IETF-124 I2ICF Side Meeting

Agent to Agent (A2A) Protocol

November 6, 2025

Montreal

Members: Jaehoon (Paul) Jeong, Joseph Ahn, Jiwon Suh, and Jiwon Yoo

Department of Computer Science and Engineering at SKKU

Email: {pauljeong, ahnjs124, sjw6136, sowra1}@skku.edu

Google’s Agent to Agent (A2A)

➢ Google A2A Protocol Announcement (April 9, 2025)

- https://developers.googleblog.com/ko/a2a-a-new-era-of-agent-interoperability/

➢ A2A Protocol

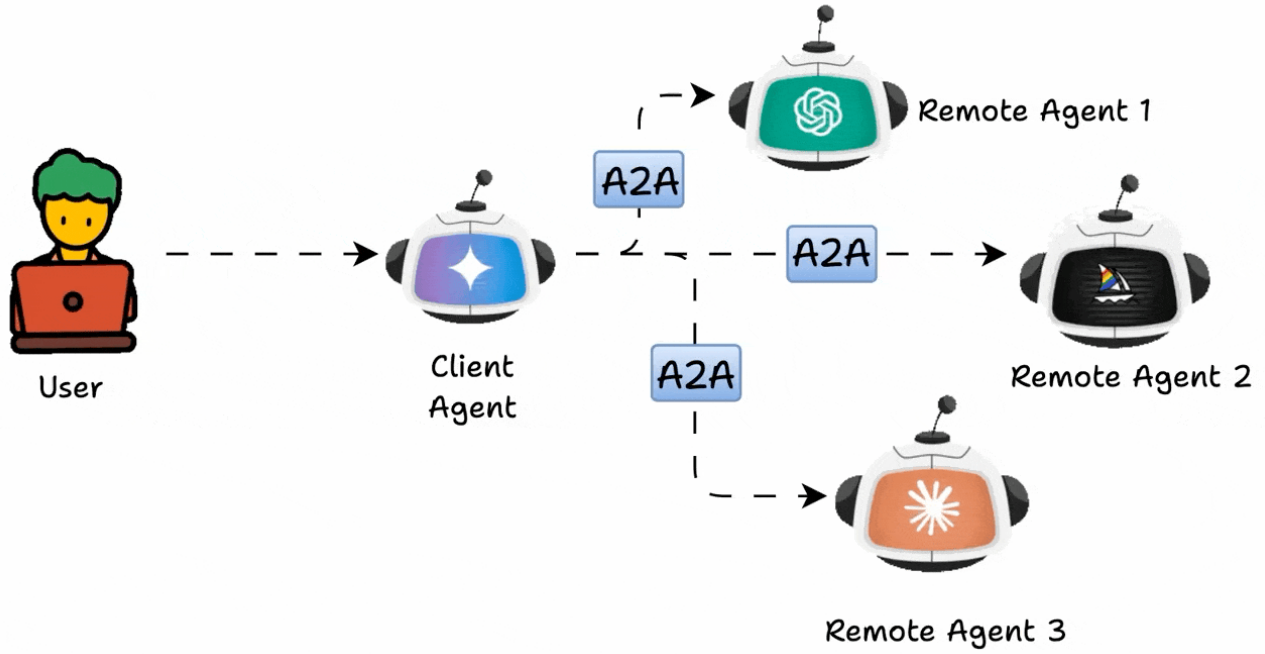
- **AI agents** communicate with each other across various enterprise platforms and applications, securely **exchanging information and coordinating tasks**.
- A2A is an open protocol that complements Anthropic's MCP, providing agents with useful tools and context.
  - MCP (Model Context Protocol)

<Source : Google>

Google’s A2A Framework

➢What is A2A protocol?

- The Agent-to-Agent (A2A) protocol is revolutionizing how autonomous agents discover and communicate with each other.
- Instead of complex integration setups, A2A provides a standardized way for agents to find each other and exchange information seamlessly.



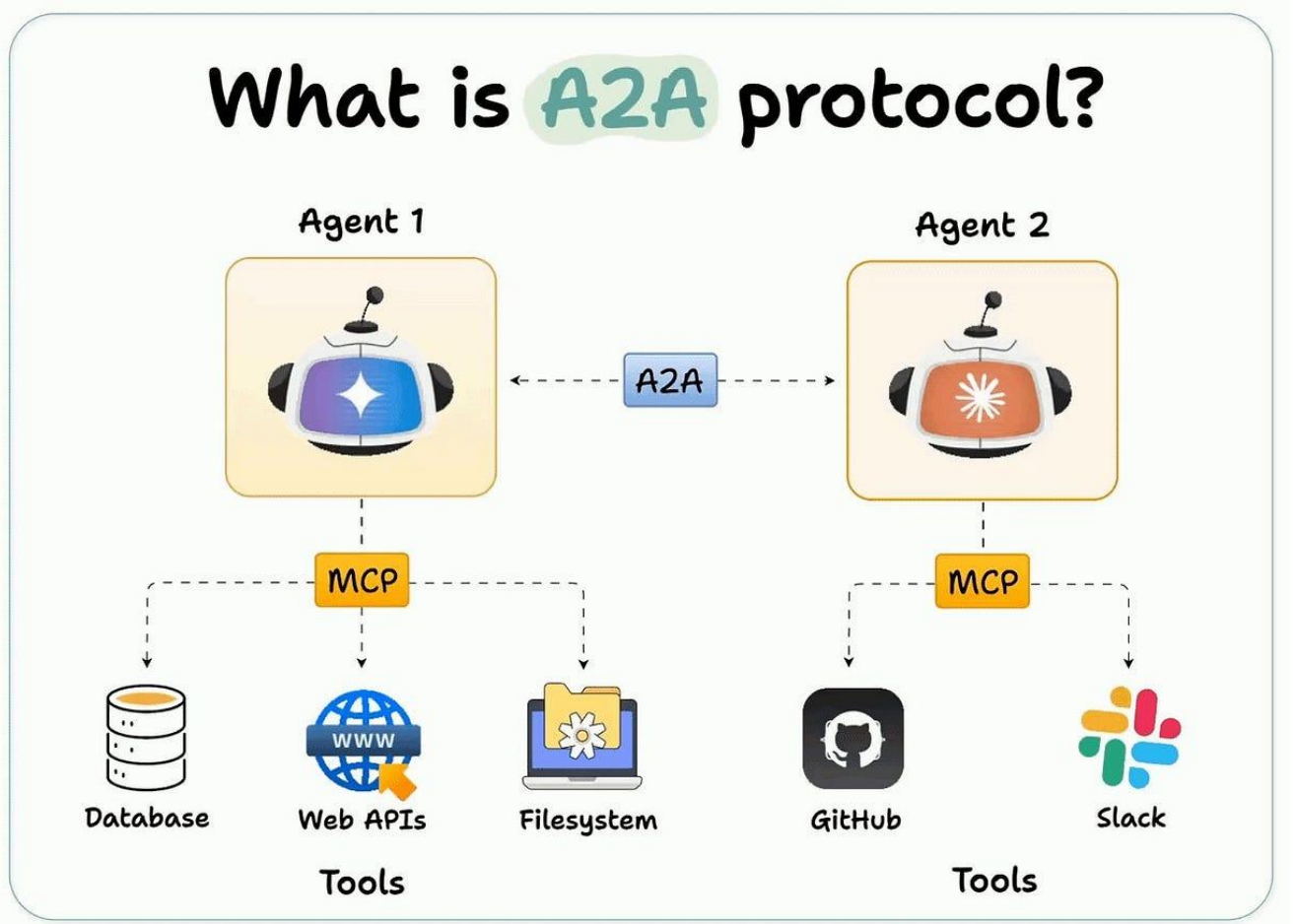
<Source : https://ai.plainenglish.io/>

Google’s A2A Framework

➢What is A2A protocol?

- The Agent2Agent (A2A) Protocol is an open standard that enables seamless communication, coordination, and task-sharing between AI agents — regardless of which teams built them, which technologies they use, or which organizations own them.

