

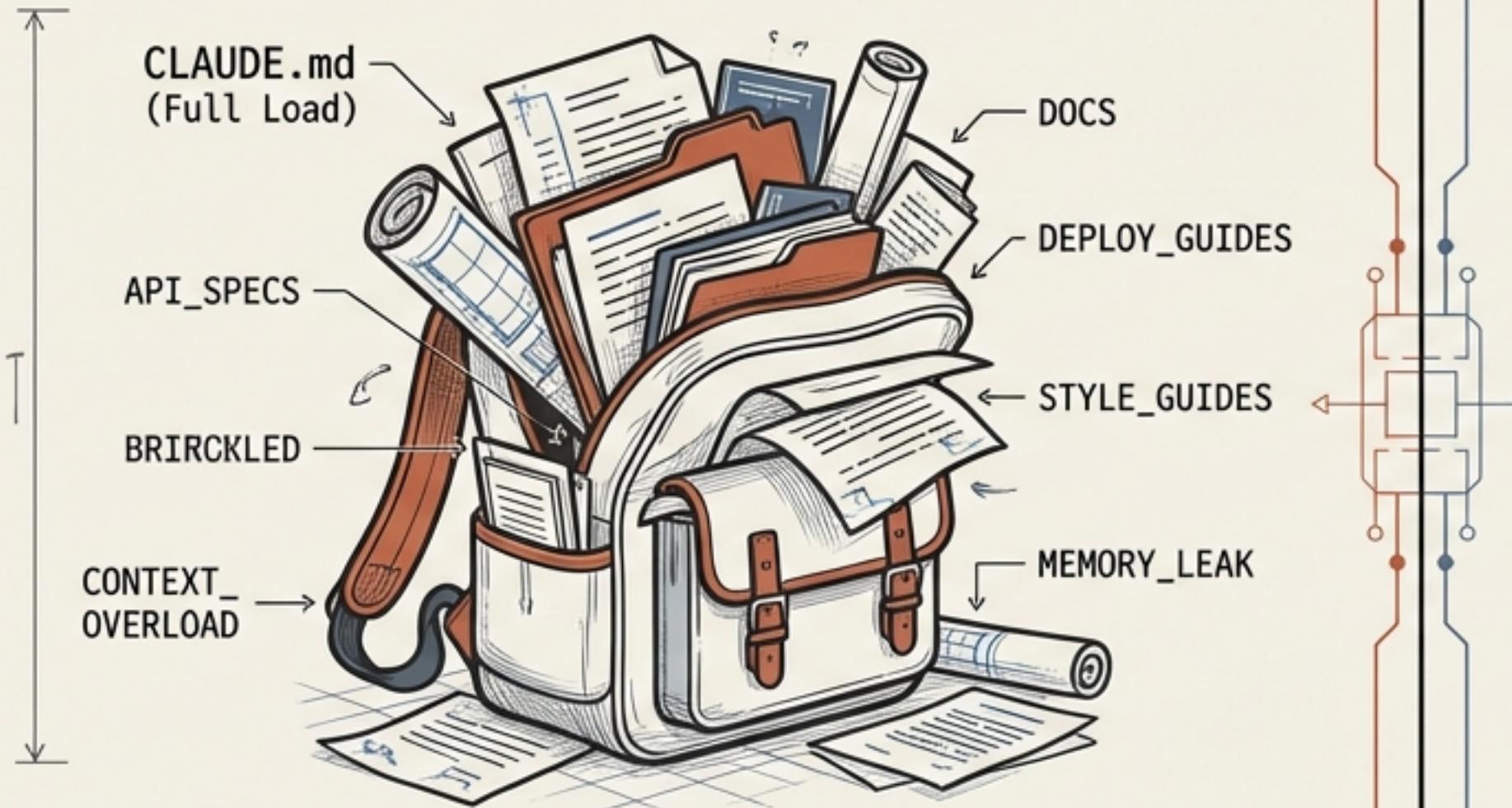
# Claude Code Skills : 讓 AI 變身專業工匠

從通用對話到領域專家——Agent Skills 架構與實戰解析

Based on the technical analysis by Kao Chien-Lung (高見龍) & Agent Skills Open Standard.

