VERTICAL-DRIVEN ARCHITECTURES · MARITIME & PORTS · DESIGNED WITH PRAXIS · NOVEMBER 2026

Industrial Ontology Examples: A Port Equipment and Berth Object Model in Saudi Arabia

Maritime & Ports Watch: One Living Picture of the Operation

Muhammad Umar, Cofounder, CodeNinja

For the leaders accountable for maritime & ports operations, their supervisors, and the data and platform 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

The Answer Already Exists in the Systems the Operation Runs

Berth productivity drops and nobody can say whether the cause was the crane's hydraulic fault, the yard block that could not keep up, the vessel's stowage or the berth window the authority granted; this paper, one of the industrial ontology examples in this series, is a port equipment and berth object model that lets a container terminal operator in Saudi Arabia answer that question.

The design reads 9 source systems through 2 adapter families into one object model of 16 objects, runs 10 services over it and opens 4 surfaces to the people who decide, with 3 models placed where the data is. A governed industrial ontology of the terminal's berths, equipment and moves, built from the operator's eight named systems of record and kept current move by move, with three drafting agents (productivity, equipment, commissioning) whose outputs managers confirm and which write back only through named approvals. Reasoning runs on GLM 5.3 open weights self hosted by a container terminal operator in Saudi Arabia on one 141 GB HBM class node, with a kept record of decisions and outcomes that the next decision reads, memory improving every next trace without training.

The paper covers the operation and the problem its systems cannot answer, the constraints, the stack, the object model, ingestion, inference placement, the models and their licenses, the rollout and its proof, ownership, and finally how Praxis contextualized and reasoned the design.

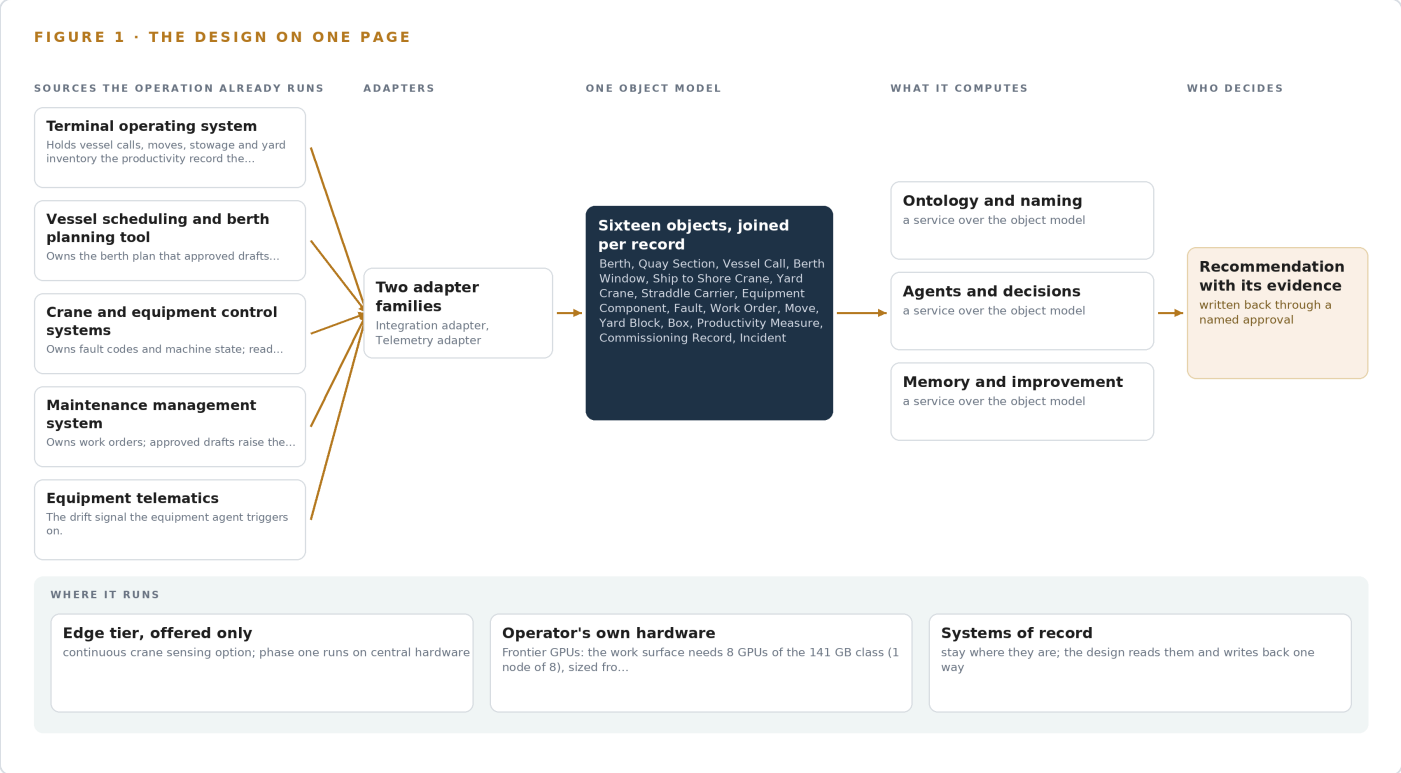


Figure 1. Maritime & Ports Watch on one page: the sources the operation already runs, one object model, what it computes, and the person who decides.

Contents

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

	Abstract · The Answer Already Exists in the Systems the Operation Runs	Executive
PART I · THE PROBLEM		
1	A Daily Question Goes Unanswered in the Systems That Hold the Data	Executive
2	Every System Sees One Slice of the Operation	Executive · Team Lead
PART II · THE DESIGN		
3	Four Constraints Shape Everything Downstream	Team Lead
4	One Stack Runs From the Systems of Record to the Surfaces	Team Lead · FDE
5	Sixteen Objects Turn Operational Data Into One Argument	FDE

6	Every Source Enters Through an Adapter, Never Directly	FDE
7	Inference Belongs Where the Data Is	FDE
8	The License Decides What the Operator Can Own	FDE Executive
PART III · THE ROLLOUT		
9	Shadow Mode Comes Before Any Flag Is Trusted	Team Lead Executive
10	The Intelligence Should Stay with the Operation That Produced It	Executive
PART IV · HOW IT WAS DESIGNED		
	Conclusion · The Pattern Repeats Wherever Systems of Record Do Not Meet	Executive
11	How Praxis Contextualized and Reasoned This Design	Team Lead FDE
	Sources	Reference

PART I · CHAPTER 1

A Daily Question Goes Unanswered in the Systems That Hold the Data

The operation holds every fact the question needs and no system answers it.

The abstract promised one of the industrial ontology examples in this series, a system that answers a daily operational question from data the operation already holds. This chapter states that question exactly, documents what its absence costs, and sets out the scenario the rest of the design answers.

1.1 The Question, the Data and the Regimes

When berth productivity drops on a working vessel call, the person accountable needs one ranked answer drawn from four candidate causes: a crane fault such as a hydraulic failure, a yard block that could not keep up, the vessel's stowage, or the berth window the authority granted. A terminal that can rank those causes daily belongs to the family of industrial ontology examples: reasoning held in country, on the operator's own hardware, projected over the systems of record rather than copied out of them. The data the ranking requires already exists, but only as separate facts in separate systems: moves with their seconds and equipment identity, fault codes with clearing times, yard occupancy and reshuffle ratios, stowage plan references, window grants against actual arrivals, work orders with their berth plan effect, and the commissioning status of newly installed cranes.

The regulatory regimes shape the design as much as the data does. Marine casualty investigation standards require timely, evidence-based reconstruction of what equipment did and when (IMO 2008),

and a fatal injury at a port shows how such reconstruction depends on joining equipment behavior to the sequence of operations (TAIC n.d.). Port facility security obligations put a named officer's approval on anything touching the secured area. National cybersecurity and communications controls demand auditable segmentation of operational technology. The kingdom's personal data protection law sets transfer conditions, including its Article 29 assessment, before gang, incident and driver records reach group ports. Export control rules attach to the accelerator class the reasoning runs on.

1.2 The Documented Cost

The cost of an unanswered attribution question is not one bad shift but a recurring one. Ports carry national weight: ports deliver 95 per cent of the country's imports and exports, including much of its food, fuel and materials, according to the ports safety code (Service 2025), so every unexplained fall in berth productivity propagates into supply chains rather than stopping at the quay. The safety record compounds the same failure, because accidents are documented to be underreported to vessel accident databases (NIH 2011), and an incident register that cannot be joined to the equipment and the vessel call it involved leaves every investigation to rebuild the same links by hand, under the timeliness the investigation code expects (IMO 2008).