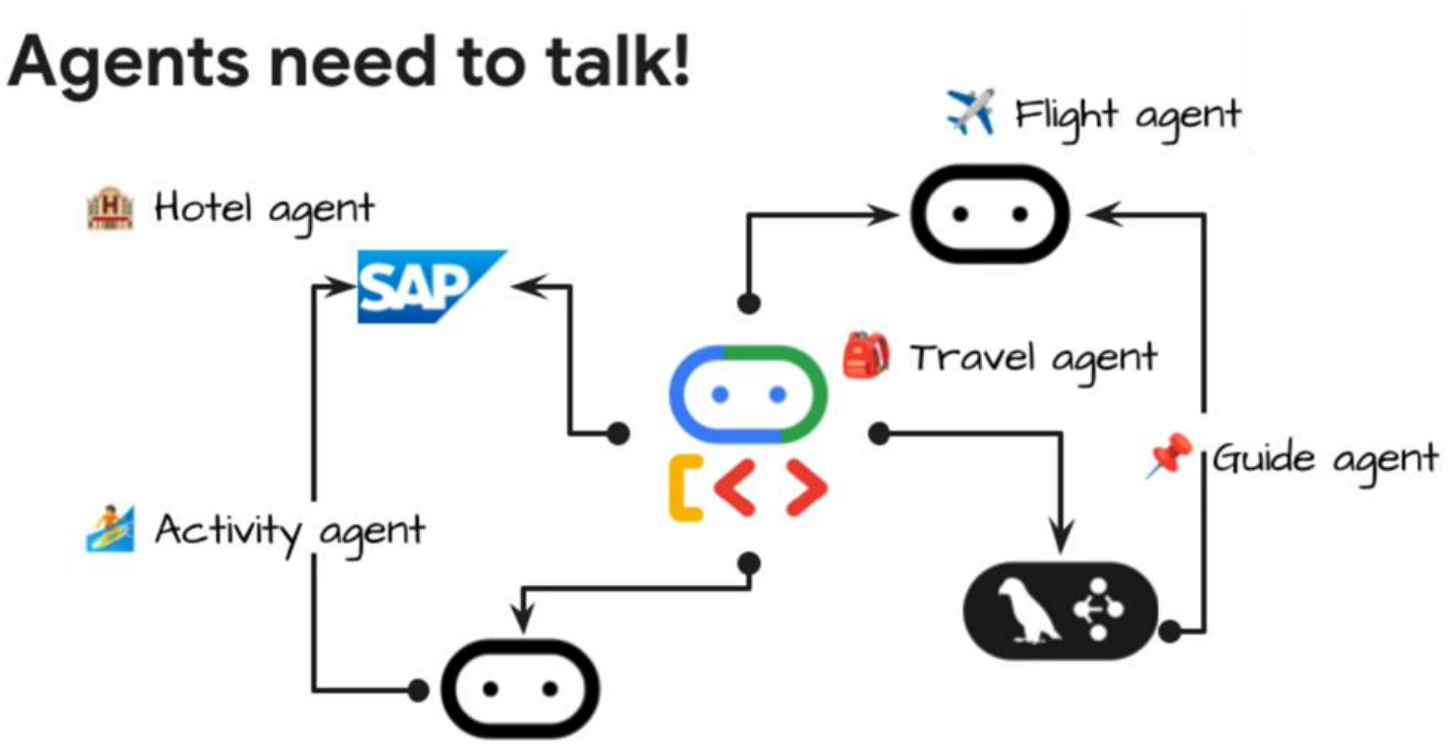
- It’s designed to help create intelligent, decentralized ecosystems where agents can collaborate, no matter who made them or where they live.



<Source : https://ai.plainenglish.io/>

Google’s A2A Framework

➢Example of Agents Collaboration



<Source : Google>

Google’s A2A Framework

➢Partners contributing to the A2A protocol

<Source : Google>

Google’s A2A Framework

➢A2A Capabilities

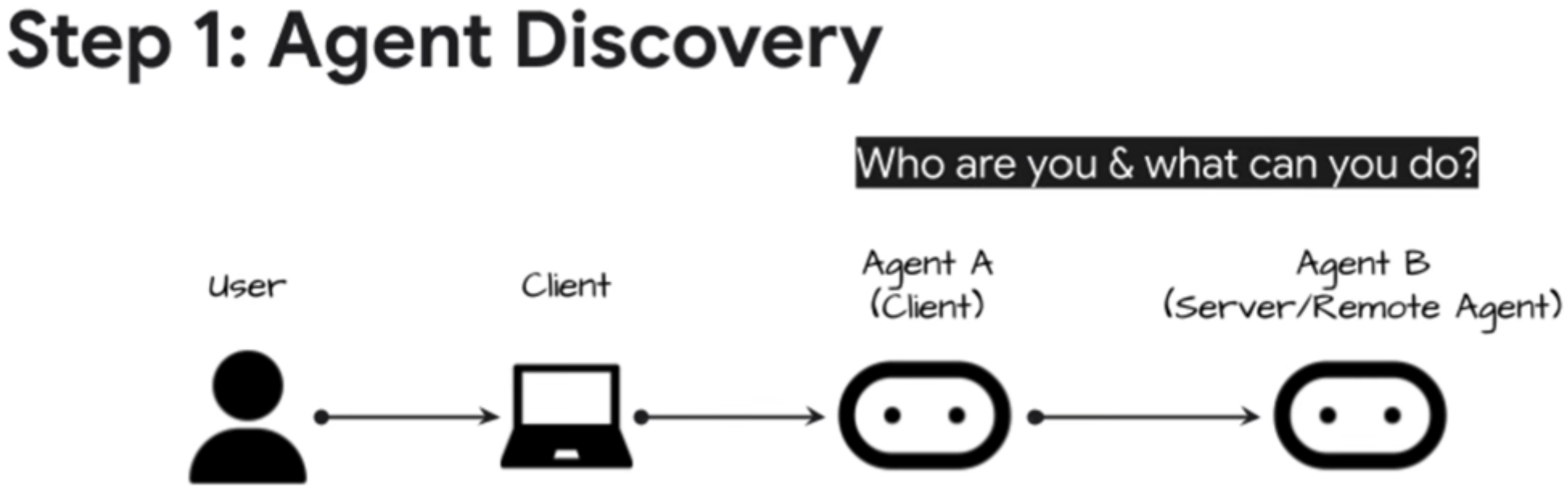
- **A2A Discovery → Protocol Negotiation → Task and State Management → Collaboration to satisfy user requirements**

<Source : Google>

Google’s A2A Framework

➢ Agent Discovery

- For inter-agent communication to occur, the server or remote agent must be identified and known.

