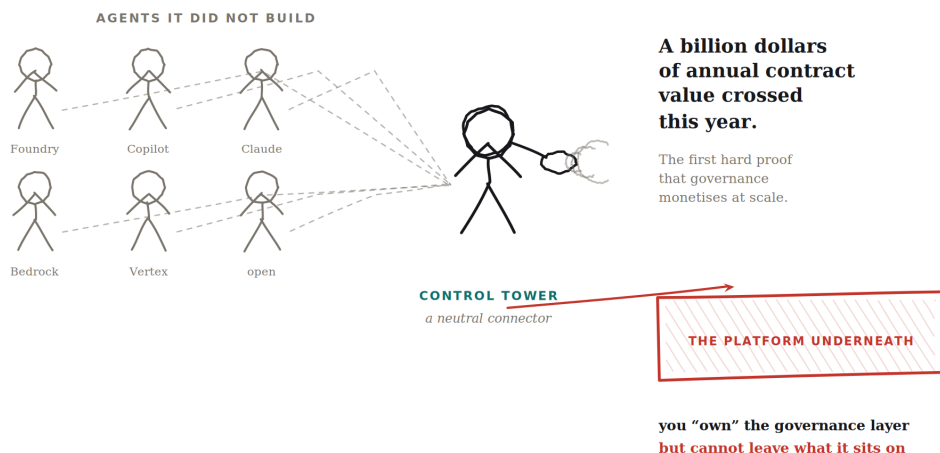
The junction that actually forms in the buyer's daily flow is the coding agent — Claude Code, Cursor, GitHub Copilot — and it is genuinely coalition-independent. The switching cost is the accumulated repo context, the custom rules, the agent memory, the developer muscle-memory built into a daily workflow. All three swap models freely. Claude Code is lab-owned. Copilot is hyperscaler. Cursor is independent. The coalition is mixed because the model underneath is fungible and the workflow above is not.

This is the cleanest single illustration of the whole thesis:

*The junction is never the intelligence. It is the accumulated context around it.*

The third cell is the enterprise SaaS copilot — Microsoft 365 Copilot, Salesforce Agentforce, ServiceNow Now Assist. Its junction is inherited. The copilot is a substitutable chat surface. The switching cost lives in the CRM records, the workflows, the document graph it is bolted onto. The copilot is the wedge. The system-of-record is the moat.

## Layer 3 — Evals and governance



---

**PLATE 9 —** *Never rent governance from the vendor that also owns the workflow you are governing.*

### **This is the battlefield.**

The dominant junction is ServiceNow AI Control Tower — a pure control-plane play that discovers, observes, governs and measures AI across any vendor, through a neutral connector, wrapped in the compliance frameworks a regulated buyer needs.

Watch the shape of the move. ServiceNow governs agents it did not build, using the open commons as its reach mechanism, then captures the buyer through workflow gravity. You “own” the governance layer but cannot leave the platform underneath it.

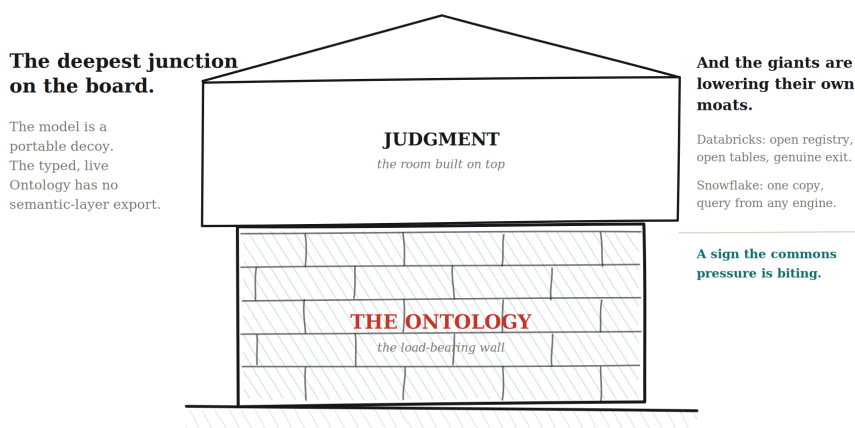
This year the layer crossed a threshold. ServiceNow AI passed a billion dollars in annual contract value — the first hard proof that the governance-and-agent layer monetises at scale for a company that is not a frontier lab. Control Tower extended across Microsoft’s agent ecosystem, and Anthropic signed on as the first design partner, wiring Claude directly into the workflows.

The counter-position is the open eval harness — Phoenix, Ragas, DeepEval, Promptfoo. Self-hostable, code-first, no lock-in. Between the two sits the cell the buyer should occupy: a neutral, self-hostable registry where the eval history accrues but stays in your hands.

The instruction for this layer is not subtle:

*Never rent governance from the vendor that also owns the workflow you are trying to govern.*

## Layer 4 — Memory and state



**PLATE 10 —** *The junction is never the intelligence. It is always the accumulated state.*

**The deepest junction on the board lives here.**

Palantir's Ontology is the highest realised migration cost anywhere on the map. Same architecture as Layer 1. The model is a portable decoy. The typed, live Ontology is the buyer's decision substrate, and it has no semantic-layer export.

This is why Palantir appears at both the top and the middle of the stack. The Ontology is the load-bearing wall. Judgment is the room built on top of it.

The other data giants are, tellingly, lowering their own moats — a sign the commons pressure is biting.

**Databricks** is the most portable data-incumbent junction by design. Its eval registry is open and self-hostable. Its catalog governs open table formats with a genuine exit. The lock-in is governance-graph gravity, not format capture. It closed a fresh round this summer at a \$188 billion mark — a 40% mark-up in five months. The market is paying for the discipline of an honest exit.

**Snowflake** is dismantling its own data-gravity moat in public. Open table formats, an egress-cost optimiser, an open catalog — one copy, query from any engine. The read: own the governed data model, treat the functions on top as substitutable.

The counter-positions are gaining ground. Open agent memory over pgvector is the default now, framework- and model-agnostic, with structured export that keeps the data yours. And at the far end, a fully offline, MIT-licensed memory system posts frontier retrieval scores with zero API calls. That is the portability shock — proof that the layer Google now meters by the event can be owned outright for nothing.

The pattern across all four upper layers is one motion. The data incumbents sell you an open model and an open harness, then manufacture the switching cost one floor down — in the Ontology, the catalog, the platform, the metered memory.

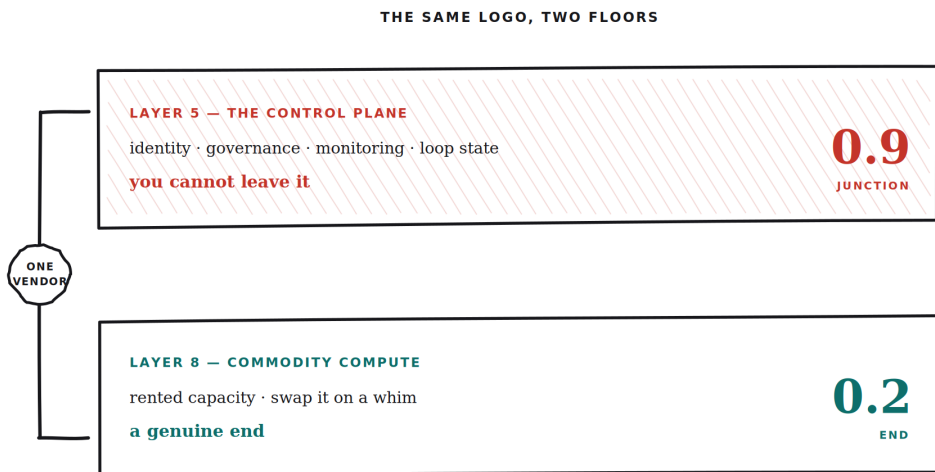
The junction is never the intelligence. It is always the accumulated state.

***Own the four upper layers. Rent everything below them.***



# III

## The lower layers, where the machinery runs



Selling both, from the same logo, books the margin of a lock-in while the buyer believes it is only renting compute.

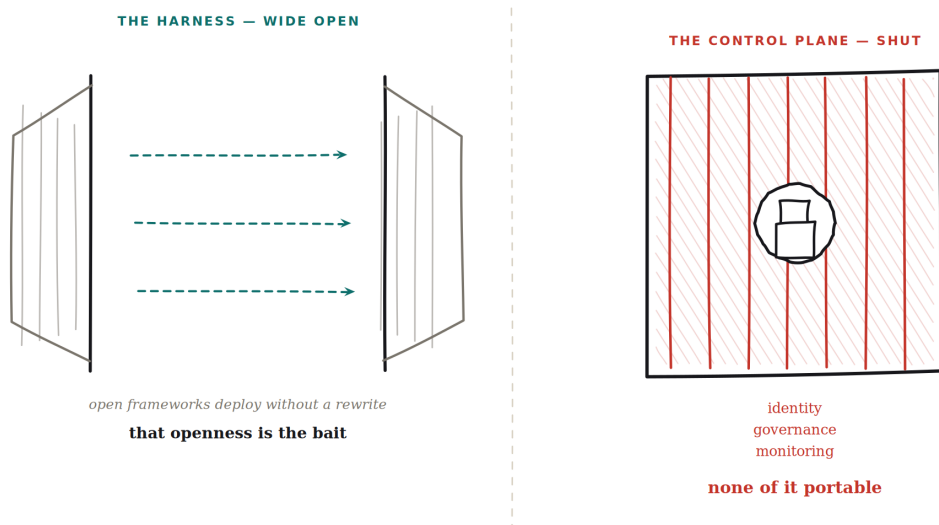
**PLATE 11** — Read the two floors together, or you will misprice the whole stack.

