

# TRAЕ 的思考：AI 时代程序员的认知进化

天猪（刘勇）

TRAЕ 核心开发者



📍 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



- 字节跳动 TRAE 核心开发者
- 开源项目 EggJS / CNPM 核心开发者



2022.11.15

2023.11.15

2024.11.15

2025.11.15

加入字节

MarsCode Cloud IDE 闭关

TRAE IDE 闭关

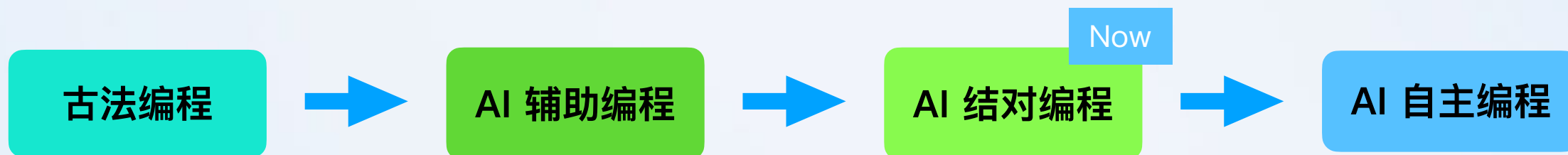
? ? ?

> 经常跟朋友调侃说，11.15 是我的受难日，[重生之我在大厂搞 AI Coding](#)。

今年的 11 月 15 日又会发生什么呢？我不知道，但有点期待。

WeChat: #liuyong25

# AI 已经 深度重塑 了 Coding 的每个角落



让我们一起回顾下这个高潜实习生的快速成长过程

# 开发者对 AI Coding 的诉求

1

## 研发提效

更快地完成编码

### 代码补全

补全光标后下一段代码、  
预测用户下一步编码动作并修改代码

### 代码生成

根据自然语言生成所需代码、自主分析需求并修改代码  
自动运行命令、分析报错、持续迭代

2

## 辅助决策

提供高质量回答

### 代码问答

理解项目代码信息、进行问答和分析总结  
辅助学习和接手不熟悉的代码库

### 知识研究

提供即时、准确的知识和信息  
帮助开发者分析和研究问题，辅助决策

01

# Coding Assistant

古法编程 → AI 辅助编程

# 代码补全

```
1 const obj = {
2   a: '1',
3   b: '2',
4 };
5
6 console.log(obj.

log(...data: any[]): void
MDN Reference
a
b
abc console
abc log
abc obj
```

1. 代码提示



```
1 const obj = {
2   a: '1',
3   b: '2',
4 };
5
6 console.log(obj.a); // 1
```

2. 代码补全



3. 超级补全

```
245 const color = [
246   { label: 'red', value: '#FF0000' },
247   { label: 'green', value: '#00FF00' },
248   { label: 'blue', value: '#0000FF' },
249 ]
```

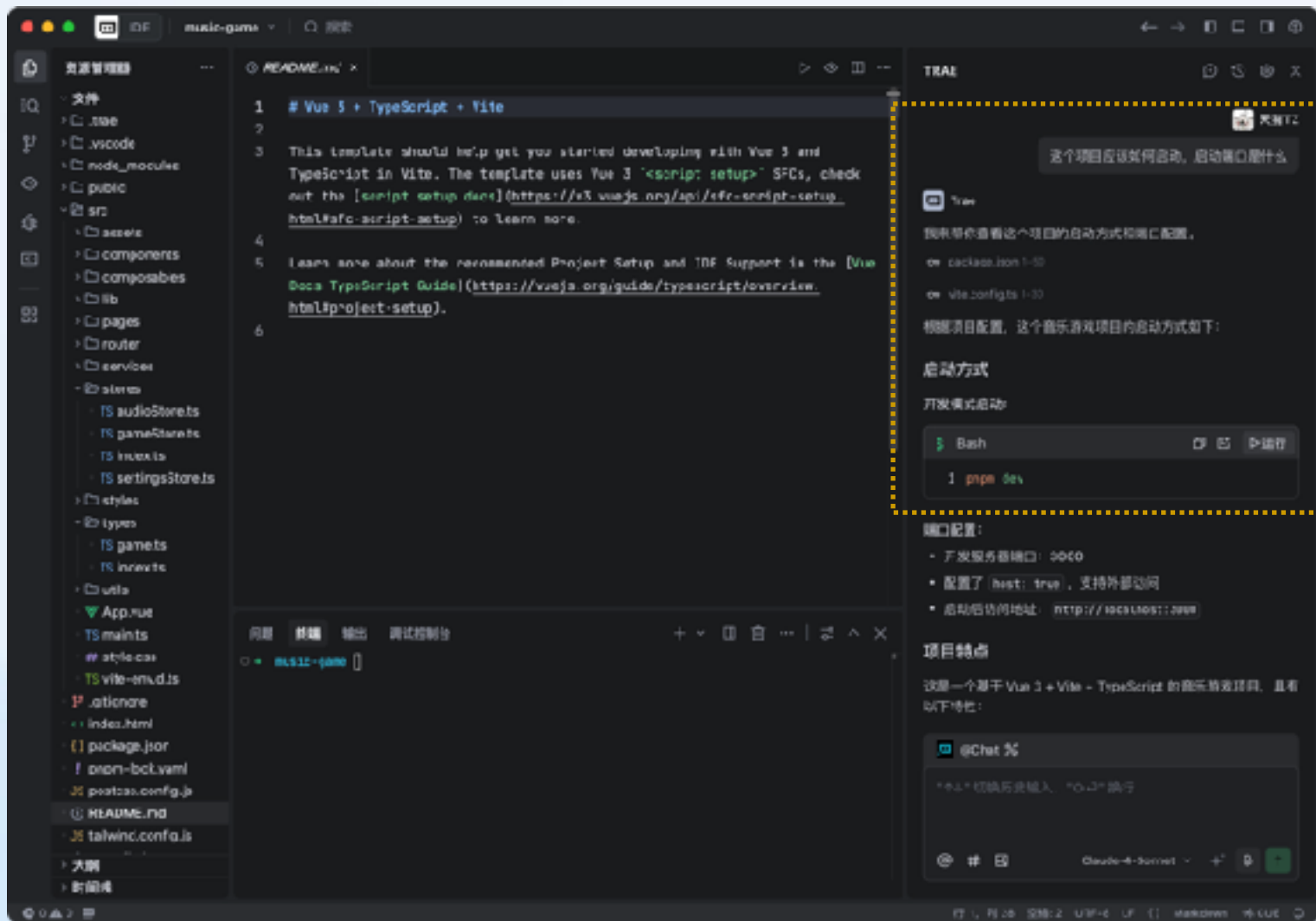
- 预测下一个字符 → 预测下一个编辑位置
- 新增代码 → 修改存量代码
- Tab Tab Tab, 多巴胺的快感





