

# Unboxing ADK & A2A

Next Wave AI Meetup

Leon Jalfon

Jonathan Jalfon



May 15, 2024

# commit

---



Google Cloud

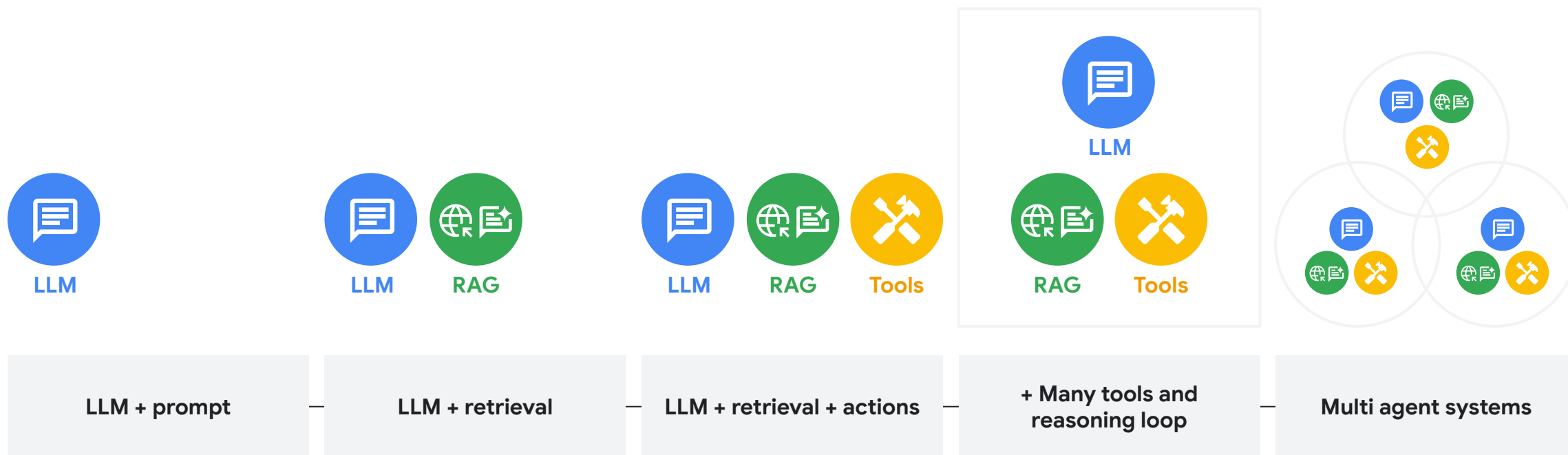



The AI revolution has redefined **possible**,  
clearing the path for **innovation**...

...so, **stop wondering** and **start building**, your  
ability to build **AI agents** is here and **now!**