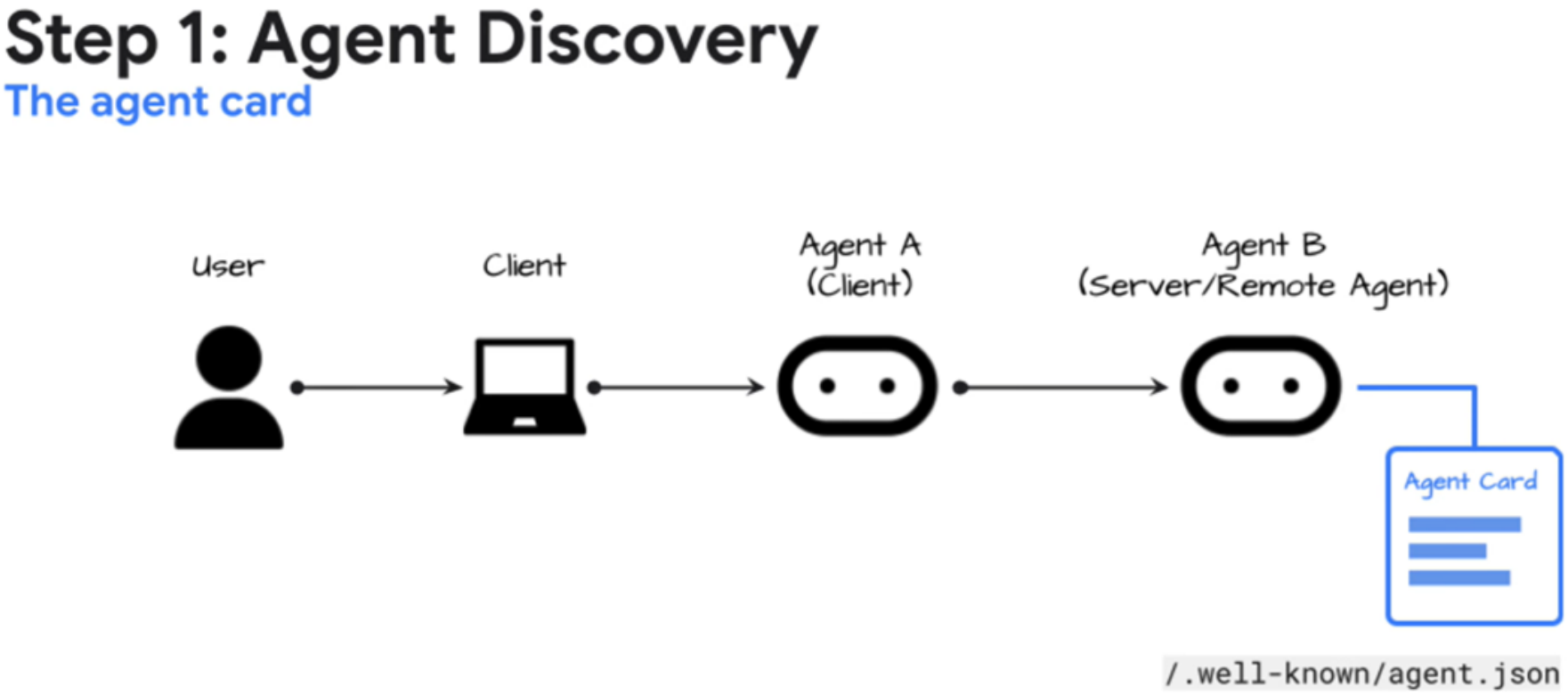


<Source : Google>

Google’s A2A Framework

➢Agent B Share their information through Agent Card to Agent A

- Provides information such as the protocol to use for communication, the information to provide, and the authentication method.
  - Name, HTTP endpoint URL, and Agent Specific Skill

