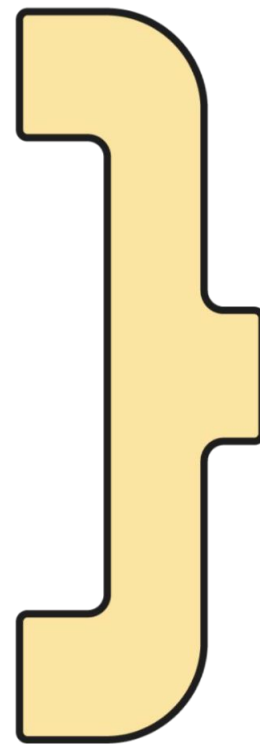


Build with AI



North America

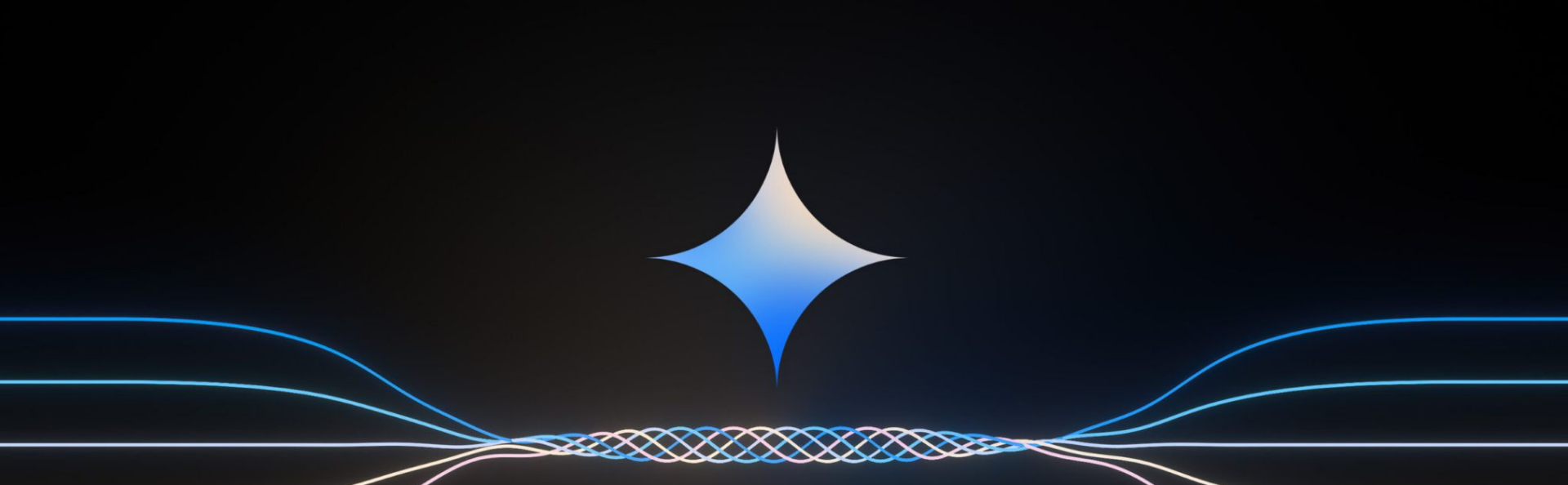
March 2025



Keynote

Ankur Kotwal

Head of Google Cloud Global Advocacy

The background features a dark blue gradient. At the top center is a four-pointed starburst shape with a blue-to-white gradient. Below it, a series of horizontal, wavy lines in blue, yellow, and pinkish-purple extend across the frame, converging towards the center. In the center of this convergence is a small, intricate pattern of interlocking loops in the same colors.

Gemini

State of the art native multimodal LLM

TEXT

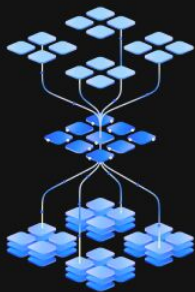
AUDIO

IMAGE

VIDEO

The Gemini 2.0 model family

Our most capable model yet built for the
agentic era.



EXPERIMENTAL

2.0 Pro

Our best model yet for
coding performance and
complex prompts.



GENERAL AVAILABILITY

2.0 Flash

Our powerful workhorse model
with low latency and enhanced
performance, built to power
agentic experiences.



EXPERIMENTAL

2.0 Flash Thinking

Our enhanced reasoning model,
capable of showing its thoughts to
improve performance and
explainability.



PUBLIC PREVIEW

2.0 Flash-Lite

Our most cost-efficient model yet.

