VERTICAL-DRIVEN ARCHITECTURES · SUPPLY CHAIN & LOGISTICS · DESIGNED WITH PRAXIS · NOVEMBER 2026

Sovereign AI Architecture for a Freight Corridor and Customs Clearance in Saudi Arabia

Corridor Command: Industrial AI for the Whole Consignment Story

Muhammad Umar, Cofounder, CodeNinja

For the corridor and customs lead, the corridor planners and yard controllers in the loop, and the platform, data and reliability engineers who would build and run it.

Vertical-Driven Architectures is a CodeNinja series of system designs. Every design in the series is driven by a real-world problem and scenario in a single industry, and every one is designed on Praxis, CodeNinja's platform for designing physical AI systems. Operations are described by class, never by name.

ABSTRACT

Where a Box Is Should Not Take Six Systems to Answer

A sovereign AI architecture for a freight corridor in Saudi Arabia starts from the one question the operation needs answered continuously: where is this consignment, and when will it clear. It cannot be answered today because the answer lives in six systems of record carrying six identifiers, and it is assembled by hand by a planner whenever a consignee asks. Corridors of more than two thousand kilometers mix rail legs with road legs run by contracted hauliers, each consignment crossing the port terminal, the customs authority's clearance system, the operator's rail yard, a haulier, the dry port and the consignee, while the authority's single declaration transit with pre clearance and smart inspection now runs on some lanes of a national truck fleet of around half a million vehicles. Dwell at the dry port grows when a declaration stalls, and the stall is learned from the consignee rather than from the systems; a late road leg is visible in telematics but never joined to the train it will miss.

The design is a sovereign corridor and customs system built for a national freight and logistics operator in Saudi Arabia: seven read-only source systems, two adapter families covering integration and telemetry,

fifteen objects in one governed object model, seven services covering governance, baselines, the ontology, agents and approvals, the sovereign platform, memory and the command center, and three surfaces consisting of the command center, the connection view and the answer queue. Every part runs on the owner's own central node in its in-country data center, on open weights the owner holds, with no edge inference and no third party model service, so cargo, consignee and declaration data never leave the country. The model register carries no committed weights on this plan: no open weight model survived this run's live verification, so the choice of weights is recorded open work with its sizing line, the inference cluster class (GPU nodes for 300B to 1T parameter open weight models), held ready rather than fixed.

The paper covers the industry problem and the join failure across the six systems, the constraints and the scoping decisions they force, the layered stack and the fifteen-object model of the consignment story, the adapter tier and event backbone that feed it, the placement of all reasoning on the owner's central node, the model register and the licenses that decide what the owner can own, the three phases with their item counts and gates, and who owns the intelligence the system produces, closing with the chapter on how Praxis contextualized and reasoned the design.

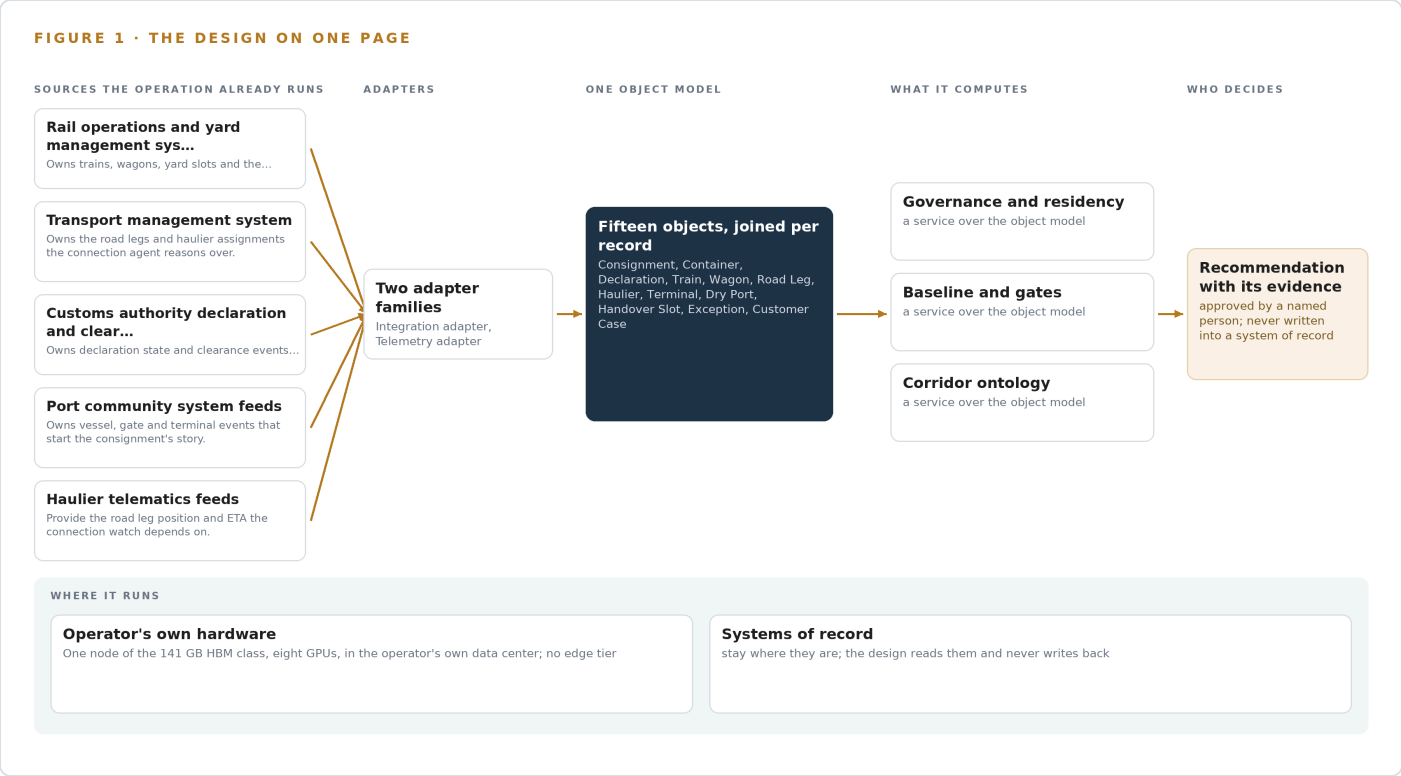


Figure 1. Corridor Command on one page: the sources the operation already runs, one object model, what it computes, and the person who decides.

Contents

Each chapter is tagged for the reader it serves most directly: Executive, Team Lead, FDE, Reference.

	Abstract · Where a Box Is Should Not Take Six Systems to Answer	Executive
PART I · THE PROBLEM		
1	Industrial AI Starts with the Consignment, Not the Systems	Executive
2	Six Identifiers Hide One Consignment From Every Planner	Executive · Team Lead
PART II · THE DESIGN		
3	Sovereignty and Six Systems Shape Everything Downstream	Team Lead
4	One Stack Runs From Records to Reasoning In-Country	Team Lead · FDE
5	Fifteen Objects Turn Corridor Data Into One Story	FDE
6	Every Source Enters Through an Adapter, Never Directly	FDE
7	All Reasoning Runs on the Operator's Own Node	FDE
8	The Register Holds No Weights Until Verification Stands	FDE · Executive
PART III · THE ROLLOUT		
9	A Measured Quarter Gates Every Corridor Extension	Team Lead · Executive
10	The Intelligence Stays with the Corridor That Produced It	Executive
PART IV · HOW IT WAS DESIGNED		
	Conclusion · One Projection, One Memory, One Answer, Held Sovereignly	Executive
11	How Praxis Contextualized and Reasoned This Design	Team Lead · FDE
	Sources	Reference

PART I · CHAPTER 1

Industrial AI Starts with the Consignment, Not the Systems

The corridor's cost is not a missing system but a missing object: the consignment no single system holds, in a country whose customs programs and new corridors are moving faster than the operators' ability to see one box end to end.

The abstract put the whole sovereign AI architecture on one page: seven sources of record, two adapter families, fifteen objects, seven services and three surfaces, all inside the owner's own hardware in the country. This chapter begins where industrial AI has to begin, with the question the operation cannot

answer today, the documented cost of that silence, and the scenario the design must fit without replacing anything.

1.1 The Question, the Data and the Regimes

The question the operation needs answered is narrow and constant: where is this container, and when will it clear? The answer requires one key and every story told about it. The container number carries the whole join, and around it sit the vessel, gate and terminal events that open the consignment's story; the declaration with its lane, broker and clearance state; the train, its wagons and the handover slot; the road leg with its haulier and its ETA; the yard state at the destination dry port; and the case log of questions already asked and answered. The question must also be answered inside four regimes at once. The customs authority now offers single declaration transit with pre clearance and smart inspection on some lanes, which the design reads but never acts in. The national data protection regime fixes residency, cross-border transfer bases and the data protection officer's records of processing duties. The national AI authority's ethics risk tiers govern how the reasoning tier is classified. The road transport law of 2025 links the haulier fleet to the authority's electronic platform, which decides which telematics can be relied upon.

1.2 The Documented Cost

The cost of not answering lands in three places: planner hours spent assembling an answer by hand from six identifiers, dwell hours that grow at the dry port while a stalled declaration goes unnoticed, and the same questions answered week after week for the authority and for cargo owners. The exposure is documented at industry scale. Cross-border data transfers are integral to international supply chains, which depend on information moving across borders for sourcing, finance, logistics and risk management (Globaldataalliance n.d.), and third-party breaches doubled as a share of all security incidents in the most recent year, with regulators across Europe and the United States now holding leadership personally accountable for vendor risk (Cybelangel 2026). When an operator's data flows stop, the physical operation stops with them: a ransomware attack on a logistics company in October 2025 encrypted its data and caused a large-scale service stoppage, with leakage of the information it held confirmed by the company's own investigation (Askullogist 2025). An operator that cannot see one consignment across its systems cannot even rank which flows it depends on.

