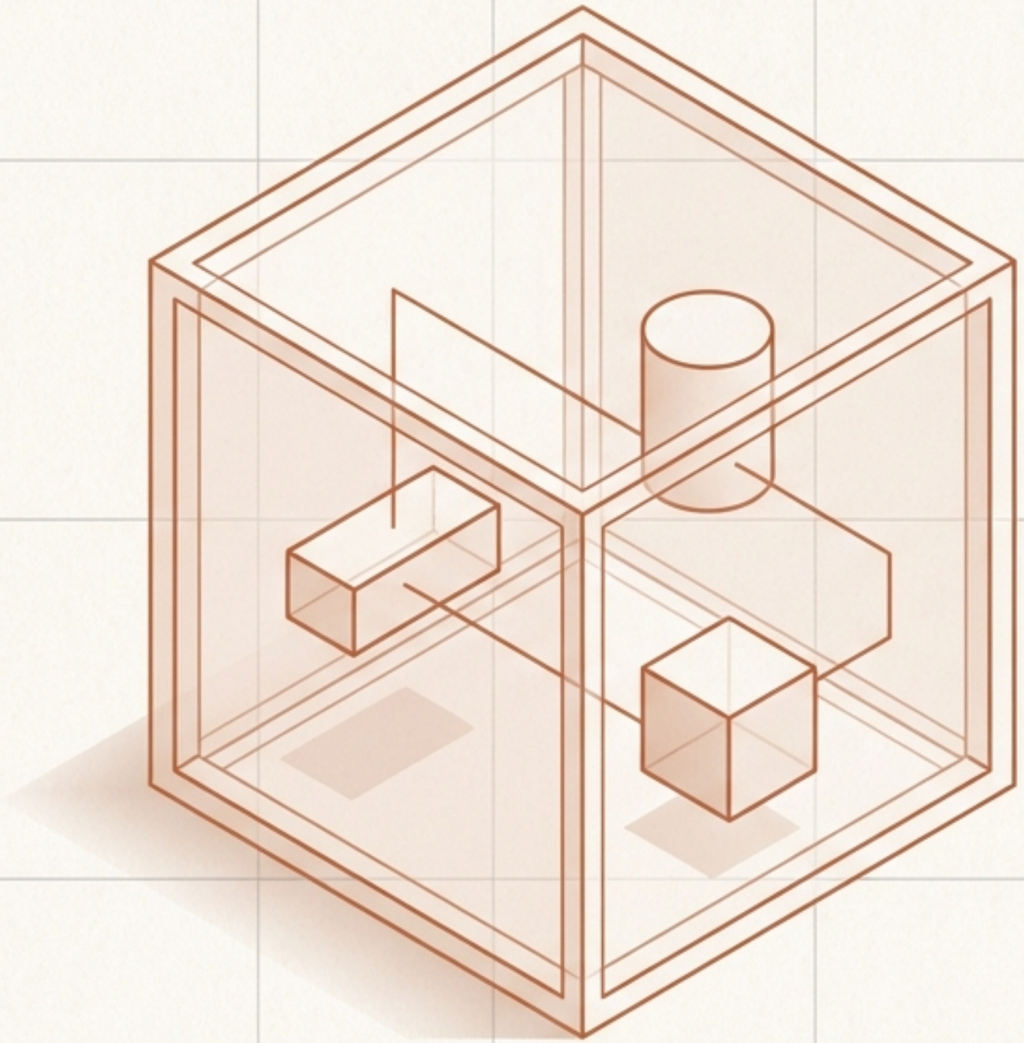


交互架构：多 Agent 赛道缺失的平台层

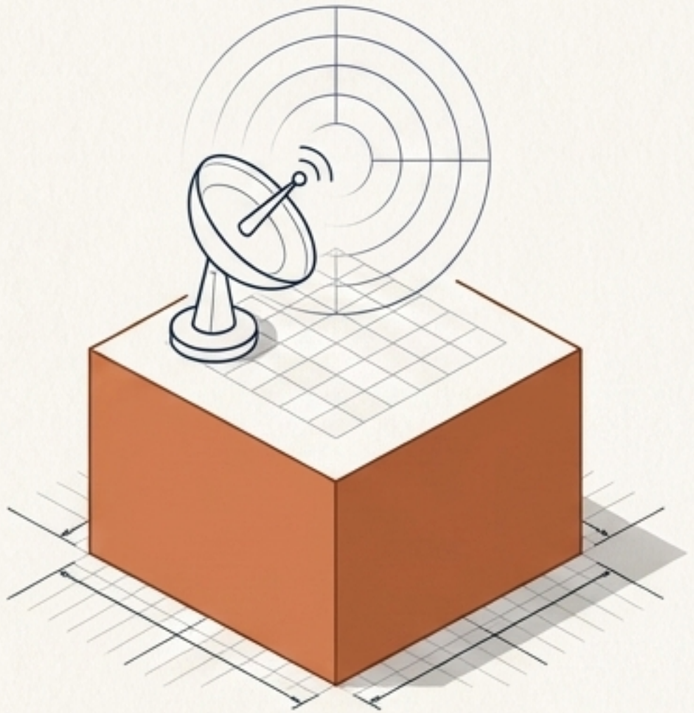
二十个开源项目的调研启示与大一统模型

巨大的需求，尴尬的现实

让三个顶级 AI 坐在一起辩论话题或跑一局狼人杀？GitHub 上有超过二十个相关项目，头部星数高达六万。

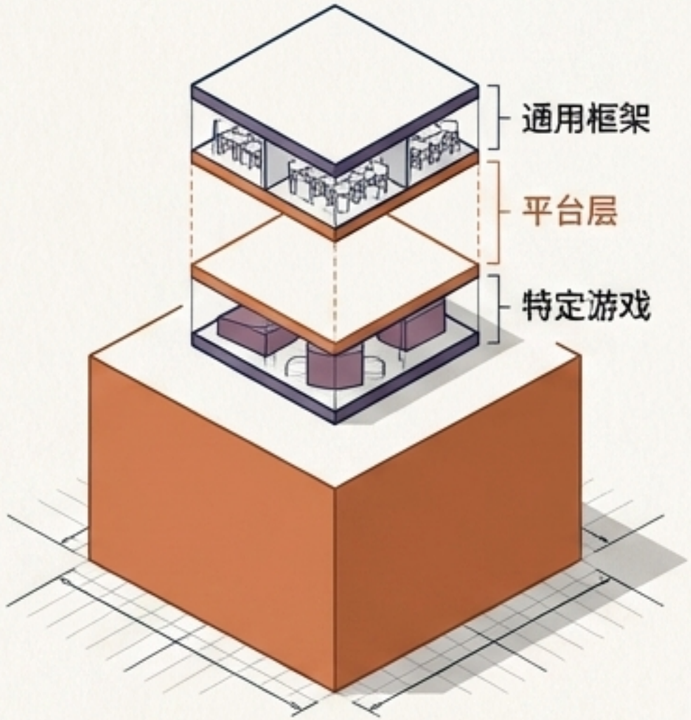
- 但经过实际运行测试，没一个能直接用。
- 爆款 Demo 层出不穷，但每一次都是无法复用的“一次性 Hack”。

