

AI Agent 基礎

AI Agent

觀察 Observation

可整合資訊

推理 Reasoning

根據上下文思考下一步（如 LLM + 工具）

記憶 Memory

可回顧過往資訊做出更好策略

行動 Action

主動執行任務、發送指令、呼叫系統

Feedback loop



呼叫



Machine Learning
Deep Learning

靜態模型

分類器、預測器、強化學習演算法

被動，無任務管理邏輯

接收輸入，輸出結果

單一任務，精準數據

辨識一張影像、預測一筆資料結果

核心



GPT - 4

Claude

BY ANTHROPIC

LLama 3

大型語言模型

模型架構
大量資料
參數量



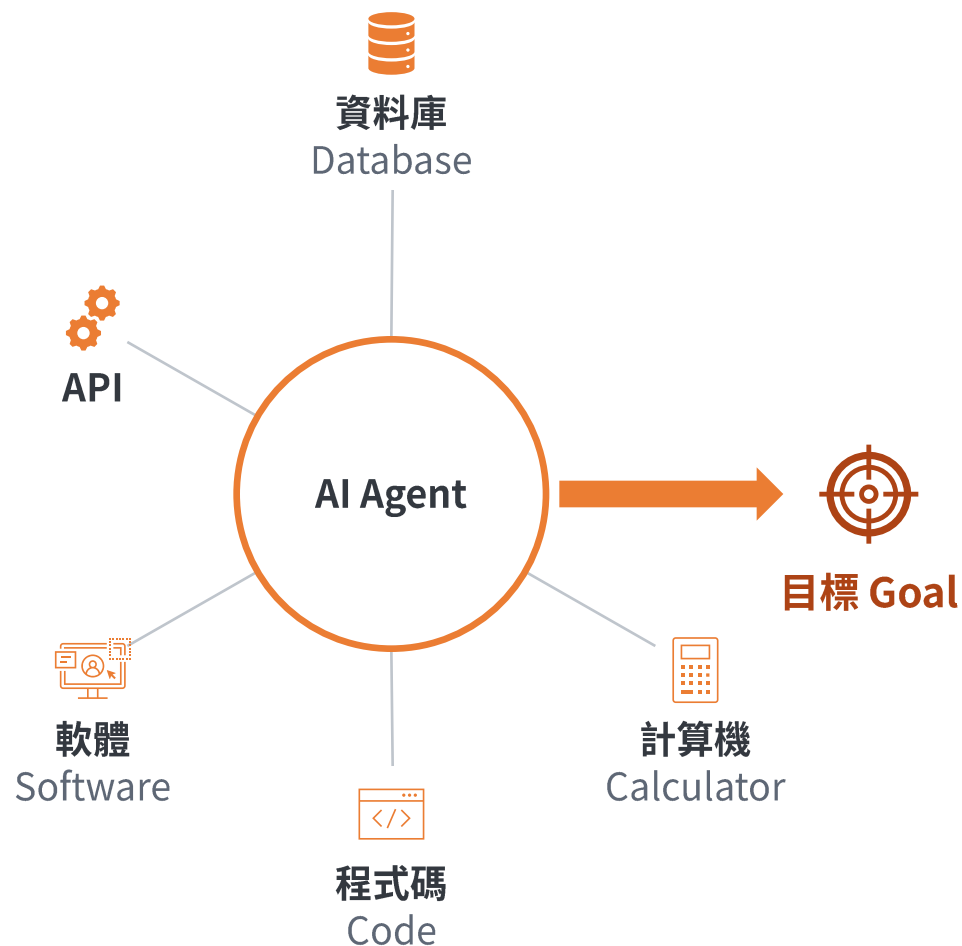
靜態模型

- 分類器、預測器 — 被動,無任務管理邏輯
- 接收輸入 → 輸出結果,單一任務
- 紡織例:輸入布樣照片 → 輸出瑕疵類別
- 精準,但「我知道」≠「我做不到」

