Proprietary + Confidential

06

Retrieval Augmented Generation (RAG)

The information in this presentation is classified:

Google confidential & proprietary

⚠️ This presentation is shared with you under NDA.

- Do **not** record or take screenshots of this presentation.
- Do **not** share or otherwise distribute the information in this presentation with anyone **inside** or **outside** of your organization.

Thank you!

Proprietary + Confidential

In this module, you learn to ...

01 Architect RAG solutions for real-world customer problems

02 Choose the right embedding technology for creation, storage and serving

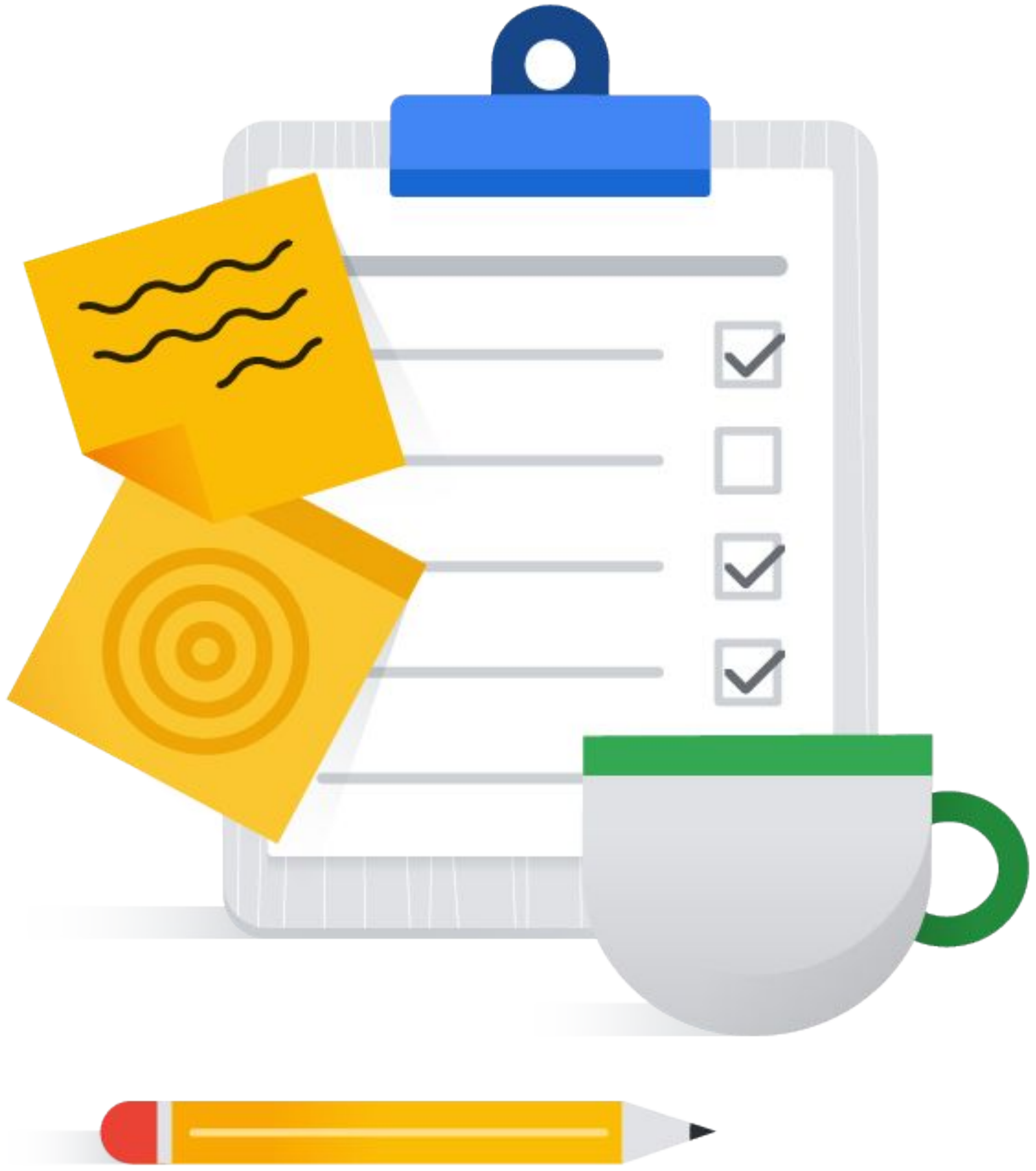
03 Optimize workflows and RAG solutions

Proprietary + Confidential

Topics

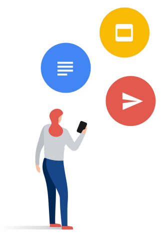
01 Retrieval Augmented Generation

02 RAG Optimization



Proprietary + Confidential

Customer problem



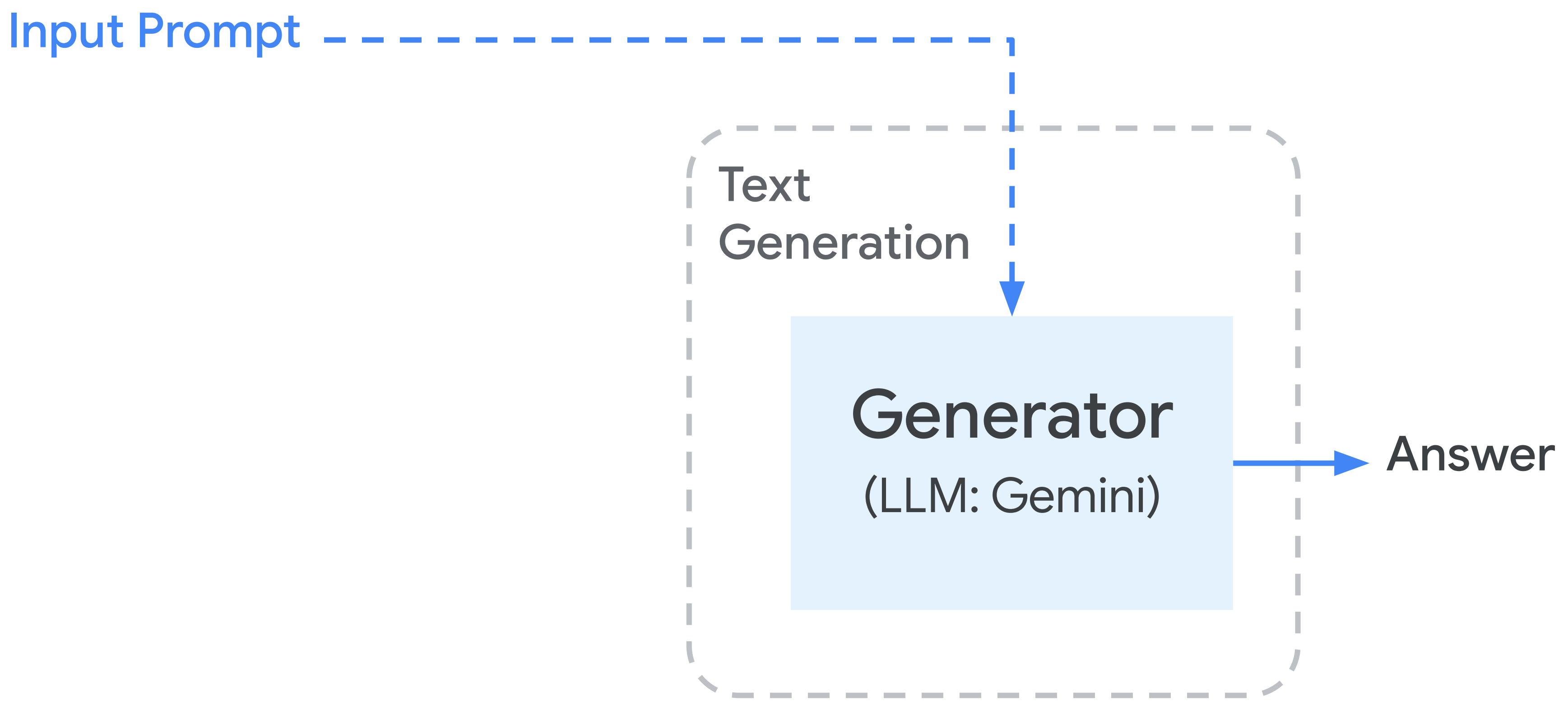
Provide semantic search to find answers from a proprietary dataset and obtain responses with summarizations grounded on the data

- Search answers should not paraphrase the stored data; they should answer the question
- Search answers should have a pointer to the document where the answer was found to be able to verify that answers are grounded on customer data
- Answers should not be made up if the data is not found in the search results
- Dataset is proprietary to the customer and it needs to stay private
- Dataset is constantly growing (it’s not stale)

What is Retrieval Augmented Generation (RAG)?

