RADIANT

GUIDE

AI as Critical Infrastructure: A Practical Guide to Sovereign AI

radiant.co

Contents

Why Control Planes Matter More Than Data Centers

Most “sovereign AI” deployments today satisfy data residency requirements but fail operational sovereignty test. The control plane - the system that schedules jobs, allocates resources, and manages the cluster - often requires connectivity to vendor infrastructure outside national borders introducing a number of risks.

This creates a structural dependency: you can own the hardware, control the facility, and enforce data residency, but if the orchestration layer depends on external services, you don’t have sovereignty. You have a licensing agreement that works until geopolitical conditions change, vendor relationships terminate, or compliance requirements evolve.

The question sovereign buyers must ask isn’t “where is my data stored?” but “can this system operate when all external connectivity is severed?” Many commercial platforms answer with variations of “yes, but...” The “but” is where the continuum of sovereignty exists.

For some countries, there are acceptable measures of risk in having a control plane that is subject to foreign control. Increasingly, however, that risk feels too great.

Definitions Used in This Paper

For clarity, this paper distinguishes three layers that are often conflated:

- **Data Plane**: The compute, storage, and networking resources where workloads execute.
- **Control Plane**: The systems that make operational decisions such as scheduling, admission control, policy enforcement, identity integration, quota management, and tenancy boundaries.
- **Management Plane**: Fleet-level lifecycle services like upgrades, telemetry aggregation, licensing, vendor support tooling, and remote administration.

Many commercial platforms keep parts of the management plane - and sometimes critical elements of the control plane - outside the customer’s jurisdiction. Sovereign AI requires both planes to be independently operable domestically.

Many sovereign AI initiatives follow a similar high-level stack pattern:

Layer 1	Domestic data center facility (power, cooling, physical security)
Layer 2	AI compute hardware (GPUs, accelerators, high-performance CPUs)
Layer 3	High-performance storage and data services
Layer 4	Orchestration and scheduling (for example Kubernetes, Slurm, or equivalent)
Layer 5	Platform control and management services (identity, policy, telemetry, upgrades, licensing)

On paper, this appears sovereign because the physical infrastructure and primary workloads run in-country. In practice, sovereignty often breaks at the control and management layers.

Even where Layers 1–4 are domestically deployed, critical operational functions are frequently tied to vendor-operated external services. These dependencies are not always visible in architecture diagrams or procurement summaries, but they materially affect operational independence.

In technical due diligence, external dependencies most commonly appear in the following forms:

- License or entitlement validation calls to vendor servers
- Identity or token issuance from external identity providers
- Telemetry and metrics pipelines bound to vendor SaaS dashboards
- Remote configuration or optimization services
- Upgrade and patch orchestration controlled from vendor infrastructure
- Container or model artifact registries that are not fully mirrored locally
- Key management or metadata services hosted outside jurisdiction
- Vendor support tunnels required for administration or incident response

If these services are unreachable, platforms may not fail loudly. Instead, they may enter degraded states: scheduling restrictions, authentication failures, blocked upgrades, expiring credentials, or disabled management functions.

Because these dependencies are often optional in marketing material but mandatory in real operation, a direct question is required during evaluation:

“Can this platform continue operating production workloads if all connectivity to vendor infrastructure is severed?”

In many cases, the accurate answer is conditional — requiring configuration changes, emergency procedures, or vendor intervention.

For high-assurance sovereign deployments, the target design standard is stronger: disconnected operation should be a tested and supported mode, not an exception path.

What Sovereignty Actually Requires

A sovereign control plane must satisfy five architectural requirements:

1. Control plane domesticity

All job scheduling, resource allocation, and system orchestration executes on hardware within your jurisdiction. You don’t need to own the control plane, but you need domestic control over it - meaning it runs on infrastructure you operate, under domestic legal jurisdiction.

2. Zero external dependencies

The system operates at full capacity with zero outbound connectivity. No remote procedure calls to external infrastructure for operational decisions. Container images, model weights, and system dependencies are cached locally. External services (updates, support, monitoring) are pull-based (you initiate), not push-based (vendor initiates).

3. Operational independence

Your team can operate, upgrade, and troubleshoot the system without vendor intervention. Identity, policy enforcement, and audit logging run on local databases. Skills transfer is contractual with measurable milestones, not optional.

4. Hardware agnosticism

You’re not structurally dependent on a single GPU vendor, networking vendor, or storage vendor. The architecture accommodates substitution without rearchitecting the platform.

5. Auditability and explicit dependencies

You can demonstrate to regulators exactly where every component runs, who operates it, and what data crosses what boundaries. Any external dependency fails visibly rather than silently degrading. Hybrid or burst configurations require explicit policy enablement. Telemetry and monitoring data stays in-country by default.

Many commercial platforms satisfy #1 partially, #2 rarely, #3 almost never, #4 through procurement rather than architecture, and #5 through documentation rather than design.

The primary test: Can you sever all external connectivity and continue operating production workloads without vendor intervention?