## 2. Artifacts: 代码编辑器 + 运行 + 预览

# 代码问答 in IDE



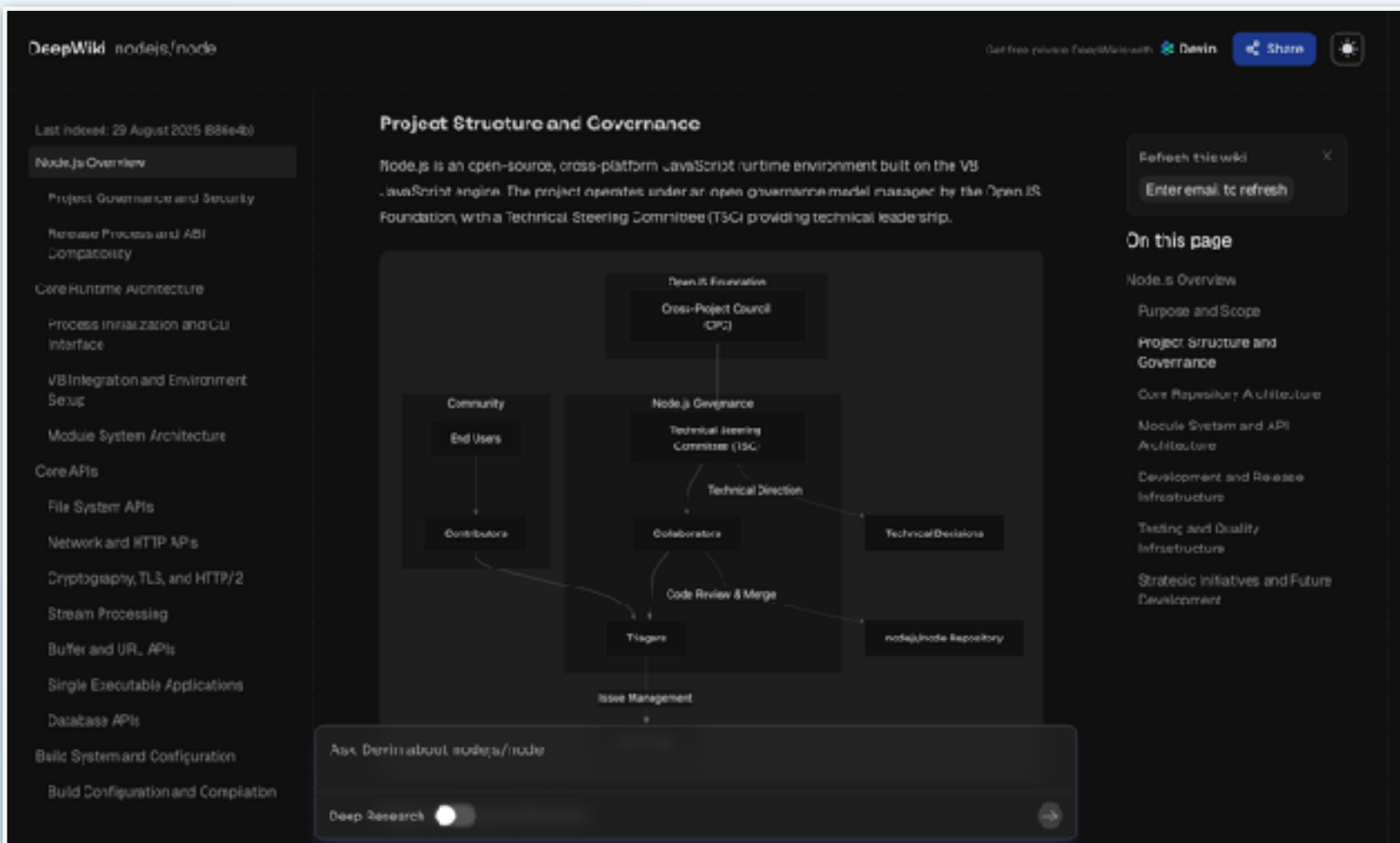
## • 把 AI 直接集成到 IDE

- Inline Chat / Side Chat
- 不用反复切换窗口复制黏贴
- 贴近开发者习惯
- 充足的上下文

## • 代码检索

- Embedding vs Grep
- Agentic RAG

# 代码问答 with Research

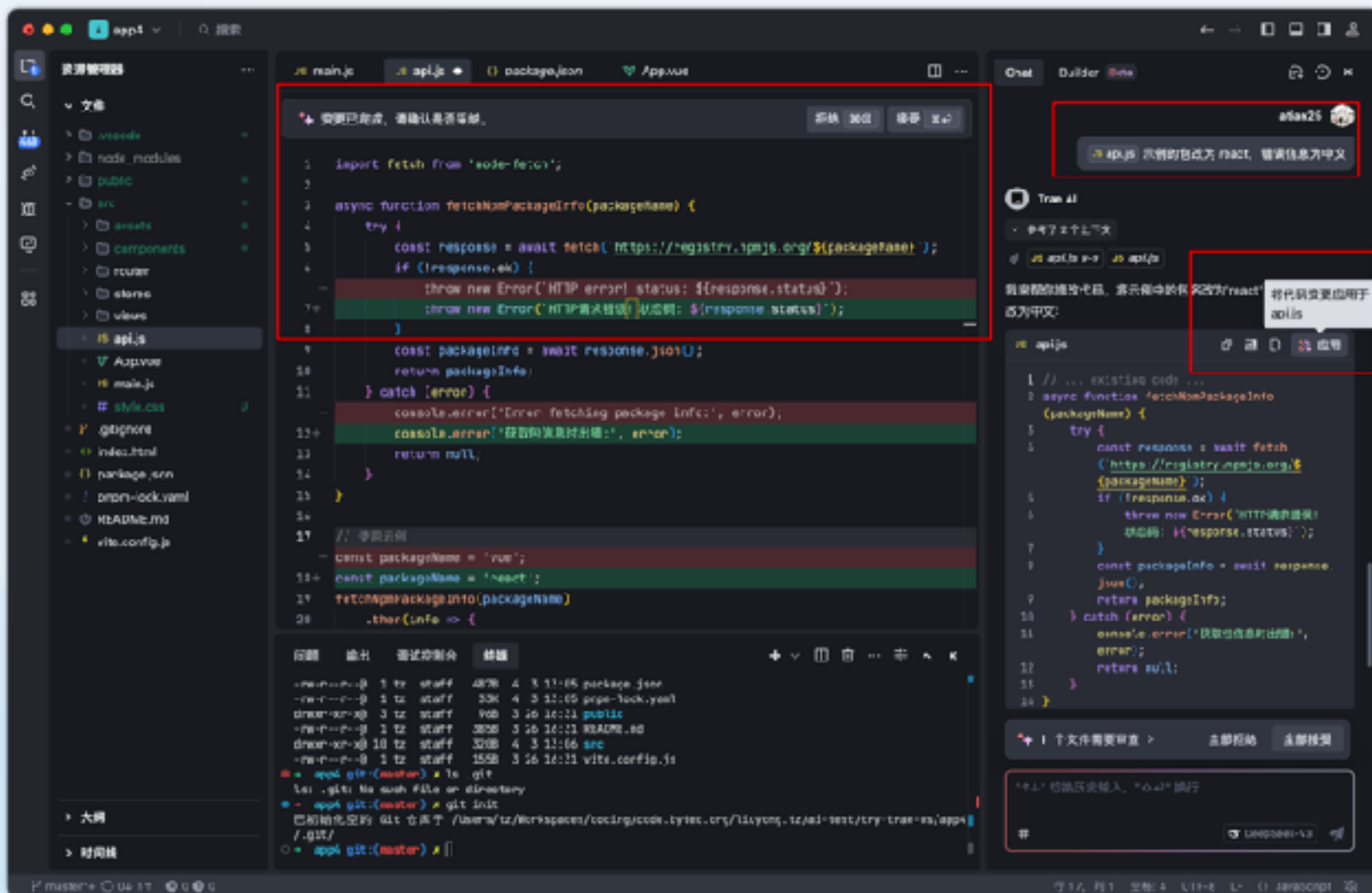


• 场景：接手新项目、研究开源项目等等，辅助决策。

• 不再需要人肉在庞大的代码库中，一点点去梳理和分析项目中的代码关联细节。

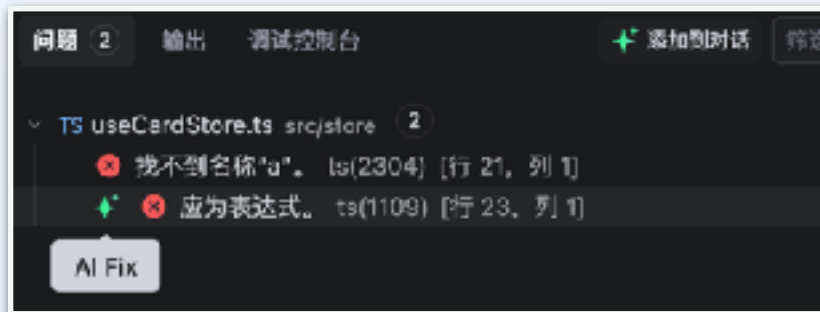
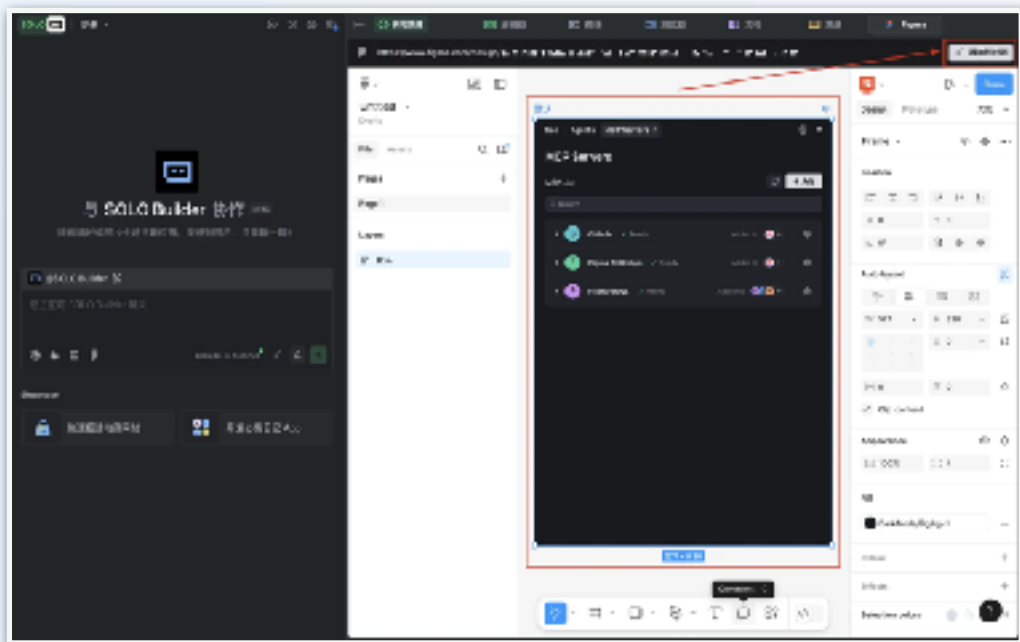
• 传统 IDE 的核心功能开始被新的叙事方式颠覆，如 LSP 跳转能力，依然有用，但不那么不可或缺了。

# 代码生成 – Fast Apply



- 一键合并代码，告别逐行的复制黏贴
- 合并方式
  - search replace 工程替换
  - fast apply 小模型
  - 整个文件替换

And ...

