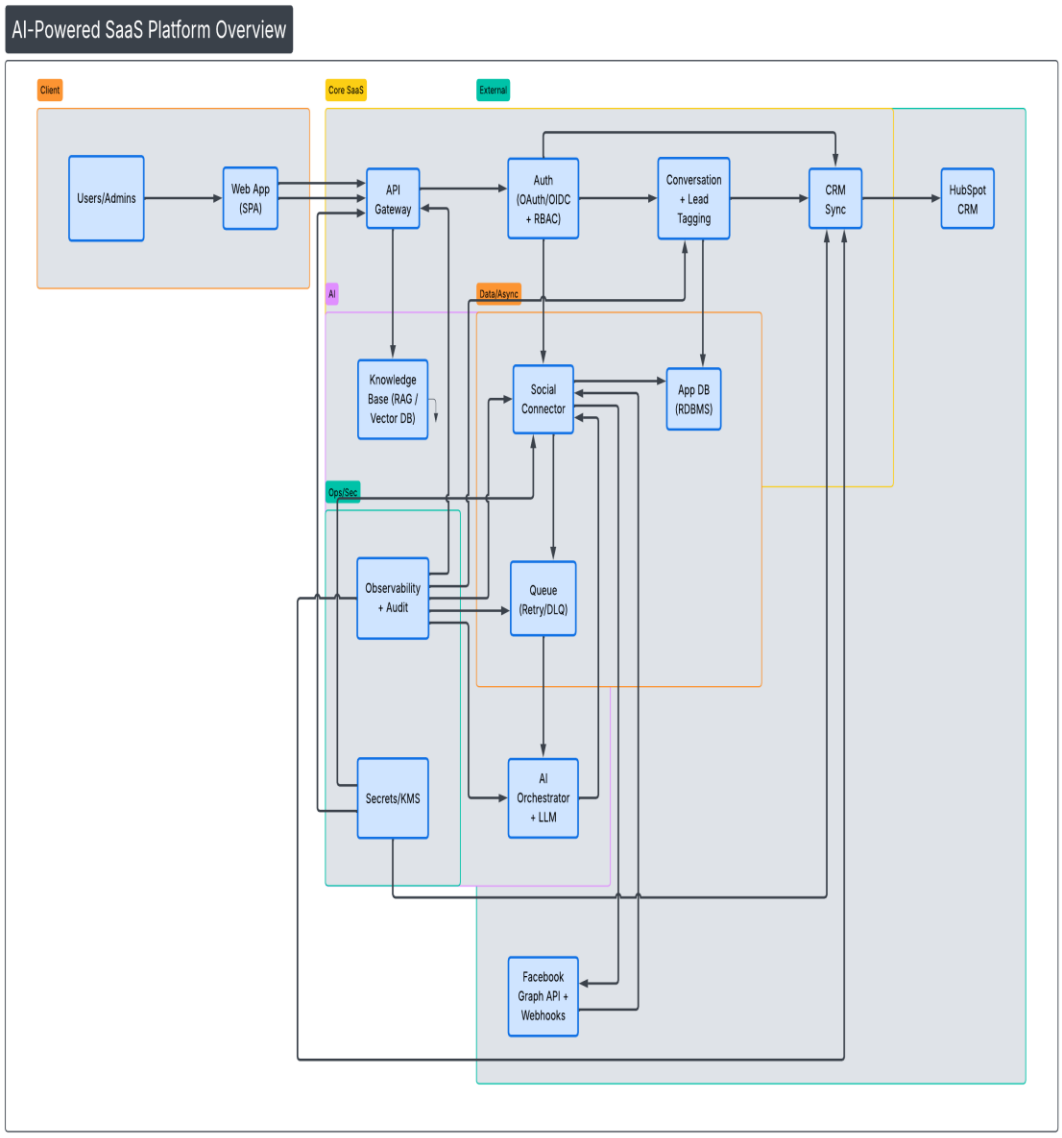
AI Agent System Design for Web-Based Saas Platform

Objective: End-to-End AI Agent Integration for Automated Social Media Engagement

Prepared by: Shirin Mahbuba

This project has been developed for an extendable web-based SaaS platform that allows any company to enable AI in managing social media interactions. The system will automatically respond, classify the lead with intent, and synchronize all data with any CRM, such as HubSpot.

System Architecture Overview



Note: The architecture is designed around a multi-tenant model of microservices, enabling high availability, security, and scalability.

AI agents and services involved

- **AI Orchestrator & LLM:** Functions as a central logic system. It receives each message event, obtains the conversation context, and uses LLMs to create contextual responses.
- **Knowledge Base (RAG / Vector DB):** This service is based on the technology used in Assessment 2. The FAQs and policies are stored in the Vector Database (FAISS). The AI's responses are backed up by actual company facts.
- **Conversation & Lead Tagging Service:** A specialized service that labels every interaction as Hot, Warm, or Cold using rule-based logic and light model classifiers

End-to-End Data Flow

A. Setup Flow: The admins will use the Web App (SPA) to connect their Facebook Page using OAuth 2.0. Integrations store tokens securely in Secrets/KMS.

B. Runtime Messaging Flow:

- **Ingestion:** Events are sent to Social Connector from the Facebook Webhooks
- **Queueing:** Messages are buffered for reliability within the Retry/DLQ (Dead Letter Queue).
- **Generation:** The AI Orchestrator retrieves relevant snippets from the Knowledge Base and prepares a draft of the answer.
- **Action:** Reply via Facebook Graph API; update the lead status in App DB.
- **Synchronization:** Using logics from Assessment 1, the CRM Sync module will upsert data to HubSpot CRM.

Security, Privacy, and Authentication

- **Authentication:** Leveraging OAuth 2.0 / OIDC for secure user login and RBAC for tenant isolation.
- **Data Security: * TLS Encryption:** Applied for all data in transit.
- **Encryption of data at Rest:** For App DB and Vector store.
- **Secrets Management:** Secret API keys are managed using a Key Management Service (KMS).
- **Auditability:** In "Centralized Observability & Audit," all admin changes and message handling actions are recorded.

Operational Strategies

Cost Optimization

- **Model Tiering:** Employ inexpensive small models to filter leads and keep expensive high-parameter models for generating complex RAG-based responses.
- **Asynchronous Scaling:** The Async Queue allows the background workers to scale at peak traffic independently, thus reducing server costs.

Failure Scenarios & Recovery

- **API Resilience:** In case of failure by either LLM or Facebook API, the system will perform exponential backoff retries.
- **Dead-letter Queue Handling:** Failed messages are routed to a dead-letter queue for human intervention. CRM Outages: Data stays in App DB, retrying the sync once CRM is up again.

This architecture represents the strategic evolution of functional components previously validated through hands-on prototyping. n8n was used in **Assessment 1** to prove the efficiency of basic automation and CRM synchronizations, which have shown reliable data flow across platforms. More importantly, the core intelligence of the platform is anchored on the RAG framework developed in **Assessment 2**, which validated the use of FAISS and LLMs to ensure accurate and factual information retrieval. Consequently, putting these elements together yields a robust and technically feasible solution for enterprise-grade AI engagement.