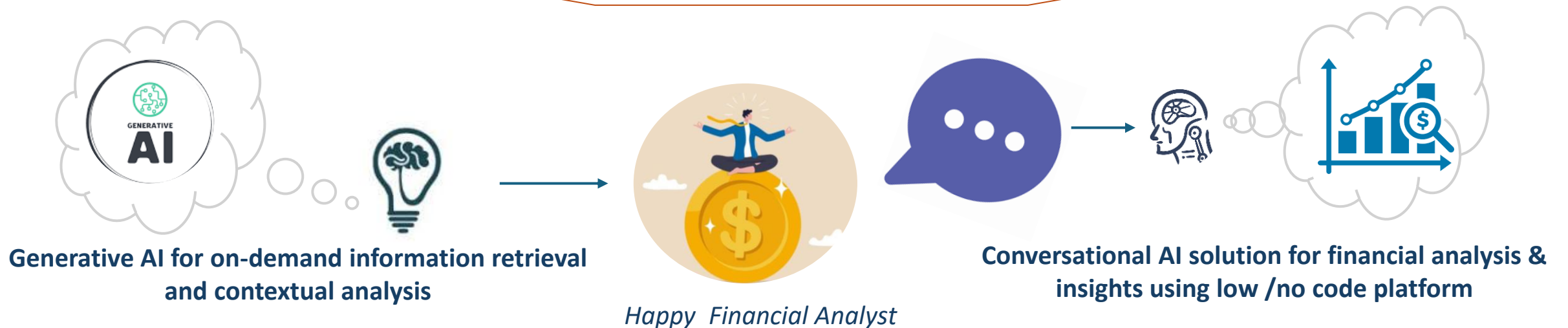# Competitive Intel Agent

- **Conversational AI Co-Pilot Solution**

Competitive Intel Agent:

Analysis and deriving data-driven insights from complex financial artefacts is effort intensive, time consuming and error prone.