多 Agent 交互赛道的三个核心判断



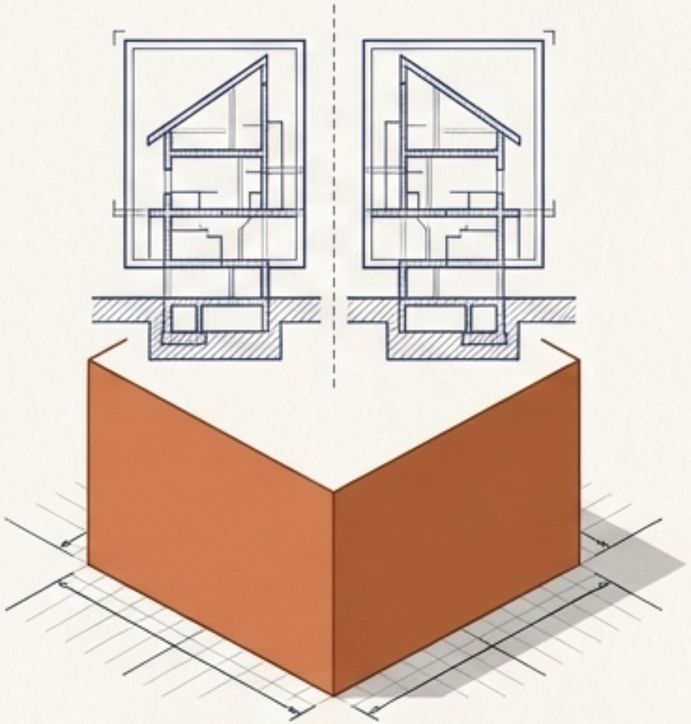
1. 赛道几乎是空的

没有任何一个专注多 Agent 游戏或交互的开源项目超过 100 星。



2. 两端拥挤，中间断层

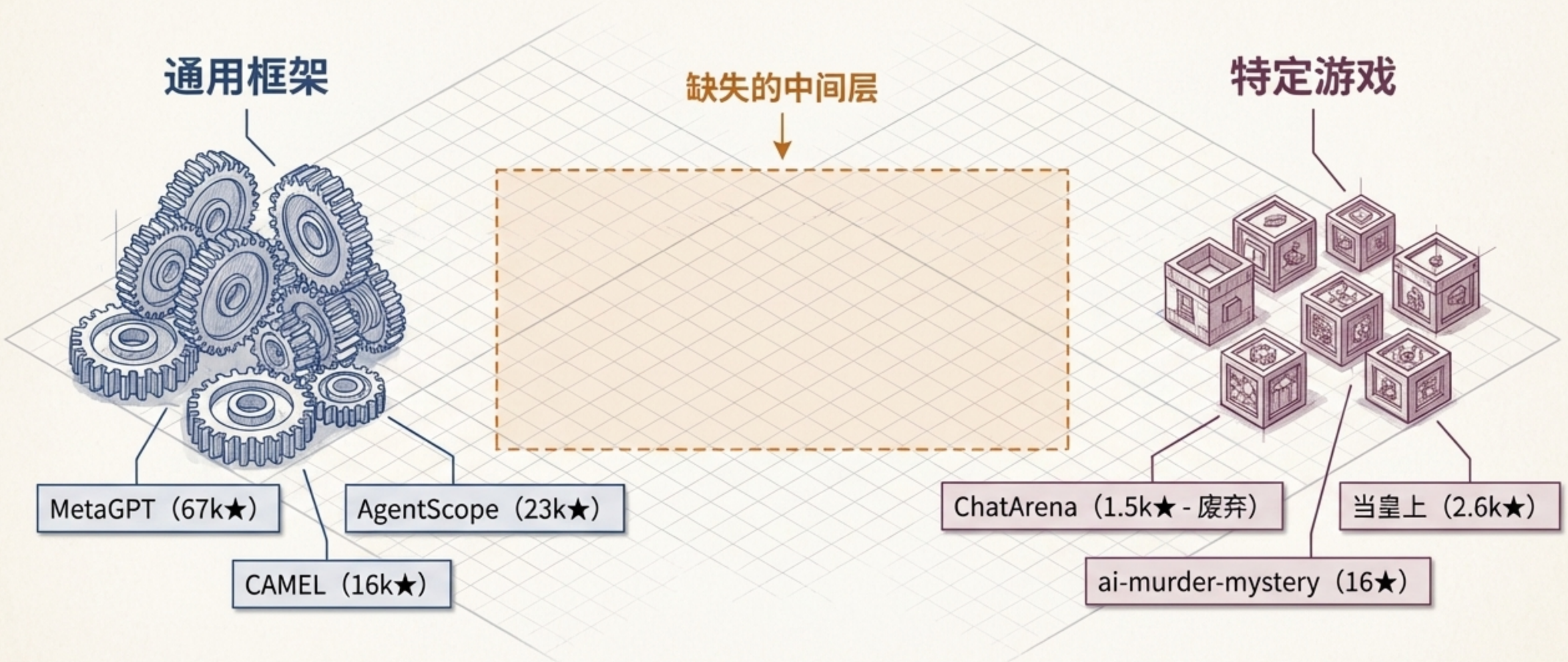
通用框架和特定游戏在走极端，中间缺失一个至关重要的“平台层”。



3. 大一统的基础设施

做好多人游戏和做好多人协作，底层所需的基础设施完全一模一样。

两头拥挤的赛道与缺失的中间层

