

QCon
全球软件开发大会

前端工程 3.0

企业级智能研发与 Agent 系统落地

郭华翔

支付宝高级前端技术专家



📍 北京

QCon

全球软件开发大会

会议时间：4月10-12日

- 大模型赋能 AI Ops
- 越挫越勇的大前端
- 云上业务架构演进
- 多模态大模型及应用
- 海外 AI 应用创新实践

📍 北京

AiCon

全球人工智能开发与应用大会

会议时间：6月27-28日

- 端侧智能
- AI Agent
- 多模态大模型
- 金融+大模型

📍 上海

QCon

全球软件开发大会

会议时间：10月23-25日

- AI Agent
- 大模型训练推理
- 端侧 AI
- 搜推深度融合
- 数智金融

4月

5月

6月

8月

10月

12月

📍 上海

AiCon

全球人工智能开发与应用大会

会议时间：5月23-24日

- 通用大模型
- AI Agent
- 垂直领域应用
- 模型可解释性

📍 深圳

AiCon

全球人工智能开发与应用大会

会议时间：8月22-23日

- 模型效率与部署
- 多模态大模型
- 大模型安全
- 智能硬件

📍 北京

AiCon

全球人工智能开发与应用大会

会议时间：12月19-20日

- 通用大模型
- 智能硬件
- LM Ops
- 具身智能

极客邦科技 2025 年会议规划

促进软件开发及相关领域知识与创新的传播



参会咨询



查看会议

目录

01

前端工程在 AI 时代的演进

02

上下文工程 – 企业级前端研发的上下文管理

03

智能的工具体系 – 让工具具备智能

04

智能工作流 – 基于 Multi-Agent 的 Agentic Workflow 实现

05

总结与未来展望

01

前端工程在 AI 时代的演进

前端工程 3.0 背景及智能研发的思考

企业级前端研发领域面临的挑战

前端领域知识的复杂性

前端基础领域知识繁杂且灵活，领域知识多样
(如基础框架、业务框架、领域知识、组件文档...)

研发工具的碎片化

技术栈带来的丰富多样的工具链复杂且不统一
(如编码风格、构建打包、调试预览、发布部署...)

企业级前端研发

面向用户的高标准还原

2C 产品对还原度和体验要求更高，且需持续迭代
(像素级还原、交互复杂多样、代码可维护性...)

自动化质检难以落地

UI、代码风格差异大，自动化手段成本高效果差
(高体验+高质量要求，AI 产物的鲁棒性欠佳...)

