compeople

High-Code Agents in Gemini Enterprise

powered by ADK and Vertex AI

Meet the compeople Team

Frauke, Software Developer & Cloud Engineer

Nick, Professional Cloud Developer

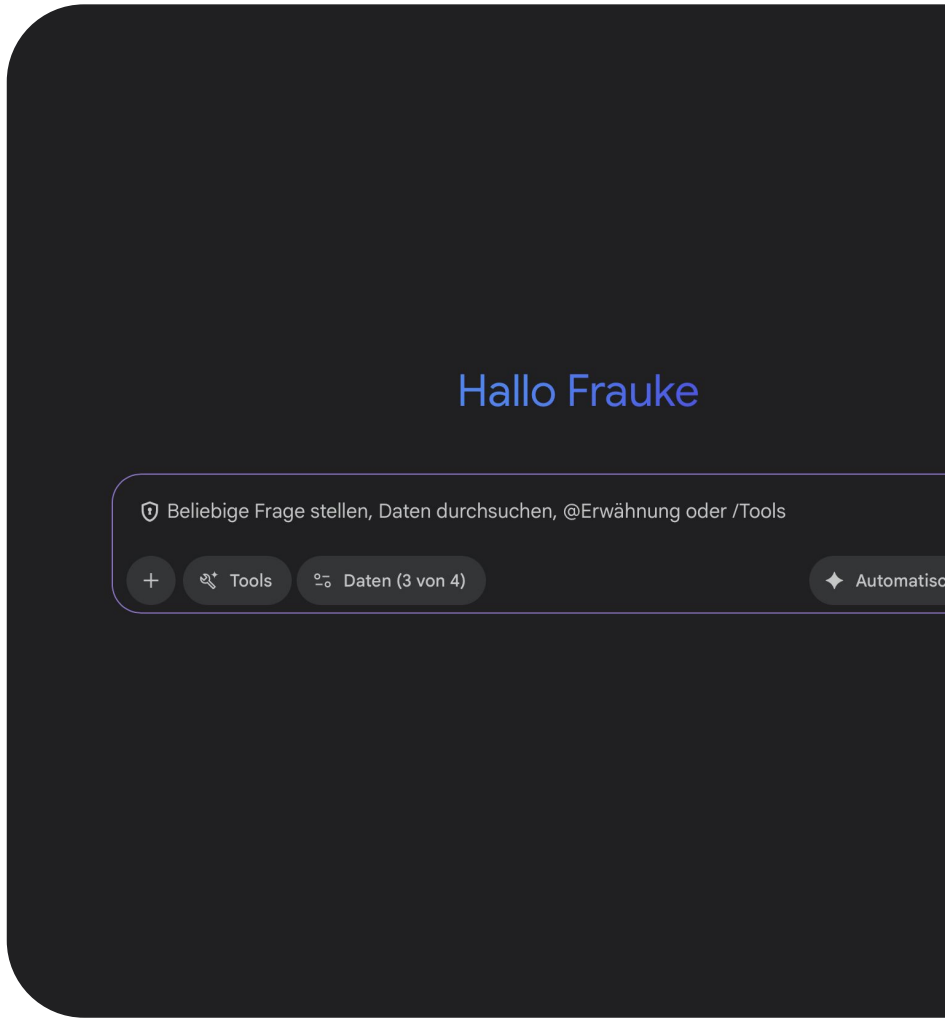
Christian, Data Scientist

Benni, BU Manager Application Development

compeople

Workshop Agenda

1. **Overview**
   - Fundamentals of Gemini Enterprise (GE) & Agent Development Kit (ADK)
   - Core architecture principles
2. **Custom Agent Creation**
   - Hands-on session custom agent development
3. **Lunch Break**
   - 1:00 PM - 2:00 PM
4. **Presentation & Future**
   - Best practices, Q&A Session



compeople

Question:

What are your expectations?