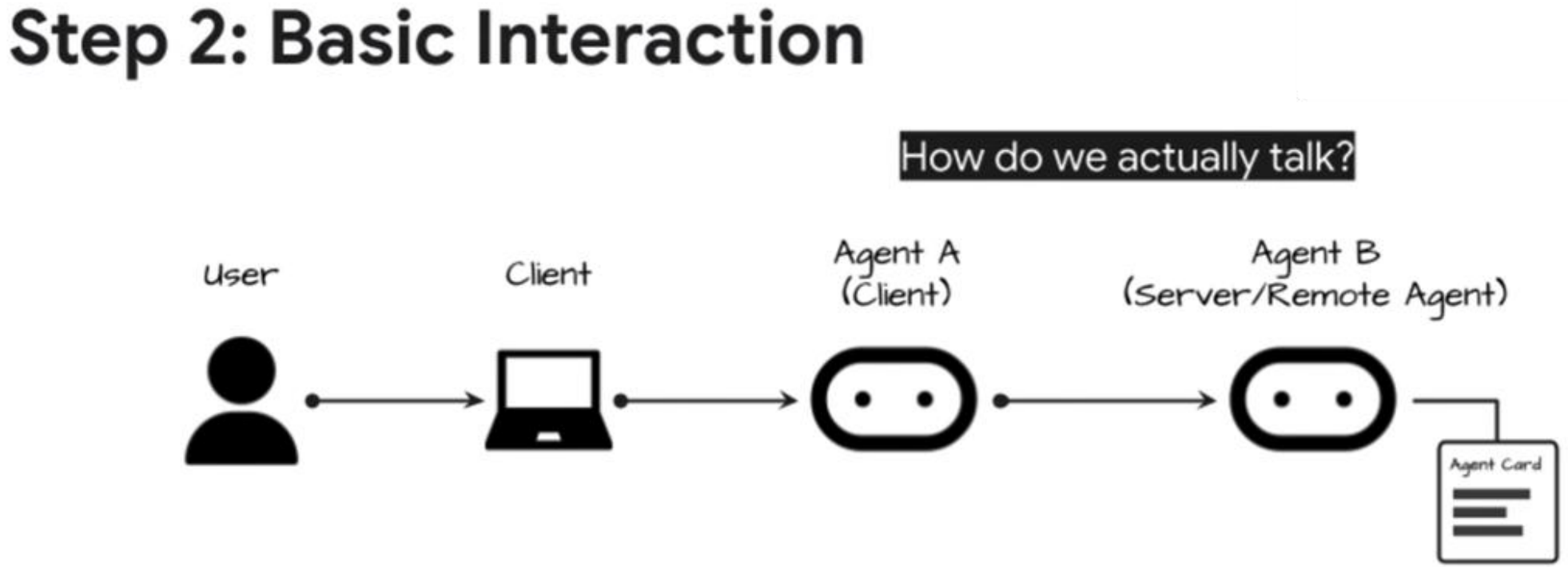


<Source : Google>

Google’s A2A Framework

➢Basic Interaction

- Negotiate information sharing based on exchanged information

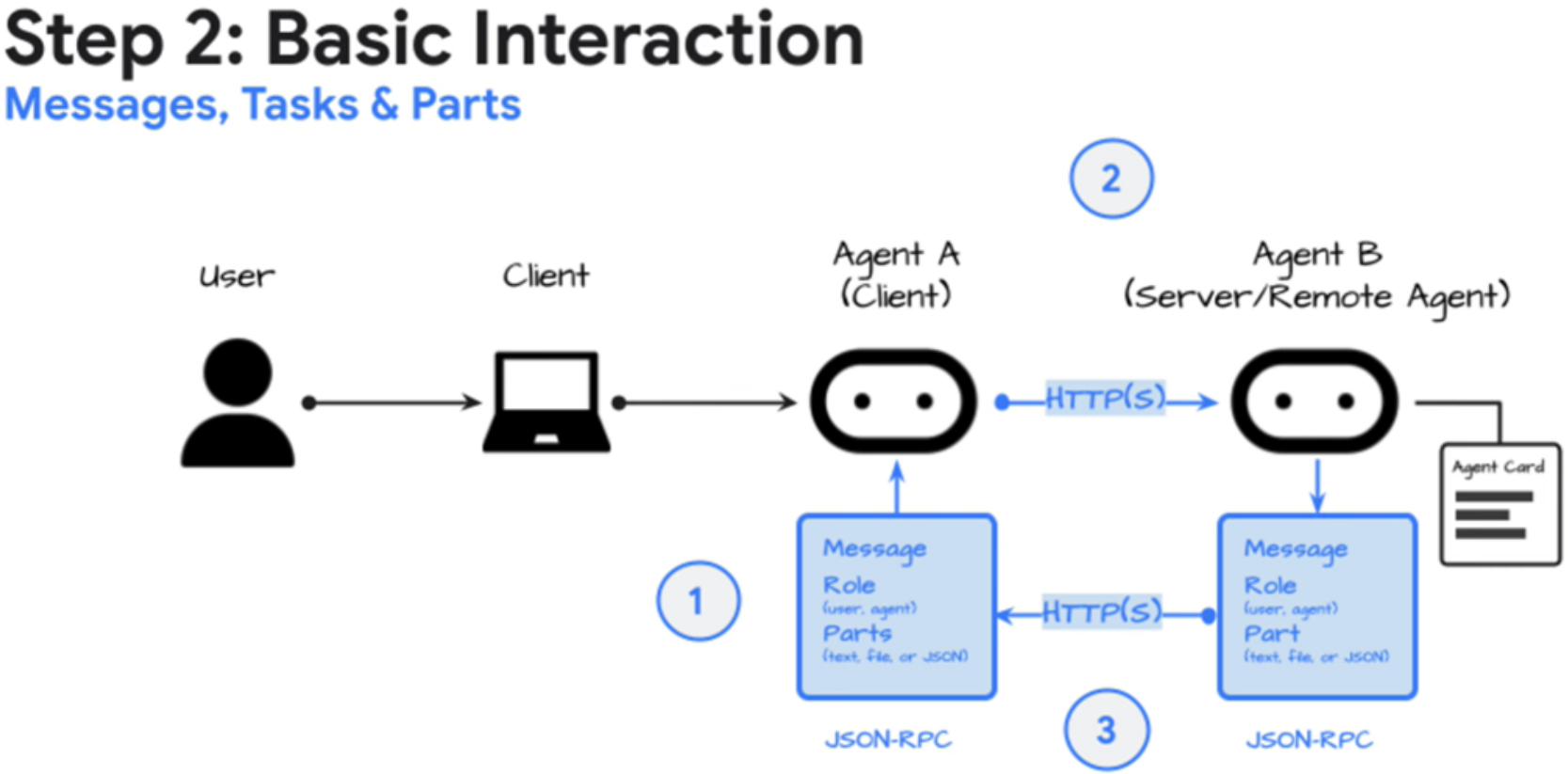


<Source : Google>

Google's A2A Framework

➢ Agent A requests the necessary infromation from Agent B

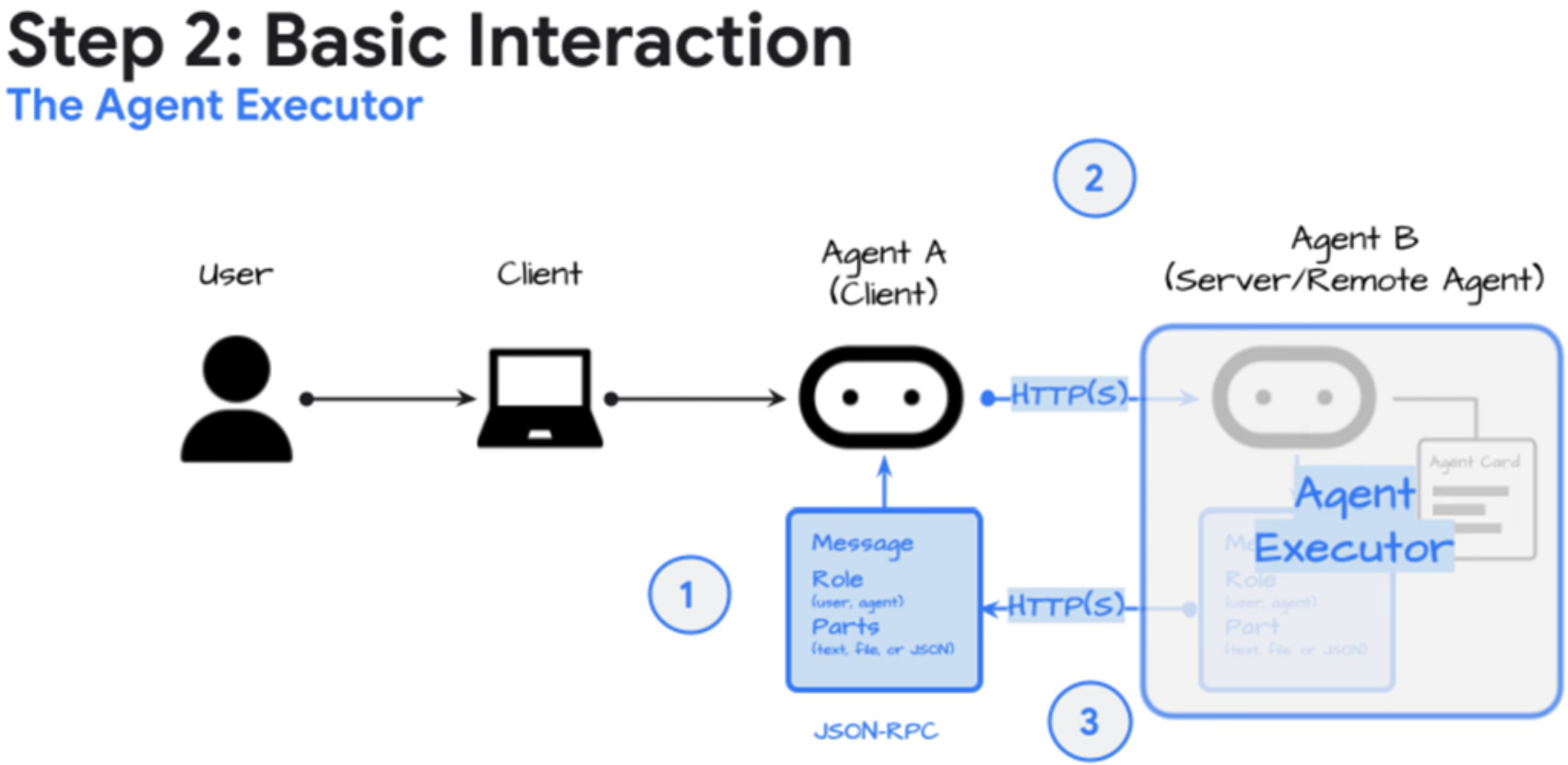
- Questions the desired request in JSON-RPC format over HTTP(S)
- Agent B responds to the request in JSON-RPC format



<Source : Google>

Google’s A2A Framework

➢ **Agent B performs the request from Agent A**

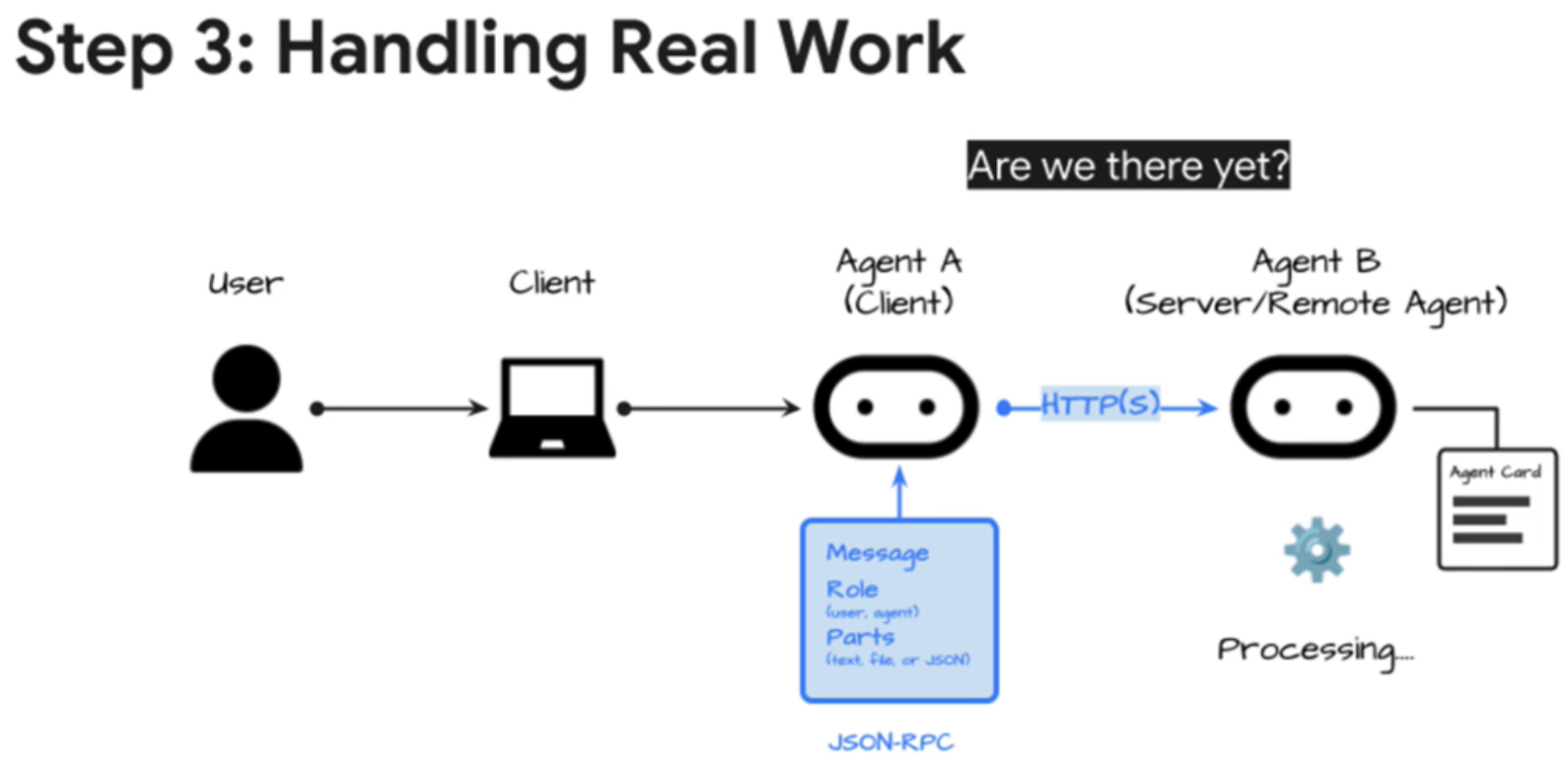


<Source : Google>

Google’s A2A Framework

➢Agent A waits for Agent B's response

- What if Agent B takes a long time to process the answer to a question?