Software Engineering

Frontend Engineering

前端工程在 AI 时代的演进 – 前端工程 3.0



Software Engineering

Frontend Engineering

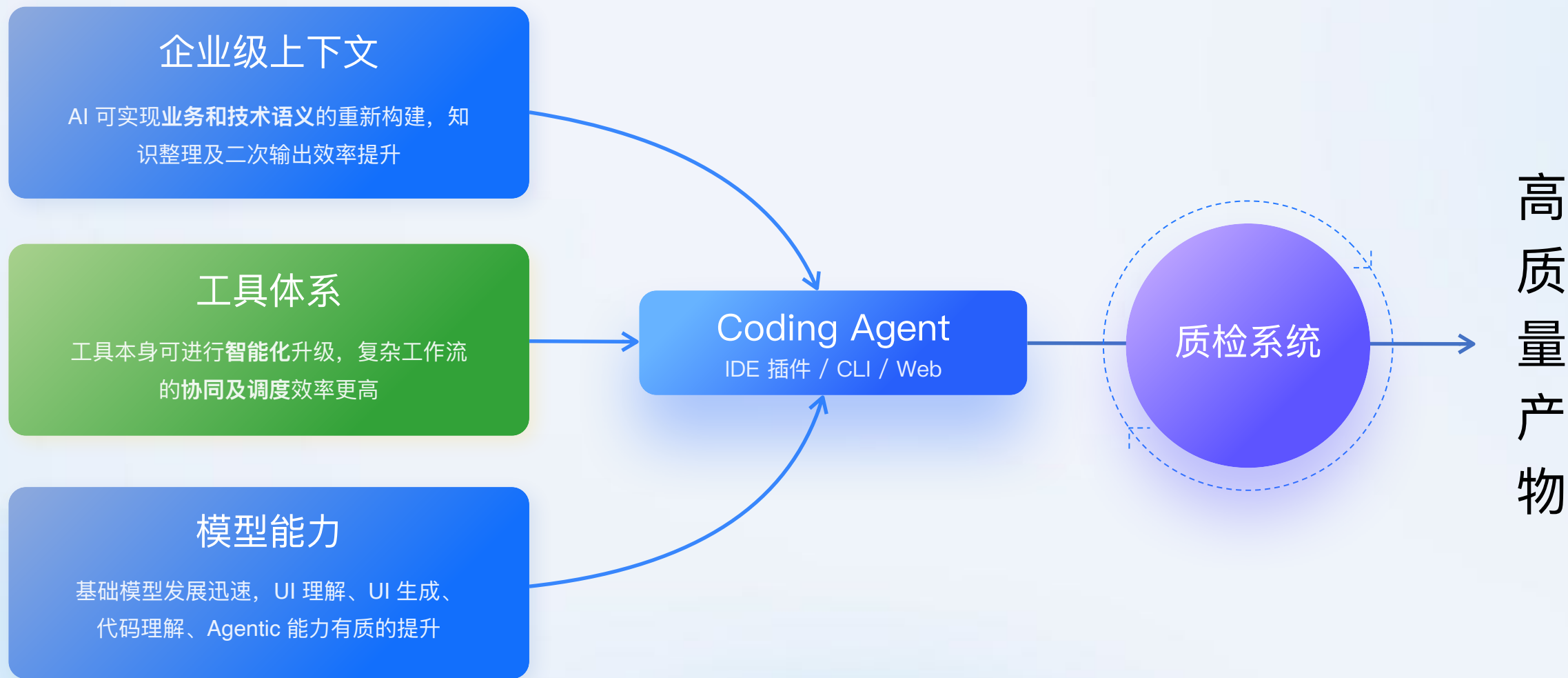
前端工程在 AI 时代的演进 – 前端工程 3.0



AI for Software Engineering

AI for Frontend Engineering

AI 时代解决前端工程问题的机会



● 核心原则及关键解题思路

企业级 Coding Agent

需要解决的三大核心问题

上下文工程

企业级前端研发的上下文

理解前端的技术语义，提升复杂项目的代码生成效果

智能的工具体系

让工具具备智能

完成单个研发环节的智能化升级，同时覆盖更多研发节点

智能工作流

基于 Multi-Agent 的 Agentic Workflow 实现

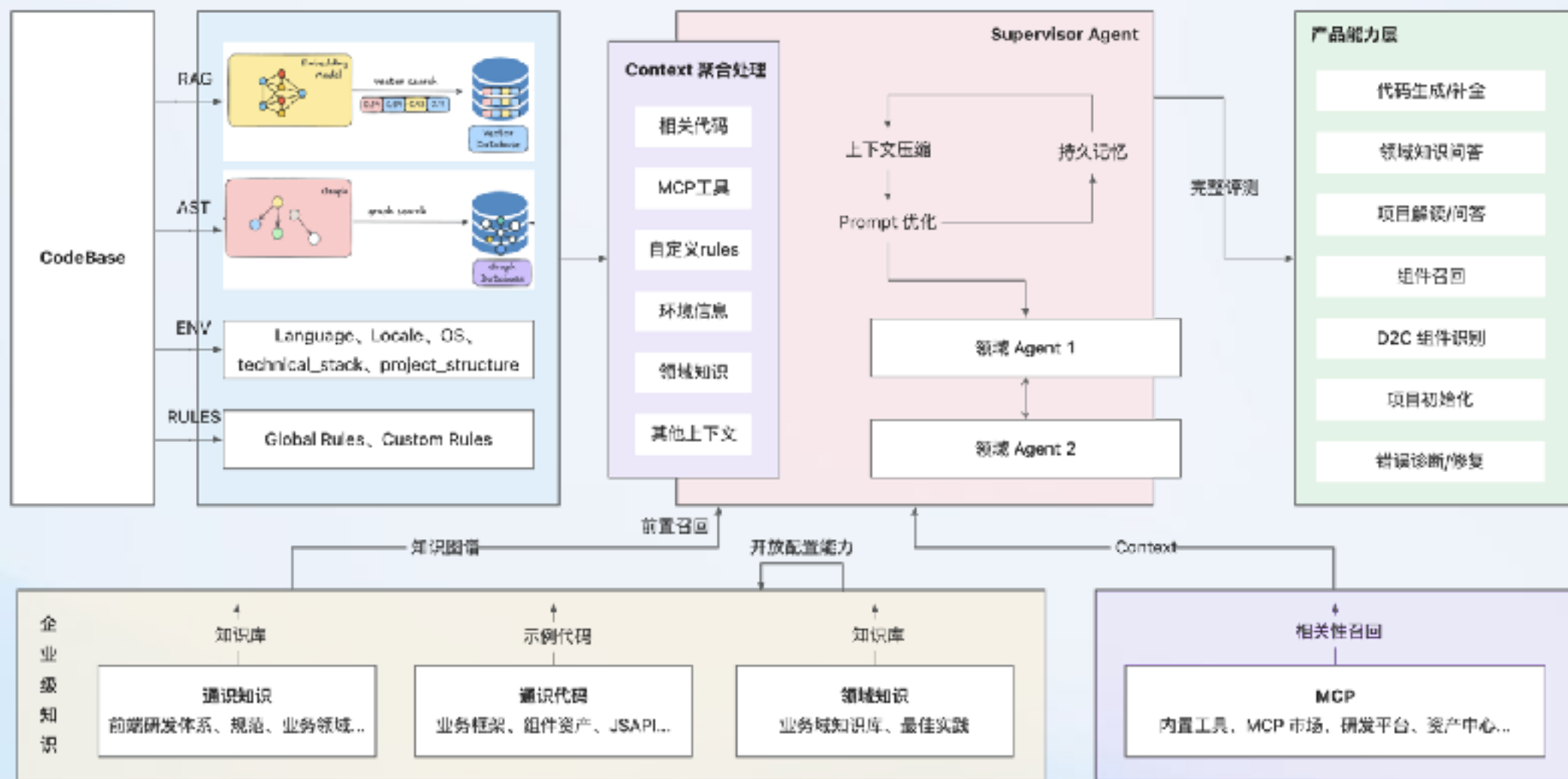
结构化工作流和智能决策的有机结合，实现研发节点的高效流转

02

上下文工程

企业级前端研发的上下文管理

企业级前端研发的上下文工程



