Google Cloud

Next '24

Roll up your sleeves:

Craft real-world generative AI Java in Cloud Run

Generative AI adoption starts from business needs, not technological aspects

A Journey

Business needs drive technology

Understand clearly my business needs

Why Gen AI in the Enterprise

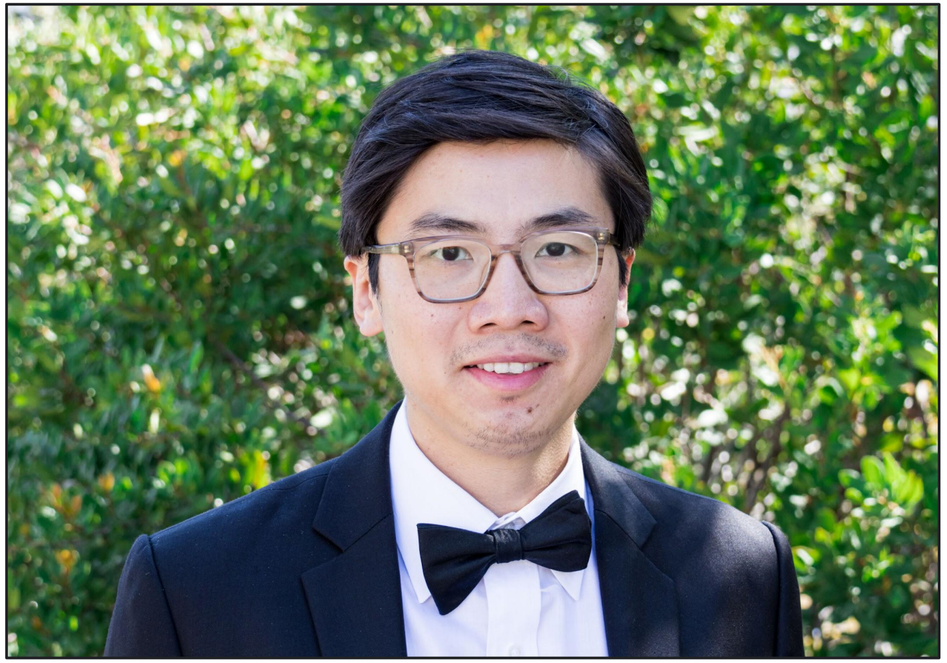
How can it help my business

Java meets Gen AI

Build, test, deploy in the enterprise

Production-ready Java with Gen AI

Fast, reliable, scalable, secure



Dan Dobrin

Enterprise App Architect,
Google Cloud

Yanni Peng

Customer Engineer,
Application Modernization,
Google Cloud

Business case

University operates the internal *University Book Review* website for the school, where students and faculty can submit or search reviews of different public and private books, documents and research papers.

The university wishes to improve this popular website.

Current Architecture

Low scalability

Application deployed on prem in a Java app server, storing materials on a server in a file system and local database

Challenging development

New features requested by the user base are either difficult or impossible to add with the current tech stack

Maintenance and security

Codebase built on older Java tech stack with multiple CVEs encountered.
Outages during deployment

Costs

Hardware and application maintenance costs significantly increased

The ask

Modern TechStack

Build and deploy the app using a modern application stack, following modern best practices

Scalability and costs

Leverage a cloud platform to scale up seamlessly during peak usage time; control costs in a pay per usage model

Flexible architecture

Architect the system with the ability to easily expand and add new business features

Feature requirements

Support on-demand or automated ability for analysis, review, classifications and summarization, issue recommendations

On-demand, automated, scale

- Ingest large scale of public and internal university materials
- Perform automatic book reviews, on demand or at scale
- On demand book analysis by keywords, refined by user
- Infer references across multiple documents, public or internal
- Find related books and up-to-date availability and pricing

Generative AI Terminology

Concepts

LLM

Machine Learning Models trained on vast amounts of data that can comprehend and generate human language text

Prompts & templates

Inputs to the LLM to generate a response or perform a tak. Prompt Engineering critical to model output

Embeddings

Numeric representation of text into a format to be processed by LLMs, captures text semantics

Vector databases

Used to store and manage embeddings with the goal of finding relevant text based on semantic similarity

