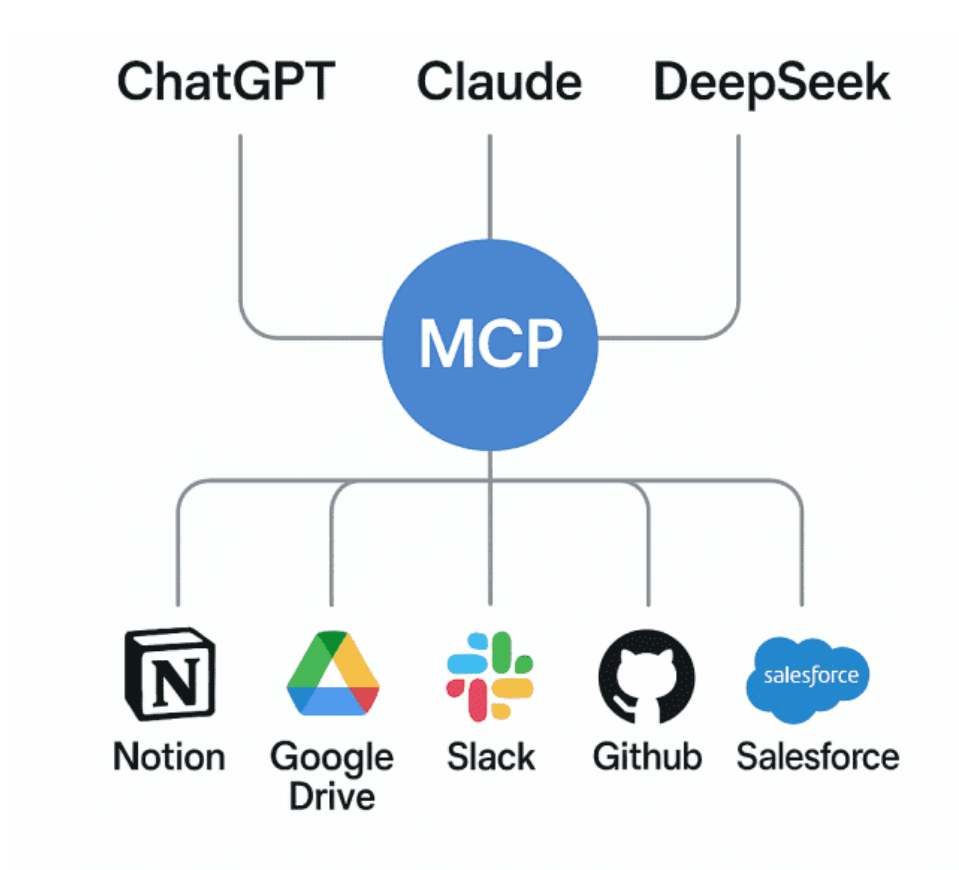




Model Context Protocol



MCP 개요

MCP(Model Context Protocol)은 LLM(Large Language Model) 애플리케이션과 외부 데이터 소스 및 도구들 간의 원활한 통합을 가능하게 하는 JSON-RPC 기반 개방형 프로토콜

