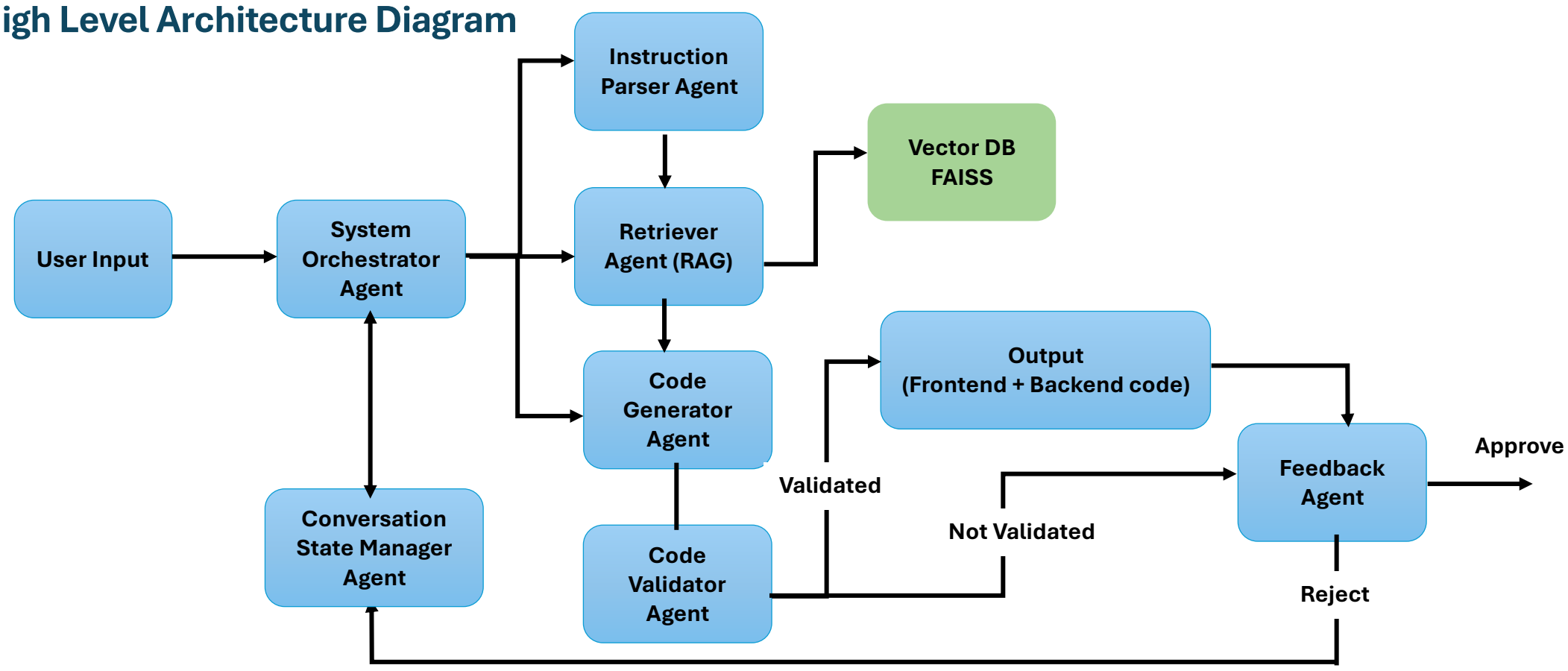
Agentic AI System

High Level Architecture Diagram



Complete flow and summary of functionalities of each agent.

About System Orchestrator Agent (Main Controller)

- It reads prompt and decides:
  - What needs to be done?
  - Which agents should work on what part?
  - In what order?
- Sends prompt to the Instruction Parser Agent.

About Instruction Parser Agent

- Breaks request into small tasks.
- These smaller tasks are sent back to the System Orchestrator.

About Retriever Agent (RAG - Retrieval-Augmented Generation)

- System Orchestrator will send these tasks to the RAG and RAG will search for relevant APIs from the documentation CSV file.
- Used **BGE-small-en** (**Bilingual General Embedding**) model to create embeddings of both API docs and user query. (Embeddings of 384 dimensions)
- Used **FAISS** to store the embeddings and separate dictionary to store the API metadata
- Sent the RAG output back to System Orchestrator

About Code Generator Agent

- Combined the outputs from both Instruction Parser agent and RAG agent
- Extracted the API documentation parameters from the web page using document URL of the API provided using **Playwright and Beautifulsoup.**
- Ask the user about the dev stack they want
- Construct a prompt including all the required details and send it to LLM.
- Used GPT-4 to generate the code.

About Code Validator Agent

- Code generator agent will pass the code for validation.
- This agent will check if the code is correct by Syntax Checking, Linting, API call validation and security checks
- If code is validated, then it will be sent to user.
- If code is not validated, then it will be sent to feedback agent where it will prefill the response of that session and user can also add their feedback to the same response.
- This response now will be sent to Conversation state manager agent store validation results + feedback as a part of session history.
- Then to orchestrator, based on this response it will trigger the agents again and construct the new prompt

About Feedback Agent

- Once the code is generated user will be asked to approve or reject and if they have any feedback.
- Capture their feedback into a proper structure and send it to state manager agent
- State manager agent will store the session history of responses + feedbacks and keep the regeneration count of that session
- That response will be sent back to system orchestrator and orchestrator will take actions based on response. (trigger agents if necessary and construct prompt again and pass it to code generator agent)

About Conversation State Manager Agent

- New user query = Session Initialization in System Orchestrator and new record in state manager
- Receive O/Ps from all agents to keep track of the history
- Basically, it stores chat history, feedback, validation results, context management

Note: Autogen and Lang chain to be integrated.