第1章 3つの作例を通じてMCPによるLLM外部拡張のポイントを学ぶ

ラズパイでも手軽に試せる！自作AIエージェントの構成

佐藤 聖

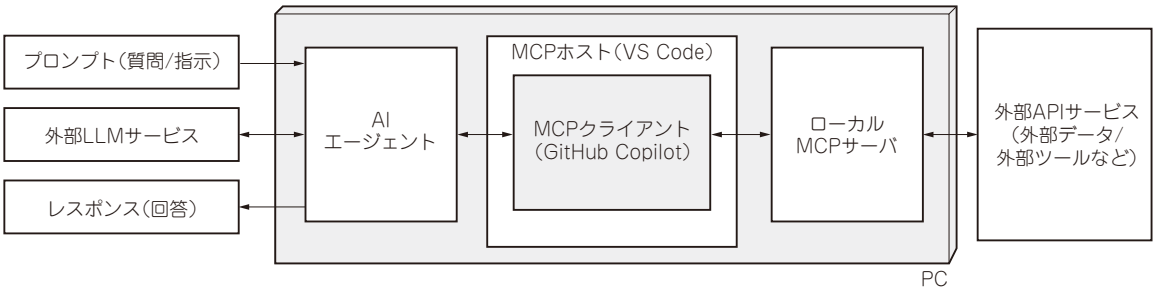

図1　第2部でやること…LLMとMCPを連携させるAIエージェントを自作する
実験のシステム構成．MCPサーバはローカル環境に構築する

MCPによるLLM外部連携はAIエージェントが鍵となる

● 定型的な指示やツールの設定を事前に組み込んでおける

第2部では，LLM（Large Language Models）とMCP（Model Context Protocol）を組み合わせることで，AIエージェントが外部サービスや環境とどのように連携できるのかを，具体的な実験を通して理解します．通常，LLMへの質問や指示，さらにはMCPサーバの各種ツールの指定まで，毎回プロンプトで入力する必要があります．しかし，AIエージェントを活用すれば，定型的な指示やツールの設定を事前にシステム・プロンプトとして組み込んでおけるので，ユーザは簡単なプロンプトを入力するだけで済みます．

さらにLLMには，過去リクエストなどを記録するメモリ機能が備わっていることも多く，過去の処理や会話を覚えてくれるため，AIエージェントを使えば使うほど効率良く作業を進められるようになります．AIエージェントを構築できると，LLMの力を最大限に引き出し，複雑な作業や繰り返しタスクもスムーズに処理できるようになります．

● AIエージェントの主要素

AIエージェントは，次の3つの要素で動作します．

- LLM（例：ChatGPT）→エージェントの頭脳．指示を理解し，必要に応じて外部サービスを使おうとする
- MCPクライアント→LLMの指示を受け取り，「このURLからデータを呼び出してほしい，ツールで処理してほしい」という要求をMCPサーバへ渡す
- MCPサーバ→外部サービスのAPIを実際に呼び出し，その結果をクライアントに返す

● 構成…MCPサーバはローカルに配置

前項の3つは，別の場所に置くこともできますが，第2部の実験では，LLM以外（MCPサーバとMCPクライアント）を自分のPC内に置きます（**図1**）．

この構成で外部サービスとやり取りすると安全性はどうなるのでしょうか．ローカルMCPサーバを使った場合，AIエージェントを起動すると，MCPクライアントとMCPサーバの通信は全て手元のPC内で完結します．つまり，外部と通信するのはMCPサーバ⇔外部サービスAPIの部分だけです．

この構成のメリットは，セキュリティ対策の対象が大幅に絞れることです．PC内部での通信は外へ露出しないため，複雑な内部通信の安全対策を心配する必要はありません．注意すべきは，外部APIへアクセスする部分だけです．今回の実験でも，このローカル構成を採用し，安全にAIエージェントと外部APIの連携を検証していきます．