{ DevFest }

GDG Cloud Surabaya

2025

Hands-On: AI Agent Development with ADK and MCP Server on Cloud Run

Ananda Dwi Rahmawati
Google Developer Expert - Cloud

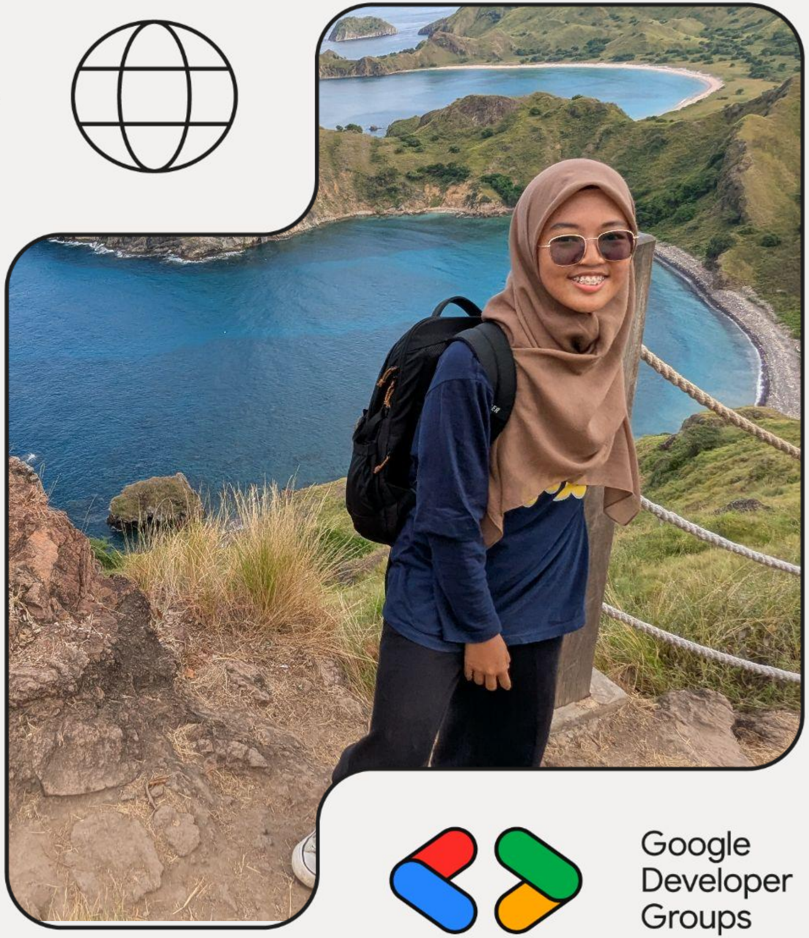
Google Developer Groups

{ DevFest }

GDG Cloud Surabaya

Ananda Dwi Rahmawati

- ❏ Cloud & DevOps Engineer, Singapore
- ❏ Google Developer Expert Cloud - Modern Architecture
- ❏ Master of Computer Science - University of Texas at Austin
- ❏ **go.gov.sg/misskecupbung**



Google Developer Groups

Prerequisites

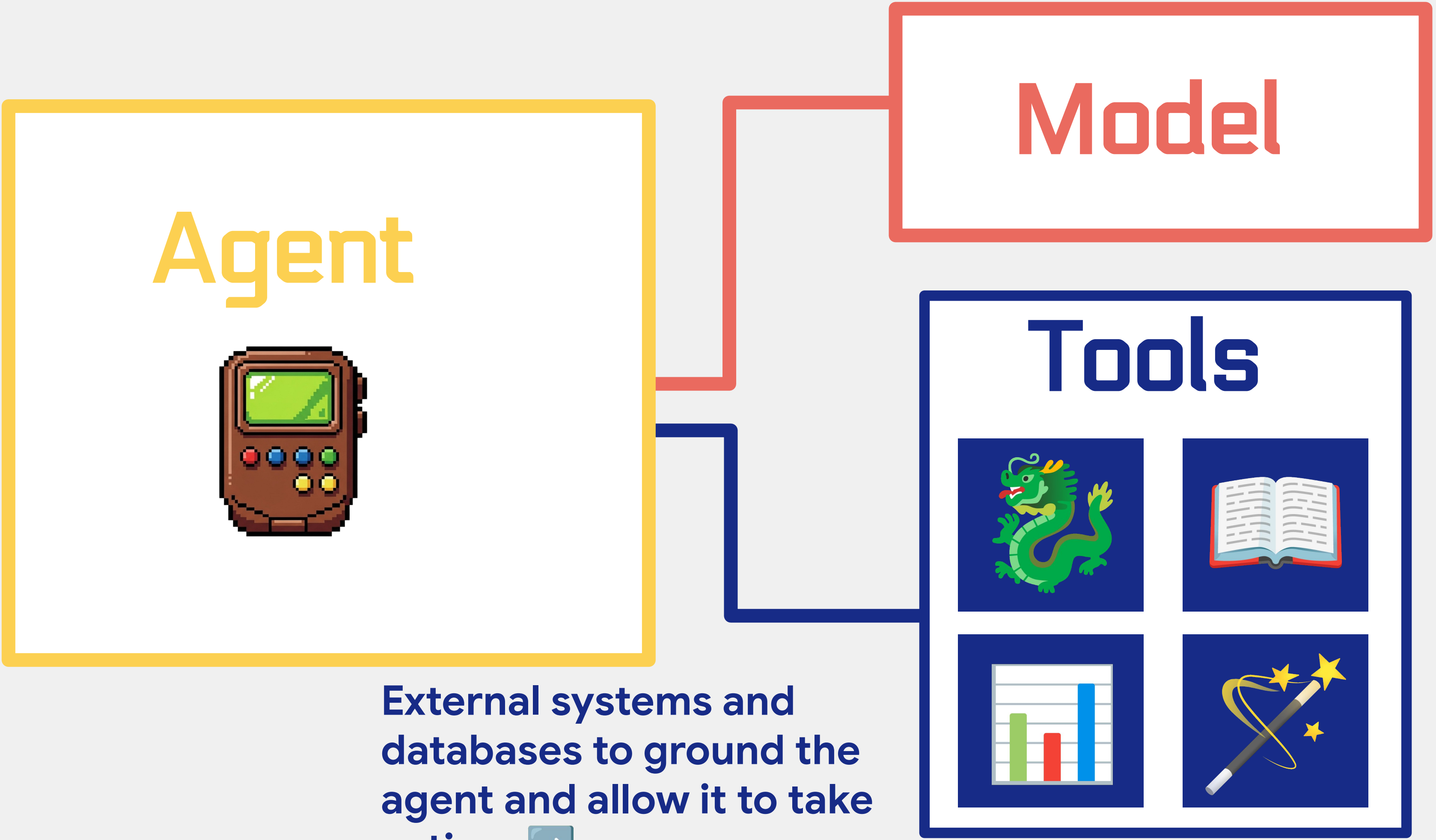
- A running MCP server on Cloud Run or its associated Service URL.
- A Google Cloud project with billing enabled.

