

Clinical Supply Chain Control Tower (Agentic AI Architecture Design)

Executive Summary

Global Pharma Inc. faces two systemic problems in its clinical supply chain:

- 1. **Reactive management** → stock-outs when enrollment spikes.
- 2. **Invisible waste** → high-value drug batches expiring unused.

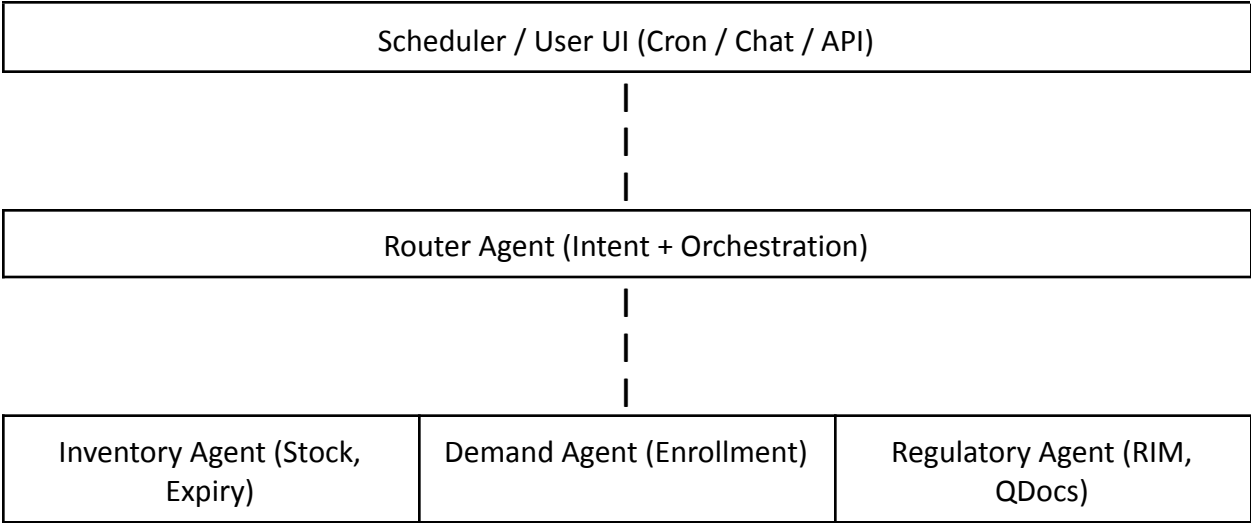
This solution proposes a **Multi-Agent AI Control Tower** that:

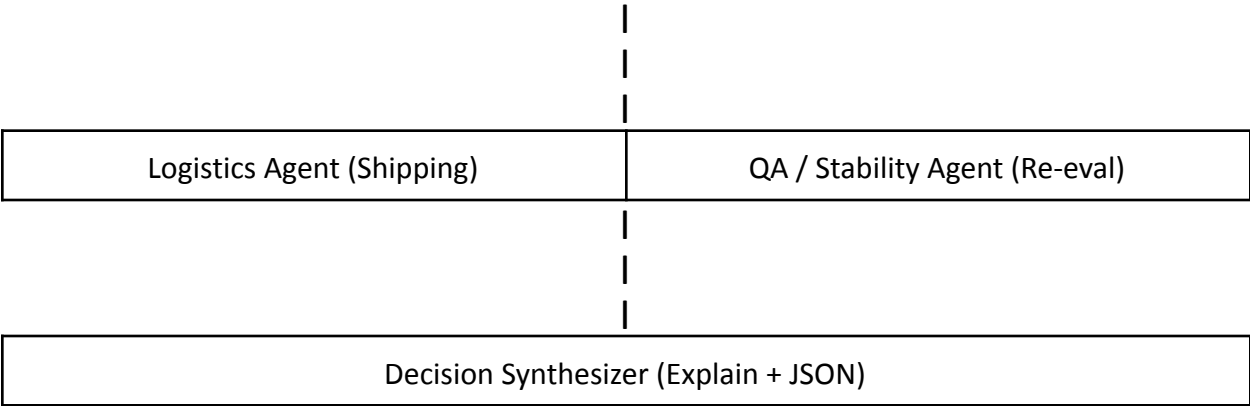
- **Autonomously monitors risk daily** (Supply Watchdog)
- **Assists human decision-makers conversationally** (Scenario Strategist)
- **Is auditable, explainable, and self-healing**
- Works directly on a **PostgreSQL-backed fragmented schema ( around 40 tables)**

The design intentionally separates **reasoning, data access, regulatory logic, and decision justification.**

PART 1: ARCHITECTURAL DESIGN (THE BLUEPRINT)

1. High-Level Architecture





**2. Agent Definitions & Responsibilities**

**2.1 Router Agent (Brain & Traffic Controller)**

Responsibility	Description
Intent detection	Distinguishes between autonomous monitoring vs user queries
Agent routing	Decides which domain agents to invoke
Memory	Stores conversation context and resolved entity mappings

**2.2 Inventory Agent**

Aspect	Details
Tables	Affiliate_Warehouse_Inventory, Allocated_Materials, Available_Inventory_Report
Core Logic	Expiry risk, batch availability, quantity aggregation
Outputs	Expiry buckets, stock availability

**2.3 Demand Agent**

Aspect	Details
--------	---------

Tables	Enrollment_Rate_Report, Country_Level_Enrollment
Core Logic	Demand forecasting using recent enrollment velocity
Outputs	Projected weekly consumption

#### 2.4 Logistics Agent

Aspect	Details
Tables	Distribution_Order_Report, IP_Shipping_Timelines
Core Logic	Lead time validation
Outputs	Earliest feasible execution dates

