

MongoDB Day Frankfurt

From Data to Agent: Build AI-Powered Apps with MongoDB, BigQuery, and Vertex AI

Daniel Quinlan

Data Management Customer Engineer Specialist

Wojciech Marusiak

Infrastructure Modernisation Customer Engineer



Agenda

- 01 Google Cloud Generative AI
- 02 Google and MongoDB integrations
- 03 Building production GenAI with RAG
- 04 Building Industry Agentic Apps

01. Google Cloud Generative AI



Google cloud delivers the most comprehensive platform for building and adopting agents

Integrated Gemini assistants

Gemini for Google Cloud | Gemini for Google Workspace

AI Applications and Agents

Google pre-built | Customer built | Partner built

Google Agentspace

Enterprise Search | NotebookLM | Agent Gallery

Vertex AI

Model Garden | Model Builder | Agent Builder

Data

Multimodal | Vector Search | AI Insights | Data Science

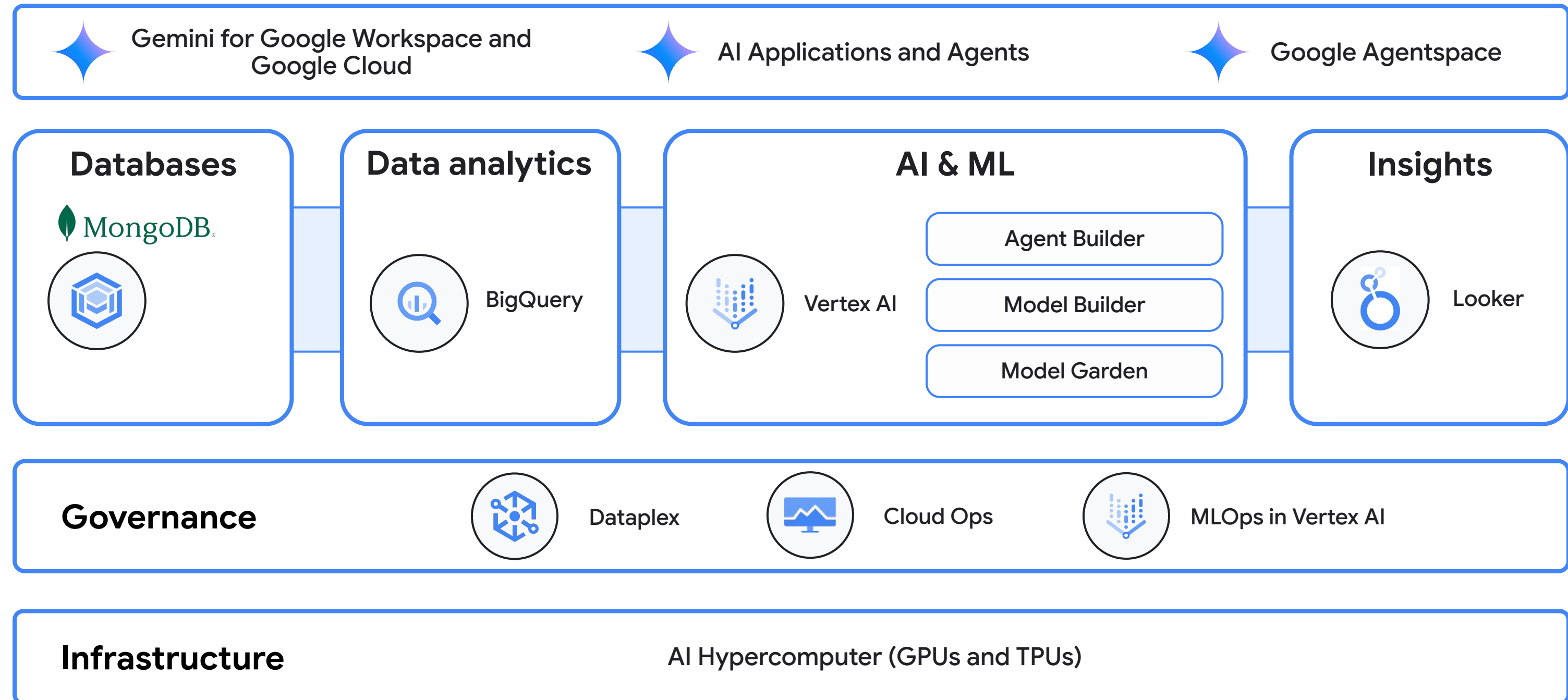
Models

Gemini | Imagen | Veo | Partner | Open

AI Hypercomputer

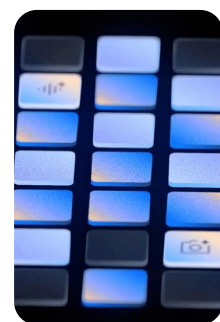
Performance-optimized hardware | Open software | Flexible consumption

A unified platform from data to deployment and for all your predictive, generative, and agentic needs



Vertex AI Model Garden

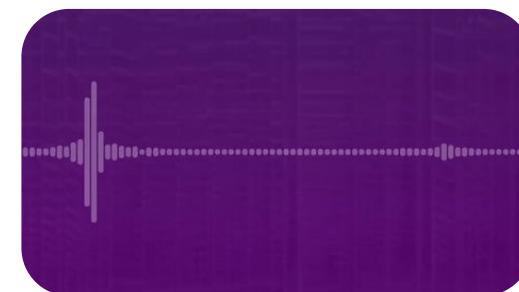
200+ models from Google and the industry



Gemini 2.5



Imagen 3



Veo 2

Lyria



Llama 3.1, 3.2,
and 3.3



Large, Nemo,
Codestral

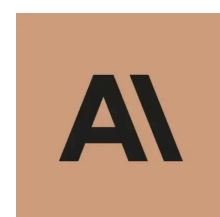
Embeddings API

Text and multimodal
Embeddings

Chirp 3



Jamba



Claude 3, 3.5, & 3.7



Gemma



..and many more

Foundation Models

(Google, open, and partner)

Generative Media on Vertex

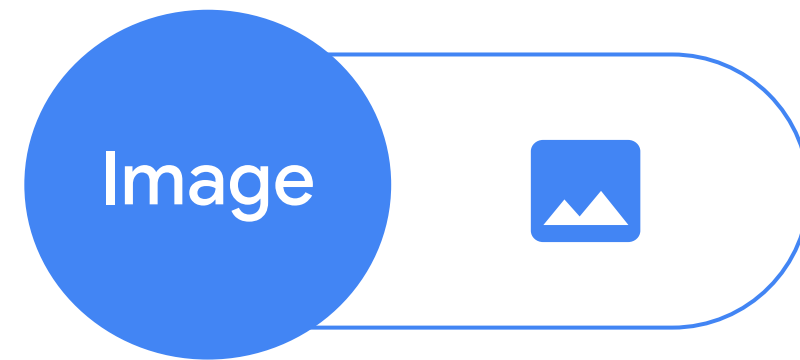
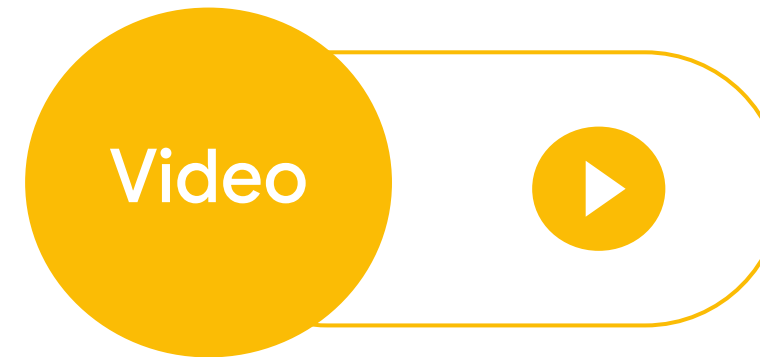


