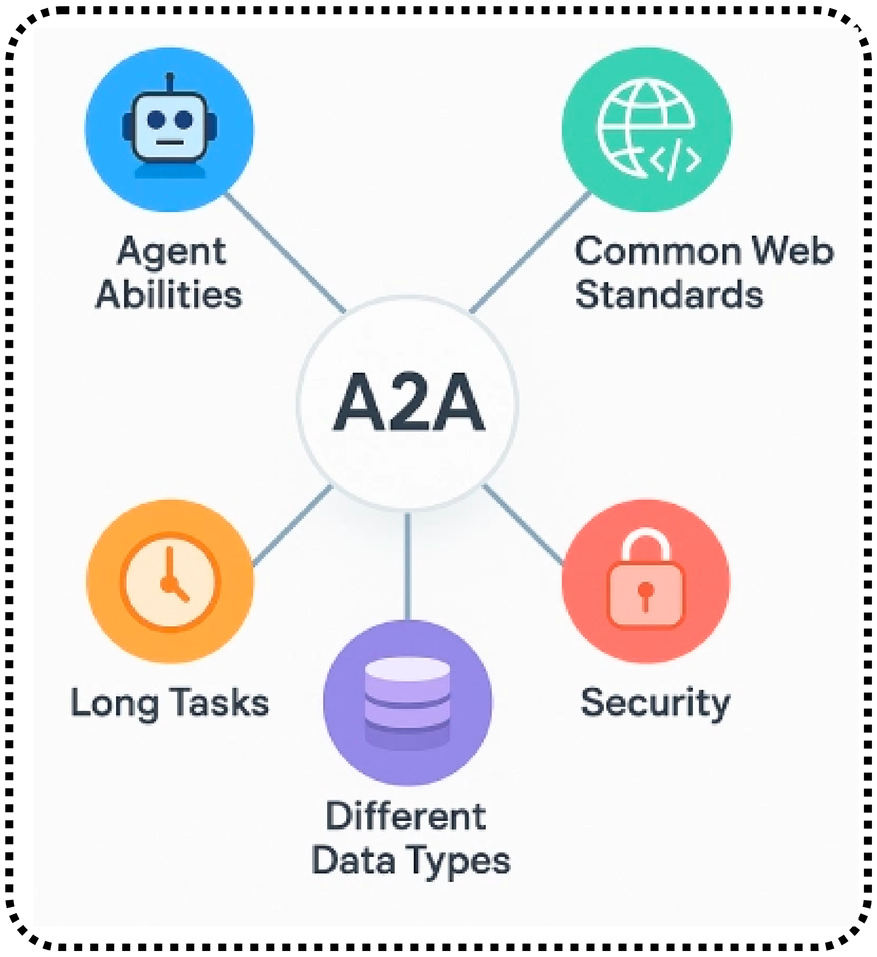
