



國立臺灣科技大學
TAIWAN TECH
NATIONAL TAIWAN UNIVERSITY OF SCIENCE AND TECHNOLOGY

LangChain

Official Tutorials Basics part4. Agent

Agent 是什麼?



AI Agent

- 翻譯中文：AI代理，不太直觀
- 簡單來說：能夠自主決策並執行行動的系統
- 特點：善於處理非結構化的問題

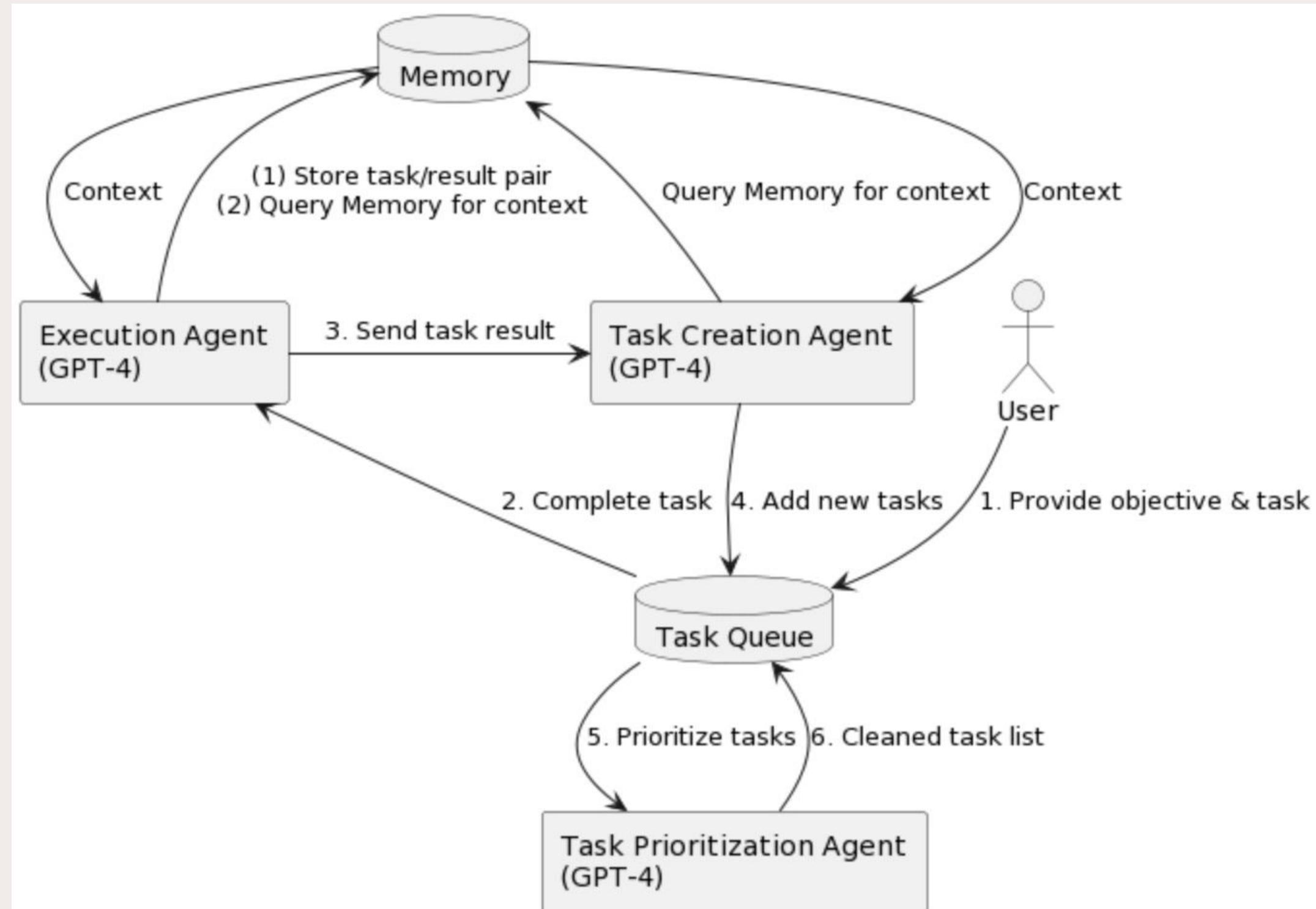
舉例：AutoGPT

前陣子剛發表就很紅的[AutoGPT](#)，旨在讓AI自動解決複雜問題

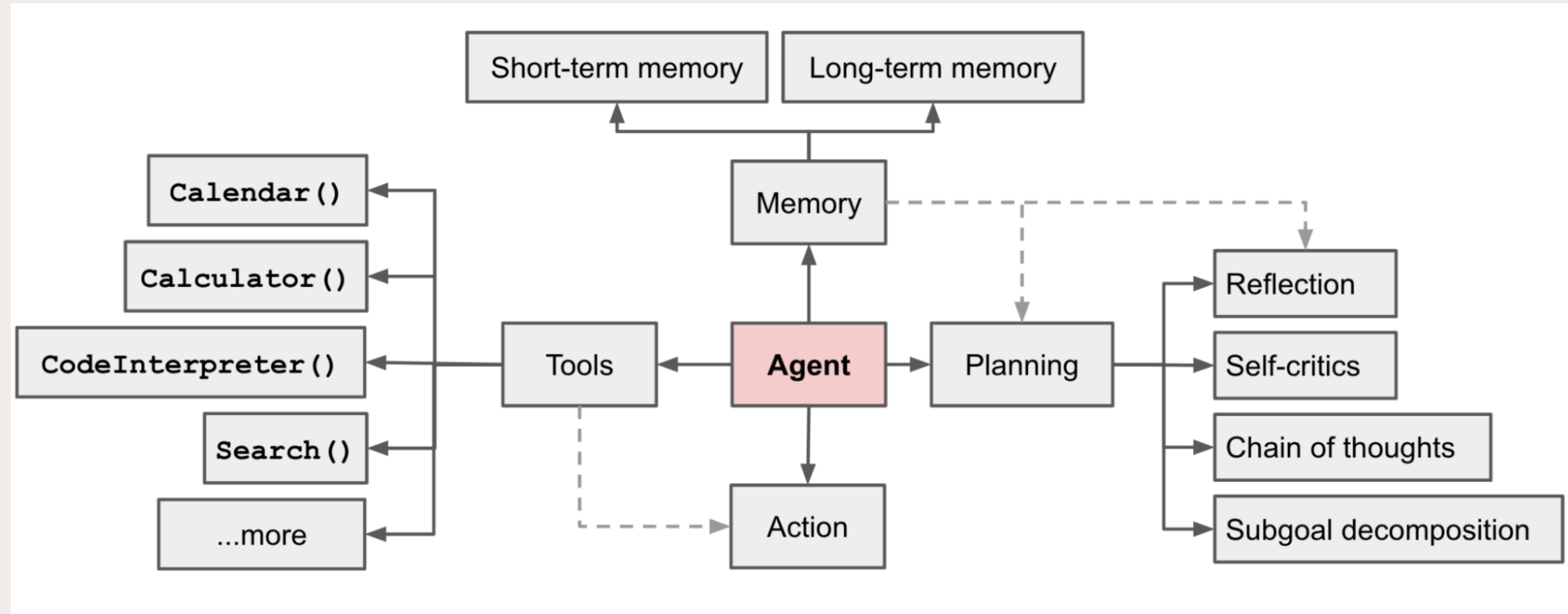
簡要流程：

1. 使用者給予複雜的問題（任務）
2. Agent 將任務**拆解、分發**給多個LLM模型（子任務）
3. 讓 LLM 或使用者評論任務的完成品質，獲得回饋
4. 納入回饋並重新執行任務，直到可被接受
5. 重複以上流程直到任務完成度足夠

舉例：AutoGPT



AI Agent Overview



AI Agent 技術應用

關鍵字：助理

網購助理

- 查詢訂單 送貨時間
- 下訂單
- 訂單問題解決（商品瑕疵或有誤）

程式設計助理

- 能完成 high level 任務：建立能負擔月流量100人次的網站
- 擁有檢視整體架構的能力
- 遵從 best practice，符合 coding style
- 例如：[GitHub Copilot Workspace](#), [Cursor](#)



