Real-Time LLM Chat System: Deployment Architecture Summary

1. User Interaction Layer

- **UI - Web Frontend**: The primary user interface for sending and receiving messages.
- **WebSocket Server**: Facilitates **real-time bidirectional communication** between UI and backend. The user query is sent via a WebSocket GET request, and the LLM response is returned over a WebSocket PUT/POST connection.

---

2. Backend Services

- **Session Manager**:
    - Manages user sessions and routes queries to the processing pipeline.
    - Sends the user query to Kafka Topic 1 for ingestion.
- **Pre-Processor**:
    - Performs tasks like **profanity filtering** and **request logging** (user ID, timestamp, session ID).
    - Forwards the cleaned input to the Embedder.
- **Embedder**:
    - Converts the input query into vector embeddings for semantic search.
- **Retriever**:
    - Uses **FAISS or Qdrant** to search for top-k relevant documents.
    - Performs re-ranking and returns the most relevant context to the Prompt Builder.
- **Prompt Builder**:
    - Assembles the final input for the LLM by combining the user query, session history, and retrieved documents.
    - Sends the prompt to Kafka Topic 2.

---

3. LLM Node

- **LLM Response Generator**:
    - Consumes from Kafka Topic 2.
    - Generates the AI response using an LLM model.
    - Publishes the response to Kafka Topic 3.

---

4. Kafka Messaging System

- **Kafka Cluster** (Zookeeper, Brokers, Topics):
    - **Kafka Topic 1**: Receives raw user queries.
    - **Kafka Topic 2**: Holds fully-built prompts.
    - **Kafka Topic 3**: Contains final LLM responses.
    - Ensures decoupling, scalability, and reliability of communication between services.

---

5. Vector Database

- **FAISS / Qdrant**: Used for **high-speed semantic search** on embedded queries, enabling context-aware response generation.

---

6. Monitoring & Logging

- **Logging/Monitoring Service**:
    - Records all interactions and logs for analysis, troubleshooting, and auditing.

This architecture is optimized for **real-time, scalable, and secure LLM-based chat systems** with context retention, profanity filtering, and monitoring built in.