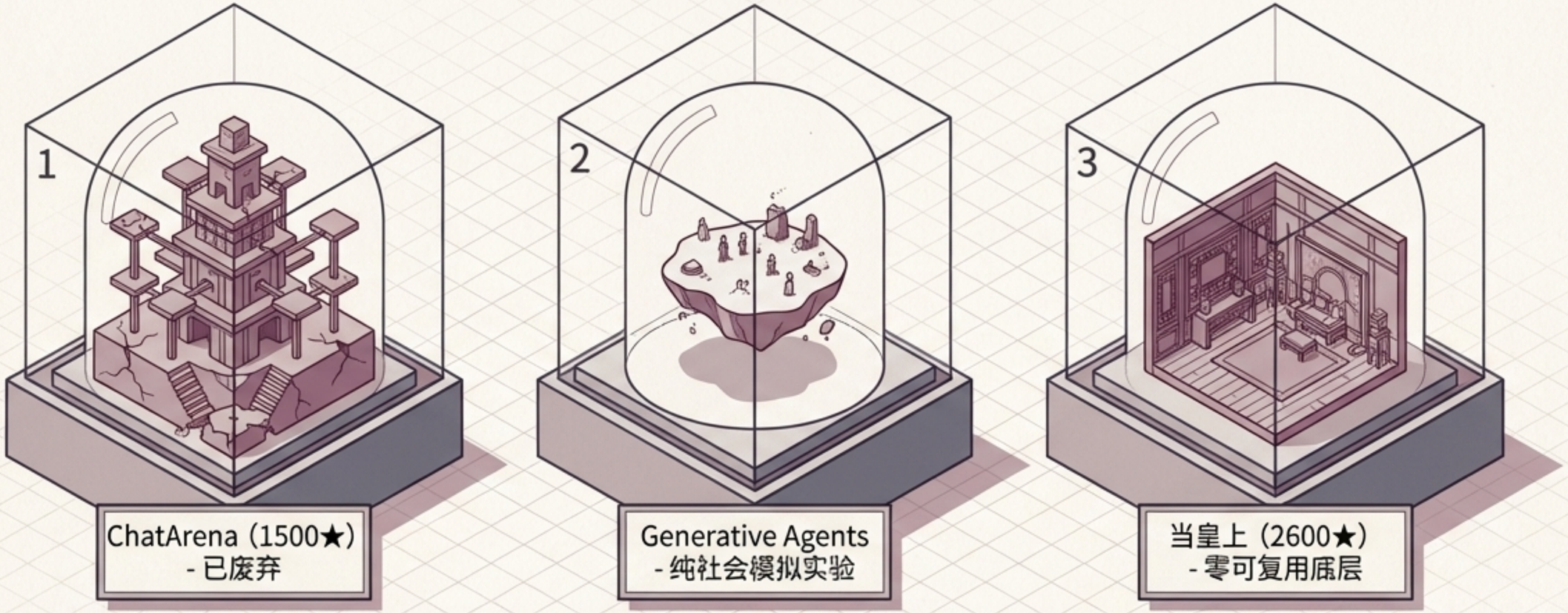


开源生态呈现两极分化。左侧是极其底层、无模式抽象的通用框架；右侧是高度定制、缺乏复用底层的一次性游戏应用。没有人为交互场景提供开箱即用的容器。

通用框架的局限：缺失“房间”概念

- 尽管拥有极佳的抽象和消息分组能力，但底层框架解决的只是“怎么让多个 Agent完成一个任务”。
- 痛点：MsgHub 仅是代码级别的上下文管理器。
- 结果：无法通过点击鼠标创建“狼人杀房间”或动态拉入 Agent。一切都需要硬写 Python 代码。

