Claude の Model Context Protocol (MCP)

AIと外部システムの連携を可能にする標準プロトコル

MCPとは何か？

- Anthropicが開発した**オープンプロトコル**
- AIアシスタントと外部システムを接続する**標準インターフェース**
- 「AIアプリケーション向けのUSB-Cポート」のような役割
- AIが様々なツールやデータソースと安全に対話できる仕組み

概念図: Claude AIと外部システム（ファイルシステム、データベース、APIなど）がMCPを介して双方向に通信する標準インターフェース

MCPの主な特徴

✅ 標準化されたインターフェース
✅ 双方向通信をサポート
✅ セキュリティを確保
✅ 拡張性と相互運用性

MCPの仕組み

MCPの基本構成

- **MCP Server**: 外部システムとの接続を提供
- **MCP Client**: AIアシスタント側（Claude）
- **Tool**: MCPサーバーが提供する機能単位
- **Transport**: 通信方式（標準入出力、HTTP等）

ツールとコンテキスト

- AIからツールを呼び出し、結果を受け取る
- 複数のツールを組み合わせて複雑なタスクを実行
- ツールは型付きスキーマで定義される

アーキテクチャ概要:

- Claude AI → MCP Client → Transport → MCP Server
- MCP Serverは各種ツール（ファイル操作、データベース、API）を提供
- 各ツールは対応する外部システムと連携

MCPは多層構造で安全な通信を実現

MCPで接続可能なシステム・サービス

カテゴリ	具体例	主な用途
ローカルシステム	ファイルシステム、SQLite/PostgreSQL、実行環境	ローカルファイルの読み書き、データベース操作、コード実行
クラウドサービス	Google Drive, Slack, GitHub, Google Calendar	ドキュメント管理、コミュニケーション、コード管理、予定管理
開発ツール	Git, Puppeteer, VS Code	バージョン管理、ウェブ自動化、コード開発
その他	Figma, YouTube, Pandoc, Blender	デザイン連携、動画分析、文書変換、3Dモデリング

拡張性: 新しいシステムやサービスに対応するMCPサーバーを自作することも可能

MCPサーバーの実装例

基本的なMCPサーバー

```
import { Server } from "@modelcontextprotocol/sdk/server";
import { StdioServerTransport } from
  "@modelcontextprotocol/sdk/server/stdio";

const server = new Server({
  name: "hello-world-server",
  version: "1.0.0",
});

// ツールの定義
server.tool(
  "hello", // ツール名
  { name: { type: "string" } }, // 入力スキーマ
  async ({ name }) => {
    return {
      content: [{ type: "text", text: `Hello, ${name}!` }]
    };
  }
);

// サーバー起動
const transport = new StdioServerTransport();
await server.connect(transport);
```

ファイル操作MCPサーバー

```
import { Server } from "@modelcontextprotocol/sdk/server";
import { StdioServerTransport } from
  "@modelcontextprotocol/sdk/server/stdio";
import fs from "fs/promises";

const server = new Server({
  name: "filesystem-server",
  version: "1.0.0",
});

server.tool(
  "read_file",
  { path: { type: "string" } },
  async ({ path }) => {
    const content = await fs.readFile(path, "utf-8");
    return {
      content: [{ type: "text", text: content }]
    };
  }
);

// サーバー起動
const transport = new StdioServerTransport();
```

MCPの設定と使用方法

設定方法

1. MCPサーバーのコードを作成・実行
2. Claude for Desktopの設定ファイルを編集

```
{
  "mcpServers": {
    "myserver": {
      "command": "node",
      "args": ["path/to/your/mcp-server.js"]
    }
  }
}
```

3. Claudeを再起動

利用方法

- Claudeに直接指示するだけで連携が可能

実際の活用例

- ローカルファイルの分析と要約
- データベースからの情報取得と可視化
- 外部APIを使った情報収集と整理
- 複数のシステム間でのワークフロー自動化

MCPの主なメリット

✅ データをクラウドに送信せずローカル処理
✅ 複数のシステムを統合的に利用可能
✅ カスタムツールで独自機能を拡張
✅ セキュリティとプライバシーの確保

外部API連携のMCPサーバー例

```
import { Server } from "@modelcontextprotocol/sdk/server";
import { StdioServerTransport } from "@modelcontextprotocol/sdk/server/stdio";
import axios from "axios";

const server = new Server({
  name: "weather-server",
  version: "1.0.0",
});

// 天気情報取得ツール
server.tool(
  "get_weather",
  { city: { type: "string" } },
  async ({ city }) => {
    const apiKey = process.env.WEATHER_API_KEY;
    const url = `http://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${apiKey}`;

    const response = await axios.get(url);
    const weather = response.data.weather[0].description;
    const temp = response.data.main.temp - 273.15; // Convert to Celsius

    return {
      content: [{ type: "text", text: `${city}の天気は${weather}で、気温は${temp.toFixed(1)}°Cです。` }]
    };
  }
);

const transport = new StdioServerTransport();
await server.connect(transport);
```

MCPの将来性と発展

今後の展望

- **エコシステムの拡大**: より多くのサービス・ツールとの連携
- **複雑なワークフロー**: 複数のツールを組み合わせた高度な処理
- **標準化の進展**: AIエージェント間の相互運用性向上
- **セキュリティ強化**: より安全な認証・権限管理の仕組み

課題

- MCPサーバー開発の敷居の低減
- パフォーマンスと安定性の向上

MCPの応用領域

- **企業内システム連携**: 社内ツールとAIの橋渡し
- **パーソナルアシスタント**: 個人のデータと連携した支援
- **開発者支援**: コード生成・テスト・デプロイの自動化
- **データ分析**: 複数ソースからのデータ統合と分析

エコシステム図: Claudeが様々なシステムと連携可能

- **ローカルシステム**: ファイルシステム、データベース、コード実行環境
- **クラウドサービス**: Google Drive、GitHub、Slack

まとめ

MCPの主要ポイント

- AIと外部システムを**標準化されたプロトコル**で接続
- ローカルシステム、外部API、各種サービスとの連携が可能
- **JavaScript/TypeScript**を使った簡単なサーバー実装
- Claude for Desktopでの利用がすぐに可能

学習リソース

- Anthropic MCP公式ドキュメント
- Model Context Protocol SDK
- Claude for Desktop設定ガイド

MCPの本質

MCPは単なる技術仕様ではなく、AIと既存システムの共存・連携の基盤として、AIの実用性と有用性を大きく高める可能性を秘めています。

オープンなプロトコルとして公開されることで、エコシステム全体の発展に貢献し、AIの適用範囲を大きく広げるでしょう。

「MCPはAIアプリケーション向けのUSB-Cポート」
- Anthropic より -