The problem:

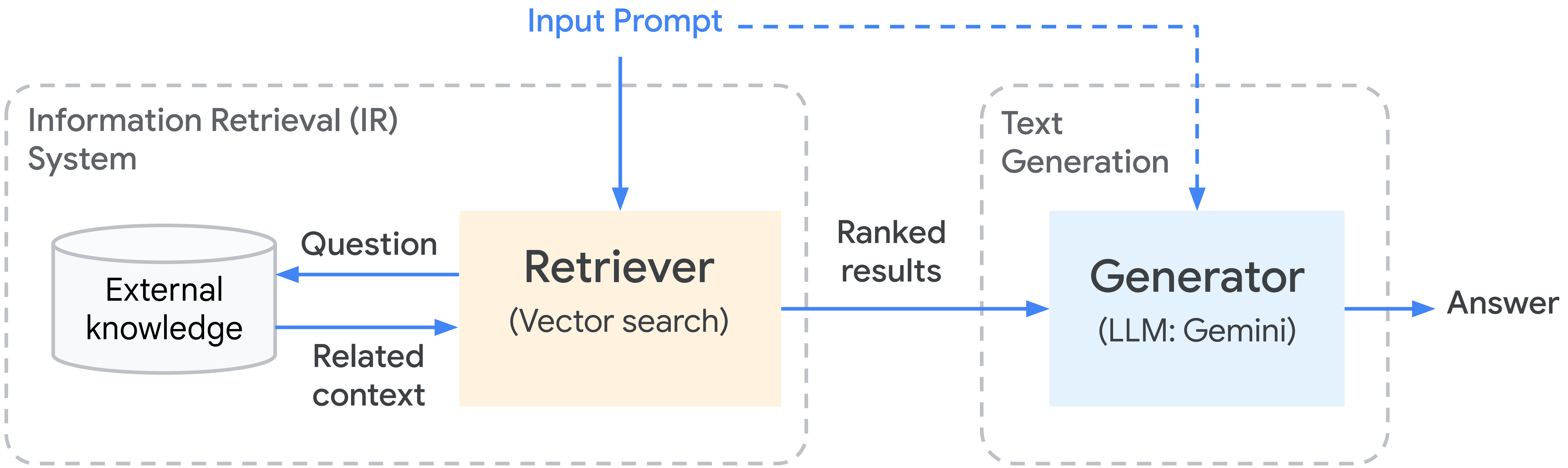
- LLMs don't know your business proprietary or domain specific data
- LLMs don't have real-time information
- LLMs find it hard to provide accurate citation



What is Retrieval Augmented Generation (RAG)?

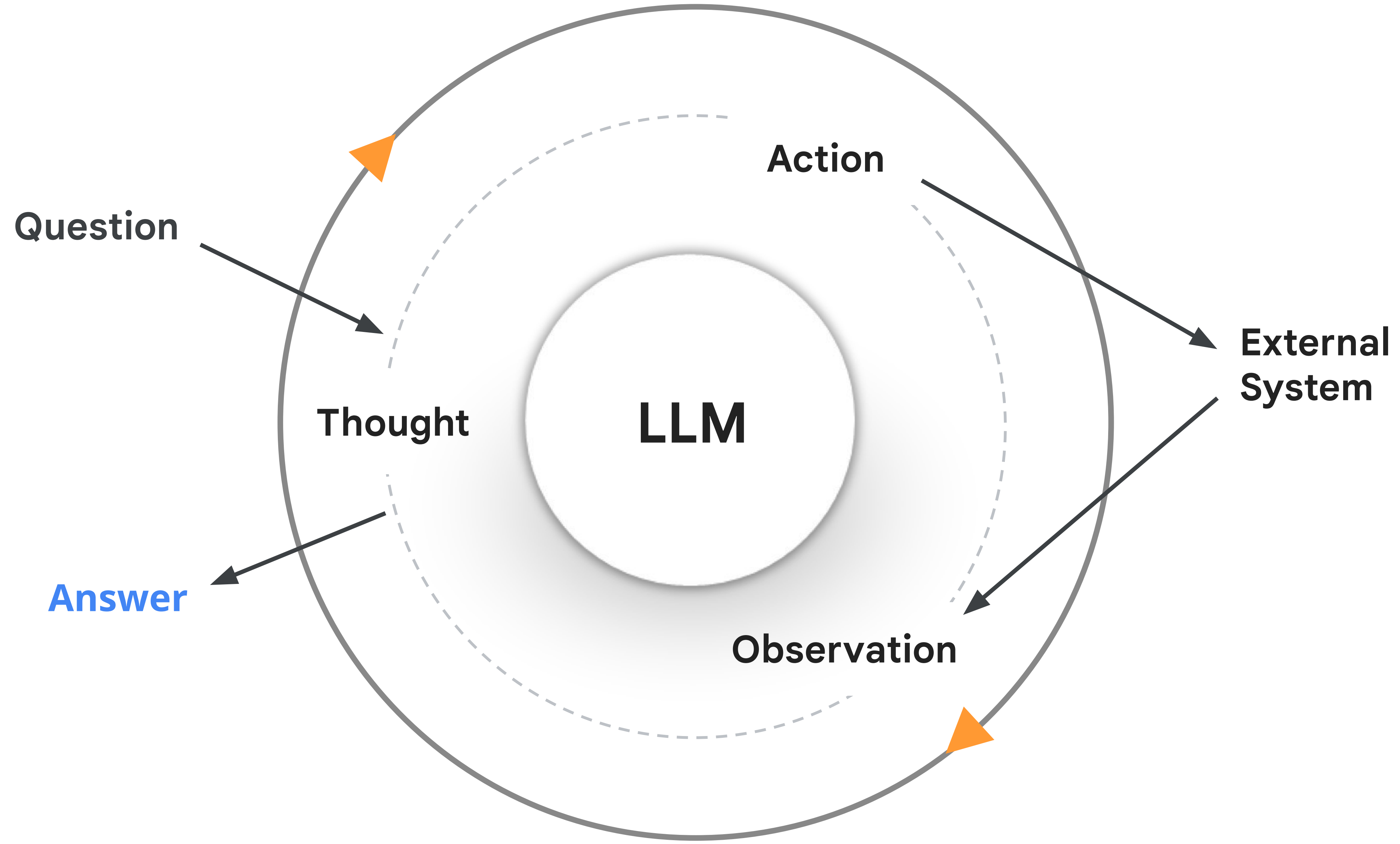
The solution:

- Feed the LLM relevant context in real-time, by using an information retrieval system

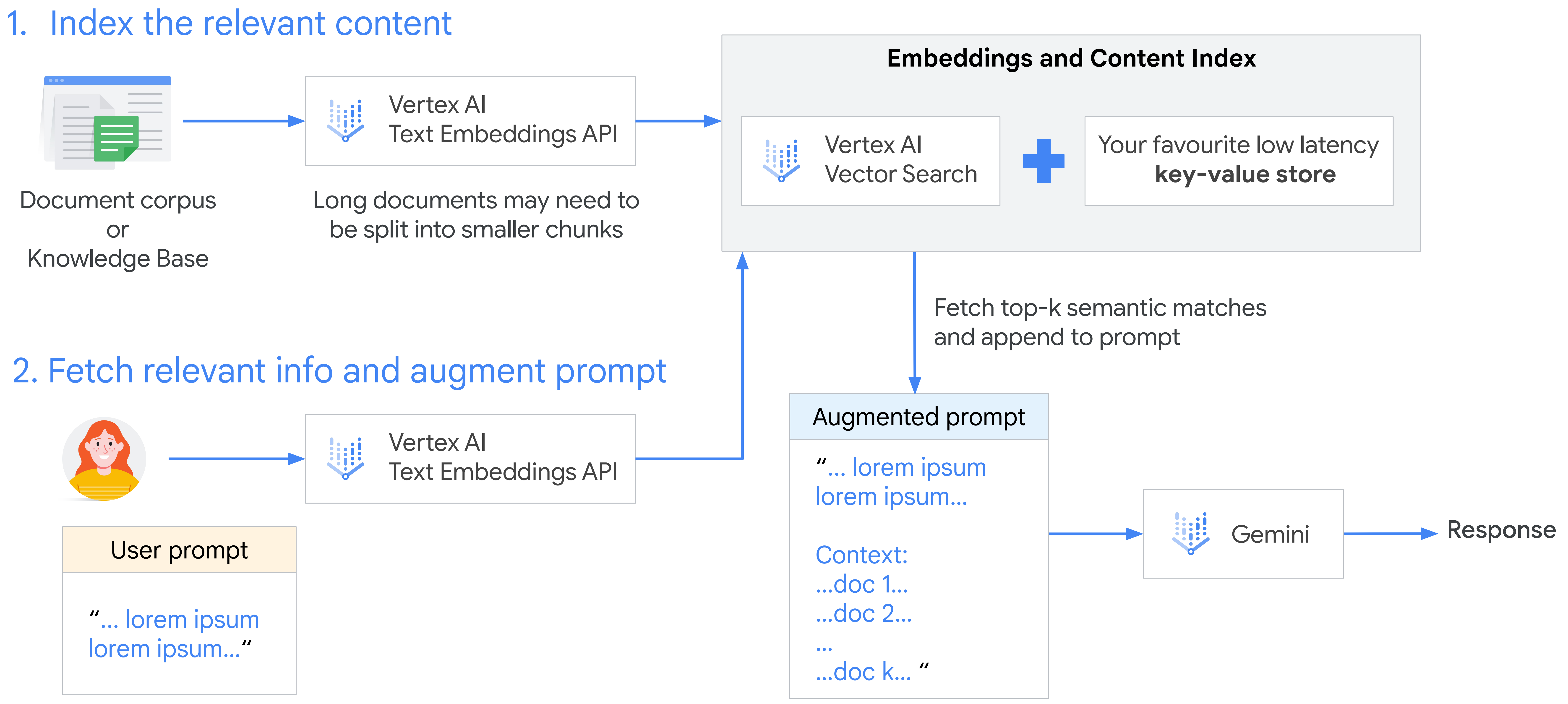


RAG is an implementation of the ReAct pattern

- A repository of data external to the LLM
- Embeddings and vector search to find the data relevant to a user query
- The LLM answers the question based on the data provided