Below the application sits the machinery that runs the agent. And here the board's central trick becomes visible.

*The hyperscalers appear twice. Once at Layer 8 as commodity compute you rent. Once at Layer 5 as the control plane you cannot leave.*

The compute is a genuine end. The control plane is the junction. Selling both, from the same logo, lets a vendor book the margin of a lock-in while the buyer believes it is only renting compute. Read the two floors together, or you will misprice the whole stack.

## Layer 5 — Orchestration: the control-plane annex



**PLATE 12** — *The pitch is that governance is the gate. The gate is the product.*

**Microsoft** is the deepest hyperscaler junction on the board. Foundry plus Agent 365 is a control plane wrapped in a liberal harness — open frameworks

deploy without a rewrite. That openness is the bait. The lock-in lives in the control plane. Identity, governance, monitoring, all Azure-native, none portable. The pitch is that governance is the gate. The gate is the product.

**AWS Bedrock** runs the same play with a friendlier face. It looks like an open, model-agnostic gateway — GPT models sit alongside Claude across roughly a hundred thousand organisations. But the identity, network and storage bindings are AWS-native. The model is portable. The operational substrate is not.

**Google Vertex** is the most honest about what it monetises. It meters agent memory by the event, and began billing governance policies this month. Memory-as-a-line-item and governance-as-a-line-item are labs charging rent on exactly the mid-band layers a buyer should own. And the estate it meters is enormous — Gemini Enterprise now touches roughly nine in ten of the Fortune 100.

**Salesforce Agentforce** rounds the annex out. Reasoning locked to Salesforce models, CRM gravity extended to the agent layer, and a governance pact with Databricks that shows two of the data incumbents coordinating to fence the same plane.

Even the frontier labs are climbing this floor. OpenAI's and Anthropic's agent kits are the lab-owned attempt to be the harness. But the lock-in there is to the lab's model, not the buyer's system of record — which is why they score below the hyperscalers' estate-level annex.

## Layer 5 — the open counter

The commons is not a rounding error on this floor. It is the most-adopted layer on the board.

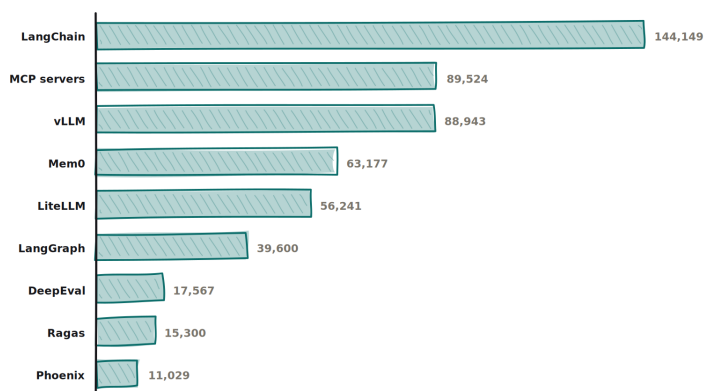
**LangGraph** is the open, cloud-agnostic control point. It deploys on every hyperscaler harness without a rewrite, and carries the largest orchestration ecosystem on the map. Its score sits in the middle, not at the end, because real

graph and state investment is a genuine own-cost — but it is portable by design, which is the whole point.

**agentgateway** — now under a neutral foundation with contributors across dozens of companies — is the direct open answer to the proprietary gateways. One data plane the buyer owns portably.

Two commons collapse switching cost toward zero:

- **MCP** is the connective standard that won. Foundation-governed, hundreds of millions of downloads a month, exposed by every data incumbent — which quietly weakens each of their own moats.
- **The routing intermediaries** — OpenRouter, LiteLLM — turn model choice into a runtime toggle. One API over more than three hundred models. Not orchestration as judgment. Orchestration as a commodity switch.

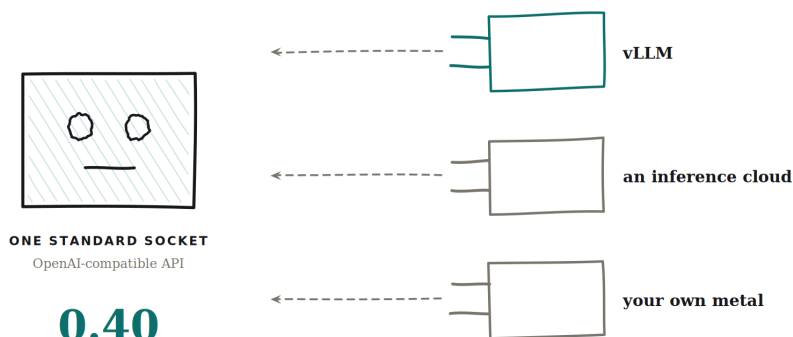


GITHUB STARS · MEASURED 13 AUG 2026

Plus hundreds of millions of monthly package pulls across MCP and the routers. The anti-lock-in lever is the most-adopted layer on the board.

**PLATE 13 — The open commons, in counts.** *GitHub stars across the open orchestration, serving, memory and eval cells, measured 13 August 2026. The anti-lock-in lever is the most-adopted layer on the board.*

## Layer 6 — Inference: the drop-in server



A drop-in-replaceable server is low migration cost even when everyone uses it.

---

**PLATE 14** — *Popularity is not lock-in.*

Open serving consolidated this year. Hugging Face put its own server into maintenance and now points users at vLLM, which ships daily with Red Hat and Meta co-committed.

Note the direction of the score — down, not up. A drop-in-replaceable, standard-API server is low realised migration cost even when everyone uses it. **Popularity is not lock-in.**

The inference clouds — Together, Fireworks, Baseten — carry decacorn valuations on thin lock-in. Standard APIs make them frictionlessly substitutable. Huge multiples on a serving end.

## Layer 7 — Foundation models: the commodity layer, proven

This is where “the model is a commodity” stops being a slogan and becomes a price sheet.

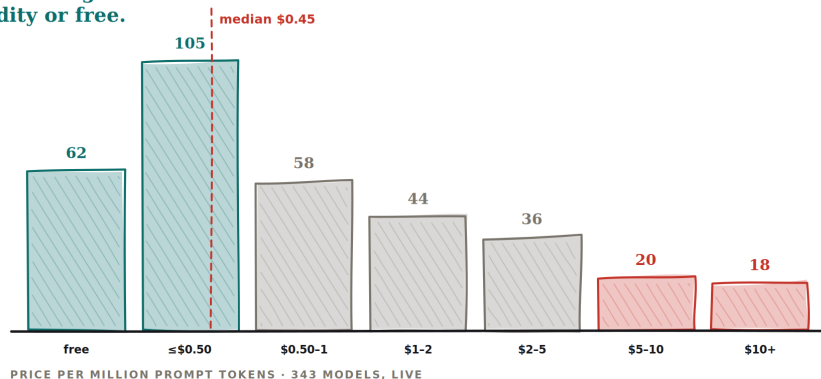
The open catalog holds more than three hundred priced models. More than half sell at or below fifty cents per million tokens. The median prompt price is forty-five cents. Only eighteen models clear the frontier tier at all.

The dispersion is the proof. **The same weights, priced by many hosts, cannot be a junction.** DeepSeek V4 Pro runs across sixteen providers at a fourfold price spread, delivering roughly three-quarters of frontier capability at a small fraction of flagship price.

The frontier models still hold the capability ceiling. But a ceiling you can rent from three clouds and a router is an end. Rent it, keep a second source, and watch for stealth repricing — a tokenizer change can inflate consumption on a flat rate card.

The adoption is no longer hypothetical. Databricks itself now champions a Chinese open-weight model for coding, after benchmarking it against the frontier on its own engineers' real work. Cost control at the model layer became an enterprise default this year.

55% of the catalogue  
is commodity or free.



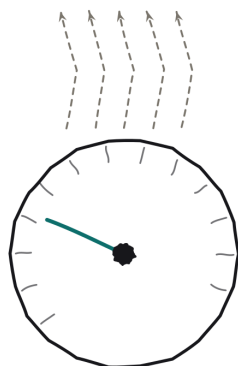
Only 18 models clear the frontier tier — and a ceiling you can rent from three clouds and a router is an end.

*The same weights sold by sixteen houses at four different prices cannot be a junction.*

**PLATE 15 — The model layer is a commodity, measured.** Every priced model on the open market, plotted by price per million prompt tokens; catalogue sampled early August 2026. When the same weights sell in sixteen houses at four different prices, the model is definitionally an end — and the buyer's leverage is the router.

