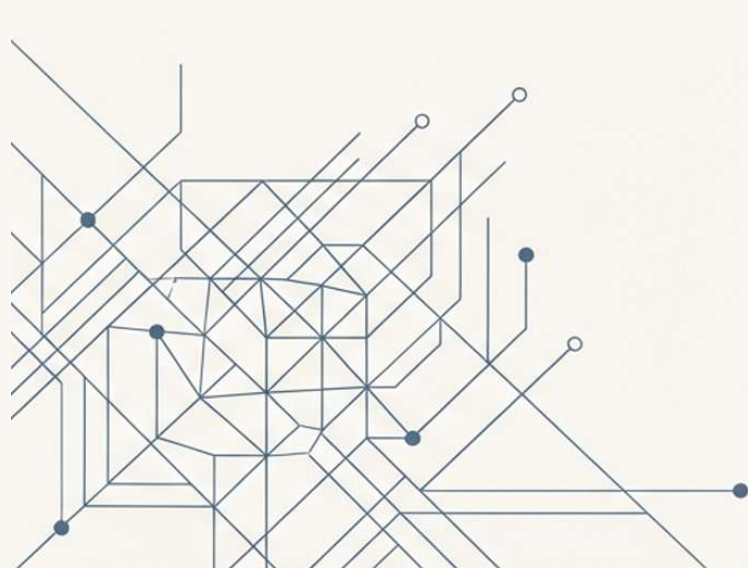


# AI 的進化： 從 RAG 到自主 Agent

從「隨身助理」到「自主執行者」的技術躍遷

報告人: 方欽賢 | 2025 年



# 報告大綱



## 1. 大腦與侷限

AI 的基礎能力與其根本缺陷。

## 2. 開卷考試

賦予 AI 外部記憶，解決知識侷限。

## 3. 精準查資料

優化 AI 的記憶檢索能力，確保準確性。

## 4. 手與腳

讓 AI 超越問答，具備執行任務的能力。

## 5. 實踐與未來

整合所有能力，展望 AI 的下一戰場。

# LLM：強大但健忘的大腦

LLM 是 AI 的核心，擁有強大的語言理解與生成能力，  
但基礎存在兩個致命缺陷，使其無法單獨應對真實世界的任務。

## ▲ 兩大缺陷



### 知識停滯 (Static Knowledge)

模型訓練完成後，其知識庫便不再更新。



### 幻覺 (Hallucination)

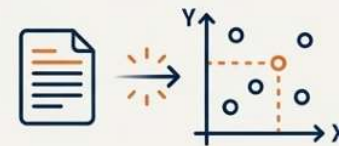
當模型不知道答案時，傾向於捏造看似真實的資訊，也就是「一本正經說瞎話」。

## 解決方案：給 AI 開卷考試

RAG (檢索增強生成)讓LLM 在回答前先「查資料」,從根本上解決知識停滯與幻覺問題。

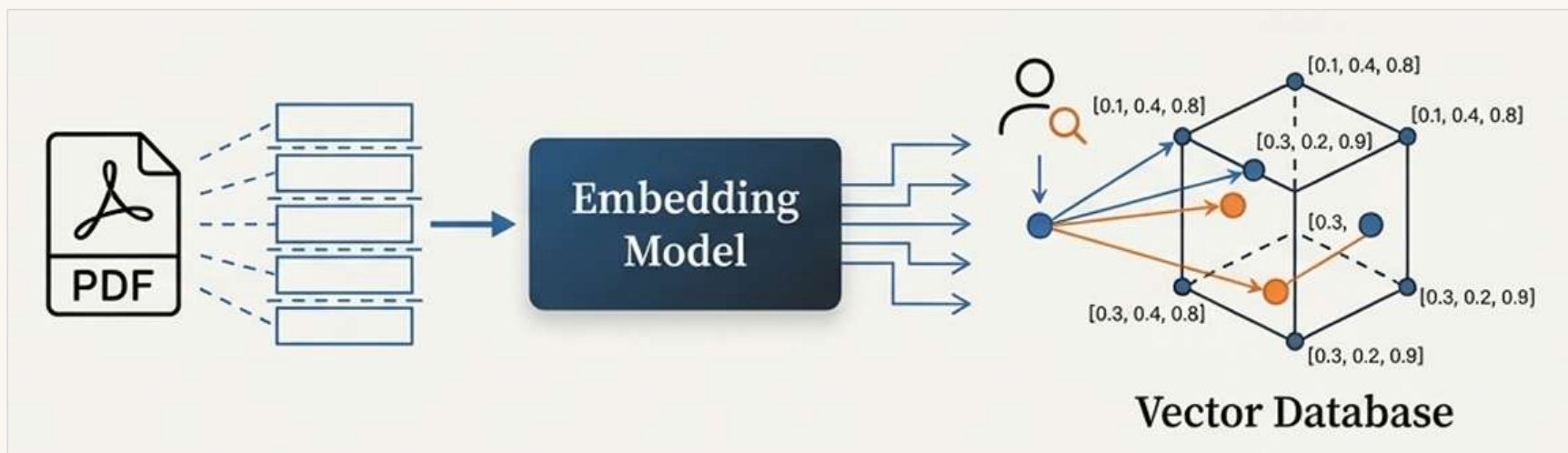


**關鍵機制** 知識透過 Embedding 技術被翻譯成「數學座標」，存儲在向量資料庫中,實現基於「意義」的快速查找。



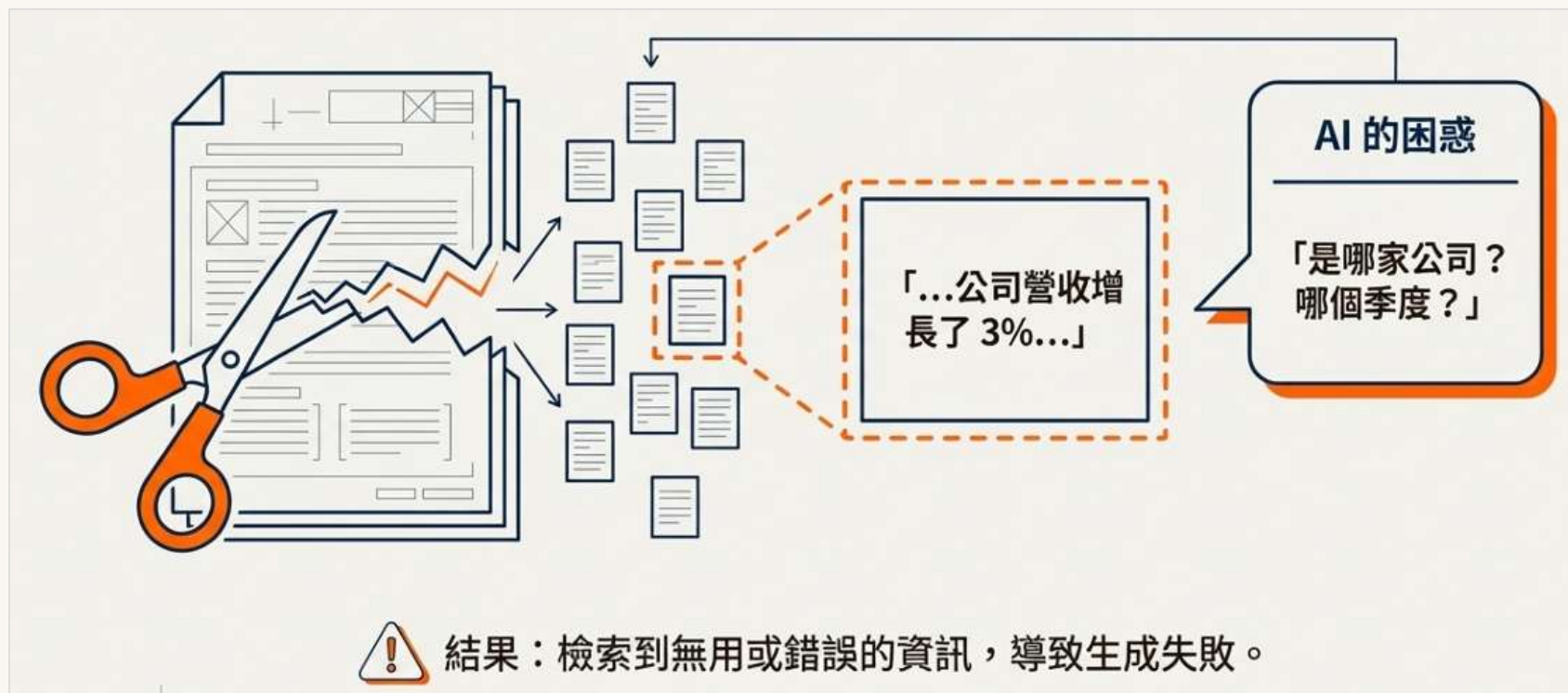
# Embedding：將「知識」翻譯成「座標」

- ✓ 1. 切塊 (Chunking)  
將長文件切分成小段落。
- ✓ 2. 向量化 (Embedding)  
將文字轉化為多維數學座標。
- ✓ 3. 儲存 (Storage)  
