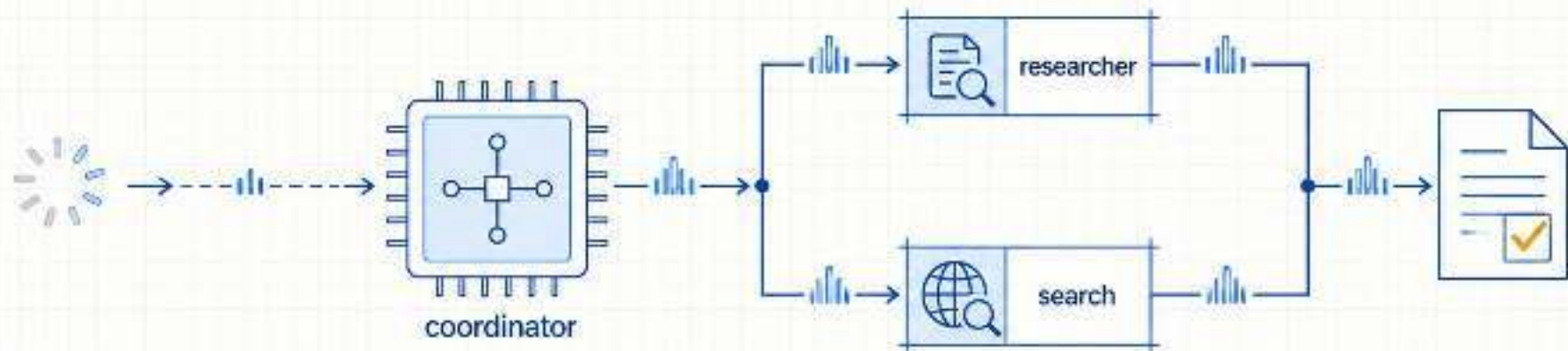


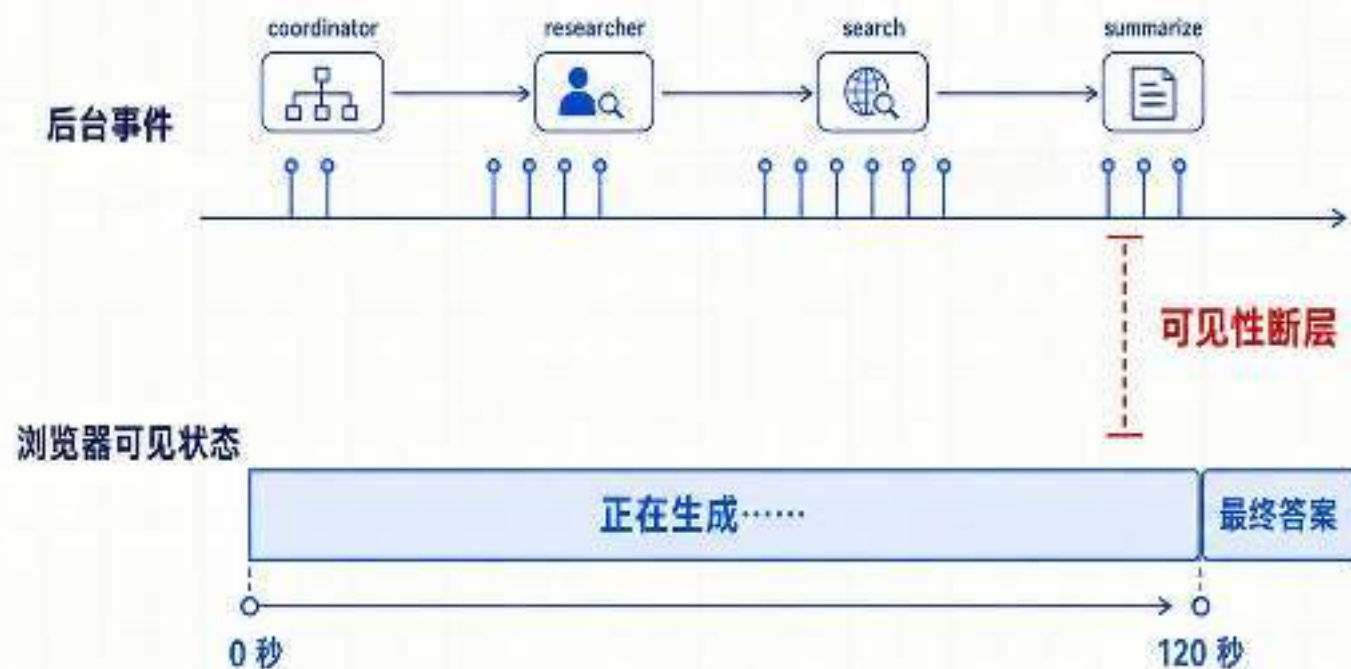
Deep Agents 实战

第 18 讲: Streaming — 实时观察主 Agent、子 Agent 与工具调用



最终答案能返回，页面依然像卡住了

后台已启动 researcher 并调用搜索；前台两分钟只有“正在生成……”