What you'll learn

- How to structure a Python project for ADK deployment.
- How to implement a tool-using agent with google-adk.
- How to connect an agent to a remote MCP server for its toolset.
- How to deploy a Python application as a serverless container to Cloud Run.
- How to configure secure, service-to-service authentication using IAM roles.
- How to delete Cloud resources to avoid incurring future costs.

What you'll need

- A Google Cloud Account and Google Cloud Project
- A web browser such as Chrome



External systems and databases to ground the agent and allow it to take action. ➡️

Why build an agent?

Flexible

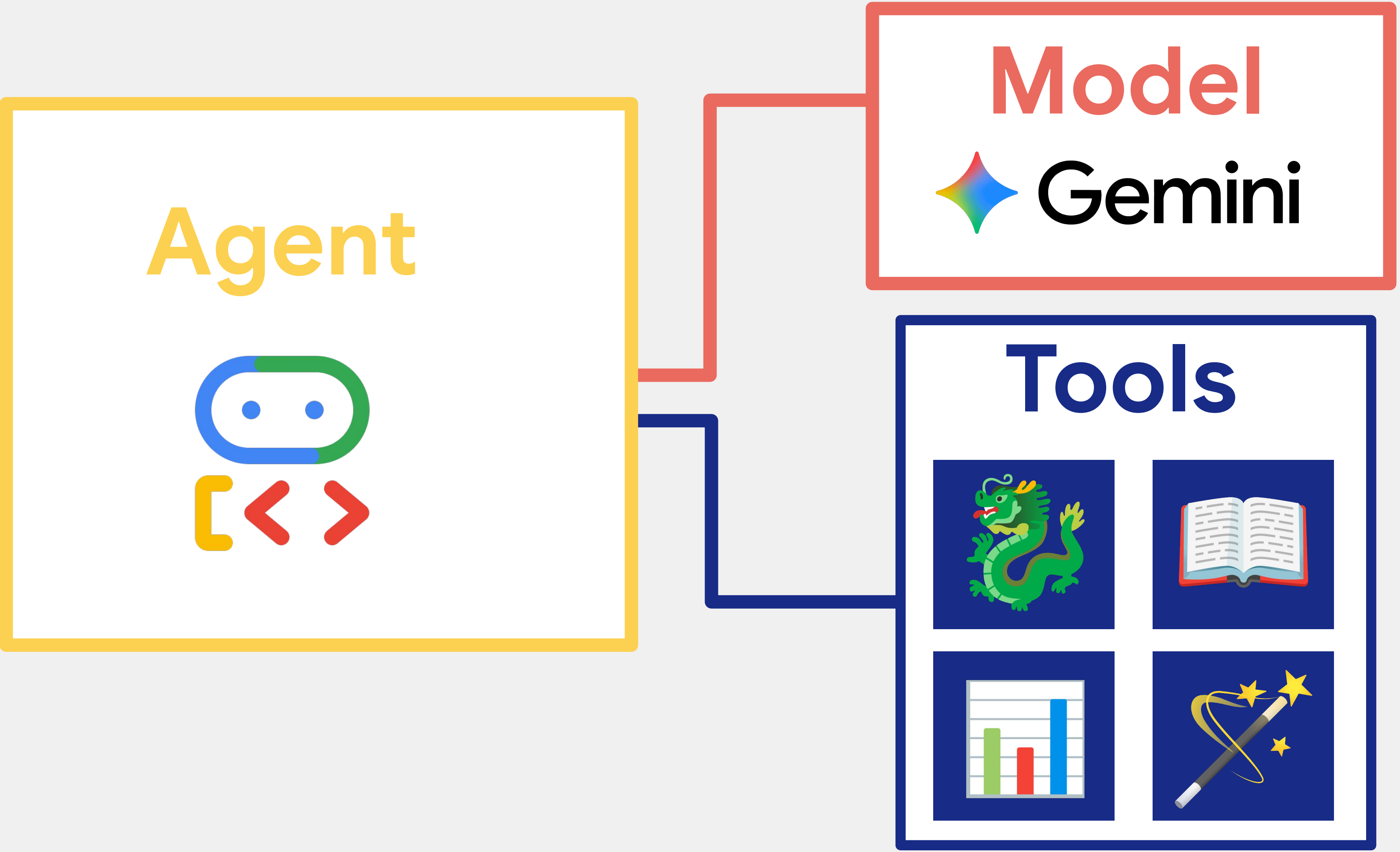
Can **reason** based on personalized inputs, handle unexpected **edge cases**, and respond in **natural language. Outcome-driven** rather than path-driven.