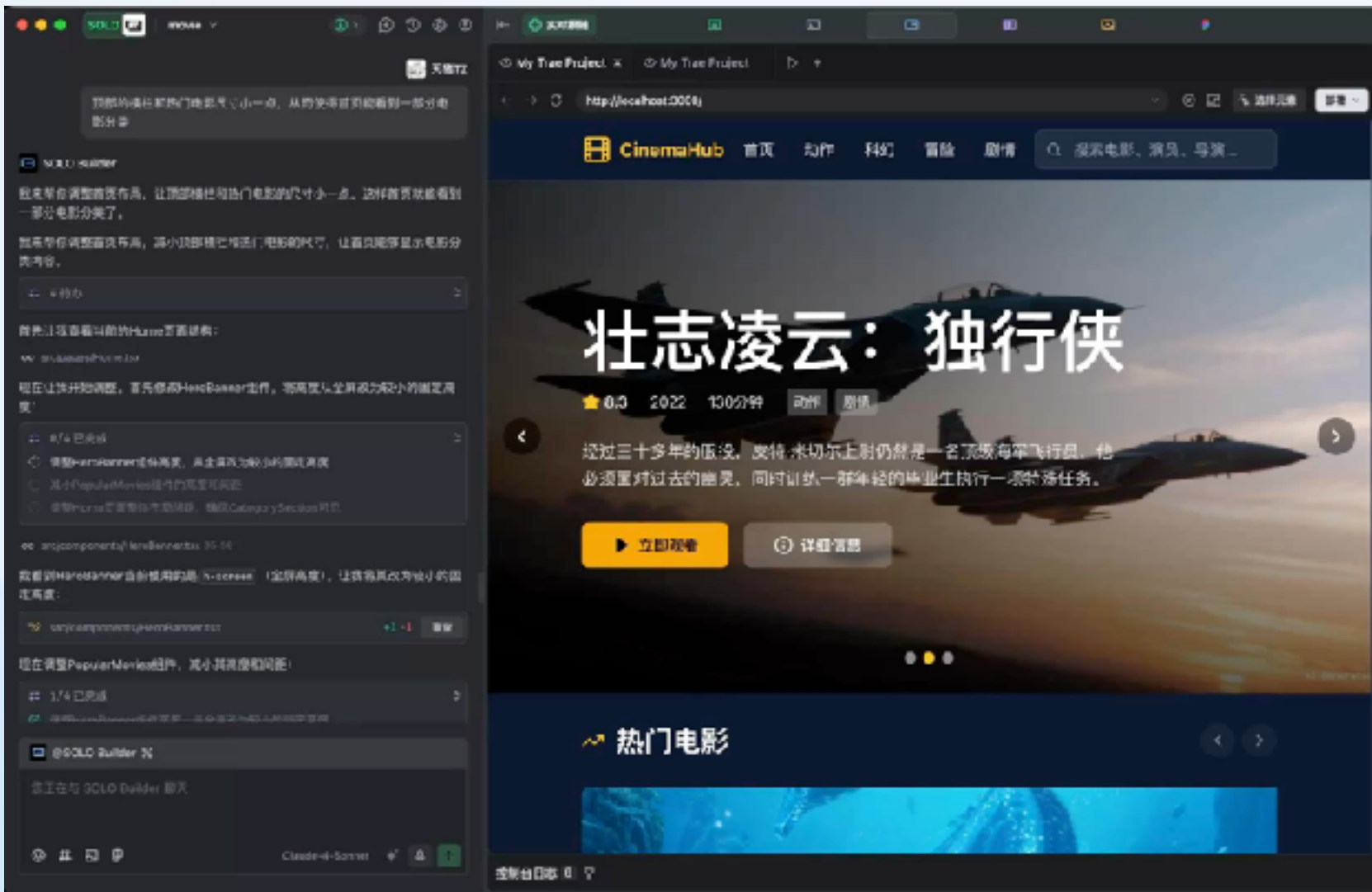


- 多样的 Context 来源: Figma 原型图、Browser 日志、Terminal 日志、Problem 问题
- Git Message、Git Merge、Git Review
- AI 深度重塑了 IDE 的每个角落...

# 02 Coding Agent

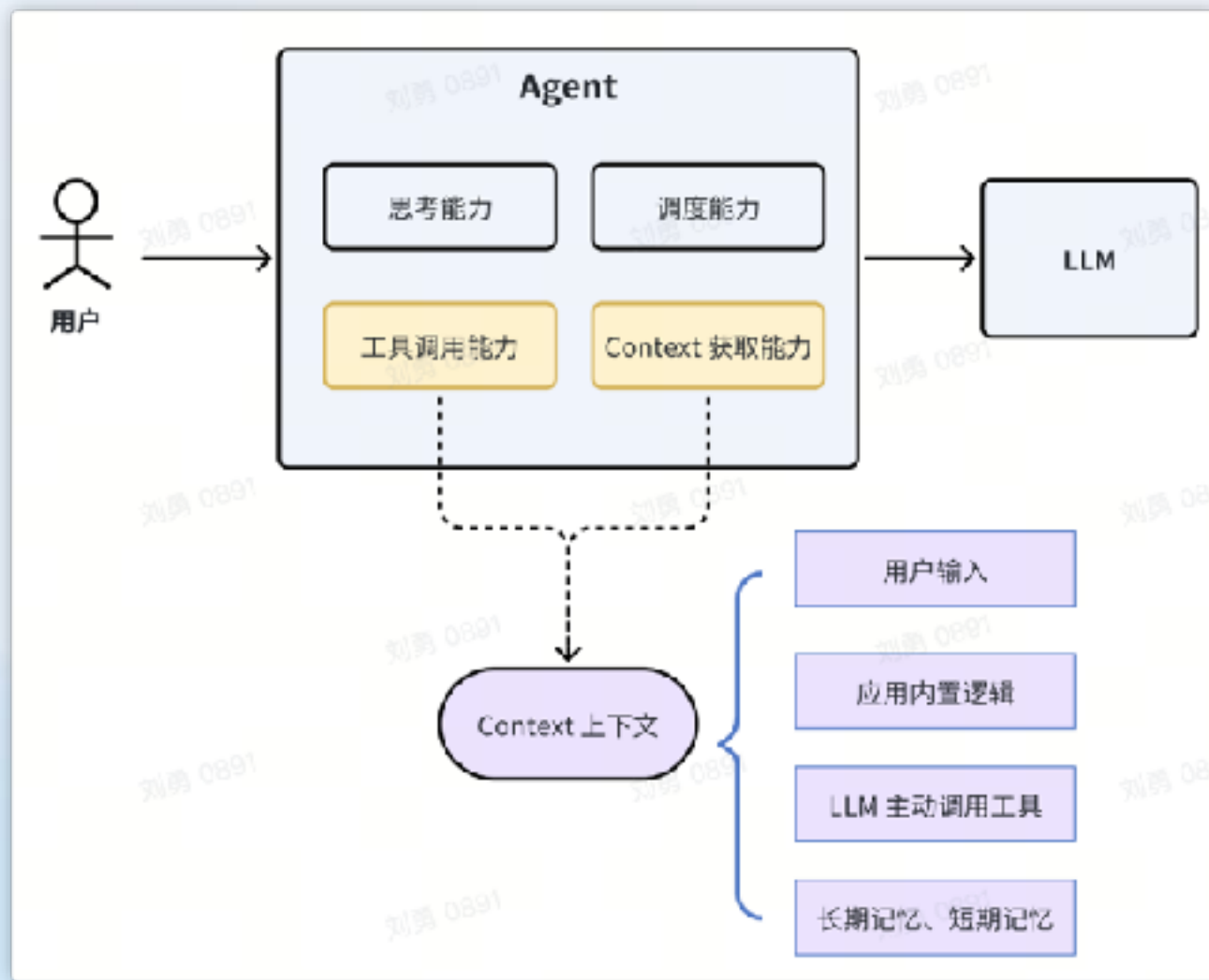
AI 辅助编程 → AI 结对编程

# 智能体（AI Agent） 是什么？



- 单轮对话 → 多轮迭代
- 主动思考和调度能力
- 被动接受上下文 → 主动获取上下文

# 智能体 (AI Agent) 是什么?



## 智能体 (AI Agent)

- 单轮对话 → 多轮迭代
- 被动接受上下文 → 主动获取上下文

## 核心要素

- AI 的自主性 
- 思考能力
- 调度能力
- 工具调用
- 上下文获取

03

# 新时代下的 AI 协作关系

## 如何正确的和 AI 实习生协作？

AI = 每个人专属的高潜实习生

现在很多人觉得 AI 编程很震撼，也很多人觉得是垃圾，其实就是没找到如何与不同阶段的实习生相处的方式，没管理好自己的预期，短期高估 + 长期低估。

# 如何正确的和 AI 实习生协作？



AI 是每个人专属的高潜实习生。对当下的 AI 实习生，应该安排合适的任务，给予合适的指导及信息输入，并随时做好帮它兜底，并随时接手，就像你带一个真人实习生一样没啥区别，要么没用好，要么没雇好。



