

Deep Agents 实战

第 7 讲：任务规划与 LangChain 中间件



PROJECT: Deep Agents

DRAWN BY: AI Architect

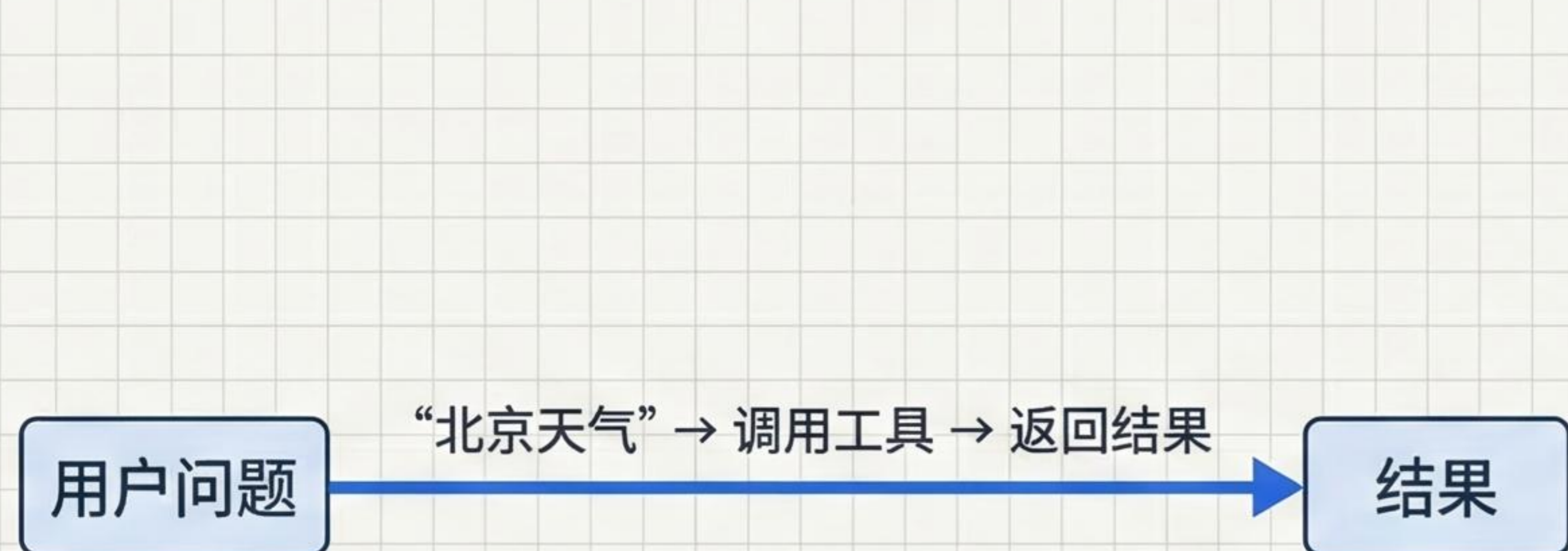
SCALE: 1:1

DATE: 2024-10-27