Sovereign architectures fall into tiers:

Disconnect-native	Continues operating with no configuration changes
Disconnect-tolerant	Continues operating with pre-staged artifacts and policy switches
Disconnect-degraded	Partial function only
Disconnect-failed	Operations stop

Only the first two tiers qualify as operationally sovereign.

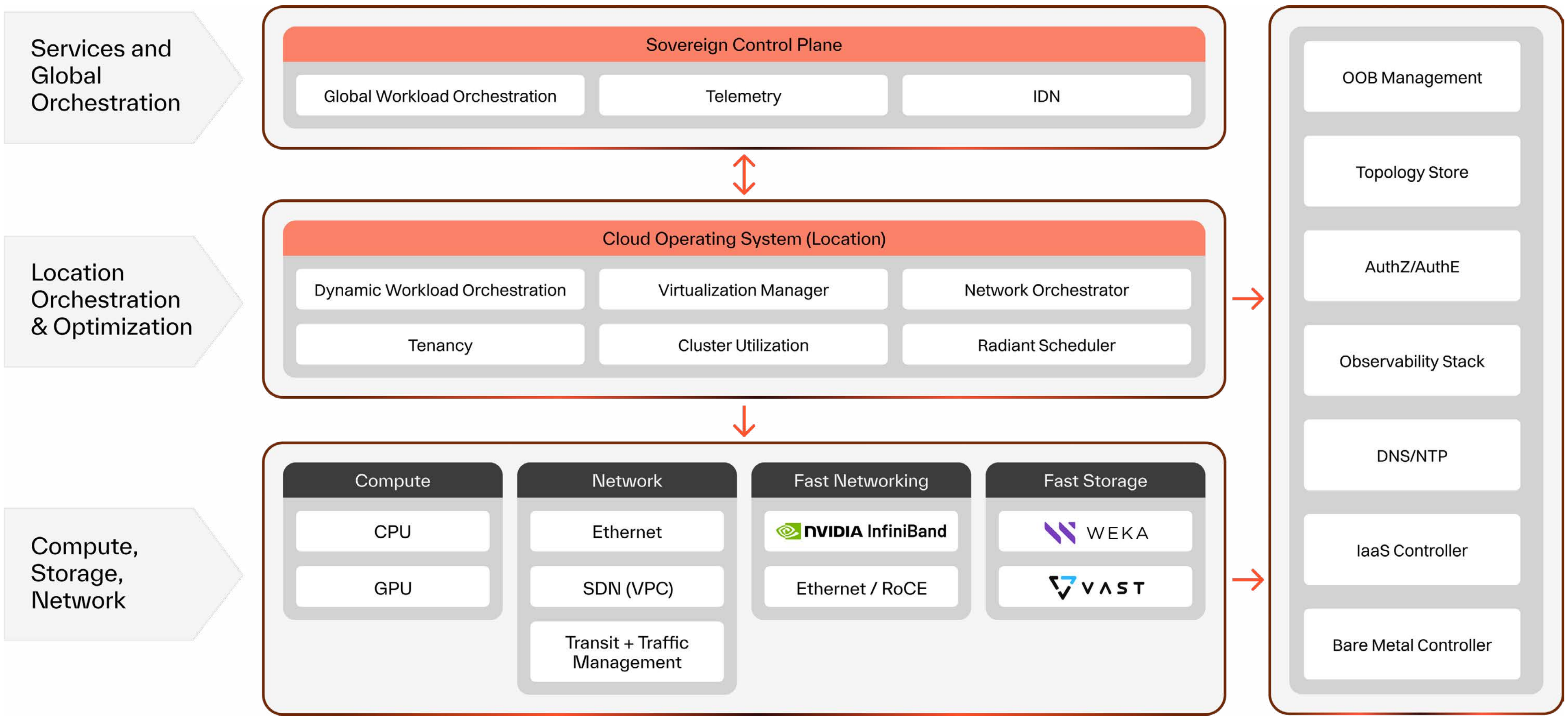
The Sovereign Blueprint: Designing for the Disconnect

If we accept that true sovereignty requires a system that functions in total isolation, the architecture must shift from a "Cloud-First" to a "Site-First" logic. Traditional cloud architectures centralize control plane functions in vendor-operated infrastructure while distributing compute resources to customer sites. Achieving sovereignty requires co-locating both control and compute within the same jurisdictional boundary. This requires fundamental re-engineering of the platform stack:

- **Site-level autonomy**: Each data center location must be designed as a self-contained operational unit capable of managing its own identity, scheduling, and telemetry without "calling home."
- **Domestic root of trust**: Instead of external SaaS-based administration, the system must recognize a local root of authority - one that answers only to domestic administrators.
- **Operational transfer**: The transition from vendor-managed to customer-managed operations must be a structured architectural decoupling where the software is designed to be operated by a third party (the nation-state) from day one.

