



Workflow Atelier workflows 坊

ASP.NET Core 視覺化平台與 Agent/MCP 自動化實戰

24 小時大師級架構藍圖

系統願景與核心哲學：劃定邊界的藝術



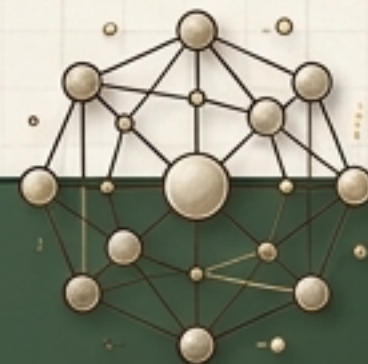
workflow引擎 (Workflow Engine)

職責

負責順序控制、條件分支、週期排程與錯誤處理。

特質

絕對的「決定論」，確保企業邏輯的穩定與可預測性。



AI 代理 (Agent Runtime)

職責

讀取指定 Skill、理解非結構化任務、提出工具呼叫 (Tool Calls)。

特質

受控的「非決定論」，賦予系統靈活的適應與解析能力。

核心原則：將「工作流程式控制」與「AI 任務執行」徹底分開。

我們將建造什麼？ 最終交付成果解析

MCP 整合橋樑：
連接模型意圖與授權工
具的雙向通道

MVC / API 雙核心：
處理前端頁面渲染與非
同步 JSON 請求

AGENTDB：
SQL Server 驅動，集中管理
權限、流程版本與執行日誌



視覺化編排器：
基於 Vue 3 與 Vue Flow，
可拖曳、連線、儲存與發布

Agent Skills 技術包：
經過嚴格審核上架、具備
固定版本的指令集

Worker 執行緒：
獨立運作的排程與背景任
務處理中心

系統分層架構圖 (Separation of Concerns)

Layer 1: Client (Vue 3 / Vue Flow)

職責: 僅負責視覺化與拖曳， workflow 語意由後端解釋。

Layer 2: Controllers (MVC & Web API)

職責: 頁面渲染與 DTO 驗證，絕對不直接操作資料庫。

Layer 3: Services (業務邏輯)

職責: 處理權限授權、流程發布與審核。

Layer 4: Engine & Runtime
(Workflow Engine / Agent Runtime)

職責: 流程狀態轉移與模型對話，不包含資料庫存取。

Layer 5: Data Access (Repositories & 整合服務)

職責: SQL Server 存取、Ollama / MCP 連線。

現代化企業級技術堆疊矩陣



後端核心 (Backend Core)

ASP.NET Core MVC + Web API

.NET 10 LTS

EF Core



前端視覺 (Frontend Visual)

Vue 3

Vue Flow

Razor Views, Bootstrap (古典書房與工坊風格)



排程與執行 (Schedule & Worker)

Quartz.NET

SQL Server 持久化

獨立 .NET Worker



AI 與工具 (AI & Tools)

Ollama Cloud (Gemma 4 31B 模型)

MCP Client (雙向工具整合)

A decorative graphic in the bottom right corner featuring two large, interlocking golden-brown gears. A network of thin, light-brown lines connects several points, including the centers of the gears and other nodes, creating a web-like structure. The background is a light beige color with a subtle grid pattern.



典型 workflows 解析：從觸發到保存



跨節點資料傳遞

節點可引用上游結果 (例如：
`steps.queryOrders.output.orders`)，
由平台受限制的解析器處理，杜絕任意程
式碼注入。

Agent Skills 技術包治理與版本控制



不可變版本原則：已上架內容不得原地覆寫，修改必須建立新版本；目錄路徑由平台解析，防禦路徑穿越攻擊。

AI Runtime：Ollama 與 Gemma 模型的安全整合

1. 取得設定：
從節點讀取
Skill ID 與固
定版本。

2. 安全驗證：
檢查執行者
權限、上架
狀態與檔案
雜湊。

3. 載入資產：
從核准目錄
載入
SKILL.md 與
參考資料。

4. 組裝
Prompt：
結合任務輸
入、平台規
則與允許使
用的工具清單。

5. 模型呼叫：
呼叫 Ollama
Cloud
(gemma4:31
b)。

6. Schema
驗證：
驗證最終
JSON 輸出，
保存結果與
稽核紀錄。



金鑰隔離：API Key 由伺服器端環境變數提供，絕不暴露於前端、日誌或版本控制中。

MCP 工具整合：雙向迴圈與授權邊界

直接 MCP Task (明確指定)

流程中直接指定 Tool，程式驗證參數後直接呼叫。
適用於確定性的 API 整合。



Agent Tool Call (模型自主提出)

模型從「允許清單」中提出 Tool Call，平台驗證授權與 Schema 後呼叫 MCP，將結果交回模型。

授權不可繞過：模型產生的工具名稱與參數僅為「執行請求」，平台強制阻擋未經授權的越權呼叫，工具輸出亦不得修改平台規則。

持久化排程引擎與 Worker 執行緒

即使會員關閉瀏覽器，自動化巨輪仍持續運轉。



排程綁定 (Version Binding)