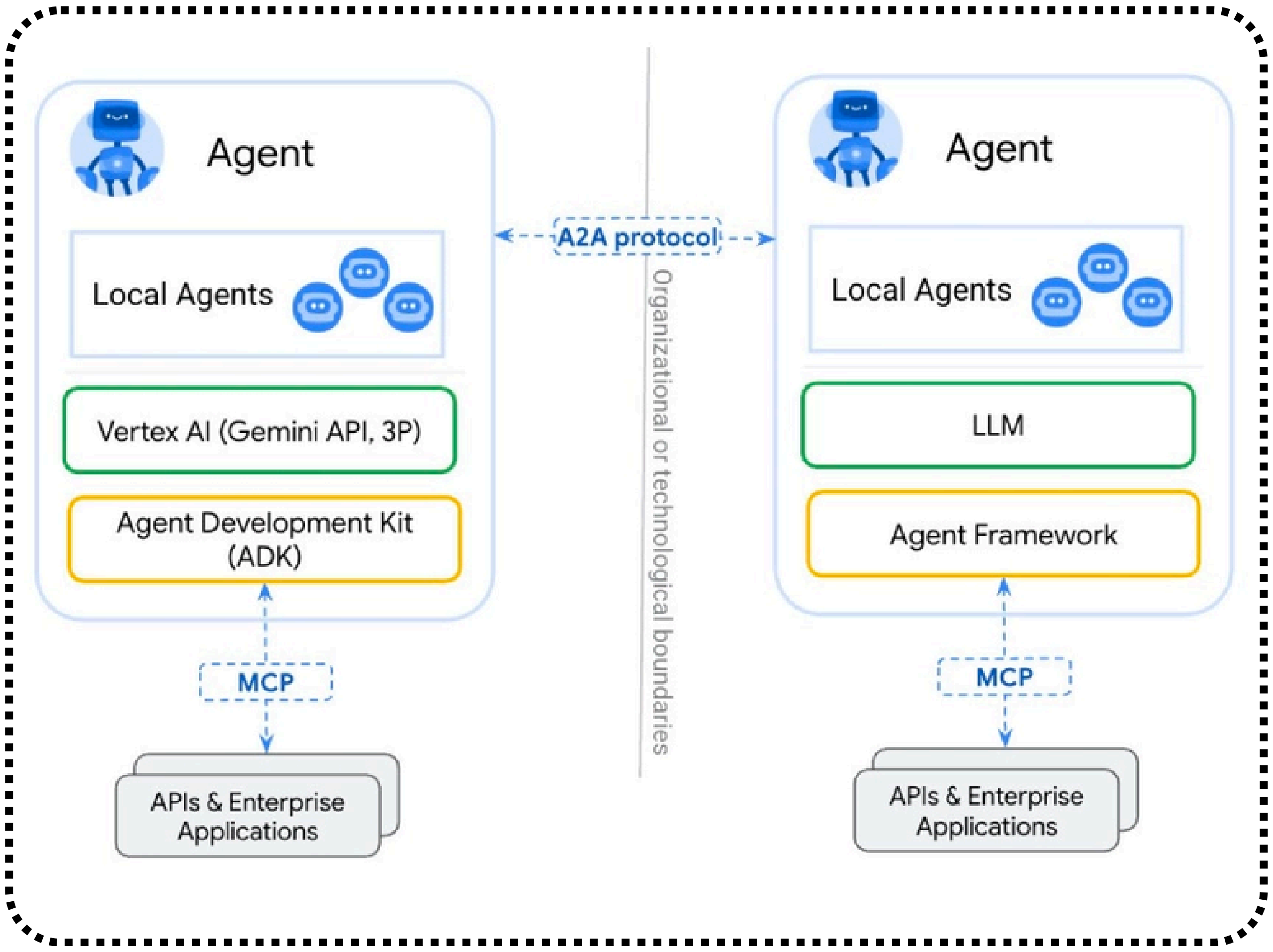
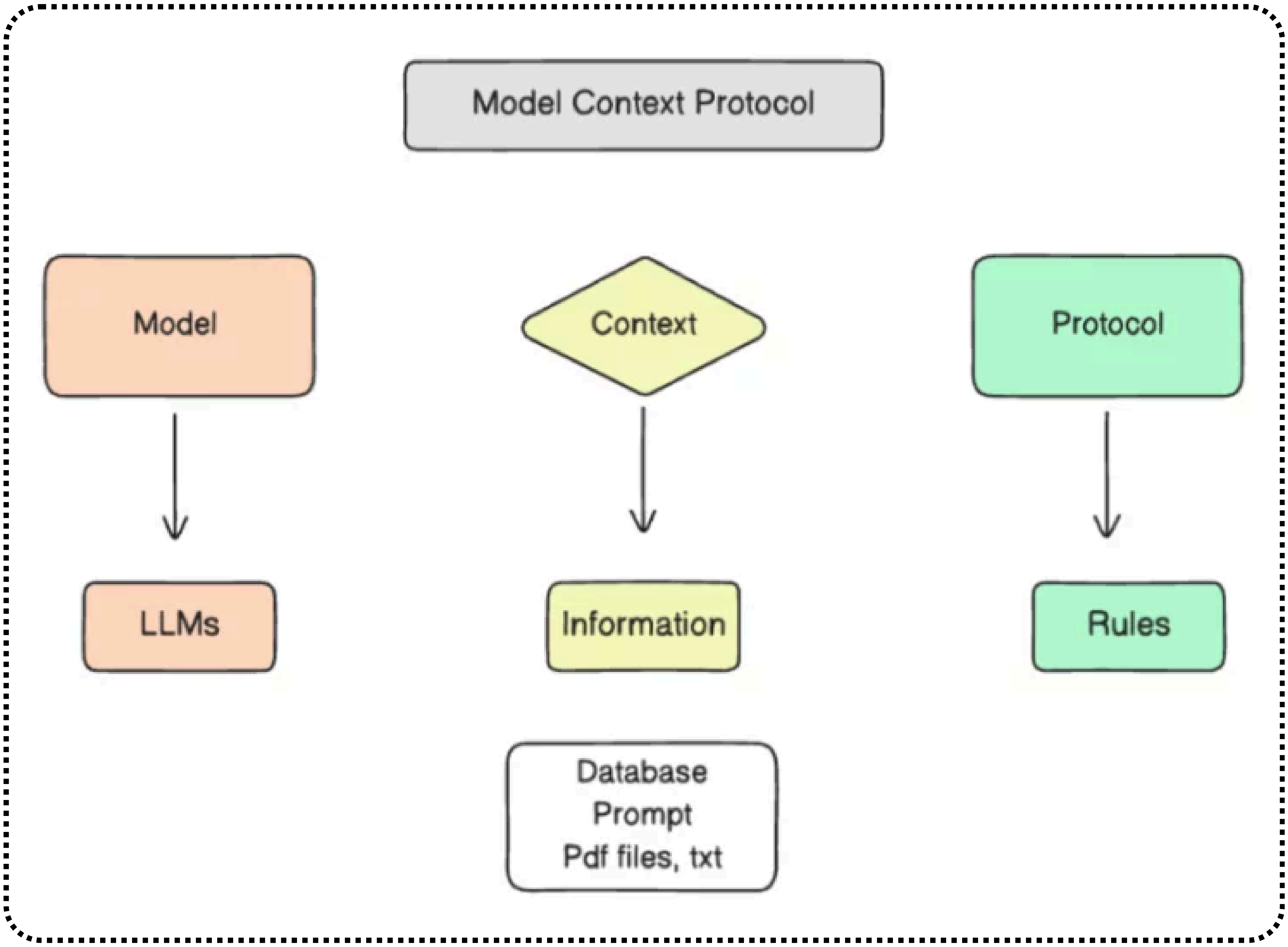
Difference between A2A and MCP



