Knowledge Relay

AI-based Knowledge sharing Agent

Presented by: *Team AI বদ্দা*

Md Omar Faruqe Riyad
Arpita Ghosh
Zawad Abdullah Yusuf

🧩 The "Oh No, He Left!" Problem

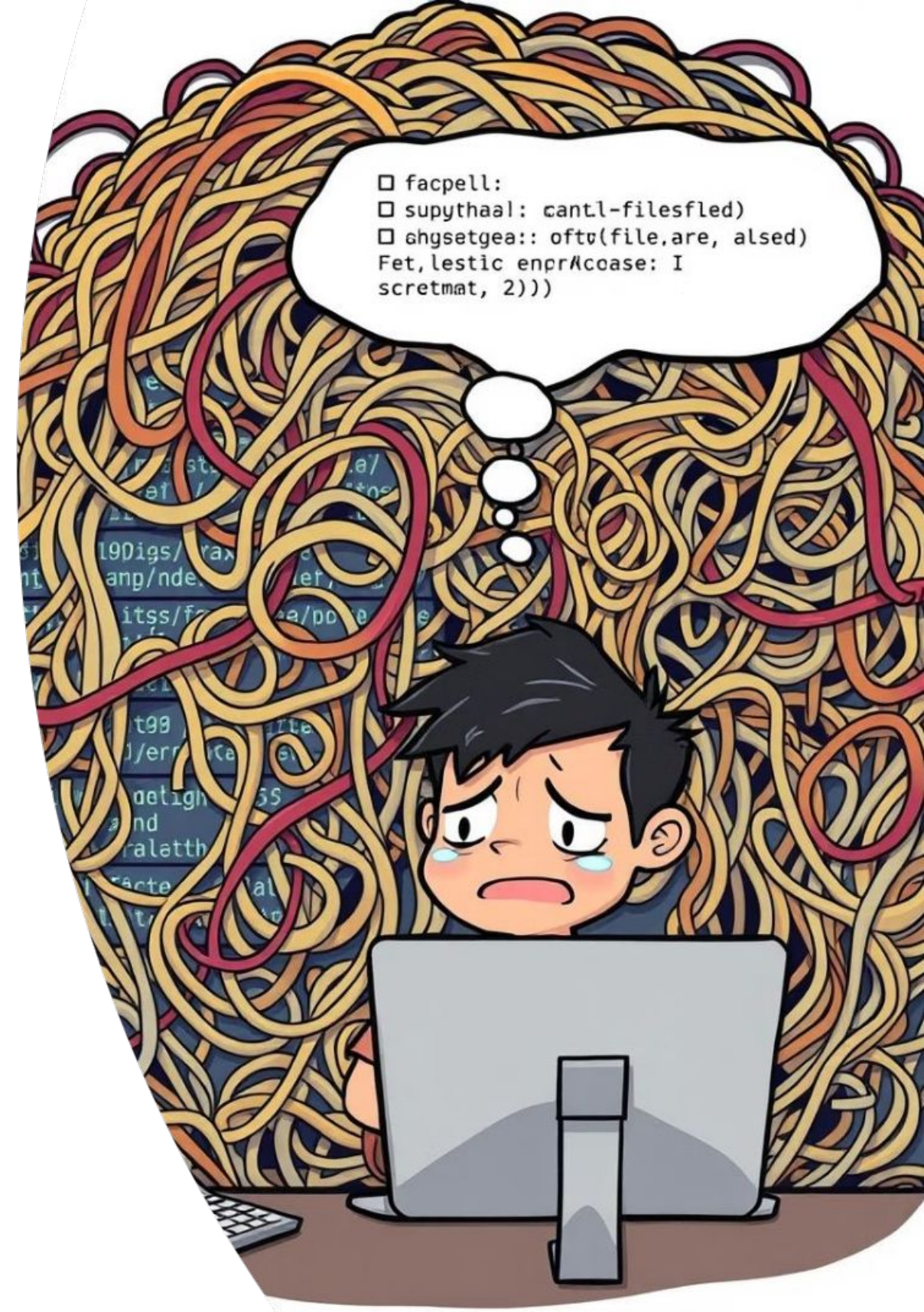**Dev A**

Holds project secrets and buried code mysteries 💀

Dev B

Arrives motivated but clueless about the codebase

Knowledge Lost

Info vanishes like a deleted mystery branch without merge

🦾 Enter: AI to the Rescue

1 **Dev A uploads knowledge**

2 **AI absorbs it all**

Not literally, but close enough 😎

3 **Dev B asks questions**

- "What's the flow?"
- "Where's the PDF generator?"
- "Why 'doNotTouchThisMethod()'?"

