Whitepaper #01: Sovereign AI Infrastructure

Subtitle: *Architecting Intelligent Systems for Disconnected Environments* **Author:** Dustin J. Ober, PMP

1. Executive Summary

The Bottom Line Up Front (BLUF): While the commercial AI revolution is driven by cloud-hosted APIs (e.g., OpenAI, Anthropic), organizations in defense, healthcare, and critical infrastructure face a stark reality: they cannot send their data to the cloud. For these sectors, the future of AI is not "Cloud First"—it is **"Sovereign First."**

The Core Problem: Strict regulatory frameworks (ITAR, HIPAA, CUI) and Zero Trust security mandates often prohibit the use of external inference endpoints. Data leakage risks, however small, are unacceptable when the data involves national security or sensitive intellectual property. Furthermore, mission-critical systems often operate in "air-gapped" or disconnected environments where internet connectivity is physically impossible.

The Solution: This brief outlines the reference architecture for a **Sovereign Inference Unit (SIU)**—a fully offline, self-contained AI pipeline. By repatriating workloads to local infrastructure, organizations achieve 100% data sovereignty, eliminate external dependencies, and ensure continuity of operations even during network blackouts. This document serves as a technical guide for architects sizing the hardware (VRAM), network topology, and software stack required to deploy Large Language Models (LLMs) behind the firewall.

2. The Operational Challenge: Why "Local" is Hard

Deploying AI in a connected enterprise is relatively simple: provision an API key, install a Python library, and start sending JSON requests. Deploying AI in a disconnected environment is a fundamental systems engineering challenge. It forces architects to solve problems that cloud providers usually abstract away.

2.1 The Cloud Disconnect

Modern AI development assumes ubiquitous connectivity. Toolchains rely on "lazy loading" resources from the internet:

- `pip install` fetches libraries from PyPI.
- `docker pull` fetches containers from Docker Hub.
- `AutoTokenizer.from_pretrained()` fetches weights from Hugging Face.

In a sovereign environment, all of these commands fail immediately. The "Operational Challenge" is not just running the model; it is rebuilding the entire supply chain of dependencies that allows the model to exist.

2.2 The "No-Egress" Mandate

The defining constraint of a secure facility is the **No-Egress Policy**. Data cannot leave the network. This creates two specific friction points:

- **Ingress Difficulty:** Getting open-source innovation *into* the environment requires a rigorous "sneaker-net" or "diode" transfer process, involving scanning, hashing, and manual approval.
- **No Telemetry:** You cannot use monitoring tools like Datadog or LangSmith. You must build your own observability stack (e.g., Prometheus/Grafana) to know if your model is hallucinating or failing.

2.3 The Throughput Reality Check

Stakeholders often expect "ChatGPT-level speed." This expectation must be managed.

- **Cloud Reality:** A massive cluster of H100 GPUs serving thousands of users with auto-scaling.
- **Local Reality:** A single workstation or small rack serving a specific team.

While local hardware cannot match the raw token-per-second generation of a massive cloud cluster, it offers a superior trade-off: **Consistency.** Your local latency is predictable, your queue is yours alone, and your service never goes down because a cloud provider has an outage.

3. Hardware Sizing: "Sizing the Iron"

In a disconnected environment, you cannot simply "scale up" an instance class when a model crashes. Hardware is fixed, procurement cycles are long, and the cost of under-provisioning is a failed deployment. Therefore, calculating Video Random Access Memory (VRAM) requirements is the single most critical step in architecting a sovereign inference unit.

3.1 The VRAM Equation

The total VRAM required (M_total) is the sum of three distinct components: the static model weights, the dynamic Key-Value (KV) cache (the "context window"), and the temporary activation buffers.

VRAM Equation

Component	Description
P	Number of parameters (e.g., 7B, 70B)
B_p	Bytes per parameter (FP16 = 2, Q4 = 0.5)
C_ctx	Context length (e.g., 4096, 8192)
L	Number of layers
H	Hidden dimension size
B_kv	Bytes per KV cache entry
O_sys	System overhead (~500MB-2GB)

3.2 Quick Reference Sizing

Tier	Example Model	Quantization	VRAM Required	Hardware Example
Small (Edge)	Llama-3-8B	Q4_K_M	~6 GB	RTX 3060, Jetson Orin
Medium (Dev)	Mixtral 8x7B	Q4_K_M	~24 GB	RTX 4090, A5000
Large (Pro)	Llama-3-70B	Q4_K_M	~48 GB	A6000, Dual 3090s
Enterprise	Llama-3-405B	Q4_K_M	~200 GB	Multi-GPU (8x A100)

Key Takeaway: For sovereign environments, **VRAM is the bottleneck, not compute.** A slower token generation speed is acceptable for a chatbot; an Out-Of-Memory (OOM) crash is not.

---

4. Network Architecture: The "Zero Trust" Setup

In a sovereign environment, the network is a constraint. We achieve functionality by building a "Local Internet"—a mirrored ecosystem that lives entirely within the secure boundary.