Memory and state

Used for context retention over multiple interactions

SDKs and APIs

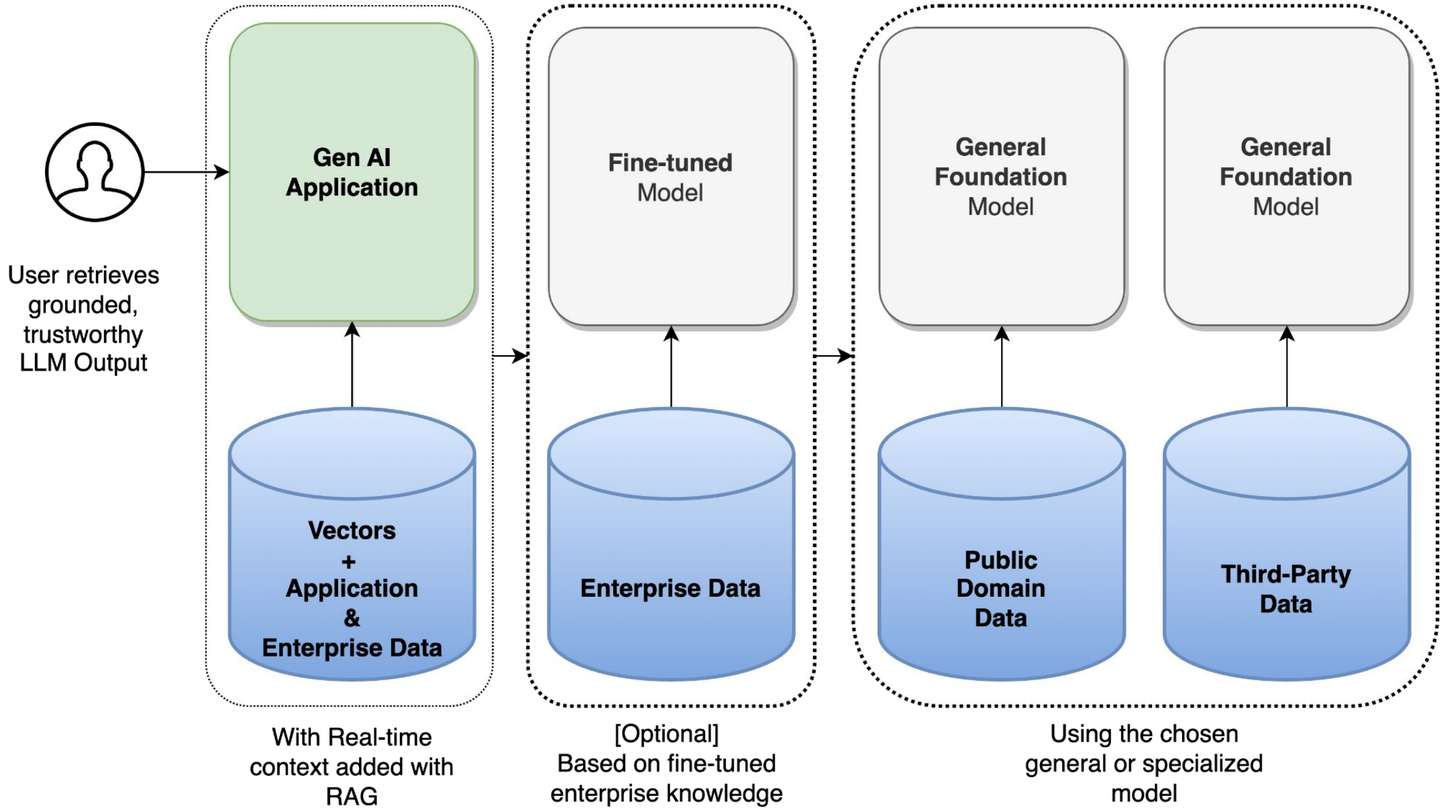
Help build applications that run anywhere

Function calling

Uses functions registered as tools in Gen AI apps. Model indicates function to call and its parameters

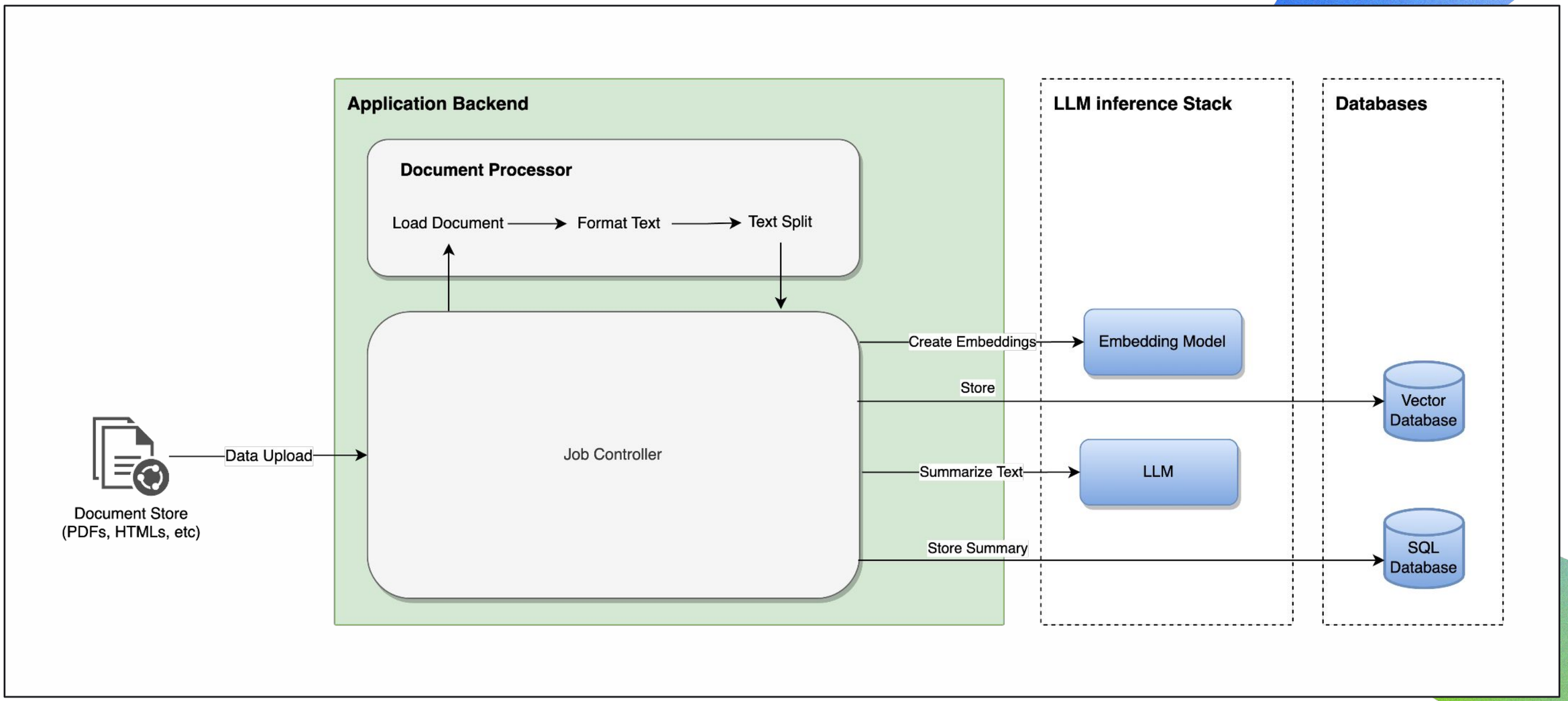
Retrieval Augmented Generation (RAG)

