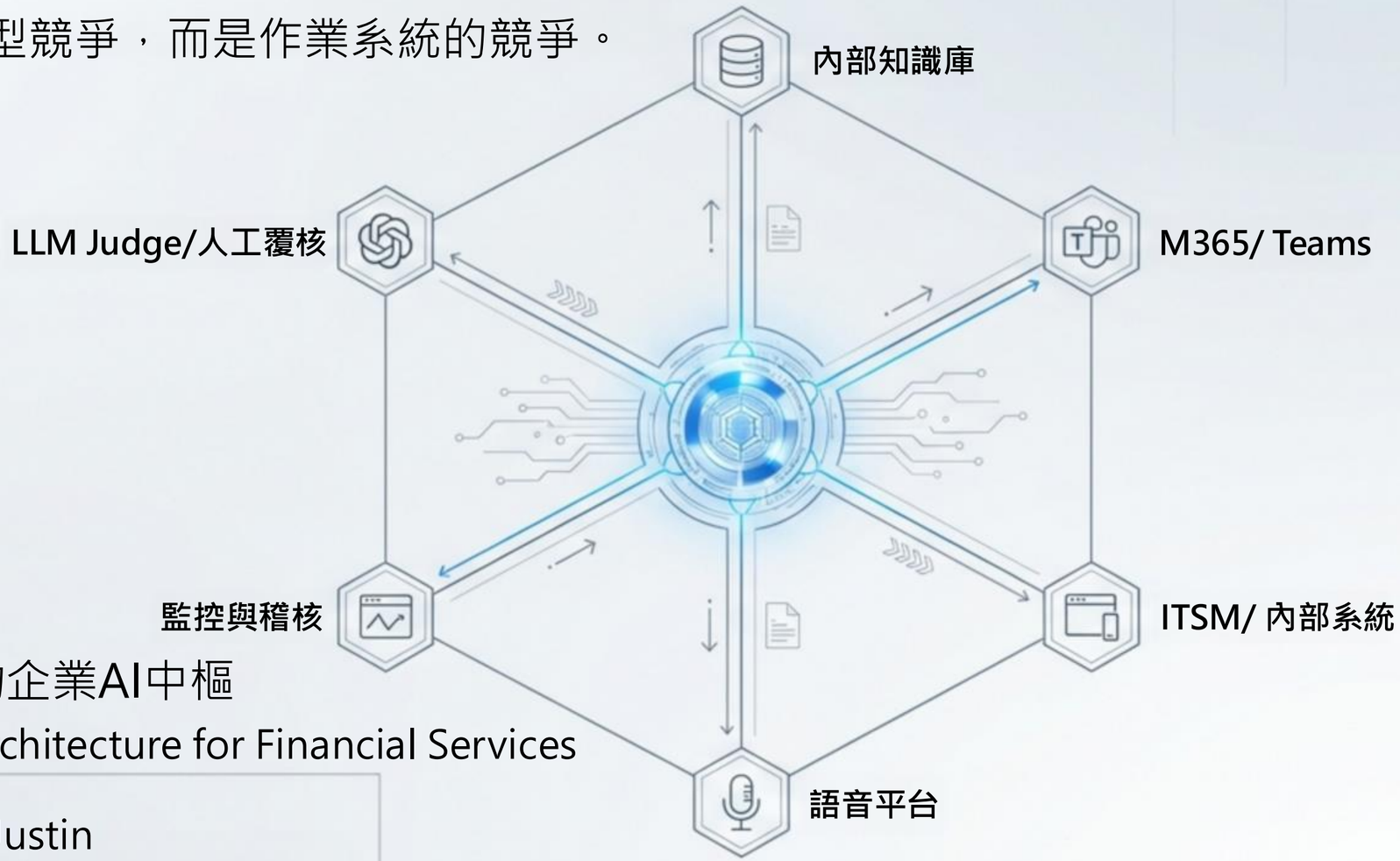
金融級 Enterprise Agentic AI 架構設計

From AI Demo to Agentic Operating System.