游戏专项的局限：无法复用的单点 Hack



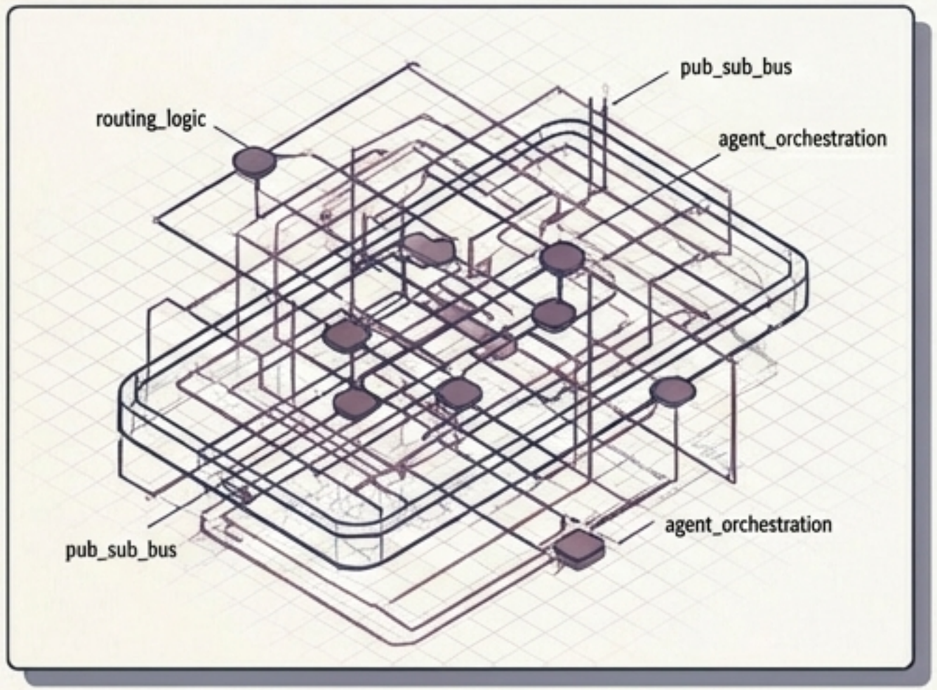
- 聚焦特定场景的项目往往设计精巧，但由于缺乏可复用的平台层支撑，最终要么走向废弃，要么沦为孤立的单场景演示。
- 没有任何超过 100 星的活跃交互开源项目。

多 Agent 市场全景诊断矩阵

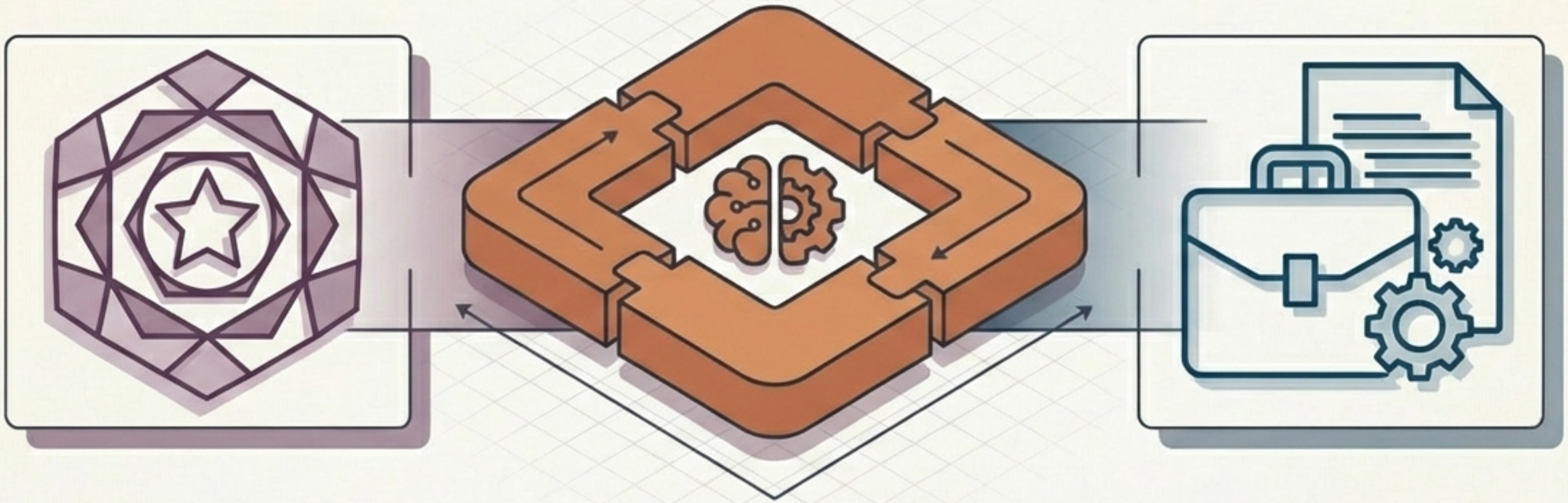
类别	典型代表（Stars）	核心痛点	房间/容器概念	可复用底层
通用框架	AgentScope（23k），MetaGPT（67k）	面向工作流/工程，太底层	无（需手写代码）	强
游戏专项	ChatArena（1.5k），当皇上（2.6k）	单场景 Hack，已废弃或孤立	强（但写死）	无
剧本杀/TRPG	jubensha-ai（89），ai-murder-mystery	极度细分，活跃度低	半固定	无
闭源参考	阿里 Accio Work	闭源不可用	完美（群聊隐喻）	强