存入向量資料庫 (Vector DB)。
- ✓ 4. 搜尋 (Search)  
計算座標距離 (Cosine Similarity) 尋找近鄰。



## 傳統 RAG 的痛點：上下文失憶

傳統 RAG 在處理文件時,會將其「粗暴地切塊」(Chunking),導致每個區塊內的資訊與其原始上下文脫鉤。



# 各種RAG方案：精準度 VS 實現成本

架構類型	實現複雜度	索引成本 (CapEx)	查詢成本/延遲 (OpEx)	維護難度
Naive RAG	低	低	低	低
Hybrid + Rerank	中 (需複雜重排引擎)	中	中 (需昂貴計算)	中
Agentic RAG	複雜 (需编排與 Prompt 工程)	低	極高 (串聯請求)	高 (Prompt 漂移)
GraphRAG	高 (需路徑引擎)	極高 (LLM 使用)	中	高 (需索引維護)
LazyGraphRAG	高	極低 (延遲計算)	中/高 (需路徑引擎)	中
LightRAG	中/高	低	低	低 (極易更新)





# 零程式碼 RAG 工具：Google NotebookLM

## 我的筆記本



### AI進化：從RAG到自主Agent架構

2025年11月22日 · 1 個來源



### Text-to-SQL技術與開源方案研究

2025年11月23日 · 9 個來源



### 混合搜尋：精準與廣泛的雙重保障

2025年11月22日 · 1 個來源



### 生成式AI與提示工程進階實踐

2025年11月22日 · 10 個來源



### RAG

2025年11月15日 · 9 個來源



上傳來源

請將檔案拖曳到這裡，或是選擇檔案上傳

支援的檔案類型：PDF, .txt, Markdown, 音訊 (例如 MP3), .avif, .bmp, .gif, .ico, .jp2, .png, .webp, .tif, .tiff, .heic, .heif, .jpeg, .jpg, .jpe