企业级的上下文管理能力 – 精准召回

项目结构与环境信息

技术栈知识

前置召回与引用

引用池关注

编辑变更追踪

编辑副作用感知

基于标记的上下文片段清理

批量操作反馈信息聚类



构建具有仓库与技术栈信息的 Agent 编程环境



提供 **mention**(引用机制), 引入文件、文件夹、网页、工作空间、终端、知识库、mcp作为上下文注入

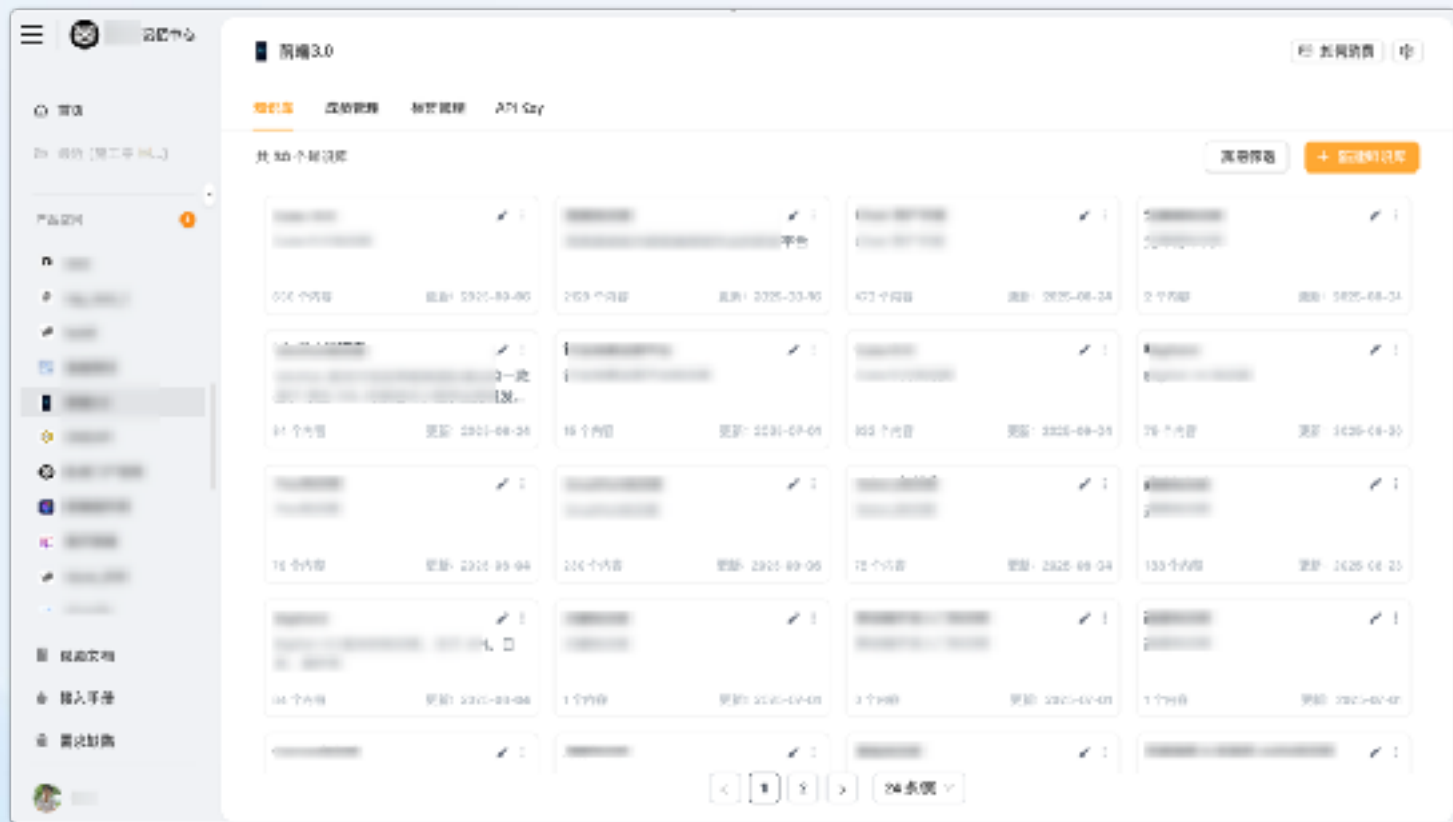


核心围绕编码过程上下文展开, 控制编码信息的**实时性**, 关注点聚焦性、可纠正性



提升上下文**利用率**、减少上下文负担

企业级的上下文管理能力 – 系统管理



Ragas 大模型评测结果

精准度评分 0.64 → 0.84

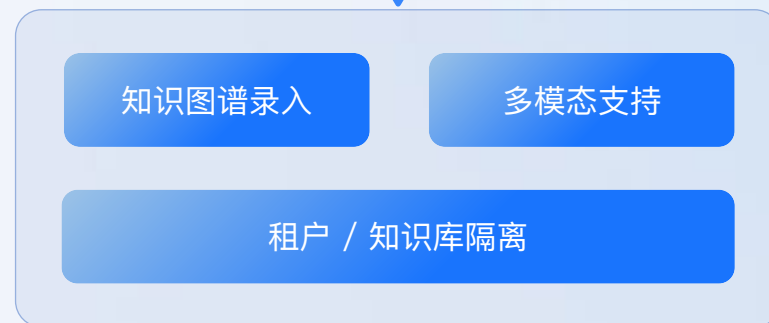
召回率评分 0.55 → 0.67

回答生成评分 0.59 → 0.74

知识录入



知识管理



预处理



组件召回

代码召回

日常答疑

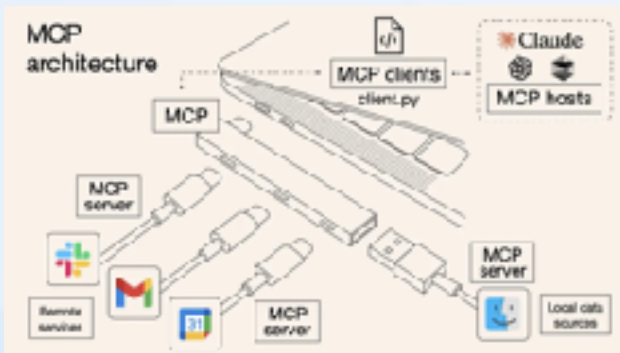
03

智能的工具体系

让工具具备智能

智能的工具体系

MCP 很好!
但在用户侧真的易用吗?