LangChain Agent 概念

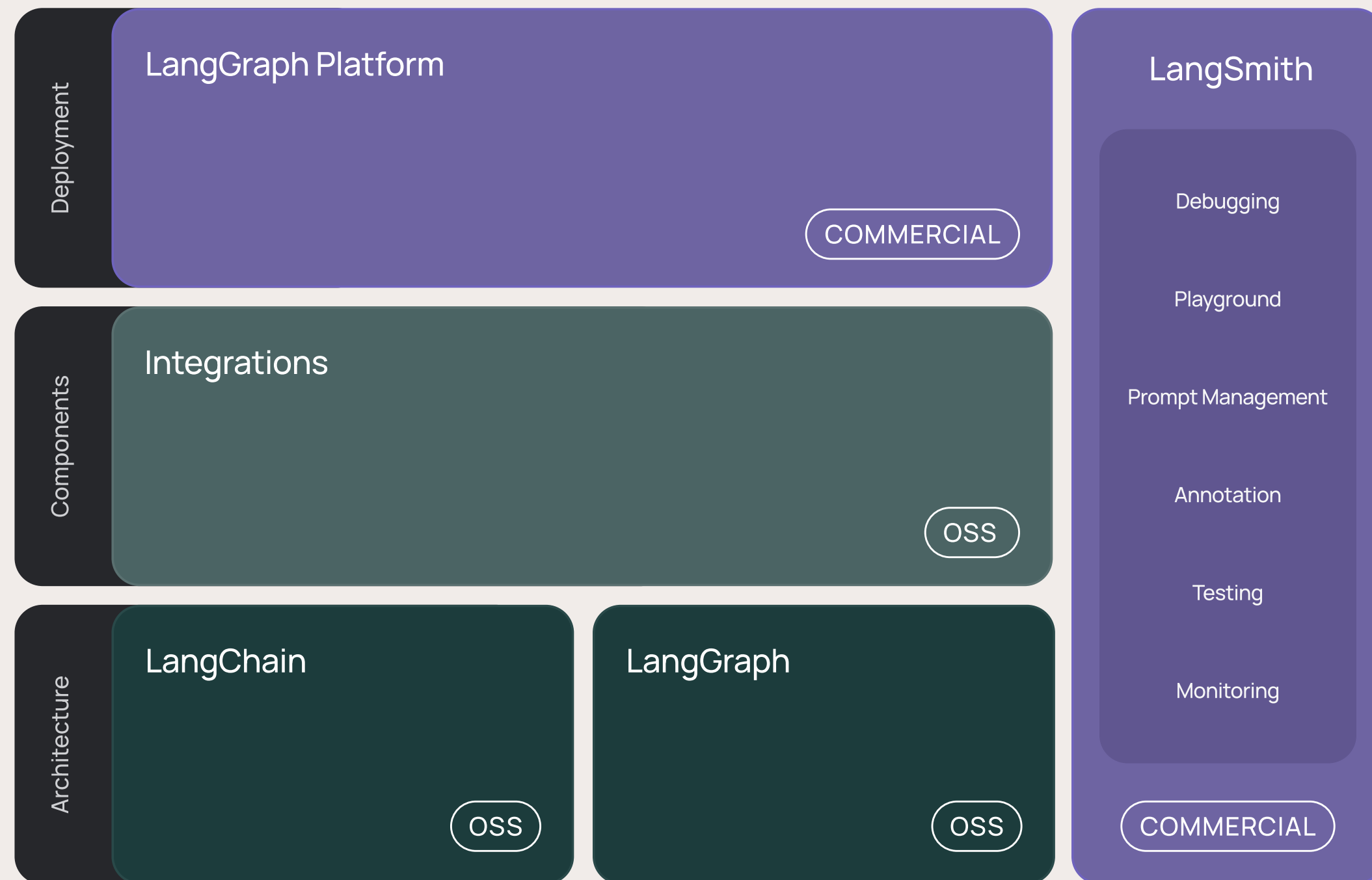
spoiler: Agent 並不是用 LangChain 建構




1. Agent 建立

```
1 from langgraph.prebuilt import create_react_agent
2
3 agent_executor = create_react_agent(model, tools)
```

- 用LangGraph建立，使用 [Prebuilt library](#)



2. Tool



```
1 tools = [search, calculator, translator]
2 agent_executor = create_react_agent(model, tools)
```

- `tool[]` 可放入多個自定義的工具供Agent選擇

3. Memory

```
1 from langgraph.checkpoint.memory import MemorySaver
2
3 memory = MemorySaver()
4 agent_executor = create_react_agent(model, tools, checkpointer=memory)
```

Checkpointner 指整個 graph 的紀錄，含不同 session 和使用者
MemorySaver() 指儲存在記憶體，透過Thread存取同一 graph完整狀態

Official tutorial



此 Tutorial 大綱

- Agent
 - 用LangGraph建立
 - 使用 [Prebulit library](#)
- Tool
 - 定義 tool
- Memory
 - Thread 綁定，in memory 儲存，即MemorySaver()
 - 輸入到checkpointer(完整Graph紀錄)

Code

<https://github.com/hank1224/DataTeam-RAG-training/tree/main/1126-LangChain-Tutorials4-Agent>



