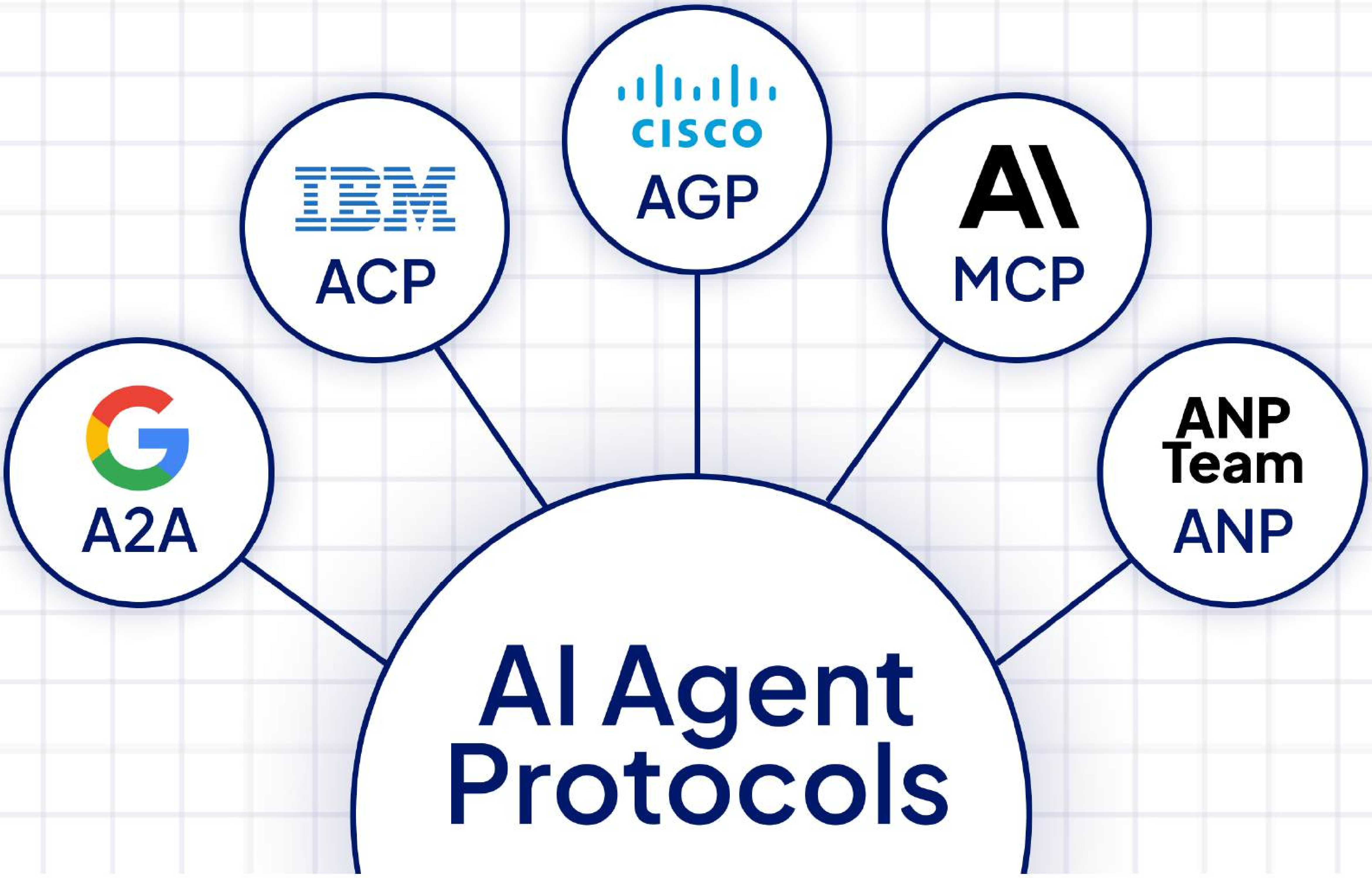


A2A

ACP

AGP

MCP

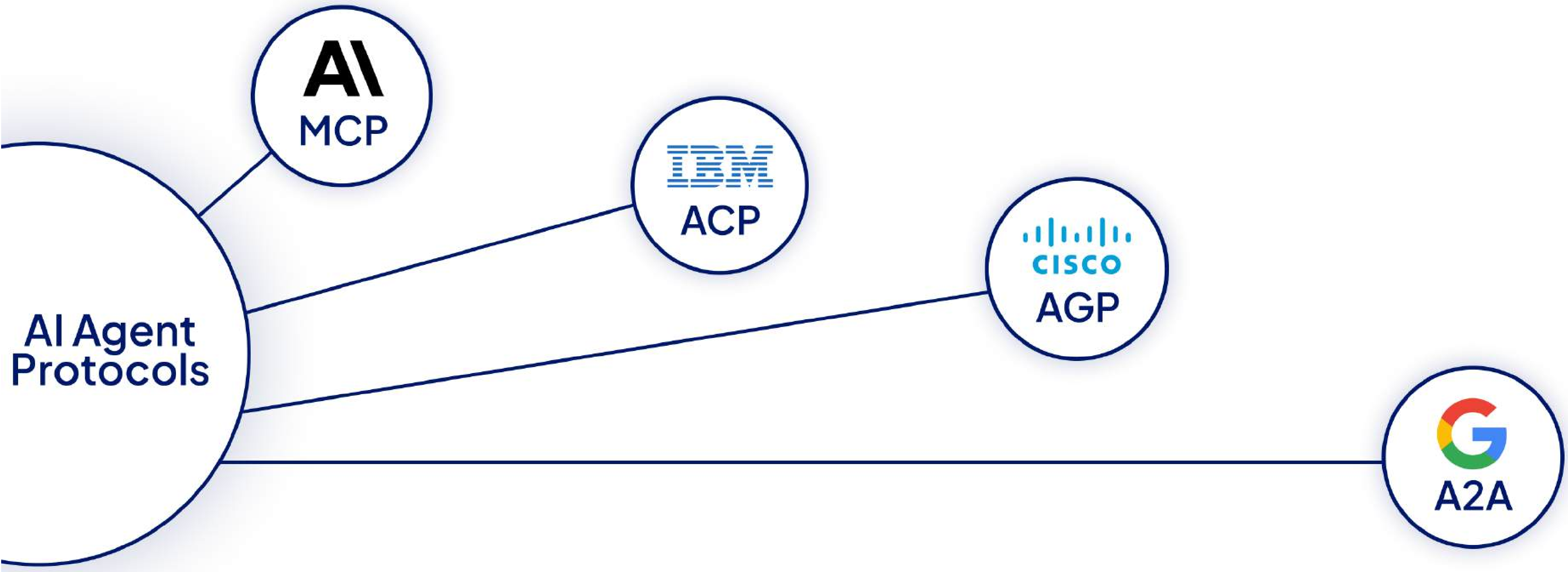
ANP Team

ANP

AI Agent Protocols

Overview of AI Agent Protocols

@rakeshgohel01



Intro

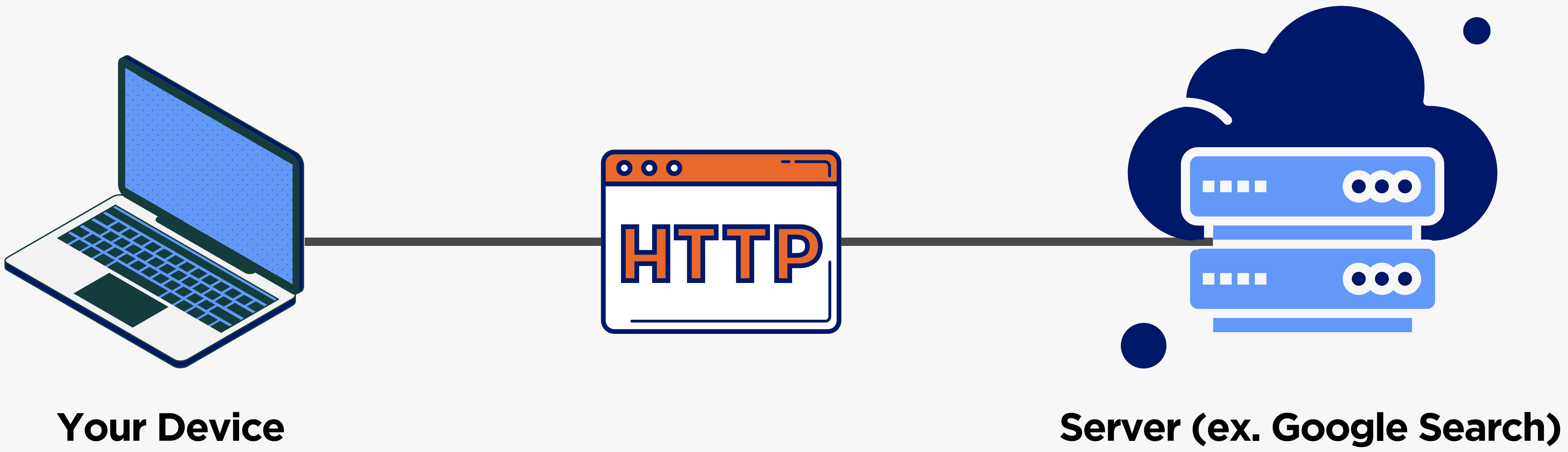
This year companies are going big on AI Agent Protocols

Today, we will take a look at few of these popular protocols and overview their workflow to understand which is best suited to what type of workflow →

WHAT ARE AI AGENT PROTOCOLS?

PROTOCOLS ARE A COMMON LANGUAGE FOR MULTI-AGENT SYSTEMS TO COMMUNICATE BETWEEN THE HETEROGENEOUS AGENTS.

"AI AGENT PROTOCOLS ARE LIKE HTTP FOR AI AGENTS"



WHY DO WE NEED THESE PROTOCOLS?

THIS IS BECAUSE

EARLIER APPROACHES WERE FRAMEWORK-SPECIFIC, AD HOC, AND NON-STANDARDIZED

THIS WAS LIKE PRIVATE CHATROOMS WHERE EVERYONE SPEAKS THEIR OWN NATIVE LANGUAGE.

The new age interoperability protocols changed the way we interact with multi-agents and even simplified tools integration for each agent.

