

## 1. 系統目標

針對 20 間目標企業自動化產出具備「財務洞察」、「語意摘要」與「供應鏈推論」的高品質研究報告，並透過 **Semantic Cache** 與 **Parallel Execution** 確保系統在有限 GCP 資源下達到極致性能。

---

## 2. 核心技術棧 (Tech Stack)

- **Orchestration:** LangGraph (Supervisor Pattern)
- **LLM:** Vertex AI (Gemini 2.5 Pro )
- **Data Warehouse:** BigQuery (Structured Data + Supply Chain Graph)
- **Data Lake:** Google Cloud Storage (Original PDFs & Images)
- **Memory / Cache (Local VM):**
  - 長期記憶:**Vector DB:** ChromaDB / FAISS (用於 Local RAG & Semantic Cache)
  - 短期記憶:**State DB:** SQLite (用於 LangGraph Checkpointer / 短期記憶)
    - 附註:Cache 用 Redis 可能是最高, 但考慮到規模不大, 不須處理高併發就先不往下挖深這部分.
- **Json Config File:**



(Graph 可以再討論作法 或是用pdf找參考資料給他也可以)

- Graph:
  - {
  - "nodes": [
  - {
  - "id": "2330",
  - "name": "TSMC",

```

■ "category": "Manufacturing",
■ "tags": ["Foundry", "Advanced Node"]
■ },
■ {
■   "id": "AAPL",
■   "name": "Apple",
■   "category": "Fabless",
■   "tags": ["Consumer Electronics", "Customer"]
■ }
■ ],
■ "edges": [
■   {
■     "source": "AAPL",
■     "target": "2330",
■     "relation": "Client",
■     "source_info": "Official_List_Semiconductor"
■   },
■   {
■     "source": "ASML",
■     "target": "2330",
■     "relation": "Supplier",
■     "description": "Lithography Equipment",
■     "source_info": "Official_List_Semiconductor"
■   }
■ ]
■ }

```

### 3. Agent 角色定義與分工

#### A. 管理與輔助層 (Orchestration & Utility)

Agent 名稱	職責說明	核心工具 (Tools)
<b>Root (Supervisor)</b>	任務調度中心。解析問題、拆解任務並分派給專家。	LangGraph Router
<b>Cache Gateway</b>	門衛節點。利用 <b>Local Vector DB</b> 進行比對，減少重複生成成本。	Local Vector DB

<b>Template Node</b>	格式化節點。將輸出填入 Markdown 模板，確保排版統一。	Python Formatter
----------------------	---------------------------------	------------------

## B. 專家執行層 (Specialized Experts)

Agent 名稱	職責說明	Instruction 重點	數據來源
<b>Financial Analyst</b>	提取財報數值，執行 QoQ/YoY 計算與趨勢分析。	專注數據準確度與財務比率解釋。	BigQuery SQL
<b>Q&amp;A Analyst</b>	檢索法說會 (Earnings Call) 文本。	專注經營層對未來展望 (Outlook) 的事實提取。	<b>Local Vector DB</b>
<b>News Agent</b>	整理近期重大新聞，過濾產能、地緣政治動態。	專注時效性與市場情緒 (Sentiment) 分析。	<b>Local Vector DB</b>
<b>Supply Chain Expert</b>	查詢供應鏈圖譜，推論上下游聯動關係。	專注風險傳導與關聯影響評估。	JSON Config (Graph)  A. 標註 Graph 節點關係來源。例如：APPLE / Fabless /

## C. 品質監控層 (Quality Assurance)

Agent 名稱	職責說明	Instruction 重點	數據來源
<b>Evaluation Agent</b>	執行 Guardrail 任務。確保無幻覺、引用來源明確。	Consistency / Citation	BigQuery + <b>Local Vector DB</b>  A. 交叉比對標籤。若為 BQ 標籤則發送

			SQL 核對；若為 PDF 標籤則比對本地文本。 <b>B. 全用 PDF 文件比對</b> 並且利用 RAG 的 Meta data 標註來源
--	--	--	--

## 4. 數據契約 (Data Contract - AgentState)

為確保 Agent 間溝通不失真，全局狀態 AgentState 定義如下：

Python

```
class AgentState(TypedDict):
    company_id: str      # 股票代號 (如 2330)
    basic_info: dict     # 公司基本 Profile
    finance_results: dict # 結構化財務數據
    rag_summaries: list   # 法說會與新聞摘要
    sc_analysis: dict     # 供應鏈推論結果
    validation_status: bool # 是否通過品質檢查
    final_report_md: str   # 最終渲染的 Markdown 文字
```

---

## 5. 功能特色

### 5.1 供應鏈圖譜推論 (Supply Chain Reasoning)

透過 Json config 儲存的 Graph 搜尋關係：

- 邏輯：Source → Target → Market
- 價值：當新聞提到 A 公司失火，Agent 能自動推論出對目標公司 B 的營收衝擊與影響。

### 5.2 語意快取機制 (Semantic Cache)

為了應對重複請求並節省 Token：

- **Similarity**: 比較 User\_Query 與 Cache 紀錄的 Embedding 相似度。
- **Thersholt**: 相似度 > 0.95 則直接調用快取。

---

備註：優先考量「資料一致性」與「可追蹤性」。

所有 LLM 的回答皆需附帶 source\_file 標籤，以滿足可靠性需求。