AI-Powered Communication Assistant - Architecture & Approach Documentation

🏗️ System Architecture Overview

INTRO- This system automates customer email processing and response generation with a three-layer structure: Backend API, Frontend Dashboard, and Knowledge Base. It mixes automated AI workflows with human oversight to keep communication accurate and professional.

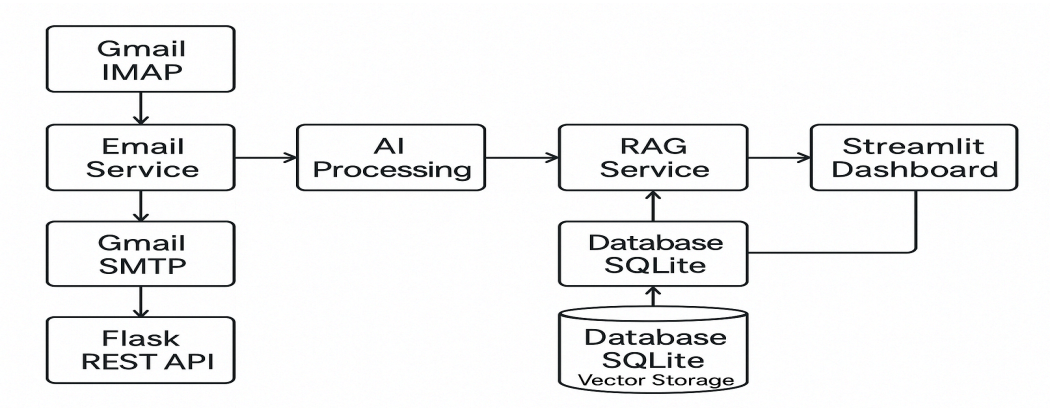
Working Process -Emails are fetched from Gmail using IMAP and sent to the backend for sentiment analysis, categorization, and response generation. Relevant knowledge is retrieved through a semantic search system (RAG). The generated responses appear in a Streamlit-based dashboard where support staff can review, edit, and send replies through Gmail SMTP. All emails, responses, and knowledge entries are stored in a structured database.

Components

1.Backend (Flask API) manages email services, AI processing, semantic search, and database operations.

2.Frontend (Streamlit Dashboard) provides a real-time interface for email queues, AI responses, analytics, and settings.

3.Knowledge Base contains FAQs, policies, and templates, stored with embeddings for semantic retrieval.



Workflow (Design Approach)

STEP 1: The system fetches support-related emails periodically from Gmail.

Each email is analyzed for sentiment and urgency.

STEP 2: Relevant knowledge is retrieved using sentence-transformer embeddings.

STEP 3: A context-aware draft response is generated.

STEP 4: Support agents review and edit the response in the dashboard before sending.

Data & Storage

Structured data, such as emails, responses, and metadata, is stored in SQLite, while embeddings enable semantic search. JSON fields hold flexible metadata.

Key Features

Automated fetching, filtering, sentiment analysis, and prioritization of emails.

RAG-enhanced responses using semantic search across internal documents.

Real-time dashboard for monitoring, editing, and analytics.

Scalable design that supports high volumes of emails with efficient processing.

Security & Reliability

The system uses environment variables for secure credentials, along with input validation, error handling, and retry mechanisms. Regular backups and monitoring help ensure operational reliability.

Design Approach

Built with modular, loosely connected components, the system prioritizes maintainability, real-time feedback, and ease of use. Human oversight remains central, combining AI efficiency with professional standards for customer communication.