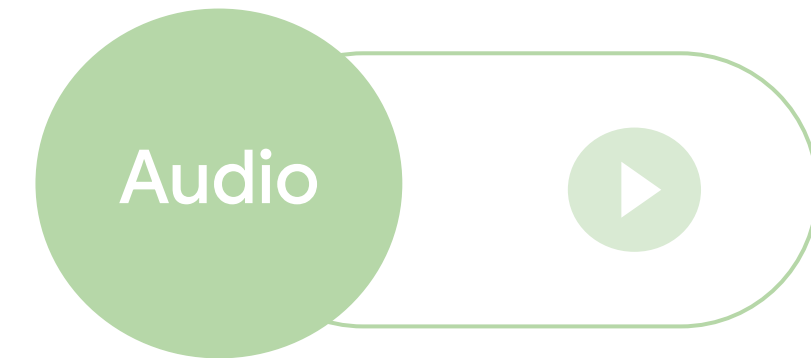
Imagen 3

High quality image generation and editing



Veo 3

Exceptional video generation, editing and advanced camera controls



Chirp and Lyria

Lyria- high quality music
Multi-speaker voices and voice cloning



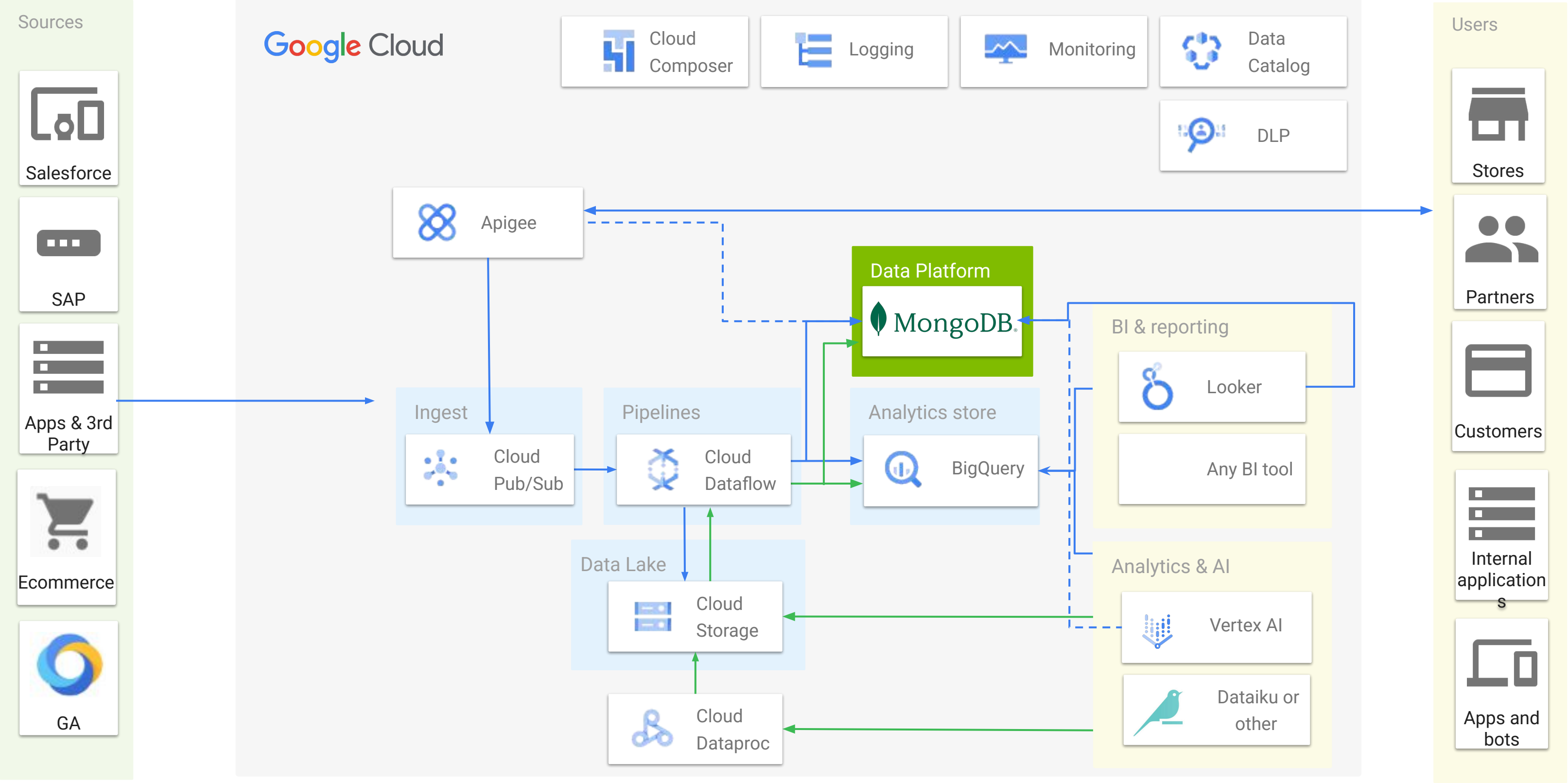
Gemini - analyze, enhance prompts, create stories