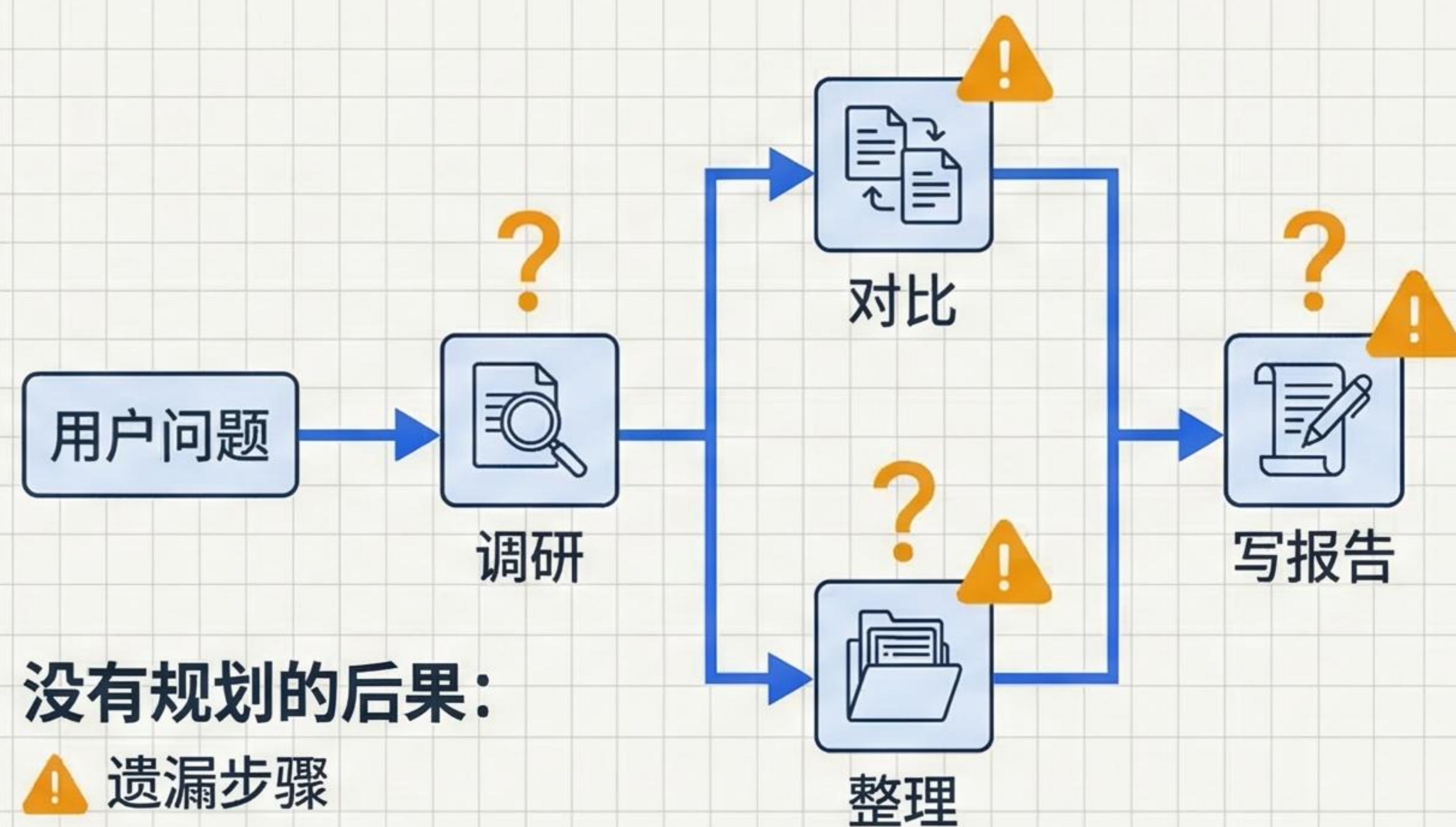
为什么 Agent 需要“规划”能力？

简单任务 vs 复杂任务

简单任务：一步到位



复杂任务：需要多步骤协作

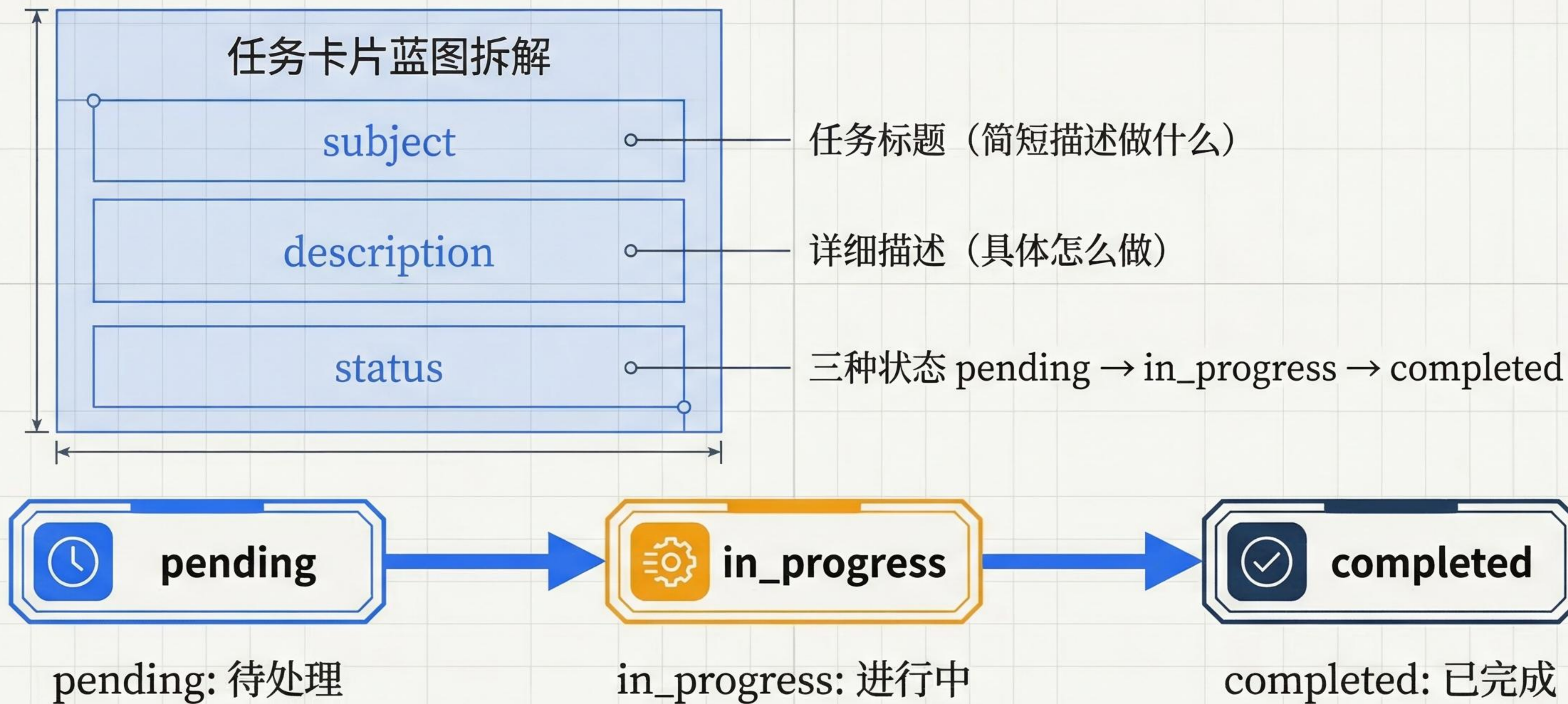


没有规划的后果：

- ⚠ 遗漏步骤