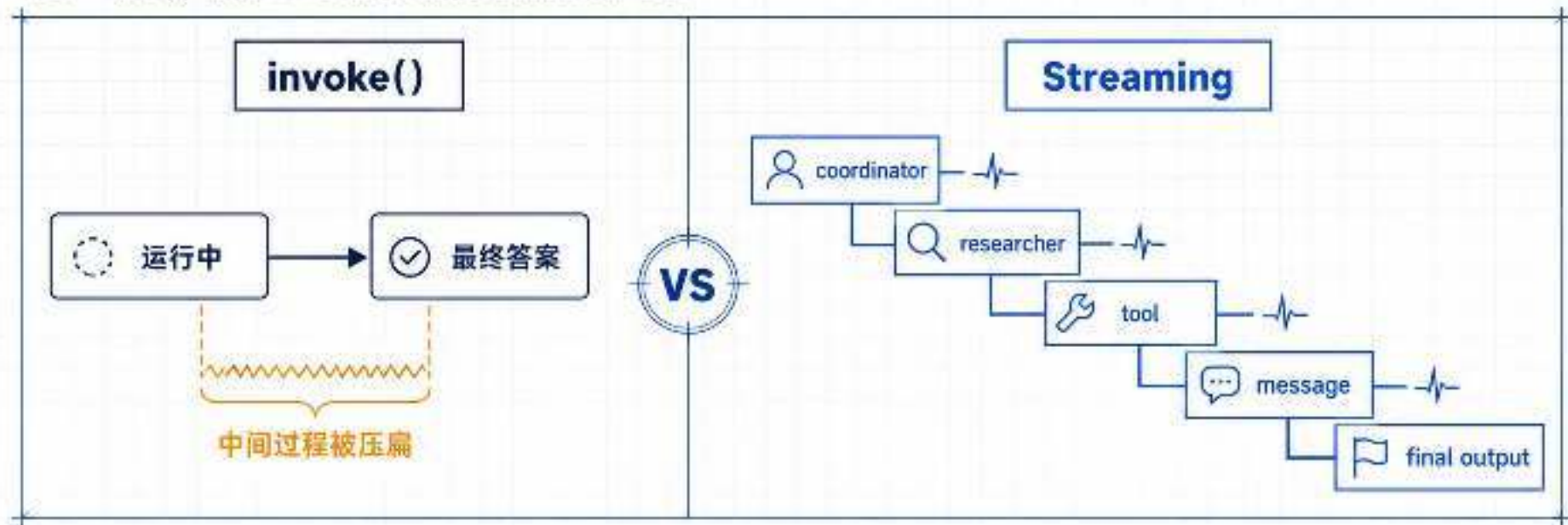
① 1. 委派是否发生？

② 2. 工具正在运行，还是已经失败？

③ 3. 中间过程去了哪里？

invoke() 只交付结果，Streaming 交付运行过程

同一请求，体验差异来自中间事件是否可见



第一步不是做漂亮进度条，而是先显示谁正在工作。

运行 Streaming: v3 看对象, v2 看事件包

每个版本只保留一个入口, 并锁定三个真正需要消费的点

v3 · 产品视图

```
stream = agent.stream_events(..., version="v3")
```

- stream.subagents
找到委派; 读取 path / status
- subagent.messages
进入这个 subagent 自己的消息流
- message.text
拿到可直接展示的文本

v2 · 图执行协议

```
agent.stream(..., stream_mode="updates",  
             subgraphs=True, version="v2")
```

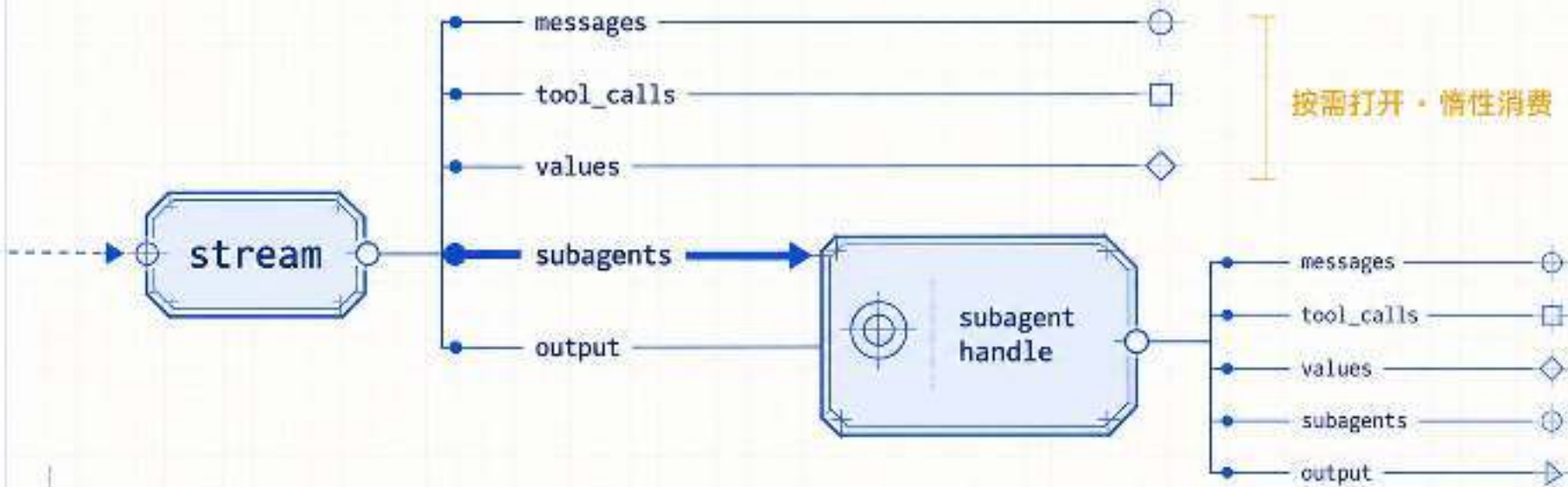
- chunk["type"]
先筛 updates, 忽略其他类型
- chunk["ns"]
空路径 = main; 非空路径 = subagent
- chunk["data"]
完成路由后, 再解释 payload



