Use Case: RAG with PDF - Google Cloud Reference Architecture

Let's create a reference architecture for a document query system (RAG-based GenAI system) on the Google Cloud platform. The system, which currently processes and analyses a PDF about the impact of the Indian Premier League on Test cricket in India, will be reimagined using Google Cloud services including VertexAI to improve scalability, performance, and cost-effectiveness.

Notebook of RAG with PDF standalone Use Case

Key Google Cloud Services and Migration Strategy:

1. Document Storage:

- Migrate from local PDF storage to **Google Cloud Storage** for secure, scalable document storage.

2. Text Processing and Chunking:

- Utilize **Cloud Run Functions** to handle PDF parsing and text chunking, triggered by Cloud Storage events when new documents are uploaded.

3. Text Embedding:

- Use **VertexAI Text Embeddings API or Huggingface based Embedding Model deployed as VertexAI Endpoint** for generating embeddings, offering efficient and cost-effective text embedding generation without managing infrastructure.

4. Vector Database:

- Replace the local FAISS implementation with **Vertex AI Vector Search** (formerly Matching Engine) for efficient similarity search at scale.

5. Large Language Model (LLM):

- Use **VertexAI Gemini 1.5/1.0 Pro** for text generation.

6. Question & Answer Pipeline:

- Implement the retrieval-augmented generation process using Python, LangChain or LlamaIndex with **Cloud Run Functions** to orchestrate the workflow between Vector Search, Text Embeddings API, Gemini 1.5 Pro, and other Google Cloud services.

7. API Layer:

- Create an API using **Cloud API Gateway** and **Cloud Run Functions** to handle user requests and responses.

8. Front-end and Load Balancing:

- Implement **Cloud Load Balancing** for global load distribution and **Cloud CDN** for content delivery optimization.

9. Authentication and Authorization:

- Use **Cloud Identity Platform** for secure user authentication and authorization.

10. Data Processing Pipeline:

- Use **Dataflow** for ETL processes
- **Cloud Run Functions** for Data Fetching & Embedding generation

11. Monitoring and Analytics:

- Implement **Cloud Monitoring**, **Cloud Logging**, and **Looker** for comprehensive monitoring, logging, and analytics.

12. Security and Compliance:

- Leverage **Cloud KMS** for secret management
- **Cloud Armor** for web application firewall and DDoS protection
- **Cloud DLP** (Data Loss Prevention) for sensitive data handling

13. Scalability and Reliability:

- Utilize **Cloud Run** for serverless container deployment
- **Cloud Load Balancing** with multiple regions
- **Cloud Storage** with multi-region configuration for high availability

Additional Considerations:

Development and Testing:

- Use **Cloud Workstations** for development environments
- **Cloud Build** for CI/CD pipelines
- **Artifact Registry** for container image storage

Cost Optimization:

- Implement appropriate pricing tiers for VertexAI services
- Use Cloud Functions and Cloud Run for serverless compute to optimize costs
- Configure caching strategies to minimize API calls

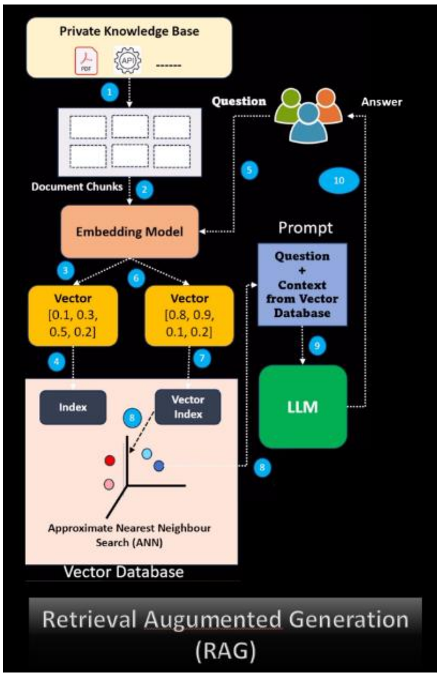
Best Practices:

1. Implement retry logic for API calls
2. Use batch processing for large document sets
3. Implement proper error handling and monitoring
4. Set up appropriate IAM roles and permissions
5. Regular backup and disaster recovery planning

