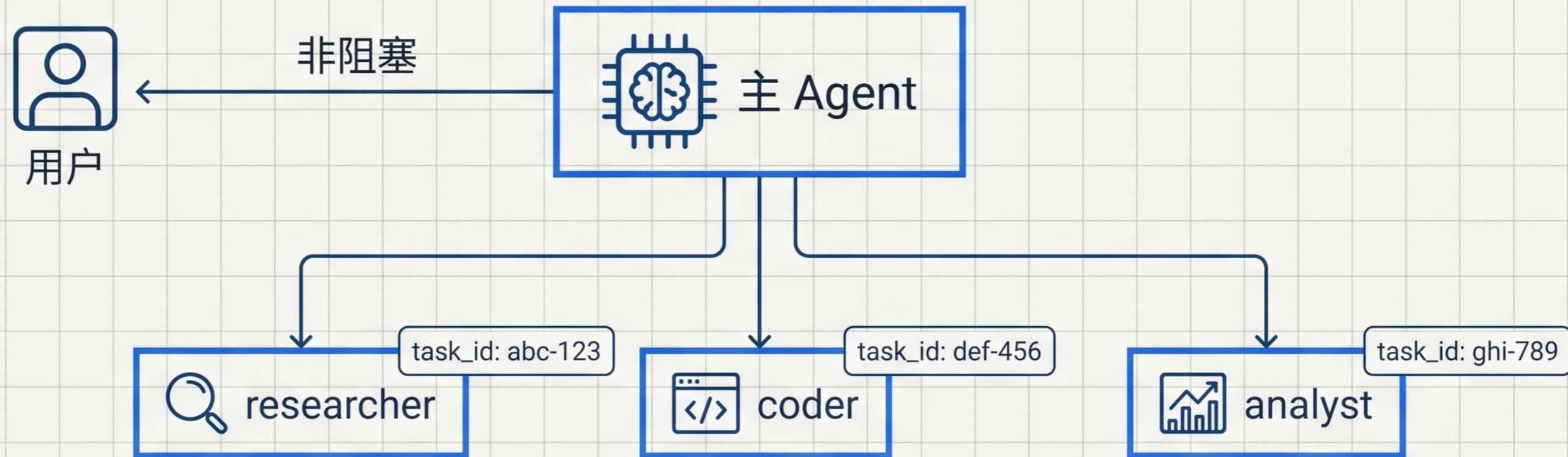


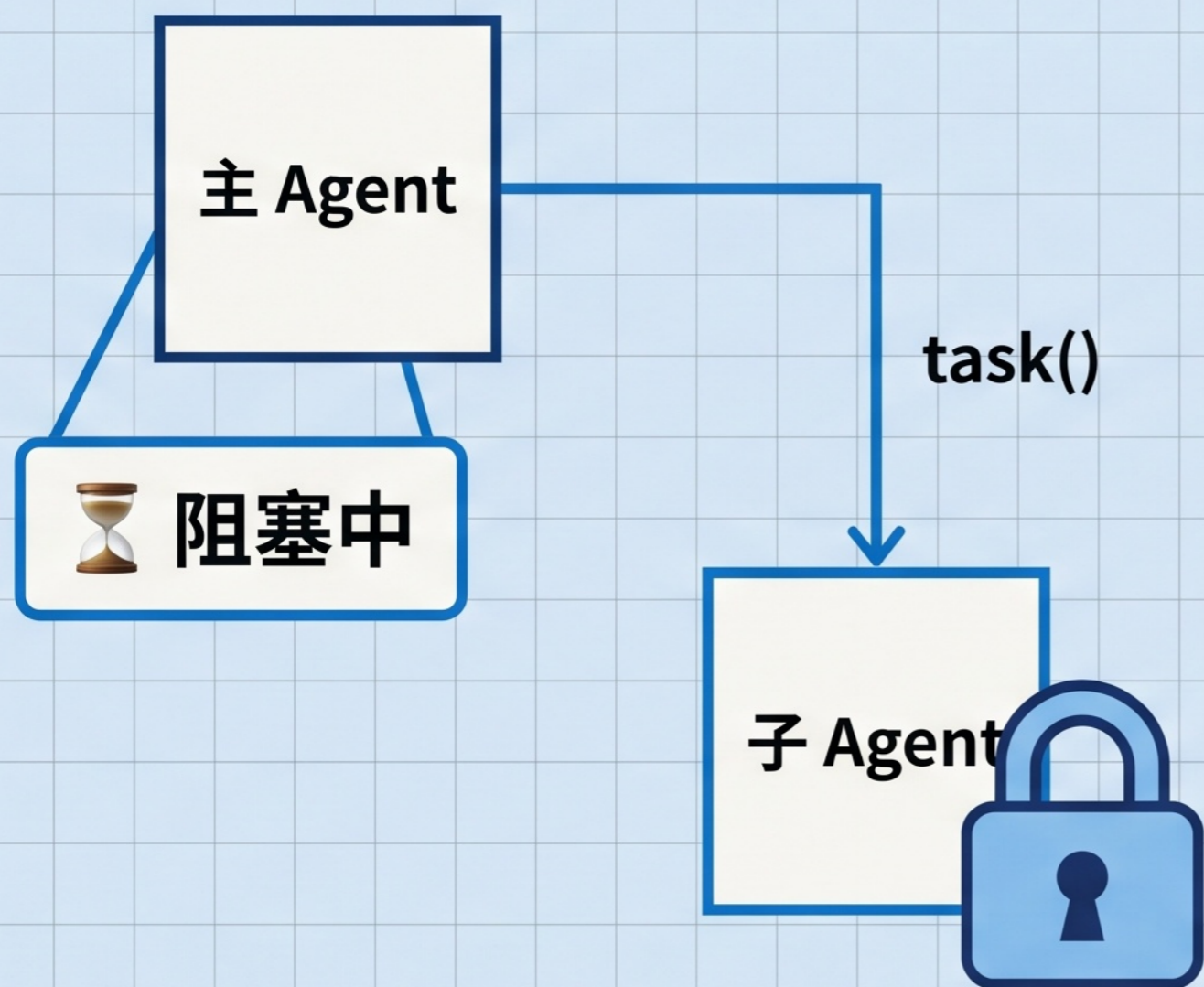
# Deep Agents 实战

## 第 8 讲：异步子 Agent 与并行编排



# 同步子 Agent 的瓶颈

长程任务下，主 Agent 与用户都被卡住



无法插话调整方向



无法取消

- 任务时长从分钟到小时：调研 / 迁移 / 批处理
- 想取消？同步模式没有取消按钮

# 同步 vs 异步：六维对比

5 秒能完成 → 同步；几分钟以上 + 需要可交互 → 异步

判定法则：5 秒 → 同步；分钟级 → 异步

维度	同步子 Agent	异步子 Agent
执行模型	阻塞，等到完成	立即返回 task_id
并发性	并行但整批阻塞	完全并行 + 不阻塞
中途追加指令	✗	✓ update_async_task
取消	✗	✓ cancel_async_task
状态性	无状态	有状态，会话持续累积
适用场景	秒级快速委派	分钟级以上的长程任务

# 配置 AsyncSubAgent

一个 Agent Protocol 服务器, 多张 graph

```
from deepagents import AsyncSubAgent, create_deep_agent

async_subagents = [
    AsyncSubAgent(
        name="researcher",
        description="深度研究 Agent",
        graph_id="researcher",
        # 不传 url → ASGI 进程内传输
    ),
    AsyncSubAgent(
        name="coder",
        description="编码 Agent",
        graph_id="coder",
        url="https://coder.langsmith.dev", # 可选 → 走 HTTP 远程
    ),
]

agent = create_deep_agent(
    model="google_genai:gemini-3.5-flash",
    subagents=async_subagents,
)
```



