InfoQ 极客传媒

从辅助到驱动：构建研发智能体的实践与思考

演讲人：杜沛

汽车之家 / 客户端架构师

AiCon

全球人工智能开发与应用大会

AiCon
全球人工智能开发与应用大会

目录

01 AI Coding在企业落地的挑战与思考

02 Coding Agent的多层记忆与能力增强设计

03 案例1：跨技术栈代码转换

04 案例2：Design to code 高UI还原度的技术突破

05 总结与展望

极客邦科技 2026 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

4月 · 北京 · 1200人

QCon 全球软件开发大会

会议时间：4月16-18日

- Agentic Engineering
- AgentOps
- 下一代模型架构与推理优化
- AI 原生基础设施
- 知识工程实践
- AI 安全

6月 · 上海 · 1000人

AiCon 全球人工智能开发与应用大会

会议时间：6月26-27日

- AI Infra 系统工程
- 多 Agent 协作与实践
- 多模态融合
- 模型训练与推理创新
- 数据平台与特征服务

8月 · 深圳 · 1000人

AiCon 全球人工智能开发与应用大会

会议时间：8月21-22日

- Agentic AI
- 轻量化与高效推理
- 多模态应用
- AI + IoT 场景实践
- AI 工业化落地

10月 · 上海 · 1200人

QCon 全球软件开发大会

会议时间：10月22-24日

- AI Agent
- Vibe Coding
- 智能可观测
- 推理基建
- 模型攻防
- AI x 创造力

12月 · 北京 · 1000人

AiCon 全球人工智能开发与应用大会

会议时间：12月18-19日

- 大模型架构创新
- 多模态 AI 产业融合
- 具身智能
- AI for Science
- 大模型安全

InfoQ 极客传媒

01

AI Coding在企业落地的挑战与思考

开发者的 AI Coding核心诉求

人引领
- 通用大模型
- 单指令交互

AI驱动
- 复杂任务规划
- 工具执行
- 观察与修复

AI引领

1 代码续写

2 代码问答

3 助理模式

4 代码智能体

5 自适应全自主

深度理解业务需求

希望AI能够基于项目历史需求与问题经验,自动理解日志、性能等隐性技术要求

AI代码输出一致性

不同开发者使用AI生成的代码保持统一的风格和模式,确保代码质量的一致性

UI高还原度与组件复用

AI精准理解设计稿，自动匹配并复用企业级组件库，覆盖多端场景

面临的挑战

复杂项目上下文工程

大型项目涉及多层架构、多种技术栈和复杂依赖关系,单一记忆体系难以准确把握项目全貌。AI在处理跨模块需求时,容易遗漏关键上下文,导致生成代码与实际架构脱节

规范的结构化注入

团队规范散落在文档、wiki、历史代码中，隐性知识难以显性化；难以整理成可理解的结构化形式。且规范本身也在持续演进，同步更新成本高

多模态识别准度低

AI难以精准理解设计稿中的视觉细节(间距、圆角、阴影等)、交互规则(响应式布局、动画效果)，生成的界面与预期存在偏差,需要大量手动调整。

工程质量保障

AI在代码质量自检环节缺少明确指引,容易遗漏性能优化、安全防护、异常处理等关键检查项。自检不准确导致生成代码直接用于生产环境的风险较大。

思考：让Agent变成'团队成员'