Process of optimizing the output of a LLM, by using additional knowledge base in addition to its training data before generating a response

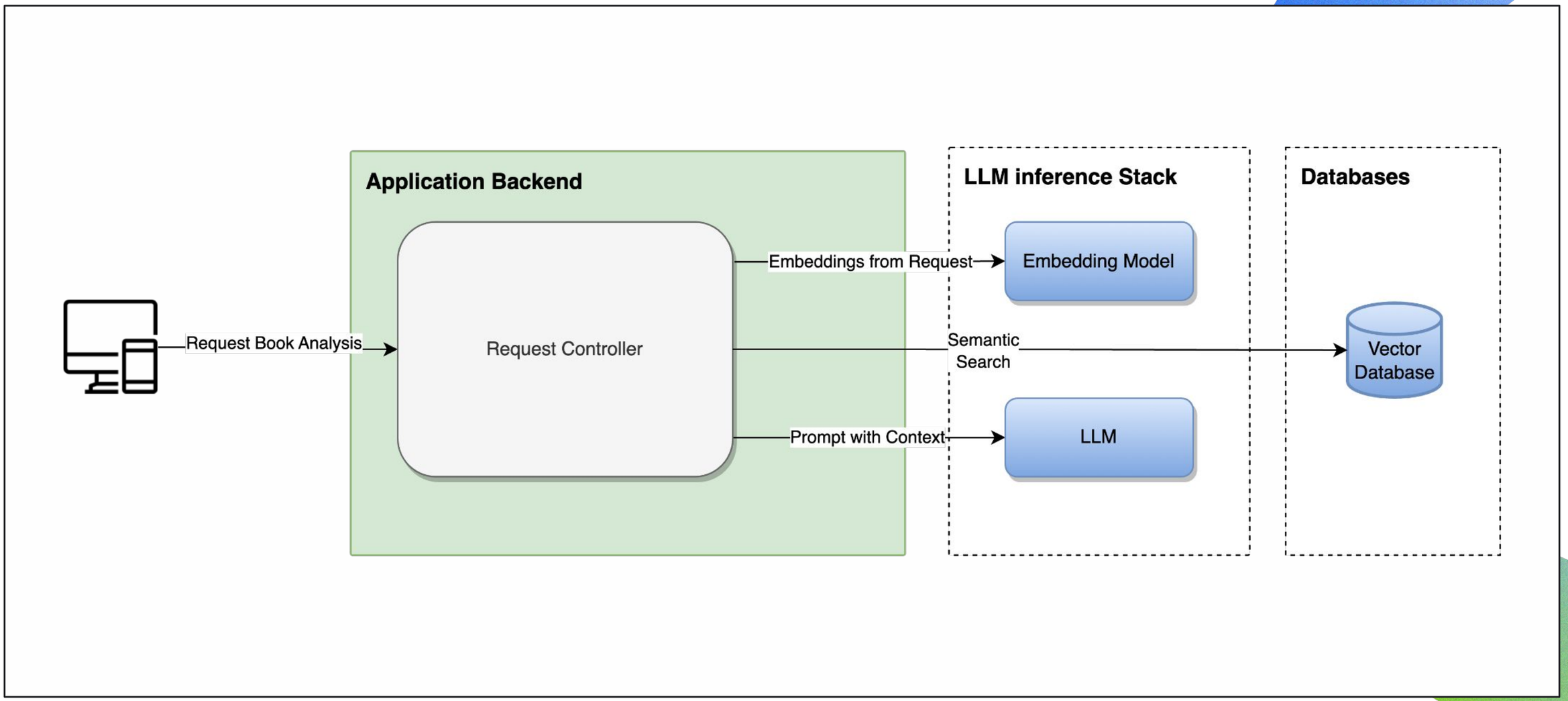


Conceptual Architecture

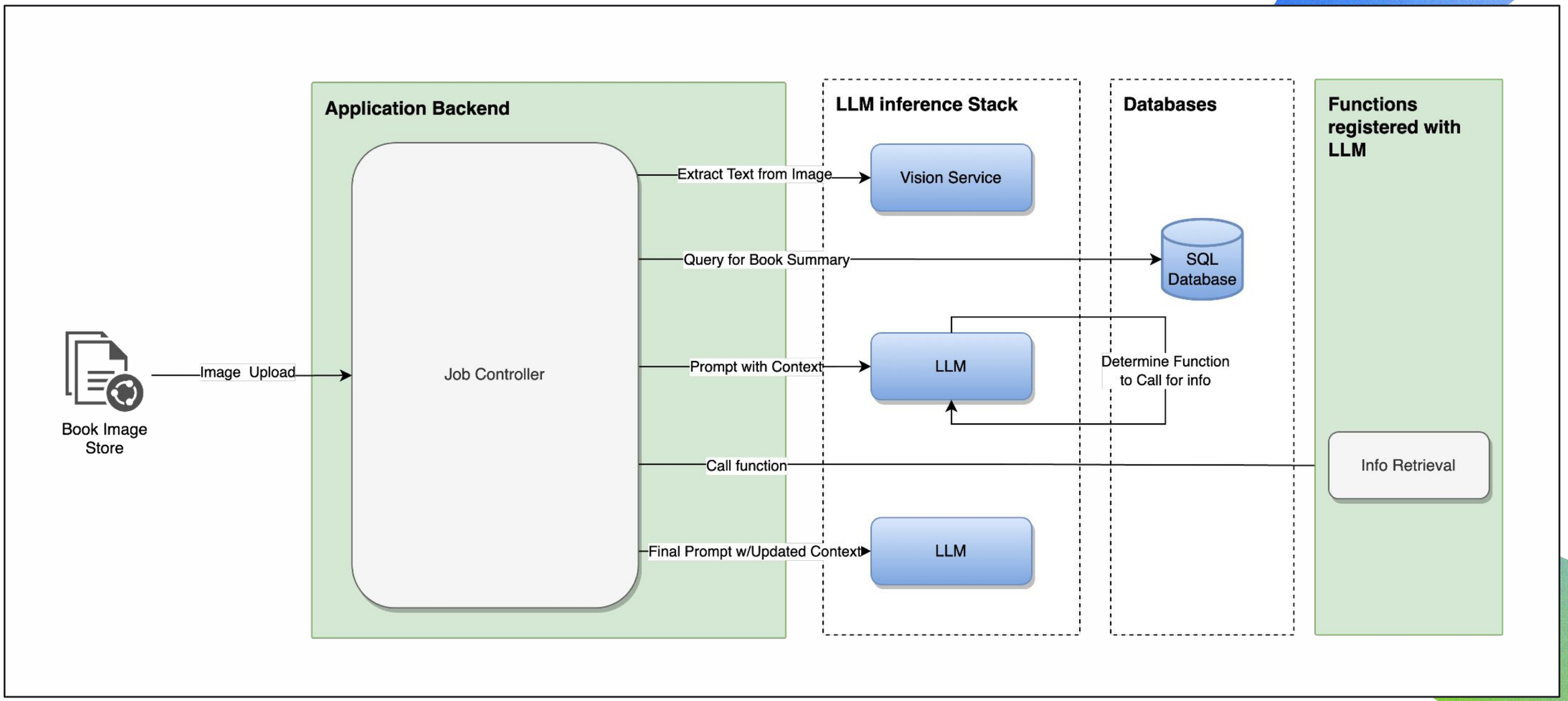
Data ingestion



Book analysis query

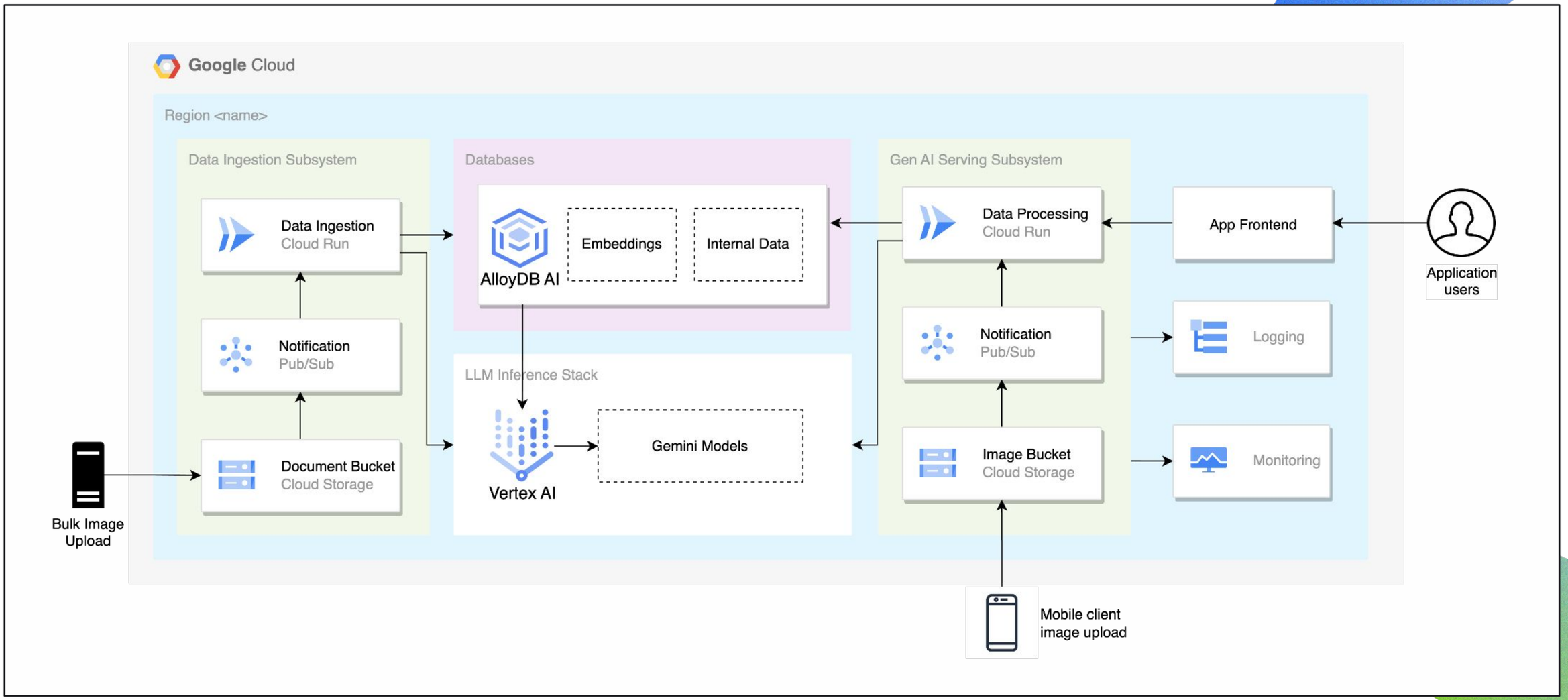


