Google Cloud

An overview of AI agents

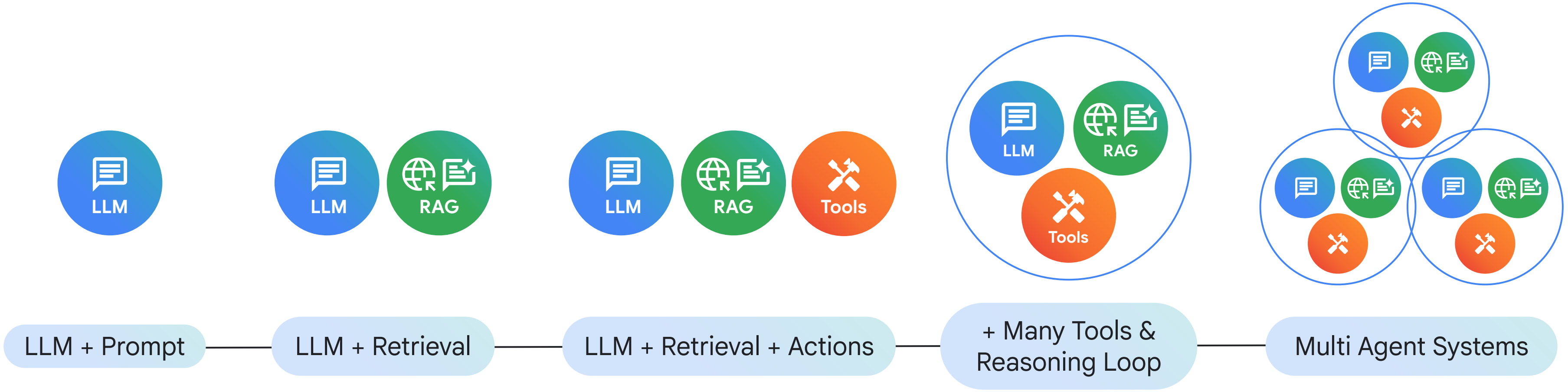
Architectures, key concepts and applications

Dr. Wafae Bakkali

Staff Generative AI Specialist, Blackbelt, Google

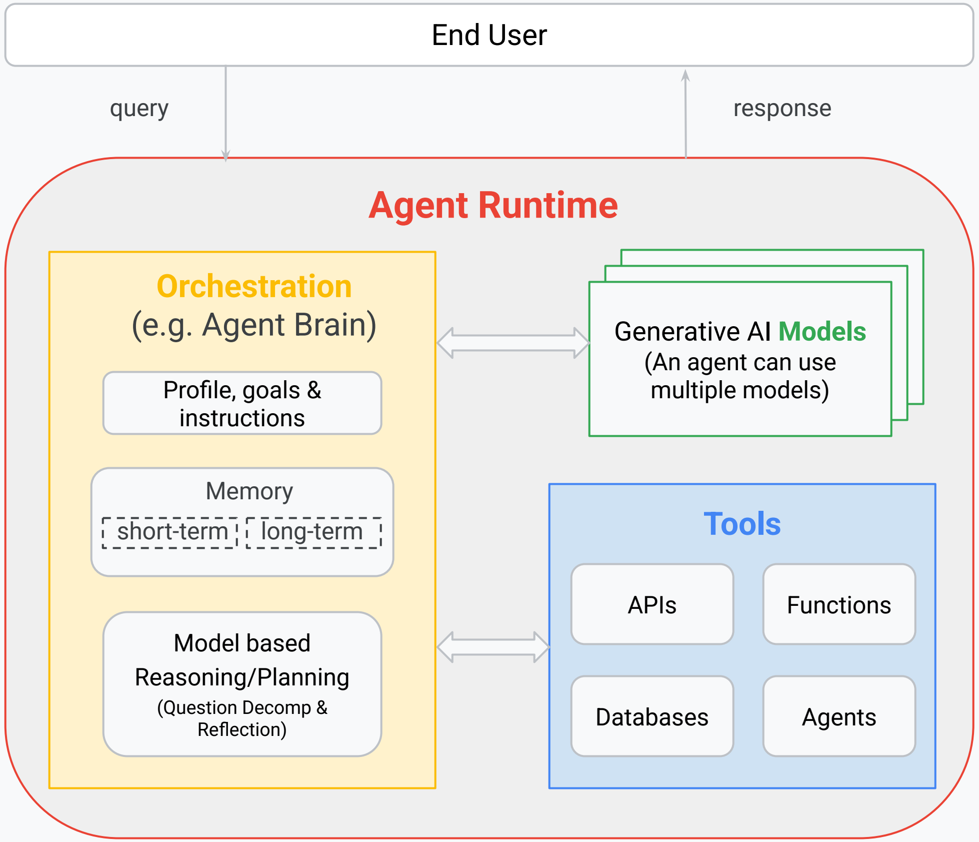
AI for Good Global Summit, July 2025

AI Agents
An Evolution



What is next?

The Agent Architecture



Four Key Components

- **Model:** 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

Agent architectures (complexity and architecture)

Single agent Architecture

Powered by a **single LLM** that performs all the reasoning, planning and actions. Simplest architecture to set up.

Benefits

- Easier to implement

Challenges

- More prone to get stuck in execution loop

Multi-Agent Architecture

Powered by two or more agents that can be used to coordinate, collaborate & specialize

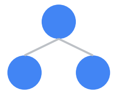
Benefits

- Use specialized agents for specific tasks and to drive efficiencies

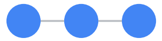
Challenges

- More complex to setup and maintain
- Horizontal architectures can lead to group chat and loss of focus
- Vertical architectures susceptible to leading agent not sending critical information to other agents

Hierarchical



Horizontal



Agent Design

Levels of Abstraction

Level 4: No-code platforms

Level 3: Agent framework

Level 2: Low-level orchestration framework

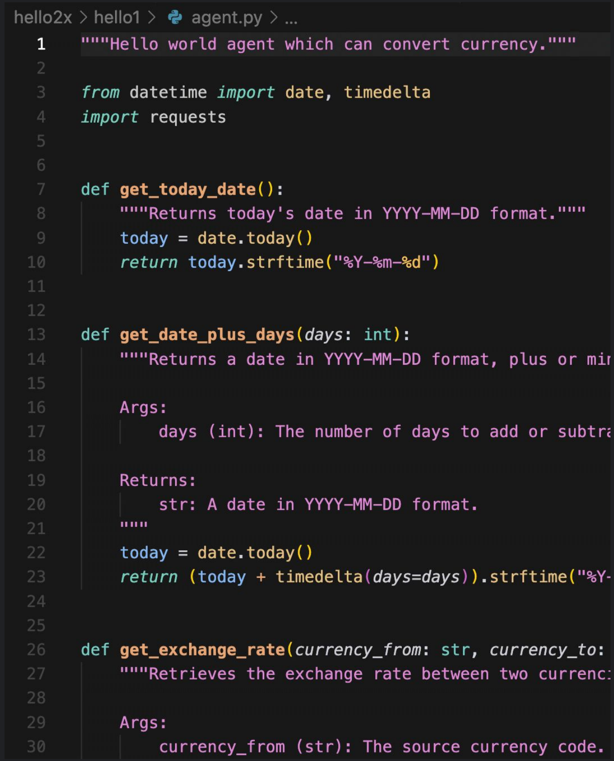
Level 1: Low-level LLM framework

Level 0: DIY

Agent Development Kit (ADK)

Develop Agents Easily

- ✓ Develop Multi-agent Solutions Easily
- ✓ Robust Session Management
- ✓ Multimodal is the Present
- ✓ Asynchronous First
- ✓ Transform Agents to Live
- ✓ Open Source including UI



