系統架構與技術策略藍圖

AI Agent

如何製作一個有行動力的AI 助理

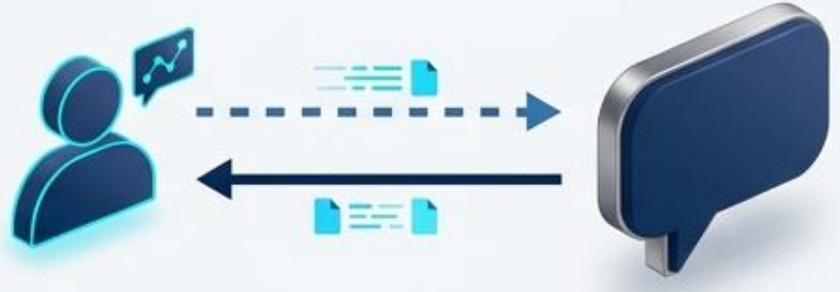


智慧科技產研中心
陳啟彰教授

跨越「回答」與「行動」的界線



傳統 LLM (被動回答)



「使用者提出問題，模型產生回答。」
依賴人類詳細指定每一個操作步驟。



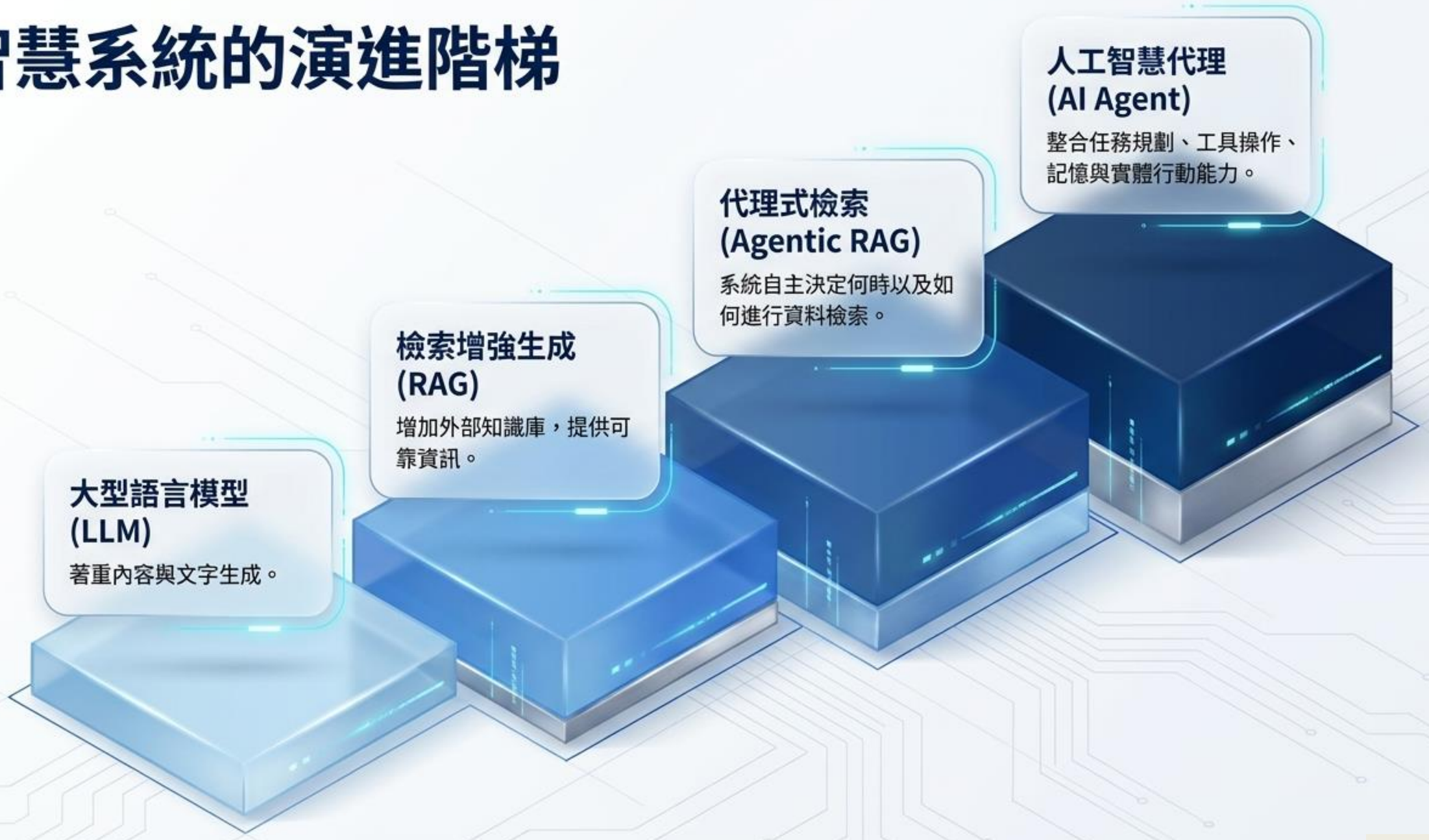
AI Agent (自主行動)