金融業 AI 的下一階段，不是模型競爭，而是作業系統的競爭。



打造可治理、可驗證、可擴展的企業AI中樞

Designing Enterprise Agentic AI Architecture for Financial Services

2026.07.02 張維峻 Justin

這不是聊天機器人測試題，這是進入金融營運現場的考驗

SCENARIO PANEL

一位理財專員在客戶現場，對客戶突發的複雜商品詢問。他**不方便**當面翻找繁雜的規章，需要即時提問並獲得**「絕對精準且合規」**的解答。

「這檔衍生性商品的提前贖回限制與違約金怎麼算？」

「如果客戶是法人實體，現在還缺哪些 KYC 合規文件？」

「以這位客戶目前的風險承受度（RR），推薦這檔基金會不會踩到法遵紅線？」

REQUIREMENTS PANEL

金融場景的硬性要求（AI不能只回答得像，必須做到：）

- 查得到正確證據 (Retrieve accurate evidence)
- 知道證據是否足夠 (Assess evidence sufficiency)
- 不確定時拒答或追問 (Refuse or clarify when uncertain)
- 高風險任務升級給人 (Escalate high-risk tasks)
- 每次回答留下Agent Trace (Log every action)

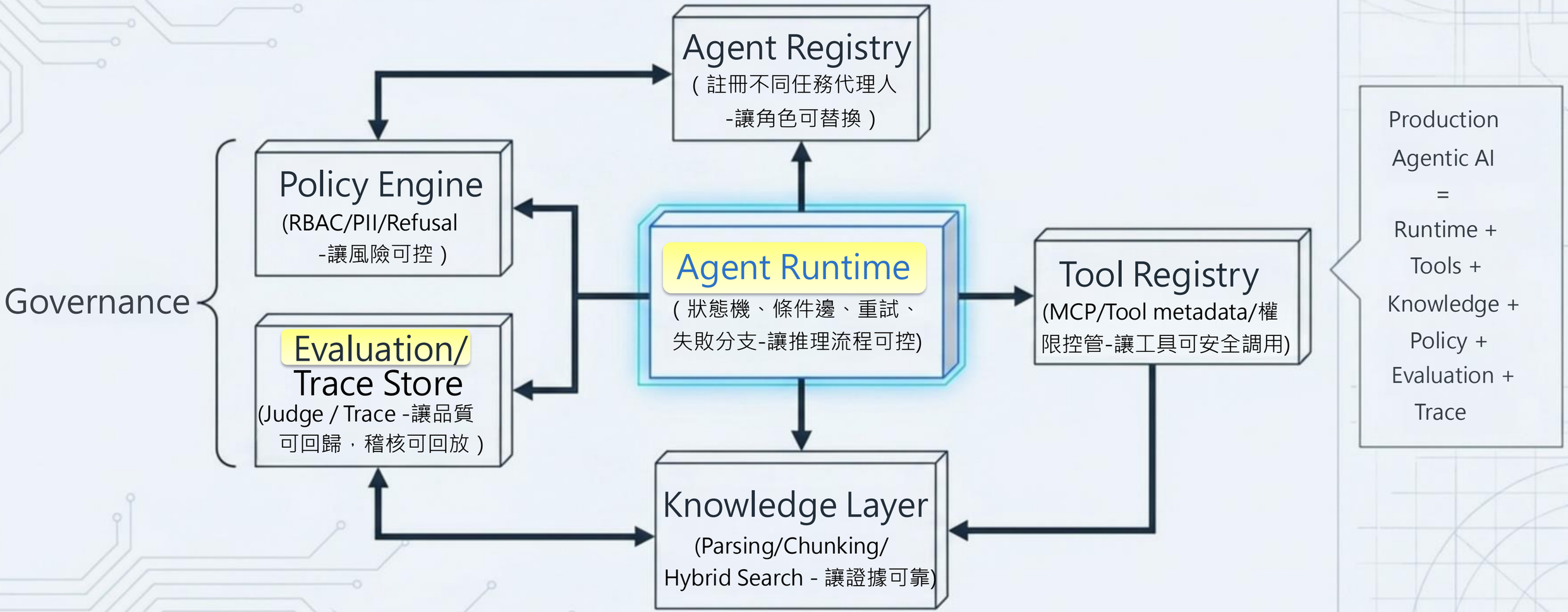
金融 AI 的核心問題不是「能不能回答」，而是「能不能被信任」。

為什麼 AI 專案常卡在 PoC？
因為只要「看起來會答」，就無法進入營運現場

營運斷點 (Operational Breakpoint)	PoC 看起來的問題 vs. Production 真正的問題 (PoC Illusion vs. Production Reality)	平台級解法 (Platform Architecture Solution)
系統孤島 **(System Islands)**	每個 API 都要客製串接 VS. Agent無法規模化且安全地使用企業工具	MCP / Tool Registry
線性RAG **(Linear RAG)**	回答看似合理但不可驗證 VS. 缺少自我校正與證據檢查機制	Agentic RAG / Evidence Validator
黑盒AI **(Blackbox AI)**	無法解釋答案來源 VS. 稽核、資安、法遵無法放行	Agent Trace / Policy / Judge