Organic and Inorganic Growth

Trend and Ratio Analysis

Competitive Intelligence

Multiple Artifacts with different data types

Overwhelmed Financial Analyst

Competitive Intel Agent

Generative AI for on-demand information retrieval and contextual analysis

Happy Financial Analyst

Conversational AI solution for financial analysis & insights using low /no code platform
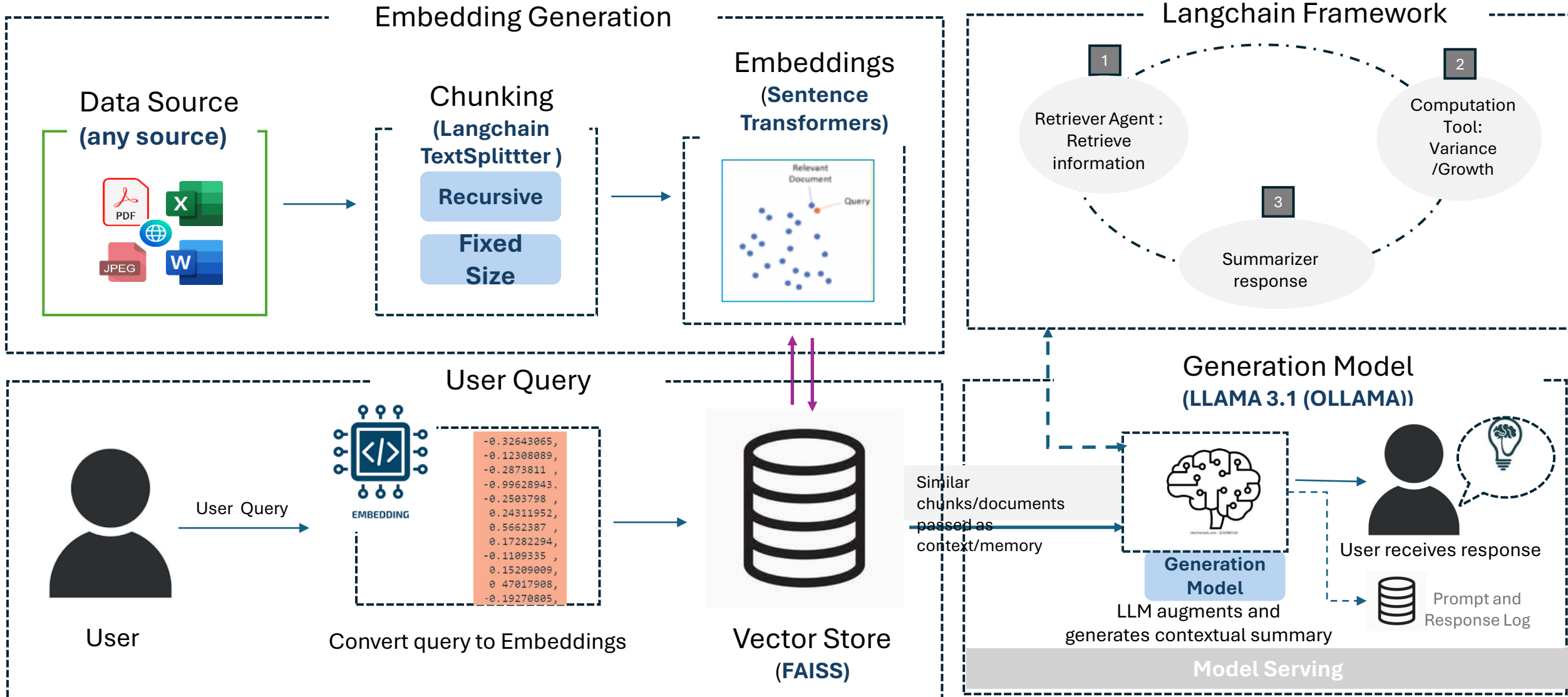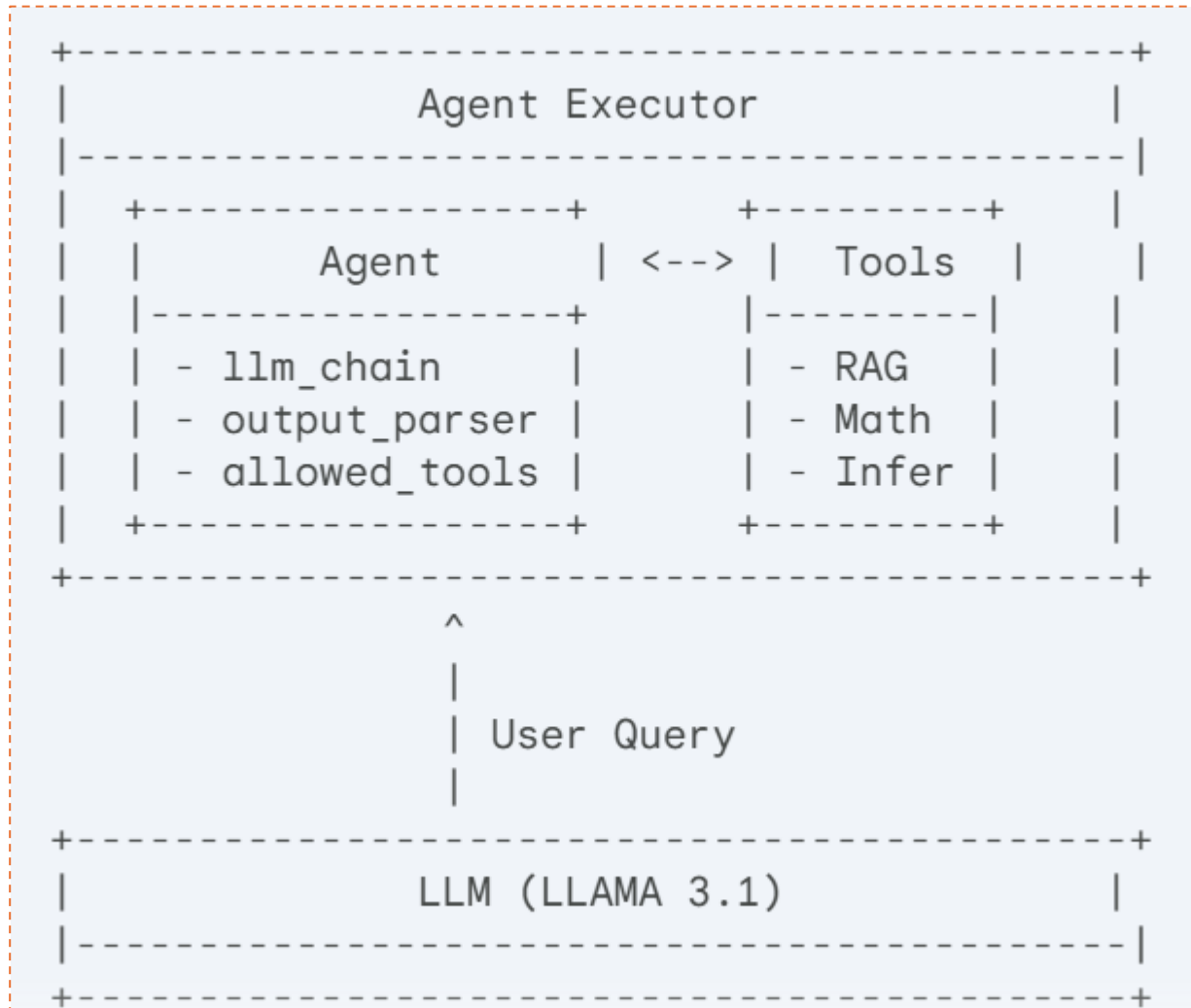
Solution Architecture (Backend)