Easy to Prototype

Reduced dev time **integrating** with external APIs and databases, and in building deterministic, “if this then that” systems.

Proactive

Can run an agent in response to a **user prompt,** on a **trigger** or **schedule** (autonomously).



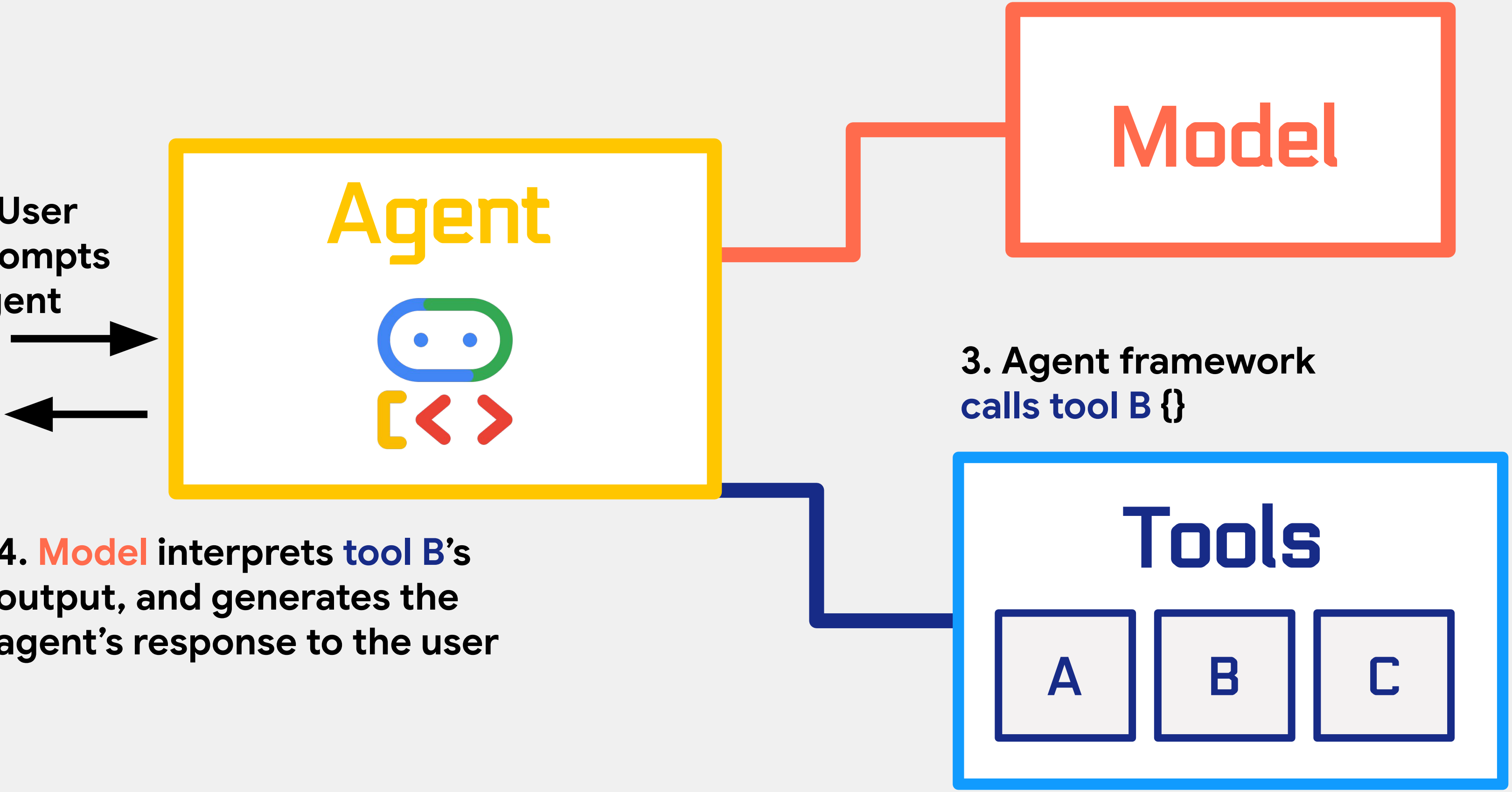
Agent

Model

Gemini

Tools

How do tools work?