1.3 The Operation as a Scenario

The owner runs inland corridors linking seaports to dry ports and freight yards by rail and road, with corridors of more than two thousand kilometers mixing rail legs with road legs run by contracted hauliers, and new corridors opened in 2026. The national truck fleet it draws on is around half a million vehicles. Each consignment crosses several hands: the port terminal, the customs clearance system, the rail yard, a haulier, the dry port and the consignee. Three roles sit in the loop by name: corridor planners who own train and slot decisions on shift, yard controllers who hold re booking authority at the dry ports, and service desk agents who work the case queues. The physical environments are the seaport terminals, the rail corridors, the yards and dry ports inland, and the road network between them. The regulatory ground

is the national data protection regime, the AI authority's ethics tiers, the road transport law of 2025 and the customs authority's clearance programs. The estate holds seven named systems of record, three named roles and four classes of place, and every system is read, never replaced.

PART I · CHAPTER 2

Six Identifiers Hide One Consignment From Every Planner

Each system of record sees one slice of the consignment's story and none can answer the question they collectively hold, which is why dwell grows silently and knowledge of stalling lanes, brokers and hours lives only in experienced planners' heads.

Chapter 1 stated the question and the scenario that cannot answer it. This chapter walks each of the seven systems of record in turn, because the shape of the design follows directly from what each one sees and what each one misses.

2.1 What Each System Sees

The rail operations and yard management system owns trains, wagons, yard slots and the handover, the moment the whole corridor story turns on; it cannot see the declaration state behind a wagon's box, so it schedules a train that clearance will hold. The transport management system owns the road legs and the haulier assignments; it cannot see the train each leg feeds, so a late truck is late in isolation. The customs authority's declaration and clearance interfaces own declaration state, the stall signal and the single declaration transit lanes; they cannot see the physical position of the box a declaration covers, or the slot its delay consumes. The port community system feeds own the vessel, gate and terminal events that start the story; they go quiet at the gate. The haulier telematics feeds provide road leg position and ETA; they carry no commodity, no declaration and no yard state. The container tracking system owns the container number that joins every system to one consignment; it is a register rather than a story, holding no state any other system owns. The customer service case log holds the questions asked and the answers sent; it records what was said about a consignment, not what was true of it. Figure 2 sets these seven slices side by side, and the register at the end of this chapter states each system's sees and misses in one line.

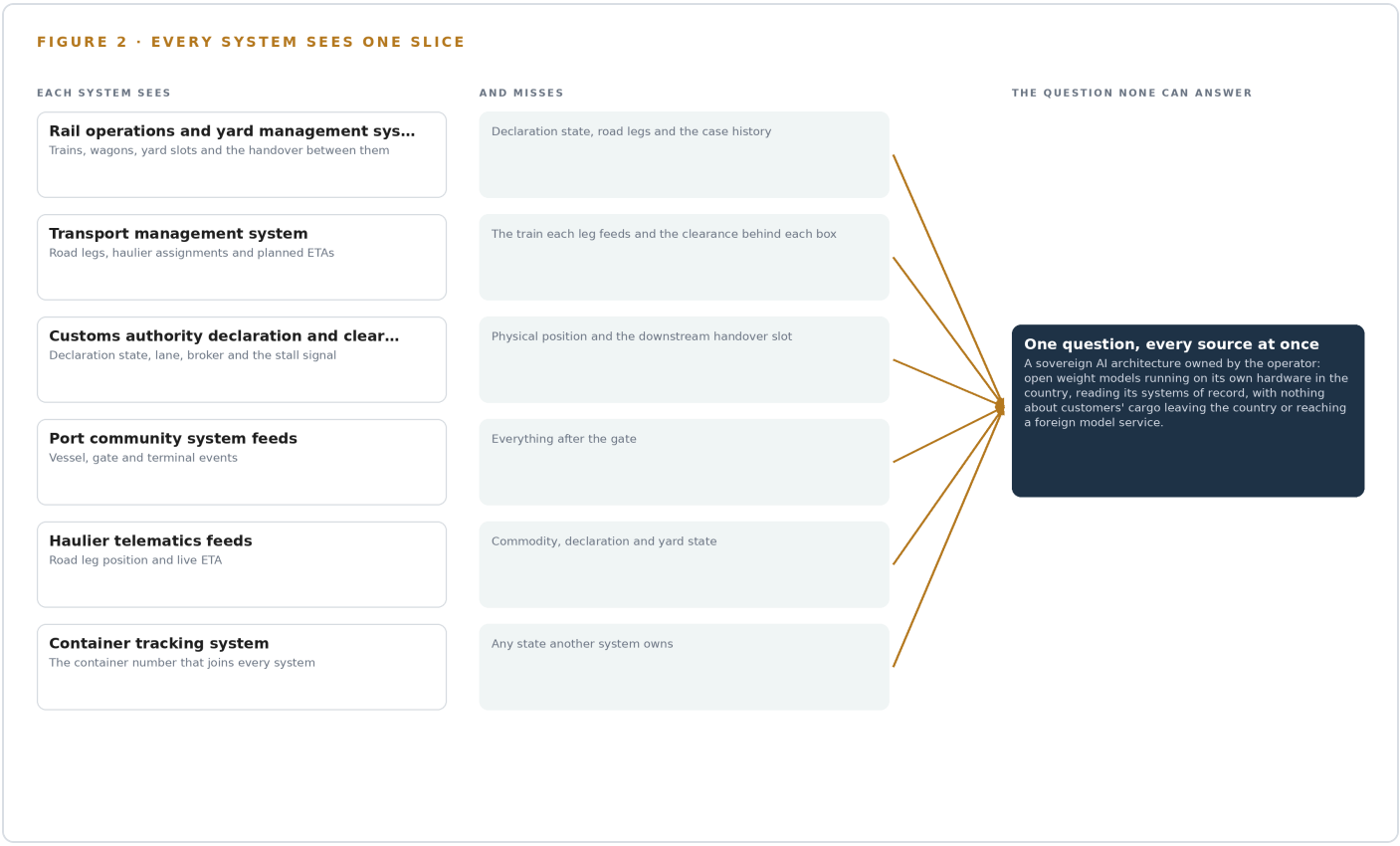


Figure 2. Seven systems, each seeing one part of the answer. The question needs all of them in one place at once.

2.2 What None of Them See Together

What no system holds is the consignment as one object with one identifier and a continuous state, and the costs follow directly. Dwell at the dry port grows while a declaration stalls, and the operation learns of the stall from the consignee rather than from any system. A road leg that runs late is visible in telematics but never joined to the train it will miss, so the handover slot is lost rather than re booked. Experienced planners know which lane, which broker and which hour of the day stall most often, and none of that is written anywhere, so it leaves with the shift. The same questions about the same consignments arrive week after week, because the answer is assembled by hand each time. The join failure is therefore not a reporting gap but a structural one: the data exists, in seven systems, and no identifier in any one of them reaches across all six others.

PART II · CHAPTER 3

Sovereignty and Six Systems Shape Everything Downstream

Four constraints, in-country residency under the national data protection regime, ownership of the weights, projection over the existing systems of record, and a named person behind every action, decide the architecture before any component is chosen.

Chapter 2 ended with the join failure: one consignment, six identifiers, and knowledge of stalling lanes, brokers and hours living only in planners' heads. This chapter sets the four constraints that decide the architecture before any component is chosen, then records what the design bought, what it refused and what sits outside scope altogether.

3.1 Residency Under the National Data Protection Regime

The first constraint is that cargo, declaration and case data stay on the owner's own hardware in the country, with nothing reaching a foreign model service. This is hard in this industry because supply chains are inherently cross-border: their information flows underpin sourcing, finance, logistics and risk management across jurisdictions (Globaldataalliance n.d.), and haulier telematics in particular may transit a foreign cloud before they reach the estate, which is a covenant and contract question settled at phase zero with egress monitoring at the boundary. The regime's transfer bases, the data protection officer's records of processing duties and the residency rules are carried as phase-zero approvals, not as assumptions the build outlives.

3.2 Ownership of the Weights

The second constraint is that the reasoning tier runs open weights the owner holds on its own hardware in the country. This is hard for two reasons the industry cannot negotiate away. The GPU class the sizing rule names faces export licensing whose timeline sits outside anyone's control, so the license determination is a phase-zero checkpoint before purchase. And model licenses carry triggers, including one that fires for a corporate group running a Model as a Service business above 10 billion USD, which is checked against the owner's group at phase zero, with an MIT-licensed fallback standing behind it.

3.3 A Projection Over the Systems of Record

The third constraint is that the one model of the corridor is a projection over the seven systems, never a migration and never a shadow copy. A projection differs from the industrial ontology examples a reader may know: it owns no copy of any record, only the crossing mappings that answer the questions no single system can. This is hard because each system is a system of record with its own identifier and the join runs on the container number alone, so the design carries a weekly join-success gate at 95 per cent with failures triaged by source system, on the grounds that a join failing silently is worse than no join at all.