两个循环分开消费, 再在应用层统一

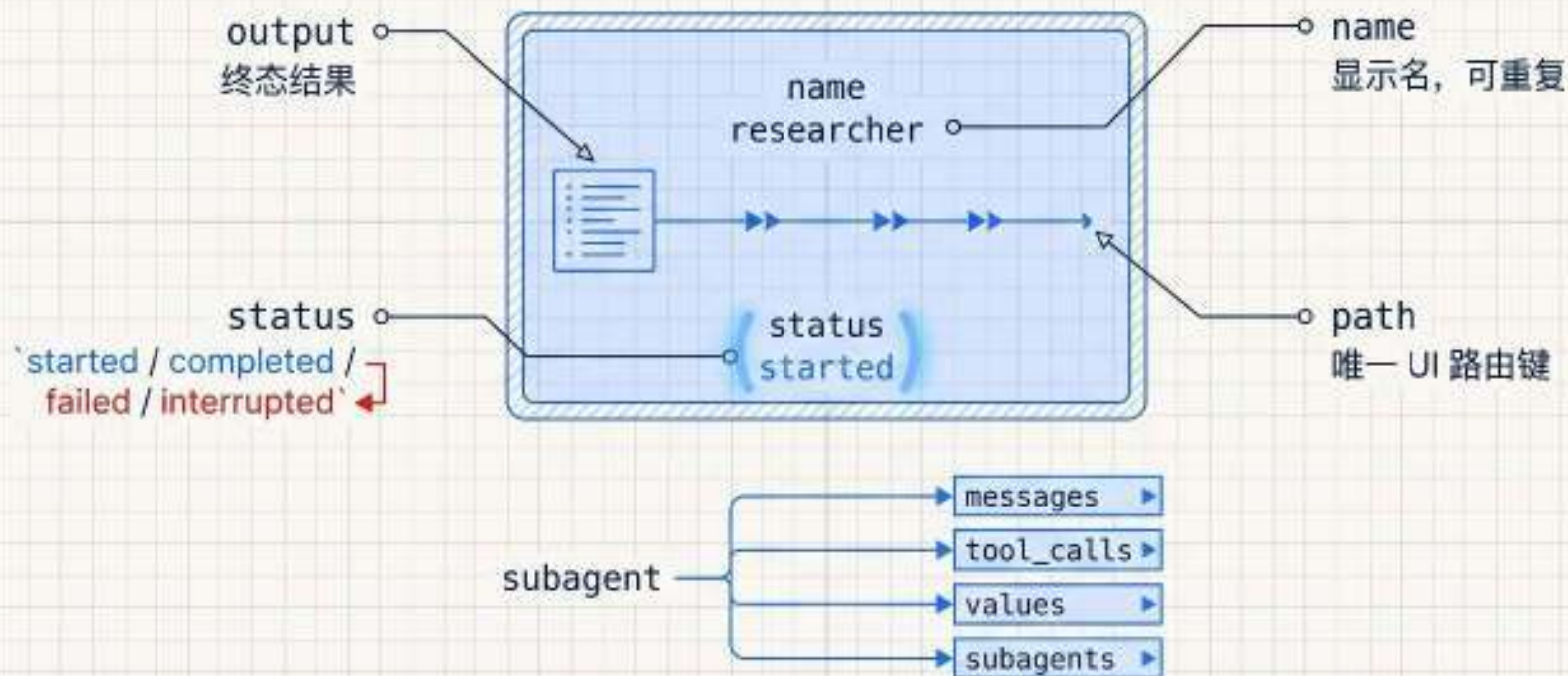
v3 把复杂事件投影成五条可消费细流

新应用优先从产品语义出发，而不是从 graph node 猜状态



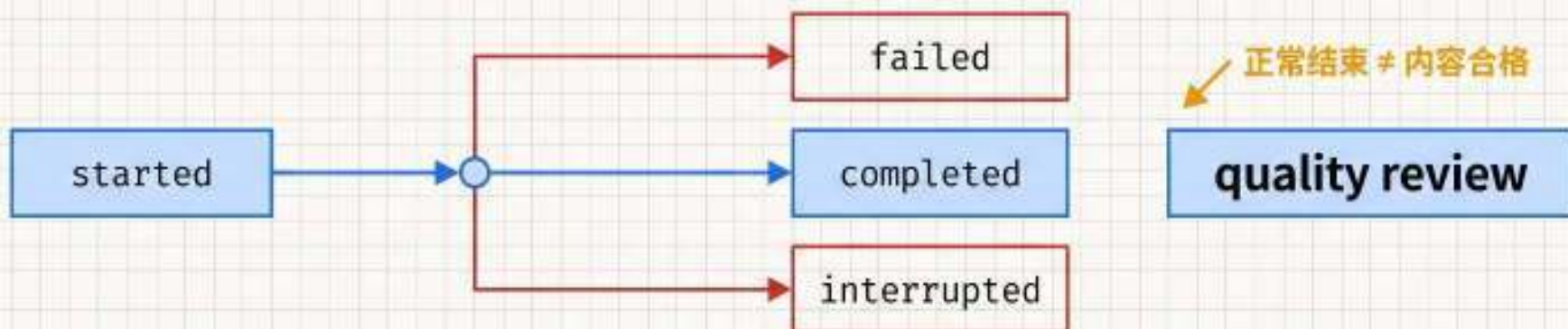
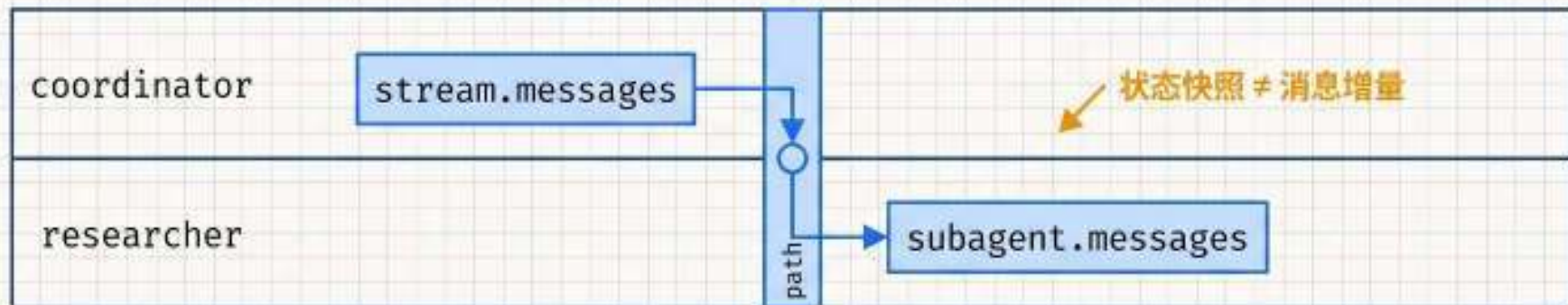
subagent handle 把一次委派变成可路由卡片