1.3 The Operation as a Scenario

The operator is a container terminal operator in Saudi Arabia. It berths container ships against a quay served by ship to shore cranes, moves boxes with yard cranes, straddle carriers and terminal tractors, and hands them to road and rail at the gate. A terminal expansion under way lengthens the quay, deepens the berths, adds ship to shore cranes and a new yard block, and raises capacity toward two and a half million containers a year; by the operator's own assessment the expansion roughly doubles the objects under management and adds a second builder's equipment carrying its own vocabulary. The people in the loop are operations managers, engineering managers, maintenance planners, berth planners, the port facility security officer and the authority's vessel traffic staff; the planners who know which crane behaves how are few, and name reconciliation happens in meetings today. The physical environments are the quayside in Gulf heat and salt air, the yard blocks, the terminal machine room, and the road and rail gate. The operation runs eight named systems of record, counted roles spanning operations, engineering, maintenance, planning, security and the authority, and counted places spanning berths, yard blocks, the gate and the machine room, and the design in Part II stands on those counts.

PART I · CHAPTER 2

Every System Sees One Slice of the Operation

Each system of record sees its own part of the answer and none sees the whole.

Chapter 1 stated the question and priced its absence. This chapter shows why none of the existing systems can answer it: each sees one slice of the operation, and each names the same physical thing differently.

2.1 What Each System Sees

The terminal operating system holds vessel calls, moves, stowage and yard inventory, which makes it the productivity record, but it never sees the fault code that slowed the crane, the window the authority actually granted, or the commissioning state of the newest machine. The vessel scheduling and berth planning tool owns the berth plan, the quay length and the crane split, yet cannot see whether the crane it assigned is under repair. The crane and equipment control systems own machine state and fault codes, but not the move sequence around each fault or the yard pressure behind it. The maintenance management system owns work orders and component history, but not what any work order did to berth productivity. Equipment telematics carries fuel state, utilization and position, but cannot join a drift signal to the call it affected. The port authority's vessel traffic and berth window data holds the granted window, but nothing about performance inside it. The civil works and commissioning records hold the expansion's objects and the builders' cited documents, but nothing about how those objects perform once handed over. The safety incident register holds incidents, but not, on its own, the equipment component and the vessel call each incident belongs to. Figure 2 sets these slices side by side.

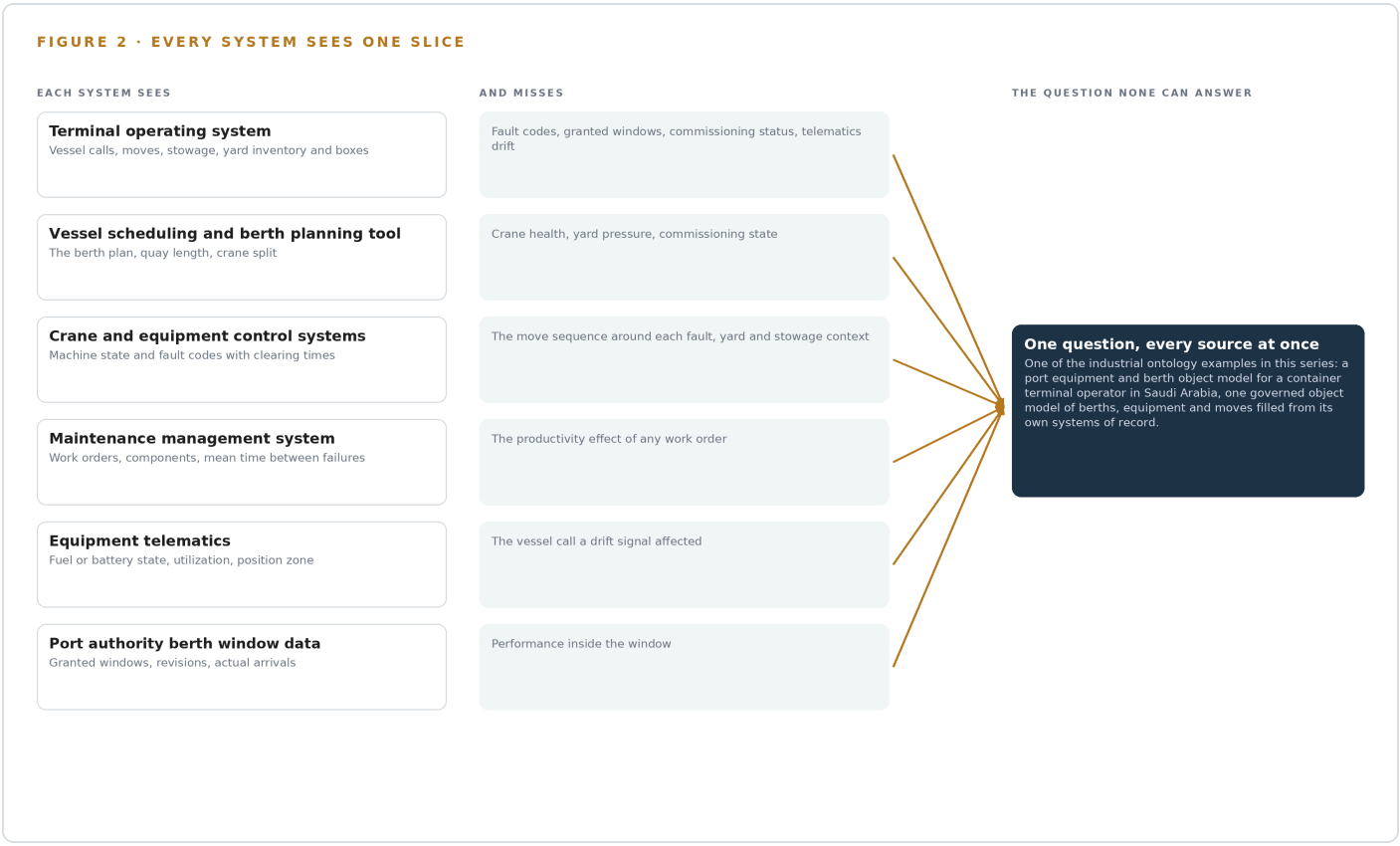


Figure 2. Eight systems, each seeing one part of the answer. The question needs all of them in one place at once.

2.2 What None of Them See Together

The question of Chapter 1 crosses all eight slices at once, and no system crosses even two. In practice the cost is paid in meetings: managers resolve names by hand, because a crane is an asset in maintenance, a resource in operations, a controller in automation and a line item in the commissioning records. The few planners who know which crane behaves how become a single point of failure for the whole operation, and the expansion doubles the object count while the second builder's vocabulary reopens the reconciliation the first builder's equipment already needed, so the manual join gets harder at exactly the moment productivity matters most.

PART II · CHAPTER 3

Four Constraints Shape Everything Downstream

The goals, the scoping decisions and the exclusions were fixed before any component was chosen.

Chapter 2 showed eight systems, each holding one slice and none able to join even two. This chapter fixes the goals and exclusions that every component choice in Part II answers to.

3.1 The Systems of Record Stay in Place

The design sits beside the systems of record, never replacing them, never modifying them, never loading their databases out. This is hardest exactly where it matters most: the terminal operating system is the operational clock the design reads productivity from, and the crane and equipment control systems are safety-rated, so writing to them is not a shortcut but a boundary breach. The expansion runs alongside live operations, which removes any window in which systems could be swapped.

3.2 The Weights Stay in the Country

Open weight models run on the operator's own hardware inside the country, and nothing about cargo, vessels or equipment leaves the network. The easy path in this industry is a hosted frontier API, which the requirement forbids; the sovereign path carries real friction, because the accelerator class the reasoning needs attracts an export license determination for the destination country, national cybersecurity controls demand auditable segmentation of the operational technology zone, and the operator, not a vendor, carries the hardware.

3.3 Named People Decide Every Write

Agents draft, managers confirm, and the single write path back to a system of record is an approved draft, raising a work order under the confirmer's name or adjusting the berth plan through the planning tool. The difficulty is behavioral rather than technical: approvals drift into rubber-stamping, and a false confirmation

poisons the memory the system improves on, so approval latency is instrumented and known-answer probes are injected into the draft queue from the start of phase one.

3.4 Improvement Comes From Memory, Not Training

The system improves by reading a memory of confirmed outcomes before each run, never by updating weights. This inverts the standard machine learning loop, whose natural answer is fine tuning, and it holds because the requirement says the next call and the next crane must be modelled better without training a model on the fly; the constraint also protects the boundary, since no training pipeline means no data ever leaves for one.

3.5 Scoping Decisions

Table 1 records the three decisions that bound the work.

Table 1 · Scoping Decisions

DECISION	WHAT IT BUYS	WHAT IT COSTS
Prove the object model on two berths before any agent runs	A first gate that can stop the work cheaply, with joins validated against production data	Value is held back from the rest of the terminal until the gate passes
Commit to existing telematics and fault logs, with continuous crane sensing quoted as an option	A committed price built on signals the operator already owns	If drift resolution proves too coarse, the sensing tier is exercised later rather than in place
Write back only through named approvals	An auditable decision record and a memory built on true confirmations	Every draft waits on a person, so approval latency becomes a managed quantity

3.6 What the Design Chose Against

Table 2 records the rejected alternatives and the exclusions that close the scope.