3.4 A Named Person Behind Every Action

The fourth constraint is that no agent files, amends, re books or cancels anything without a named approval, and approval authority is an explicit property of the actor objects in the model. This is hard precisely because the value of the design lies in the actions it proposes: drafted answers with citations, re booking proposals, stall alerts. The design resolves the tension by agreeing a per-shift alert budget before thresholds are set and making the action rate a phase gate, so the surface earns attention rather than demanding it and quietly losing it.

3.5 Scoping Decisions

Three scoping decisions follow from the four constraints, and each buys something at a stated cost, recorded in Table 1: the ontology as a projection rather than a migration, the phase-one corridor only with extension by measured gates, and memory loops instead of weight changes. Each decision is revisited only at a gate and never mid-phase, because a constraint traded quietly mid-build is a constraint that was never held. The gates themselves spend the quarter's measured evidence of dwell, stalls, missed connections and answered questions against the recorded baseline.

Table 1 · Scoping Decisions

DECISION	WHAT IT BUYS	WHAT IT COSTS
One ontology as a projection over the seven systems of record	The one argument about the consignment without touching any system of record	Adapter discipline on every source, and a weekly join-success gate at 95 per cent
Phase one corridor only, extension by measured gates	A recorded baseline of dwell, stalls, missed connections and answered questions before any scale	A narrow first proof, and a quarter of measurement before the other corridors
Memory loops instead of weight changes	Improvement the owner owns outright, with nothing trained into weights	Curation of the decision record every run reads before it acts

3.6 What the Design Chose Against

Table 2 records where the design made a choice and what it refused in each place, from the serving footprint to the boundary of scope. The refusals are the architecture as much as the choices are: what is excluded, from model training to autonomous declaration filing, is what keeps every constraint in Chapter 3's first four subsections true after the build exists. The final row states the scope boundary in full, including what the owner supplies and what remains outside this design's price.

Table 2 · What the Design Chose Against

WHERE	WHAT WAS PICKED	INSTEAD OF, AND WHY
Model serving footprint	One node of the 141 GB HBM class at FP8	Two 80 GB class nodes, or BF16 on a 256 GB class node: the weights, about 756 GB at FP8, fit one node with about 280 GB of KV cache ceiling, and one node serves the phase-one corridor
Memory and learning	Memory loops that each run reads before it acts	Fine tuning on stall histories: the requirement asks for improvement without training on the fly, and nothing is trained into weights

WHERE	WHAT WAS PICKED	INSTEAD OF, AND WHY
The one model of the corridor	A governed ontology held as a projection	A warehouse or a migration of the systems of record: the seven systems stay in place, joined by the container number they already share
Decision authority	Approval authority as a property of the actor objects	Autonomous agents acting on exceptions: no declaration is filed, amended or re booked without a named person
Scope boundary	Phase one corridor, live feeds and the agreed opening balances	Model training, cameras and roadside sensing, replacement of the systems of record, corridors beyond phase one, autonomous declaration filing, the hardware and its support contracts, third-party license fees, historic data migration, estate-wide end user training and round the clock support

PART II · CHAPTER 4

One Stack Runs From Records to Reasoning In-Country

The design puts the systems of record below, one governed object model in the middle, and services and surfaces above, all on the owner's own central node with an event backbone holding the tiers together.

Chapter 3 fixed the four constraints that shape everything downstream: sovereignty over data and weights, no replacement of the systems of record, a named person on every decision, and a first gate that can stop the work cheaply. This chapter lays out the stack that satisfies them, and it is the shape a sovereign AI architecture in this trade arrives at: records below, one governed model in the middle, agents and surfaces above.

4.1 The Pattern and Why It Fits

Three layers in a fixed order. Below sit the seven systems of record, read only, never replaced, never modified, never loaded. In the middle sits one governed object model of the corridor, a projection over those systems rather than a copy of them, so the model can be rebuilt from the sources at any time and never becomes a second book of truth. Above sit the applications and agents that draft answers, rebookings and exception chases, all of which become real only when a named person approves them. Figure 3 shows the layered stack with its counts: seven sources, two adapter families, fifteen objects, seven services and three surfaces.

The pattern fits because the owner's problem is a join failure, not a data shortage. Every fact the consignment story needs already exists inside the estate; what is missing is one place where the facts are already joined. International supply chains ordinarily depend on information moving across borders

(Globaldataalliance n.d.), and this design inverts that habit for the operator's cargo, customer and declaration data, which stay in the country on the owner's own central node; there is no edge tier, and the one piece of equipment the design adds is that node, planned at about ten kilowatts of rack power and cooling in the owner's data center. An event backbone, Apache Kafka 4.3 in KRaft mode, holds the tiers together by carrying every change the adapters emit, in order. Table 7, printed by the platform from the plan, gives the layer by layer register of components and choices with the reason for each.

FIGURE 3 · THE LAYERED STACK: 3 SURFACES, 7 SERVICES, 15 OBJECT MODEL, 2 ADAPTERS, 7 SOURCES

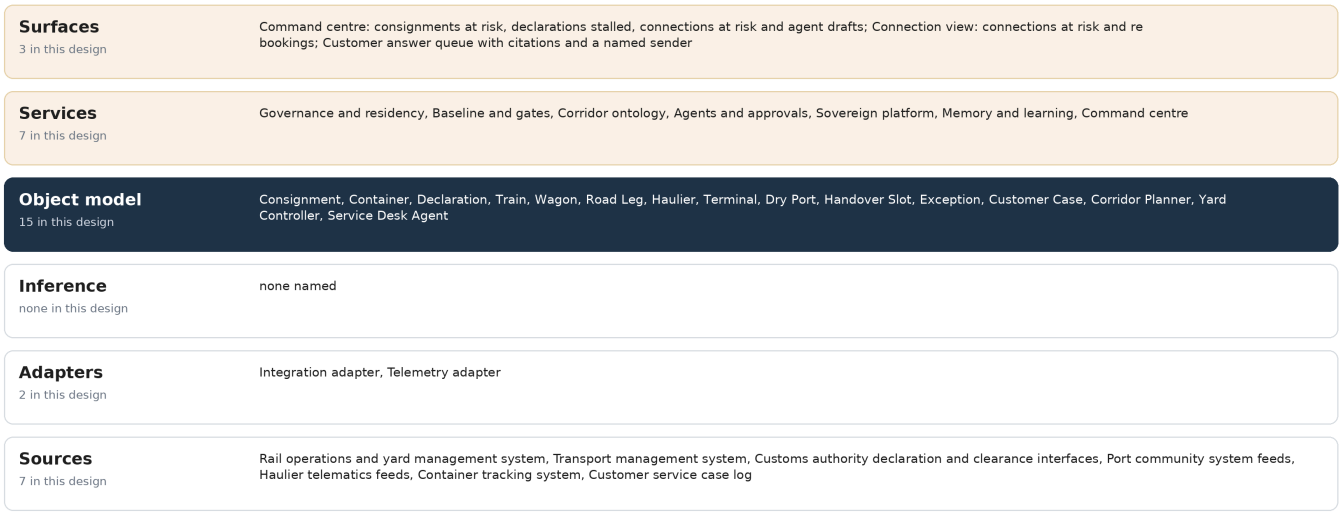


Figure 3. The layered stack: 7 sources, 2 adapter families, 15 objects, 7 services and 3 surfaces.

Table 7 · Architecture, Layer by Layer

LAYER	COMPONENT	CHOICE	WHY
Event streaming and messaging	by class, no product named	Product	Apache Kafka 4.3 KRaft
The ontology or graph store	by class, no product named	Product	Hyper Ontology v0.9
Time series and telemetry storage	InfluxDB 3 Core, TimescaleDB, QuestDB	Class	The shelf holds InfluxDB 3 Core, TimescaleDB and QuestDB; the choice waits on the phase-one corridor's measured event volumes.
Video ingest, recording and management	by class, no product named	Not needed	No cameras; the design watches records and telematics, not video.
Edge model serving	by class, no product named	Not needed	No edge inference; all reasoning runs on the owner's central node, as the intake fixed.
On site language model serving	by class, no product named	Not needed	No model in the register survived this run's live checks, so nothing is served on this layer yet. The

LAYER	COMPONENT	CHOICE	WHY
			layer is model open work: it is chosen and sized when a verified model is.
Object tracking and trajectory forecasting	by class, no product named	Not needed	Road legs arrive as telematics positions and ETAs, not video tracks.
Edge orchestration of containers	by class, no product named	Not needed	No edge compute tier exists in this design.
An air gapped artifact and model registry	by class, no product named	Product	Harbor
Labelling and retraining	by class, no product named	Not needed	Under system of context nothing is trained; improvement runs through its memory loops.
Time synchronisation across sensors and systems	by class, no product named	Product	Chrony
Observability of the running system	by class, no product named	Product	Prometheus, Grafana and Loki

4.2 The Stack Stage by Stage

Table 3 walks the same stack stage by stage, and each stage names its components. A supporting spine runs beside the stages: Chrony disciplines the clocks that event ordering depends on, Prometheus, Grafana and Loki observe the running system, Keycloak holds identity and access, and Harbor keeps artifacts and model images in an air-gapped registry. No model row survived this run's live checks, so language model serving is model open work: the hardware class is fixed at the 141 GB HBM class, and weights are committed only when a verified model stands in the register.