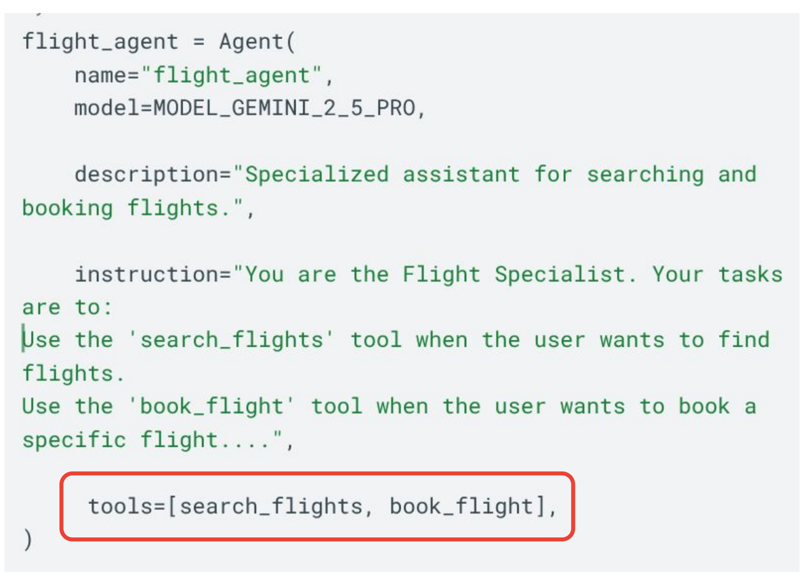
hello2x > hello1 > agent.py > ...

```python
"""Hello world agent which can convert currency."""

from datetime import date, timedelta
import requests


def get_today_date():
    """Returns today's date in YYYY-MM-DD format."""
    today = date.today()
    return today.strftime("%Y-%m-%d")


def get_date_plus_days(days: int):
    """Returns a date in YYYY-MM-DD format, plus or mi

    Args:
        days (int): The number of days to add or subtra

    Returns:
        str: A date in YYYY-MM-DD format.
    """
    today = date.today()
    return (today + timedelta(days=days)).strftime("%Y-


def get_exchange_rate(currency_from: str, currency_to:
    """Retrieves the exchange rate between two currenc

    Args:
        currency_from (str): The source currency code.
```

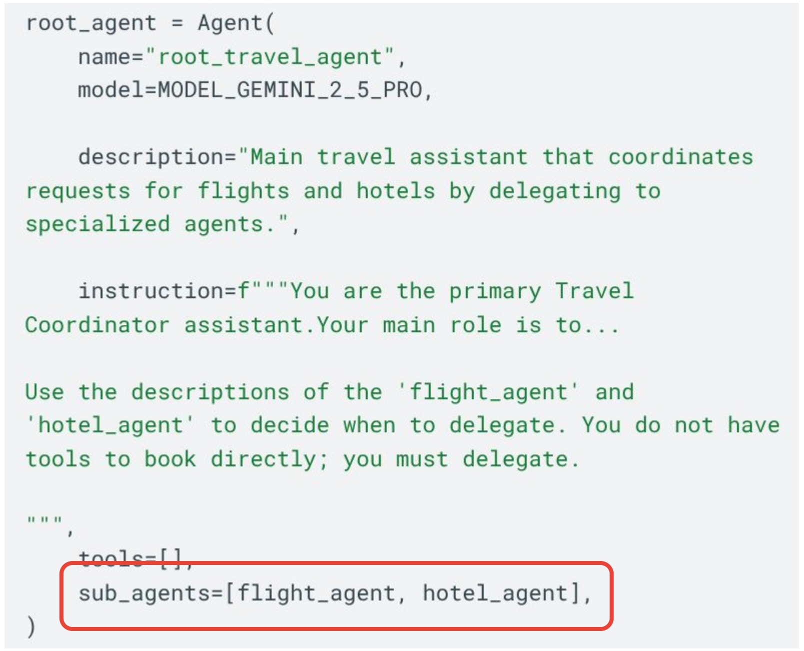
Models | Instructions | Tools (APIs and Data) | Session/Memory | Multi-agent Orchestration

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

Minimal boilerplate code



```
flight_agent = Agent(
    name="flight_agent",
    model=MODEL_GEMINI_2_5_PRO,

    description="Specialized assistant for searching and
booking flights.",

    instruction="You are the Flight Specialist. Your tasks
are to:
Use the 'search_flights' tool when the user wants to find
flights.
Use the 'book_flight' tool when the user wants to book a
specific flight....",

     tools=[search_flights, book_flight],
)
```



```
root_agent = Agent(
    name="root_travel_agent",
    model=MODEL_GEMINI_2_5_PRO,

    description="Main travel assistant that coordinates
requests for flights and hotels by delegating to
specialized agents.",

    instruction=f"""You are the primary Travel
Coordinator assistant.Your main role is to...

Use the descriptions of the 'flight_agent' and
'hotel_agent' to decide when to delegate. You do not have
tools to book directly; you must delegate.

""",
    tools=[],
    sub_agents=[flight_agent, hotel_agent],
)
```

Support for building different types of Agents

A programmatic SDK offers a high degree of flexibility

Interactive Chat Agents

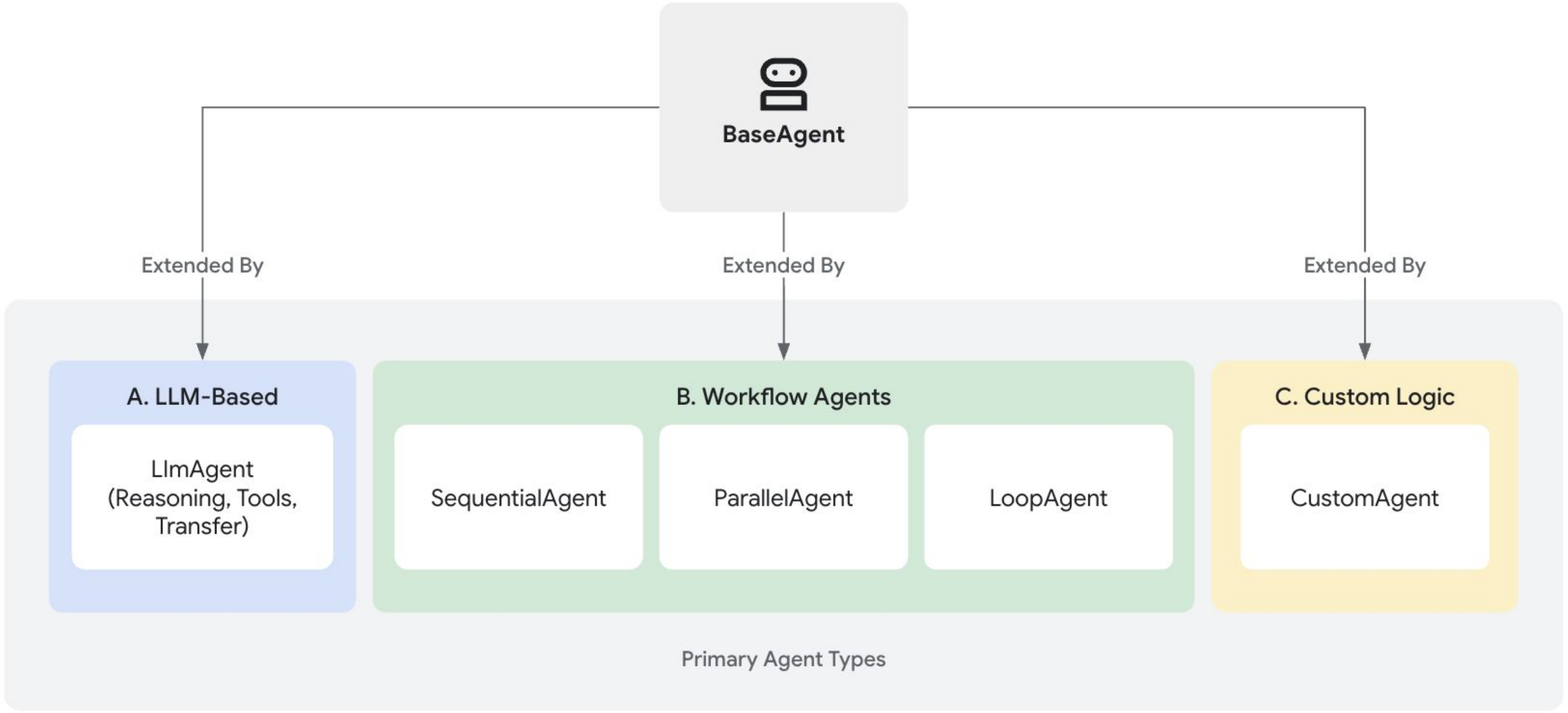
Perhaps the most common type of agent. Interact with the agent via chat messages, and get responses back with low latency.