name 用来显示, path 才是唯一键, status 只表示生命周期



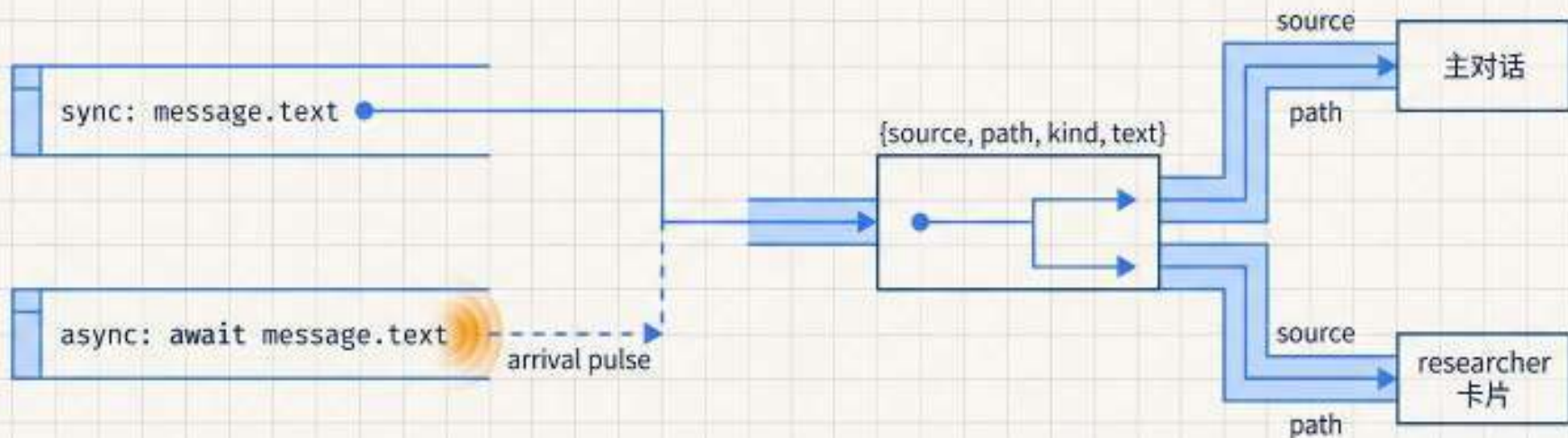
Projection 不会替你跨层合并, completed 也不是质量结论

coordinator 与 researcher 是两条独立事件范围



message.text 解决“显示什么”，projection 决定“来自谁”