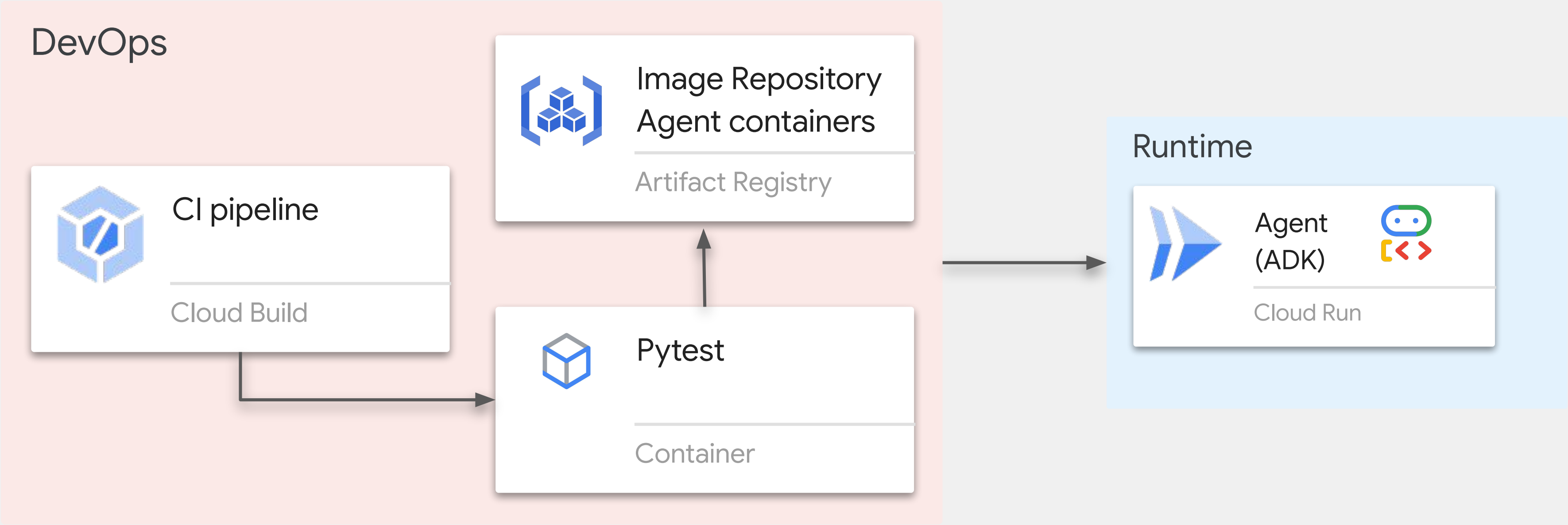
1. User prompts agent

2. Model selects tool B and formats tool request body {} ("function-calling")

3. Agent framework calls tool B {}

4. Model interprets tool B's output, and generates the agent's response to the user

Deploy your agent to Google Cloud



DevOps

CI pipeline

Cloud Build

Image Repository
Agent containers

Artifact Registry

Pytest

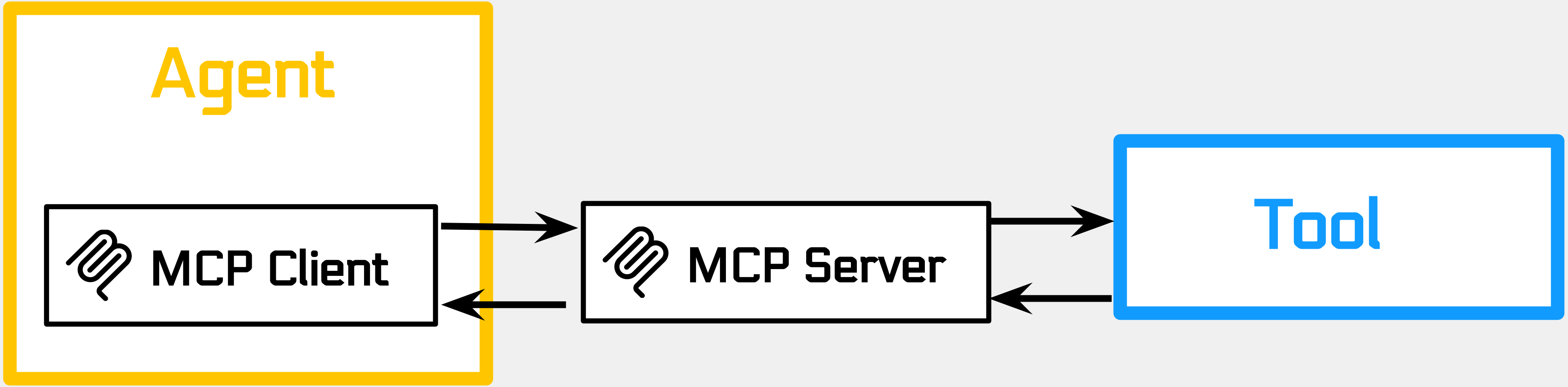