- ▶ 实现 MCP 的开放度高，质量参差不齐
- ▶ 工具描述不清晰，容易冲突

工具质量

- ▶ 人机交互界面不友好，尤其是需要持续补充上下文时
- ▶ 用户显式地指定工具的成本高

交互方式

- ▶ MCP 调用全部交由单个 Agent，调用时机难以控制（尤其工具过多时）
- ▶ MCP 的 tools list 功能单一且不内聚

工具调用

智能的工具体系 – 需要解决的问题

传统工具

确定性的输入和输出

解决单点问题

Agents
as
Tools

智能工具

工具本身具备智能决策能力，更具泛化性

系统性地解决复杂问题，各模块实现分治

LLM 智能地调用工具

智能的工具

智能的工具体系 – 以 UI 自动化为例

用户Query 帮我生成一个订单列表页数据为空的测试用例



准备 Mock 数据

场景组编排

用例编写

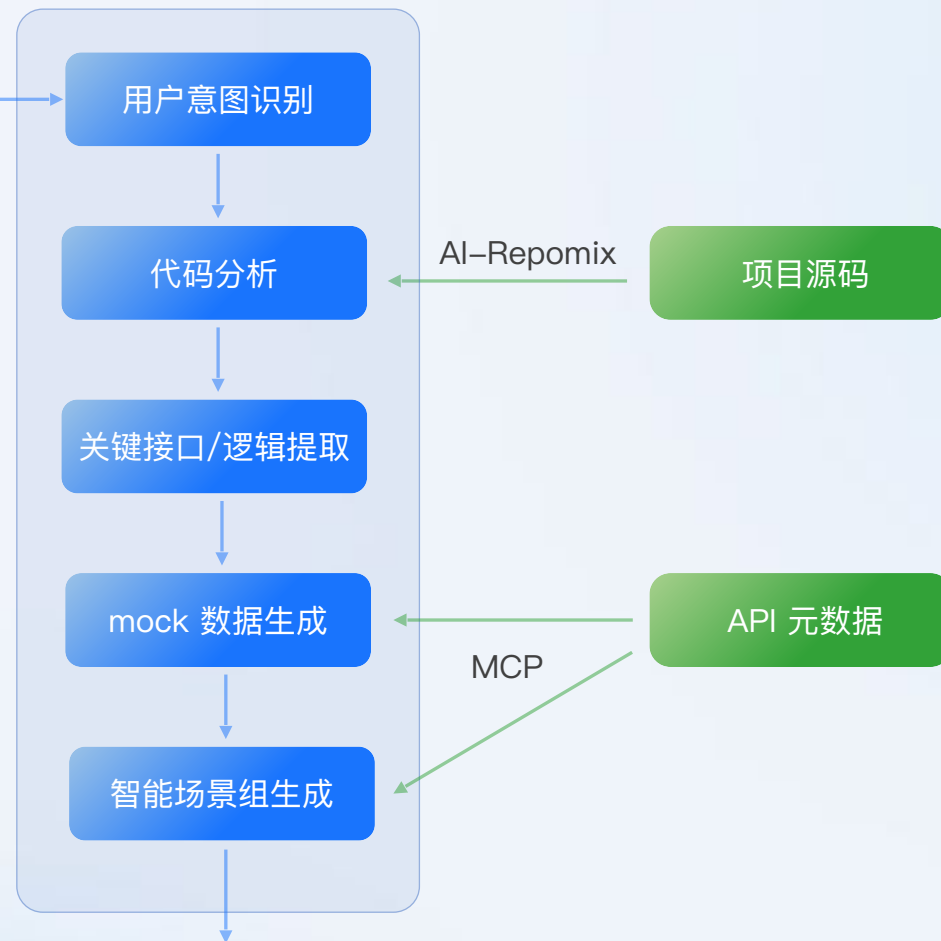
用例执行

智能的工具体系 – 以 UI 自动化为为例



Mock 数据&场景组智能生成

用户 query





智能的工具体系 – 以 UI 自动化为例

等待页面加载完成，热门景区推荐包含上海博物馆，点击展馆展览，包含上海博物馆景点

已获取当前项目页面和场景组信息。

您已选择页面: pages/buy-tickets/index (景区购票页面)

并已创建场景组: 上海博物馆热门景区

根据您的操作描述:

1. 等待页面加载完成
2. 验证热门景区推荐包含“上海博物馆”
3. 点击“展馆展览”
4. 验证页面包含“上海博物馆”景点

我将基于以上步骤生成自动化测试脚本。

```
typescript
import Vision from '@alipay/neo-vision';

Vision.config({
  page: 'pages/buy-tickets/index',
  mockGroup: '上海博物馆热门景区',
});

Vision.run(async ({ agent, delay, ESWIPE_DIRECTION, ESWIPE_PRESET_ACTION }) => {

  await agent.waitUntilIAAssertsTrue('页面不为loading态');

  await agent.aiAssert('热门景区推荐包含上海博物馆');

  await agent.aiTap('展馆展览');

  await delay(2000);
```

用户
query

PE 设计

用户意图识别

代码逻辑分析

场景组生成/选取

用例生成
(基于 NL 描述)

```

Vision.config({
  page: 'pages/xxx/index',
  mockGroup: '订单列表页面数据为空',
})

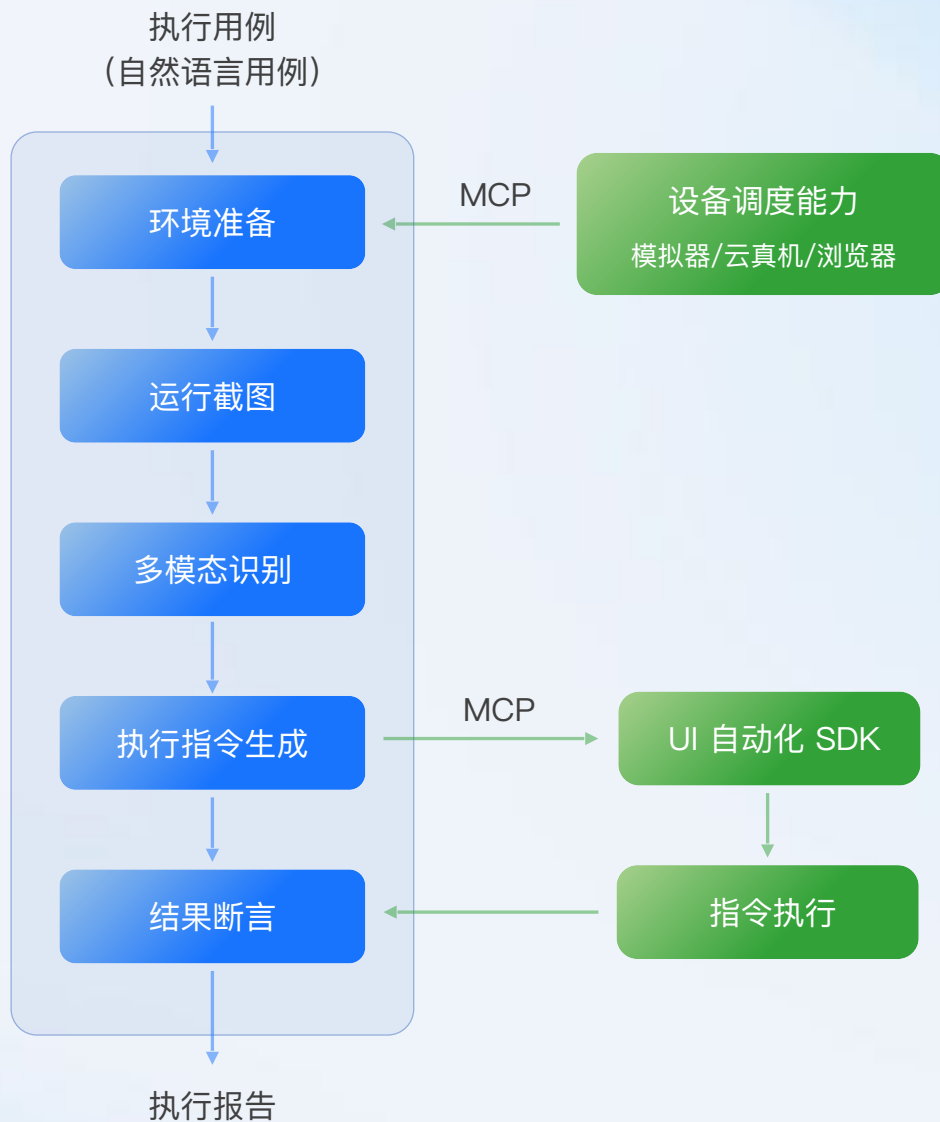
Vision.run(async ({ agent, delay }) => {
  await agent.aiTap('点击订单列表 tab');
  await delay(2000);
  await agent.aiAssert('列表内容为空状态');
});
```