In many cases, achieving disconnected-first operation cannot be reached through configuration alone and requires substantial architectural changes to platform control and management services.

The control plane makes sovereignty operational: SSO, fine-grained RBAC, quotas, audit logging, observability, and multi-org management - every action traceable, every resource policy-governed. Tenants (ministries, national champions, research institutions) are isolated via soft, strict, or fully private tenancy, letting sensitive organizations share a national AI fleet without compromising security, compliance, or performance.



Why Global Clouds Fail the Sovereignty Test

The "Sovereign Cloud" offerings from global hyperscalers - such as the AWS European Sovereign Cloud, Azure EU Data Boundary, or Google Cloud Sovereignty Controls - are sophisticated products. For 90% of commercial enterprises, they are excellent solutions for data residency.

However, for a nation-state treating AI as critical infrastructure, they fail on three structural levels:

1. The Control Plane Tether

Hyperscaler "sovereignty" is frequently defined by the **Data Plane** (where your data sits) but never by the **Control Plane** (where the "brain" sits). Even in their sovereign regions, the software that schedules your jobs, manages your encryption keys, and authenticates your users is a SaaS service operated by the vendor.

The Failure: If the vendor's global management network is severed, or if the vendor is compelled by foreign law to suspend your account, your domestic cluster may lose scheduling and management capability even if the hardware remains available. You have residency, but no autonomy.

2. The Operational Dependency Trap

Global cloud providers are built on proprietary, "black box" orchestration layers. They do not - and cannot - train your national workforce to operate their stack independently. Their business model relies on you staying dependent on their proprietary SRE teams and automated workflows.

The Failure: Sovereignty requires that your own citizens can operate, troubleshoot, and repair the system. You cannot achieve "Capability Transfer" with a vendor whose software is a trade secret.

3. Economic Rent-Seeking vs. Infrastructure Financing

Hyperscalers are optimized for high-margin, short-term "renting." Public cloud GPU services are typically priced at substantial gross margins relative to infrastructure cost, based on industry analyst estimates and public financial disclosures. For a 5-year national AI program, this results in "margin stacking" - where a government pays a massive premium for a service model they don't actually need for a fixed, long-term asset.

The Failure: True infrastructure (like a power plant or a bridge) is financed at a low cost of capital (4-6%). Hyperscalers force governments to pay venture-level returns on what should be a utility-priced asset.

The Litmus Test: If a foreign regulator can "turn off" your domestic AI cluster with a single API command from outside your borders, you do not have a sovereign solution. You have a high-end rental agreement.

Domestic Alternatives: Why They're Not Enough

Many countries have domestic cloud providers (OVHcloud in France, sovereign initiatives in Germany, regional providers in Middle East). These satisfy the "domestically operated" requirement but fail on technical capability.

Common gaps:

1. **GPU density**: Domestic providers often retrofit general-purpose data centers with GPUs added as an afterthought. Power density is 10-15kW/rack (adequate for CPU workloads, insufficient for GPU clusters requiring 30-60kW/rack).
2. **Networking**: Standard Ethernet fabrics (1-10Gbps) instead of RDMA-capable networks (InfiniBand or RoCE at 200-400Gbps). For distributed training, this makes multi-node jobs impractical.
3. **Software maturity**: Custom orchestration layers that don't integrate with standard ML frameworks. Data scientists end up writing infrastructure code instead of training models.
4. **No capability transfer**: Because the platform is proprietary, there's no external expertise to draw from. You're dependent on a single domestic vendor.

Domestic alternatives work for inference workloads and small-scale training. They don't work for national AI programs that need to train foundation models or run large-scale research programs.

The strategic pattern that emerges: sovereign platforms that combine global-class AI infrastructure capability with domestic operational control for national-scale training and research. Domestic providers serve complementary roles in edge inference, citizen-facing applications, and politically sensitive deployments where any international partnership creates unacceptable risk.

The Importance of Building on Bare Metal

Sovereign AI benefits from performance predictability and minimized abstraction layers. The most effective approach is to implement a bare-metal foundation, which reduces overhead and simplifies dependency auditing, though sovereignty can also be achieved with virtualized stacks if they are fully domestically operable and dependency-free.
By eliminating the hypervisor, bare-metal architectures avoid virtualization overhead that independent benchmarks commonly measure in the high-single-digit to low-double-digit percentage range, depending on workload type and I/O profile. This helps workloads operate closer to the theoretical performance envelope of the underlying silicon, subject to software stack and workload characteristics.

Bare metal does not imply rigidity. Modern bare-metal platforms layer cloud-native AI services directly on physical infrastructure. Supercomputers spanning hundreds or thousands of GPUs, virtual machines, Kubernetes environments, inference endpoints, and fine-tuning pipelines are all provisioned from the same hardware pool. This allows governments to evolve AI capabilities over time without re-architecting infrastructure or locking into a single operational model.