Two major shifts in AI in 2024

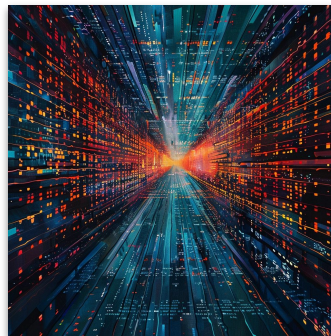
Single modal



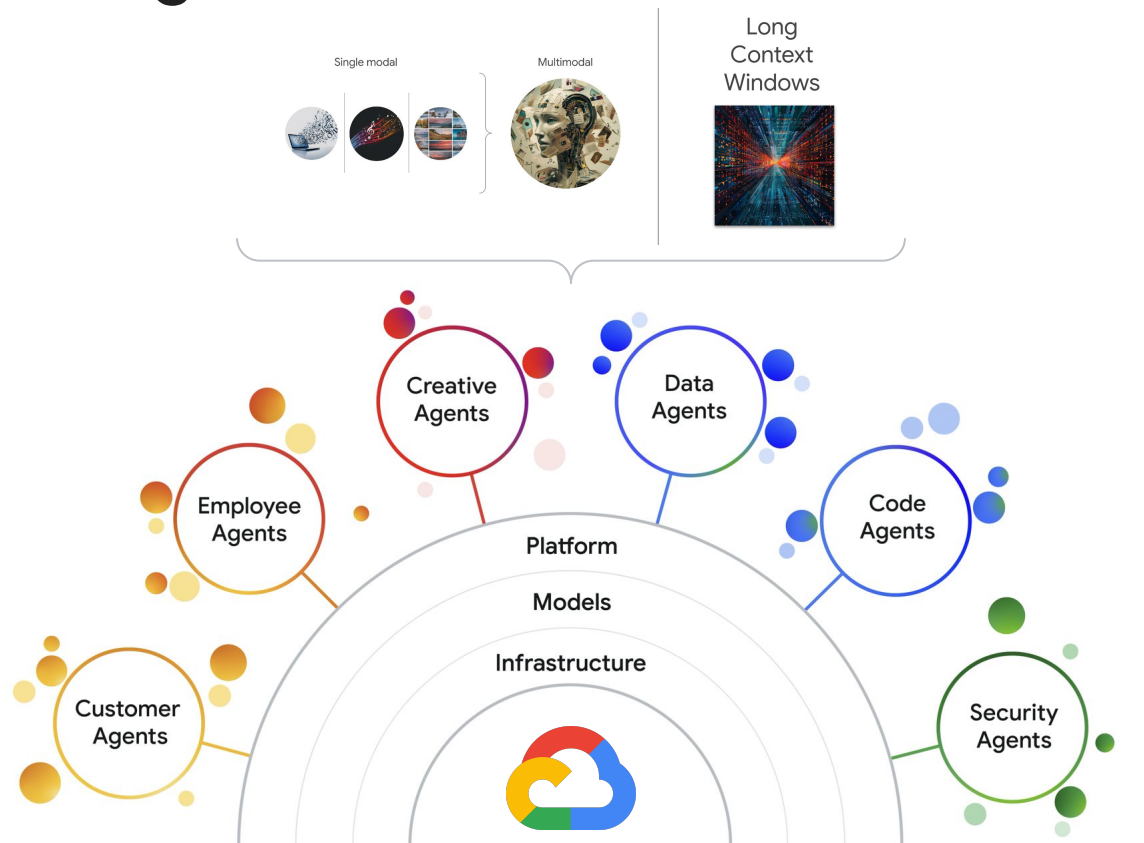
Multimodal



Long
Context
Windows



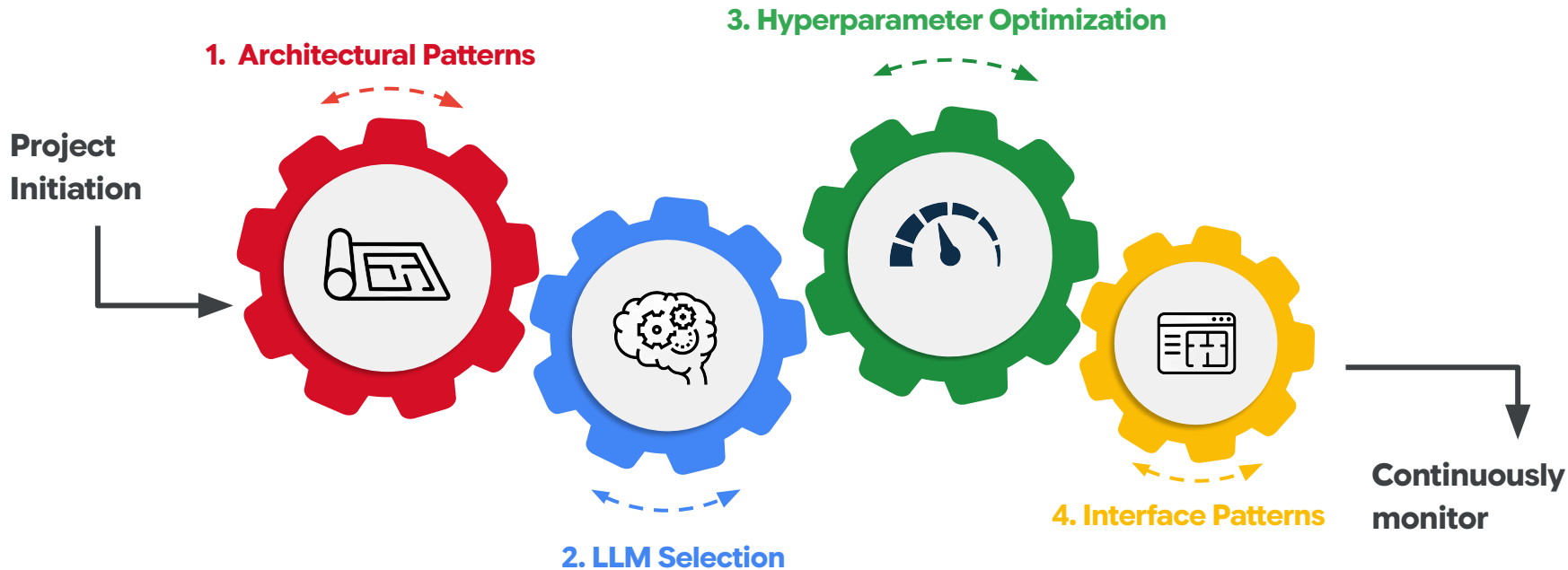
2025 - The Agentic Era



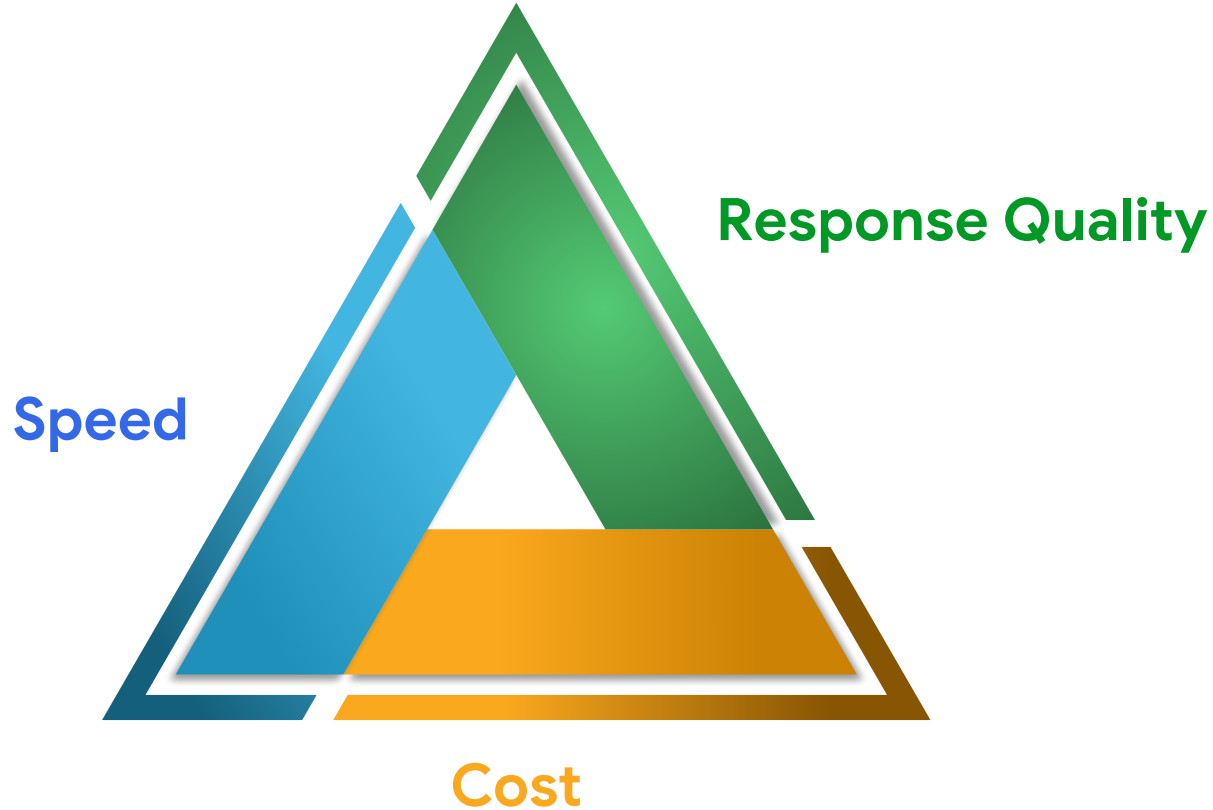
What is an AI Agent?

Autonomous software that can operate independently, make decisions and take actions without constant human intervention

Blueprint for Agentic Success



Constraints Triangle of Agentic systems



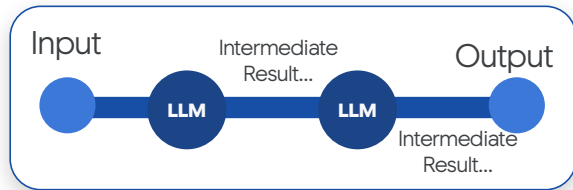
1. Architectural Patterns

Single Model Architecture



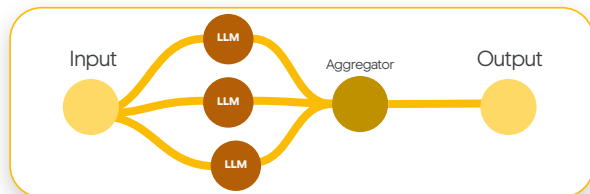
One LLM for all tasks

Sequential Models



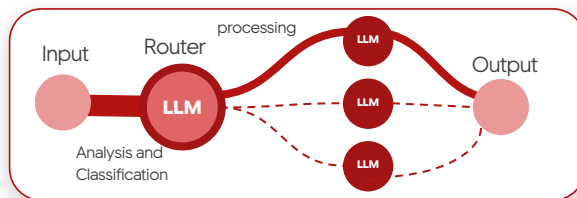
Chains multiple specialized LLMs in a pipeline.

Parallel Models



Multiple LLMs concurrently and aggregate outputs

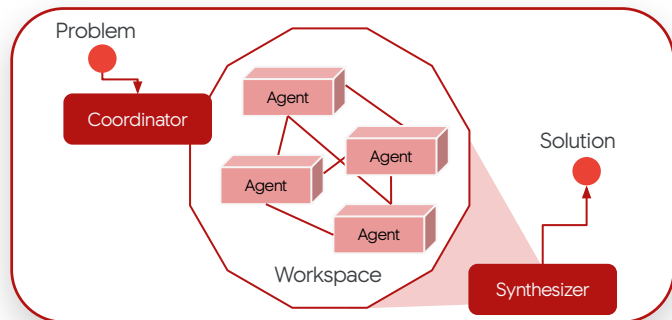
Hierarchical Routing



Router model to direct inputs to specialized LLMs based on content or intent

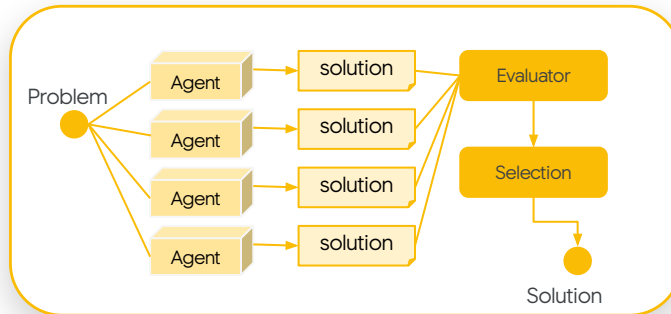
Multi-Agent Interaction

Cooperative Multi-Agent System



Multiple LLM-based agents collaborating to solve problems

Competitive Multi-Agent System



Independent LLM agents to propose competing solutions, with evaluator model selecting best or integrating elements

2. LLM Selection



Model Capabilities

- Task Suitability
- Language Support
- Context Window Size
- Reasoning Abilities
- Fine-tunability



Performance Metrics

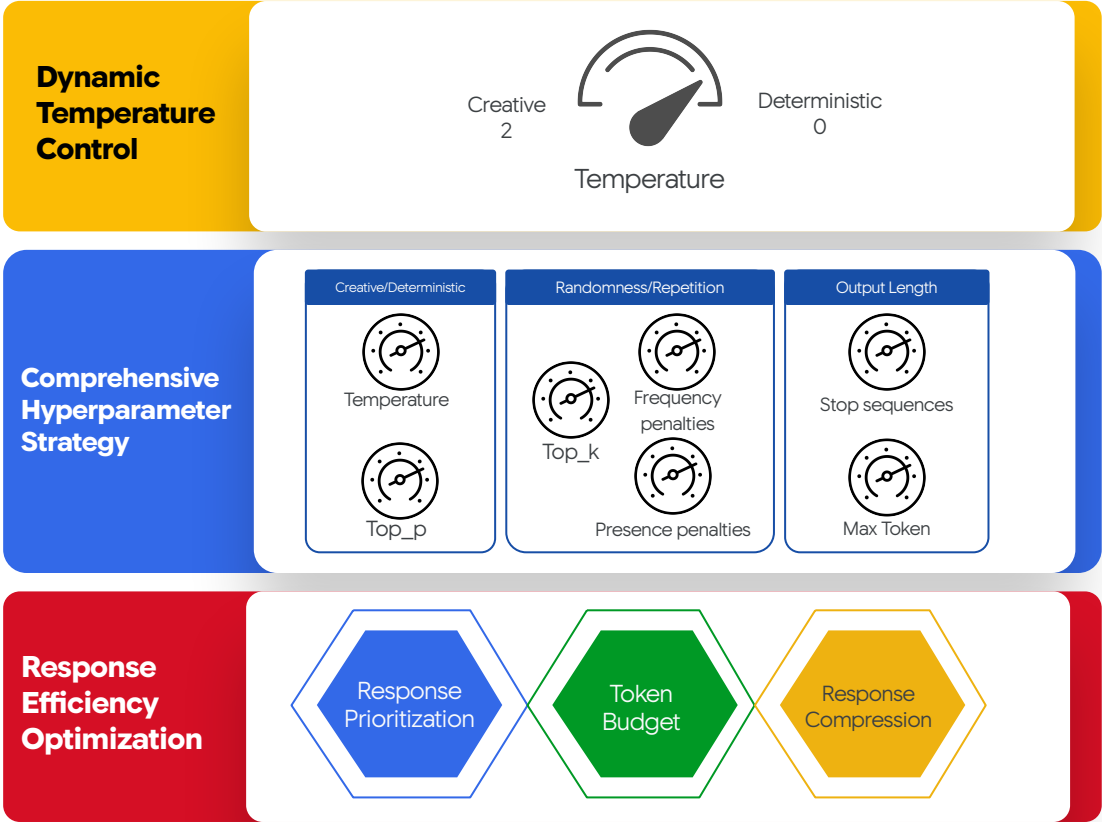
- Accuracy
- Fluency
- Latency
- Throughput
- Robustness