Table 3 · The Stack, Stage by Stage

STAGE	WHAT IT IS RESPONSIBLE FOR	HOW
Sources	Holding every fact the corridor story is made of: trains, wagons, yard slots, road legs, declarations, gate events, container identity and service cases	Seven named systems of record, all read inbound over agreed channels and never replaced
Sensing	Watching the corridor through records and telematics already flowing	Haulier telematics provide road leg position and ETA; no cameras and no new positioning hardware
Adapters	Putting every source behind one guarantee of lineage, ordering and staleness flags	Two families, the integration adapter and the telemetry adapter, built on Debezium with

STAGE	WHAT IT IS RESPONSIBLE FOR	HOW
		Apache Camel
Object model	One governed model of the consignment story, fifteen objects joined by typed links	Corridor ontology held in Hyper Ontology v0.9, a projection over the sources, never a copy
Inference	Reasoning over the model: stall detection, connection watch, drafted answers	Central node in the owner's data center on open weights; the model row is open work until a verified model stands in the register
Services	The seven workstreams: governance and residency, baseline and gates, corridor ontology, agents and approvals, sovereign platform, memory and learning, command center	Each service reads the object model and writes only through approvals; memory runs through a kept record of decisions and outcomes that the next decision reads, never into weights
Surfaces	The three places people work: the command center, the connection view, the customer answer queue	Rendered on the work surface, with citations on every answer and a named sender before it goes

4.3 A Reusable Pattern

The reusable shape is the whole middle of this design: read-only systems of record below, a projection ontology joined by typed links, an event backbone carrying changes in order, adapters that guarantee lineage and staleness flags, and above it all agents that draft while people approve. A team carrying a similar brief keeps that shape and swaps the contents: their own source systems, their own object list grown from the nouns and relationships of their operation, their own thresholds, baselines and approval authorities, and their own hardware class once a model stands in their register. What they should not swap is the discipline. Sources are never written to, the model is never a second book of truth, and no action is real until a named person makes it so.

PART II · CHAPTER 5

Fifteen Objects Turn Corridor Data Into One Story

The object model makes the consignment the focal object and joins container, declaration, train, wagon, road leg, haulier, terminal, dry port, handover slot, exception, case and the three human roles through typed links a document store cannot traverse.

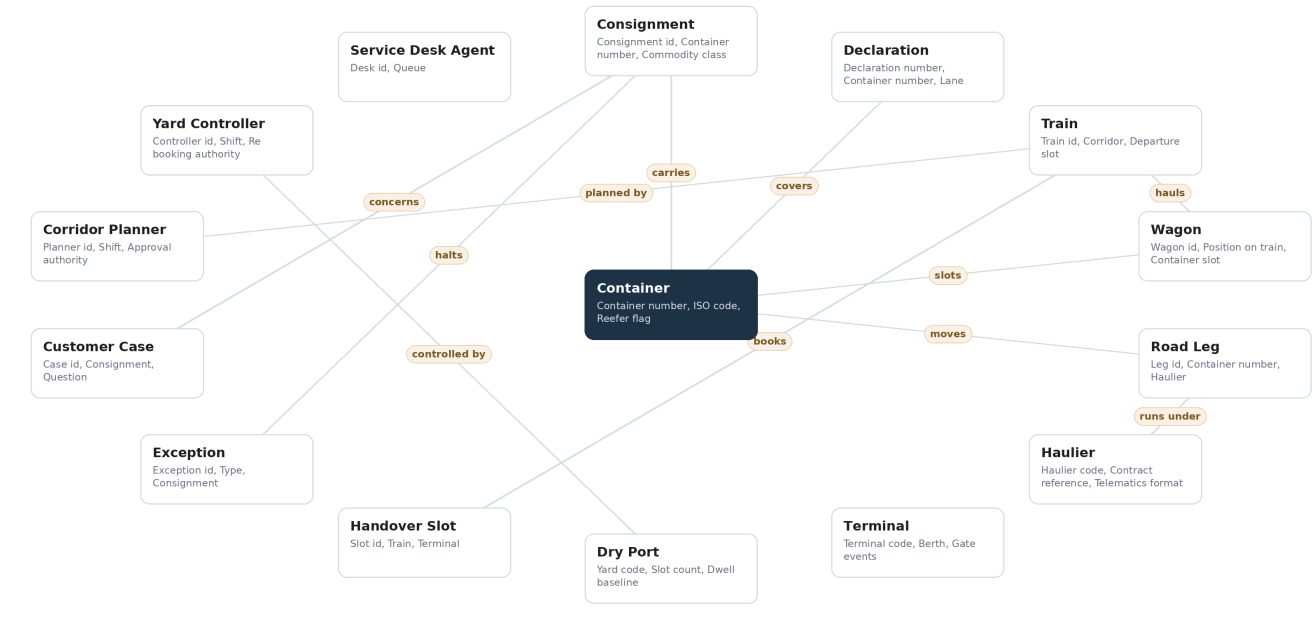
Chapter 4 laid the stack from records to surfaces. This chapter opens the middle layer, the corridor ontology, where fifteen objects and their typed links turn seven fragmented records into one story a planner can read.

5.1 Fifteen Objects, One Story

The model holds fifteen objects: Consignment, Container, Declaration, Train, Wagon, Road Leg, Haulier, Terminal, Dry Port, Handover Slot, Exception, Customer Case, and the three human roles, Corridor Planner, Yard Controller and Service Desk Agent. The consignment is the focal object, anchored in the container tracking system, with a status vocabulary that runs from Booked through At port, Cleared, In transit rail and At dry port to Handed over, with Exception as the turning point. Every other object attaches through typed links: the declaration joins the container by container number, the train carries its wagon list and each wagon carries the container in its slot, the road leg names its container and its haulier, the handover slot books its train, and the exception and the customer case name the consignment they concern. Figure 4 shows every object and every link.

What the links buy is traversal. Ask for consignments whose declaration has sat Queried past the lane's normal time, and the model returns a path, not a list: declaration to container to consignment, to the train and the handover slot it will miss, to the road leg and haulier downstream of that miss, to the open customer case asking about the same box, and to the corridor planner who holds approval authority over that train. A document store keyed on consignment cannot make that traverse, because the joins are not in the documents; they are in the typed links, so the traversal is a query rather than a project. Many industrial ontology examples stop at naming entities, and this model earns its keep one level up, in the links between them. Table 8, printed by the platform from the plan, gives each object with the system it is anchored in and its typed links.

FIGURE 4 · THE OBJECT MODEL: 15 OBJECTS AND 11 TYPED LINKS



The navy card is the object the most links reach. A link label is the property the plan carries, or the verb the design names.

Figure 4. The fifteen objects of the model and the typed links that let a query reach across them.

Table 8 · The Object Model

OBJECT	ANCHORED IN	TYPED LINKS
Consignment	Container tracking system	carries: Container
Container	Container tracking system	no typed link recorded
Declaration	The customs authority's declaration and clearance interfaces	covers: Container
Train	Rail operations and yard management system	hauls: Wagon; planned by: Corridor Planner
Wagon	Rail operations and yard management system	slots: Container
Road Leg	Transport management system	moves: Container; runs under: Haulier
Haulier	Transport management system	no typed link recorded
Terminal	Port community system feeds	no typed link recorded
Dry Port	Rail operations and yard management system	controlled by: Yard Controller
Handover Slot	Rail operations and yard management system	books: Train
Exception	Raised by the design's stall detection and connection watch	halts: Consignment
Customer Case	Customer service case log	concerns: Consignment

5.2 The Human Loop and the Hosting Posture

The human loop lives in three places: the corridor planner holds approval authority, the yard controller holds rebooking authority over handover slots, and the service desk agent works the answer queue. The single write path is a named approval; no agent files, amends or cancels a declaration on its own, and agents draft while people decide. The hosting posture keeps everything on the owner's own hardware in the country. Identity and access run on Keycloak with permissions scoped to the three human roles, which matters because logistics data leaks from inside as well as from outside: at one parcel operator a former employee moved more than 26,000 customer records to outside firms, one of which used them for sales (Nikkei 2025). The estate makes no external model calls; there is no third party model service, and the boundary holds cargo, customer and declaration data in the country. Memory runs through a kept record of decisions and outcomes that the next decision reads, so each decision reads the record of earlier ones before it acts, and nothing is trained into weights.

5.3 One Object in Its Recorded Form

The Consignment, in its recorded form as the platform prints it below, carries its identifier, the container number it carries, its commodity class, its origin seaport and destination dry port, its current hand, its status vocabulary from Booked to Exception, and the link that places the container inside its story; the declaration, with its lane, broker, filing time, clearance state and last movement time and a status vocabulary from Filed to Held, is recorded the same way.


```
{ "id": "consignment", "label": "Consignment", "kind": "record", "anchored_in": "Container
tracking system", "properties": [ "Consignment id", "Container number", "Commodity class",
"Origin seaport", "Destination dry port", "Current hand", "Story link" ],
"status_vocabulary": [ "Booked", "At port", "Cleared", "In transit rail", "At dry port",
"Delivered", "Exception" ], "links": [ { "to": "container", "label": "carries" } ] }
```


PART II · CHAPTER 6

Every Source Enters Through an Adapter, Never Directly

Two adapter families, integration and telemetry, put seven named source systems behind one guarantee of lineage, ordering and staleness flags, so a stall is detected when the data says so rather than when the consignee calls.

Chapter 5 showed what the object model holds. This chapter shows how data gets into it: through two adapter families, never directly, so that lineage, ordering and staleness are guaranteed before any agent reasons over a single fact.