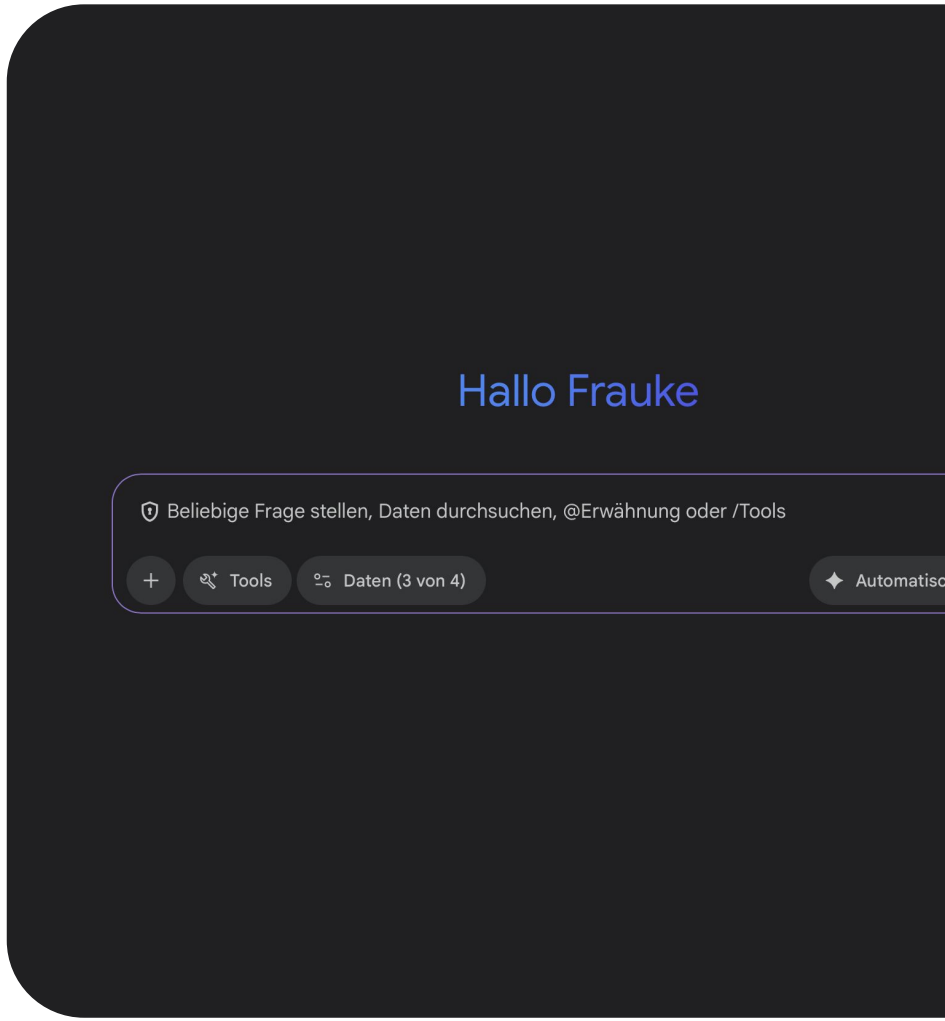
compeople

Overview

Custom Agent Set-up and leveraging ADK

Gemini Enterprise

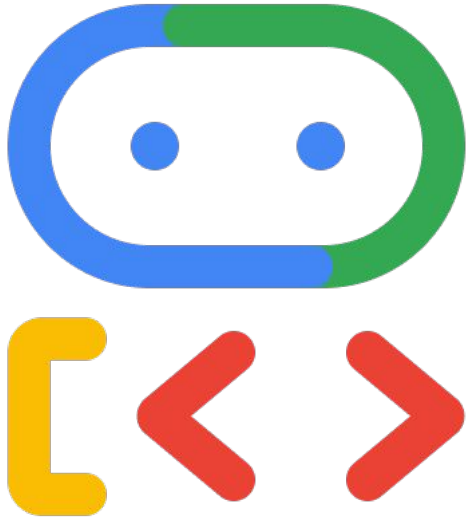
- Provides conversational assistance and answers complex questions
- Hosts custom AI agents that apply generative AI contextually
- Connects content across your organization to generate grounded and personalized answers
- Includes pre built connectors for the most commonly-used third-party applications such as Confluence, Jira and Microsoft SharePoint
- Gives employees a single multimodal search interface with permissions-aware access to enterprise information



compeople

Agent Development Kit (ADK)

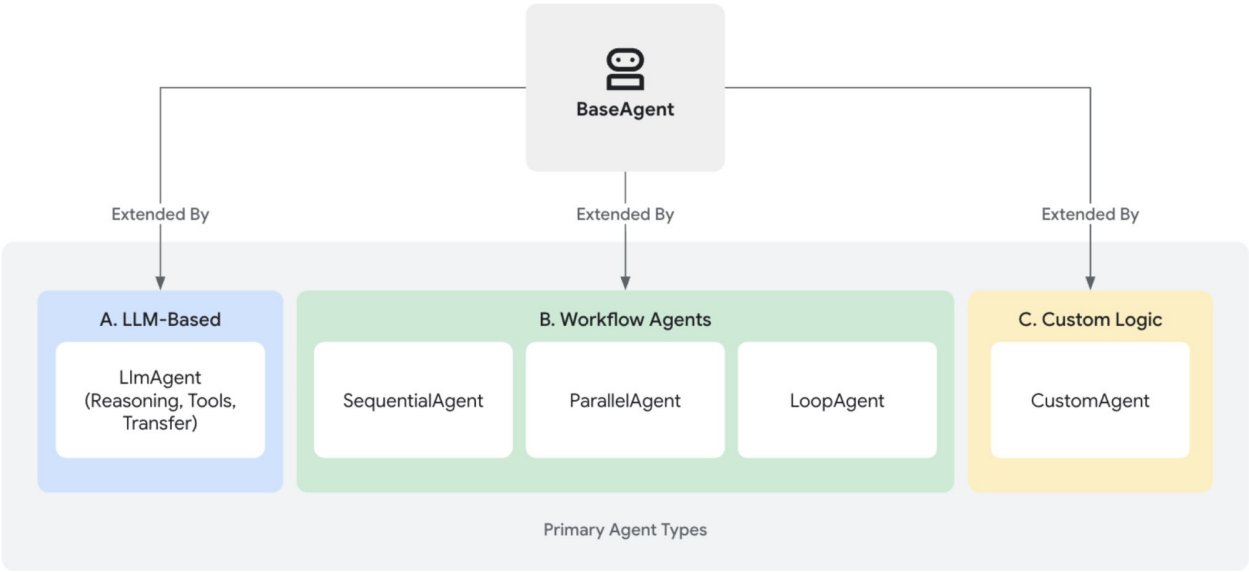
- An open-source, code-first Python framework for building, evaluating, and deploying sophisticated AI agents with flexibility and control
- **Rich model ecosystem:** optimized for Gemini and the Google ecosystem but built for compatibility with other frameworks
- Multi-Agent Support
- Agent Orchestration mechanisms



compeople

ADK Agents

- Agent = autonomous AI system that automates multi-step workflows by interacting with enterprise data sources and tools based on natural language instructions
- Different Agent Types solving different purposes