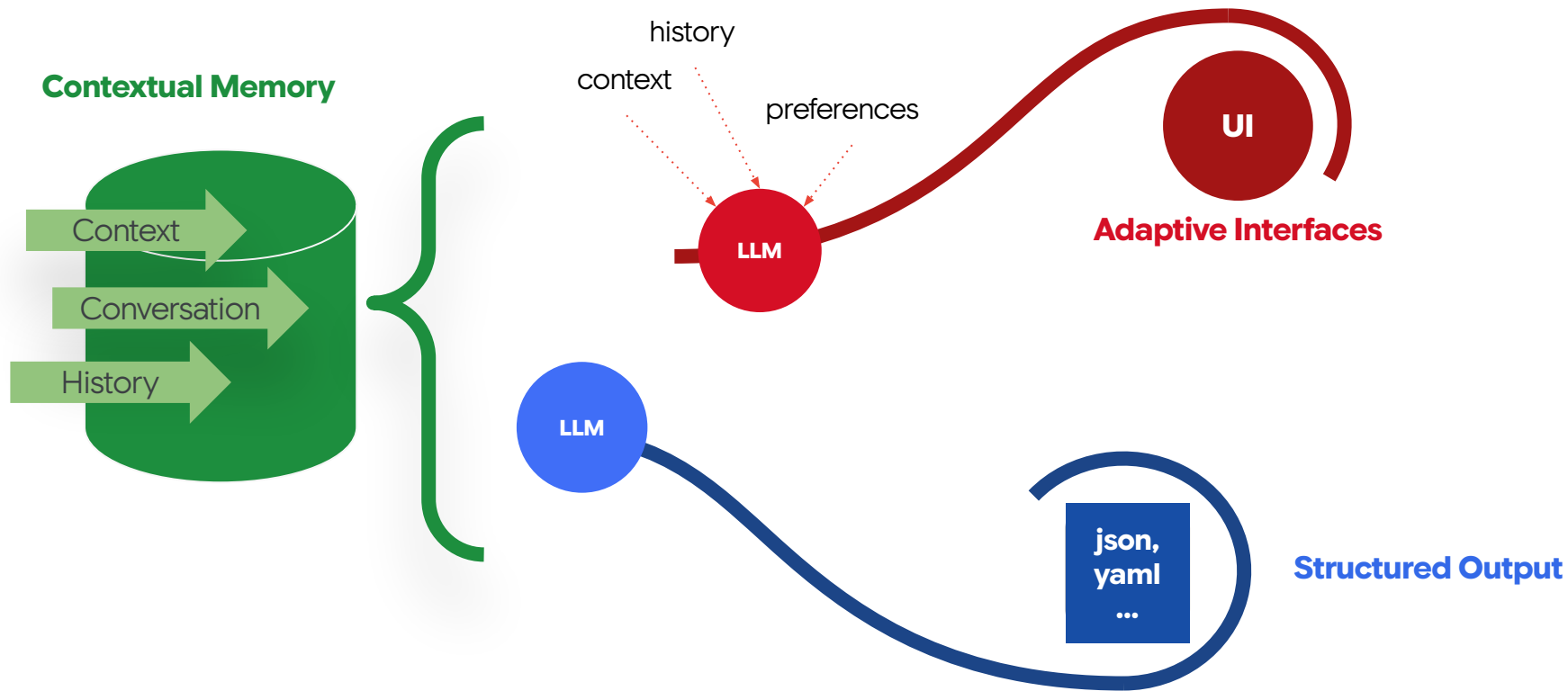
Cost

- Inference
- Fine-tuning
- Infrastructure

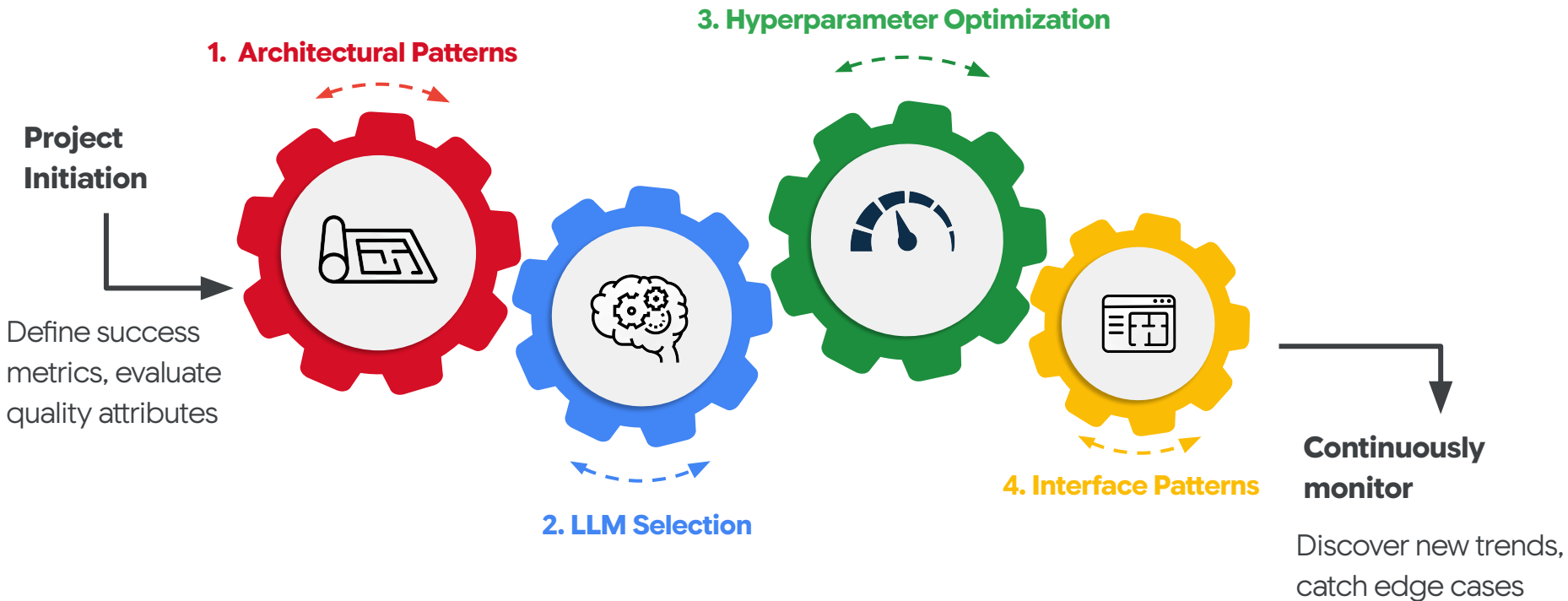
3. Hyperparameter Optimization



4. Interface Patterns



Blueprint for Agentic Success



What will you build with



+



Google Cloud

Agentic AI

Christina Lin

Head of Google Cloud Americas Advocacy

Blueprint for Agentic Success

Architectural Patterns

Consider speed quality cost trilemma defined, determine which architecture suits the most

Hyperparameter Optimization

Tune design based on testing the system for specific requirement.

Interface Patterns

Implement Contextual memory and select what type of interface best suits requirement

Project Initiation

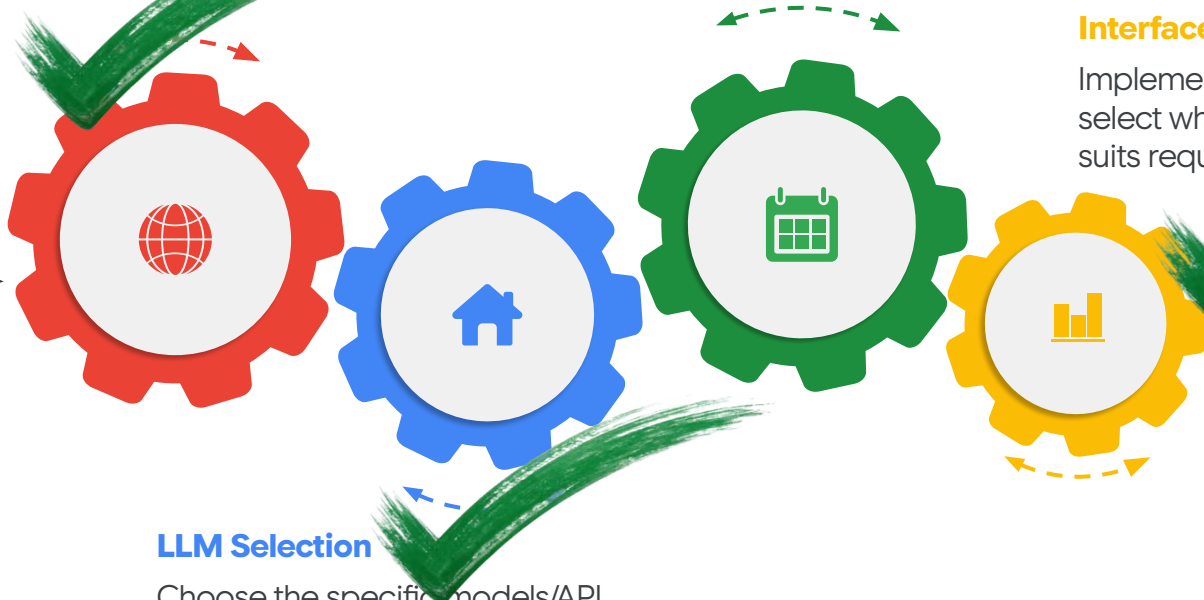
Define success metrics, evaluate quality attributes

LLM Selection

Choose the specific models/API that meet cost, performance and objective target defined

Continuously monitor

- discover new trend
- catch edge cases





What is in an **Agent**?

Input

Receive info, encompass user queries, sensor data, and real-time events, to understand the current situation and user intent.

Tools

Provide the agent with the ability to interact with the real world and external systems by leveraging function calling

LLM

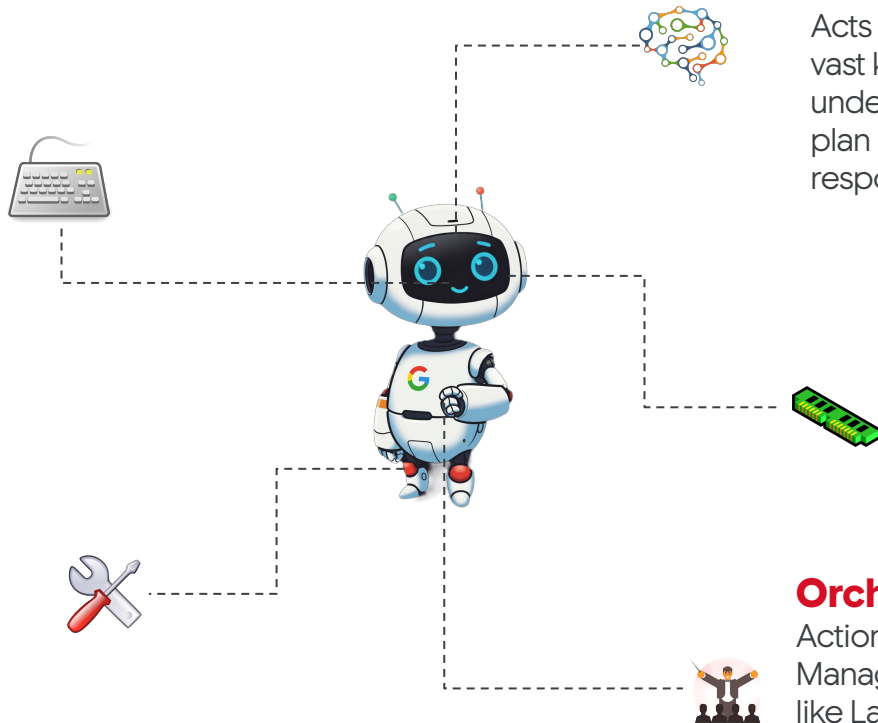
Acts as the "brain" of the agent, using its vast knowledge and language understanding to reason about the input, plan a sequence of actions, and generate responses

State

Stores both short-term memory (recent interactions) and long-term knowledge (user profiles, past experiences)

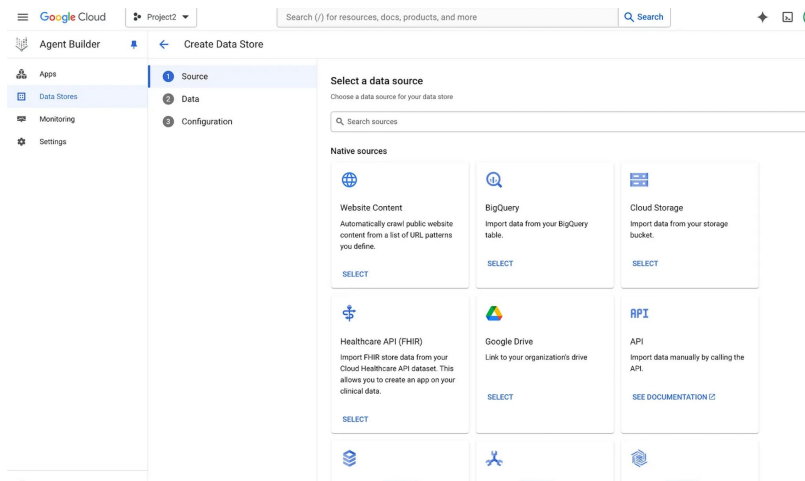
Orchestrate

Action Sequencing and Workflow Management (Powered by Frameworks like LangGraph, crewai)

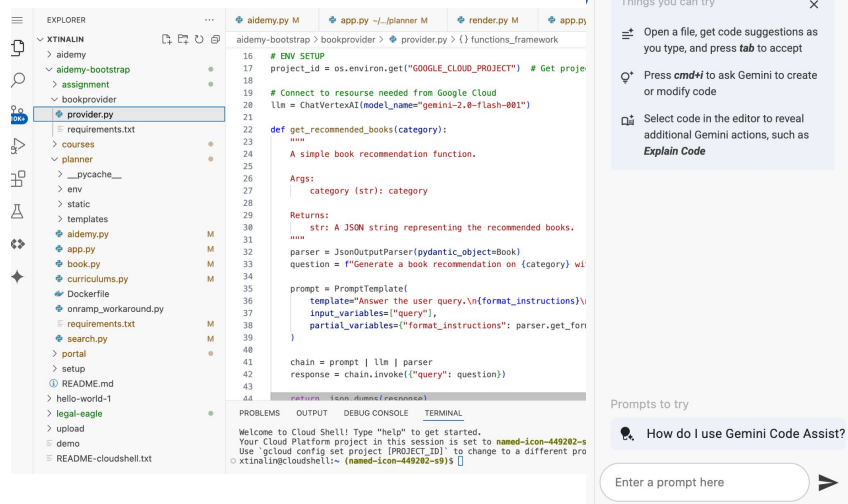


Building Agents with Google

Agent Builder



Custom Agent

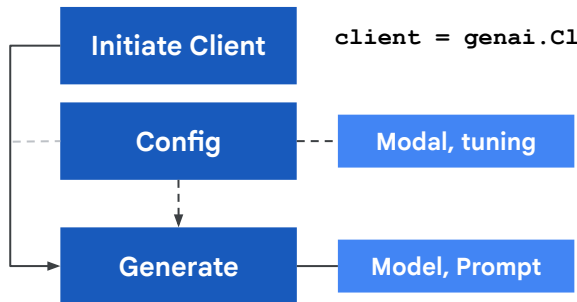


Single Model Architecture



Uses one LLM for all tasks in a system.