基于自然语言的智能用例生成

智能的工具体系 – 以 UI 自动化为例



支持模拟器/真机/浏览器多环境的调度和用例执行





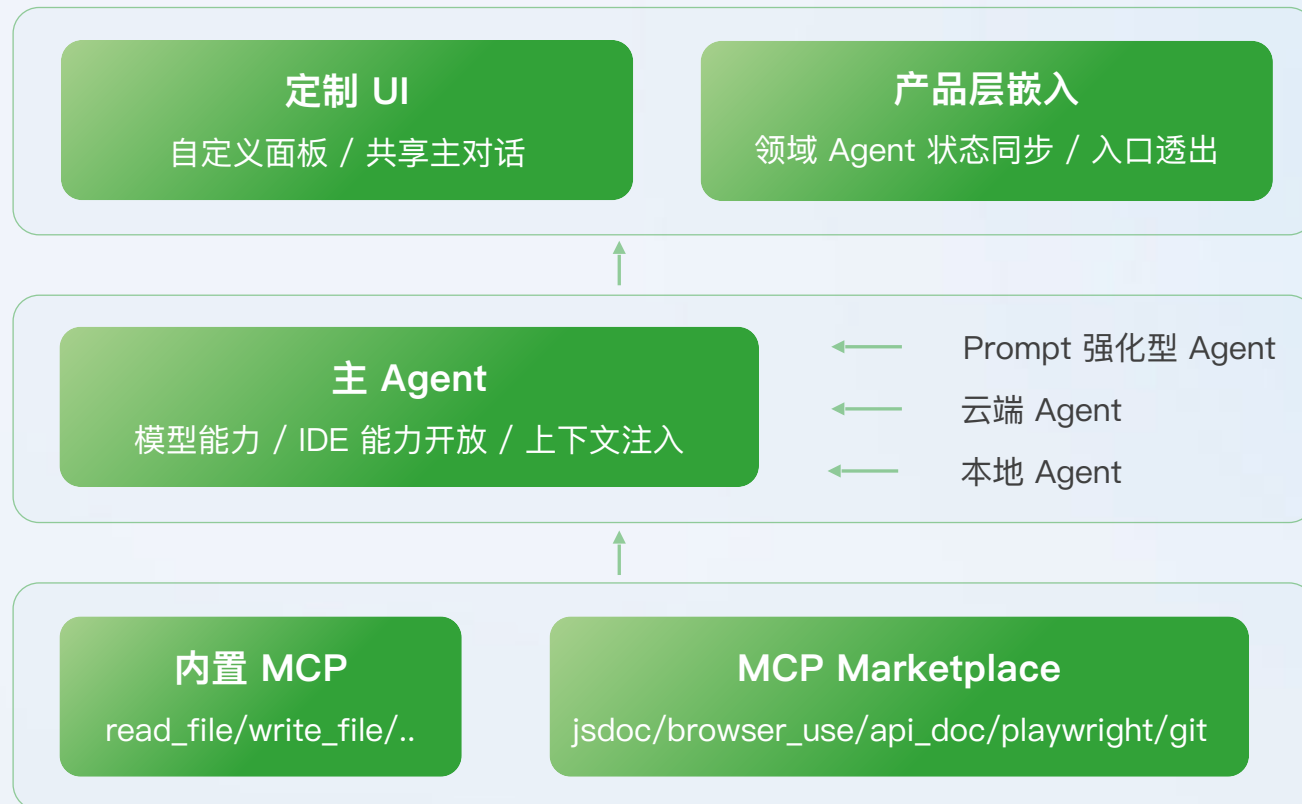
智能的工具体系 – 全研发生命周期覆盖



智能的工具体系 – 开放体系



三层开放能力设计



企业级研发场景的必要设计

04

智能 workflows

基于 Multi-Agent 的 Agentic Workflow 实现

为什么需要智能工作流

用工生态复杂

专家经验难以沉淀

新人多

工具碎片化

流程推进难

项目规范多样化

智能编码

智能调试

智能 Mock

智能审查

智能质检

如何让每一个研发者具备前端专家的 know-how 经验?