## OLD WAY

# 你的 AI 是一個健忘的新進員工嗎？



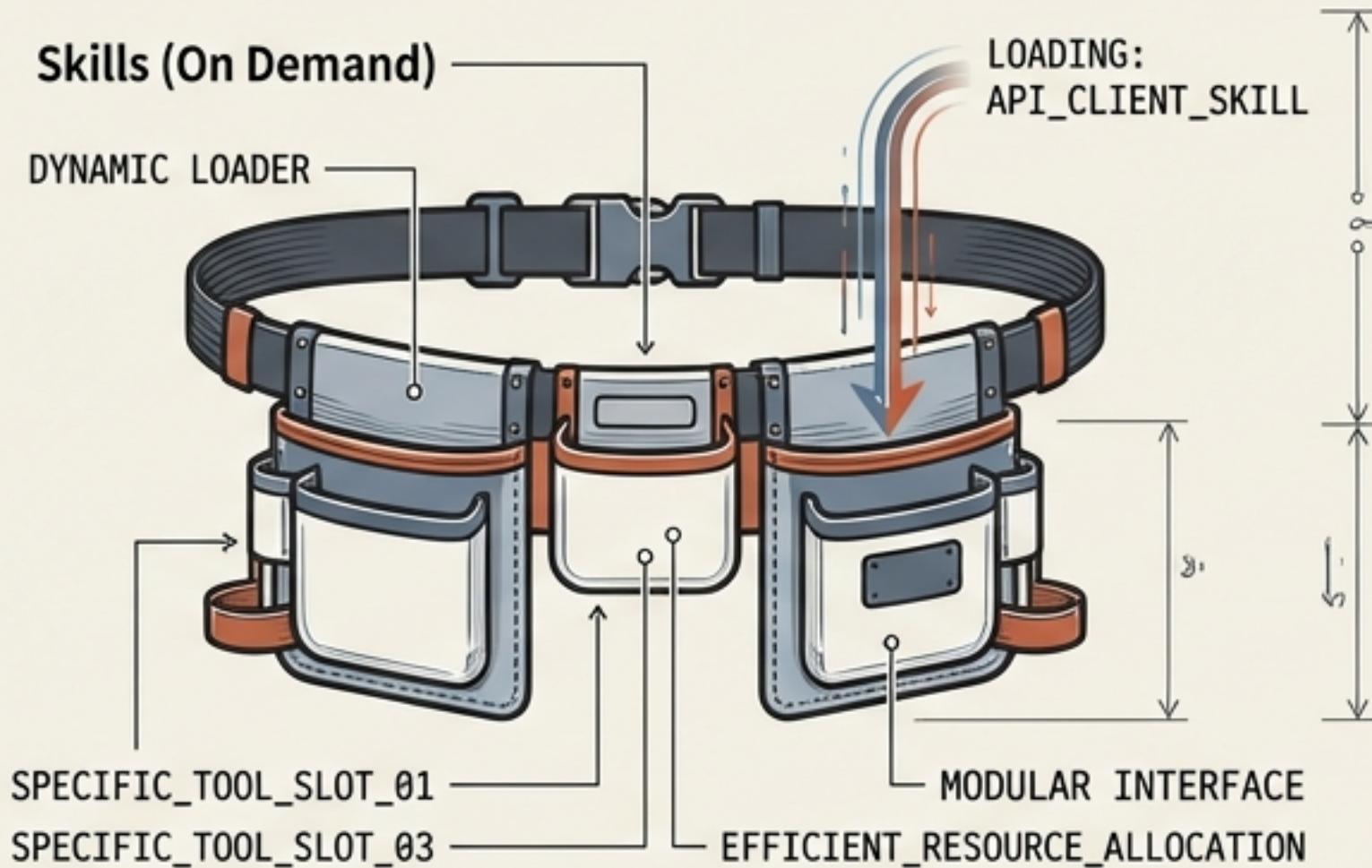
如果沒有 Skills，每個新專案都像在帶實習生。你必須不斷重複：「這是我們的 Coding Style」、「Deploy 流程是這樣」、「去串接這個 API」。