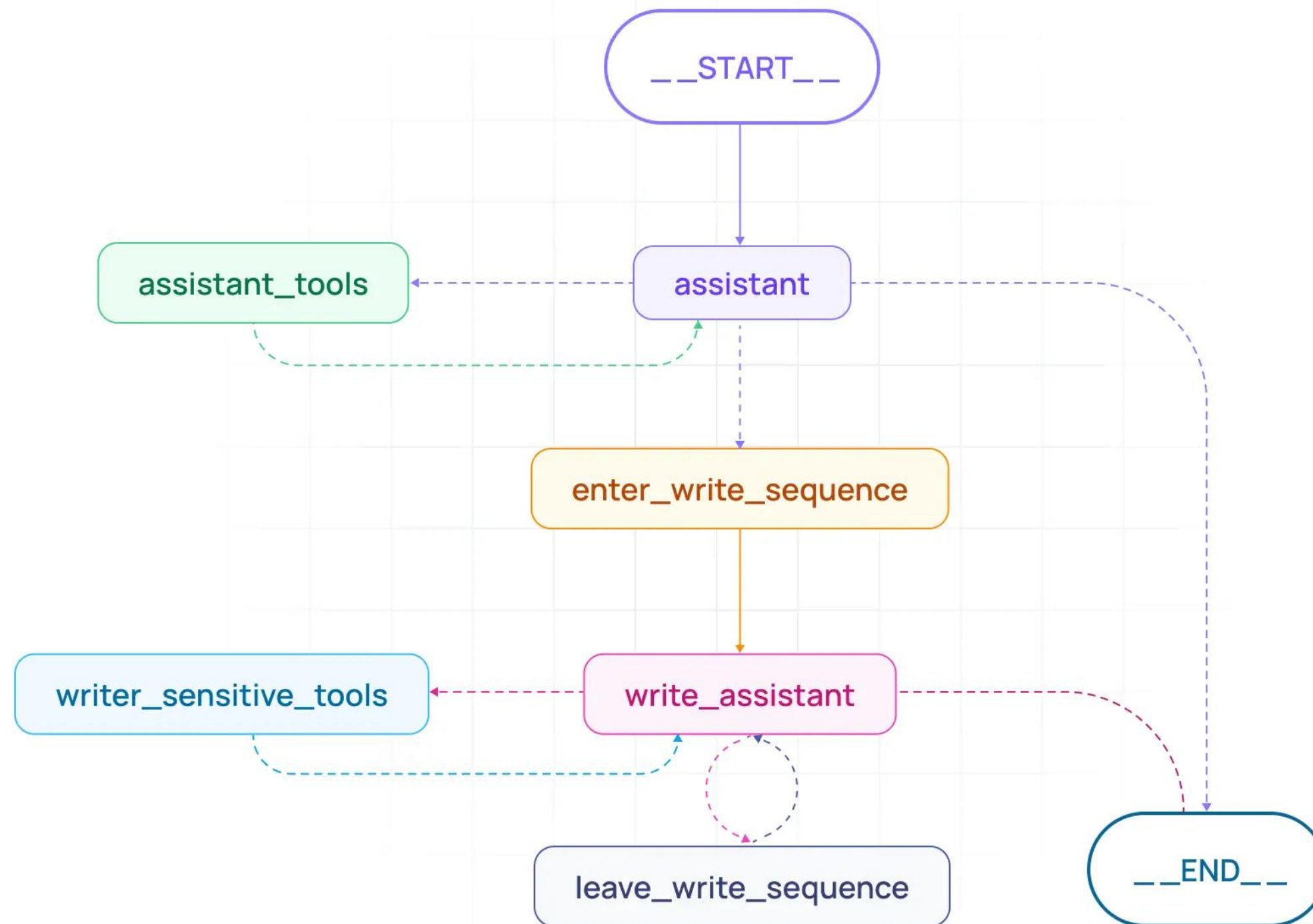
LangGraph

Building language agents as graphs



LLM-powered autonomous agents



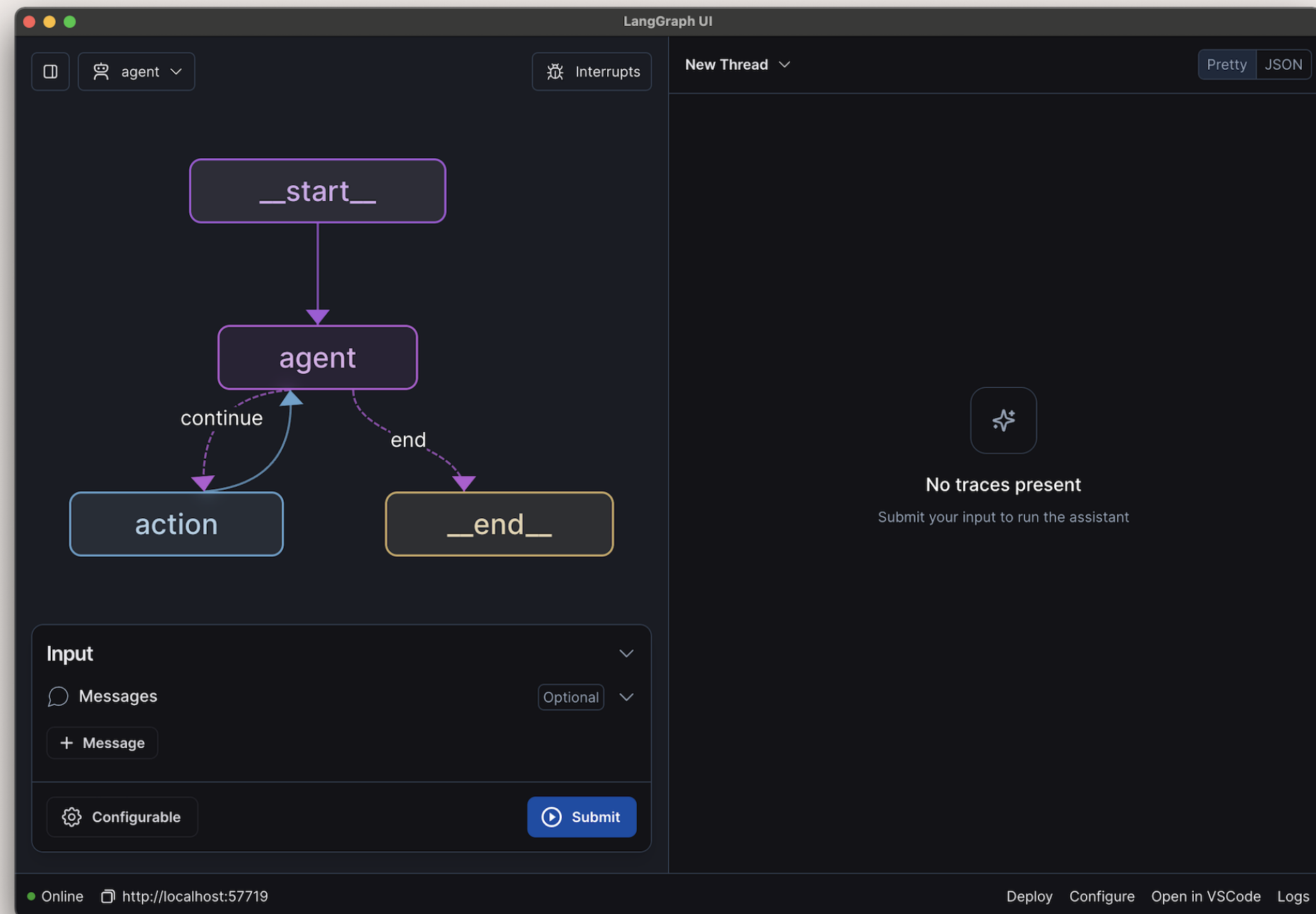


LangGraph 特色

- **Cycles and Branching:**
實現流程循環(for, while)和條件(if)。
- **Persistence 持久性：**
儲存每步驟的狀態（ e.g., via a database ），隨時暫停和恢復執行。
- **Human-in-the-Loop:**