智能工作流 – Agentic Workflow

领域经验 + 工具供给 + 执行规则

传统 Workflow

泛化性不足 无法智能决策

大模型 + 自主决策工具

纯 Agent 架构

可控性低 可靠性差

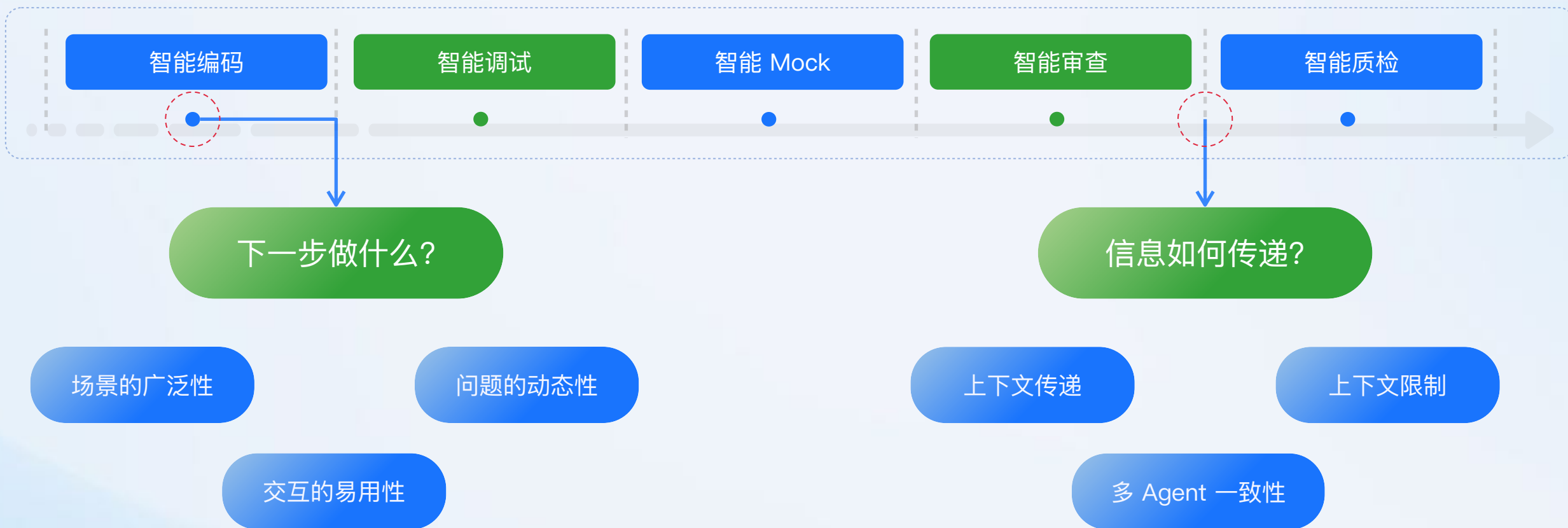
泛化 SOP + 智能决策

Agentic Workflow

专家经验可复用 智能决策 更高可控性

来自 Anthropic 的实践经验 「结构化工作流和自主决策的有机结合能有效提升 Agent 效果」

智能 workflows - 核心问题



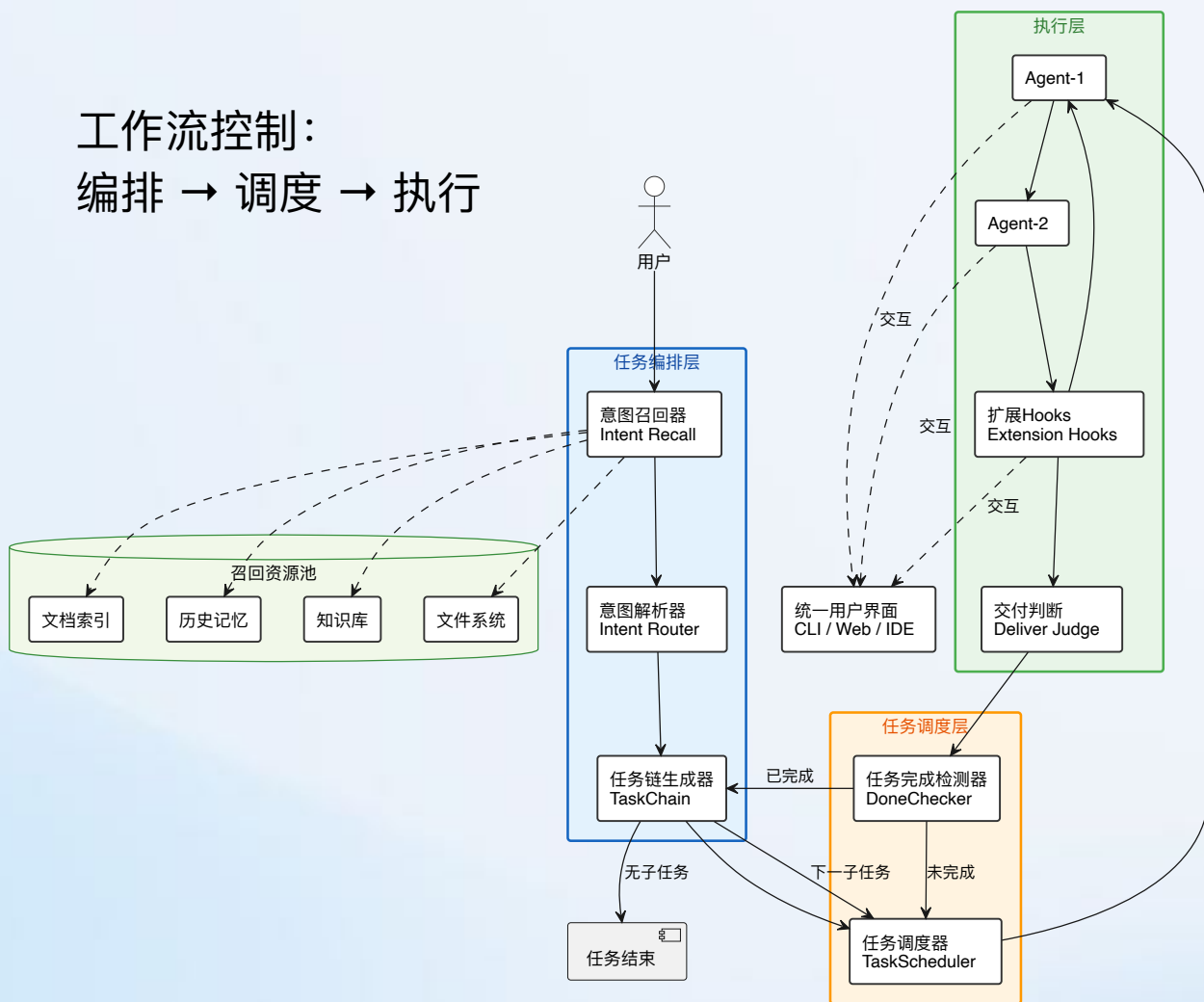
Multi-Agent 视角要解决的核心问题

智能 workflow - workflow 控制



智能工作流 – 工作流控制

工作流控制：
编排 → 调度 → 执行



上下文召回与任务编排

- **有效召回与融合意图**：基于用户输入，进行资源召回与判断，融合上下文后进行任务意图分析
- **意图拆解与可控流程**：完成意图拆解，得出任务完成计划以及对应的Agent执行链起始节点，进行任务链制定，进入具体的调度与执行

