Schema Mapper System - Design Document

**Overview**

Converts unstructured text into structured JSON using Azure OpenAI GPT-4 and Streamlit. Processes .txt and .json files with user-defined schemas.

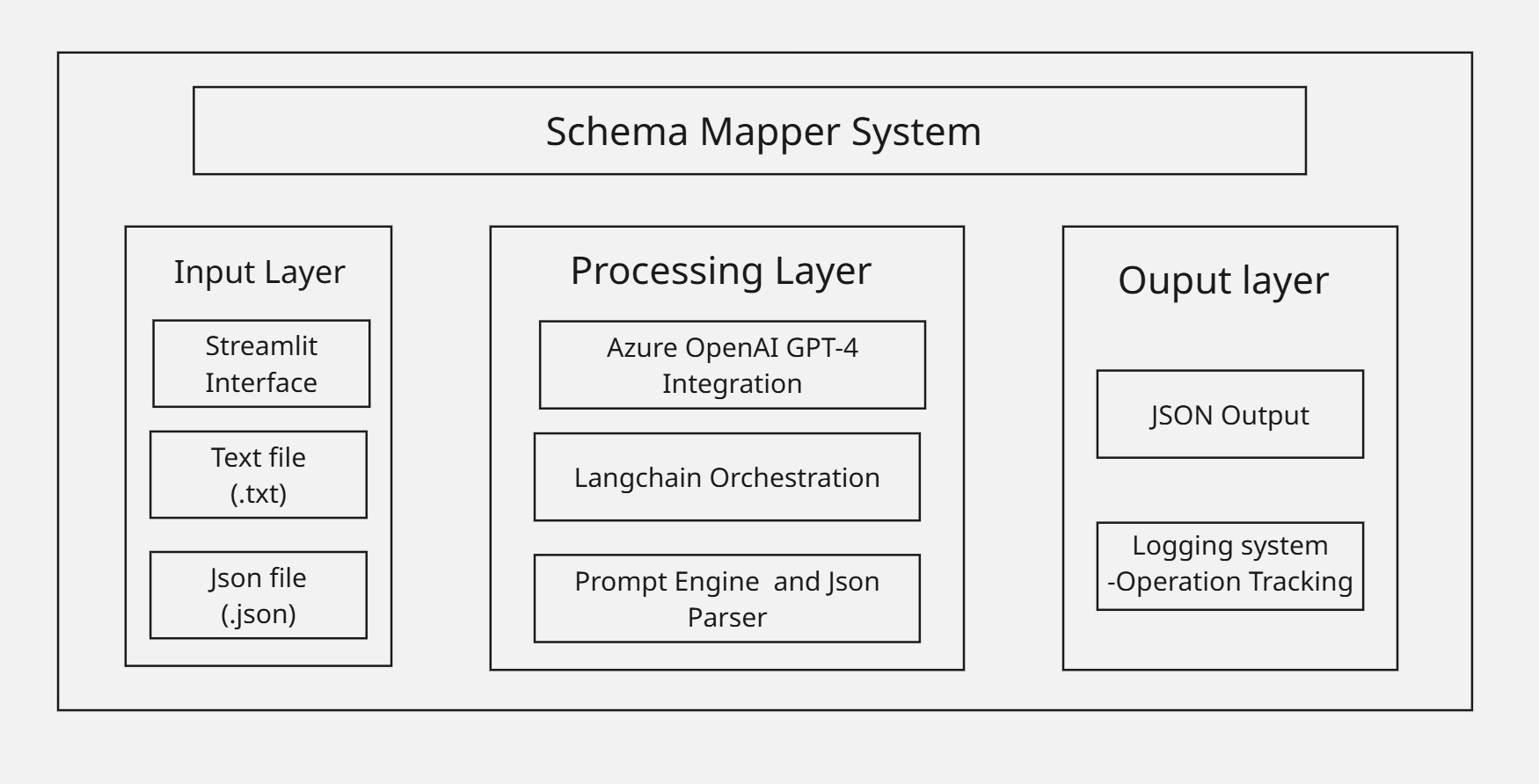
**Architecture**

Figure: Architecture of Schema Mapper System

**Input Layer:**

* **Streamlit web interface:** User-friendly web UI for file uploads and interaction
* **Text files (.txt) and JSON files (.json):** Accepts unstructured text and schema definitions as input