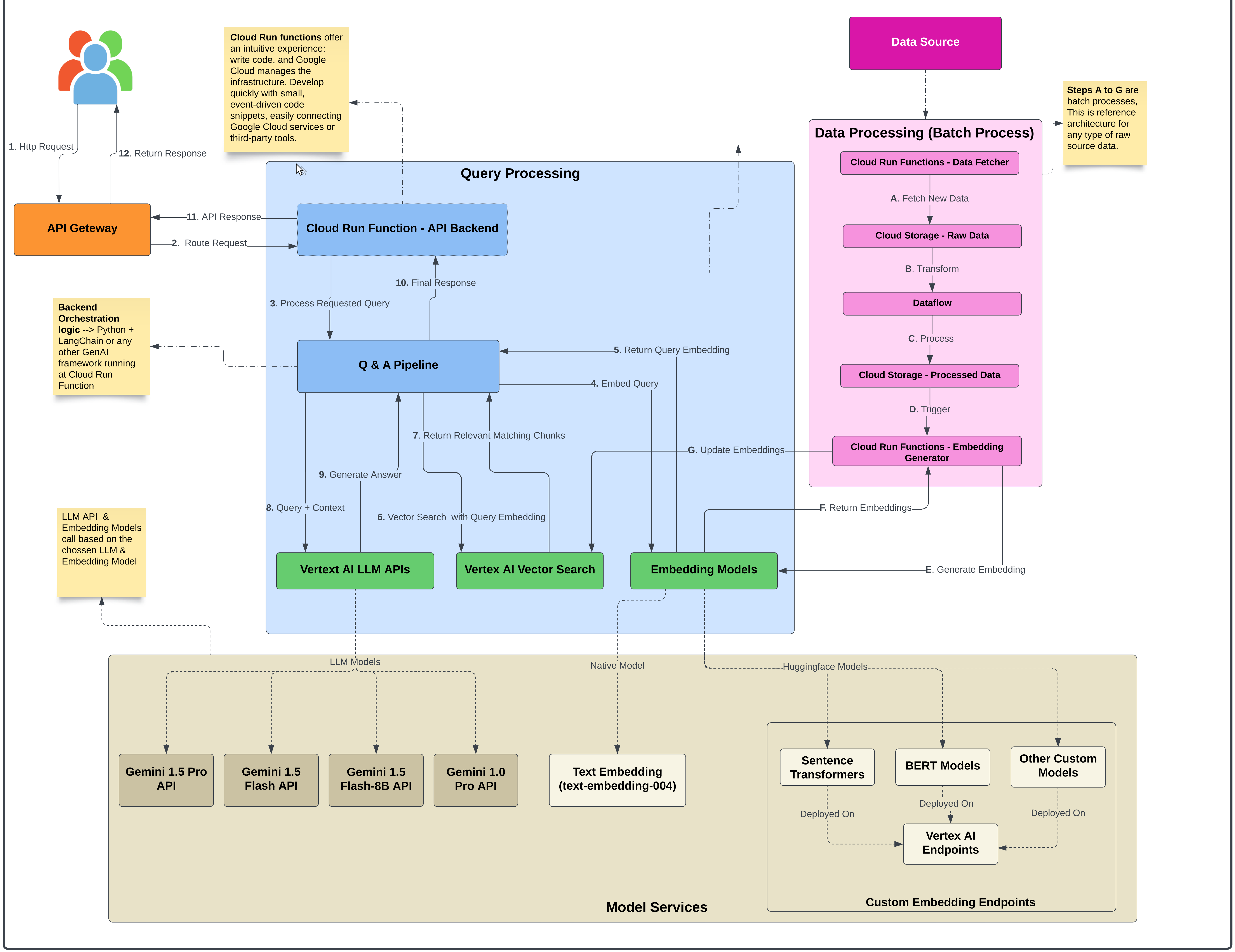
This Google Cloud migration will transform the solution into a cloud-native, serverless architecture, offering:

- Better scalability through managed services
- Enhanced security with Cloud-native security tools
- Cost optimization through pay-as-you-go pricing
- Reduced operational overhead
- Integrated AI capabilities through Vertex AI
- Simplified management and monitoring

The architecture leverages Google Cloud's comprehensive set of AI and machine learning services, particularly VertexAI, which provides a unified platform for both traditional ML and modern GenAI applications. This allows teams to focus on improving the document query system.