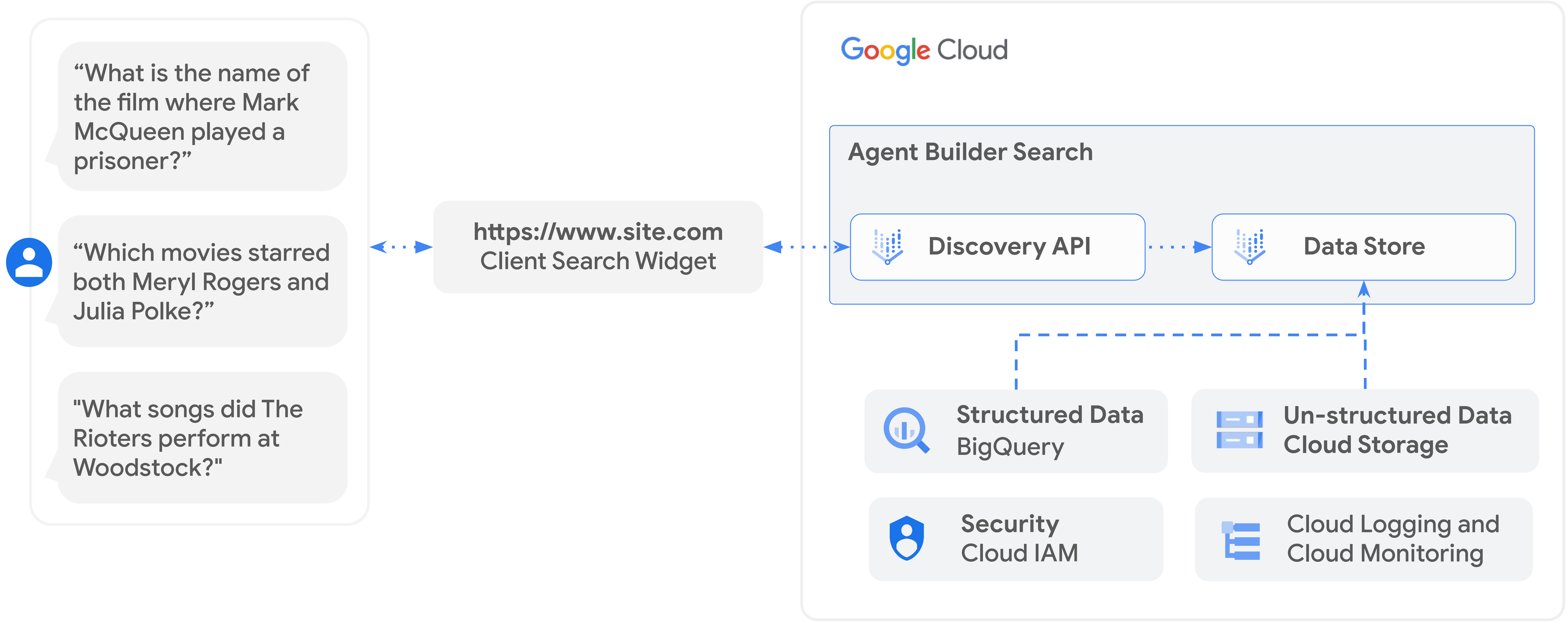


RAG architecture example: embeddings and Vector Search



Proprietary + Confidential

Automate RAG with Google Agent Builder Search Apps



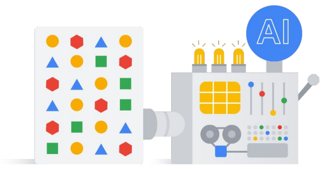
Google Cloud

Discussion out-of-the-box (OoTB) vs do-it-yourself (DIY)

Out-of-the-box Agent Builder Search	**Do-it-yourself** Embeddings + Vector Search + Document AI
Implementation in minutes	Implementation in hours or days
Only batch data refresh available	Batch and streaming data refresh available
Supports: ● BigQuery tables, HTML, PDF with embedded text, TXT format ● Preview: PPTX and DOCX Does not support: Images, videos, audio	Can be used with any data format Parsing documents with Document AI before ingesting them as embeddings provides better outcomes than OOTB
Prompt templates in Preview	Can create a prompt template and send it to the LLM

Proprietary + Confidential

Benefits of RAG vs Fine Tuning



RAG

- You can use different versions of the LLM with the same knowledge base, without the need to re-train
- You can keep ingesting documents on-demand
- You can ground the answer in a specific known document source
- RAG is for supplying the LLM external data

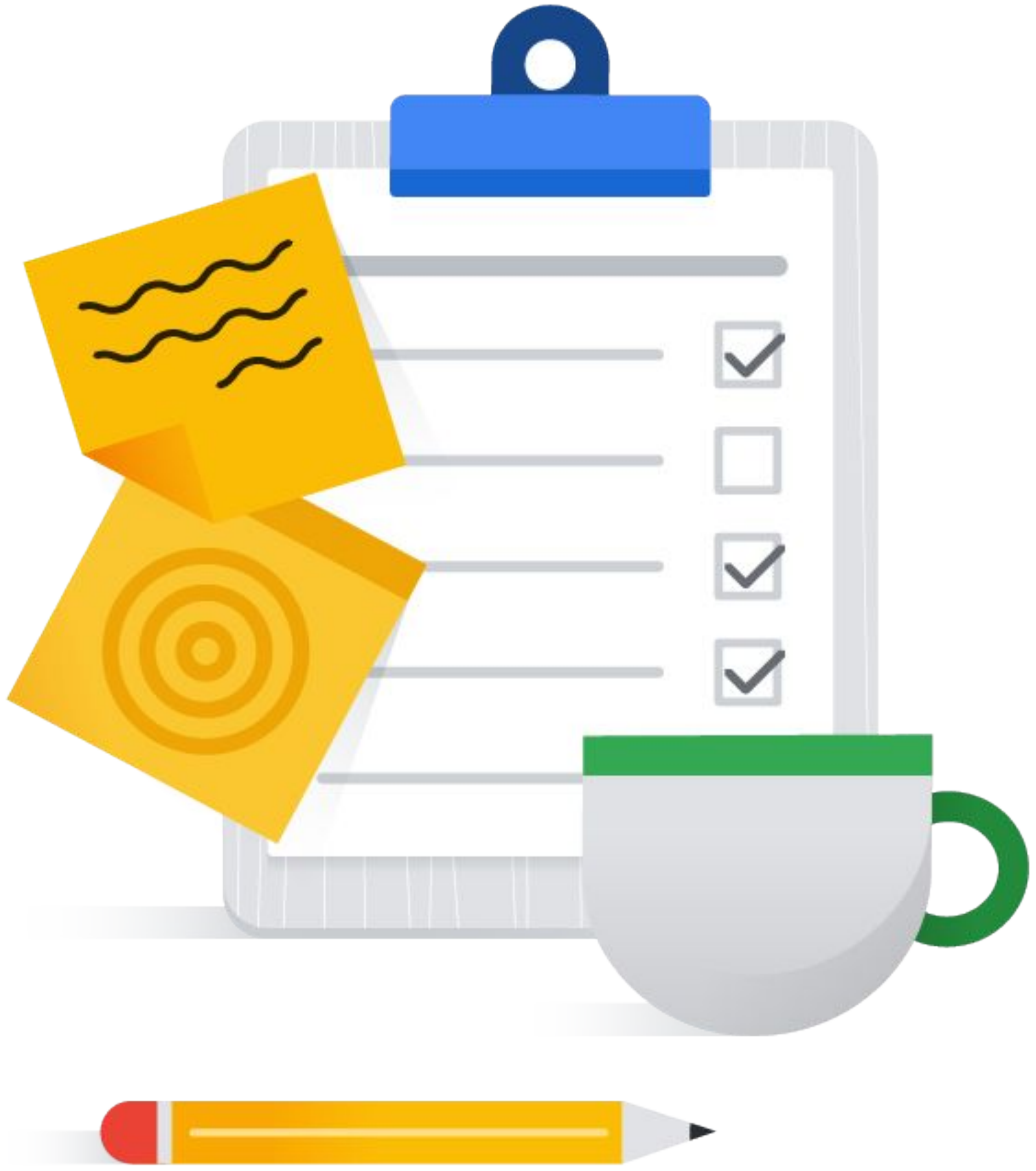
Fine Tuning

- Fine tuning an LLM and keeping it up-to-date is more expensive than creating embeddings in RAG
- Inferencing in bigger fine-tuned LLMs can be more costly (sometimes) than an LLM with the context provided by the embedding
- Fine tuning is used to show the model how to format its answers