■ 컨텍스트 공유 표준화

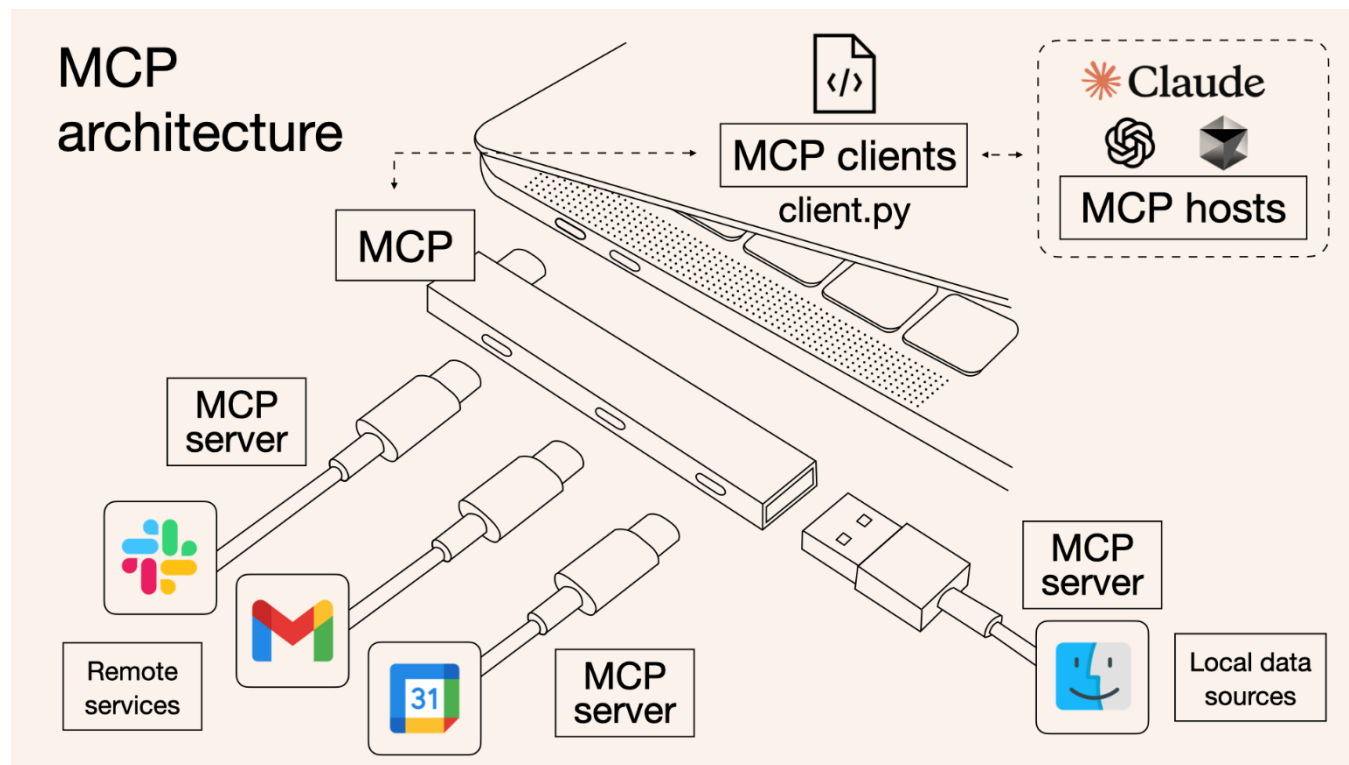
- LLM 애플리케이션과 데이터 소스 간의 상호작용을 위한 표준 프로토콜 제공
- 구조화된 방식으로 컨텍스트 정보를 전달하고 관리

■ 도구와 기능 노출

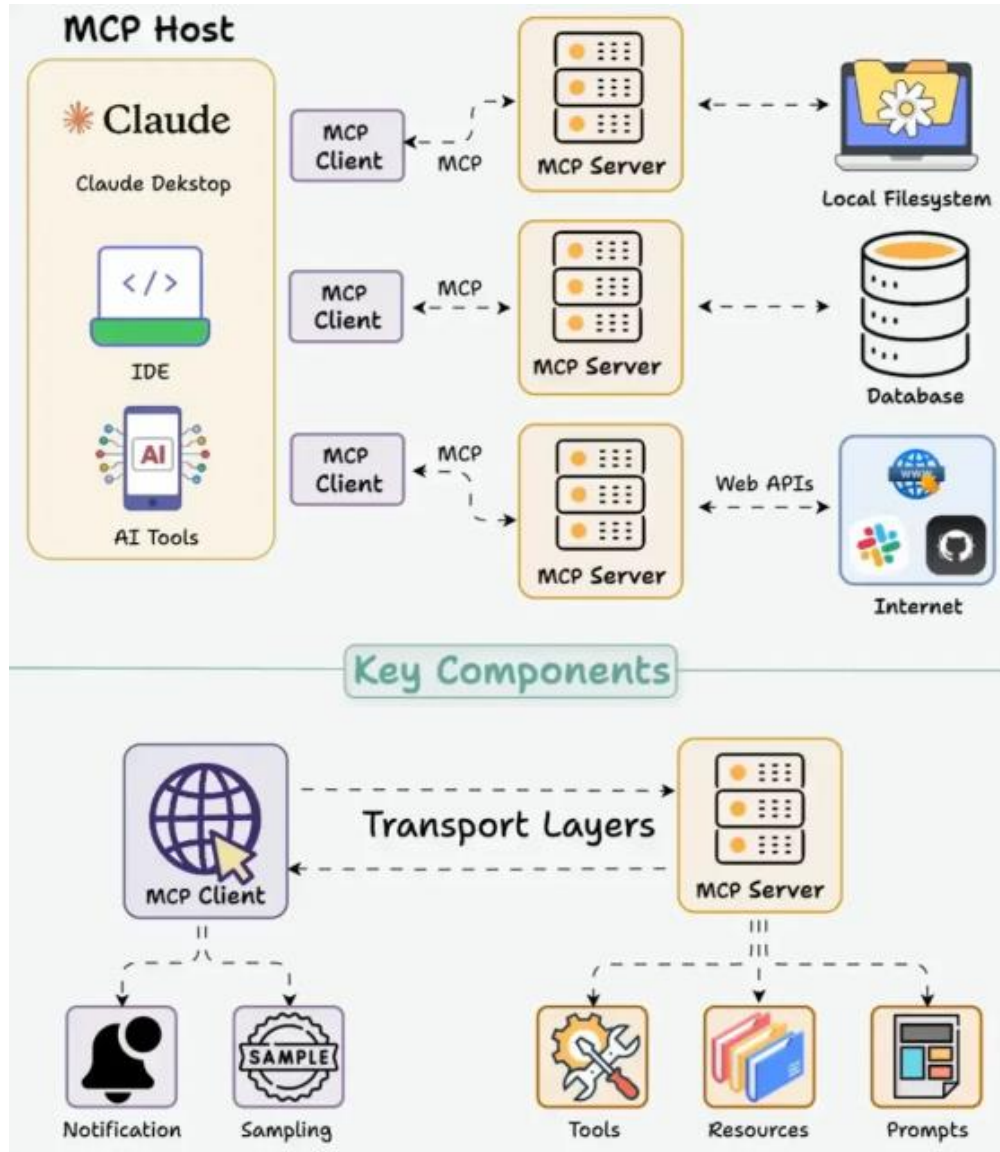
- AI 시스템에 로컬 또는 원격 도구들을 안전하게 노출
- 표준화된 방식으로 기능을 정의하고 호출

