

.NET에서의 MCP 활용 가이드

이종인
DesignMe AI / Microsoft MVP



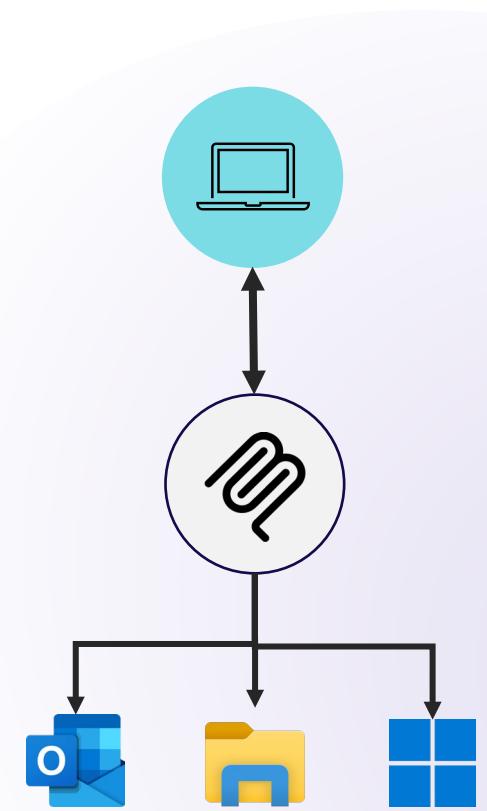
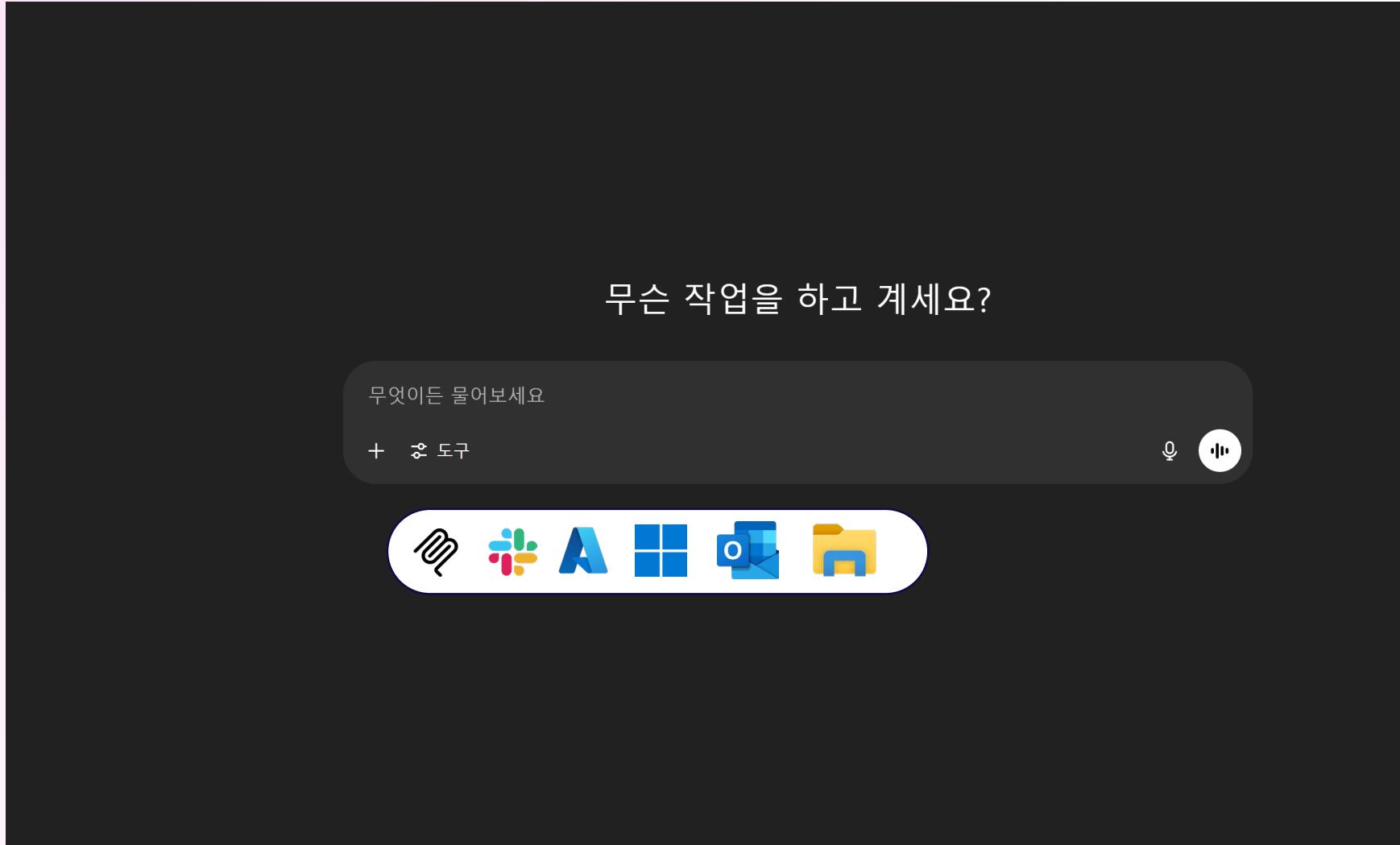
저는요!