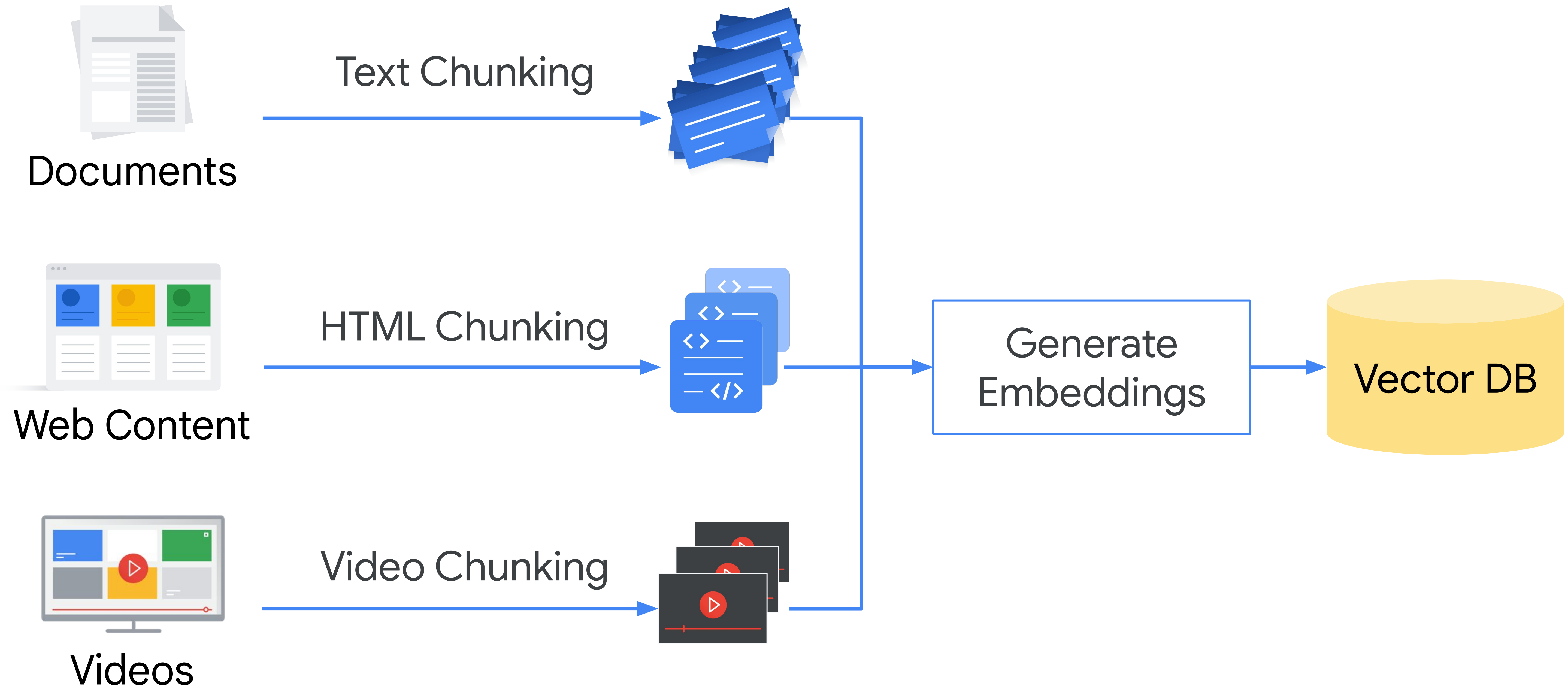
Proprietary + Confidential

Topics

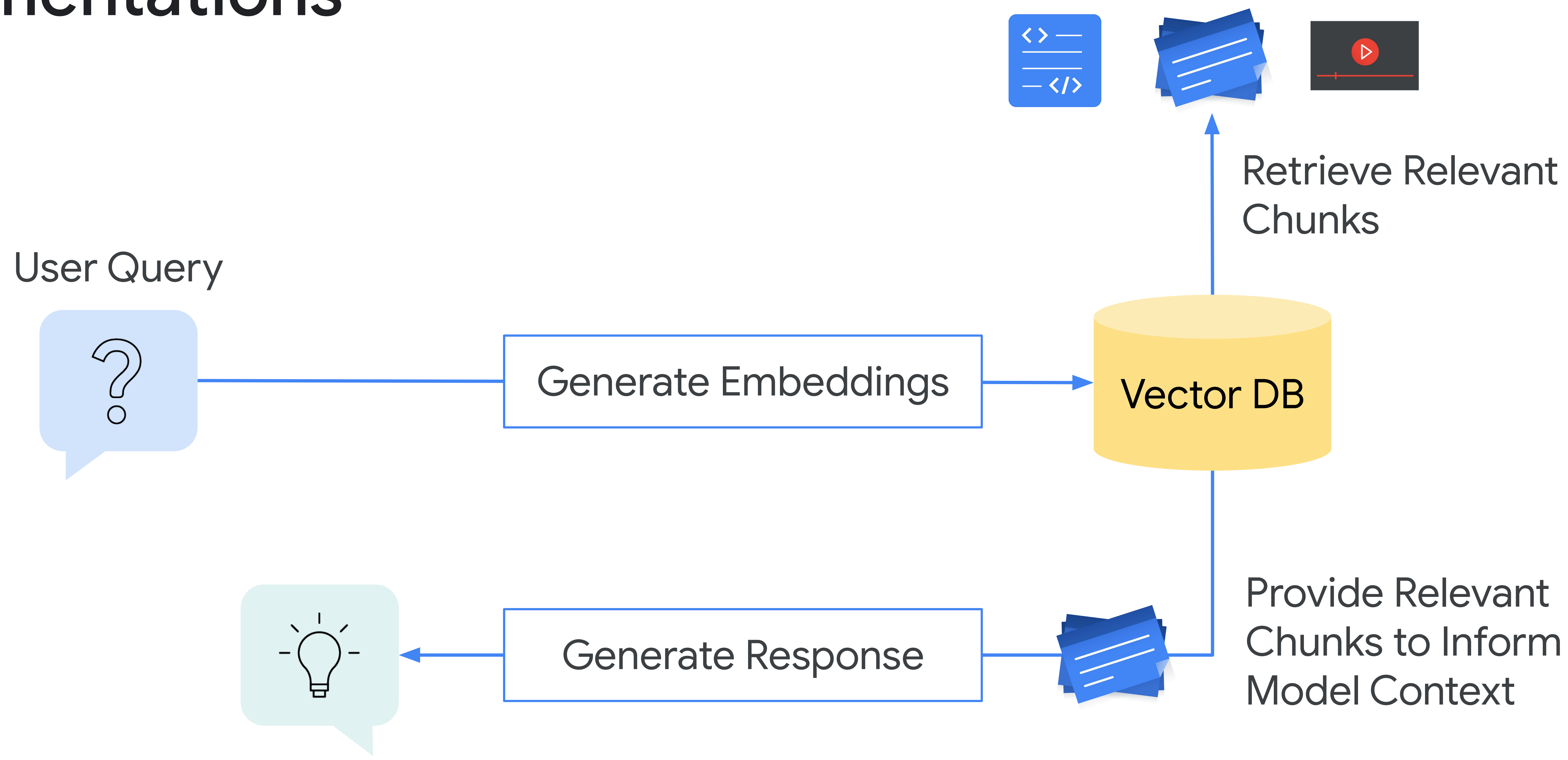
01	Retrieval Augmented Generation
02	RAG Optimization



