Gen AI Orchestrator for Email and document triage/Routing

**Diagram Structure**

A diagram of a software system

AI-generated content may be incorrect.

**Architecture Components**

**1. Input Layer**

* **User Interface (UI)**:
  + Angular-based web application.
  + Allows users to upload .eml files and input additional attributes (e.g., routing preferences).
  + Sends data to the backend API.

**2. Backend API Layer**

* **Flask API**:
  + Handles file uploads and processes input data.
  + Reads the email body from .eml files.
  + Extracts relevant attributes (e.g., amount, coordinates, address) using Python logic or regex.
  + Sends extracted data to the AI model for further processing.

**3. AI Processing Layer**

* **Generative AI (Gemini Model)**:
  + Processes the extracted email/document content.
  + Generates insights or routes the document based on predefined rules or AI predictions.
  + Example: Extracts key attributes like amount, coordinates, and address and determines the appropriate routing.

**4. Data Storage Layer**

* **File Storage**:
  + Stores uploaded .eml files temporarily for processing.
* **Database (Optional)**:
  + Stores metadata, extracted attributes, and routing history for auditing and analytics.

**5. Output Layer**

* **Response to UI**:
  + Sends processed data or routing decisions back to the Angular UI.
  + Displays extracted attributes and routing results to the user.

**6. External Services (Optional)**

* **Email Services**:
  + Integrates with email servers (e.g., Gmail, Outlook) to fetch .eml files directly.
* **Third-Party APIs**:
  + Calls external APIs for additional processing or validation (e.g., address verification).

**Flow Description**

1. **User Interaction**:
   * User uploads an .eml file and provides additional input via the Angular UI.
   * The UI sends the file and input data to the Flask API.
2. **Backend Processing**:
   * The Flask API reads the .eml file and extracts the email body.
   * Extracts key attributes (e.g., amount, coordinates, address) using regex or AI.
3. **AI Processing**:
   * The extracted data is sent to the Generative AI model (e.g., Gemini).
   * The AI model processes the data and generates routing decisions or insights.
4. **Response to UI**:
   * The Flask API sends the processed data and routing results back to the Angular UI.
   * The UI displays the results to the user.
5. **Storage**:
   * The .eml file and extracted data are optionally stored for future reference or auditing.