Security and isolation are enforced at the hardware level. With a bare metal orientation it is possible to implement deep, hardware-rooted isolation across compute, storage, and networking, enabling secure resource sharing without fragile software boundaries. Sensitive workloads remain protected even as infrastructure is efficiently shared across agencies, ministries, or national organizations. Isolation becomes a property of the system itself, not an operational compromise.

At national scale, this bare-metal foundation unlocks powerful economies of scale.
As sovereign AI fleets grow from dozens to thousands of GPUs, performance gains compound rather than degrade. Utilization improves, cost per compute unit decreases, and operational complexity is reduced through uniform control and orchestration. The result is AI infrastructure that scales linearly with ambition, turning capital investment into a durable, high-performance national asset built to serve sovereign needs for decades.

A Tenancy Spectrum for Diverse Workloads

Sovereign AI infrastructure must reconcile two forces that are often treated as incompatible: absolute isolation for sensitive national workloads, and efficient utilization of scarce, high-value compute resources. This tension can be addressed through a single, unified control plane that orchestrates multiple tenancy models across one physical fleet. Rather than fragmenting infrastructure by workload type or organizational boundary, unified control planes enable governments to serve every class of user, from experimental research teams to defence and critical national services, within a coherent, policy-driven platform.

Tenancy Mode	Description	Benefits
Soft	GPU nodes are shared between tenants, with namespace-level isolation	Maximizes cluster utilization; end-customers control compute usage
Strict	Entire GPU node(s) reserved for a tenant with ring-fenced compute, storage and NVIDIA Infiniband networking	No workload sharing; predictable performance without “noisy neighbour” effect
Private	Dedicated, single-tenant deployment plus independent platform management capabilities	Ultimate level of isolation and data privacy; full control over underlying management

At the beginning of the spectrum is soft tenancy, a model optimized for efficiency and collaboration. GPU nodes are shared between tenants using platform-enforced namespace isolation, allowing multiple teams, departments, or agencies to safely operate on the same hardware. This maximizes fleet yield, minimizes idle capacity, and enables rapid access to compute for public-sector innovation, research, and internal development - without compromising governance or visibility.

For workloads requiring stronger guarantees, strict tenancy provides ring-fenced isolation across compute, storage, and networking. Entire GPU nodes are reserved for a single tenant, eliminating resource contention and “noisy neighbor” effects. This model is suited to business-critical systems, regulated environments, and sensitive government applications where predictable performance, compliance, and auditability are essential. Isolation is enforced at the infrastructure and control-plane level, not through fragile overlay mechanisms.

At the highest level of assurance, private tenancy provides a fully dedicated, single-tenant environment with independent platform management. Designed for defence, intelligence, and national-security use cases, this model fully segregates infrastructure, control, and operations, enabling air-gapped deployments, bespoke governance, and complete autonomy over data, models, and workloads.

Together, these tenancy modes form a trusted spectrum rather than a rigid choice. A single national AI fleet can support innovation, efficiency, and the most sensitive missions simultaneously - without duplicating infrastructure or surrendering control. Orchestrated through a unified control plane, tenancy becomes a matter of design, not compromise.

Storage: The Sovereignty of Data-at-Rest

In a sovereign AI architecture, storage is more than a performance requirement - it is a jurisdictional anchor and the ultimate guarantor of data persistence.

Many "cloud-adjacent" sovereign solutions fail because while the GPUs are local, the metadata services or encryption key management for the storage layer reside in a vendor's global cloud. If the connection to that global service is severed, the data becomes an encrypted brick.

A truly sovereign storage architecture requires:

- **Local Metadata and Key Management**: All file system metadata and KMS (Key Management Service) operations must execute within the domestic security boundary.
- **Direct Pathing**: To realize the value of bare-metal compute, the storage layer must support RDMA and GPUDirect protocols, bypassing the CPU to feed data directly from the disk to the GPU.
- **S3-Compatible Protocol, Domestic Implementation**: We leverage high-performance, S3-compatible object storage - not because we are tied to a specific vendor, but because it provides the scale-out flexibility AI demands without the proprietary lock-in of legacy file systems.

Sovereign platforms should not lock deployments into specific storage providers. The architecture should provide high-speed fabric and local orchestration that allows any high-performance storage - whether Weka, VAST, MinIO or domestic alternatives - to function as a fully sovereign, air-gapped asset.

Cost Structure: Infrastructure vs Rent-Seeking

The traditional AI supply chain extracts margin at every layer:

1. **Data center operator** (15-20% margin on power/space)
2. **Colocation provider** (10-15% margin on rackspace)
3. **Hardware integrator** (8-12% margin on GPU procurement)
4. **Cloud platform vendor** (40-60% margin on orchestration/control)
5. **Support services** (20-30% margin on SRE/operations)

Radiant's model collapses layers 1-4 into a single entity financed at infrastructure cost of capital (~5% vs 20%+ for venture-backed platforms). The result: fixed-price compute locked in for 5+ years. This is how infrastructure financing works. Bridges and power plants are financed at 4-6% because the cash flows are predictable and the assets are long-lived. GPUs should be financed the same way.