AI Agent(核心:Feedback Loop)

- 觀察 Observation:整合資訊(查資料庫、讀文件)
- 推理 Reasoning:根據上下文思考下一步(LLM+ 工具)
- 記憶 Memory:回顧過往資訊,做出更好策略
- 行動 Action:主動執行任務、發指令、呼叫系統
- 紡織例:染機稽核 Agent 自行完成「查結構→SQL 統計→分類→報告」

AI Agent:從「回應指令」到「自主行動」



定義 Definition

AI Agent 結合 LLM 的推理能力與 RAG 的知識汲取能力，並被賦予「目標」與「工具 (Tools)」的自主系統。



自主性 Autonomous

無需持續的人工提示，能主動執行任務。



主動規劃 Planning

拆解並執行多步驟的複雜任務，直到達成目標。



善用工具 Tool Use

串接 API、操作軟體、讀寫資料庫、執行程式碼。

Tools、MCP 與 Skills

Agent 要會「做事」，不只是會聊天

WHY THIS MATTERS

一個只會生成文字的模型，無法查資料庫、寫入 MES、產出報表。要讓它成為能在工廠落地的 agent，必須讓它能「連接外部系統」並「按正確流程執行」。這正是 Tools、MCP、Skills 要解決的三件事。



純語言模型

只能根據訓練知識生成文字，無法存取
即時、私有或外部的資料與系統



缺口

「我知道」≠「我做不到」。模型需要動
作能力、連接能力與流程知識



可落地的 Agent

能讀寫資料、呼叫服務、依標準流程完
成任務，並可被治理與稽核

為什麼需要「標準」：M×N 整合難題

THE INTEGRATION PROBLEM

沒有標準之前

M 個 AI 應用 × N 個工具 = M×N 套客製整合

- ✗ 每個整合自創一套認證、沙箱、資料處理方式
- ✗ 模型難以可靠地「發現」有哪些工具可用
- ✗ 換模型就要重寫整合 → vendor lock-in
- ✗ 維護成本隨工具數量爆炸成長

有了開放標準之後

M + N：寫一次，到處可用

- ✓ 工具發現標準化，host 自動找到可用能力
- ✓ 可複用：server 建一次，任何相容 host 都能接
- ✓ 換模型不用改整合，避免綁定
- ✓ 認證、傳輸、會話有一致規範

三個關鍵詞，先建立心智模型

TOOLS · MCP · SKILLS



Tools

「做動作」

可被呼叫的單一函式——搜尋、計算、查詢、API。一個 tool = 一個動作、一步。



MCP

「給連接」

把 agent 連到外部工具與資料的開放標準。提供「存取」與一致的接法。



Skills

「懂方法」

把一整套工作流程與 know-how 打包。教 agent 「怎麼做才對」、何時算完成。

一句話記法：Tools 做動作 · MCP 給連接 · Skills 懂方法 — 三者互補，不是互相取代。



Tools : 可被模型呼叫的單一函式

FUNCTION CALLING · 基礎

Tool 是 agent 能執行的「一個操作」——想成「一步」，不是「一整套流程」。模型在需要時，會輸出一個工具呼叫 (function call)，帶上結構化參數。

一個 tool 的定義 (概念)

```
name:  query_database
desc:  對生產資料庫執行查詢
input: { sql: string,
         limit: number = 100 }
output: [{ ...rows }]
```

優點

- 精準：一個輸入 → 一個輸出，好測試、好除錯
- 可控：清楚知道發生了什麼，可組合串接

限制

- 不編碼「怎麼做」：先後順序每次可能不同
- 沒有「完成的定義」，格式指示得一講再講



MCP：連接 AI 與外部世界的開放標準

MODEL CONTEXT PROTOCOL

由 Anthropic 於 2024 年 11 月發布，2025 年 11 月捐給 Linux Foundation (與 Block、OpenAI 共組 Agentic AI Foundation)。基於 JSON-RPC 2.0 的開放標準，被稱為「AI 界的 USB-C」——任何相容的 host 都能連接任何 MCP server。