Book image query

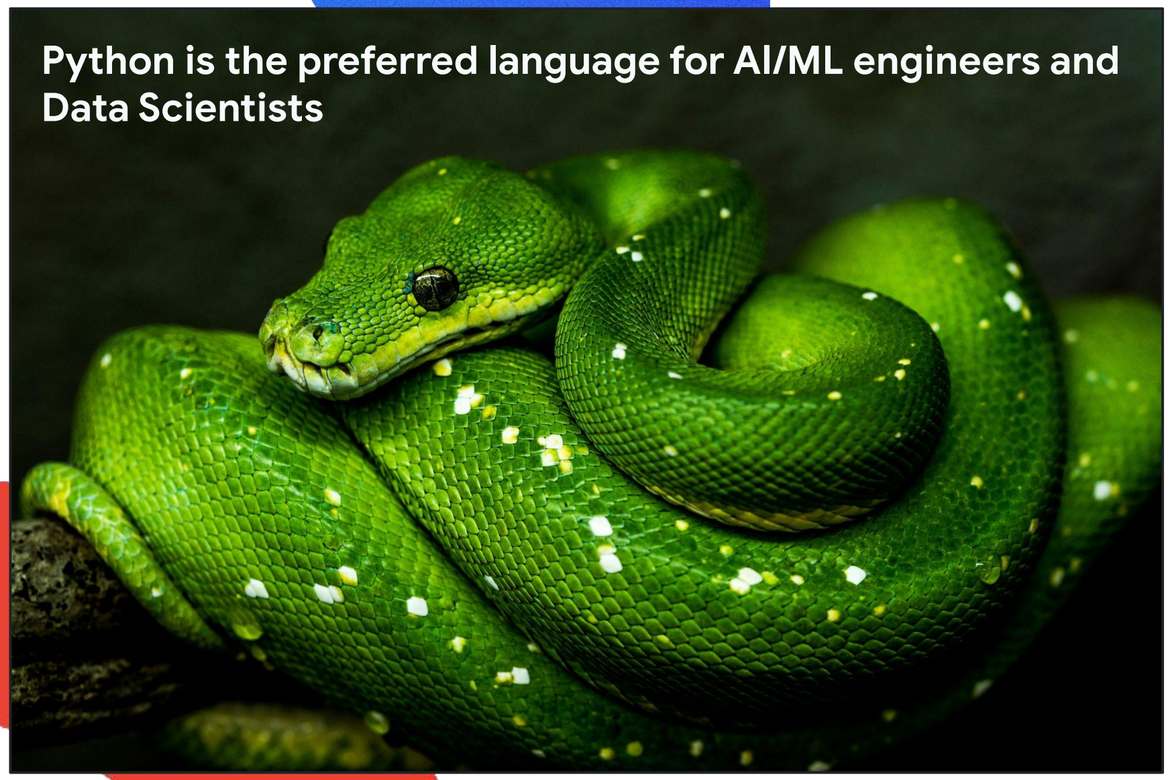


Technology Stack

High-level architecture



Python is the preferred language for AI/ML engineers and Data Scientists



Java is the dominant language of the enterprise

Why Java with Gen AI

Reduced upskilling

Dev Teams with existing Java expertise enjoy a gentle learning curve in building Generative AI apps

Integration with Gen AI

Rich ecosystem of Java libraries and frameworks, expanding with SDKs from model creators and new frameworks

Performance and scalability

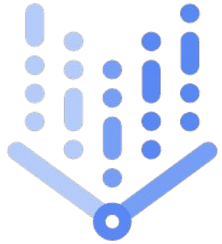
Excellent performance with modern JVMs, tools and optimizations.

Concurrency? Perfect for Gen AI Apps!