The savings come from eliminating margin stacking, not from cutting corners.

Why Architecture Matters: Performance and Economics

Meeting sovereignty requirements on paper means nothing if the implementation is slow or expensive. The architectural choices that enable sovereignty - bare metal deployment, hardware-agnostic design, local control planes - also determine performance and total cost of ownership.

In a sovereign deployment, where infrastructure is not shared with external customers, a virtualization penalty provides zero operational benefit while creating a massive economic drain. This performance gap compounds over the lifecycle of the asset: efficiency losses compound across the infrastructure lifecycle: higher energy consumption for equivalent computational output, extended training times that delay model deployment, and increased capital expenditure to achieve target performance levels. For a state-scale deployment, this technical inefficiency translates directly into tens of millions of dollars in wasted capital and operational expenditure. Slower inference doesn't just cost more - it results in a degraded, high-latency user experience for citizen-facing applications.

Our benchmark testing indicates that Radiant's bare-metal driven architecture reduces inference latency considerably by avoiding the virtualization overhead. For example, **Time-to-first-token (TTFT) for the gpt-oss-20b model was measured at 0.165 seconds which is comparable to the leading provider on Artificial Analysis' leaderboard for latency** (Groq - 0.16 seconds) and better than other inference providers from across the industry.

Bare metal matters for inference because latency predictability determines user experience. Shared infrastructure introduces jitter. For sovereign deployments serving citizen-facing applications, P99 latency is often a contractual SLA.

The Path Forward: AI as Infrastructure, Not as Service

If you're evaluating sovereign AI options:

Question 1: Can your system operate with zero outbound connectivity? Not "in degraded mode" - at full capacity.

Question 2: Show me your control plane architecture. Where does it run? Who operates it? What happens if I sever all connectivity to your infrastructure?

Question 3: How long until my team can operate this system independently? Show me the training plan with measurable milestones.

Question 4: What's your cost of capital? If it's >10%, you're financing this like a tech company, not like infrastructure. I'll be paying that premium.

Question 5: Can I replace any component (GPU vendor, storage vendor, networking vendor) without rearchitecting the entire platform?

If the vendor hesitates on Questions 1-3, they don't have sovereignty. If they can't answer Questions 4-5, they're economically or technically locked-in.

Common Deployment Patterns

Sovereign AI deployments typically follow one of four architectural patterns, each trading off sovereignty depth, speed to production, and operational complexity:

Pattern 1: Managed Cloud with Domestic Presence

- Infrastructure: Vendor-owned data centers in-country
- Operations: Vendor SREs
- Control plane: Vendor-managed, domestically deployed
- Trade-off: Fastest time-to-production; least operational sovereignty
- Appropriate for: Hybrid strategies, commercial workloads

Pattern 2: Customer Infrastructure, Vendor-Managed Operations

- Infrastructure: Customer-owned or leased facility
- Operations: Vendor SREs with contractual capability transfer
- Control plane: Deployed in-country, vendor-managed initially
- Trade-off: Data sovereignty immediately; operational sovereignty after transfer period
- Appropriate for: Governments building domestic capability

Pattern 3: Customer Infrastructure, Customer-Managed Operations

- Infrastructure: Customer-owned
- Operations: Customer SREs (vendor provides training + support contracts)
- Control plane: Customer-operated from deployment
- Trade-off: Maximum sovereignty; requires existing technical capability
- Appropriate for: Nations with HPC/cloud operations expertise

Pattern 4: Hybrid Federation

- Infrastructure: Mix of domestic (customer-owned) + vendor cloud
- Operations: Coordinated (customer operates domestic, vendor operates cloud)
- Control plane: Federated (unified view, regional autonomy)
- Trade-off: Burst capacity available; complexity in policy enforcement
- Appropriate for: Staged buildouts, workloads with mixed sensitivity

Most sovereign programs start with Pattern 2 and transition to Pattern 3 after capability transfer completes.

Evaluation Checklist: What to Demand from Vendors

Section 1: Operational Sovereignty (Non-Negotiable)

- □ Control plane runs on hardware we control, in facilities we operate
- □ System operates at full capacity with zero outbound connectivity
- □ No mandatory external SaaS dependencies for scheduling, orchestration, or identity
- □ All telemetry and logging stays in-country by default (external access is pull-based, not push-based)
- □ Complete audit trail of all operator actions (who did what, when, from where)

If any box is unchecked, you have a data residency solution, not a sovereign solution.