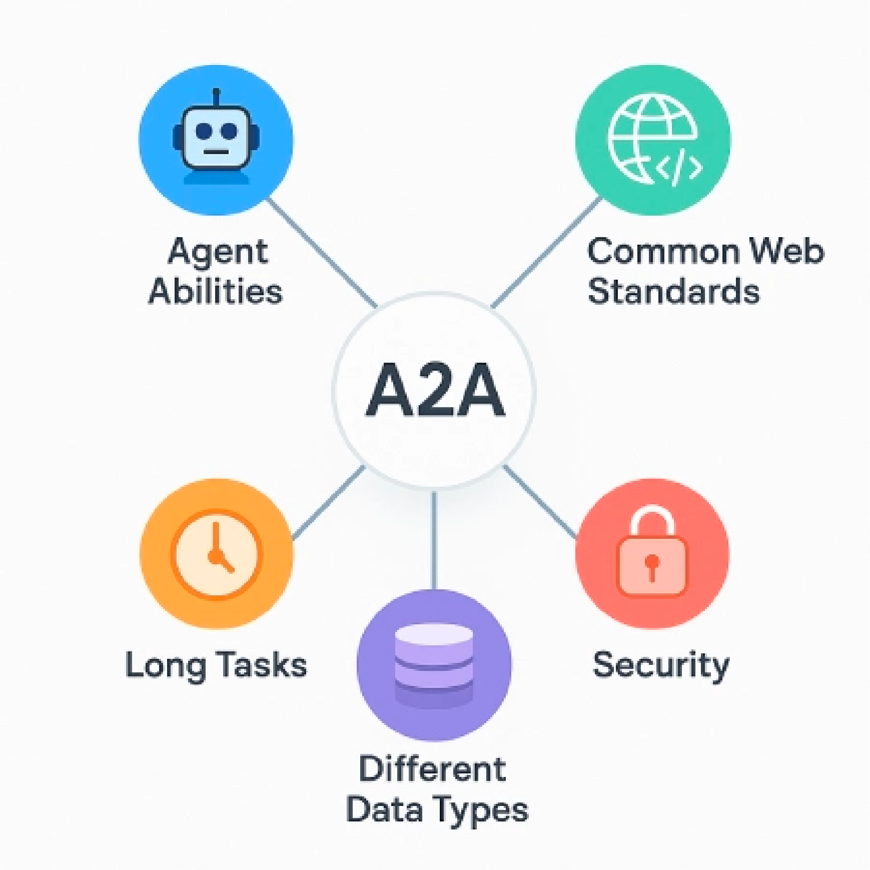

Agent
Local Agents
Vertex AI (Gemini API, 3P)
Agent Development Kit (ADK)
MCP
APIs & Enterprise Applications
A2A protocol
