



2025.5.26

MCP勉強会 -技術深掘り-



ソフトウェアエンジニア

大坪 悠 Yu Otsubo



職歴：

新潟県出身。大学卒業後、2016年にKDDIに新卒入社。
通信設備の運用、社内データ基盤構築のPM、Webチャットボット開発などを経験する。
スタートアップへの転職を経て2024年にKAG入社。
KAGでは生成AIを活用したアプリケーション開発に従事。

著書：

「K.A.G Tech Book」(KDDIアジャイル開発センター)
「やさしいMCP入門」(秀和システム)

 @tubone24

 tubone-project24.xyz

 mugimugi.cutedog

趣味：

愛犬と遊ぶこと
個人開発



モットー：

早く小さく失敗する



プリミティブ

MCPはツールが主要機能ですが、他にも機能があります

- リソース
- プロンプト
- ツール**

(とはいえ、ほとんどツール利用用途でMCPを使っている)

(規格はあるけど実装は進んでいないもの)

- サンプリング
- ルート

<https://modelcontextprotocol.io/clients>

Example Clients

Copy page

A list of applications that support MCP integrations

This page provides an overview of applications that support the Model Context Protocol (MCP). Each client may support different MCP features, allowing for varying levels of integration with MCP servers.

Feature support matrix

Client	Resources	Prompts	Tools	Discovery	Sampling	Roots	
Sire	✗	✗	✓	?	✗	✗	Supports to
AgentAI	✗	✗	✓	?	✗	✗	Agent Libra with tools s
AgenticFlow	✓	✓	✓	✓	✗	✗	Supports to resources fr agents and workflows.
Amazon Q CLI	✗	✓	✓	?	✗	✗	Supports pr
Apify MCP Tester	✗	✗	✓	✓	✗	✗	Supports re and tool dis
BeeAI Framework	✗	✗	✓	✗	✗	✗	Supports to workflows.
BoltAI	✗	✗	✓	?	✗	✗	Supports to
Claude.ai	✓	✓	✓	✗	✗	✗	Supports to resources fr servers.
Claude Code	✗	✓	✓	✗	✗	✗	Supports pr
Claude Desktop App	✓	✓	✓	✗	✗	✗	Supports to resources fr MCP server
Cline	✓	✗	✓	✓	✗	✗	Supports to
Continue	✓	✓	✓	?	✗	✗	Supports to resource

リソースはLLMの知識拡張に使う読み取り専用オブジェクト

よくMCPで表現される図はリソースを

指していそうな図

(厳密にはリソースだけじゃないですが…)