Google GenAI SDK



```
client = genai.Client(vertexai=True, project=PROJECT_ID, location=region)
```

```
response = client.models.generate_content(
    model='gemini-2.0-flash-001',
    contents='Why is sky blue?'
)
```

LangChain



Chain

Brings together components to create meaningful responses from language models.

```
chain = prompt | llm
```

Prompt

The entry point for interacting with LLMs, directing the flow of information.

```
prompt = PromptTemplate(
    template="Why is sky blue?"
)
```

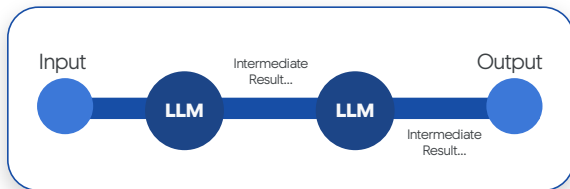
Model

The core of LangChain, consisting of three primary types: LLMs, Chat Models, and Text Embedding Models.

```
llm =
    ChatVertexAI(model_name="
    gemini-2.0-flash-001",
    temperature=0...)
```

Config

Sequential Models

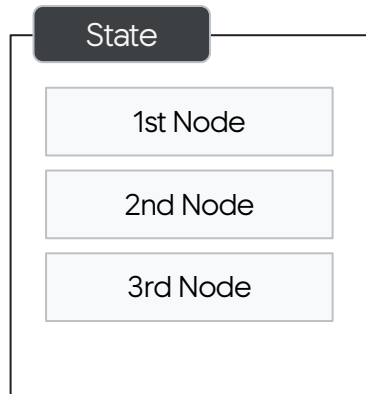
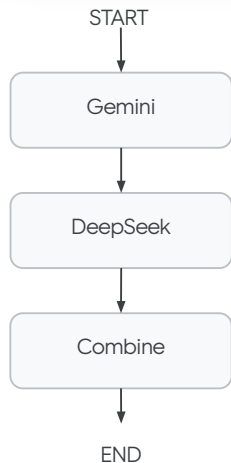
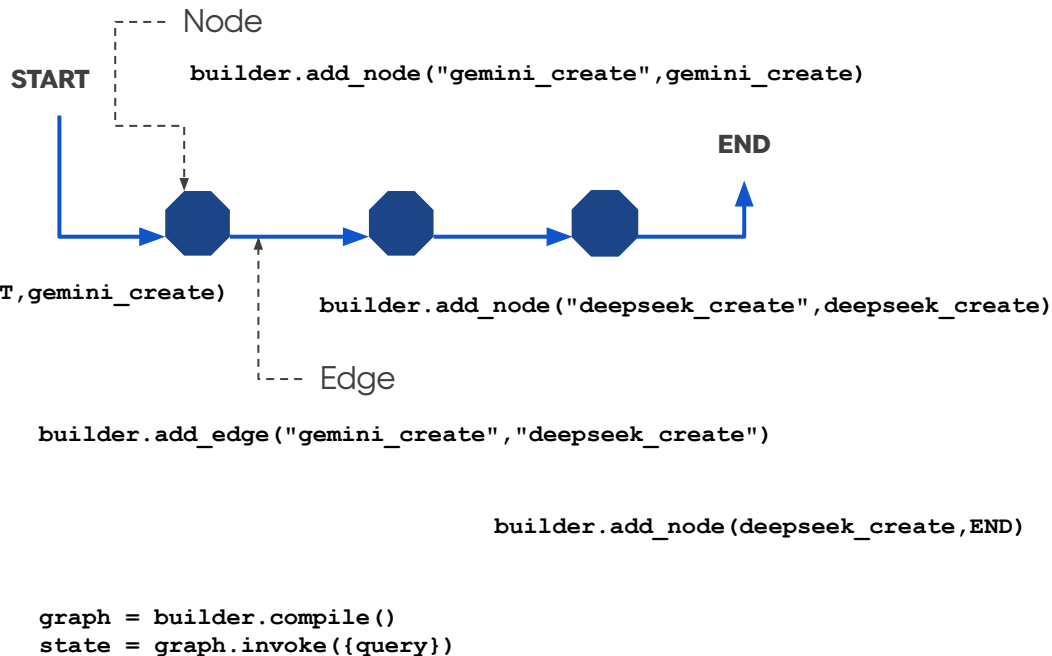


Chains multiple specialized LLMs in a pipeline.

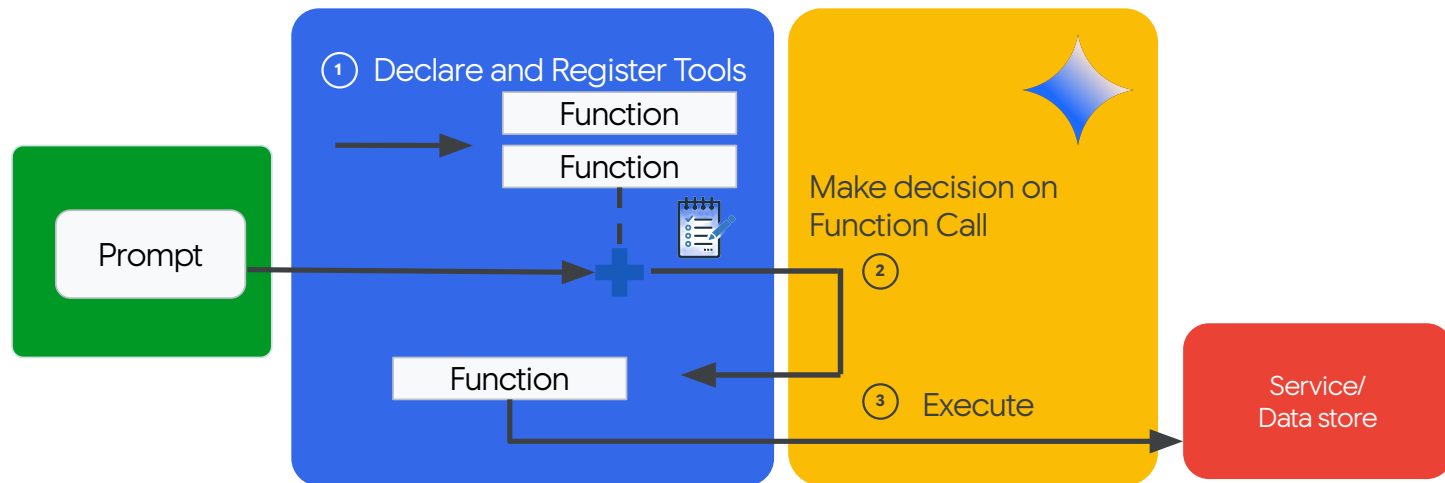
LangGraph



```
builder = StateGraph(State)
```



Function Calling



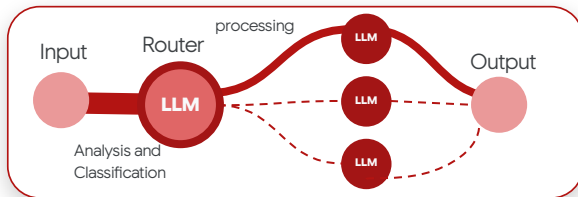
LangChain



```
tools = [function_one, function_two]
llm_with_tools = llm.bind_tools(tools)
tool_calls=llm_with_tools.invoke(query).tool_calls
```

┌
└-----> Function name and
Parameters

Hierarchical Routing



Central router model to direct inputs to specialized LLMs based on content or intent.

LangGraph

```
builder = StateGraph(State)
```

LangChain

```
Function -> determine_tool
tools = [function_one, function_two]
llm_with_tools = llm.bind_tools(tools)
tool_calls=llm_with_tools.invoke(query).tool_calls
```

```
builder.add_node(START,determine_tool)
```

START

END

Conditional Edge

```
builder.add_conditional_edges("determine_tool",tools_condition)
```

Tool Node

```
builder.add_node("tools",
ToolNode(tools))
```

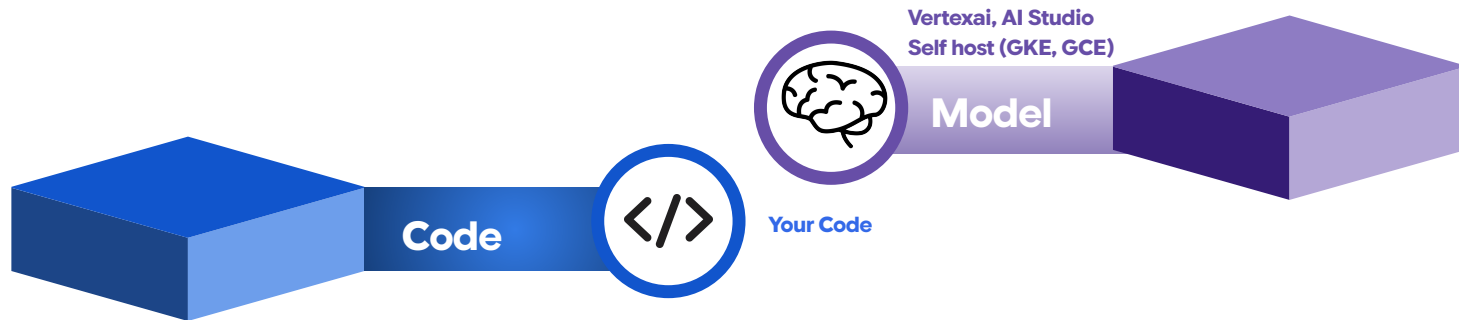
```
builder.add_edge("tools", "determine_tool")
```