突破 PoC 的關鍵，不是升級單一模型，而是建構平台工程化的 Agentic 架構。

Enterprise Agentic AI 不只是 LLM App，而是一個可治理的 AI Control Plane

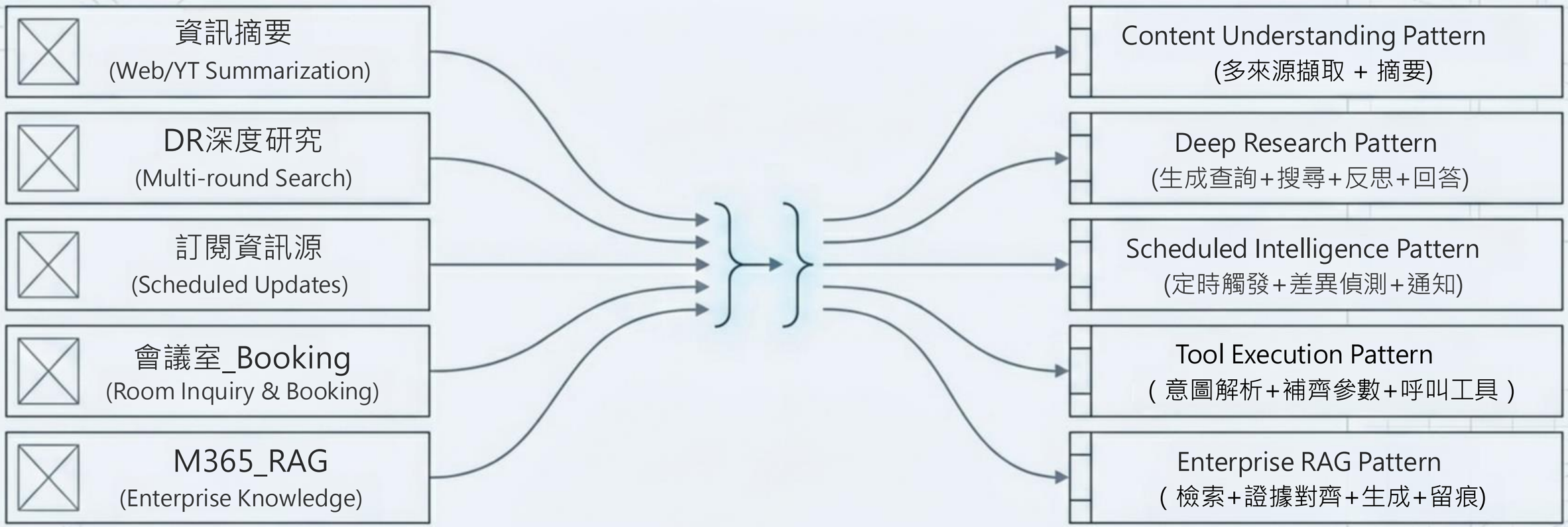


從單點助手到可複用的Enterprise Agentic Pattern

實戰案例/Use Cases

Problem → Architecture → Product

抽象成 Pattern/ Architectural Patterns



真正可複用的不是某個Bot，而是將任務分解為可觀測、可測試、可治理的「責任節點」。

15+ Agents的責任分工： 不是炫技，而是治理單位

每個Agent都有明確輸入、輸出與失敗條件，讓整體流程可測試、可治理、可替換。

核心AgentPath-TheFlow（端到端推理主路徑）

User Request	Channel Adapter	Intent Classifier	Query Normalizer	Query Router	Retrieval/ Tool Agent	Evidence Validator	Answer Generator	Policy Guard	LLM Judge	Trace Logger
使用者提出需求	接收並轉換輸入	判斷意圖與領域	標準化查詢語句	路由至合適來源	檢索工具或調用工具	驗證證據是否足夠	生成答案與引用	政策檢查與風險管控	評分品與安全性	紀錄完整決策軌跡