- **让Agent知道"我是谁"**

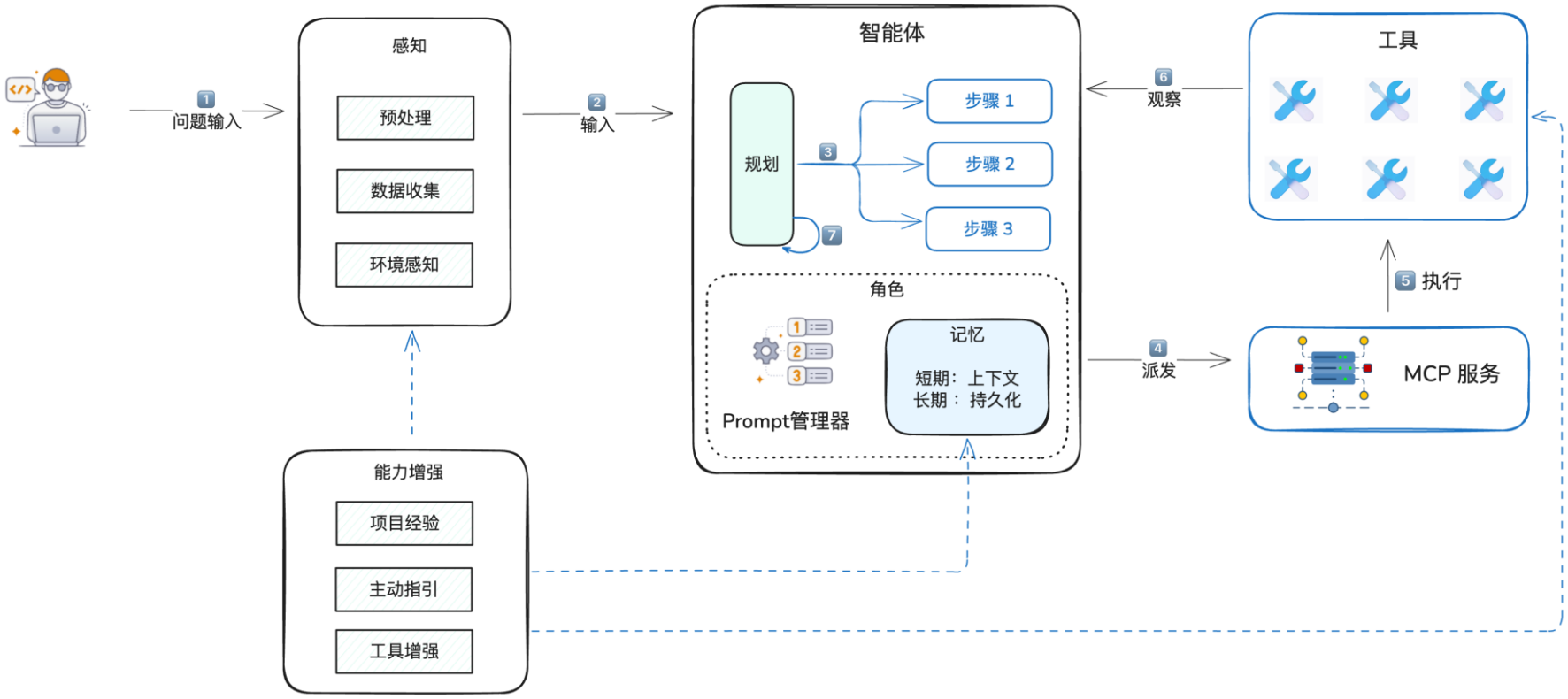
  理解团队、项目和历史上下文，不再每次都失忆

- **让Agent知道"我会什么"**

  工具是手段，技能才是目的，是真正具备处理特定领域的技能

- **让Agent知道"怎么干"**

  沉淀最佳实践与执行路径，按团队方式做事，而非从零试错



02

Coding Agent的多层记忆与能力增强设计

Agent架构设计

- **多层记忆体系**

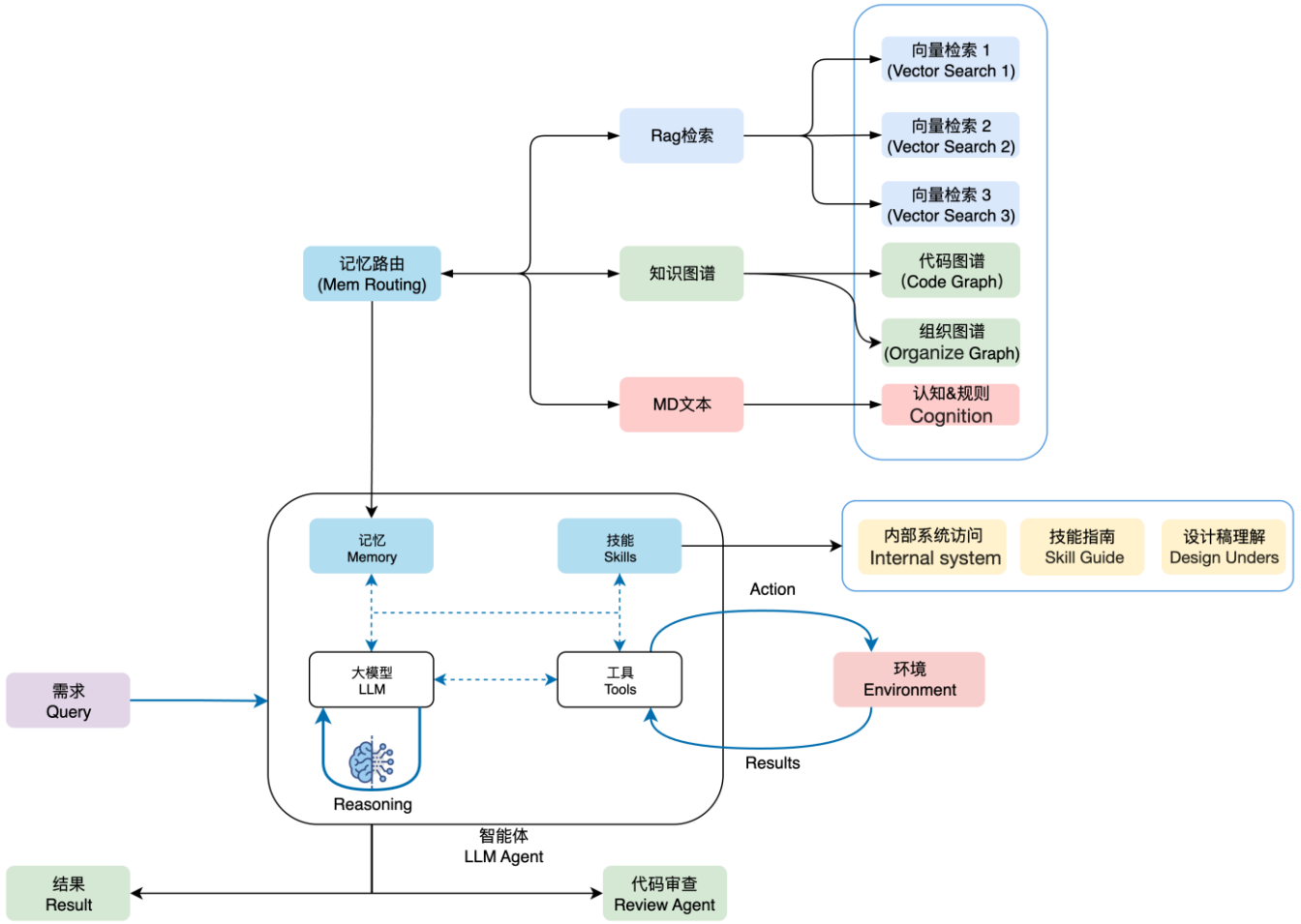
  构建个人记忆、组织记忆、项目经验多层记忆结构，让Agent既懂”我”，也懂”团队”，更懂”项目”

- **领域技能内置**

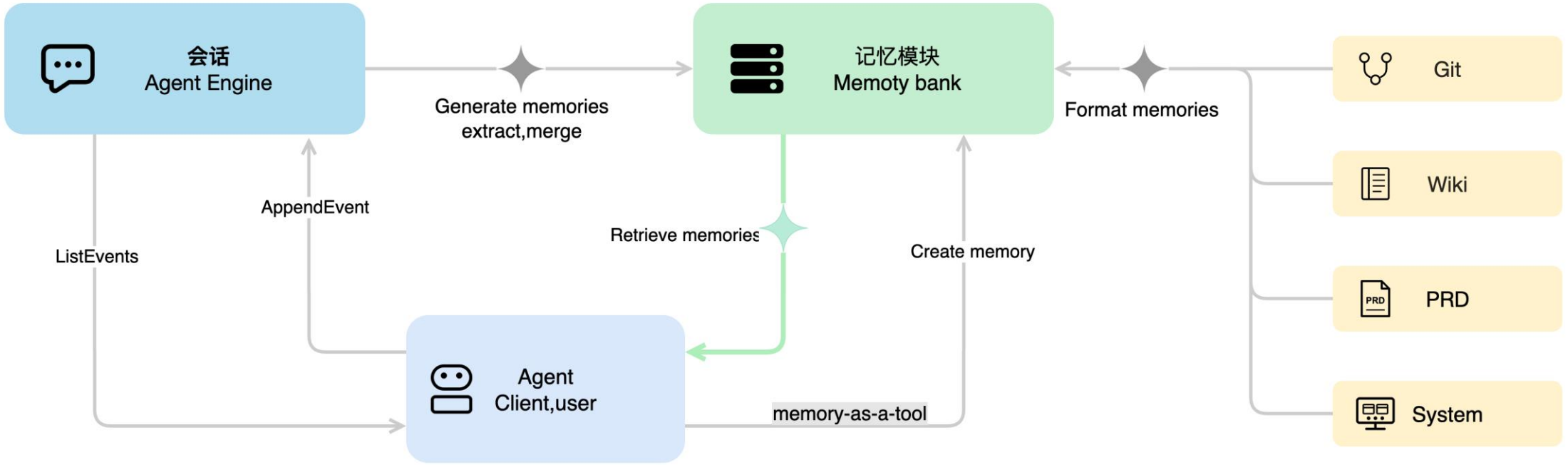
  集成内部系统访问，技能指引、设计稿理解等专业技能，让智能体具备特定领域的专业能力