UX 突破口：“群聊”作为交互隐喻

- 闭源项目揭示了最重要的产品设计参照：群聊式多 Agent 交互。
- 不需要向用户解释什么是 “Agent 编排” 或 “消息路由”。建一个 Agent 团队就像建微信群，分配任务就像发消息。这是解决多

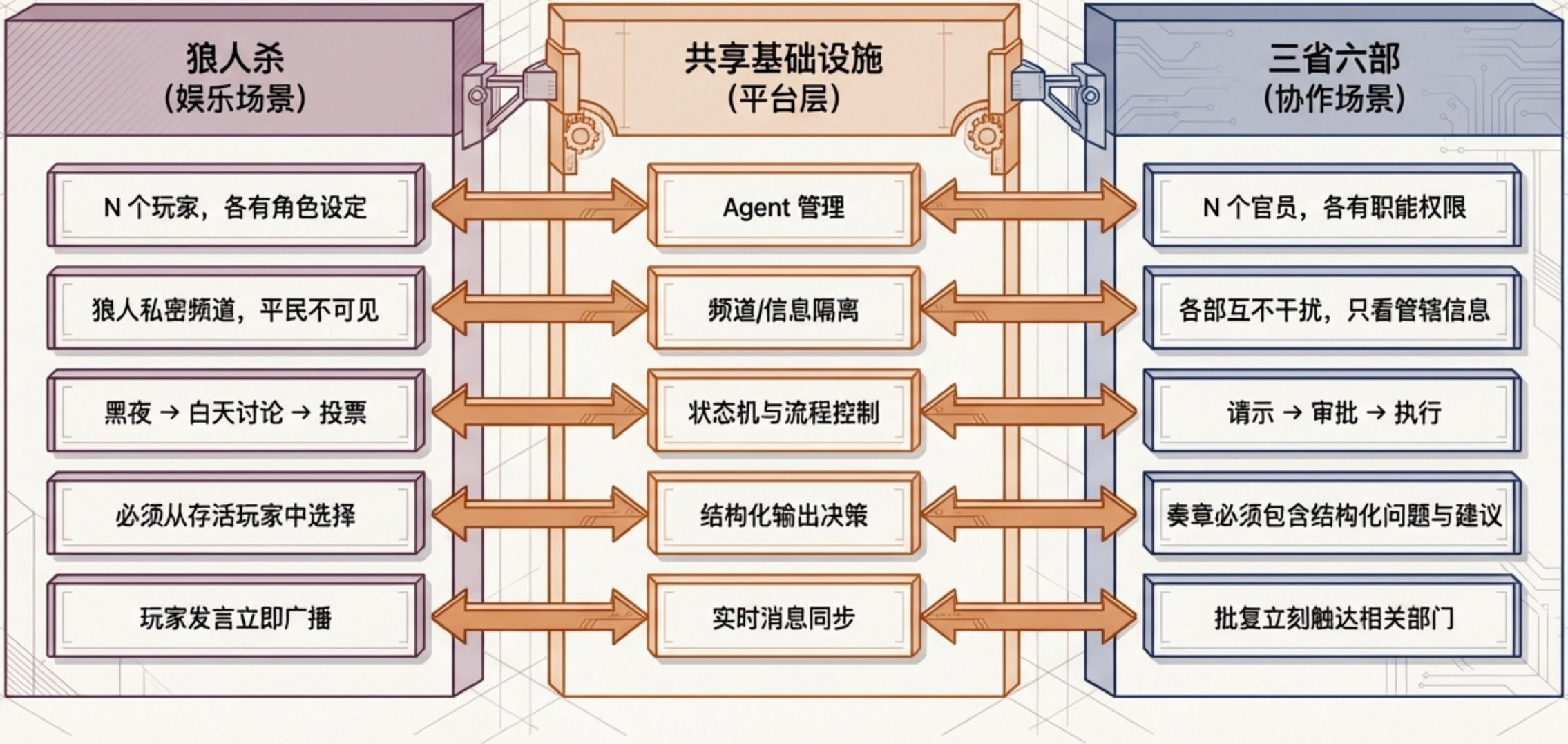


调研中最震撼的发现：