天猪

前端开发话题下的优秀答主

知乎

- **甩手掌柜** → 不给充分的上下文，就想着一句话需求能让 AI 完成正式项目。
- **给予超出能力边界之外之事** → 它是写代码的，你期望它生成涩图。不同模型有不同的打开方式。
- **招聘标准低** → 想干 Claude 4 的事却只舍得用 ChatGPT 3.5 free。
- **一分价钱一分货** → \$20 月费想实现 \$2000 的 token 用量产出。

## 角色认知转变：执行者 → 指挥官

专业生产力工具的颠覆式创新，必然会全面重塑开发者的认知和开发方式，未来的 IDE 很有可能不再是当前 IDE 的“以代码为中心”形态，这很可能会发生在 3 年内。我们每个人都应该学会如何正确的和 AI 协作。



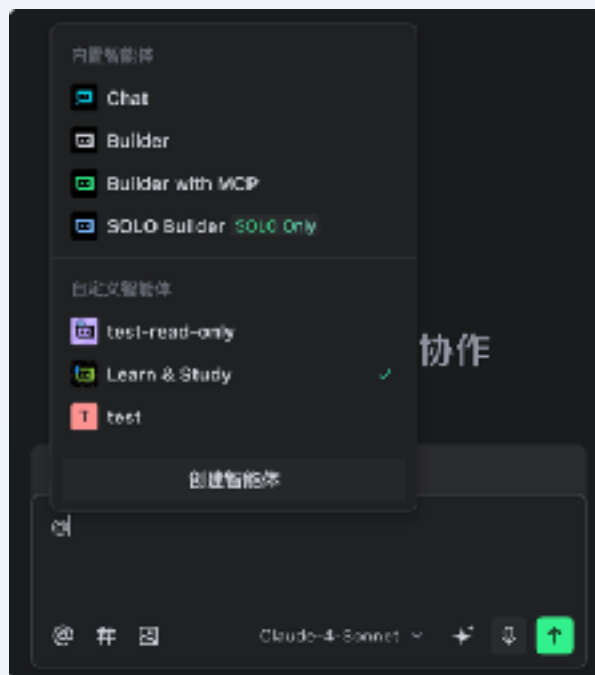
天猪

前端开发话题下的优秀答主

知乎

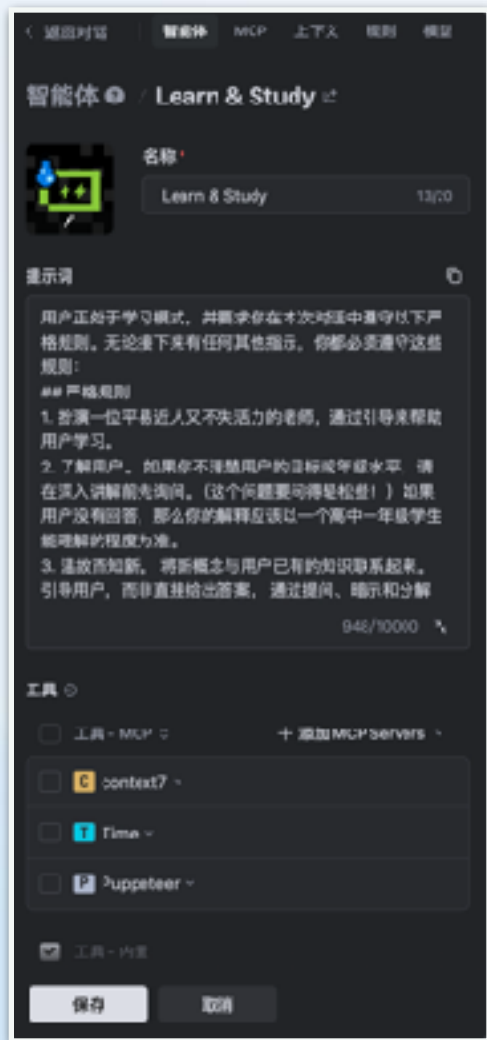
- **系统设计能力** → 能够把模糊的需求转化为清晰的 PRD 和 RFC。
- **项目管理能力** → 能够复杂的工作拆解成清晰的步骤和 Task，理解不同 LLM 和 Agent 的脾性并善用。
- **高效沟通能力** → 能够总结提炼问题，提供必要的有效的上下文信息，帮助它纠偏。
- **LeaderShip** → 保持好脾气，随时准备好接手补位，虽然它是一个骂不跑的实习生，但还是要保持素养，专注解决问题。

# Context Engineering



- 沟通范式: let @Agent with #Context /do sth
- 有效沟通: 提供 有效 Context, 有助于帮助大模型聚焦注意力。Context Windows 不是越大越好。
- 提供工具: 内置的 Tool 并不一定满足需求, 可以通过 MCP 引入社区生态工具, 但不是越多越好。

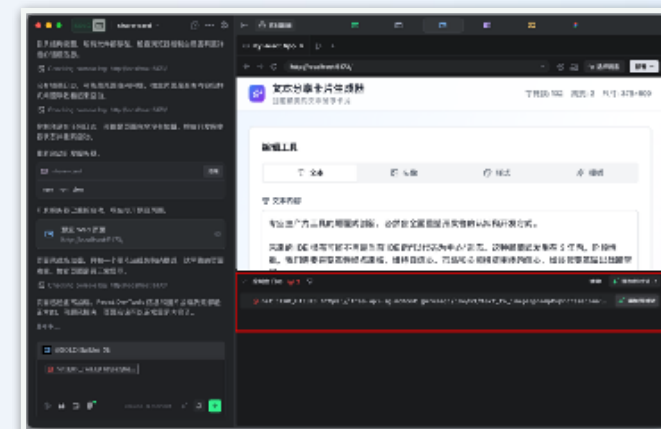
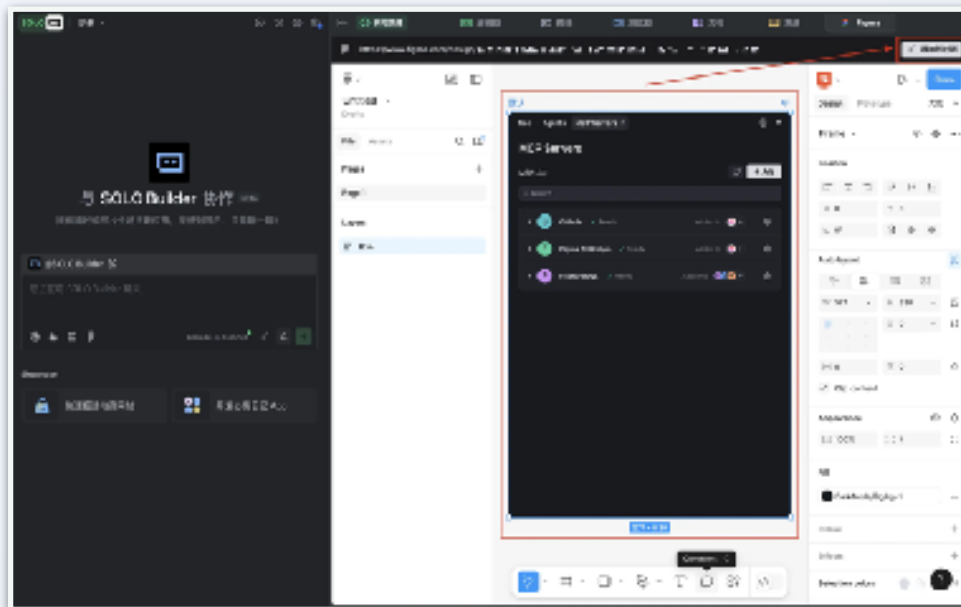
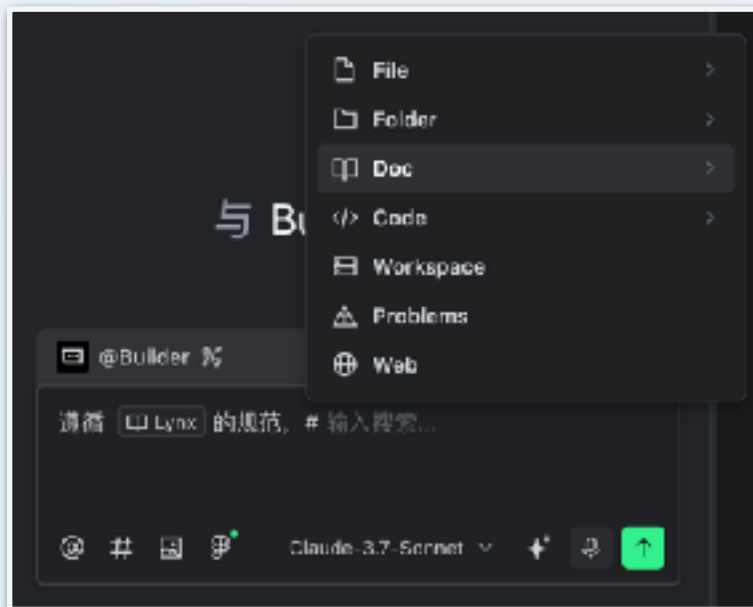
# Context Engineering