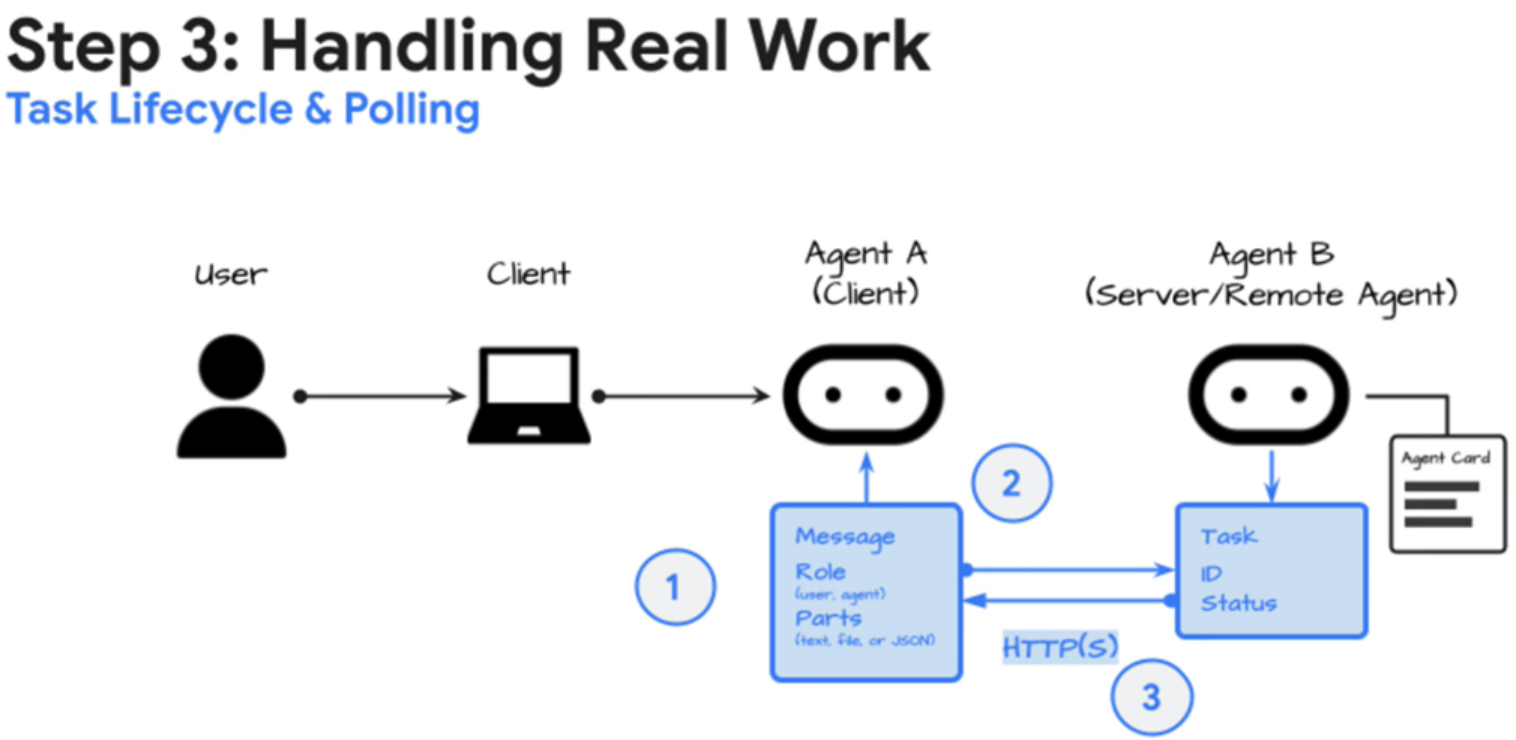


<Source : Google>

Google's A2A Framework

➢ Manage requests/responses by Task ID (1, 2, 3, etc.)

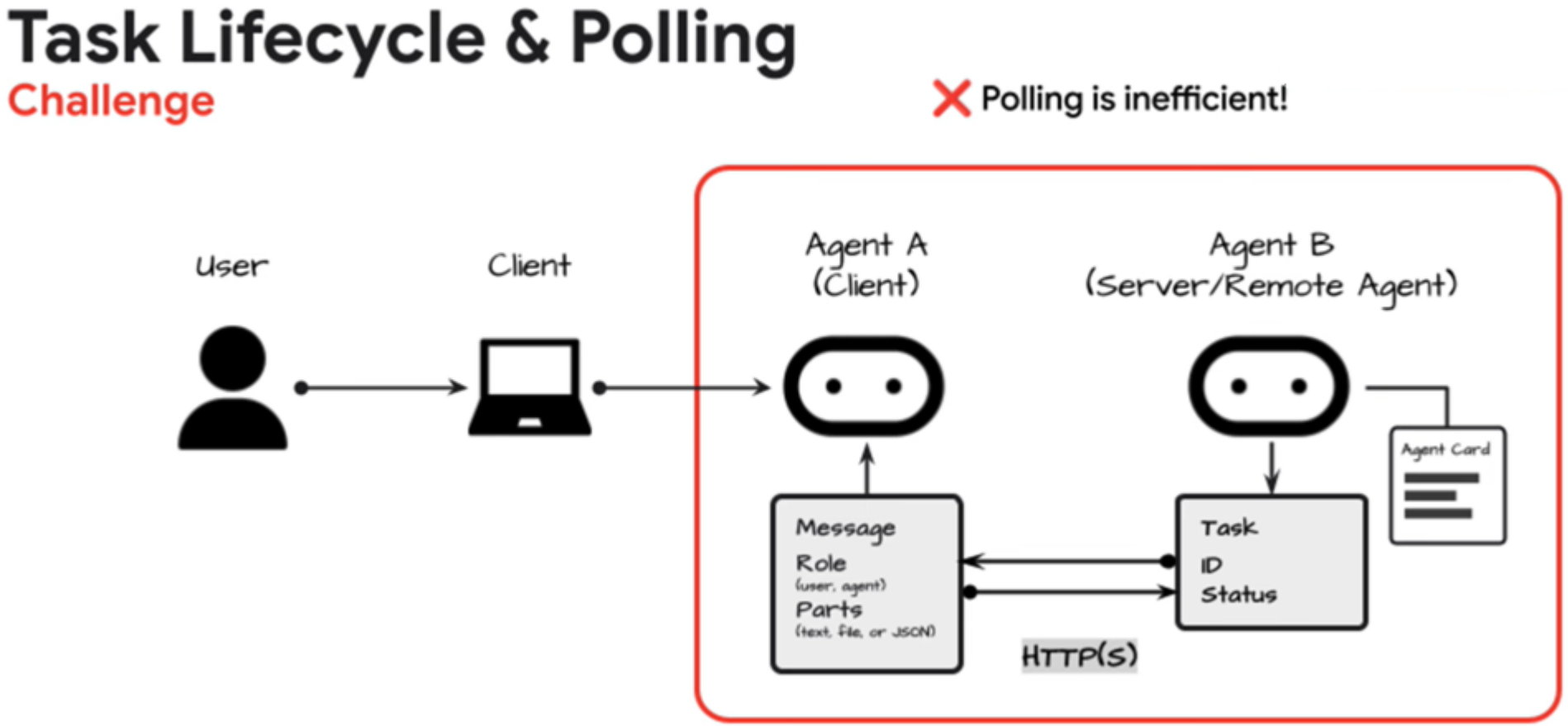
- Agent B → More information needed for Task ID 1 → Agent A responds → Agent B executes