6.1 Seven Sources, All Read

Figure 5 maps the seven named source systems and their adapter paths. The rail operations and yard management system owns trains, wagons, yard slots and the handover the whole corridor story turns on. The transport management system owns the road legs and haulier assignments the connection watch reasons over. The customs authority's declaration and clearance interfaces own declaration state and clearance events, the stall signal and the single declaration transit lanes. The port community system feeds own the vessel, gate and terminal events that start the consignment's story. The haulier telematics feeds provide the road leg position and ETA the connection watch depends on. The container tracking system owns the container number that joins every system to one consignment. The customer service case log holds the questions the answer queue drafts against and the record of answers sent. Every source carries the operator's own provenance class: each fact enters already owned by the operation, read inbound over agreed channels, with nothing scraped or purchased, and all seven are read only.

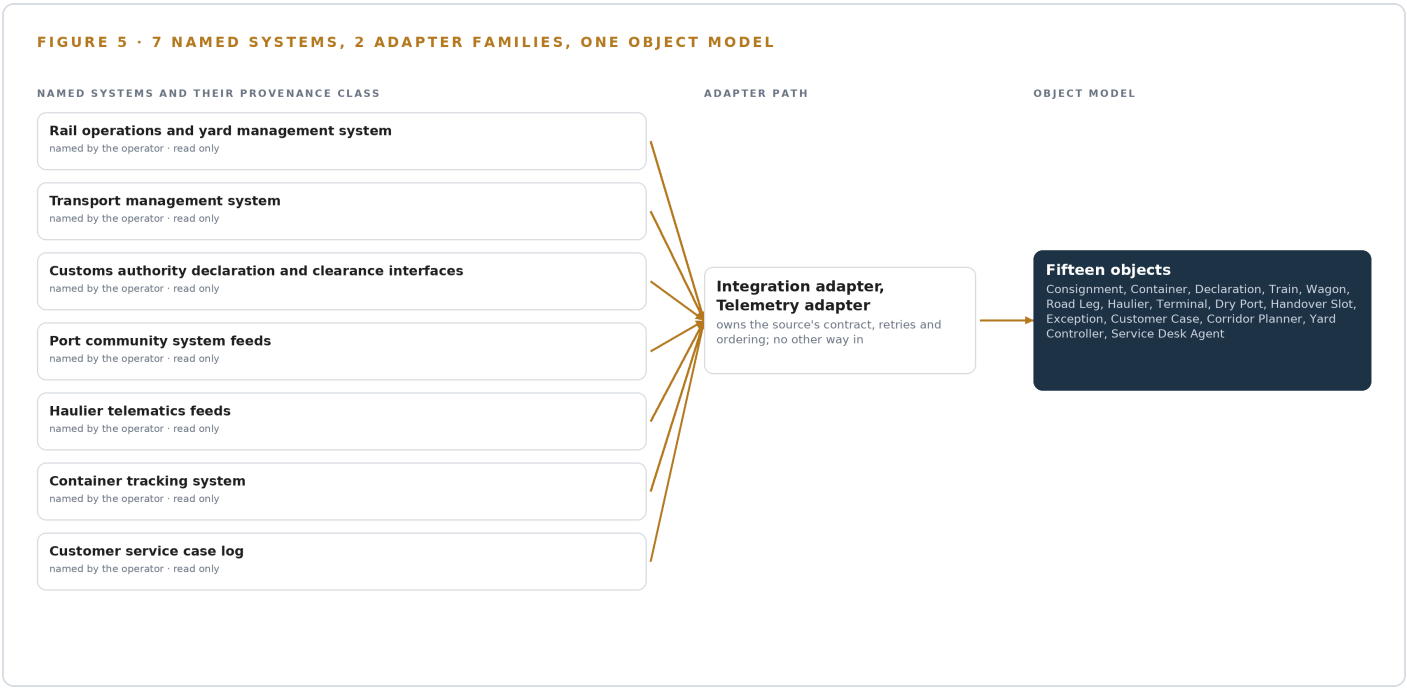


Figure 5. The 7 named systems, the adapter path each one takes, and the object model they all map into.

6.2 What the Adapter Tier Guarantees

Two families stand between the sources and the backbone. The integration adapter, built on Debezium with Apache Camel, brings the systems of record in as change events; the telemetry adapter brings the haulier feeds in as positions and ETAs. Together they guarantee three things. Lineage: every property on every object traces to a source record and a timestamp, so an answer can cite where each fact came from. Ordering: clocks across the estate are disciplined by Chrony, so a gate event and a clearance event on the same consignment can be sequenced. Staleness flags: each interface's cadence is published in the command center, so a declaration that has not moved reads as no news rather than as cleared, and a stall is detected when the data says so. One audited seam also has a security value, because logistics estates are breached through shared infrastructure and third parties: personal data left one provider through a shared network station (Warehousetotaal 2026), and third-party breaches doubled as a share of incidents in the most recent year (Cybelangel 2026).

6.3 The Event Backbone

Apache Kafka 4.3 in KRaft mode carries every change the adapters emit. Ordering is per partition keyed on the container number, so each consignment's story is sequenced within its own lane even when the whole estate streams concurrently. Delivery is at least once with idempotent handlers keyed on source record ids, so a replayed event cannot double-write a state change. Buffering is the retained log: when an interface drops or a scheduled file delivery slips, events wait, and the object model marks the gap rather than guessing. Replication stays in country, across the owner's own data center racks, and nothing replicates outside the boundary. Replay is also the recovery path, because a ransomware event stopped a logistics operator's services outright and encrypted the data it held (Askullogist 2025); the backbone's retained logs let the object model be rebuilt from the sources without ever writing back to them.

All Reasoning Runs on the Operator's Own Node

There is no edge tier in this design: the central node in the owner's in-country data center serves every agent and surface, the latency budget is set by the corridor's decision cadence, and the boundary admits only inbound reads.

Chapter 6 described how seven source systems enter through two adapter families and never directly. This chapter fixes where the reasoning over that joined data runs, and the answer is short by design: industrial AI on this corridor does not need an edge tier, because the decisions the system supports run on a cadence of minutes and hours rather than milliseconds, and the sovereignty the owner demands is easiest to hold in one place.

7.1 One Tier: the Central Node in Country

There is one inference tier, the owner's central node in the in-country data center, and it serves every agent and every surface. The design commits no cameras, no roadside compute, no edge orchestration of containers and no edge model serving, because the corridor is watched through records and telematics rather than through instruments, and haulier telematics already provide position and ETA. The plan sizes the estate for the phase-one corridor with this single node, with any scaling decided at the extension gates rather than assumed at the start.

The memory arithmetic comes from the sizing rule the register records, and it is stated here as that rule rather than as a committed model row, a distinction Chapter 8 carries in full. GLM 5.3, the model the requirement names, occupies about 756 GB of weights at FP8 precision, taken as 753B parameters. One node of the inference cluster class (GPU nodes for 300B to 1T parameter open weight models), with eight GPUs and about 1,038 GB of usable memory, holds those weights and leaves about 280 GB of KV cache ceiling, the working memory that determines how much concurrent context the node can serve. The node is planned at about ten kilowatts of rack power, which is ordinary data-center power and cooling rather than a field engineering problem. Figure 6 sets the placement out: one node, seven read-only source systems below it, three surfaces above it, and no second tier anywhere in the picture.

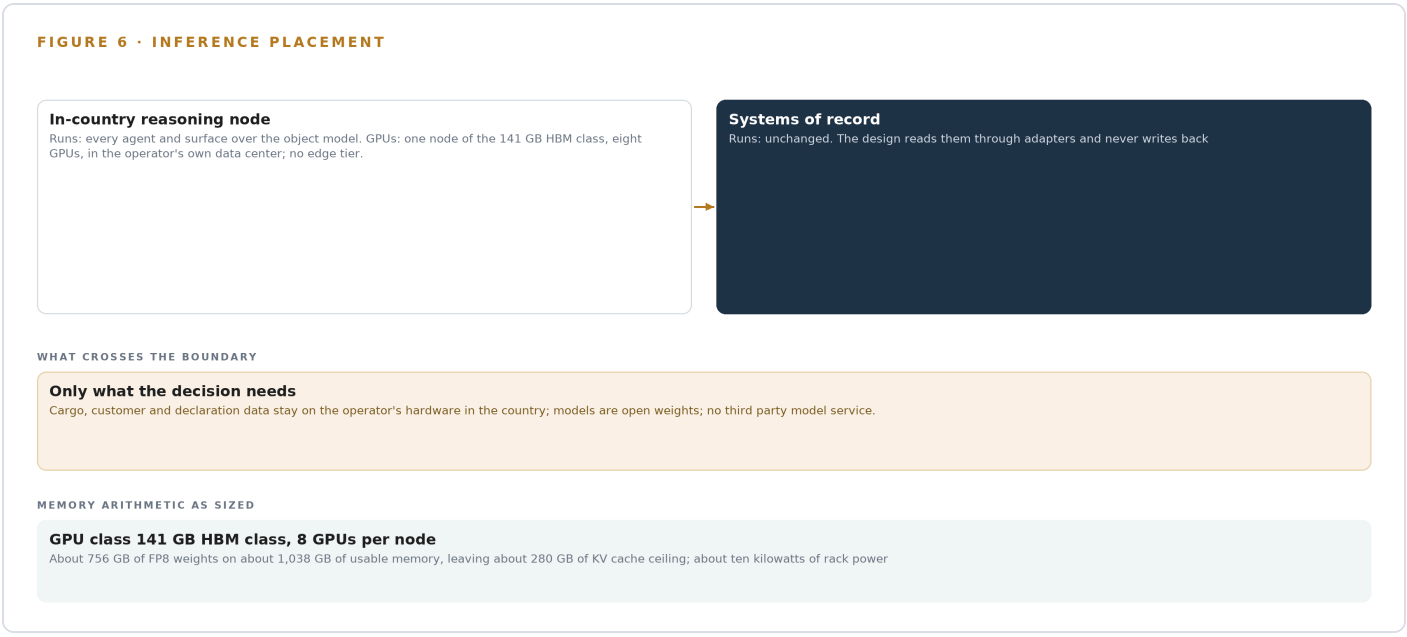


Figure 6. Where each tier runs, what runs there, and the narrow set of outputs that cross the boundary.