4 **AI acts like calm Yoda**

Answers amidst commit chaos

Made with GAMMA

Agent Process Flow

Overall Project QA

LLM generates initial questions from system prompt, adapting based on answers.

Document upload and QA
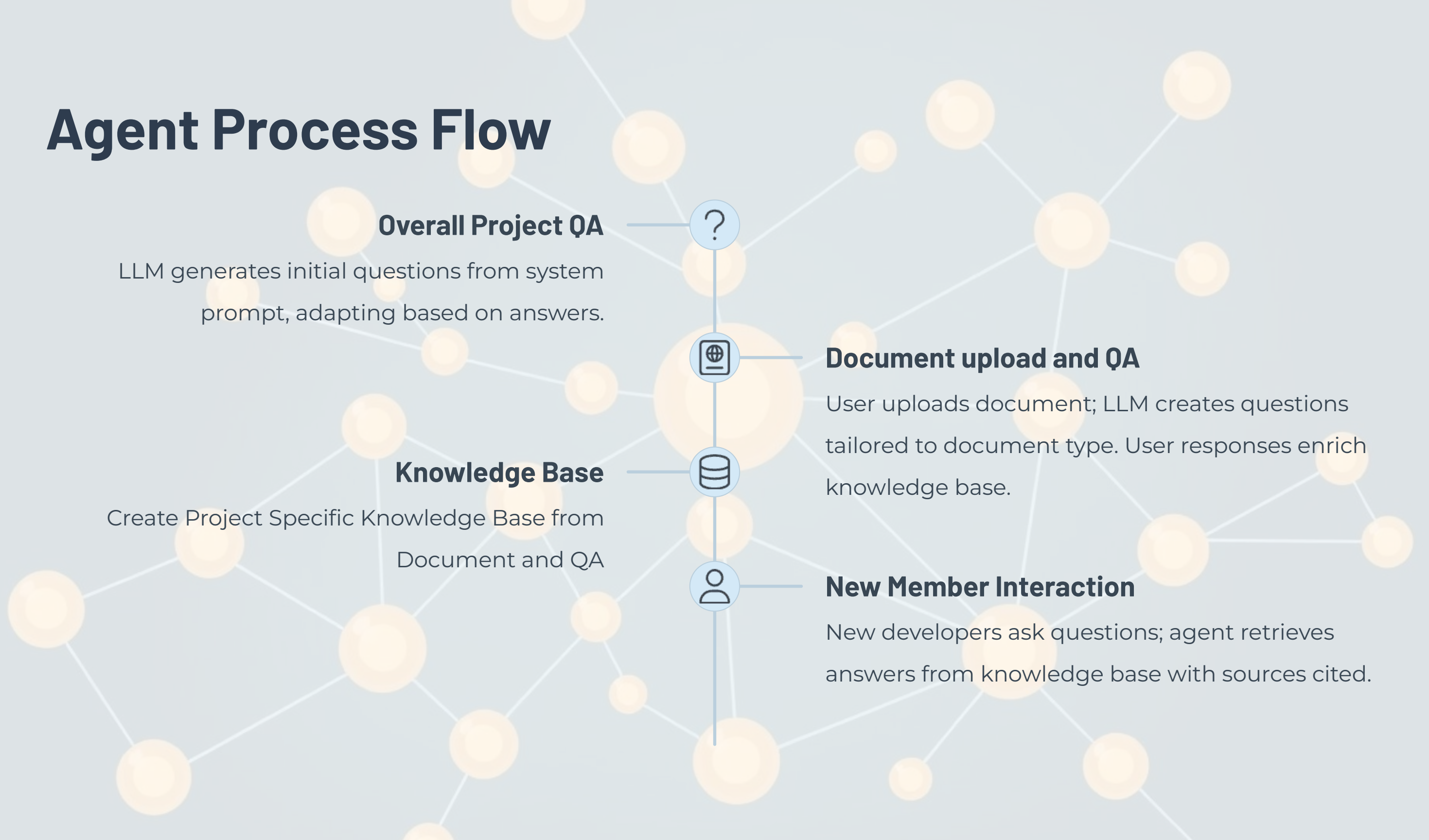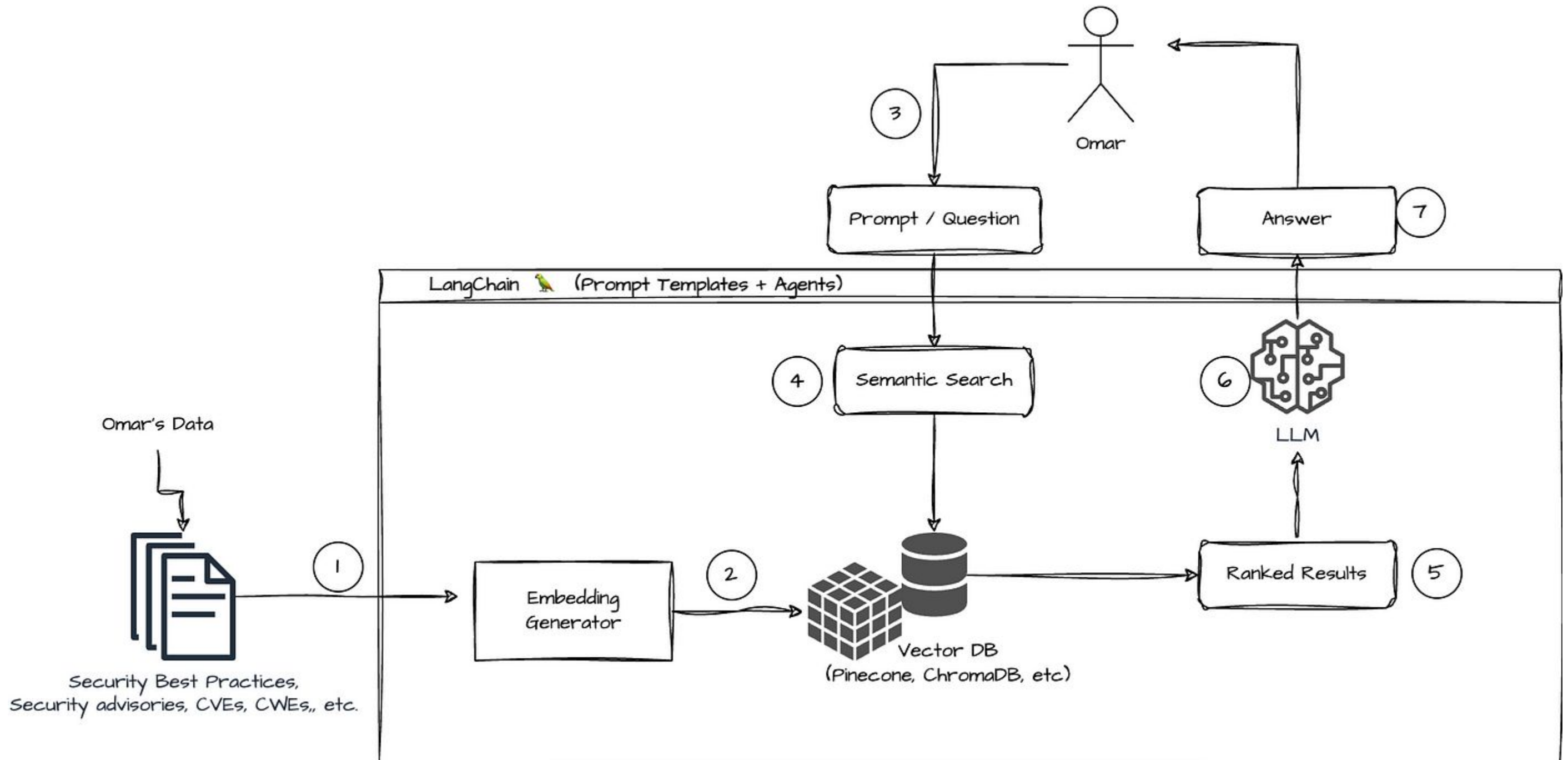
User uploads document; LLM creates questions tailored to document type. User responses enrich knowledge base.