可以人為製造斷點並介入，審核或修改已規劃好的 plan。
- **Streaming Support**
- **Integration with LangChain and LangSmith**

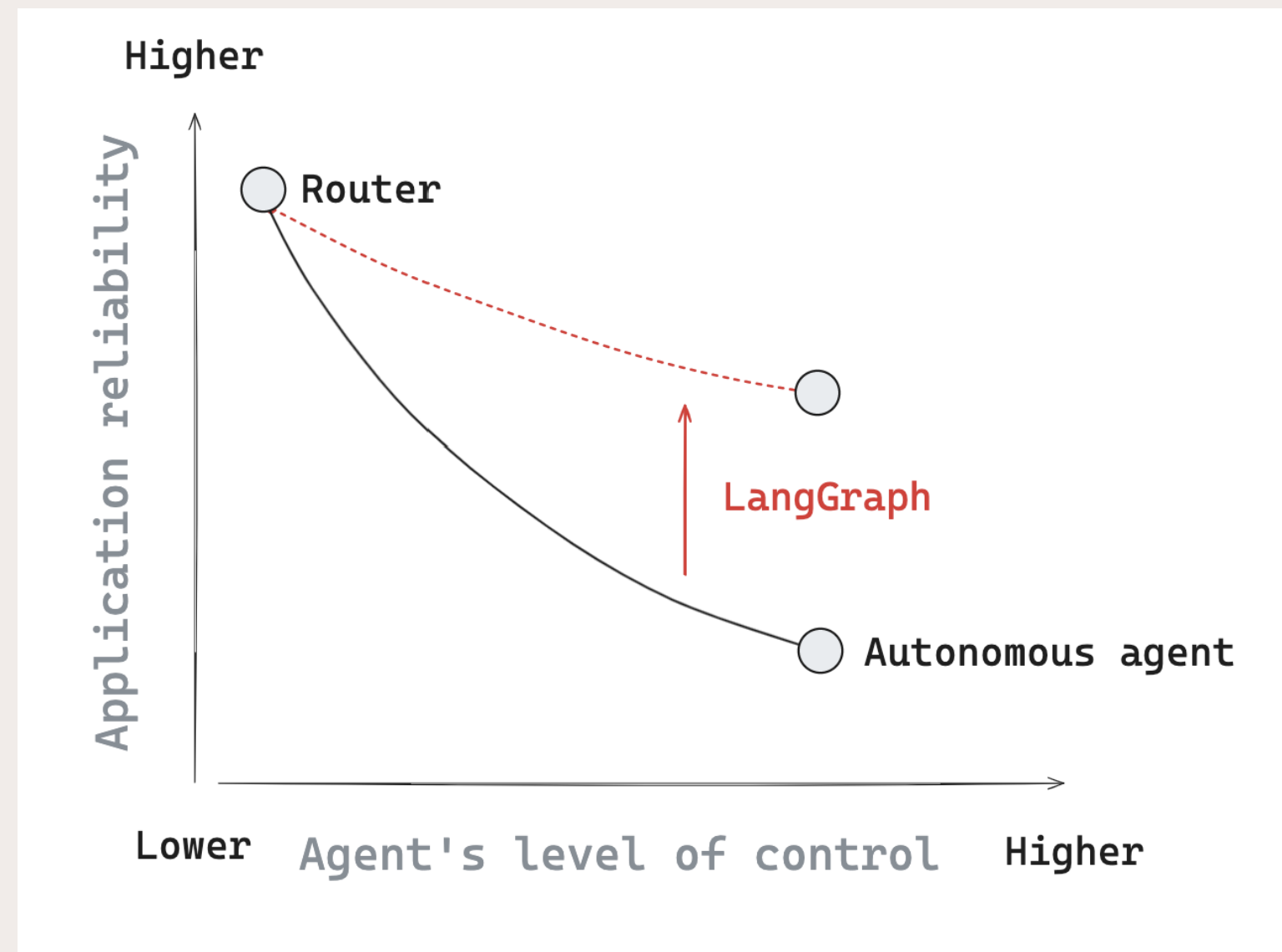
LangGraph 特色

- LangGraph Studio
用於 LangGraph 應用程式
視覺化和調整的 IDE
- 尚屬於早期開發中



LangGraph 理念

- 給予 Agent 越高的自由度, 你的 AP 就越不可靠。
- 給予 Agent 越低的自由度, 建構系統就越麻煩。
- LangGraph 旨在給予更方便的制定流程, 同時保持 AP 的可靠性。