Enterprise grade

Reliability, security and focus on large-scale apps with deployment portability in the hybrid cloud

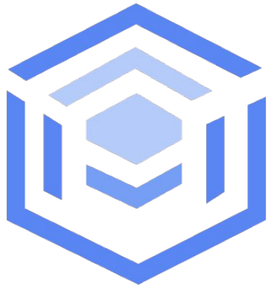
Vertex AI Java SDK



Everything you need to build Java generative AI apps

- Generate text and images
- Use the Gemini language multimodal models to generate text from image or text input.
- Designed for developers and enterprises for use in scaled deployments
- Offers features such as enterprise security, data residency, performance and support

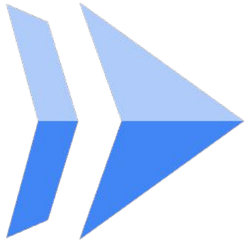
AlloyDB AI: AlloyDB + Gen AI



Enterprise generative AI apps with AlloyDB AI

- Familiar PostgreSQL interface for dev work across data, vectors and models in Java
- Optimized for enterprise Gen AI apps with real-time and high accuracy needs in mind
- Simplified database management experience in the enterprise
- Enterprise-level scalability, availability, and security

Cloud Run



Google Cloud's Serverless Engine

- Build fast-scaling, scale-to-zero, API endpoints to serve requests
- Portable containers run your Java Gen AI apps, interoperable with GKE
- Pay only when your code is running
- Idiomatic to developers, with high deployment velocity

Function calling

Use functions as tools in Gen AI apps to provide added information to LLMs

- Supported in the Vertex AI Java SDK by Gemini models
- Developers create function descriptions and provide them to the model in a request
- Function calling returns a structured data object function in JSON format and args
- Developer uses the data to call the function and provides the response back to the model

Build, test, deploy, test again

LangChain4J

- Simplifies integration of AI/LLM capabilities into Java apps
- Combines ideas from LangChain, LlamaIndex and wider community
- Unified APIs across LLM providers and embedding (vector) stores
- Comprehensive Toolbox for prompting, templating, RAG, etc
- Growing sample base for easier developer onboarding

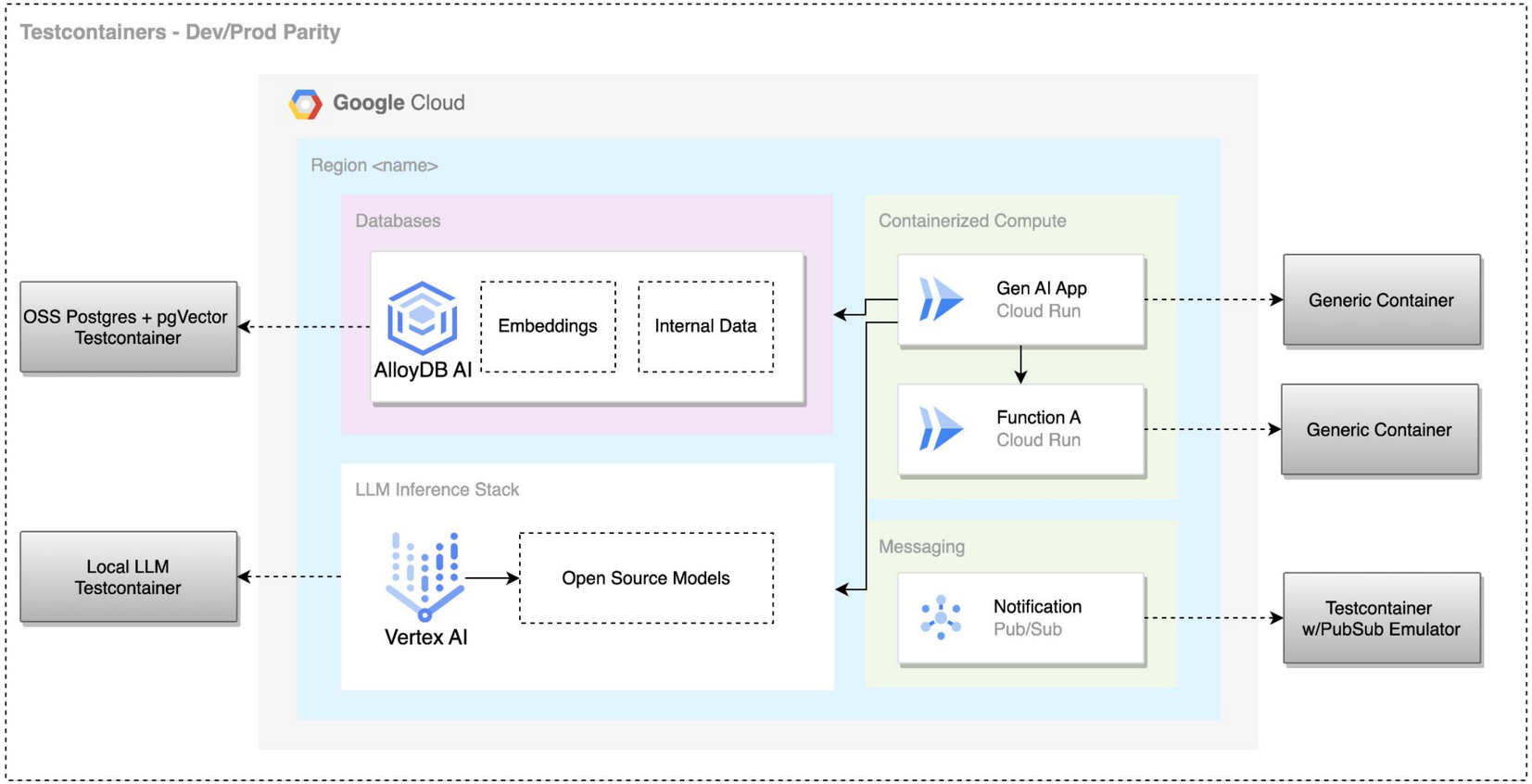
Spring AI

- Reduce complexity of AI/LLM functionality integration with Java
- Draws inspiration from LangChain, LlamaIndex as well
- Existing Spring expertise minimizes developers learning curve
- Robustness and Scalability battle-tested in Production
- Huge developer ecosystem for enterprise-level applications