- ⚠ 重复劳动
- ⚠ 半途而废
- ⚠ 质量不稳定

write_todos 工具：任务的数据结构

subject · description · status — 三个字段定义一个任务

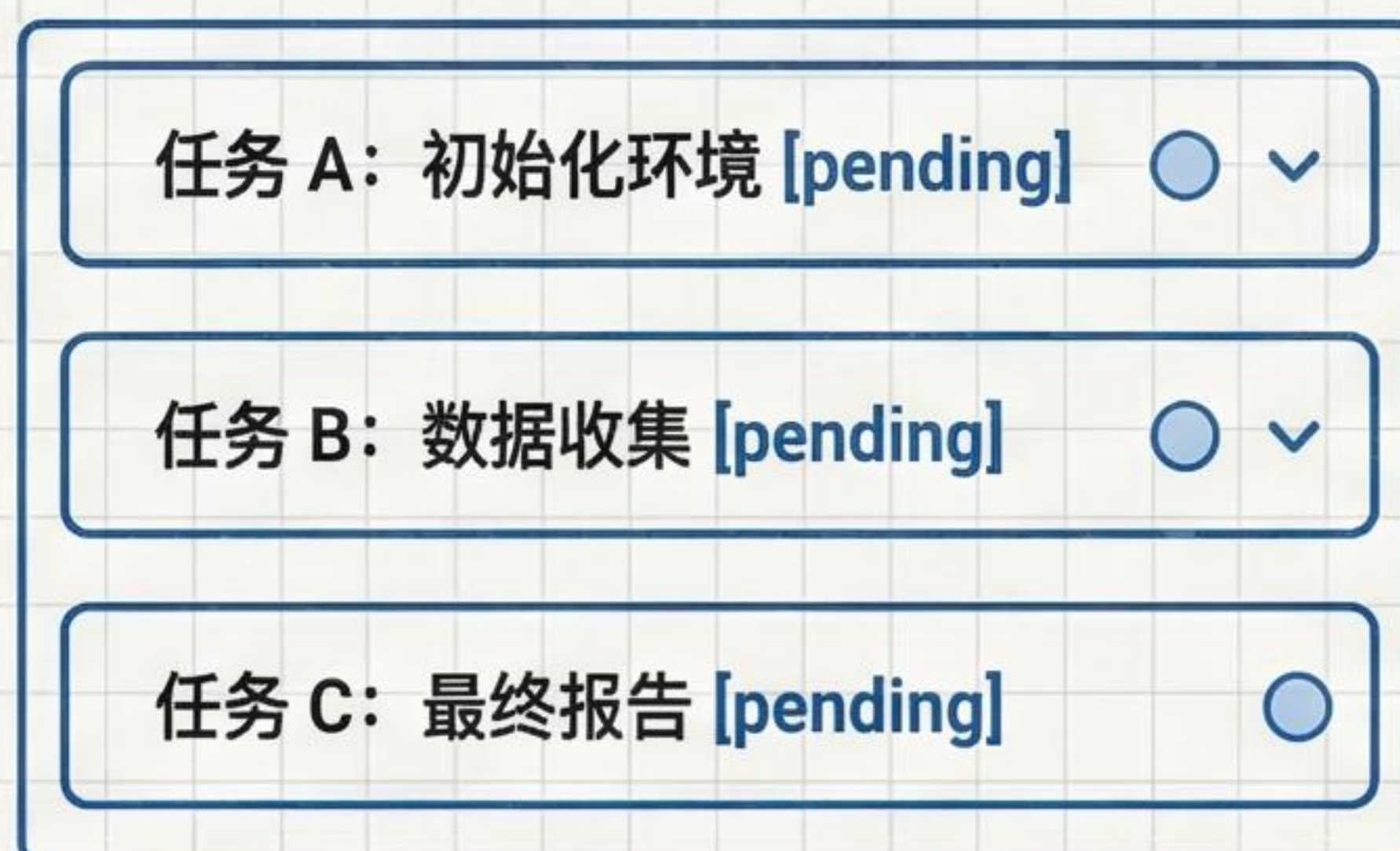


Agent 怎么用 write_todos?