<Source : Google>

Google’s A2A Framework

➢ **Agent A should not poll Agent B**

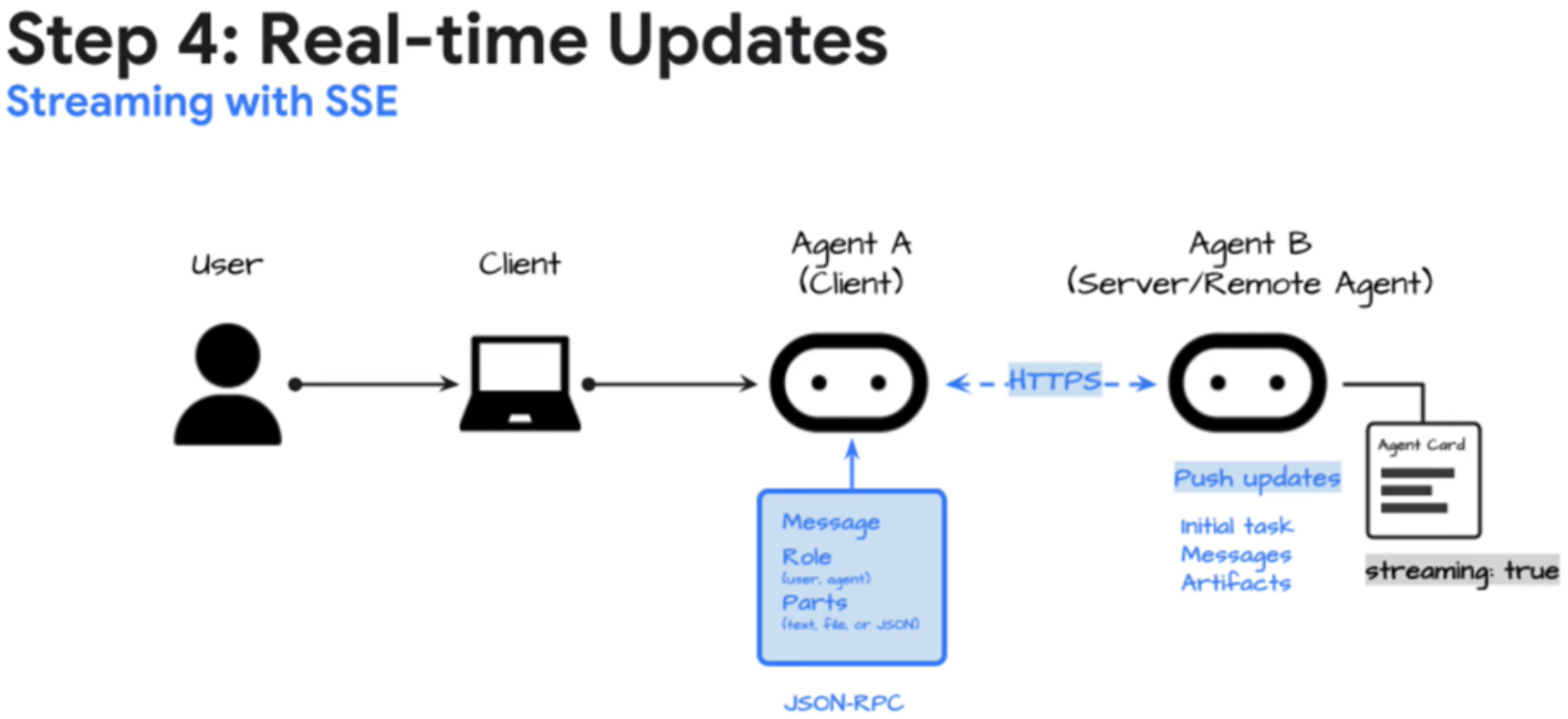


<Source : Google>

Google’s A2A Framework

➢Real-time Updates

- When a task is completed, Agent B → Agent A updates the response while connected to HTTP
- It's convenient to use because it provides a Live Progress update when the response is ready.

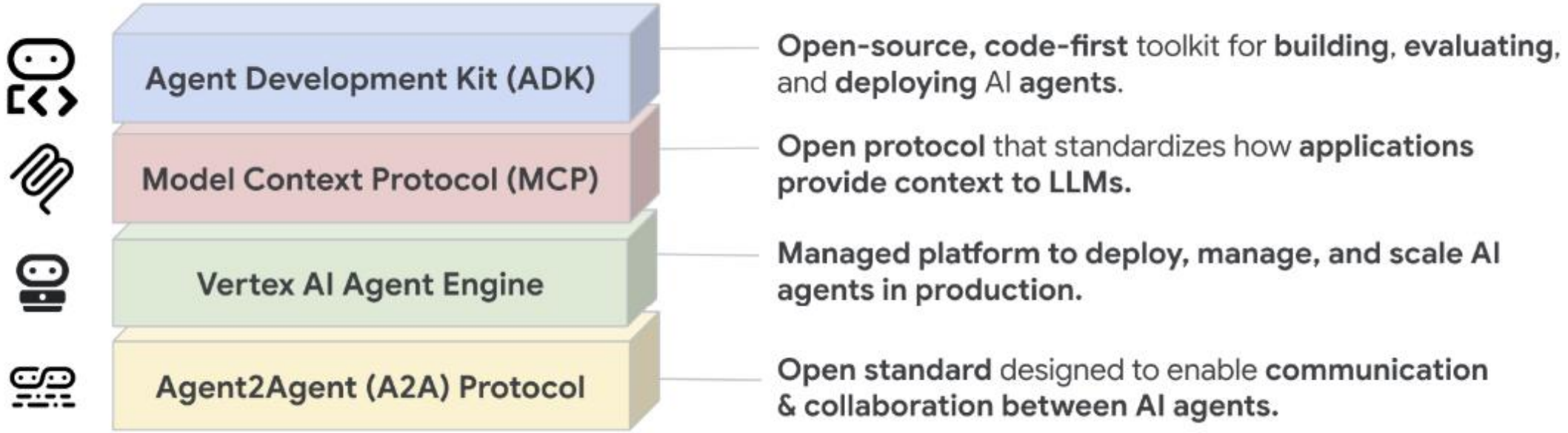


<Source : Google>

A2A agent stack

➢ Agent Stack

- A2A is situated within a broader agent stack



<Source : Google>

A2A and MCP

➢Relationship between A2A and MCP (1/2)

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.

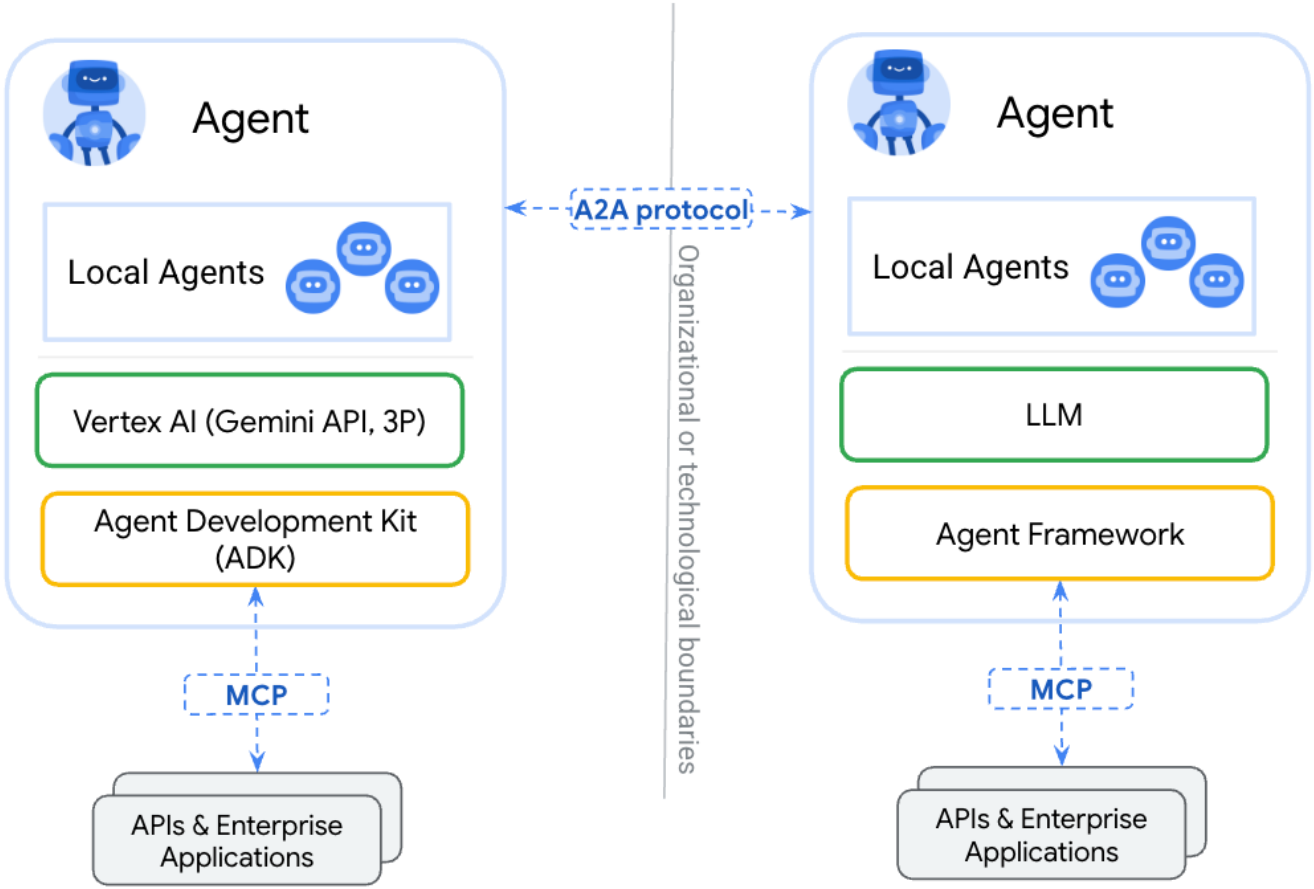
Agent2Agent Protocol (A2A)

- 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.