7.2 The Latency Budget Is the Decision Cadence

Each decision the system supports has its own clock, and the budget is set by that clock rather than by a target written in milliseconds. Stall detection watches declarations against lane normal-time baselines, and it must fire while the stall still costs hours of a nominal 96 hour dry port dwell rather than after the consignee calls, so its budget is the interval between a declaration's last movement time and the moment a planner could still act. The connection watch joins a road leg's telematics ETA to the handover slot window, and its budget is the gap between the ETA update and the window start, because an answer that arrives after the train has departed is a record rather than a decision. The answer queue drafts replies with citations and a named service desk agent sends them, so its budget is a working shift and completeness of citation matters more than a second saved. Scheduled file drops make some interfaces staler than others, so the command center publishes a per-interface cadence and flags staleness rather than pretending to a real time it does not have.

7.3 The Boundary Admits Only Inbound Reads

What crosses the network boundary is inbound reads, nothing else: seven interfaces over agreed channels, each read-only, with the systems of record never replaced, modified or loaded, and the single write path in the design running from the adapter tier into the object model. Cargo, consignee and declaration data stay on the owner's hardware in the country, and no third party model service is ever called. The boundary is drawn against the grain of the industry, since cross-border data transfers are integral to how supply chains ordinarily run (Globaldataalliance n.d.), and the owner's requirement inverts that default deliberately. It is also drawn against rising third-party exposure: third-party breaches doubled as a share of all security incidents in the most recent year (Cybelangel 2026), which is why egress monitoring sits at the boundary and why the telematics question is asked at phase zero, because haulier feeds may transit a foreign cloud before they reach the estate.

Each failure path has a stated behavior. When the link fails, staleness flags rise on the affected interfaces, the object model holds last known state, and exceptions recorded on the feed's return carry their true detected-at time rather than a smoothed one. When power fails, the node rides the data center's existing protections, and its about ten kilowatts sit within ordinary rack capacity. When the update path is exercised, weights and artifacts enter only through the air-gapped artifact registry, verified before anything is served, so an update can never become an unexamined ingress.

PART II · CHAPTER 8

The Register Holds No Weights Until Verification Stands

The license decides what the owner can own, and no open weight model survived this run's live checks, so the register records the sizing rule, the inference cluster class (GPU nodes for 300B to 1T parameter open weight models) and the GLM 5.3 license question as open work rather than committing weights it cannot stand behind.

Chapter 7 placed every agent and surface on the owner's central node and drew the boundary as inbound reads only. This chapter states what that node is licensed and sized to run, and why the register commits no weights it cannot stand behind. Figure 7 opens the chapter with the model stack: one planned slot, the work surface model class open weights, self hosted, with the model row itself marked as open work rather than a filled commitment.



Figure 7. The model stack: no committed model row; the design names classes and roles, and a model is chosen at the first gate.

8.1 What the Requirement Names, and What the License Allows

GLM 5.3 is the model the requirement itself names: class open weights, self hosted for sovereignty, of the frontier class the plan records. Its role is the conversational command center the requirement asks for, the interface that answers questions about a consignment with citations back into the object model, and it runs on the single central node of Chapter 7. The sizing arithmetic already stated, 753B parameters at FP8, about 756 GB of weights on one 141 GB class node with about 280 GB of KV ceiling, is the rule that fixes its footprint. Its license carries a trigger: it bites a corporate group running a Model as a Service business above 10 billion USD, so the owner's corporate group is checked at phase zero, and GLM 5.2

under MIT stands as the fallback if the trigger is met. Open weights are what make the ownership real, because the owner holds the weights on its own hardware and no third party model service ever sees cargo, consignee or declaration data.

What the register does not do is commit that row. No open weight model survived this run's live checks: each catalog entry the design cites was verified by a person on the date it carries and was not rechecked live for this build. That verification is outstanding work before any weights are committed, and the register records it as open work rather than naming weights it cannot stand behind. The honest position is that the design runs on weights the owner holds, so which weights is part of what is being bought, and a design file that invented certainty here would spend the owner's money on it.

8.2 Equipment, Rules and the Register

Table 4 gathers the register in one place. On hardware, the choice is one node of the inference cluster class (GPU nodes for 300B to 1T parameter open weight models) with eight GPUs, because it holds the FP8 weights with the KV ceiling stated above, while two 80 GB class nodes and BF16 on a 256 GB class node were set aside. Sizing runs through the Model GPU Footprint Sizing Rule (vLLM, SGLang) and the GLM 5.3 Node Count By GPU Class, which fix the node count from weights, usable memory and KV ceiling before any purchase is made. On sensing, the register holds none: the design reads declarations, slots, gate events and telematics ETAs, not video. The pattern the design stands on is system of context, with the ontology as a projection over the seven systems of record and improvement running through a kept record of decisions and outcomes that the next decision reads instead of fine tuning, because the requirement asks for improvement without training a model on the fly. The ground is the owner's own hardware in an in-country data center, with weights and artifacts entering only through the air-gapped registry.

That ground carries one rule the design never contradicts. Saudi Arabia sits in US Country Groups D:3 and D:4, so the 141 GB class ships there only under a US export license granted case by case, a determination against ECCN 3A090 and 4A090 is required before any GPU class is quoted, and the license checkpoint sits in phase zero before purchase, with delivery riding timelines outside the vendor's control. Cross-border flows are how supply chains normally work (Globaldataalliance n.d.); this design keeps the cargo's data inside the country while the containers themselves cross borders, and the register says so in rows rather than in promises.

Table 4 · Model and Equipment Register

THE CHOICE	WHAT WAS PICKED	WHY HERE
Work surface model	GLM 5.3, class open weights, self hosted, frontier class	Named in the requirement; open weights on the owner's own hardware keep every answer in country

THE CHOICE	WHAT WAS PICKED	WHY HERE
License fallback	GLM 5.2 under MIT	Escapes the GLM 5.3 trigger for a corporate group running a Model as a Service business above 10 billion USD
GPU class	One node of the 141 GB HBM class, eight GPUs	About 756 GB of FP8 weights fit with about 280 GB of KV ceiling; phase one is sized on one node, gates scale it
Sizing rules	Model GPU Footprint Sizing Rule (vLLM, SGLang); GLM 5.3 Node Count By GPU Class	Fix the node count from weights, usable memory and KV ceiling before any purchase
Sensing	None; records and telematics only	The corridor is watched through declarations, slots and ETAs, not instruments
Pattern	System of context	The ontology is a projection over seven systems of record; a kept record of decisions and outcomes that the next decision reads improves the plan without training weights
Verification	No model row stands until live checks pass	Each catalog entry was verified by a person on its date and not rechecked live for this build
Ground	The owner's own hardware, in-country data center, air-gapped artifact registry	Cargo, consignee and declaration data never leave the country; no third party model service
Export license	Determination against ECCN 3A090 and 4A090 before quoting any GPU class	Saudi Arabia sits in Country Groups D:3 and D:4; the 141 GB class ships only under license, checked at phase zero

PART III · CHAPTER 9

A Measured Quarter Gates Every Corridor Extension

Three phases with seventeen items carry the build from baseline and approvals through the phase-one corridor and its quarter of measured dwell, stalled declarations, missed connections and answered questions, and the numbers, not opinion, open the further corridors.

Chapter 8 closed the model register and the license terms that decide whether the owner can hold its own weights. This chapter turns to the build itself: how seventeen items in three phases move from approvals to a measured quarter on the first corridor, and why the numbers, not opinion, open the corridors beyond it.

9.1 The Phases and Their Gates

The rollout carries three phases and seventeen items, shown with their workstreams and gates in Figure 8. Phase 0 holds two items in two workstreams, governance and residency together with baseline and gates: residency and data protection compliance, and the phase-one baseline measurement. Nothing moves until the residency approvals close and the baseline is recorded, because a savings claim without a recorded starting point is an opinion. Phase 1 holds fourteen items in six workstreams: agents and approvals, baseline and gates, command center, corridor ontology, governance and residency, and memory and learning. Its exit gate is source system bindings and lineage: every property on every object must trace to the system of record that anchors it, and the joins must hold on one real week of production data before any agent is allowed to trigger. Phase 2 holds a single item, the extension gates to further corridors, and it spends nothing but the quarter's measured evidence. The requirement register holds fifteen requirements, each traced to a phase item, and coverage is reviewed at every gate so a gap surfaces as a gate failure rather than as a surprise.