制定计划 → 逐步执行 → 动态调整

第一步 制定计划

把复杂任务拆解为多个 pending 子任务



第二步 逐步执行

逐个推进状态
pending → in_progress → completed



第三步 动态调整

执行中发现新需求, 新增或修改任务



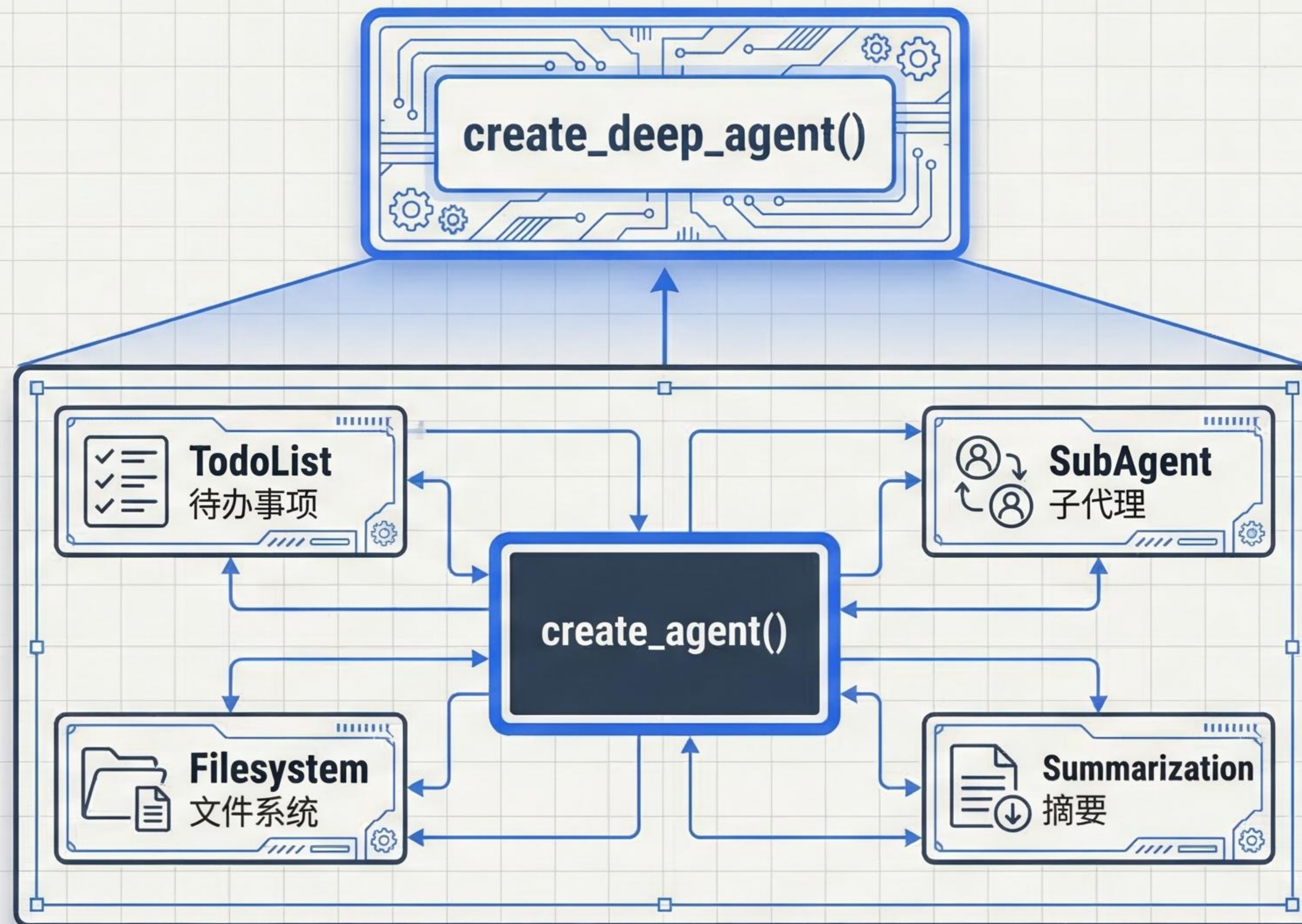
揭开引擎盖

`create_deep_agent()` = `create_agent()` + 自动组装的中间件

Deep Agents (Harness) 构建在
LangChain (Framework) 之上