必填三件套

name / description / graph\_id



url

缺省 → ASGI; 填了 → 远程 HTTP



headers

自托管服务自定义鉴权头

# 主 Agent 的 5 把遥控器

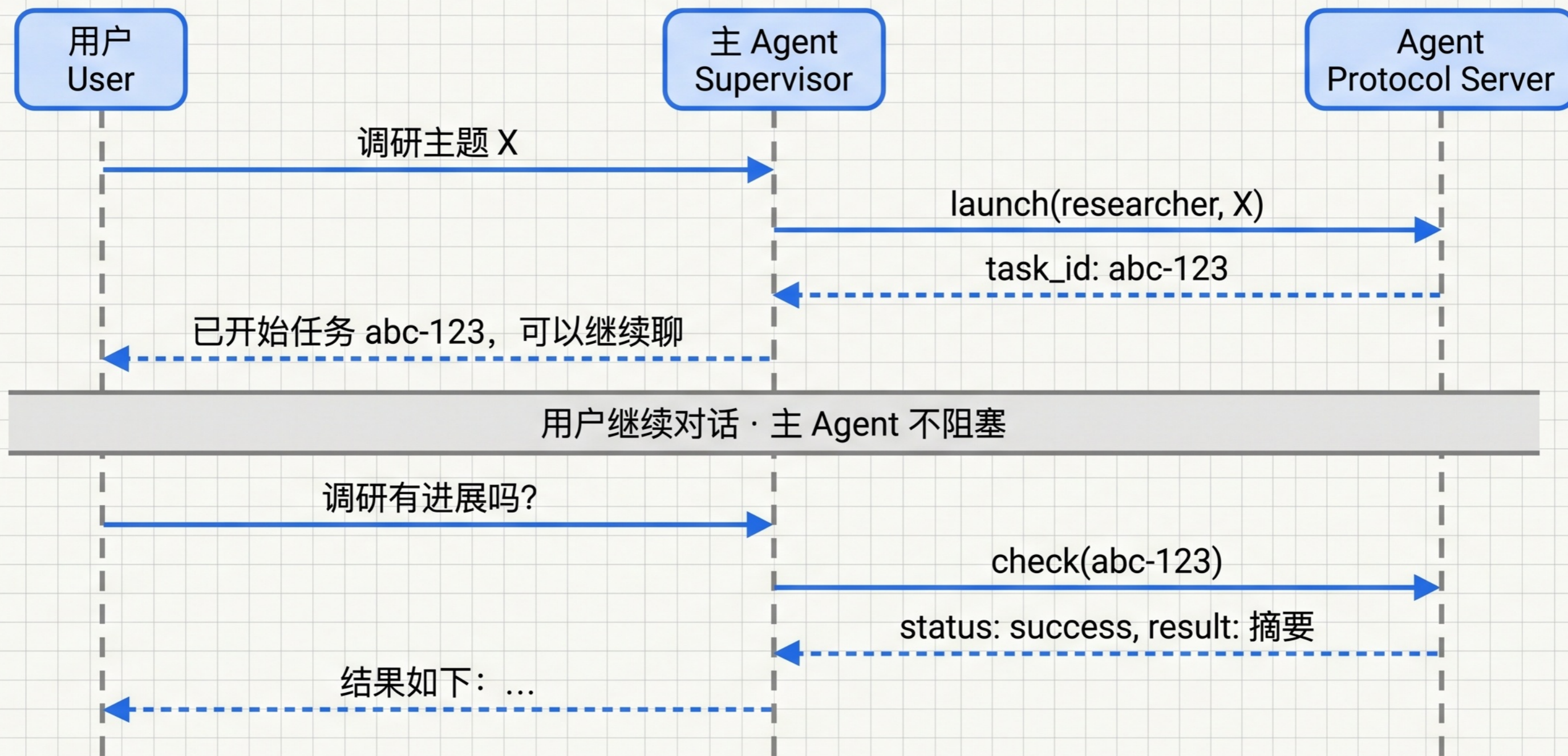
AsyncSubAgentMiddleware 自动注入



像普通工具一样调用 · 中间件负责线程与状态持久化

# 一次完整生命周期

User · Supervisor · Agent Protocol Server

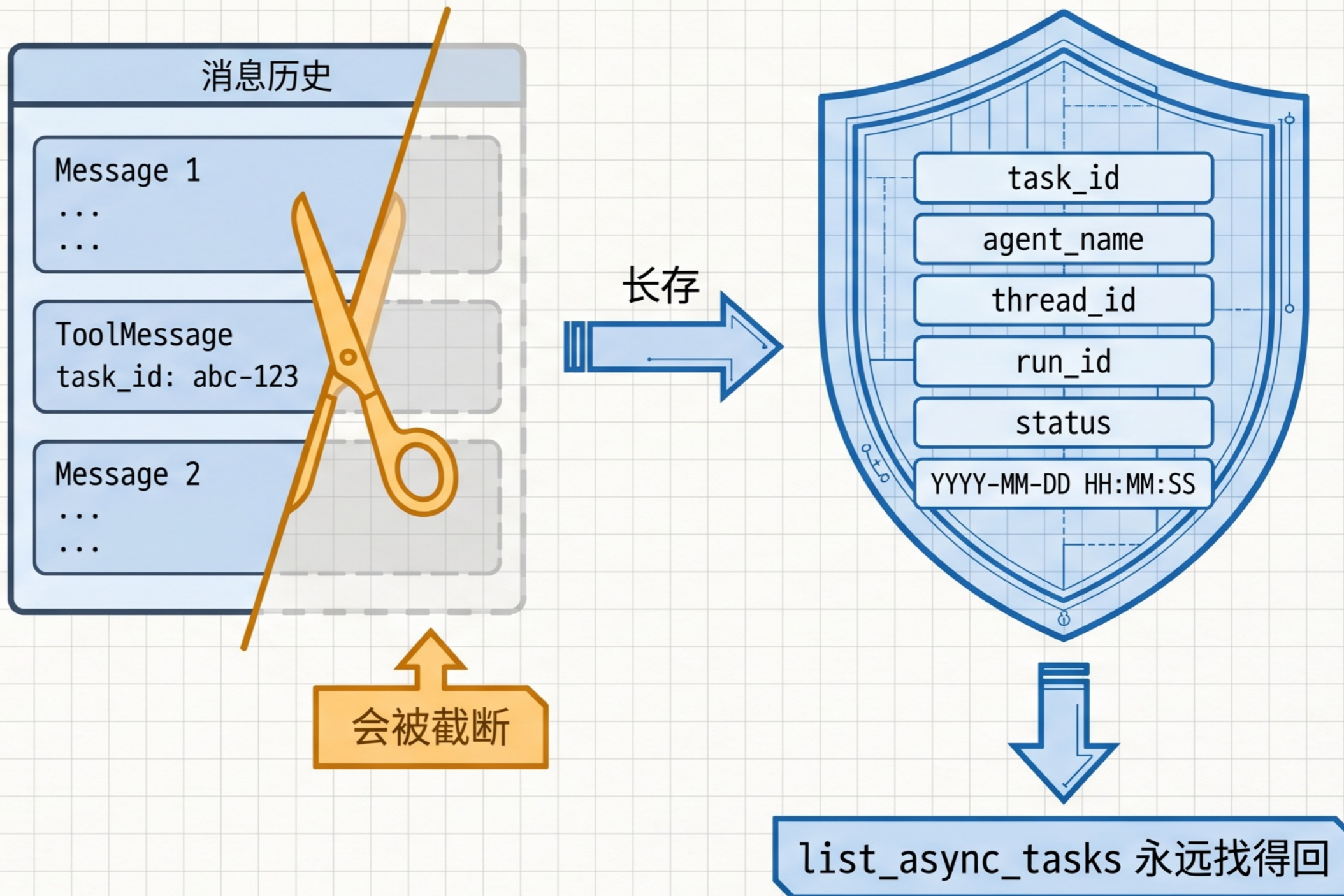


launch 不轮询, 由用户驱动 check