이종인

- DesignMe AI
- Microsoft MVP | Developer Technologies
- jongin@designme.ai



폴더에서 어제 회의록을 요약해서 메일을 쓰고 아웃룩으로 보내줘.



Azure AI Foundry MCP 서버



오픈 AI 모델을 배포해줄 수 있니?

Azure AI Foundry에 Open AI 모델을 배포하려면 어떠한 스텝을 거쳐야 하니?

내가 사용할 수 있는 모델은 어떠한 것들이 있니?

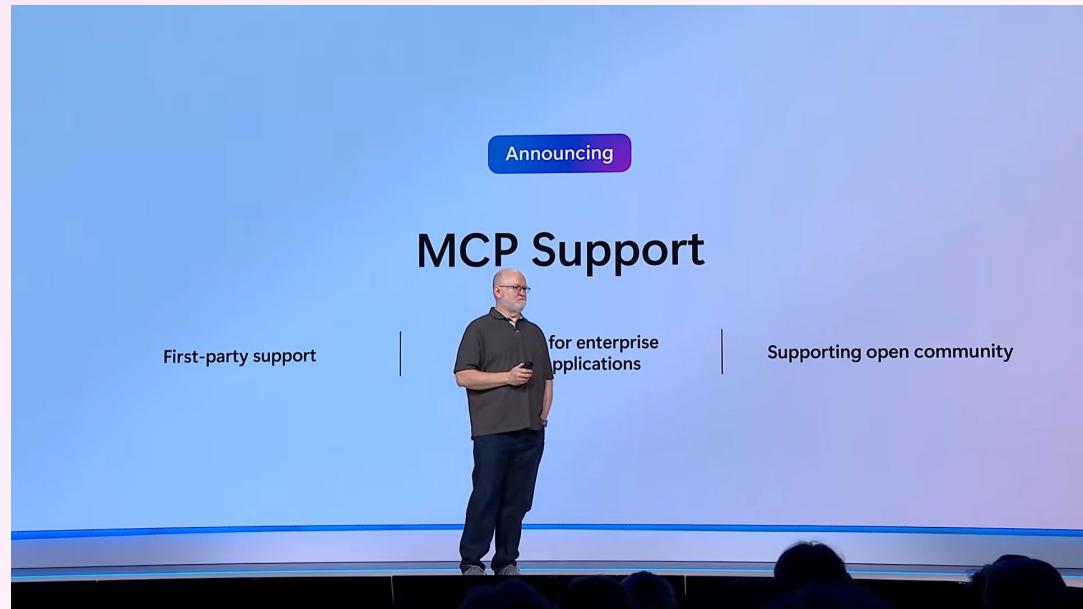
...



Open AI 4o 모델을 배포하고 사용할 수 있는 코드를 작성해줘.