⚠️ 問題：Context Window 脢腫與效能下降

Resource Drain: High  
Efficiency: Low

## NEW WAY

# 你是在用 AI，還是在當 AI 的助理？



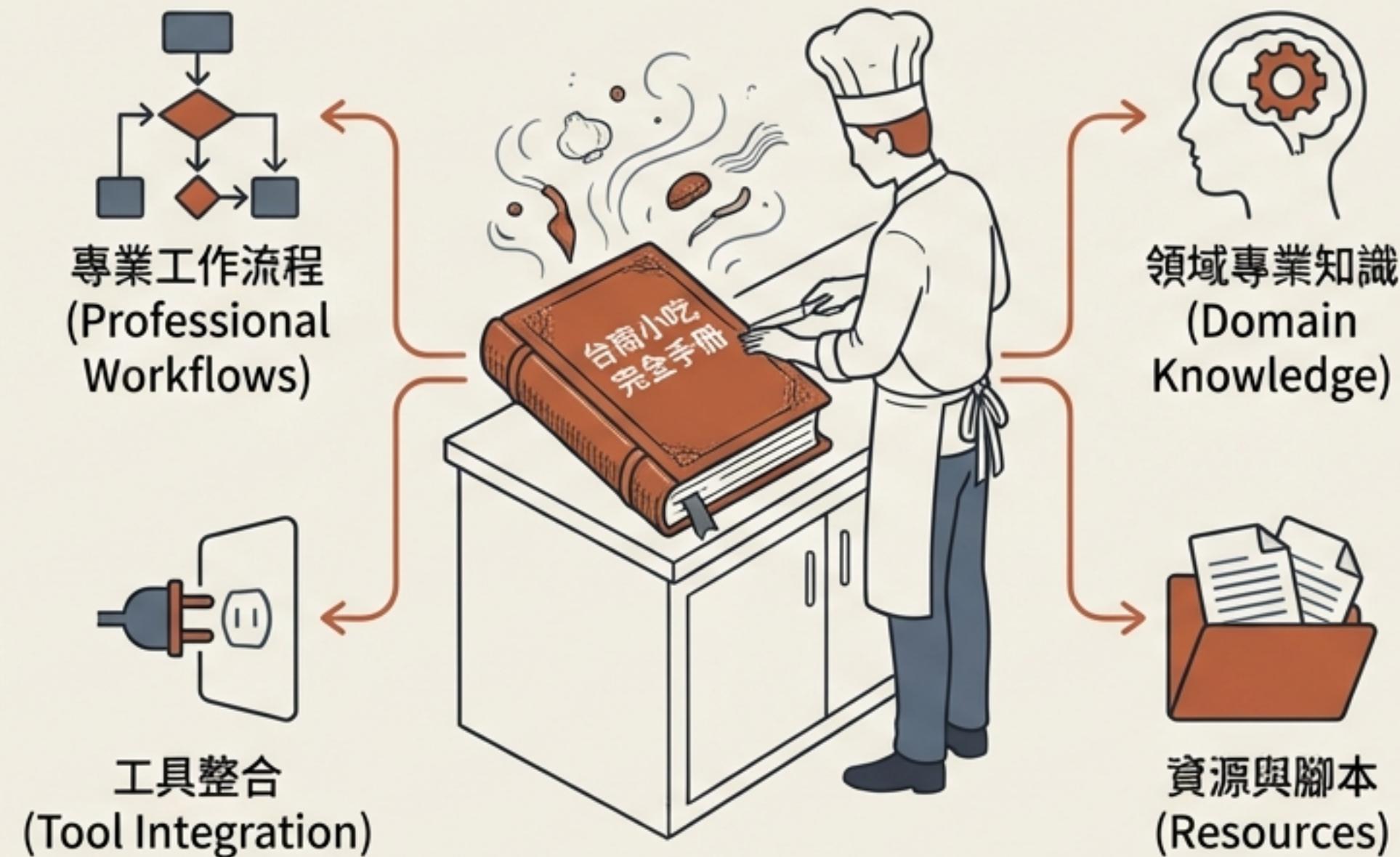
CLAUDE.md 的限制在於它會在啟動時載入所有內容。不管任務是否相關，它都佔據了記憶體與運算資源。