Background Processing Agents

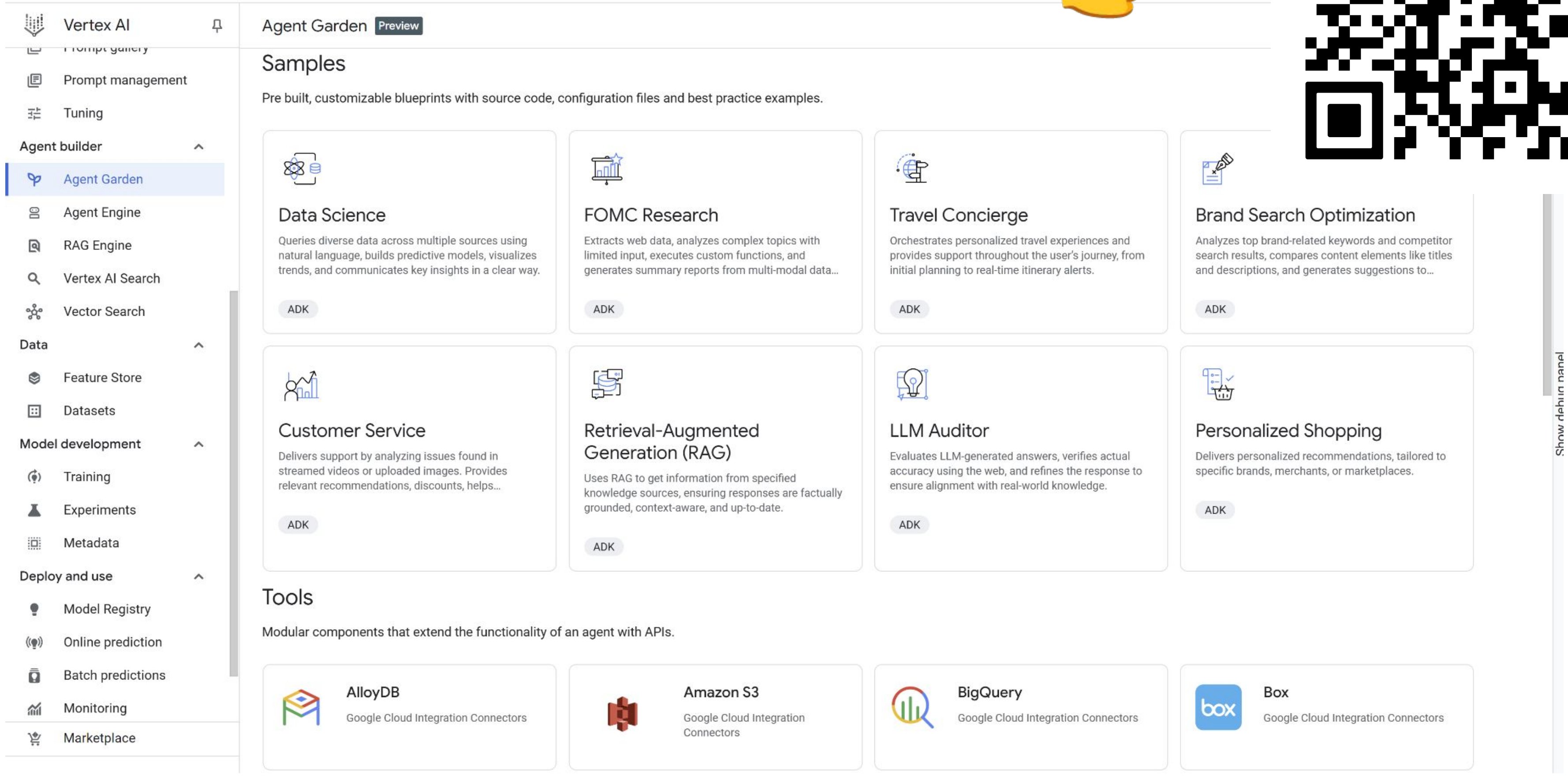
Agents that run in the background, monitor systems or data, make intelligent decisions, perform offline data processing and notify humans and escalate only if required. These could be one-off or recurring jobs.

Real-time Audio/Video streaming Agents

Build agents that can have low latency bi-directional voice and video interaction with end users. This enables building agents that one can have natural human like interactions with.


BaseAgent
Extended By
Extended By
Extended By
A. LLM-Based
LlmAgent (Reasoning, Tools, Transfer)
B. Workflow Agents
SequentialAgent
ParallelAgent
LoopAgent
C. Custom Logic
CustomAgent
Primary Agent Types

Sample agents



Vertex AI

Prompt gallery

Prompt management

Tuning

Agent builder

Agent Garden

Agent Engine

RAG Engine

Vertex AI Search

Vector Search

Data

Feature Store

Datasets

Model development

Training

Experiments

Metadata

Deploy and use

Model Registry

Online prediction

Batch predictions

Monitoring

Marketplace

Agent Garden Preview

Samples

Pre built, customizable blueprints with source code, configuration files and best practice examples.

Data Science
Queries diverse data across multiple sources using natural language, builds predictive models, visualizes trends, and communicates key insights in a clear way.
ADK

FOMC Research
Extracts web data, analyzes complex topics with limited input, executes custom functions, and generates summary reports from multi-modal data...
ADK

Travel Concierge
Orchestrates personalized travel experiences and provides support throughout the user's journey, from initial planning to real-time itinerary alerts.
ADK

Brand Search Optimization
Analyzes top brand-related keywords and competitor search results, compares content elements like titles and descriptions, and generates suggestions to...
ADK

Customer Service
Delivers support by analyzing issues found in streamed videos or uploaded images. Provides relevant recommendations, discounts, helps...
ADK

Retrieval-Augmented Generation (RAG)
Uses RAG to get information from specified knowledge sources, ensuring responses are factually grounded, context-aware, and up-to-date.
ADK

LLM Auditor
Evaluates LLM-generated answers, verifies actual accuracy using the web, and refines the response to ensure alignment with real-world knowledge.
ADK

Personalized Shopping
Delivers personalized recommendations, tailored to specific brands, merchants, or marketplaces.
ADK

Tools

Modular components that extend the functionality of an agent with APIs.

AlloyDB
Google Cloud Integration Connectors

Amazon S3
Google Cloud Integration Connectors

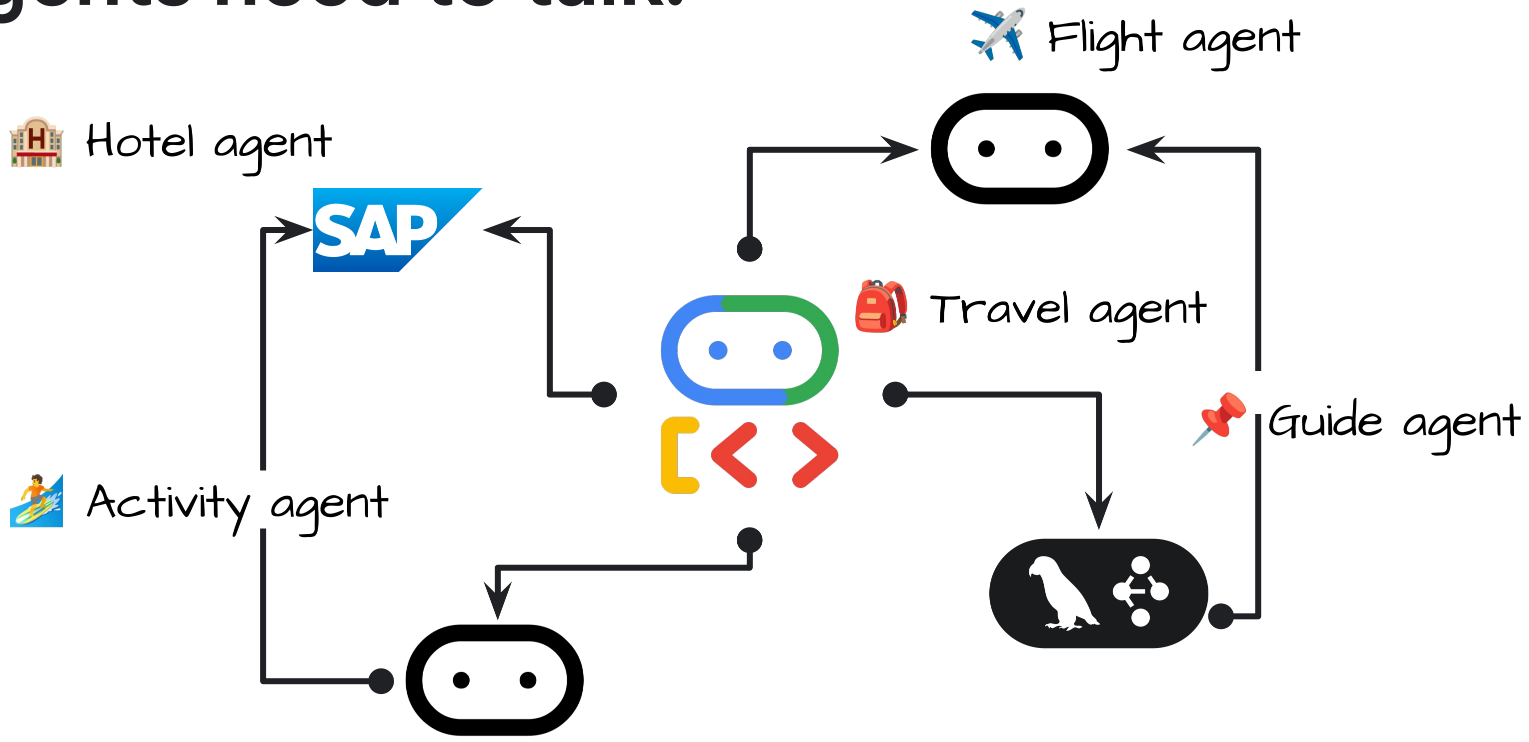
BigQuery
Google Cloud Integration Connectors