# AI Agents

History of the agents evolution





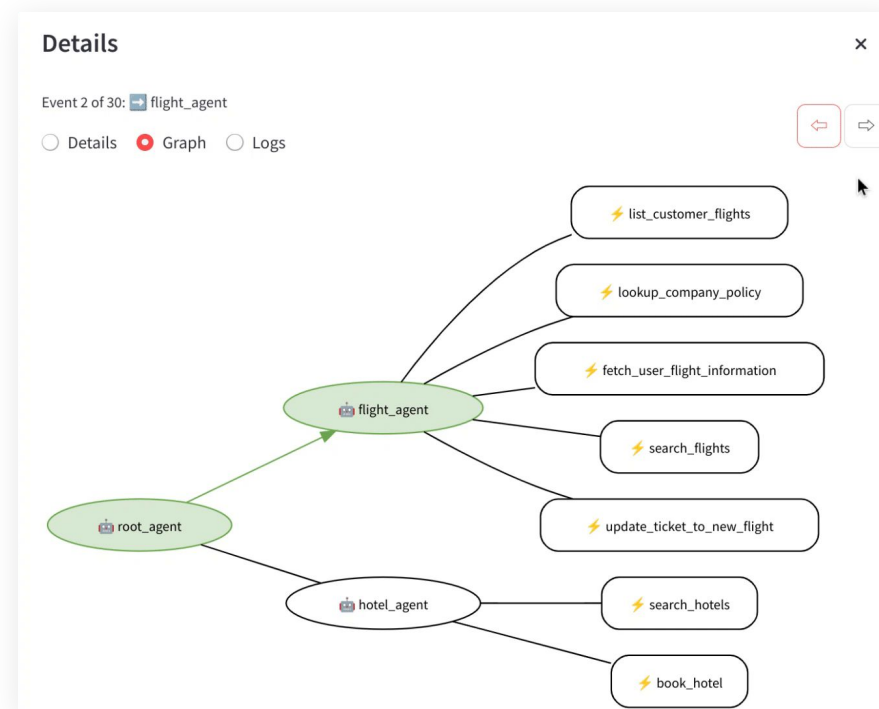
# Unboxing ADK & A2A

**(Agent Development Kit & Agent2Agent Protocol)**

# 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.

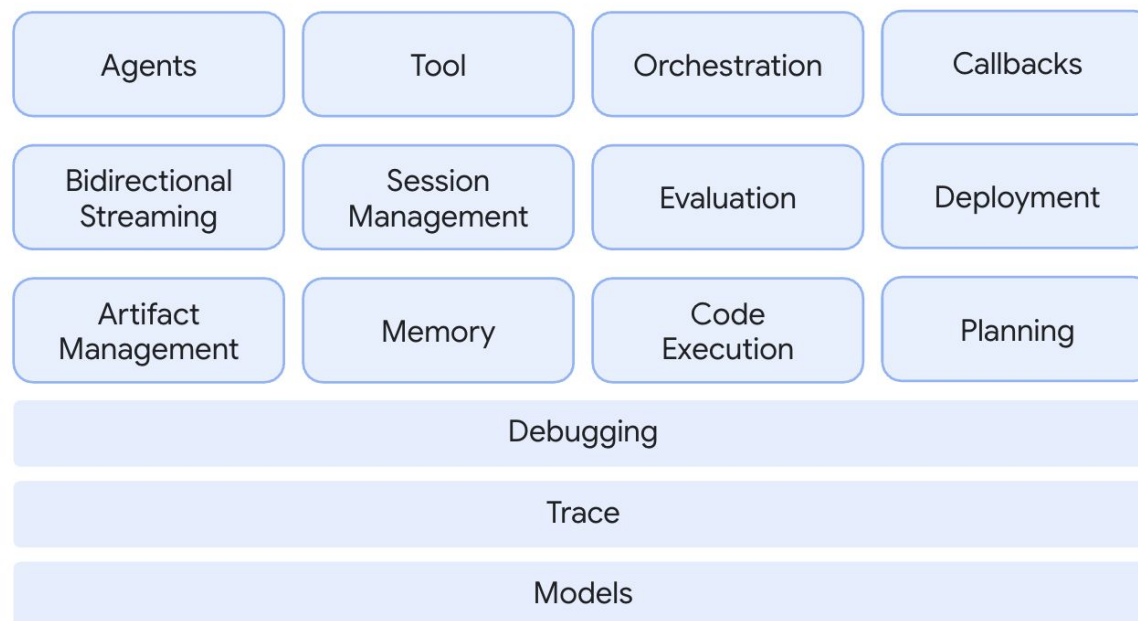
## Developer Experience:

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

## Discover, build and deploy agents

### Agent Development Kit

Client side SDK to define multi-agent applications for complex, real world scenarios



# Building our First Agent with ADK

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

```
1  from google.adk.agents import Agent
2  from google.adk.tools import google_search
3
4  ✓ root_agent = Agent(
5      model='gemini-2.0-flash-001',
6      name='root_agent',
7      description='A helpful assistant for user questions.',
8      instruction='Answer user questions to the best of your knowledge',
9      tools=[google_search],
10 )
```

# MCP Tools in ADK

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

```
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
```