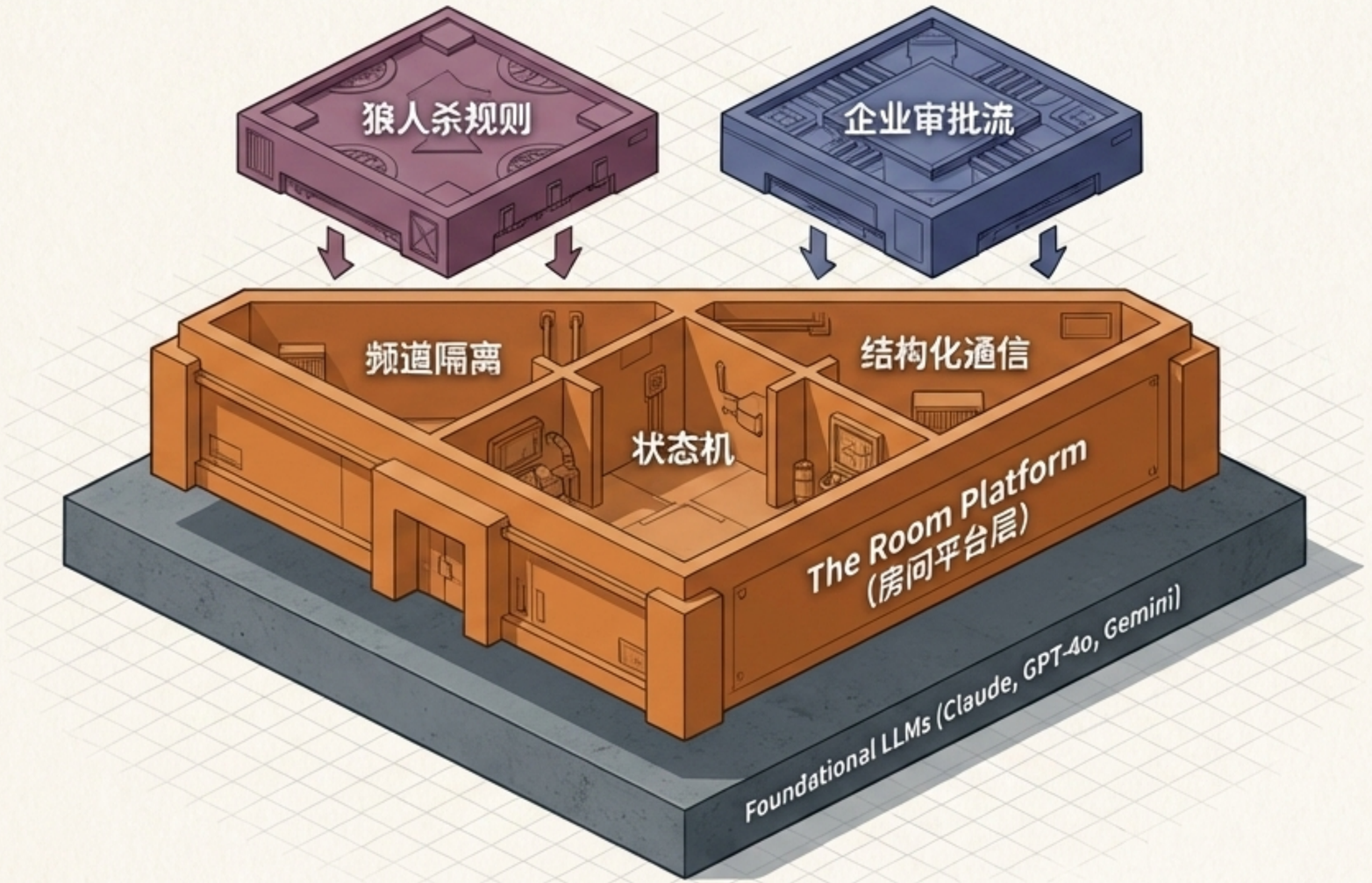


好游戏和好协作，做的是同一件事。

多 Agent 大一统矩阵：业务逻辑的高度同源



架构重塑：插拔式的平台层

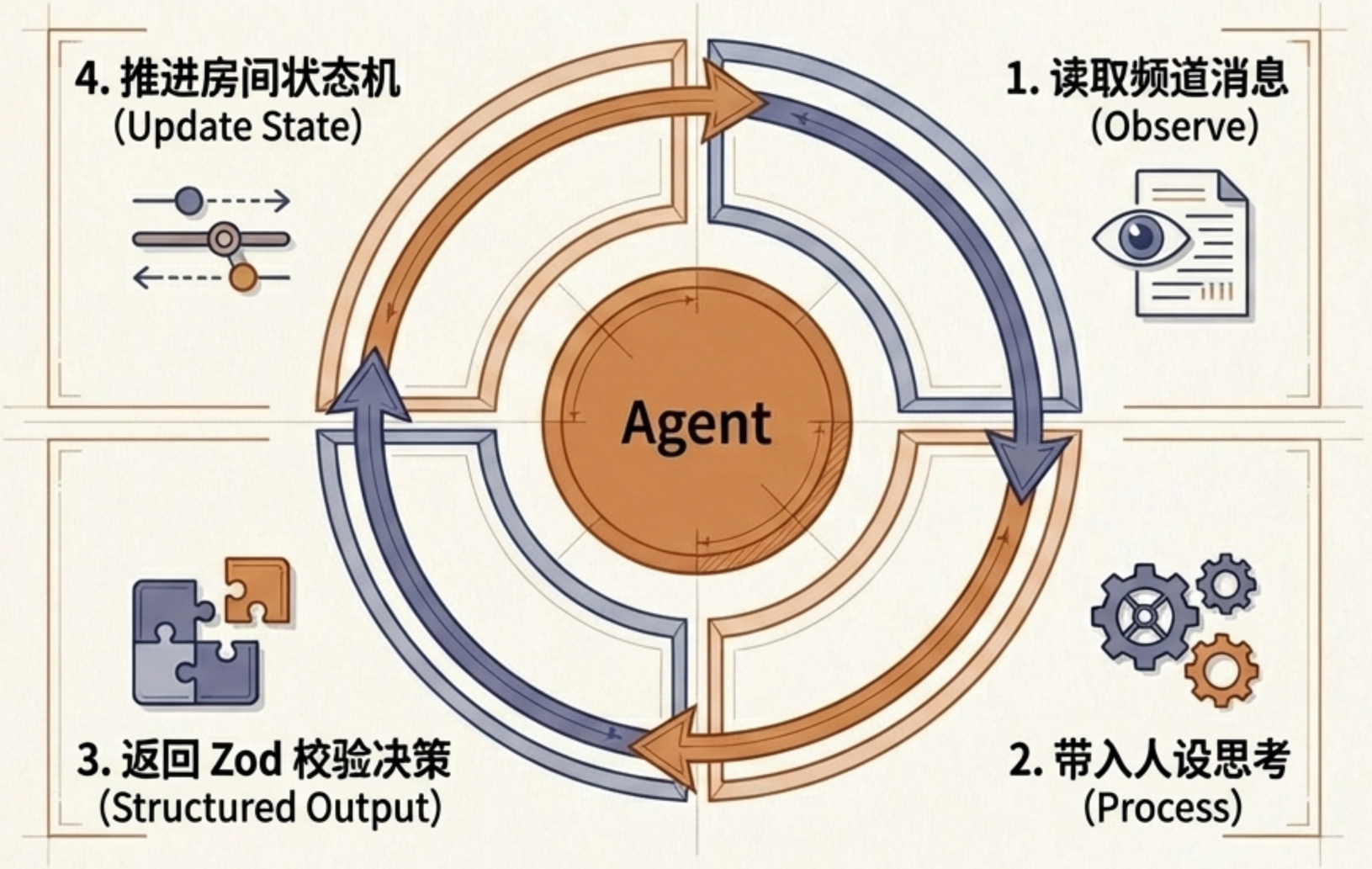


差异只在规则配置，基础设施完全复用。

只要构建一次中间的“房间”平台层，核心能力（隔离、通信、流程）就全部就绪。