## 支援的檔案類型：

PDF, .txt, Markdown, 音訊 (例如 MP3), .avif, .bmp, .gif, .ico, .jp2, .png, .webp, .tif, .tiff, .heic, .heif, jpeg, .jpg, .jpe

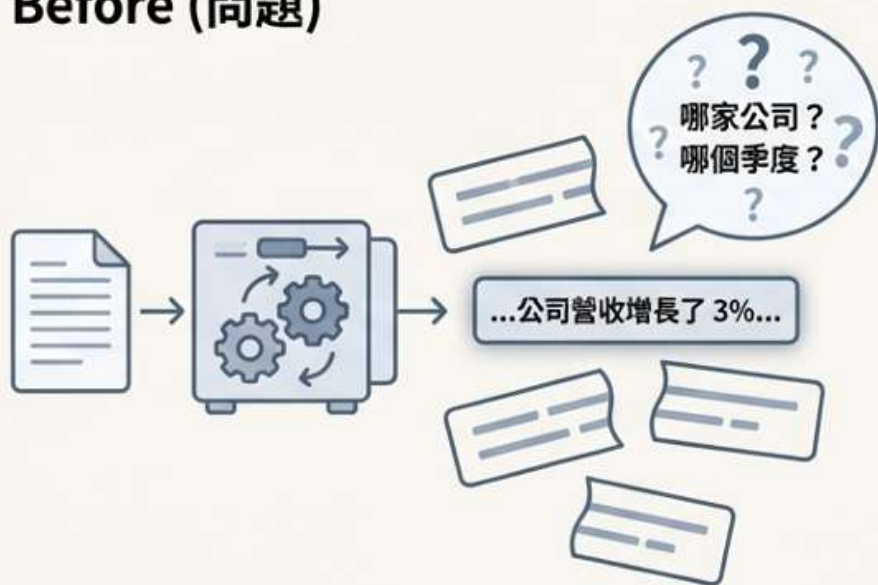
功能項目	免費版	Pro 版
可建立筆記本數量	最多 100 個	最多 500 個
每個筆記本的來源上限	50 個來源	300 個來源
每個來源字數上限	50 萬字	50 萬字 (相同)
每日對話查詢次數	50 次	500 次
每日語音生成次數	3 次	20 次
每日影片生成次數	3 次	20 次