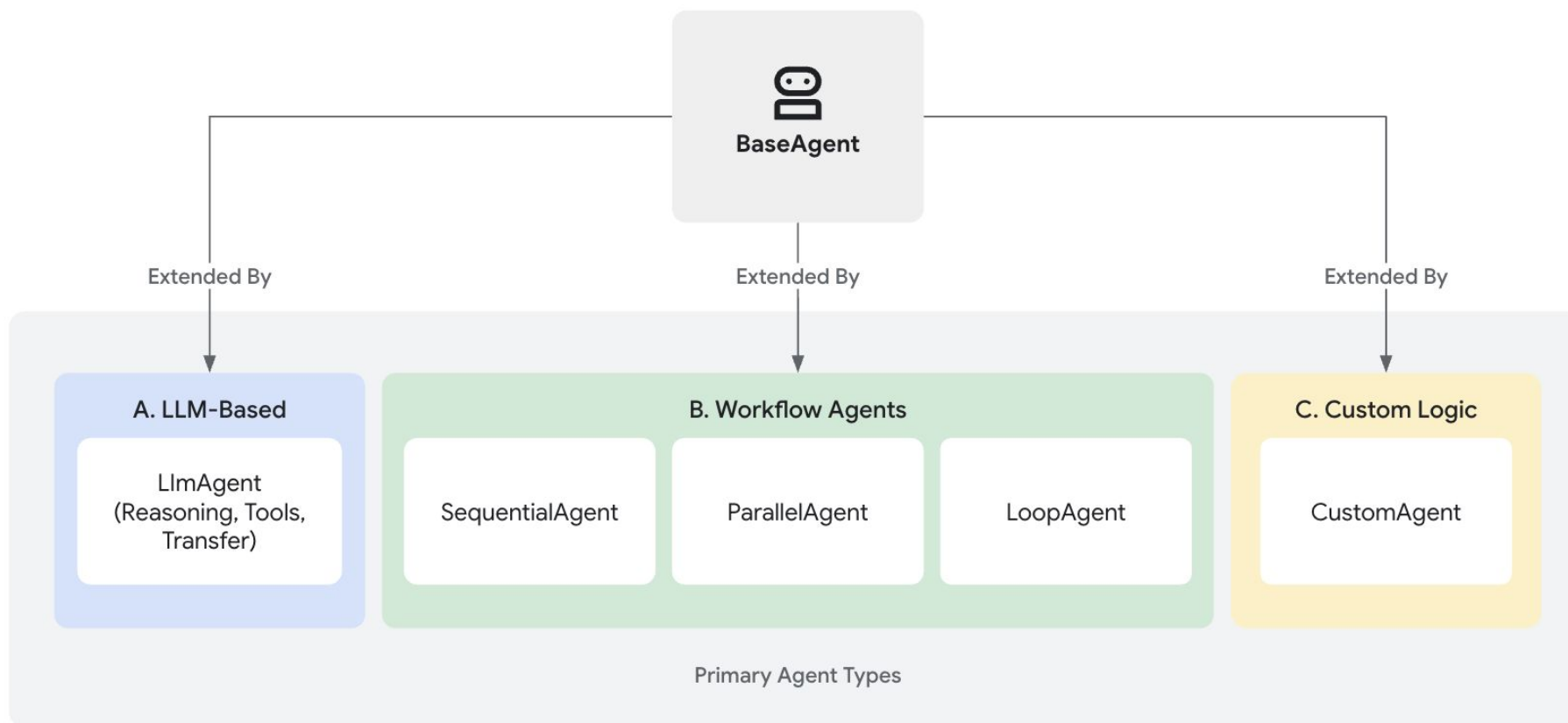
## Primary Integration Patterns

- **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

The screenshot displays the Commit Development UI interface. The top bar shows the session ID: e43cd5fe-2998-4437-b381-2304e06b27d2. The interface is divided into two main sections: a left sidebar and a right main area.

**Left Sidebar:**

- Event 5 of 11
- Event, Request, Response tabs
- Diagram showing a host agent connected to list\_remote\_agents and send\_task functions.
- Content area showing a JSON response:

```
content:
  parts:
    0:
      text: "Okay, I can do that. I will delegate the tasks to the appropriate agents. Here's the plan: 1. **Calendar Agent:** Check my calendar for this week. 2. **Notion Agent:** Check if there is something about peleg in Notion, specifically in the 'Agent Collection' page. 3. **Deep Research Agent:** Do a quick search about Comm-it and look for info on Leon Jalfon who works at Comm-it. "
```