**The Future is Modular: Load What You Need, When You Need It.**

Resource Drain: Optimized  
Efficiency: High

# Skills：你的數位武功秘笈

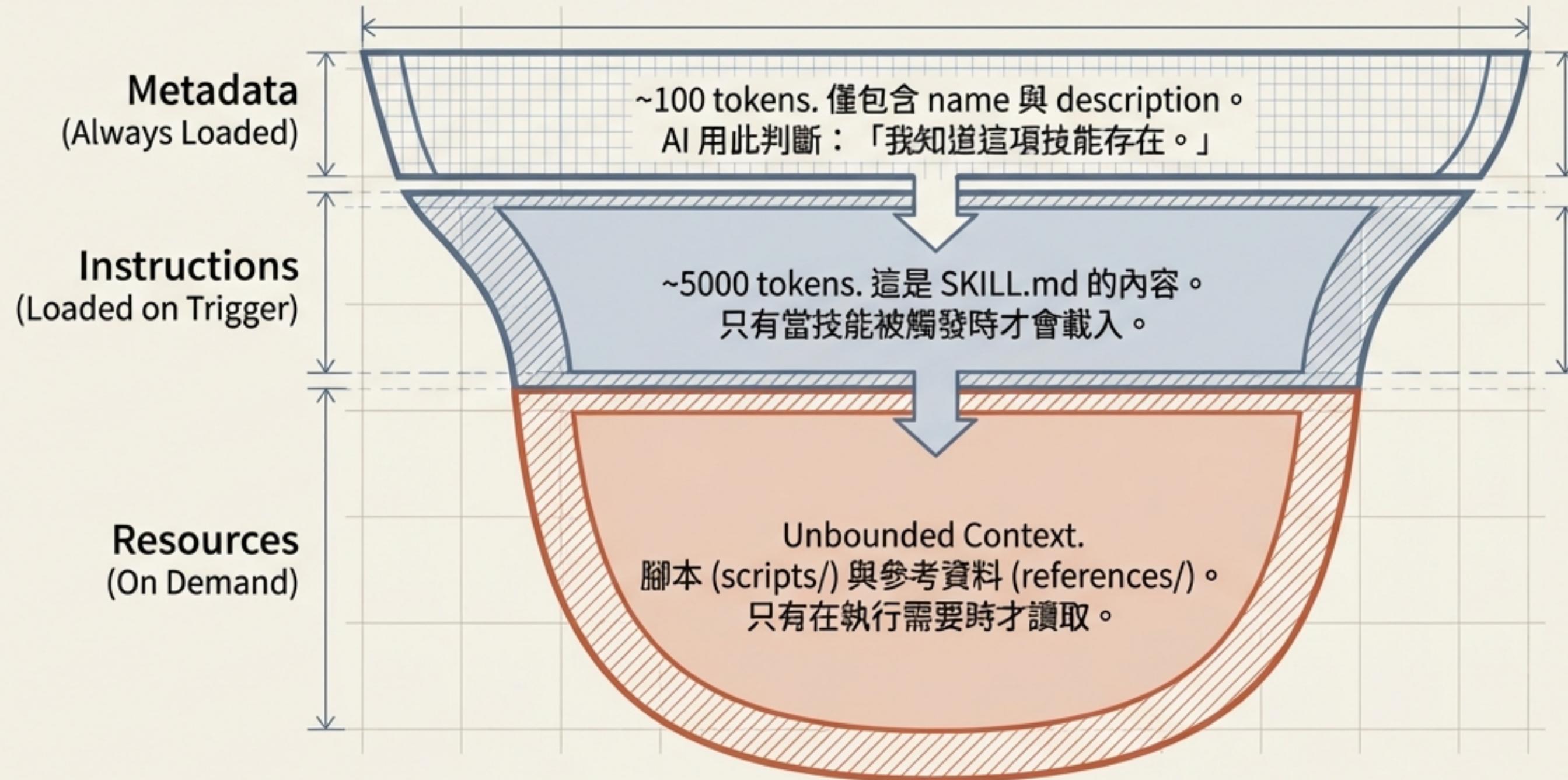
THE DIGITAL ATELIER: AI WORKFLOW OPTIMIZATION



Skills 是模組化、自包含的知識包。想像你是一位通才廚師，Skills 就是那本特定的食譜，教你精確的湯頭比例、肉品部位與獨門醬汁。它將 AI 從「略懂」轉變為「專精」。

# 漸進式揭露：如何放入無限的知識？

The Secret Sauce: Progressive Disclosure



像 Google Maps 導航：  
只在路口揭露下一步，  
而不是一次唸完所有路徑。

# 生態系地圖：Skills 與其他機制的差異

THE DIGITAL ATELIER: SYSTEM MECHANISMS MAP

## Skills (知識/大腦)



- 角色: 賦予 Agent 領域知識
- 觸發: 自動 (由 Context 判斷)
- 場景: 程序性工作流 (例如 Code Review)