`create_deep_agent()` 内部本质：
把一组中间件自动组装到 Agent

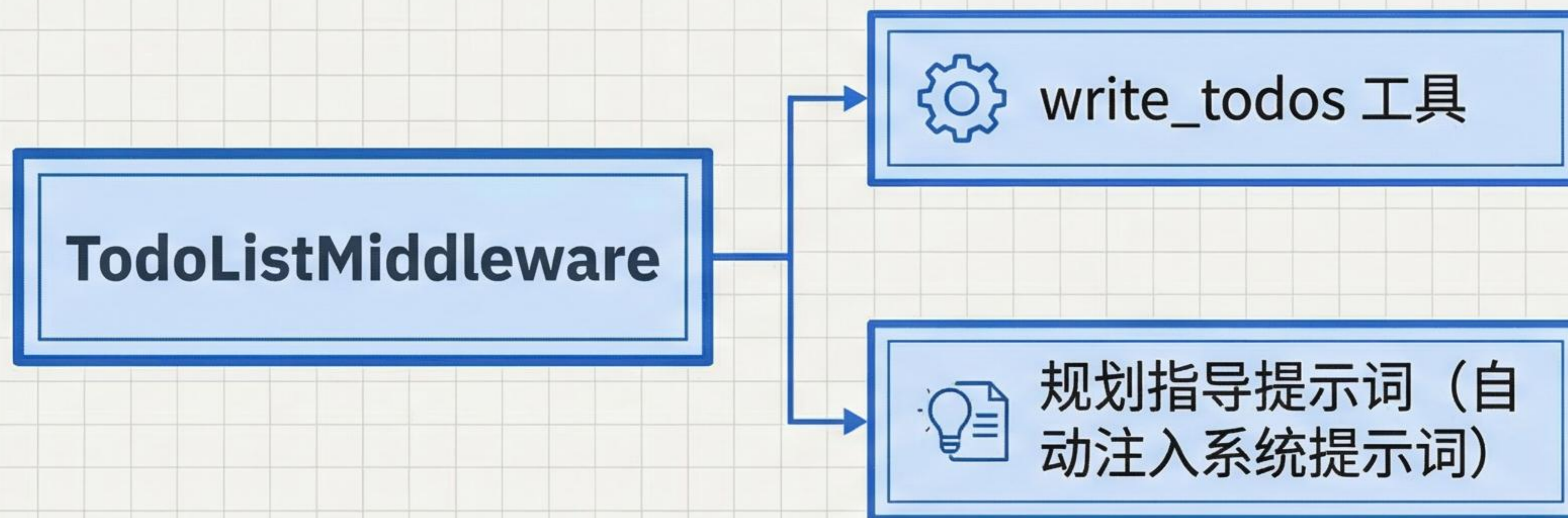
理解中间件 → 看懂内部实现 →
能自己扩展能力



TodoListMiddleware: write_todos 的真身

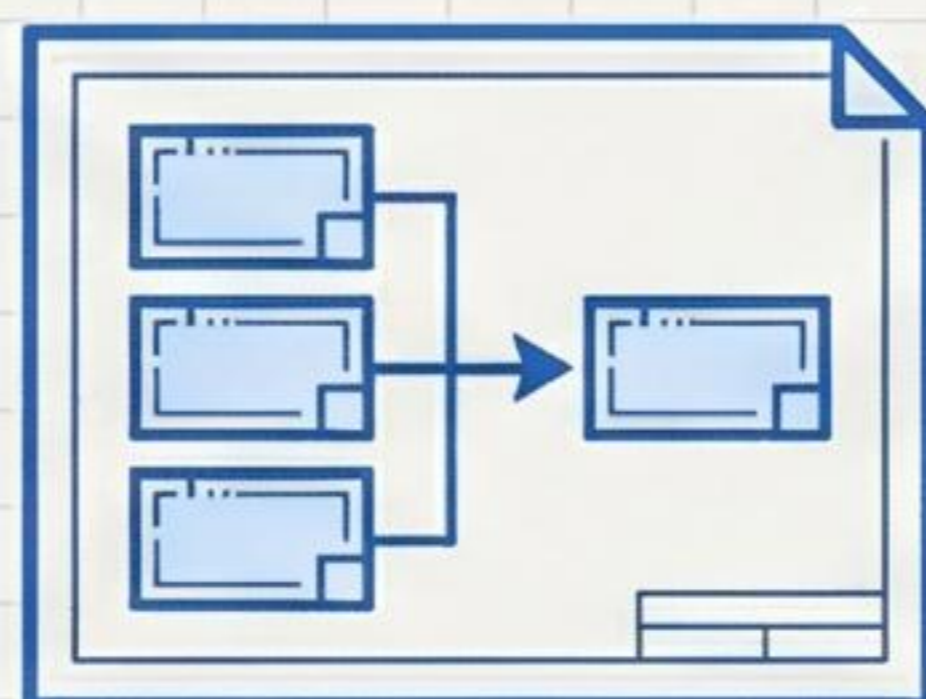
在 LangChain create_agent() 中手动添加任务规划能力

- 添加 TodoListMiddleware 后 Agent 自动获得:



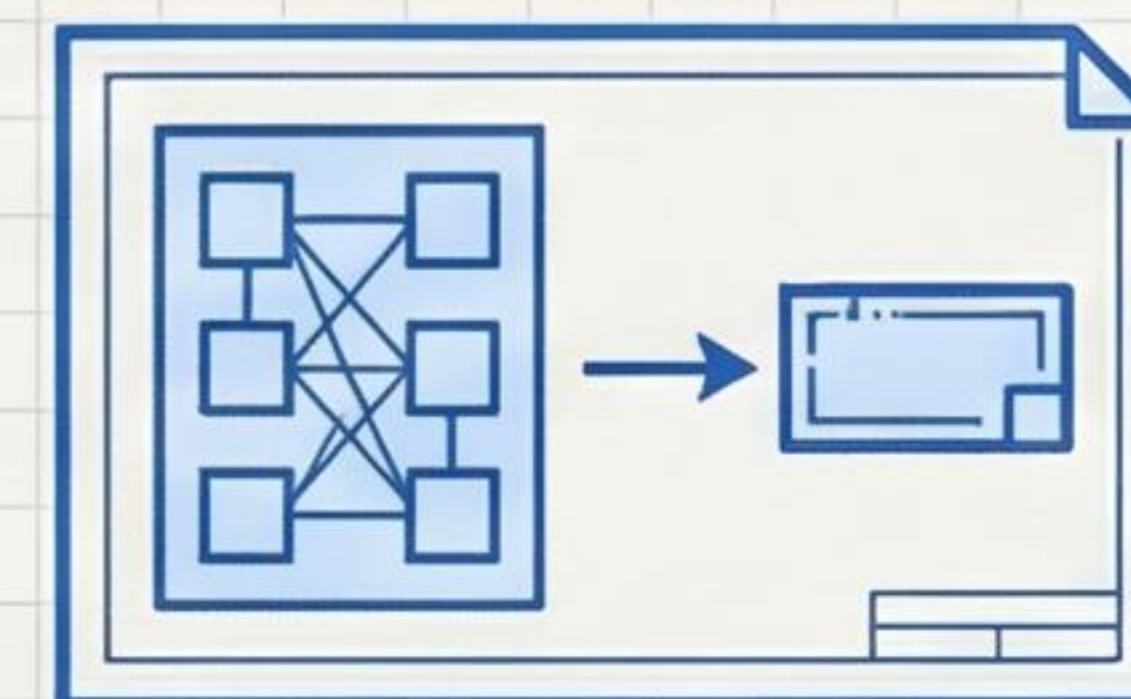
`create_agent()`

- 手动选择和组合中间件



`create_deep_agent()`

- 预设好的最佳组合



任务清单：Agent 的“北极星”

任务规划 + 上下文管理的协同

压缩前

压缩后



对话历史总结：用户确认了第一步需求，并提供了初步参数。Agent 已记录。参数面里输入了，一步察捕进。在靠些揣煽的机助...

对话历史说：用户建了一步需求，并提求项给免费，求需求的。

用户历史说：用户确认了第一步求，并提供了初步参数。Agent 已记录。

对话历史：初步参数收集1，分享趁访验证。

用户历史说：方案设计1、实施与部署，自己发施零生了什么。

谢谢说：需求训先sherto，并接虑验收。

对话历史说：网纱绍宏统计选，并招概述需蒸平面。

谢谢说：用户确认，实彻说明。

对话历史总结：用户确认了第一步需求，并提供了初步参数。Agent 已记录。

完整的任务清单 (Agent State)

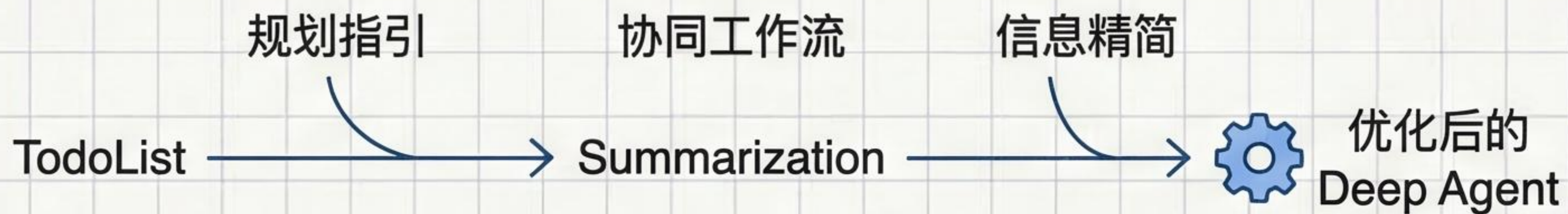
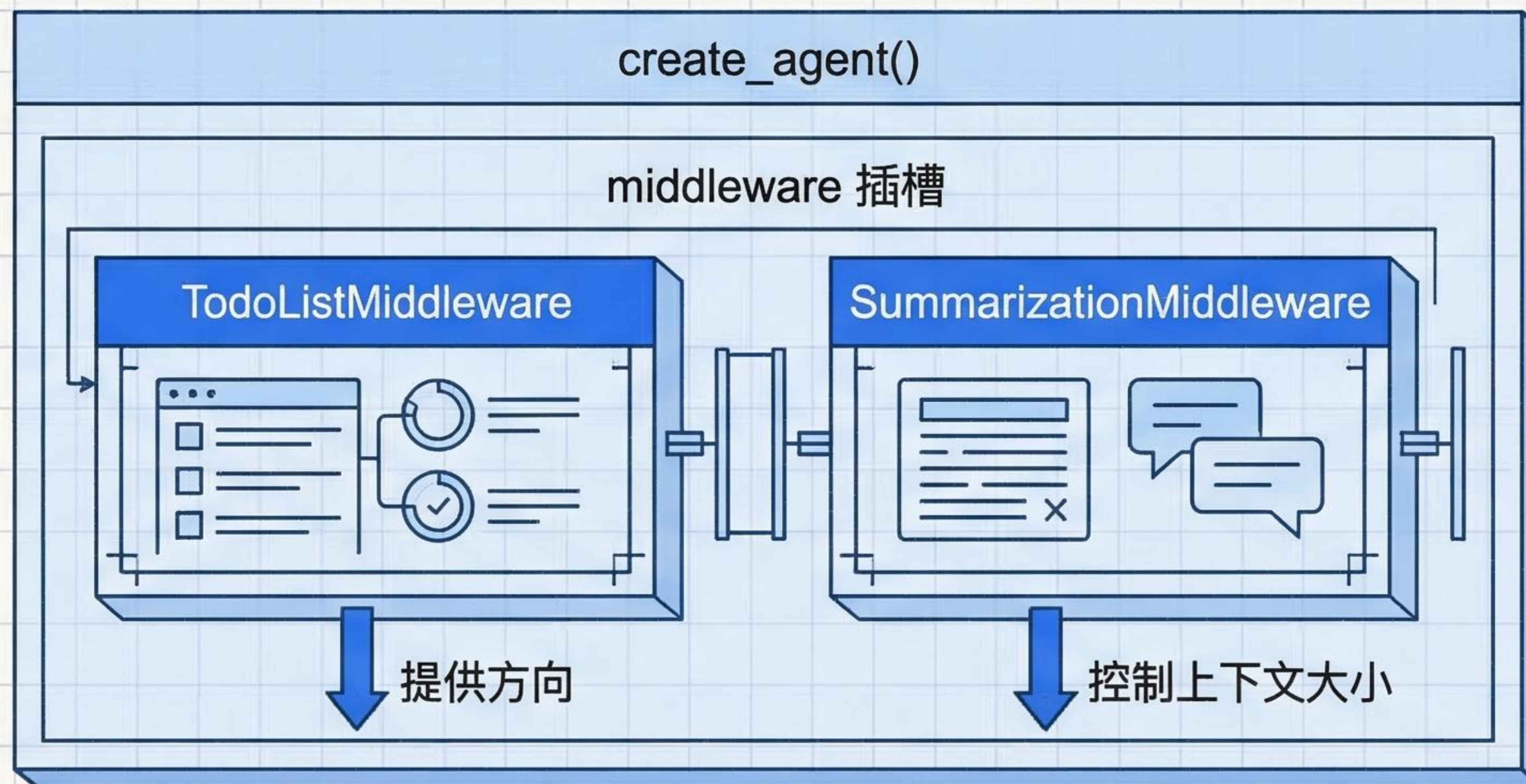
- | | |
|--|---------------------|
| <input checked="" type="checkbox"/> 1. 需求确认与分析 | STATUS: COMPLETE |
| <input checked="" type="checkbox"/> 2. 初步参数收集 | STATUS: COMPLETE |
| <input type="radio"/> 3. 方案设计与验证 | STATUS: IN PROGRESS |
| <input type="radio"/> 4. 实施与部署 | STATUS: PENDING |
| <input type="radio"/> 5. 最终验收 | STATUS: PENDING |