これが、MCP = RAGと言われてしまう所以

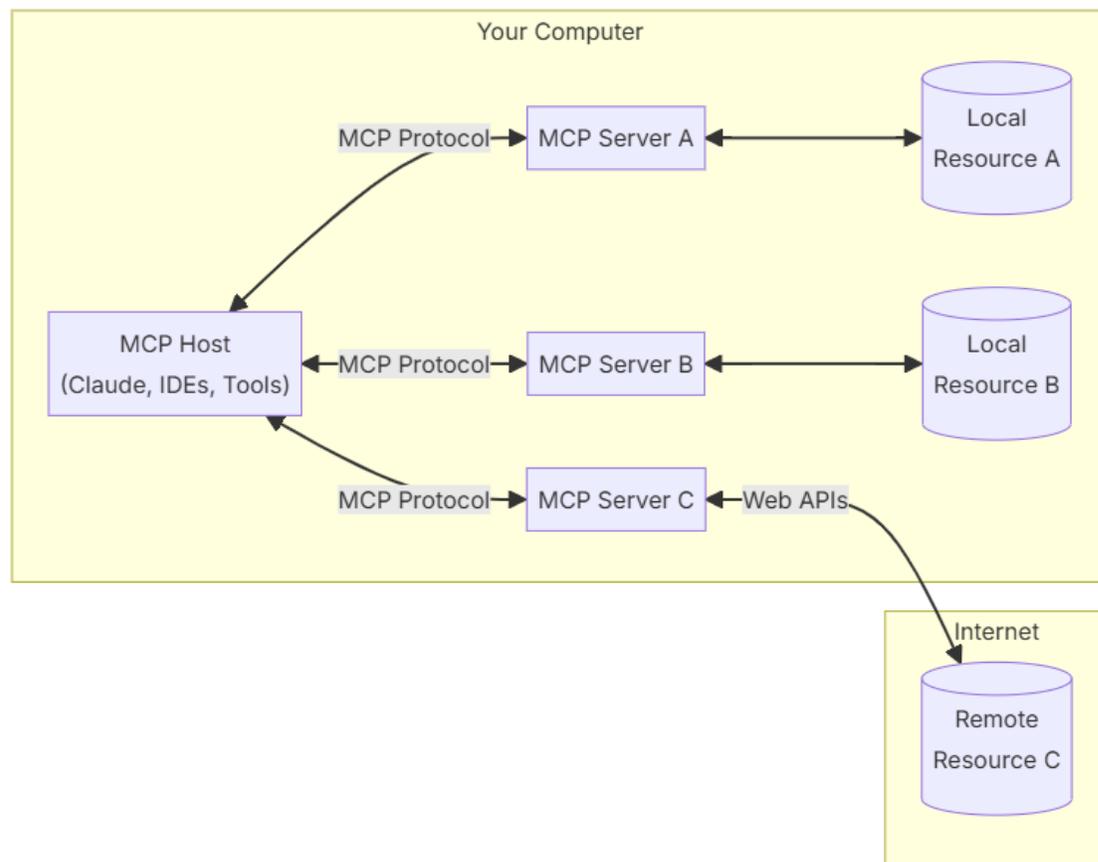
次のようなURIで表現される

note://

Schema

hoge hoge/text.md

Contents



<https://modelcontextprotocol.io/introduction#general-architecture>

プロンプトは用意されたプロンプトテンプレートを提供

Share context with Claude

Claude can receive information like prompts and attachments from specialized servers using Model Context Protocol. [Learn more about MCP.](#)

Choose an integration ▾

weather

weather

利用するプロンプトを指定する

Fill Prompt Arguments

city

Palo Alto

Submit

プロンプトに埋める内容を入力

weather.txt

0.03 KB • 1 line • Formatting may be inconsistent from source

What is the weather in "Palo Alto"?

- Christmas Day: Mostly sunny with a high of 14°C
- Tuesday Night: Clearing up to mostly clear skies, cooling to 8°C

The weather pattern shows some rain and unsettled conditions through Christmas Eve, but Christmas Day is looking nice with mostly sunny skies. Would you like me to provide