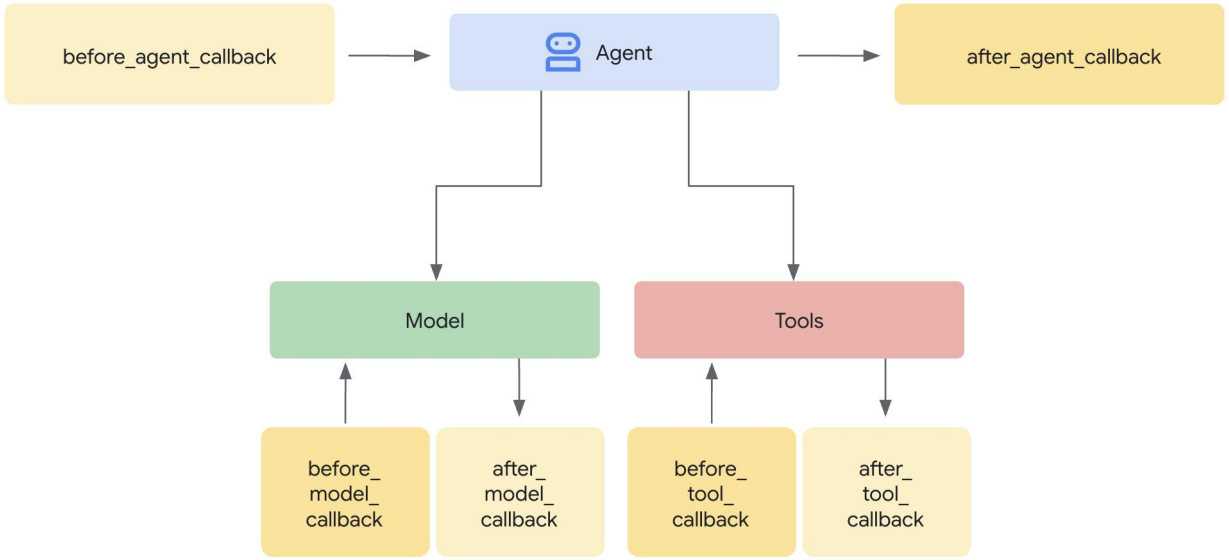


source: https://google.github.io/adk-docs/agents/

compeople

ADK Callbacks

- `before_XXXX_callback` and `after_XXXX_callback`
- used to execute code on specific times in agent execution