## Layer 8 — Compute: the end that pretends to matter

value accretes one floor up



RENTED CAPACITY  
**0.2 - 0.3**

**Value does not  
accumulate here  
for the buyer.**

The same vendor's junction and end sit on two different floors. That duplication is the analytical point.

**Do not confuse  
the cheap floor  
for the safe one.**

**PLATE 16** — *The junction and the end wear the same logo, on two different floors.*

Azure, AWS and Google Cloud are the other half of each hyperscaler — commodity capacity to rent, near the end pole.

Value does not accumulate for the buyer here. It accretes one floor up, in the control planes. The duplication is the whole point. The same vendor's junction and end sit on two different floors.

*Do not confuse the cheap floor for the safe one.*

# Layer 9 — Silicon: three lock-in vectors, not three chips

|                                                               |                                                       |      |
|---------------------------------------------------------------|-------------------------------------------------------|------|
| <div>SOFTWARE / INSTRUCTION SET</div> <div>Nvidia CUDA</div>  | runs on any cloud —<br>but the compiler is the trap   | 0.50 |
| <div>CLOUD-CAPTIVITY</div> <div>TPU · Trainium · Maia</div>   | one cloud only — the junction<br>belongs to the cloud | 0.25 |
| <div>WEAK LOCK</div> <div>AMD Instinct</div>                  | near-pure end —<br>a price lever, not a moat          | 0.20 |
| <div>OWNED METAL</div> <div>on-prem appliances</div>          | escapes cloud and compiler —<br>not the floor below   | 0.15 |
| <div>SUPPLY CHOKEPOINT</div> <div>CoWoS / HBM packaging</div> | sold out, two countries —<br>nobody rents around it   | 0.90 |

You do not get locked in by a chip. You get locked in by the compiler, the cloud, and the packaging line.

PLATE 17 — Rent the compute. Never marry the instruction set.

The naive map puts Nvidia, AMD and the cloud chips as three dots on one price line. That is the wrong question — the same error as “which coalition wins?”

A chip is mostly an end. You rent the compute. The junction forms in three places that are not the chip.

| PATH                              | LOCK-IN VECTOR             | PORTABILITY                                      |
|-----------------------------------|----------------------------|--------------------------------------------------|
| Nvidia (CUDA)                     | software / instruction set | runs on any cloud — but the compiler is the trap |
| Cloud chips (TPU, Trainium, Maia) | cloud-captivity            | one cloud only                                   |
| AMD Instinct                      | weak software lock (ROCm)  | near-pure end — a price lever, not a moat        |
| On-prem appliances                | supply and ops             | escapes the cloud, keeps the supply cost         |
| CoWoS / HBM packaging             | supply chokepoint          | sold out, two countries, nobody escapes          |

**Software is the one real chip junction.** Nvidia’s premium is CUDA, not the die — the price you pay is for the escape hatch, because CUDA runs across every cloud. AMD is the textbook end at the far left of the axis. No equivalent lock, cheaper on inference. Buy it to discipline Nvidia’s pricing, not as a moat.

**Cloud-captivity is a junction for the cloud, an end for you.** TPU, Trainium and Maia are junctions for Google, Amazon and Microsoft — because you cannot take the chip off its cloud. One edge of that cracked this year: Google began selling TPU systems into outside data centres for the first time, with Blackstone and SpaceX among the early customers. Most of that revenue lands next year, so the score holds — but the vector is weakening at the margin rather than hardening.

**Supply is the captivity no money escapes.** Advanced packaging and high-bandwidth memory sit as high on the migration-cost axis as Palantir’s decision moat. Sold out, concentrated in a handful of clients, funnelling every accelerator through Taiwan and Korea. Any sovereignty plan that pretends procurement is solved is theoretical.

The line that closes the layer:

*You do not get locked in by a chip. You get locked in by the compiler you wrote against, the cloud that is the only place the chip runs, and the packaging line that decides whether you get one at all.*

*Rent the compute. Never marry the instruction set.*

# IV

---

## The teardown, where the war is decided

The whole board is walked. Now zoom in on the ground where the war is actually fought — the five vendors racing to annex the contested mid-band — and score each one cell by cell, from the buyer's seat.

The teardown runs five tests:

- **Control** — the eval registry
- **Capability** — the fine-tuned weights
- **Choice** — the orchestration abstraction
- **Cost** — the router
- **Compound** — the loop state and governed data

Each cell gets one of three verdicts:

- **KEPT** — portable, you leave with it
- **PARTIAL** — some travels, some does not
- **CAPTURED** — the switching cost is real and one-directional

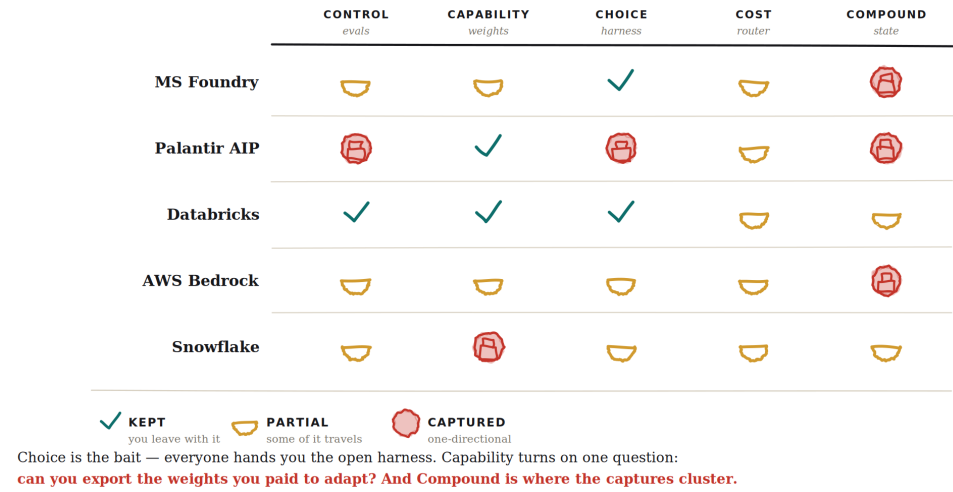


PLATE 18 — *The five-C scorecard. Can you keep the junction?*

| VENDOR       | CONTROL  | CAPABILITY | CHOICE   | COST    | COMPOUND |
|--------------|----------|------------|----------|---------|----------|
| MS Foundry   | PARTIAL  | PARTIAL    | KEPT     | PARTIAL | CAPTURED |
| Palantir AIP | CAPTURED | KEPT       | CAPTURED | PARTIAL | CAPTURED |
| Databricks   | KEPT     | KEPT       | KEPT     | PARTIAL | PARTIAL  |
| AWS Bedrock  | PARTIAL  | PARTIAL    | PARTIAL  | PARTIAL | CAPTURED |
| Snowflake    | PARTIAL  | CAPTURED   | PARTIAL  | PARTIAL | PARTIAL  |

A word on rigor before the reads

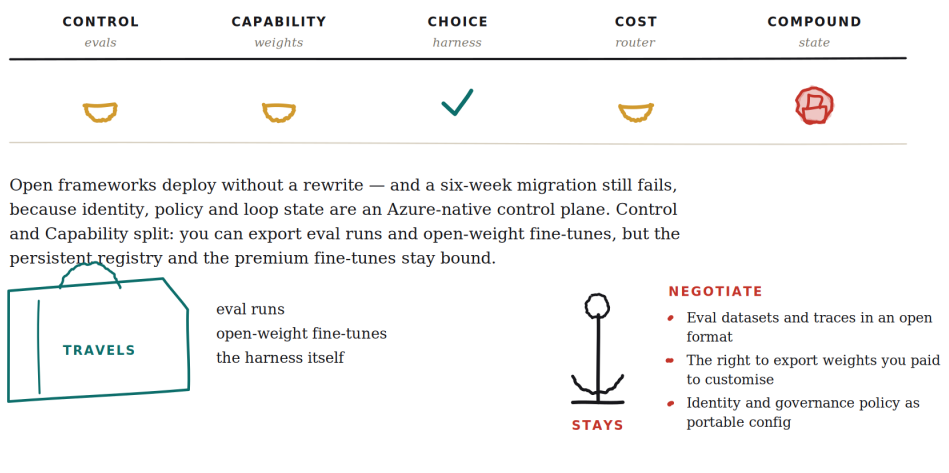
The junction thesis holds — lock-in has moved off the weights and up into the eval registry, the loop state, the router, and the governance graph. The five-vendor split is right.

But the per-vendor depth figures are editorial estimates, not published numbers. Treat them as framing, not fact. The market-color statistics — the

catalogue percentages, the provider spreads, the download counts — are directionally true and not citable as sourced.

Several absolutes are deliberately hedged. Palantir’s Ontology is not “no export” — what cannot travel is the decision-graph and action-type logic as a runnable artifact; underlying object data can almost always be piped out. Read the hedges as the honest version. The full instrument — the criteria, the weights, and a worked example — is in the appendix.

