

# Motia OrderFlow: Architecture Strategy

Winning the Hackathon: The Multi-Tenant SaaS Pivot

## Project Manager's Directive

---

Our current status features a polished UI and a functional backend. However, to win the hackathon, we must shift from a “single-player” demo to a **Multi-Tenant SaaS Platform**. We need to prove that Merchant A and Merchant B can utilize the same engine simultaneously with total data isolation.

## Core Architectural Enhancements

---

### 1. Intelligent Fraud Guard (The Saga “Pause”)

We have implemented “Step A → Conditional Wait → Step B.” This reflects real-world enterprise logic where high-risk transactions are paused for manual or automated verification before inventory is committed.

### 2. Elastic Webhook Engine

To prove our platform is truly **Headless**, this engine enables real-time synchronization with external ERPs (SAP) and CRMs (Salesforce) via signed POST requests, moving beyond a simple standalone website.

### 3. Real-Time Saga Execution Pulse

Instead of a static progress bar, we have engineered an interactive, animated **Live Flow Graph** that visualizes the order’s journey through the backend steps in real-time.

- **Visual Proof of Event-Driven Logic:** By showing “data particles” flying between nodes, we prove the system is orchestrating a distributed process rather than just calling a function.
- **Dynamic Multi-Tenancy:** During the demo, we can toggle features (like “Fraud Check”) and show the graph structure dynamically changing its path for one merchant while remaining direct for another.

## The Sprint Plan: “The Multi-Tenant Pivot”

---

### Phase 1: Backend “Hardening” (High Priority)

- ✓ **Tenant Identity:** Implement mandatory X-Store-ID header checks. Return 400 Bad Request if missing.
- ✓ **Data Siloing:** Transition storage logic from global keys to partitioned paths: /inventory/{storeId}. This prevents cross-tenant data leaks.
- ✓ **Real Payments:** Connect to the payment\_gateway microservice via HTTP to demonstrate microservice orchestration.

## Phase 2: Frontend “Realism” (Medium Priority)

- ✓ **Live Flow Graph:** Use `react-flow` to “light up” nodes as the backend updates the order status via the `merchant_stats` Stream.
- ✓ **Merchant Switcher:** Integrate a UI dropdown to swap between “Store 123” and “Store 456” to demonstrate live multi-tenancy and unique flow paths.

## Phase 3: The “Wow” Factor (Hackathon Polish)

- ✓ **The Webhook Demo:** Send an automated POST request to a webhook tester upon order completion to prove external extensibility.

### Success Metrics for the Demo

**Isolation:** Can two browser tabs with different Store IDs see unique inventories and flow graphs?

**State Recovery:** Does order progress persist accurately after a page refresh?

**Real-Time:** Does the flow graph update instantly upon backend state changes?

### Hackathon Demo Statement

*“Not only is our platform multi-tenant, but it also supports custom fraud logic and real-time ERP synchronization via our webhook engine, all visualized through a live, dynamic Saga Execution Pulse.”*

*Strategic Roadmap — Version 1.6 — Ready for Execution*