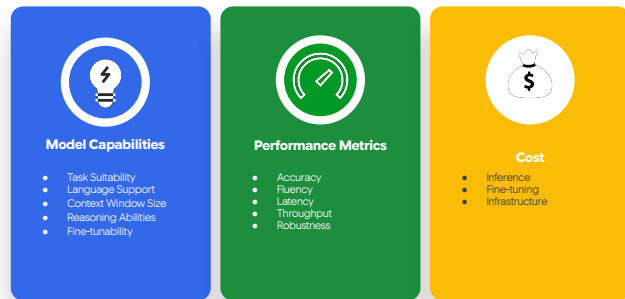
Tool Node

runnable

Execute

Deploying on Google Cloud





**Gemini 2.0 Flash
GA**

Production-ready

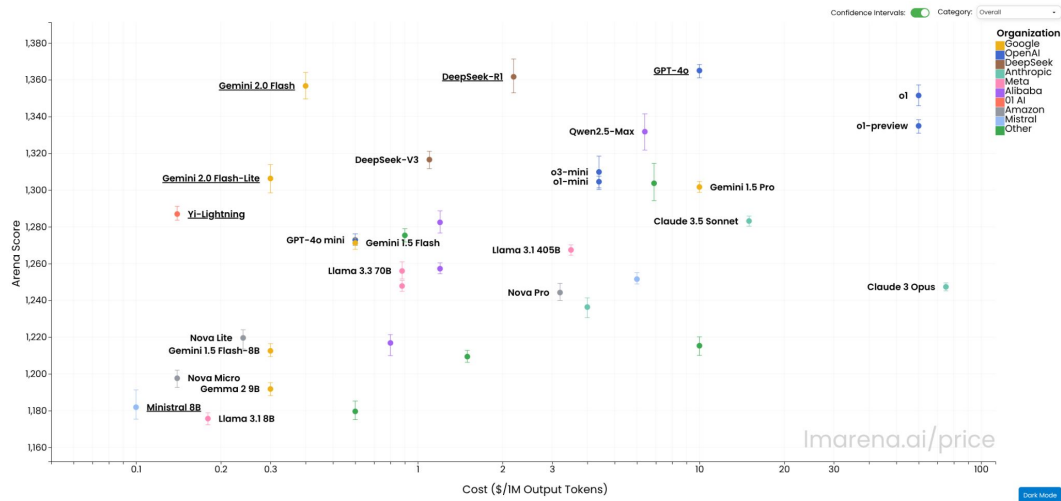
**Gemini 2.0
Flash-Lite
Preview**

Low-cost,
high-performance

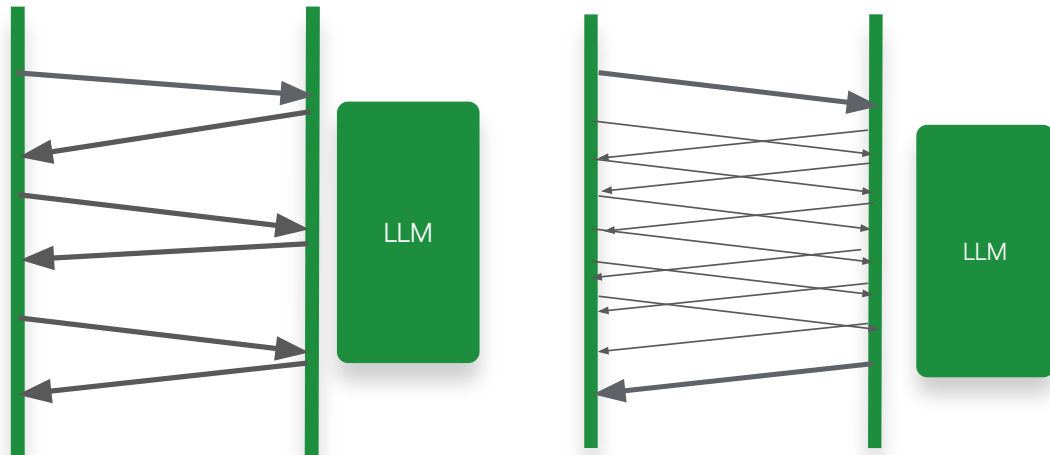
**Gemini 2.0 Pro
Experimental**

Superior
intelligence with a
2M context window

Pick your LLM!



Multimodal Live API



- Bidirectional streaming API
- Real-time **voice + vision** interactions and natural conversation mechanics like interruptions

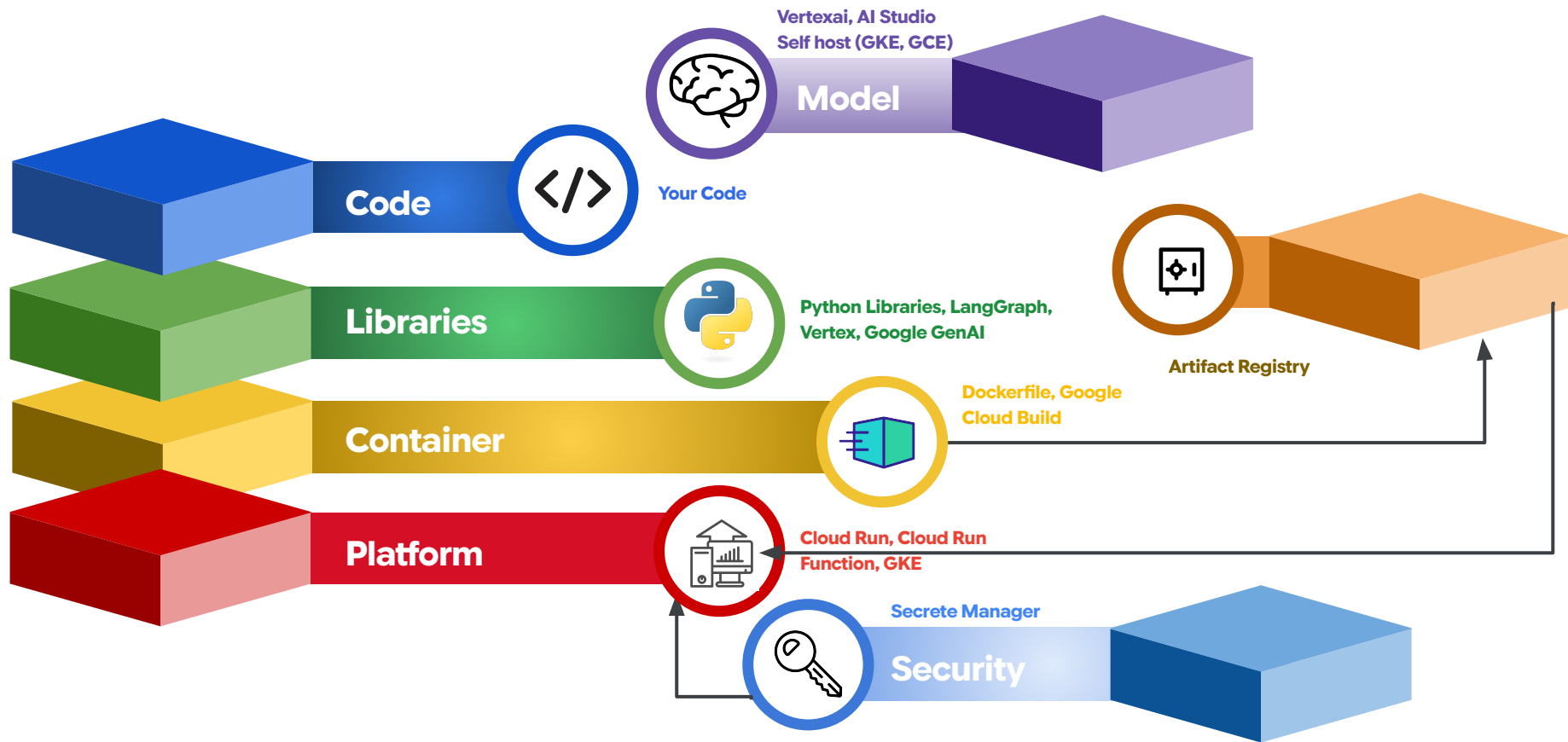
```
config = LiveConnectConfig(
    response_modalities=["AUDIO"],
    speech_config=SpeechConfig(
        voice_config=VoiceConfig(

prebuilt_voice_config=PrebuiltVoiceConfig(
    voice_name="Charon",
    )
),
),
)

client = genai.Client(vertexai=True,
project=PROJECT_ID, location="us-central1")
async with client.aio.live.connect(
    model=MODEL_ID,
    config=config,
) as session:
    await session.send(input=prompt)

    async for message in session.receive():
```

Deploying on Google Cloud



Cloud Run

Container



Deploy any OCI-complaint container image

Support repositories:

- Artifact Registry
- Docker Hub

Function



Single purpose functions.

Supported languages:
Python, Node.js, Go, Java,
Ruby, PHP, and .NET

As well as Dockerfile

HTTP

Out-of-the-box TLS endpoint

- HTTP
- websockets
- HTTP/2
- gRPC

Events

- Pub/Sub
- Cloud Storage
- 150+ event sources with 7000+ events from GCP services and third parties via EventArc and Firebase

Gemini



Cloud Run
Functions

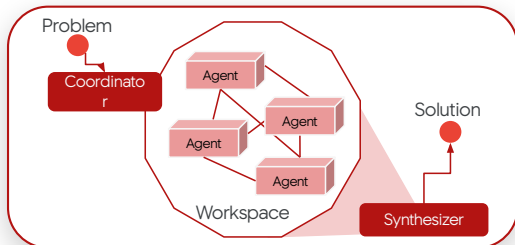


Cloud Run



GPU

Cooperative Multi-Agent System

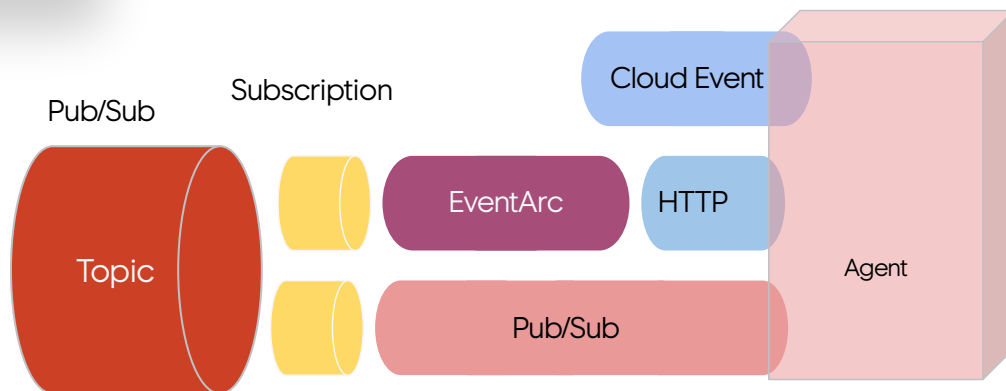
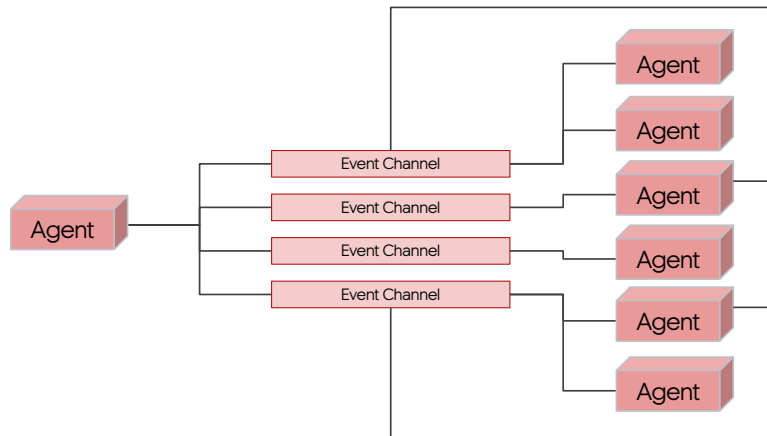


Use multiple LLM-based agents collaborating to solve complex problems.