# 为什么要单开 async\_tasks channel?

上下文会被压缩，任务 ID 不能丢

- 主 Agent 上下文逼近上限会自动压缩消息历史
- 任务 ID 若只活在 ToolMessage 里，压缩后会丢失
- 每条任务记录：task\_id / agent\_name / thread\_id / run\_id / status / 时间戳
- 单独 channel 确保 list\_async\_tasks 永远找得回所有任务



# 两种传输 · 三种拓扑

起手用单部署，按工程需要再拆

## 传输方式

### ASGI (不传 url)



进程内传输，零延迟，零鉴权

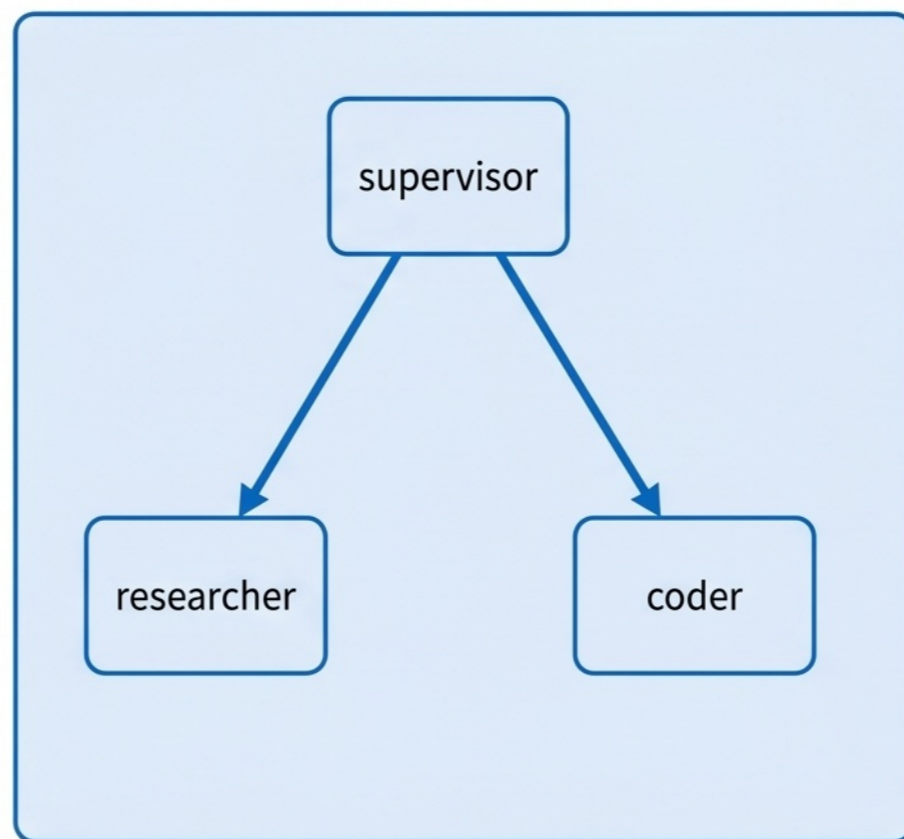
### HTTP (传 url)