- **认知规范审查**

  认知规范确保输出符合安全、合规和质量标准，约束智能体异常输出行为



记忆模块设计

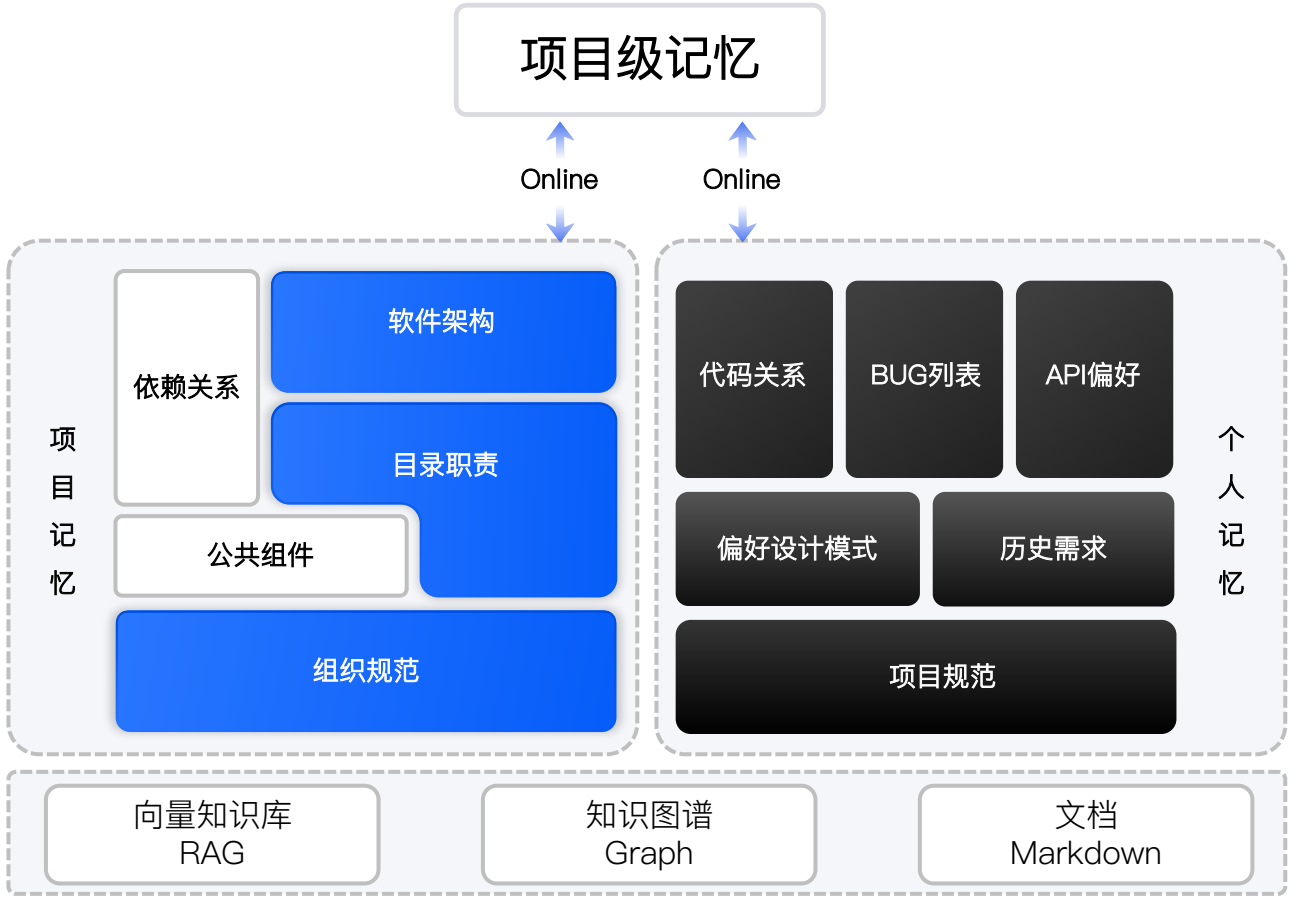


记忆来源于历史对话总结、经验灌输、文档资料、产品需求等多维输入

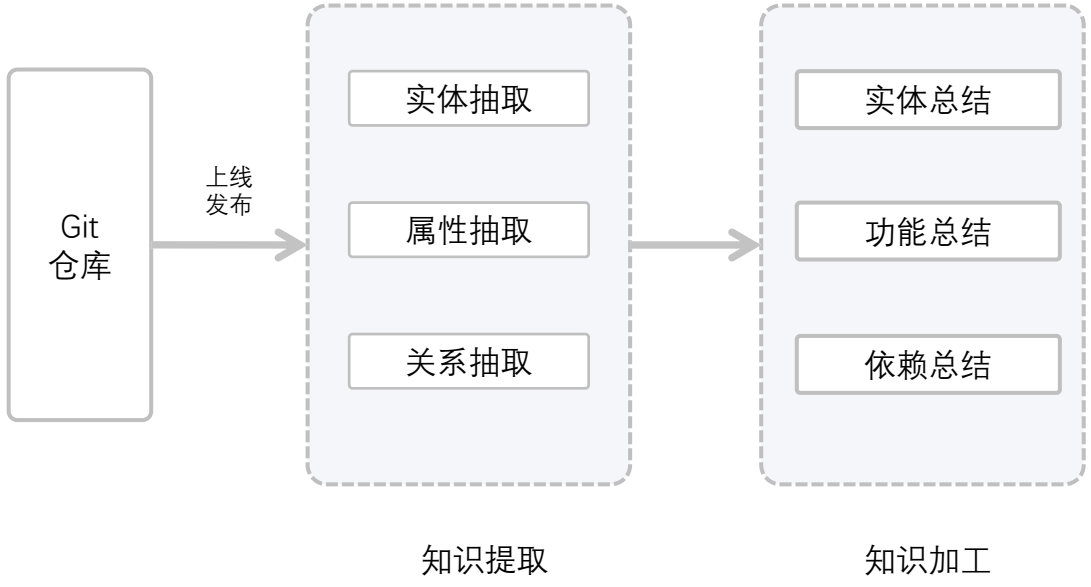
- 短期记忆：从对话上下文中动态提取和压缩关键信息
- 长期记忆：沉淀历史成功案例（项目级别），开发者组织全貌、项目细节、开发规范、公共组件等

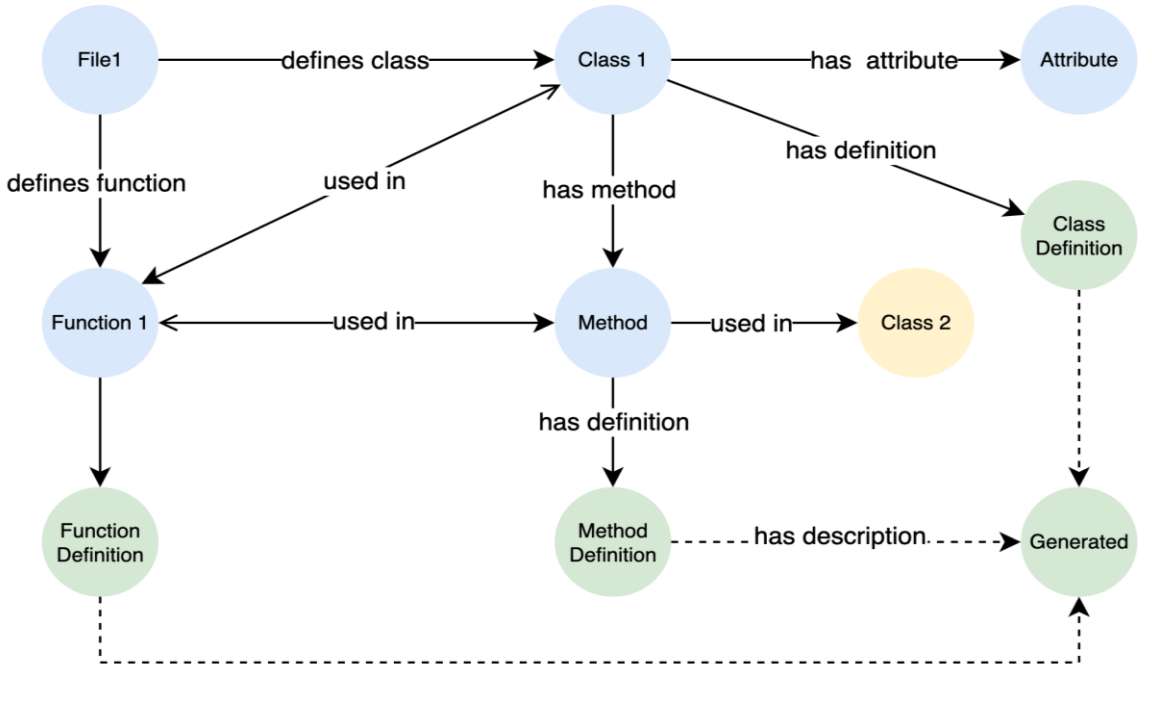
项目级记忆结构

- 依赖的项目或库（包括版本号、包管理工具如npm、pip）
- 网络架构（如REST API、GraphQL、WebSocket、网关配置）
- UI布局习惯（如组件结构、网格系统、响应式设计方法）
- 样式的选择（如CSS框架、预处理器、主题变量）
- 设计模式（如MVVM、工厂模式、观察者模式在项目中的具体应用）
- 通信方式（如HTTP客户端、消息队列、RPC框架）
- 状态持久化方案（如数据库类型、ORM工具、缓存策略）
- 路由管理（如前端路由库、后端路由配置、权限控制）
- 标准样式指南（如代码格式化规则、命名约定、linting配置）
- 项目目录职责（如文件夹结构、模块划分、资产存放位置）
- 测试策略（如单元测试框架、集成测试方法、 mocking 工具）
- 安全实践（如认证机制、加密标准、漏洞防护）
- 项目配置细节（如构建工具、环境变量、配置文件路径）
- 项目的目录结构职责 (例如/src/utils存放工具类函数)
- 使用的软件架构（如微服务、单体、事件驱动、分层架构）



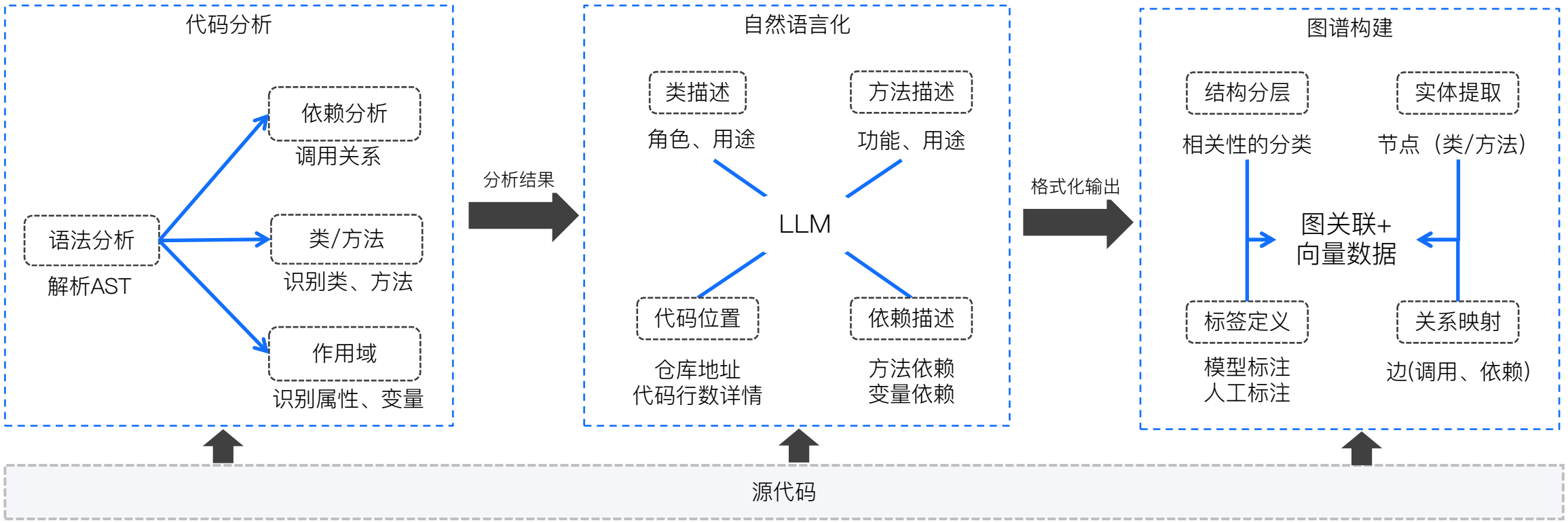
从代码到知识：实体–功能–依赖的结构化提取





构建结构化代码知识图谱，让大模型能够快速精准地理解代码结构和依赖关系，支撑代码生成。

代码图谱构建


代码分析
依赖分析
调用关系
语法分析
解析AST
类/方法
识别类、方法
作用域
识别属性、变量
分析结果
自然语言化
类描述
角色、用途
方法描述
功能、用途
LLM
代码位置
仓库地址
代码行数详情
依赖描述
方法依赖
变量依赖
格式化输出
图谱构建
结构分层
相关性的分类
实体提取
节点（类/方法）
图关联+
向量数据
标签定义
模型标注
人工标注
关系映射
边(调用、依赖)
源代码

