MODEL CONTEXT PROTOCOL

(MCP)

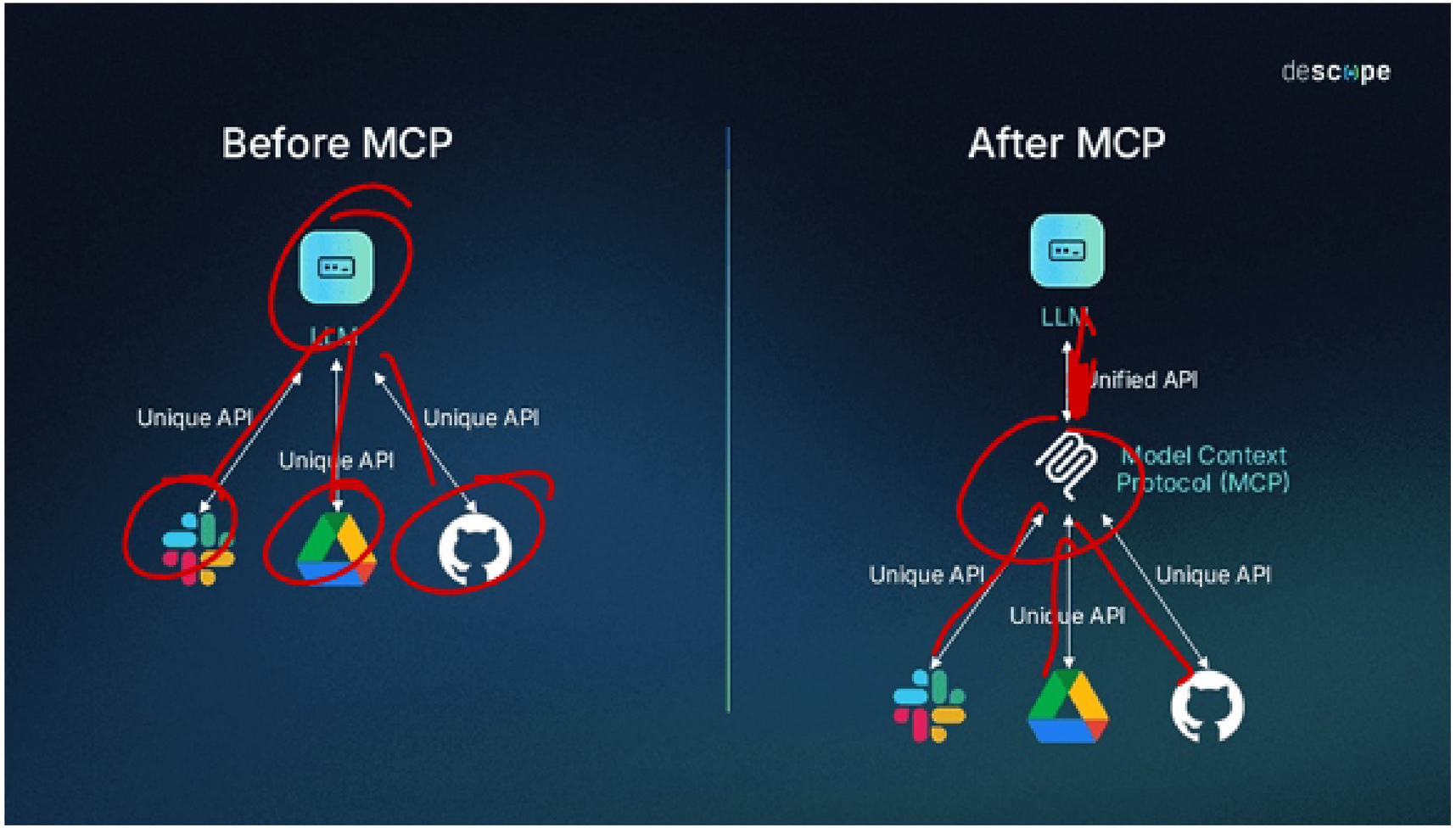
ĐỊNH NGHĨA

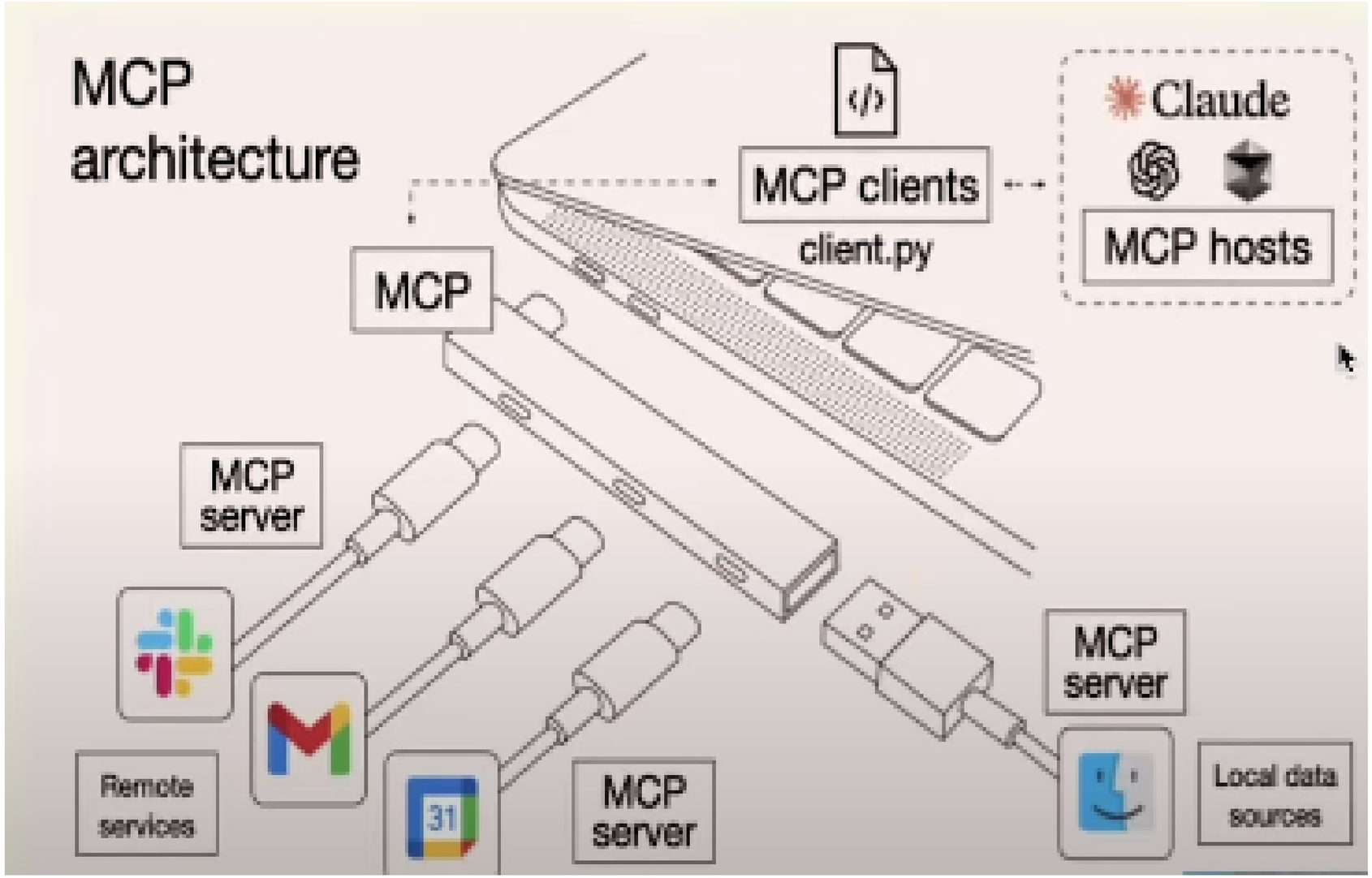
Là một tiêu chuẩn mã nguồn mở giúp AI kết nối và tương tác với các nguồn dữ liệu, công cụ và APi bên ngoài. Nó hoạt động như một "cổng USB" cho AI, giúp đơn giản hóa việc truy cập dữ liệu theo thời gian thực

MỤC ĐÍCH

Mục đích chính của MCP là:

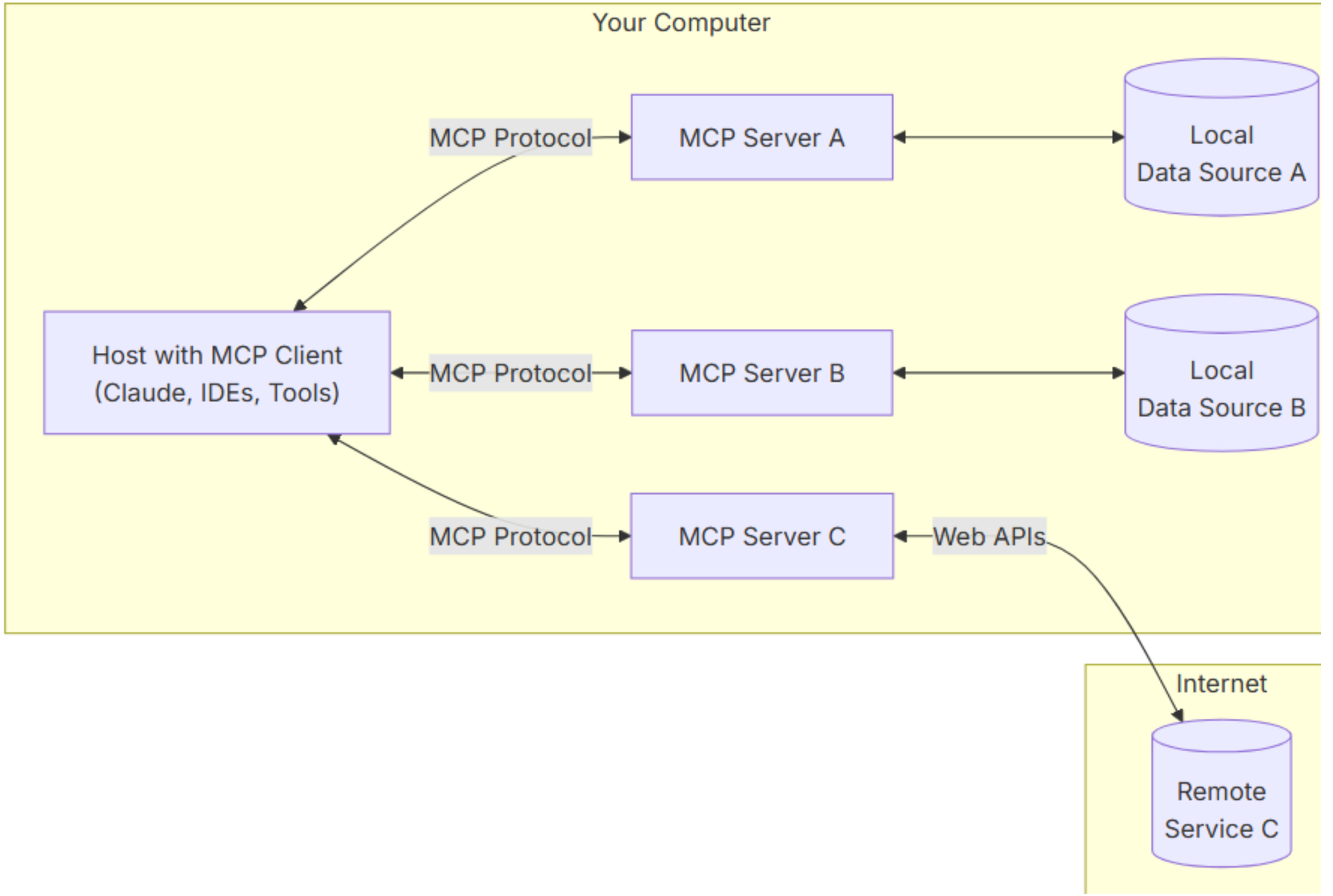
- Hỗ trợ xây dựng các agent AI và quy trình làm việc phức tạp trên LLM.
- Cung cấp danh sách tích hợp sẵn có, cho phép LLM dễ dàng kết nối với các công cụ và dữ liệu.
- Đảm bảo khả năng chuyển đổi giữa các nhà cung cấp LLM và công cụ, đồng thời áp dụng các thực hành tốt nhất để bảo mật dữ liệu trong hạ tầng.


descope
Before MCP
Unique API
Unique API
Unique API
After MCP
Unified API
Model Context Protocol (MCP)
Unique API
Unique API
Unique API


MCP
architecture
MCP clients
client.py
Claude
MCP hosts
MCP
MCP
server
MCP
server
MCP
server
Remote
services
Local data
sources