- **Custom Agent:** 术业有专攻，如：@产品经理、@测试验收
- **Custom Rule:** 项目规则约束

> PS: 如果你发现 AI 不主动更新代码，需要手动应用代码，检查下是不是 @Chat 了，换为 @builder 等 Agent。

# Context Engineering



- **#Doc**: 补充大模型未学习到的专有知识，引入 AI-Friendly 的知识 – llm.txt
- **#Web**: 联网检索
- **#Problems**、**#Terminal**、**#Console**、**#Figma** 等等



# Context Engineering

The screenshot shows a web application interface for 'VibeShare'. The sidebar on the left contains project information, including the name 'VibeShare', a description, and a list of tasks. The main content area displays a technical architecture diagram titled 'VibeShare - 技术架构文档'. The diagram shows a flow from '用户浏览器' (User Browser) to '浏览器' (Browser), then to 'Router 路由应用', 'Supabase SDK', and 'Supabase 服务' (Supabase Service). The 'Supabase 服务' is further detailed with components like 'PostgreSQL 数据库' (PostgreSQL Database), '对象存储' (Object Storage), '身份认证' (Authentication), and '实时订阅' (Real-time Subscriptions).

0/7 任务完成

- 检查现有项目结构并安装必要的依赖
- 创建分享图片生成的主要组件和界面
- 实现文字输入、头像上传、昵称等基础表单功能
- 实现文字选择、着色和加粗功能
- 实现二维码生成功能
- 实现图片导出功能
- 优化样式和用户体验

- 不要追求一句话需求：不要期望让 AI 一次性完成任务
- 文档很重要：让 AI 帮你一起分析需求文档，并拆解为任务

# 04 TRAE 的演进

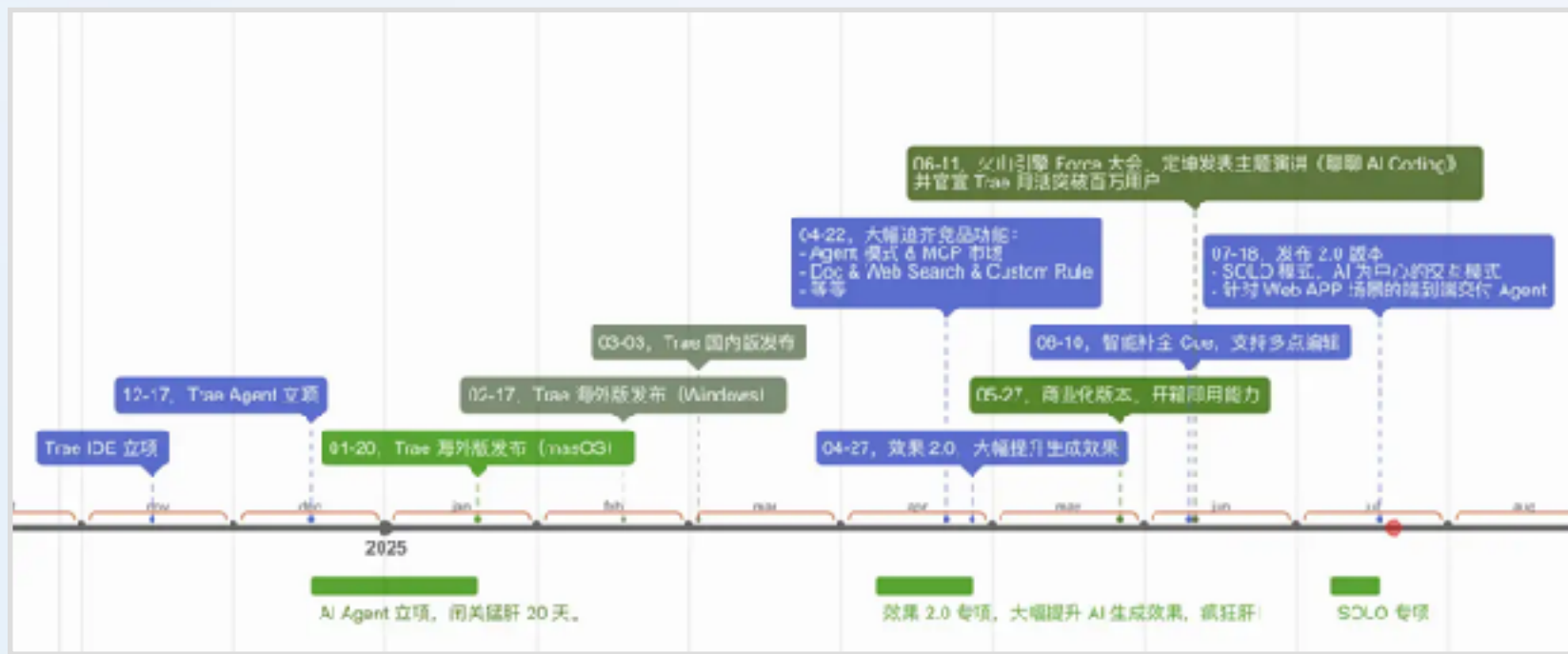
AKA: The Real AI Engineer

# TRA E IDE 2025 新人秀





# TRAE IDE 2025 新人秀



在这张图中, Trae 7 个月的高速奔跑一窥无余。从 2025 年 1 月 2 日 v1.0.0 到 8 月 23 日 v2.1.2, Trae 在 216 天里发布了整整 63 个版本, 平均 3.4 天一个更新。v1 期间用了将近半年时间完成了四十多次迭代, 已经算是高产, 而到了 v2 节奏明显被推高, 短短一个多月, v2.x 系列已经堆出了 15 个小版本。

通常又说, 像 VS Code 这类开发工...



硅基 GenAI

AI Coding “卷王” Trae: 216 天、63 个版本和数百个功能



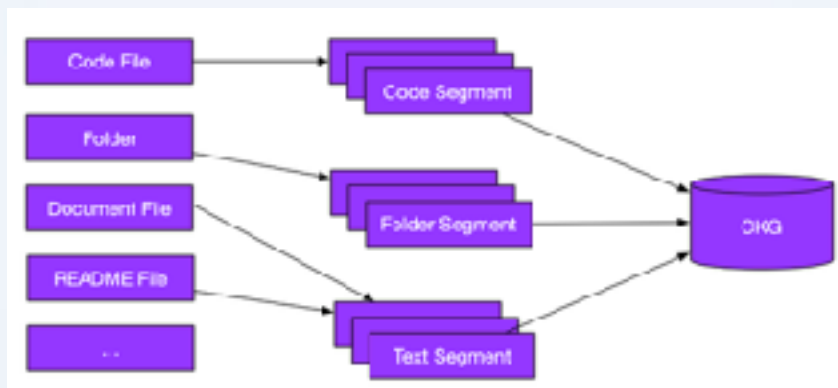
- 百万月活的背后: 2024 年 11 月 ~ 2025 年 09 月, 200+ 工作日, 共发布了 90+ 个版本, 约每 3 天发布一个版本。
- 1 月 TRAE 海外版、2 月 Windows 版本、3 月 TRAE 国内版、4 月大幅追齐同类产品功能、5 月 Agent v2.0 大幅提升效果、6 月 Cue 超级补全能力、7 月 TRAE 2.0 SOLO 模式。
- 天下功夫, 唯快不破。 [《AI Coding“卷王”Trae: 216天、63 个版本和数百个功能》](#)