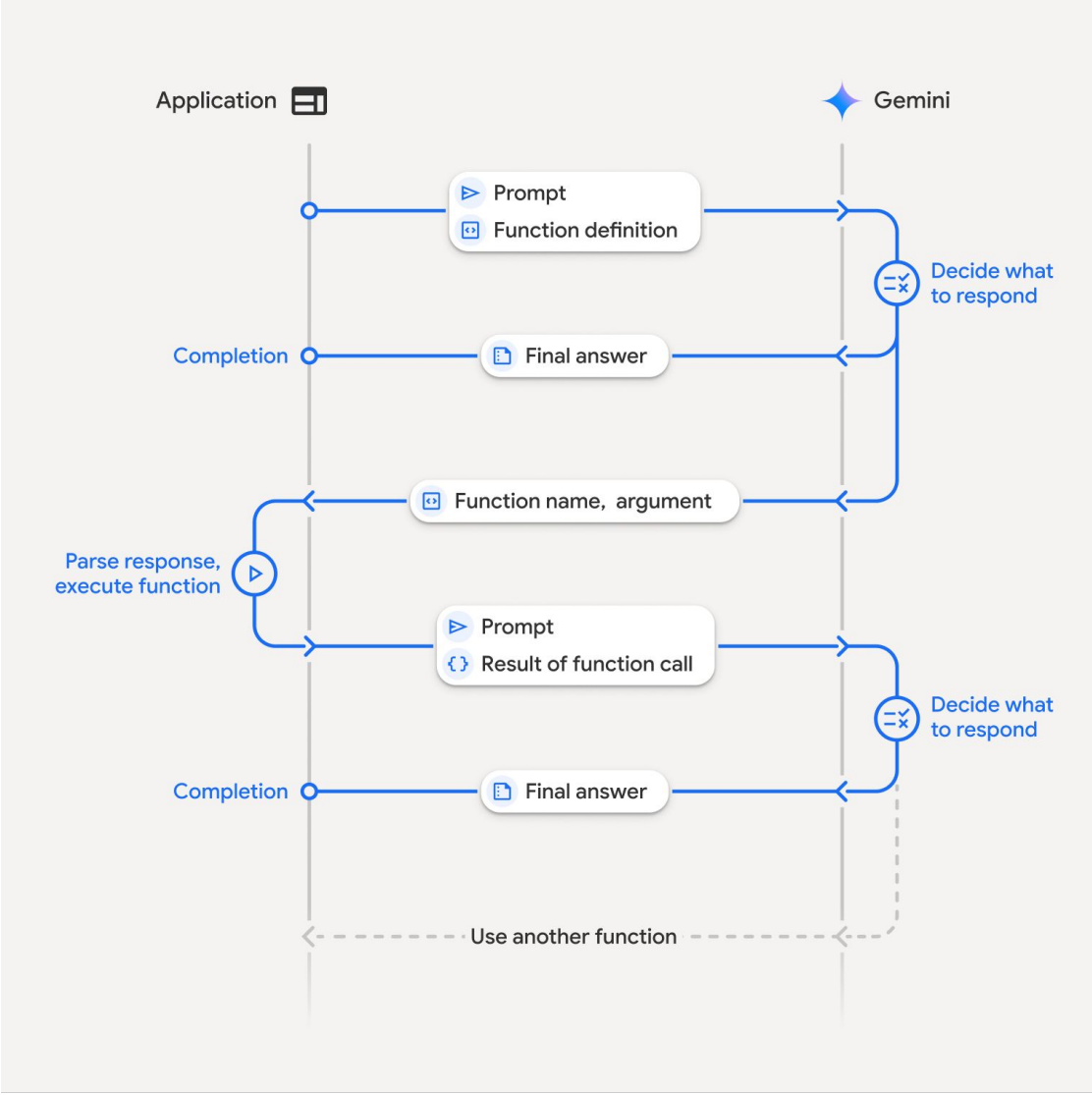


source: https://google.github.io/adk-docs/callbacks/

compeople

ADK
Tool Calls

- extend LLM by providing function definitions
- LMM can execute those functions
- tool response will be embedded in LLM answer
- best practices:
  - descriptive function signature
  - docstring
  - return dictionary, with “status” key
- usage: deterministic responses, information injection, external systems

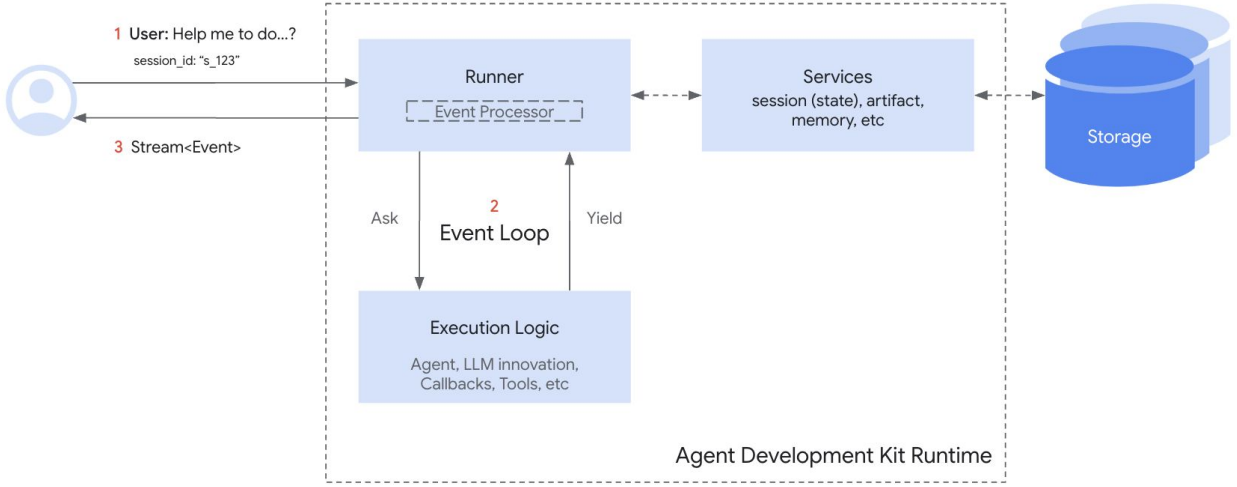


source: https://ai.google.dev/gemini-api/docs/function-calling?hl=de&example=meeting

compeople

ADK
Session and State

- Session is created when user interacts with Agent
- Stores everything related to this chat
- Consists of
  - **Identification** (id, appName, userId): Unique labels for the conversation
  - **History** (events): Event objects – user messages, agent responses, tool actions
  - **Session State** (state): store temporary data relevant only to specific, ongoing conversation
  - **Activity Tracking** (lastUpdateTime): timestamp indicating the last time an event occurred in this conversation thread



source: https://google.github.io/adk-docs/sessions/session/#sessionservice-implementations

compeople

Existing Tools

Apigee API Hub
Turn any documented API from Apigee API hub into a tool

Application Integration
Link your agents to enterprise applications using Integration Connectors

BigQuery Tools
Connect with BigQuery to retrieve data and perform analysis

Bigtable Tools
Interact with Bigtable to retrieve data and and execute SQL

Google Search
Perform web searches using Google Search with Gemini

Code Execution
Execute code using Gemini models

GKE Code Executor
Run AI-generated code in a secure and scalable GKE Sandbox environment

Spanner Tools
Interact with Spanner to retrieve data, search, and execute SQL

MCP Toolbox for Databases
Connect over 30 different data sources to your agents

Vertex AI RAG Engine
Perform private data retrieval using Vertex AI RAG Engine

Exa
Search and extract structured content from websites and live data

Firecrawl
Empower your AI apps with clean data from any website

GitHub
Analyze code, manage issues and PRs, and automate workflows

Vertex AI Search
Search across your private, configured data stores in Vertex AI Search

