

QCon
全球软件开发大会

前端领域知识库构建与辅助 代码研发中的创新实践

杨颢

蚂蚁集团





自我介绍



支付宝体验技术部
引领体验科技，驱动数字生活

姓名：杨颢（花名：颢阳）

背景：浙江大学计算机硕士

研究方向：计算机视觉、多模态大模型、AIGC等。

主要工作：

蚂蚁集团-支付宝体验技术部-智能体验团队算法方向负责人。

目前主要负责前端和体验技术领域的算法研究和工程化落地（AI For Frontend）。曾参与支付宝前端 3.0 智能研发、UI 理解生成大模型 UI-UG、智能 UI 推荐、用户体验度量、动效 AIGC等项目。

目录

01 前端领域知识库建设

02 代码仓库描述新范式

03 智能召回体系实践

04 领域知识辅助研发案例

05 总结与未来展望

📍 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 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

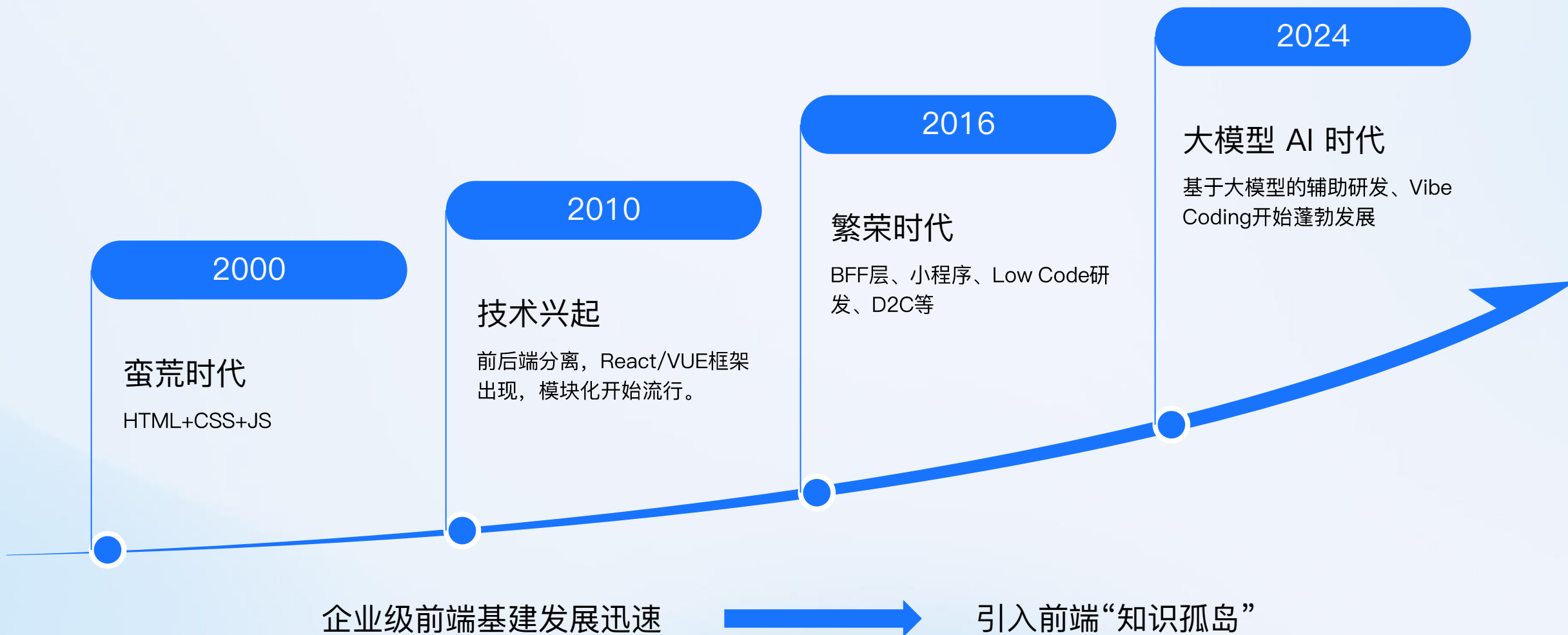


查看会议

01

前端领域知识库建设

前端研发演进历程



大模型对于领域知识的局限性

通用大模型无法理解领域知识

包括：

🔑 大模型无法理解企业级前端框架、组件、JSAPI、平台、工具等。

📖 大模型无法理解企业级前端域内代码范式。

Q: Bigfish 是什么？

A: 大鱼，一首歌曲，一部电影，一家游戏公司...