15-Factor apps for Gen AI

Testcontainers

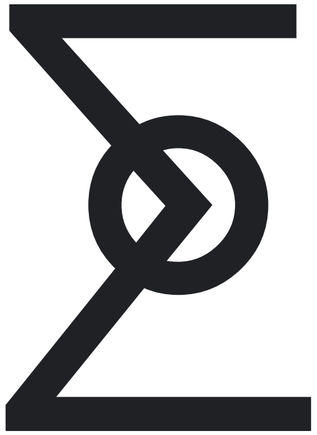
Test your dependencies with disposable, lightweight instances of database, message brokers, web browser, LLM containers

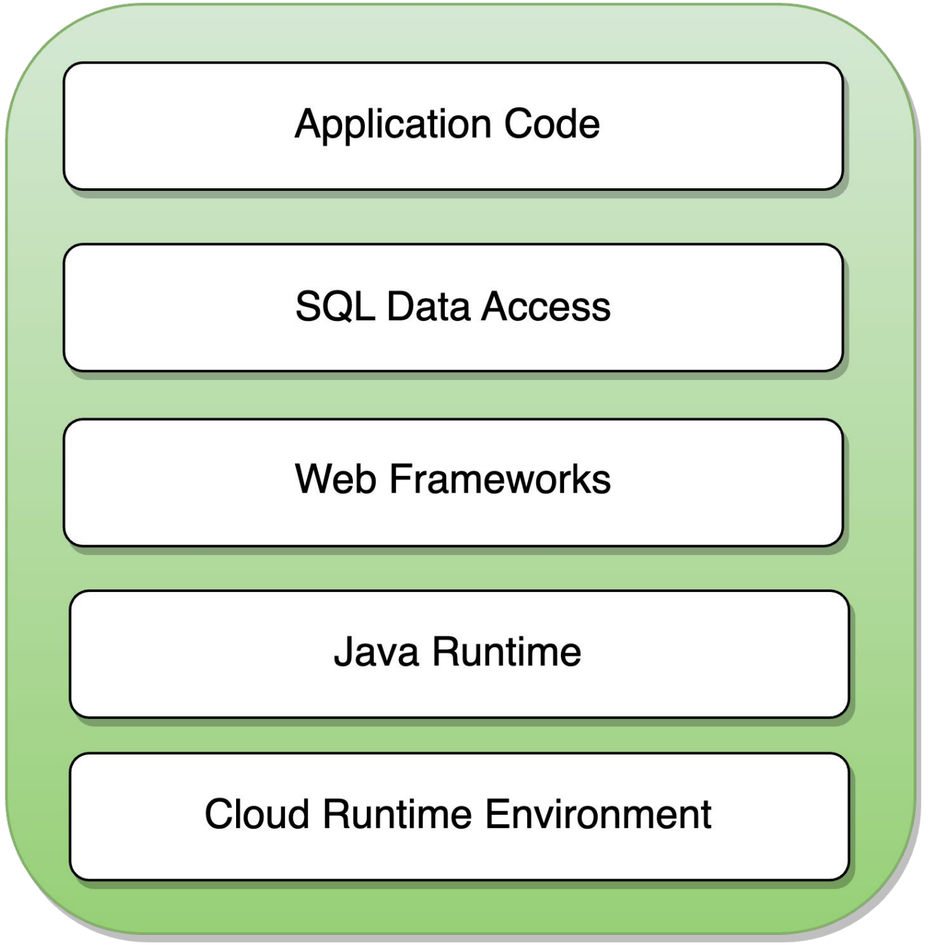


Gen AI app optimization

The Idea

Basic Arithmetics

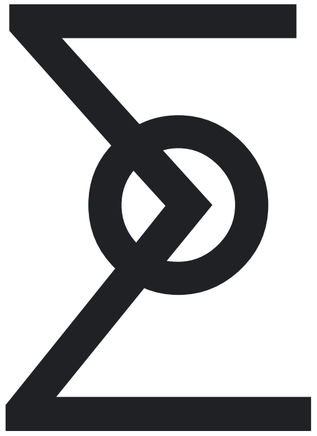


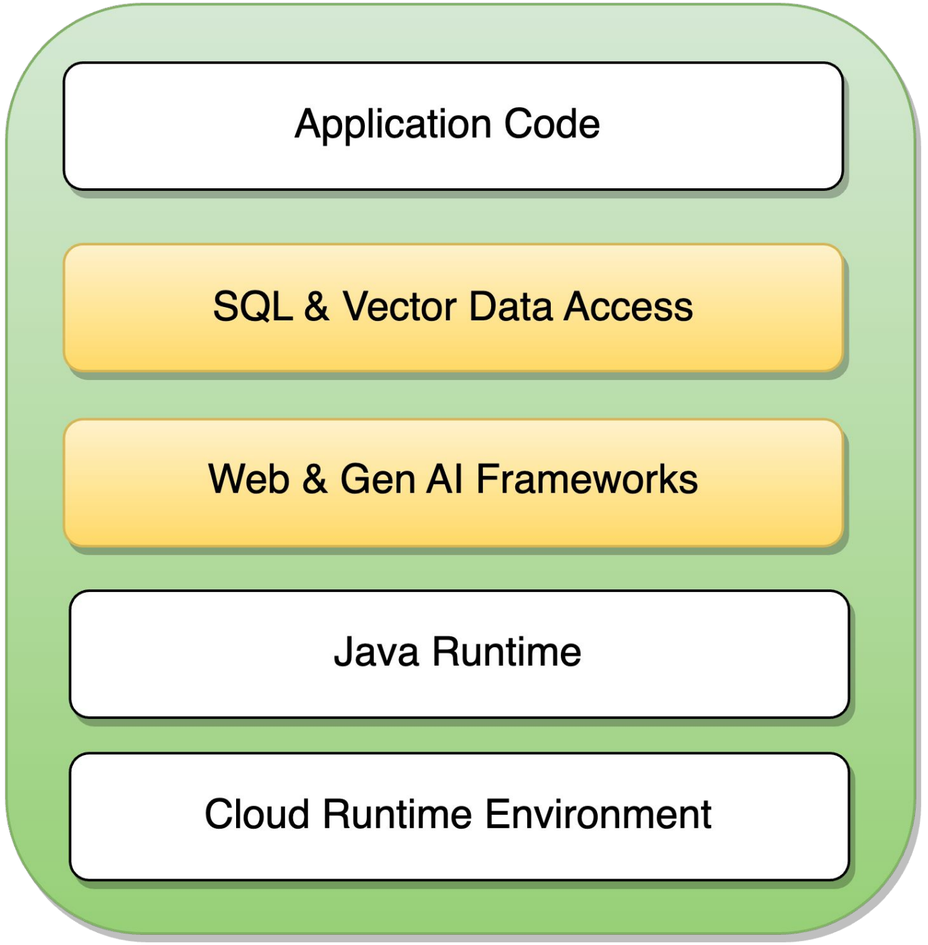


Gen AI app optimization

The Idea

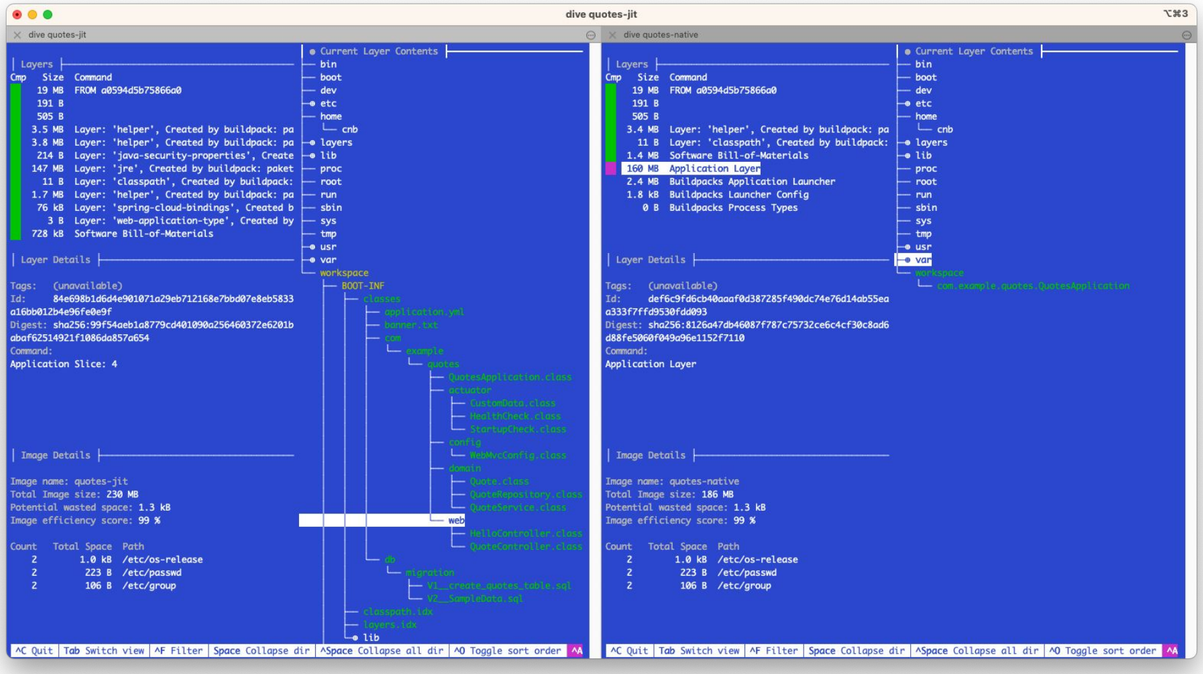
Basic Arithmetics





Native Java meets Gen AI

- Self-contained executable Java apps
- Run without the need of a JVM
- Super-fast startup time
- Peak performance from the first request
- Lower CPU and memory usage



Takeaways

1. Business needs drive Gen AI technology adoption!

2. Experienced with Java? Build GenAI apps with Java!

3. Java + Gen AI in Production? You can do it today!

Ready to build what's next?

Tap into **special offers** designed to help you **implement what you learned** at Google Cloud Next.

Scan the code to receive personalized guidance from one of our experts.

Or visit **g.co/next/24offers**


Thank you