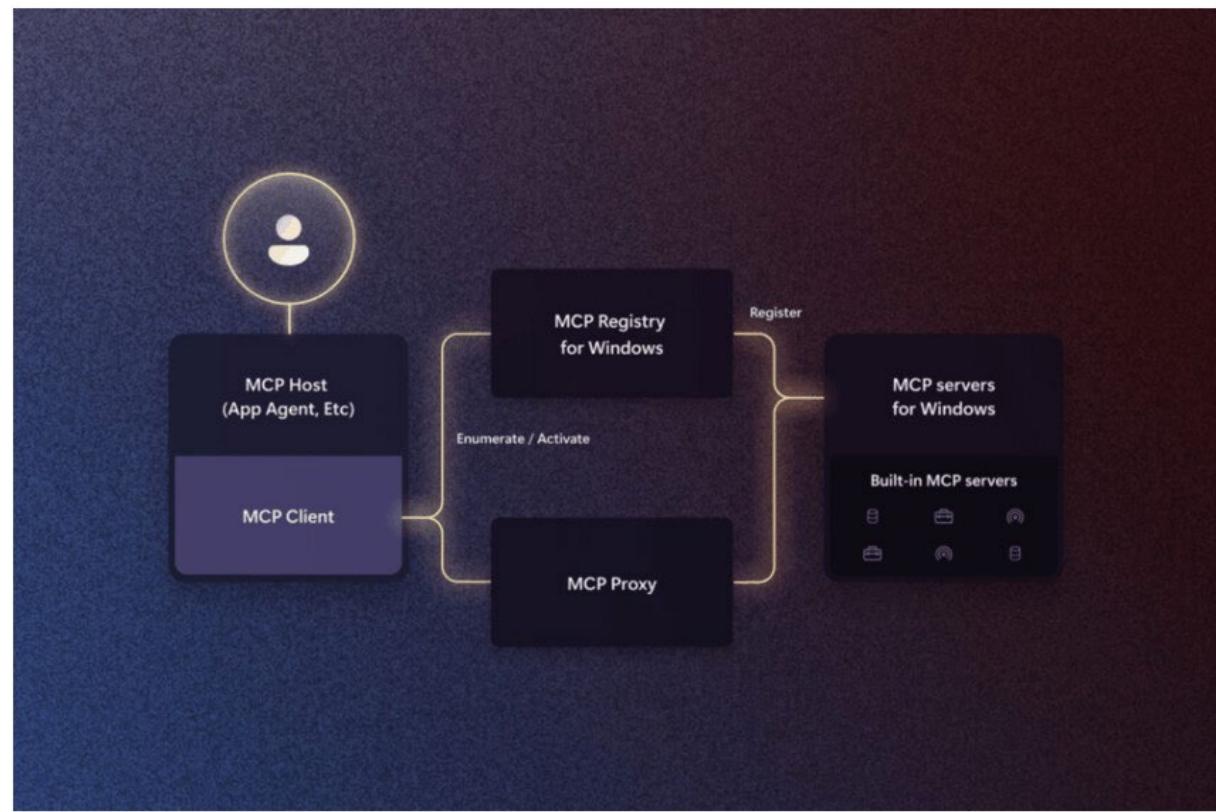
- Open AI 4o를 생성중
- 코드를 생성 중

Azure 및 Microsoft 플랫폼과의 통합



- MCP를 Microsoft에서 전사적으로 지원하겠다는 계획을 발표
- Azure, Windows, GitHub, Dynamics 365, Semantic Kernel, Microsoft 365 Copilot 등 Microsoft 에코시스템 전반에 MCP를 통합
- 일관된 에이전트 플랫폼을 구축