「使用者給予目標，系統自主完成任務。」
系統自行規劃工作、搜尋資料、使用外部工具並根據中間結果修正行動。例如：分析論文並製作摘要。

智慧科技學院

智慧系統的演進階梯



智慧科技學院

AI Agent 的基本解剖圖

指令規範 (Instructions)：
定義 Agent 的角色、總體目標與行為限制。

記憶系統 (Memory)：
保存當前任務狀態與過去對話的上下文。



外部工具 (Tools)：
與世界互動的介面（網路搜尋、資料庫、API）。

領域知識 (Knowledge)：
提供任務所需的特定專業背景資訊。

執行環境 (Agent Runtime)：
控制實際執行過程與循環機制的動力引擎。

智慧科技學院

驅動自主性的核心引擎：Agent Loop

Runtime 負責控制 Agent 的反覆運算與自我修正。



智慧科技學院

釐清關鍵迷思



LLM 本身並不等於 AI Agent。它僅是推理的「大腦」，必須加上控制機制與外部工具，才能形成完整的行動系統。

智慧科技學院

AI Agent Skill



LLM（大腦）

負責理解問題、推理與產生回答。



Tool（工具/四肢）

負責執行動作（例如：搜尋資料、呼叫 API、讀取檔案）。



Skill（技能/工作手冊）

告訴 AI Agent 遇到特定任務時，應該如何使用工具、遵循哪些步驟，以及如何產生結果。

AI Agent（統籌執行者）— 整合 LLM、Tools、Skills，規劃並執行任務。

智慧科技學院

Skill 的底層架構：SKILL.md



skill_folder/
- SKILL.md
- rules.txt
- examples/
- scripts/

SKILL.md

[目標]：描述技能的用途。

[工具權限]：宣告所需的 Tools 與系統權限。

[執行流程]：
Step 1. 選擇工具
Step 2. 執行檢查
Step 3. 整理結果

⚠ 注意：Skill 本身不會自動取得執行權限。AI Agent 必須具備對應的工具與系統權限（如執行 nvidia-smi），才能實際操作。

智慧科技學院

建立方式對比：程式化開發 vs. 自然語言

評估維度	程式化開發 (Code)	自然語言建構 (No-code)
代表工具	OpenAI Agent SDK, Claude Agent SDK, Python	視覺化節點平台、自然語言描述平台(e.g., Claude Cowork, ChatGPT Workspace Agents, Google Workspace Studio)
優勢	高度客製化、精確控制迴圈與資料庫	開發快速、學習門檻極低、維護容易
適用場景	大量執行的正式系統、複雜軟體工程任務	教學、原型設計、快速概念驗證
自由度	極高 (完整掌控 API 與錯誤處理)	中等 (受限於平台提供的 Trigger 與 Action)

智慧科技學院

決定 Token 成本與效能的六大驅動因子

無論使用 Code 或 No-code，推論能力的成本取決於系統架構設計。



模型選擇
(Model Choice)

75%



上下文長度
(Context Length)

50%



LLM 呼叫次數
(LLM Calls)

90%



Agent 迴圈次數
(Loop Count)

No-code 平台可能導致大增

100%



多代理數量
(Multi-Agent Count)