Table 2 · What the Design Chose Against

WHERE	WHAT WAS PICKED	INSTEAD OF, AND WHY
Reasoning	GLM 5.3 self hosted on one 141 GB class node at FP8	A hosted frontier API or a mid-size dense model: the operator names GLM 5.3 and forbids third party services, and one node holds the FP8 weights with room for KV cache

WHERE	WHAT WAS PICKED	INSTEAD OF, AND WHY
Improvement	A kept record of decisions and outcomes that the next decision reads: memory of confirmed outcomes	Fine tuning on confirmed traces: the requirement states the next call and the next crane are modelled better without training a model on the fly
Control system egress	A hardware data diode enforcing one-way transfer of fault logs and telematics	A firewalled bidirectional conduit: the control zone stays strictly read-only, and the diode settles the argument in hardware rather than in policy
Crane sensing	Existing telematics and fault logs, with a camera tier on critical cranes offered and priced separately	Committed continuous crane vision: existing signals get a fair test first, and the offered tier carries its own gate
Scope boundary	Equipment control changes, model training, third party model service, group ports, hardware and infrastructure, historic migration and round the clock cover are out of scope	A broader build: control changes breach the safety boundary, training breaches the improvement doctrine, group ports wait on the phase one gate, and hardware, vendor fees and support beyond working hours stay with the operator

PART II · CHAPTER 4

One Stack Runs From the Systems of Record to the Surfaces

Systems of record below, one object model in the middle, services and surfaces above.

Chapter 3 fixed the four constraints and the scoping decisions they force. This chapter turns those decisions into a stack, in a fixed order from the systems of record up to the surfaces the managers watch.

4.1 A Projection Over the Systems of Record

The pattern is a projection, not a replacement. The eight systems of record stay below, untouched and still authoritative, each keeping its own names and its own duties. Above them sits one object model of 16 objects, filled move by move and never holding a shadow copy of any record, and above the model sit ten services, four surfaces and three models. This is the shape that sovereign AI reference architectures converge on when cargo data cannot leave the country: bind the records where they are, govern one vocabulary over them, and keep the inference inside the boundary on owned hardware. Figure 3 shows the layered stack with the count at each layer, and Table 7 names the component chosen at each layer and the reason for it.

FIGURE 3 · THE LAYERED STACK: 4 SURFACES, 10 SERVICES, 16 OBJECT MODEL, 3 INFERENCE, 2 ADAPTERS, 10 SENSING, 9 SOURCES

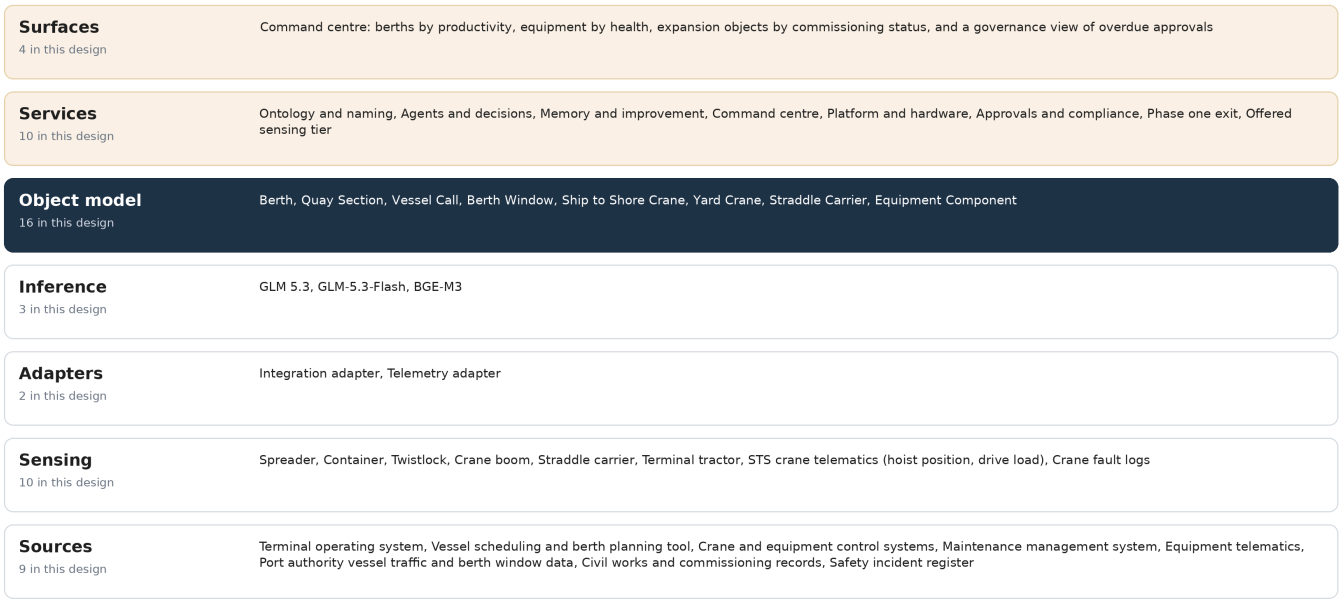


Figure 3. The layered stack: 9 sources, 2 adapter families, 16 objects, 10 services and 4 surfaces.

Table 7 · Architecture, Layer by Layer

LAYER	COMPONENT	CHOICE	WHY
Event streaming and messaging	by class, no product named	Product	Apache Kafka 4.3.x (KRaft)
The ontology or graph store	by class, no product named	Product	Hyper Ontology v0.9 (Semantics and Kinetics)
Time series and telemetry storage	by class, no product named	Product	InfluxDB 3 Core
Video ingest, recording and management	by class, no product named	Product	MediaMTX
Edge model serving	by class, no product named	Product	OpenVINO
On site language model serving	by class, no product named	Product	vLLM
Object tracking and trajectory forecasting	by class, no product named	Not needed	No tracking duty; moves carry equipment identity from the TOS and ECS.
Edge orchestration of containers	by class, no product named	Not needed	Phase one runs on the operator's central hardware; only the offered crane tier would add gateways.

LAYER	COMPONENT	CHOICE	WHY
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 or distilled into weights; memory of confirmed outcomes is the improvement path the requirement names.
Time synchronisation across sensors and systems	by class, no product named	Product	Chrony (NTP) with linuxptp
Observability of the running system	by class, no product named	Product	Prometheus 3.13 LTS with Grafana and Loki

4.2 The Stack Stage by Stage

Table 3 walks the seven stages, what each is responsible for, and how. The Sources stage is the eight operator systems, all read only, named with their provenance in Chapter 6. The Sensing stage turns four feeds into ordered streams: crane telematics with hoist position and drive load, crane fault logs, move events and the authority's berth window feed, with an offered camera tier of three fixed cameras priced as an option. The Adapters stage is two families, an integration adapter over APIs and read-only views and a telemetry adapter into the time series store, the control systems reached only through a hardware data diode. The Object model stage holds the 16 objects in a governed graph store keyed to one naming service, so a crane is one thing whichever system it arrived from. The Inference stage runs GLM 5.3 at FP8 on one node of eight 141 GB GPUs under vLLM, with GLM-5.3-Flash on the builders' commissioning documents and BGE-M3 on retrieval. The Services stage carries ten services, from ontology and naming through agents and decisions, memory and improvement, command center and approvals to the scaling workstreams. The Surfaces stage is four command center views: berths by productivity, equipment by health, expansion objects by commissioning status, and a governance view of overdue approvals.

Table 3 · The Stack, Stage by Stage

STAGE	WHAT IT IS RESPONSIBLE FOR	HOW
Sources	Hold the authoritative records the model projects	Eight operator systems, all read only, from the terminal operating system to the safety register
Sensing	Turn machine and authority feeds into ordered streams	Crane telematics, fault logs, move events, berth window feed; offered tier adds three fixed cameras
Adapters	Own each source's contract and provenance	Integration adapter on APIs and read-only views; telemetry adapter to the time series store; control systems via data diode

STAGE	WHAT IT IS RESPONSIBLE FOR	HOW
Object model	Keep one governed vocabulary of 16 objects	Graph store filled move by move, keyed to the naming service
Inference	Read the model and the builders' documents	GLM 5.3 at FP8 on one node of eight 141 GB GPUs under vLLM; GLM-5.3-Flash and BGE-M3 beside it
Services	Draft, confirm, remember and scale	Ten services, from ontology and naming and agents and decisions to command center and approvals
Surfaces	Show managers what to decide on	Four command center views: productivity, equipment health, commissioning status, overdue approvals

4.3 A Reusable Pattern

The pattern underneath this design is one a team carrying a similar brief can reuse: systems of record below, one adapter per system family, one object model in the middle, and services and surfaces above that resolve every answer back to that model.

In this design the shape holds 9 sources, 16 objects, 4 surfaces. What a reader swaps is the sources, the objects and the models; what they keep is the adapter tier, the one object model and the rule that inference runs where its latency budget allows.

PART II · CHAPTER 5

Sixteen Objects Turn Operational Data Into One Argument

The object model is the join the systems never made.

Chapter 4 ended the stack at the object model, the one layer every service reads and no source system writes. This chapter opens that layer: what it holds, where people enter it, and the exact form one object takes in the record.