■ 통합 워크플로우 구축

- 여러 데이터 소스와 도구를 조합한 워크플로우 생성
- 재사용 가능한 프롬프트 템플릿 제공



MCP 구조



- MCP는 클라이언트-서버 모델을 기반으로 하며, AI 시스템이 다양한 데이터 소스와 연결될 수 있도록 설계된 프로토콜
- MCP Host : MCP 실행 환경
 - Claude Desktop: AI 모델이 실행되는 환경
 - IDE : VSCode, Cursor 같은 개발 환경
 - AI Tools: MCP와 통합된 다양한 AI 애플리케이션
- MCP Client : MCP Server에 연결되어 데이터를 요청하고 응답을 받음
- MCP Server : 다양한 데이터 소스(로컬 파일시스템, 데이터베이스, API 등)와 직접 연결됨. 클라이언트의 요청을 받아 필요한 데이터를 전달하거나 AI의 명령을 실행함
- MCP Server 제공 기능
 - Tools : AI가 실행할 수 있는 기능 (예: GitHub PR 생성, 코드 리뷰)
 - Resources : 데이터베이스, 파일 시스템 등의 정보
 - Prompts : 재사용 가능한 프롬프트 템플릿
 - Notification : 특정 작업이 완료되었을 때 클라이언트에 알림
 - Sampling : AI가 데이터를 효율적으로 활용하기 위한 샘플링 기능

MCP 메시지 유형과 전송 메커니즘

요청 메시지

```
{
  "jsonrpc": "2.0",
  "id": "string | number",
  "method": "string",
  "param?": {
    "key": "value"
  }
}
```

응답 메시지

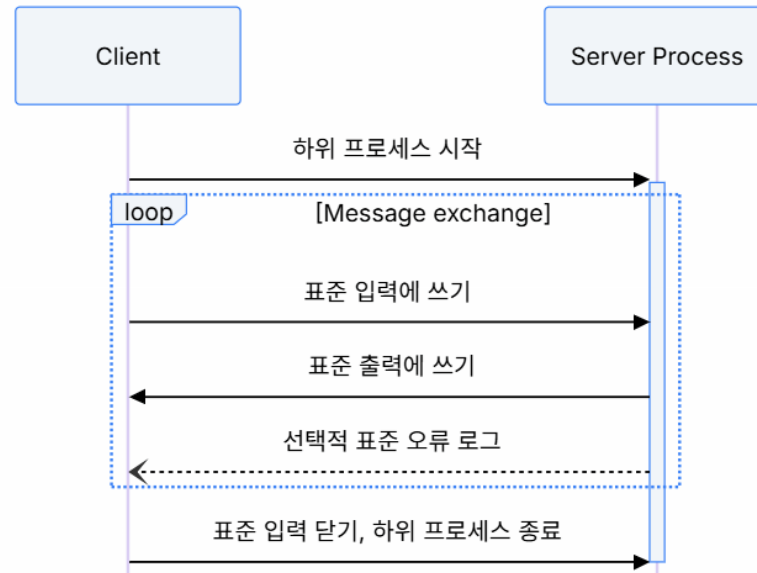
```
{
  "jsonrpc": "2.0",
  "id": "string | number",
  "result?": {
    "[key: string]": "unknown"
  },
  "error?": {
    "code": "number",
    "message": "string",
    "data?": "unknown"
  }
}
```