30%



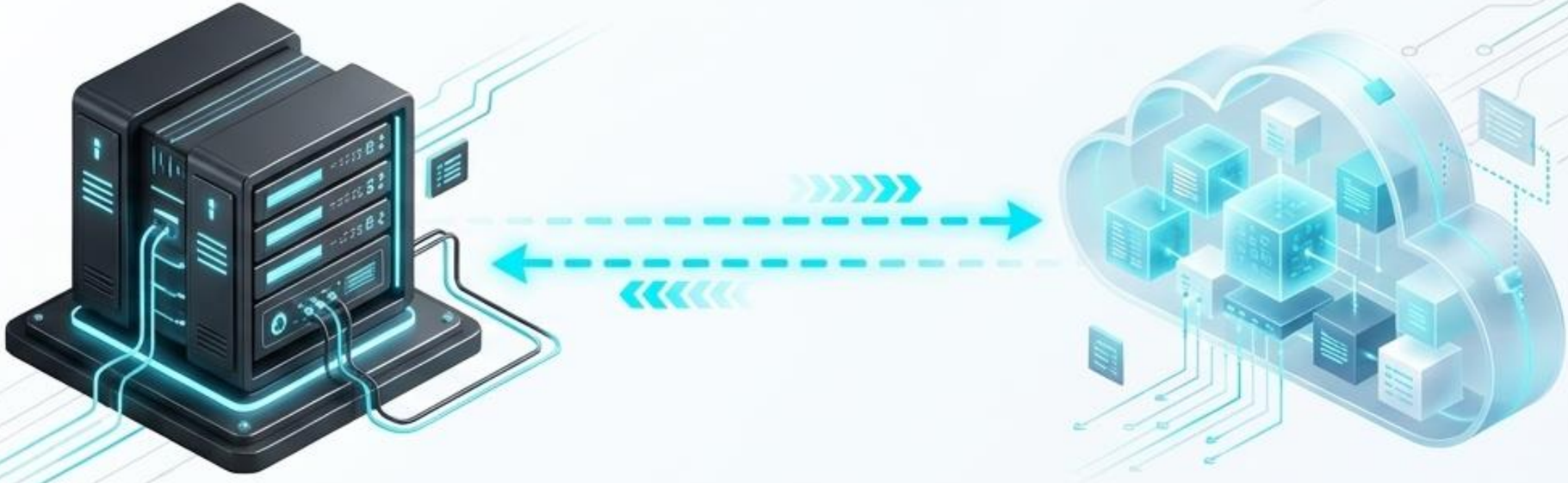
工具回傳資料量
(Payload Size)

60%

智慧科技學院

架構解耦：Agent Runtime 與 LLM 運行地

必須區分「Agent 在哪裡執行」與「LLM 在哪裡執行」。



本地端(Local Server)