Knowledge Base

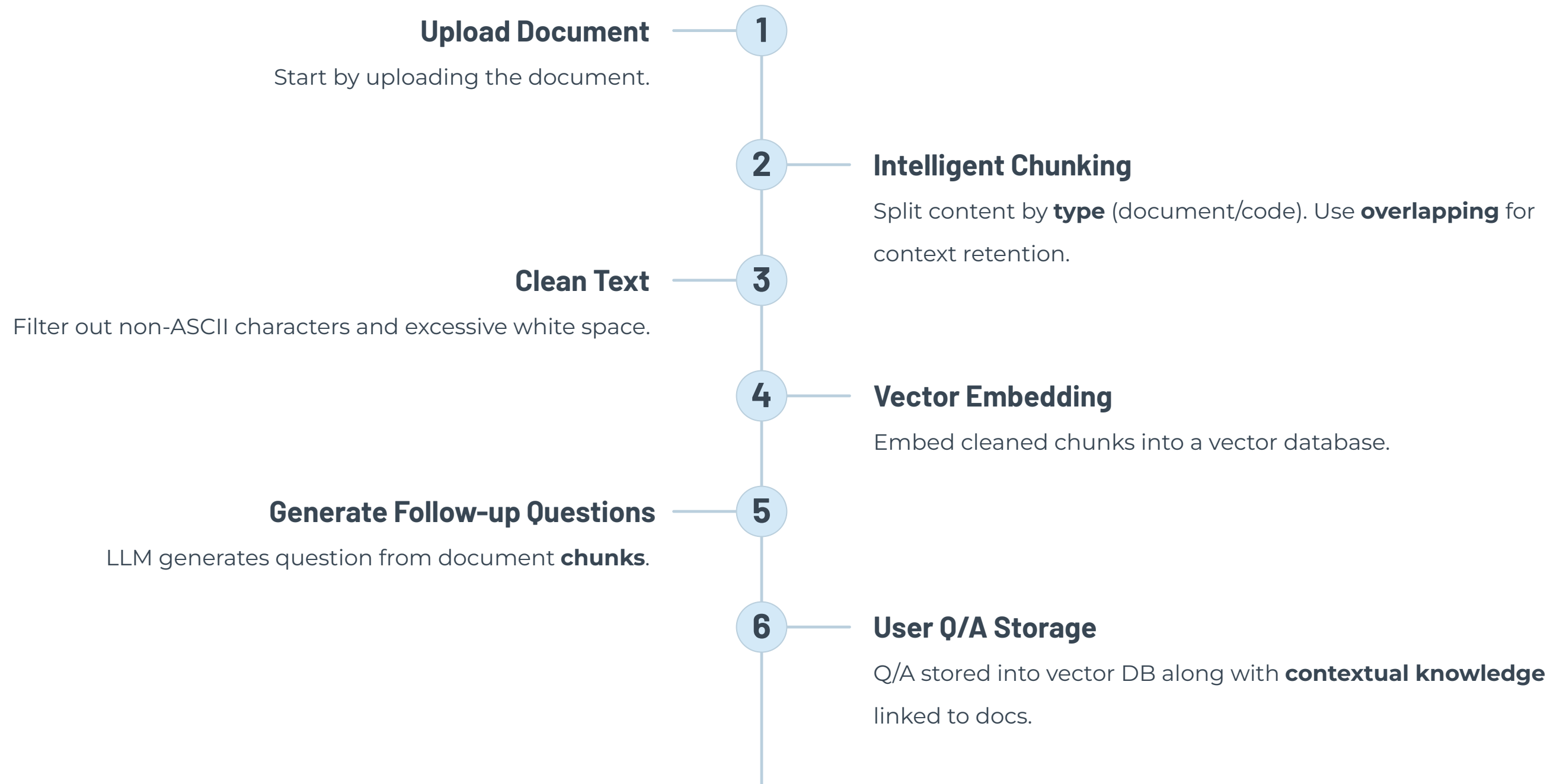Create Project Specific Knowledge Base from Document and QA

New Member Interaction

New developers ask questions; agent retrieves answers from knowledge base with sources cited.