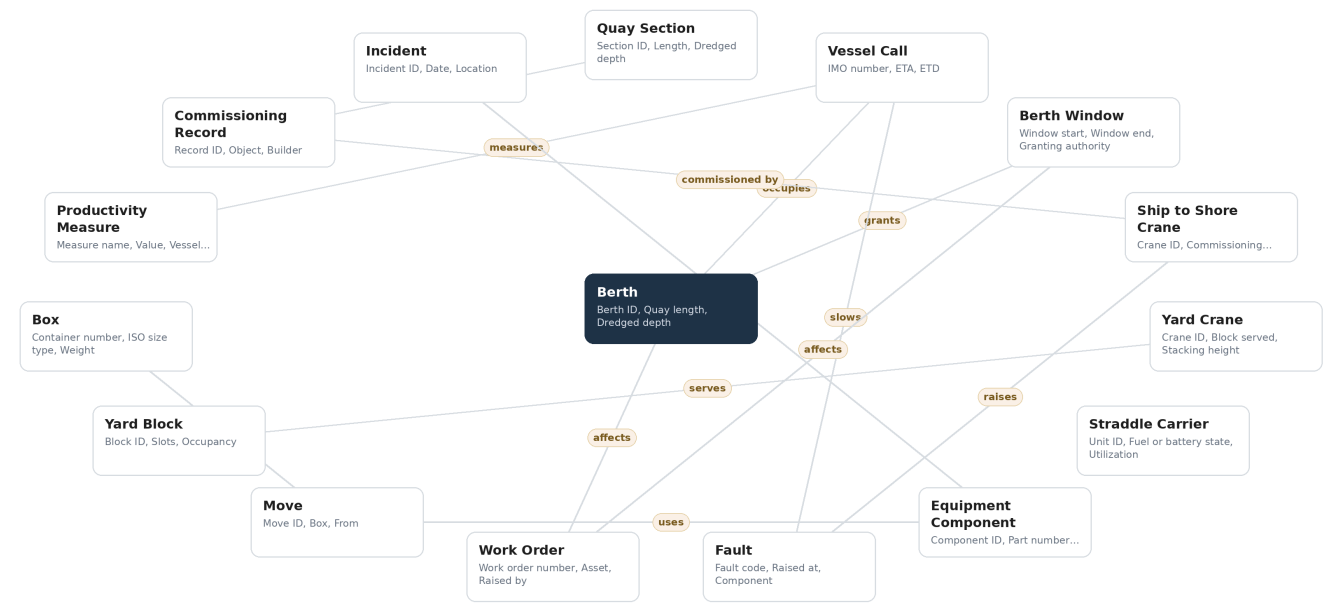
5.1 Sixteen Objects and the Links Between Them

The model holds sixteen objects in six kinds: three sites (Berth, Quay Section, Yard Block), four assets (Ship to Shore Crane, Yard Crane, Straddle Carrier, Equipment Component), five events (Vessel Call, Berth Window, Fault, Move, Incident), two records (Work Order, Commissioning Record), one material (Box) and one measure (Productivity Measure). Each object is anchored in exactly one system of record, the system whose word on that object is final: the Berth in the vessel scheduling and berth planning tool, the Move and the Box in the terminal operating system, the Fault in the crane and equipment control

systems, the Incident in the safety incident register. Table 8 sets out every object with its anchor and its typed links. Anchoring is what keeps the model a projection over the systems of record rather than a second copy of them: the ontology claims nothing the sources do not hold, and it never becomes the place a record is kept.

The typed links are the join the systems never made. Move carries Box and names Equipment Component; Productivity Measure names its Vessel Call; a Work Order records its berth plan effect against both Berth and Berth Window; an Incident names its Equipment Component. A query can therefore start at a berth's productivity measure, walk back through the vessel call to the individual moves and the equipment that performed them, continue to the faults that equipment raised and the work orders those faults produced, and end at the berth window the authority granted, all in one traversal. Figure 4 draws every object and every link. A document store holds each of these facts somewhere, but it cannot answer the operator's question, whether the drop was the crane, the yard, the stowage or the window, because that answer lives in the joins, and typed links are the one structure that carries them. The same traversal is why the Incident object matters: maritime accidents are documented to be underreported to official databases (NIH 2011), and an incident that joins to its equipment and its call in the same model as the fault codes is an incident that can be found.

FIGURE 4 · THE OBJECT MODEL: 16 OBJECTS AND 13 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 sixteen 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
Berth	Vessel scheduling and berth planning tool	grants: Berth Window
Quay Section	Civil works and commissioning records	carries: Commissioning Record
Vessel Call	Terminal operating system	occupies: Berth
Berth Window	Port authority vessel traffic and berth window data	no typed link recorded
Ship to Shore Crane	Crane and equipment control systems	commissioned by: Commissioning Record; raises: Fault
Yard Crane	Crane and equipment control systems	serves: Yard Block
Straddle Carrier	Equipment telematics	no typed link recorded
Equipment Component	Maintenance management system	no typed link recorded
Fault	Crane and equipment control systems	slows: Vessel Call
Work Order	Maintenance management system	affects: Berth; affects: Berth Window
Move	Terminal operating system	carries: Box; uses: Equipment Component
Yard Block	Terminal operating system	no typed link recorded

5.2 The Human Loop and the Hosting Posture

People enter the model at the confirmation step. The three drafting agents, for productivity, equipment and commissioning, propose; managers confirm; and the single write path out of the ontology is the named approval: an approved work order lands in the maintenance management system under the confirmer's name, and an approved berth plan adjustment lands in the planning tool. No agent dispatches equipment, and no agent writes to a control system. The hosting posture follows the same line: the model runs on the operator's own hardware inside the country, identity is issued on site so every confirmation carries a person, the port authority feed is the one inbound external link, and a hardware data diode carries fault logs out of the control zone one way. Nothing about cargo, vessels or equipment leaves the network, the boundary that sovereign AI reference architectures in regulated industries draw.

5.3 One Object in Its Recorded Form

The Berth, the object a productivity argument starts from, is recorded in the form the platform prints beneath this section: identifier, label, kind, properties, status vocabulary and links; the Ship to Shore Crane, the object every productivity argument passes through, is recorded the same way.

{ "id": "berth", "label": "Berth", "kind": "site", "anchored_in": "Vessel scheduling and berth planning tool", "properties": ["Berth ID", "Quay length", "Dredged depth", "Crane split"], "status_vocabulary": [], "links": [{ "to": "berth-window", "label": "grants" }] }

PART II · CHAPTER 6

Every Source Enters Through an Adapter, Never Directly

Nothing reaches the object model except through an adapter that owns its contract.

Chapter 5 showed sixteen anchored objects with typed links between them. This chapter shows how data reaches them: every source enters through an adapter that owns its contract, and nothing reaches the object model directly.

6.1 Every Named Source and Its Provenance

Eight named source systems feed the model. The terminal operating system holds vessel calls, moves, stowage and yard inventory, the productivity record the ontology is built on. The vessel scheduling and berth planning tool owns the berth plan that approved drafts adjust. The crane and equipment control systems own fault codes and machine state, read through a one-way path with no write back. The maintenance management system owns work orders, and receives the ones approvals raise. Equipment telematics is the drift signal the equipment agent triggers on. The port authority's vessel traffic and berth window data supplies the granted window, one of the four ranked causes of a productivity drop. The civil works and commissioning records hold the expansion's objects and the builders' cited documents. The safety incident register feeds the Incident object, the record safety investigations are built on under the international code for casualty investigation (IMO 2008). All eight carry first-party provenance, systems the operator or its authority owns, named by role because their products are confirmed on site; all are read except the single approved write back to work orders. Figure 5 maps each system, its provenance class and its adapter path.