**Right Main Area:**

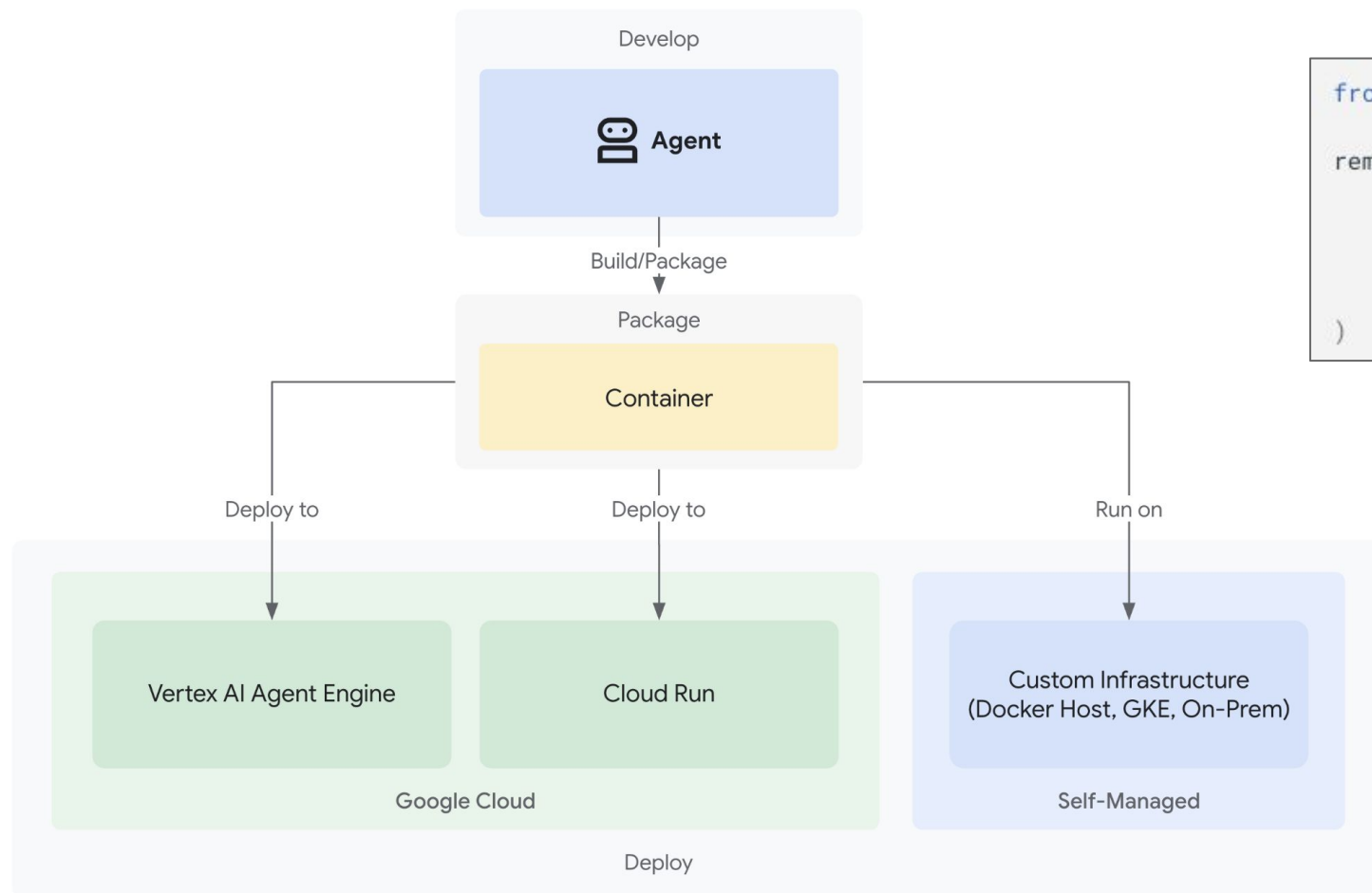
- Token Streaming toggle and New Session button.
- Five send\_task buttons, each preceded by a status icon (lightning bolt or checkmark).
- Text area: "Okay, here's what I found:"
- List of findings:

- Calendar:** Your calendar for this week (May 12th to May 18th) has events scheduled for Monday, May 12th, and Tuesday, May 13th. There are no events listed for Wednesday, May 14th, through Sunday, May 18th.
- Notion:** I can search for pages or databases by title across your Notion workspace, but I cannot perform a search specifically *within* the content of the 'Agent Collection' page or its subpages using the available tools. So, I couldn't find info about Peleg within the 'Agent Collection' page.
- Deep Research:**
  - Comm-it (also known as Commit):** A multinational technology services company providing consultancy, technology, and IT outsourcing services.
  - Leon Jalfon:** A Cloud Architect and GCP Tech Lead at Commit (formerly Comm-IT).