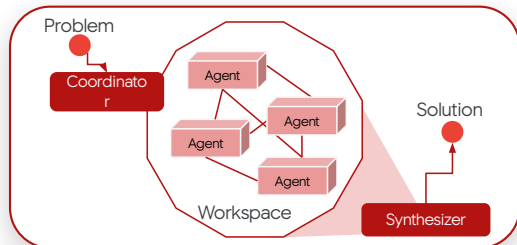
Prompt
Context(Summarized)
metadata



Event Driven Architecture

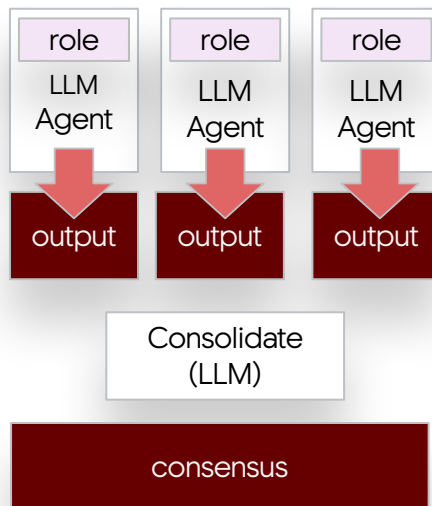


Cooperative Multi-Agent System

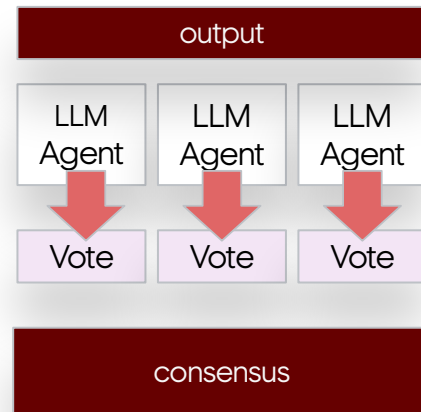


Use multiple LLM-based agents collaborating to solve complex problems.

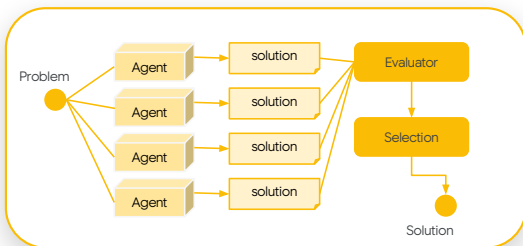
Role Based



Vote Based

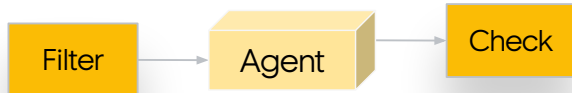


Competitive Multi-Agent System

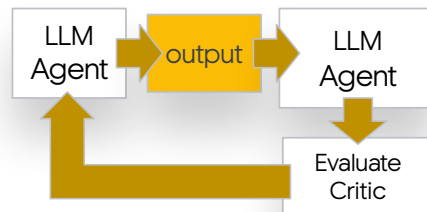


Deploys independent LLM agents to propose competing solutions, with an evaluator model selecting the best or integrating elements.

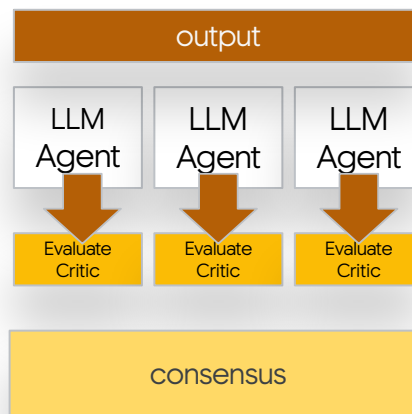
Guardrails



Evaluation



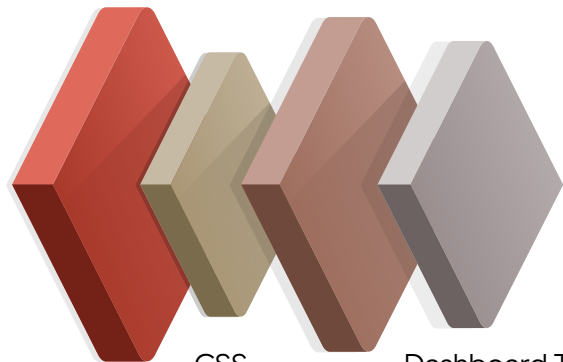
Debate Based



Adaptive Interfaces



Dynamically adjust the UI based on user context, history, preferences, and LLM capabilities.



html

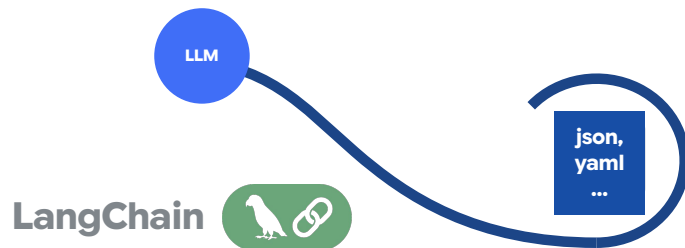
CSS

Dashboard One

Dashboard Two

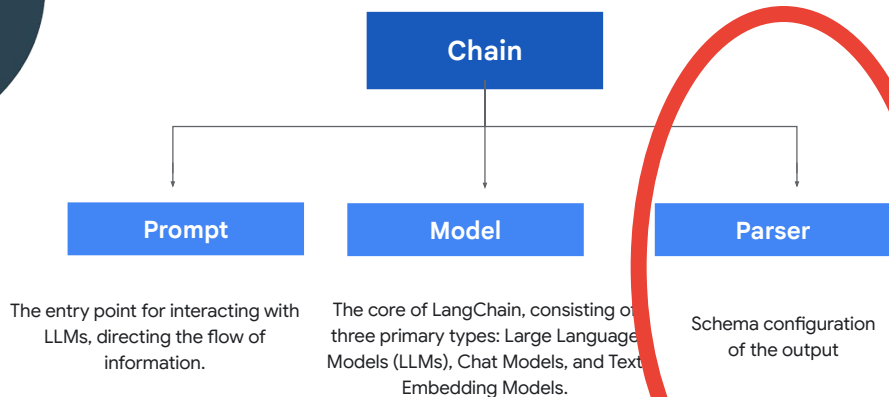


Structured Output



LangChain

`chain = prompt | llm | parser`



Blueprint for Agentic Success

Architectural Patterns

Consider speed quality cost trilemma defined, determine which architecture suits the most

Hyperparameter Optimization

Tune design based on testing the system for specific requirement.

Interface Patterns

Implement Contextual memory and select what type of interface best suits requirement

Project Initiation

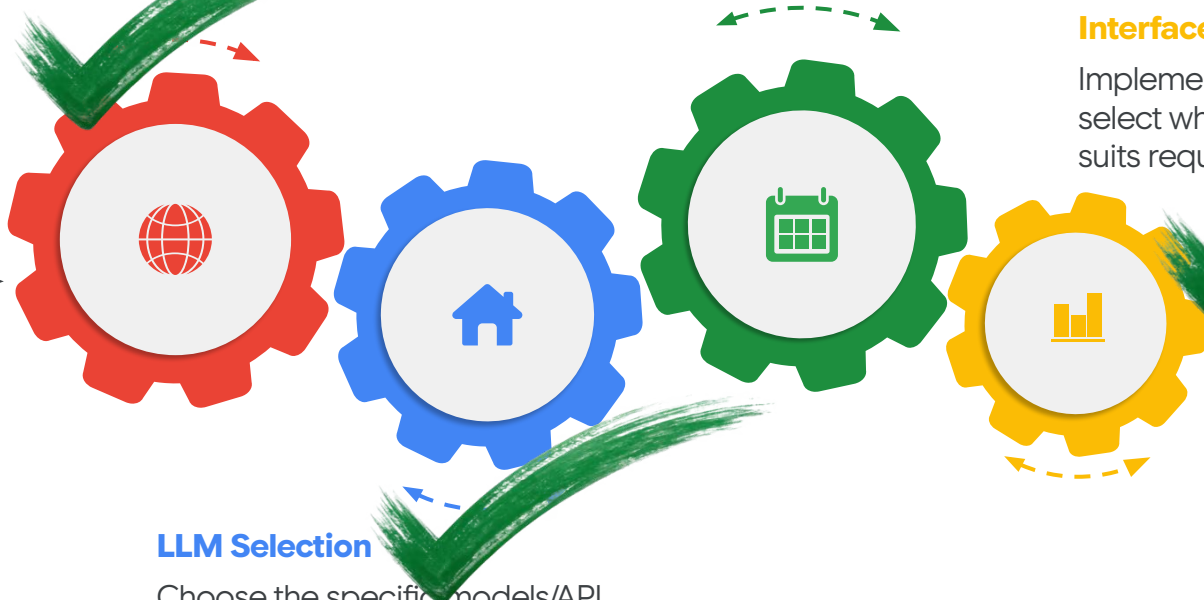
Define success metrics, evaluate quality attributes

LLM Selection

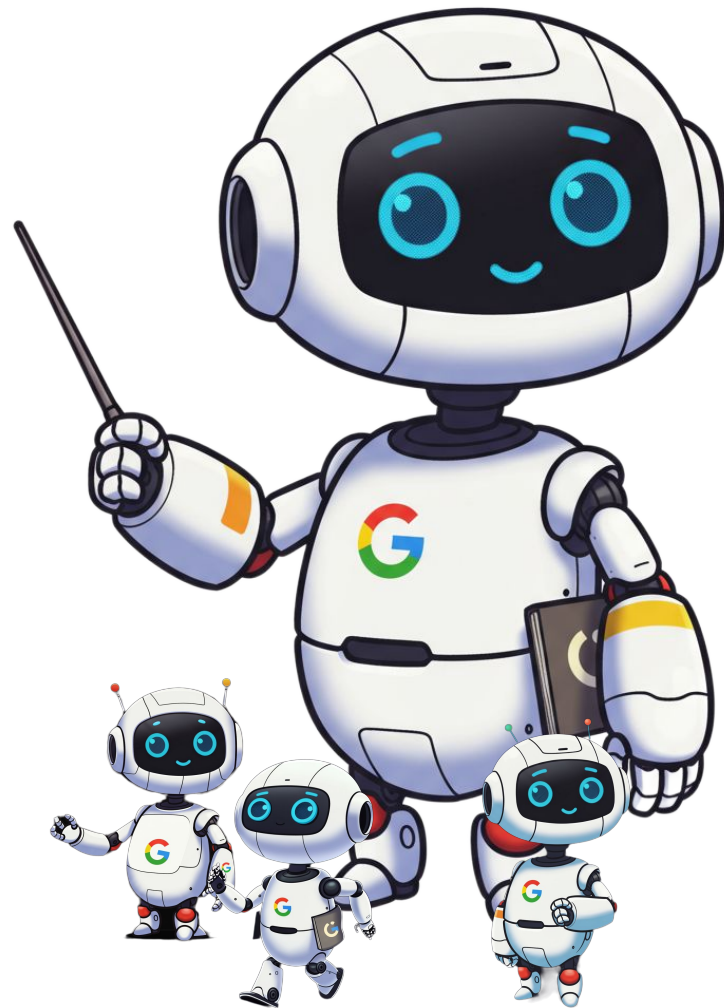
Choose the specific models/API that meet cost, performance and objective target defined

Continuously monitor

- discover new trend
- catch edge cases



Let's
build
some
agents!