完成版の"プロンプト"が完成

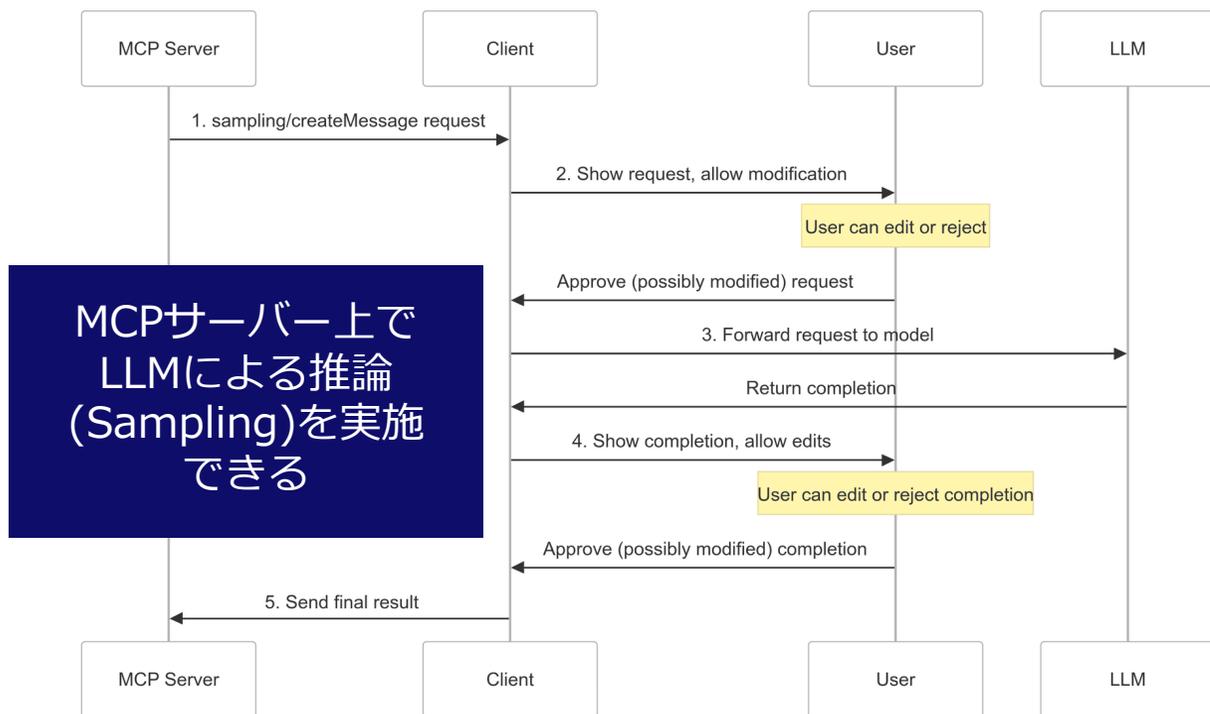
【ユースケース】

プロンプトエンジニアリング難しいユーザーへの補助として。

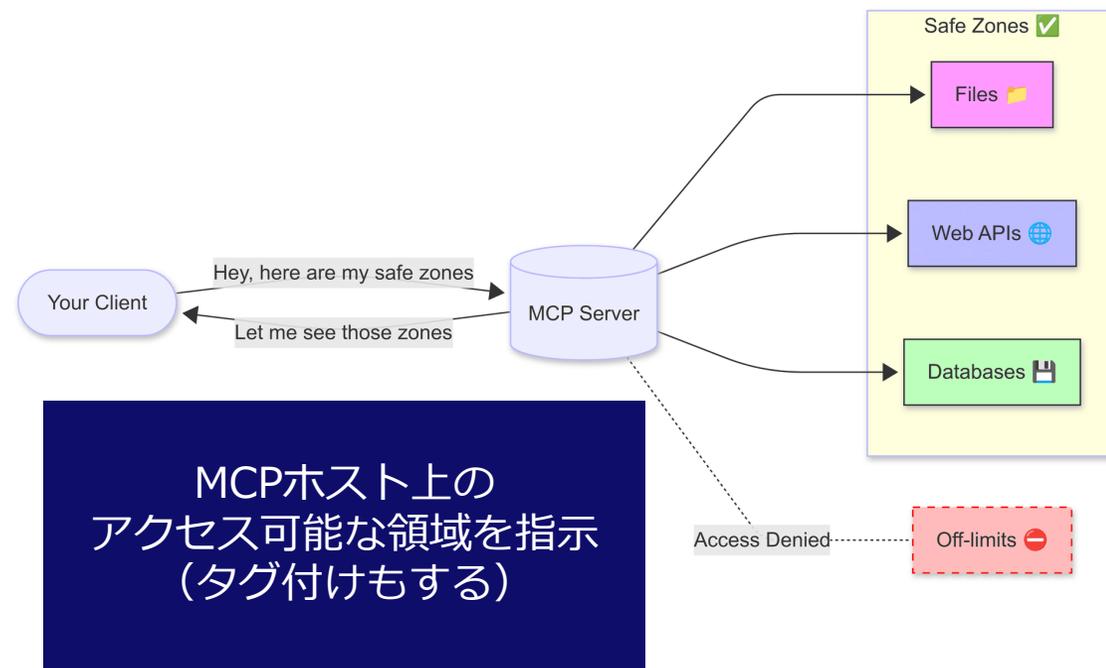
アプリ: AWSブログレベル判定くんとか？