AI as an inflection point for the enterprise

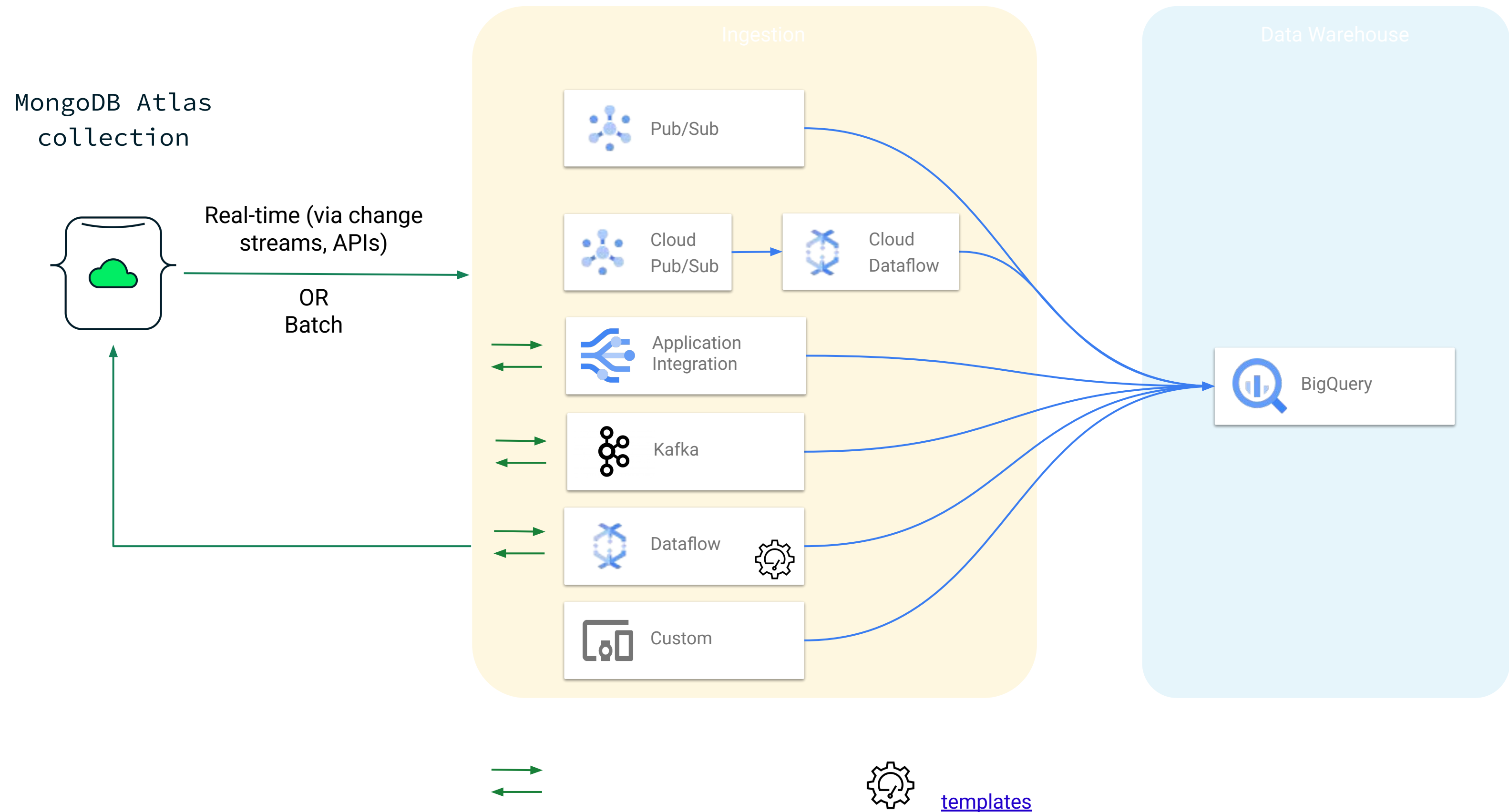
02 Google and MongoDB integrations

MongoDB Atlas integration with Google Cloud

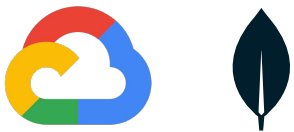
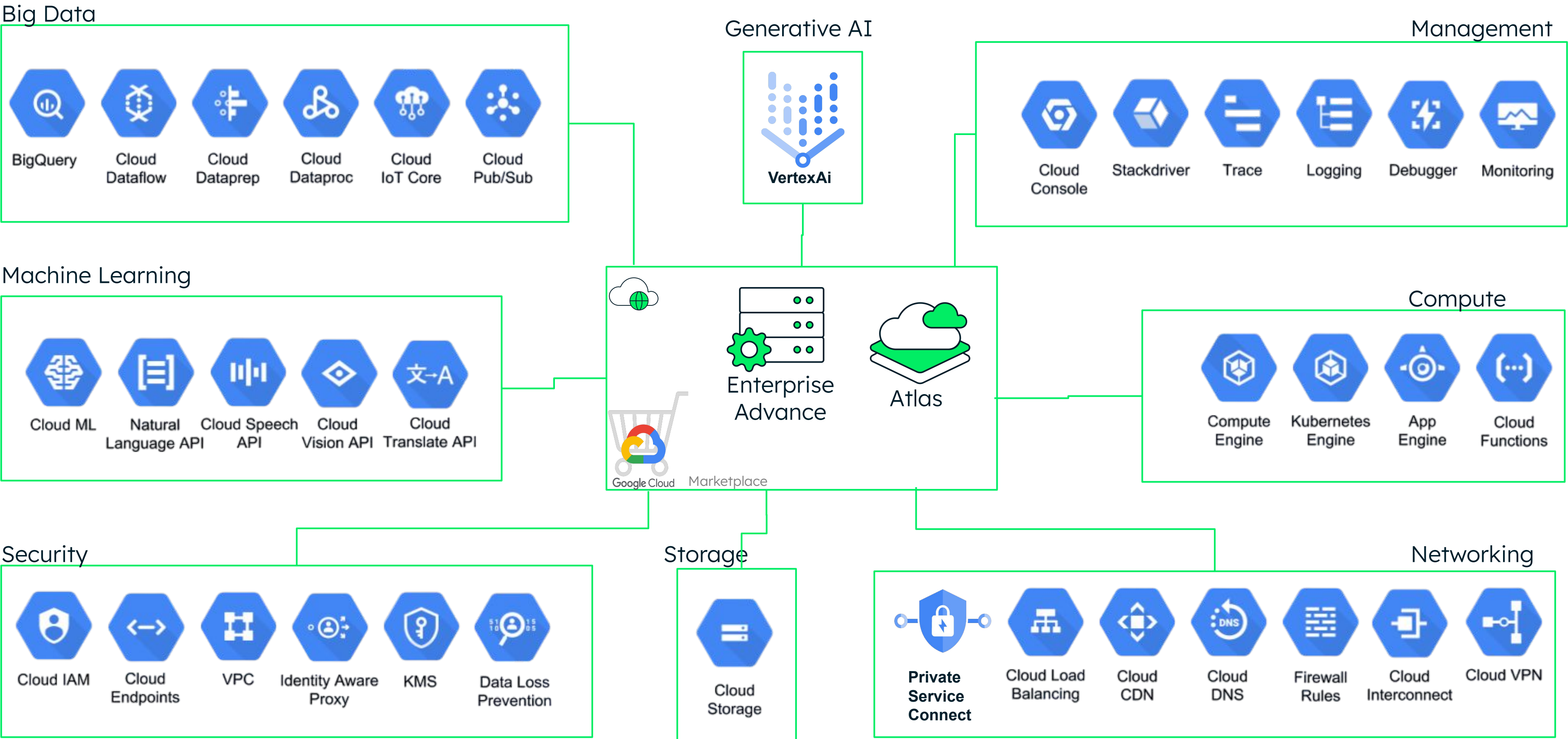
Architecture: Data & Analytics - End to end Serverless Analytics Platform



MongoDB & BigQuery integration options Options



From data to intelligence



03. Building production GenAI with RAG

How to build effective search on
proprietary documents

What is RAG?