Container

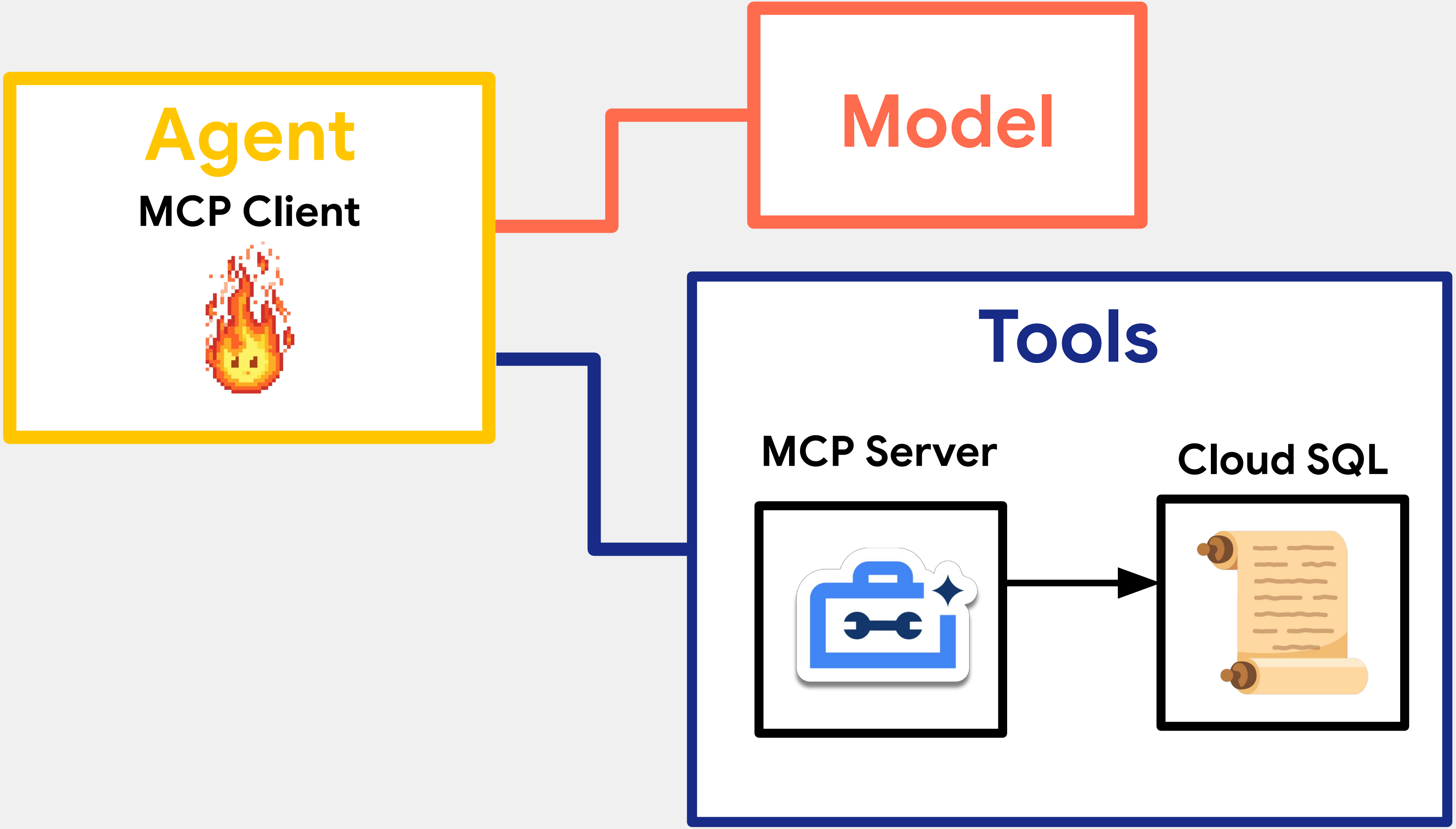
Runtime

Agent
(ADK)

Cloud Run

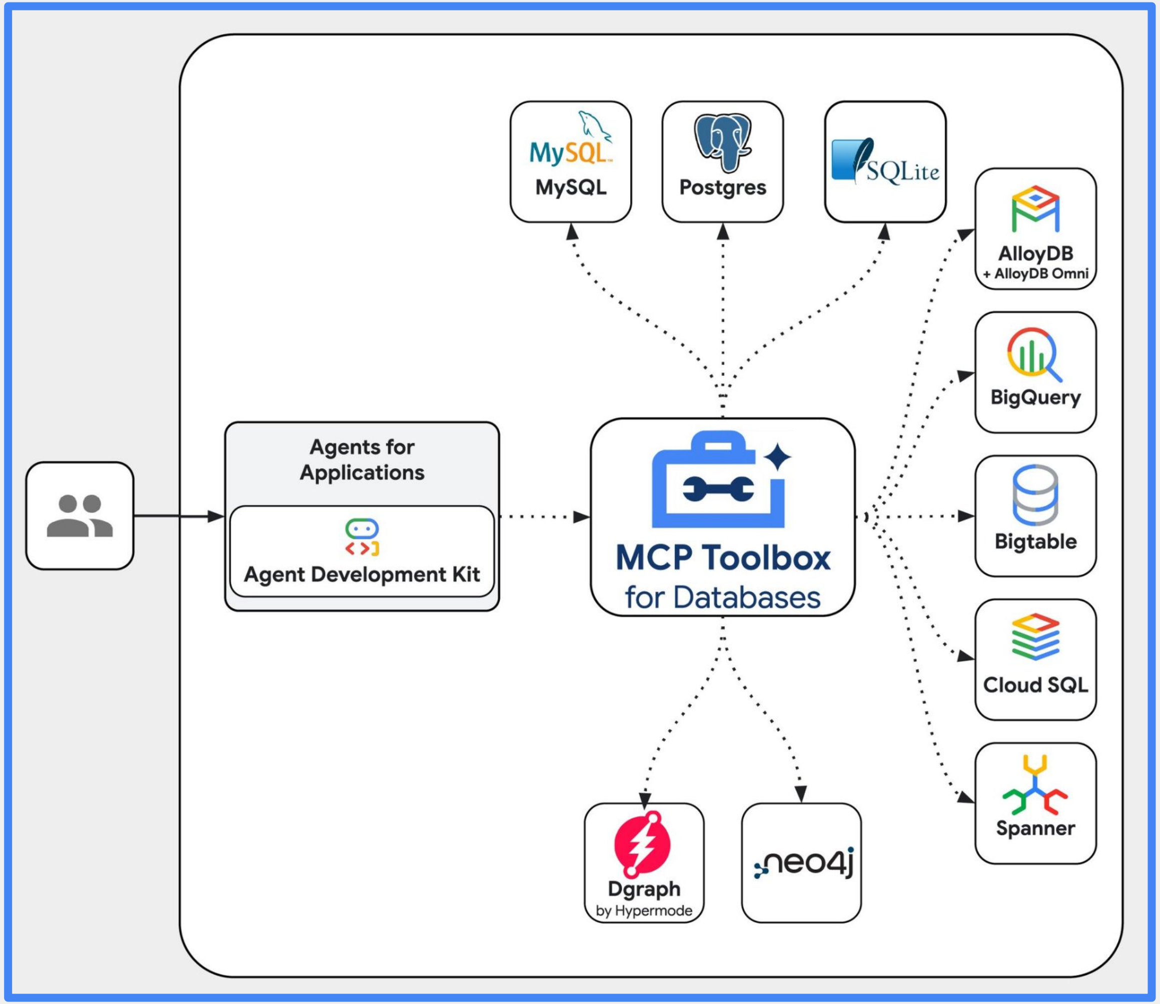
How does MCP work?




Agent
MCP Client
Model
Tools
MCP Server
Cloud SQL

The **MCP Toolbox for Databases** is a Google-built, open-source MCP server for databases.

It handles specialized database connection pooling, DB auth, and observability with OpenTelemetry.



What does it take to operate agents?

