Unboxing ADK & A2A

Next Wave AI Meetup

Google Cloud

Leon Jalfon

Jonathan Jalfon

May 15, 2024

The AI revolution has redefined **possible**,

clearing the path for **innovation**...

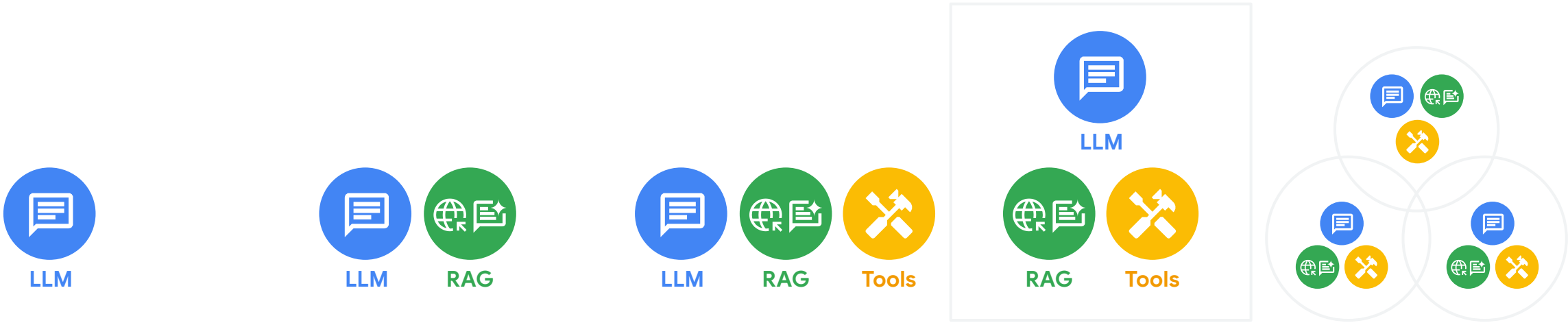
...so, **stop wondering** and **start building**, your

ability to build **AI agents** is here and **now!**

COMMIT

AI Agents

History of the agents evolution



LLM + prompt	LLM + retrieval	LLM + retrieval + actions	+ Many tools and reasoning loop	Multi agent systems

COMMIT

Unboxing ADK & A2A

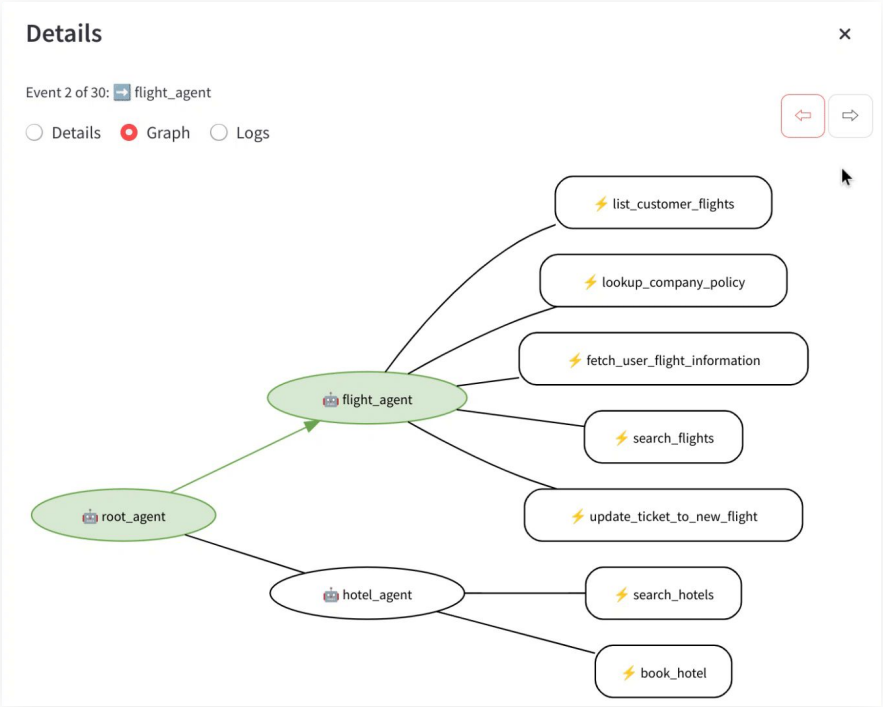
(Agent Development Kit & Agent2Agent Protocol)

COMMIT

What is ADK?