サンプリング・ルートは現時点サポートされているMCPホストが皆無 今後のサポートに期待しましょう！

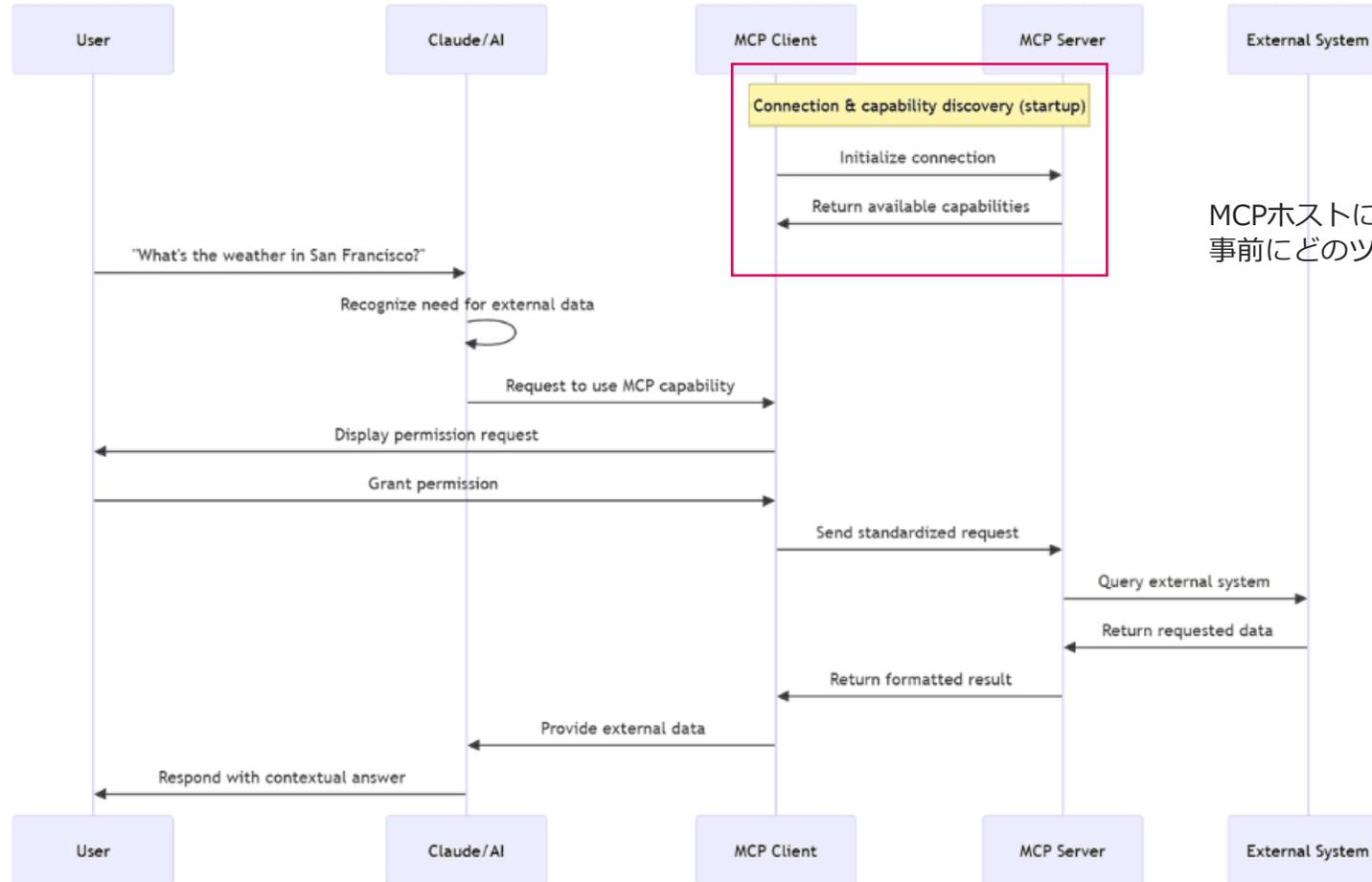
サンプリング



ルート



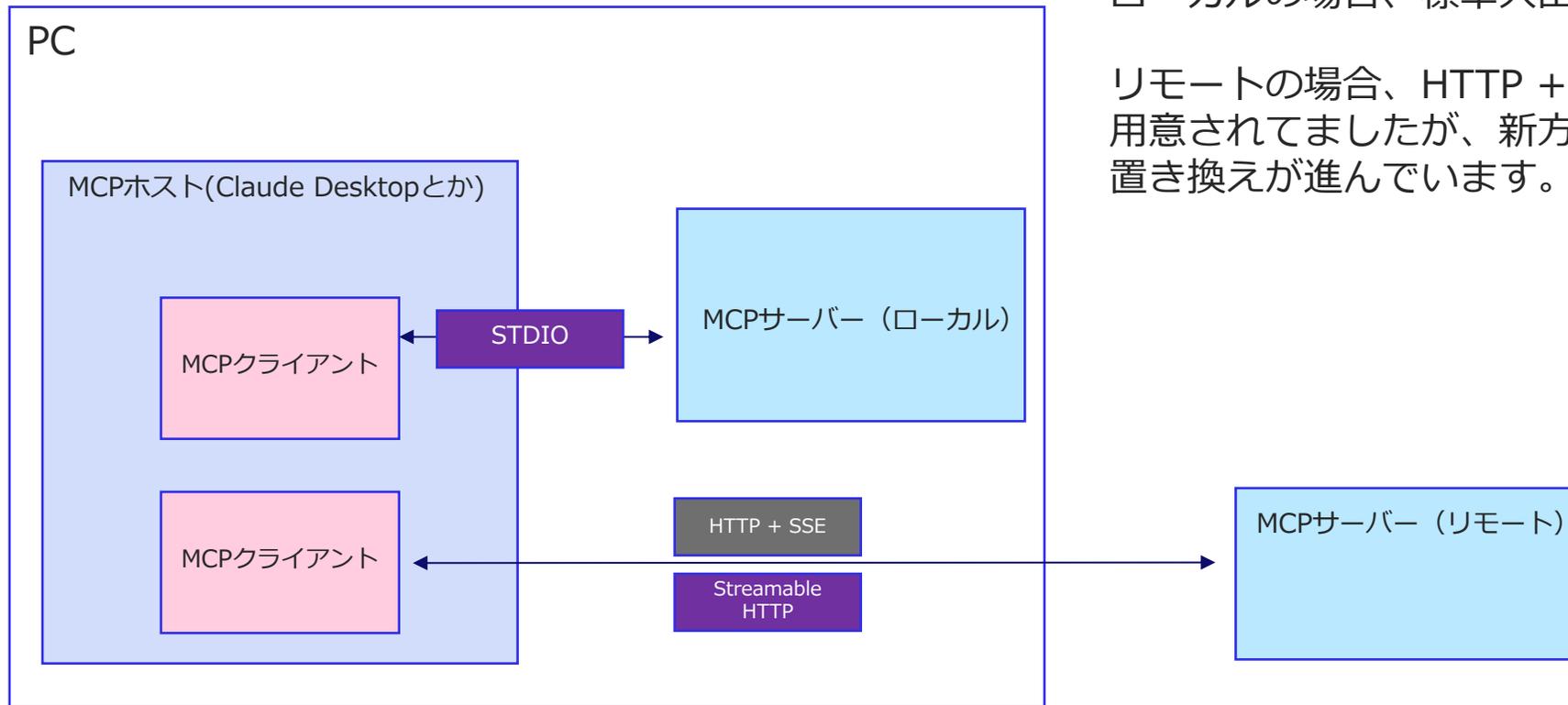