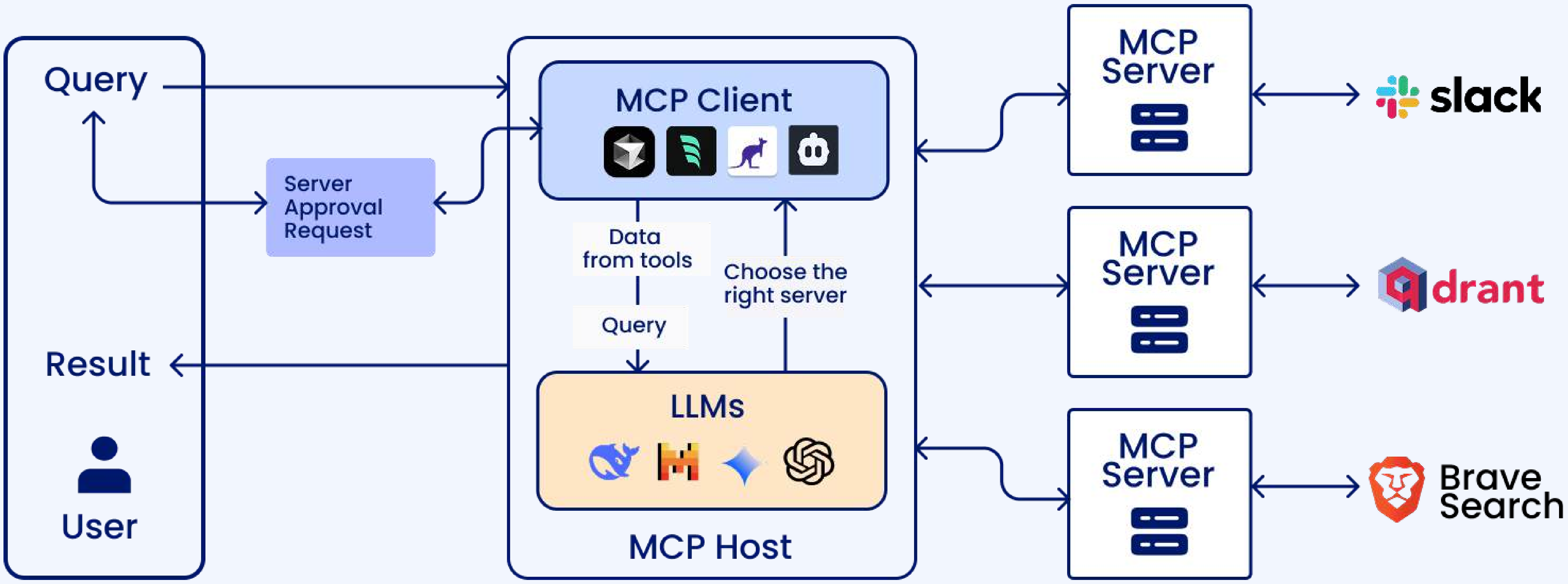
MCP (MODEL CONTEXT PROTOCOL)

—Started at Anthropic

MCP is an Open standard protocol for two-way communication between an LLM (client) and external systems like tools (servers).

Think of it like a USB-C Port for all the external data sources.

Overview of MCP Architecture



Let's understand the workflow behind the architecture

MCP Workflow

1. Query: This can be a prompt given to an MCP client asking to build an AI Agent that can do a specific task.

2. MCP Client: The MCP client intercepts the query and shares it with the Large Language Model.

3. Query: The initial query is sent to the LLM by the client.

4. LLMs: MCP Client uses an LLM, and that particular LLM is responsible for generating answers based on the query and also for choosing the right tool.

5. Chooses the right server: After understanding the context of the query, the LLM sends a response to the Client to choose an appropriate MCP server for the task.

6. Server Approval Request: After the LLM sends a server selection request, the client optionally shares an approval request with the user for Human-in-Loop security.

7. MCP Server processing: The chosen server is then used to complete the given task by the user, utilising the user's query and the tools' data.