LangGraph Use Case

- Chatbots
- RAG
- Agent Architectures
 - Multi-Agent Systems
 - Planning Agents
 - Reflection & Critique

LangGraph 核心概念

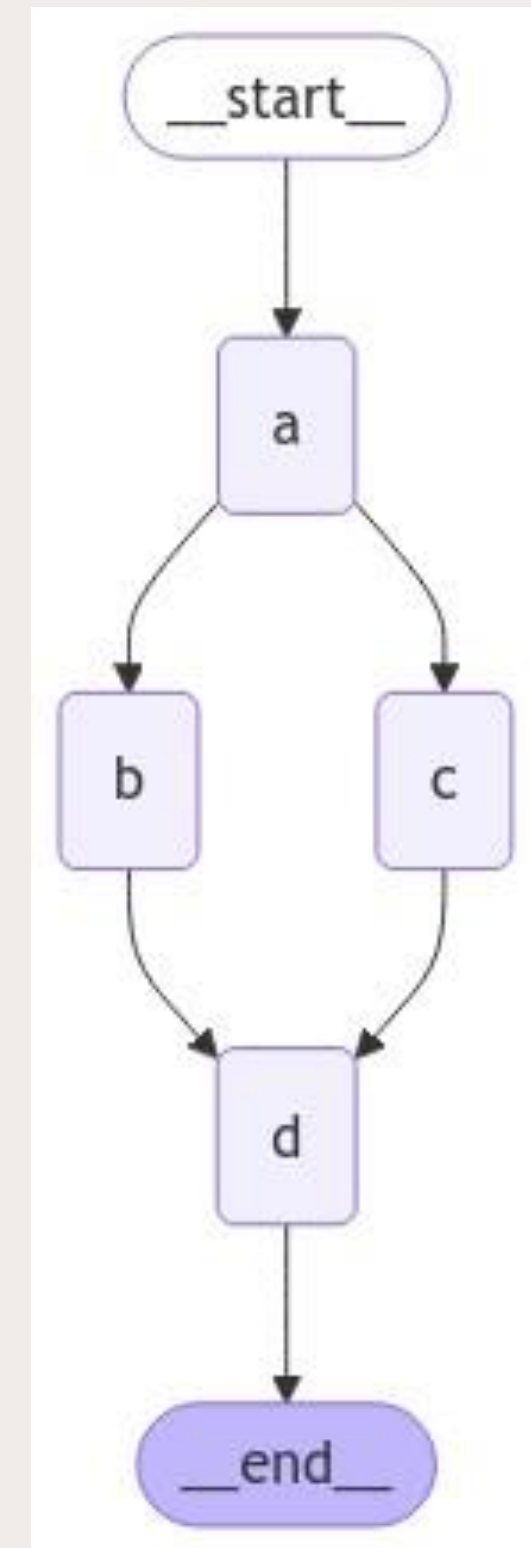
- State: 自定義變數組，可儲存/修改，似執行結果snapshot
- Node: 執行
- Edge: 下一步

註: 這是非常簡易的理解方法

LangGraph 核心概念 – Node, Edge



```
1 builder = StateGraph(OverallState, input=InputState, output=OutputState)
2 builder.add_node("node_1", node_1)
3 builder.add_node("node_2", node_2)
4 builder.add_node("node_3", node_3)
5 builder.add_edge(START, "node_1")
6 builder.add_edge("node_1", "node_2")
7 builder.add_edge("node_2", "node_3")
8 builder.add_edge("node_3", END)
9
10 graph = builder.compile()
11 graph.invoke({"user_input": "My"})
12 {'graph_output': 'My name is Lance'}
```