You can embed content from many types of source documents



Based on this basic RAG framework, let's consider some augmentations



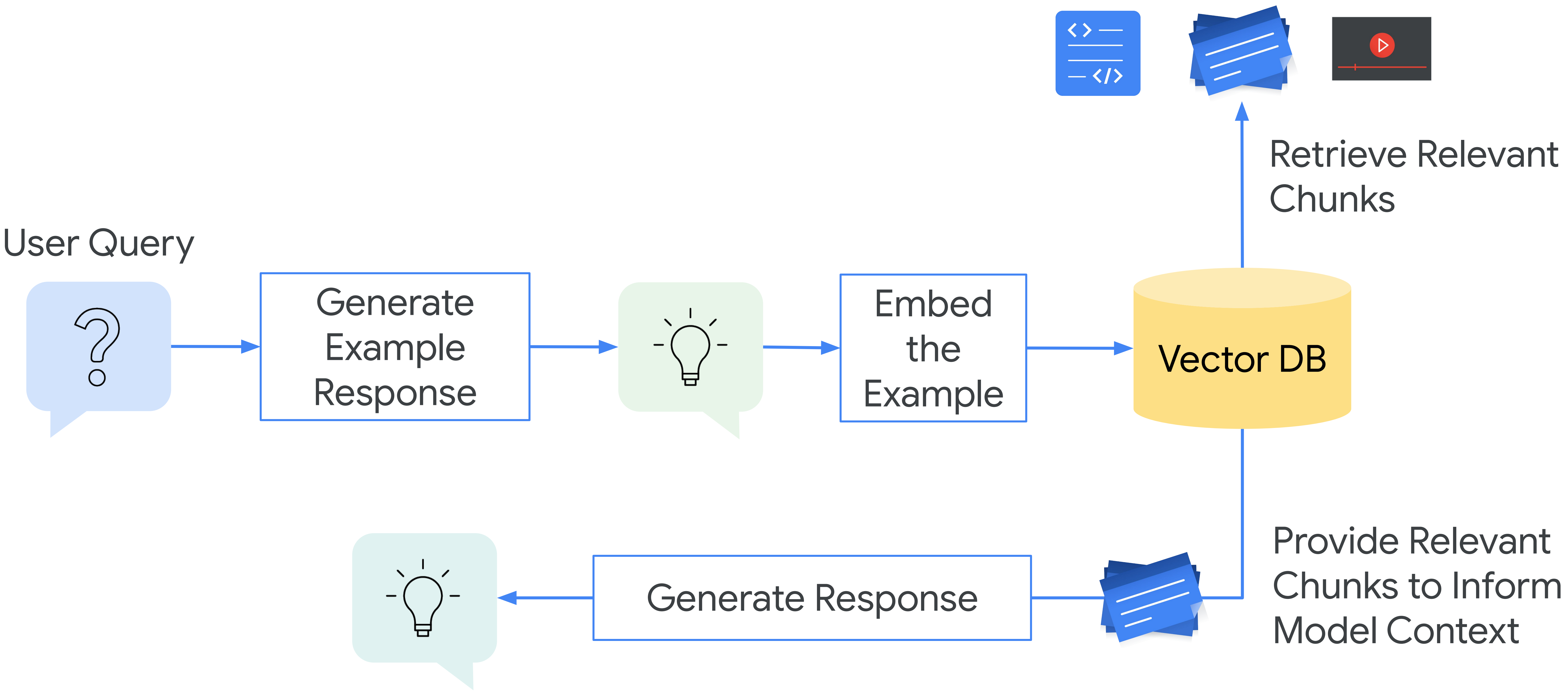
Proprietary + Confidential

A response can look very different from a query

Why is the sky blue?

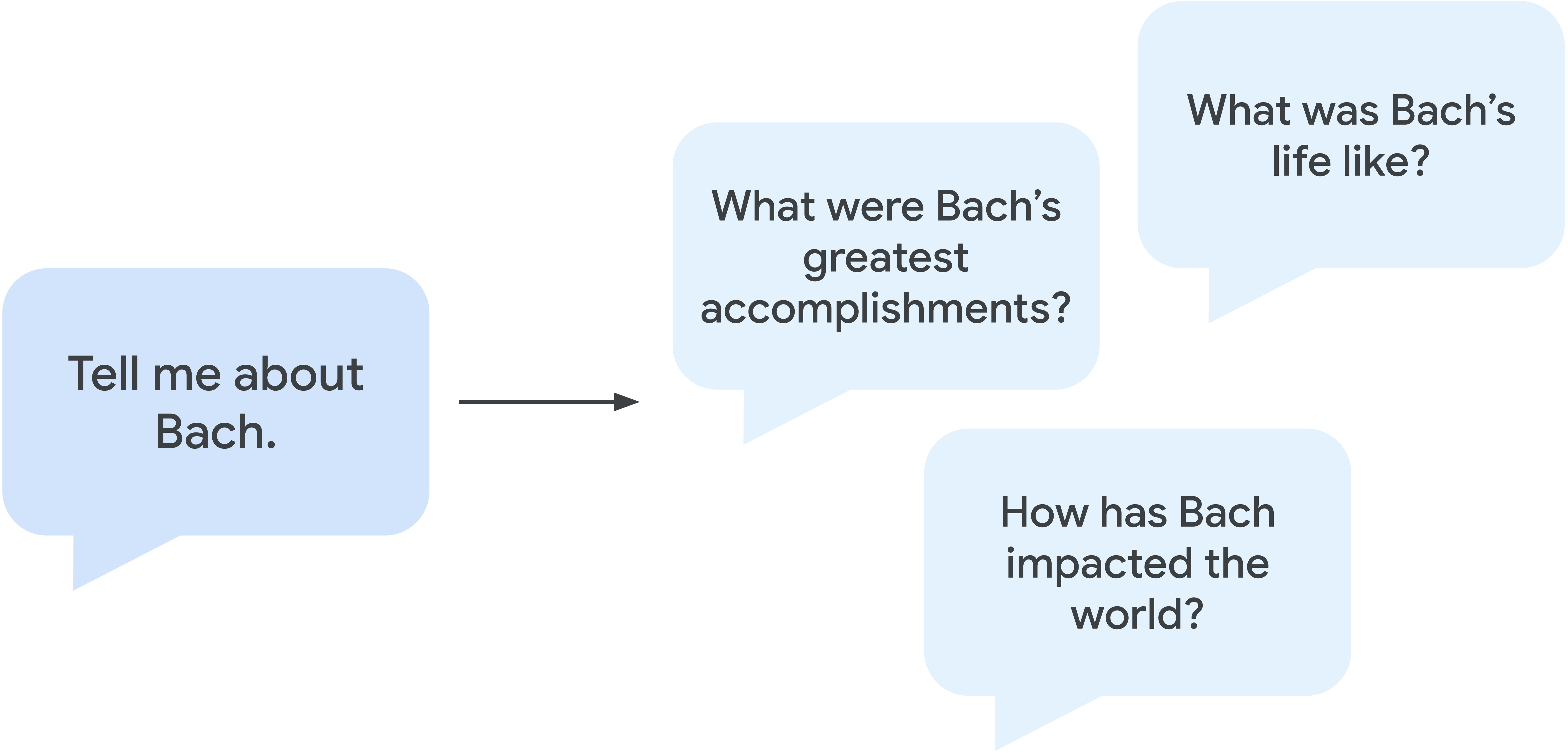
Sunlight reaches the atmosphere and is scattered in all directions...

RAG: HyDE (Hypothetical Document Embeddings)

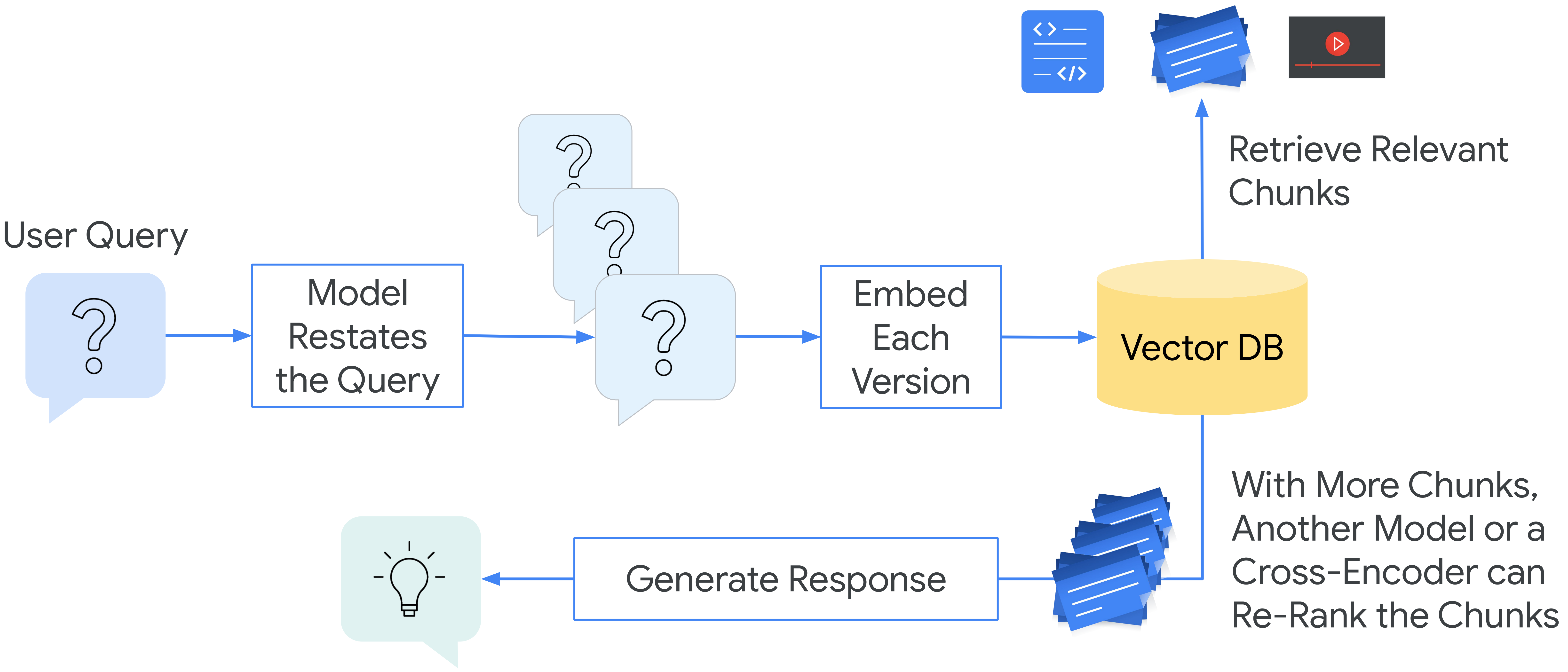


Proprietary + Confidential

There are also many ways to ask questions



RAG: Query Expansion



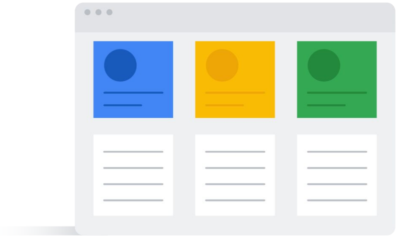
Proprietary + Confidential

You may also wonder what content informed a response

We have 27 trucks in the fleet.