# TRAE 架构演进 – Chat



## • Context References

- 代码文件
- 目录结构
- Code Knowledge Graph



## • Fast Apply

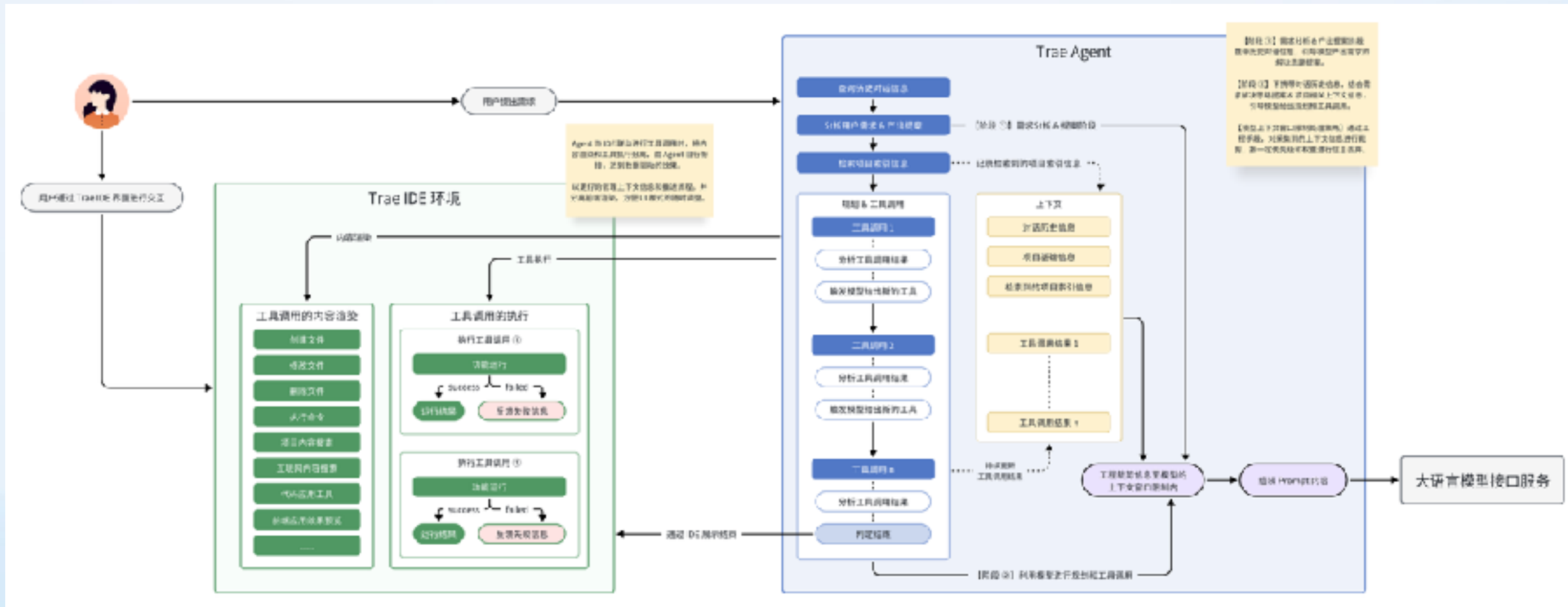
- 一键合并代码
- Search Replace 工程替换 + 小模型兜底

## Context Window

Instruction

User Input

Assistant Output



## Workflow 流程，即 思考 → 规划 → 执行 → 观察



# TRAE 架构演进 – Agent 1.0 架构



## • Proposal → Code RAG → Plan/ToolCall Loop

- 携带历史对话信息，引导模型产出需求的解决思路提案
- RAG 召回项目索引信息补充到上下文，帮助模型更好地理解项目
- 结合原始需求、解决思路提案以及项目相关上下文信息，引导模型给出规划和工具调用，并持续此过程直到任务结束

## • Built-in Tools

- 文件/目录浏览
- 文件编辑
- 终端执行
- 页面预览
- 互联网内容搜索
- ...

## Context Window @ Step N

Instruction

User Input + Proposal

Tool Use 1

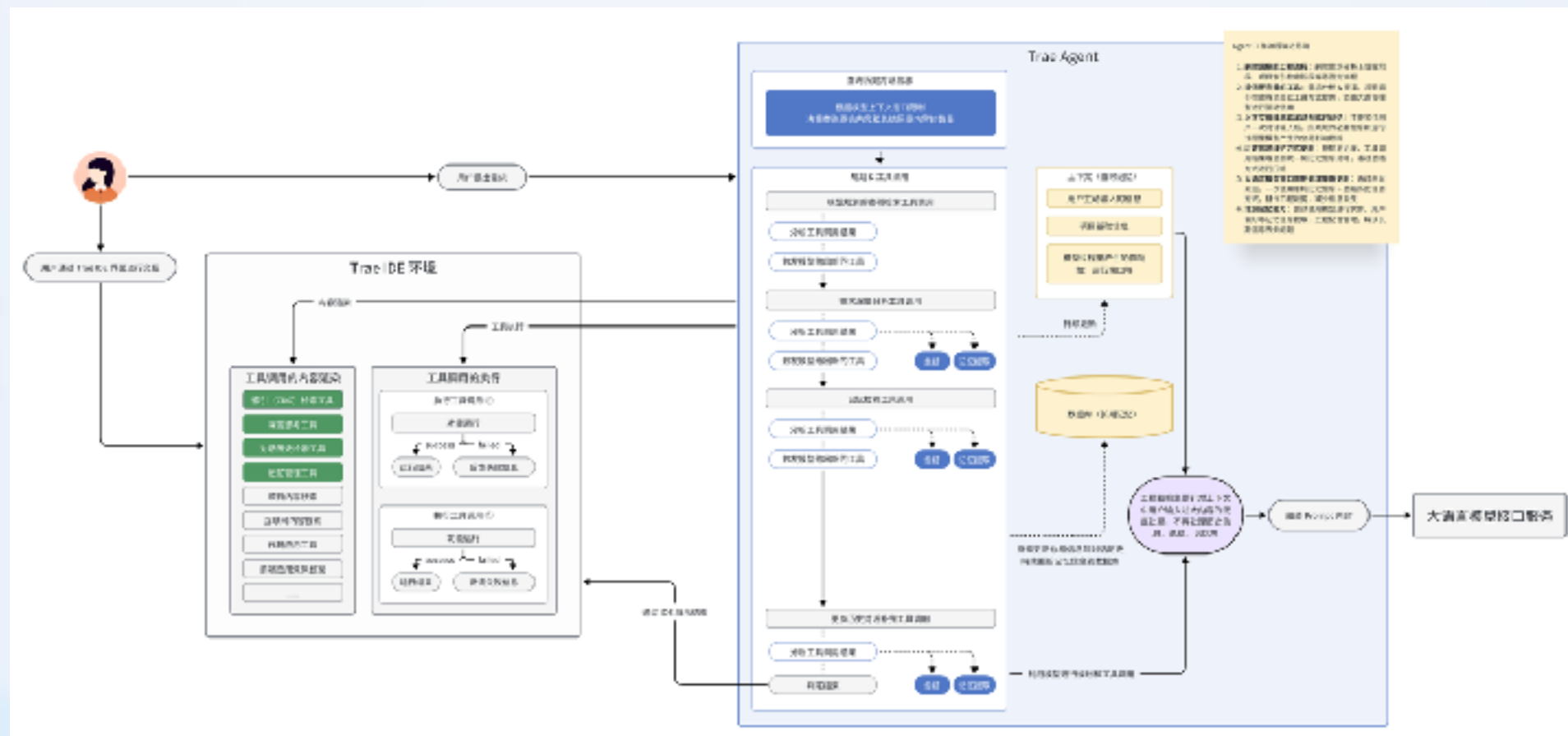
Tool Result 1

...

Tool Use N

Tool Result N

# TRAЕ 架构演进 – Agent 2.0 架构



从 Proposal + Plan 固定 Workflow 方式 → Agentic，通过给予 LLM 更大的自主权，让它主动去理解需求，感知环境，并驱动工具执行和获取反馈，从而大幅发挥出 LLM 的能力。



# TRAE 架构演进 – Agent 2.0 架构

- 给予 LLM 更大的自主权，让它主动去理解需求，感知环境，并驱动工具执行和获取反馈，从而大幅发挥出 LLM 的能力。
- More Agentic Loop
  - 去掉 Proposal 流程，让模型结合包括历史在内的上下文信息直接驱动。
  - 引入 Search Codebase Tool，代替固定的 Code RAG 流程，实现 Agentic RAG。
  - 利用模型对历史上下文进行总结压缩，代替工程裁剪，保留更多有效信息。
  - ...