Client-Server 架構

Host (AI 應用) 建立多個 Client，每個 Client 對應一個 Server，維持有狀態的連線



標準化發現

Server 連線時宣告自己提供哪些能力，host 透過 tools/list 自動發現



讀 + 寫一次到位

和只做檢索的 RAG 不同：MCP 讓模型能讀資料、也能執行動作



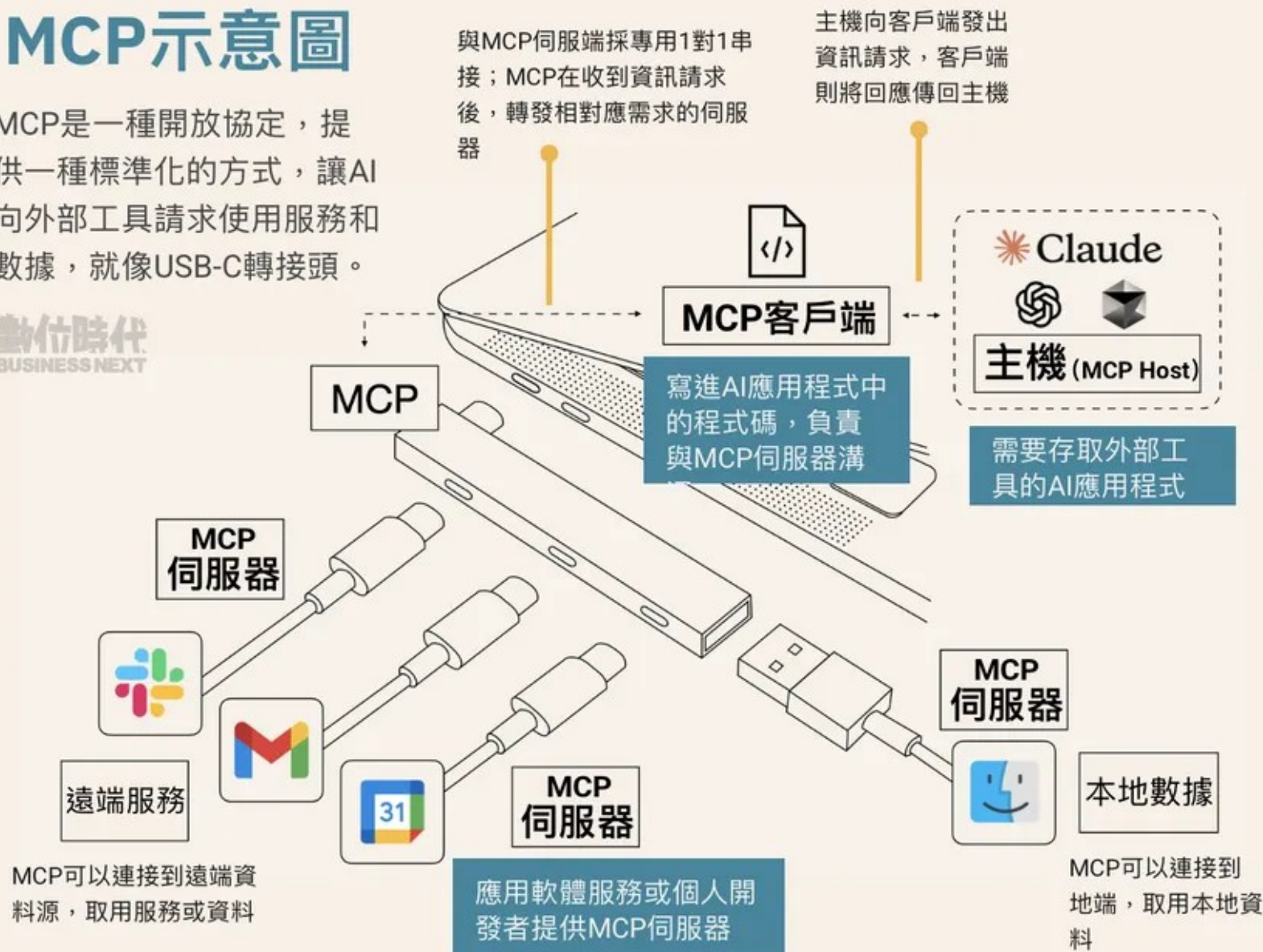
建立在 function calling 之上

保留函式呼叫機制，額外加上發現層、傳輸層與會話層

MCP示意圖

MCP是一種開放協定，提供一種標準化的方式，讓AI向外部工具請求使用服務和數據，就像USB-C轉接頭。

數位時代
BUSINESS NEXT



圖片來源：Norah Sakal；吳玲臻 重製

MCP 架構：Host · Client · Server

JSON-RPC 2.0



傳輸方式：stdio (本機、同機行程) | Streamable HTTP (遠端、多用戶 ; 2025-11 規格取代舊的 SSE)

MCP 三大原語：Tools · Resources · Prompts

WHAT A SERVER EXPOSES



Tools

可執行的動作



模型可呼叫的函式



會產生副作用、能「寫入」



例：執行 SQL、呼叫 API、寫 MES



Resources

唯讀的資料



模型可讀取的資料來源



用 URI 定址、依需求取用



例：檔案內容、資料表、設定檔



Prompts

可複用的模板



預先定義、可帶參數的提示模板



引導固定的工作流程



例：分析 log、產出標準報告

MCP 生態系與安全須知

ECOSYSTEM & SECURITY

10,000+

公開 MCP server (GitHub 、 Postgres 、 Notion 、 Slack 、 Stripe 、 Figma...)

業界共識