알림 메시지

```
{
  "jsonrpc": "2.0",
  "method": "string",
  "params?": {
    "[key: string]": "unknown"
  }
}
```

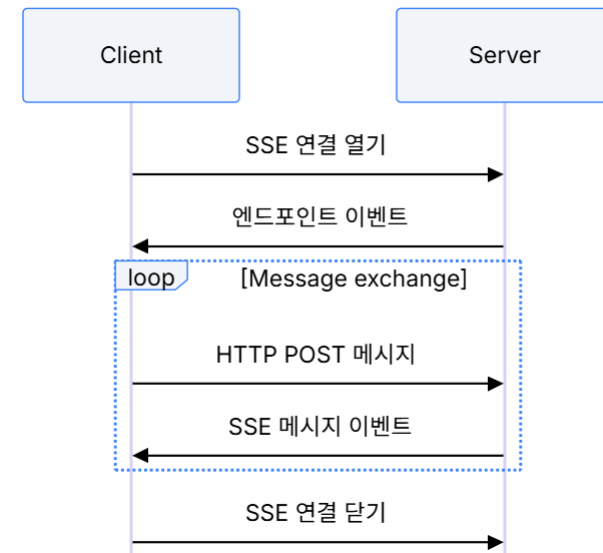
Stdio 전송 메커니즘

클라이언트는 stdio를 가능한 한 지원해야 함



SSE(Server-Sent Events) 전송 메커니즘

서버가 실시간 업데이트를 클라이언트에 전송할 수 있는 HTTP 서버 푸시 기술



서버는 2 개의 엔드포인트를 제공해야 합니다.

- SSE 엔드포인트 - 연결을 설정하고 서버에서 메시지를 수신하는 데 사용
- HTTP POST 엔드포인트 - 클라이언트에서 서버로 메시지를 보내는 데 사용

MCP 장단점

■ 장점

- 데이터 연결 단순화 : AI 애플리케이션과 데이터 소스 간의 연결을 단순화
- 향상된 데이터 상호작용 보안 : 표준화된 프로토콜을 통해 데이터 흐름을 더 잘 제어하고 데이터 유출을 방지
- 향상된 AI 애플리케이션 기능 : AI 애플리케이션의 기능 범위를 확장하고 더 많은 비즈니스 시나리오에서 역할을 수행할 수 있게 함
- 우수한 확장성 : MCP 프로토콜은 Prompts, Tools, Sampling과 같은 기능을 제공하여 우수한 확장성을 갖추고 있음
- 다양한 데이터 형식 지원 : 파일 내용, 데이터베이스 레코드, API 응답, 실시간 시스템 데이터, 스크린샷 및 이미지, 로그 파일 등 다양한 데이터 형식을 지원

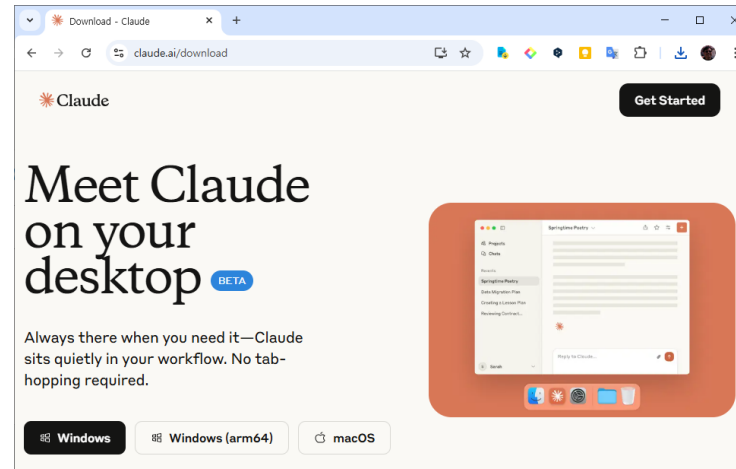
■ 단점

- 높은 산업 표준 경쟁 압력
- 잠재적 호환성 문제
- 프로토콜 본질에 대한 의문
- 현재 제한된 응용 범위 : 여러 기기나 분산 환경에서 데이터 상호작용이 필요한 일부 응용 시나리오의 경우, 현재 MCP 프로토콜이 요구 사항을 충족시키지 못할 수 있음