Google Agent Development Kit (ADK) ← How Google builds their own multi-agent solutions

- Define **multi-agent** applications orchestrating actions across many agents and tools
- Intuitive local **dev UI** for fast iteration; visualize agent topology and trace agent's actions (observability & eval are baked in)
- Interleave **deterministic logic** with actions driven by **gen AI reasoning** for effortless, hybrid agents
- Built-in support for **long-running sync tools / human-in-the-loop**
- Gemini is the default, but **any generative AI model** is supported, including fine-tuned models
- Support to call remote agents and thousands of existing tools via **MCP** and **A2A** protocols



Core Principles

ADK is designed to empower developers to build, manage, evaluate and deploy AI-powered agents

Code-First Approach:

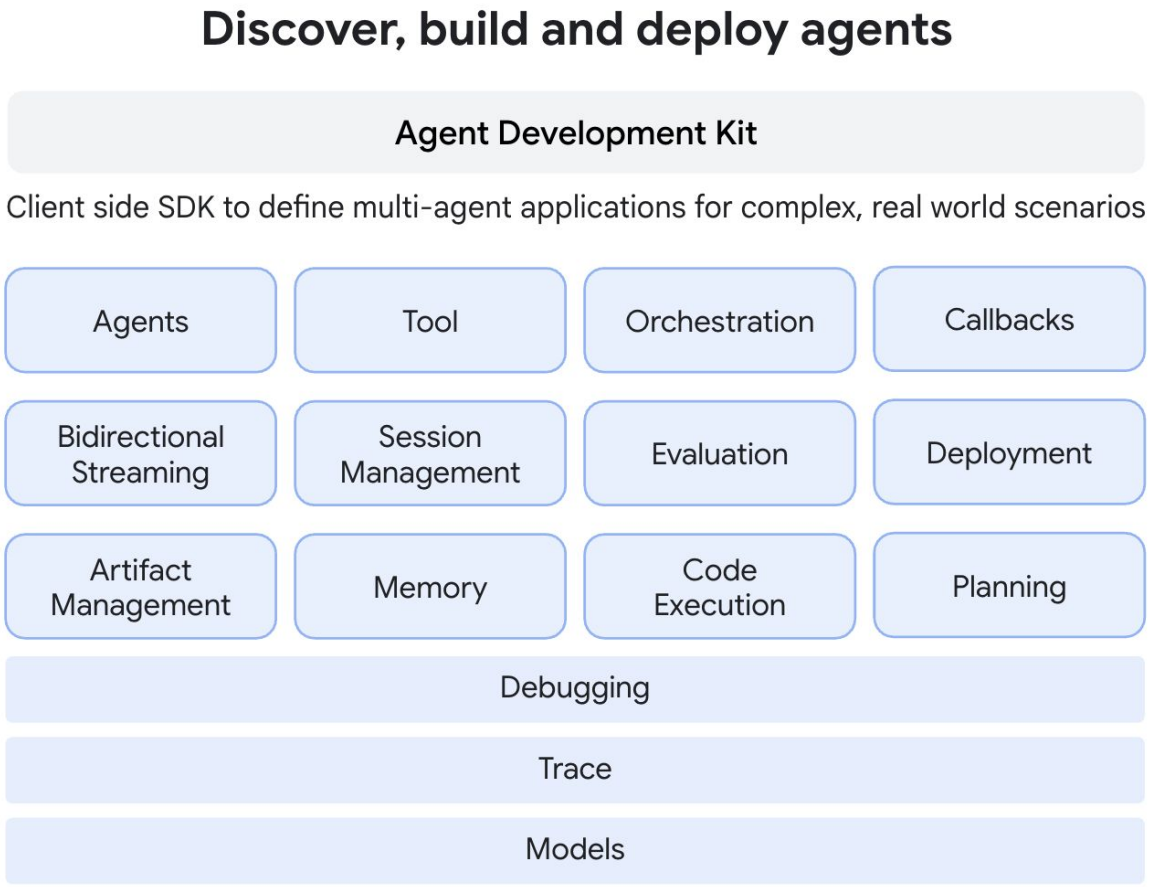
- Empowers developers to use familiar Python programming paradigms.
- Offers fine-grained control and customization.
- "Designed to make agent development feel like traditional software development."