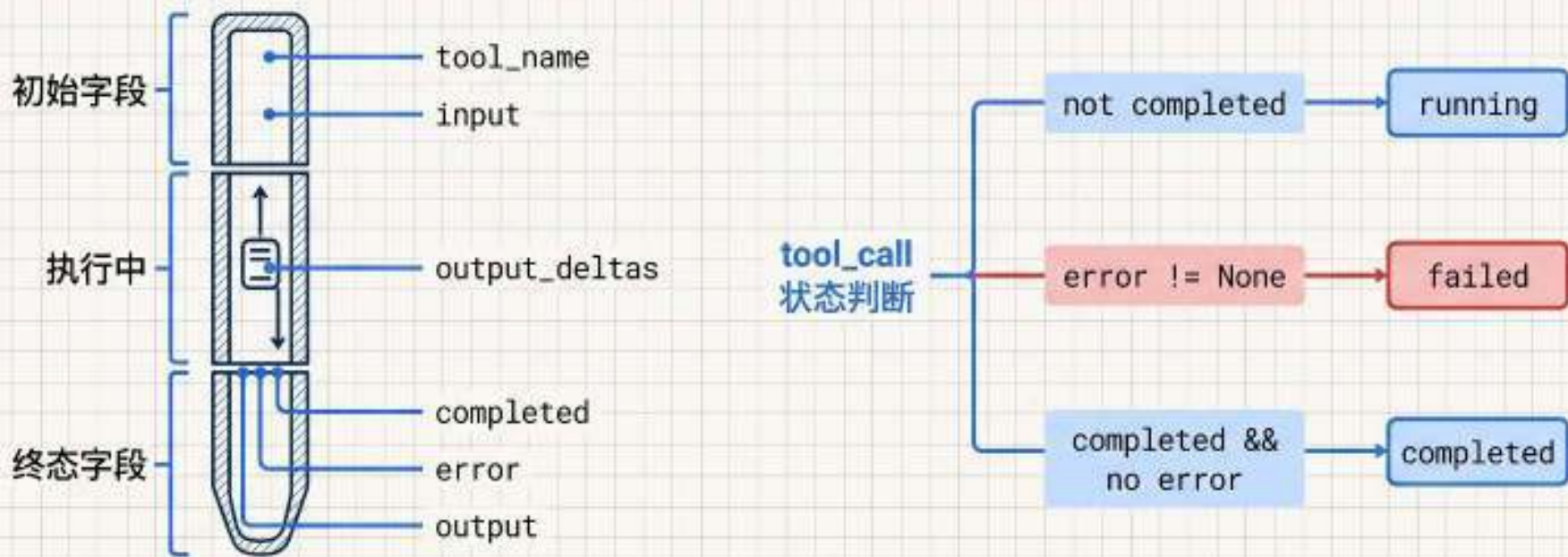
同步直接读取；异步文本仍在到达，需要 await



message.text 不是精确 token 顺序接口

tool_call 必须同时判断 completed 和 error

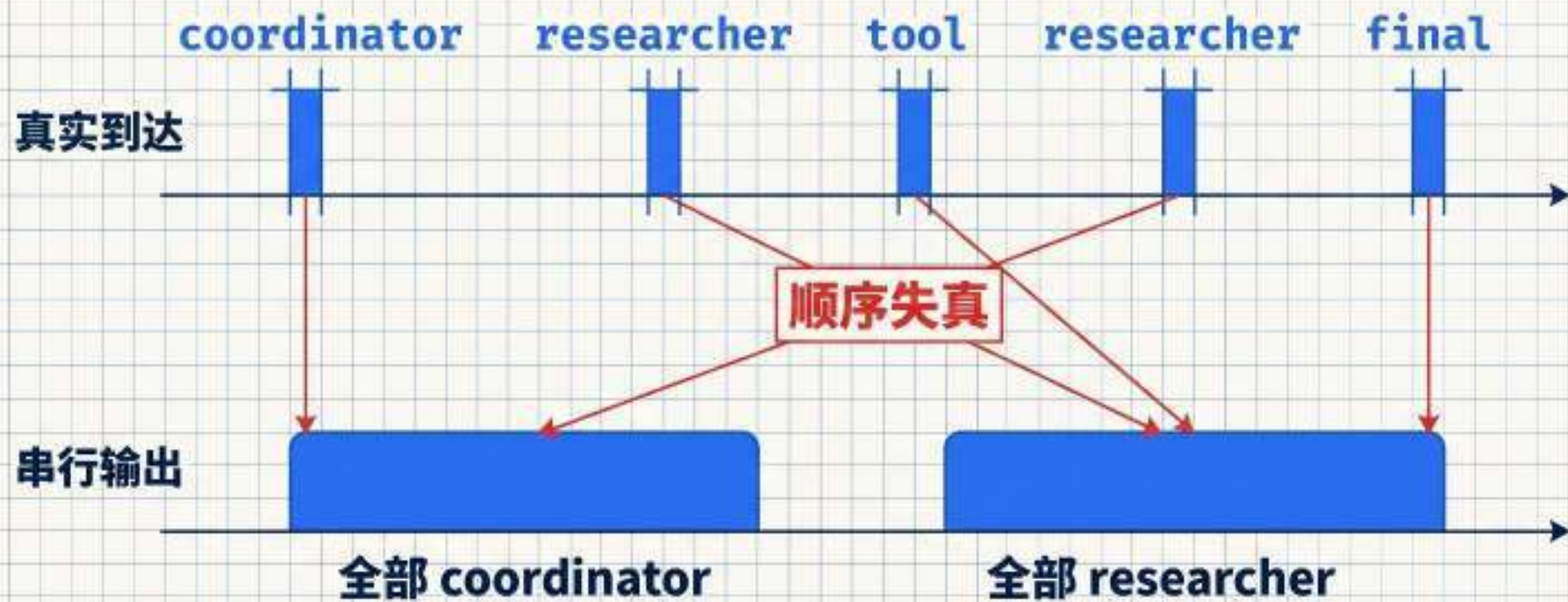
