MCP 簡介

Model Context Protocol (MCP)，中文譯為模型上下文協議。
規範應用程式如何回覆 LLM 經過標準化的上下文，讓 LLM 在處理相關內容時，能夠更加精準與減少幻覺，在此同時也提供 LLM 應用上更多的可能性。

MCP 優劣勢

優勢 (Pros)

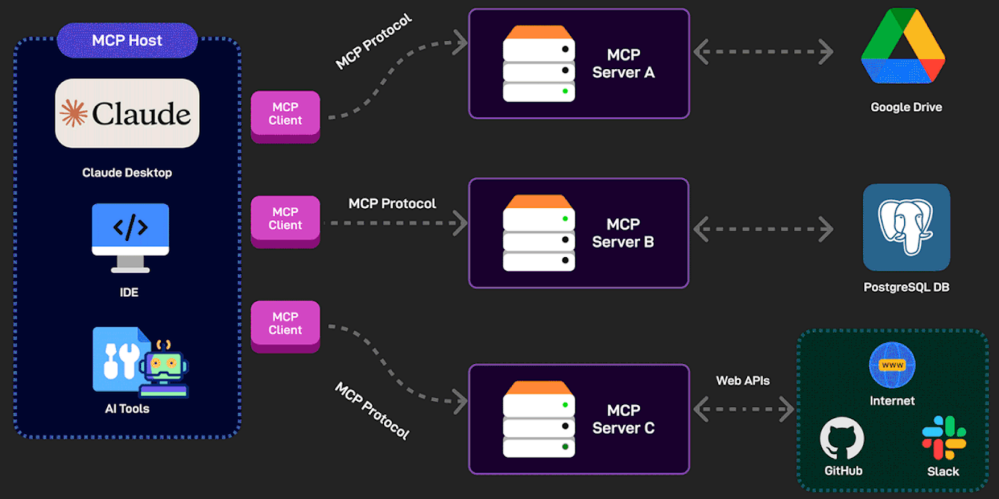
- **標準化協議**: 提供統一的介面規範，降低整合複雜度
- **減少幻覺**: 透過結構化上下文提供，提升 LLM 回應準確性
- **模組化設計**: 可獨立開發和部署不同功能的 MCP Server
- **擴展性強**: 支援多種資源類型和工具整合
- **生態系統**: 與主流 AI 應用程式良好整合

劣勢 (Cons)

- **學習成本**: 需要理解協議規範和實作細節
- **開發複雜度**: 相較於直接整合，需要額外的協議層
- **效能考量**: 多層通訊可能帶來延遲
- **相依性管理**: 需要維護 MCP Server 的穩定性和版本相容性
- **除錯困難**: 分散式架構增加問題排查複雜度
- **安全性問題**: 需要確保 MCP Server 的安全性和權限控制

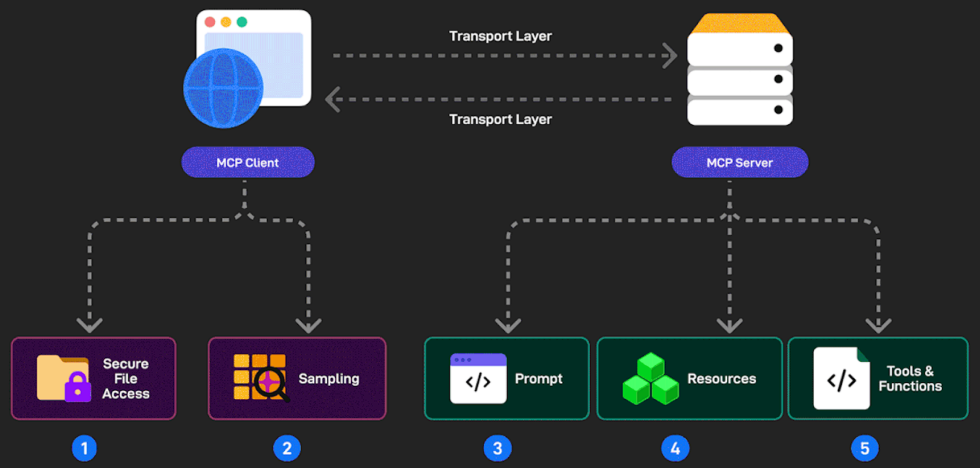
MCP 基礎架構

- **MCP Host**: 處理客戶端與伺服器之間的通訊，像是 Claude Desktop / Cursor / Raycast
- **MCP Client**: 發起請求的應用程式或服務，通常是需要獲取上下文資訊的 LLM 應用，大多數時候，MCP Host 與 MCP Client 由同一個角色承擔責任。
- **MCP Server**: 提供特定領域知識或功能的服務端，回應客戶端的請求並提供標準化的上下文資料



MCP Server 上下文來源

- **Prompts**: 預先撰寫固定的提示詞，當 LLM Agent 需要時能夠可攜帶參數的調用預先定義好的提示詞使用。
- **Resources**: 預先設置好的靜/動態資源，以供 LLM Agent 需要時可以獲取對應資源內容，資源格式可以是單純文本或是 binary
- **Tools/Functions**：以標準函式描述方式 function schema 提供的互動操作，允許 LLM Agent 觸發特定行為，ex. 寫入資料、整合第三方應用。



如何配置 Prompts

```
const PROMPTS = {
  "git-commit": {
    name: "git-commit",
    description: "Generate a Git commit message",
    arguments: [
      {
        name: "changes",
        description: "Git diff or description of changes",
        required: true
      }
    ]
  },
}
server.setRequestHandler(ListPromptsRequestSchema, async () => {
  return {
    prompts: Object.values(PROMPTS)
  };
});
```

More on the next slide

```
server.setRequestHandler(GetPromptRequestSchema, async (request) => {
  const prompt = PROMPTS[request.params.name];
  if (!prompt) {
    throw new Error(`Prompt not found: ${request.params.name}`);
  }

  if (request.params.name === "git-commit") {
    return {
      messages: [
        {
          role: "user",
          content: {
            type: "text",
            text: `Generate a concise but descriptive commit message for these changes:\n` +
                  `${request.params.arguments?.changes}`
          }
        }
      ]
    };
  }

  throw new Error("Prompt implementation not found");
});
```

如何配置 Resources / Toos

Google Drive MCP Server 範例參考 : Google Drive MCP Server

當紅炸子 MCP Servers

Awesome MCP Servers

用過都說讚 :

Brave MCP Server / Brave API

Playwright MCP Server

FileSystem MCP Server

族繁不及備載

生態系

Official Clients List

Awesome MCP Clients

類型	名稱	連結
Editor	VSCode	VSCode
Editor	Cursor	Cursor
Editor	Windsurf	Windsurf
Extensions	Cline	Cline

More on the next slide

類型	名稱	連結
LLM Agent	Claude Desktop	Claude Desktop
LLM Agent	Chatwise	Chatwise
LLM Agent	Cherry Studio	Cherry Studio
Command Launcher	RayCast	RayCast

Model Context Protocol

謝謝聆聽

延伸資源

- MCP 官方文件
- Awesome MCP Servers
- Awesome MCP Clients