At the bottom, there is a "Type a Message..." input field with icons for attachments, voice, and video.

# Deploying Your Agent

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



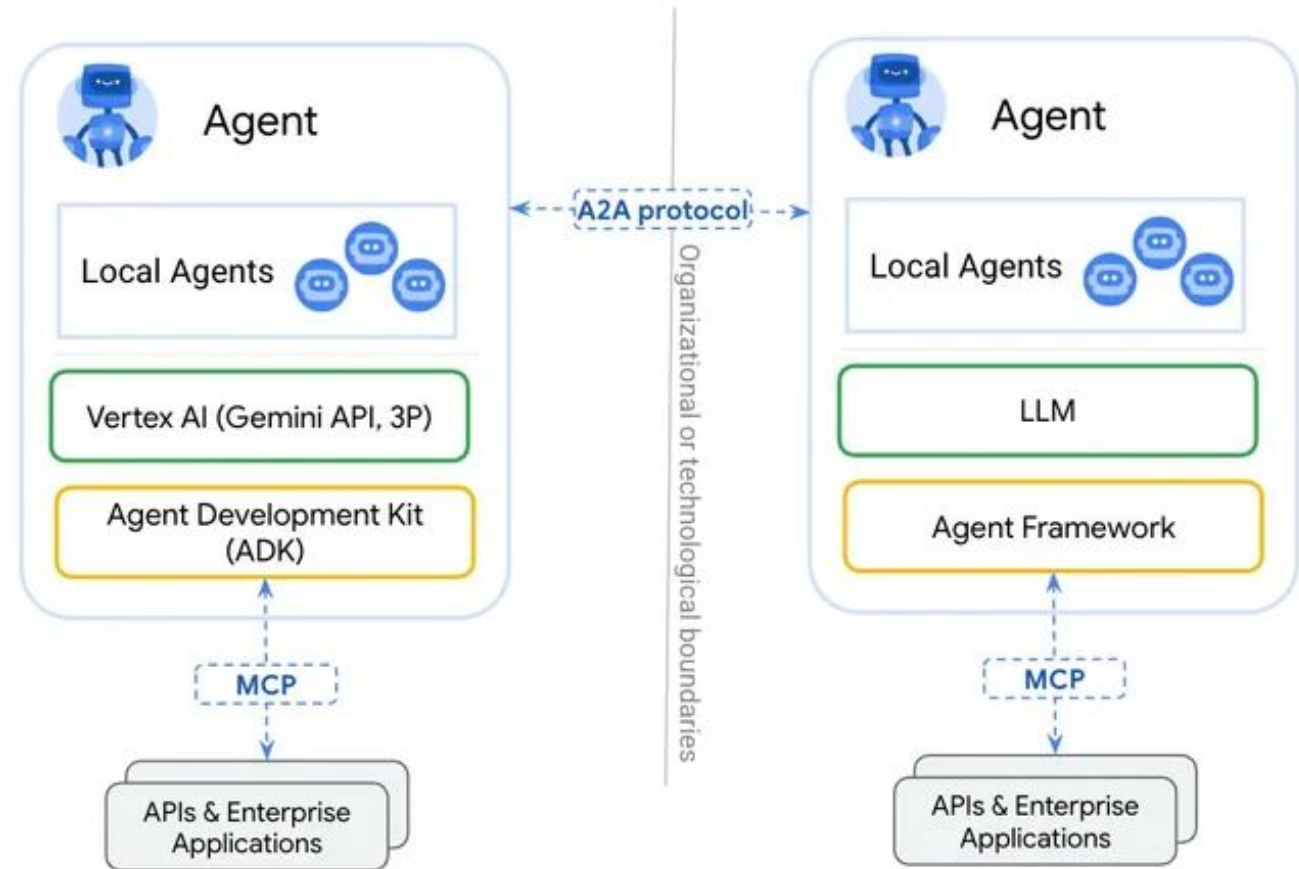
```
from vertexai import agent_engines

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

## <> Open Standard

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

## ⚙️ Agent Collaboration

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



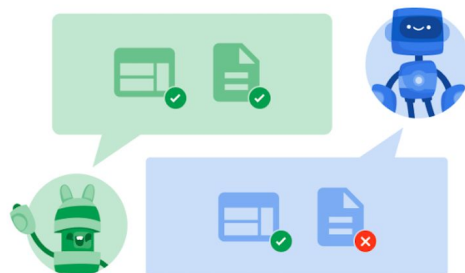
## Secure by Default

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



## Multi-Modal Support

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



## Real-Time Updates

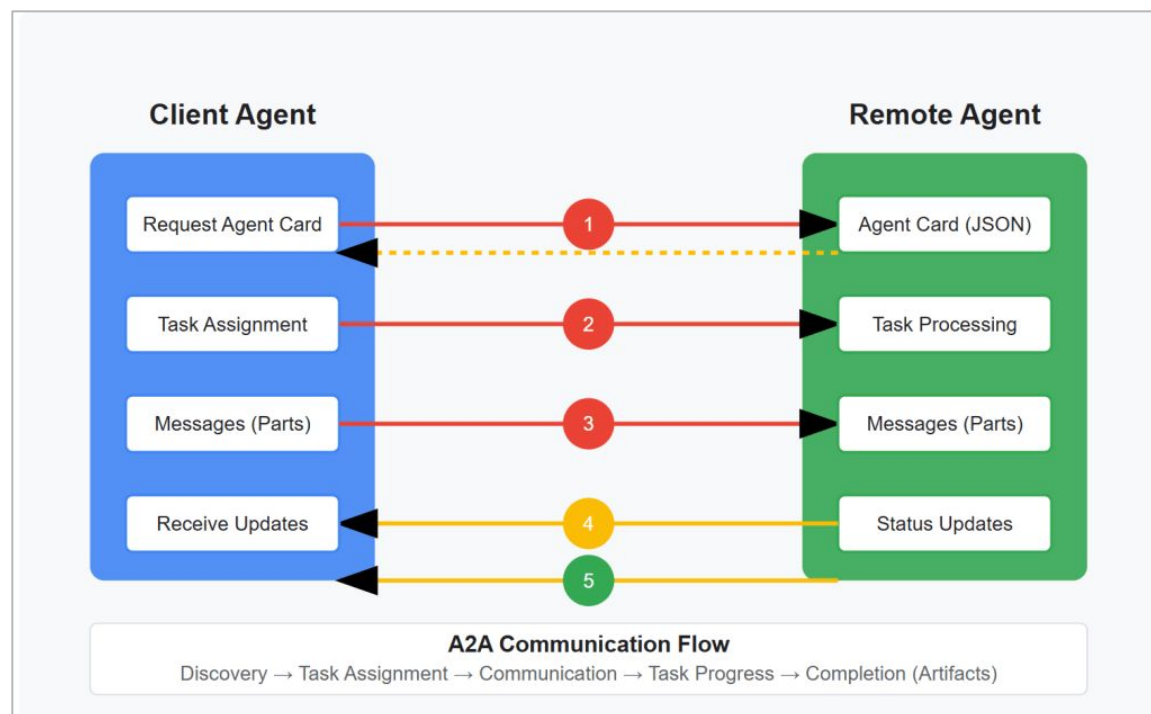
Provides real-time feedback, notifications, and state updates throughout the task lifecycle



## Long-Running Tasks

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

# How it Works



```
{
  "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"
      ]
    }
  ]
}
```

```
1 from .host_agent import HostAgent
2 from google.adk.agents import LlmAgent
3
4 python_dev_agent_address = "http://localhost:10000"
5 people_info_agent_address = "http://localhost:10001"
6 google_calendar_agent_address = "http://localhost:10002"
7 deep_research_agent_address = "http://localhost:10003"
8 notion_agent_address = "http://localhost:10005"
9
10
11 root_agent = HostAgent(
12     [
13         python_dev_agent_address,
14         people_info_agent_address,
15         google_calendar_agent_address,
16         deep_research_agent_address,
17         notion_agent_address
18     ]
19 ).create_agent()
```





# Thank You

Leon Jalfon  
Jonathan Jalfon



[commitgcp/commit-adk](https://github.com/commitgcp/commit-adk)