Section 2: Security Architecture

- □ Multiple tenancy modes (soft, strict, private) defined architecturally
- □ Air-gapped deployment option available without feature degradation
- □ Hardware-enforced isolation (not just namespace/VLAN isolation)
- □ Encryption at rest and in transit with customer-managed keys
- □ No vendor backdoors or maintenance access without explicit authorization

Section 3: Supply Chain Resilience

- □ GPU vendor can be substituted (not locked to NVIDIA)
- □ Networking vendor can be substituted (not locked to InfiniBand)
- □ Storage vendor can be substituted (not locked to specific parallel FS)
- □ Architecture minimizes single points of failure
- □ Exposure to export controls is documented and mitigated

Section 4: Capability Transfer

- □ Formal training program with measurable outcomes (not “documentation”)
- □ Defined handoff milestones (joint ops → primary ops → independent ops)
- □ National operators certified before vendor exit
- □ Runbooks written collaboratively (not delivered as read-only documents)

Section 5: Economic Certainty

- □ Fixed-price commitment available (not “subject to GPU market pricing”)
- □ Cost of capital aligned with infrastructure projects (not venture returns)
- □ No hidden margin stacking (single entity controls full stack)
- □ TCO model includes power, cooling, operations - not just hardware

Capability Transfer: The Concrete Plan

Many commercial platforms promise “knowledge transfer” and deliver PowerPoint decks. Here’s what actual capability transfer looks like:

Phase 1: Shadowing (Months 1-3)

- Vendor SREs on-site full-time
- National operators shadow all procedures
- Every incident, upgrade, and configuration change is documented collaboratively
- Outcome: National team can perform routine operations under supervision

Phase 2: Primary Operations (Months 4-9)

- National operators take primary on-call rotation
- Vendor SREs available for escalation only
- National team performs system upgrades with vendor review
- Monthly competency assessments with documented pass/fail criteria
- Outcome: National team operates independently for 90% of scenarios

Phase 3: Full Autonomy (Months 10-12)

- National team operates completely independently
- Vendor provides quarterly reviews and upgrade planning
- SRE support available under SLA (response times: P0=1hr, P1=4hr, P2=24hr)
- Annual security audits and architecture reviews
- Outcome: National team certified, vendor exits day-to-day operations

Certification criteria (must pass before Phase 3):

- Perform hardware replacement (GPU swap, NIC replacement) without assistance
- Diagnose and resolve network performance issues (InfiniBand fabric troubleshooting)
- Execute system upgrade (control plane, orchestration layer, storage firmware)
- Respond to security incident (unauthorized access attempt, data exfiltration detection)
- Provision new tenant with strict isolation requirements

These aren’t aspirational. They’re contractual milestones.

Conclusion: Infrastructure vs Theater

The sovereign AI market is full of vendors who will tell you “yes” to every requirement. The technical evaluation separates real sovereignty from compliance theater.

Sovereign AI isn’t about nationalism or paranoia. It’s about infrastructure you can operate when geopolitical conditions change, vendor relationships terminate, or compliance requirements evolve.

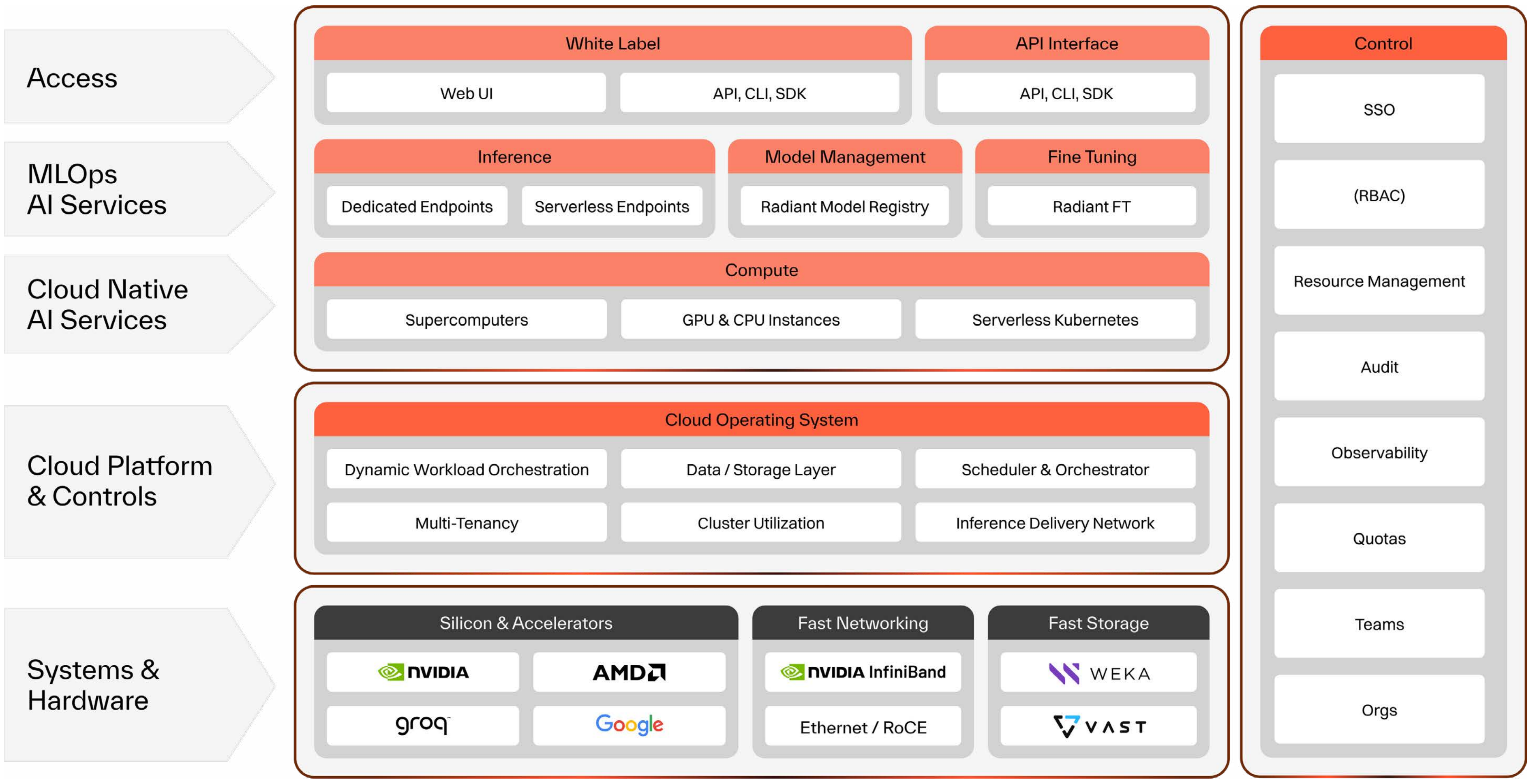
If you’re evaluating sovereign AI, demand specificity. Demand proof. Demand an architecture you can audit. And demand an answer to the simplest question: “If I cut all connectivity to your infrastructure, does this system still work?”