实际上：

蚂蚁集团内部的企业级前端开发框架。

More：

如何使用 Bigfish 框架、研发规范、示例代码等。

于是，我们重新思考：

- 大模型缺的不是算力，而是「企业级前端世界观」。
- 大模型缺的不是上下文窗口，而是「上下文质量」。

💡 解决方案：

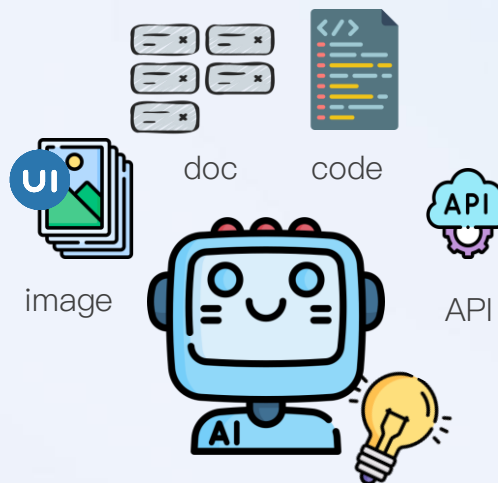
前端定制的RAG（检索增强生成）

💪 挑战：

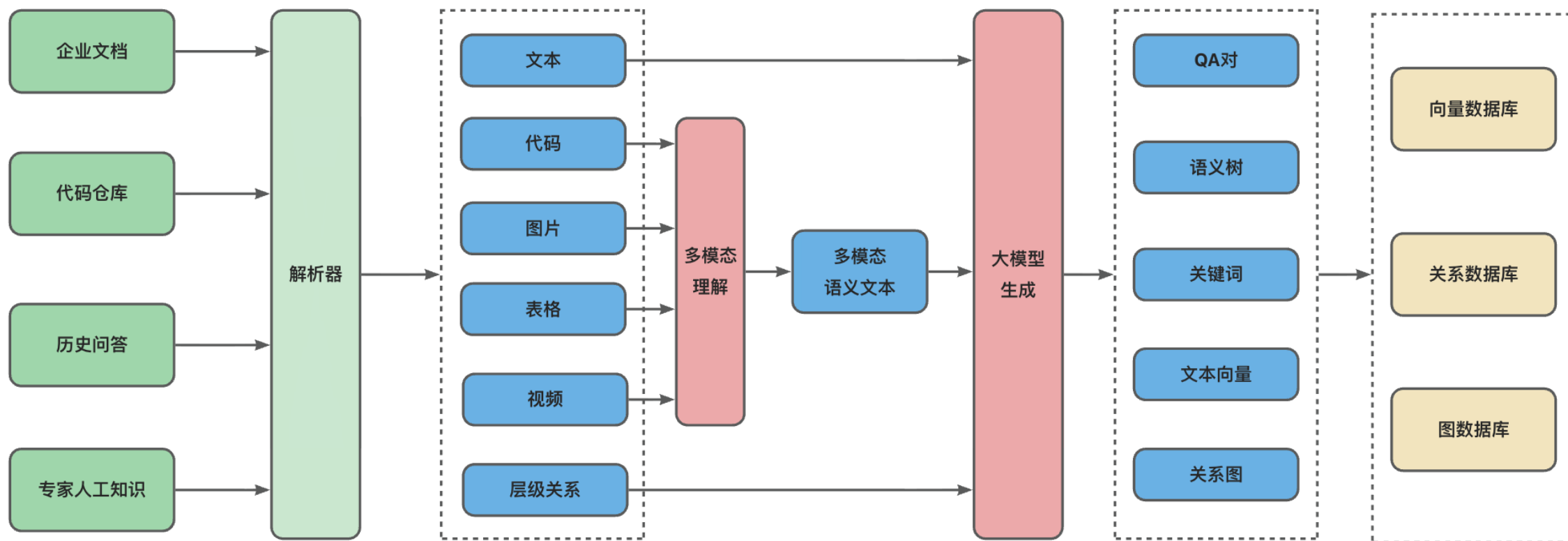
前端的各种模态问题

上下文限制

召回准确率