<Source : Google>

A2A and MCP

➢Relationship between A2A and MCP (2/2)

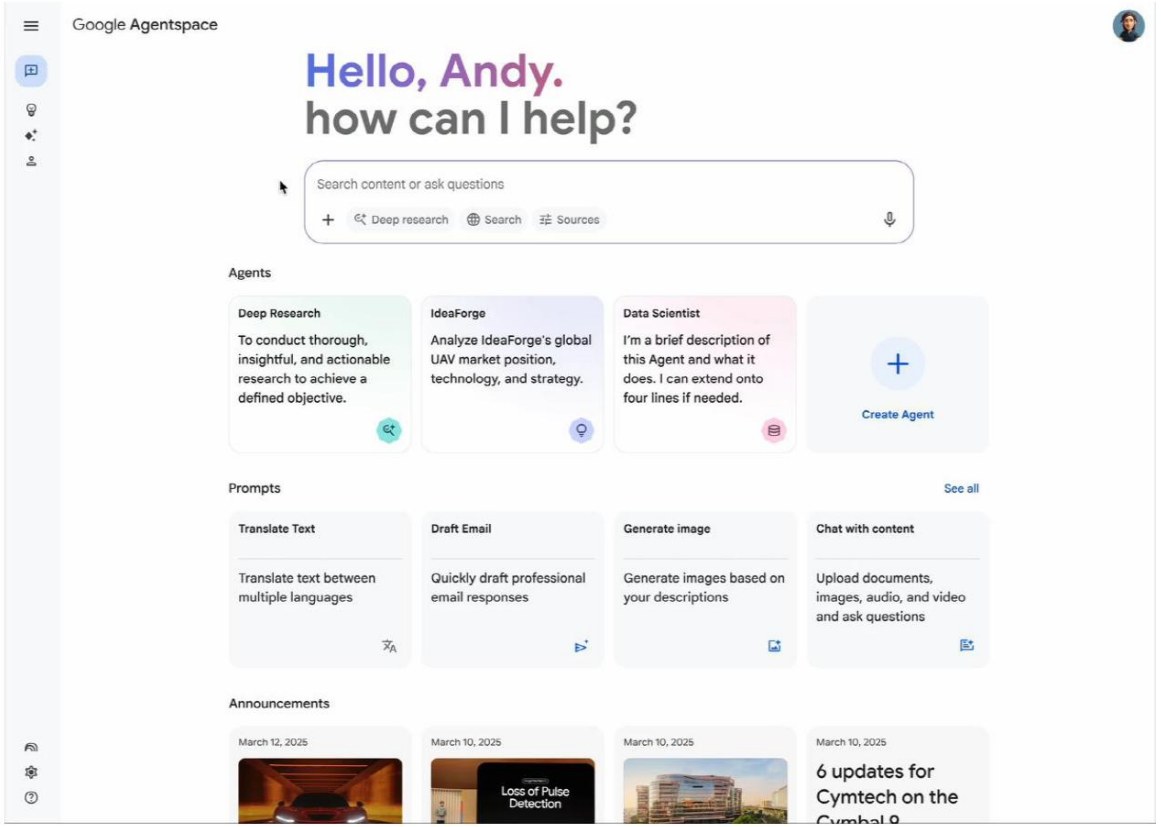


<Source : Google>

A2A candidate sourcing

➢ A real-world example: candidate sourcing

- Hiring engineers becomes easier with A2A collaboration
- The agent works with other agents to find candidates
- The user's agent schedules interviews to simplify hiring



<Source : Google>

APPENDIX

Core Concepts and Components in A2A

➢Fundamental Communication Elements

- The following table describes the fundamental communication elements in A2A:

Element	Description	Key Purpose
Agent Card	A JSON metadata document describing an agent's identity, capabilities, endpoint, skills, and authentication requirements.	Enables clients to discover agents and understand how to interact with them securely and effectively.
Task	A stateful unit of work initiated by an agent, with a unique ID and defined lifecycle.	Facilitates tracking of long-running operations and enables multi-turn interactions and collaboration.
Message	A single turn of communication between a client and an agent, containing content and a role ("user" or "agent").	Conveys instructions, context, questions, answers, or status updates that are not necessarily formal artifacts.
Part	The fundamental content container (for example, TextPart, FilePart, DataPart) used within Messages and Artifacts.	Provides flexibility for agents to exchange various content types within messages and artifacts.
Artifact	A tangible output generated by an agent during a task (for example, a document, image, or structured data).	Delivers the concrete results of an agent's work, ensuring structured and retrievable outputs.

<Source : Google>

Google’s A2A Framework

➢A2A Design Principles

- A2A is an open protocol that provides a standard way for agents to collaborate with each other, regardless of the underlying framework or vendor.

  - **Embrace agentic capabilities**: A2A focuses on enabling agents to collaborate in their natural, unstructured modalities, even when they don’t share memory, tools and context.
  - **Build on existing standards:** The protocol is built on top of existing, popular standards including HTTP, SSE, JSON-RPC, which means it’s easier to integrate with existing IT stacks businesses already use daily.
  - **Secure by default**: A2A is designed to support enterprise-grade authentication and authorization, with parity to OpenAPI’s authentication schemes at launch.
  - **Support for long-running tasks:** We designed A2A to be flexible and support scenarios where it excels at completing everything from quick tasks to deep research that may take hours and or even days when humans are in the loop.
  - **Modality agnostic:** The agentic world isn’t limited to just text, which is why we’ve designed A2A to support various modalities, including audio and video streaming.

<Source : Google>

How A2A works

A2A enables communication between 'Client' Agents and 'Remote' Agents

Remote Agent is responsible for acting on those tasks in an attempt to provide the correct information or take the correct action.

Client Agent is responsible for formulating and communicating tasks.

<Source : Google>

How A2A works

A2A enables communication between 'Client' Agents and 'Remote' Agents

① Capability Discovery:

- Agents can advertise their capabilities using an "Agent Card" in JSON format.
- It allows the client agent to identify the best agent that can perform a task and leverage A2A to communicate with the remote agent.

② Task Management:

- The communication between a client and remote agent is oriented towards task completion, in which agents work to fulfill end-user requests.
- This "task" object is defined by the protocol and has a lifecycle. It can be completed immediately or, for long-running tasks, each of the agents can communicate to stay in sync with each other on the latest status of completing a task.

<Source : Google>

How A2A works

A2A enables communication between 'Client' Agents and 'Remote' Agents

③ Collaboration:

- Agents can send each other messages to communicate context, replies, artifacts, or user instructions.

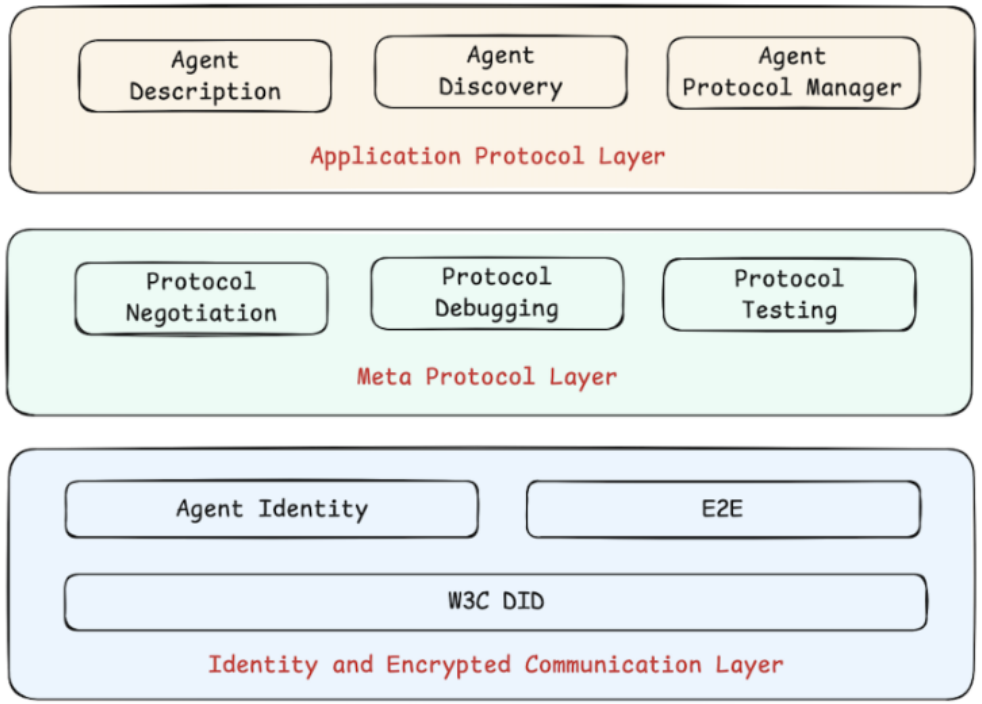
④ User Experience Negotiation:

- Each message includes "parts," which is a fully formed piece of content, like a generated image.
- Each part has a specified content type, allowing client and remote agents to negotiate the correct format needed and explicitly include negotiations of the user's UI capabilities–e.g., iframes, video, web forms, and more.