# 手刻 RAG 改善方案一：Contextual Retrieval

方法原理: 從「上下文失憶」到「情境檢索」

## Before (問題)



傳統的「粗暴切塊」會導致關鍵資訊遺失,造成「上下文失憶」。

## After (解決方案)



為每個區塊「注入上下文」,讓其自帶說明書。根據Anthropic 的研究,此方法可使檢索失敗率降低49%。

# 手刻 RAG 改善方案一：Contextual Retrieval

實現工具包: LLM + N8N + Supabase

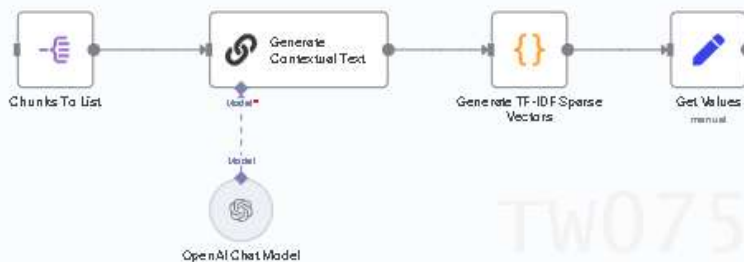
## 1. Import Document PDF



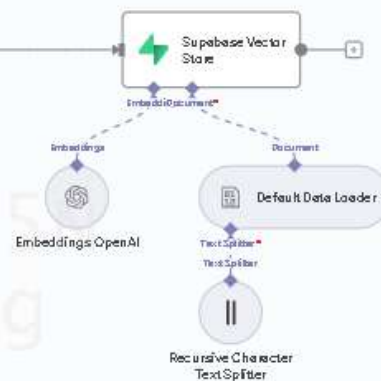
## 2. Split Document Into Chunks



## 3. Generate Sparse Vector and Contextual Text For Chunk



## 4. Insert Docs to Qdrant (via Langchain Code Node)



## 手刻 RAG 改善方案二：混合搜尋 (Hybrid Search)

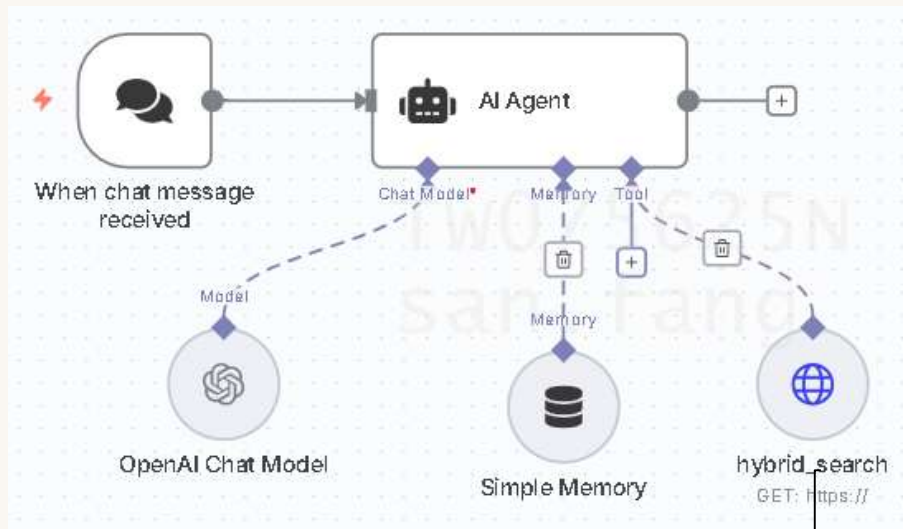
方法原理: 關鍵字 vs 語意? 能不能兩者都要

僅依賴語意搜尋(Semantic Search)是不夠的。某些查詢需要絕對的精確度。



## 手刻 RAG 改善方案二：混合搜尋 (Hybrid Search)

實現工具包: LLM + N8N + Supabase



### Definition

The language below should be written in sql.

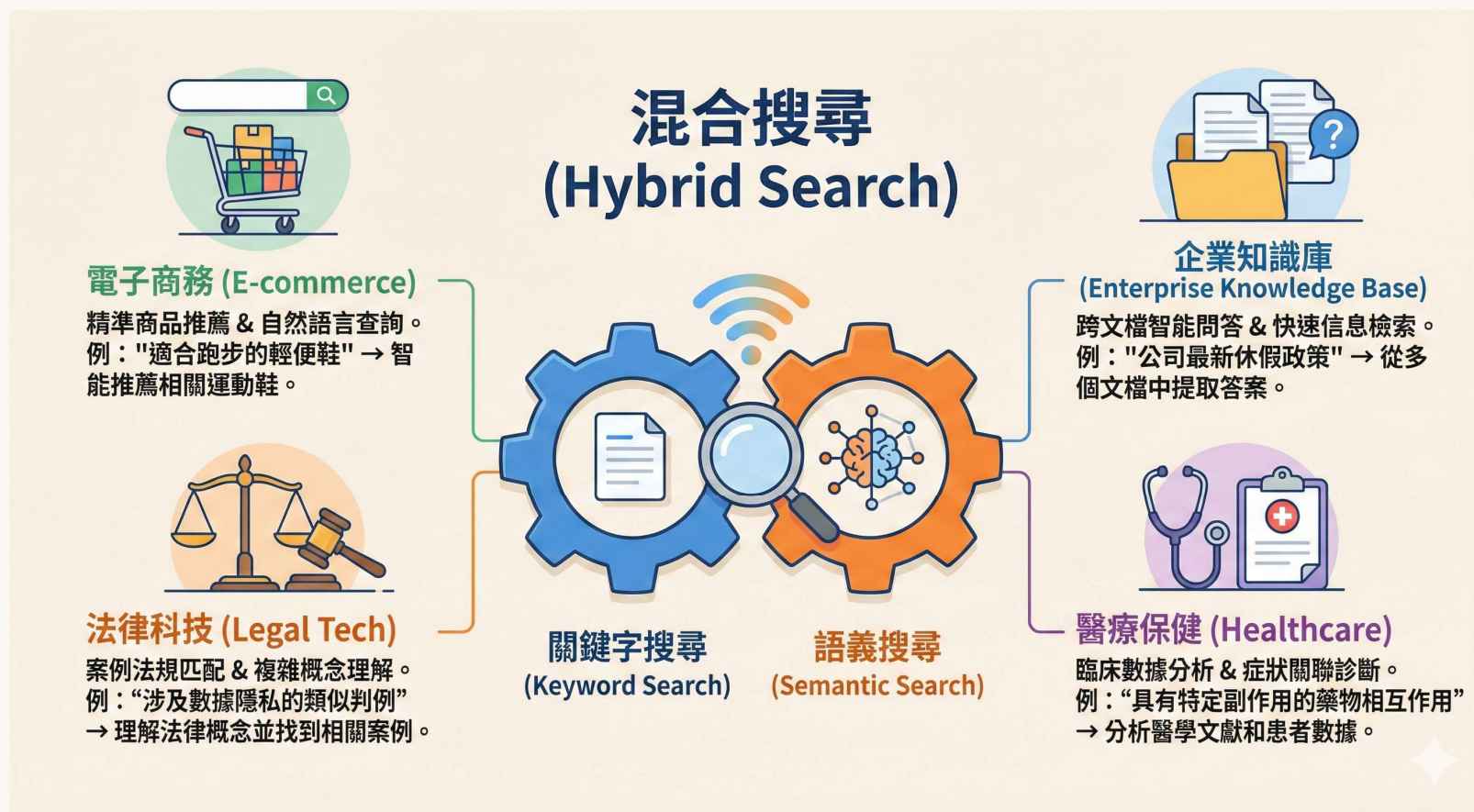
```
1
2 with full_text as (
3   select
4     id,
5     -- Note: ts_rank_cd is not indexable but will only rank matches of
6     -- the where clause
7     -- which shouldn't be too big
8     row_number() over(order by ts_rank_cd(fts, websearch_to_tsquery
9       (query_text)) desc) as rank_ix
10  from
11    documents
12  where
13    fts @@ websearch_t
14    and metadata @> fi
15  order by rank_ix
```

建立 Hybrid Search  
自動產生 API

### INVOKE FUNCTION

```
let { data, error } = await supabase
  .rpc('my_hybrid_search', {
    filter,
    full_text_weight,
    match_count,
    query_embedding,
    query_text,
    rrf_k,
    semantic_weight
  })
if (error) console.error(error)
else console.log(data)
```