记忆的持续学习

- **知识沉淀**

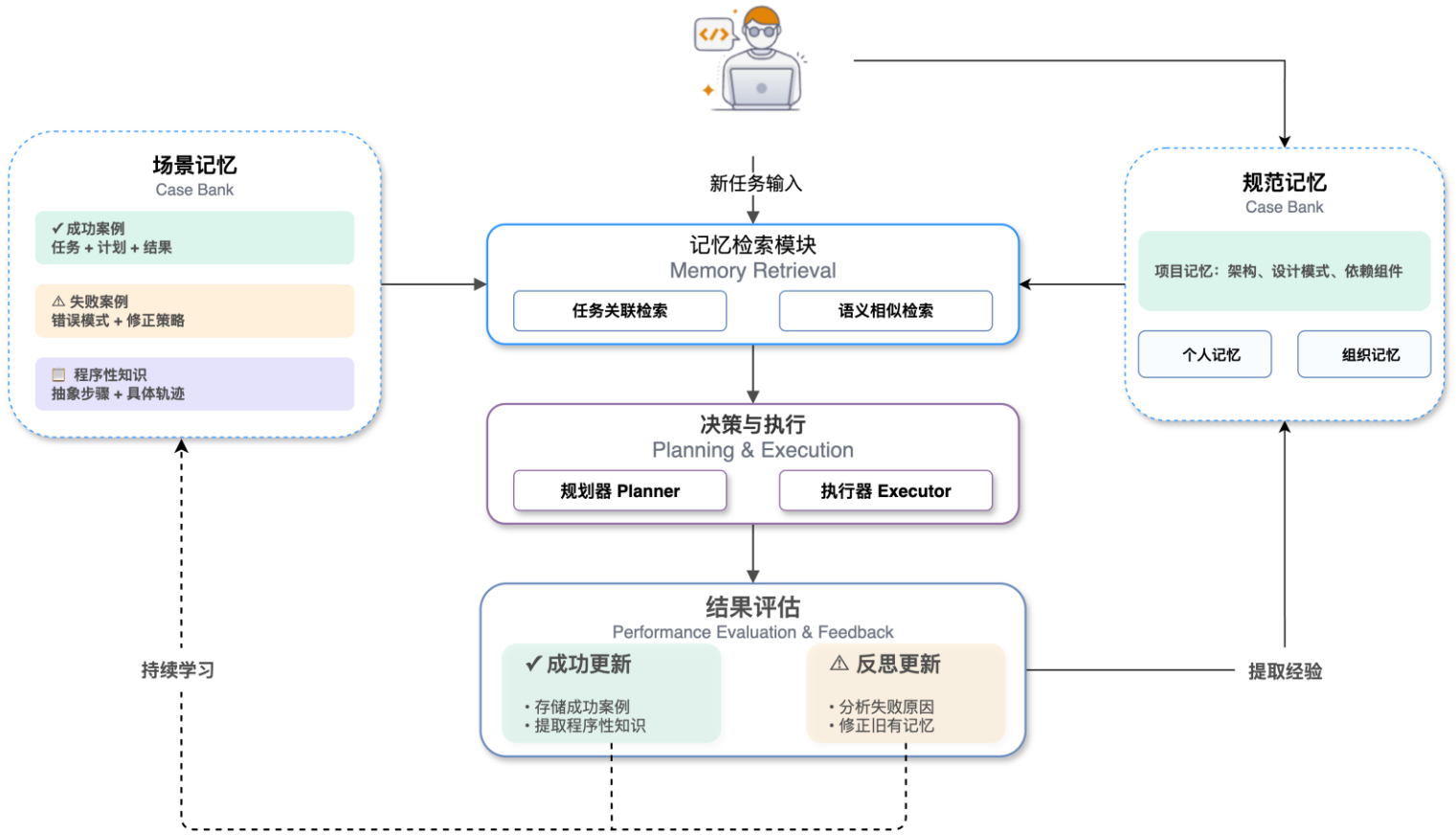
  规范记忆沉淀团队标准，定义”应该怎么做”；场景记忆从任务执行中积累，提供”怎么做才对”的经验参照

- **记忆检索**

  识别任务意图，按需召回对应记忆；动态组合场景经验与规范约束

- **闭环进化**

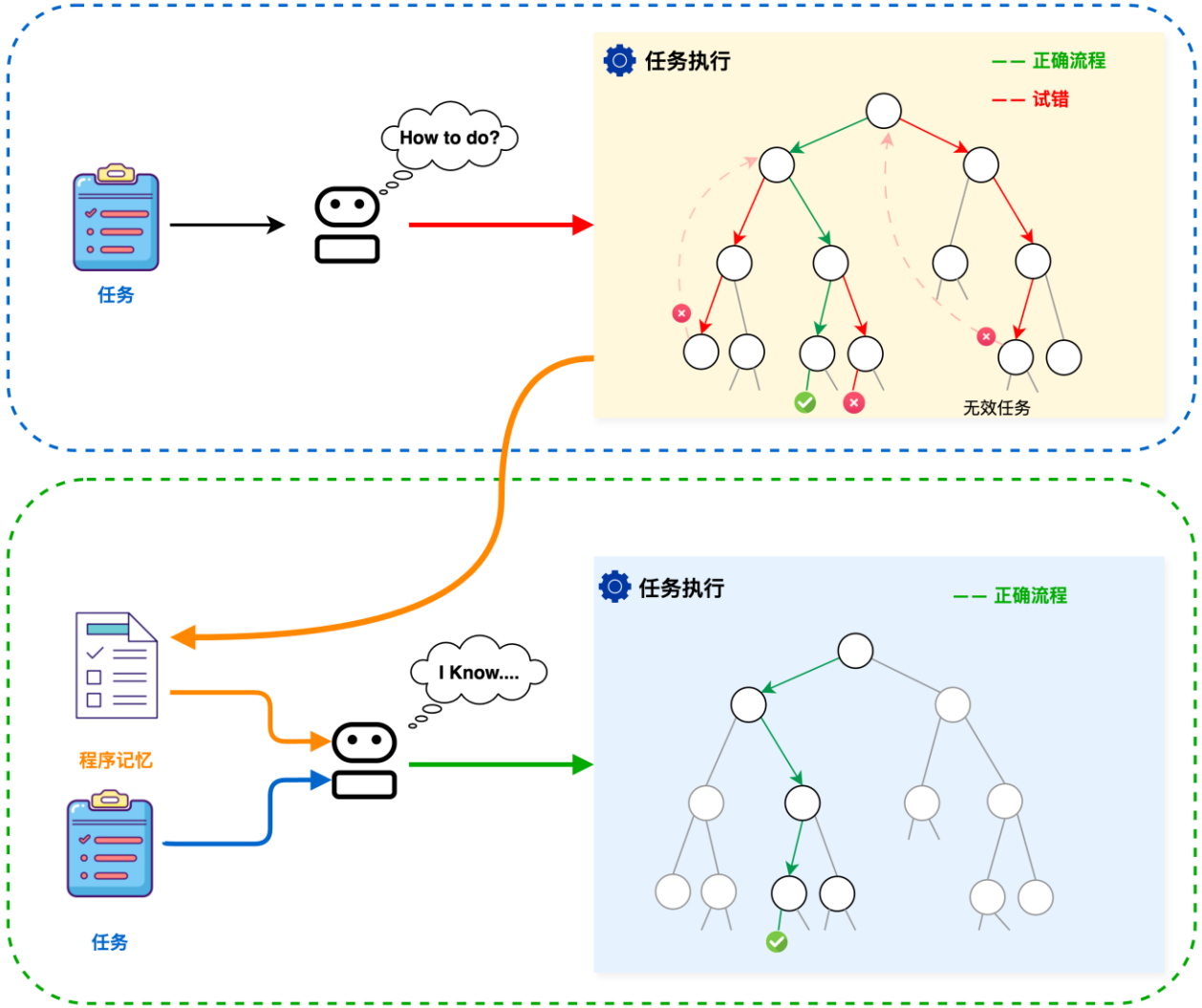
  执行结果自动反哺记忆库：成功案例沉淀为场景记忆，通用规律提炼为规范记忆，持续自我优化



“学会”正确做事

经验指引，精准执行：

- 消除产出不确定性：从“How to do?”升级为具备主动判断力的 “I Know....”
- 规范内化为先验： 将复杂决策树简化为可预测的、规范化的最短路径
- 指引式收敛：通过结构化的指引经验，Agent跳过不必要的探索和验证环节