Modularity & Flexibility:

- Build agents from reusable components (skills, tools)
- Easily integrate with existing systems and APIs.
- Adaptable to a wide range of agent complexity and use cases.

Developer Experience:

- Streamlined tools for testing, debugging, and iteration.
- Aims for rapid prototyping and "instant productivity"



Building our First Agent with ADK

Very easy to define an agent, or a multi-agent application

```python
from google.adk.agents import Agent
from google.adk.tools import google_search

root_agent = Agent(
    model='gemini-2.0-flash-001',
    name='root_agent',
    description='A helpful assistant for user questions.',
    instruction='Answer user questions to the best of your knowledge',
    tools=[google_search],
)
```

MCP Tools in ADK

ADK helps you both use and consume MCP tools in your agents

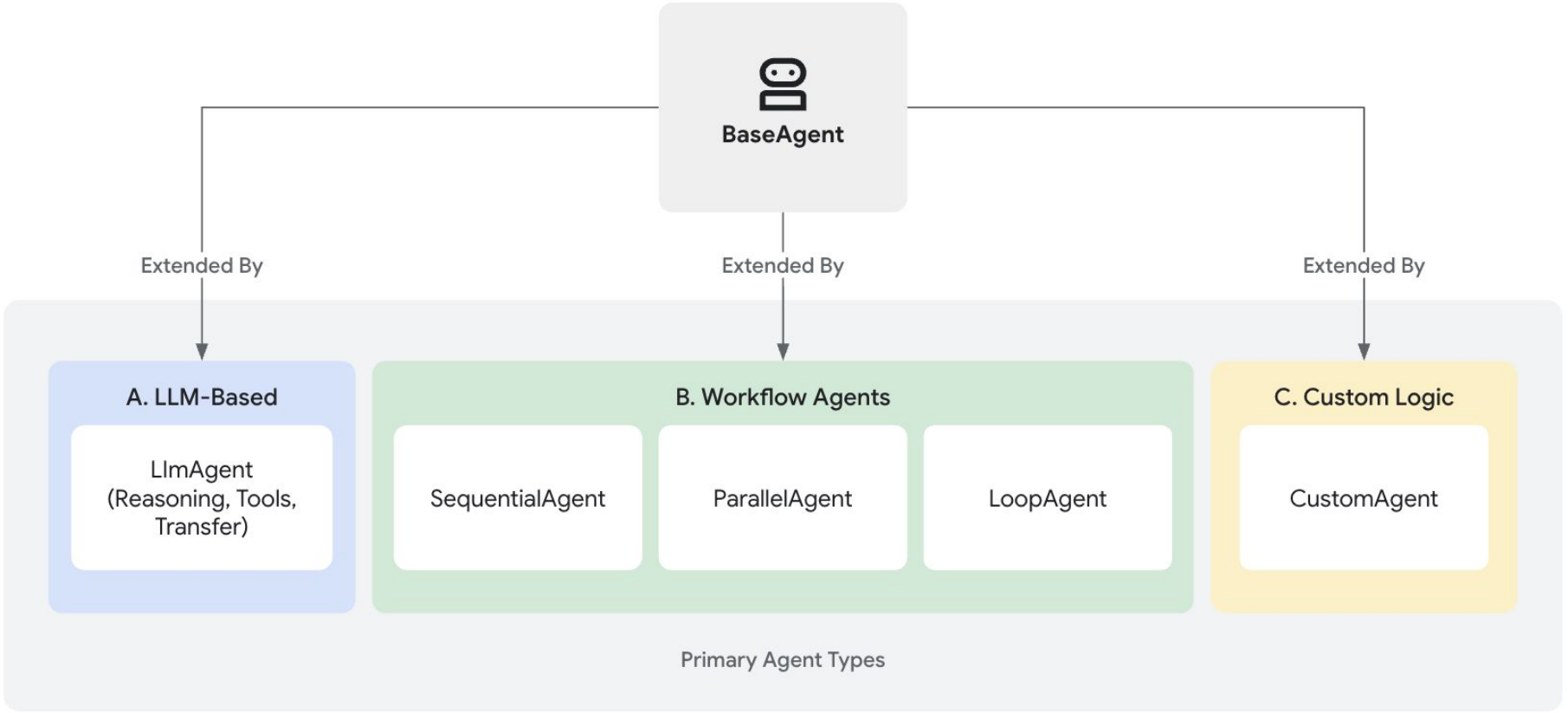
```python
import os
from google.adk.tools.mcp_tool.mcp_toolset import (
    MCPToolset,
    StdioServerParameters,
)
from dotenv import load_dotenv

load_dotenv()

NOTION_API_KEY = os.getenv("NOTION_API_KEY")

async def get_notion_tools():
    """Gets tools from the File System MCP Server."""
    print("Attempting to connect to MCP Filesystem server...")
    tools, exit_stack = await MCPToolset.from_server(
        # Use StdioServerParameters for local process communication
        connection_params=StdioServerParameters(
            command="npx",  # Command to run the server
            args=[
                "-y",
                "@notionhq/notion-mcp-server"
            ],
            env={
                "OPENAPI_MCP_HEADERS":
                f"{{\"Authorization\": \"Bearer {NOTION_API_KEY}\", \"Notion-Version\": \"2022-06-28\" }}"
            }
        )
    )
    print("MCP Toolset created successfully.")
    return tools, exit_stack
```