Box
Google Cloud Integration Connectors

Show debug panel

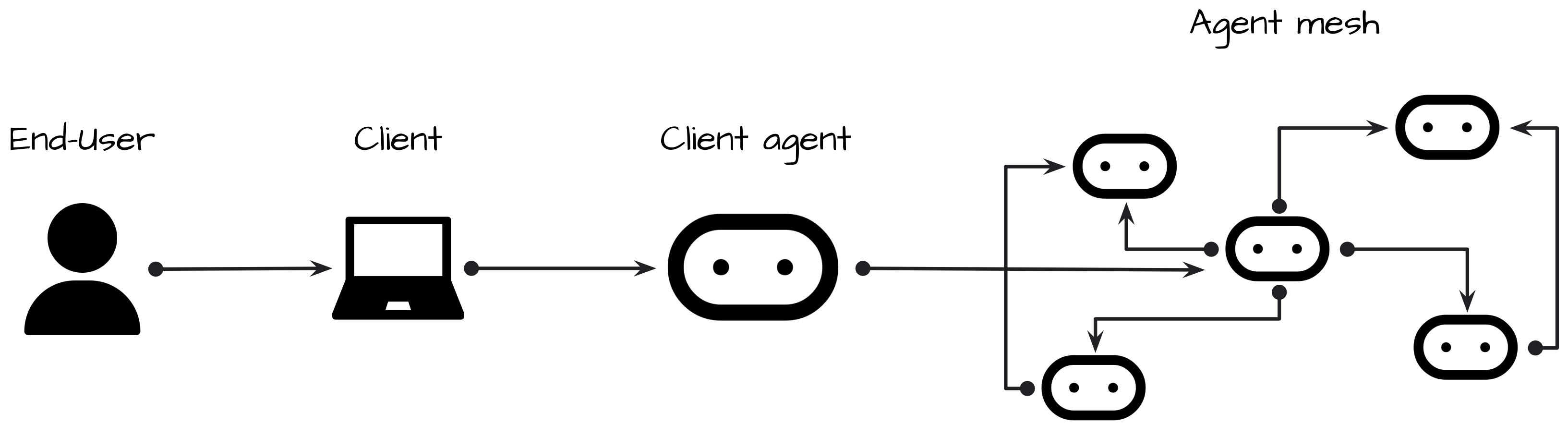
Agents interoperability

Agents need to talk!



Agent2Agent (A2A) protocol

Open protocol to handle agent collaboration



Google Cloud

Partners contributing to the Agent2Agent protocol

accenture arize Articul8 ask-ai ATLASSIAN BCG box C3.ai Capgemini

chronosphere cognizant cohere Collibra contextual·ai cotality DATADOG DataRobot DATASTAX

Decagon Deloitte. devnagri elastic Ema <epam> glean GrowthLoop harness

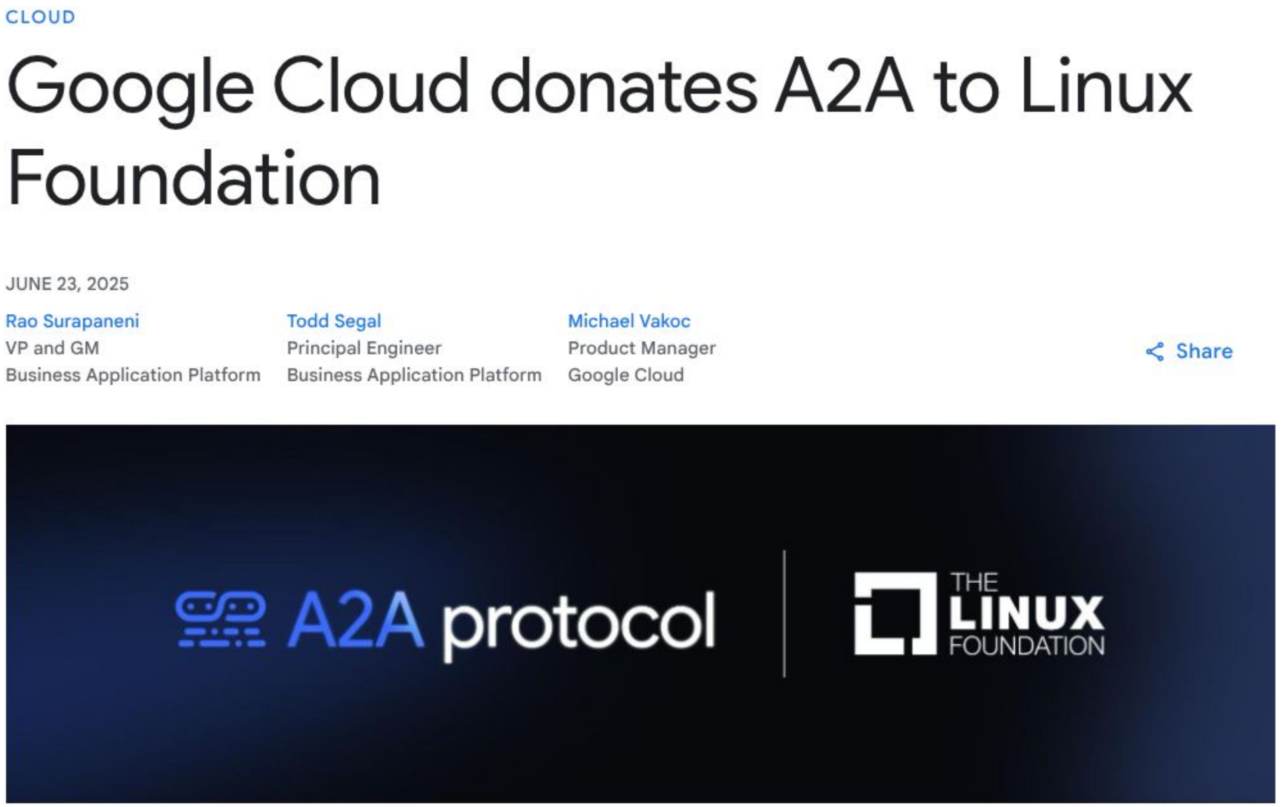
HCLTech IN—COR—TA Infosys INTUIT JET BRAINS JFrog KPMG Labelbox LangChain

LIVEX.AI LTIMindtree lyzr McKinsey & Company mongoDB. neo4j new relic ORACLE pendo

pwc quantiphi Solving What Matters S&P Global salesforce SAP servicenow Supertab TCS TATA CONSULTANCY SERVICES Typeface

Microsoft UKG Weights & Biases wipro workday WRITER ZEOTAP zoom

A2A's open governance



CLOUD

Google Cloud donates A2A to Linux Foundation

JUNE 23, 2025

Rao Surapaneni	Todd Segal	Michael Vakoc
VP and GM	Principal Engineer	Product Manager
Business Application Platform	Business Application Platform	Google Cloud

Share

A2A capabilities

🔍

Discovery

Agents must advertise their capabilities so clients know when and how to utilize them for specific tasks.

🤝

Negotiation

Clients and agents need to agree on communication methods like text, forms, iframe, or audio/video to ensure proper user interaction.

⚙️

Task and State Management

Clients and agents need mechanisms to communicate task status, changes, and dependencies throughout task execution.

🤗

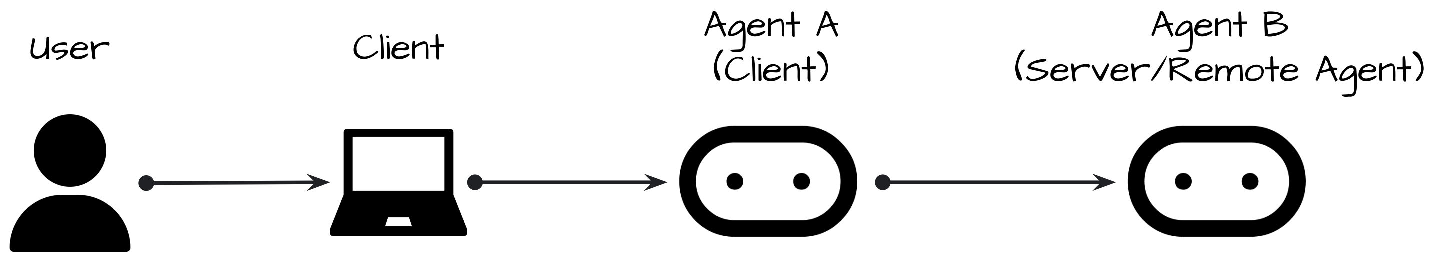
Collaboration

Clients and agents must support dynamic interaction, enabling agents to request clarifications, information, or sub-actions from client, other agents, or users.