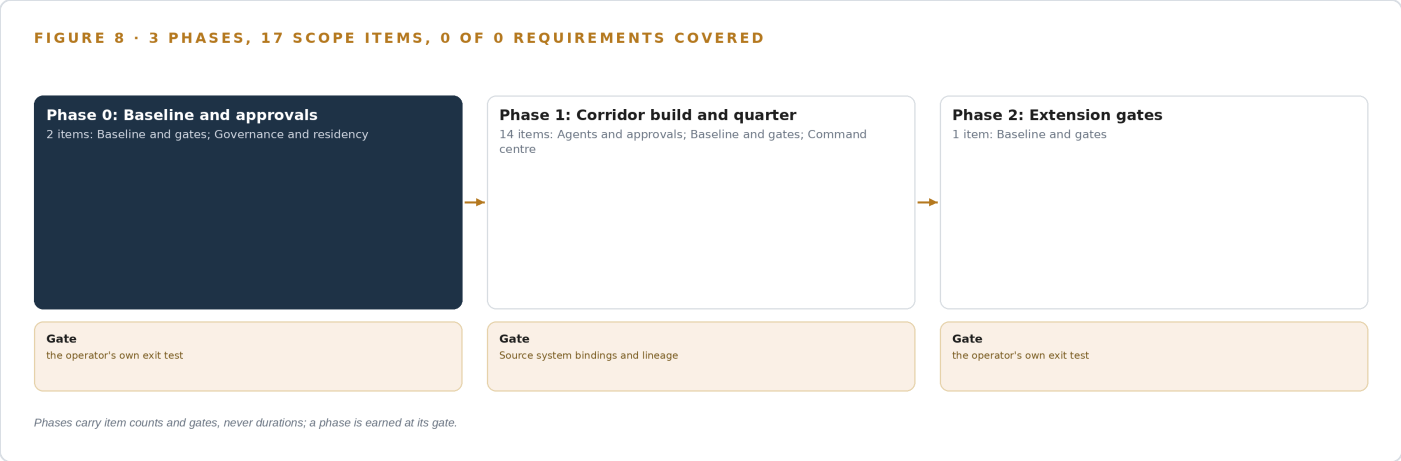


Figure 8. The three phases and their gates, and coverage of the 15 requirements across them.

9.2 What the Rollout Measures

The primary measure is dwell at the dry port, in hours, against a nominal of 96, with a breach when dwell runs above it; the fault the design watches for is a declaration stall at the seaport, because a stalled declaration is what turns a scheduled handover into yard congestion the design learns about from the consignee. Around that headline the quarter counts three more things: stalled declarations by lane, broker and hour of the day; missed connections, where a road leg running late in telematics is joined to the train it will miss; and answered questions, the cases the service desk closes with citations from the system rather than by a planner assembling six identifiers by hand. Two guard measures sit underneath. A weekly join-success gate at 95 per cent, with failures triaged by source system, protects the container number join the whole argument rests on. An action rate, the share of drafted answers and re-bookings the planners actually act on, is itself a gate, because a command center nobody acts on is a screensaver.

9.3 Failure Modes

Table 5 lists what fails and what the design does about it. The telematics row deserves its place on evidence: supply chains depend on information moving across borders (Globaldataalliance n.d.), and third-party breaches doubled as a share of security incidents in the most recent year, with regulators holding leadership accountable for vendor risk (Cybelangel 2026), so a feed that transits a foreign cloud before reaching the estate is a boundary failure, not a technicality.

Table 5 · Failure Modes

WHAT FAILS	WHAT THE DESIGN DOES
The container number join silently fails across the six identifiers	Weekly join-success gate at 95 per cent, with failures triaged by source system
The GLM 5.3 license triggers for a corporate group running a Model as a Service business above 10 billion USD	Check the owner's corporate group at phase zero; GLM 5.2 under MIT is the fallback
Scheduled file drops make stall detection stale	Staleness flags and per-interface cadence published in the command center
Planners ignore drafts and the command center becomes a screensaver	Per-shift alert budget agreed before thresholds are set; action rate is a phase gate
Telematics transit a foreign cloud before reaching the estate	Covenant and contract question at phase zero; egress monitoring at the boundary
Export license delay on the 141 GB class GPU import	License checkpoint in phase zero, before purchase

9.4 Lessons

Bind the join before the agents. The ontology is a projection over seven systems of record, and the container number is the one key that crosses all of them; an agent reasoning over a broken join produces confident nonsense, which is why the source system bindings and lineage gate stands before any agent triggers and why the join-success gate runs weekly forever after. Publish the cadence of every interface. A declaration event that arrives on a scheduled file drop is stale by construction, and a stall detector built on stale events reports yesterday's problem this morning; the command center therefore carries staleness flags and a per-interface cadence so a planner can see how old every signal is before acting on it. Agree the alert budget before the thresholds. Thresholds chosen in a vacuum either flood the shift or whisper; the per-shift alert budget is negotiated with the planners who will receive the alerts, and the action rate gate holds the design to the bargain. Check licenses and covenants at phase zero, not at purchase. The GLM 5.3 license trigger, the telematics cloud path and the GPU export license are all contract and regulatory questions, and each is cheap to ask before anything is bought and expensive to discover after.

9.5 What Is Still Open

Four questions remain open, and each one changes the plan when it settles. The export license determination for the 141 GB HBM class, against the control categories that govern advanced computing items, is a phase-zero checkpoint before purchase; settling it sets the order date and the phase-one hardware timeline, and an unfavourable answer moves the footprint to a different GPU class and a different node count. The register holds no open-weight model verified live on this run: each catalog entry was verified by a person on the date it carries and was not rechecked for this build, which is outstanding work before any weights are committed; settling it fills the serving layer and turns the unsized work surface footprint into a node count. The telematics covenant question, whether any haulier feed transits a foreign cloud before reaching the estate, decides whether egress monitoring closes a real gap or guards an empty one; a logistics peer in the region absorbed a ransomware attack that encrypted data and stopped services at scale (Askullogist 2025), and an insider leak of more than twenty-six thousand records at another carrier shows the boundary faces people as well as networks (Nikkei 2025). Finally, whether chrony's time synchronisation proves sufficient for event ordering decides whether a GNSS grandmaster class is added at the central site; settling it either confirms the software-only answer or adds one hardware class to the register.

PART III · CHAPTER 10

The Intelligence Stays with the Corridor That Produced It

The object model, the weights and their fine tunes if ever made, the decision record and the boundary all belong to the owner, because the memory of which lane stalls at which hour was earned on its corridors and should compound inside its estate.

Chapter 9 showed a quarter of measured dwell, stalled declarations, missed connections and answered questions gating every extension. This chapter settles the question those numbers raise: who owns the intelligence the corridors produced, and where it compounds.

10.1 What the Operator Owns

The owner owns the object model. Fifteen objects, their status vocabularies and their typed links form a projection over the seven systems of record, never a copy of them, and the crossing mappings that answer questions no single system can are the owner's asset; the model survives any change of tooling beneath it because it is the argument, not the storage. The owner owns the weights and any fine-tunes, if ever made: the design commits open weights running on the owner's own hardware in the country, so whatever model the register settles on, the weights sit inside the estate, and under system of context nothing is trained into them, which means the improvement path never creates a derivative the owner does not hold outright. The owner owns the decision record: every exception with its confirmed cause, every approval by a named planner, every re-booking by a named yard controller, and every case

answered with citations, held so the next decision reads what the last one learned about which lane, which broker and which hour stalls most often. That memory was earned on the owner's corridors and should compound inside its estate, not inside a vendor's. The owner owns the boundary: cargo, consignee and declaration data stay on its hardware in the country, no third-party model service is called, and the egress monitoring at the boundary exists to keep that covenant true rather than to assert it.

10.2 The Offer Behind the Design

CodeNinja designed this system on Praxis, its platform for designing physical AI systems, and the design maps directly onto what CodeNinja builds: the object model that projects seven systems of record into one corridor argument is Hyper Ontology; the command center and the work surface where the owner's own people build and adjust agents is Hyper Pragma; the kept record of decisions and outcomes that each next decision reads, which improves the operation without touching model weights, is Hyper Engram; the posture of open-weight models on the owner's own hardware, inside the country, with the export-license and residency questions carried as phase-zero checkpoints, is Sovereign Infrastructure. The design is the argument that these pieces belong together, and the ownership chapter is where the argument lands: everything the quarter measures, and everything the memory learns from it, stays with the corridor that produced it.

PART IV · CONCLUSION

One Projection, One Memory, One Answer, Held Sovereignly

The design is one projection over six systems of record that never replaces or modifies them: an object model of fifteen objects joined by typed links anchored on the container number, adapters that guarantee lineage for every event, all reasoning on the owner's own in-country node under open weights, a memory of decisions and outcomes that improves the corridor without training any model, and a command center where a named planner, yard controller or service desk agent approves every action and every answer carries citations. The first measured quarter gates every extension beyond the phase-one corridor, so the system grows only on evidence the operation itself produced.

Running the same shape elsewhere takes three things the operation must already hold: systems of record whose identifiers can be resolved to one anchoring key, an in-country place for the node and its power, and named people willing to own each decision the agents draft. Swap the object vocabulary for the local corridor's nouns, rebind the adapters to the local source systems, re-run the export license and residency checks for the destination country, and the same fixed order of layers, the same rule of projection rather than migration and the same gated phases carry a different corridor to the same sovereign answer.

How Praxis Contextualized and Reasoned This Design

Every design in the series is produced on Praxis, and this chapter traces each choice in the paper back to the records read, the eight lenses applied and the patterns adopted, so a reader can audit the reasoning rather than take it on trust.

Chapter 10 closed the ownership argument: the model, the weights, the decision record and the boundary all stay with the owner. This chapter opens the reasoning itself, so a reader can audit how the design was reached rather than take it on trust. Every design in the series is produced on Praxis, and this chapter lets a reader trace any choice in the paper back to what justified it (Figure 9).