Organizational or technological boundaries
Agent
Local Agents
LLM
Agent Framework
MCP
APIs & Enterprise Applications



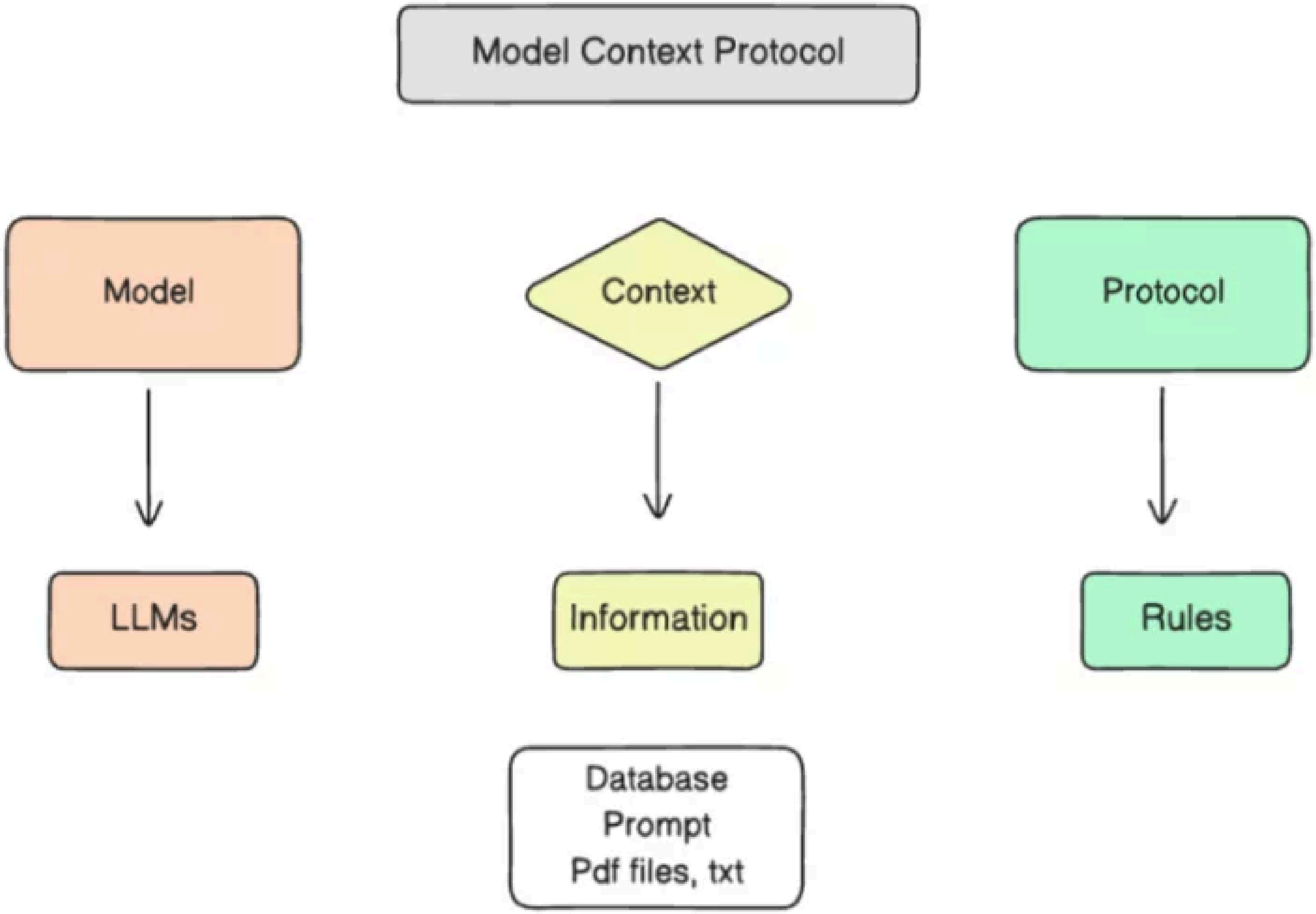
What is A2A (Agent-to-Agent)?