THÀNH PHẦN

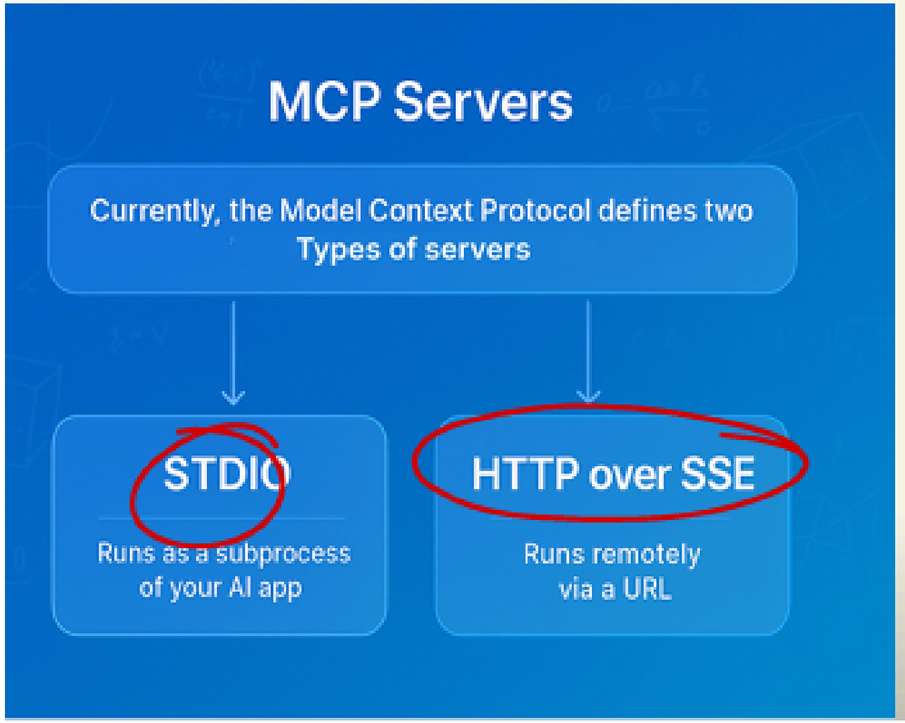
- Tools: Các hàm hoặc chức năng mà LLM có thể gọi để thực hiện hành động, được điều khiển bởi mô hình AI. Ví dụ: Gọi API thời tiết, tạo kho GitHub.
  - Resources: Các nguồn dữ liệu mà LLM có thể truy cập, giống như endpoint GET trong REST API, không có tác dụng phụ. Ví dụ: Truy cập tệp tin, cơ sở dữ liệu.
  - Prompts: Các mẫu tương tác được tối ưu hóa, giúp LLM hiểu rõ yêu cầu của người dùng, được điều khiển bởi người dùng. Ví dụ: Mẫu gợi ý cho code review.

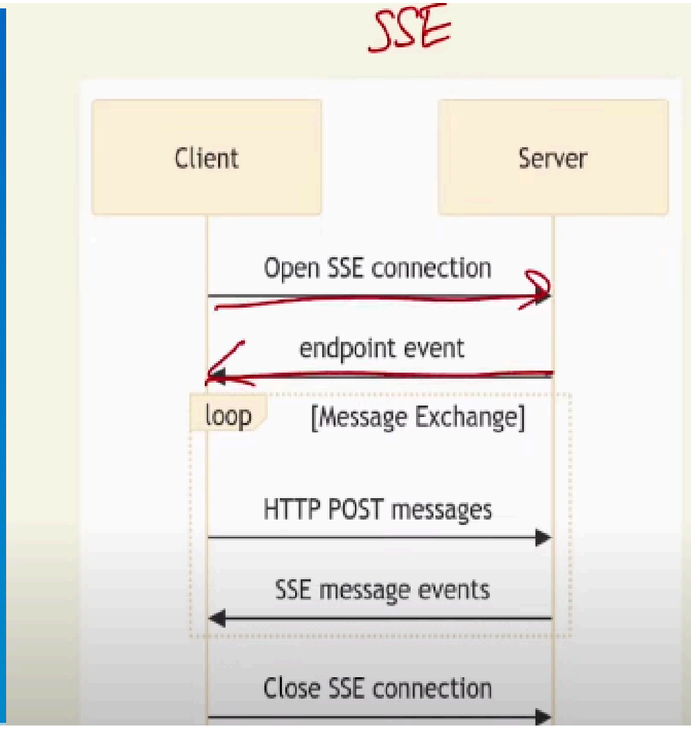

Your Computer
MCP Protocol
MCP Server A
Local
Data Source A
Host with MCP Client
(Claude, IDEs, Tools)
MCP Protocol
MCP Server B
Local
Data Source B
MCP Protocol
MCP Server C
Web APIs
Internet
Remote
Service C

