■■ AuroSys Enterprise Architecture Guide

This document contains detailed specifications to help you create an architecture diagram for the AuroSys Enterprise system.

1. High-Level Architecture (The "Hybrid" Model)

The system is divided into three distinct layers: 1. ■ **EDGE (Vehicle)**: Real-time sensor processing, safety logic, and bandwidth optimization. 2. ■■ **CLOUD (OEM Backend)**: Heavy-lift AI analysis, supply chain integration, and fleet management. 3. ■ **USER (Interface)**: Dual interfaces for Engineers (Dashboard) and Drivers (Mobile App).

---

2. Component Logic & Data Flow

A. The "Edge" Layer (In-Car)

Input Streams:

- **Vibration Sensors (400Hz)**: Simulated Accelerometer data.
- **CAN Bus**: RPM, Speed, Throttle Position, Engine Temp.

Active Agents:

1. **Telematics Agent**: Aggregates raw sensor data into `TelemetryFrame`.
2. **Driver Behavior Agent**: Calculates `Safety Score` & `Driver DNA` locally using physics models (Speed/RPM ratio).

   Diagnosis Agent (GenAI): Runs local inference on vibration waveforms.

   - *Logic*: Detects "Rod Knock" vs "Misfire" patterns.

   Comms Module:

   - *Function*: Bandwidth Optimization.
   - *Output*: Compresses 100MB Raw Data -> 1.5MB "Blackbox Dump" (JSON + Critical Waveform Snippet) for transmission.

B. The "Cloud" Layer (OEM Backend)

Orchestrator:

- **Master Agent**: The central controller that receives the "Blackbox Dump" and routes it to specialized sub-agents.

Specialized Agents:

RCA Agent (Forensics):

- Correlates specific fault codes with Manufacturing Database (e.g., "Batch-2023-A" defects).

Inventory Agent:

- Queries Logistics Database for part availability (e.g., "Connecting Rod Bearing" at "Chennai Hub").

Financial Agent:

- Calculates `Total Estimate` (Parts + Labor Cost in INR).

Scheduling Agent:

- Finds nearest "Hero Hub" service center based on GPS and books a slot.

5. **Compliance & OTA Agents**: Checks regulatory safety and deploys software patches if the fault is software-related.

Storage:

- **SQLite DB (`audit_log`)**: Stores an immutable trace of every agent's decision (`AgentLogStep`).

C. The "User" Layer (Presentation)

Engineering Console (Streamlit Web App):

- **3D Spectrograms**: Visualizes the raw vibration data (Order Analysis).
- **Trace Viewer**: Displays the step-by-step logic log from the Master Agent.
- **Strategic Decision Card**: Shows high-level business actions (Recall/OTA) via `ui_lib.py`.

Driver Companion (Mobile Simulator):

- **Notification System**: Receives the "Driver Friendly Message" (e.g., "Critical Issue Detected").
- **Booking UI**: Allows the user to confirm the appointment reserved by the Scheduling Agent.

---

3. Data Flow Scenario ("The 4-Second Loop")

1. **[0.0s] Event**: Vibration spike detected (Rod Knock).
2. **[0.1s] Edge Processing**: `Diagnosis Agent` flags severity -> `Comms Module` packages payload.
3. **[0.5s] Telemetry**: JSON Payload transmitted via 5G/4G.

[1.2s] Cloud Orchestration: `Master Agent` wakes up.

- -> `RCA Agent` confirms "Batch Defect".

    - -> `Inventory Agent` reserves part.
    - -> `Financial Agent` sums costs.
5. **[3.0s] Decision**: `Logic Engine` (`core.py`) creates a Strategic Decision (e.g., "Targeted Recall").
6. **[3.5s] Notification**: Driver App buzzes; Engineer Dashboard updates live.

---

4. Key Technologies

- **LLM**: Google Gemini 2.0 Flash (Reasoning & Text Generation)
- **Frontend**: Streamlit + Plotly + Custom CSS/HTML
- **Backend**: Python (Pandas / NumPy / SQLite)

5. Diagram Suggestions

- **Layout**: Left-to-Right flow (Car -> Cloud -> Users).

  Icons:
    - *Car*: Sensor/Chip icon.
    - *Cloud*: Server/Database icon.
    - *Agents*: Robot/Brain icons (One for each specific agent).
    - *Users*: Mechanic (Laptop) and Driver (Phone).

  Colors:
    - Create a distinction between "Real-time Data" (Red arrows) and "Agent Queries" (Blue arrows).