According to...

Which Documents?

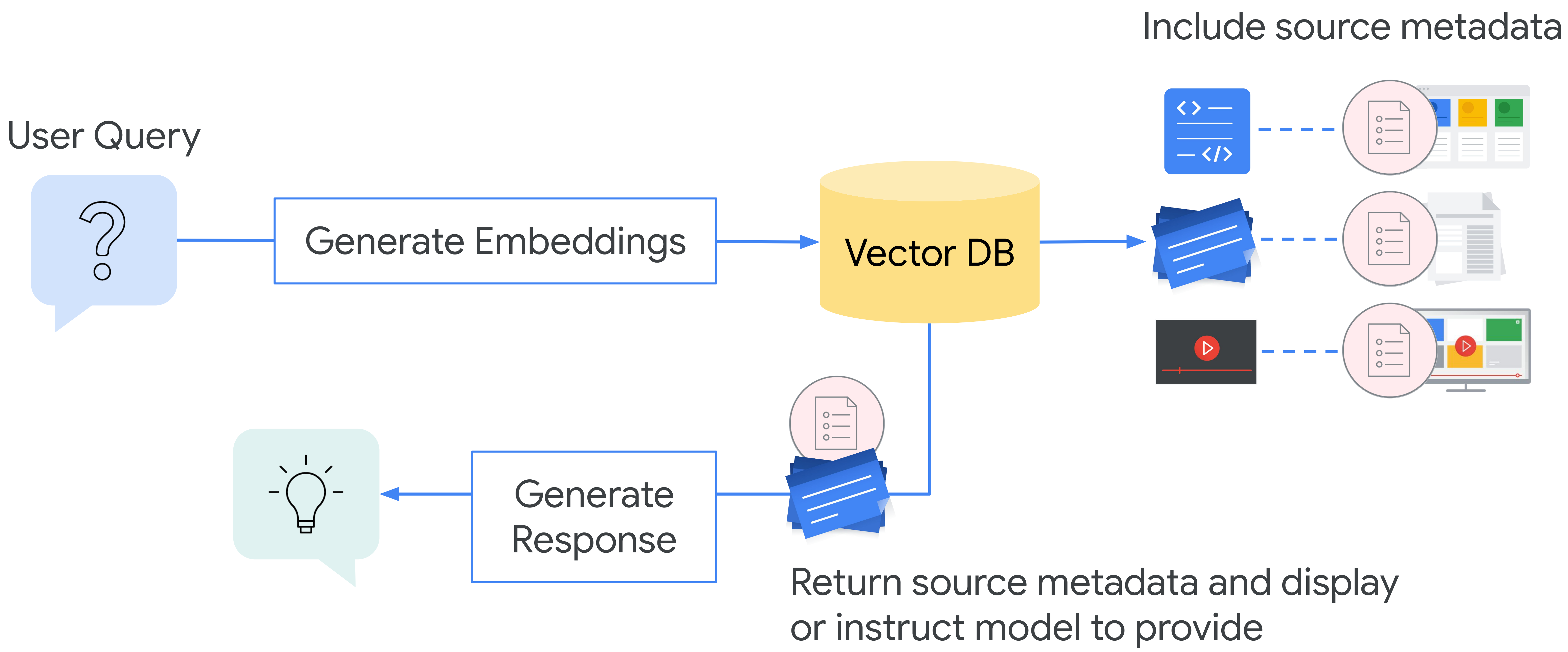


What Web Content?

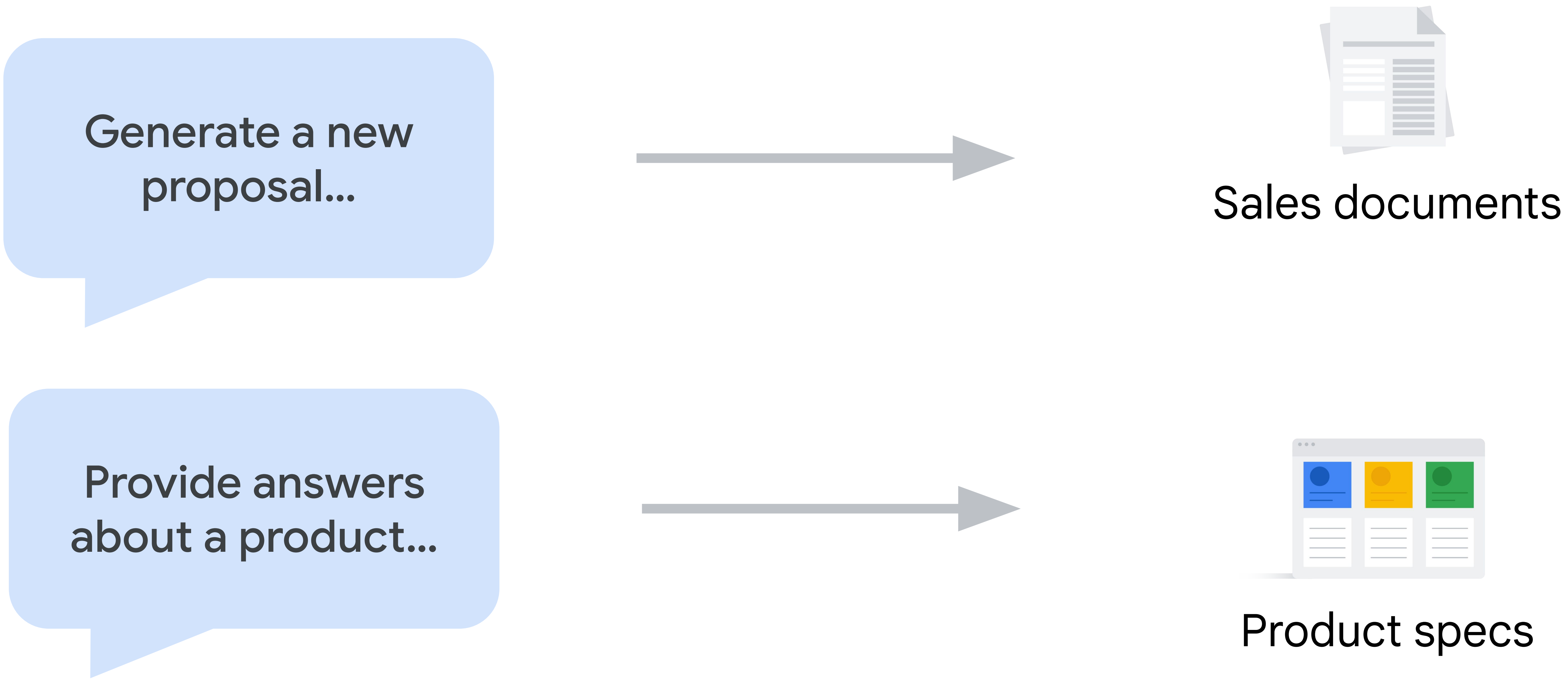
Which Video(s)?