Windows 11의 MCP 네이티브 지원



MCP Registry

- 중앙 집중식, 보안된 레지스트리
- 디바이스에 설치된 MCP 서버 탐색 및 기능 조회 가능

MCP Servers

- 파일 시스템 관리, 창 관리, WSL 등 핵심 Windows 기능
- AI 에이전트가 시스템 기능에 직접 액세스하여 고급 시나리오 지원

표준화된 통합 프레임워크

- 네이티브 Windows 앱과 AI 에이전트 결합
- 에이전트의 기술 및 기능 확장

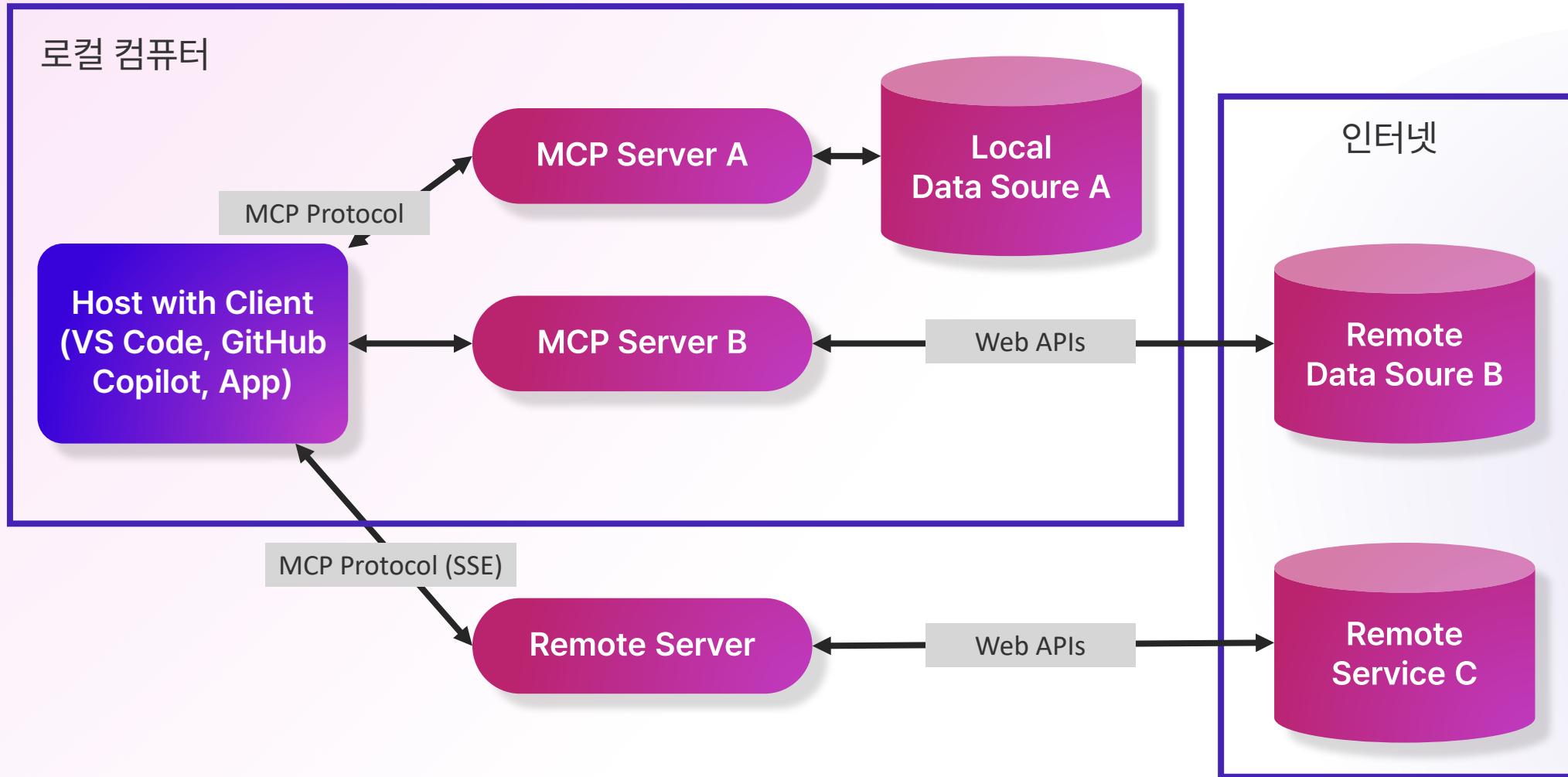
데이터 프라이버시 및 보안

MCP 개요

- AI 모델과 애플리케이션 간의 문맥(Context) 교환을 표준화한 개방형 프로토콜
- 메시지 기반 요청·응답 구조와 플러그인 아키텍처로 유연한 확장성 제공
- 파일 시스템·데이터베이스·웹 API 등 다양한 외부 리소스를 동일한 방식으로 연동
- Anthropic의 주도로 개발되며 Visual Studio Code, Cloude, GitHub Copilot 등 주요 AI 도구에서 채택 중