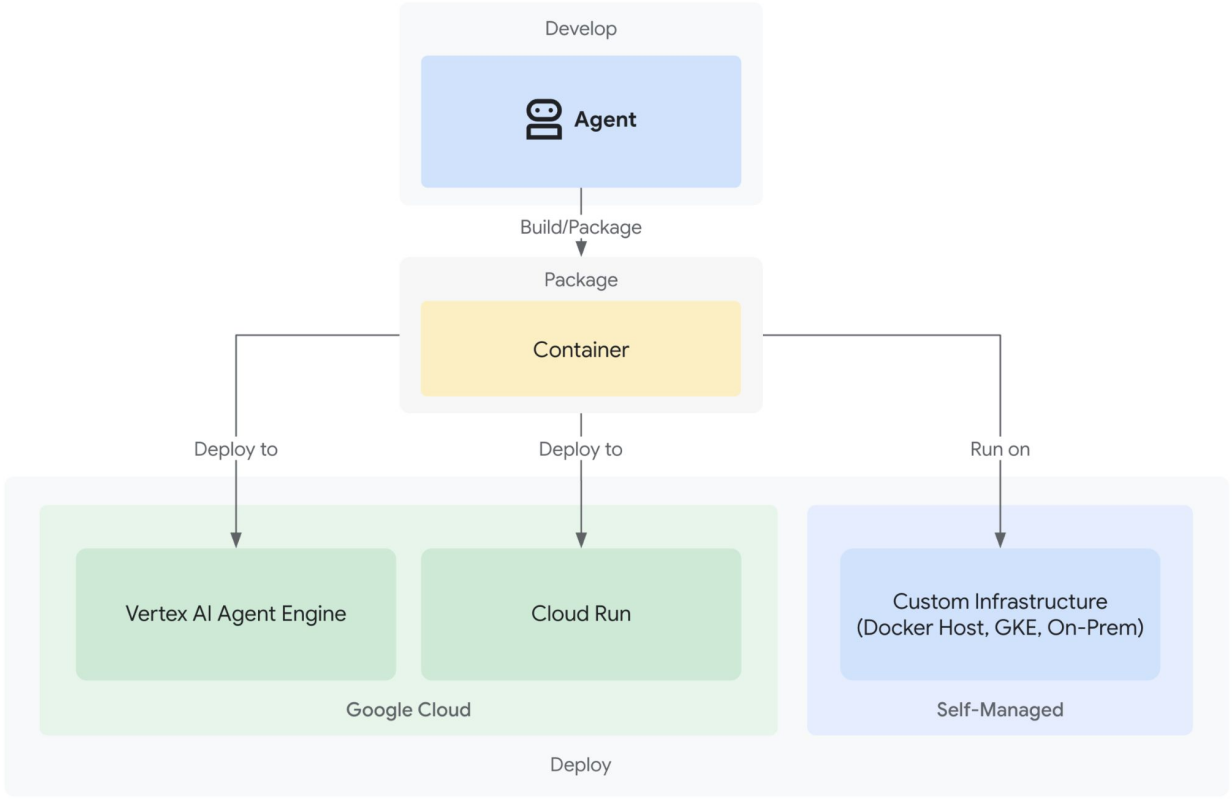
Notion
Search workspaces, create pages, and manage tasks and databases

Hugging Face
Access models, datasets, research papers, and AI tools

compeople

Deployment

- **Cloud Run**: managed auto-scaling compute platform on Google Cloud for container-based applications
- **Agent Engine**: specific Agent Runtime, fully-managed and autoscaling. Required for deploying to Gemini Enterprise
- **GKE**: managed Kubernetes Engine, to run containerized apps
- **Custom**: run in own environment

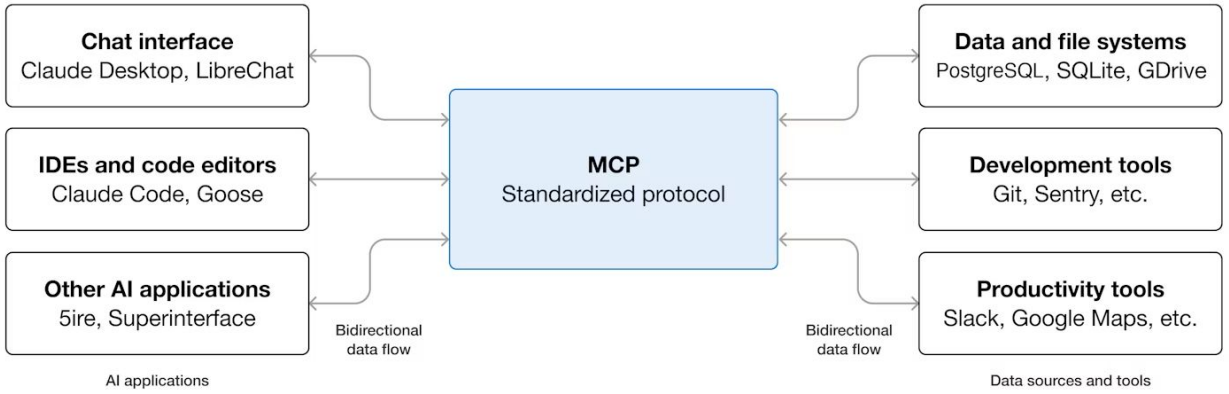


source: https://google.github.io/adk-docs/deploy/

compeople

MCP: Standard Protocol for Context & Actions

- Avoids M × N integrations and connects LLMs/agents with tools via a unified protocol.
- It's an api with a description in natural language.