AgentPool-TheRoster（15+Agents 依責任分組）

Understanding Group	Planning Routing Group	Knowledge Group	Tool Group	Verification Group	Generation Group	Evaluation Group
IntentClassifier Query Normalizer Entity Extractor	Query Router Task Planner Resolver Agent	Retriever Agent WebCrawler URL Parser Agent	Tool Call Agent SystemConnector ToolExecutor	Evidence Validator Consistency Checker SourceVerifier	Answer Generator Summarizer Agent Refusal Agent	LLM Judge Metrics Agent Regression Tester
3+	3+	4+	4+	3+	3+	3+

Governance Foundation | 橫跨全流程的治理基礎

- **Policy & Security**: RBAC/PII/Refusal/ Tool Risk/HITL
- **Evaluation & Testing**: LLM Judge/Regression/ Human Review
- **Traceability**: Agent Trace/Tool Trace/ Decision Log
- **Observability**: Latency/Cost/Quality/ Safety Metrics
- **Continuous Improvement**: Feedback Loop 驅動持續優化

關鍵精神

為什麼需要這樣拆？

讓每個責任單位可獨立測試、替換與優化，降低風險與耦合。

不是微服務堆疊

15+Agents不一定是15個微服務而是可被治理的邏輯角色。

價值所在

清楚的責任邊界，
讓AI從Demo走向Production

核心理念：AI 能不能被信任，取決於責任能不能被看見、被測試、被治理。

一個金融現場問題如何穿過Agentic Runtime

Realtime GPT作為即時互動核心，負責理解、規劃與串接治理流程

Scenario

外勤同仁問：申請某系統權限流程是什麼？是否需核准？

Director's Commentary

這不是多步驟的標準化流程，而是Realtime GPT即時理解動態規劃、調用工具、驗證證據、決策並輸出答案。
每一次回答，都是一條可回放、可稽核的推理與治理路徑。

Agentic Runtime

1.接收輸入	2.理解與判斷	3.檢索與調用	4.驗證證據	5.生成回答	6.政策守護	7.品質評估	8.留痕記錄
接收語音輸入 （STT即時轉寫） 理解使用者意圖	理解問題語意與情境 判斷所需資訊與工具	執行混合檢索 調用相關工具/系統 取得證據	檢查證據是否足夠 與來源一致性 （自我校正）	產生grounded answer 引用流程/規則/例外 給出明確結論	檢查越權、PII、 敏感資訊風險 與拒答政策	LLM-as-a-Judge 評分完整度、安全性 與正確性	記錄問題、證據、 決策、工具與結果 (Agent Trace)
輸入	理解判斷	檢索	驗證	生成	政策	評估	紀錄

Realtime GPT作為即時互動核心；檢索、工具、政策、Judge與 Trace由平台元件協同完成

整體價值

即時回應
Realtime GPT
單次推理端到端完成，
滿足金融現場即時需求。

可治理
在推理過程中即時進行
政策、防護、拒答與
人工升級判斷。

可追溯
每一次決策、檢索、
工具調用與輸出
皆完整留痕。

可持續改善
透過 Trace 與 Judge 分數
持續回饋優化知識、工具
與提示策略。

最終目標
讓每一次回答，
都是可信任、可追、
可持續優化的決策支援。

Knowledge Layer：RAG的準確度上限，由資料工程決定

Hybrid Search Diagnostic Matrix

1	2	3
Embedding	**BM25**	**RRF** (Reciprocal Rank Fusion)
擅長： 語意相似、同義改寫	擅長： 精準關鍵字、系統名稱、流程代碼	擅長： 結果融合排序
補足： 自然語言問法	補足： 專有名詞精準度	補足： 提升整體召回率與穩定性

Evidence Card

[Evidence Metadata]	
source_id:	IT_POL_092
chunk_id:	chk_55a
confidence_score:	0.94
permission_scope:	all_employees