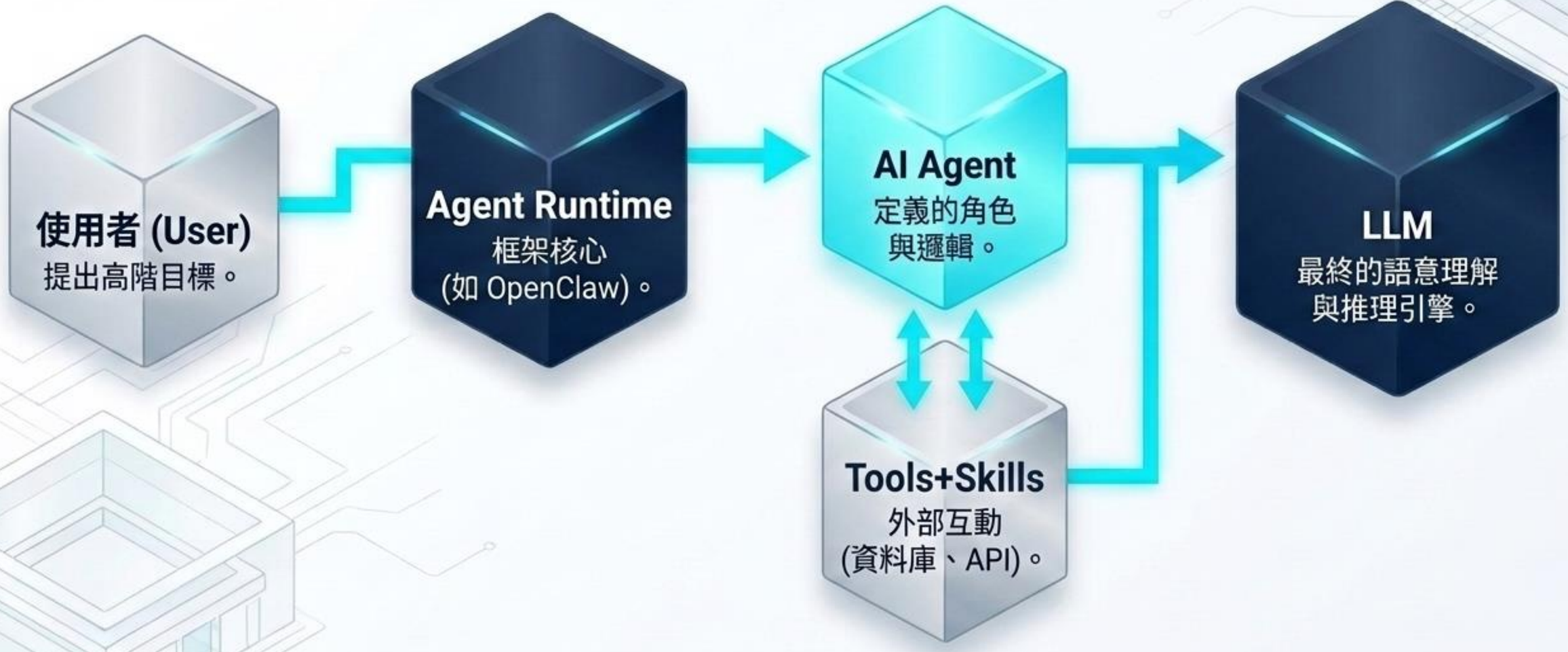
Agent 平台 (如 OpenClaw) 提供工作環境與工具整合。可安裝於校內 Ubuntu 或個人電腦。

雲端 (Cloud LLM) / 邊緣模型

代理所使用的模型。可透過 API 呼叫雲端的 OpenAI、Claude，或連接本地的 vLLM。

智慧科技學院

指令穿透系統的部署資料流



智慧科技學院

戰略決策：何時真正需要使用 Agent？

任務是否具備「高度動態性」？
(需自主判斷是否查庫、用網及應對資料不足)