MCP 실습

<https://www.claudemcp.com/ko/docs/quickstart>

- Claude 데스크톱 앱 설치
<https://claude.ai/download>



- Claude 데스크톱의 설정 파일 열기 : 설정 → 개발자 → 설정 편집
- claude_desktop_config.json 에 서버 정보 추가

```
{
  "mcpServers": {
    "sqlite": {
      "command": "uvx",
      "args": ["mcp-server-sqlite", "--db-path", "/Users/Administrator/test.db"]
    }
  }
}
```

MCP 실습 - 데이터베이스 자연어 조회

<https://www.claudemcp.com/ko/docs/quickstart>



- DB Browser for SQLite 설치
<https://sqlitebrowser.org/dl/>
- test DB 생성
- products 테이블 생성

```
DB Browser for SQLite - C:\Program Files\DB Browser for SQLite\test.db
파일(F) 편집(E) 보기(V) 도구(T) 도움말(H)
새 데이터베이스(N) 데이터베이스 열기(O) 변경사항 저장하기
데이터베이스 구조 데이터 보기 Pragma 수정 SQL 실행
SQL 1
1 CREATE TABLE products (
2   id INTEGER PRIMARY KEY,
3   name TEXT,
4   price REAL
5 );
6
```

- products 테이블에 데이터 추가

```
INSERT INTO products (name, price) VALUES
('Widget', 19.99),
('Gadget', 29.99),
('Gizmo', 39.99),
('Smart Watch', 199.99),
('Wireless Earbuds', 89.99),
('Portable Charger', 24.99),
('Bluetooth Speaker', 79.99),
('Phone Stand', 15.99),
('Laptop Sleeve', 34.99),
('Mini Drone', 299.99),
('LED Desk Lamp', 45.99),
('Keyboard', 129.99),
('Mouse Pad', 12.99),
('USB Hub', 49.99),
('Webcam', 69.99),
('Screen Protector', 9.99),
('Travel Adapter', 27.99),
('Gaming Headset', 159.99),
('Fitness Tracker', 119.99),
('Portable SSD', 179.99);
```

MCP 실습 - 구글맵 MCP 서버로 길찾기

■ claude_desktop_config.json 에 서버 정보 추가

```
{
  "mcpServers": {
    "filesystem": {
      "command": "npx",
      "args": [
        "-y",
        "@modelcontextprotocol/server-filesystem",
        "/Users/사용자명/Desktop",
        "/Users/사용자명/aiagent"
      ]
    },
    "google-maps": {
      "command": "npx",
      "args": [
        "-y",
        "@modelcontextprotocol/server-google-maps"
      ],
      "env": {
        "GOOGLE_MAPS_API_KEY": "YOUR_GOOGLE_MAPS_API_KEY"
      }
    },
    "tavily-mcp": {
      "command": "npx",
      "args": ["-y", "tavily-mcp@0.1.2"],
      "env": {
        "TAVILY_API_KEY": "YOUR_TAVILY_API_KEY"
      }
    },
    "sequential-thinking": {
      "command": "npx",
      "args": [
        "-y",
        "@modelcontextprotocol/server-sequential-thinking"
      ]
    },
    "mcp-youtube": {
      "command": "npx",
      "args": [
        "@anaisbetts/mcp-youtube"
      ]
    }
  }
}
```

Directions from Bundang Sunae Middle School to Sunae Station

DP 분당 수내중학교에서 분당선 수내역까지 어떻게 가?

분당 수내중학교에서 분당선 수내역까지 가는 방법을 알려드리겠습니다. 정확한 경로를 확인하기 위해 지도 검색을 해보겠습니다.

google-maps의 maps_directions에서 결과 보기(로컬) >

주소를 더 정확하게 검색해 보겠습니다.

google-maps의 maps_geocode에서 결과 보기(로컬) >

google-maps의 maps_geocode에서 결과 보기(로컬) >

google-maps의 maps_directions에서 결과 보기(로컬) >

정확한 위치 정보를 검색해 보겠습니다.

google-maps의 maps_search_places에서 결과 보기(로컬) >

google-maps의 maps_search_places에서 결과 보기(로컬) >

google-maps의 maps_directions에서 결과 보기(로컬) >

정확한 좌표를 사용해 검색해 보겠습니다.

google-maps의 maps_directions에서 결과 보기(로컬) >

다른 방식으로 검색해보겠습니다. 대중교통 옵션으로 시도해보겠습니다.

google-maps의 maps_directions에서 결과 보기(로컬) >

수내중학교에서 분당선 수내역까지 가는 방법을 알려드립니다:

총 거리는 약 1.6km이며, 소요 시간은 약 25분입니다.