About Radiant’s Sovereign Solutions

The Radiant architecture is the physical realization of the Sovereign Blueprint. By centering the entire stack around a fully local control plane, Radiant ensures uninterrupted government authority under all conditions.

Radiant’s differentiation is architectural, not rhetorical. The Radiant Sovereign Control Plane is deployed in-country to perform all critical functions without outbound coordination:

- **Autonomous Scheduling**: Manages GPU clusters and schedules jobs without external dependency or remote signaling.
- **Local Persistence**: Enforces RBAC, quotas, and audit logging entirely on local databases within the domestic jurisdiction.
- **Domestic Caching**: Maintains container images, model weights, and system dependencies on-site to ensure immediate availability during disconnected scenarios.
- **Isolated Tenancy**: Manages multi-tenant isolation through namespace-level and node-level policies executed locally.



Access

MLOps AI Services

Cloud Native AI Services

Cloud Platform & Controls

Systems & Hardware

White Label: Web UI; API, CLI, SDK

API Interface: API, CLI, SDK

Inference: Dedicated Endpoints; Serverless Endpoints

Model Management: Radiant Model Registry

Fine Tuning: Radiant FT

Compute: Supercomputers; GPU & CPU Instances; Serverless Kubernetes

Cloud Operating System: Dynamic Workload Orchestration; Data / Storage Layer; Scheduler & Orchestrator; Multi-Tenancy; Cluster Utilization; Inference Delivery Network

Silicon & Accelerators: NVIDIA; AMD; groq; Google

Fast Networking: NVIDIA InfiniBand; Ethernet / RoCE

Fast Storage: WEKA; VAST

Control: SSO; (RBAC); Resource Management; Audit; Observability; Quotas; Teams; Orgs

At its foundation is sovereign AI infrastructure: in-country GPU compute integrated with high-performance networking and storage. Radiant supports hardware optionality—NVIDIA, AMD, and emerging accelerators; InfiniBand or RoCE; and leading platforms such as Weka and VAST. This multi-vendor architecture reduces supplier dependency, mitigates export and supply risks, and preserves flexibility as hardware evolves.

Radiant's Cloud Operating System, purpose-built for sovereign AI, delivers dynamic orchestration, topology-aware scheduling, fleet-wide utilization management, and deterministic inference at national scale. Compute is exposed as flexible compute primitives. Administrators can provision supercomputers for large-scale training, GPU and CPU instances for mixed workloads, or Serverless Kubernetes for elastic inference and applications.

Radiant also provides MLOps and lifecycle services essential to model sovereignty. Dedicated and serverless inference endpoints enable secure production deployment. Radiant Model Registry ensures ownership of weights, versions, and provenance, while Fine-Tuning supports national and sector-specific models trained on sovereign data across language, energy, healthcare, defence, and public services.

Developer access is delivered through APIs, Web UI, CLI, and SDKs, along with white-label interfaces, enabling governments to operate a highly customizable national AI platform.