增量、终态与失败是三个不同时间点



不要把 delta 与最终 output 重复追加

串行消费会制造一条假的时间线

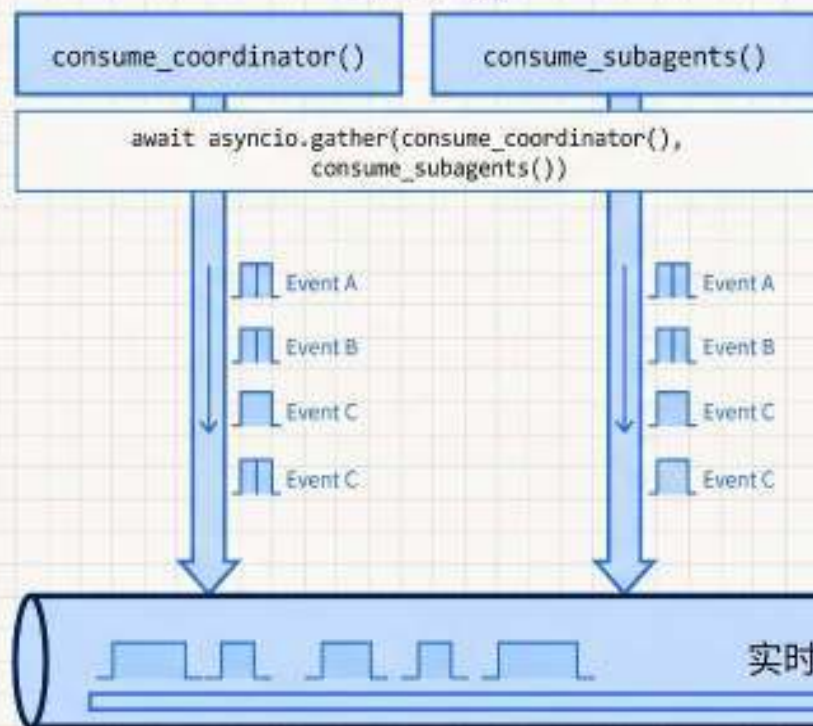
researcher 明明先完成，却被排在 coordinator 最终总结之后