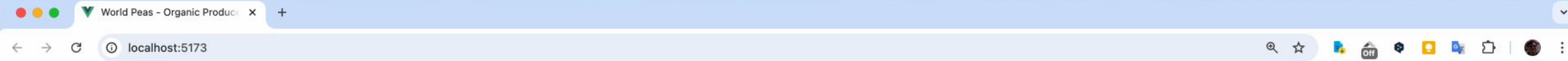
- 수내중학교에서 출발하여 푸른마을관리사무소 방향으로 도보로 이동합니다 (약 3분, 0.2km)
- 롯데백화점 방향 버스를 탑승합니다 (약 7분, 1.4km)
- 하차 후 수내역까지 도보로 이동합니다 (약 1분, 42m)

만약 버스를 이용하지 않고 도보로만 이동하고 싶으시다면, 약 1.6km 거리이므로 약 20-25분 정도 소요될 것으로 예상됩니다. 수내중학교에서 분당선 수내역 방향으로 걸어가시면 됩니다.

+ ↺ ↻ 22 ✕

Claude 3.7 Sonnet

↑



World Peas

[Shop](#)[Newstand](#)[Who we are](#)[My profile](#)[BASKET \(3\)](#)

WHAT WE BELIEVE

We believe in produce. Tasty produce. Produce like:

Apples. Oranges. Limes. Lemons. Guavas. Carrots. Cucumbers. Jicamas. Cauliflowers. Brussels sprouts. Shallots. Japanese eggplants. Asparagus. Artichokes—Jerusalem artichokes, too. Radishes. Broccoli. Baby broccoli. Broccolini. Bok choy. Scallions. Ginger. Cherries. Raspberries. Cilantro. Parsley. Dill.

What are we forgetting?

Oh! Onions. Yams. Avocados. Lettuce. Arugula (to some, "rocket"). Persian cucumbers, in addition to aforementioned "normal" cucumbers. Artichokes. Zucchini. Pumpkins. Squash (what some cultures call pumpkins). Sweet potatoes and potato-potatoes. Jackfruit. Monk fruit. Fruit of the Loom. Fruits of our labor (this website). Sorrel. Pineapple. Mango. Gooseberries. Blackberries. Tomatoes. Heirloom tomatoes. Beets. Chives. Corn. Endive. Escarole, which, we swear, we're vendors of organic produce, but if you asked us to describe what escaroles are...



Central California — The person who grew these was located in Central California and, er, hopefully very well-compensated.

- 개요
- 설치
 - 파이썬 프로젝트에 MCP 추가하기
 - 독립형 MCP 개발 도구 실행
- 빠른 시작
- MCP란 무엇인가요?
- 핵심 개념
 - Server
 - Resources
 - Tools
 - Prompts
 - Images
 - Context
- 서버 실행
 - 개발 모드
 - 클로드 데스크톱 통합
 - 직접 실행
 - 기존 ASGI 서버에 마운팅
- 예시
 - 에코 서버
 - SQLite 탐색기
- 고급 사용
 - 저수준 서버
 - MCP 클라이언트 작성
 - MCP Primitives
 - Server Capabilities