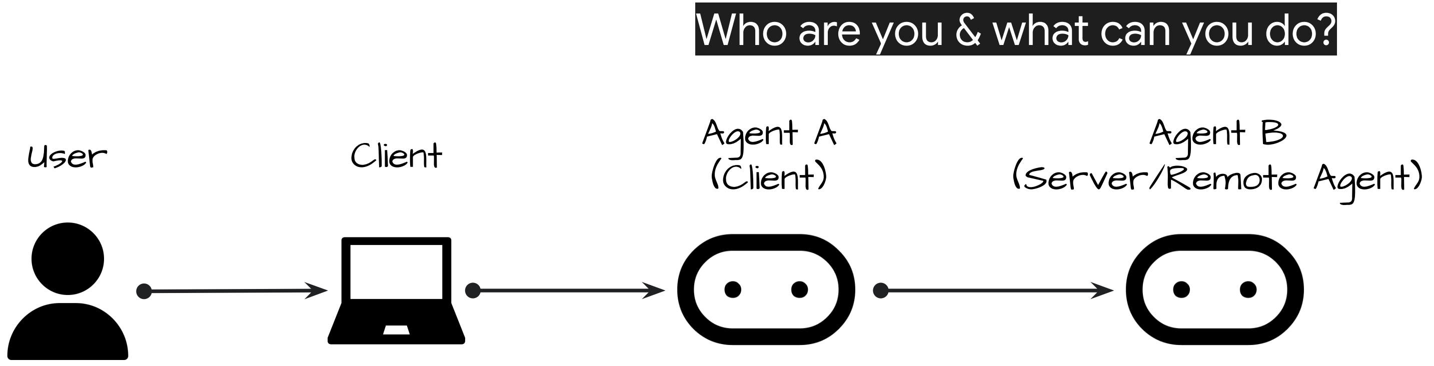
📚

How A2A works

Building a simple agent system

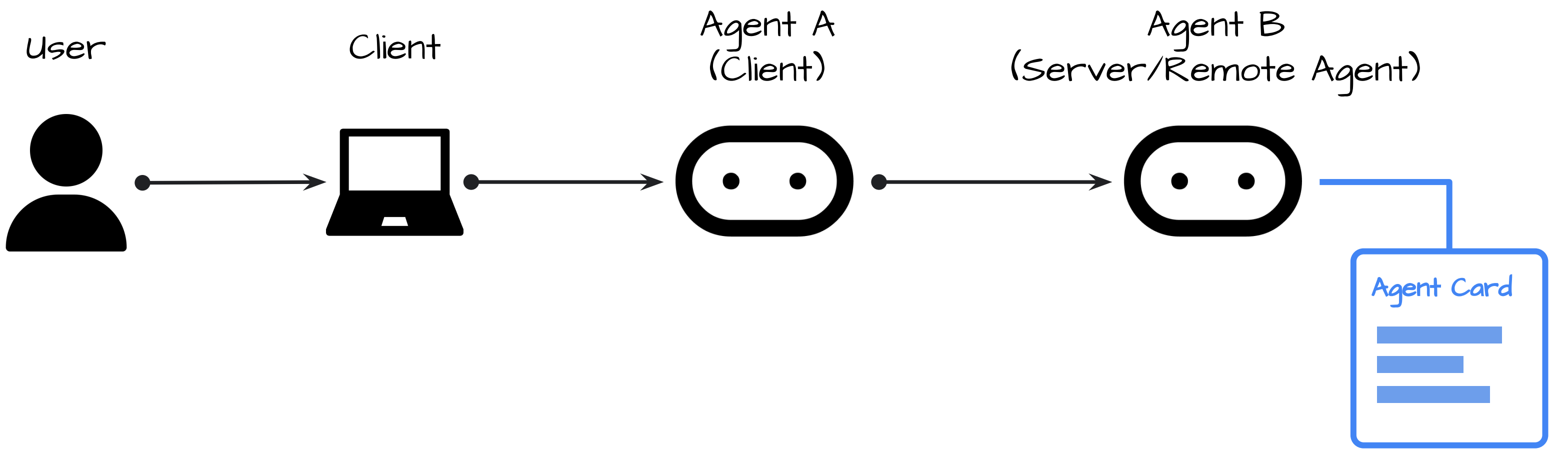


Step 1: Agent Discovery



Step 1: Agent Discovery

The agent card



`/.well-known/agent.json`

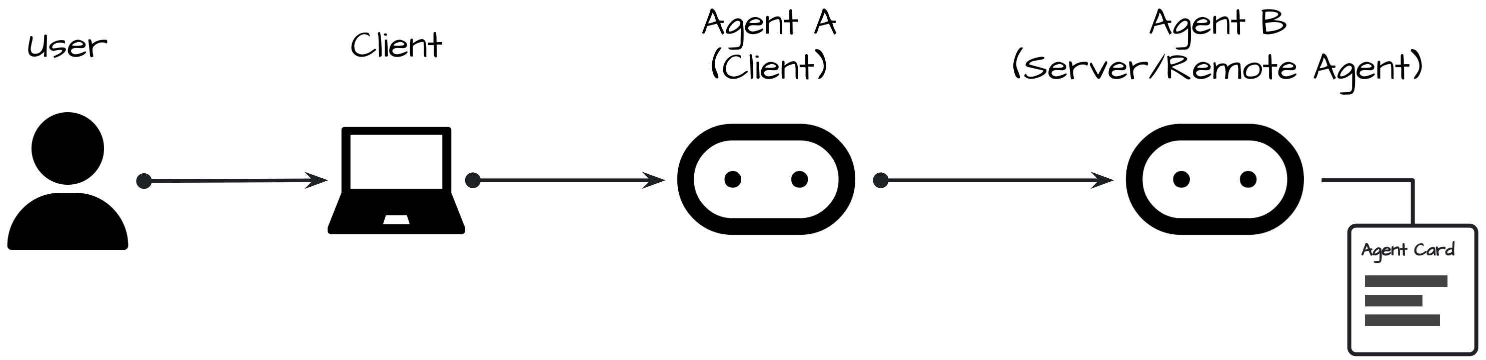
Agent Card

Example

```python
agent_card = AgentCard(
    name='Currency Agent',
    description='Helps with exchange rates for currencies',
    url=f'http://{host}:{port}/', # e.g., http://localhost:10000/
    version='1.0.0',
    defaultInputModes=CurrencyAgent.SUPPORTED_CONTENT_TYPES, # Usually ['text/plain']
    defaultOutputModes=CurrencyAgent.SUPPORTED_CONTENT_TYPES,
    skills=[skill],
)

# ... (Server setup using this agent_card) ...
```