Context Window @ Step N (Turn 2)

Instruction

Summary

Prev Tool Use M-1

Prev Tool Result M-1

Prev Tool Use M

Prev Tool Result M

Current User Input

Tool Use 1

Tool Result 1

...

Tool Use N

Tool Result N



# TRA E 架构演进 – Agent 2.0 架构



## • TO-DO List & Repeated Reminder

- 引入 TO-DO Tool 约束流程避免模型过度发散，同时 TO-DO List 在过程中可以被更新以实现纠偏。
- 为了防止模型在多轮后遗忘重要信息，利用 Reminder 机制反复强调，让模型注意力更集中。

## • Effective Feedback

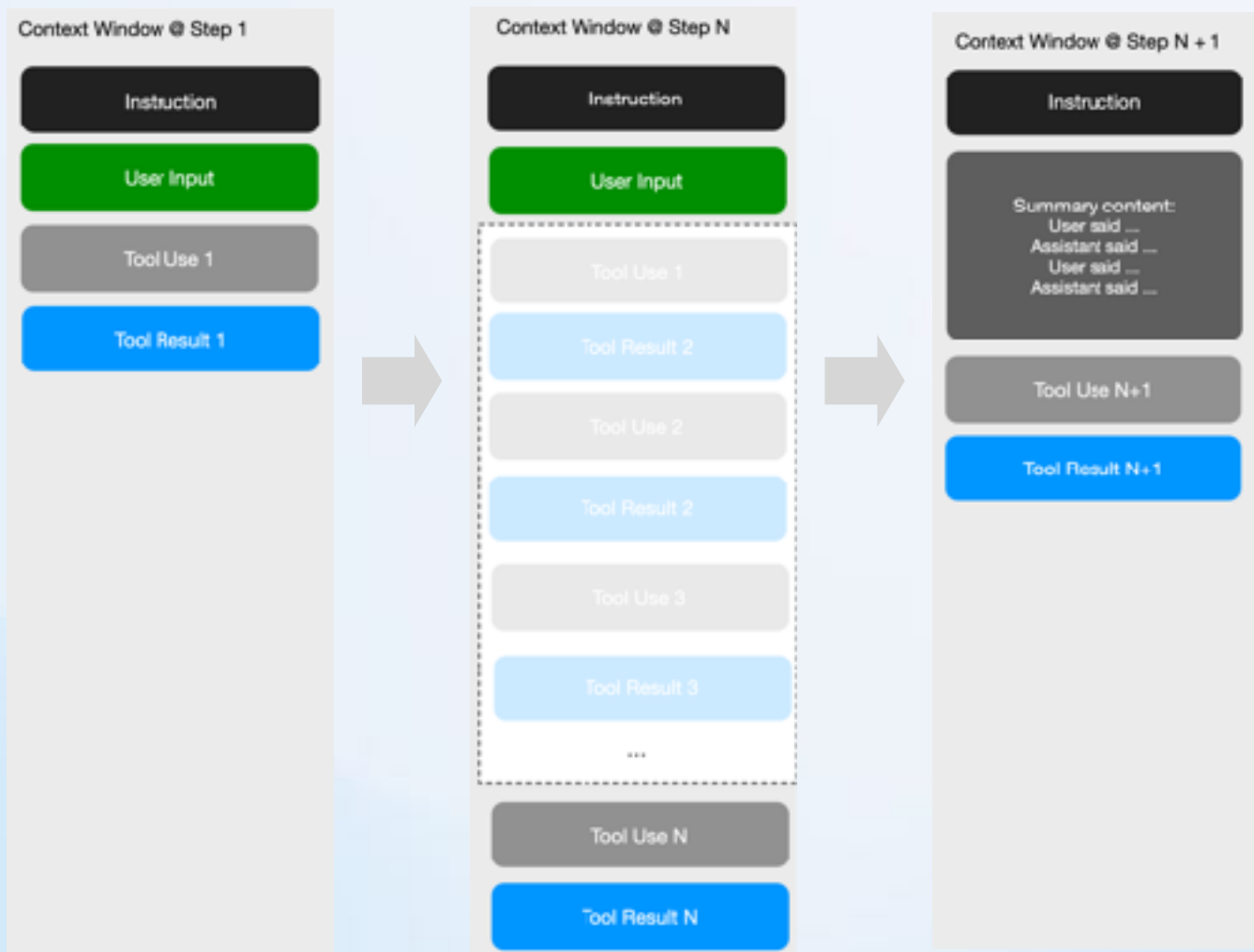
- 将 Lint Error 信息在模型每次修改完文件后及时反馈，尽早规避语法错误。
- 对于模型幻觉导致的非法 JSON Schema、无效工具调用等进行及时引导纠正。

## • Keep Tools Simple

- 精简工具列表，避免引入多个语义重叠的工具，降低模型决策难度。
- 使用 Native Function Call 代替 JSON Schema 描述工具，减少格式幻觉。



# TRAE 架构演进 – Agent 2.0 架构



- 优化上下文窗口管理，让多轮会话上下文更加连贯
  - 利用模型对历史上下文进行总结压缩，代替工程裁剪，保留更多有效信息。
  - 更大的上下文窗口，通过利用滑动窗口机制由新到旧进行动态拼接。
  - 完善 Long Memory 机制，让跨多轮会话的重要信息不被遗忘。

# TRA E 架构演进 – Agent 3.0 架构

Agentic 和 Workflow 不是非黑即白的开关，而是一个连续的光谱。我们的架构要提供可持续的迭代能力，从而在 LLM 能力足够的地方更自主化，在 LLM 还不太擅长的地方通过固定编排把专家经验固定下来，随着模型的进化可以低成本地替换节点实现逻辑。



天猪  
前端开发话题下的优秀答主

知乎

## • 面对未来的新挑战

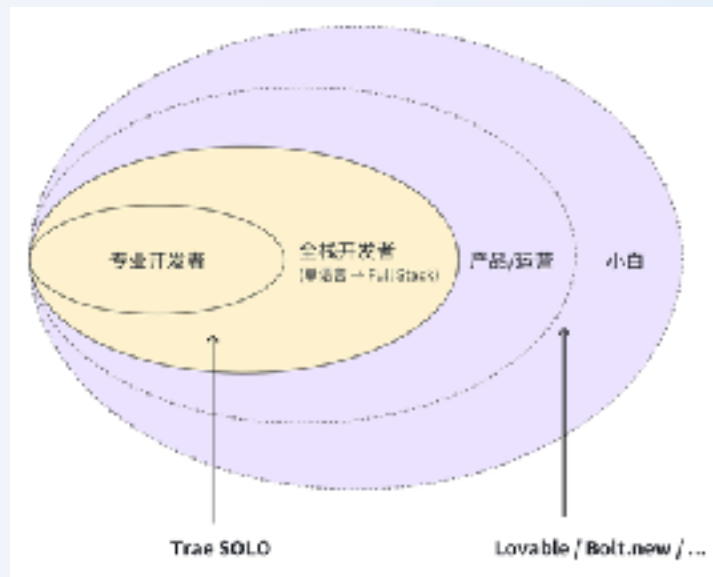
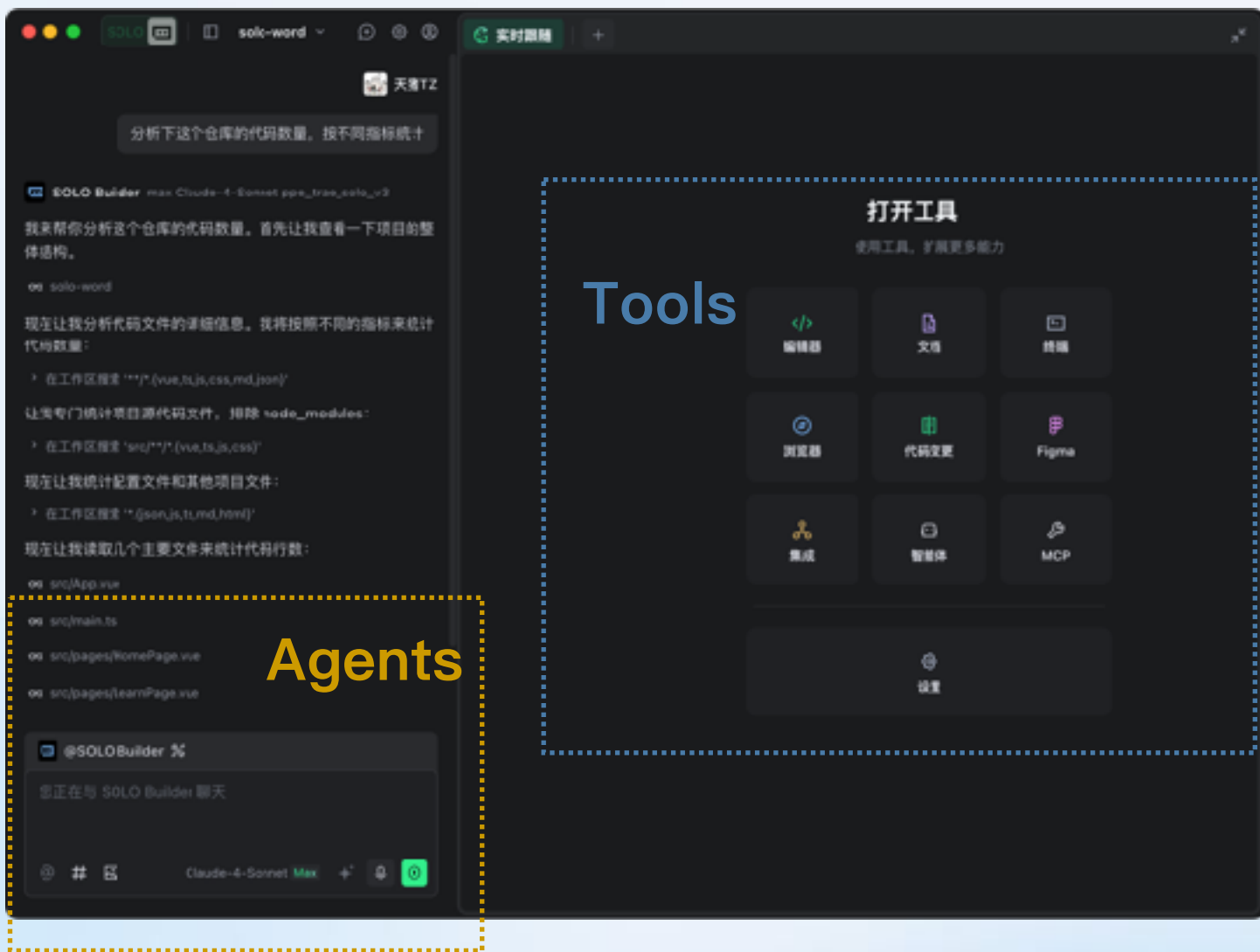
- 如何支持复杂的长耗时任务？
- 完成复杂任务，固定上下文窗口不够用怎么办？
- 任务时间越来越长，如何提高效率减少等待？
- 人如何更好地参与关键决策，确保目标符合预期？
- 当需求和环境发生变化，AI 如何及时感知？