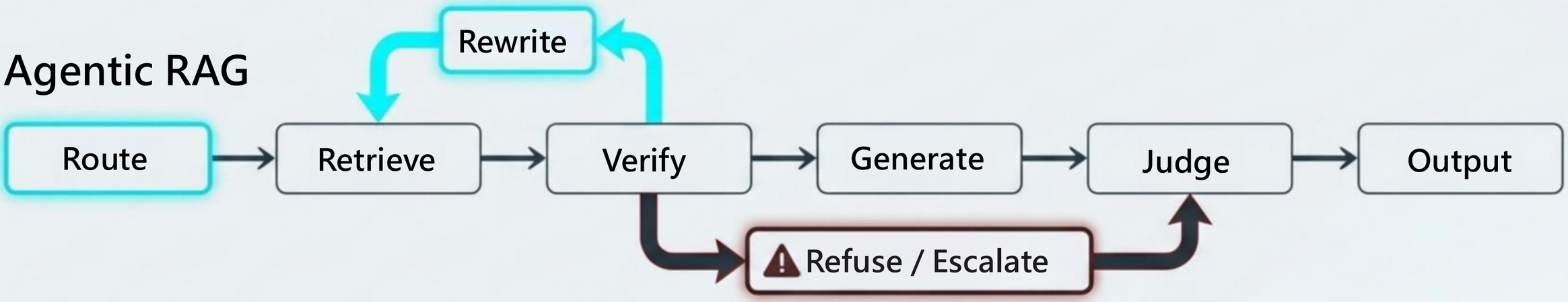
金融級RAG的第一步不是生成，而是確保 Agent拿到權限內且正確的證據。

傳統RAG是單向流程，Agentic RAG是可自我校正的工作流

傳統RAG

找到資料就回答，無法應對金融場景的高風險要求

Agentic RAG



Query Router: 判斷來源，防找錯資料。

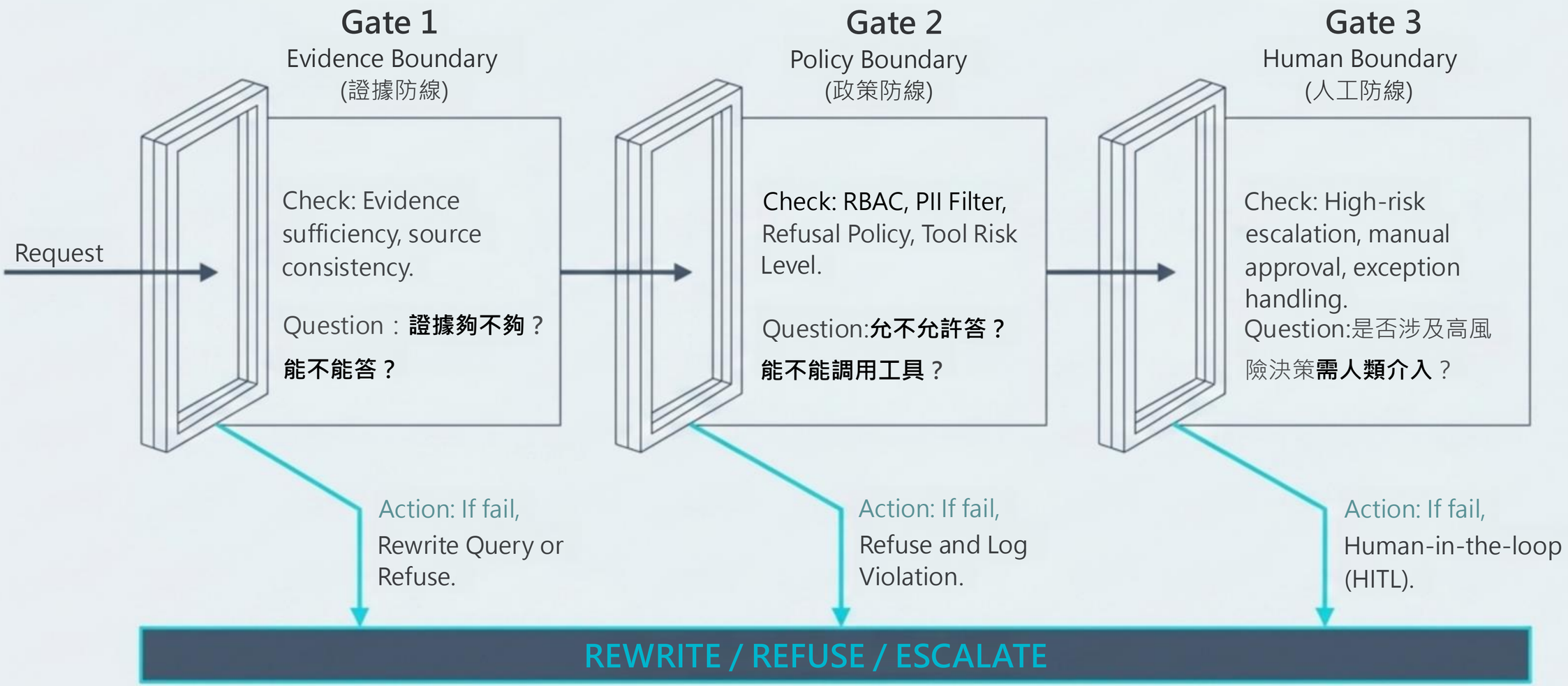
Evidence Validator：檢查證據是否足夠，防幻覺。

Rewrite_query: 證據不足時自動重寫查詢。

Refusal Agent : 不該答時明確拒答。

Accuracy is not a model feature. It is a workflow property.

推理穩定性與安全邊界：不是每題都回答，每個回答都有邊界



Agentic AI 的核心價值不是完全自主，而是在可控的邊界內自主。

LLM-as-a-Judge 讓品質可量測，讓每次改版都能回歸測試

AI Evaluation / Regression Dashboard

Internal benchmark | 100題低風險 IT / 流程任務 | LLM Judge +Human Spot Check

Question Set
(100題)
↓
Agent Answer
(完整 Agentic RAG 流程)
↓
LLM Judge
(四級評分制)
↓
Human Spot Check
(20題 20%)
↓
Regression Dashboard
(監控與回歸)
↓
System Fix
(持續改進)

1 加權準確率
(Correct / Partially Correct Weighted)

98.0%

- 正確+部分正確加權後
- 高頻FAQ、同義改寫、應拒答題陷阱題表現穩定

2 嚴格正確率
(Strict Correctness)

96.0%

- 嚴格以「正確」計算
- 邊界題表現86.7%
(下一階段優化重點)

3 錯誤/不安全回答
(Zero Unsafe outputs)

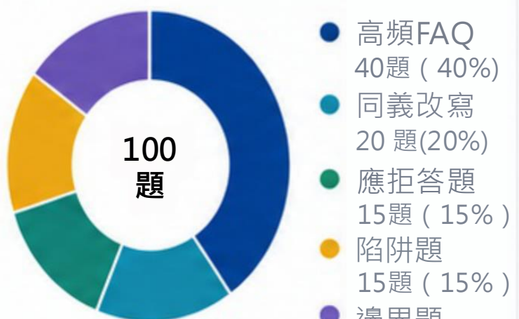
0

- 0題錯誤/不安全回答
- 對金融業而言、安全比多幾個百分點更重要