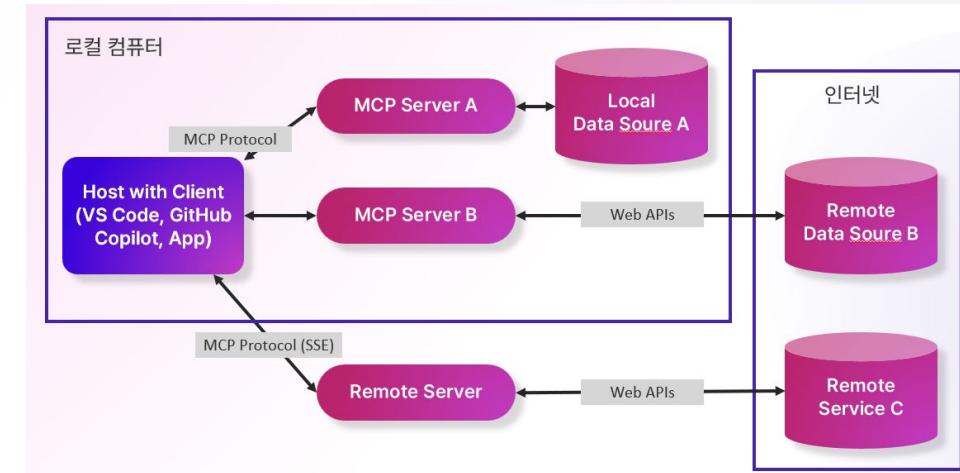


MCP의 구조



MCP의 구조

- **클라이언트-서버 아키텍처** 기반으로, 호스트 애플리케이션이 내부 MCP 클라이언트를 통해 MCP 서버와 통신
- **호스트(Host)**: VS Code, IDE, 채팅 앱 등 AI 기능을 탑재한 주체로, 사용자 인터페이스와 AI 워크플로우 관리
- **MCP 클라이언트**: 호스트 내부에서 동작하며, MCP 서버와의 1:1 연결 관리 및 메시지 교환 담당
- **MCP 서버**: 파일 시스템, 데이터베이스, 외부 API 등 특정 기능·데이터 접근을 경량화된 서비스로 제공
- 하나의 호스트는 **복수의 MCP 서버**에 동시 연결 가능해, 코드 편집기에서 파일·Git·DB 등 다양한 소스에 AI 접근 확장 가능



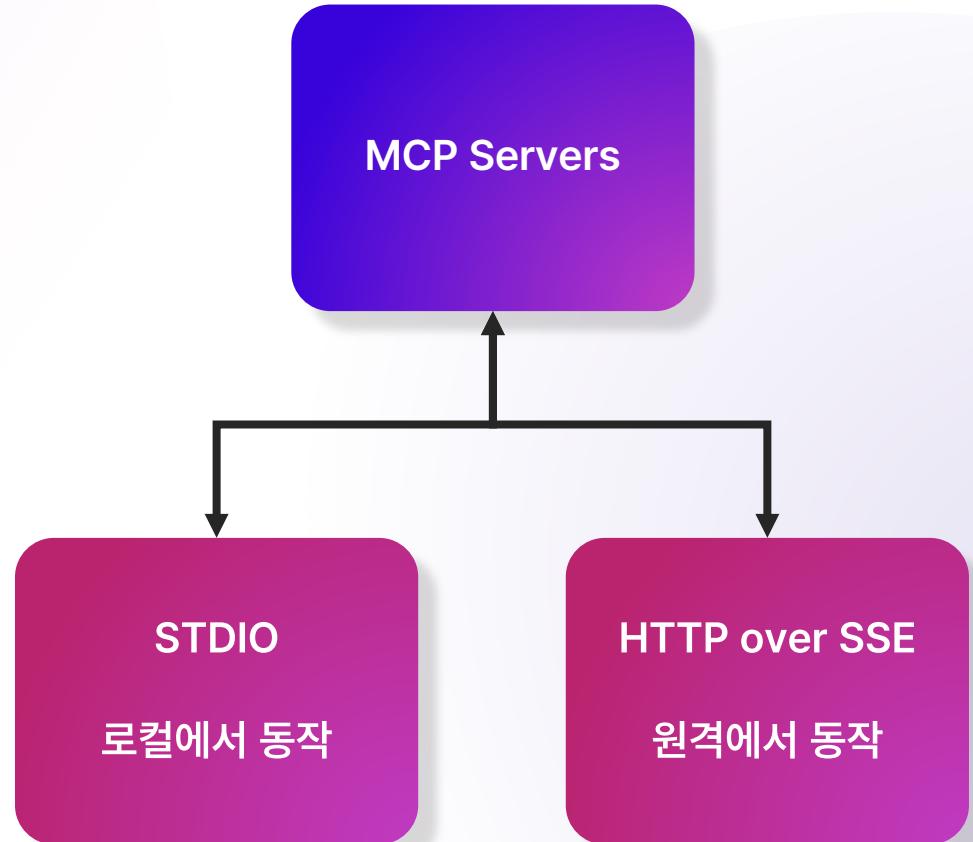
MCP의 전송 방식

로컬 통신: Standard Input/Output (stdio)

- MCP 서버와 클라이언트를 같은 머신(또는 같은 컨테이너)에서 프로세스 간 통신으로 동작시키는 방식
- JSON-RPC 메시지를 표준 입력(stdin)과 표준 출력(stdout)을 통해 주고받음

원격 통신: HTTP over Server-Sent Events (SSE)

- MCP 서버를 웹 서비스로 띄워, HTTP 엔드포인트를 통해 통신
- SSE(Event Stream)로 서버→클라이언트 스트리밍, HTTP POST로 클라이언트→서버 요청을 처리하는 하이브리드 방식