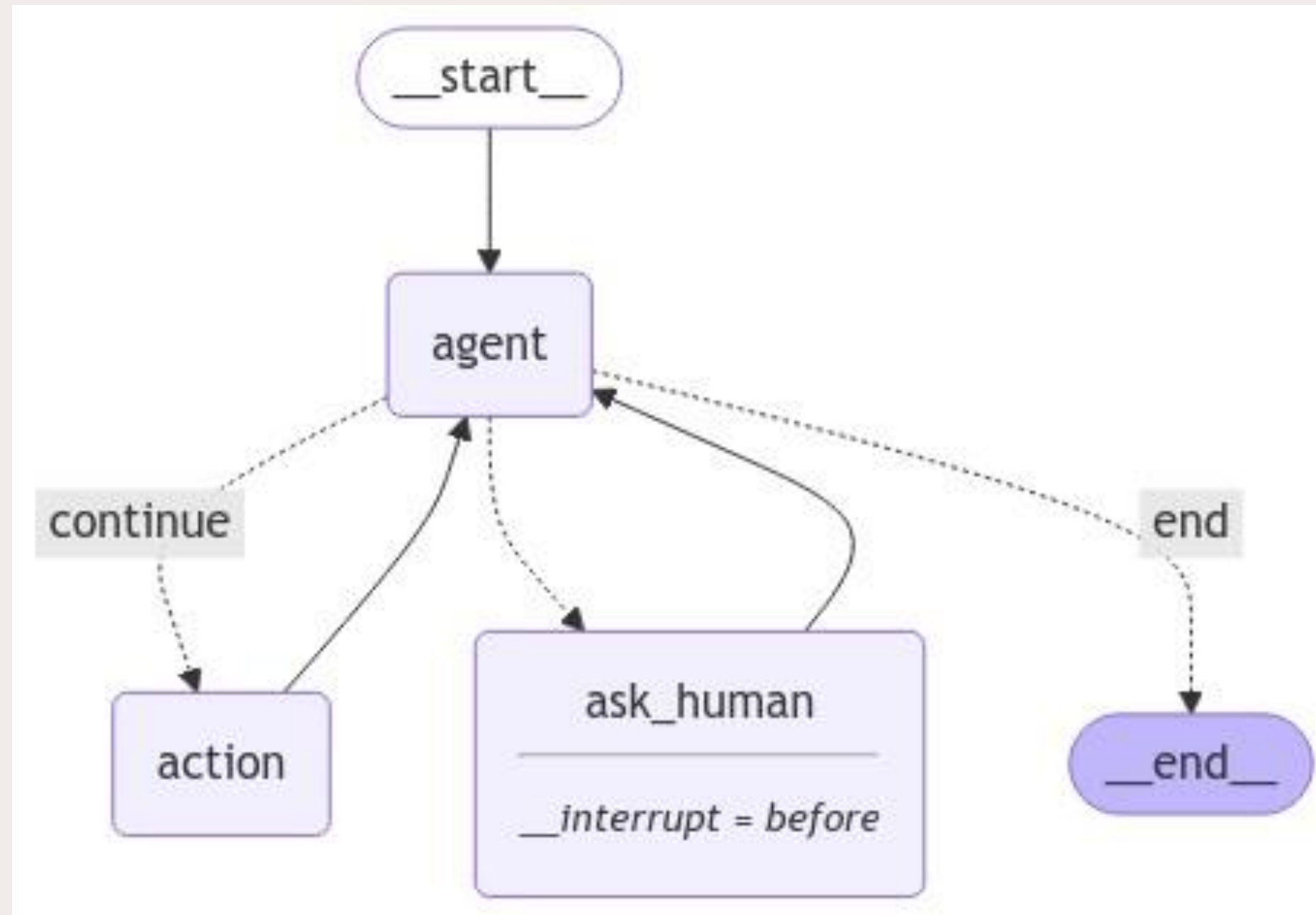
LangGraph 核心概念 – State

- **State:**
指執行狀態、執行結果儲存
- **Checkpoint:**
是指整個 Graph 完整記錄儲存，
含不同使用者、session



```
1 class MyState(TypedDict):  
2     i: int  
3     j: int  
4  
5 def fn2(state: MyState):  
6     i = state["i"]  
7     return {"i": i+1}
```

範例：人為介入



[https://langchain-ai.github.io/langgraph/how-tos/human in the loop/wait-user-input/#agent](https://langchain-ai.github.io/langgraph/how-tos/human%20in%20the%20loop/wait-user-input/#agent)

範例：Customizing State

複習：

- State 是指執行狀態、執行結果儲存
- Checkpointer 是指整個 Graph 完整記錄儲存，含不同使用者、session

<https://langchain-ai.github.io/langgraph/tutorials/introduction/#part-6-customizing-state>

LangGraph 結論

LangGraph 專用於設計 Agent 流程，上手難度較高(自學8hr)

LangGraph Studio 專門的設計 IDE

能給予高度自訂的 Agent 流程，並減少建構時間
可用圖形化界面設計是他的下一步，但現在還太早期

LangGraph 學習資源

- [LangGraph 101: it's better than LangChain](#)
- [LangGraph Deep Dive: Build Better Agents](#)
- [LangGraph Tutorial - Build an AI Agent That Gets You HIRED!](#)
- [LangGraph Official How-to Guides](#)
- [LangGraph 快速入門](#)

建議先看官方 core concept 再看下面這些

- [LangGraph: LangChain Agent 的殺手鐮 \(入門\)](#)
- [LangGraph: LangChain Agent 的殺手鐮 \(進階\)](#)
- [用 LangGraph 寫 LeetCode 解題機器人 Agent](#)



Other Agent Framework

CrewAI

Framework for orchestrating role-playing AI agents.

- Role-based agent design
- Multi-agent collaboration
- Flexible memory system
- Built-in error handling

Other Agent Framework

Microsoft AutoGen

Framework for building multi-agent conversational systems.

