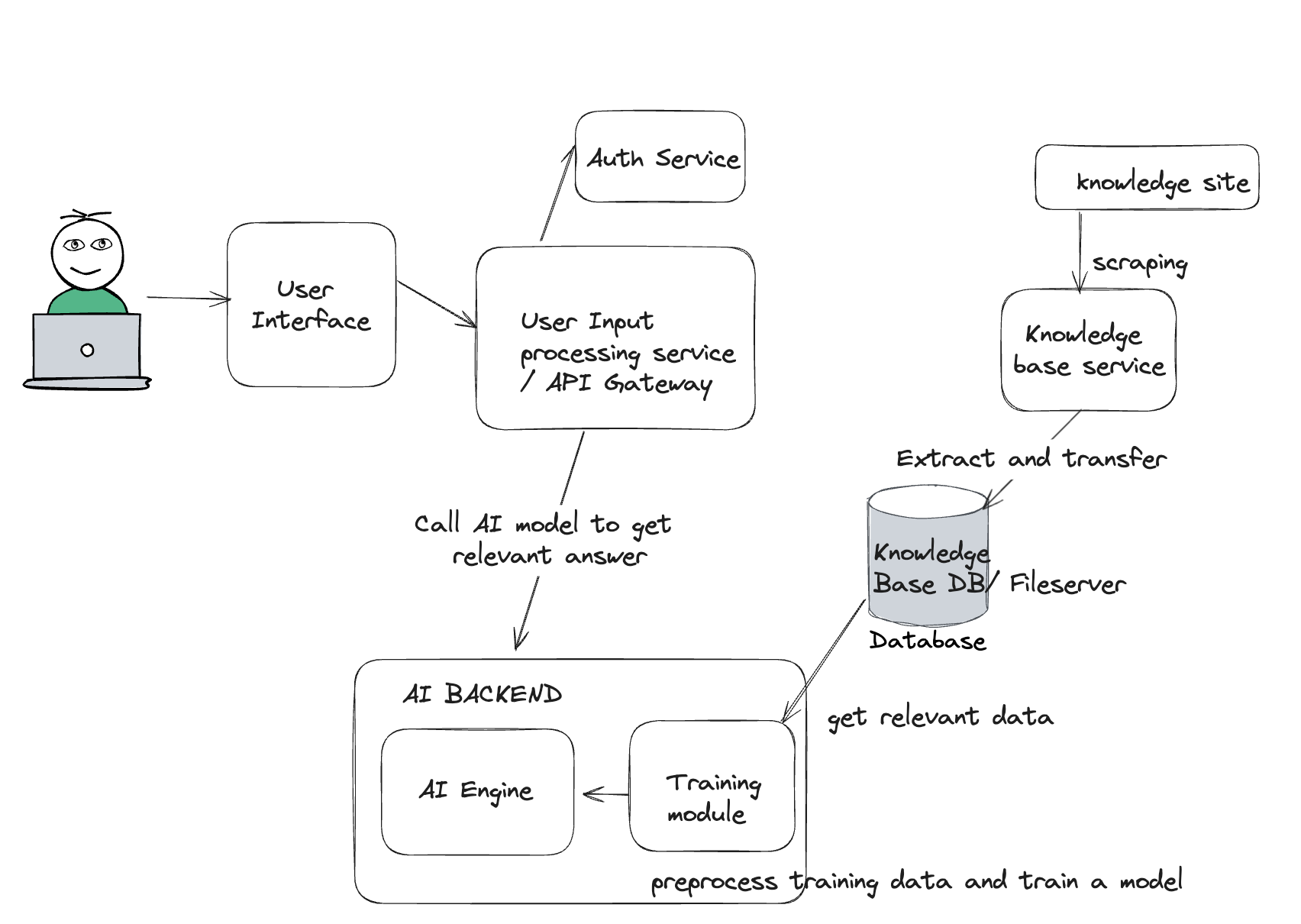
# **AI-Powered Support Bot - Architecture Document**



## **1. Introduction**

This document outlines the architecture for an AI-powered support bot that interacts with users, processes their natural language queries, and retrieves relevant information from a knowledge base. The system is designed with components that handle natural language processing (NLP), dialog management, response generation, and error handling.

## **2. System Architecture Overview**

The architecture is built with the following components:

* **User Interface (UI)**: Users can interact with the bot by entering queries.
* **User Input processing service:**  Handles incoming requests and routes them to appropriate services. Also handles authentication. This is the main backend which interacts with the UI and exposes APIs.
* **AI Engine**: Processes and interprets user queries to extract intent and entities through trained models.
* **Training Module:** This is part of the AI backend and is used to train models as per relevant usecase and data.
* **Knowledgebase**: Stores the raw data that the bot uses to answer user queries.
* **Knowledgebase service:** This service serves as the entry point for webscraping the data and transferring it to the knowledgebase DB.

## **3. Components Overview and Data Flow**

### **3.1 User Interface (UI)**

* **Purpose**: The UI is the point of interaction between the user and the bot. It can be a web-based chat interface or a mobile application.
* **Technology**: HTML/CSS, JavaScript (or ReactJS for a modern front-end).
* **Data Flow**: Users enter their queries, and the interface sends these queries via an API call to the backend.

# **3.2 API Gateway/ User Input processing service**

* **Purpose**: Acts as the entry point for all user queries. It routes the requests to the user input processing service and handles response delivery.
* **Technology**: Flask/FastAPI in Python.
* **Data Flow**: Receives the user query, passes it to the AI engine, and returns the response back to the UI.

### **3.3 AI Engine**

* **Purpose**: Processes and interprets the user's natural language query. Extracts intent and entities from the input, through trained models
* **Technology**: SpaCy, Hugging Face Transformers, or NLTK for NLP tasks.
* **Data Flow**: The query is tokenized, and the intent is identified. Based on the processed input, entities like service requests or keywords are extracted and passed to the dialog management system.

### **3.4 Training Module**

* **Purpose**: To train models as per different usecases
* **Technology**: SQLite, Pandas (for CSV processing), or any simple relational database, SpaCy, Hugging Face Transformers, or NLTK for NLP tasks.
* **Data Flow**: Based on the usecase (chatbot or some other) this module comes into action to train a specific model which is stored in the AI engine and is then used to answer chatbot queries.

### **3.4 Knowledgebase**

* **Purpose**: Stores all the relevant information the bot uses to respond to user queries. The knowledgebase can be in the form of a database, CSV, or JSON dataset.
* **Technology**: SQLite, Pandas (for CSV processing), or any simple relational database.
* **Data Flow**: Based on the processed query, the knowledgebase is searched for matching questions and corresponding answers.

### **3.6 Knowledgebase Service**

* **Purpose**: This service serves as the entry point for web scraping the data and transferring it to the knowledgebase DB.
* **Technology**: Python, BeautifulSoup, CSV parser, Pandas.
* **Data Flow**: This could be a cron job to scrape data periodically and periodically retrain a model or else could also be trigger-based.