tools/listという利用できるツールの一覧をMCPサーバーが返すことで、 実際の質問の際に適切なツール呼び出し（そのパラメータ含め）ができる



MCPホストによってタイミングは異なるが、
事前にどのツールが使えるかLLMに渡している

通信方式

ローカルでMCPホストとサーバーと通信する方法が有名ですが、 リモートのサーバーと通信する方法もあります



ローカルの場合、標準入出力を使った通信。

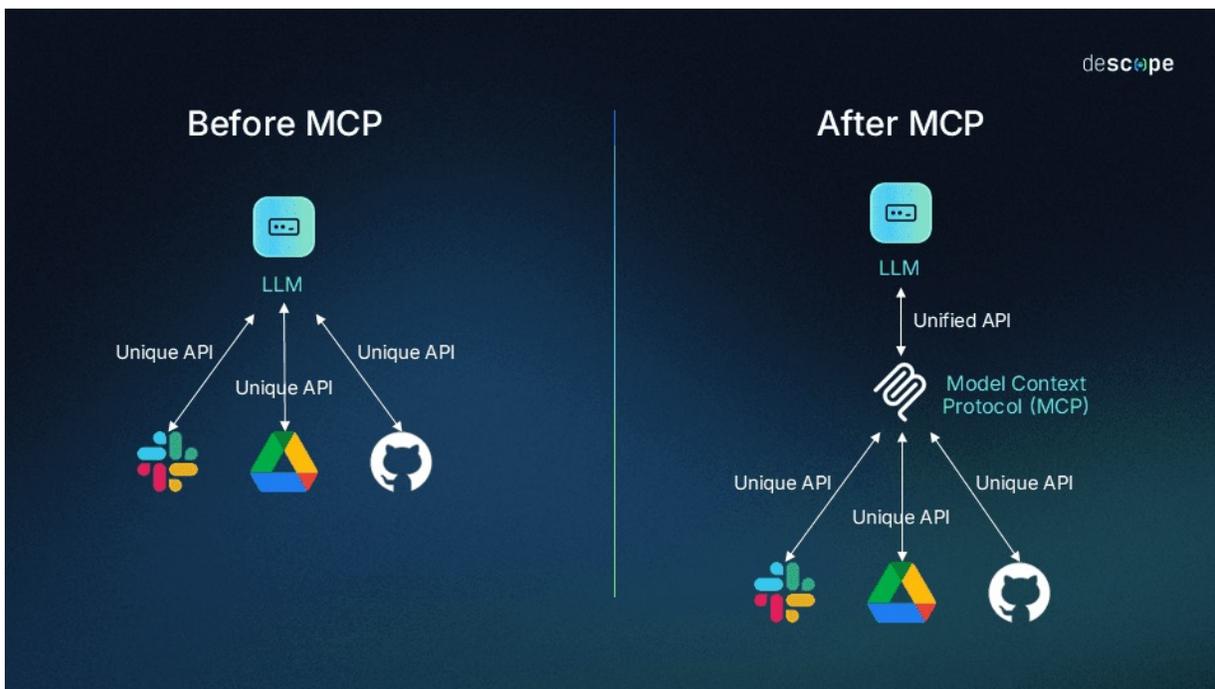
リモートの場合、HTTP + SSE(Server Sent Event)が用意されてましたが、新方式としてStreamable HTTPへの置き換えが進んでいます。