MCP 서버 개발

<https://www.devshorts.in/p/how-to-build-your-own-mcp-server>

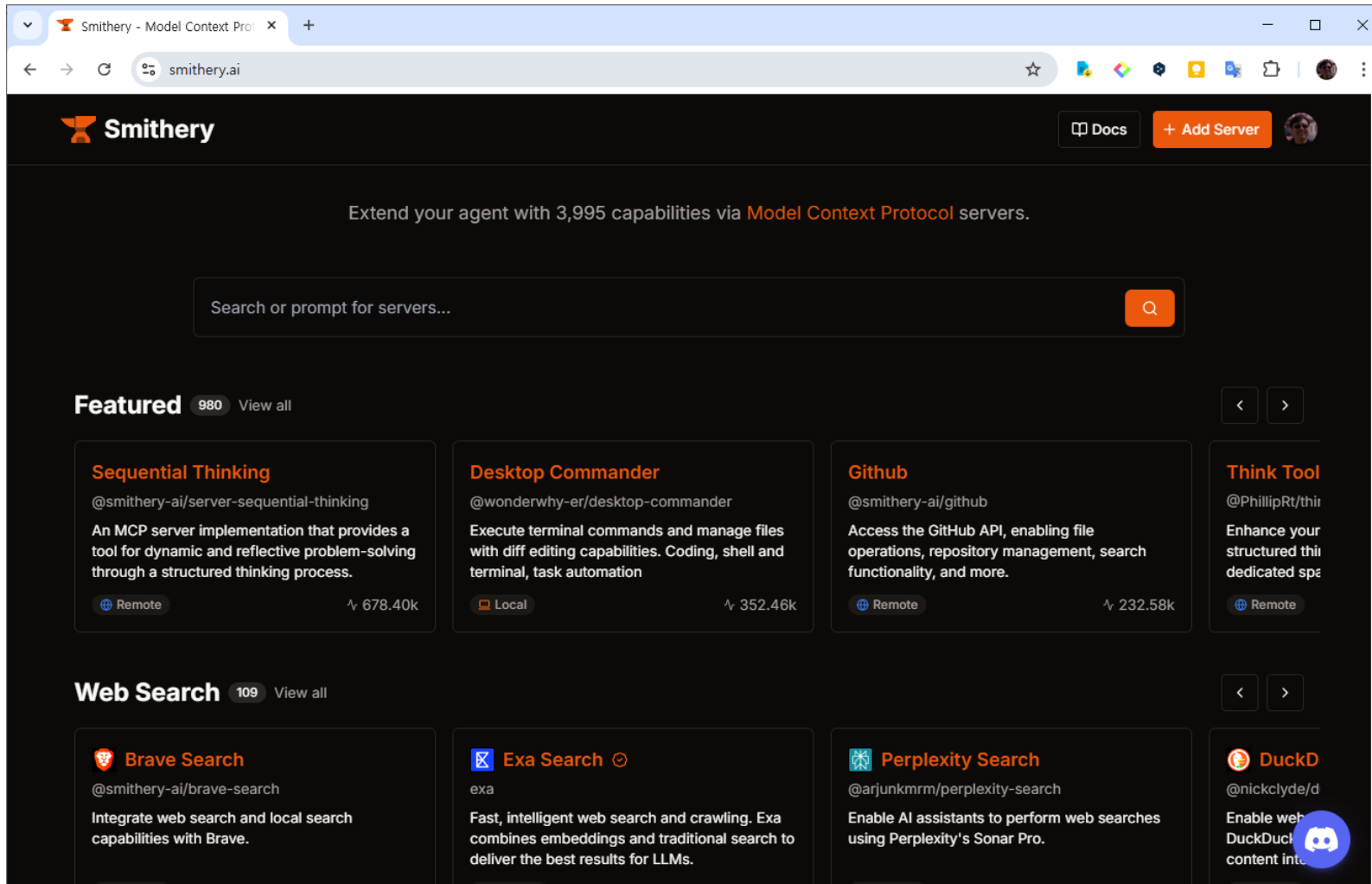
```
1 from mcp.server.fastmcp import FastMCP
2 from google.oauth2.credentials import Credentials
3 from google_auth_oauthlib.flow import InstalledAppFlow
4 from google.auth.transport.requests import Request
5 from googleapiclient.discovery import build
6 from email.mime.text import MIMEText
7 import base64
8 import os
9 import pickle
10 import logging
11 import time
12 import datetime
13
14 logging.basicConfig(level=logging.INFO)
15 logger = logging.getLogger("gmail_mcp_server")
16
17 # Combined scopes for both Gmail and Calendar
18 SCOPES = [
19     'https://www.googleapis.com/auth/gmail.modify',
20     'https://www.googleapis.com/auth/calendar'
21 ]
22
23 mcp = FastMCP("gmail_mcp_server")
```

```
51 @mcp.tool()
52 def send_email(to: str, subject: str, body: str) -> dict:
53     """Send an email using Gmail"""
54     try:
55         service = get_gmail_service()
56         message = MIMEText(body)
57         message['to'] = to
58         message['subject'] = subject
59         raw = base64.urlsafe_b64encode(message.as_bytes()).decode()
60
61         result = service.users().messages().send(
62             userId='me',
63             body={'raw': raw}
64         ).execute()
65
66         return {"result": f"Email sent successfully (Message ID: {result['id']})"}
67
68 except Exception as e:
69     logger.error(f"Send email failed: {e}")
70     return {"error": str(e)}
```