## MCP (手腳/工具)



- 角色: 連接外部世界
- 觸發: 按需呼叫
- 場景: 查詢資料庫、API、檔案存取

## Custom Commands (巨集)



- 角色: 快速指令 (/fix)
- 觸發: 手動輸入
- 場景: 固定的 Prompt 模板

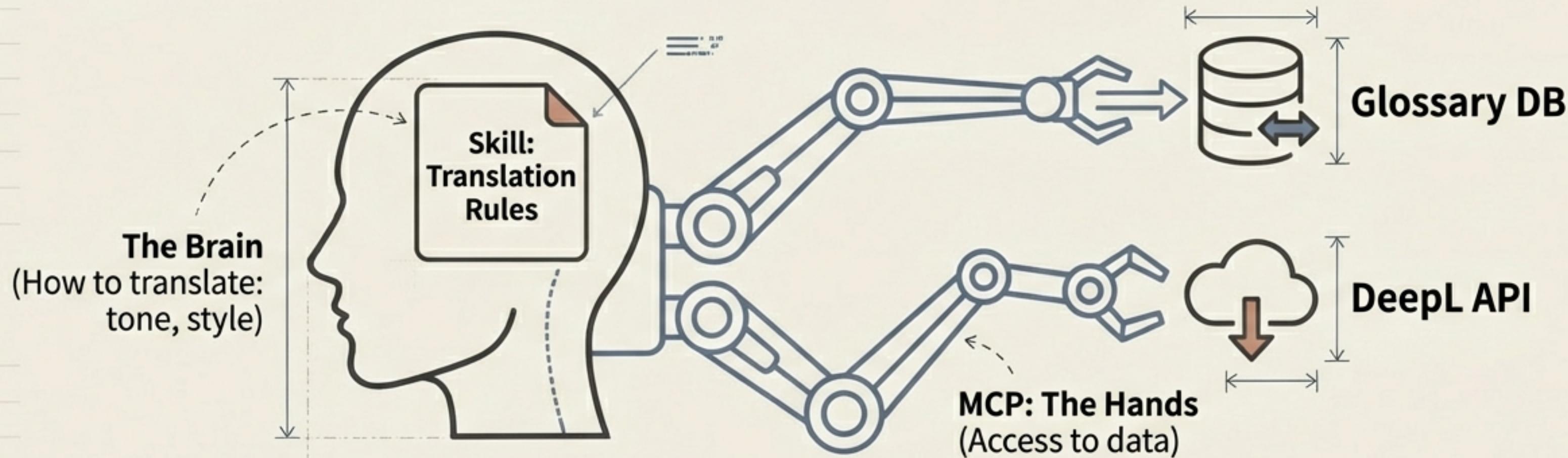
## Subagents (委派)



- 角色: 平行處理
- 觸發: 自動/手動
- 場景: 複雜且耗時的多步驟任務

# Skills vs. MCP : 大腦與手腳的協作

Complementary, not competitors.



- **Skills 指導行為 (How)** : 告訴 Agent 翻譯的語氣與禁語。
- **MCP 執行連接 (Access)** : 讓 Agent 有權限去查閱術語表或呼叫 API。

**“Skills give the Agent a brain; MCP gives the Agent hands and feet.”**

# 解剖 Skill：檔案結構與規範

skill-name/



SKILL.md

The Core: Metadata + Instructions (Required)



scripts/

Executable Python/Bash/JS (Optional)



references/ Docs, Specs, Style Guides (Optional)