# 混合搜尋 (Hybrid Search)：應用場景



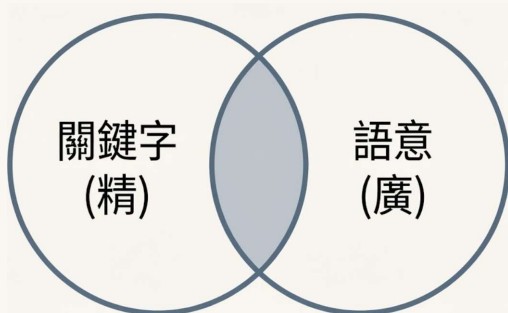
# 精準檢索: AI 的雙重搜尋策略

為了應對不同類型的資料, AI需要結合多種檢索工具。

## 針對非結構化文本(如 PDF, TXT)

### 策略：混合搜尋 (Hybrid Search)

結合關鍵字搜尋(Keyword)的「精」與語意搜尋(Semantic)的「廣」,實現「精確匹配」與「相關推薦」的雙重保障。



## 針對結構化數據(如資料庫)

### 策略：Text-to-SQL

將「人話」的自然語言查詢,直接翻譯成「SQL程式碼」,讓AI 能夠讀懂並分析資料庫中的表格數據。





## 結構化資料庫:Text-to-SQL 讓AI 讀懂資料庫

AI 的能力不應侷限於讀懂 PDF或網頁。Text-to-SQL技術讓AI 能夠直接查詢結構化資料庫,徹底解鎖其數據分析潛力。

 輸入 (用戶語言)

「上個月誰是我的超級VIP？」

過程  
(AI 翻譯)



```
SELECT user, SUM(revenue)
FROM sales
WHERE month='last'
GROUP BY user
ORDER BY SUM(revenue) DESC
LIMIT 1;
```



 輸出 (精確答案)

「是**王先生**，他上個月消費了**\$50,000 元**。」



## 從數日到數秒:數據查詢的進化

### 過去的低效循環



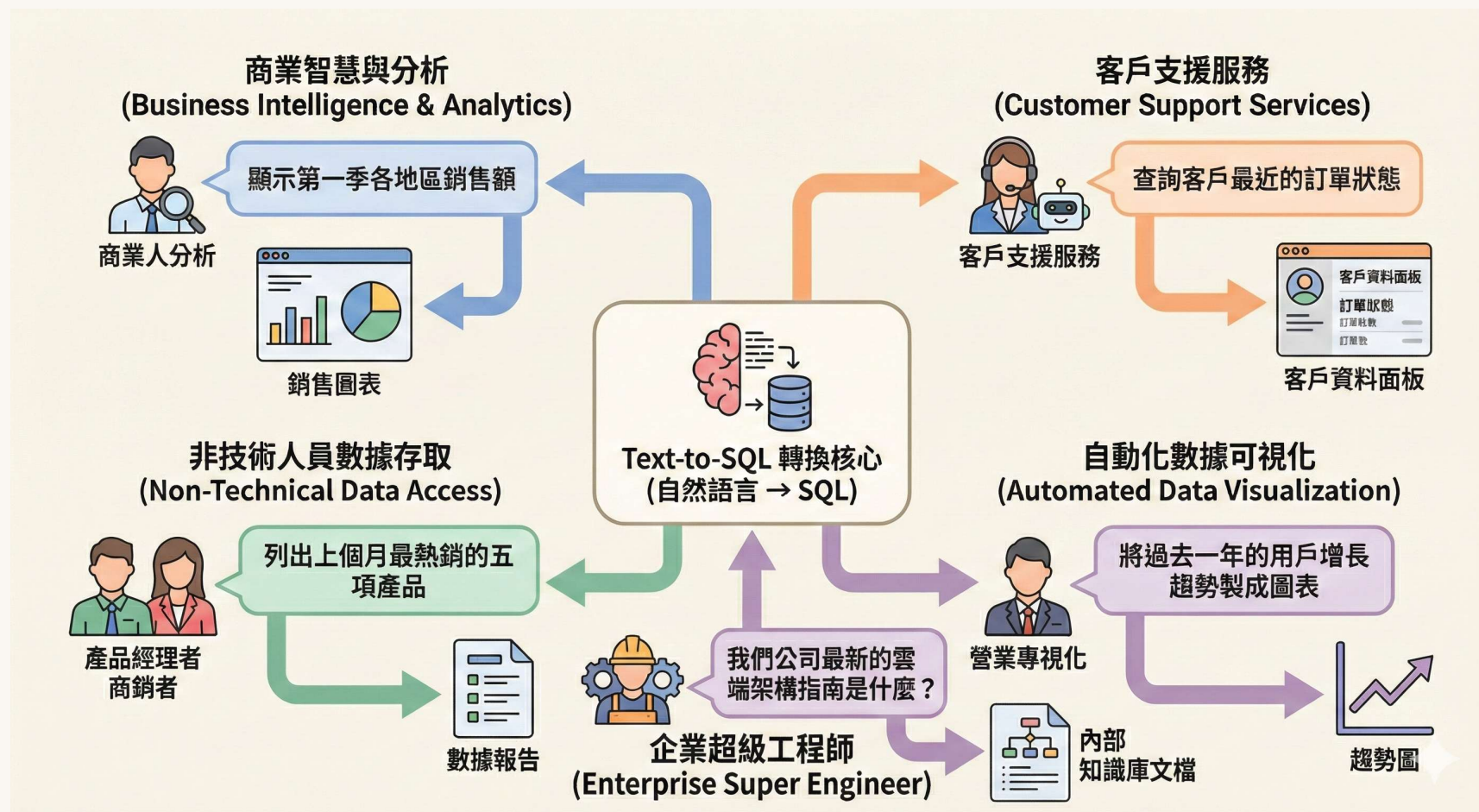
「過去，領導、決策者或業務人員想要統計個數據、做個報告，總是繞不開技術人員的支持……整個過程非常低效！」