#### 2.5 Regulatory Agent

Aspect	Details
Tables	RIM, Material_Country_Requirements
Core Logic	Country-wise approval status
Outputs	Approved / Pending / Rejected

#### 2.6 QA / Stability Agent

Aspect	Details
Tables	Re_Evaluation, QDocs, Stability_Documents
Core Logic	Past shelf-life extensions

Outputs	Technical feasibility
---------	-----------------------

2.7 Decision Synthesizer Agent

Responsibility	Description
Explanation	Converts agent outputs into human-readable reasoning
JSON output	Structured payload for email or dashboards
Auditability	Explicit citation of tables & logic

PART 2: TECHNICAL IMPLEMENTATION STRATEGY

1. Tooling Layer (Agent Tools)

```
def run_sql_query(query: str) -> List[Dict]:
    """
    Executes read-only SQL on Postgres.
    Returns rows as list of dicts.
    """

def get_schema(table_name: str) -> Dict:
    """
    Returns column names + descriptions for a table.
    """

def log_decision(payload: Dict) -> None:
    """
    Stores AI decisions for audit and compliance.
    """
```

2. Teaching Schema Without Context Overload

Strategy Used

- 1. Schema Registry
Each agent has access only to its own table schemas

## 2. On-Demand Schema Fetching

`get_schema("Allocated_Materials")`

## 3. Column Alias Map

Canonical names used internally

Example:

```
{  
  
  "Allocated_Materials": {  
  
    "batch_id": ["batch_no", "material_batch"],  
  
    "expiry_date": ["exp_date", "shelf_expiry"],  
  
    "quantity": ["qty", "available_qty"]  
  
  }  
}
```

## 3. System Prompt – Supply Watchdog Agent

You are the Supply Watchdog Agent.

Goal:

Detect inventory and demand risks autonomously.

Rules:

- You run daily.
- You ONLY use SQL through `run_sql_query`.
- You MUST output structured JSON.
- You MUST classify risks as CRITICAL, HIGH, or MEDIUM.
- You MUST cite which tables and columns were used.

Risk Definitions:

- Expiry Risk:

Allocated batches expiring in  $\leq 90$  days.

Critical  $< 30$ , High  $< 60$ .

- Shortfall Risk:

If projected demand exceeds available inventory  
within the next 8 weeks.

Do not hallucinate.

If data is missing, report uncertainty explicitly.

#### **4. SQL Logic – Shortfall Prediction (Postgres)**

##### **Step 1: Weekly Demand Projection**

WITH weekly\_demand AS (

SELECT

country,

trial\_id,

AVG(enrollment\_rate) \* 7 AS weekly\_consumption

FROM enrollment\_rate\_report

WHERE report\_date  $\geq$  CURRENT\_DATE - INTERVAL '28 days'

GROUP BY country, trial\_id

)

## Step 2: Current Inventory

```
, available_stock AS (  
  
    SELECT  
  
        country,  
  
        trial_id,  
  
        SUM(available_quantity) AS total_inventory  
  
    FROM available_inventory_report  
  
    GROUP BY country, trial_id  
  
)
```

## Step 3: Shortfall Detection

```
SELECT  
  
    d.country,  
  
    d.trial_id,  
  
    a.total_inventory,  
  
    d.weekly_consumption,  
  
    (a.total_inventory / NULLIF(d.weekly_consumption,0)) AS weeks_of_cover  
  
FROM weekly_demand d  
  
JOIN available_stock a  
  
ON d.country = a.country  
  
AND d.trial_id = a.trial_id  
  
WHERE (a.total_inventory / NULLIF(d.weekly_consumption,0)) <= 8;
```

## 5. Output JSON (Example)

```
{  
  "risk_type": "SHORTFALL",  
  "severity": "HIGH",  
  "trial": "Trial_ABC_v2",  
  "country": "Germany",  
  "weeks_of_cover": 5.2,  
  "source_tables": [  
    "Enrollment_Rate_Report",  
    "Available_Inventory_Report"  
  ],  
  "recommended_action": "Expedite replenishment or reallocate stock"  
}
```

## **PART 3: EDGE CASE HANDLING**

### 1. Ambiguous Entity Resolution

**User:** "Trial ABC"

#### **Resolution Flow**

1. Fuzzy match on **trial\_name**
2. Rank by:
  - Exact match
  - Version proximity



- Recent activity
3. Ask clarification if confidence < 0.8

```
{  
  "possible_matches": [  
    "Trial_ABC",  
    "Trial_ABC_v2"  
  ],  
  "confidence": 0.65  
}
```

## 2. Invalid SQL – Self-Healing Loop

### Recovery Mechanism

SQL Execution Error → Parse error message → Send error + query to SQL-Fix Agent → Generate corrected SQL → Re-run query (max 3 attempts) → Escalate to human if unresolved

## 3. Scenario Strategist – Shelf-Life Extension Example

### User Query:

“Can we extend expiry of Batch #123 for the German trial?”

### Agent Checks

Constraint	Agent	Result
Technical	QA Agent	Yes (Re-evaluated before)
Regulatory	Regulatory Agent	Approved in DE
Logistical	Logistics Agent	21 days buffer

### Final Answer

```
{
```

```
"decision": "YES",  
  
"reasoning": {  
  
  "technical": "Batch #123 successfully re-evaluated previously",  
  
  "regulatory": "German authority approval found in RIM",  
  
  "logistical": "Shipping timelines allow execution"  
  
}  
  
}
```

#### **N8N Integration (High Level)**

Component	Implementation
Scheduler	Cron node
AI Agent	HTTP → LLM
Database	Postgres node
Alerts	Email / Slack node
Audit	Write to DB