ÆNIX

Sovereign AI Decision Guide

Seven decisions for designing AI infrastructure that meets sovereignty, auditability, and regulatory constraints for regulated workloads.

For CTOs, Heads of AI/ML & data-sovereignty leads · aenix.io

This guide walks through seven decisions, in order, for architecting AI infrastructure for regulated workloads. It pairs a decision tree with per-decision deep dives, four reference architectures, and GPU sizing tables for common workloads. Numbers are rough operating estimates valid as of the version date; validate against actuals before committing capacity.

Master decision tree

Walk through seven decisions in order. Each decision narrows the architecture; earlier choices constrain later ones. Most enterprise projects discover several triggers at once — pick the most binding one and layer the rest on top.

1. **Trigger** — What is your trigger for sovereign AI? Regulated data / Inference economics / Auditability mandate / Air-gap requirement.
2. **Regulator** — Which regulator(s) apply? DORA (financial) / NIS2 (essential entities) / Sectoral / National sovereign mandate.
3. **Model class** — Open-weight (Llama, Mistral, Qwen, DeepSeek, Phi, Gemma) or proprietary (closed-weight, vendor-licensed)? Sovereignty and auditability constraints usually narrow the choice to open-weight.
4. **Hardware** — Which GPU class? H100/H200 / A100 / L40S / Blackwell (B100/B200) / CPU-only or smaller accelerators.
5. **Multi-tenancy** — Single-tenant / Multi-tenant via Tenant CRD (shared cluster, isolated logically) / Cluster-per-tenant (strict isolation).
6. **Sovereignty** — Standard (in-region baseline) / Strict (customer-controlled keys, supplier transparency, audit-isolated) / Air-gapped (offline, out-of-band updates only).
7. **Operations** — Customer-operated / Vendor-operated (Aenix managed) / Hybrid (vendor control plane + customer data plane, or vice versa).

The seven decisions

Decision 1 — Trigger

What is your primary trigger for sovereign AI? More than one may apply; the primary trigger drives the architecture.

Option	Trade-offs	When to pick	Interaction with other decisions
Regulated data	Cannot leave jurisdiction or vendor environment. Drives encryption, key custody, auditability.	Regulated industry (finance, public sector, healthcare). Workload uses customer-personal or business-sensitive data.	Pushes Decision 6 (sovereignty) toward "Strict" or "Air-gapped".
Inference economics	At-scale inference becomes cheaper on owned infrastructure than per-token API.	High-volume inference (millions of tokens/day or steady production load). RAG-heavy applications.	Decision 4 (hardware) optimised for inference (L40S, H100) over training.

Option	Trade-offs	When to pick	Interaction with other decisions
Auditability mandate	Need full reproducibility — model + data + prompt + response logged immutably.	Regulated decisioning (loan approval, insurance underwriting, public-services workflows).	Decision 7 (operations) pushed toward customer-operated or strict hybrid.
Air-gap requirement	Workload must operate offline — no provider call-home, no SaaS dependencies.	Defence, sovereign cloud, isolated industrial environments.	Decision 6 forced to "Air-gapped". Decision 3 forced to open-weight.

> Most enterprise AI projects discover two or three triggers simultaneously. Pick the most binding one — the one that, alone, would force sovereign deployment. That drives the architecture; the others are layered on top.

Decision 2 — Regulator

Which regulator or regulatory framework applies to your AI workload?

Option	Trade-offs	When to pick	Interaction with other decisions
DORA (Regulation (EU) 2022/2554)	Operational resilience, ICT third-party risk, supplier exit. Architecture-heavy expectations.	Regulated financial entity or ICT third-party serving one.	Pushes Decision 7 toward customer-operated or strict-contract vendor-operated. Pushes Decision 5 multi-tenancy to provable isolation.
NIS2 (Directive (EU) 2022/2555)	Risk management, incident reporting, supply chain. Broad scope (energy, transport, telco, healthcare, public admin, etc.).	Essential or important entity per Member State transposition.	Influences Decision 6 sovereignty and Decision 7 operations significantly.
Sectoral (e.g., national healthcare data laws, GDPR special categories)	Domain-specific data classification and processing constraints.	Healthcare, legal, certain public services.	Often forces Decision 6 to "Strict"; Decision 4 hardware unaffected.
National sovereign mandate	Country-specific (e.g., GAIA-X membership, sovereign cloud designation). Often most restrictive.	Public sector, defence, critical national infrastructure.	Decision 6 forced to "Strict" or "Air-gapped"; Decision 7 customer-operated typical.

> Multiple regulators commonly stack. A bank in scope of DORA is also typically in scope of GDPR, sectoral national supervisory expectations, and — if cross-EU — NIS2. Architecture must satisfy the most restrictive, not the average.