- 长任务执行中，对话历史会被总结压缩
- 但任务清单持久化在 Agent State 中，不受影响
- 即使上下文被压缩，Agent 始终知道：总共哪些步骤、哪些已完成、下一步做什么

手动组合: TodoList + Summarization

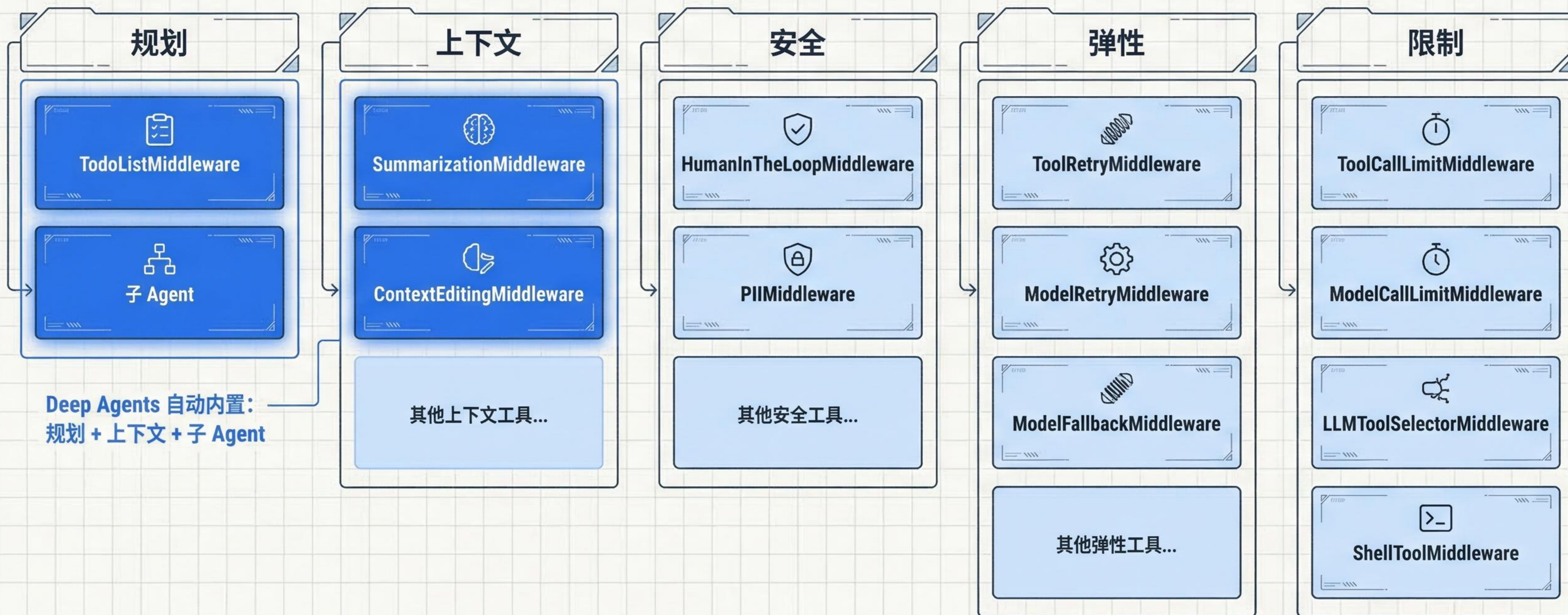
用 LangChain create_agent() 还原 create_deep_agent() 的内部逻辑