Primary Integration Patterns

- **Using Existing MCP Servers within ADK:** An ADK agent acts as an MCP client, leveraging tools provided by external MCP servers.
- **Exposing ADK Tools via an MCP Server:** Building an MCP server that wraps ADK tools, making them accessible to any MCP client.

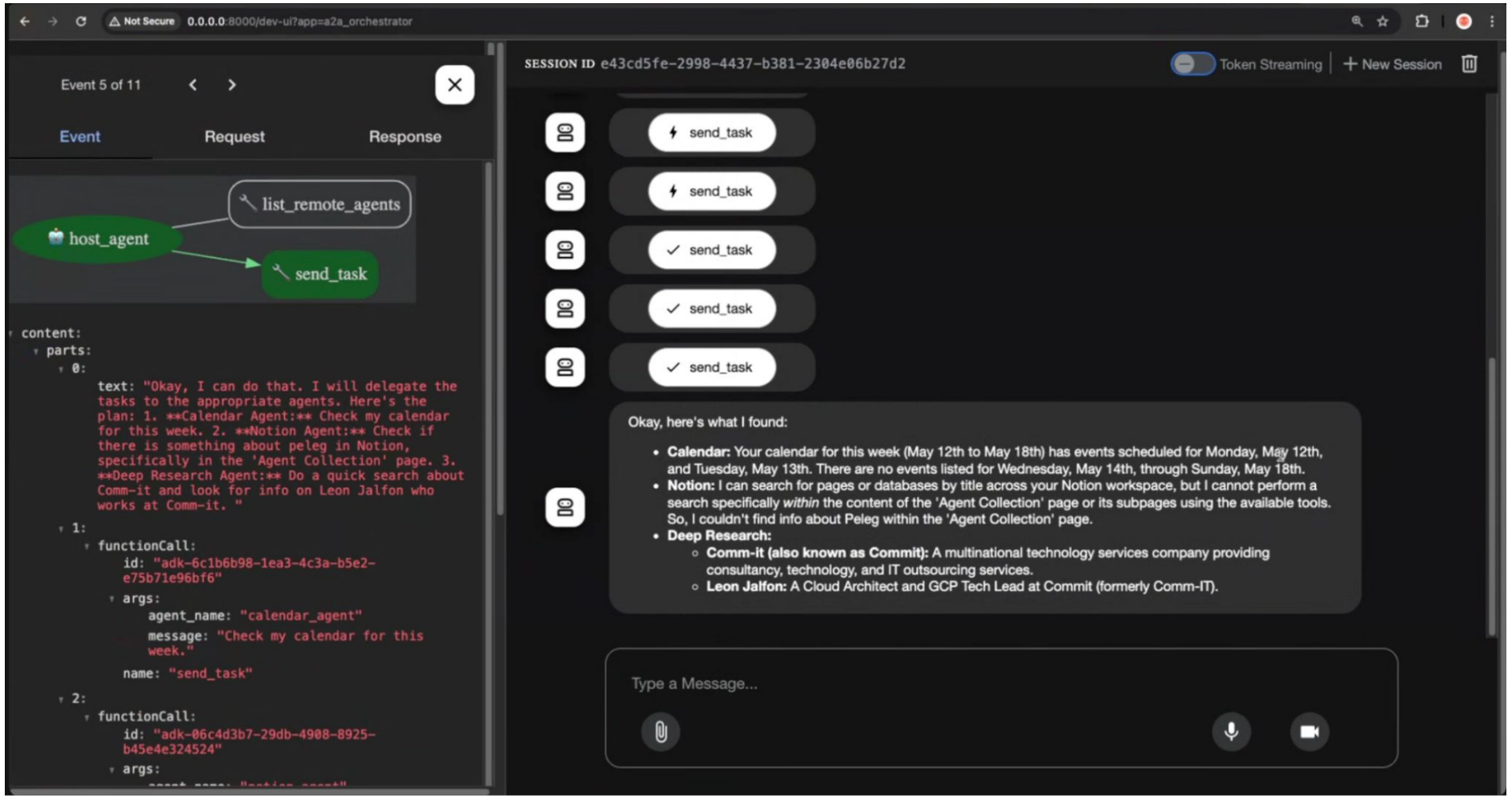
The “BaseAgent” Class