Decision 3 — Model class

Open-weight or proprietary?

Option	Trade-offs	When to pick	Interaction with other decisions
Open-weight (Llama, Mistral, Qwen, DeepSeek, Phi, Gemma, etc.)	Customer controls weights; can fine-tune; fully reproducible; auditable. License terms vary (Llama license vs Apache 2.0 vs others — verify per model).	Sovereignty or auditability is binding. Customer wants fine-tuning ownership. Cost predictability.	Decision 4 hardware sized for chosen model; Decision 6 sovereignty fully achievable.
Proprietary (closed-weight, vendor-licensed, e.g., GPT-4 family, Claude, Gemini)	Higher peak quality on certain tasks (today). Vendor controls weights and updates. Limited fine-tuning. Auditability constrained.	Quality requirements not yet met by open-weight. Workload not in regulator scope.	Decision 6 sovereignty severely constrained; Decision 7 operations rarely customer-controlled.

> Open-weight quality has narrowed the gap to proprietary substantially in 2024–2026. For most enterprise workloads (RAG, code assistance, structured generation, classification, summarisation), open-weight models of 70B and above produce production-grade results. Proprietary still leads on certain frontier tasks; verify per use case.

Decision 4 — Hardware

Which GPU class — sized for which workloads?

Option	Best for	Approximate sizing	Trade-offs
NVIDIA H100 / H200 (80GB / 141GB HBM)	Flagship inference, large-model fine-tuning, high-throughput production.	7B–70B inference comfortably; 70B–405B with multi-GPU sharding.	Highest cost, longest lead time. Justified for production at scale.
NVIDIA A100 (40GB / 80GB)	General-purpose; cost-effective for 7B–70B inference and small fine-tuning.	7B–13B comfortably single-GPU; 70B with multi-GPU.	Older generation; supply better than H100. Good price/performance for inference.
NVIDIA L40S (48GB)	Cost-effective inference, especially for 7B–30B production.	7B–13B comfortably; 70B requires careful sharding.	No NVLink — not suitable for large-model training. Excellent inference $/token.
NVIDIA Blackwell (B100/B200)	Largest training and inference workloads (2026+).	405B+ inference with appropriate cluster size.	Newest generation; supply constrained.
CPU-only / smaller accelerators (e.g., Intel Gaudi, AMD MI series)	Small models (≤7B), RAG retrieval, embedding generation, batch inference.	Llama 7B Q4 on CPU is viable; embeddings on CPU work well.	Lower throughput. Useful for tail workloads and sovereignty edge cases (no NVIDIA).

> Mixed-hardware fleets are common. L40S for steady inference plus H100 for fine-tuning bursts is a frequent pattern. Cozystack supports heterogeneous GPU pools per tenant.

Decision 5 — Multi-tenancy

Do multiple workloads, customers, or business units share infrastructure — and how strictly are they isolated?

Option	Trade-offs	When to pick	Interaction with other decisions
Single-tenant	Simplest. One workload class per cluster. Highest cost per workload.	Single critical workload; no cross-customer sharing required; simplest path.	Decision 6 sovereignty straightforward.
Multi-tenant via Tenant CRD (Cozystack pattern)	Shared infrastructure plus logical isolation: namespace, NetworkPolicy, identity, and storage class per tenant. Cost-efficient.	Multiple workloads, business units, or customers needing efficient resource pooling with provable isolation.	Most flexible across other decisions. Recommended for most cases.
Cluster-per-tenant	Strict isolation at cluster level. Highest operational overhead. Highest cost.	Hostile multi-tenancy; sovereign customer requiring own cluster; regulatory expectation of physical/virtual separation.	Operationally expensive — automate cluster lifecycle (Cluster API / Cozystack provisioning).

> Tenant CRD is the most scalable pattern when isolation can be logical rather than physical. It supports identity, network, storage, and observability isolation per tenant on shared infrastructure. Verify the regulator's expectation before committing to "cluster-per-tenant" — it is rarely literally required.

Decision 6 — Sovereignty

How strictly does the deployment need to constrain who can see, key-control, audit, and modify the workload?

Option	Trade-offs	When to pick	Interaction with other decisions
Standard	In-region (e.g., EU-only) deployment. Vendor/provider can manage; customer-controlled where reasonable.	Workload not under strict sovereignty mandate; cost-sensitive.	Compatible with most other choices.
Strict	Customer-controlled encryption keys (BYOK / HYOK). Supplier transparency to second hop. Audit-isolated environment. Provider personnel access logged and time-limited.	Regulated workloads (DORA, NIS2). Sovereign cloud customers. National-mandate workloads.	Decision 7 typically customer-operated or hybrid with strict contract.
Air-gapped	Offline operation. No provider call-home. Updates via signed bundles delivered out-of-band.	Defence, isolated industrial environments, certain national-security workloads.	Forces Decision 3 to open-weight, Decision 7 to customer-operated. Operational overhead high.