1

Retrieve relevant documents

Convert the user's question into a vector and search for semantically similar documents in the database.

2

Enhance prompt with returned chunks

Combine the retrieved documents with the original query to provide context.

3

Send enhanced prompt to the LLM

Feed the enhanced prompt to the LLM, which generates a more accurate and informed response.

Fetch some data that is relevant to the problem.

Text completion output is more relevant and reliable.

Retrieval Augmented Generation

Add it to the prompt which is fed into the LLM → modified prompt

Modified Prompt Template

You are an intelligent assistant helping the users with their questions on **{{company | research papers | ...}}**. Strictly Use ONLY the following pieces of context to answer the question at the end. Here are the rules:

Do not try to make up an answer:

- If the answer to the question cannot be determined from the context alone, say "I cannot determine the answer to that."
- If the context is empty, just say "I do not know the answer to that."

CONTEXT:

{{retrieved_information}}

QUESTION:

{{question}}

Helpful Answer:

{{}} -> Dynamic values

External data

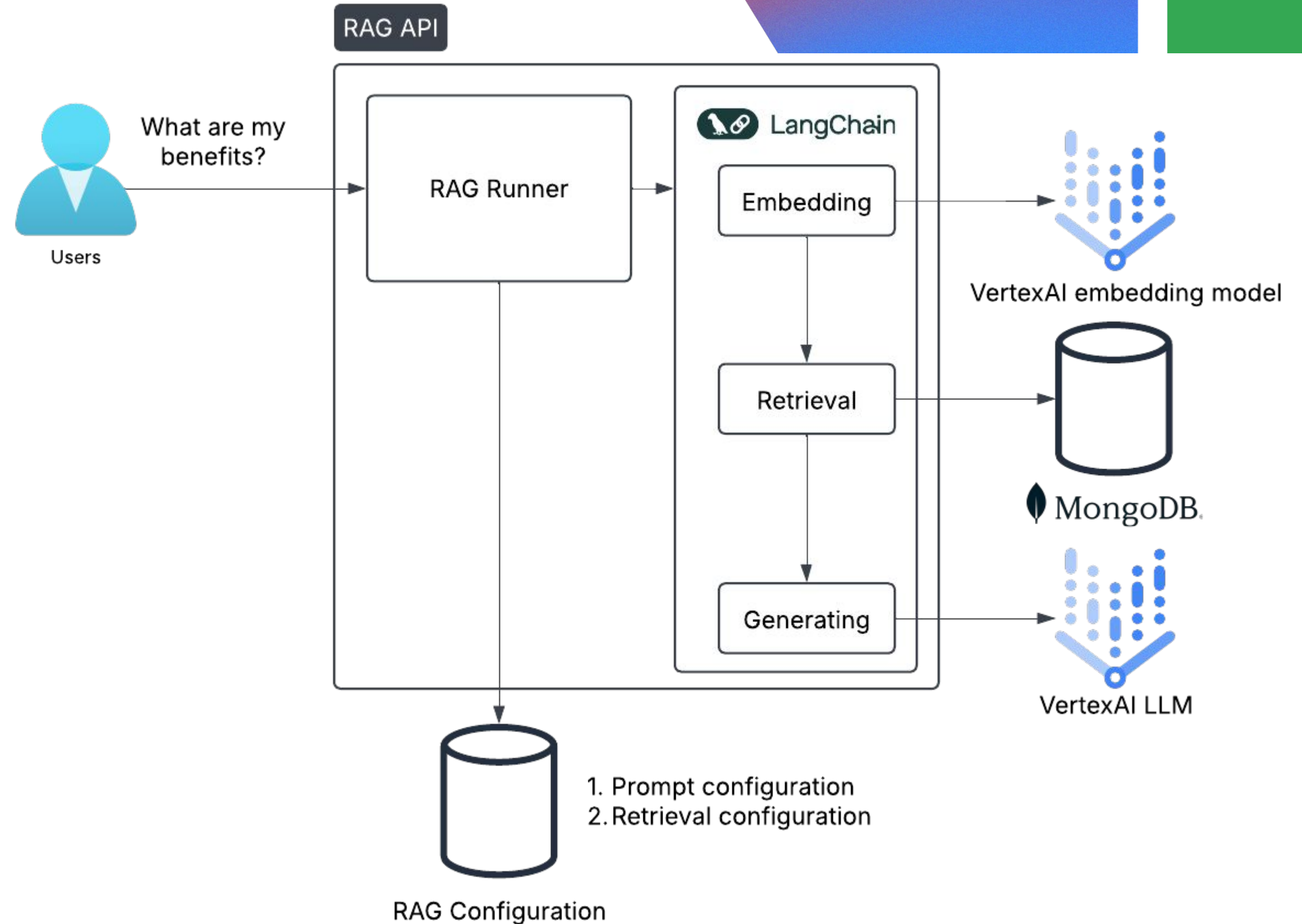
**No hallucinations
grounding**

**External data
fetched**

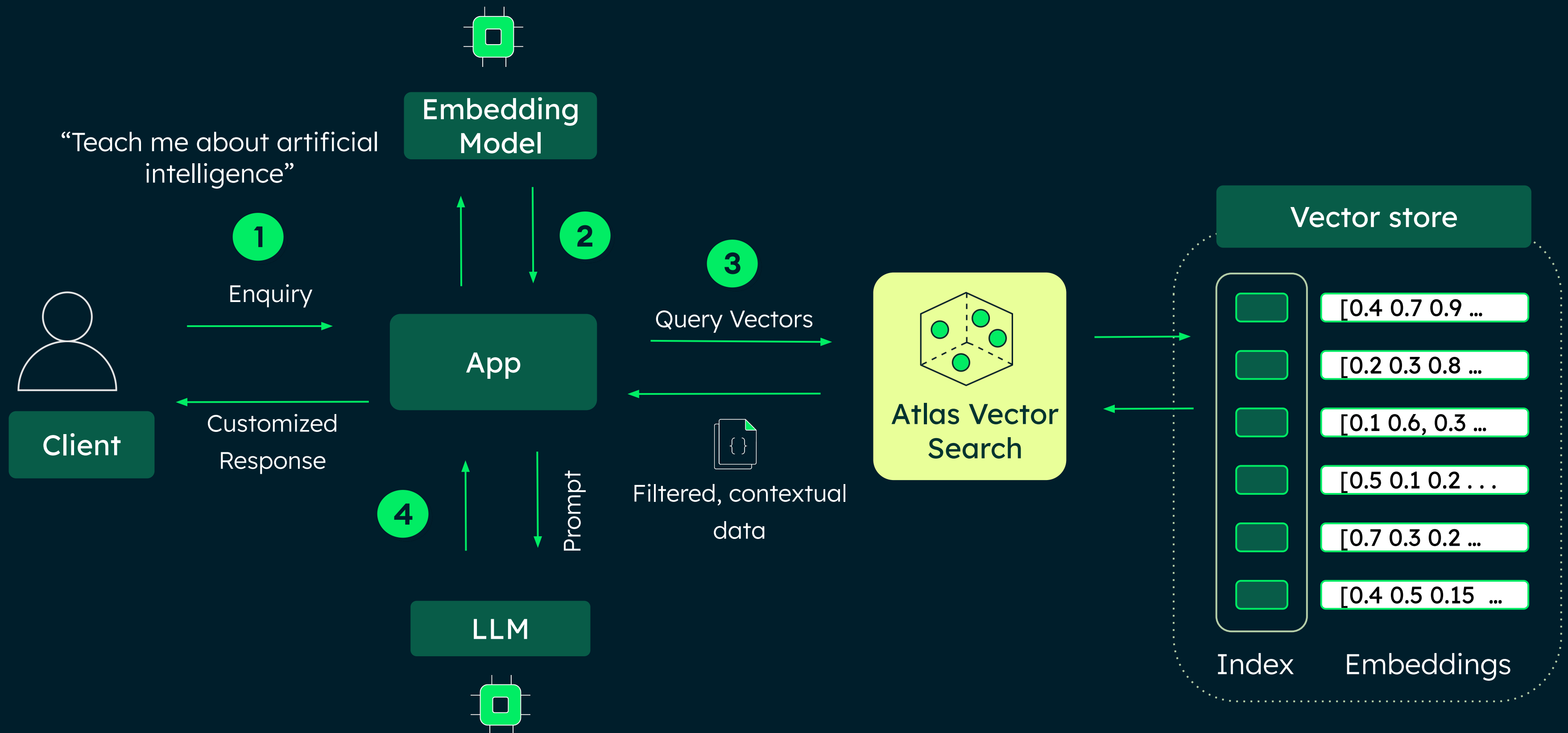
Initial user prompt

Retrieval process

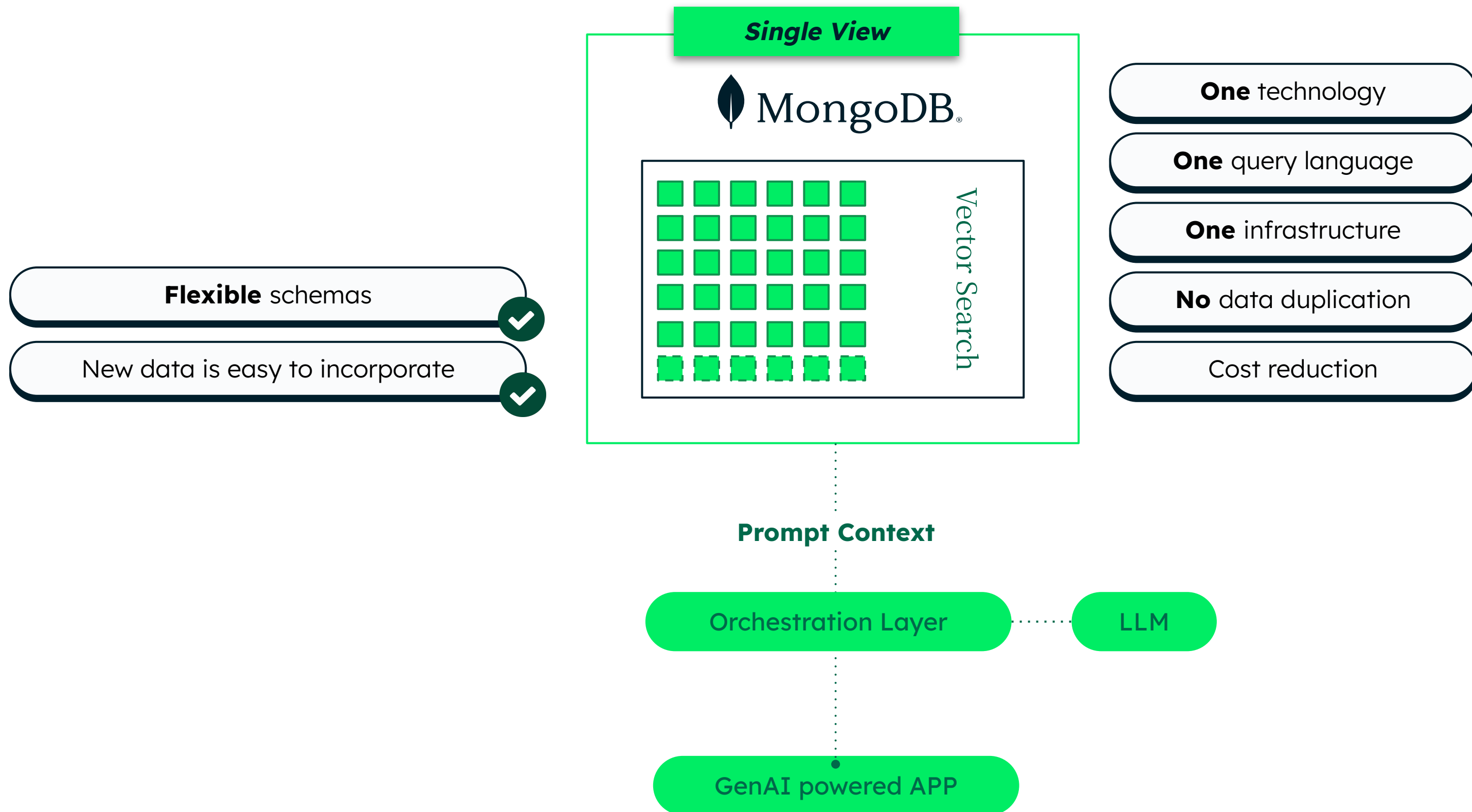
1. Fetch RAG configuration
2. Embed query
3. Retrieve chunks
4. Generate final answer