### 未來的即時對話



**數據民主化：賦能非技術人員**

# Text-to-SQL : 正在賦能千行百業



# | Text-to-SQL：超級工程師

實現工具包: LLM + Python+ Nodejs + 結構化資料庫

歷史對話：

目前對話 ▾

清除

知識助理：

智能/建模/AI專案 ▾

智能/建模/AI專案

MAERB

前段效益管理平台

PM知識助理

深根報告

ILS OUI DL人力

異常通報

Hello, 我是智能/建模/AI專案知識助理，有甚麼可以協助您的？

範例一：請整理AOI專案內容

範例二：用表格呈現八大應用領域下的專案數量

範例三：列出知識管理的專案、負責人、時程、效益

範例四：智能專案數量

請輸入您的問題...

複製內容

下載CSV

送出

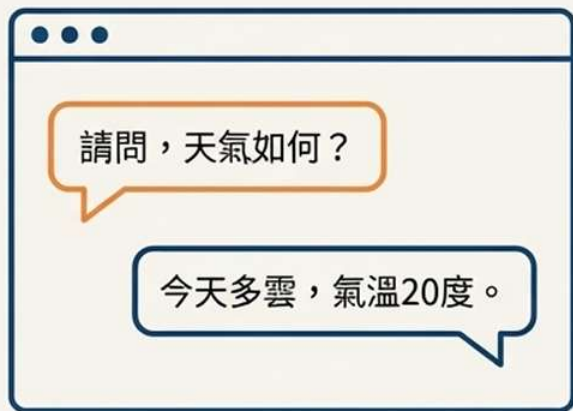
開新對話

# AI 的下個進化:從「聊天」的AI到「做事」的Agent

Agent = LLM（大腦）+ Tools（手與腳）

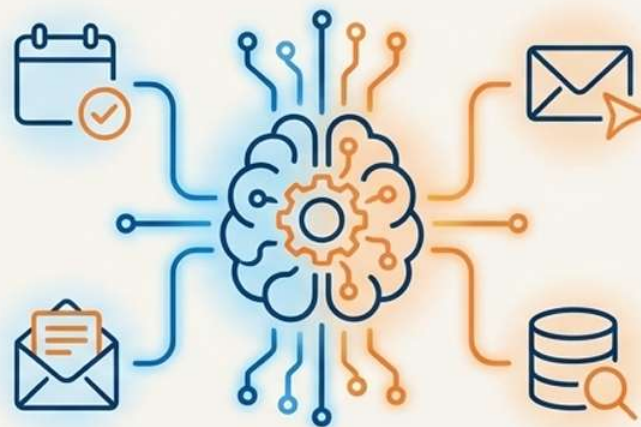
## 傳統 GenAI

你問，我答 (You ask, I answer)



## AI Agent

你給目標，我搞定 (You give goal, I get it done)