> "Customer-controlled keys" must extend to all data-bearing layers — primary store, replicas, backups, observability data, model weights at rest, and training-data caches. A common gap is encrypted production data with vendor-controlled keys protecting the backups.

Decision 7 — Operations

Who runs the platform day-to-day?

Option	Trade-offs	When to pick	Interaction with other decisions
Customer-operated	Customer controls everything. Highest sovereignty. Highest in-house skill requirement.	Strong platform engineering team in place; air-gapped or strict-sovereignty requirement; long-term self-sufficiency goal.	Compatible with all other choices. Slowest to first production.
Vendor-operated (Aenix managed)	Aenix runs the platform under SLA. Customer focuses on workloads. Faster to production.	No internal platform team for AI infrastructure; want production capacity in 60–90 days; sovereignty achievable via strict contractual and technical controls.	Constrains Decision 6 sovereignty options unless contract/architecture guarantees customer-key and audit-export at every layer.
Hybrid	Most common in regulated industries: vendor-operated control plane plus customer-operated data plane (or vice versa). Clear interface defined contractually.	Customer wants to retain control over data and workloads but accelerate operations. Common in DORA/NIS2 contexts.	Most flexible for regulated workloads. Requires up-front clear interface definition.

> Hybrid is often the right answer for regulated workloads. The customer keeps control over data, identity, encryption keys, and workload definitions (auditable, exportable). The vendor operates the platform substrate under contractual scope. Both sides contribute to incident handling per DORA Article 23 and NIS2 Article 23.

Four reference architectures

Four common patterns, from smallest sovereign footprint to maximal-sovereignty air-gapped deployment. Match your seven decisions to the closest pattern.

Architecture 1 — Single-tenant inference cluster

- One Kubernetes cluster, one customer, one workload class (e.g., a RAG-backed chatbot).
- Components: 4–8 inference nodes (L40S or H100 GPU), 2 control-plane nodes, shared storage (LINSTOR), ingress (Cilium), observability (VictoriaMetrics + VictoriaLogs).
- Single tenant boundary at cluster level.

Footprint: Smallest viable footprint for sovereign inference. Roughly 4–8 GPU nodes. One to two platform engineers, part-time.

Architecture 2 — Multi-tenant inference fleet (Tenant CRD pattern)

- One Cozystack cluster, multiple Tenant CRDs (each = isolated namespace, NetworkPolicy, storage class, identity, and quota).
- 8–32 GPU nodes shared across tenants by the scheduler.
- Per-tenant observability views; cross-tenant no-sharing enforced by NetworkPolicy.

Footprint: Scalable multi-customer inference. Cost-efficient pooling. Logical isolation provable per tenant.

Architecture 3 — Inference + fine-tuning + RAG

- Heterogeneous cluster: H100/H200 nodes for fine-tuning; L40S for steady inference; CPU nodes for RAG retrieval and embedding.
- Vector DB (e.g., pgvector via PostgreSQL operator, or Qdrant) for RAG.
- Object storage (S3-compatible, MinIO or similar) for training data and model checkpoints.

Footprint: Full-stack pattern for organisations doing both fine-tuning and steady inference. Storage plus vector DB plus heterogeneous GPU.

Architecture 4 — Air-gapped sovereign deployment

- Isolated network. No internet egress. Updates delivered via signed bundle imported on an air-gap bridge.
- Open-weight model (Llama / Qwen / Mistral). Self-contained registry (Harbor or similar). Self-hosted observability.
- Customer-controlled keys (HSM-backed). All audit logs exported to a customer-controlled SIEM.

Footprint: For air-gapped sovereign deployments. Operational overhead high; sovereignty maximal.

Sizing reference

> The numbers below are rough operating estimates as of the version date. Validate against actuals before committing capacity. Refresh every quarter as the model and hardware landscape evolves.

Per-model sizing (open-weight, FP16 baseline; lower with quantisation)

Model family	Size	Min GPU memory (single-card)	Inference TPS reference	Notes
Llama 3 family	7B	16 GB (FP16) / 6 GB (Q4)	~80–120 TPS on L40S	Single-card viable on most modern GPUs
Llama 3 family	70B	140 GB (FP16) / 40 GB (Q4)	~25–40 TPS on H100 (FP16)	Multi-GPU sharding for FP16; Q4 single-H100 viable
Llama 3 family	405B	800 GB (FP16)	~15–20 TPS on 8×H200	Production scale; multi-node typical
Mistral family	7B / Mixtral 8×7B	16 / 90 GB	similar to Llama equivalents	MoE: lower active params than total