8. Result: Finally, after the processing is done, the result is then shared with the user.

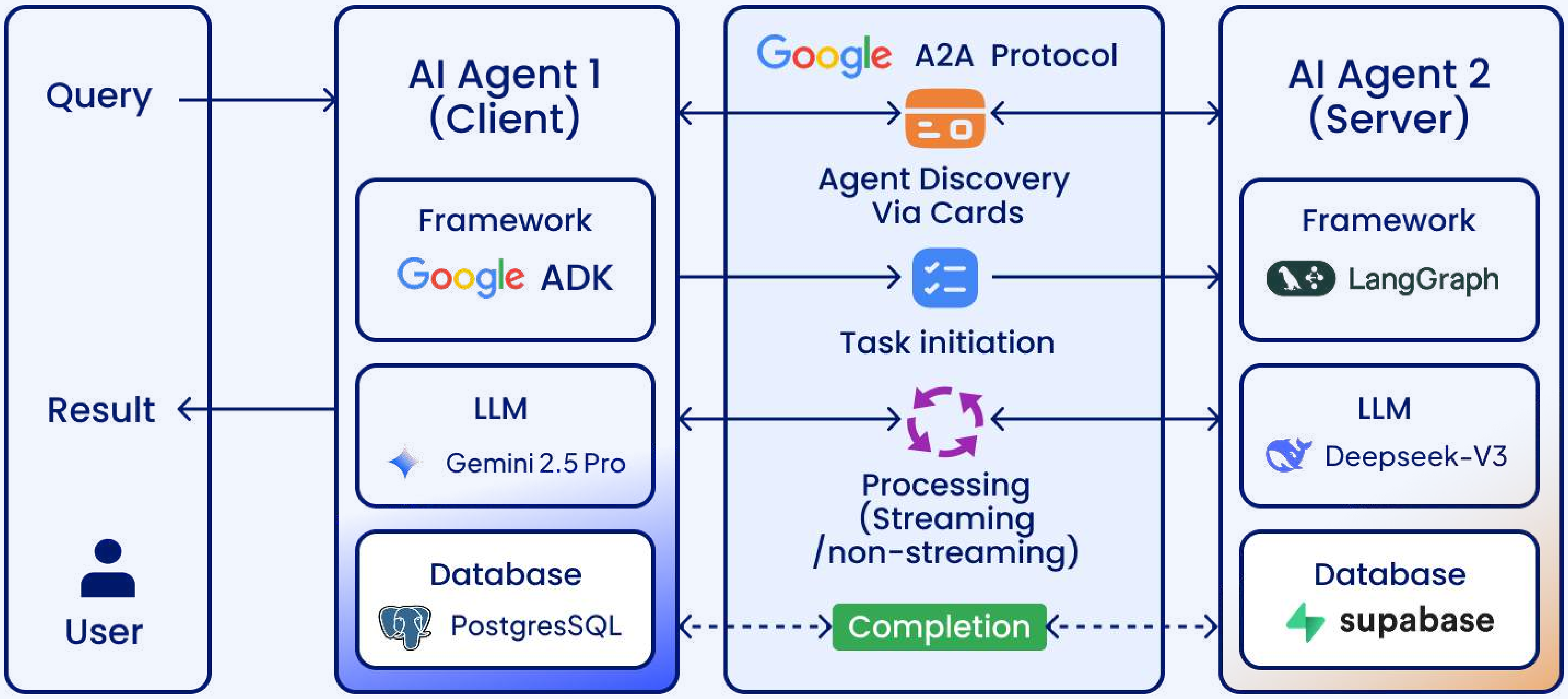
A2A (AGENT 2 AGENT PROTOCOL)

—Started at Google

It is an open, standard protocol that defines how AI agents—even if made by different vendors or built on different frameworks—can discover, communicate, and collaborate.

A2A handles agent ↔ agent interoperability, while MCP (Model Context Protocol) by Anthropic enables agents to call external tools or data sources.

Overview of A2A Architecture



Let's understand the workflow behind the architecture

A2A Workflow

1. Query: The user sends a query to AI Agent 1 (Client), requesting a specific task or information.

2. Agent Card: A public JSON file with an agent's capabilities, skills, endpoint URL, and authentication needs acts as a discovery card for clients.

- Through this Agent Card client discovers the capabilities of other agents, which helps them choose the best one for their current need.

3. Task: Task is the central unit of work. A client initiates a task by sending a message, and each Task have a unique ID and progresses through states.

4. Processing: The server either streams SSE events (status updates, artifacts) as the task progresses or processes the task synchronously, returning the final Task object in the response.

- **Interaction (Optional):** If a task requires input, the client sends further messages using the same Task ID via tasks/send or tasks/send Subscribe.