02

案例1：跨技术栈代码转换

跨端代码转换

面临的挑战

- **平台特性差异**

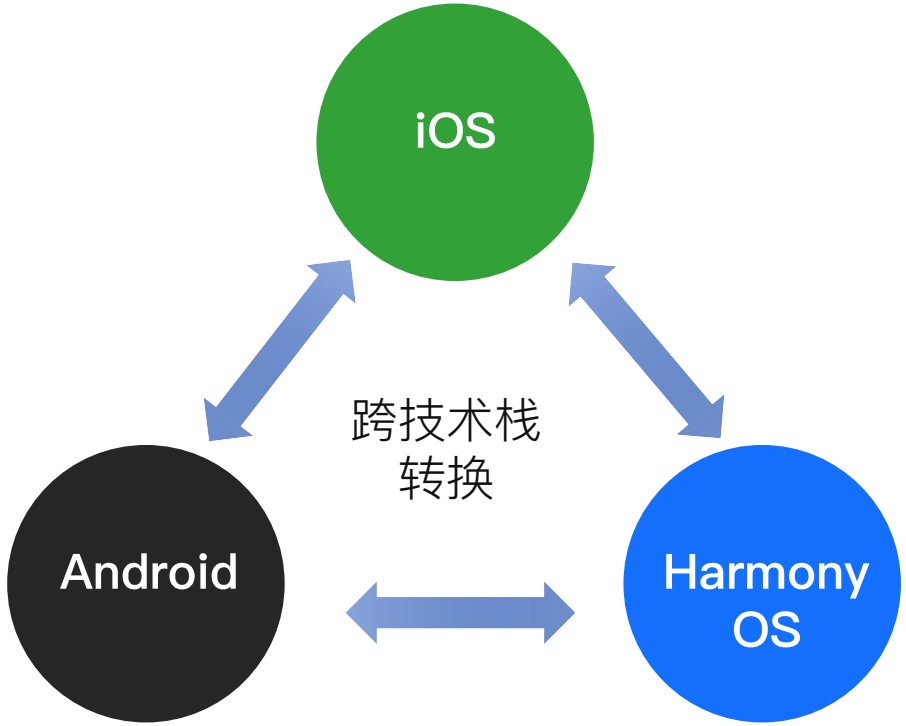
  各平台生命周期、权限管理、导航模式等底层机制不同，无法直接平移，需针对性适配

- **API对应关系**

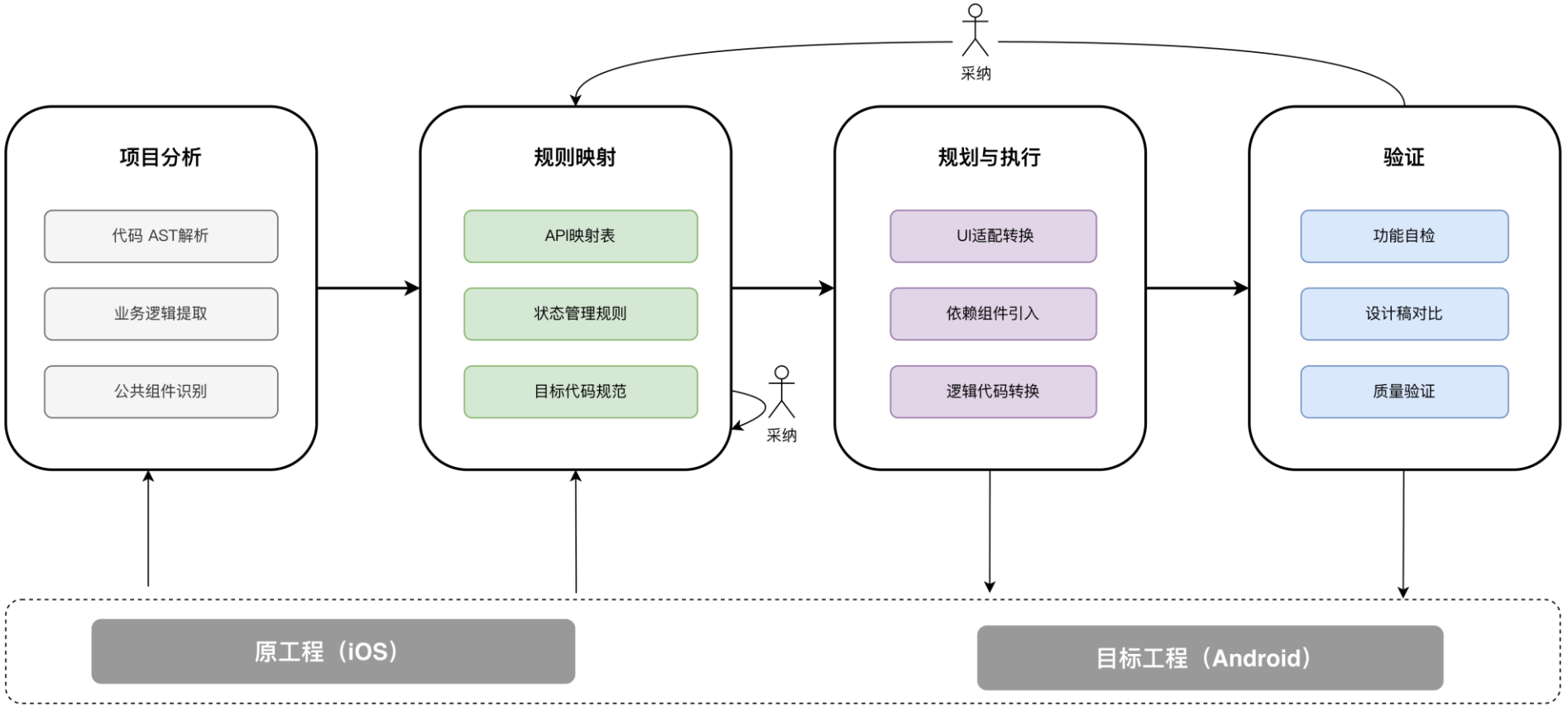
  不同平台依赖的三方组件、API差异等，如果没有清晰的对应关系，模型不知道怎么转