在上层插上“狼人杀”规则卡带，它就是游戏；插上“企业审批”卡带，它就是专业组织模拟。

基础设施微观解析：Agent 运行的循环机制



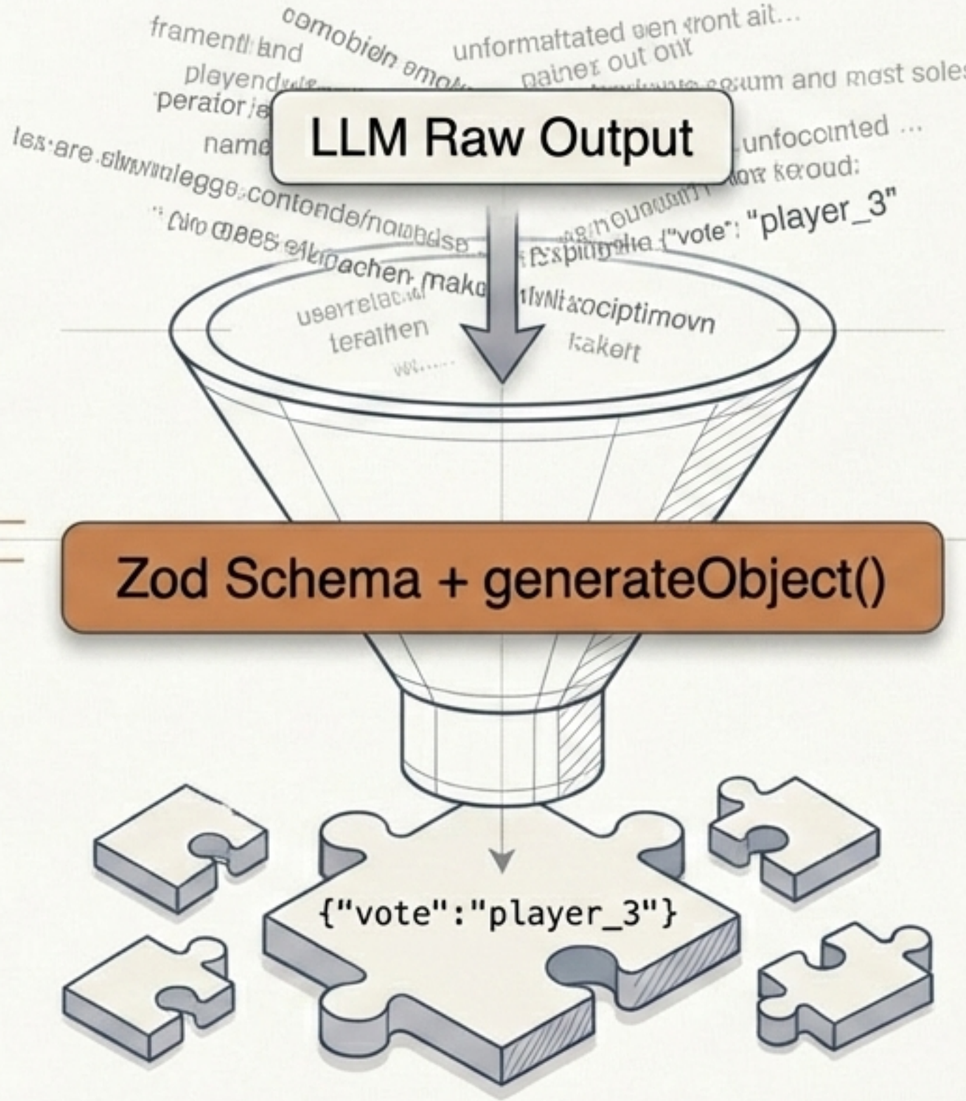
在统一的平台层下，无论是投票淘汰还是盖章审批，Agent 的每一次行动都被标准化为极其严谨状态机循环。