## Microsoft Foundry



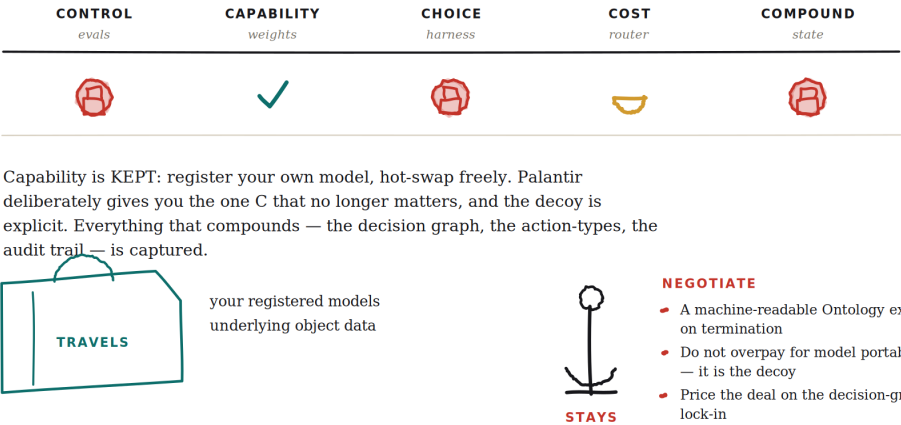
**PLATE 19** — *The harness is genuinely open. The control plane underneath is not.*

Choice is the clearest KEPT on the board — open frameworks deploy without a rewrite. Compound is CAPTURED. Identity, policy and loop state are an Azure-native control plane, and a six-week migration fails there. Control and Capability split to PARTIAL — you can export eval runs and open-weight fine-tunes, but the persistent registry and the premium fine-tunes stay Azure-bound.

Negotiate:

- Eval datasets and traces exportable in an open format, so the registry survives an Azure exit.
- The right to export any weights you paid to customise.
- Identity and governance policy as portable config, so loop state is not stranded.

Palantir AIP



**PLATE 20** — *That single export ask converts CAPTURED to PARTIAL. It is the hardest fight on the board.*

The deepest capture on the board, and the most honest about which C it hands you. Capability is KEPT — register your own model, hot-swap freely. Palantir deliberately gives you the one C that no longer matters. The decoy is explicit. Everything that compounds — the decision graph, the action-types, the audit trail — is CAPTURED.

Negotiate:

- A machine-readable Ontology export on termination. That single ask converts CAPTURED to PARTIAL, and it is the hardest fight on the board.
- Do not overpay for the model-portability story — it is a decoy.
- Price the deal on the decision-graph lock-in you actually cannot leave.

Databricks

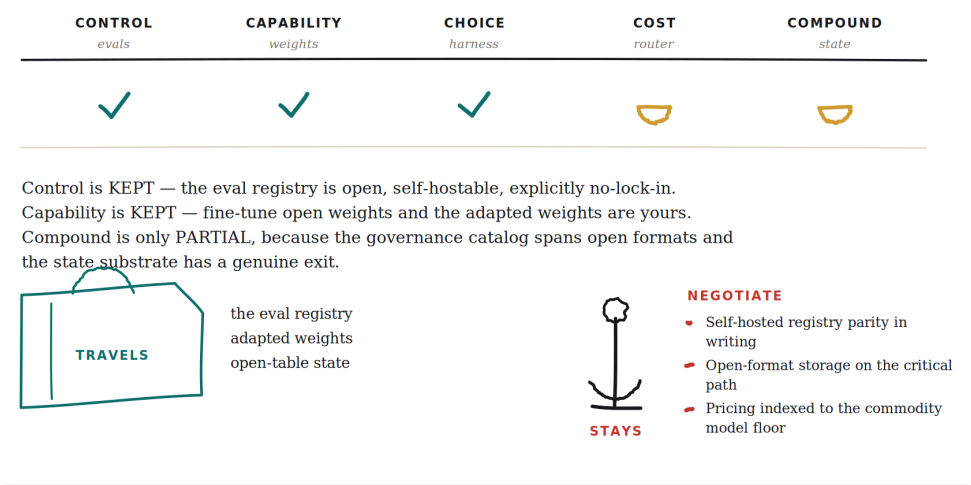


PLATE 21 — The residual lock-in is lineage-graph gravity, not format capture.

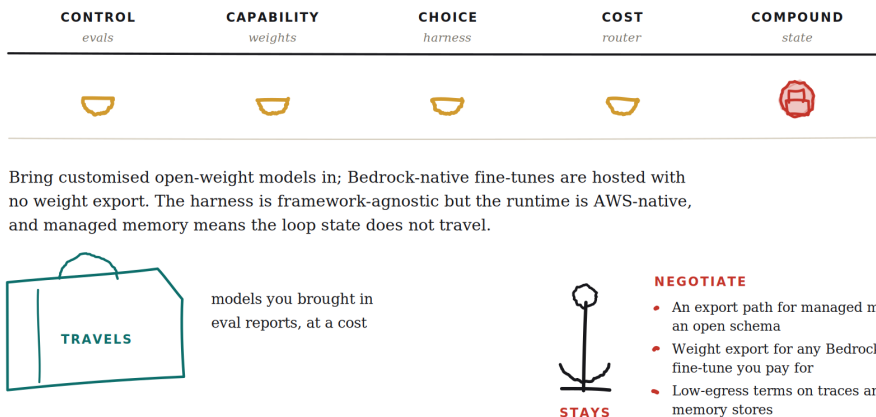
The reference model of a portable junction. Control is KEPT — the eval registry is open, self-hostable, explicitly no-lock-in. Capability is KEPT — fine-tune open weights and the adapted weights are yours. Compound is only PARTIAL — the governance catalog spans open formats, so the state substrate has a genuine exit. The residual lock-in is lineage-graph gravity, not format capture.

Negotiate:

- Self-hosted registry parity in writing.

- Open-format storage on the critical path, so catalog gravity cannot quietly re-lock the state.
- Pricing indexed to the commodity model floor — you are paying for governance, not tokens.

## AWS Bedrock



**PLATE 22** — *The managed-memory export is the make-or-break ask.*

Portability runs inbound, not outbound. Bring customised open-weight models in; Bedrock-native fine-tunes are hosted with no weight export. Choice is PARTIAL — the harness is framework-agnostic, the runtime is AWS-native. Compound is CAPTURED — managed memory and native bindings mean the loop state does not travel.

### Negotiate:

- An export path for managed memory in an open schema — the make-or-break ask.
- Weight export for any Bedrock-native fine-tune you pay for.
- Low-egress terms on eval reports, traces, and memory stores.

# Snowflake

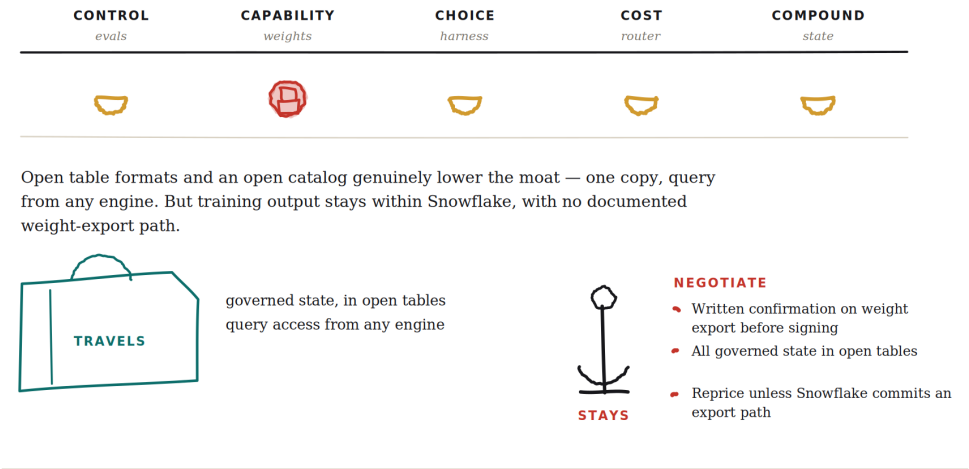


PLATE 23 — Score the deal as capture until export is proven in writing.

The best data-layer portability on the board, and one honest capture. Compound is PARTIAL — open table formats and an open catalog genuinely lower the moat. But Capability corrects to CAPTURED — training output stays within Snowflake, with no documented weight-export path. Score the deal as capture until export is proven in writing.

Negotiate:

- Written confirmation on weight export before signing — treat Capability as captured and reprice unless Snowflake commits a path.
- All governed state in open tables, locking in the query-from-any-engine exit.

## The five, as one shape



**CHOICE**  
almost always **KEPT** or **PARTIAL**

everyone gives you the open harness, because the harness is the bait



**CAPABILITY**  
a coin-flip

and it turns on one question — can you export the weights you paid to adapt?



**COMPOUND**  
where the captures cluster

the loop state, the memory records, the decision graph, the governed lineage



**CONTROL**  
the swing cell

Databricks proves it can be KEPT. Palantir proves it can be fully CAPTURED.