ChatGPT 、 Cursor 、 Gemini 、 Copilot 、 VS Code 皆已採用

開放治理

Anthropic 發起 · 現由社群 / Linux Foundation 維護

安全須知 (工廠落地特別重要)



遠端連線強制 OAuth 2.1 + PKCE 並用 Resource Indicators (RFC 8707 · 2025-06 規格) 避免 token 被轉給錯的 server



Tool 描述是模型會讀的文字 可能被 prompt injection / 「tool poisoning」操控——只接信任來源的 server



最小權限 + 稽核軌跡 限制 server 能存取的範圍 · 記錄每一次工具呼叫 · 便於追溯與治理

Skills：把「怎麼做」打包成可複用能力

AGENT SKILLS · SKILL.md

許多 agent 互動不是一次性的提問，而是每次都照同樣步驟、用同樣脈絡、套同樣邏輯的「可重複流程」。Skill 就是把這套流程打包起來。



一個 SKILL.md 檔

YAML 結構化中介資料 + Markdown 步驟指示，人類可讀、模型好懂



教的是「方法」

流程步驟、最佳實踐、輸出格式、以及「怎樣才算完成」



漸進式揭露

只在需要時載入相關內容，避免塞爆 context



可自我精煉

因為就是檔案，agent 可讀可寫，隨使用持續改進

Tools vs MCP vs Skills

SIDE BY SIDE

	Tools	MCP	Skills
本質	可呼叫的函式	連接的協定 / 標準	打包的工作流程知識
回答的問題	「做什麼動作」	「怎麼連、怎麼存取」	「怎麼做才對」
形態	函式 + 結構化參數	Client–Server / JSON-RPC	SKILL.md + 腳本 / 附件
提供	單一動作	存取外部工具與資料	流程、SOP、完成定義
類比	一個按鈕	USB-C 連接埠	操作手冊 / SOP
何時用	自訂單一操作	標準化整合外部服務	可重複、要一致的流程
關係	三者互補，常一起使用：Skill 指揮、MCP 連接、Tool 執行		