Context retrieval and long term memory

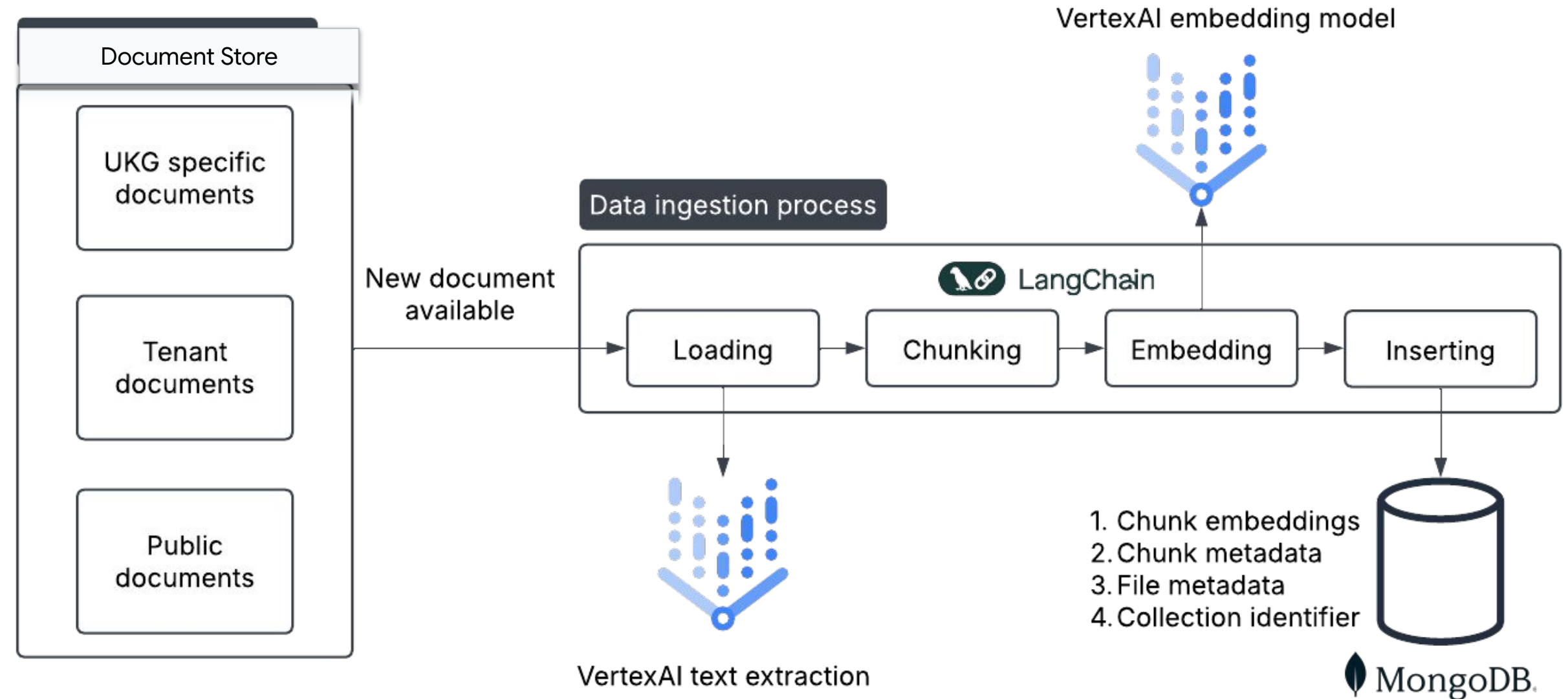


Streamlining RAG implementation



Ingestion process

1. Text extraction
2. Text splitting
3. Embedding creation
4. Indexing in a vector store



04. Building Industry Agentic Apps

From idea to agent: speeding up AI
agents development

Technology evolution for businesses



AI agents

Boost creativity and fully offload mundane tasks



AI assistants

Assist employees with tasks



Task specific AI

Automation for specific tasks within a domain



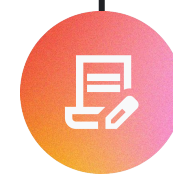
Web software

Semi-automation across businesses



Computer software

Semi-automation at business



Pen and paper

Fully manual process

Agentic development and usage choices for employees of all technical backgrounds and development preferences

Build agents with Vertex AI Agent Builder

Agents collaborate and communicate with Agent2Agent Protocol

Employees use agents with Google Agentspace

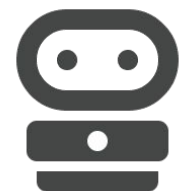


ADK Open Source
Code agents like Google

Open protocol for agent collaboration

A2A protocol complements MCP

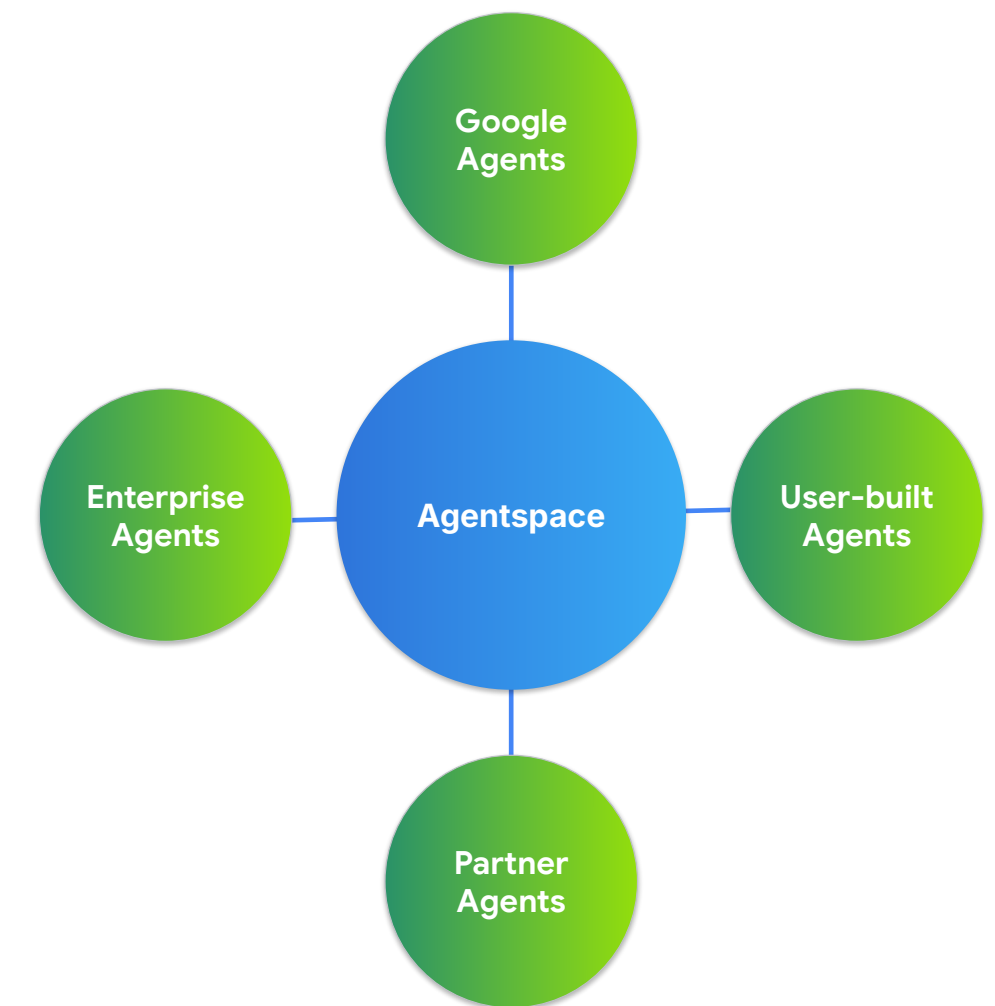
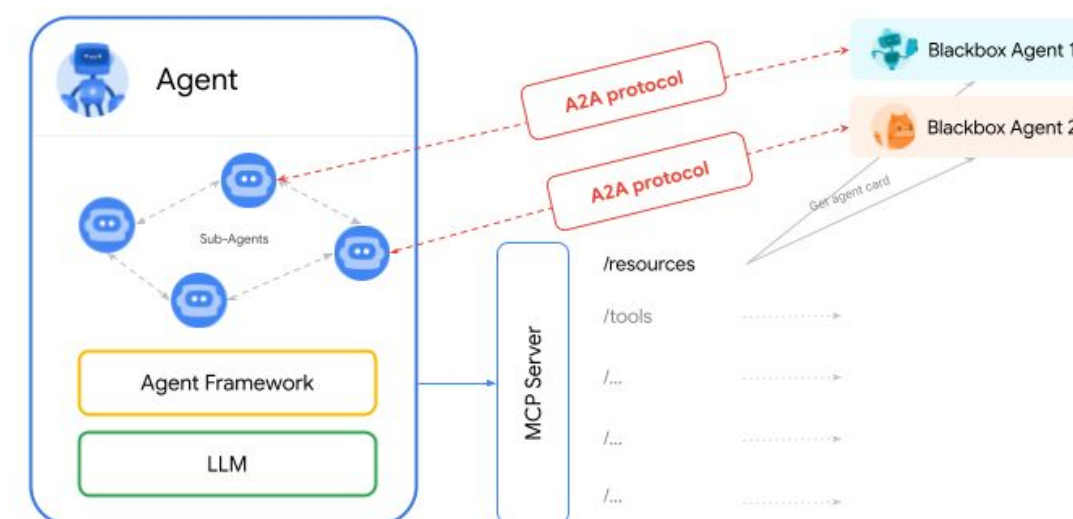
- MCP for resources and tools.
- A2A for applications and agents.