ADK provides distinct agent categories to build sophisticated applications



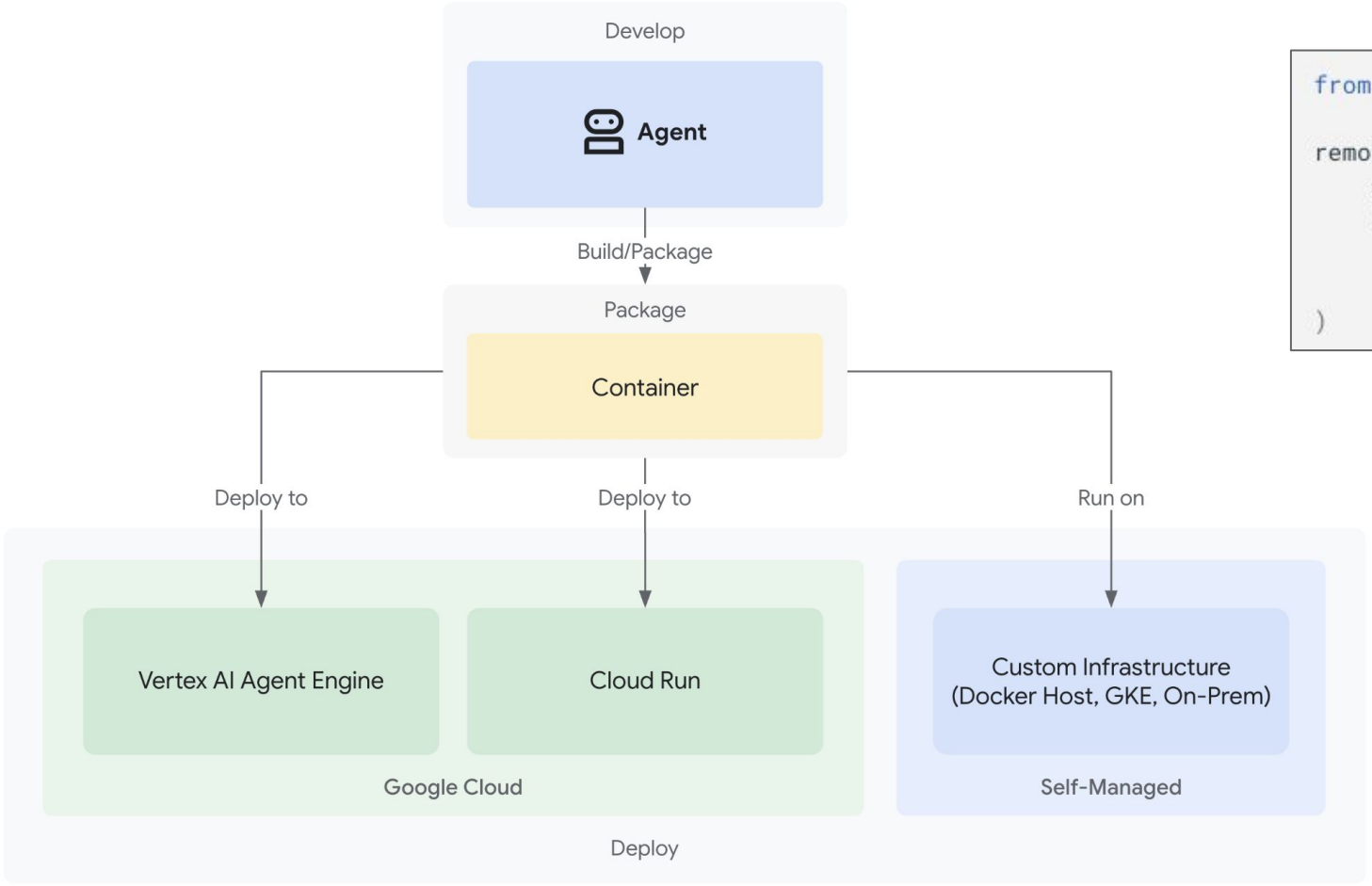
Development UI

A built-in development UI to help you test, evaluate, debug, and showcase your agents



Deploying Your Agent

Move your agent from your local development machine to a scalable and reliable environment