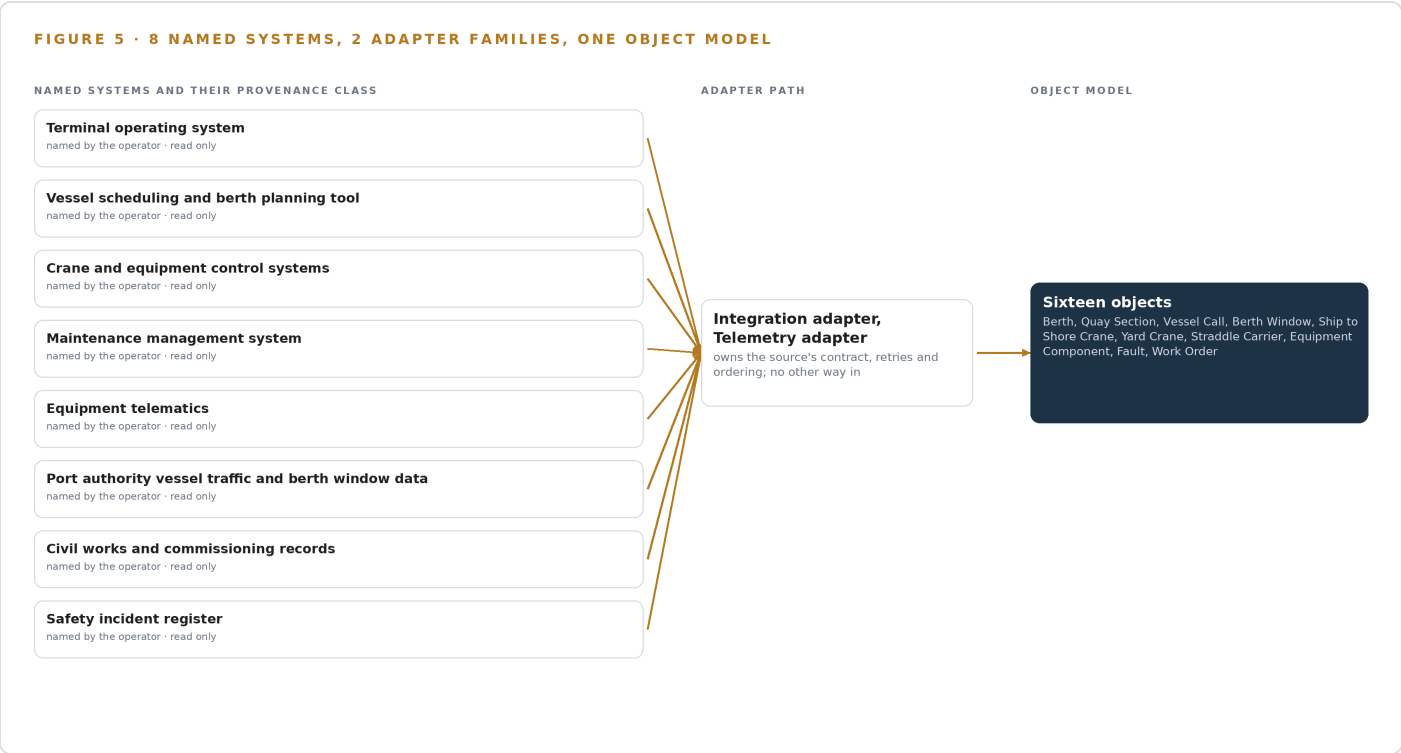


Figure 5. The 9 named systems, the adapter path each one takes, and the object model they all map into.

6.2 What the Adapter Tier Guarantees

Two adapter families cover the eight systems. The integration adapter serves the transactional sources, the operating system, the planning tool, the maintenance system, the commissioning records and the incident register, over APIs and read-only views built per system. The telemetry adapter serves the high-rate sources, the control systems' fault logs and the equipment telematics, and carries the hardware data diode that makes the control zone read-only in silicon rather than in policy. The tier guarantees five things: each adapter owns its contract, so a source change breaks one adapter and not the model; every event is stamped with its source, provenance class and read time; source names are resolved to governed object identities, so the second builder's vocabulary maps rather than invents; events are buffered and replayed when a link fails; and no adapter ever writes to a source except the approval path already described.

6.3 The Event Backbone

The backbone is Kafka 4.3.x in KRaft mode. Ordering is per key: move events are keyed by vessel call so a call's moves arrive in sequence, and fault events are keyed by crane so a fault history reads true. Delivery is at least once, made safe by idempotent upserts keyed on object identifiers, so a replayed event converges on the same record instead of duplicating it. Brokers replicate, adapters buffer against source outages, and the terminal's own network carries everything committed. Time is the quiet dependency: operating system events, control system fault logs and telematics join on one clock disciplined by Chrony and linuxptp under an OCP Time Card grandmaster, because a fault joined to the wrong shift is worse than a fault not joined at all.

Inference Belongs Where the Data Is

Each tier runs what its data and its link can carry.

Chapter 6 described how every source enters through an adapter and never directly, and how the event backbone carries each record to the object model in order. This chapter decides where the models that read those records run, and why each tier of hardware carries only what its data and its link can support.

7.1 Three Tiers and the Arithmetic That Fixed Them

The committed design runs inference on three tiers, and the memory arithmetic of the frontier tier fixed the whole placement. The central tier holds GLM 5.3, the frontier-class work surface model, on the operator's own hardware in country: one node of eight GPUs of the 141 GB high bandwidth memory class, which is 1,128 GB of usable memory. The model's filed parameter count is 753 billion, served at FP8 precision, where each parameter occupies one byte, so the weights alone occupy 753 GB. The 1.2 planning factor reserves a further 151 GB for the KV cache, the per-conversation attention memory a running model allocates, and for activations, so the node must hold 904 GB in total; it holds that with margin, leaving 375 GB beside the bare weights. Serving runs on vLLM, and because the node is inside the country and owned by the operator, no reasoning request crosses a border.

The terminal tier hosts GLM-5.3-Flash on a two-socket air-cooled server class in the terminal machine room, reading the builders' commissioning documents and carrying retrieval volume so the frontier node stays reserved for reasoning. The embedding tier runs BGE-M3 on the central node. A fourth tier is offered, not committed: continuous crane sensing, which would add edge compute of the fanless, sealed, sunshielded class rated for Gulf quayside heat, serving detection through OpenVINO, with a development-board measurement as the gate before any unit is fixed. Figure 6 sets the four tiers side by side, with what runs on each and what crosses between them.

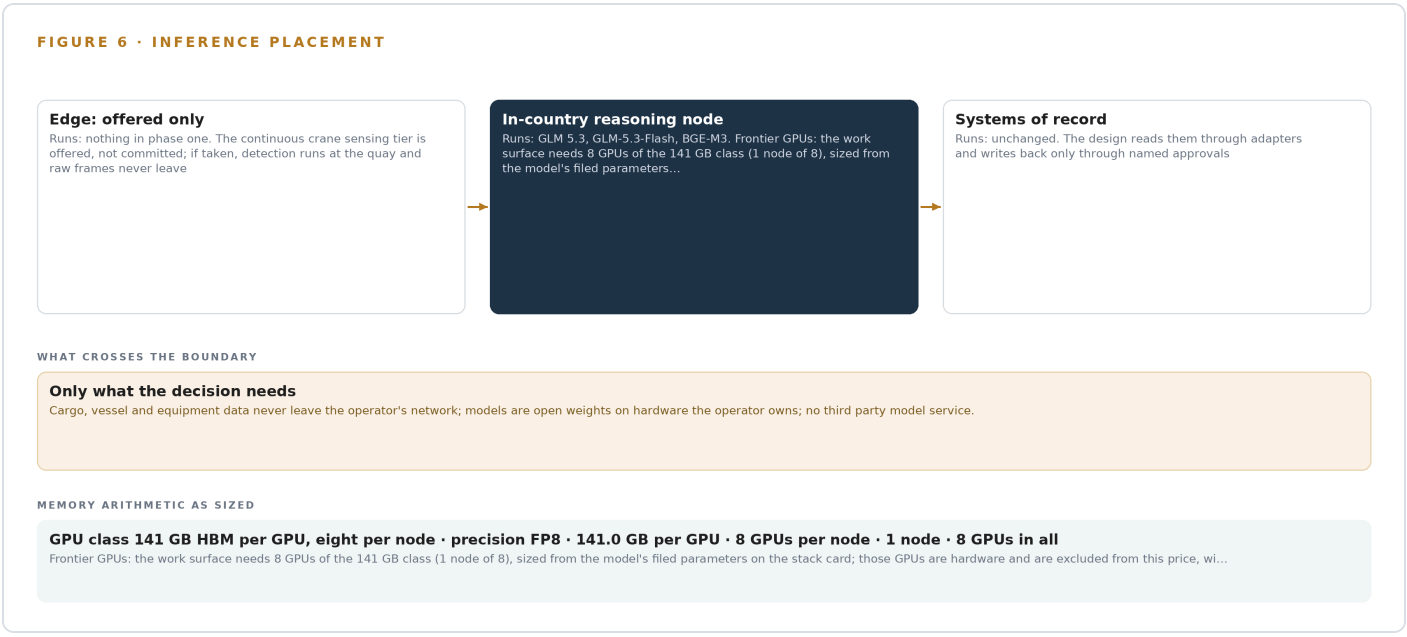


Figure 6. Where each tier runs, what runs there, and the narrow set of outputs that cross the boundary.

7.2 The Latency Budget

The budget is set by the decision each output serves, not by a number quoted for a component. Move events must join in stream order on the backbone, because a productivity argument assembled out of order is wrong, not merely late. The productivity question itself is answered after the vessel call, on the terminal operating system's own clock, because that is the clock the measure is defined against. Drift signals from telematics must surface within the shift, so a superintendent can act while the crane is still the problem of this shift rather than the next one's investigation. Commissioning document reading is working-hours work, and conversational retrieval answers within a session. Nothing committed needs a model to answer faster than a manager can read it, because agents draft and managers decide; the one latency the ontology itself carries is event ordering, which time synchronization through Chrony with linuxptp and the backbone's ordering guarantee hold.

