Generative AI - Conversational Agent

Context

A Generative AI based conversational agent designed to facilitate seamless question-and-answer interactions, summarization, and more within a specified knowledge base. This innovative tool operates with intuitive functionality, enabling users to engage in collaborative exploration and retrieval of information in a natural conversational manner.

Features:

- An agnostic solution covering Databricks, Qdrant vector database and cloud managed services
- Can be deployed into any cloud environment using Terraform scripts(bundled with the product)
- Scalable, robust, flexible and secured solution
- Can evolve and industrialize, easily tailored into a bespoke solution with a minimal coding changes
- Domain independent and adaptable for use across various domains

Deliverables include :
- Design of the demonstration solution, including solution architecture and frontend mock-up
- A mvp deployed in Databricks and Azure
- Detailed documentation, including design of the storyline for each use case

Product Evolution

1. Demonstrate POV

Description

Demonstrate whether the current solution is feasible with Client's requirement.

Demonstrate as it is solution:

- Demonstrate how users can submit a question about a set of text documents and receive an AI-generated short answer
- Demonstrate the high level and low-level solution design
- The solution is agnostic and can be deployed in any cloud with a minimal development effort
- Currently the solution is hosted in Databricks and Azure, the same solution also hosted in a single cloud

Outcome

- **Based on Client's interest a POC can be conducted**

2. Perform POC

- Conduct **proof of concept by**
- define the scope
- Identify the environment
- Access data sources (this can be anonymous if performed on Vendor's environment)
- Define a timeline and expected outcome
- **Perform code changes based on the defined scope and measure the solution if this is working as expected**

- **Based on the POC outcome further evolution can be planned by industrializing the solution**

3. Industrialize and Scale-up

- Industrialize the solution by make it client specific
- Build and unit testing of the end-to-end solution (with new changes) in Client's environment
- Maintain security and network standards
- Deployment to production environment
- setting up governance and maintenance processes, adoption & change management strategy
- identify KPI/metrics for adoption and value generated and build dashboard/report, setup change advisory board
- Training and documentation