# Agent 如何使用「工具」？語言與協定

## 指令語言- 函數呼叫 (Function Calling)

Agent 使用工具的技術基礎, 是一種比提示詞 (Prompt) 更穩定、更結構化的 API 請求。

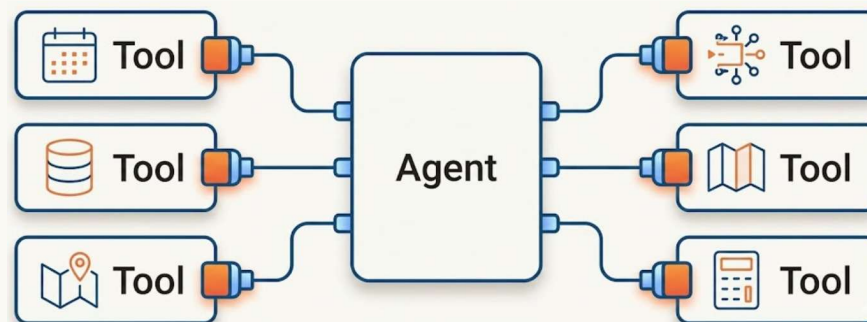


## 通用協定-模型上下文協定 (Model Context Protocol)

MCP 是「AI 的 USB」 

**解決的痛點：**「每家公司的工具/API 格式都不同, 整合困難。」


**MCP 的作用：**「建立 Agent 和 Tool 之間的「通用語言」, 讓任何 Agent 都能「隨插即用」任何工具, 實現跨平台的工具生態系。」



# MCP 範例一：圖表生成工具 ECharts

OI gpt-4.1 今天 09:47

這是你所需要的半年財報曲線圖連結：

👉 點我看半年財報數據曲線圖 

圖表中，你可以清楚看到未來新股份有限公司在2024年上半年，各項財務指標如營收、毛利、淨利等的走勢。從Q1到Q2的明顯成長，證明公司策略奏效！

如果你需要不同項目分開呈現，或是想要更多細項數據，也可以再告訴我喔！👏

1 個來源



< 工具 3

🔑 PPC\_達產



🔑 圖表工具



傳送 🔑 Chrome DEV TOOLS



+  🔑 1

) 分別做一張單獨的曲線圖嗎？

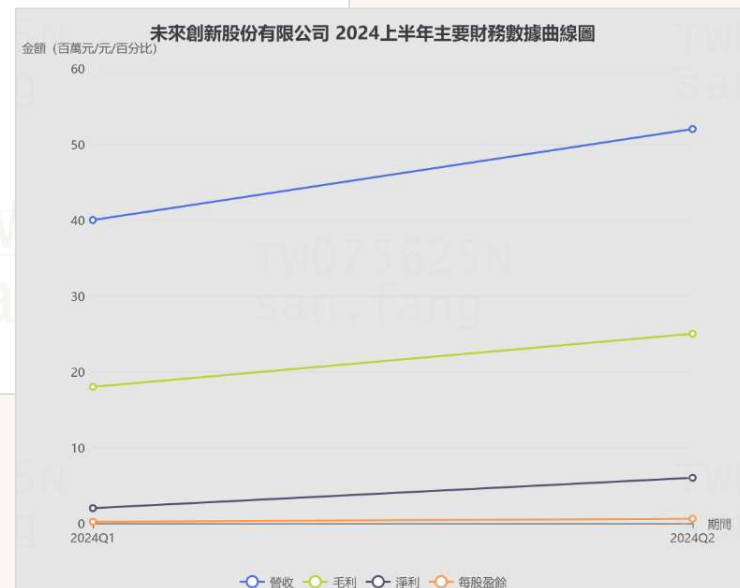


較嗎？

Apache ECharts

An Open Source JavaScript Visualization Library

[Get Started](#) [Docs](#) [Icons](#)