7.3 The Boundary, and What Happens When It Fails

What crosses the network boundary crosses it inward, and only as a read. The crane and equipment control systems are read through Lidi over a hardware data diode, so fault logs and machine state leave the control zone and nothing can return: the control-zone segmentation required of operational technology is satisfied by construction rather than by firewall policy that a misconfiguration could undo. Nothing about cargo, vessels or equipment leaves the country, because the frontier node is the operator's own hardware in country and no third party model service is called anywhere in the design.

The failure paths are each specified. When the link between the terminal and the central node drops, the backbone buffers events and replays them in order on recovery, so the object model catches up rather than gaps. When power fails, each tier restarts against its last committed offsets and time resynchronizes through Chrony with linuxptp, so joins made after restart still line up by timestamp. When the update path

is the failure, model artifacts arrive through Harbor, the air-gapped artifact and model registry: they are staged, verified and applied on the operator's schedule rather than pulled live from outside. The systems of record are never the point of failure for the intelligence, because the intelligence writes to none of them, and the one write path that exists goes through named approvals into the maintenance system and the berth plan, never into the control zone.

PART II · CHAPTER 8

The License Decides What the Operator Can Own

Every model was chosen against its license before its accuracy.

Chapter 7 fixed where each tier runs and what each node must hold. This chapter turns to the models themselves, and to the licenses that decide whether the operator can keep what it runs: a model the operator cannot legally hold on its own hardware is not a sovereign choice, however accurate it is.

FIGURE 7 · THE MODEL STACK: 3 MODELS, THEIR PLACEMENT AND THEIR WORK

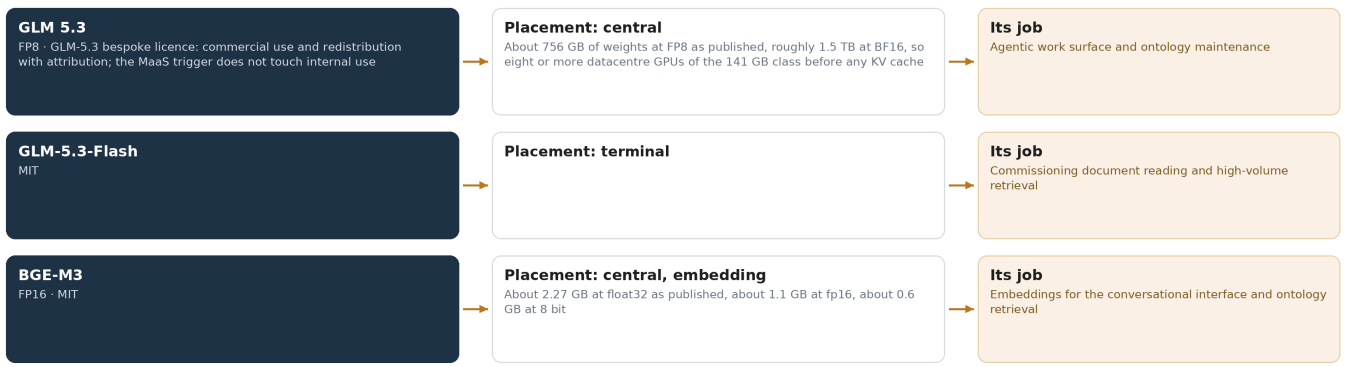


Figure 7. The three models, their placement, and the work each one does.

8.1 Three Models and the Terms That Bind Them

Figure 7 sets the model stack on one page: a frontier reasoning model on the central node, a multimodal reader on the terminal tier, and an embedding model beside the reasoning model, each with its license written under it. GLM 5.3 is the work surface model, the frontier-class reasoning tier where the drafting agents work and the ontology is maintained. Its filed parameter count is 753 billion, served at FP8, where each parameter occupies one byte: about 756 GB of weights as published, and roughly 1.5 TB if carried at BF16, the two-byte precision, which is why the design serves it at FP8 on one node of eight 141 GB GPUs with the KV cache ceiling Chapter 7 sets out. Its license is the GLM-5.3 bespoke license, whose terms permit commercial use and redistribution with attribution, and whose model as a service trigger governs offering the model as a service to others; it does not touch a terminal operator's internal use. The

operator names this model and forbids third party services, and this license is what lets the operator hold the weights, host them in country and keep them.

GLM-5.3-Flash is the terminal tier reader: 320 billion total parameters with 18 billion active per token, a mixture of experts shape that carries a large vocabulary of knowledge at a small serving cost, and natively multimodal, so it reads scanned drawings and photographed pages of the builders' commissioning records as well as text. It carries the MIT license, which permits use, copying, modification and redistribution with the license text carried and carries no trigger of any kind, so the operator can hold and redistribute the weights without a determination. Its work is deliberately volume work: reading documents at the rate the expansion produces them, and leaving the frontier node to reasoning.

BGE-M3 is the embedding model, the piece that turns text into the vectors search runs on. It performs dense, sparse and multi-vector retrieval in a single pass over an 8,192 token context, occupies about 1.1 GB at FP16 on the central node, and carries the MIT license. The sparse leg matters more than the size: container numbers and fault codes are exact strings, and a purely dense vector blurs them, while the sparse leg keeps them retrievable exactly.

8.2 The Model and Equipment Register

The license is the line of sovereign AI reference architectures most often left vague: which weights the operator may hold, on what ground, under whose trigger. Table 4 closes that vagueness by registering not only the three models but everything the design stands on: the GPU class and the sizing rule that fixed the node count, the diode and the thermal camera legality line, the sensing the offered tier would add and the telemetry signals the committed design already holds, the patterns the architecture rests on, and the ground it all runs on. One entry deserves its emphasis stated plainly: the improvement path is a kept record of decisions and outcomes, a memory of confirmed outcomes which the next trace reads before it acts and which does not update model weights, so no license with a training or distribution trigger can ever be tripped by the system getting better at its job.

Table 4 · Model and Equipment Register

THE CHOICE	WHAT WAS PICKED	WHY HERE
Work surface model	GLM 5.3, 753 billion parameters at FP8, served with vLLM on the frontier node	Named by the operator; the bespoke license permits commercial use and redistribution with attribution and its MaaS trigger does not touch internal use
Document reading model	GLM-5.3-Flash, 320B total with 18B active, natively multimodal, on the terminal tier	MIT terms let the operator hold the weights outright; it reads commissioning documents at volume and keeps load off the frontier node

THE CHOICE	WHAT WAS PICKED	WHY HERE
Embedding model	BGE-M3 at FP16, dense, sparse and multi-vector retrieval over 8,192 tokens, on the central node	MIT; the sparse leg keeps container numbers and fault codes exactly retrievable
GPU class	141 GB HBM GPU class, eight per node	The node holds the FP8 weights, the KV cache reserve and headroom beside them
Sizing rule	Frontier Model GPU Footprint Sizing Rule for vLLM and SGLang	One byte per FP8 parameter, times the 1.2 factor for KV cache and activations, fixes the node count at one
One-way transfer	Lidi over a hardware data diode	Control-zone egress ends in hardware; read-only is a property of the wire, not of policy
Thermal camera legality	Any US-sourced thermal unit held at 9 Hz or less	Stays under ECCN 6A003.b.4.b, which otherwise requires a BIS license for the operator
Sensing	Cameras on hoist machinery, the quay crane rail and a yard block transfer zone, detecting spreader, container, twistlock, crane boom, straddle carrier and terminal tractor	Belongs to the offered crane tier only; existing telematics is the committed signal
Telemetry signals	STS crane telematics, crane fault logs, TOS move events and the port authority berth window feed	The drift and productivity arguments run on records the operator already holds
Pattern	System of context, with Hyper Ontology v0.9 Semantics and Kinetics	The ontology is a projection over the systems of record, never a shadow copy of them
Pattern	a work surface where the operator's people build their own agents with a kept record of decisions and outcomes that the next decision reads	Agents draft in a work surface inside the boundary, and memory of confirmed outcomes improves the next trace without training
Ground	The operator's own hardware, in country, on the terminal's own network	Cargo, vessel and equipment data never leave the country, and no third party model service is called

PART III · CHAPTER 9

Shadow Mode Comes Before Any Flag Is Trusted

The rollout earns each phase at a gate the operator can read.

Chapter 8 settled which models run, where they run and under what licenses. This chapter sets out how the build earns trust in stages, what each stage must prove before the next begins, and what the operator should still be asking about.

9.1 Three Phases, Eighteen Items, Two Hard Gates

The rollout is shaped by the requirement's own phasing, and Figure 8 shows it whole: three phases, 18 items, two gates, and the register of 21 requirements read against the build at every gate. Phase 1 covers two berths and carries 15 items across six workstreams: ontology and naming, agents and decisions, approvals and compliance, command center, memory and improvement, and phase one exit. Its named items include the port equipment and berth object model, the source system mappings and joins, and name reconciliation across systems. Its exit gate is the commissioning agent for expansion objects: the agent must draft commissioning records from the civil works and commissioning records for the new berth sections and cranes, and a manager must confirm them, before anything widens. Phase 2 carries two items across the offered sensing tier and terminal scale workstreams: the whole-terminal rollout and the continuous crane sensing option. Its gate is that option itself, taken on phase-one evidence rather than assumption, which is why the sensing tier is priced and quoted separately rather than built in. Phase 3 carries a single item, the group ports worked example, on the terminal scale workstream; the group's other ports inherit the worked example only when the running terminal has proven it, and that inheritance is the requirement's stated purpose, that the ontology becomes the worked example for the rest of the ports in the group.