Agent2Agent (A2A) is an open protocol by Google that standardizes how AI agents communicate and collaborate. Essentially, A2A allows independent AI agents built by different vendors or running on different platforms to form a common language for cooperation. Using A2A, agents can exchange goals, share context, and invoke actions with each other in a secure, structured manner.

The protocol was explicitly designed to allow multi-agent workflows that span across different clouds, applications, or services. A2A is built on familiar web standards such as HTTP, making it easier to integrate it into existing IT stacks.

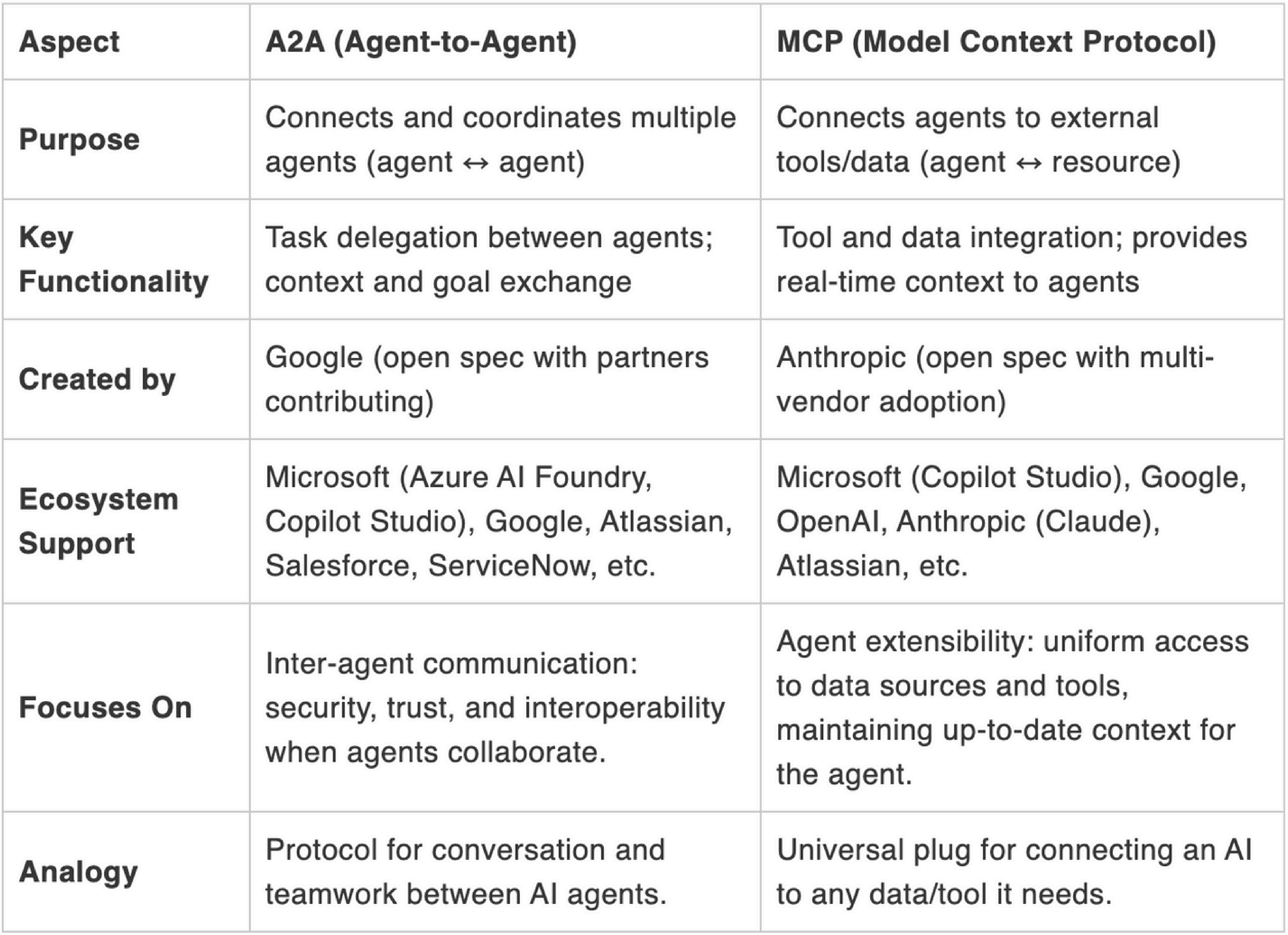


What is MCP



The Model Context Protocol (MCP) was introduced by Anthropic (parent company of Claude), which allows connecting AI agents (or LLMs) to external tools. If A2A is about agent-to-agent communication, MCP is about agent-to-resource integration. It provides a unified, standardized way for AI models to access various data sources, knowledge bases, and services that are outside the model's own parameters. That is why it is commonly referred to as the "USB-C port" for AI applications.

A2A vs MCP



Aspect	A2A (Agent-to-Agent)	MCP (Model Context Protocol)
Purpose	Connects and coordinates multiple agents (agent ↔ agent)	Connects agents to external tools/data (agent ↔ resource)
Key Functionality	Task delegation between agents; context and goal exchange	Tool and data integration; provides real-time context to agents
Created by	Google (open spec with partners contributing)	Anthropic (open spec with multi-vendor adoption)
Ecosystem Support	Microsoft (Azure AI Foundry, Copilot Studio), Google, Atlassian, Salesforce, ServiceNow, etc.	Microsoft (Copilot Studio), Google, OpenAI, Anthropic (Claude), Atlassian, etc.
Focuses On	Inter-agent communication: security, trust, and interoperability when agents collaborate.	Agent extensibility: uniform access to data sources and tools, maintaining up-to-date context for the agent.
Analogy	Protocol for conversation and teamwork between AI agents.	Universal plug for connecting an AI to any data/tool it needs.