Benefits

- Reusable
- Platform agnostic

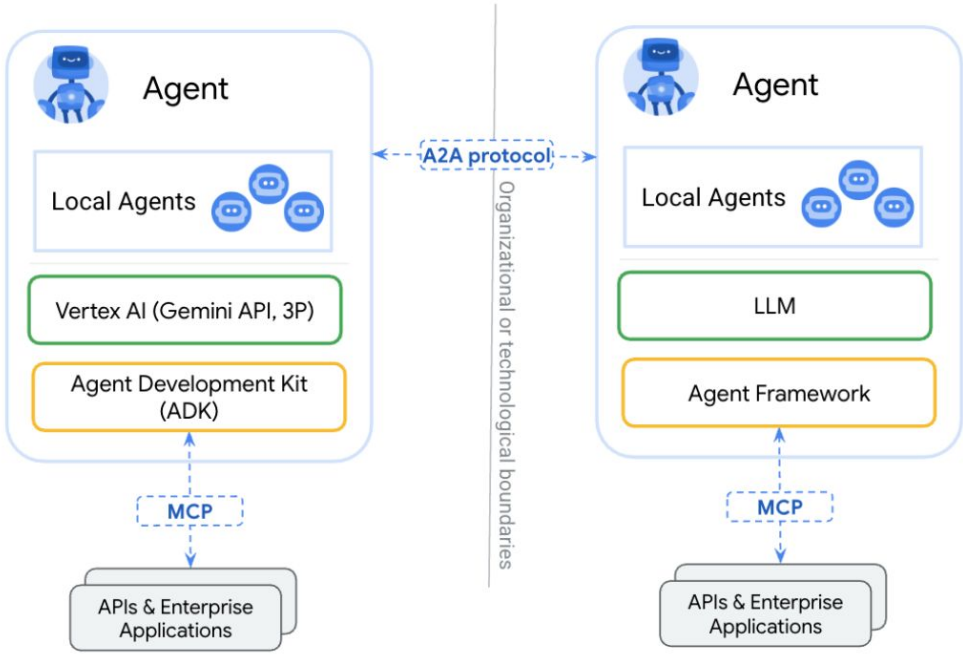
MCP Types

- Tools (functions)
- Resources
- Prompts

compeople

Connect agents with Agent2Agent protocol

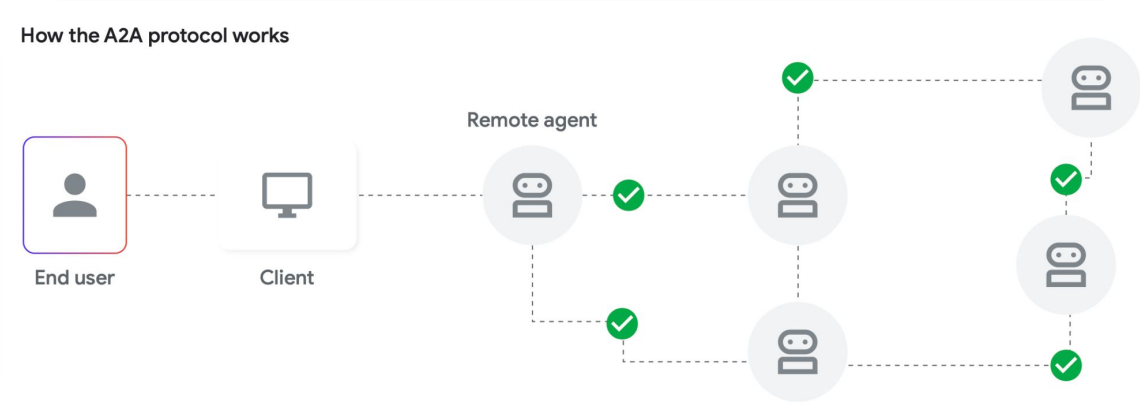
- Google offers the Agent2Agent (A2A) protocol
- Open standard that ensures the agents you build can discover, communicate with and securely coordinate actions with other agents regardless of what framework they use
- Commitment to an open, interoperable ecosystem
- Central part of Google Cloud's agent strategy



compeople

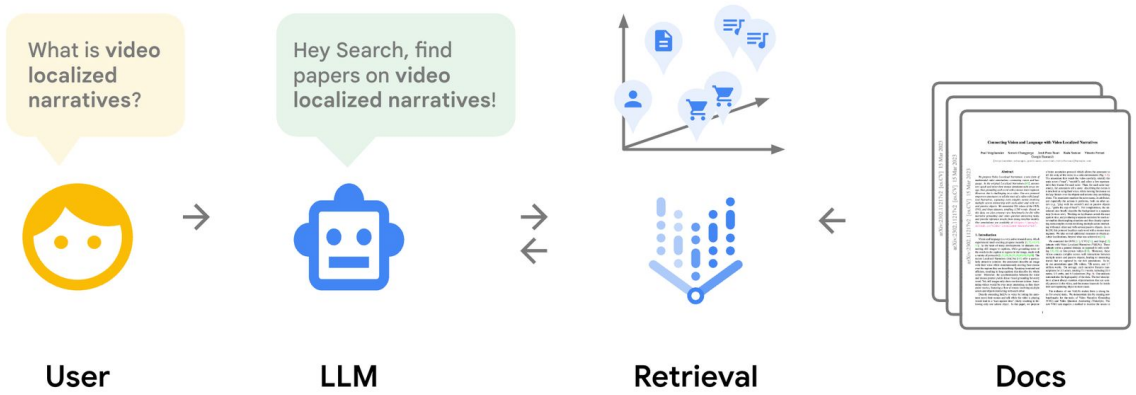
Key concepts of the A2A protocol

- **Agent card:** A digital “business card” (typically a JSON file) that an agent uses to advertise its capabilities, endpoint URL, and authentication requirements, enabling discovery by other agents
- **Task-oriented architecture:** Interactions are framed as “tasks.” A client agent sends a task request to a server agent, which processes it and returns a response. An agent can act as both a client and a server
- **Modality agnostic:** A2A supports text, audio and video communication to reflect the multimodal nature of agent interactions



compeople

RAG - Retrieval Augmented Generation



Combines traditional information system (like DBs) with LLMs.

Used to enhance the LLMs Context with custom knowledge.