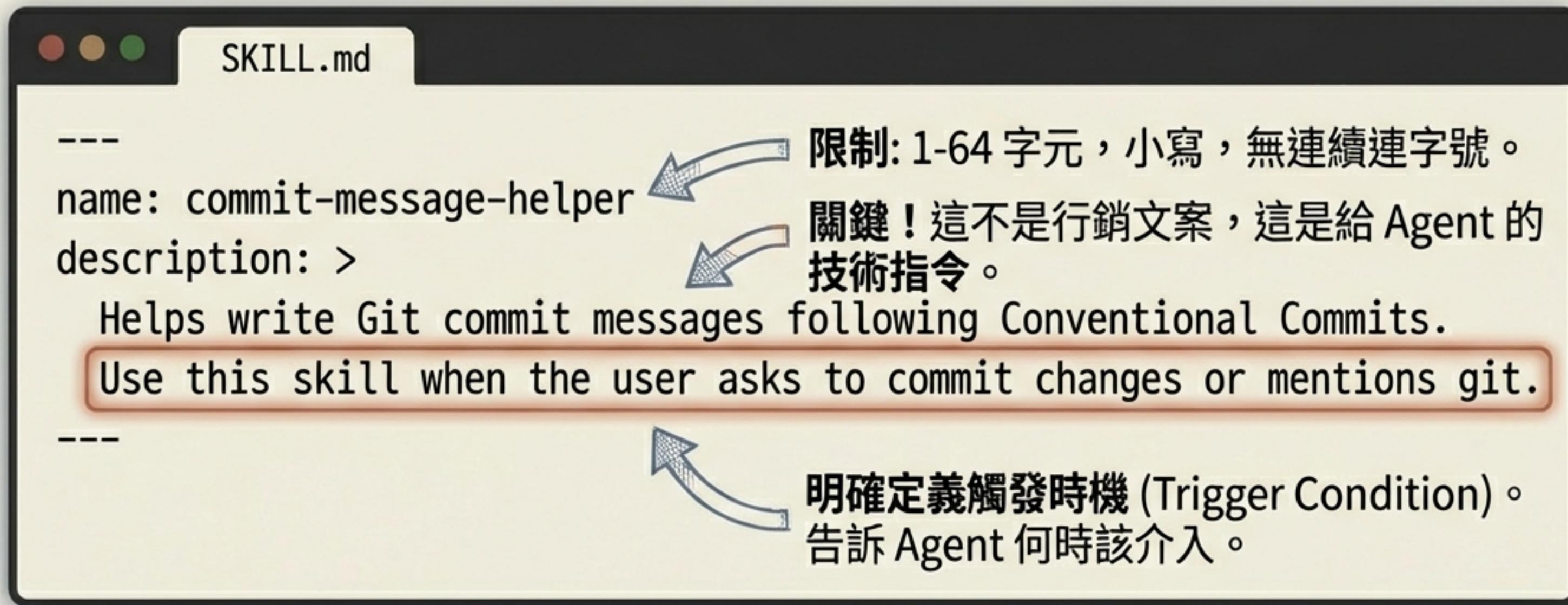


assets/

Templates, Images (Optional)

**標準化結構**：目錄名稱必須與 Skill  
名稱一致。Claude 透過 SKILL.md  
中的引用來發現其他檔案。

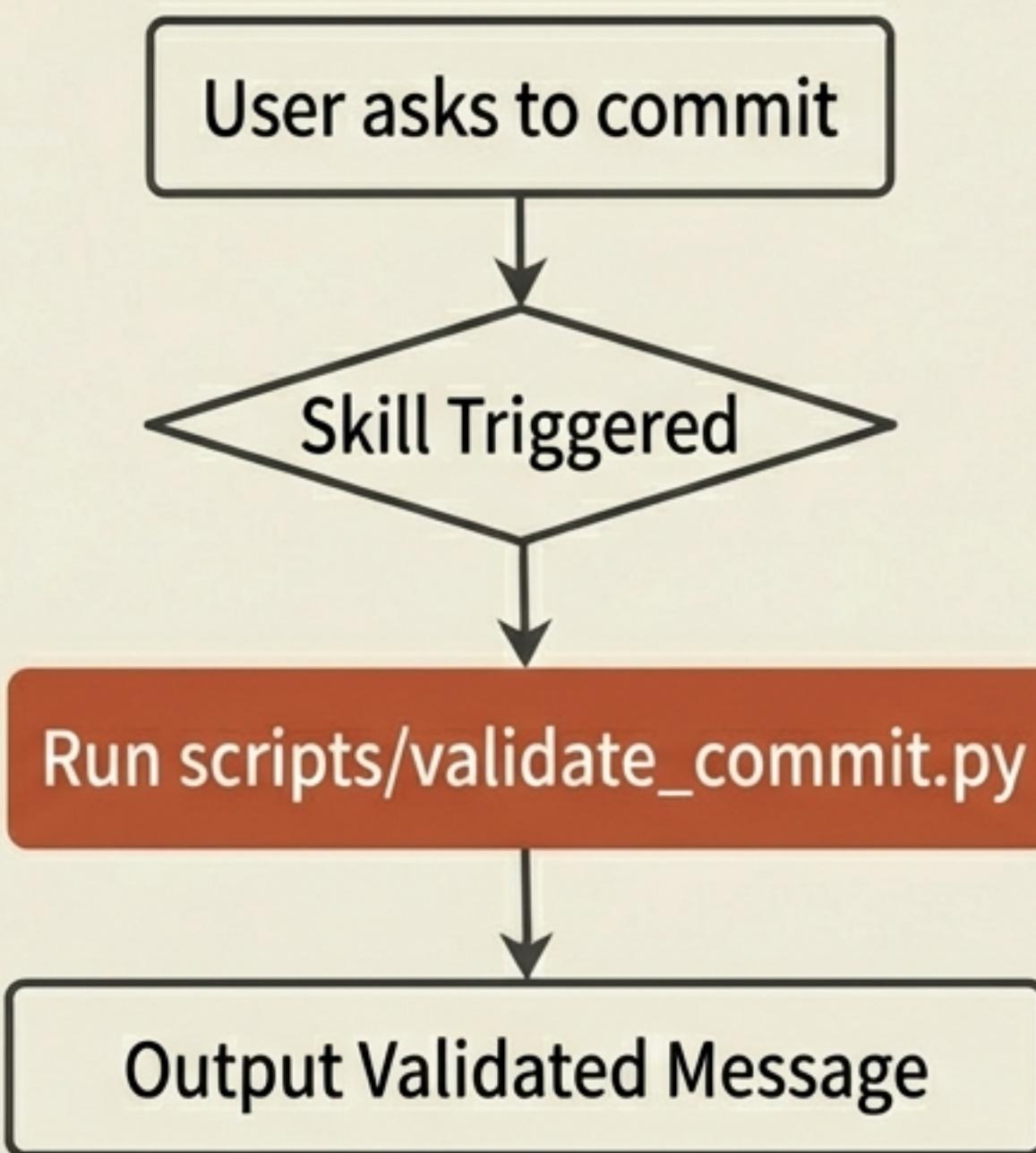
# 核心啟動器：SKILL.md 與 Metadata



Description 的品質決定了 Skill 是否能被正確喚醒。

# 能力升級：加入可執行腳本

從純文字指導到程式碼驗證



```
scripts/validate_commit.py
```

```
def validate_commit_message(message):
    # Regex pattern check
    pattern = r'^feat|fix|docs'
    if not re.match(pattern, message):
        return False, "Format Error"
    return True, "Valid"
```