排程嚴格綁定工作流的「已發布版本」，後續草稿修改絕不影響既有排程。



容錯與重疊 (Fault Tolerance)

支援防重複鍵。若前次任務未完，預設略過重疊執行並記錄。

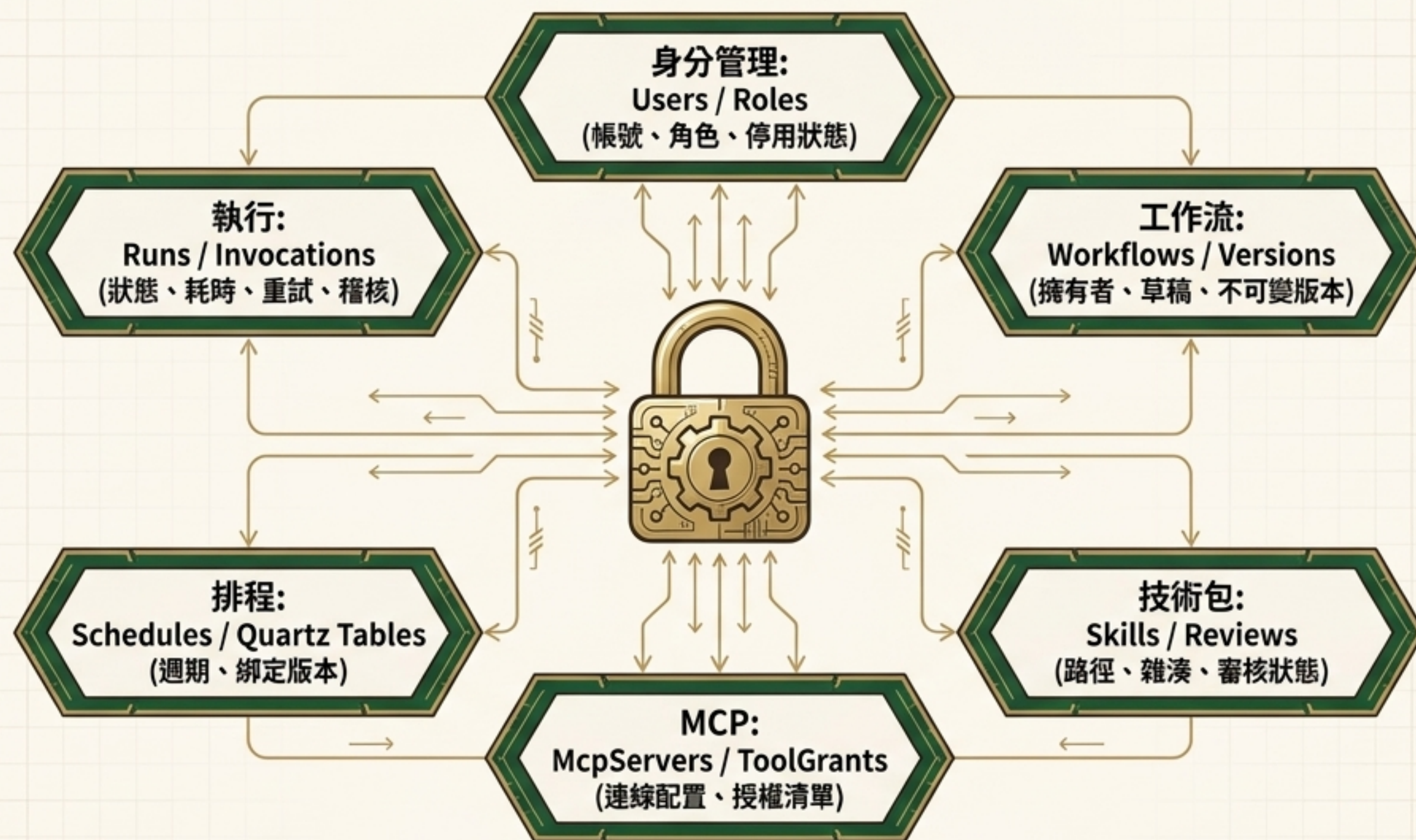


中斷恢復 (Resilience)

伺服器重啟後，由 Quartz.NET 與 SQL Server 保護排程恢復。對外部寫入不明的節點標記待查核。



AGENTDB 資料模型與權限防護圈



雙重驗證邊界：畫面隱藏按鈕不能取代後端授權；API 與背景 Worker 執行時，均須嚴格驗證「資料擁有者」與「授權關係」。

24 小時實戰課程地圖 (The 24-Hour Roadmap)

引擎與 AI (Sec 05-08)

無 AI 工作流引擎 → Skill 治理與審核 → Ollama 與 Prompt 組裝 → MCP Server 與授權工具節點。

進階與維運 (Sec 09-12)

