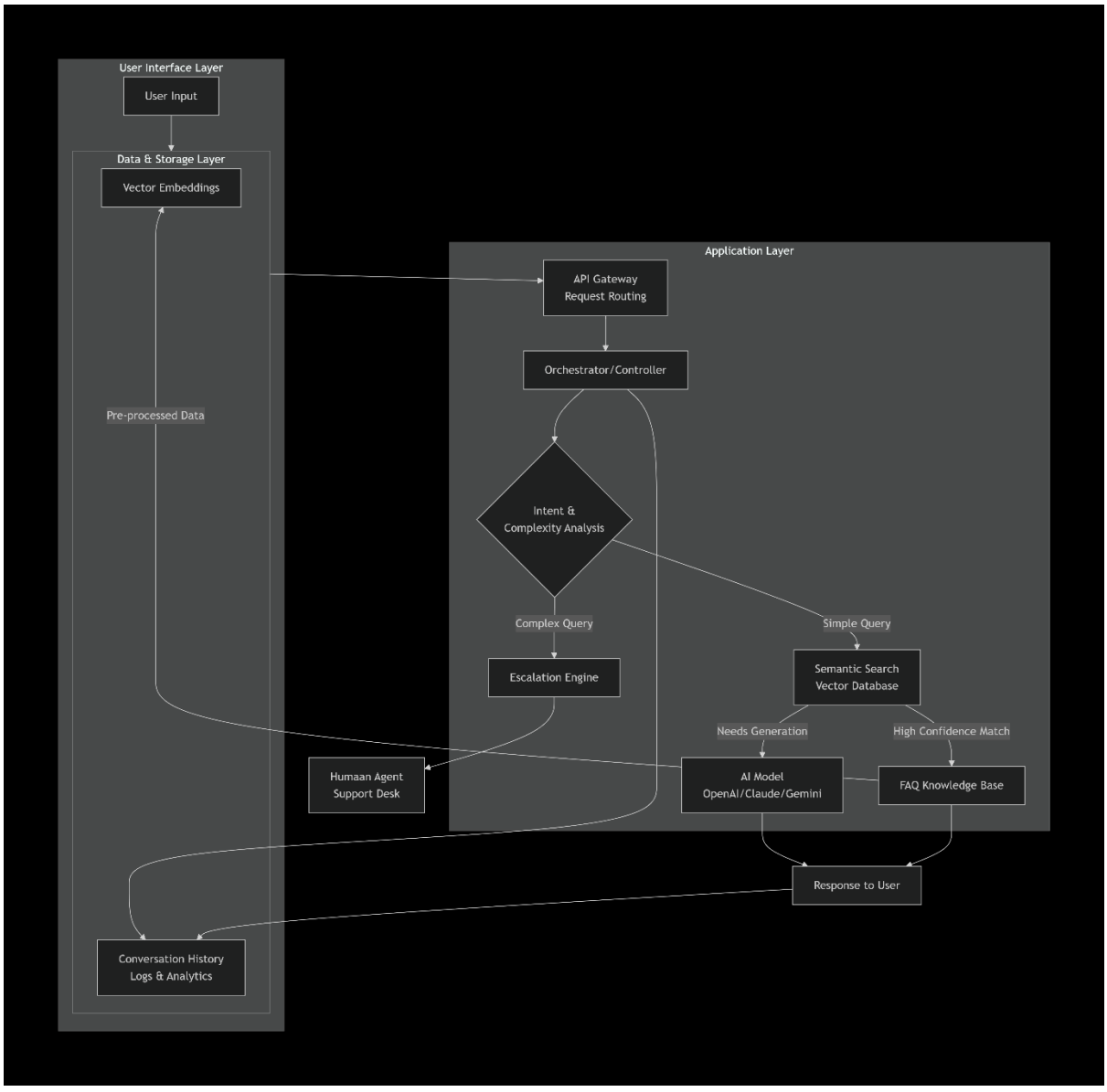
**Architecture of an AI Support Assistant**

An AI Support Assistant designed to resolve FAQs and escalate complex queries typically follows a layered architecture that processes user input, finds the best response, and decides when a human should take over. Here is a breakdown of its core components and data flow.




**Key Components and Technologies**

- **User Interface (UI):** A web-based chat interface (like your single HTML file) or mobile app that users interact with. It sends queries to and displays responses from the backend server.
- **API Gateway:** Acts as a single entry point for all client requests, handling routing, rate limiting, and authentication.
- **Orchestrator (Core Controller):** This is the brain of the operation. It coordinates the workflow: receiving a user query, calling the intent analysis, deciding which service (FAQ lookup or AI model) to use, and formulating the final response.
- **Semantic Search & Vector Database:** For FAQ resolution, this is a crucial component.

    - **FAQ Knowledge Base:** A storage system (like a database or JSON files) containing pre-defined questions and answers.
    - **Vectorization Model:** A machine learning model (like a Universal Sentence Encoder) that converts text into numerical vectors (embeddings).
    - **Vector Database:** A database (e.g., ChromaDB, Pinecone) that stores the vector embeddings of your FAQs. It performs a similarity search to find the most relevant answer to a user's query, even without exact keyword matches.
- **AI Model (LLM - Large Language Model):** A general-purpose model like OpenAI's GPT, Claude, or Gemini. It's used to generate answers for queries that aren't perfectly matched in the FAQ database but don't require escalation. It can also rephrase or enrich answers from the knowledge base.
- **Escalation Engine:** A rules-based or model-based classifier that analyzes the user's query and conversation history for keywords (e.g., "speak to a human," "manager") or signs of complexity/frustration. When triggered, it routes the conversation to a human agent and can create a support ticket.
- **Data Storage:** Persists conversation history for training, analytics, and providing context in ongoing chats. It also stores logs for debugging and performance metrics.

**How the Components Interact**

1. **User Input:** A user submits a question through the chat interface.
2. **Request Handling:** The API gateway receives the query and forwards it to the orchestrator.
3. **Intent & Complexity Analysis:** The orchestrator first checks if the query should be escalated based on predefined rules (keywords like "complaint," "lawyer," "not working"). If yes, the escalation engine takes over.
4. **FAQ Resolution (for non-complex queries):** The orchestrator sends the query to the semantic search module. The user's question is converted to a vector, which is compared against all FAQ vectors in the database. The most similar FAQ answer is returned if it meets a confidence threshold.
5. **AI Generation (fallback):** If no good FAQ match is found, the query is sent to the LLM to generate a helpful, context-aware response.
6. **Response & Escalation:** The chosen response (from FAQ or LLM) is sent back to the user. If the conversation is escalated, the user is notified that a human agent will take over, and a ticket is created in the support system.
7. **Logging:** The entire interaction, including the final response and any escalation action, is logged to storage for future analysis.