MCP 서버 Registry

<https://smithery.ai/>

Smithery는 MCP 서버의 중앙 HUB 역할을 하며, MCP 서버를 쉽게 검색하고 활용할 수 있도록 지원합니다



1 MCP 서버 검색 및 관리

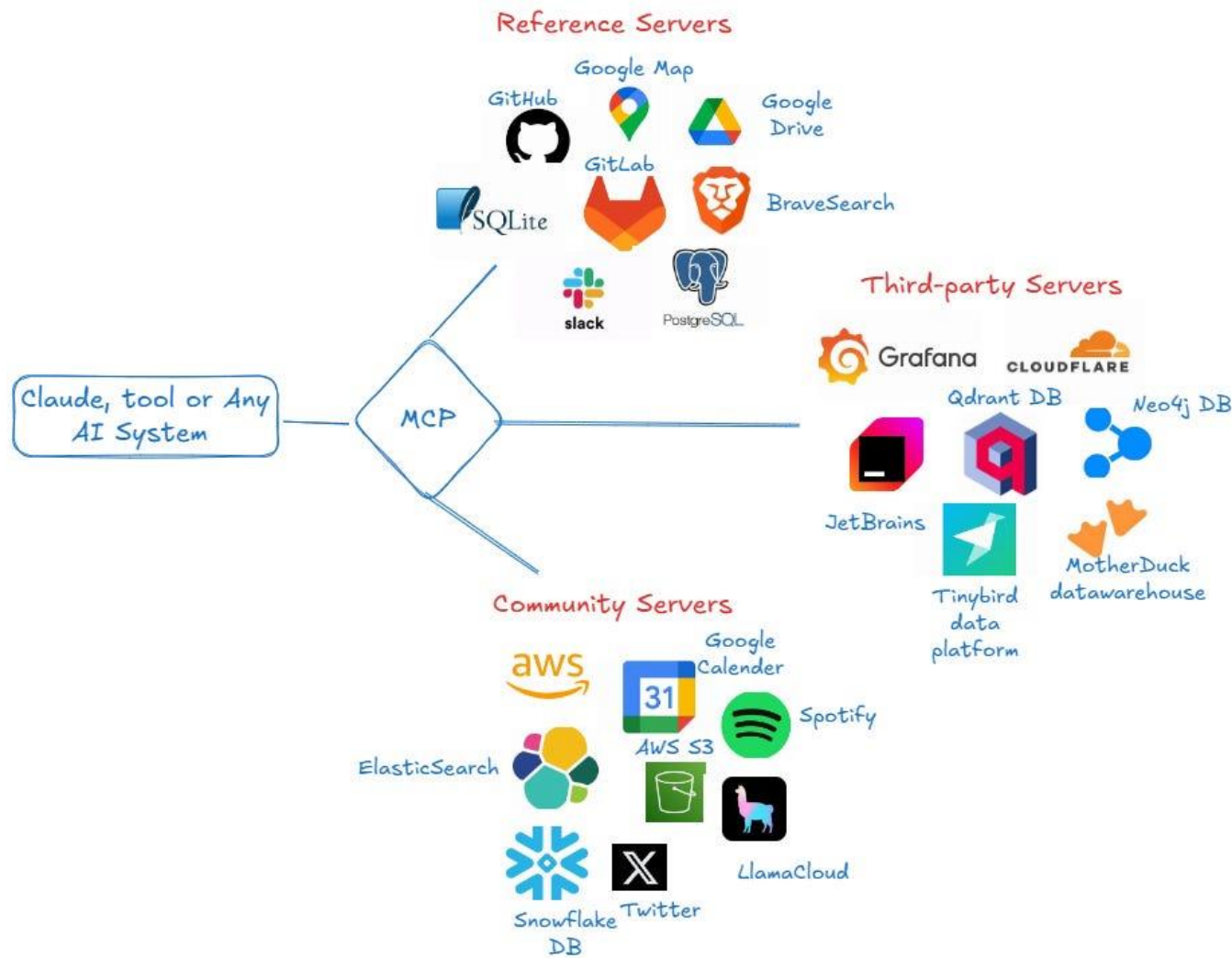
2 MCP 서버 호스팅 및 배포

3 표준화된 인터페이스 제공

MCP 서버 Repository

<https://github.com/modelcontextprotocol/servers>

사전 구축된 MCP 서버는 GitHub, Cloudflare, Slack과 같은 주요 플랫폼에서 사용할 수 있습니다.



Thank you 😊