A screenshot of a terminal window titled 'scripts/validate\_commit.py'. It contains Python code for validating commit messages using a regular expression pattern.

Skill 不再只是靜態文本。透過 `scripts/` 目錄，我們可以要求 Agent 在輸出前執行程式碼進行驗證。

# Skill Engineering：打造高品質技能的秘訣

## 1. Description is King

精準描述觸發條件。如果 Skill 沒反應，先檢查這裡。

## 2. Keep the Core Lean

SKILL.md 控制在 500 行以內。細節請移至 `references/`。

## 3. Examples are Mandatory

人類看範例學習，AI 也是。提供 Input -> Expected Output 對照。

## 4. Handle Edge Cases

定義如何處理資訊不足或模糊的指令。

# 安全至上：防禦惡意腳本

With great power comes great responsibility.



Claude always asks permission before first use.



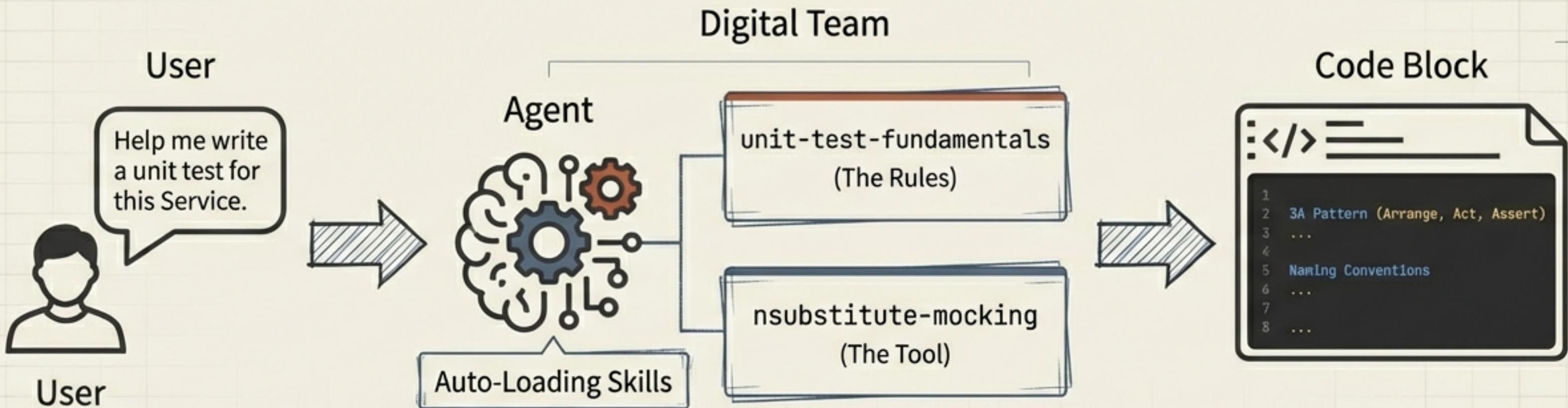
Inspect `scripts/`



- ✓ 驗證來源：只從信任的 Repos 安裝。
- ✓ 檢查腳本：確認`scripts/`內無惡意外部連線(curl/wget)。
- ✓ 審閱指令：確保 SKILL.md 不會洩漏.env 敏感資訊。

# 實戰案例：.NET Testing Agent

iThome Ironman 2025 Winner - Kevin Tseng



包含 27 個專屬 Skills。當開發者要求測試時，AI 自動載入最佳實踐，確保生成的程式碼符合團隊規範。

# 超越 Claude：開放標準的未來

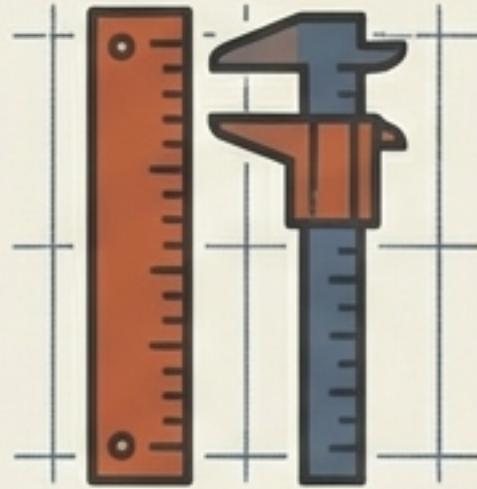
Write once, run anywhere.