Deployment Models

Radiant offers four paths to production, allowing nations to calibrate their balance of speed and autonomy:

- **Model 1**: Radiant Public Cloud – Fastest path to production; infrastructure and operations managed by Radiant in-country.
- **Model 2**: Customer Infrastructure, Radiant Managed – Data sovereignty in customer facilities with Radiant-managed operations and a defined 24-month handoff plan.
- **Model 3**: Customer Infrastructure, Customer Managed – Maximum sovereignty. Customer-operated from day one with Radiant providing intensive training and support.
- **Model 4**: Hybrid Federation – Combines domestic customer-owned nodes with Radiant cloud capacity for burst workloads, governed by unified, regional policy.

Together, these layers form a self-contained system where control, resilience, and security reinforce one another. The result is AI infrastructure that functions as a national utility: reliable under stress, secure by construction, and governed within domestic boundaries.

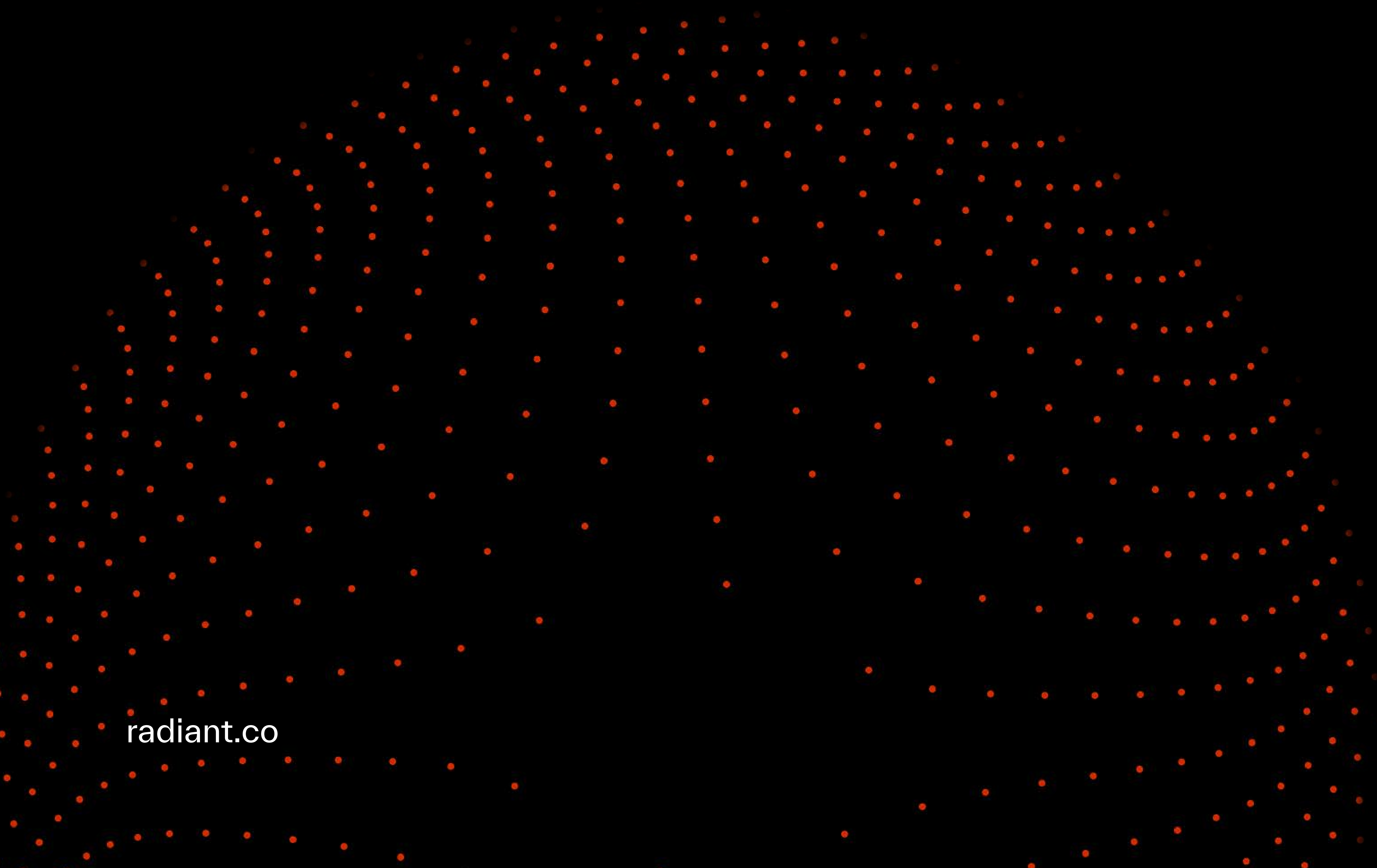
Learn more at www.radiant.co

RADIANT

About Radiant

Radiant is building the utility model for compute - ubiquitous, always on and offering superior economics. Through vertical integration of capital, powered land, compute, and software, we deliver sovereign-grade control for nations, turnkey certainty for enterprises, and limitless capacity for builders. This model will democratize AI, equipping every individual with the ability to unlock their creative potential and tackle world-changing problems.

LEARN MORE AT RADIANT.CO



radiant.co