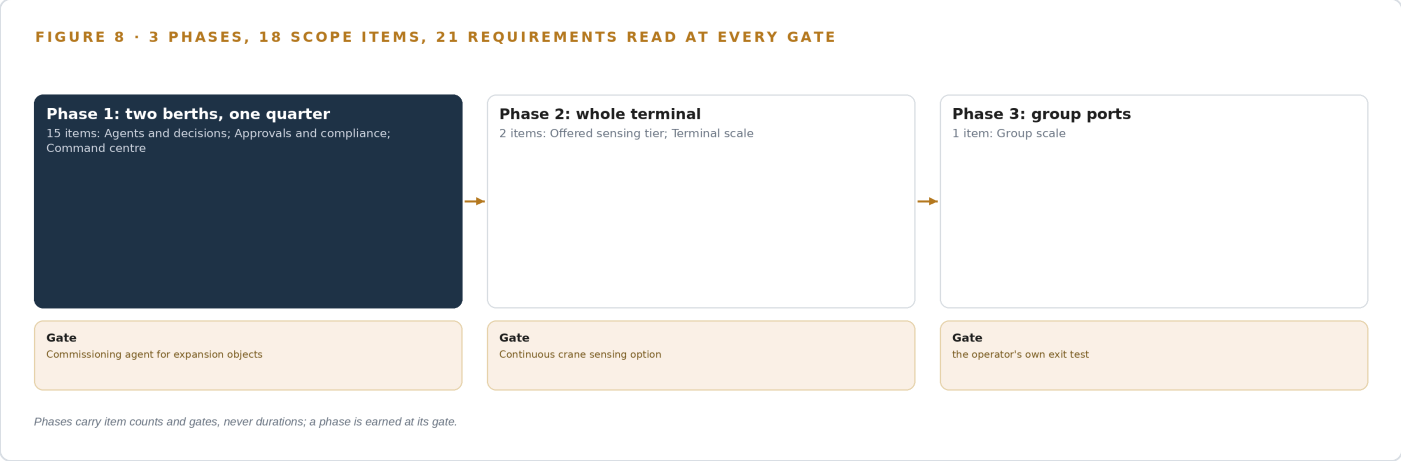


Figure 8. The three phases and their gates, and coverage of the 21 requirements across them.

9.2 What the Rollout Measures

The phase-one measurements are chosen so the gate reads evidence, not opinion. The object model is proven on one week of production joins before any agent runs: berths, windows, cranes, faults, moves and measures must join across the eight systems of record with every object resolved to one governed name. Approval latency is instrumented from the start, because the confirmations are the training data of the memory, and known-answer probes are injected into the draft queue so rubber-stamping shows up as

a number rather than a suspicion. Drift resolution is measured on the phase-one cranes at shift sensitivity, which decides whether existing telematics can carry the equipment agent's trigger or the offered continuous sensing tier is needed. Attribution is measured against confirmed outcomes: when a manager confirms a ranked cause among the crane's fault, the yard block, the vessel's stowage and the granted window, that confirmation is recorded and the next similar call must rank better. The incident join is measured too, and the join rate matters beyond the terminal because underreporting of maritime accidents to official databases is documented (NIH 2011), so a safety register that reaches equipment and vessel calls through the ontology is itself a finding.

9.3 Failure Modes

Table 5 states what fails and what the design does about it, and every row is a live risk with a named mitigation rather than a residual accepted in silence.

Table 5 · Failure Modes

WHAT FAILS	WHAT THE DESIGN DOES
TOS, ECS and telematics products are unnamed, so connector effort is unpriced	Confirm products and interfaces in the first week of phase one; scope the connectors by system role now
Approvals drift into rubber-stamping and poison the memory with false confirmations	Instrument approval latency and inject known-answer probes into the draft queue
The 141 GB GPU class needs an export license determination before order	Run the determination at phase one start and name the importer of record
Existing telematics may not resolve a hoist drift within the shift	Measure drift resolution on phase-one cranes; the continuous sensing tier is the fallback
The second builder's vocabulary diverges and reopens name reconciliation	Govern the vocabulary centrally so new builders map to it, not invent beside it
Personal data in gang, incident and driver records reaches group ports without a transfer basis	Complete the Article 29 assessment and safeguards before any group-wide rollout

9.4 Lessons

Shadow the drafts before anyone signs them. The agents begin in shadow mode, drafting productivity explanations, equipment health notes and commissioning summaries that managers read and correct before any draft can raise a work order or adjust a berth plan, and the probes in the queue keep the confirmation honest. Trust in a ranked cause is earned one confirmed explanation at a time, and the gate trusts nothing that a named person has not confirmed. Govern the vocabulary centrally or the second builder reopens the war. The expansion doubles the objects and adds a second builder's equipment with its own names, and the design answers with one naming service that every new object passes through,

so the builder's terms are mapped to the governed vocabulary instead of accumulating beside it. Price the connectors before promising the joins. The eight systems are named by role, not by product, and the join quality of the whole ontology rests on interfaces that are confirmed in the first week of phase one, which is why the system survey is an item with a gate behind it and not a formality.

9.5 What Is Still Open

Four questions remain open, and settling each changes something concrete. The system survey settles the connector effort and with it the true cost of the join layer. The export license determination for the 141 GB GPU class settles the license path the hardware order follows, and names the importer of record. The drift resolution measurement settles whether the offered continuous sensing tier moves from option to commitment. The Article 29 assessment under the personal data protection law settles what, if anything, may flow to the group's other ports in phase 3 beyond the worked example itself.

PART III · CHAPTER 10

The Intelligence Should Stay with the Operation That Produced It

What the design builds belongs to the operator.

Chapter 9 showed how the build earns each phase at a gate the operator can read. This chapter settles the quieter question behind the gates: who owns what the build produces.

10.1 The Model, the Weights and the Record Belong to the Operator

The object model is the operator's asset in the fullest sense: 16 objects, from Berth and Vessel Call through Ship to Shore Crane, Fault, Move and Commissioning Record, each anchored in one named system of record, linked by typed joins, and named through one governed vocabulary. Nothing in the model is a shadow copy of a vendor's database; it is a projection over systems that stay in place, which means it cannot be repossessed by any of them. The requirement asks that this model become the worked example for the rest of the ports in the group, and that is a statement of ownership before it is a statement of reuse. The weights are equally the operator's to hold: GLM 5.3 runs under its bespoke license, whose commercial terms and attribution requirement permit self-hosted internal use without touching its model as a service trigger, and GLM-5.3-Flash and BGE-M3 run under MIT, so the operator holds every model it depends on, on its own hardware, in country. There are no fine-tunes to own, and that is deliberate: improvement runs through the memory of confirmed outcomes, which records what managers decided and never touches a weight. The decision record is the third asset: every approved draft carries the confirmer's name, every raised work order is raised under that name in the maintenance management system, and the accumulated memory of confirmed explanations stays inside the operator's boundary as the corpus the next decision reads. The boundary itself is the fourth: a hardware data diode

enforces one-way egress from the control systems, nothing about customers' cargo leaves the country, and the single write path into any system of record runs through a named approval.

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: Hyper Ontology holds the governed object model of berths, equipment and moves; Hyper Pragma is the work surface where the operator's own people extend the drafting agents in the command center; Decision Systems is the ranking and attribution layer that puts four candidate causes in front of a named manager who decides; Adaptive Operations is the offered sensing tier that would watch crane behavior continuously; and Sovereign Infrastructure is the posture the whole design stands on, with open weights on the operator's own hardware inside the country and hardware-enforced one-way egress from the control zone. The intelligence stays with the operation that produced it, and the platform that produced the design is the reason it could be specified that way.

PART IV · CONCLUSION

The Pattern Repeats Wherever Systems of Record Do Not Meet

Seen whole, the design is one object model over the systems a container terminal operator in Saudi Arabia already runs, with inference placed where the data is and every decision left to a named person.

Running the same shape elsewhere takes the same four things: the named systems, an object model anchored in them, a first gate that can stop the work cheaply, and the license posture that lets the operator own what it learns.

PART IV · CHAPTER 11

How Praxis Contextualized and Reasoned This Design

Every choice above traces back to something Praxis recorded reading.

Chapter 10 settled where the intelligence lives. This chapter opens the record behind it, so a reader can trace any choice above back to what justified it. Every design in this series is produced on Praxis, the design team platform for designing physical AI systems, and this chapter lets a reader audit the reasoning rather than take it on faith. Figure 9 shows the ask as Praxis received it, the family and industry it assigned, what it read, the eight lenses it read through, the patterns it adopted and set aside, and the equipment classes the reasoning landed on.