Agent 工具呼叫迴圈 → Quartz 持久化排程 → 執行監控與故障處理 → 端到端部署與驗收。

基礎架構 (Sec 01-04)

關注分離架構 → 會員授權與資料模型 → Vue Flow 拖曳畫布 → 流程儲存與不可變發布。

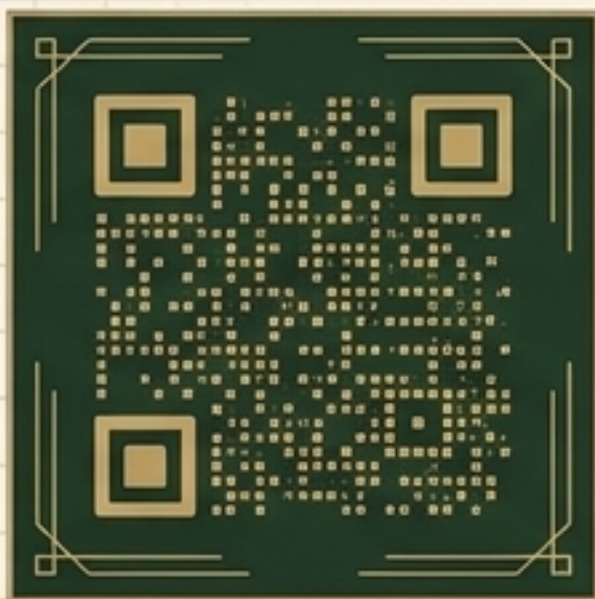
企業級品質與端到端驗收標準



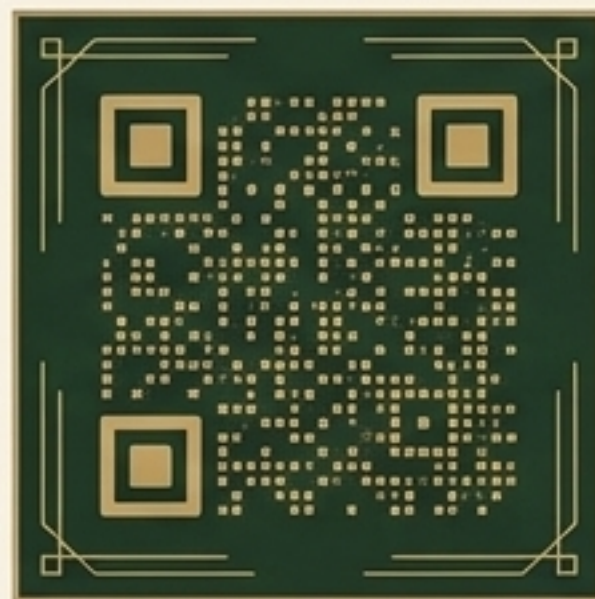
✓	[A02] 絕對資料隔離	驗證會員無法透過篡改 ID 越權讀取他人流程與結果。
✓	[A04] 嚴格 Skills 治理	驗證未上架、遭停用或雜湊不符的 Skill 絕對無法被 Invoke。
✓	[A08] 背景排程韌性	驗證關閉瀏覽器與伺服器重啟後，排程與 Worker 仍能精準觸發。
✓	[A10] 企業級秘密保護	驗證 Ollama API Key 與機敏資料絕不外流於前端 JSON、瀏覽器或系統日誌。

結語：重塑工作流的未來

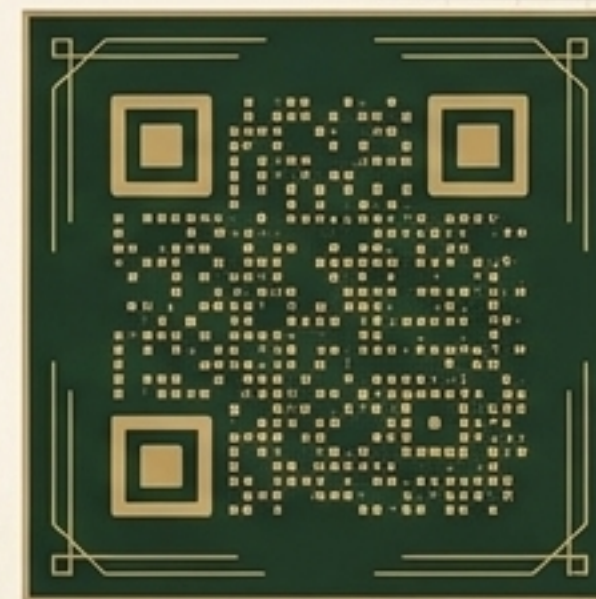
Workflow Atelier 不僅是一門 24 小時的實戰課程，更是掌握 AI 時代企業自動化核心架構的藍圖。它證明了「嚴謹的傳統工程邊界」與「靈活 AI 推論能力」能夠在 ASP.NET Core 架構下完美共存。



Vue Flow 開發文件



Agent Skills 規格書



MCP 官方標準