**Skills** 已成為開放標準 (Dec 2025)。這是一次投資，隨處通用的資產。  
從單純的 Chatbot 演進為具備判斷力的 **Digital Artisan**。

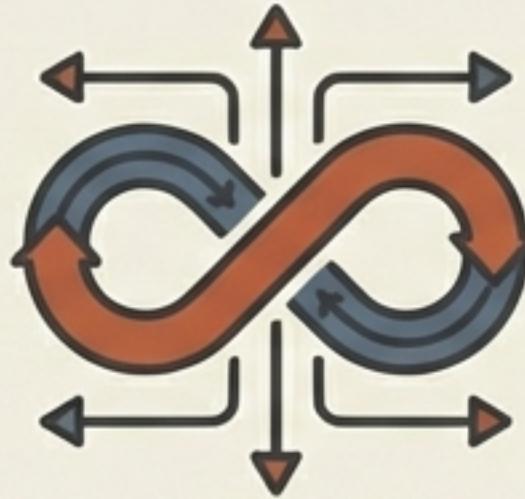
# 為你的團隊建立「數位大腦」

## Standardize (標準化)



確保所有成員（與 AI ）遵循相同的 Linting 與 Commit 規則。

## Scale (擴展性)



漸進式揭露允許無限的 Context，無效能負擔。

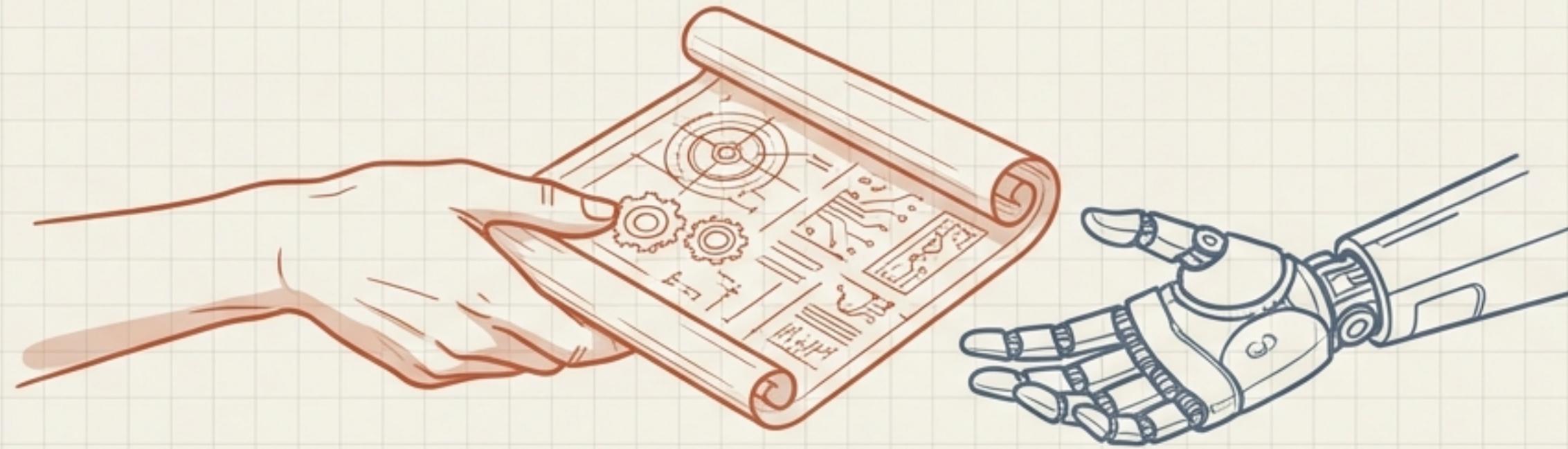
## Share (共享)



將個人知識轉化為組織資產，建立中央 Skill Repository。

# 工具再強，核心還是在「你」

## 為你自己學 (Learn for yourself).



**Skills** 讓你將知識「外包」給 AI，但前提是**你必須先擁有那些知識。**  
**Digital Artisans** 打造工具，其他人只是使用工具。

- Anthropic Documentation
- Agent Skills Repository (GitHub)
- Kevin Tseng's .NET Testing Skills
- Article by Kao Chien-Lung (高見龍)