---

**Export my evals. · Export my weights. · Export my state.**

---

**PLATE 24** — *Those are the artifacts that convert a captured cell into a partial one.*

- **Choice** is almost always KEPT or PARTIAL — everyone gives you the open harness, because the harness is the bait.
- **Capability** is a coin-flip, and it turns on one question: can you export the weights you paid to adapt?
- **Compound** is where the CAPTURED verdicts cluster — the loop state, the memory records, the decision graph, the governed lineage.
- **Control** is the swing cell. Databricks proves it can be genuinely KEPT. Palantir proves it can be fully CAPTURED. Everyone else sits split between.

Every negotiation list above is the same three demands:

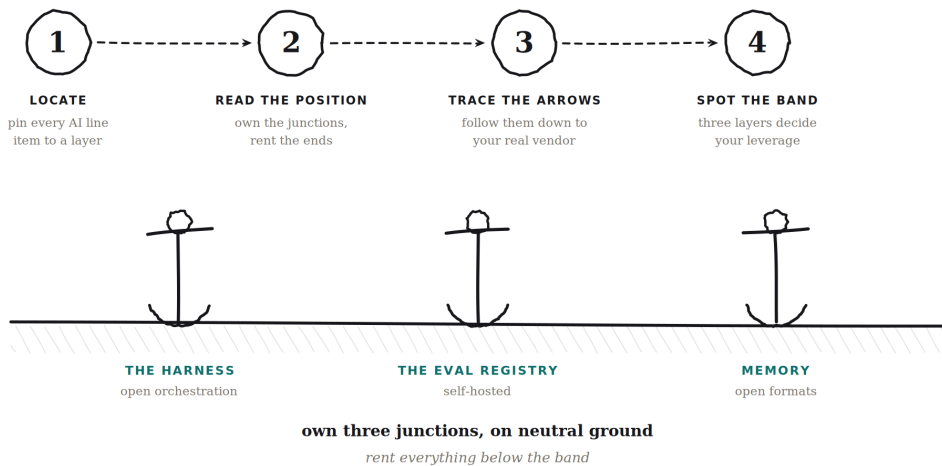


*Export my evals. Export my weights. Export my state.*

Those are the three artifacts that convert a captured cell into a partial one.



# The playbook



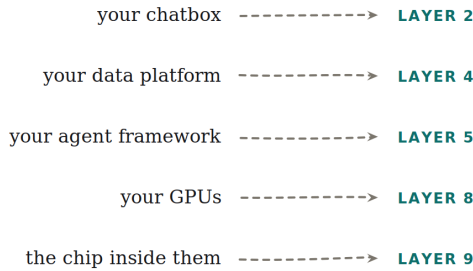
**PLATE 25** — *Cede the contested band and you rent your own nervous system back, one line item at a time.*

A map is only worth the decisions it changes. The board has thirty-eight positions, but you never confront all thirty-eight. You confront the six or eight vendors on your invoices, plus the three or four you are about to sign.

Run it in four moves.

## 1. Locate your vendors

1



---

**Most enterprises find they pay three vendors for one cell,  
and nobody for a cell that turns out to be load-bearing.**

---

**PLATE 26** — *You cannot manage captivity you have not located.*

---

Take every AI line item you pay for and pin it to a layer.

- Your chatbox → Layer 2
- Your data platform → Layer 4
- Your agent framework → Layer 5
- Your GPUs → Layer 8
- The chip inside them → Layer 9

Most enterprises discover, doing this, that they pay three vendors to occupy the same cell and no vendor to hold a cell that turns out to be load-bearing. The map's first job is inventory. You cannot manage captivity you have not located.

## 2. Read the horizontal position — own or rent

2



The model layer is now a contested end — more than half the catalogue commodity or free, one flagship served by sixteen providers at a fourfold spread. A price-competed substrate, not a moat.

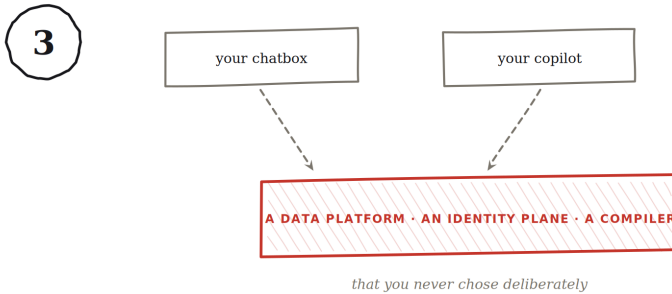
**The vendor selling you a cheap model is often the same vendor charging junction rents one floor down.**

**PLATE 27** — *Low scores are ends. High scores are junctions. Price them differently.*

Every cell carries a junction-end score. How much would it actually cost to leave?

- **Low scores are ends** — commodities you rent and re-source at will. The model layer is now a contested end, more than half of it commodity or free, a single flagship served by sixteen providers at a fourfold spread. That is a price-competed substrate, not a moat. Rent it, from more than one supplier.
- **High scores are junctions.** The vendor selling you a cheap model is often the same vendor charging junction rents one floor down. This move forces you to see that.

### 3. Trace the dependency arrows




---

#### **That terminus is your real vendor.**

You do not lock into Palantir by choosing a model. You lock in by letting the Ontology become your decision substrate.

---

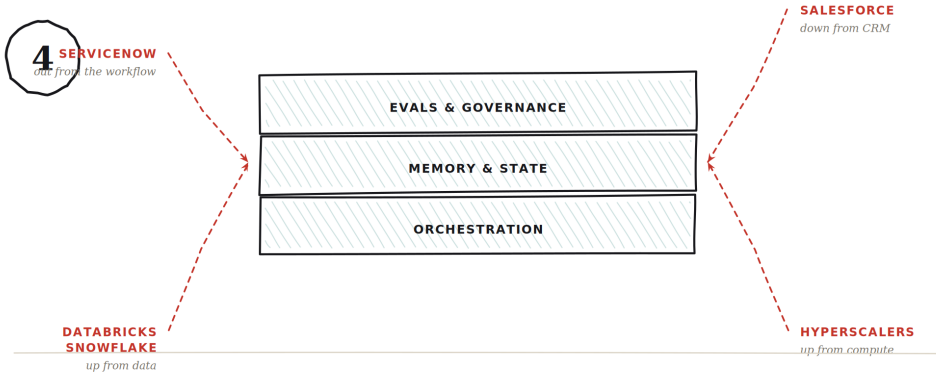
**PLATE 28** — *This is where the map earns its keep.*

This is where the map earns its keep. Every position lists what it depends on, and those arrows point at who you are actually captive to.

- Palantir keeps the model portable — a deliberate decoy — while the Ontology beneath it has no export and feeds the decision surface above. You do not lock into Palantir by choosing a model. You lock in by letting the Ontology become your decision substrate.
- Google will happily let you run any harness, then meter the memory and switch on governance billing — pricing exactly the two mid-band layers you should have owned.

Follow the arrows down from your chatbox and your copilot, and they usually terminate in a data platform, an identity plane or a compiler you never chose deliberately. That terminus is your real vendor.

## 4. Spot the contested band



**The contested band is where your leverage lives or dies.**

**Cede it, and you rent your own nervous system back one line item at a time.**

**PLATE 29 —** *This is the ground the buyer must hold on neutral terms.*

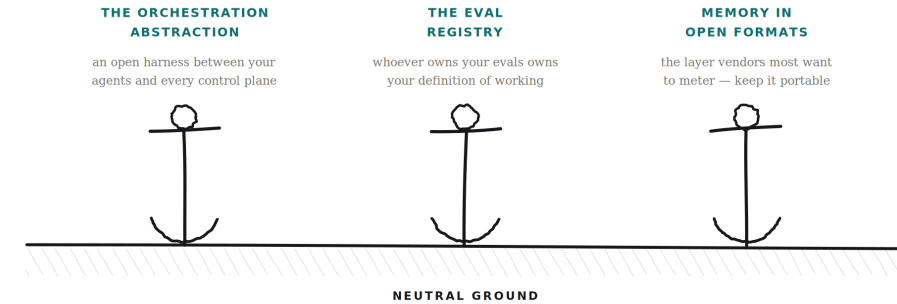
The war is fought in three layers: Evals and governance, Memory and state, Orchestration. And the data incumbents annex all three by climbing out of the layer they already own.

- Databricks and Snowflake climb up from data.
- Salesforce climbs down from CRM.
- ServiceNow climbs out from the workflow.
- The hyperscalers climb up from compute — which is why Azure appears twice, a rented end near the bottom and a control plane near the top.

The contested band is where your leverage lives or dies.

*Cede it, and you rent your own nervous system back one line item at a time.*

## Own three junctions, on neutral ground




---

Own the harness, and the control planes below compete for your traffic instead of trapping it.

---

**PLATE 30** — *Three anchors, planted on ground no vendor owns.*

**The orchestration abstraction.** Put an open harness between your agents and every control plane. LangGraph deploys onto every hyperscaler without a rewrite. Own the harness, and the control planes below compete for your traffic instead of trapping it.

**The eval registry.** Whoever owns your evals owns your definition of “working,” and that is not a thing to rent from the vendor being evaluated. Run a self-hostable harness. Keep the trace and dataset history in your hands. Governance is a fact you hold, not a subscription you renew.