4.1 The Physical Air-Gap Topology

The Sovereign Inference Unit (SIU) resides on a dedicated subnet with physically severed uplinks.

```
┌─────────────────────────────────────────────────────────────────────┐
│                         OPEN INTERNET                               │
│  (Ingest Station: Sanitization for scanning/hashing assets)        │
└──────────────────────────────┬──────────────────────────────────────┘
                               │
                               │
                        ┌──────▼──────┐
                        │ DATA DIODE  │  (Unidirectional Transfer)
                        └──────┬──────┘
                               │
```

```
┌─────────────────────────────────▼──────────────────────────────────┐
│                     CLOSED SYSTEM (Secure Zone)                      │
│                                                                    │
│  ┌─────────────┐  ┌─────────────┐  ┌──────────────────────────┐  │
│  │ PyPI Mirror │  │ Model Store │  │ Sovereign Inference Unit│  │
│  │   (Nexus)   │  │   (MinIO)   │  │      (GPU Cluster)       │  │
│  └─────────────┘  └─────────────┘  └──────────────────────────┘  │
└────────────────────────────────────────────────────────────────────┘
```

4.2 Internal Repositories

- **PyPI Mirror:** Use **Sonatype Nexus** or **JFrog Artifactory** to host Python wheels internally. Configure pip to use your internal index:

  ```
  [global]
  index-url = https://nexus.internal.lab/repository/pypi-hosted/simple
  trusted-host = nexus.internal.lab
  ```

- **Model Registry:** Use **MinIO** or a shared NAS to host GGUF model files (`.gguf`) so developers don't store 40GB files on their laptops.

4.3 Containerization: Docker vs. Apptainer

Feature	Docker	Apptainer (Singularity)
Use Case	Open Internet builds	Closed System / HPC
Privilege	Requires `root` daemon	Rootless execution
Format	Layered images	Single `.sif` file
Security	Good	Excellent (verifiable)
Transfer	`docker save -o image.tar`	Single file transfer

Recommendation: Build with Docker on the open side, convert to Apptainer `.sif` at the boundary using:

```
apptainer build my-model.sif docker-archive://my-model.tar
```

---

5. Software Stack: The "Sovereign Inference" Layer

5.1 Operating System

Production units should run on **Ubuntu LTS** or **RHEL**.

- **Critical:** You must manually install the NVIDIA drivers using the local `.run` file and strictly pin your CUDA version (e.g., 12.1) to match your PyTorch binaries. Version mismatches cause silent failures.

5.2 The Inference Engine

Engine	Best For	Key Feature	API
Ollama	Developers, single-user chat	Easy model management	OpenAI-compatible
vLLM	Production, high-throughput RAG	PagedAttention (24x perf)	OpenAI-compatible
Llama.cpp	Edge, CPU-only inference	Minimal dependencies	Custom
TGI	Enterprise, multi-model serving	Optimized batching	HuggingFace API

5.3 Quantization (GGUF)

Quantization reduces model size by using fewer bits per parameter. We recommend **4-bit Medium Quantization (Q4_K_M)** as the optimal balance:

Quantization	Bits	Size Reduction	Quality Impact
FP16	16	Baseline	None

Quantization	Bits	Size Reduction	Quality Impact
Q8_0	8	50%	Minimal
Q4_K_M	4	**75%**	**~1% perplexity loss**
Q2_K	2	87.5%	Noticeable degradation

A Llama-3-70B model at Q4 retains ~99% of its reasoning capability but fits into 40GB of VRAM instead of 140GB.

6. Deployment Checklist

Before deploying your Sovereign Inference Unit, verify:

- [ ] **Hardware:** VRAM calculated and validated for target model
- [ ] **Drivers:** NVIDIA drivers installed offline; CUDA version pinned
- [ ] **Network:** Air-gap topology established; internal mirrors configured
- [ ] **Models:** GGUF files transferred and integrity verified (SHA256)
- [ ] **Containers:** Apptainer `.sif` images built and tested
- [ ] **Monitoring:** Prometheus/Grafana stack deployed for observability
- [ ] **Security:** No egress routes; all external DNS blocked
- [ ] **Testing:** Load testing completed within VRAM constraints

7. Conclusion

Building a Sovereign AI capability is a systems engineering challenge that marries hardware constraints, network security, and operational discipline. While the upfront architectural cost is higher, the strategic value—**complete data sovereignty**—is non-negotiable for mission-critical sectors.

The key principles to remember:

1. **VRAM is King:** Size your hardware for the context window, not just the model.
2. **Mirror Everything:** Your internal environment must replicate the open-source supply chain.
3. **Quantize Intelligently:** Q4_K_M offers the best balance of size and quality.
4. **Containerize for Portability:** Build with Docker, deploy with Apptainer.
5. **Monitor Internally:** You cannot rely on external observability tools.

Next in this Series:

- **Whitepaper #02:** *The Disconnected Pipeline: Solving Dependency Management & Containerization in Secure Facilities.*

About the Author

Dustin J. Ober, PMP, M.Ed. *AI Developer & Technical Instructional Systems Designer*

Dustin J. Ober is a specialist in the intersection of Artificial Intelligence, Instructional Strategy, and secure systems architecture. With a background spanning over two decades in the United States Air Force and defense contracting, he focuses on deploying high-impact technical solutions within mission-critical environments.

Unlike traditional developers who focus solely on code, Dustin bridges the gap between **technical capability** and **operational reality**. His expertise lies in architecting "Sovereign AI" systems—designing offline, air-gapped inference pipelines that allow organizations to leverage state-of-the-art intelligence without compromising data security or compliance.

He holds a Master of Education in Instructional Design & Technology and is a certified Project Management Professional (PMP). He actively develops open-source tools for the AI community, focusing on DSPy implementation, neuro-symbolic logic, and verifiable agentic workflows.

Connect:

- **Web:** aiober.com
- **LinkedIn:** linkedin.com/in/dustinober

- **Email:** dustinober@me.com

Suggested Citation: Ober, D. J. (2025). *Sovereign AI Infrastructure: Architecting Intelligent Systems for Disconnected Environments* (Whitepaper No. 01). AIOber Technical Insights.