- **框架适配**

  UI框架组件体系不兼容，功能相似但实现机制、属性配置、事件处理方式都不同，只是一对一转换可能出现大量不适配问题

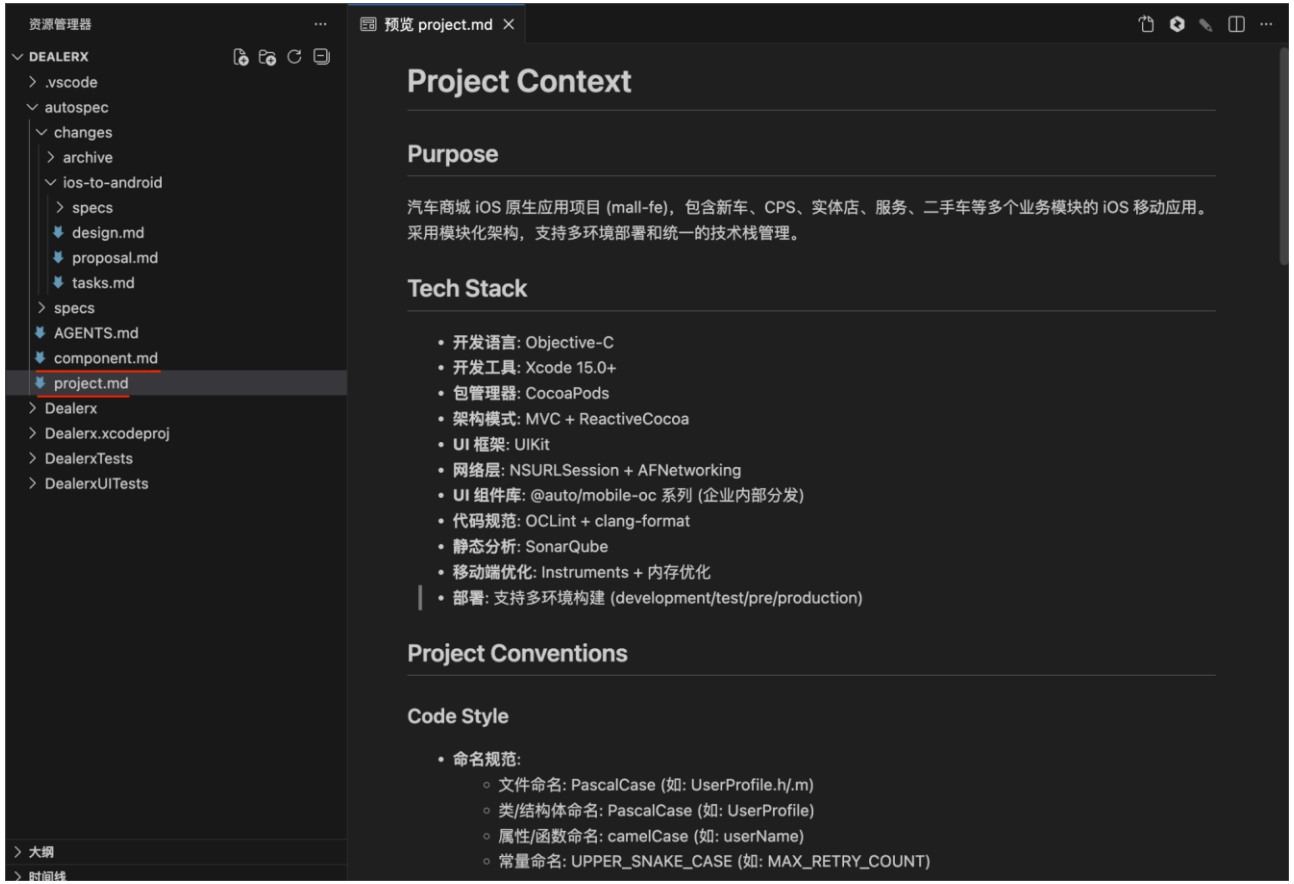


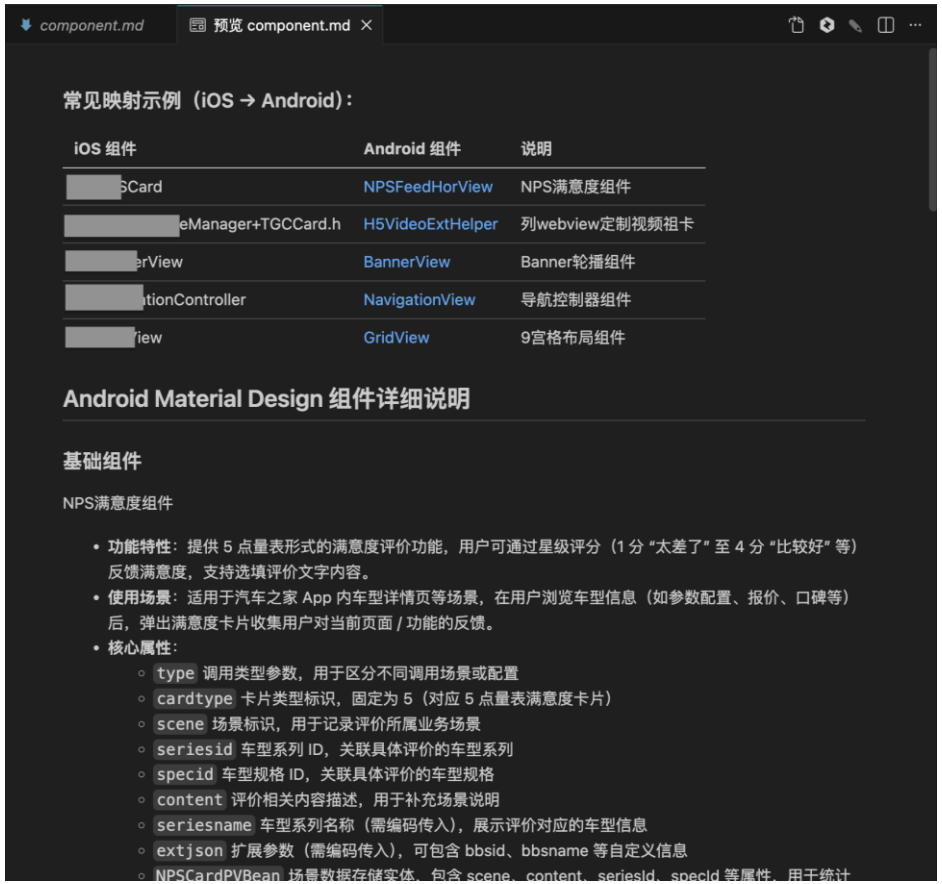
基于Spec的跨技术栈代码转换方案



将任务结构化为Rules/Templates/Workflow，LLM基于明确的规范执行转换，确保输出的准确性。

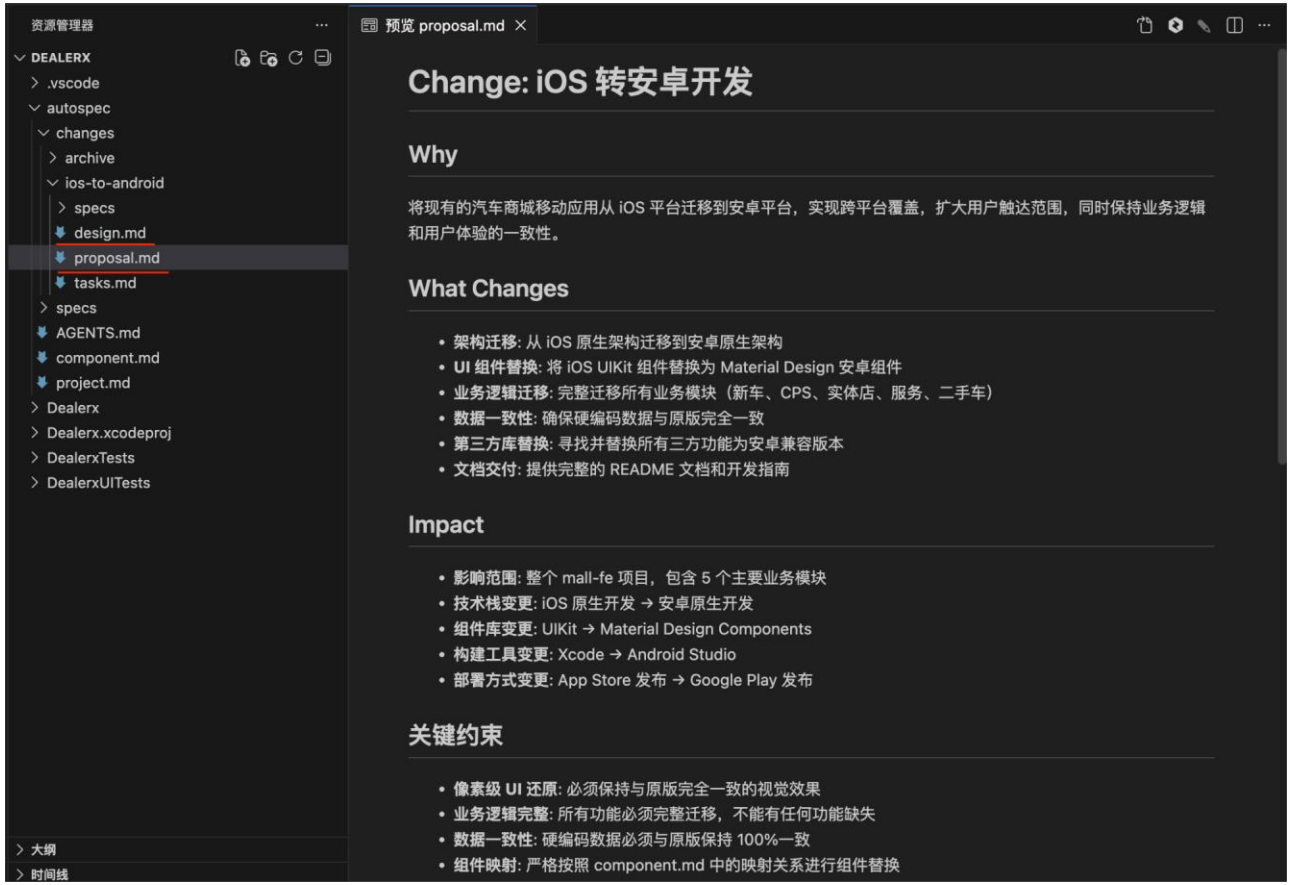
工程描述与组件映射

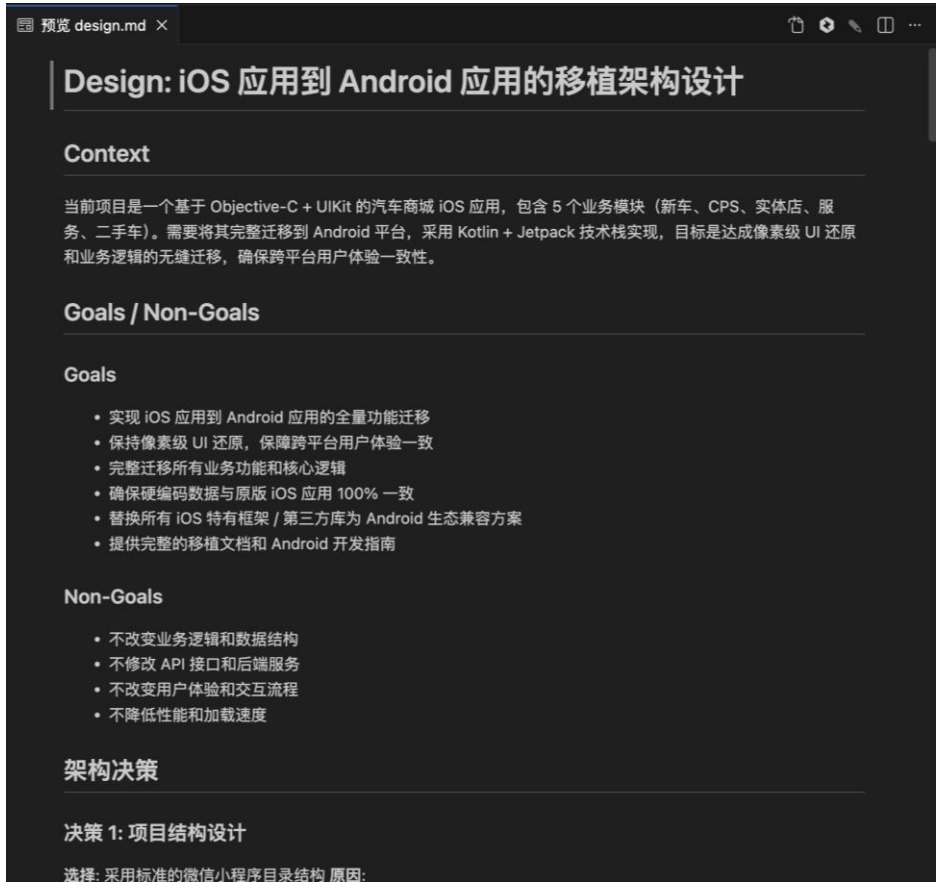




解析源工程技术栈（project.md），建立 iOS → Android 组件映射表(component.md)。

提案与架构设计





基于工程分析自动生成 proposal.md（迁移方案与约束）和 design.md（架构决策与技术选型）

任务拆解与执行计划