Proprietary + Confidential

RAG: Grounding

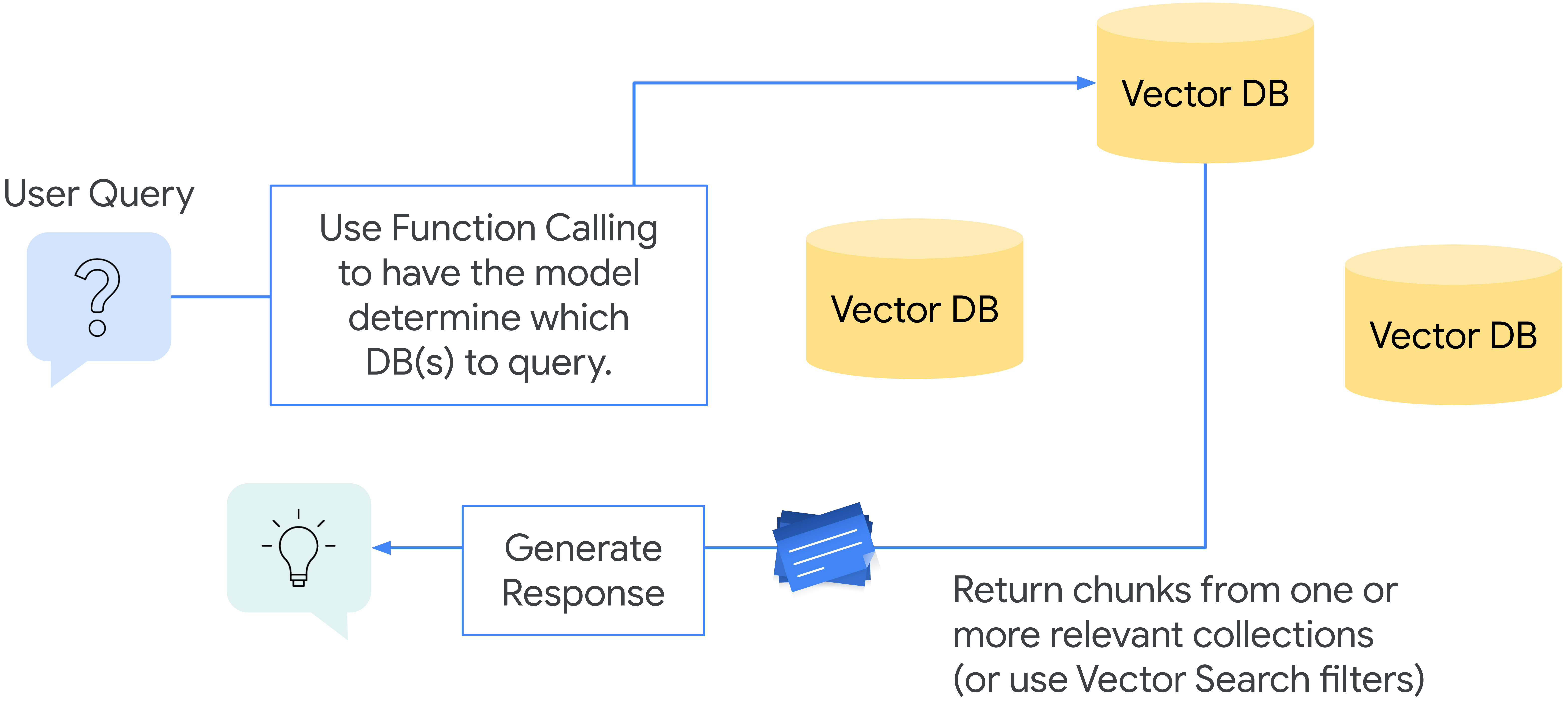


You may want to provide context from different sources depending on the request



Proprietary + Confidential

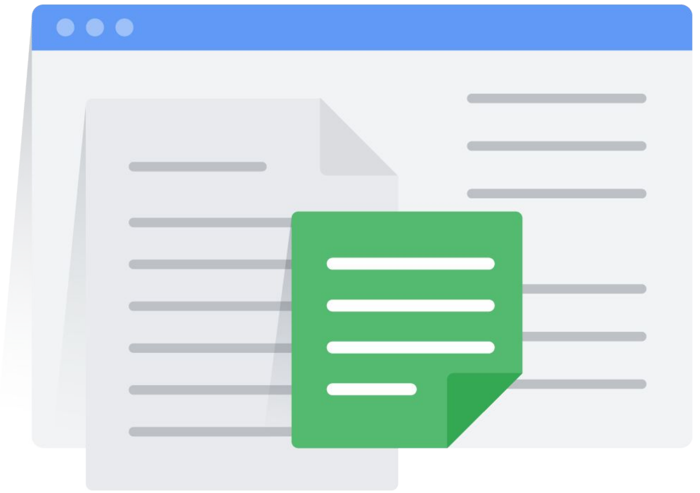
RAG: Routing



Proprietary + Confidential

You may want to find answers in text or in images (plots or tables), this is called Multimodal RAG

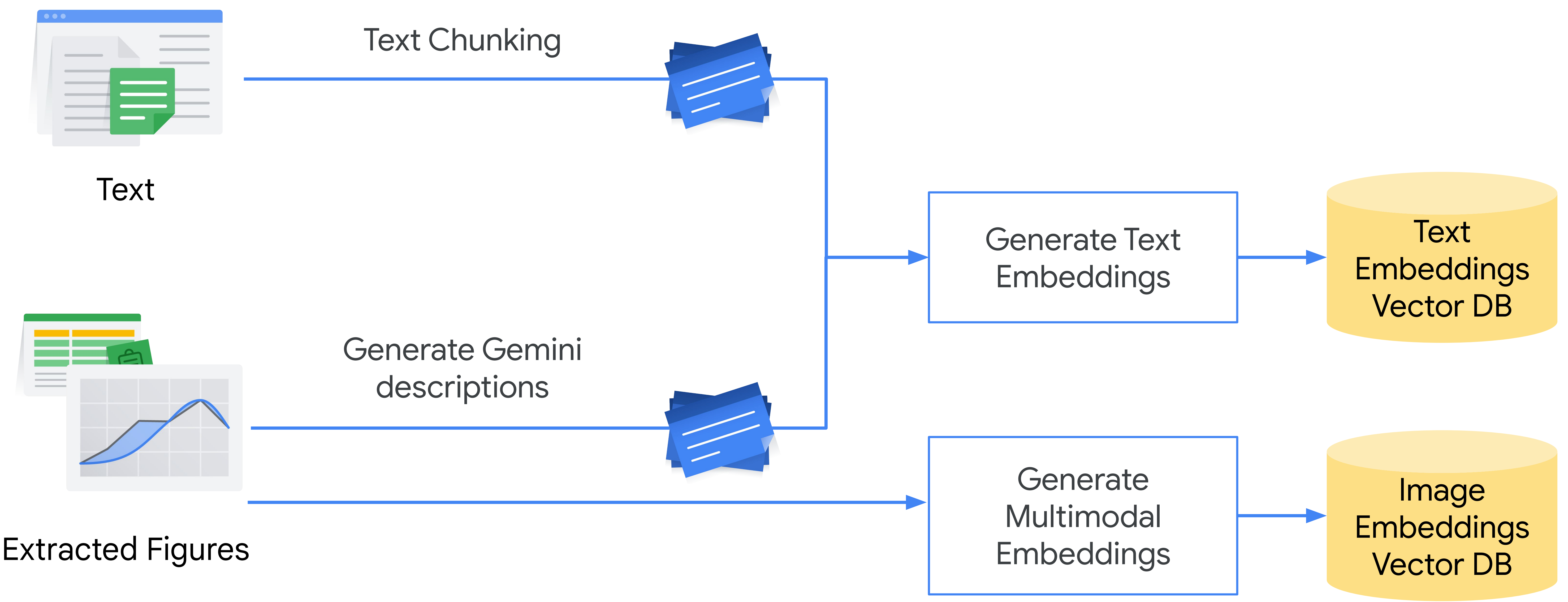
How have our earnings changed in the last 6 quarters?



The answer could be in text...

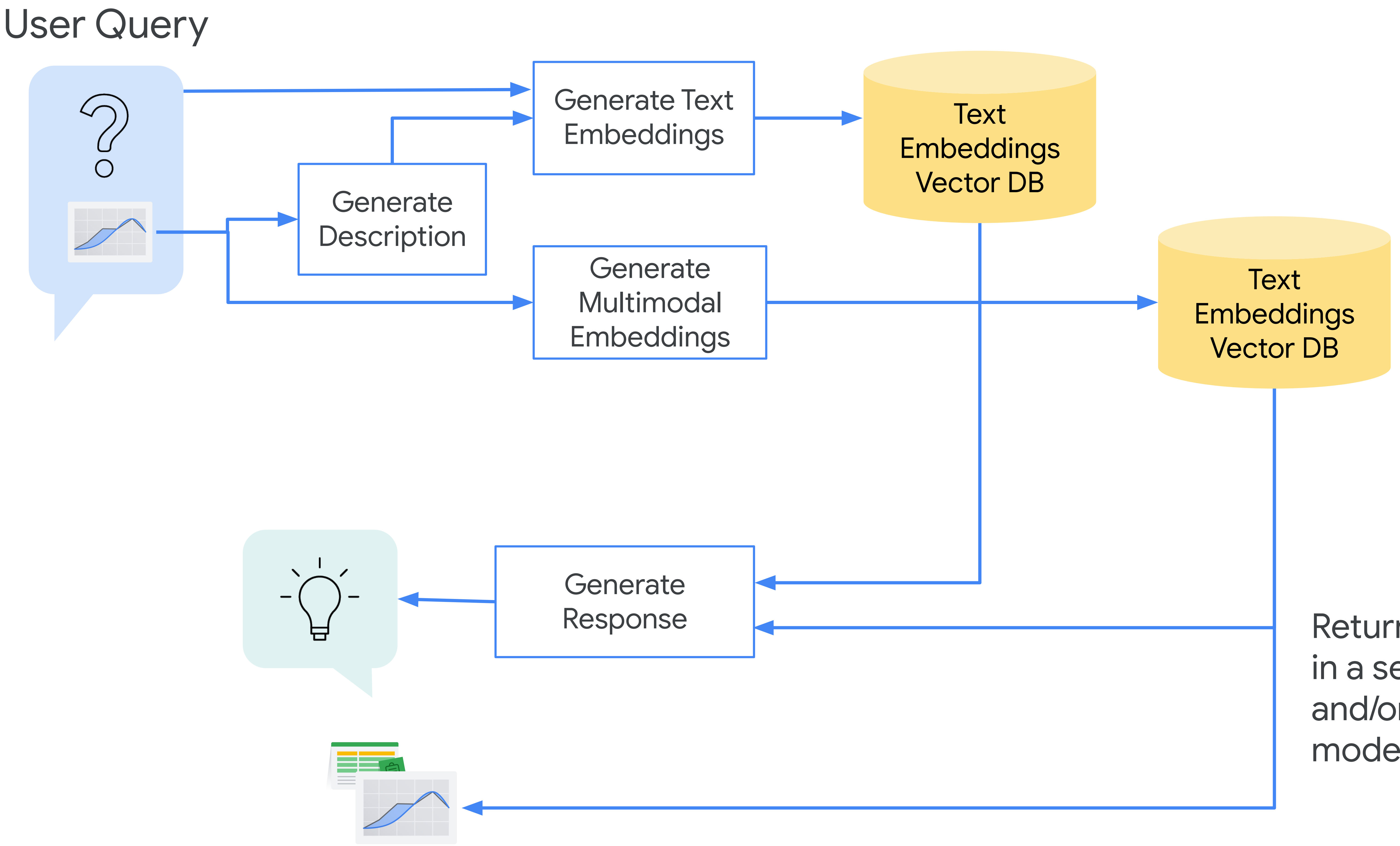
...or in an embedded table/plot.

