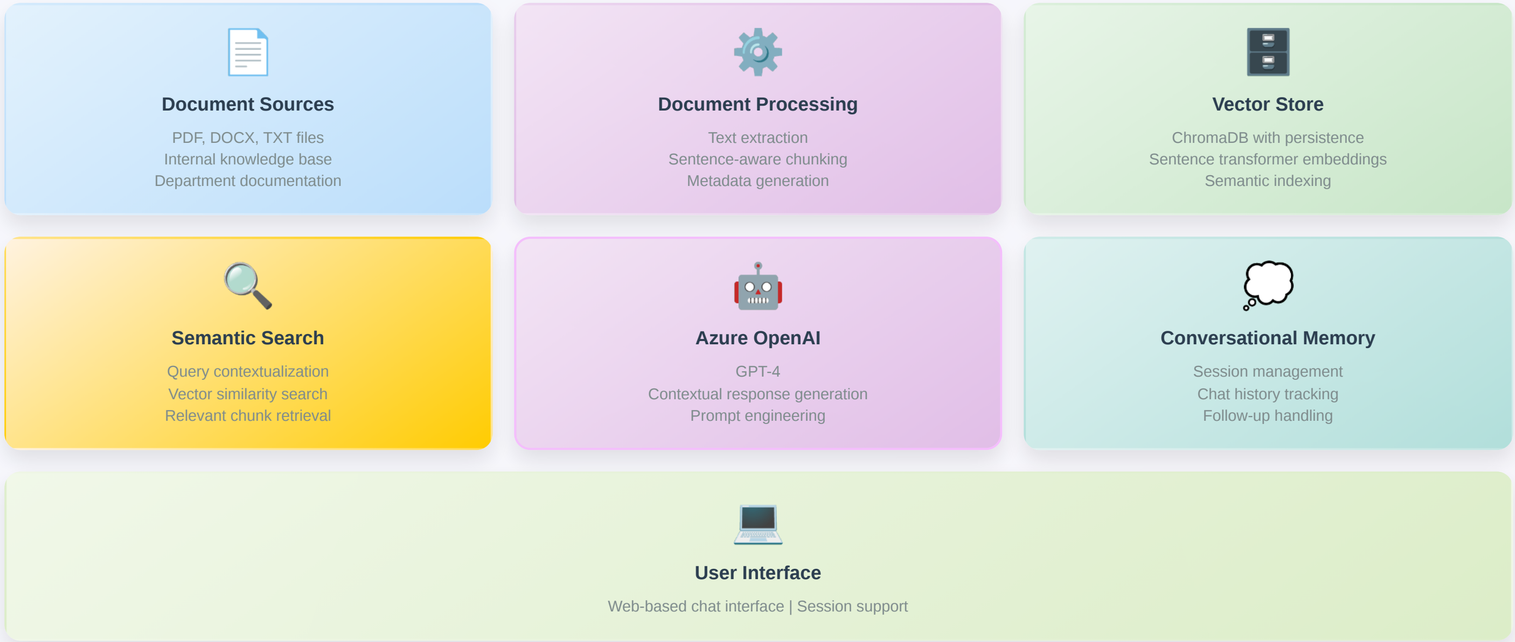
🏗️ RAG Architecture Diagram

Framework-Free RAG Implementation with Azure OpenAI



🛠️ Technology Stack

Azure OpenAI GPT-4, GPT-3.5-turbo	**ChromaDB** Vector database	**Sentence Transformers** all-MiniLM-L6-v2	**Document Processors** PyPDF2, python-docx	**Python** Core implementation

🔄 RAG Workflow Steps

1. **Document Ingestion**
   Process department documents (PDF, DOCX, TXT) and extract text content with metadata
2. **Text Chunking**
   Split documents into sentence-aware chunks (~500 chars) for optimal retrieval
3. **Embedding Generation**
   Generate vector embeddings using sentence-transformers model
4. **Vector Storage**
   Store embeddings in ChromaDB with persistent storage and metadata
5. **Query Processing**
   Contextualize user queries using conversation history
6. **Semantic Retrieval**
   Find most relevant document chunks using vector similarity
7. **Response Generation**
   Use Azure OpenAI to generate contextual responses with retrieved information
8. **Memory Update**
   Update conversation history for follow-up questions and context