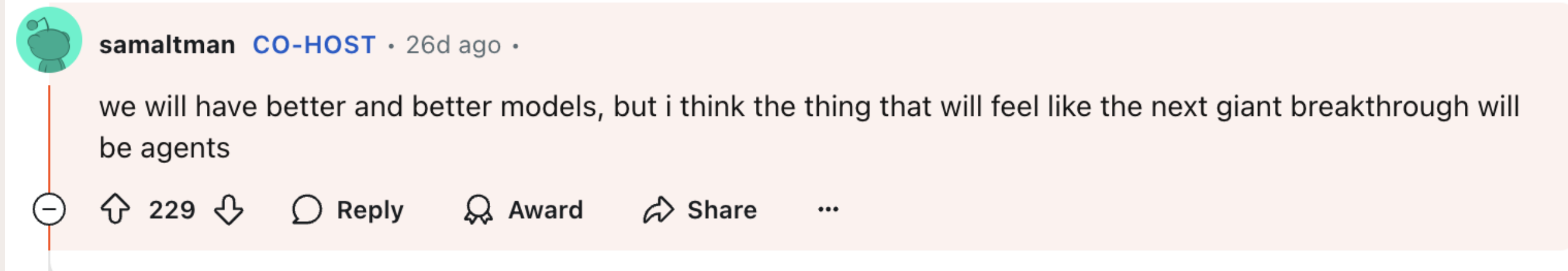
- Multi-agent architecture
- Customizable agents
- Code execution support
- Flexible human involvement
- Advanced conversation management

Other Agent Framework

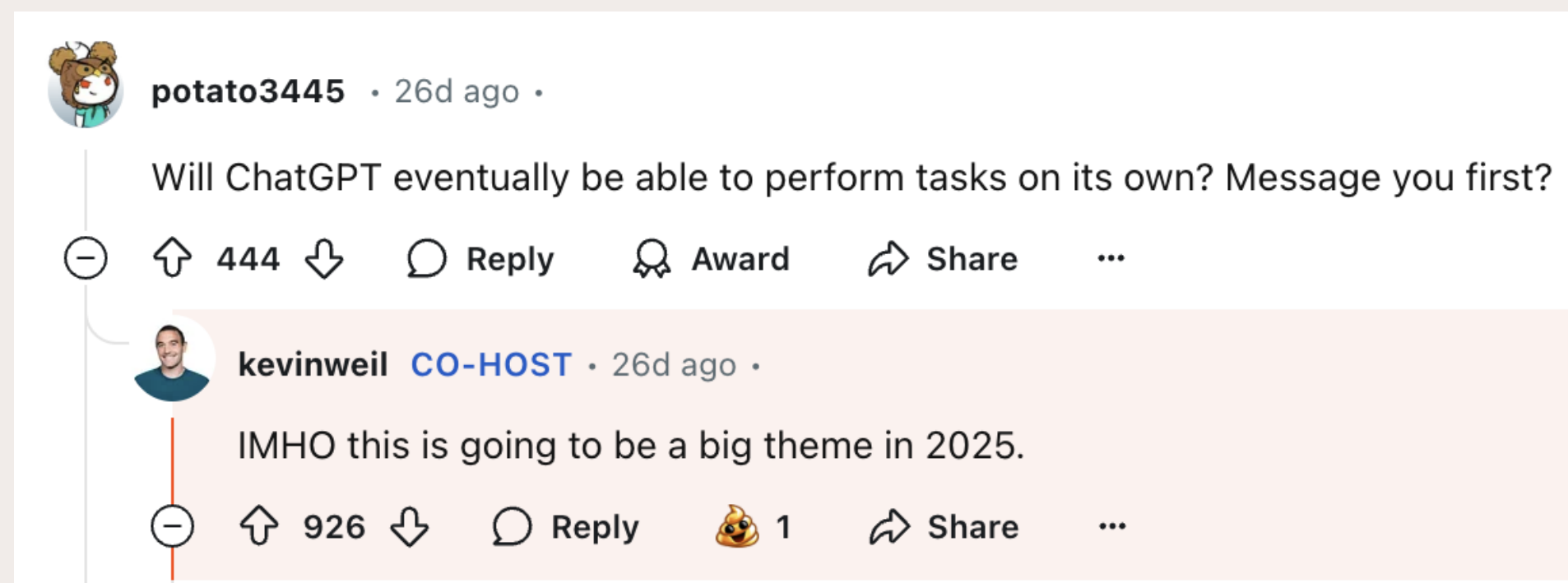
GCP Vertex AI Agent Builder

- SaaS? (prompt算code嗎?)
- 快速試驗可行性的好工具
- 多種 connectors 接入 enterprise data
- 資料混雜情況下好用 (結構化+非結構化)

OpenAI reddit AMA (Ask Me Anything)



<https://www.reddit.com/r/ChatGPT/comments/1ggixzy/comment/luqgr7l/>



<https://www.reddit.com/r/ChatGPT/comments/1ggixzy/comment/luq18fx/>