否

採用：傳統 RAG
流程：問題 → 向量檢索 → LLM → 回答
優勢：更簡單、快速且便宜

是



採用：AI Agent 架構
優勢：靈活處理未知情況，具備自我修正與規劃能力

AI Agent 並非越複雜越好，應依據任務動態性選擇架構。

智慧科技學院

最佳實踐：各司其職的混合架構

設計良好的 Agent 不是讓 LLM 處理所有事情。

LLM 負責理解、推理與規劃

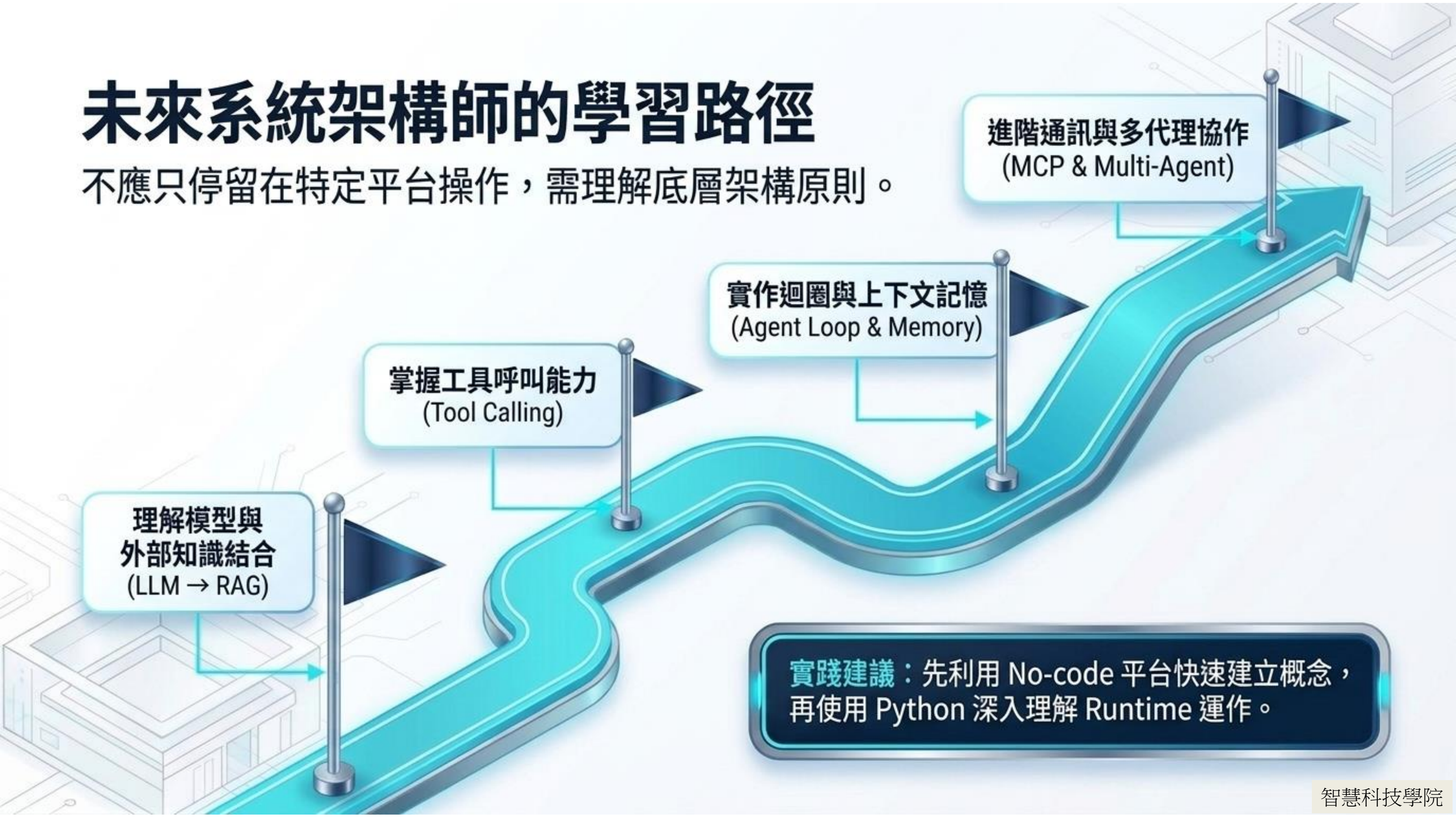


傳統程式負責 SQL 查詢、數值計算與固定商業規則

智慧科技學院

未來系統架構師的學習路徑

不應只停留在特定平台操作，需理解底層架構原則。

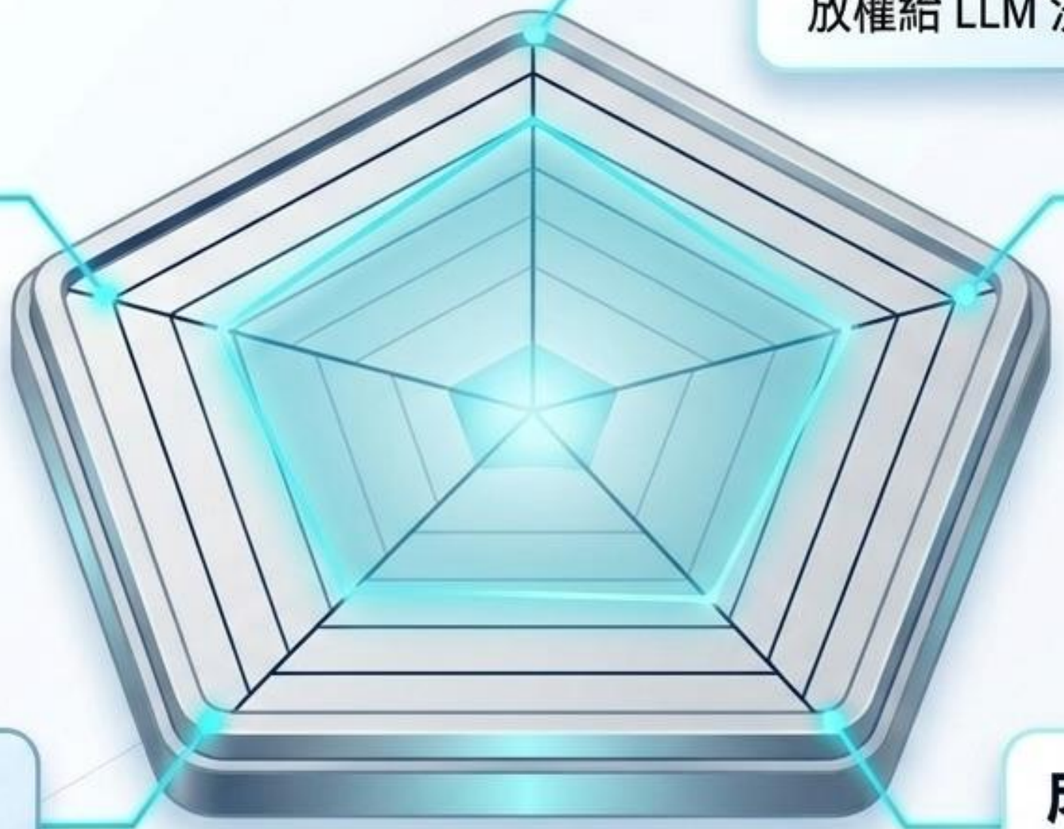


實踐建議：先利用 No-code 平台快速建立概念，再使用 Python 深入理解 Runtime 運作。

智慧科技學院

下一代智慧系統的五角平衡

未來核心能力在於系統設計的權衡藝術。



自主性 (Autonomy)
放權給 LLM 決策的程度

速度 (Speed)
減少延遲與 Loop 次數

成本 (Cost)
Token 消耗與運算資源最佳化

安全性 (Security)
工具權限與存取控制

可靠性 (Reliability)
傳統程式碼保障的確定性

智慧科技學院

結論與願景



AI Agent 代表了生成式人工智慧從單純的「內容生成」正式邁向「自主完成任務」的歷史性轉折。

未來的競爭力不在於操作單一平台，而在於具備系統架構思維：精準判斷何時使用 Agent、如何分配 LLM 與傳統程式的工作，並在複雜環境中建立極致的平衡。