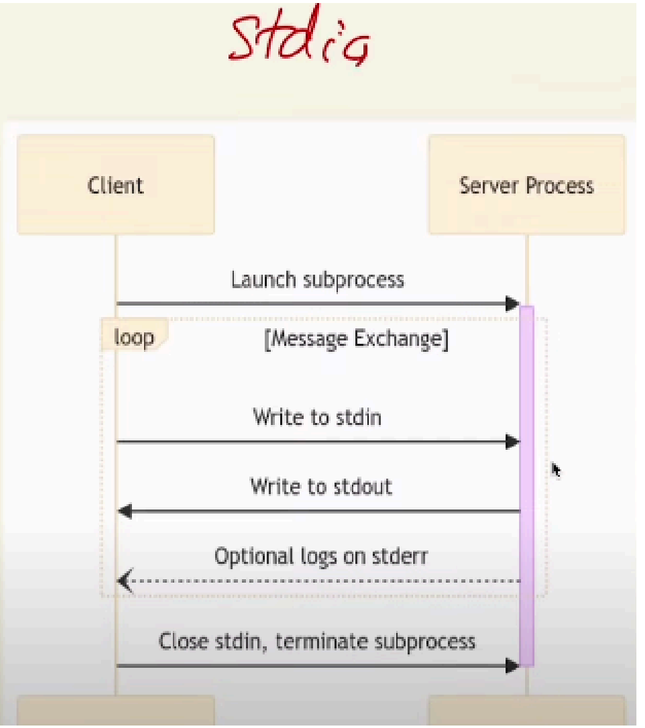
GIAO THỨC

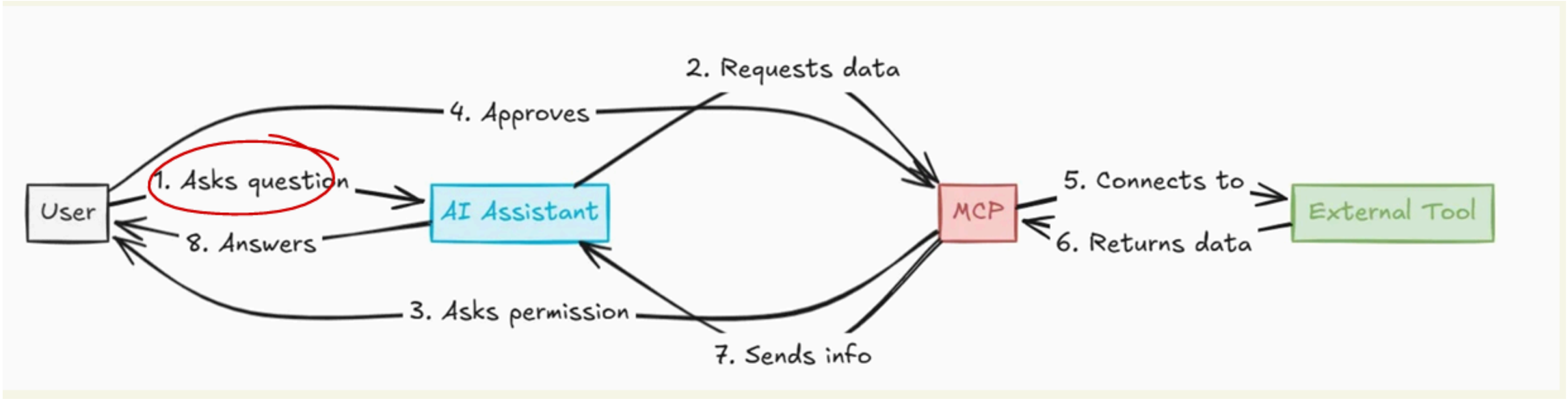
MCP có 2 giao thức

- Stdio(Standard IO): Dùng cho các tác vụ cục bộ trên cùng một máy tính, như truy cập tệp tin hoặc chạy script. Phương thức này phù hợp cho các server chạy trên cùng máy với ứng dụng AI.
- SSE(Sever send event): Dùng cho kết nối từ xa, cho phép giao tiếp thời gian thực từ server đến client, chẳng hạn khi theo dõi thay đổi tệp tin trên Google Drive.


MCP Servers
Currently, the Model Context Protocol defines two Types of servers
STDIO
Runs as a subprocess of your AI app
HTTP over SSE
Runs remotely via a URL


SSE
Client
Server
Open SSE connection
endpoint event
loop
[Message Exchange]
HTTP POST messages
SSE message events
Close SSE connection


Stdio
Client
Server Process
Launch subprocess
loop
[Message Exchange]
Write to stdin
Write to stdout
Optional logs on stderr
Close stdin, terminate subprocess


2. Requests data
4. Approves
User
1. Asks question
AI Assistant
MCP
5. Connects to
External Tool
8. Answers
6. Returns data
3. Asks permission
7. Sends info