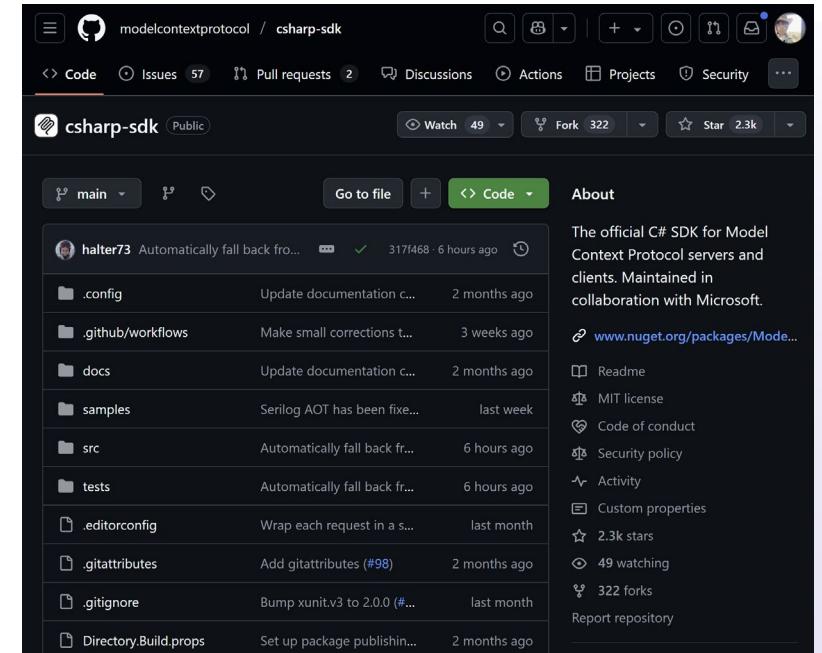


공식 C# MCP SDK

- 공식 C# SDK는 ModelContextProtocol NuGet 패키지로 제공되며, 현재 프리뷰 상태로 배포 중
- 서버/클라이언트 API와 JSON-RPC 통신 처리 로직을 모두 포함, 프로토콜 세부사항 대신 애플리케이션 로직 구현에 집중 가능
- 설치 명령 한 줄로 간단 추가:

```
dotnet add package ModelContextProtocol --prerelease
```

- Microsoft·Anthropic·MCP 오픈 소스 기여자들의 협력으로 개발 제공되며, GitHub 및 공식 문서에서 지속적 업데이트와 릴리즈 노트 확인 가능



MCP 서버의 주요 기능

Tools

- 모델이 직접 호출해 실행하는 액션(계산, 웹 요청, DB 조회, 파일 작업 등)
- ListTools → 선택 → CallTool 실행

Resources

- 파일·문서·DB 레코드·로그·이미지 등 대용량 컨텍스트 데이터
- 고유 URI로 식별 → ListResources → ReadResource로 읽기

Prompts

- 코드 분석·오류 요약 등 템플릿화된 워크플로우
- ListPrompts → GetPrompt → 인자 입력 후 AI 요청

MCP Tool

Semantic Kernel

- 기존 Kernel Function 설정을 그대로 지원
- 기존 AI Agent에 연결된 커널 함수를 그대로 MCP 서버로 만드는 것이 가능

```
public class DesktopTools
{
    [KernelFunction, Description("Open calculator from desktop")]
    참조 0개
    public static string OpenCalculator()
    {
        Process.Start(new ProcessStartInfo
        {
            FileName = "calc.exe",
            UseShellExecute = true
        });
    }

    return "Calculator opened";
}
```

Microsoft.Extensions.AI 혹은 Model Context Protocol C# SDK

- McpServerType, McpServerTool 어트리뷰트로 도구를 등록

```
namespace MyFirstMCP;

[McpServerToolType]
public static class MonkeyTools
{
    [McpServerTool, Description("Get a list of monkeys.")]
    public static async Task<string> GetMonkeys(MonkeyService monkeyService)
    {
        var monkeys = await monkeyService.GetMonkeys();
        return JsonSerializer.Serialize(monkeys);
    }

    [McpServerTool, Description("Get a monkey by name.")]
    public static async Task<string> GetMonkey(MonkeyService monkeyService, [Description("The name of the monkey to get")]
    {
        var monkey = await monkeyService.GetMonkey(name);
        return JsonSerializer.Serialize(monkey);
    }
}
```

Demo

MCP 어플리케이션 만들어보기

감사합니다