範例：AI Agent--從RAG、Agentic RAG到實際應用

無縫整合動態決策、工具呼叫與資料庫檢索的完整實作藍圖。

技術堆疊：Python × OpenAI API × Milvus × PostgreSQL



智慧科技學院

透過四項核心技術建構 AI Agent 的最小可行性架構



Python

負責主程式運行、流程控制與外部實體工具的執行。



LLM Tool Calling

扮演 Agent 的核心推理引擎，決定何時、如何使用工具。



Milvus

向量資料庫，專注於非結構化文本的相似度檢索 (Embeddings)。



PostgreSQL

關聯式資料庫，用於精確查詢固定的業務數據或客戶資料。

智慧科技學院

義大世界智能客服面臨跨資料源查詢的實戰挑戰

使用者：「請問今天義大世界購物廣場營業到幾點？附近有什麼牛排館推薦嗎？」

需要即時查詢營業時間 API（結構化數據）。

需要進行餐廳資料庫檢索與評價比對（非結構化數據）。

傳統 RAG 無法同時處理兩種不同檢索邏輯的組合任務，必須依賴 AI Agent 自動拆解問題並分派工具。

Gemini Notebook

Python Tools 賦予 AI 串接實體世界數據的能力



get_business_hours(place)

查詢特定設施營業時間。

```
Demo Data:
'義大世界購物廣場': '11:00 - 22:00',
'星巴克': '10:00 - 22:00'
```



search_restaurant()

搜尋園區內與周邊的餐飲選擇。



search_restaurant()