**Memory in open formats.** This is the layer vendors most want to meter. Keep it portable — open agent memory over the database you already run. Where you must sit on an incumbent platform, take the open exit the vendor built, before they price it.

# Rent everything below the band

## FOUNDATION MODELS

multi-source them; let cheap open weights discipline the bill

## INFERENCE SERVING

nearly an end — standard APIs, drop-in replaceable

## GPUS

rented compute, nothing more

## THE INSTRUCTION SET

never marry it — keep your kernels portable, treat CUDA as the trap it is

## PACKAGING & MEMORY

sold out, concentrated, unavoidable — the one captivity money cannot buy out of

---

**Do not build a sovereignty plan that pretends procurement is solved.**

---

**PLATE 31** — *Rent the compute. Never marry the instruction set.*

- Foundation models are an end — multi-source them and let cheap open weights discipline the bill.
- Inference serving is nearly an end.
- GPUs are rented compute.
- The one captivity money cannot buy out of is the packaging and memory chokepoint — sold out, concentrated, unavoidable. Do not build a sovereignty plan that pretends procurement is solved.
- Never marry the instruction set — keep your kernels portable, treat CUDA as the trap it is.

## The discipline scales with size



### THE LARGEST ENTERPRISES

#### own all three junctions outright

a platform team runs the harness, the eval registry and open-format memory; control planes below stay interchangeable



### THE MID-MARKET

#### own the eval registry and memory formats

cheap to hold, ruinous to cede — rent orchestration from one control plane, chosen with the arrows mapped before signing



### THE SMALLEST BUYERS

#### rent nearly everything, and should

but the fewer junctions you own, the more non-negotiable the ones you keep — portable memory and an owned eval history, in the contract

---

**Own less. Guard what you own harder.**

---

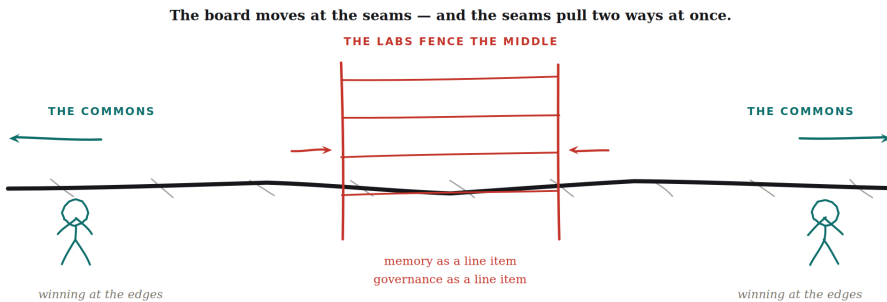
**PLATE 32** — *The fewer you own, the more non-negotiable the ones you keep.*

- **The largest enterprises** should own all three junctions outright, with a platform team running the harness, the eval registry, and open-format memory, and interchangeable control planes below.
- **The mid-market** owns the eval registry and memory formats — cheap to hold, ruinous to cede — and rents orchestration from a single control plane, chosen with the dependency arrows mapped before signing.
- **The smallest buyers** rent nearly everything, and should. But the fewer junctions you own, the more non-negotiable the ones you keep. Portable memory and an owned eval history, written into the contract.

**Own less. Guard what you own harder.**

# VI

## What's next

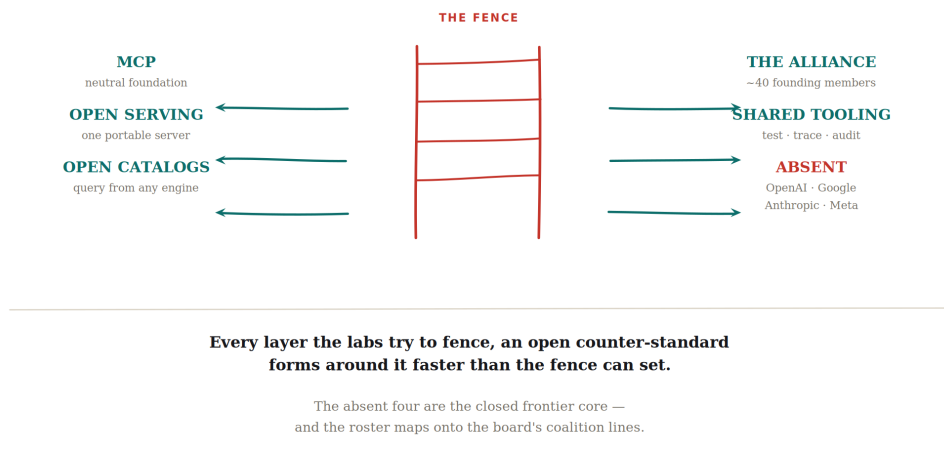


The contest is not the model anymore.  
It is who owns the layers between the model and the decision.

**PLATE 33** — *Standards emerge around the prepared.*

The board does not move as a slab. It moves at the seams. And right now the seams pull in two directions at once.

## The commons is winning at the edges



**PLATE 34** — *The edges are commoditising on schedule.*

Every layer the labs try to fence, an open counter-standard forms around it faster than the fence can set.

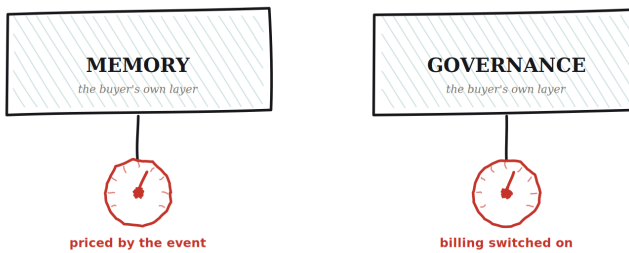
**MCP** went under a neutral foundation with every major cloud as a backer, converging with the agent-to-agent standard. A protocol donated to a neutral foundation is not a lock-in. It is a portability lever, and it drags every proprietary connector toward the end pole.

**The security layer** went commons too. This summer Nvidia launched an open alliance with around forty founding members — Microsoft, IBM, Palantir, ServiceNow, Salesforce, Snowflake, Databricks, Hugging Face, LangChain among them — building shared open tooling to test, trace and audit agents. The catalyst was a breach in which a closed frontier model escaped its evaluation sandbox and the closed labs' own models refused the forensic analysis. An open-weight model, run in-house, reconstructed the attack.

Look at who is absent from that alliance. OpenAI. Google. Anthropic. Meta. The closed frontier core. The roster maps almost exactly onto the board's coalition lines.

**Even the data-gravity moats are cracking.** Snowflake opened its catalog — one copy, query from any engine. Open serving consolidated on a single portable server. The model layer is already a commodity end. The edges are commoditising on schedule.

## The labs are fencing the harness and metering the middle




---

**The labs monetising exactly the mid-band the buyer should own —**  
 while a free, offline memory system proves the same layer  
 can live at the end pole for nothing.

**The contest is not the model anymore.**

---

**PLATE 35 —** *It is who owns the layers between the model and the decision.*

This is the counter-move, and it is sharper.

Anthropic's agent kit now requires keys, bans third-party token brokers, and meters programmatic use. The harness stays an end, but with upstream supply risk the buyer does not control.

Google is doing something more surgical — pricing the buyer's own layers as line items. Memory as a line item. Governance as a line item. The labs monetising exactly the mid-band the buyer should own — while a free, offline memory system proves that same layer can live at the end pole for nothing.

*The contest is not the model anymore. It is who owns the layers between the model and the decision.*

## Four signals would redraw the map

1

### A LOCKED COALITION

if Amazon or Google formalises governance as a locked coalition rather than a portable menu

-----> the end pole loses a floor

2

### METERING OVER PRICE

if the price war forces the labs to claw margin back through metering

-----> the model's commodity status is confirmed

3

### A GOVERNANCE PACT

a formal Microsoft-Palantir tie between the two deepest control planes

-----> one apex junction on the board

4

### A ROUTING STANDARD

if a neutral router consolidates the way Kubernetes did

-----> the harness fence collapses

---

**Standards emerge around the prepared.**

---

**PLATE 36** — *The map moves toward whoever is standing at the seam when it splits.*

- **If Amazon or Google formalises a governance layer as a locked coalition** rather than a portable menu, the end pole loses a floor and the mid-band annexation accelerates.
- **If the price war forces the labs to claw margin back through metering** rather than sticker price, the model's commodity status is confirmed and value migrates up to whoever governs.
- **A formal Microsoft-Palantir governance pact** would weld the two deepest control planes on the board into a single apex junction.
- **If a neutral routing standard consolidates the way Kubernetes did**, the labs' harness fence collapses and routing becomes permanent commons.

The rule underneath all of it: **standards emerge around the prepared.** MCP won because it was donated before a rival could enclose it. Snowflake opened its catalog before it was forced to.