Agent Engine GA
Deploy agents to a fully managed runtime

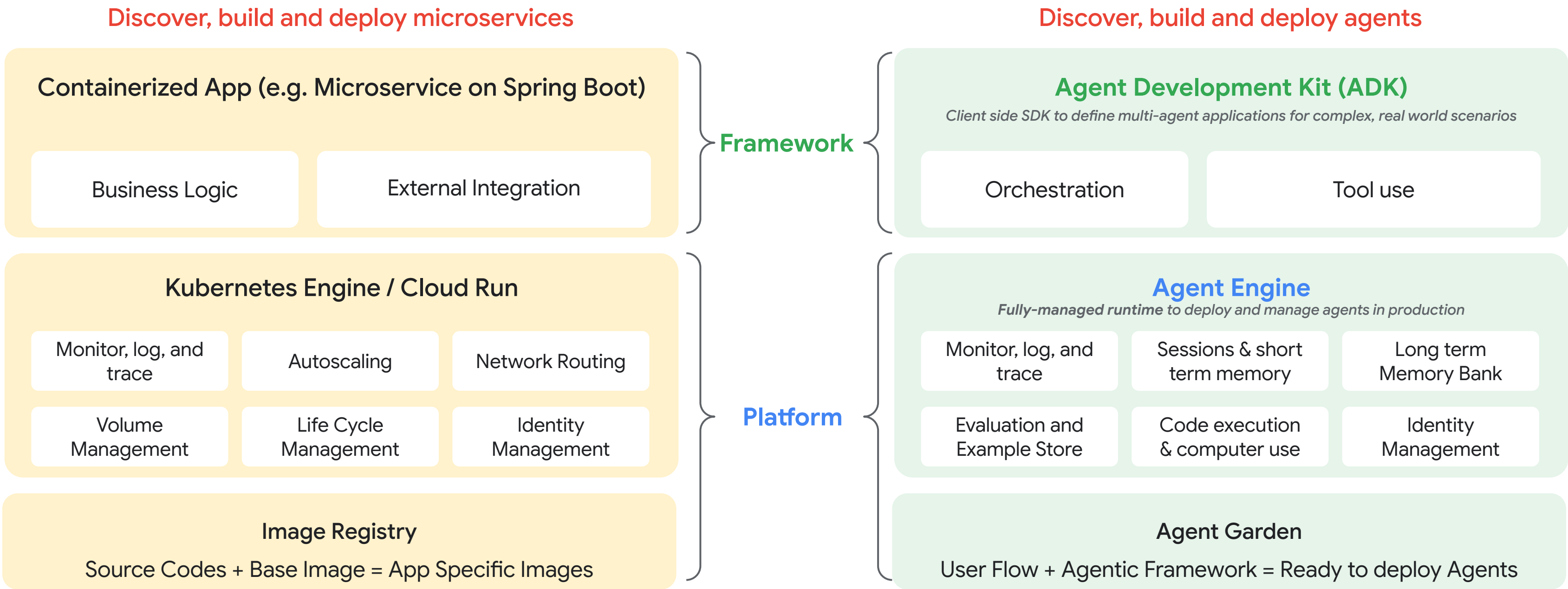


Agent Garden Open Source
Sample agents and tools get you started quickly



Simplify the process of building and deploying agents

Helping developers discover, build, and deploy agents



Simplify the process of building and deploying agents

Helping developers discover, build, and deploy agents

Scalable Deployment

- Serverless Runtime & Session Store

Agent Monitoring & Visualization

- Vertex AI Console

Evaluation

- Track & Trace
- Vertex AI GenAI Evaluation

Multiple Framework Support

- LangGraph
- LangChain
- AG2
- CrewAI
- Google ADK
- LlamaIndex

Discover, build and deploy agents

Agent Development Kit (ADK)

Client side SDK to define multi-agent applications for complex, real world scenarios