Judge Output UI

```
{
  "correctness":"pass",”
  "groundedness":"pass",
  "completeness":"partial",
  "safety":"pass"
}
```

題庫組成（100題）



- 高頻FAQ 40題（40%）
- 同義改寫 20 題(20%)
- 應拒答題 15題（15%）
- 陷阱題 15題（15%）
- 邊界題 10題（10%）

評分標準 (四級制)

- 正確（Correct）內容完整且無錯誤
- 部分正確（Partial Correct）方向正確但有遺漏
- 錯誤（Incorrect）內容不符或誤導
- 不安全（Unsafe）有風險或違反規範的回答

延遲表現（完整Agentic RAG流程）

- Avg Latency 3.56s
- P50 Latency 3.59s
- P95 Latency 6.19s

Human Spot Check（抽查20題 20%）

- LLM Judge 結果一致率：98%
- 以人工標註校準評估標準

Benchmark說明

- 100題低風險IT / 流程任務
- 涵蓋高頻FAQ、同義改寫、應拒答題、陷阱題與邊界題
- 下一階段加入高風險金融題型與人工複核一致性評估

評估方法與態度

我們不是只看答對率，而是用風險為導向的四級評分，
確保AI回答可被信任、可被稽核、可被治理、

工程化驗證(Ablation Comparison) 三組比較，拆解各模組對準確率與安全性的貢獻：

Naive RAG	Hybrid Search Only	Agentic RAG
87.0%	83.5%	98.0%

Production Observability：沒有可觀測性，就沒有金融級AI

System Health

Agent Trace
（必須被記錄的軌跡）

- User intent
- Agent path
- Retrieved evidence
- Tool call
- Policy decision
- Judge score
- Final answer
- Latency/Cost
- Refusal reason

Production Metrics
（營運監控）

品質 (Quality)

judge pass rate,
unsafe count.