Used for

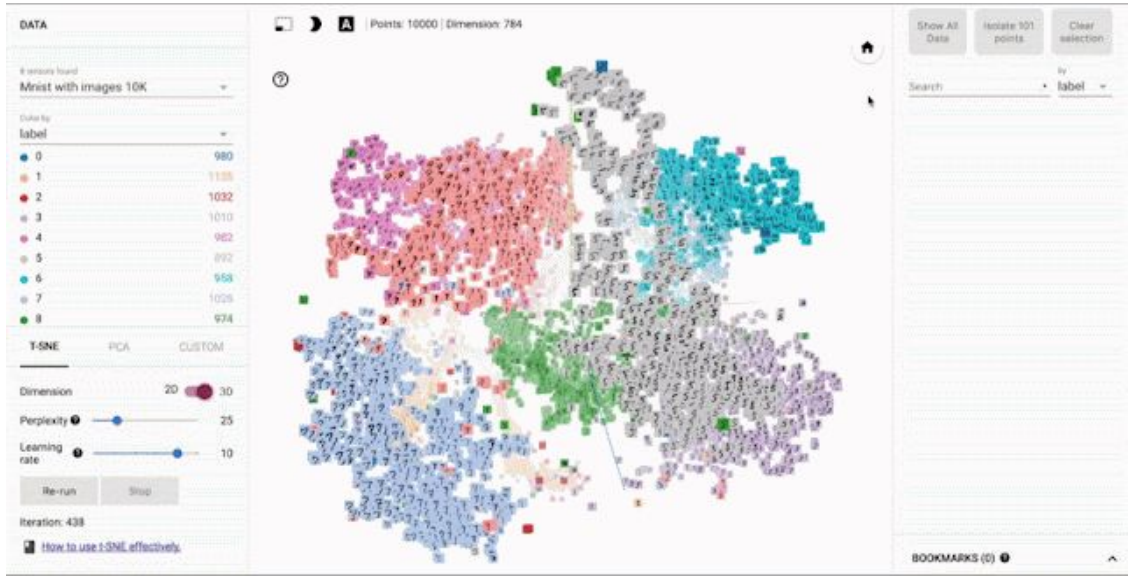
- Factual grounding
- Consumption of large unstructured information

Typical search technologies

- Vector Database (semantic search)
- Graph Database

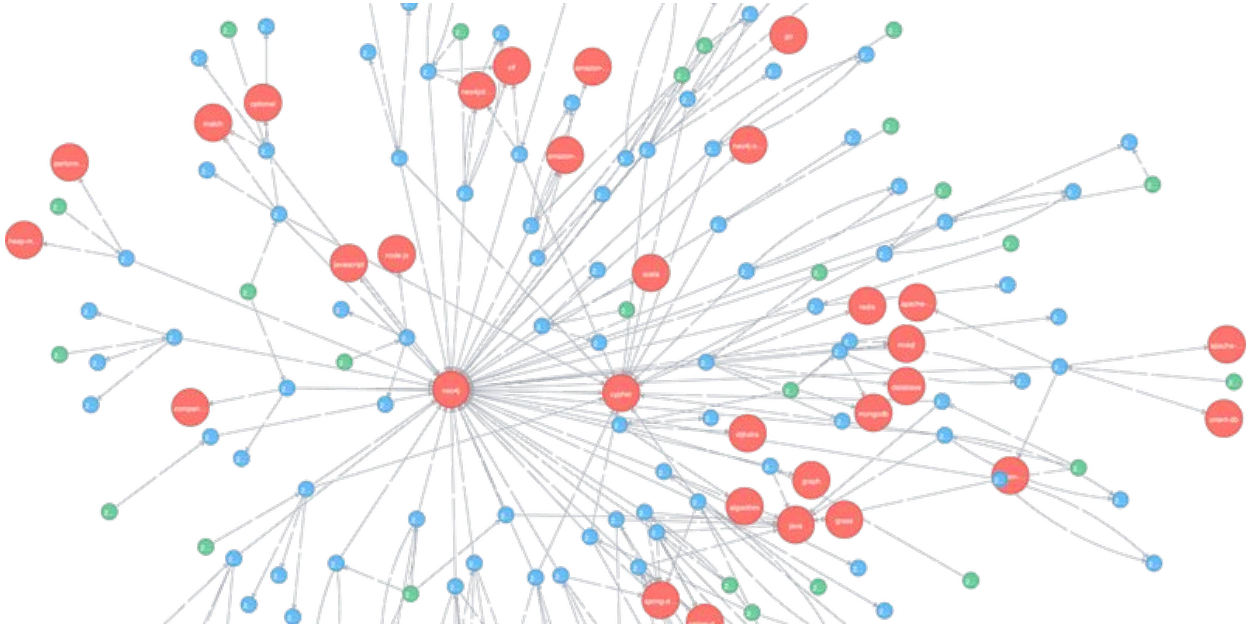
compeople

Vector Database



Embedding space of a vector DB

Graph Database



Notes of a Graph Database

compeople

Hands-On Session

Custom Agent Creation

compeople

Use Case:
IT Troubleshooting Agent

- Agent uses live system checks (via MCP/A2A Tool-Calling to internal APIs like monitoring/log systems) to diagnose and provide actionable resolution steps for IT/System incidents (e.g., slow database, authentication error, failed deployment, failed executions)
- Agent uses RAG over internal IT documentation (runbooks, past incident reports)
- Enables fast, grounded problem resolution without needing a human to manually cross-reference docs and log files.