Choose foundational infrastructure (eg. models)

Choose and manage frameworks

Build CI/CD pipelines for agents, models, and tools

Protect agents against malicious attacks

Observe agents in production

Manage auth + access into external- and internal-facing agents

An integrated & secure path to production for container workloads

Develop

- Cloud Workstations
- Code Assist
- Cloud Code

Mobile & Web Dev

- Firebase

Migrate

- Migrate to Containers

Deploy

- Cloud Build
- Cloud Deploy
- Artifact Registry

Run

- **Kubernetes Engine**
- **Cloud Run**
- **GCE**

Operate

- Cloud Logging
- Cloud Monitoring

Orchestrate

- Workflows
- Service Mesh
- Eventarc

Data Store

- AlloyDB
- Cloud SQL
- Cloud Storage
- Cloud Spanner

and more ...

Network

API GW

Data analytics

AI / ML Vertex AI

Security and identity

Gemini
Code Assist

Google Cloud Serverless Offering: Cloud Run

Any Language

Any Framework

Any Library

Ideal runtime environment for AI app development

- Serverless GPUs
- Scalable API endpoint for Agentic AI apps (e.g. AI agents, MCP servers)

Enterprise-grade platform supporting diverse workloads

- Scalable, secure, cost optimized, highly available
- A wide selection of workloads supported

Developer-centric platform designed for maximum ease-of-use and developer velocity

- Focus on code not infrastructure
- Fast deployment, easy integration with other Cloud services, language and framework agnostic
- Open and portable

Benefits of Cloud Run

Higher Velocity & Productivity.
Cloud Run allows developers to spend more time writing code and less time managing infrastructure.

Higher Reliability.
Cloud Run is redundant by default. Google is your SRE.

Lower Cost.
Cloud Run autoscales to meet your needs and scales to zero. Pay only for what you use.

95% faster deployment than legacy platforms

98% fewer interruptions to service

15% - 50% cheaper than provisioned platforms

75% cheaper than on-prem

> Our initial concern about choosing serverless was cost.
>
> It turns out that using **Cloud Run is significantly more cost-effective than running the number of VMs** we would need for a system that could survive reasonable traffic spikes with a similar level of confidence.

BBC

FORRESTER

Access study

Google Cloud

Cloud Run

Fully managed platform to run your code on top of Google's scalable infrastructure

Experience

- **Simple**
  Demand as little as possible.
- **Automated**
  Cloud Run takes care of a lot for you.
- **Top satisfaction and usability scores**
  Highest CSAT and task success.
- **Developer productivity**
  Idiomatic to developers, deployment velocity.

Runtime

- **Capable**
  Run any code, any container
- **On-demand**
  No pre-provisioning
- **Hyper-elastic**
  Scales automatically **fast** (0 to 10,000 containers in 10s). Scales to **zero**
- **No infrastructure management**
  No VM or cluster to manage

Pricing

Pay only when code is running, with a 100ms granularity.

- CPU
- Memory
- Requests (not always)

Perpetual monthly free tier

Flexible Committed Use Discounts

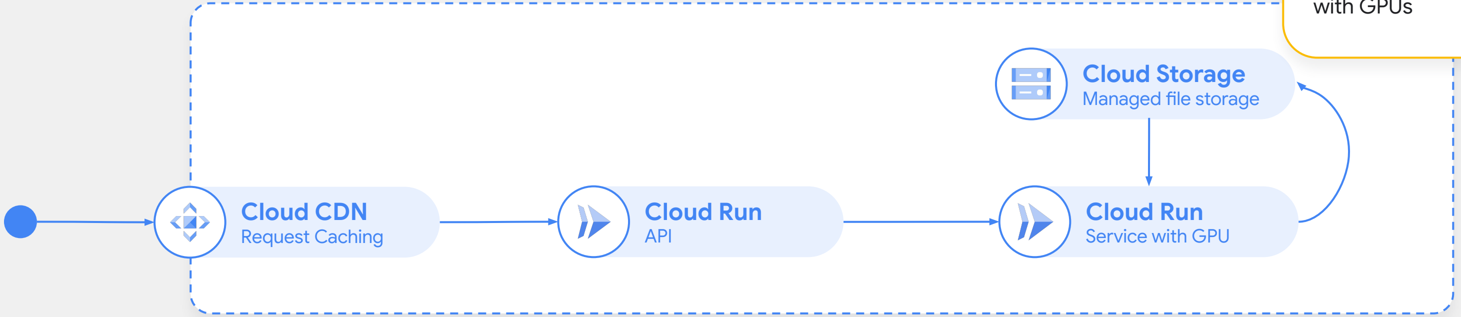
No flat fee!
"if you don't use it, you don't pay for it"

Design Patterns

On-demand AI inference with GPUs

Cloud Run Advantage

On-demand inference with <5s fast startup, autoscaling and scale to zero with GPUs



Use **Cloud Run** as a scalable API endpoint to serve requests.

Run LangChain or custom code validating requests and orchestrate calls to models

A **GPU attached Cloud Run service** runs LLM e.g. Gemma2 and fetches the LLM weights from **Cloud Storage** over a VPC with **Direct VPC egress**

Cloud Run GPU service performs inference such as text-to-text, text-to-image, send back the response via frontend.

AI agents plan, reason, and execute tasks for users

Four key components

✓ **Model(s)**

Used to reason over goals, determine the plan and generate a response.