5. Completion: The task eventually reaches a terminal state.

6. Result: After the task is completed, the result is sent back to the user through their chosen client agent.

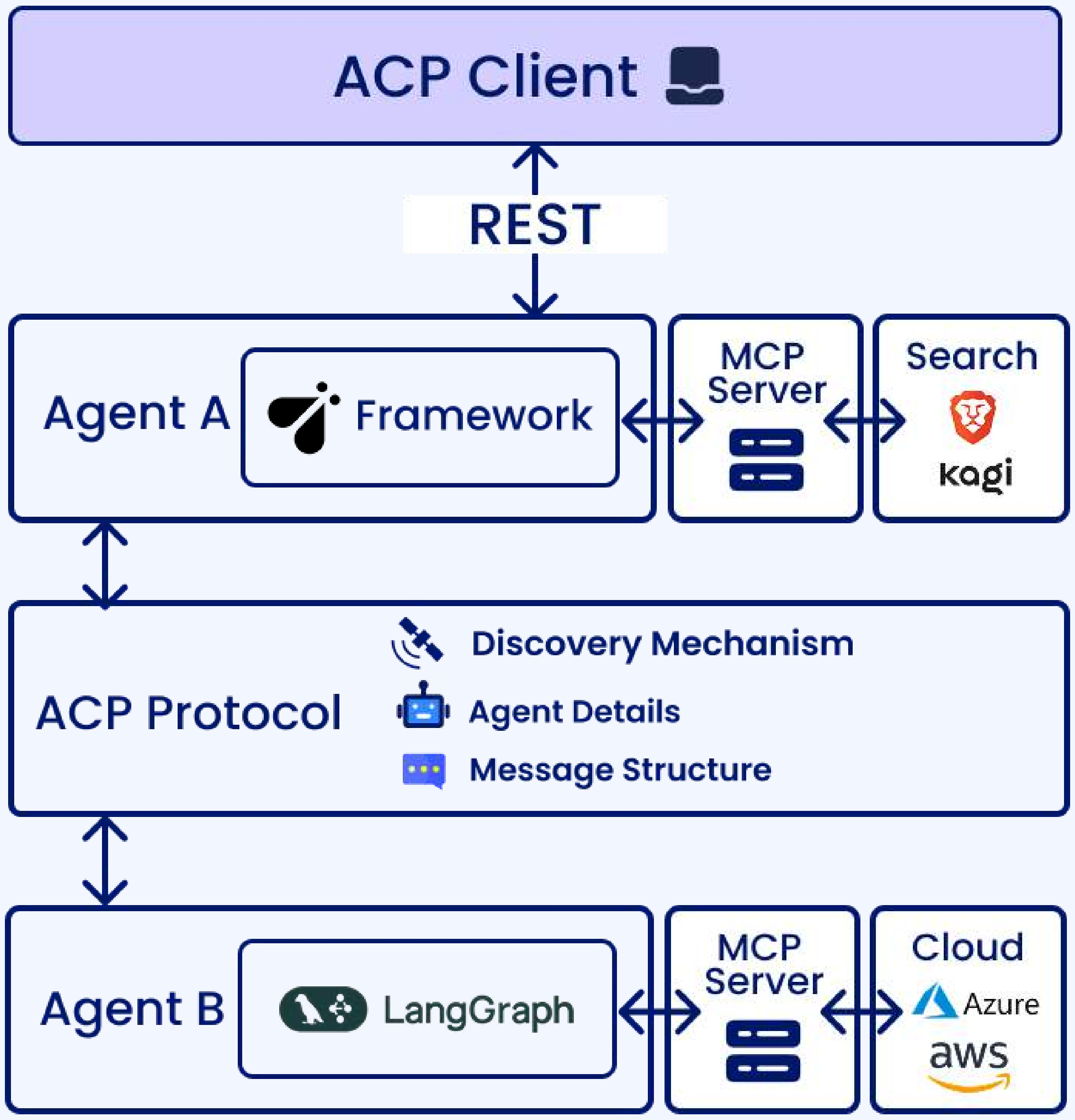
ACP (AGENT COMMUNICATION PROTOCOL)

IBM —Started at IBM

ACP is IBM's open, vendor-neutral protocol for standardizing communication between AI agents, released in May 2025 under the Linux Foundation via the BeeAI project.

It defines a RESTful, HTTP-based "wire format", enabling agents to exchange messages, assign tasks, and interact both synchronously and asynchronously.

Overview of ACP Architecture



Let's understand the workflow behind the architecture

ACP Workflow

1. Agent Discovery: The ACP Client discovers available agents (e.g., Agent A and B) via REST from a registry or shared manifest.

It retrieves metadata for Agent A and Agent B, including endpoints, capabilities, and security schemas—exposed via REST APIs.

2. Capability Identification: Agent A identifies Agent B's capabilities through metadata fetched from the manifest or registry.

3. **Token Acquisition:** Agent A obtains a signed capability token that defines what it is authorized to ask Agent B.

4. **Task Invocation:** Agent A sends a structured HTTP POST /run request to Agent B with the task details and capability token.

5. **Task Execution:** Agent B processes the request using its LangGraph framework and possibly interacts with cloud services via MCP.

6. **Response Streaming:** Agent B streams real-time progress or final results back to Agent A or the ACP Client using Server-Sent Events (SSE).

