EXHIBIT VIII: Sovereign Operational Intelligence (The Docker Orchestration)

Project Scope: Transitioning Standalone BI Tools to a Containerized Microservices Stack

Architect: David Culp, Principal Solutions Architect

Key Value: Orchestrated a triple-service environment to ensure seamless data flow and service high-availability on local high-compute hardware.

---

1. EXECUTIVE SUMMARY

Building on the "Sovereign Architect" philosophy, I evolved a suite of standalone scripts into a fully containerized **Triple-Service Stack**. This Dockerization effort ensures that the Ingestion Engine, Backend API, and Frontend Dashboard operate as a unified, portable ecosystem. By isolating dependencies and orchestrating inter-service communication, I eliminated environment variance and created a resilient, production-ready operational intelligence environment that operates entirely within the private cloud.

---

2. THE CHALLENGE

- **Environment Parity:** Original tools relied on host-specific Python paths, making consistent deployment across Proxmox nodes difficult.
- **Volume Sync Complexity:** Moving to Docker introduced "Volume Stalling" where the host and Linux containers struggled to synchronize PDF data in real-time.
- **Service Discovery:** Enabling three distinct applications to communicate securely while maintaining a single, consistent SQLite data source.

---

3. THE ARCHITECTURAL SOLUTION

I implemented a `docker-compose` orchestration strategy segmentizing business logic into three distinct layers sharing a **Sovereign Data Volume**.

- **Layer 1 (The Engine):** Optimized the **OCR-Ingestion** container for real-time batch processing of multi-site invoices.
- **Layer 2 (The Bridge):** Engineered a custom **FastAPI Networking Layer** to bridge the isolated SQLite database and the external browser, resolving routing errors via manual endpoint injection.
- **Layer 3 (The Sync):** Resolved frontend data-binding issues by aligning JavaScript calls with verified JSON API outputs (e.g., `vendor_name` field mapping).

---

4. DEPLOYMENT GUIDE: SOVEREIGN STACK

1. **Orchestration Launch:** Deploy containers via `docker-compose up -d --build` to initialize the virtual network.
2. **Database Alignment:** Access the API container (`docker exec`) to create SQL views, ensuring ORM names match the injected data.
3. **Frontend Validation:** Access the dashboard at `localhost:8001` and perform a Hard Refresh (Ctrl + F5) to pull the live records.

---

5. BUSINESS OUTCOMES

- **Rapid Portability:** The entire 3-app stack deploys in seconds to any node in the organization's high-compute environment.
- **Technical Resilience:** Successfully navigated complex container-layer networking, ensuring "Sovereign Vendor" data flows from raw PDF to the GUI with zero packet loss.
- **Extreme ROI:** Leveraged existing hardware to replace expensive third-party BI subscriptions with a 100% private solution.