**Processing Layer:**

* **Azure OpenAI GPT-4 (128k context):** AI engine that transforms unstructured data using large language model capabilities
* **LangChain orchestration:** Framework that manages AI workflow and prompt execution
* **Prompt engine and JSON parser:** Components that optimize AI instructions and validate structured output

**Output Layer:**

* **Structured JSON output:** Final transformed data conforming to provided schemas
* **Logging and operation tracking:** System monitoring and audit trail for all processing activities

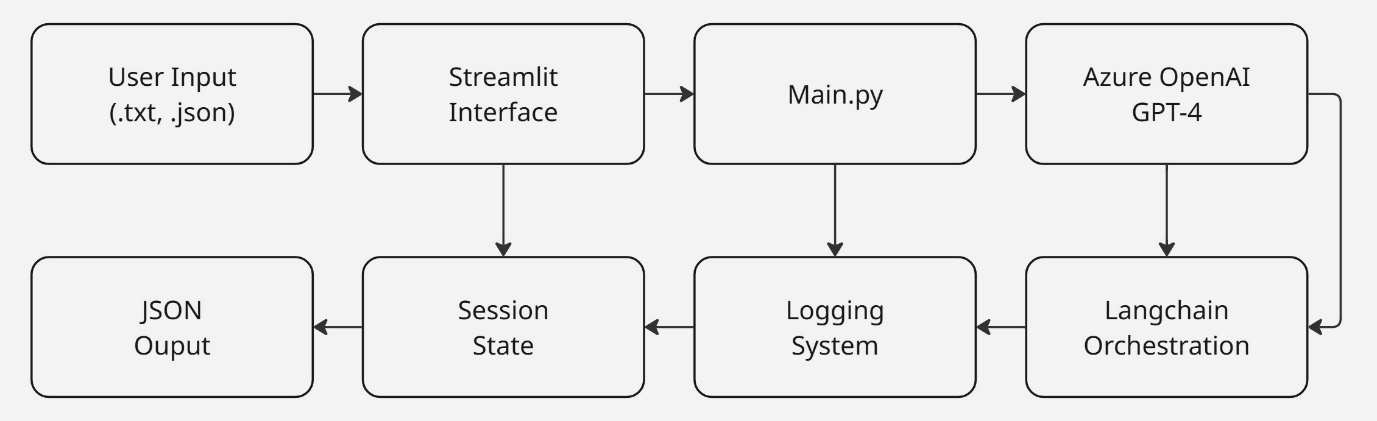
**Data Flow**

Figure: Data flow of Schema mapper

**Data Flow Explanation**

**Primary Processing Flow (Top Row - Left to Right):**

1. **User Input (.txt, .json)** → **Streamlit Interface** → **Main.py** → **Azure OpenAI GPT-4**
   * User uploads files through web interface, which routes to main processing module, then sends to AI engine

**Supporting Systems Flow (Bottom Row - Right to Left):**

1. **Azure OpenAI GPT-4** → **Langchain Orchestration** → **Logging System** → **Session State** → **JSON Output**
   * AI responses flow through orchestration layer, get logged for monitoring, maintain session context, and produce final structured output

**Key Data Flow Characteristics:**

* **Bidirectional Support:** Each component in the bottom row supports the corresponding component above it
* **State Management:** Session state maintains user context throughout the process
* **Monitoring:** Logging system tracks all operations for debugging and analytics
* **Orchestration:** LangChain manages the AI workflow and prompt execution
* **Output Generation:** Final JSON output is generated after processing through all supporting systems