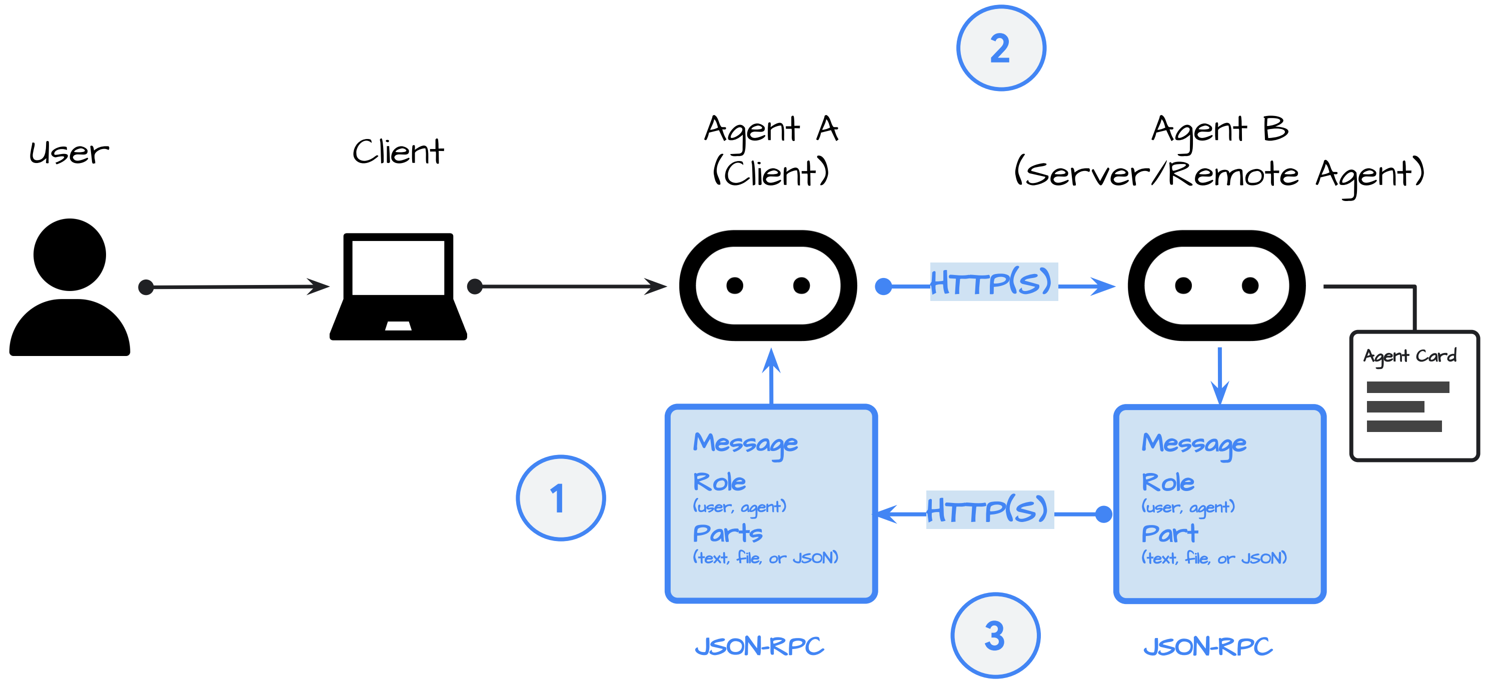
Step 2: Basic Interaction

How do we actually talk?