✓ **Tools**

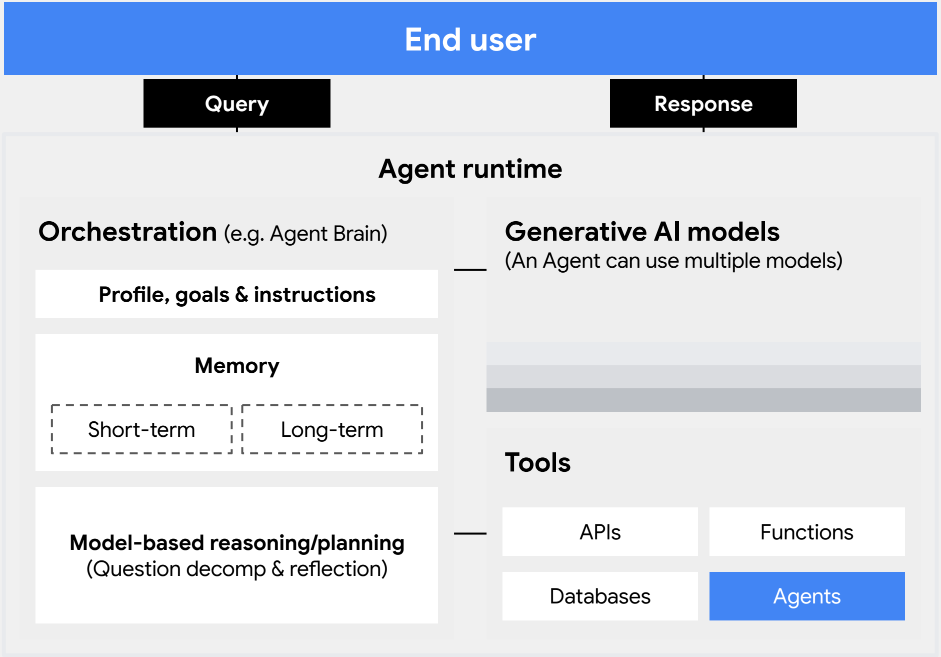
Fetch data, perform actions or transactions by calling other APIs or services.

✓ **Orchestration**

Maintain memory and state (including the approach used to plan), tools, data provided/fetched, etc.

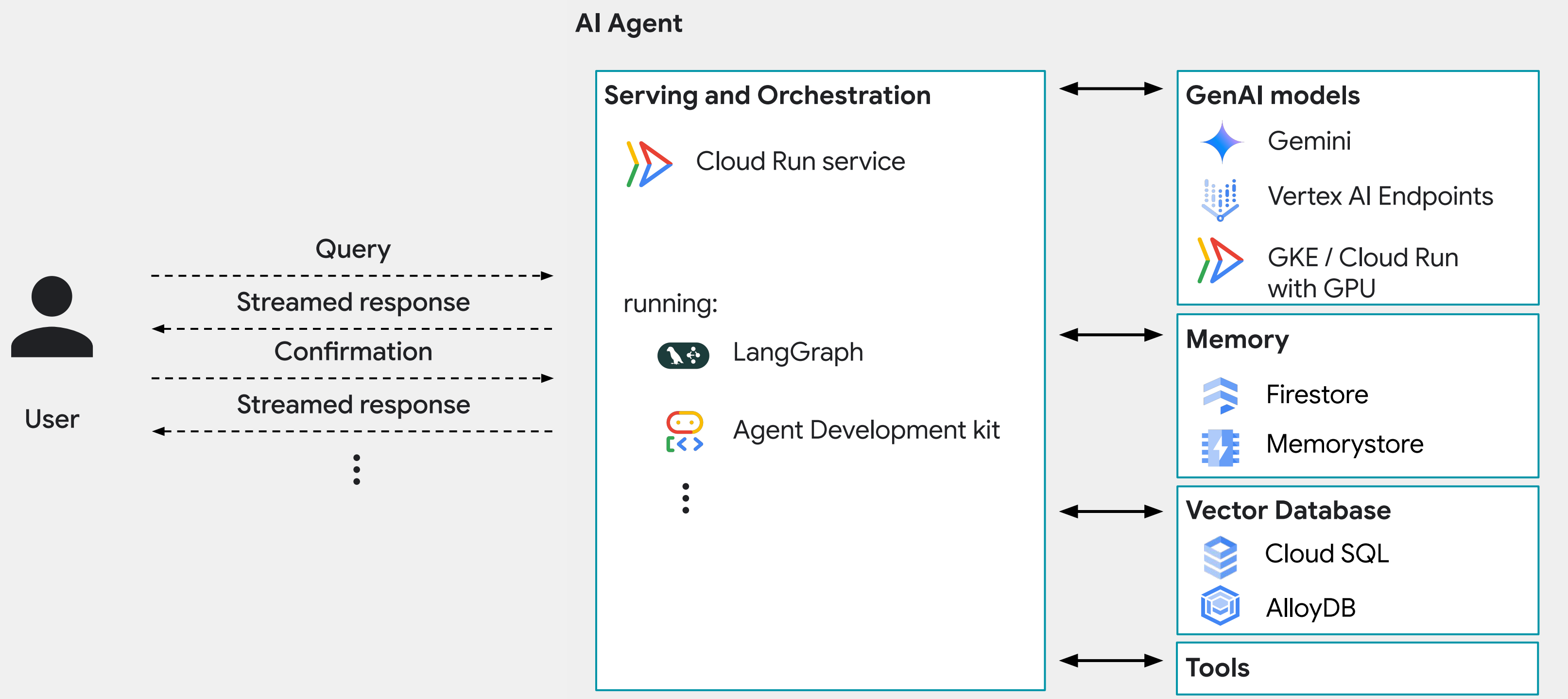
✓ **Runtime**

Execute the system when invoked.


End user
Query
Response
Agent runtime
Orchestration (e.g. Agent Brain)
Profile, goals & instructions
Memory
Short-term
Long-term
Model-based reasoning/planning
(Question decomp & reflection)
Generative AI models
(An Agent can use multiple models)
Tools
APIs
Functions
Databases
Agents

Run AI agents on Cloud Run

("Do It Yourself" on Google Cloud)



AI Agent

User

Query

Streamed response

Confirmation

Streamed response

⋮

Serving and Orchestration

Cloud Run service

running:

LangGraph

Agent Development kit

⋮

GenAI models

Gemini

Vertex AI Endpoints

GKE / Cloud Run with GPU

Memory

Firestore

Memorystore

Vector Database

Cloud SQL

AlloyDB

Tools

AI agents plan, reason, and execute tasks for users

Four key components

✓ **Model(s)**

Used to reason over goals, determine the plan and generate a response.