搜尋園區內與周邊的餐飲選擇。



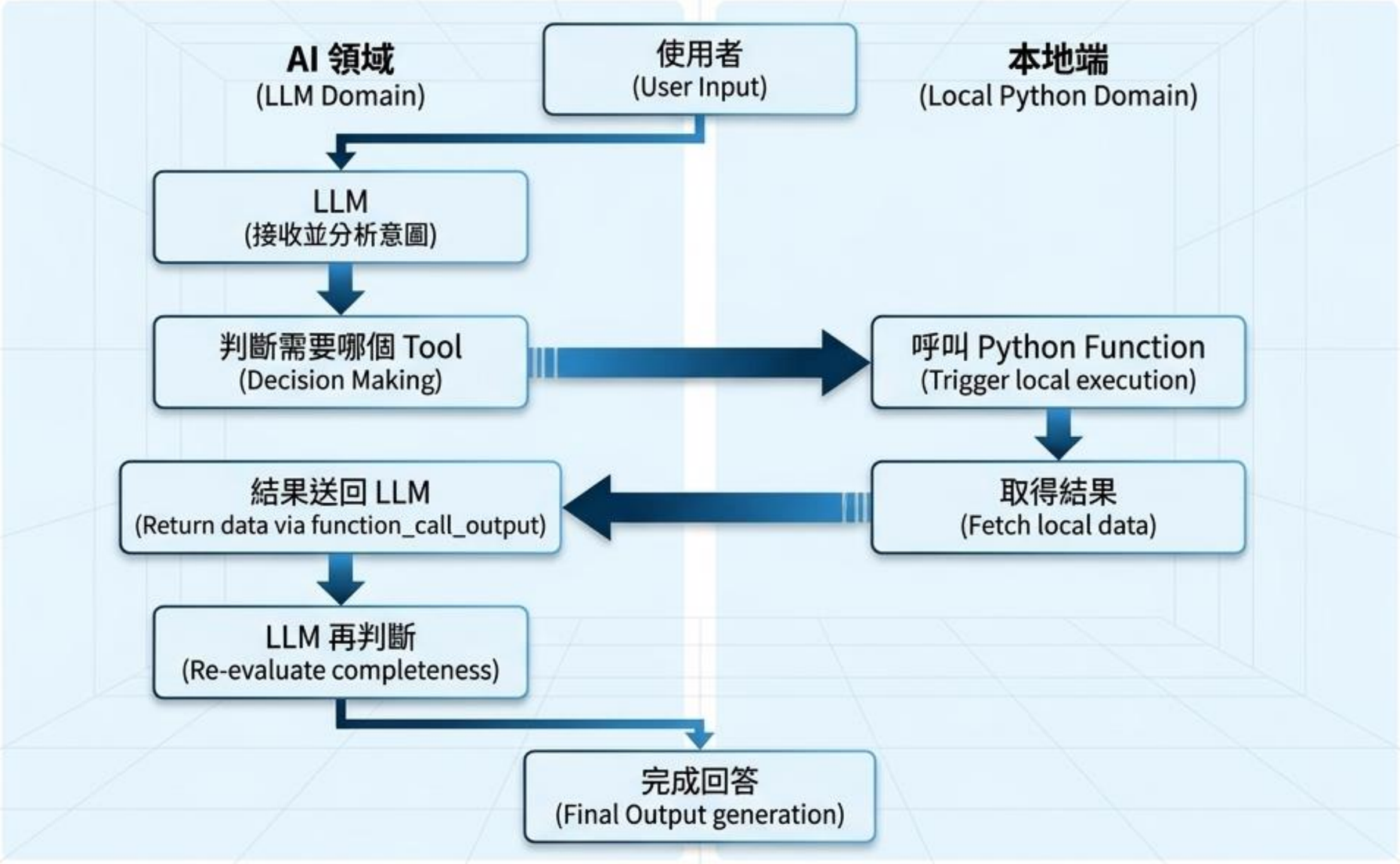
get_statistics()

獲取客服統計與相關營運數據。

註：這些工具在初始階段透過 Python dictionary 進行資料模擬，實戰中將替換為真實的 API 端點。

智慧科技學院

核心執行流程展現 AI 如何自主評估並呼叫外部工具



智慧科技學院

5 輪迴圈機制確保程式碼能確實對映思考與行動邏輯

Python

```python
for step in range(5):
    # Think & Act
    response = LLM.call(tools)
    if response == function_call:
        # Observe
        execute_python_tool()
        return function_call_output
```

Agent Loop 最大重試限制

range(5) 對應 Agent Loop 的最大重試限制，防止無限迴圈。

Act (行動) 階段

function_call 對應 Act (行動) 階段，由 LLM 指定觸發的工具。

Observe (觀察) 階段

function_call_output 對應 Observe (觀察) 階段，將真實世界的數據餵回給大腦。

智慧科技學院