compeople

Use Case - The Approach

1. Set up your own Gemini Enterprise App dh-adk-gemini-workshop
2. Create a GCS Bucket and upload testfiles (the runbooks)
3. Create a datastore out of the Bucket and connect it to your Gemini Enterprise
4. Create an ADK Agent using the code from the provided github repository: http://bit.ly/47pinaH
5. Connect ADK to MCP Server

compeople

Use Case - The Approach

- MCP Server is running on (requires IAM authentication) https://mcp-server-no-auth-38251951707.europe-west4.run.app/mcp
- env:

```
GOOGLE_CLOUD_PROJECT=dh-adk-gemini-workshop-1507

GOOGLE_GENAI_USE_VERTEXAI=true

GOOGLE_CLOUD_LOCATION=europe-west1

STAGING_BUCKET="gs://dh_tmp_agent_deployment"

AGENT_NAME="Workshop_Agent_compeople"

AGENT_DESCRIPTION="Agent to get weather infos"

GEMINI_ENTERPRISE_APP_ID="

REASONING_AGENT=""
```

compeople

Best practices

for enterprise-ready agents

compeople

Building reliable agents

Goal

- Manage your agent's lifecycle professionally
- Ensure your agent is accurate and safe before going live
- Track your agent's real-world performance, cost and errors
- Figure out why your agent made a specific decision

Best option

- Adopt AgentOps to automate processes from development to deployment and monitoring
- Implement automated evaluation in your CI/CD pipeline to rigorously test for quality, grounding and safety
- Set up monitoring using observability tools to get real-time data on latency, token usage and tool call success rates
- Inspect the agent's trajectory (its "chain of thought") using logging and tracing tools to debug its reasoning process

compeople

Building reliable agents

Goal

- Secure your agent, its data and its tool access
- Get started with AgentOps quickly

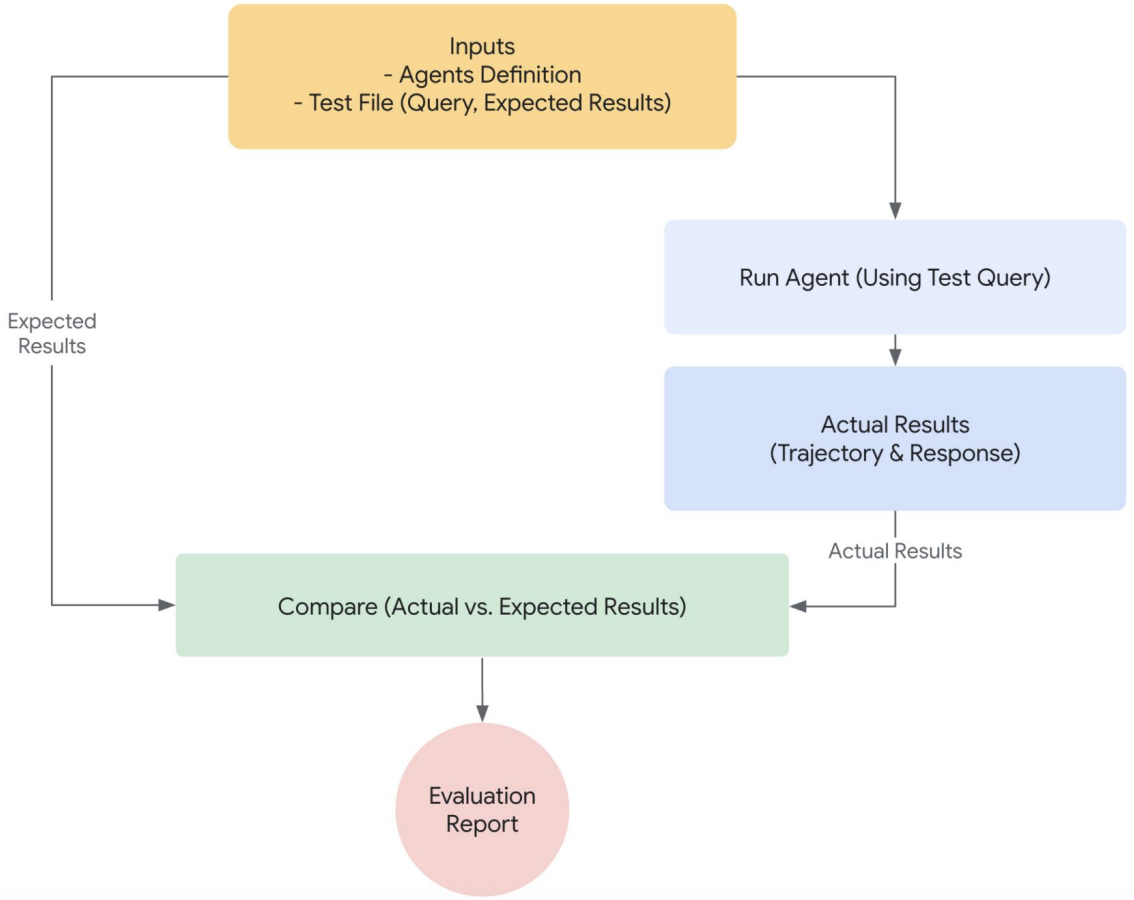
Best option

- Apply AgentOps security principles, which include infrastructure security, data governance and compliance controls
- Use the Agent Starter Pack for pre-configured templates for CI/CD, evaluation and infrastructure

compeople

Agent evaluation

- Traditionally unit tests and integration tests provide confidence that code functions as expected
- LLM agents introduce a level of variability that makes traditional testing approaches insufficient
- Define clear objectives and success criteria:
  - What constitutes a successful outcome for your agent?
  - What are the essential tasks your agent must accomplish?
  - What metrics will you track to measure performance?



compeople

Q&A

compeople

Thank you for your time! :)