Orchestration

Tool Use

Agent Engine

Fully-managed runtime to deploy and manage agents in production

Monitor, log, and
trace

Sessions & short
term memory

Long term
Memory Bank

Evaluation and
Example Store

Code execution
& computer use

Identity
Management

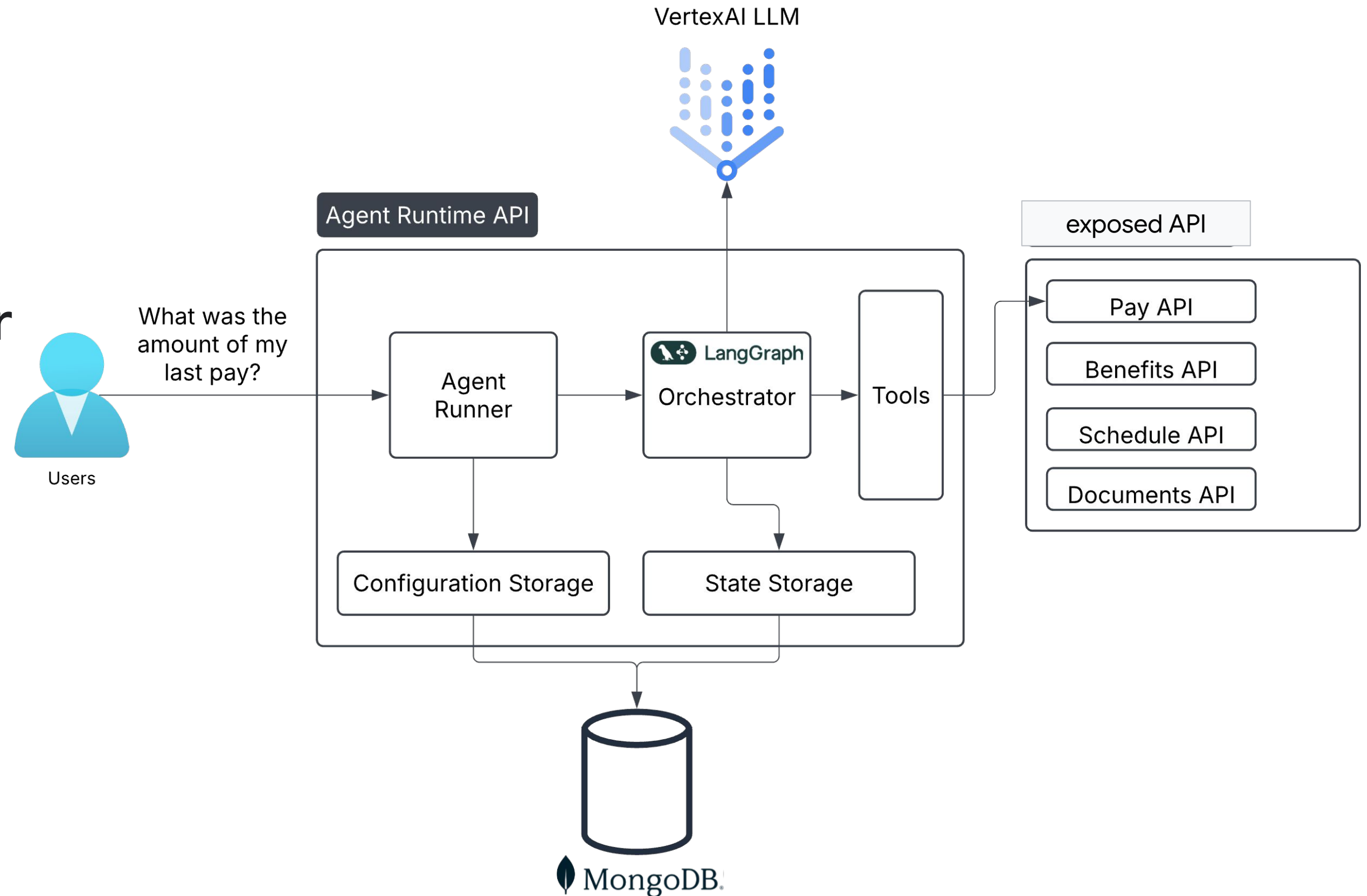
Agent Garden

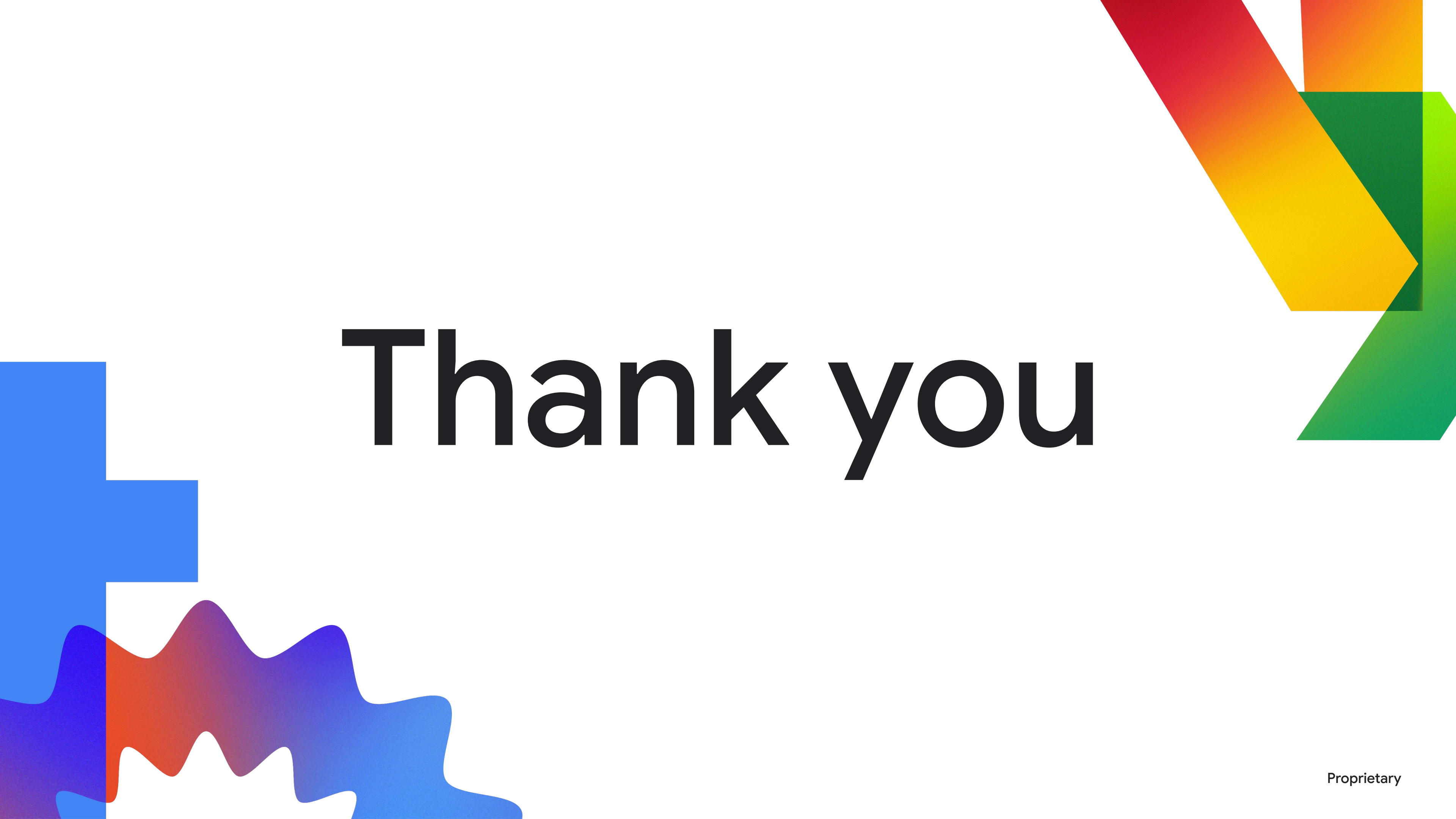
Samples and tools to accelerate agent development

MongoDB Agentic architecture

Key components

- LangGraph orchestrator
- Gemini LLM reasoning
- MongoDB configuration and state store





Thank you