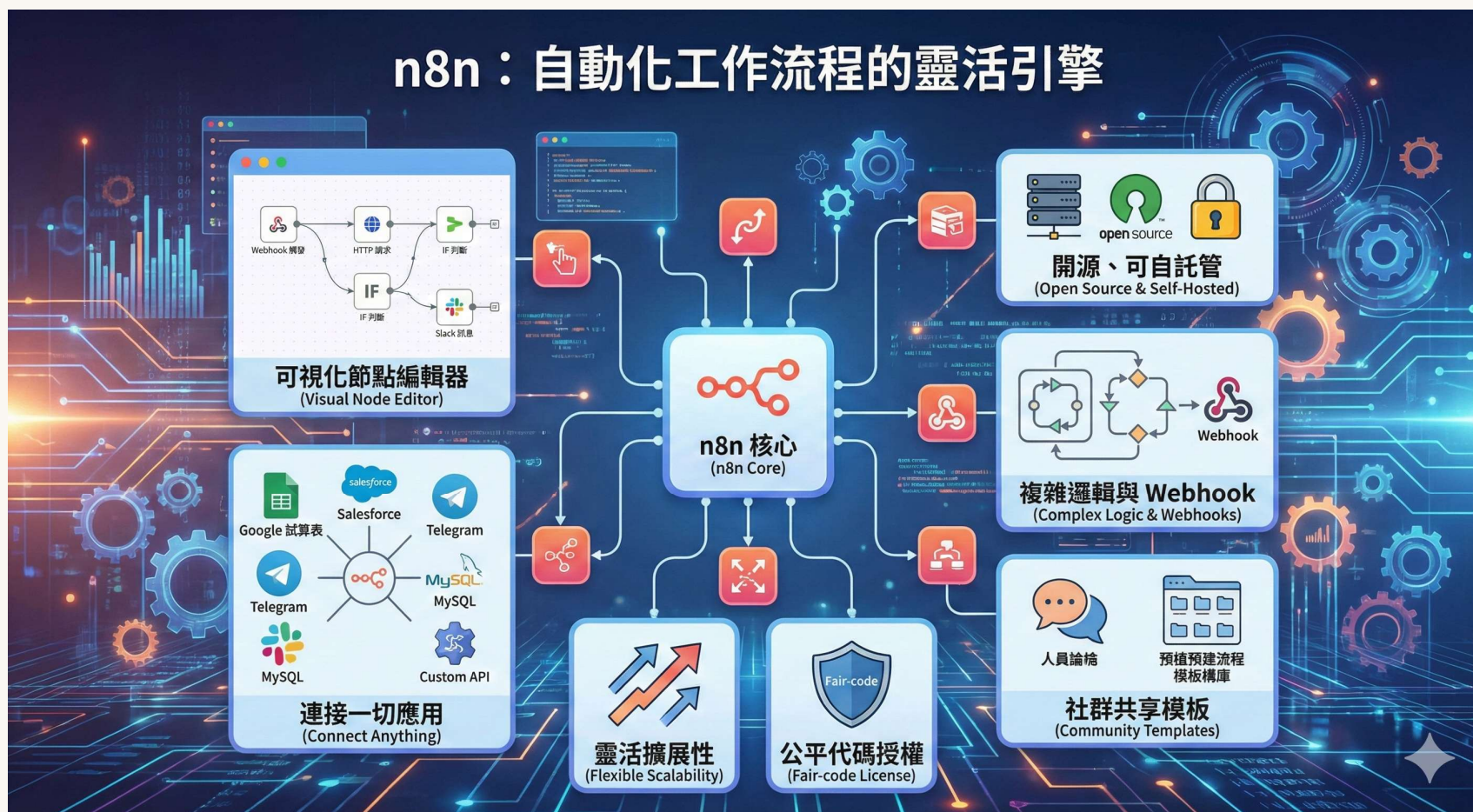
## | MCP 範例二：待補



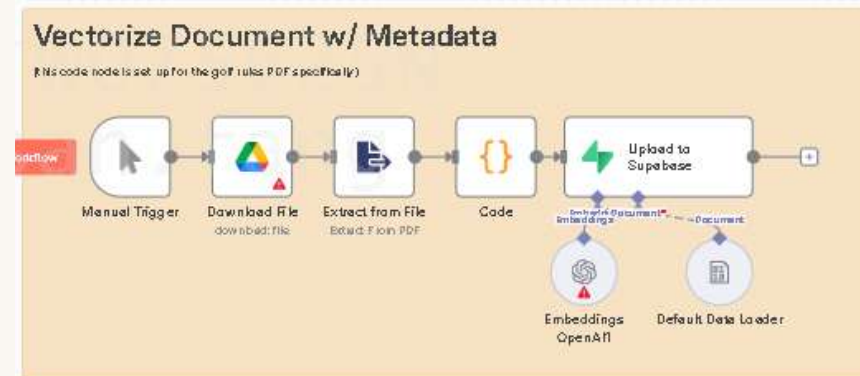
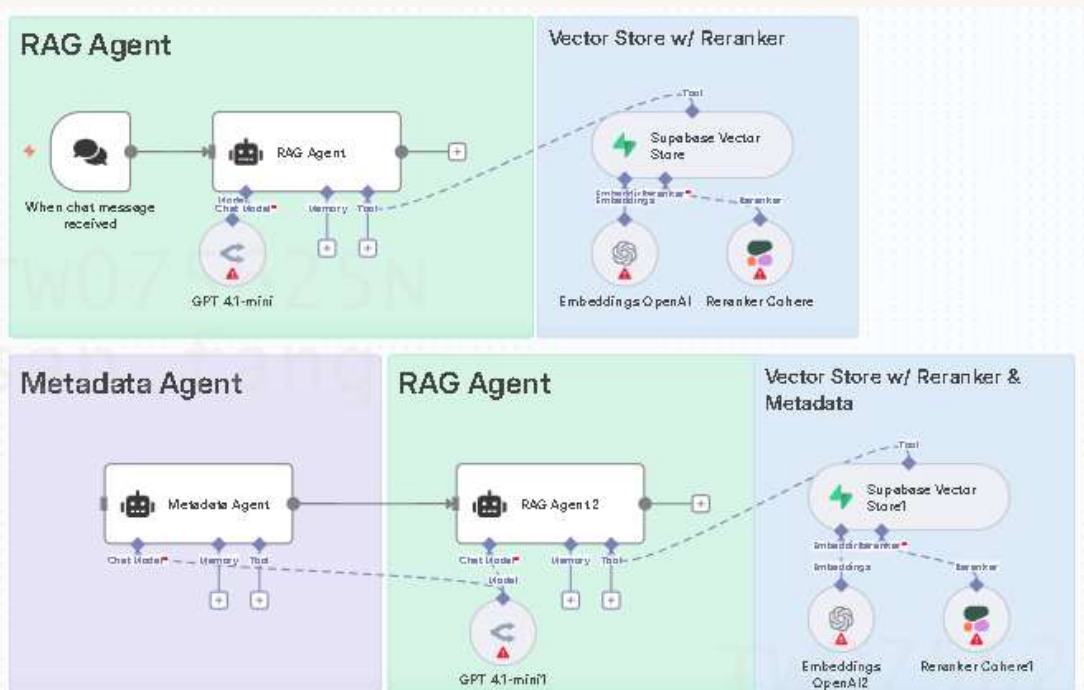
# AI Agent：從「知識庫」到「行動者」的進化路徑



# AI Agent 工具：N8N 作業流程自動化工具特點



# 實作落地一：用 N8N 組裝自己的知識小助手



## | 實作落地二：待補

# 未來展望：自主生產力時代

**RAG**

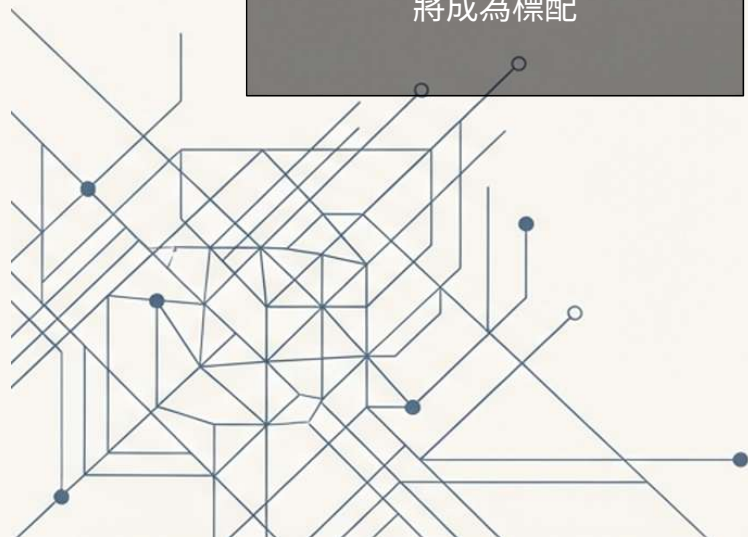
將成為標配

**Agent**

將無所不在

**MCP**

生態系護城河



# Q & A

歡迎提問與交流