- **有序编排**

  任务按执行顺序排列，前置任务完成后才推进后续任务

- **复用优先**

  优先完成组件层和通用能力，业务模块直接复用

- **粒度可控**

  每个任务明确输入输出，可独立执行、可单独验证

- **可追踪**

  任务状态清晰，便于进度跟踪、代码审查和质量验收

04

案例2：Design to code 高UI还原度的技术突破

Design to code 高还原度的关键要素

像素级的还原度

精确还原间距、字号、颜色等视觉细节，确保像素级精准

准确生成并实现卡片中的交互信息(如左右滑动、展开收起等)

01

UI组件复用

自动识别重复的UI 模块并抽离为独立组件

已有组件库中的组件,优先复用而非重复创建

02

布局适配能力

支持响应式布局，适配不同屏幕尺寸与分辨率

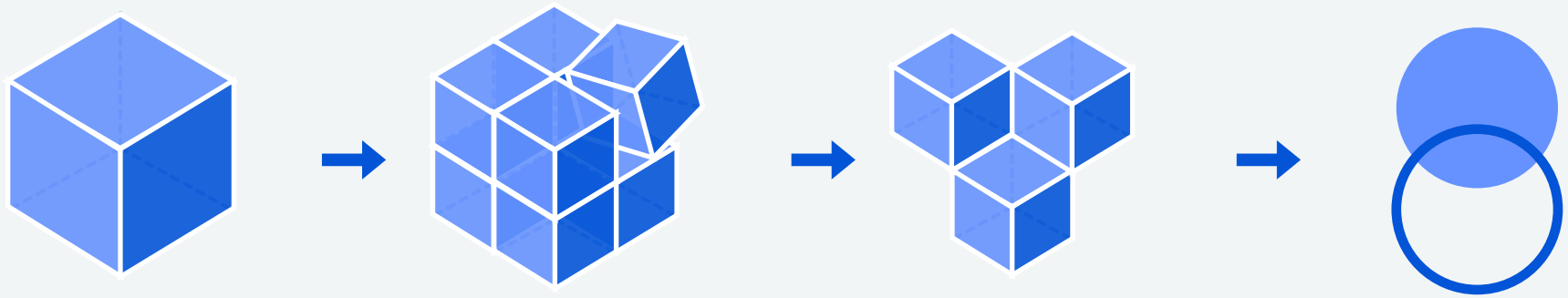
生成结构清晰的弹性代码，满足多端部署要求

03

Design to code 的工程化路径

核心概念：“切”

