短期連載

AIエージェント同士で連携！

通信プロトコル Agent2Agent

新連載

第1回 Agent2Agent（A2A）の基礎知識

鶴 英雄

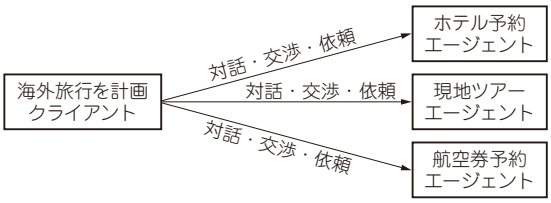

図1 A2Aは水平連携

● 登場のきっかけ

MCP（Model Context Protocol）によって，LLM（Large Language Models）はツールへのアクセスが容易になりました．LLMという知能に手足がついた感じです．

MCPはLLMがツールを使うイメージで標準化されました．その道具がさらに知性を備えた，つまりエージェント（Agent）が提供する機能だったらどうでしょうか．

今のMCPでもある程度エージェント間で仕事を割り振ることが可能です．ただし，もともとの発想がツールを使うことにあるので，自律的に動作するエージェント間の仕事分担ではその能力を十分生かしきれないでしょう．

そこでという訳ではないですがエージェント間のやり取りの標準化が検討されてきました．それがAgent2Agent（A2A）プロトコルです．本連載では，今後注目が集まるであろうA2Aについて3回に分けて解説します．

第1回 基礎知識
第2回 最小のサンプルで見るA2Aサーバの作りと通信（Hello World）
第3回 LLMと外部APIを呼び出すA2Aサーバ作り

● 現在はLinux Foundationが開発を進める

A2Aプロトコルはエージェント間のコラボレーションを主眼に設計されたプロトコルです．もともとはグーグルが開発を進めていましたが，2025年6月にLinux Foundationへ移管されました．メインとなるウェブ・ページは次の通りです．

```
https://github.com/a2aproject
https://a2a-protocol.org/
```

表1 A2AとMCPの違い

項 目	A2Aプロトコル	MCP
主な役割	エージェント同士の会話と協調を標準化	エージェント（LLM）と外部ツール＆データの接続を標準化
用途	マルチエージェント・システムにおける連携	単一エージェントの機能拡張（ツール利用）
動作イメージ	チームワークのための共通言語，対話ルール	道具（ツール）を使うための標準的な接続ポート（例：USB Type-C）
利用場面	複雑なタスクの役割分担，タスクの調整，情報交換	データベースからの情報検索，外部APIの実行，計算処理
通信レイヤ（イメージ）	エージェント間のメッセージ交換層	エージェント内から外部リソースへのアクセス層
登場の理由	複数のエージェントによる自律的な問題解決	エージェント実行能力の向上

● MCPとの違い…通信相手もエージェント

A2Aは複数のエージェントが連携してある（ロング）タスクを行う場合のやり取りの方法を標準化します．A2Aは自律的なエージェント・チームが対話をしながら仕事を進めるイメージで，対等な関係に基づくネットワーク型，水平連携志向と言えるでしょう（**図1**，**表1**）．

一方，MCPは主従関係に基づくスター型，垂直統合志向と言えそうです．MCPはPCにおけるUSBの形に近く，受動的な道具である各種リソースをホストが利用するイメージです（**図2**）．

比較的単純な機能であればMCPで機能を提供し，手元のLLMで処理を組み立てる方が簡単でしょう．一方で非常に複雑なことを事細かく制御しようとすることは大変で，間違いが起こりやすくなります．ある程度高い抽象度の作業を依頼したいですし，エージェントであればそれが可能なはずです．

例えば，海外旅行を計画することを考えます．航空