平台负责物理驱动此循环，彻底避免模型偏离规则。

为什么是现在？(1/3) 模型能力越过关键阈值

两年前，让 9 个 Agent 各扮各的角色跑完一局狼人杀是不可能的。
今天，最新模型的指令遵循能力已经彻底跨越了长文本角色扮演的物理阈值。

为什么是现在？(2/3) 结构化输出达到生产就绪

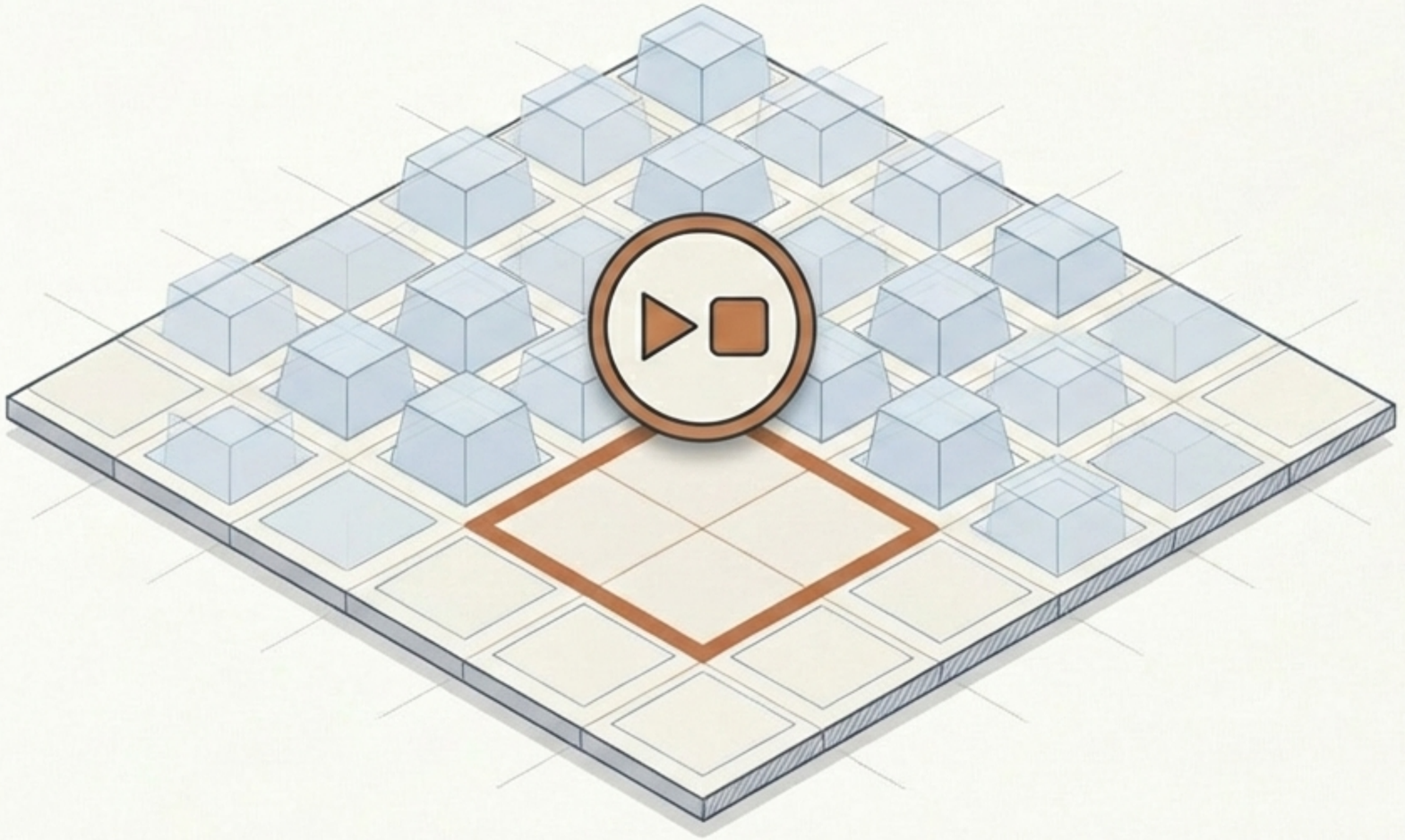


过去靠 Prompt 乞求模型：“请只在存活玩家中投票”。

现在通过类型系统（Type System）在物理层面约束 Agent 的决策空间。

投票只能输出存活玩家的 ID，Agent 物理上无法选择其他人。多 Agent 游戏从“大概率能跑通”质变为“确定能跑通”。

为什么是现在？(3/3) 绝对真空的市场与分发红利



1. 生态真空：ChatArena 废弃，通用框架不做纯交互，专注交互的平台层是 0 竞争。

2. 分发优势：AI Agent 互相博弈、辩论的过程，天然是绝佳的病毒视频素材与分享内容。

下一步决策：面对拥有两万星的成熟开源框架，
想要打造真正的交互架构，我为什么选择不 Fork？（未完待续）