Multimodal RAG: Generating the Databases



Proprietary + Confidential

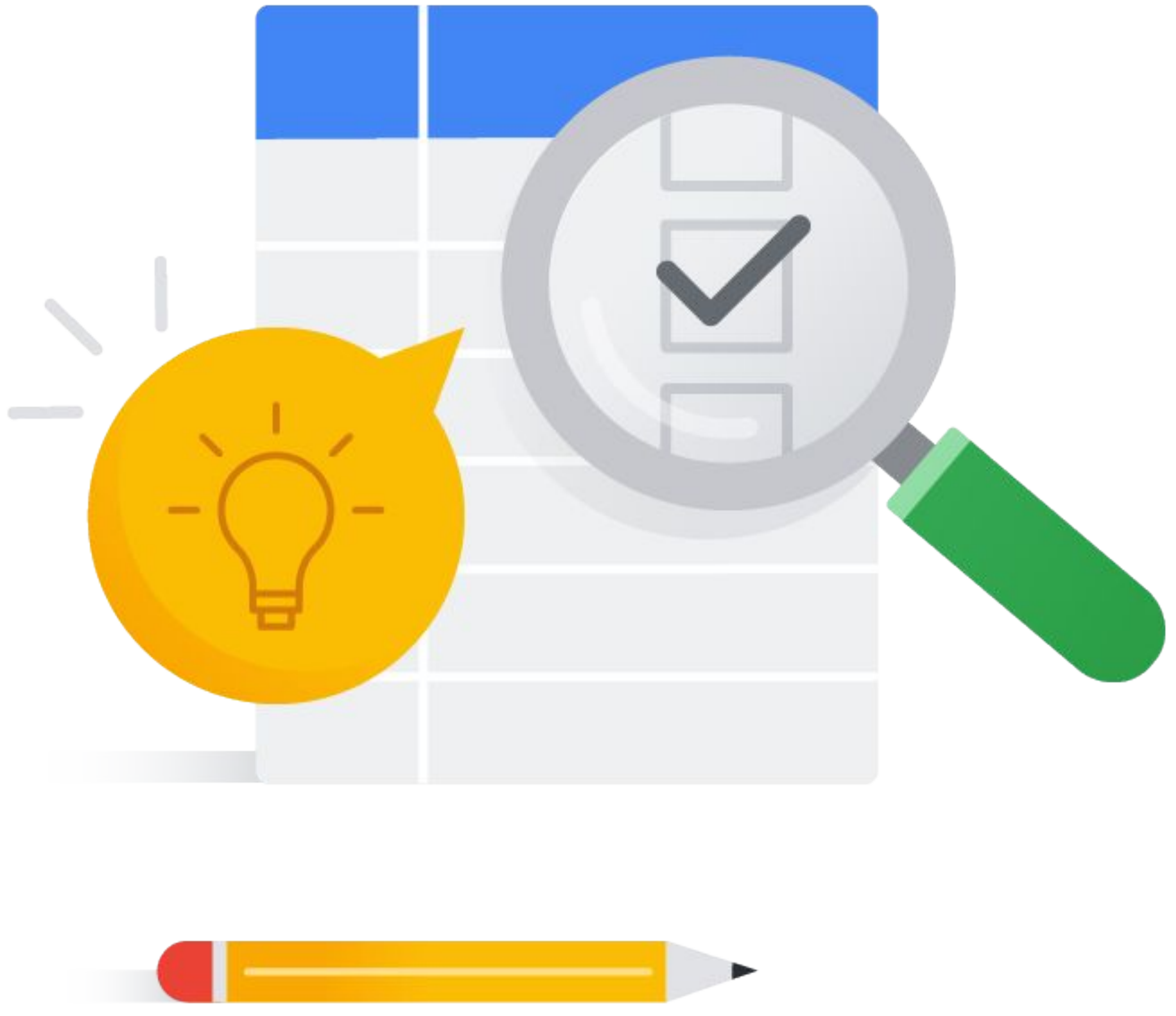
Multimodal RAG: Query Time



Return matching images in a separate UI element and/or pass them to the model for its response.

Lab

1 hour

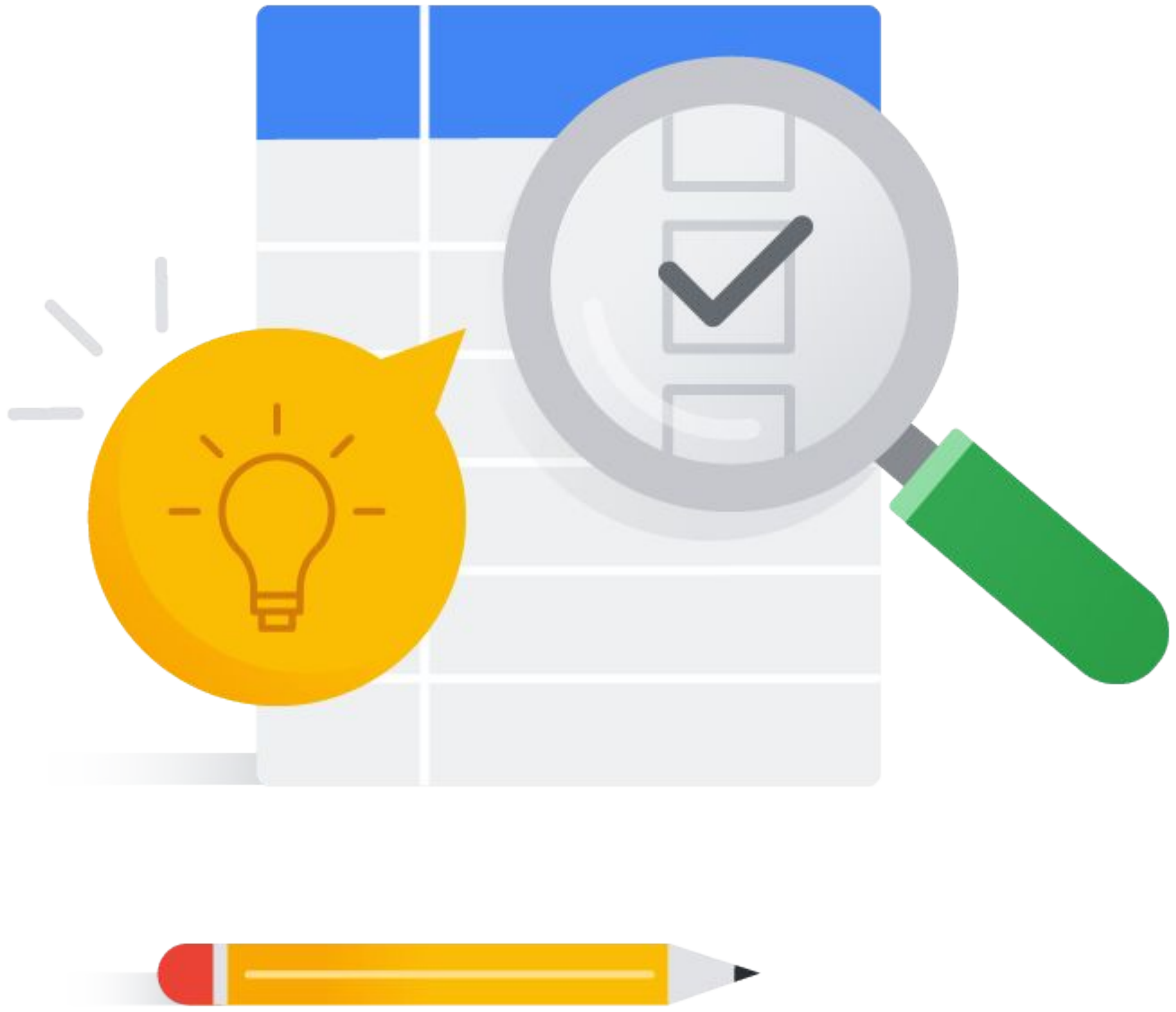


Lab: Using BigQuery Embeddings in a RAG Architecture

Proprietary + Confidential

Lab

1 hour



Lab: Multimodal Retrieval Augmented Generation (RAG) using the Vertex AI Gemini API

In this module, you learned to ...

01 Architect RAG solutions for real-world customer problems

02 Choose the right embedding technology for creation, storage and serving

03 Optimize workflows and RAG solutions

Proprietary + Confidential

Google Cloud