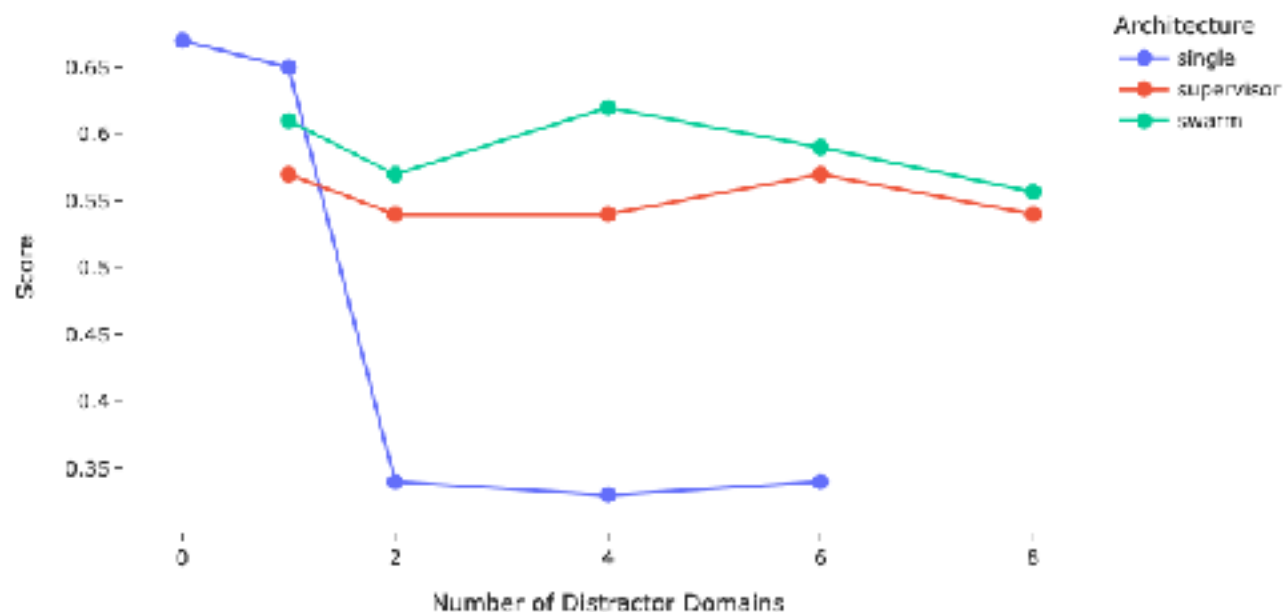
任务调度与执行

- **Agent 交接循环**：基于任务目标与执行过程结果，动态拼接 Agent 执行链，产生执行交接循环
- **可交互性**：过程中可以用户作为节点，要求补充信息、也可以以可交互 UI 界面为节点，让用户主动操作推进，完成后进行阶段性交付与整体任务推进



智能工作流 – Multi-Agent 中的 Context

Multi-agent scaling performance



Context Engineering is Critical

Multi-Agent 系统能有效缓解模型 Context window 大小的问题，但引入了新的上下文管理问题，即

「如何在多 Agent 间实现上下文的有效共享」

From LangGraph: <https://blog.langchain.com/benchmarking-multi-agent-architectures/>