11.1 Contextualizing the Ask

The ask, in the operator's own words, was an industrial ontology of the terminal, owned by the operator, filled from its systems of record, with open weight models on its own hardware in the country, and nothing about customers' cargo leaving it. Praxis assigned the design to the Physical AI family and the Maritime and Ports industry, a pin made by the forwarding design engineer on this project and read as reasoning input rather than a matched conclusion. What Praxis read was the corpus shelf of each lens, and those shelves are the corpus behind every lens that follows.

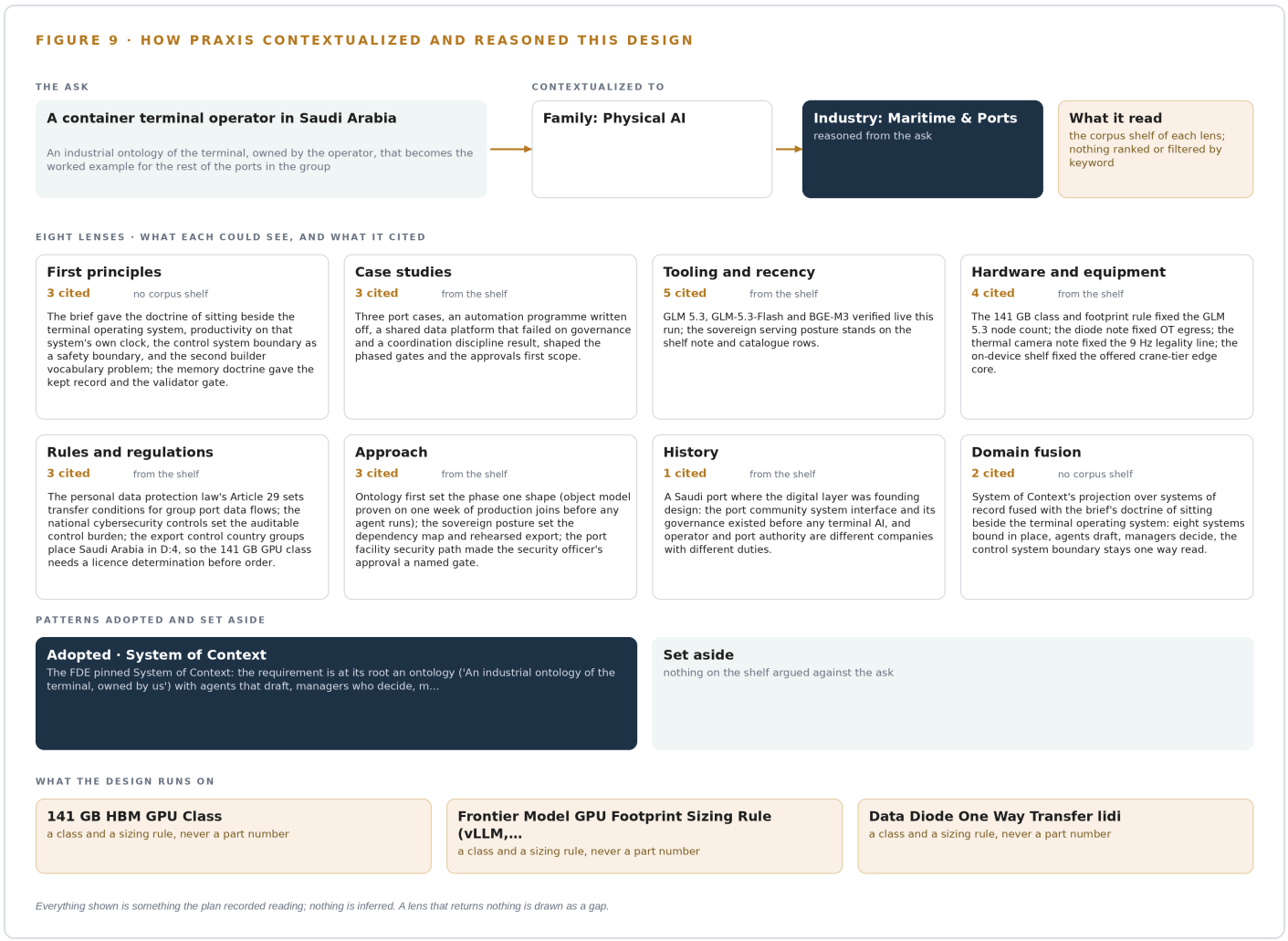


Figure 9. From the ask to the design: the family and industry Praxis assigned, the eight lenses and what each cited, the patterns adopted and set aside, and the equipment the design lands on.

11.2 The Lenses

Table 6 records each lens, what it could see, how many items it cited, and what it contributed to the design. All eight lenses returned a contribution on this project; none is shown as a gap.

Table 6 · The Lenses and What They Contributed

LENS	COULD SEE	CITED	WHAT IT CONTRIBUTED
First principles	no corpus shelf	3	The brief gave the doctrine of sitting beside the terminal operating system, productivity on that system's own clock, the control system boundary as a safety boundary, and the second builder vocabulary problem; the memory doctrine gave the kept record and the validator gate.
Case studies	the case study shelf	3	Three port cases, an automation programme written off, a shared data platform that failed on governance and a coordination discipline result, shaped the phased gates and the approvals first scope.
Tooling and recency	the tooling and recency shelf	5	GLM 5.3, GLM-5.3-Flash and BGE-M3 verified live this run; the sovereign serving posture stands on the shelf note and catalog rows.
Hardware and equipment	the hardware and equipment shelf	4	The 141 GB class and footprint rule fixed the GLM 5.3 node count; the diode note fixed control zone egress; the thermal camera note fixed the 9 Hz legality line; the on device shelf fixed the offered crane tier edge core.
Rules and regulations	the rules and regulations shelf	3	The personal data protection law's Article 29 sets transfer conditions for group port data flows; the national cybersecurity controls set the auditable control burden; the export control country groups place Saudi Arabia in D:4, so the 141 GB GPU class needs a license determination before order.
Approach	the approach shelf	3	Ontology first set the phase one shape (object model proven on one week of production joins before any agent runs); the sovereign posture set the dependency map and rehearsed export; the port facility security path made the security officer's approval a named gate.
History	the history shelf	1	A Saudi port where the digital layer was founding design: the port community system interface and its governance existed before any terminal AI, and operator and port authority are different companies with different duties.
Domain fusion	no corpus shelf	2	System of context's projection over systems of record fused with the brief's doctrine of sitting beside the terminal operating system: eight systems bound in place, agents draft, managers decide, the control system boundary stays one way read.

11.3 Patterns Adopted and Set Aside

System of context was adopted as the primary pattern, because the requirement is at its root an ontology with agents that draft, managers who decide, memory instead of training, and a command center. Its three layers sit in a fixed order: systems of record below, never replaced, never modified, never loaded;

the operator-owned object model in the middle as a projection over those systems, never a shadow copy, with crossing mappings that answer questions no single system can; and applications and agents above, governed by one small grammar in which nouns become objects, facts become properties, relationships become typed links and actions change them. Three patterns were set aside with reasons. Object tracking and trajectory forecasting were set aside because moves already carry equipment identity from the terminal operating system and the equipment control systems. Edge orchestration was set aside because phase one runs on the operator's central hardware and only the offered crane tier would add gateways. Labelling and retraining were set aside because the improvement path the requirement names is memory of confirmed outcomes, not weight updates.

11.4 Where the Reasoning Lands

The reasoning lands on four equipment classes, each carried from the shelf into the design: the 141 GB HBM GPU class that hosts the frontier model on one node of eight GPUs; the frontier model GPU footprint sizing rule for vLLM and SGLang serving, under which 753 billion parameters at FP8 are 753 GB of weights on 1,128 GB of memory, leaving 375 GB beside the weights against the 151 GB the 1.2 planning factor reserves; the hardware data diode carrying one-way transfer out of the control zone; and the fixed thermal camera class held at 9 Hz or less to stay under the export control line that would otherwise require a license. Every one of these choices, and every figure in this paper, traces to recorded reading: nothing shown was inferred, and nothing rests on a source the register does not hold.

SOURCES

Source Register

Service. 2025. Ports & Marine Facilities Safety Code April 2025. https://assets.publishing.service.gov.uk/media/67fd175bed87b81608546666/port-marine-safety-code.pdf

IMO. 2008. RESOLUTION MSC.255(84) (adopted on 16 May 2008) ADOPTION OF THE CODE OF THE INTERNATIONAL STANDARDS AND RECOMMENDED PRACTICES FOR A SAFETY INVESTIGATION INTO A. https://wwwcdn.imo.org/localresources/en/KnowledgeCentre/IndexofIMOResolutions/MSCResolutions/MSC.255(84).pdf

TAIC. n.d. Marine Inquiry 09-202: Passenger vessel Oceanic Discoverer Fatal injury, Port of Napier 19 February 2009. https://taic.org.nz/sites/default/files/inquiry/documents/09-202%20Final%20report.pdf

NIH. 2011. Underreporting of maritime accidents to vessel accident databases. https://pubmed.ncbi.nlm.nih.gov/21819835/

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.