远程 Agent Protocol 服务，需要 LANGSMITH\_API\_KEY

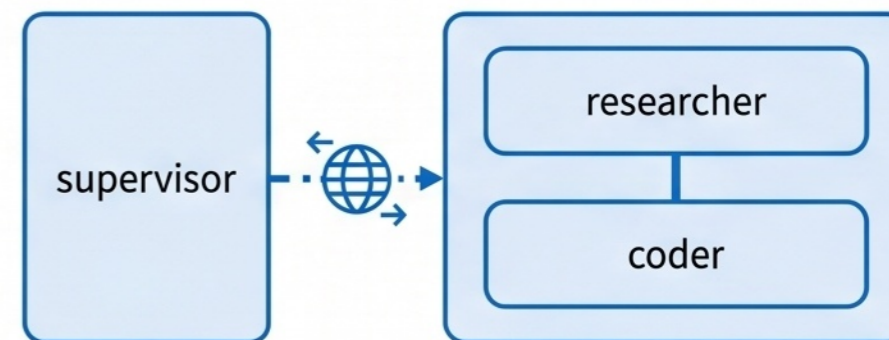
## 部署拓扑

### 单部署 (推荐起点)



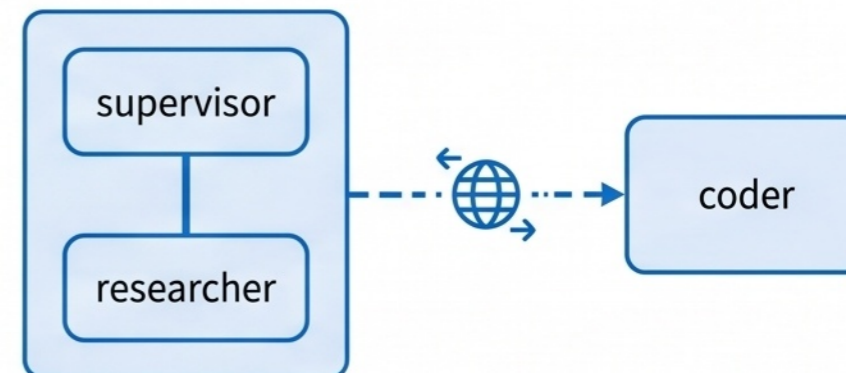
所有 graph 在同一 langgraph.json，全 ASGI

### 拆分部署



主 / 子分服务，全 HTTP (独立扩缩容)

### 混合



部分 ASGI + 部分 HTTP





# 最佳实践 · 排障速查

3 条要点 + 4 类典型坑

## 最佳实践

## 排障

- **worker pool 调大**  
运行命令: `langgraph dev --n-jobs-per-worker 10`
- **描述具体行为导向**  
否则主 Agent 不知道派给谁。
- **thread\_id == task\_id**  
串联 LangSmith 双轨 trace。

- **启动后立即轮询 check** → 强化 system\_prompt: “必须把控制权交回用户”。
- **引用过时状态** → 强制总是先 check / list 再回答。
- **task\_id 被截断 / 改写** → system\_prompt 要求完整 ID。
- **启动卡住** → worker pool 满, 调大 --n-jobs-per-worker。

# 子 Agent 双讲回顾 & 进阶预告