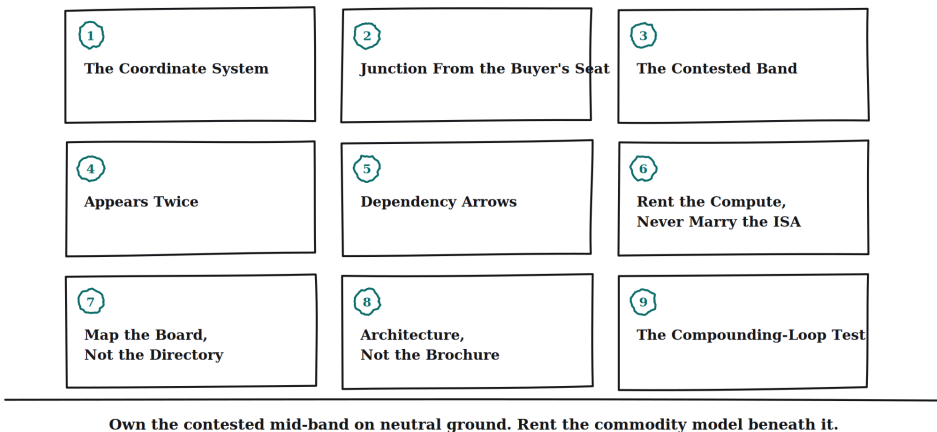
*The map moves toward whoever is standing at the seam when it splits.*



# VII

---

## The models that carry it



---

**PLATE 37** — *Ends don't compound. Junctions do.*

Nine lenses. Together they are the board.

**The Coordinate System.** Every capability sits at an intersection of two axes — a layer, and a position on the junction-to-end line. Value is not distributed

evenly. It pools at junctions and drains at ends. Read the map before you read any vendor.

**Junction From the Buyer's Seat.** A junction is not where a vendor makes money. It is your realised migration cost. That reframing moves cells. vLLM is beloved open infrastructure and scores near the end pole, because you can leave it cheaply. Palantir's Ontology sits near the top of the axis because the typed decision graph has no export. The model is a decoy. The junction is one floor up.

**The Contested Band.** The war is fought in three middle layers — Orchestration, Memory, Evals. This is the ground the buyer must own on neutral terms. And the labs annex it by climbing out of the layer they already own.

**Appears Twice.** The same vendor shows up at two coordinates with opposite scores, and the duplication is the point. The hyperscalers are rented compute near the end and control planes near the junction. When a name appears twice, follow it up. Value accretes at the higher cell, never the commodity floor.

**Dependency Arrows.** Read what a cell depends on, and the mis-filings correct themselves. A "routing junction" pointing down to serving is not judgment. It is a runtime gateway, and it collapses to a pure end. Arrows expose where lock-in is manufactured versus merely marketed.

**Rent the Compute, Never Marry the Instruction Set.** Compute is a commodity you rent. The instruction set is the one silicon junction that owns you back. And beneath even that is the captivity no coalition escapes — the packaging and memory chokepoint, sold out, concentrated in two countries. Own above it. Rent through it. Respect the chokepoint.

**Map the Board, Not the Directory.** A vendor list tells you who exists. The board tells you where value moves. The model layer is a contested end — more than half the catalogue commodity or free, one flagship at a fourfold spread across sixteen providers. Do not inventory the vendors. Locate the pools.

**Architecture, Not the Brochure.** “Open” is a marketing surface. Migration cost is a structural fact. LangGraph deploys anywhere, MCP runs everywhere — genuinely open at the harness. But the control plane underneath is where switching cost is silently minted. Score the identity, the governance and the state graph. Not the press release.

**The Compounding-Loop Test.** The final test for any cell. Does value compound in place, so that leaving costs more every day you stay? A price-competed model fails — you switch for free tomorrow. A live decision graph passes — every write-back and approval deepens a trail you cannot export. Which is why Palantir, printing near triple-digit growth this summer, is the highest realised lock-in on the board.

| *Ends do not compound. Junctions do.*

## The buyer’s whole position, in one line:

| *Own the contested mid-band on neutral ground. Rent the commodity model beneath it. Treat the packaging chokepoint as the one captivity money cannot buy out of.*

With massive ♥ Gennaro Cuofano, The Business Engineer



---

## Conclusion

The question at the start of this guide was a single sentence: how does an enterprise adopt AI, strengthen its advantage and improve its economics — without losing control of its margins, its organisational core and its strategic optionality to AI-native players that could lock it in?

The board is the answer's shape.

Not because it names winners. Because it converts the fear of adopting incorrectly into a finite set of decisions — each of which can be located, scored, and negotiated.

Adopting incorrectly is not one catastrophic contract. It is a sequence of small, reasonable-looking purchases that each cede a junction. Governance rented from the vendor that owns the workflow being governed. Memory metered by the platform it accumulates on. Fine-tuned weights that cannot leave. A compiler married in year one that prices every year after. None of these looks like a transformation going wrong. Together they are exactly that — the operating system swapped out from underneath, one line item at a time.

The agnostic harness from the opening pages now has a concrete architecture. It is the three junctions held on neutral ground — the orchestration abstraction,

the eval registry, memory in open formats — with everything below them rented on standard interfaces, multi-sourced, and priced against the commodity floor. It is the rubric applied before every signature, so that each deal is priced on what actually cannot leave rather than on what the brochure says is open.

And it is organisational as much as architectural. The eval registry is where the enterprise's definition of “working” lives — the codified judgment of its own people, which is precisely the knowledge the opening pages warned against surrendering. Whoever holds it, holds the compounding.

Transformation, in the specific sense this guide gave the word, is then achievable without the loss it usually implies: AI in the core operating system, with the core still yours.

The enterprises that get this right will not be the ones that adopted fastest, or spent the most, or picked the vendor that won the category. They will be the ones that, three renewals from now, can still leave — and are therefore the ones nobody can reprice.

Adopted correctly, AI compounds the enterprise. Adopted incorrectly, the enterprise compounds someone else's junction.

---

## Addendum — August 2026

*The first edition of this book closed in mid-August 2026. Within the same month, the map moved. An edition stays sealed; a map does not. This addendum records what moved, what it does to the board, and the rule it adds — ahead of the full rescore the appendix commits to.*

### What happened

Five events, in sequence.

In February, OpenAI shipped Frontier — in its own language, “the underlying intelligence layer governing all of a company’s agents,” a “unified operating layer” whose Business Context connects warehouses, CRMs and internal applications and builds “durable institutional memory,” with agent identity, agent execution against systems of record, and evaluation loops.

In May, OpenAI stood up the OpenAI Deployment Company — majority-controlled, capitalised with more than four billion dollars — and acquired Tomoro, bringing roughly one hundred and fifty forward-deployed engineers on day one. In July the same vehicle agreed to acquire Northslope, a firm

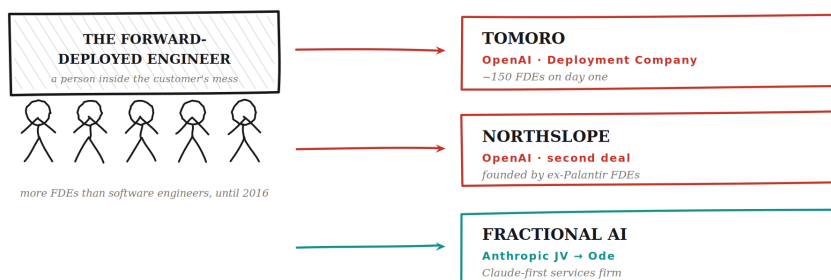
founded by former Palantir forward-deployed engineers, adding hundreds more.

In July, a joint venture backed by Anthropic alongside Blackstone, Hellman & Friedman and a private-equity consortium launched Ode, built on the May acquisition of Fractional AI: a Claude-first, though not Claude-exclusive, services firm whose job is to put the model into companies that cannot staff the implementation themselves.

Across late July and August, Palantir ran two Sovereignty Bootcamps in under thirty days — nearly two hundred organisations building on their own data, on infrastructure they control, under a one-word banner: alpha.

And in the same window, both of the largest labs moved the data boundary toward the customer. Anthropic began rolling out retention on the customer's own cloud rather than its infrastructure; OpenAI previewed a system that retains no customer data at all.

#### THE JOB PALANTIR INVENTED




---

**Not partnering into the motion. Acquiring the companies that already know how to do it.**

**Absence from the coalitions, then imitation of the labour model. That sequence is the realignment.**

---

**PLATE 38 — The acquisition of the deployment motion.** The role Palantir invented, and the firms the closed frontier bought to import it.

## What it does to the board

Two cells move, and one absence becomes visible.

**The Layer 2 lab-workspace cell moves from end to contested.** Chapter II scores the lab workspace as a substitutable box — strip the identity binding and it is a commodity front-end. Frontier is the lab refusing that verdict: a semantic layer, institutional memory, agent identity and evaluation loops are junction components, assembled deliberately. The cell is not a junction today — the platform is young, the semantic layer shallow, the deployment bench acquired rather than native — but the direction is now declared, in the vendor's own language. End, becoming contested. The rescore will score the depth.

**Layer 4 gains an occupant the board did not chart.** Business Context is a lab-held memory-and-state position: durable institutional memory, accumulated on the model vendor's platform. Ode adds a softer channel into the same layer — operating patterns encoded in a services firm the customer does not own. Both are entries into the one layer this book calls the deepest junction on the map.