This table summarizes the differentiated roles of A2A vs MCP

How They Work Independently

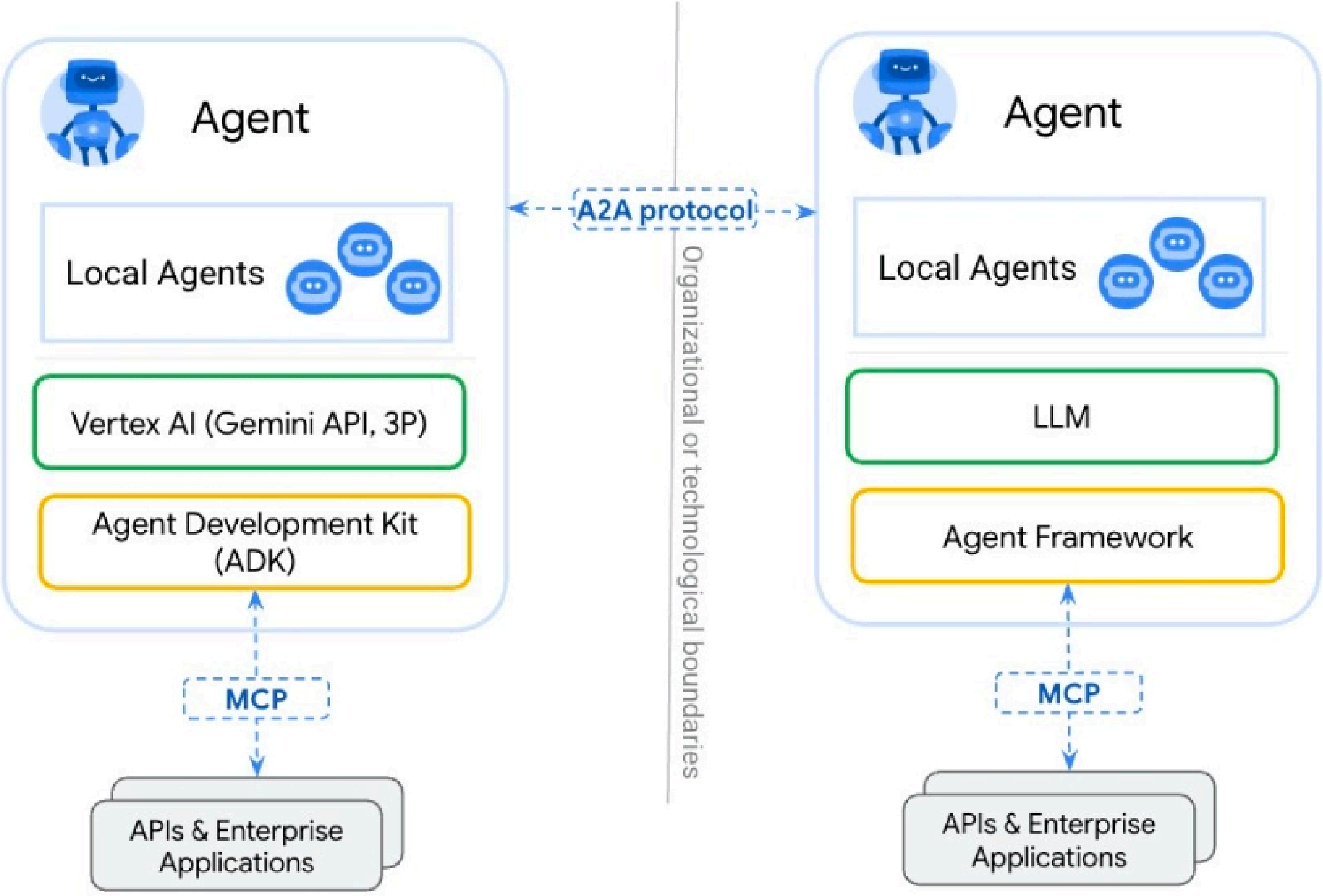
A2A Alone: Picture a company with specialized AI agents in domains such as finance, marketing, and scheduling. A master agent can delegate tasks like budgeting or timeline planning to others using A2A. Each agent contributes results back through a shared protocol.

Without MCP, though, each agent relies only on its internal knowledge or hardwired connections.
MCP Alone: Imagine a support chatbot connected to live systems such as product databases, shipping APIs, and knowledge bases using MCP.

This setup makes the agent dynamically aware and actionable in real time. Even without A2A, MCP turns it into a tool-rich, responsive assistant. However, it can’t coordinate across multiple agents to solve complex or multi-step problems.

Independently, both protocols bring clear value. A2A enables modular teamwork, whereas MCP allows agents to have external functionality.

Integration (Better Together)


Agent
Local Agents
Vertex AI (Gemini API, 3P)
Agent Development Kit (ADK)
MCP
APIs & Enterprise Applications
A2A protocol
Organizational or technological boundaries
Agent
Local Agents
LLM
Agent Framework
MCP
APIs & Enterprise Applications