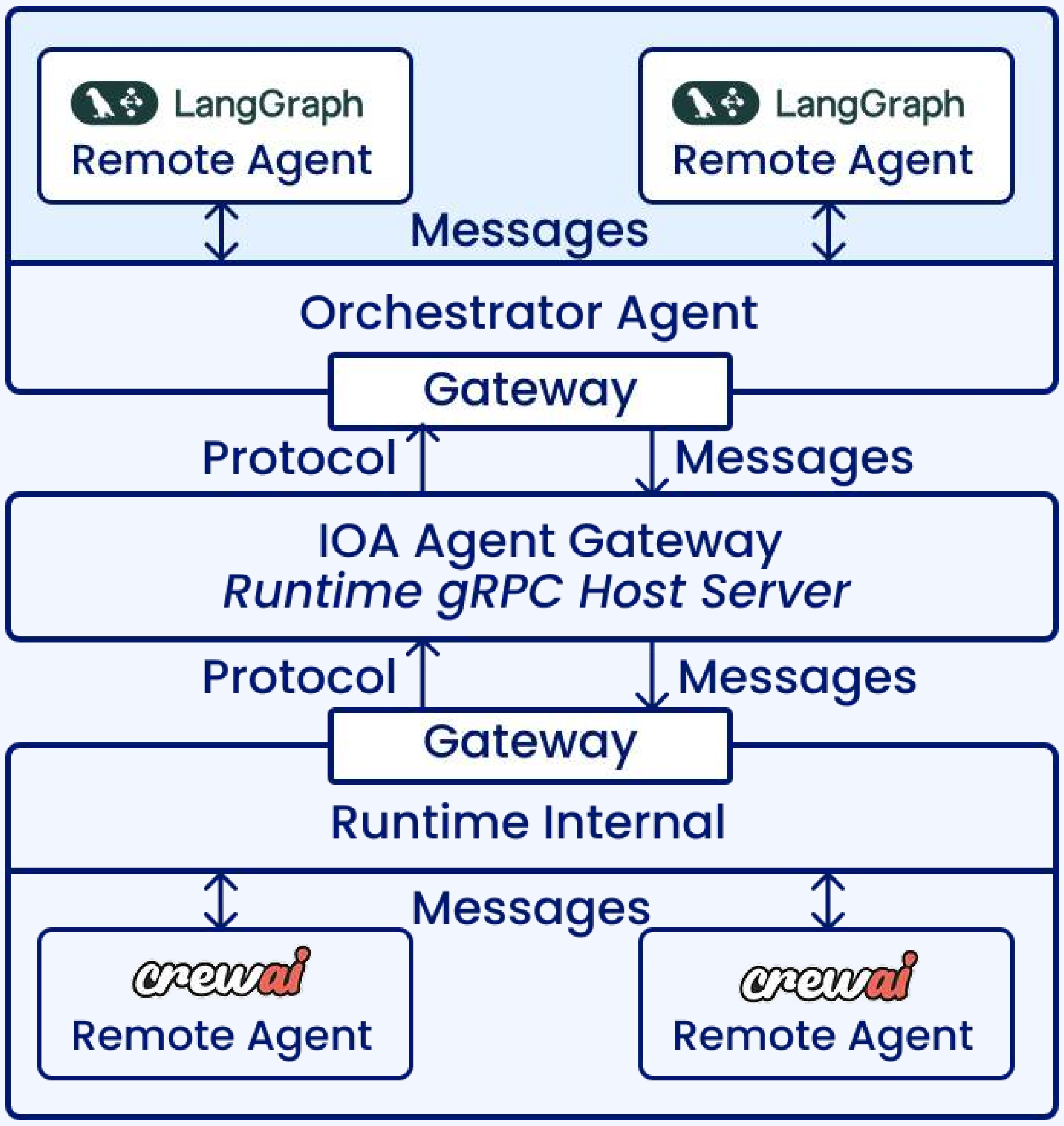
AGP (AGENT GATEWAY PROTOCOL)

CISCO —Started at Cisco

AGP is a part of Cisco's initiative called Outshift, which is focused on building the secure Internet of Agents.

Cisco AGP uses gRPC over HTTP/2 as a standardized transport layer with built-in support for streaming, pub/sub, and request-response patterns.

Overview of AGP Architecture



Let's understand the workflow behind the architecture

AGP Workflow

1. Agent Registration: Remote agents (e.g., LangGraph or CrewAI-based) register securely with a central IOA Agent Gateway using gRPC over the control plane, establishing identity and capabilities.

2. Gateway Setup: The Agent Gateway configures both control and data planes to manage authentication, messaging policies, namespaces, and agent routing.

3. Communication Patterns: Agents exchange messages through the gateway using AGP primitives that support request–response, pub/sub, fire-and-forget, and bidirectional streaming over gRPC.

4. Message Routing: The Gateway dynamically routes messages between agents—possibly including an orchestrator —applying policies and tenant isolation rules in real time.

5. Security Enforcement: All traffic is authenticated, authorized, and encrypted end-to-end, with transport security via HTTP/2 TLS and optionally payload encryption like MLS or payload-level encryption.

6. Observability & Management: The gateway provides observability, telemetry, token rotation, and discovery services (agent manifests/cards), enabling robust lifecycle and operational monitoring