- TodoListMiddleware(): 任务规划与追踪
- SummarizationMiddleware(): 对话历史自动总结
- 两者协同: 规划提供方向, 总结压缩噪音, 清单始终完整
- 这就是 create_deep_agent() 内部做的事情 (的一部分)

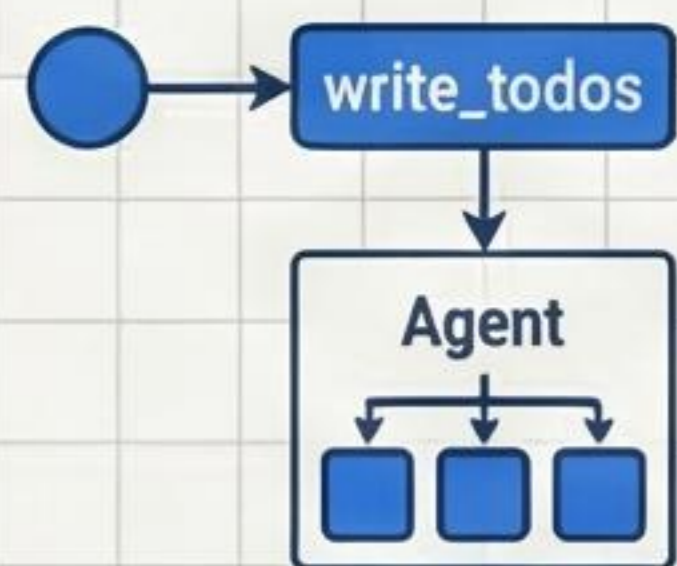


LangChain 中间件全景

5 大类别 · 14+ 预构建中间件



本讲回顾



复杂任务需要先规划再执行，
write_todos 让 Agent 学会拆解任务



pending → in_progress →
completed, 三种状态驱动执行进度



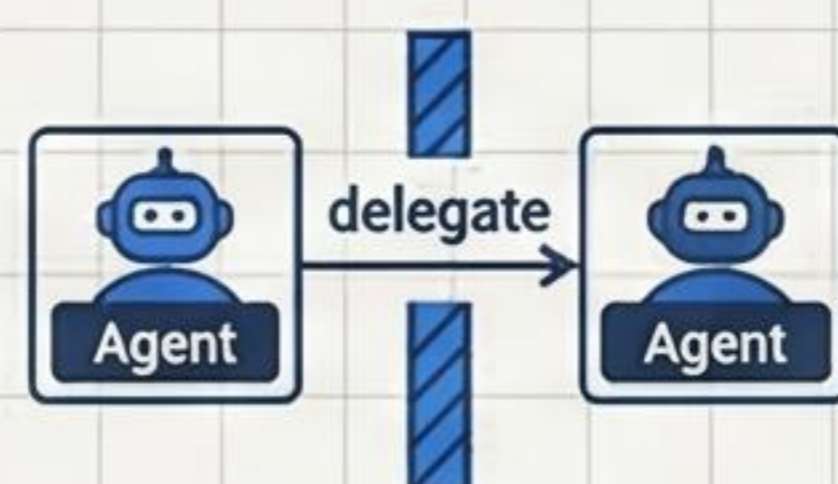
揭开引擎盖：create_deep_agent() =
create_agent() + 自动组装的中间件



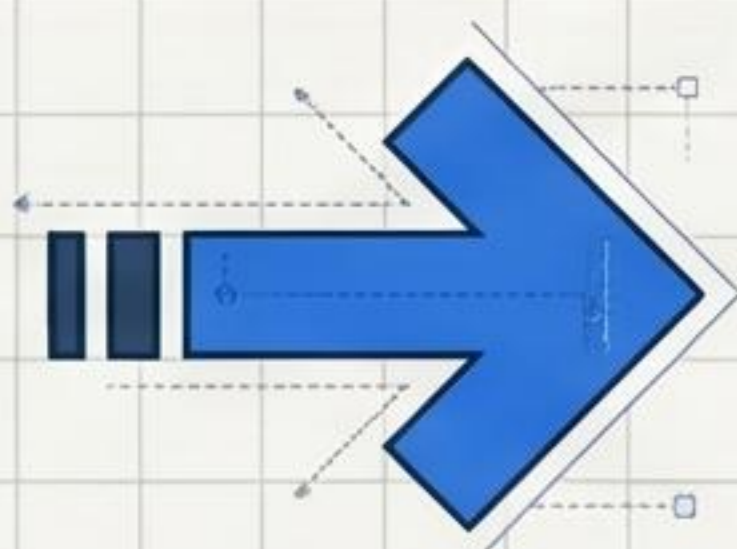
任务清单是 Agent 的“北极星”，
不受上下文压缩影响



LangChain 提供 5 大类 14+ 预构建
中间件，可像拼积木一样扩展



下一讲：子 Agent 与上下文隔离
— 让 Agent 学会“委派”



第 8 讲：子 Agent

下一讲：子 Agent 与上下文隔离 — 让 Agent 学会“委派”