【TIPS】

SSEはLLMのストリーミングを実現する技術です

Streamable HTTPは、HTTPベースで動きつつ、必要に応じてSSEも使う、というものです

ローカルMCPサーバーはLanguage Server Protocolを想像してもらおうとイメージつくかもです。



ようは、様々なエディターで使える共通の補完機能サーバーとはいつつ、実体は、エディターと通信可能なローカルプロセス。

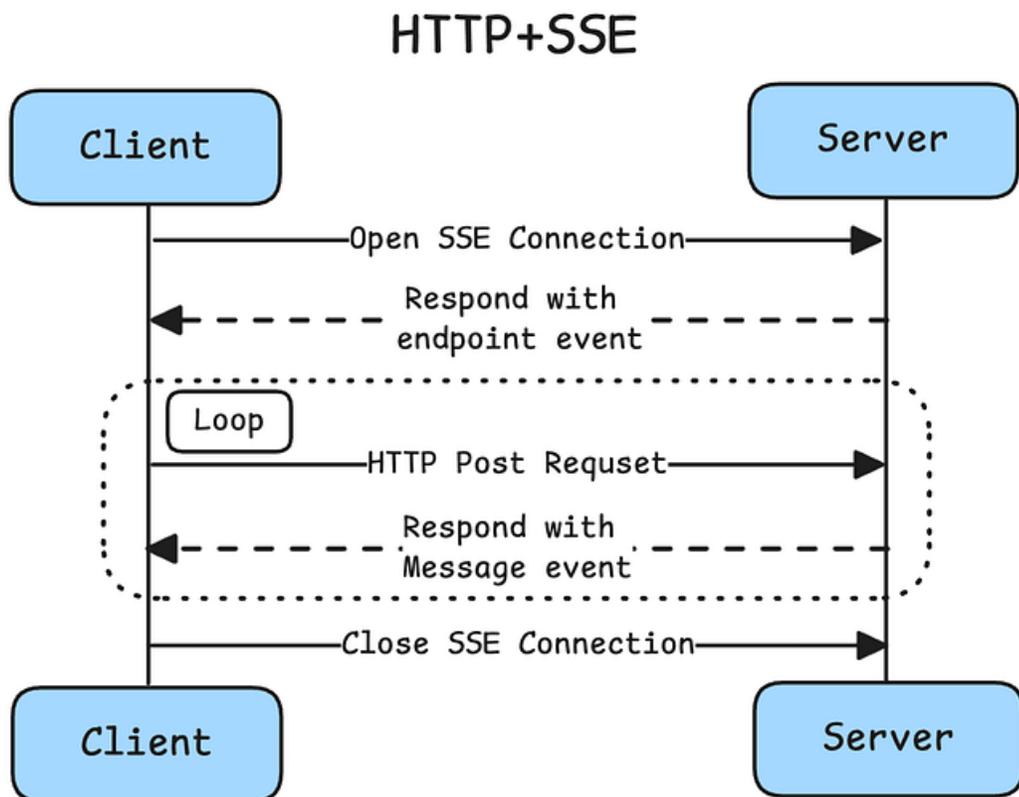
ローカルMCPサーバーもまさにこの仕組みに似ている。

SSEの場合はResponseが“text/event-stream”となる

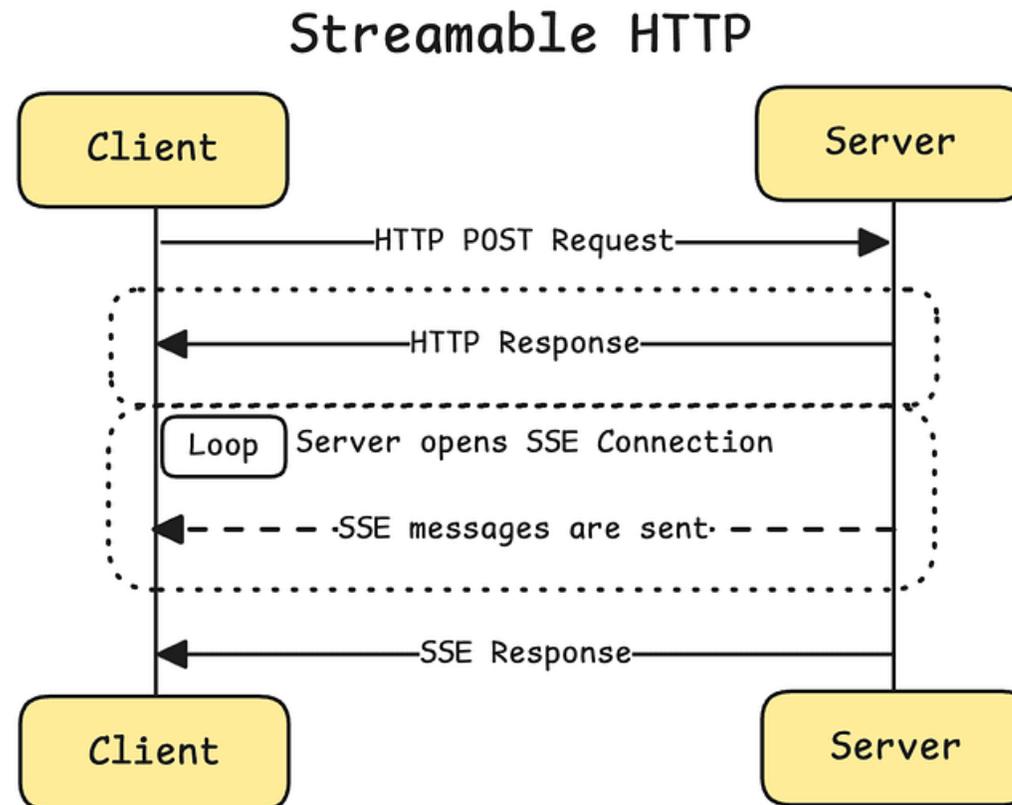
Streamable HTTPは基本がHTTPで場合によってはSSEも使える

SSEの課題

- 復帰がしにくい。(これはAIエージェントでも同じ)
- リソース効率が悪い



VS

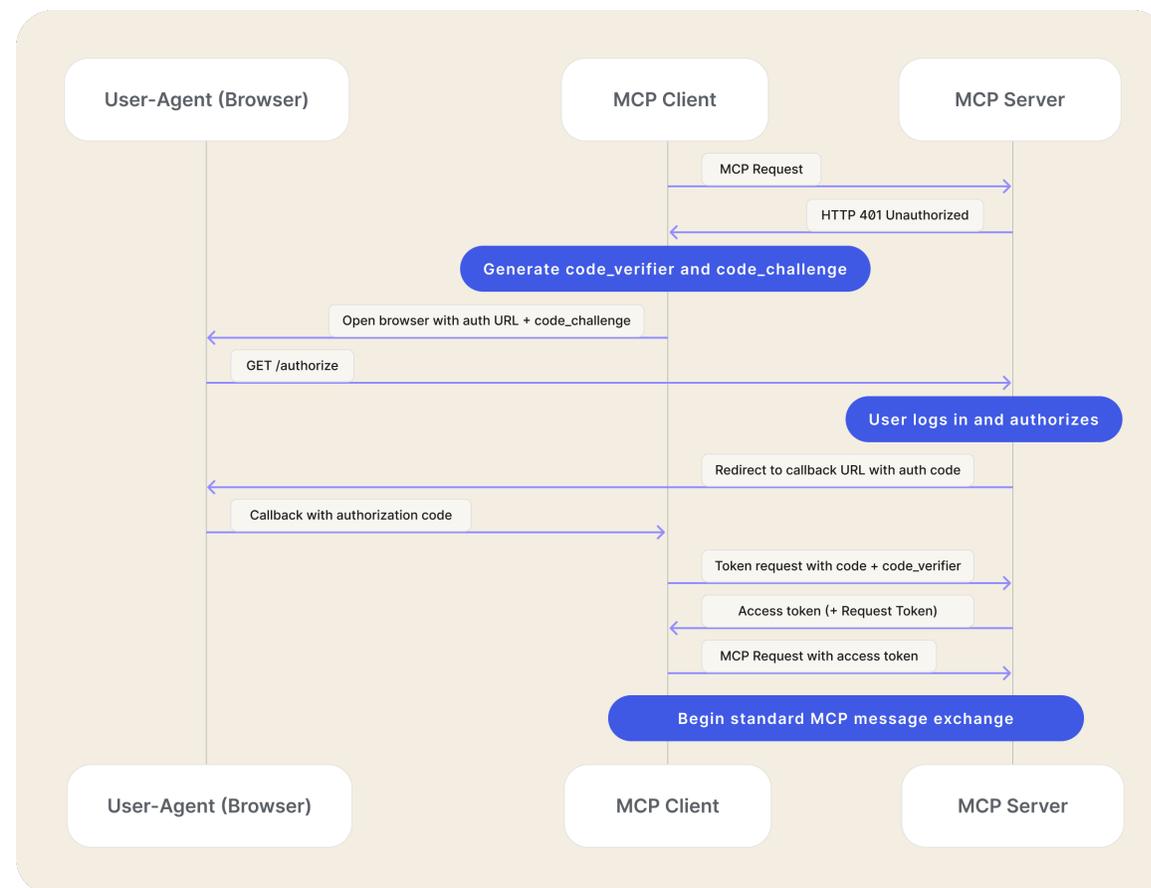


リモートMCPサーバーの場合、認証認可の仕組みが必要になります。 最近、OAuth2.1を使う規格が定まってきました。

【かいつまんでポイント】

- ユーザー認証・同意を求める画面が出てきます
 - GoogleとかのOauthを想像いただければ
- codeからアクセストークンを発行します
- 場合によってはリフレッシュトークンの発行も必要です
- これとは別に、MCPサーバー接続先のサービスの認証認可も取る必要があり、トークンのマッピングが必要になります

組み込みセキュリティベースライン(PKCE)



<https://auth0.com/blog/an-introduction-to-mcp-and-authorization/>

MCPサーバーを作るには？

Python・NodeであればFastMCPが便利ですが、公式から出ているSDKも使えます