Retrieval Augmented Generation architecture with recursive chunking and multi-modal Generative AI models

Embedding Generation

Data Source (any source)

Chunking (Langchain TextSplittter)
- Recursive
- Fixed Size

Embeddings (Sentence Transformers)

Relevant Document
Query

Langchain Framework

1 Retriever Agent : Retrieve information

2 Computation Tool: Variance /Growth

3 Summarizer response

User Query

User

User Query

EMBEDDING

-0.32643065,
-0.12308089,
-0.2873811 ,
-0.99628943,
-0.2503798 ,
0.24311952,
0.5662387 ,
0.17282294,
-0.1109335 ,
0.15209009,
0 47017908,
-0.19270805,

Convert query to Embeddings

Vector Store (FAISS)

Similar chunks/documents passed as context/memory

Generation Model (LLAMA 3.1 (OLLAMA))

Generation Model

LLM augments and generates contextual summary

User receives response

Prompt and Response Log

Model Serving

Flow of Information

```
+-----------------------------------------------------+
|                  Agent Executor                     |
|-----------------------------------------------------|
|   +-------------------+       +-----------+          |
|   |       Agent       | <--> |   Tools    |          |
|   |-------------------+       |-----------|          |
|   | - llm_chain       |       | - RAG     |          |
|   | - output_parser   |       | - Math    |          |
|   | - allowed_tools   |       | - Infer   |          |
|   +-------------------+       +-----------+          |
+-----------------------------------------------------+
                      ^
                      |
                      | User Query
                      |
+-----------------------------------------------------+
|                  LLM (LLAMA 3.1)                    |
|-----------------------------------------------------|
+-----------------------------------------------------+
```

1. The user provides a query to the Agent Executor.

2. The Agent Executor passes the query to the Agent.

3. The Agent uses the LLM to understand the query and determine the best course of action.

4. The Agent selects the appropriate tool and sends the necessary information to the tool.

5. The tool performs its task and returns the result to the Agent.

6. The Agent may use the LLM to further process the result or to decide on the next action.

7. Steps 4-6 may be repeated until the Agent has enough information to generate a final answer.

8. The Agent sends the final answer to the Agent Executor.

9. The Agent Executor presents the answer to the user.