✓ **Tools**

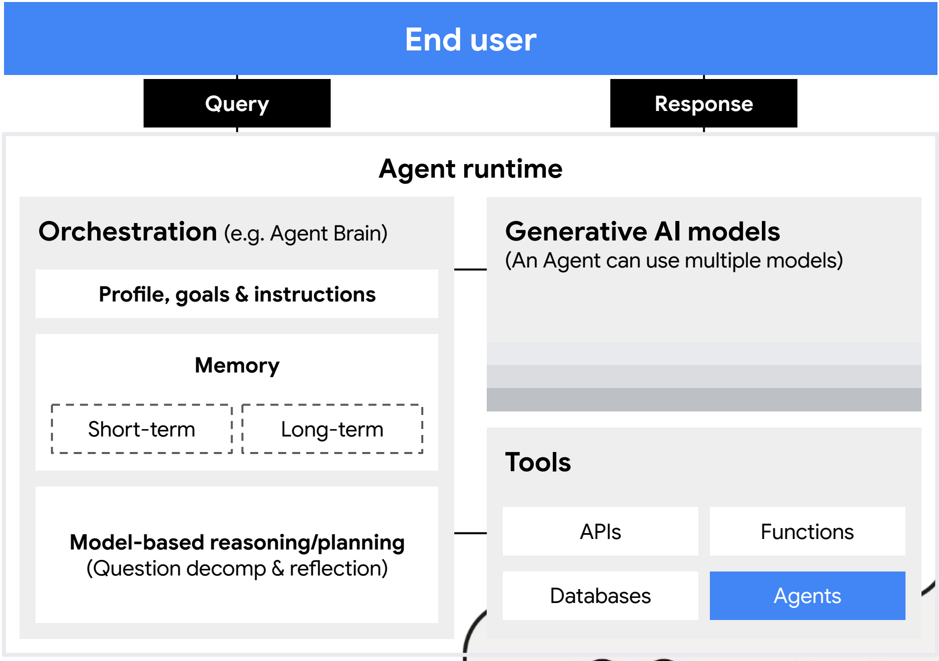
Fetch data, perform actions or transactions by calling other APIs or services.

✓ **Orchestration**

Maintain memory and state (including the approach used to plan), tools, data provided/fetched, etc.

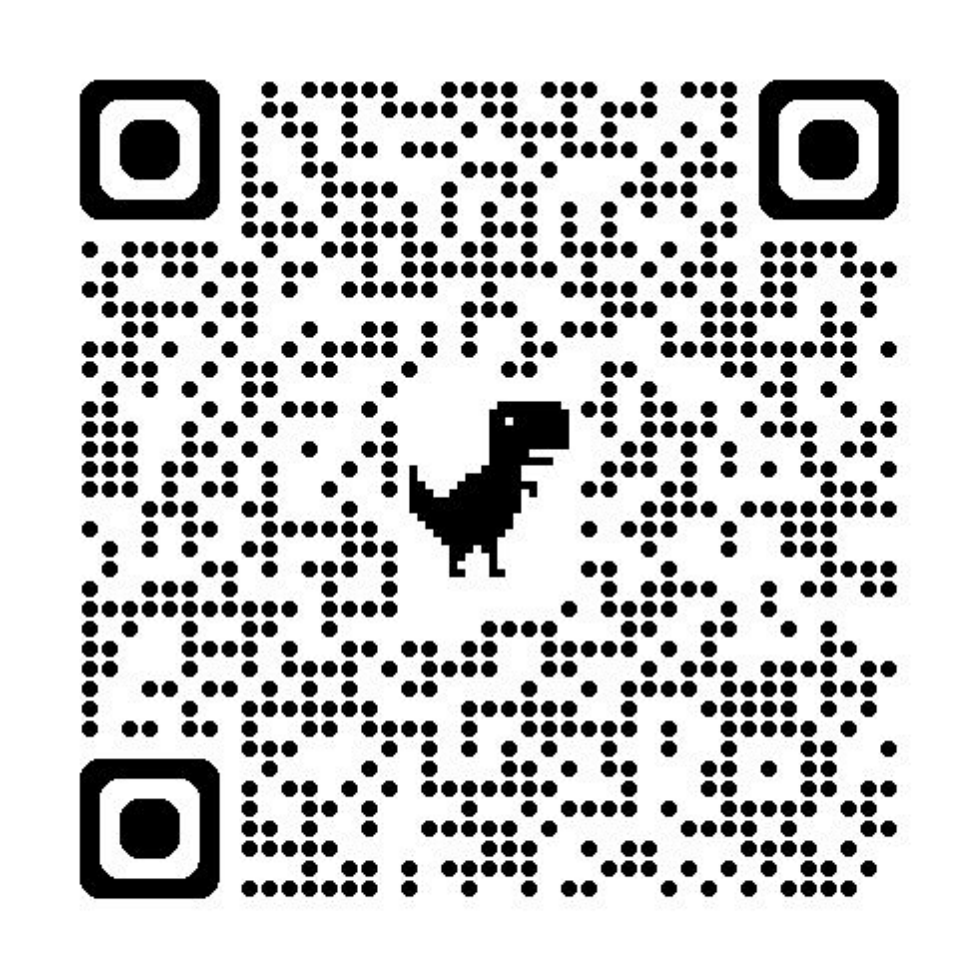
✓ **Runtime**

Execute the system when invoked.


End user
Query
Response
Agent runtime
Orchestration (e.g. Agent Brain)
Profile, goals & instructions
Memory
Short-term
Long-term
Model-based reasoning/planning
(Question decomp & reflection)
Generative AI models
(An Agent can use multiple models)
Tools
APIs
Functions
Databases
Agents

Google Developer Groups

Let's do Hands On Lab!



Thank you