把复杂页面简单化，把简单问题做精准



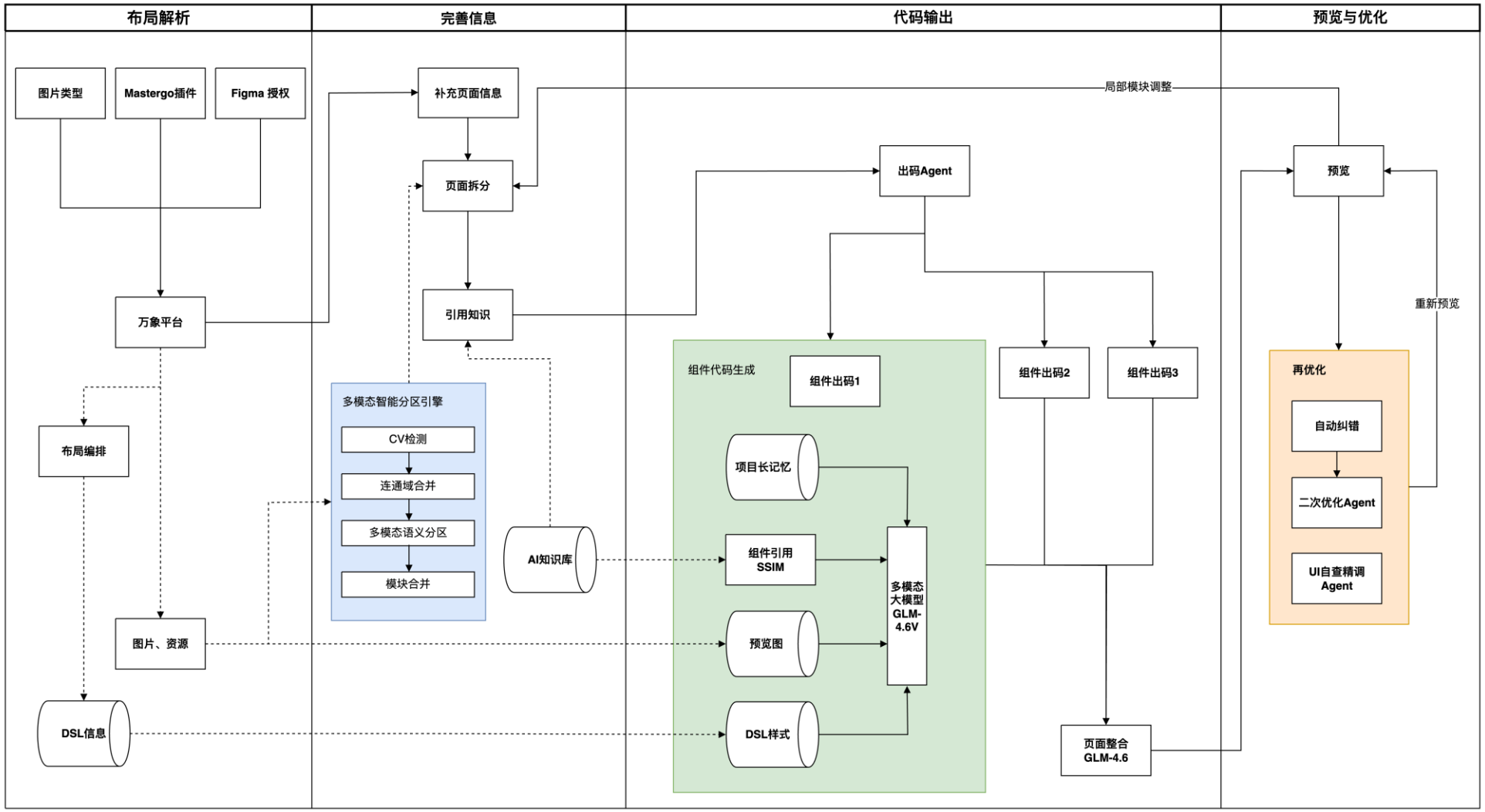
结构化分析　组件拆分生成　页面拼装　验证纠错

四大原则

1 原子化拆分　2 组件复用　3 渐进式增强　4 边界清晰化

准则　方法

D2C： 多模态分析+DSL解析



拆分示例

自动拆分卡片

将页面拆分为多个卡片区域，用户可再进行拖拽修正

自动识别组件

基于结构相似度(SSIM)查找到引用的组件

卡片要求描述

可以描述组件中的特殊交互与隐形要求

AI驱动的UI迭代优化

- **视觉差异自动检测，像素级精准对比**

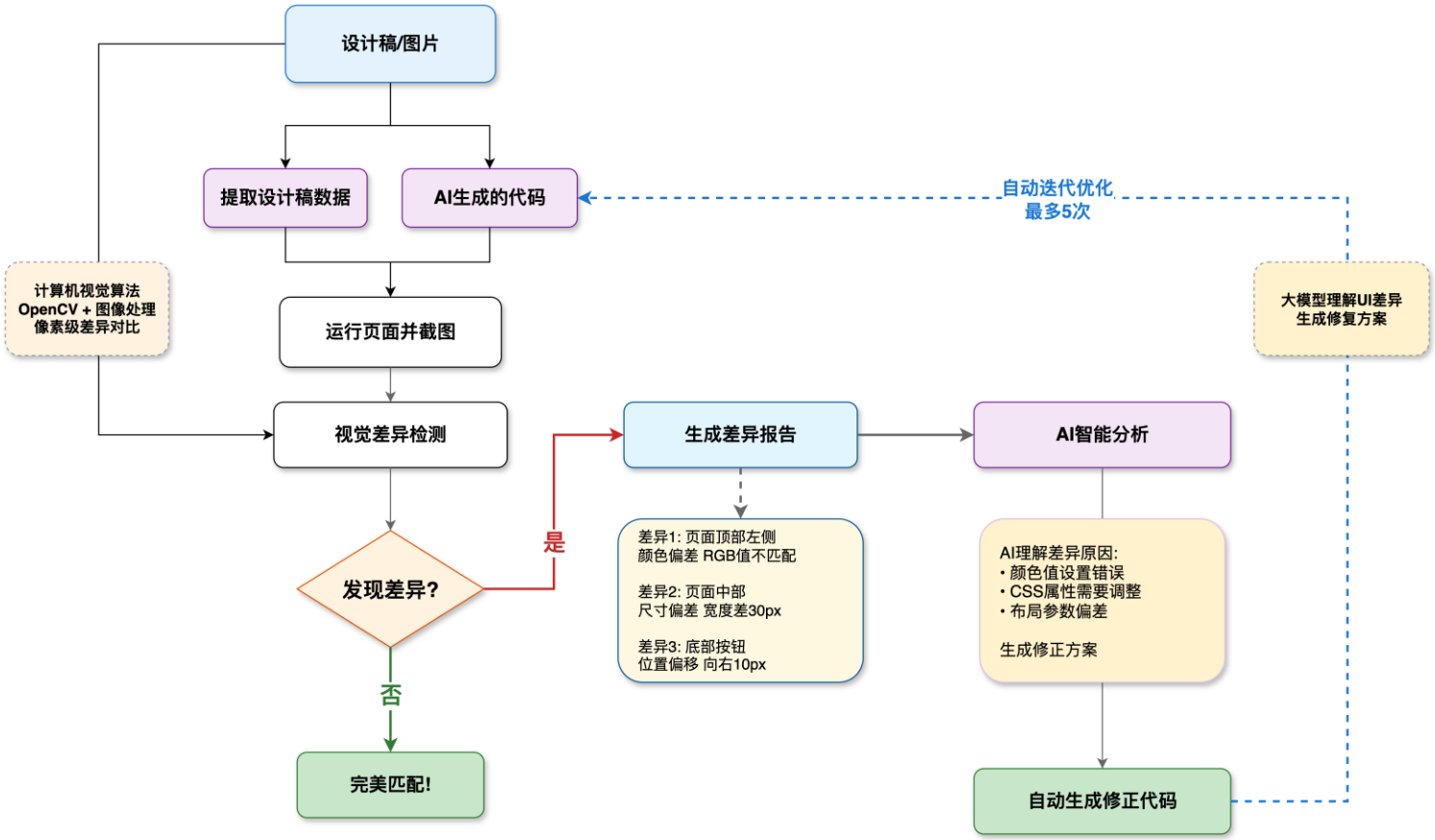
  统一设计稿和运行截图数据，基于OpenCV(热力图)+OCR，自动标注出颜色、字号、布局差异等

- **AI智能分析差异，自动生成修复方案**

  深度解析差异原因，输出针对性的代码优化建议

- **迭代优化闭环，持续提升还原精度**

  自动迭代修复直至达标，确保生成代码与设计稿高度一致



D2C演示


Chrome 文件 编辑 显示 历史记录 书签 个人资料 标签页 窗口 帮助
11月24日 周一 15:36
万象编码平台-设计稿出码
wanxiang.corpautohome.com/d2c/smartcode
有新版 Chrome 可用
WANXIANG
设计稿出码
智能体
IDE插件
AI代码审查
AI接口平台
意见反馈
关于
新聊天
历史聊天
快速上手
生成效果对比
你想要构建什么?
无论是设计稿还是图片，AI 都能帮你快速出码
描述你要实现的功能，例如：根据设计稿生成登录页
上传图片
Figma
一键优化
react
AutoDesign
关联Git仓库
生成
快速入门
通过设计稿生成代码
智能 UI 代码生成
「设计稿、图片皆可智能出码」
意见反馈
关于我们
万象支持图片、Figma、MasterGo多种格式

05 总结与展望

总结：理解AI能力边界，扬长补短

补齐短板

- 长token遗忘

  上下文分段与记忆管理

- 视觉理解的局限性

  复杂UI结构简化与拆解

- 业务理解缺失

  业务DSL与知识图谱

发挥长板

- 强大的代码生成能力

  专注AI擅长的代码编写

- 逻辑推理与决策

  智能分析与架构设计

- 跨技术栈能力

  快速适配多种技术场景

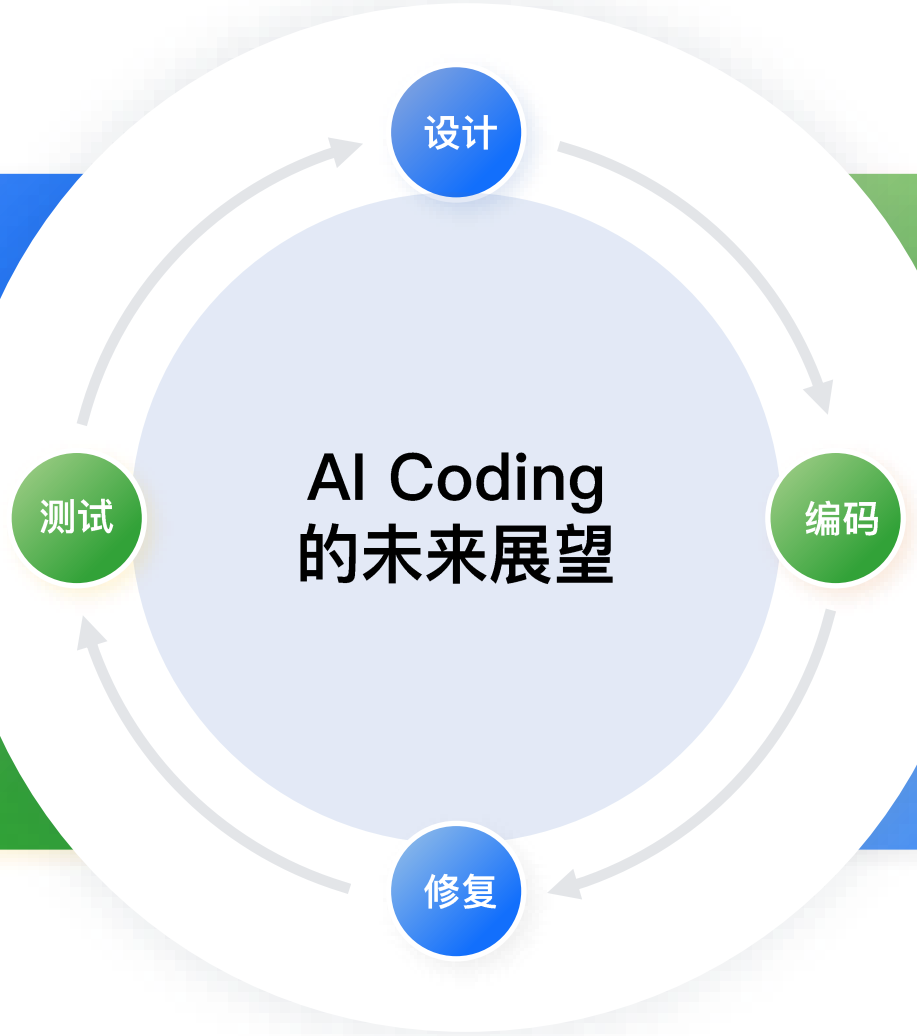
展望

从编码者到架构设计者

从编写代码转向系统设计、架构决策和质量把控

自然语言成为编程接口

开发者用需求描述替代代码编写， AI 理解意图并生成实现



自动缺陷检测

从用户视角模拟真实操作，自动发现交互异常与体验问题

自主修复形成闭环

基于检测的问题自主分析原因、生成修复方案和代码

极客邦科技 2026 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

4月 · 北京 · 1200人

QCon 全球软件开发大会

会议时间：4月16-18日

- Agentic Engineering
- AgentOps
- 下一代模型架构与推理优化
- AI 原生基础设施
- 知识工程实践
- AI 安全

6月 · 上海 · 1000人

AiCon 全球人工智能开发与应用大会

会议时间：6月26-27日

- AI Infra 系统工程
- 多 Agent 协作与实践
- 多模态融合
- 模型训练与推理创新
- 数据平台与特征服务

8月 · 深圳 · 1000人

AiCon 全球人工智能开发与应用大会

会议时间：8月21-22日

- Agentic AI
- 轻量化与高效推理
- 多模态应用
- AI + IoT 场景实践
- AI 工业化落地

10月 · 上海 · 1200人

QCon 全球软件开发大会

会议时间：10月22-24日

- AI Agent
- Vibe Coding
- 智能可观测
- 推理基建
- 模型攻防
- AI x 创造力

12月 · 北京 · 1000人

AiCon 全球人工智能开发与应用大会

会议时间：12月18-19日

- 大模型架构创新
- 多模态 AI 产业融合
- 具身智能
- AI for Science
- 大模型安全

InfoQ 极客传媒

Q&A

InfoQ 极客传媒

THANKS

探索 AI 应用边界

Explore the limits of AI applications

AiCon

全球人工智能开发与应用大会