- **MVP in production**
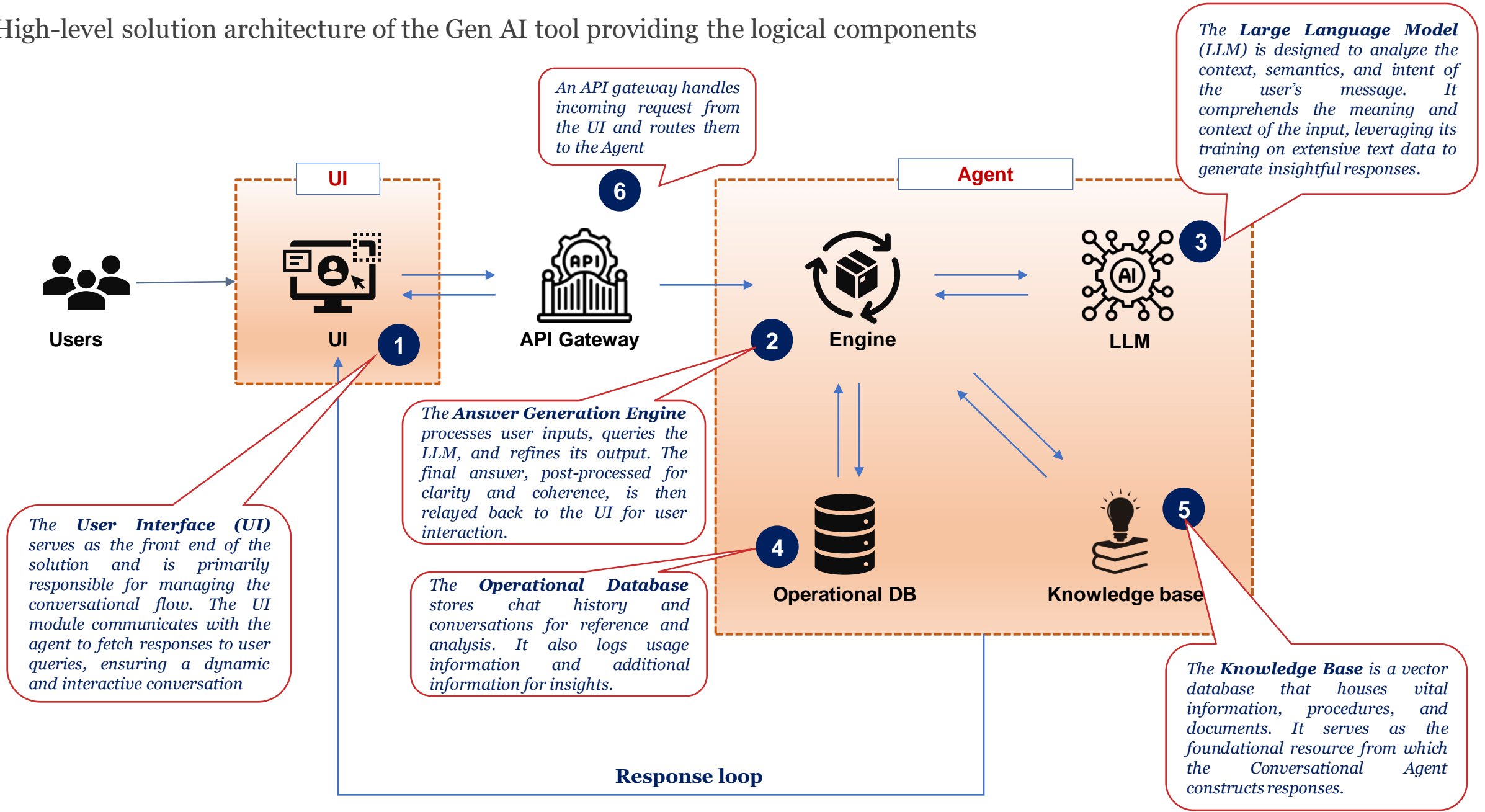- **Collect additional requirements/change Request**
- **Continuous evolution and scale-up**

High level solution architecture

High-level solution architecture of the Gen AI tool providing the logical components

An API gateway handles incoming request from the UI and routes them to the Agent (6)

The Large Language Model (LLM) is designed to analyze the context, semantics, and intent of the user's message. It comprehends the meaning and context of the input, leveraging its training on extensive text data to generate insightful responses. (3)

UI

Agent

Users

UI (1)

API Gateway

Engine (2)

LLM

The **Answer Generation Engine** processes user inputs, queries the LLM, and refines its output. The final answer, post-processed for clarity and coherence, is then relayed back to the UI for user interaction.

The **User Interface (UI)** serves as the front end of the solution and is primarily responsible for managing the conversational flow. The UI module communicates with the agent to fetch responses to user queries, ensuring a dynamic and interactive conversation

The **Operational Database** stores chat history and conversations for reference and analysis. It also logs usage information and additional information for insights.

Operational DB (4)

Knowledge base (5)

The **Knowledge Base** is a vector database that houses vital information, procedures, and documents. It serves as the foundational resource from which the Conversational Agent constructs responses.

Response loop

Low level solution design (1/2)

