Tracing and Quality Check: Validating Agent Trajectory

The system utilizes the **`minimal_tracing_plugin`** from the Google ADK, integrated with the FastAPI backend, to log the full trajectory of every query. This approach is essential for demonstrating the strategic success of the **Orchestrator Agent** and for quality assurance (QA).

Summary of Strategic Routing Success

We validated the Orchestrator's planning across two distinct domain-specific queries:

Query Scenario	Orchestrator Decision	Final Tool Execution	QA Result
Client Service (Log 1)	Routes to **`client_service_agent`**	Executes **Client Service Level Calculation**	**CORRECT:** The agent correctly selected the tool for time-based client analysis.
Inventory Aging (Log 2)	Routes to **`stock_analysis_agent`**	Executes **Inventory Aging Calculation**	**CORRECT:** The agent correctly selected the tool for obsolete stock identification.

Detailed Trace Analysis: The "Trajectory is the Truth"

The following log segments demonstrate the clean delegation and tool execution for the Inventory Aging query:

1. **Orchestrator's Decision:**
    - The `warehouse_orchestrator_agent` initiates the process (`12:47:57,673`).
    - After the first **LLM Request** (LLM Request #1), the Orchestrator correctly identifies the query's domain (obsolete stock) and delegates control in less than **1.5 seconds**.
    - The trace confirms the transfer of control: 🔍 `[TRACE] Agent 'stock_analysis_agent' started - Count: 2`.
2. **Specialized Agent's Execution:**
    - The **`stock_analysis_agent`** takes over and performs its own LLM call to determine the exact tool needed (LLM Request #2).
    - The agent successfully executes its data utility: `api - INFO - Calculated average time in warehouse for references...` (Timestamp `12:48:04,323`).
    - The total trajectory time for the specialized analysis was approximately **7 seconds** (from `12:47:57` to `12:48:04`), demonstrating efficient orchestration and execution of the core business logic.

Conclusion: The tracing capabilities confirm that the Orquestrator functions as the central **routing strategy**, consistently and accurately delegating complex requests to the appropriate specialized agents for tool execution, ensuring a robust and auditable system.