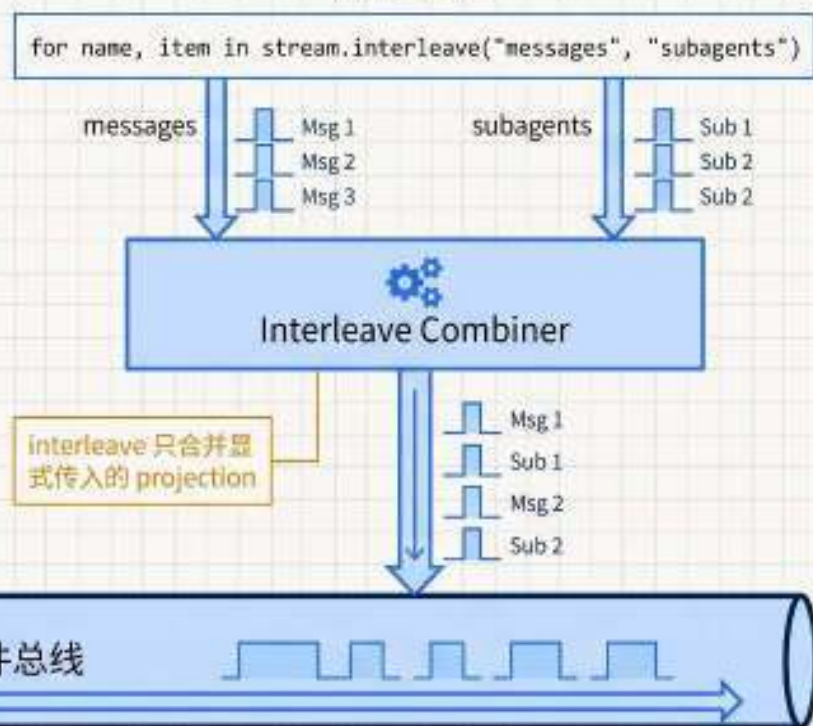
异步用 gather, 同步用 interleave

两者都避免 iterator 排队, 但不承诺跨 projection 的全局 token 顺序

异步服务

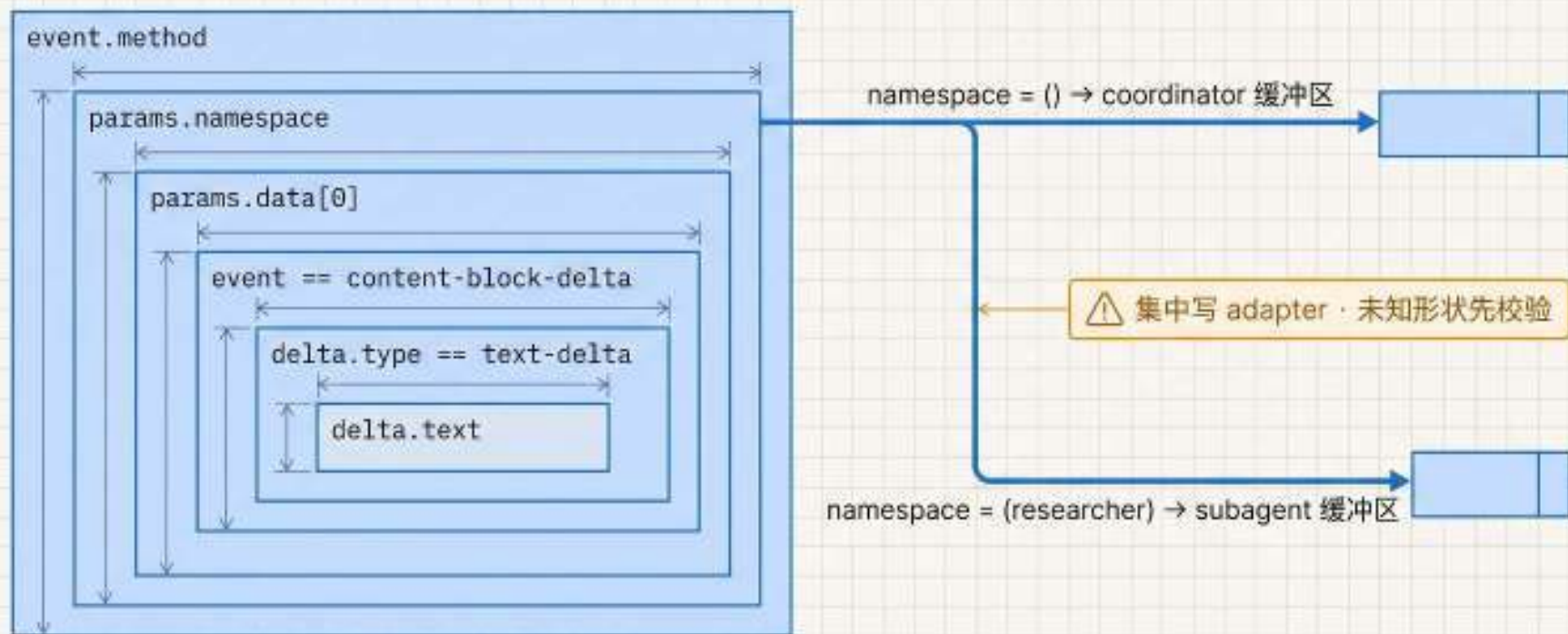


同步程序



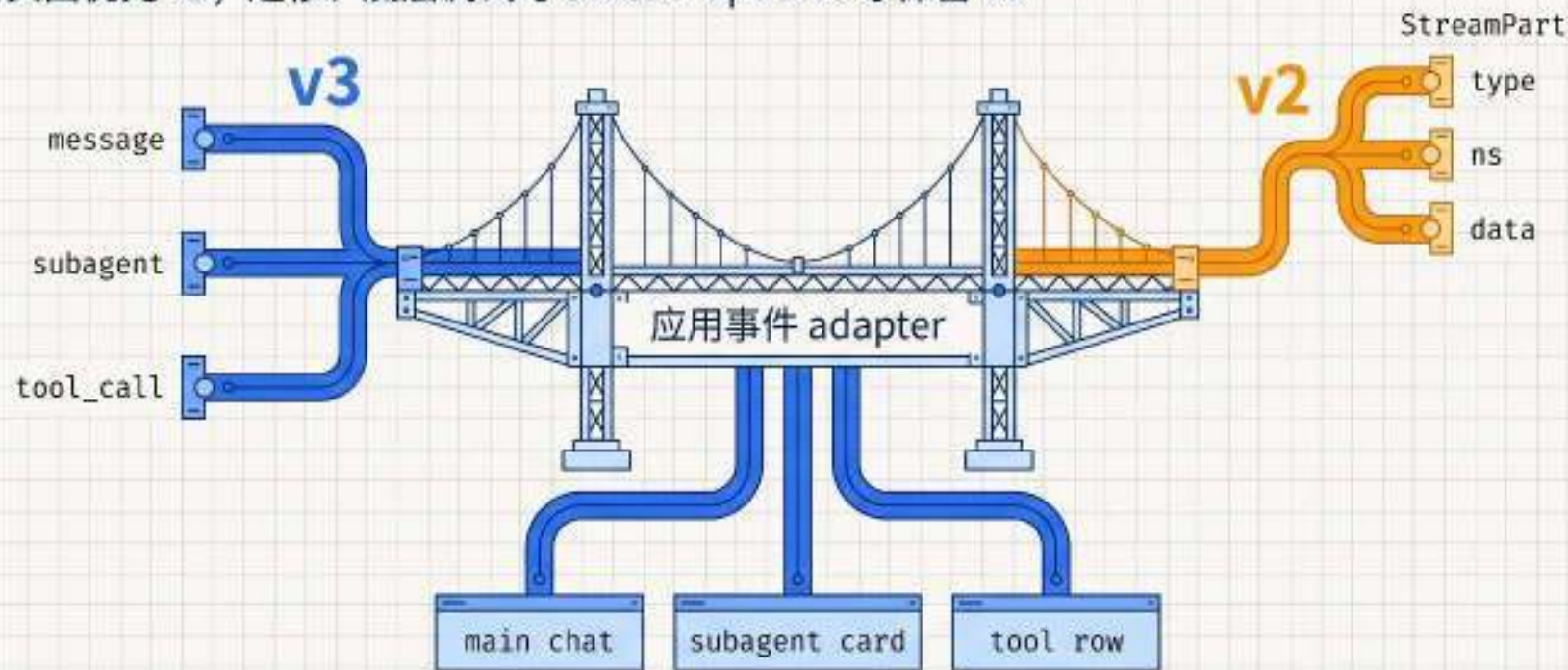
需要精确 token 顺序时，再下沉到 raw events

产品 UI 用 typed projection；调试、重放和审计才解析协议层



v3 是产品视图，v2 是图执行协议

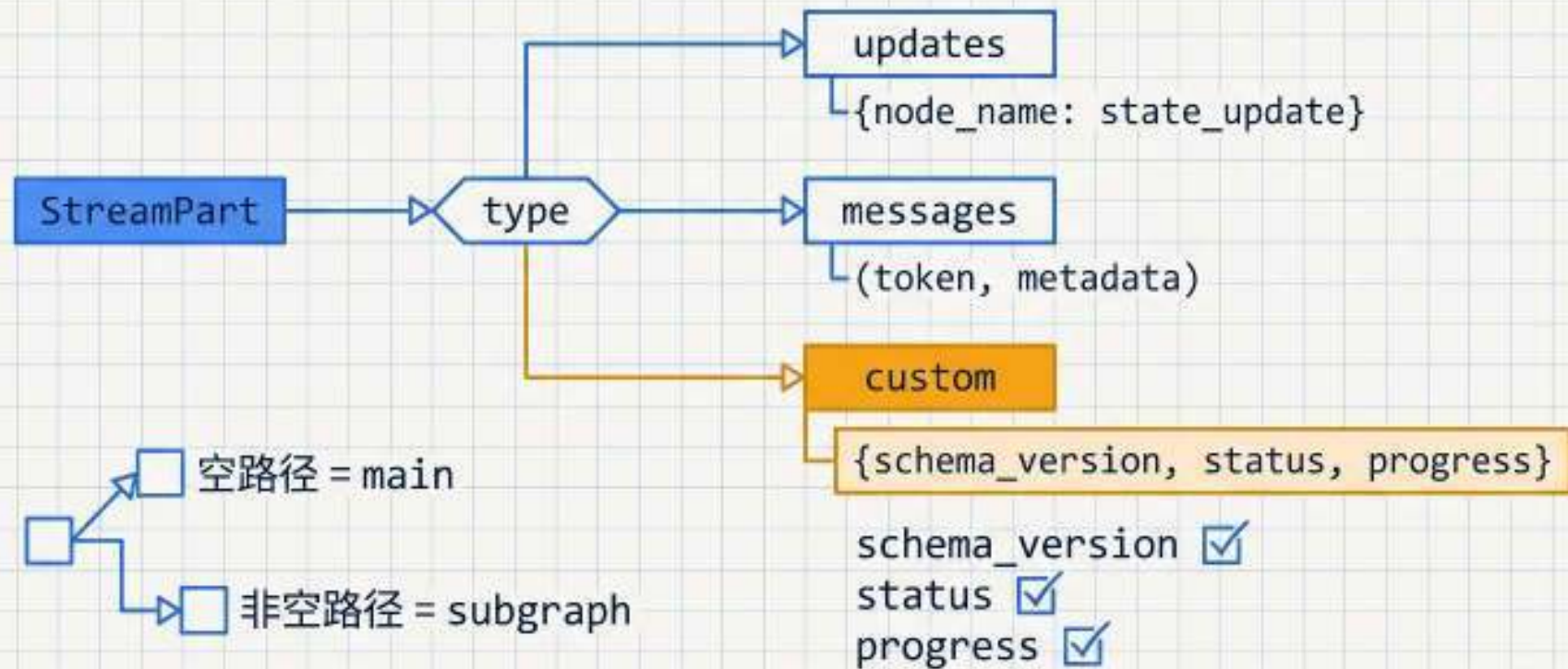
新页面优先 v3；迁移、底层调试与 custom updates 才保留 v2



不要混在同一个循环

v2 先看 type, 再按 ns 路由, 最后解释 data

data 没有固定形状; custom 事件更需要自定义 schema



拉取模板后，从 v3 projections 看见执行过程

两条命令拉取；一个入口观察 subagents 与 messages

01 · 拉取应用模板

```
uv tool install agentseek
```

```
agentseek create deepagents/streaming  
--checkout main
```



deepagents_streaming/

✅ create 成功 · 12 / 12 tests

默认模板目录，无需 --template-repo

02 · 观察 v3 event stream

```
graph.astream_events(..., version="v3")
```

subagents

name / path / status

嵌套 projections

messages

tools

subagents

messages

message.text

先让用户看见谁在工作， 再决定要看多深