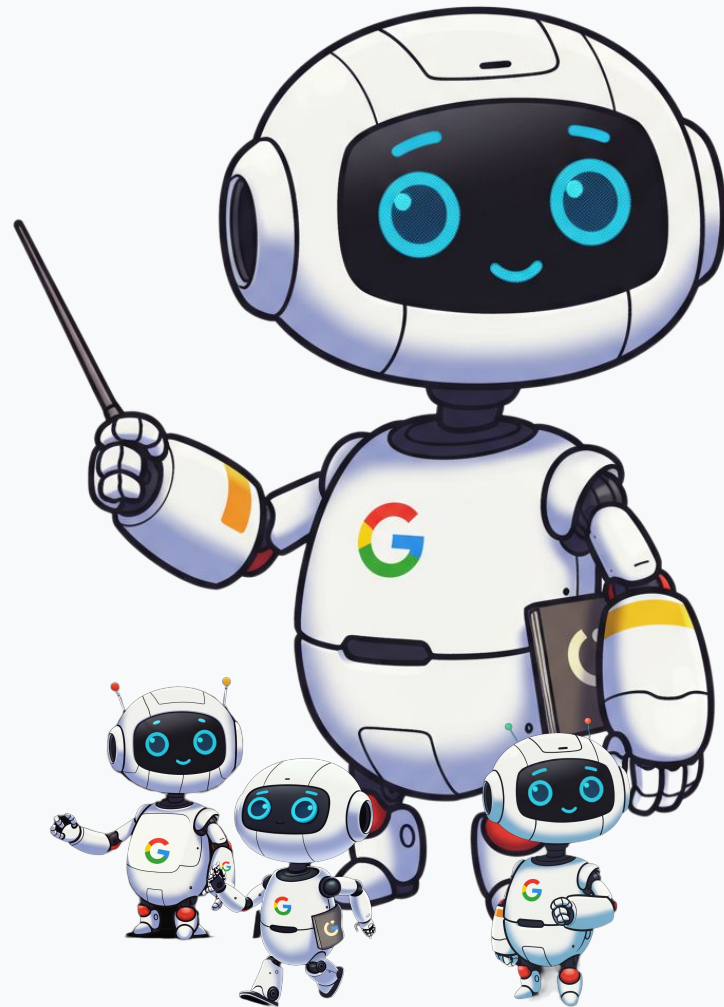


Aidemy

Agentic Workshop

Building the next generation education assistance
Multi-Agent System

NorthAM - GCA



Can you also provide after-school exercises that align with the teaching plan?

Can you create a three-week teaching plan for my Year 5 math student on Geometry?

Planner Agent

teaching plan

Here's the
teaching plan

Here are some short audio recaps of the week for students to review.

1. what's in the school curriculum.
2. Find out what's the latest information on the internet and
3. recommend few books to read.

Take some quizzes to test your understanding of the topic. If you get anything wrong, I'll be sure to explain it to you.