Architecture

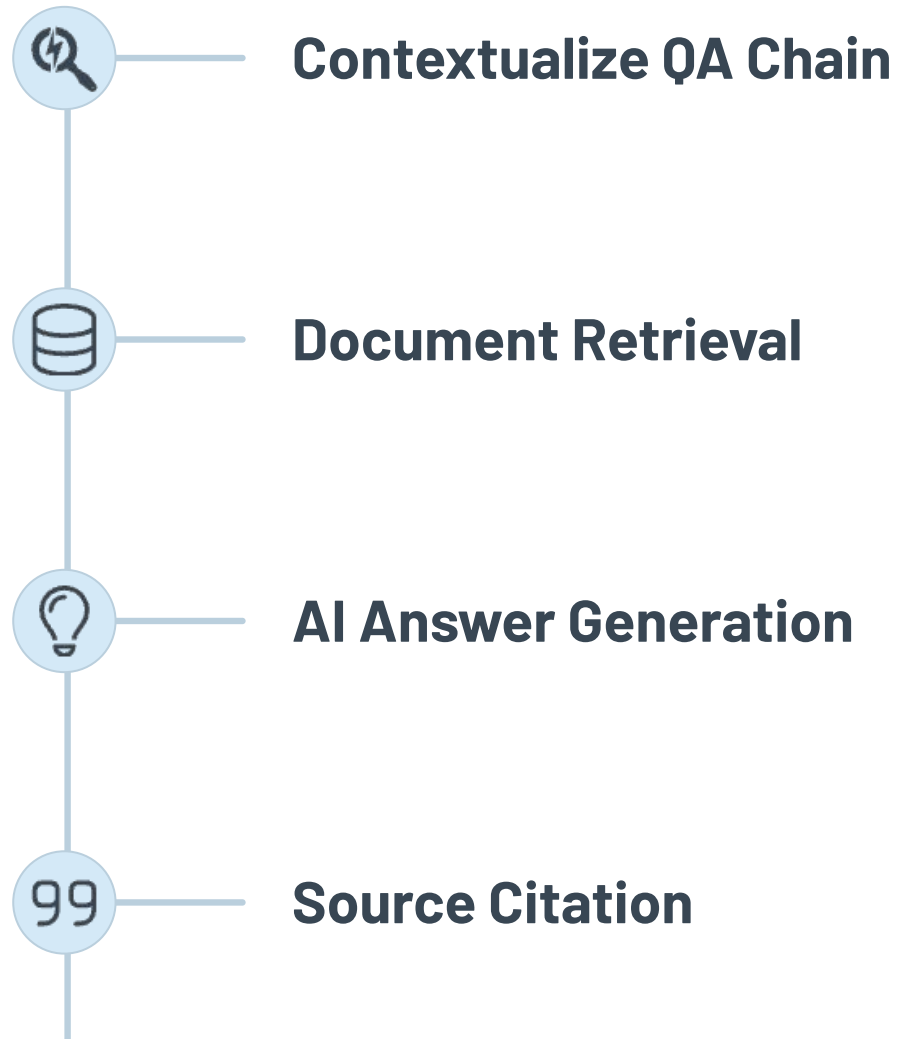# Knowledge Ingestion Flow

Upload Document — **1**

Start by uploading the document.

2 — **Intelligent Chunking**

Split content by **type** (document/code). Use **overlapping** for context retention.

Clean Text — **3**

Filter out non-ASCII characters and excessive white space.

4 — **Vector Embedding**

Embed cleaned chunks into a vector database.

Generate Follow-up Questions — **5**

LLM generates question from document **chunks**.

6 — **User Q/A Storage**

Q/A stored into vector DB along with **contextual knowledge** linked to docs.

Retrieval Flow

Contextualize QA Chain

Document Retrieval

AI Answer Generation

99 **Source Citation**

Example transformation by
Contextualized QA Chain

Original conversation:

- User: "Tell me about the authentication system"
- AI: "The system uses JWT tokens with OAuth2..."
- User: "How do we handle token expiration?"

What contextualize_q_chain does:

- Input: "How do we handle token expiration?" + chat history
- Output: "How does the authentication system handle JWT token expiration?"

💻 What's Powering the Magic?

RAG Architecture

Combines retrieval with generation

LangChain

Context-aware answer orchestration

ChromaDB

Stores developer wisdom as vectors

OpenRouter

Flexible AI model access

FastAPI backend with React chat UI for users

🚀 Future Features

- Git, Jira, Openproject integration

- Embed auto-generated documentation of code for better semantic search

Share Your Feedback