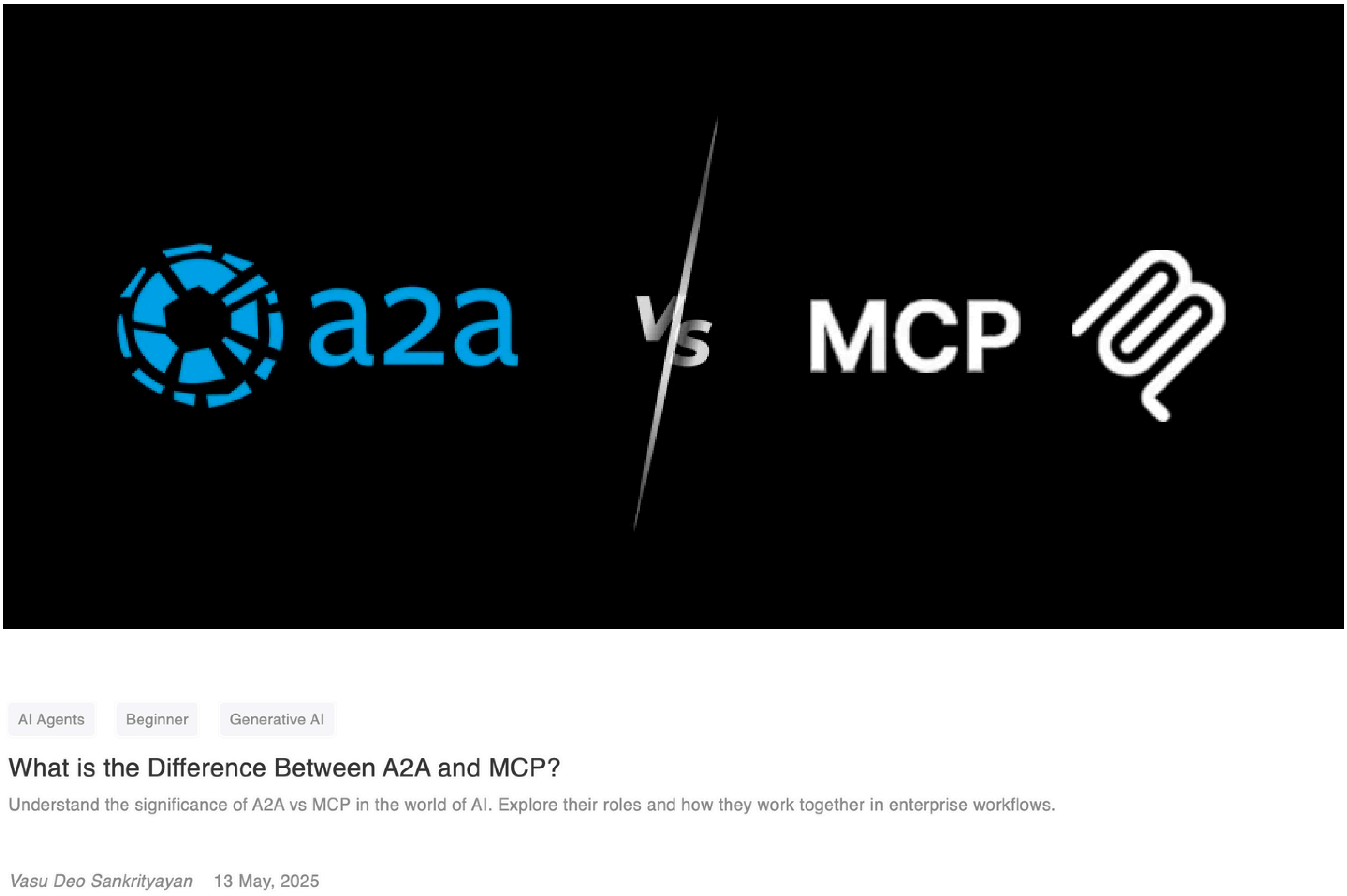
In modern GenAI systems, A2A and MCP often operate together to enable intelligent orchestration:

Layered Cooperation: Think of MCP as the foundation for tools and data access and A2A as the coordination layer that delegates tasks among agents.

In a supply chain example, agents fetch inventory data, handle procurement, and manage delivery using MCP, while A2A enables them to share tasks and results.

Unified Development Experience: Microsoft Copilot Studio showcases this integration. Developers can register MCP tools and link agent workflows via A2A, all in one interface. A2A handles the flow, and MCP handles the function.

For more information, please visit:



AI Agents Beginner Generative AI

What is the Difference Between A2A and MCP?

Understand the significance of A2A vs MCP in the world of AI. Explore their roles and how they work together in enterprise workflows.

Vasu Deo Sankrityayan 13 May, 2025