GenAI Simplified Reference Architecture: RAG on Google's Vertex AI



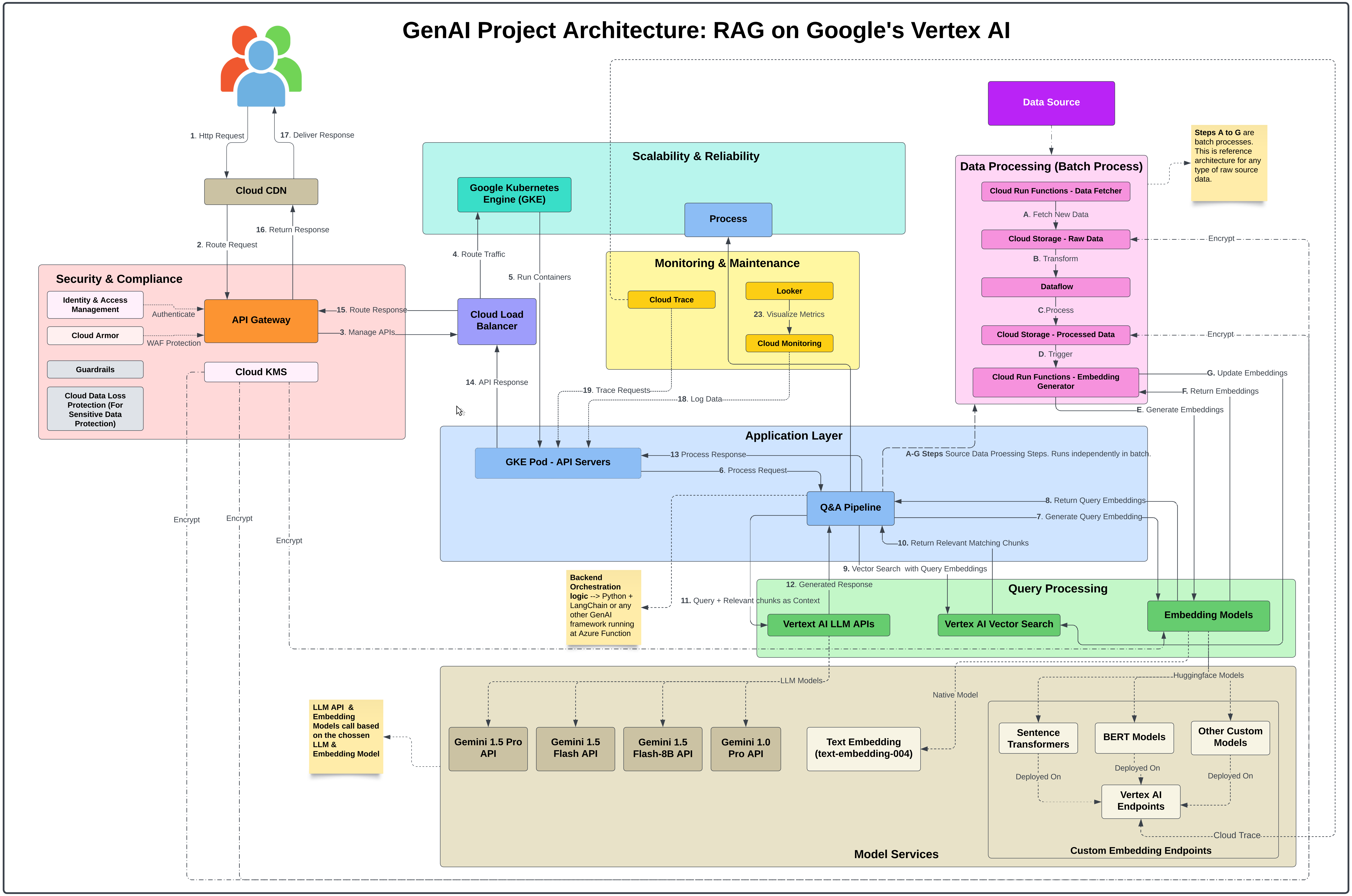
Cloud Run functions offer an intuitive experience: write code, and Google Cloud manages the infrastructure. Develop quickly with small, event-driven code snippets, easily connecting Google Cloud services or third-party tools.

Steps A to G are batch processes, This is reference architecture for any type of raw source data.

Backend Orchestration logic --> Python + LangChain or any other GenAI framework running at Cloud Run Function

LLM API & Embedding Models call based on the chossen LLM & Embedding Model

GenAI Project Architecture: RAG on Google's Vertex AI



Steps A to G are batch processes. This is reference architecture for any type of raw source data.

Backend Orchestration logic --> Python + LangChain or any other GenAI framework running at Azure Function

LLM API & Embedding Models call based on the chossen LLM & Embedding Model