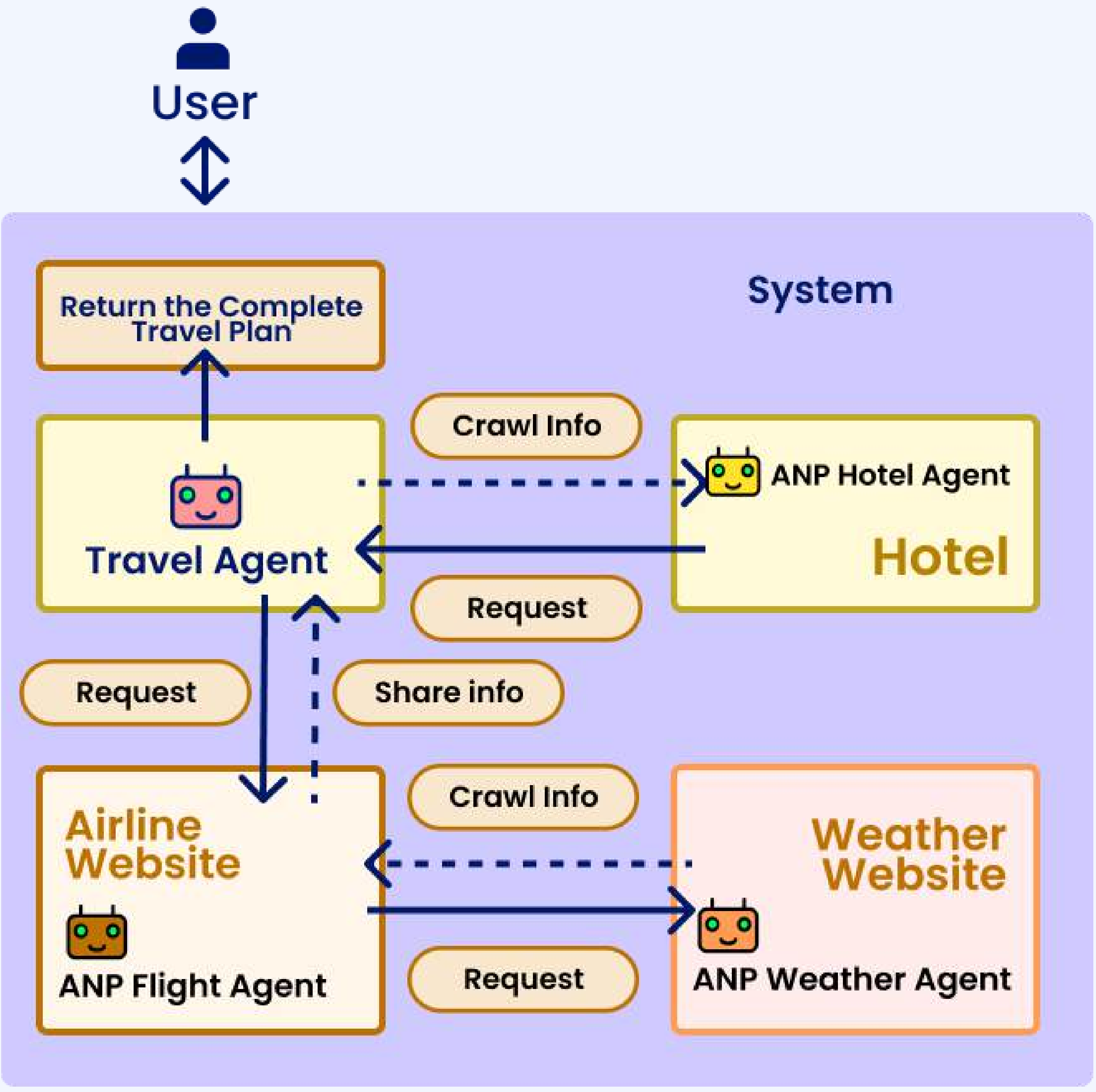
ANP(AGENT NETWORK PROTOCOL)

ANP—Started by ANP Team

ANP is an open-source framework that enables efficient interoperability among heterogeneous AI agents.

It communicates using JSON-LD, W3C Decentralized Identifiers (DID), and a layered architecture for discovery, negotiation, and collaboration.

Overview of ANP Architecture



Let's understand the workflow behind the architecture

ANP Workflow

1. Agent Discovery: A local agent queries a discovery path, typically a predefined endpoint, to retrieve a list of available agents using JSON-LD formatted metadata.

2. Accessing Agent Descriptions: The agent fetches detailed agent description files, structured with JSON-LD, which specify capabilities.

3. Interaction Initiation: The agent constructs requests adhering to the Application Protocol Layer's standardized formats

4. Authentication and Security: Leveraging W3C Decentralized Identifiers (DID), the agent appends cryptographic credentials to requests, enabling end-to-end encrypted communication via the Identity and Encrypted Communication Layer.

5. Request Transmission: The agent transmits requests over the Meta-Protocol Layer, which supports dynamic protocol negotiation using natural language.

6. Response Processing: The agent processes responses formatted according to the Application Protocol Layer, parsing JSON-LD or similar structured data to extract and utilize the provided information or task outcomes.