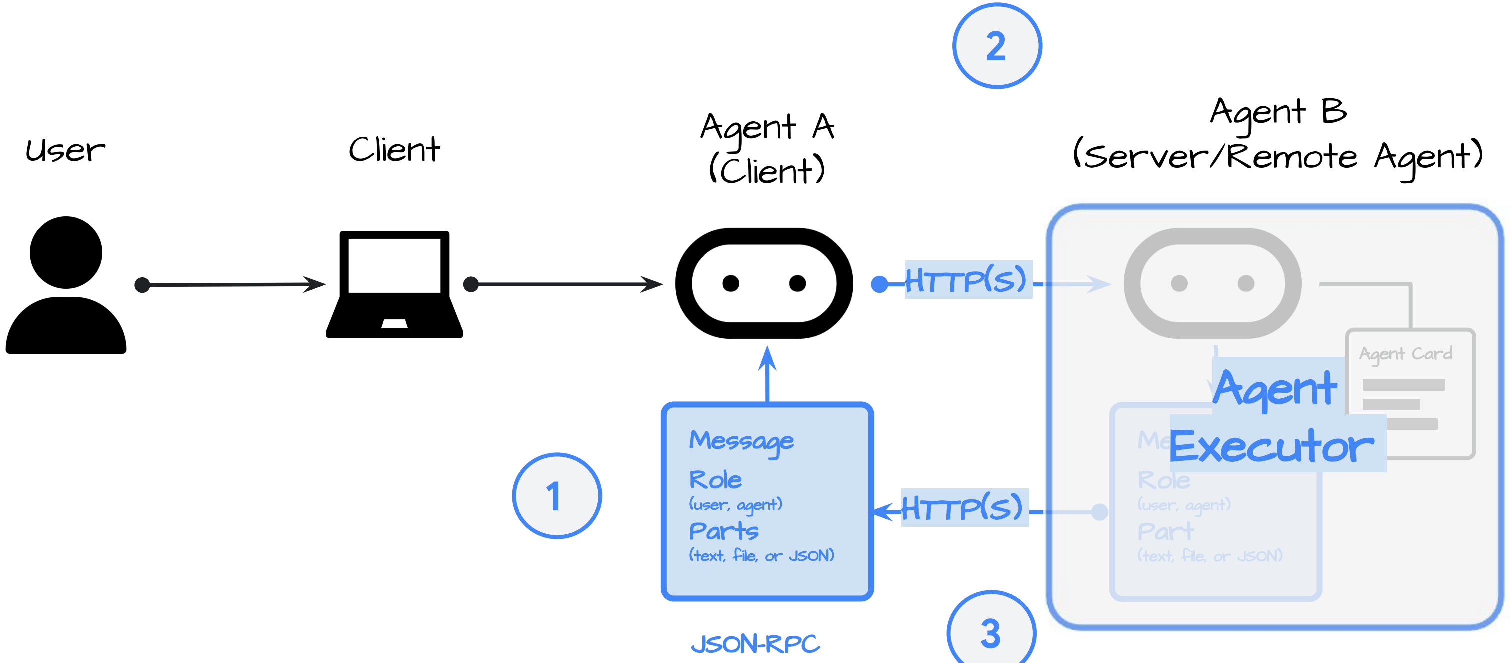
Step 2: Basic Interaction

Messages, Tasks & Parts



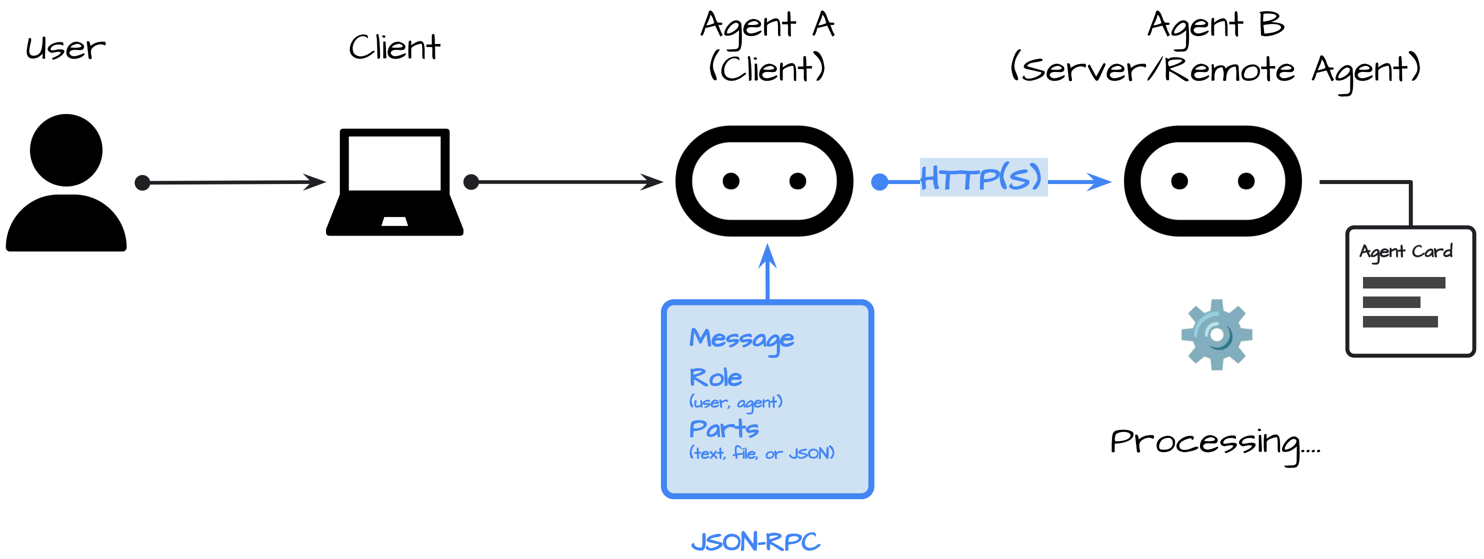
Step 2: Basic Interaction

The Agent Executor



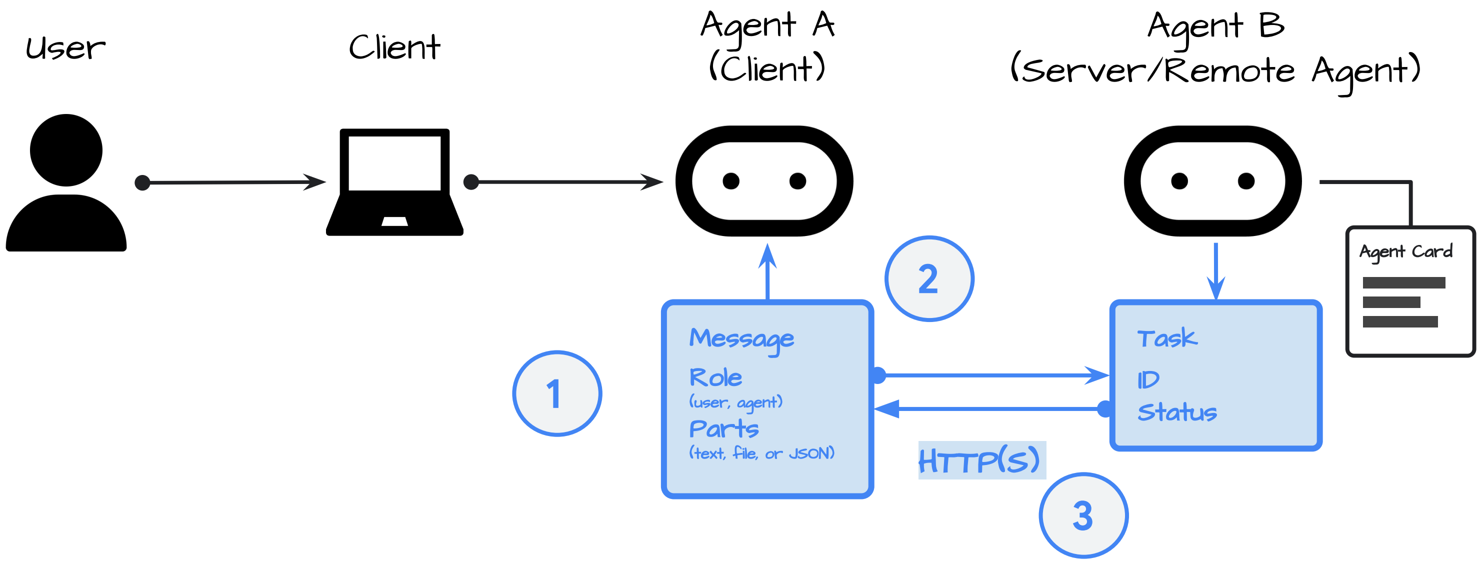
Step 3: Handling Real Work

Are we there yet?