Model family	Size	Min GPU memory (single-card)	Inference TPS reference	Notes
Qwen family	7B / 14B / 72B / 110B	similar to Llama	similar	Strong multilingual; verify per language
DeepSeek family	7B / 67B / V3 (685B)	similar to Llama; V3 needs 8×H100 cluster minimum	varies	Newer; quality/cost competitive
Phi family (Microsoft)	3.8B / 14B	8 / 28 GB	~150+ TPS on L40S	Strong small-model option for CPU-light scenarios
Gemma family (Google)	2B / 9B / 27B	4 / 20 / 56 GB	similar to Llama 7B / 70B	Small-model performance, permissive license

Hardware combinations for typical workloads

Workload profile	Recommended hardware	Why
Steady inference, 7B–13B, single-customer	4× L40S nodes	Cost-effective, single-card serving, easy to scale linearly
Steady inference, 70B, single-customer	2× H100 (NVLink) per replica, 2–4 replicas	Sharded inference; strong throughput
Multi-tenant inference fleet, mixed model sizes	Heterogeneous: 8× L40S + 4× H100, scheduled per workload	Tenant CRD assigns per-workload class; pool utilisation high
Fine-tuning 7B–13B (LoRA / QLoRA)	1–2× H100 per fine-tuning job	Memory and bandwidth for gradient updates
Fine-tuning 70B (full / partial)	4–8× H100 / H200 cluster	Multi-GPU and multi-node depending on technique
RAG-only retrieval + embedding	CPU + small GPU (L40S / A100)	Embeddings on CPU viable; retrieval is I/O bound
Air-gapped sovereign deployment	Open-weight family + on-prem H100/L40S; full self-hosted stack	Vendor independence; sovereignty maximal

> Cozystack provides the platform layer (Tenant CRD multi-tenancy, GPU scheduling, observability, identity, storage). Ænix Platform packages this with an enterprise SLA, GPU-fleet operations, and support for fine-tuning and inference workflow patterns at scale.

Common pitfalls

- [ ] **Conflating "sovereign" with "air-gapped".** Most regulated workloads need strict sovereignty (customer-controlled keys, supplier transparency, audit isolation) — but not full air-gap. Air-gap operational overhead is significant; pick it only when truly required.
- [ ] **Sizing for peak instead of steady-state.** Production inference often runs at 30–60% of peak. Right-size for steady-state with bursting headroom (autoscaling), not peak; otherwise GPU $/token is

uncompetitive.

- [ ] **Forgetting the data side.** Vector DBs, training-data storage, and observability data all need sovereignty, key control, and capacity planning. It is common to budget GPUs and forget storage.
- [ ] **Skipping evals.** Open-weight model selection should be driven by task-specific evaluations, not headline benchmarks. Build a small evaluation suite for your workloads before committing to a model family.
- [ ] **Single-vendor GPU dependency.** NVIDIA dominates in 2026, but second-source planning matters for sovereignty and supply continuity. At minimum, verify open-weight models run on alternative accelerators (e.g., AMD MI300, Intel Gaudi).
- [ ] **Underestimating fine-tuning operations.** Fine-tuning is a workflow, not an event — data versioning, model registry, evaluation harness, deployment pipeline. Plan the workflow before sizing fine-tuning hardware.

Decision summary worksheet

Record your selected option for each decision, and the reason. Then match your profile to the closest reference architecture.

#	Decision	Your choice	Why?
1	Trigger		
2	Regulator		
3	Model class		
4	Hardware		
5	Multi-tenancy		
6	Sovereignty		
7	Operations		

Closest reference architecture: Architecture 1 (single-tenant inference) · Architecture 2 (multi-tenant fleet) · Architecture 3 (inference + fine-tuning + RAG) · Architecture 4 (air-gapped sovereign).

Open questions to discuss with platform, security, and compliance teams: list the assumptions each decision depends on, the regulator expectations still to be confirmed, and the data-side controls (keys, backups, observability) not yet designed.

Next steps

Option 1 — Self-architect using this guide. Walk through it with your platform and AI team. Use the worksheet to capture decisions. Bring open questions to your next planning session.

Option 2 — Sovereign AI architecture review with Aenix (1–2 weeks). We review your decisions, validate them against regulatory requirements, and produce a target architecture sketch plus a sizing

model. Fixed-fee.

Option 3 — Phase 2: AI platform build engagement. A multi-month build of the architecture: cluster, GPU fleet, tenancy, identity, observability, model serving, and fine-tuning workflow. Scope set after Phase 1.

> Schedule a discovery call at aenix.io. Aenix is the team behind Cozystack (CNCF Project) and offers Ænix Platform — our commercial productized offering based on Cozystack.