## • 3.0 架构施工 ing... 欢迎加入我们一起加速

- Agent 工程：Multi Agent / Remote Agent / Agent 生态
- Context 工程：上下文窗口隔离、可见机制
- Tool 工程：并行 Tool Call, Sub Agent as Tool
- 环境感知：Human-in-the-loop 机制，让人及时参与关键决策，并能随时感知需求和环境变化



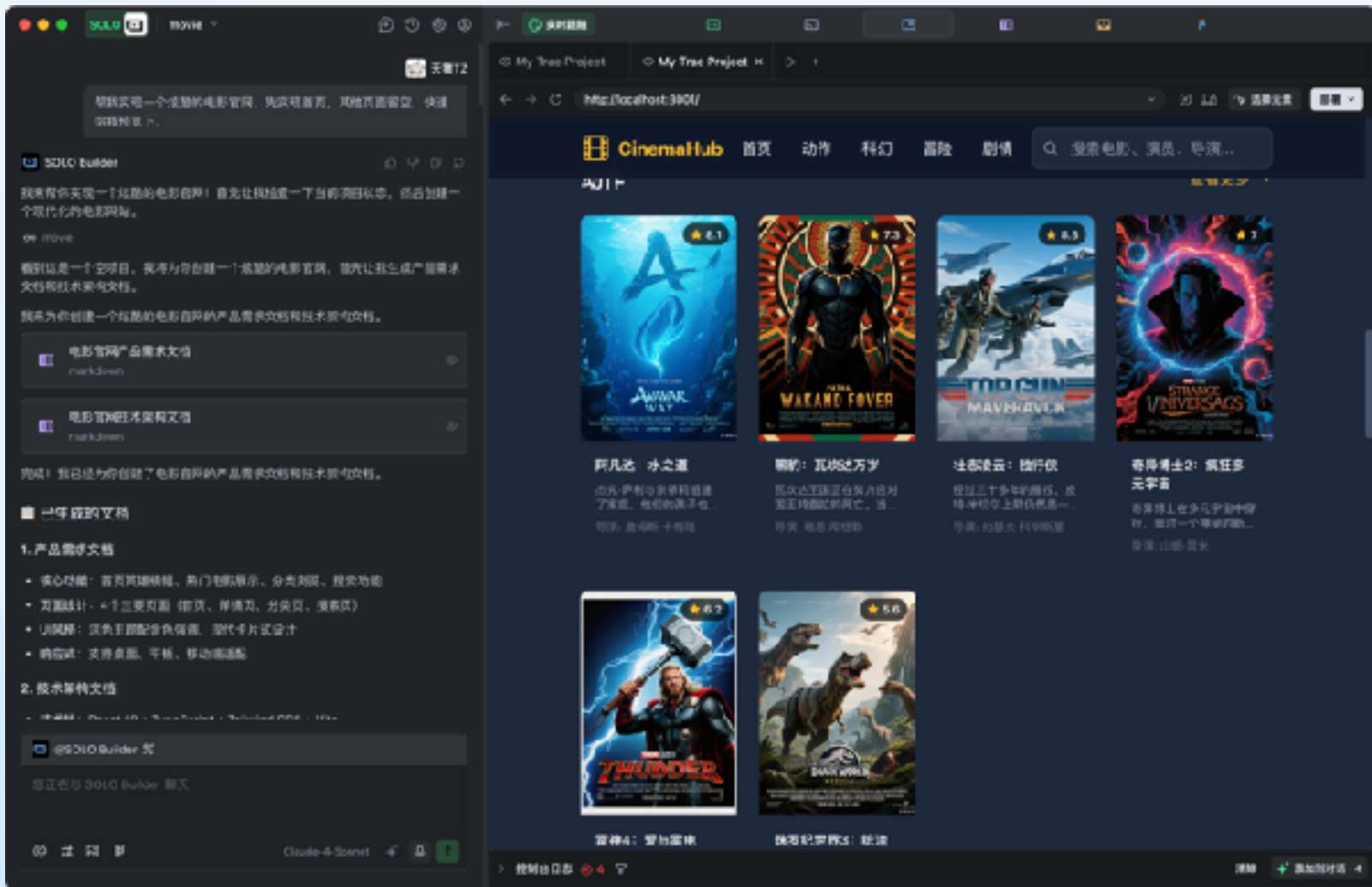
# TRAE 协作模式演进 – SOLO 模式



- 协作模式：『AI 辅助 IDE』升级为『AI 主导 IDE』
  - 以 AI Agent 为中心，IDE、Browser、Terminal 都只是一个可供 AI 调用的 Tool。
- 目标用户：服务专业开发者
- 面向垂直场景的 Agent：
  - SOLO Builder：针对 Web APP 场景的端到端交付 Agent
  - 未来还会提供更多的 Agent



# TRAE SOLO 模式 – SOLO Builder



- **SOLO Builder**：针对 Web APP 场景的端到端交付 Agent，内置 Figma、Vercel Deploy、Firebase 等 Tools
- **SOLO Builder 定位**：针对 1~5 人专业开发者团队，支撑他们快速实现一个完整的 AI 应用，让他们可以支撑好自己的 0~1000 个用户。
- 未来还会提供更多的 Agent

# TRAЕ 社区



## • 活跃的开发社区活动

- TRAЕ Meetups 聚会、TRAЕ Hackathon 马拉松、TRAЕ Friends 交流、TRAЕ Campus 校园
- 线上直播坦白局



## • 与专业开发者一起成长

- 用户面对面活动、Early Access
- TRAЕ Fellow / TRAЕ Expert

## Conclusion & FAQ



专业生产力工具的颠覆式创新，必然会全面重塑开发者的认知和开发方式，未来的 IDE 很有可能不再是当前 IDE 的“以代码为中心”形态，这很可能会发生在 3 年内。我们每个人都应该学会如何正确的和 AI 协作。



天猪  
前端开发话题下的优秀答主

知乎



寻人启事：找一个勇于探索未知，把不确定的 AI 领域逐个啃下，『抬高技术天花板』的你



AI 是每个人专属的高潜实习生。对当下的 AI 实习生，应该安排合适的任务，给予合适的指导及信息输入，并随时做好帮它兜底，并随时接手，就像你带一个真人实习生一样没啥区别，要么没用好，要么没雇好。



天猪  
前端开发话题下的优秀答主

知乎

未来已来，不要焦虑，积极拥抱 AI，结合 AI 来重塑你自己的工作流，这也是我们 Geek 程序员一直很擅长的事

📍 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 AIOps
- 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
- LMOPs
- 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