<Source : Google>

Agent Network Protocol (ANP)

- https://github.com/agent-network-protocol/AgentNetworkProtocol



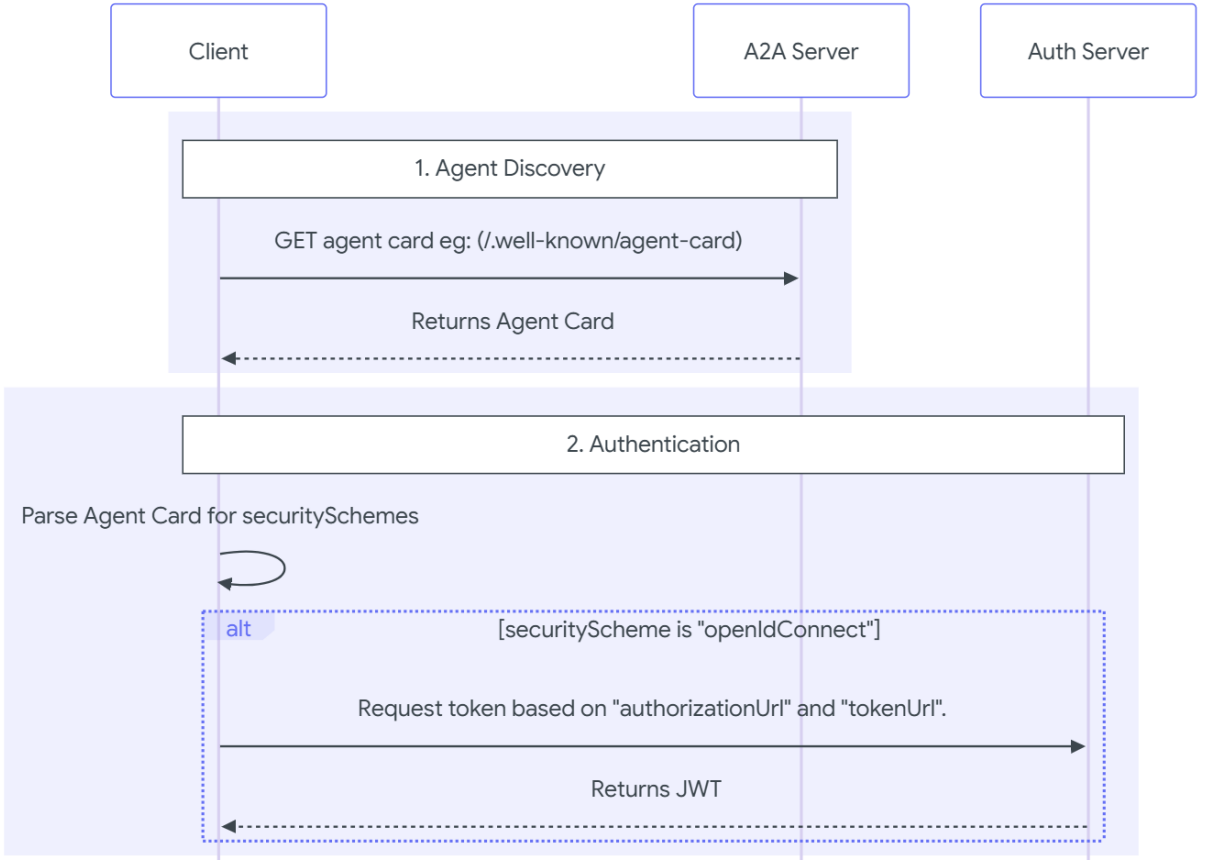
📡 **Application Layer:** Uses semantic web standards for describing capabilities and managing protocols efficiently.
🌍 **Meta-Protocol Layer:** Enables agents to self-negotiate and collaborate through protocol negotiation.
🔒 **Identity Layer:** Uses W3C DID for decentralized authentication and end-to-end encryption without central authority.

<Source : ANP Github>

A2A request Lifecycle

➢A2A request Lifecycle

- The A2A request lifecycle is a sequence that details the four main steps a request follows: agent discovery, authentication, sendMessage API, and sendMessageStream API

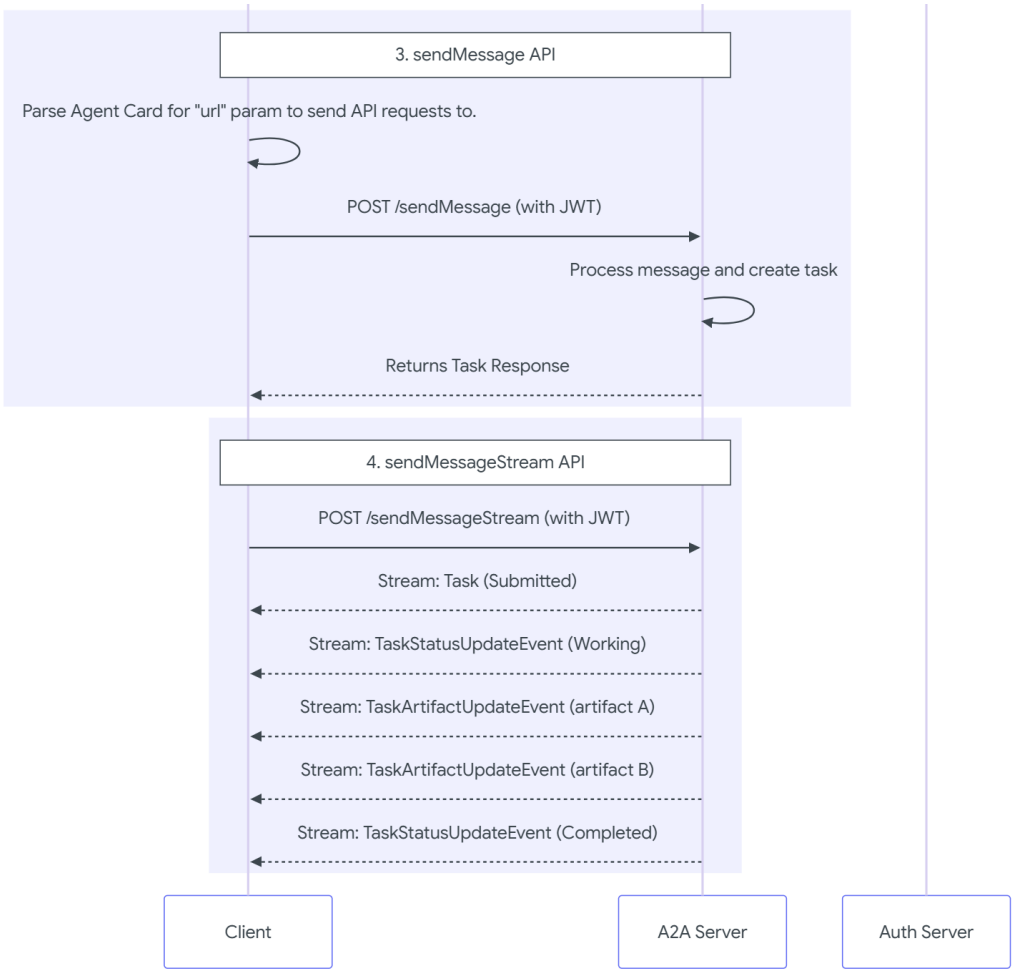


<Source : Google>

A2A request Lifecycle

➢A2A request Lifecycle

- The A2A request lifecycle is a sequence that details the four main steps a request follows: agent discovery, authentication, sendMessage API, and sendMessageStream API



<Source : Google>

Google’s A2A Framework

➢ A2A Framework Github and Overview

- Github: https://github.com/a2aproject/A2A
- A2A Overview : https://www.youtube.com/watch?v=Fbr_Solax1w

<Source : Google>

Thank You

<Source : Google>