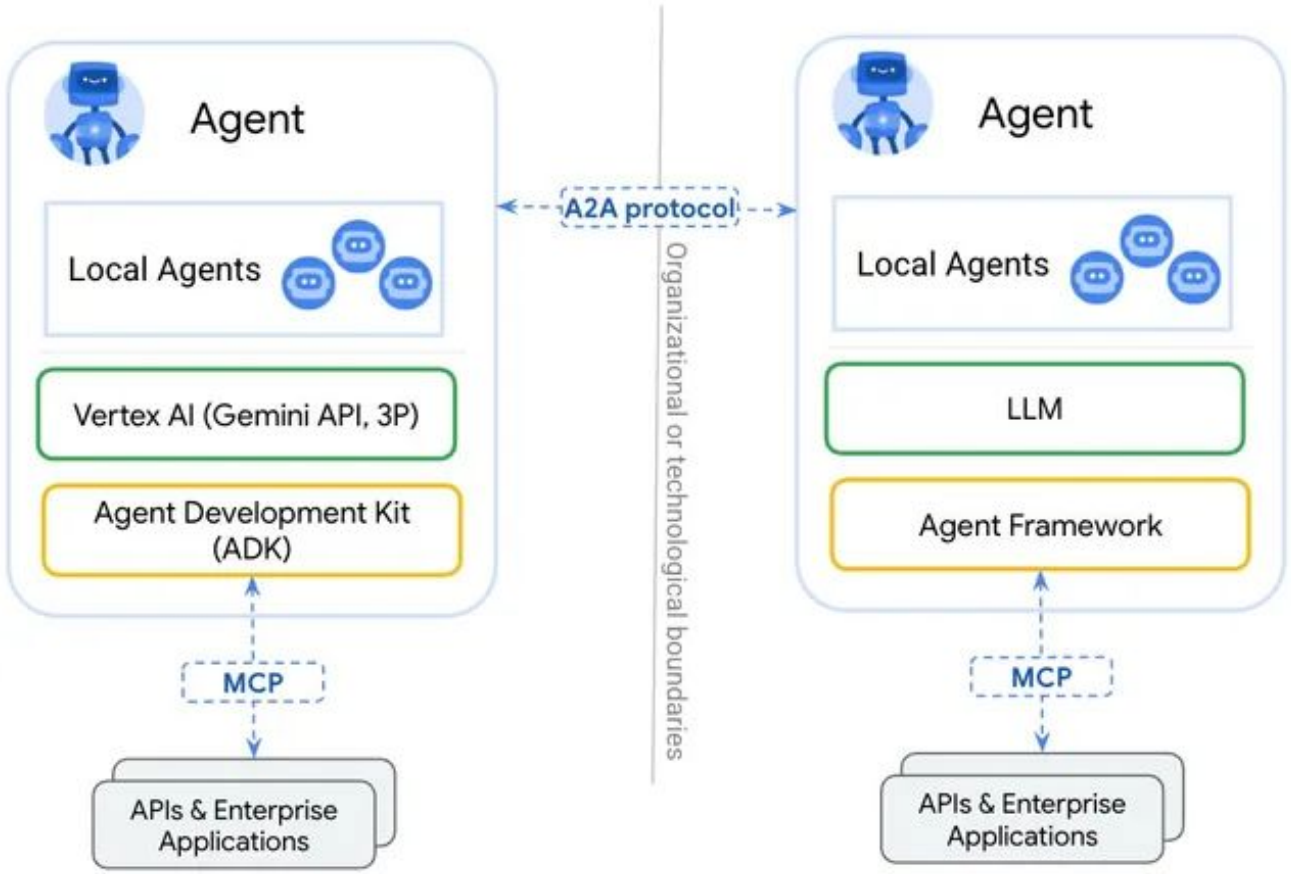



```python
from vertexai import agent_engines

remote_app = agent_engines.create(
    agent_engine=root_agent,
    requirements=[
        "google-cloud-aiplatform[adk,agent_engines]"
    ]
)
```

```
adk deploy cloud_run \
--project=$GOOGLE_CLOUD_PROJECT \
--region=$GOOGLE_CLOUD_LOCATION \
--service_name=$SERVICE_NAME \
--app_name=$APP_NAME \
--with_ui \
$AGENT_PATH
```

What is A2A?

A2A Protocol is an open standard that enables AI agents to communicate and collaborate across different platforms and frameworks, regardless of their underlying technologies. It's designed to maximize the benefits of agentic AI by enabling true multi-agent scenarios.



Key Features

Secure by Default

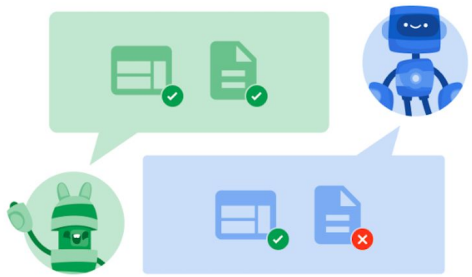
Enterprise-grade authentication and authorization with support for OpenAPI's authentication schemes

Open Standard

Built on existing standards including HTTP, SSE, and JSON-RPC for easy integration with existing IT stacks

Multi-Modal Support

Supports various modalities including text, audio, and video streaming for comprehensive agent communication



Agent Collaboration

Enables true multi-agent scenarios where agents can collaborate in their natural, unstructured modalities