なんの価値もないですが、テキストをこだまする
MCPサーバーならたった数行で実装できます。

公式のSDKで作ったサンプル↓

<https://github.com/tubone24/midi-mcp-server>

```
"""
FastMCP Echo Server
"""

from fastmcp import FastMCP

# Create server
mcp = FastMCP("Echo Server")

# create tool
@mcp.tool()
def echo(text: str) → str:
    """Echo the input text"""
    return text
```

もっと詳しく勉強したい場合

もっと詳しく勉強したい場合

- まずは公式Docを読み込むことをおすすめします！
 - <https://modelcontextprotocol.io/introduction>
- 自分で作ってみることも習得に繋がります。
 - MCPはあくまでも規格なので、利用しないような機能も多く定義されており、実装のほうがシンプルなパターンが多いです
 - ローカル実装がハードルが低いです
- インスペクターを試してみる
 - 実装したMCPサーバーのIN/OUTを見ることができるので、理解が進みます。

```
npx @modelcontextprotocol/inspector node build/index.js arg1 arg2
```
- schema.jsonを眺める
 - <https://github.com/modelcontextprotocol/modelcontextprotocol/blob/main/schema/2025-03-26/schema.json>
 - 一応MCPで定められている通信規格が全量定義されている



Be a Change Leader.

アジャイルに力を与え

共に成長し続ける社会を創る