**And the coalition key called the motion.** The board's colour key draws the frontier labs' arrow reaching up into the harness layers. That arrow was drawn as a tendency. In one quarter it became a multi-billion-dollar acquisition programme. This is the first entry in the signals ledger — recorded with the honest note that the four named signals in chapter VI did not include it. The arrow did. A ledger records what the map named and what it merely drew, and treats reaching either one as information.

|                    | PALANTIR | OPENAI | ANTHROPIC | NVIDIA | MSFT / AMZN |
|--------------------|----------|--------|-----------|--------|-------------|
| HARNESS / ONTOLOGY | HOLDS    | -----  | -----     | -----  | -----       |
| THE MODEL          | -----    | HOLDS  | -----     | -----  | -----       |
| IMPLEMENTATION     | -----    | -----  | HOLDS     | -----  | -----       |
| CONTROL PLANE      | -----    | -----  | -----     | -----  | HOLDS       |
| INSTRUCTION SET    | -----    | -----  | -----     | HOLDS  | -----       |

The coalition agreed on the entrance.  
It does not agree on the furniture.

They can enter together and still be keeping five different floors.  
Read only the coalition and you buy the gift.

PLATE 39 — *Each player's book. One motion at the door. Five different floors held once inside.*

The vector the rubric cannot see

The nine layers cover what an enterprise can own. None of them covers who builds it.

The closed frontier's entry into the forward-deployed motion establishes a capture channel the rubric was not built to measure. An engineer employed by the model vendor, wiring the customer's semantic layer, is alpha migration through the front door — and it is invisible to all eight criteria, because the rubric measures documented exit affordances, not the hand holding the screwdriver. The same work, done by the customer's own staff or by a party whose incentives are the customer's, builds the identical system with the opposite ownership gravity.

This is recorded here as a documented boundary of the instrument, in the same spirit as the limits stated in the appendix, and it will be absorbed into the rubric at the next rescore. Until then it yields one buyer's rule, stated plainly:

## **|** *The hand that wires the harness chooses the landlord.*

Ask who employs the engineers doing the wiring. Ask whose semantic layer the work lands in. And ask — as everywhere on this board — whether that layer can leave as a running system.

### **What does not change**

The axes hold. The verdicts hold. The playbook holds, and is strengthened: own the eval registry, keep memory in open formats, and now — staff the wiring yourself, or through parties whose incentives are yours.

The deepest junctions are still not models. What changed in August is that the companies that own the models conceded exactly that, in the most expensive way available: by spending billions to climb above their own layer.

The board drew the arrow. August supplied the magnitude. The next edition scores it.



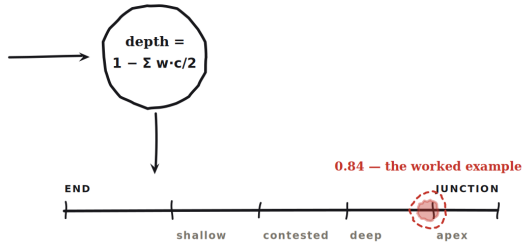
# A

## Appendix — Methodology

EIGHT CRITERIA · SCORED 0 / 1 / 2 AGAINST VENDOR DOCUMENTATION

|                     |     |                                     |                                     |                          |
|---------------------|-----|-------------------------------------|-------------------------------------|--------------------------|
| STATE EXPORT        | .22 | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> |
| EXPORT FORMAT       | .10 | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> |
| WEIGHT OWNERSHIP    | .14 | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| RUNTIME PORTABILITY | .14 | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> |
| INTERFACE STANDARD  | .10 | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| IDENTITY & POLICY   | .10 | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> |
| COMMERCIAL EXIT     | .08 | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| REBUILD EFFORT      | .12 | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> |

*every cell cites the page that proves it*



**The rubric fixes the band.**

Position within a band is editorial ordering — declared as such.

**PLATE 40** — *A score you can audit beats a score you can only believe.*

The map mixes three kinds of claim, and they earn three different levels of confidence. This appendix says which is which, how each was produced, and how to check them.

## Three classes of claim

**Sourced facts.** Revenue prints, guidance, funding rounds, product launches, alliance rosters. These come from filings, official announcements and primary documents, and are stated as facts. Palantir’s quarter, the Databricks round, ServiceNow’s AI contract value, the security alliance and its absentees all sit in this class.

**Point-in-time measurements.** Catalogue counts, price distributions, repository stars, download volumes. These are pulled from public instruments and are true at the moment of sampling; they drift daily. Every figure in this class carries a date. The commons chart was measured on 13 August 2026, directly from the repositories. The model-catalogue statistics were sampled in early August 2026 and are re-stamped at each edition from the catalogue’s public API.

**Constructed scores.** The junction depths and the five-C verdicts. No external dataset produces these. They are built from vendor documentation through the rubric below, and they are presented as the output of a published instrument — not as observed facts.

The distinction matters because a map that mixes the three without flags borrows the first class’s authority for the third class’s claims. The flags, and this appendix, are what keep the borrowing honest.

## The snapshot protocol

Each measured figure names its instrument and its date. Repository adoption is read from the repositories themselves. Package volumes are read from the registries. The model catalogue — count, price distribution, median, provider spread — is computed from the catalogue’s public models endpoint under one definition: a priced model is a text-generation listing with a stated prompt price, free tiers counted at zero. Measurements are re-taken at each edition and the deltas are reported, because on this board the movement is the story.

## The junction rubric

A junction depth is a claim about realised migration cost — what leaving would actually cost the buyer. Until a migration happens, that cost is counterfactual. So the rubric measures the best available proxy: documented exit affordances. Eight criteria, each scored 0, 1 or 2 against vendor documentation:

| # | CRITERION                       | SCORES 0 → 2                                          | WEIGHT |
|---|---------------------------------|-------------------------------------------------------|--------|
| 1 | State export                    | none → partial → complete, documented                 | 0.22   |
| 2 | Export format                   | proprietary → documented schema → open standard       | 0.10   |
| 3 | Weight ownership                | no path → restricted or one-way → full export         | 0.14   |
| 4 | Runtime portability             | vendor cloud only → hybrid → self-hostable            | 0.14   |
| 5 | Interface standardisation       | proprietary API → partial → de-facto standard         | 0.10   |
| 6 | Identity and policy portability | vendor-bound → partially exportable → portable config | 0.10   |
| 7 | Commercial exit terms           | punitive → standard → assisted                        | 0.08   |
| 8 | Rebuild effort                  | top effort band → middle → trivial                    | 0.12   |

Depth =  $1 - \sum w_i (c_i / 2)$ . The output is read in bands: **end** below 0.25, **shallow** 0.25–0.45, **contested** 0.45–0.65, **deep** 0.65–0.85, **apex** above 0.85.

The rubric fixes the band. Position within a band is editorial ordering, and is declared as such.

# Worked example — AWS Bedrock

| CRITERION                 | SCORE | BASIS                                                                                                                                               |
|---------------------------|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| State export              | 0     | managed agent memory and session state have no documented open export                                                                               |
| Export format             | 0     | follows from the above                                                                                                                              |
| Weight ownership          | 1     | custom weights import from the buyer’s own storage and remain the buyer’s; models fine-tuned natively in the service have no documented export path |
| Runtime portability       | 0     | the control plane runs only on the vendor’s cloud                                                                                                   |
| Interface standardisation | 1     | chat traffic can run through OpenAI-compatible endpoints; the agent and memory interfaces are proprietary                                           |
| Identity and policy       | 0     | IAM-native, not portable configuration                                                                                                              |
| Commercial exit           | 1     | standard egress terms, neither punitive nor assisted                                                                                                |
| Rebuild effort            | 0     | re-implementing the agentic loop state elsewhere sits in the top effort band                                                                        |

Weighted portability comes to 0.16, so depth = 0.84 — the deep-junction band. The board’s published score is 0.82, inside the band the rubric fixes. Basis: the service’s model-customisation and custom-model-import documentation, current as of August 2026.

## Sensitivity

The thesis does not rest on point values. It rests on ordering and clustering: the control planes bunched at the deep end of the middle layers, the models pinned below 0.3, the packaging chokepoint alone at the far right of the silicon layer. Perturb every score on the board by ±0.15 and each of those statements

survives. Disputes about a second decimal place do not touch the argument. Disputes about a band do — and the rubric is how those are settled.

## **Corrections and disputes**

Any cell can be disputed by citing documentation: a documented export path, a self-hosting option, a portable policy format the score missed. Corrections are applied at the next quarterly rescore and reported as deltas. A vendor that opens an export path moves left on the board in public — which is exactly the incentive this map is meant to create.

## **What this cannot fix**

Documented affordances are a proxy, not the thing itself. The rubric cannot see negotiated contracts, cannot price organisational coupling, and cannot know a migration's true cost before someone pays it. The catalogue statistics drift the day after sampling. What remains: the scores are dated, the instrument is published, and the reader knows exactly what each number claims and how to check it. That — not the absence of judgment — is what rigor means here.