Course
Recap Agent

Assignment Agent

Here is some homework for the next three weeks

Assignments

Portal

☐ Question One

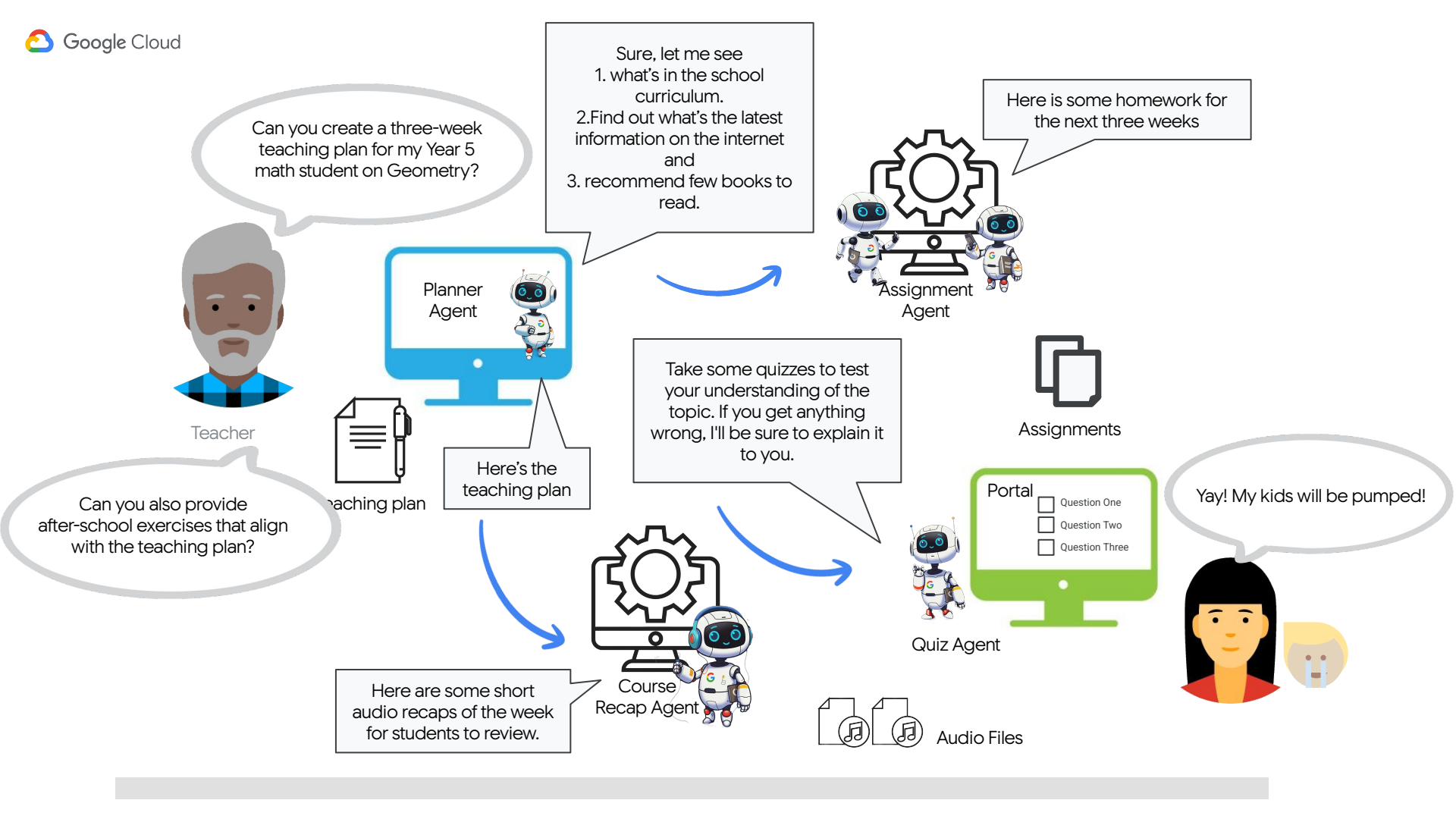
☐ Question Two

☐ Question Three

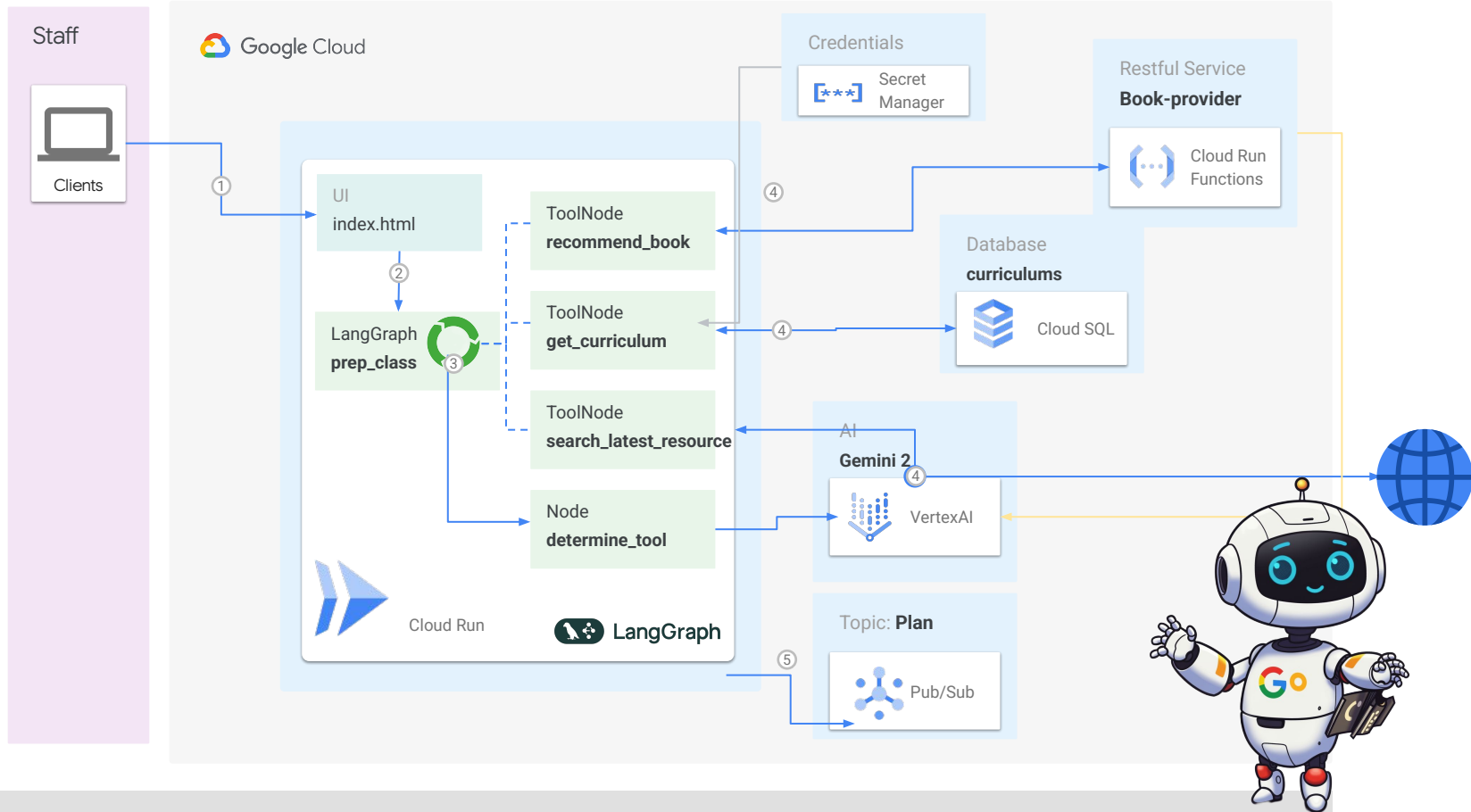
Quiz Agent

Audio Files

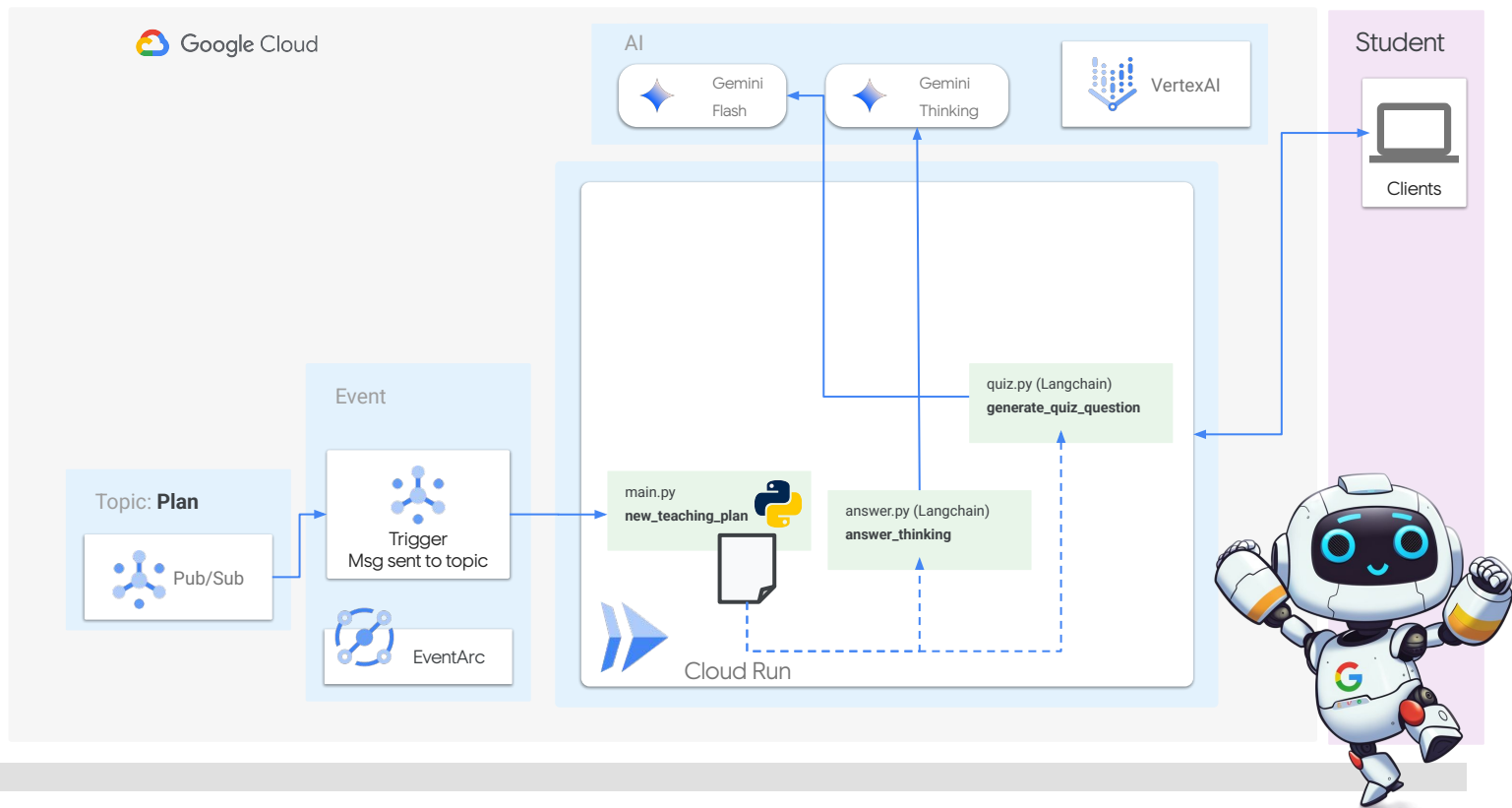
Yay! My kids will be pumped!



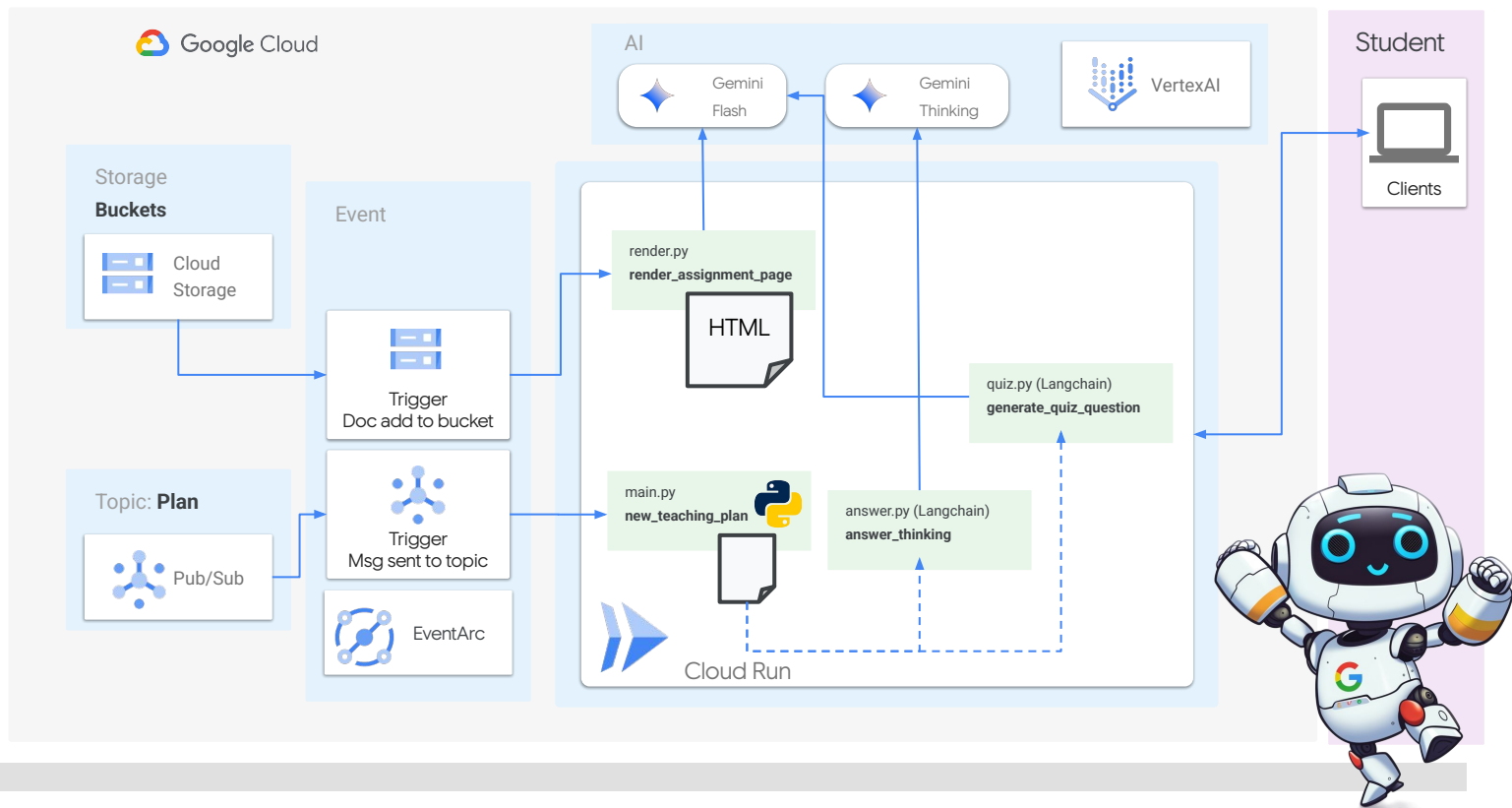
Custom Agent with Multi-Path orchestration, Function Calling and prompt organizer



Thinking model, Dynamic content delivery, Event Driven, RAG



Thinking model, Dynamic content delivery, Event Driven, RAG



Aidemy Architecture

Staff



Clients

Agents



Planner Agent
Python Flask

Cloud Run



Portal Agent
Python Flask

Cloud Run



Assignment Agent
CloudEvent

Cloud Run Function



Course Agent
CloudEvent

Cloud Run Function

Service



Book Provider
Python Flask

Cloud Run Function



Plan
Topic

Pub/Sub



Portal assignment
Trigger

EventArc



Courses-agent
Topic Trigger

EventArc

AI



Gemini 2
Flash

VertexAI



Gemini 2
Flash Thinking

VertexAI



DeepSeek
R1

GCE

Student



Clients

Data Sources



Curriculums
PostgreSQL

Cloud SQL

Events



Plan Topic
Trigger

EventArc



Assignment-agent
Bucket Trigger

EventArc

Storage



Courses
Bucket

Cloud Storage



Assignment
Bucket

Cloud Storage

Security



DB Credentials
PostgreSQL

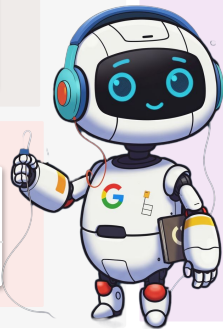
Secret Manager

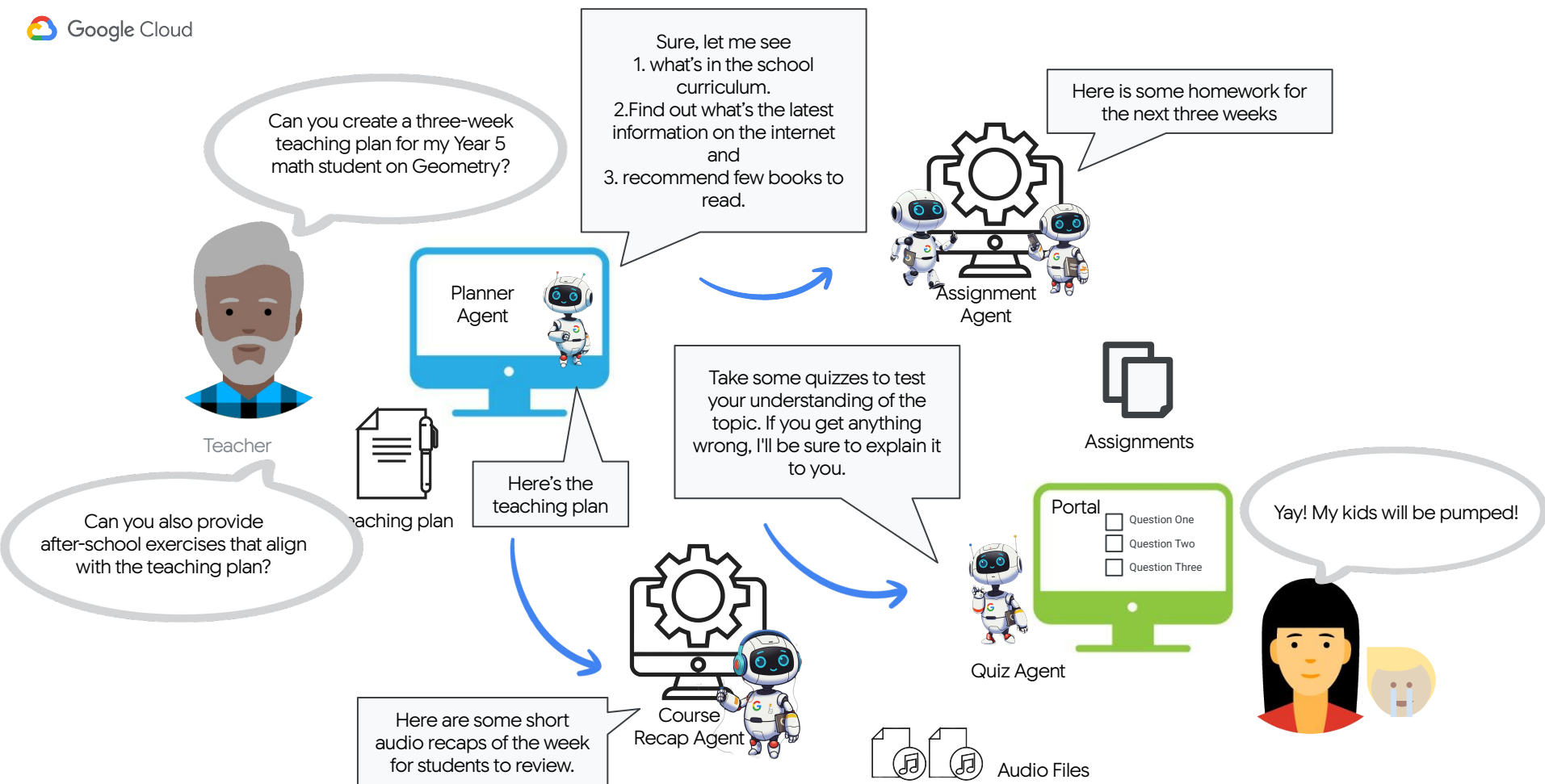
Miscellaneous



Image Repository
Agent containers

Artifact Registry





```
children: [  
  icon(icon, color: color  
  container(  
    margin: const EdgeIns  
  child:  
    label  
    style
```

When you join a [Google Developer Group \(GDG\)](#), you'll have the opportunity to learn new skills in a variety of formats. You'll also meet local developers virtually or in person with similar interests in technology.

→ goo.gle/gdg



Google Developer Groups