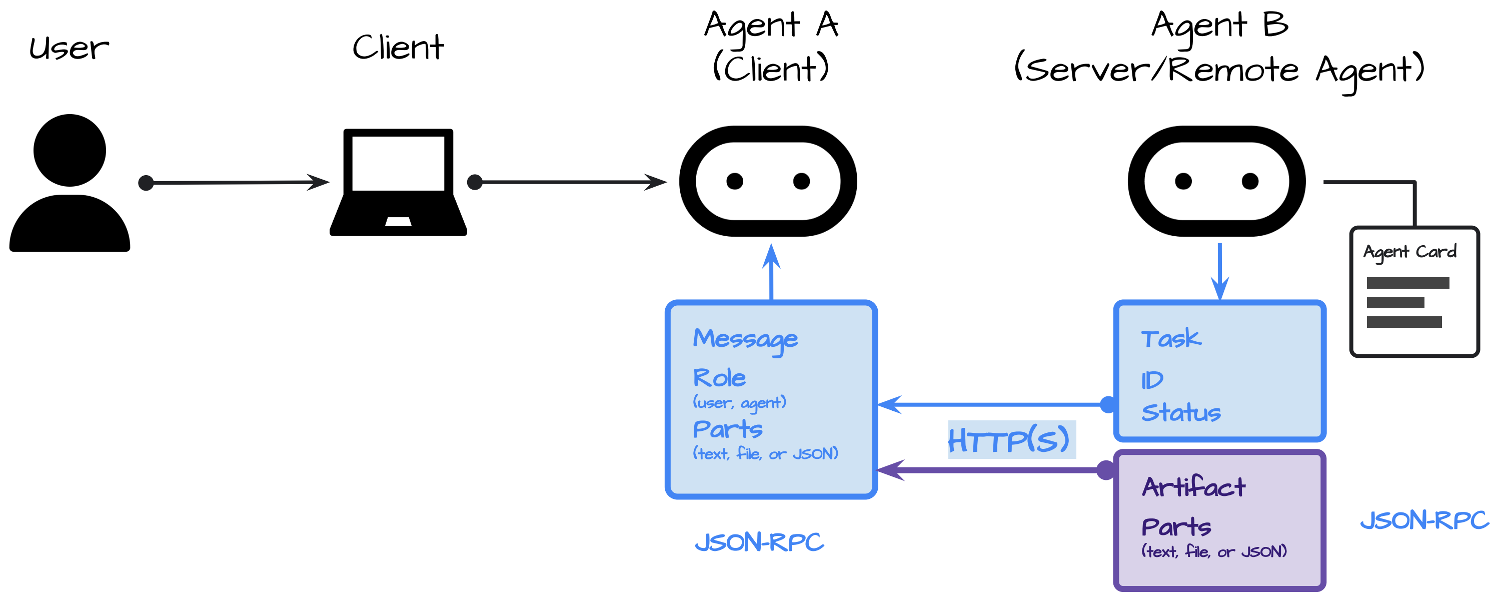
Step 3: Handling Real Work

Task Lifecycle & Polling



Step 3: Handling Real Work

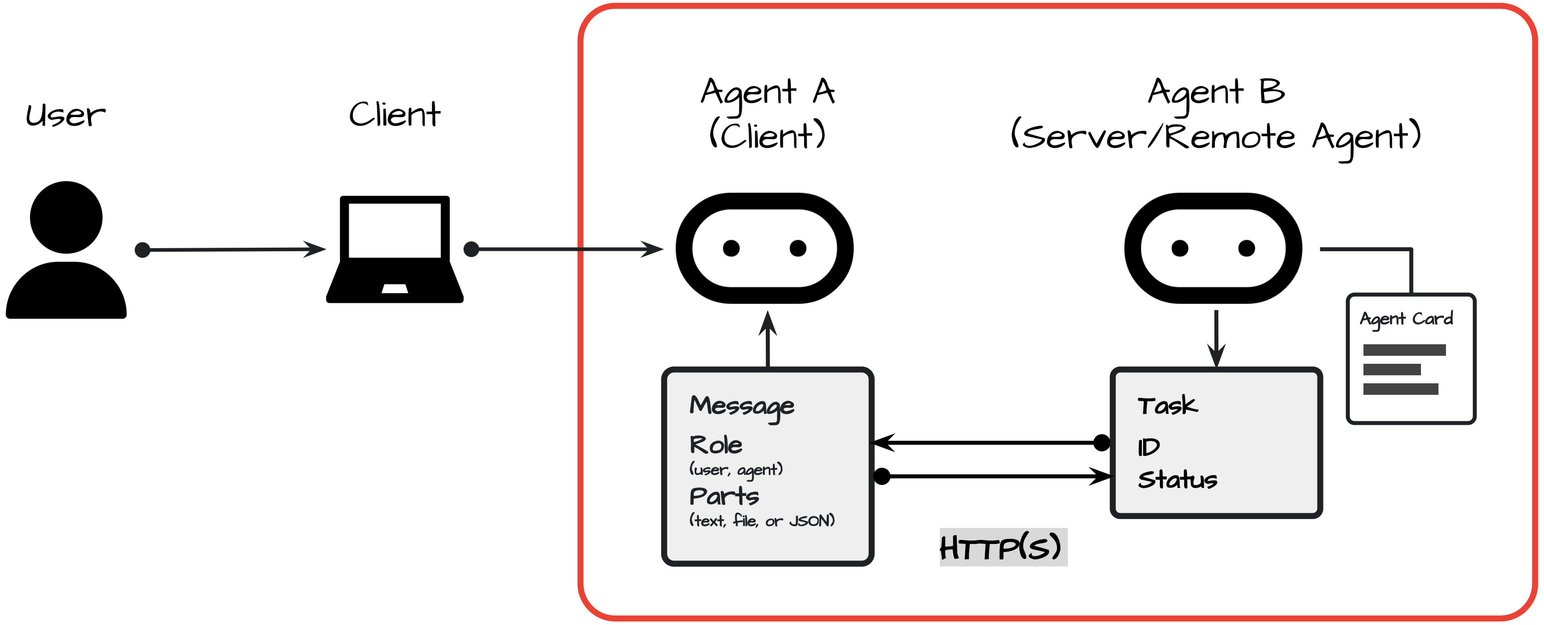
Task Lifecycle & Polling



Task Lifecycle & Polling

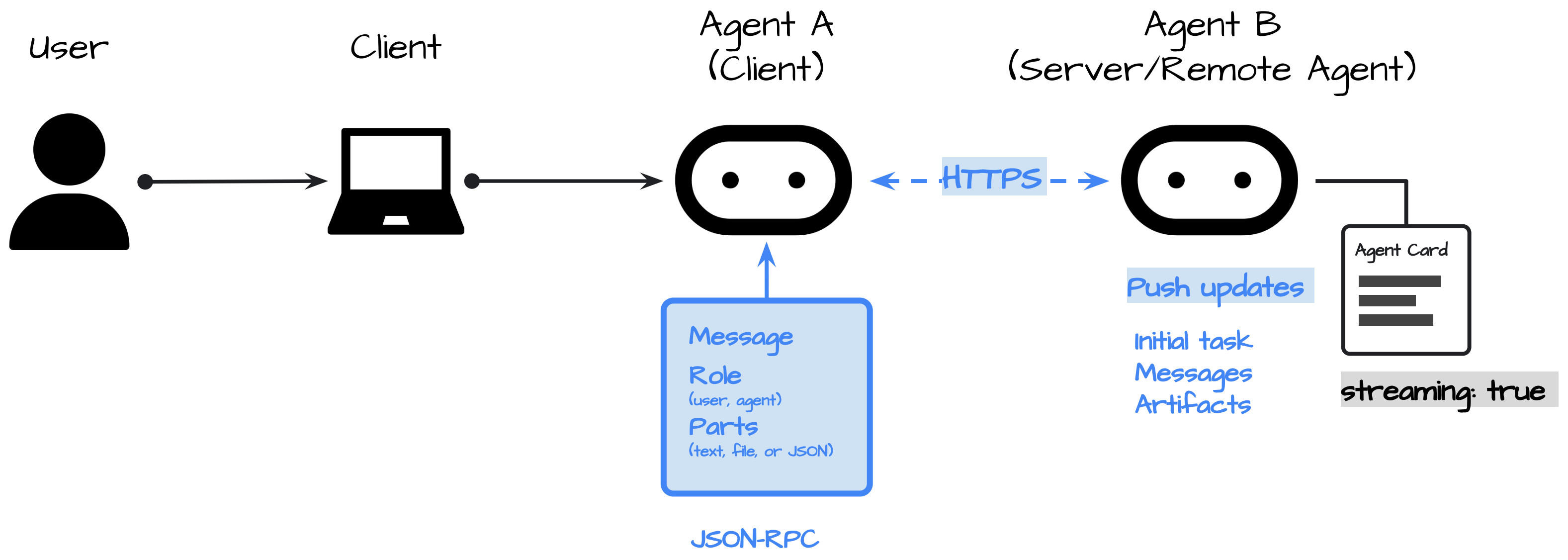
Challenge

❌ Polling is inefficient!



Step 4: Real-time Updates

Streaming with SSE



A2A vs MCP???

A2A ❤️ MCP
Complementary, Not Competing

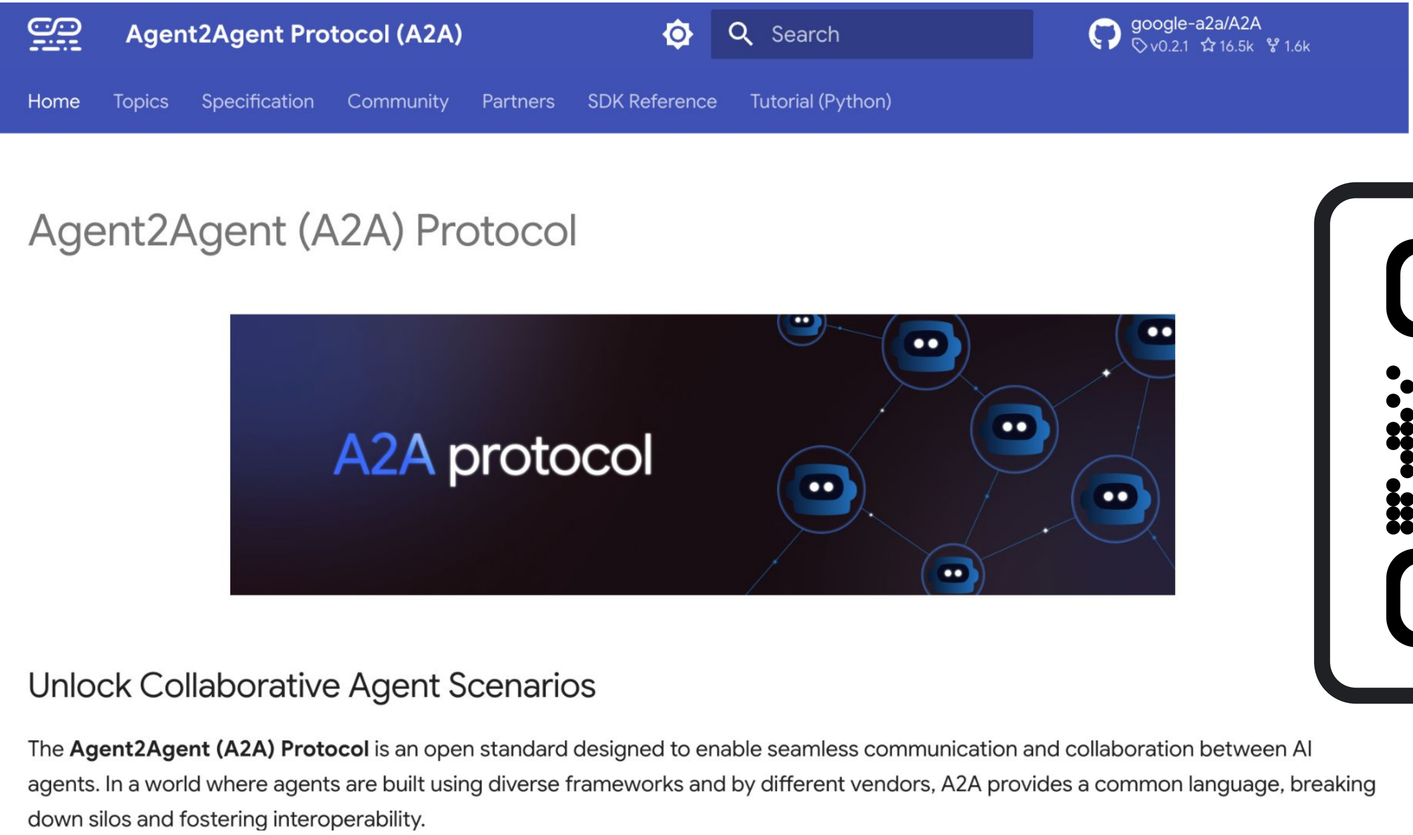
Model Context Protocol (MCP)

- Connects agents to **tools, APIs, and resources**.
- Think: *How an agent uses its capabilities (function calling)*.
- Example: Agent uses MCP to call a weather API tool.

Agent2Agent Protocol (A2A)

- Facilitates dynamic communication **between different agents** as peers.
- Think: *How agents collaborate, delegate, and manage shared tasks.*
- Example: A Travel Agent (A2A) asks a Flight Booking Agent (A2A) to find flights.

A2A Documentation


Agent2Agent Protocol (A2A)

Search

google-a2a/A2A
v0.2.1 16.5k 1.6k

Home Topics Specification Community Partners SDK Reference Tutorial (Python)

Agent2Agent (A2A) Protocol

Unlock Collaborative Agent Scenarios

The **Agent2Agent (A2A) Protocol** is an open standard designed to enable seamless communication and collaboration between AI agents. In a world where agents are built using diverse frameworks and by different vendors, A2A provides a common language, breaking down silos and fostering interoperability.

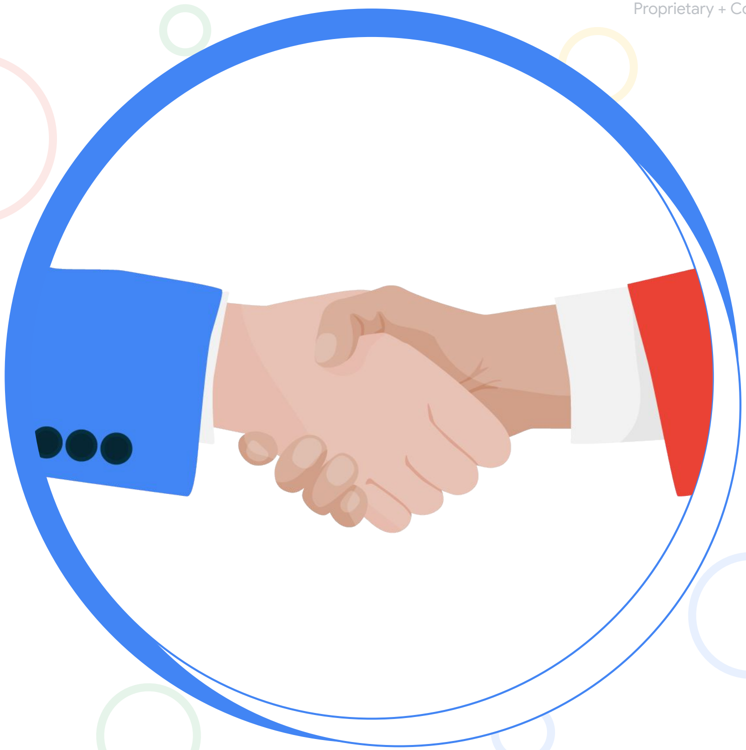
goo.gle/a2a

How about samples?

👉

goo.gle/a2a-samples

Thank you.



Dr. Wafae Bakkali

linkedin.com/in/wafae-bakkali/