全局上下文

需求输入/需求拆解信息/历史任务交付结果

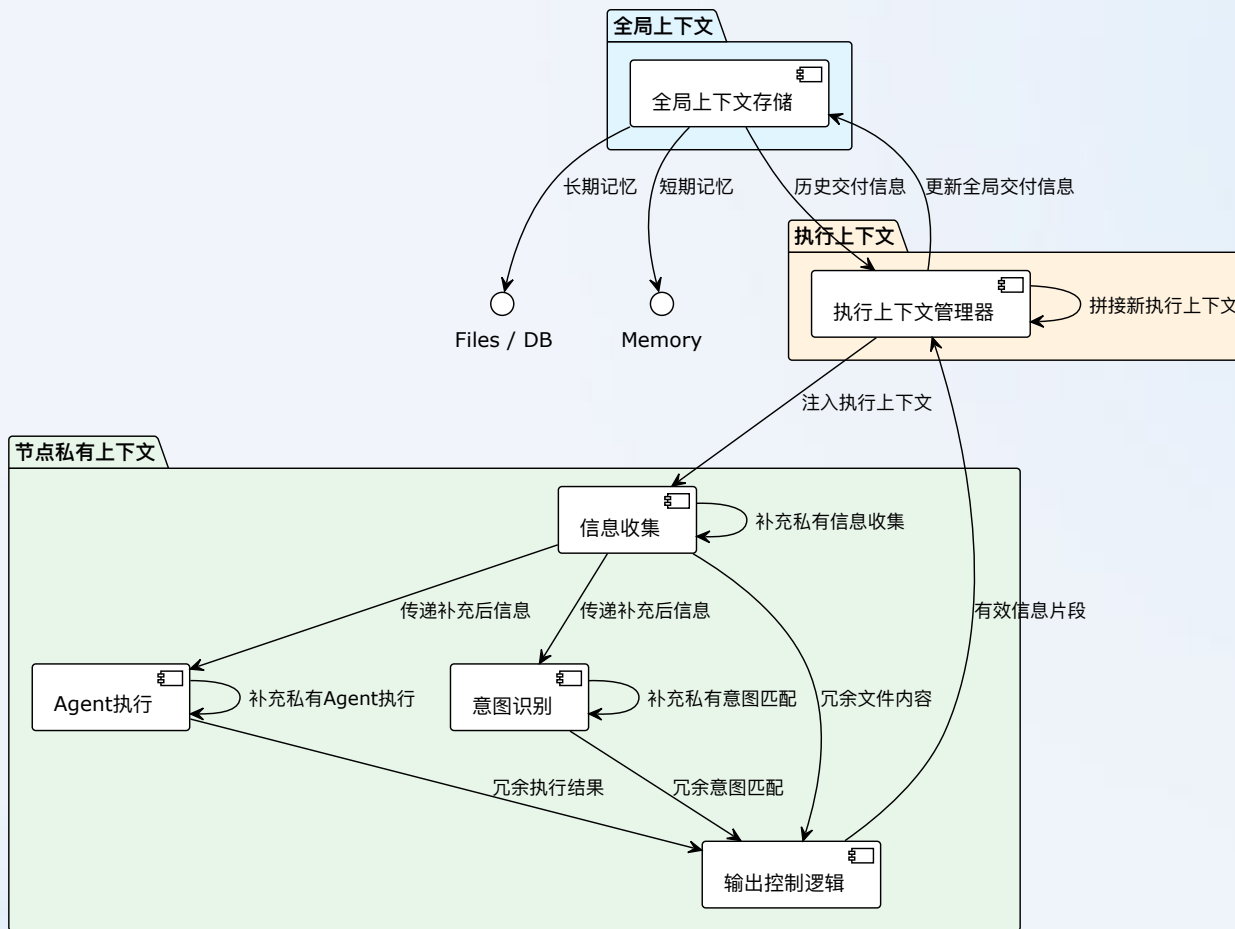
执行上下文

各个 Agent 执行得到的核心上下文信息，
驱动下游 Agent 执行以及交付

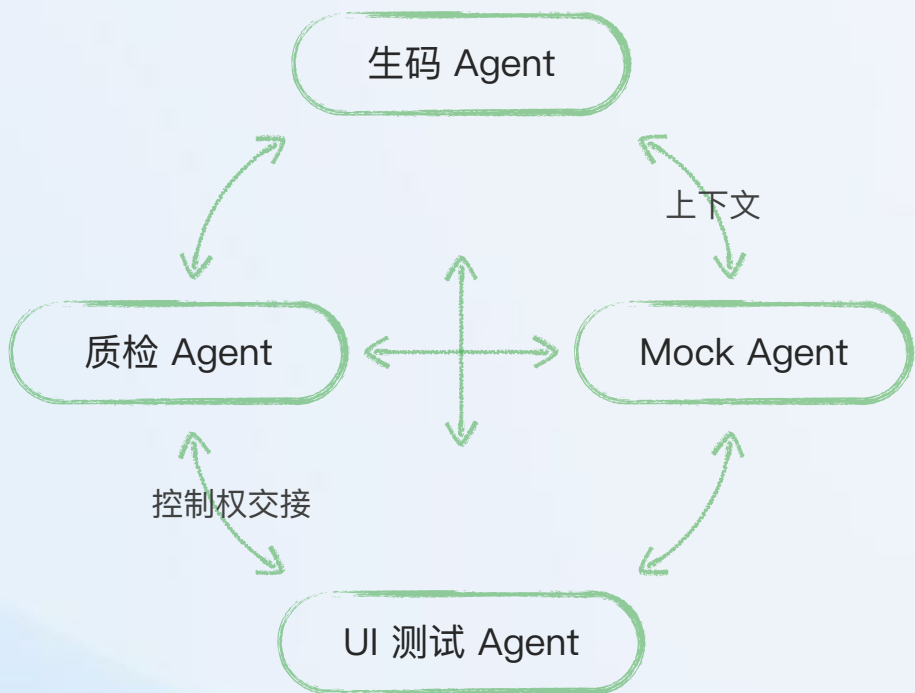
节点上下文

Agent 节点私有上下文，含执行生成过程详细信息

上下文循环：执行上下文 → 节点私有上下文 → 执行上下文



智能 workflows – Multi-Agent 中的 Context



Multi-Agent 的 Swarm 架构

选用 Swarm 架构的优势

- 各 Agent 之间是对等的，可以在 Agent 之间平等地进行控制权交接 (handoff)
- 上下文信息在 Agent 间直接共享，不需要统一通过 supervisor agent 共享
- 单一时间只有一个活跃 Agent，符合研发工作流的实际使用诉求

上下文处理技巧

- **Single Agent 的技巧此时同样有效**：如 足够精简的工具数量（关键信息）、上下文压缩（避免过度消耗上下文注意力）等
- **明确各 Agent 的职责边界**：避免发生角色重叠和冲突，上下文传递信息更准确
- **关键信息的共享记忆**：传递前先进行压缩，保留关键信息；确保任务关键描述注入每个 Agent；尽量复用共享记忆（如使用向量数据库）



智能工作流 – 关键示例



工作流思考 + 规划



下一步预测



子 Agent 执行

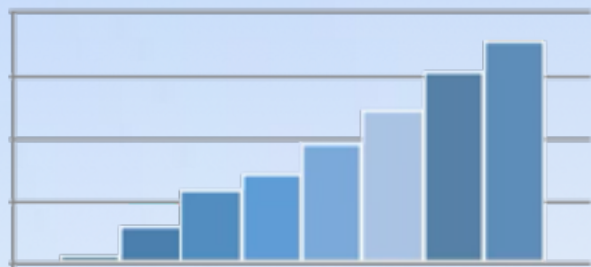
05

总结与未来展望

带来的成效和收益

用户覆盖

覆盖蚂蚁超 40% 前端研发者



场景覆盖

覆盖前端研发代码生成、UI 还原、智能调试、智能质检、智能评审、智能埋点、单测生成等 10+ 研发环节



「前端工程 3.0」 效果展示

产能提升

生码能力覆盖蚂蚁内多种私有技术栈

累计代码采纳 220w+

累计覆盖项目数 3000+

开放能力支持

服务蚂蚁 8+ BG，数十个部门

支持多个团队的定制 Agent、领域知识问答

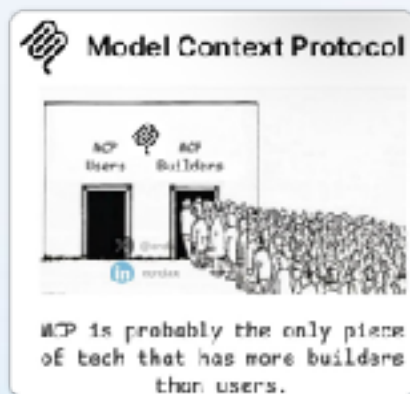
面向社区开源部分基础能力 Neovate Code

关键经验总结

「覆盖完整研发生命周期很重要，但不执着于端到端交付」



「软件工程在 AI 时代成功的机会，来自算法 + 工程 + 产品设计的三位一体」



未来规划与展望



📍 北京

QCon

全球软件开发大会

会议时间：4月10-12日

- 大模型赋能 AI Ops
- 越挫越勇的大前端
- 云上业务架构演进
- 多模态大模型及应用
- 海外 AI 应用创新实践

📍 北京

AiCon

全球人工智能开发与应用大会

会议时间：6月27-28日

- 端侧智能
- AI Agent
- 多模态大模型
- 金融+大模型

📍 上海

QCon

全球软件开发大会

会议时间：10月23-25日

- AI Agent
- 大模型训练推理
- 端侧 AI
- 搜推深度融合
- 数智金融

4月

5月

6月

8月

10月

12月

📍 上海

AiCon

全球人工智能开发与应用大会

会议时间：5月23-24日

- 通用大模型
- AI Agent
- 垂直领域应用
- 模型可解释性

📍 深圳

AiCon

全球人工智能开发与应用大会

会议时间：8月22-23日

- 模型效率与部署
- 多模态大模型
- 大模型安全
- 智能硬件

📍 北京

AiCon

全球人工智能开发与应用大会

会议时间：12月19-20日

- 通用大模型
- 智能硬件
- LM Ops
- 具身智能

极客邦科技 2025 年会议规划

促进软件开发及相关领域知识与创新的传播



参会咨询



查看会议

THANKS

大模型正在重新定义软件

Large Language Model Is Redefining The Software

