**Architecture**

**A diagram of a hybrid rag architecture

AI-generated content may be incorrect.**

**1. Document Ingestion and Preparation**

This initial phase handles the unstructured data (like papers) and prepares it for retrieval:

1. **Paper Upload & Parsing:** A user interacts with the **Web UI** to upload a document (e.g., a "Paper"). The system parses the raw document content to extract text and potentially other relevant information.
2. **Document Chunking:** The parsed document is broken down into smaller, manageable **chunks**. This is crucial for efficient retrieval, as smaller chunks often contain more focused information.
3. **Embedding/Vectorization & Storage:**
   * The document chunks are processed for **Embedding/Vectorization**, converting the text into dense numerical vectors that capture their semantic meaning.
   * These vectors are then stored in a **Vector DB** (Vector Database), making them searchable based on their semantic similarity to a user query.
   * Simultaneously, key entities, relationships, and facts from the documents are extracted and structured into a **Knowledge Graph**. This provides a symbolic, relational representation of the data, which is useful for complex, multi-hop queries.

**2. Query Processing and Retrieval**

This phase handles the user's input and retrieves relevant context:

1. **User Query Input:** The process begins when a user submits a **User Query** via the **Web UI**.
2. **Hybrid Retrieval:** The system initiates a **dual-path retrieval** process:
   * The **User Query** is vectorized and used to search the **Vector DB** for semantically similar document chunks (traditional RAG).
   * The query is also used to traverse or query the **Knowledge Graph** to find relevant entities, relationships, and structured facts.
3. **Retrieval & Fusion Layer:** This critical layer **fuses** the results from both retrieval methods:
   * **Vector Retrieval Results:** Semantically relevant text chunks.
   * **Knowledge Graph Results:** Structured facts and relational context.
   * The fusion process combines and ranks this hybrid set of contexts to provide the richest and most accurate information possible for the LLM.

**3. Generation**

The final phase uses the retrieved context to generate a coherent answer:

1. **Context Augmentation:** The fused, relevant context from the **Retrieval & Fusion Layer** is passed to the **LLM** (Large Language Model).
2. **Response Generation:** The **LLM** generates a comprehensive and fact-grounded response by conditioning its generation on the **User Query** and the provided hybrid context.
3. **Final Output:** The final generated response is delivered back to the user via the **Web UI** to address the original **User Query**.

The **hybrid approach**, leveraging both the **Vector DB** (for semantic similarity) and the **Knowledge Graph** (for structural and relational context), significantly improves the LLM's ability to answer complex, fact-based, and relational questions.