High Level Design Document

Introduction

This High Level Design (HLD) document outlines the architecture and core components for **CivicSync - Public Service Request Optimizer**. CivicSync is a government web platform leveraging agentic-AI to analyze, prioritize, and route citizen service requests, predict resource needs, optimize response times, and provide actionable reports for city officials. The platform includes secure authentication, real-time dashboards, and collaborative tools for public sector teams.

---

1. System Architecture Overview

Architecture Summary:
CivicSync is a modular, containerized web application comprising a Vue.js frontend, Python-based backend services (including agentic-AI modules), a PostgreSQL database, and secure authentication. All components are orchestrated via Docker.

Module	Description
Web Frontend	Vue.js SPA for user interaction, dashboards, and team collaboration
API Gateway	Python-based REST API, request validation, authentication
Agentic-AI Engine	Python service for request analysis, prioritization, routing, and prediction
Database	PostgreSQL for persistent storage of requests, users, and analytics
Auth Service	Secure authentication and authorization (OAuth2/JWT)
Reporting Module	Generates actionable reports for officials
Notification Service	Sends alerts/updates to users and teams

---

2. Component Interactions

Sequence Step	Interaction Description
1	User submits service request via Web Frontend
2	API Gateway authenticates and validates request
3	Request stored in Database; forwarded to Agentic-AI Engine
4	Agentic-AI analyzes, prioritizes, and routes request; predicts resource needs
5	Results and status updates sent to Web Frontend and Notification Service
6	Reporting Module aggregates data for dashboards and official reports
7	Collaborative tools enable team communication and task management

---

3. Data Flow Overview

Data Flow	Source	Destination	Purpose
Service Request Submission	Web Frontend	API Gateway	Capture citizen requests
Request Analysis & Routing	API Gateway	Agentic-AI Engine	Prioritize and route requests
Data Persistence	API Gateway	Database	Store requests, users, analytics
Status/Notification Updates	Agentic-AI	Notification Service	Inform users/teams of updates
Dashboard/Report Data	Database	Reporting Module	Generate insights for officials
Auth/Session Validation	Web Frontend	Auth Service	Secure access control

---

4. Technology Stack

Layer/Component	Technology/Framework
Frontend	Vue.js
Backend/API	Python (FastAPI/Flask)
AI/ML	Python (agentic-ai libraries)
Database	PostgreSQL
Auth	OAuth2/JWT
Containerization	Docker
Orchestration	Docker Compose/Kubernetes

---

5. Scalability & Reliability

- **Scalability:**
  - Stateless backend services enable horizontal scaling.
  - Database supports replication and partitioning.
  - AI engine can be scaled independently for high request volumes.
- **Reliability & Security:**
  - Secure authentication (OAuth2/JWT) and encrypted data transmission.
  - Role-based access control for sensitive operations.
  - Regular backups and health monitoring for all services.
  - Containerization ensures consistent deployment and isolation.

End of Document