11.1 Contextualizing the Ask

The ask arrived written in the owner's voice: a sovereign architecture for corridor and customs operations, with open-weight models running on the owner's own hardware in the country, reading the systems of record, and nothing about consignees' cargo leaving the country or reaching a foreign model service. Praxis assigned the design to the Enterprise AI family and the Supply Chain and Logistics industry, a pin made by the forward deployed engineer on the project rather than a reasoned match, with the brief still read whole as reasoning input. Praxis read the corpus shelf of each lens, with nothing ranked or filtered by keyword, and every record the lenses cite is available on request.

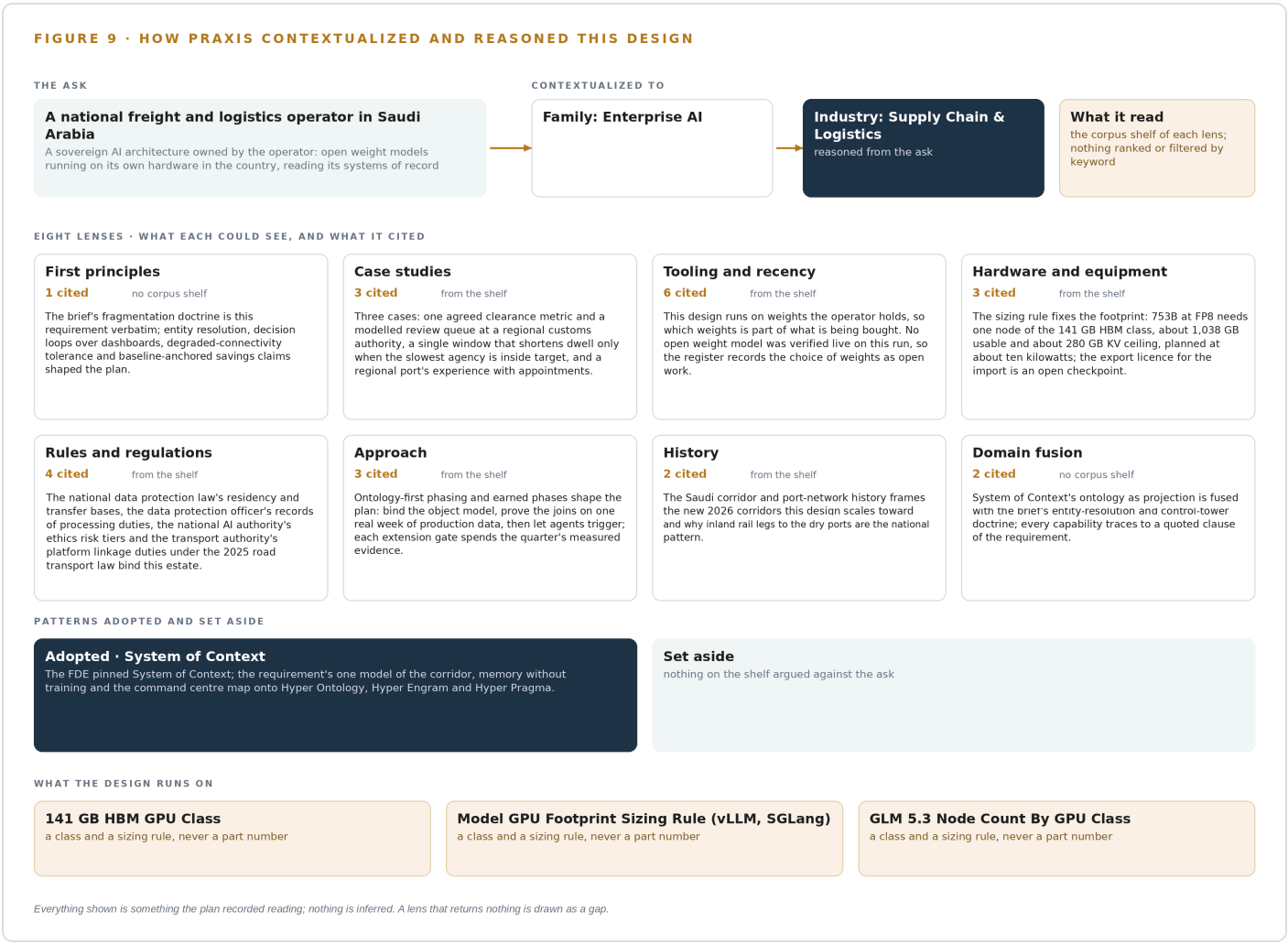


Figure 9. From the ask to the design: the family and industry Praxis assigned, the eight lenses and what each cited, the patterns adopted and set aside, and the equipment the design lands on.

11.2 The Lenses

Table 6 shows all eight lenses, what each could see, how many records each cited, and what each contributed to the design. No lens returned nothing; the tooling lens returned the one gap that matters, the absence of a live-verified open-weight model, and the design carries that gap openly rather than naming weights it cannot stand behind.

Table 6 · The Lenses and What They Contributed

LENS	COULD SEE	CITED	WHAT IT CONTRIBUTED
First principles	no corpus shelf	1	The brief's fragmentation doctrine is this requirement verbatim: entity resolution, decision loops over dashboards, degraded connectivity tolerance and baseline anchored savings claims shaped the plan.
Case studies	the case study shelf	3	Three cases: one agreed clearance metric and a modelled review queue at a regional customs authority, a single window that shortens

LENS	COULD SEE	CITED	WHAT IT CONTRIBUTED
			dwell only when the slowest agency is inside target, and a regional port's experience with appointments.
Tooling and recency	the tooling and recency shelf	6	This design runs on weights the owner holds, so which weights is part of what is being bought. No open weight model was verified live on this run, so the register records the choice of weights as open work.
Hardware and equipment	the hardware and equipment shelf	3	The sizing rule fixes the footprint: 753B at FP8 needs one node of the 141 GB HBM class, about 1,038 GB usable and about 280 GB KV ceiling, planned at about ten kilowatts; the export licence for the import is an open checkpoint.
Rules and regulations	the rules and regulations shelf	4	The national data protection law's residency and cross border transfer bases, the data protection officer's records of processing duties, the national AI authority's ethics risk tiers and the transport authority's electronic platform linkage duties under the 2025 road transport law bind this estate and are carried as phase zero approvals.
Approach	the approach shelf	3	Ontology first phasing and earned phases shape the plan: bind the object model, prove the joins on one real week of production data, then let agents trigger; each extension gate spends the quarter's measured evidence.
History	the history shelf	2	The Saudi corridor and port network history frames the new 2026 corridors this design scales toward and why inland rail legs to the dry ports are the national pattern.
Domain fusion	no corpus shelf	2	System of context's ontology as projection is fused with the brief's entity resolution and control tower doctrine; every capability traces to a quoted clause of the requirement.

11.3 Patterns Adopted and Set Aside

One pattern was adopted, as primary: System of context. Its three layers sit in a fixed order, with the systems of record below, never replaced, never modified, never loaded; the ontology in the middle as a projection over them rather than a shadow copy, its crossing mappings answering the questions no single system can; and the applications and agents above. Its grammar is small and it is the reason the object model reads as it does: nouns become objects, facts become properties, relationships become typed links, and actions change status along a controlled vocabulary. Its kinetic loop closes on the model, sense, decide, act, learn, with the return edge making the model living rather than a diagram. Nothing was set aside, and no precedent on the shelf was rejected: the proven-solutions shelf was not searched before the record closes, and that absence is carried as honest open work rather than dressed up as a choice.

11.4 Where the Reasoning Lands

The reasoning lands on three equipment classes: the inference cluster class (GPU nodes for 300B to 1T parameter open weight models), the model GPU footprint sizing rule for the vLLM and SGLang serving runtimes, and the GLM 5.3 node count by GPU class. Together they fix the footprint at one central node in the owner's own data center, about ten kilowatts of rack power and cooling, with no edge tier, no cameras and no new field sensing anywhere in the design, because the corridor is read through records and telematics rather than instruments. Every figure in this chapter, from the records read to the citations per lens to the memory arithmetic, was recorded reading, and nothing is inferred: what the reasoning did not reach is stated as open work, and what it settled is traceable to the record that settled it.

SOURCES

Source Register

Askullogist. 2025. ランサムウェア攻撃の影響調査結果および. https://www.askullogist.co.jp/pdf/20251212.pdf

Nikkei. 2025. ヤマト運輸で情報2万件超流出 元社員から、別企業が不正利用 , 日本経済新聞. https://www.nikkei.com/article/DGXZQOUC150UY0V11C25A0000000/

Warehousetotaal. 2026. Persoonlijke gegevens Ceva-medewerkers buitgemaakt via gedeeld netwerkstation CLS • Warehouse Totaal. https://www.warehousetotaal.nl/veiligheid-en-personeel/cyber-security/persoonlijke-gegevens-ceva-medewerkers-buitgemaakt-via-gedeeld-netwerkstation-cls/144988/

Globaldataalliance. n.d.. Cross-Border Data Transfers & Supply Chain Management. https://globaldataalliance.org/wp-content/uploads/2021/07/03182021gdaprimersupplychain.pdf

Cybelangel. 2026. Supply Chain Breach Timeline: 2013 to 2025 CybelAngel. https://cybelangel.com/blog/resource/the-2026-supply-chain-cyber-risk-report/

About CodeNinja

CodeNinja is a Middle Eastern American artificial intelligence lab and system of context company. It builds sovereign self-improving systems that organizations own, run inside their own boundary and improve with every decision they make.

codeatoms.ai