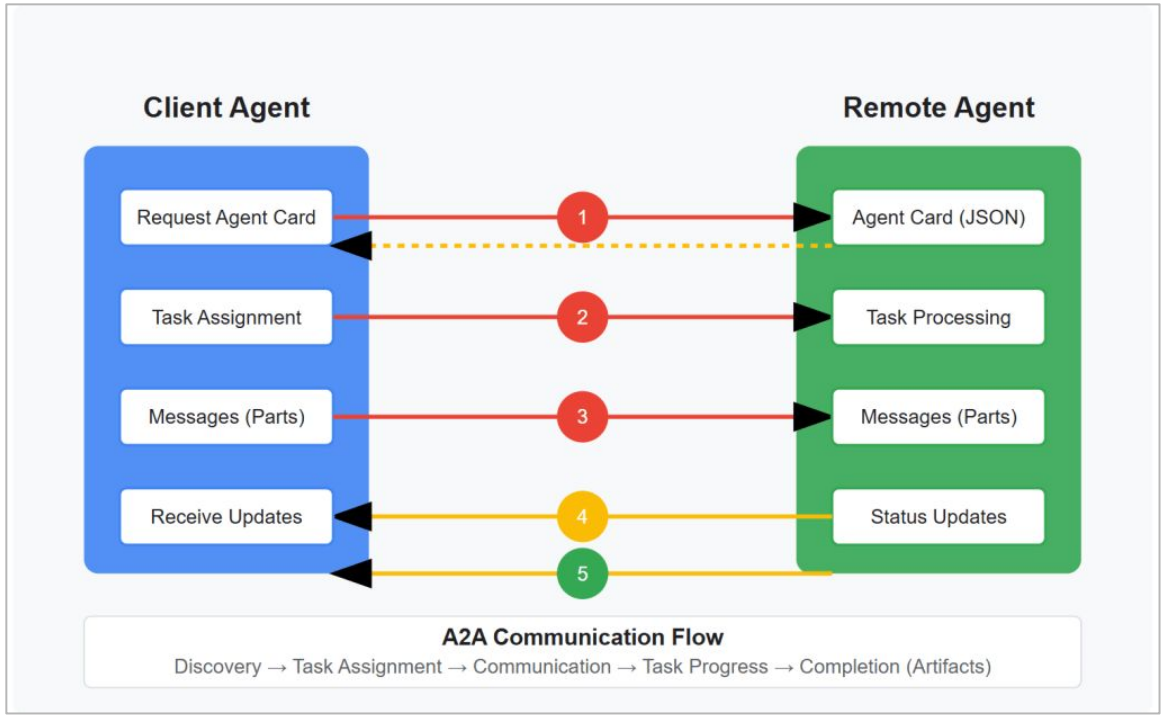
Real-Time Updates

Provides real-time feedback, notifications, and state updates throughout the task lifecycle

Long-Running Tasks

Designed to support both quick tasks and deep research that may take hours or days to complete

How it Works




```json
{
  "name": "deep_research_agent",
  "description": "A research assistant that can conduct deep research or quick search on a given topic.",
  "url": "http://localhost:10003/",
  "version": "1.0.0",
  "capabilities": {
    "streaming": true,
    "pushNotifications": false,
    "stateTransitionHistory": false
  },
  "defaultInputModes": [
    "text"
  ],
  "defaultOutputModes": [
    "text"
  ],
  "skills": [
    {
      "id": "deep_research_agent",
      "name": "deep_research_agent",
      "description": "A research assistant that can conduct deep research or quick search on a given topic.",
      "tags": [
        "research",
        "deep_dive",
        "quick_search",
        "context_gathering",
        "report_generation",
        "information_synthesis",
        "google_search",
        "multi_step_processing"
      ],
      "examples": [
        "Conduct a deep research investigation into the future of renewable energy and provide a comprehensive report.",
        "Perform a quick search on the main competitors of Tesla Inc. and summarize the findings with sources.",
        "Gather initial context on the topic of 'quantum computing applications in medicine'.",
        "Generate a detailed report on the impact of AI on the job market, including references.",
        "What are the latest developments in gene editing? (quick search)"
      ],
      "inputModes": [
        "text"
      ],
      "outputModes": [
        "text"
      ]
    }
  ]
}
```


```python
from .host_agent import HostAgent
from google.adk.agents import LlmAgent

python_dev_agent_address = "http://localhost:10000"
people_info_agent_address = "http://localhost:10001"
google_calendar_agent_address = "http://localhost:10002"
deep_research_agent_address = "http://localhost:10003"
notion_agent_address = "http://localhost:10005"


root_agent = HostAgent(
    [
        python_dev_agent_address,
        people_info_agent_address,
        google_calendar_agent_address,
        deep_research_agent_address,
        notion_agent_address
    ]
).create_agent()
```


IT'S
DEMO TIME!
makeameme.org

Thank You

commitgcp/commit-adk

Leon Jalfon
Jonathan Jalfon