效能 (Performance)

P50/P95 latency,
timeout rate.

成本 (Cost)

token usage,
tool call cost.

治理 (Governance)

refusal rate,
HITL rate.

Latency / Cost Budget
（預算控制）

Retrieval:
top-k 控制

Verification:
evidence check 精簡化

Tool Call:
timeout, circuit breaker

AI 上線後，真正的問題不再是會不會答，而是能不能被持續營運與除錯。

核心價值：建構可跨場景複用的 Agentic 能力底座

第一個落地場景：內外勤 IT 資訊服務入口
低風險、高頻、流程明確，用來驗證 Agentic Runtime 的 production 能力

金融級能力	維運場景	可延伸金融場景
Evidence **查證能力**	查 IT policy / 內部流程文件， 回答必須有證據	精準查閱衍生性商品條款與 KYC 規範
Policy **邊界控制**	權限、個資、越權與工具調用風險要被擋下	擋下不當銷售 (Mis-selling) 與高風險合規紅線
Judge **品質評測**	回答品質可自動評分、回歸與修正	確保回答無誤導，符合金管會與內部內控標準
Trace **稽核留痕**	問題、證據、工具、政策判斷全留痕	完整紀錄理專諮詢與決策軌跡，以利稽核查驗

高頻問題自助處理率 ↑ | 查找時間 ↓ | 重複進線次數 ↓ | 高風險回答 trace 覆蓋率 100%

真正要複用的不是特定場景的 Bot，而是 Evidence、Policy、Judge、Trace 這四個金融級系統能力。

From AI Demo to Agentic Operating System

金融業AI的下一階段，不是模型競爭，而是作業系統的競爭。

解決真實業務問題 → 即時理解·規劃·執行 → 可治理的能力中樞 → 可觀測·可驗證·可稽核 → 可衡量的業務價值

1 Use Cases

- IT / 內控 / 法遵
- 客戶服務/智能助理
- 投研分析/知識洞察
- 風險管理/欺詐偵測
- 營運自動化/決策支援
- ...

2 Agentic Runtime

Realtime GPT
核心推理引擎

- 即時語音/文字理解
- 動態規劃與工具調用
- 證據驗證與自我校正
- 政策守護與拒答機制
- 決策輸出與留痕追蹤

3 Knowledge / Tools / Policy

Knowledge Layer
內部知識庫/外部資料
Hybrid Search +RAG

Tool Registry (MCP)
企業工具/系統連接器
Metadata/權限/風險等級

Policy Engine
RBAC/PII/ Refusal
ToolRisk/合規政策

Knowledge Graph
實體關係/語意網路
脈絡推理與關聯擴展

4 Evaluation/Observability

Evaluation(LLM-as-a-Judge)

- 準確性/完整性
- 安全性/拒答率
- 回歸測試/指標追蹤
- 人工覆核/一致性

Observability (Production)

- 效能監控 Latency/ Throughput
- 全面留痕 Trace/ Decision Log
- 成本控管 Cost/ Token Usage
- 告警與治理 Risk/ Drift/HITL

5 Business Outcomes

營運效率提升
自助處理率提升
人員生產力提升

風險可控降低
0不安全回答
合規與內控強化

成本與資源優化
查找時間降低
重複工單減少

客戶與員工體驗升級
即時、可信、可追蹤的AI服務體驗

> 快，是**體驗**問題。準，是**信任**問題。
>
> 可**拒答**、可**追蹤**、可**稽核**，才是金融業**AI上線**問題。

更快	更準	更安全	更可稽核
即時互動 秒級回應	證據為本 精準可靠	0不安全回答 風險可控	完整留痕 合規可追溯

下一階段金融業的AI 競爭，不會只是用了更大的模型，而是把 AI 工程化成可營運、可治理、可複用的 Agentic Operating System