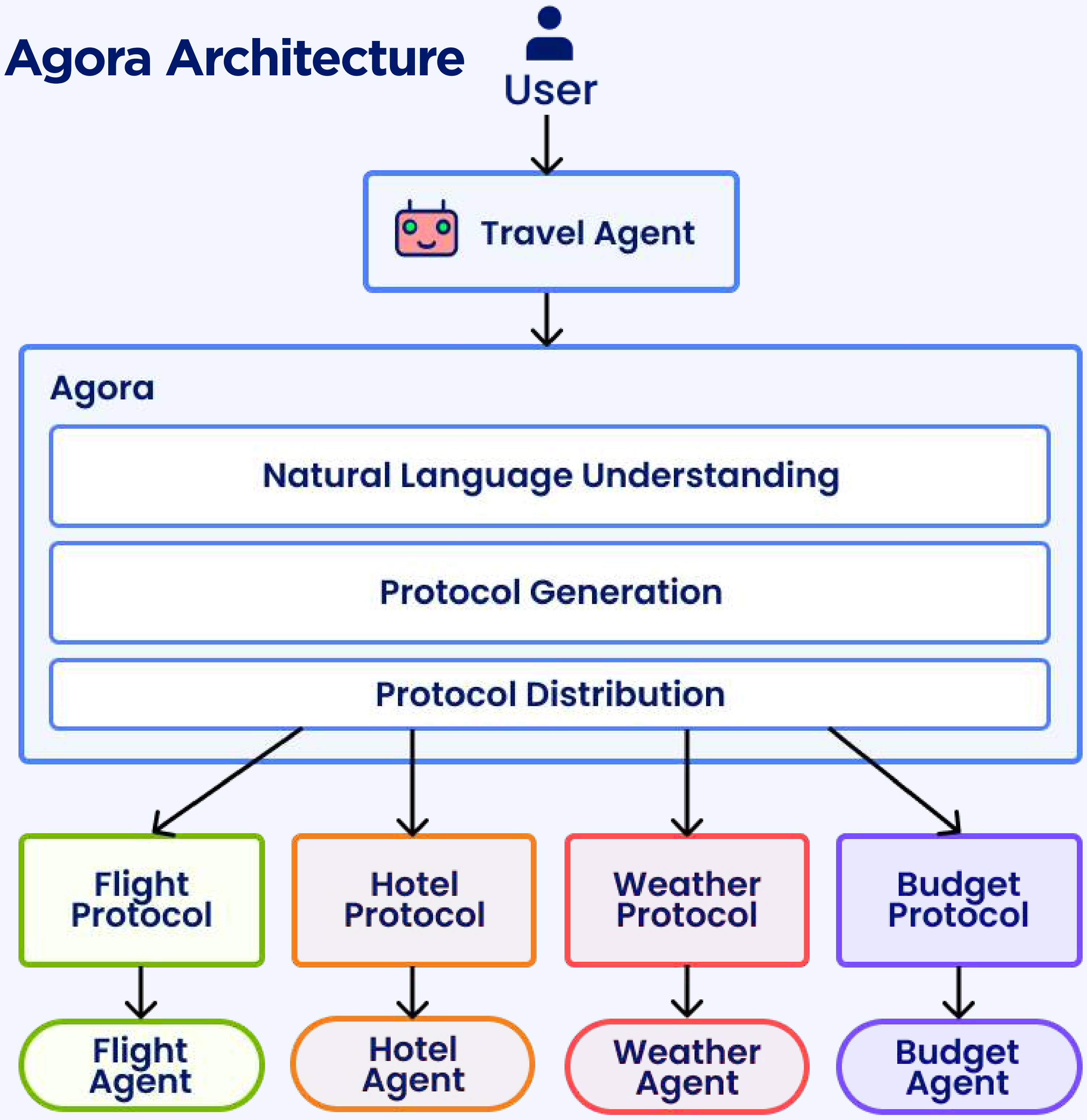
AGORA

UNIVERSITY OF OXFORD —**Started at University of Oxford**

Agora is a scalable protocol enabling LLM agents to negotiate communication protocols using natural language and JSON

Agora uniquely transforms natural language requests into standardized protocols for flexible, user-centric agent interactions

Overview of Agora Architecture



Let's understand the workflow behind the architecture

Agora Workflow

1. User Request Parsing: A local agent interprets natural language user queries, extracting structured intent components using LLM capabilities.

2. Agent Capability Query: The agent retrieves metadata about available services, detailing their supported protocols and functionalities in JSON format.

3. Protocol Generation: The agent transforms parsed intent into standardized communication protocols tailored for specific service types via Agora’s generation layer.

4. Authentication Integration: The agent embeds secure authentication tokens, ensuring trusted interactions with services using protocol-defined security mechanisms.

5. Protocol Distribution: The agent dispatches generated protocols to specialized service agents through Agora’s distribution layer for task execution.

6. Response Aggregation: The agent collects and processes structured responses from services, integrating results to fulfill the user’s request.

Best Use-case for each Protocol

MCP(MODEL CONTEXT PROTOCOL)

- Best for Allowing AI assistants to access and act on real-time, external context in a secure and standardized way.

A2A(AGENT 2 AGENT PROTOCOL)

- Best for enabling a network of enterprise modular AI agents to collaboratively solve multi-step, real-world tasks through structured, asynchronous, inter-agent communication.

ACP (AGENT COMMUNICATION PROTOCOL) IBM

- Best for enabling AI agents to work together on complex tasks, integrating legacy software, streaming data between agents, handling request cancellation, and ensuring persistency

Best Use-case for each Protocol

AGP (AGENT GATEWAY PROTOCOL) CISCO

- Best for composing event-driven multi-agent workflows, enabling agents to operate seamlessly across network boundaries.

ANP(AGENT NETWORK PROTOCOL) ANP Team

- Best for collaboration among agents from diverse providers and structures in multi-agent scenarios, specifically designed for cross-domain agent communication and interoperability.

AGORA UNIVERSITY OF OXFORD

- Best for natural language to protocol generation, allowing users to define agent interactions and workflows using natural language.

Conclusion

As Multi-Agent architectures become even more relevant in multiple AI Agent products and services,

The need for a bridge that would allow them to communicate with each other will always be an important aspect of the entire workflow.

If you want to learn more,

Each protocol's repo/documentation is attached in the caption of this document's post.

Hi, I am
Rakesh Gohel

“We help businesses 10X their growth with Cloud and AI Agents”

FOLLOW TO LEARN MORE ABOUT AI AGENTS

linkedin.com/in/rakeshgohel01