A detailed breakdown of the current architecture hosted in Databricks and Azure
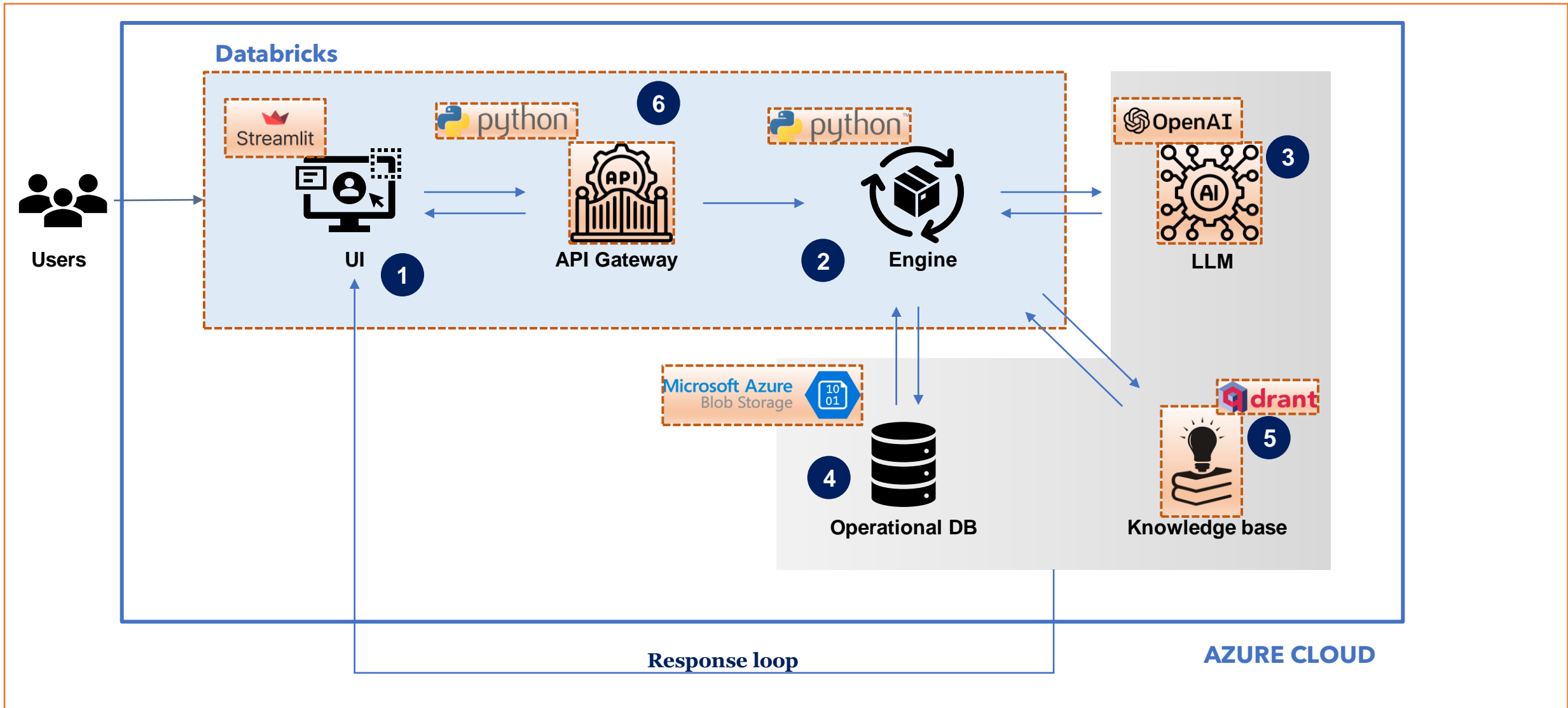
Low level solution design

A detailed description of the technology used

Id	Component	Description	Technology	Comments
1	**User interface**	Manages conversational flow and user interaction	Build a custom user interface hosted in Databricks using Streamlit/react.js	
2	**Answer generator engine**	Processes user inputs and generates responses	Build a custom Python script hosted in Databricks.	
3	**LLM**	Analyses user's message context and intent	External AI service managed by the cloud service provider	AWS bedrock/Azure Open AI/ Google Gen AI Studio
4	**Operational Database**	Stores chat history and logs usage information	No-sql database managed by cloud service provider	Dynamo DB/ Cosmos DB/ Bigtable
5	**Knowledge Base**	Vector database used to construct responses	Qdrant database hosted on Cloud environment.	A Quadrant database hosted on Kubernetes engine
6	**API Gateway**	Processes and routes requests to appropriate agents	Build a custom Python script hosted in Databricks notebooks	

Thank You

Pragyan Banerjee

LinkedIn

pragyan.banerjee@gmail.com