前端领域知识库-知识录入大图



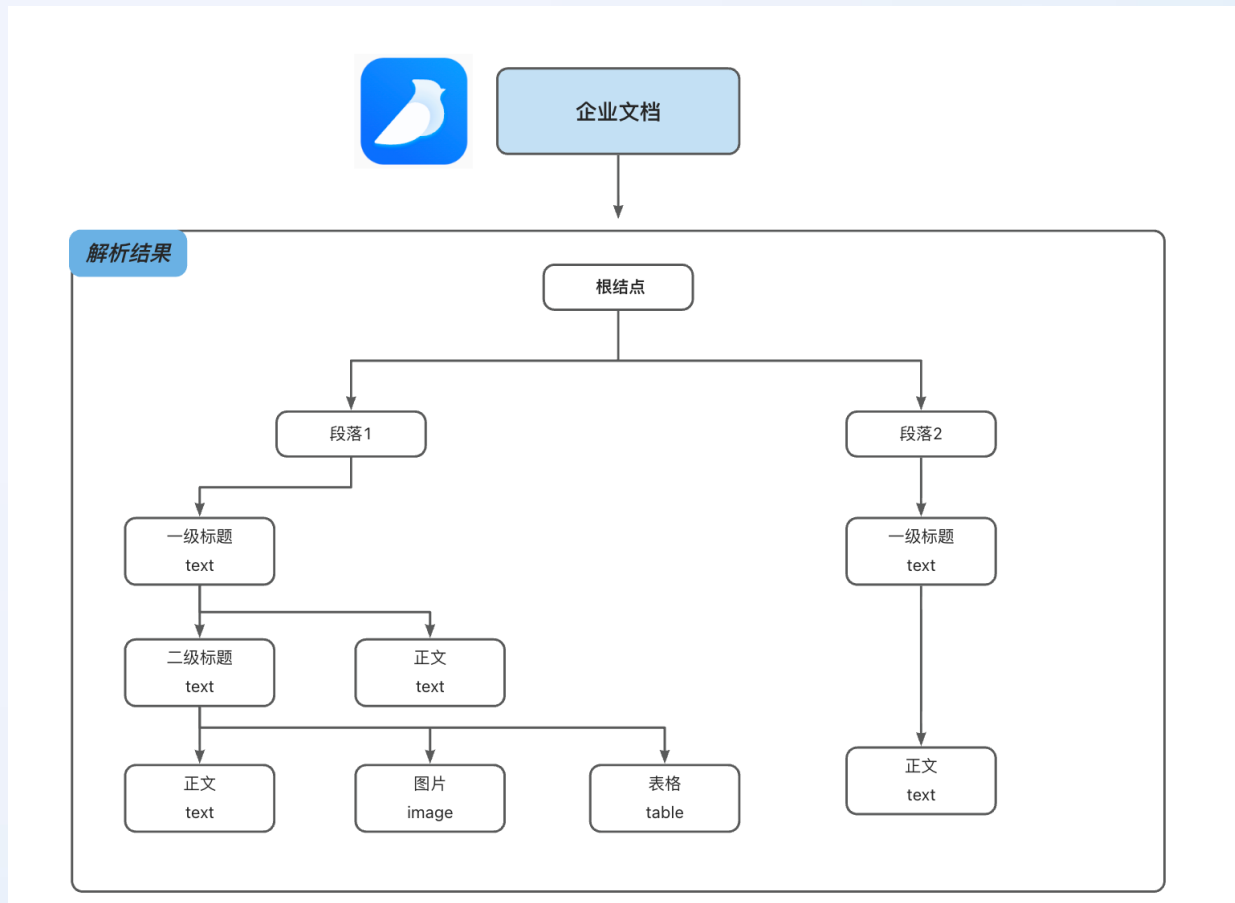
多层级知识管理

文档多层级结构处理

RAG应用的效果上限，很大程度上取决于对非结构化、半结构化和结构化数据的处理能力。因此我们需要有条理地组织知识信息，并提供更加丰富的多元化的语义信息。

对于企业级的文档，我们使用了文档解析器，把企业内部文档解析成带层级的资产树，节点为文本片段、图片、视频等。

构建树状或图状的知识结构，对提升后续召回效果至关重要。



● 不同类型的数据处理

文档型数据

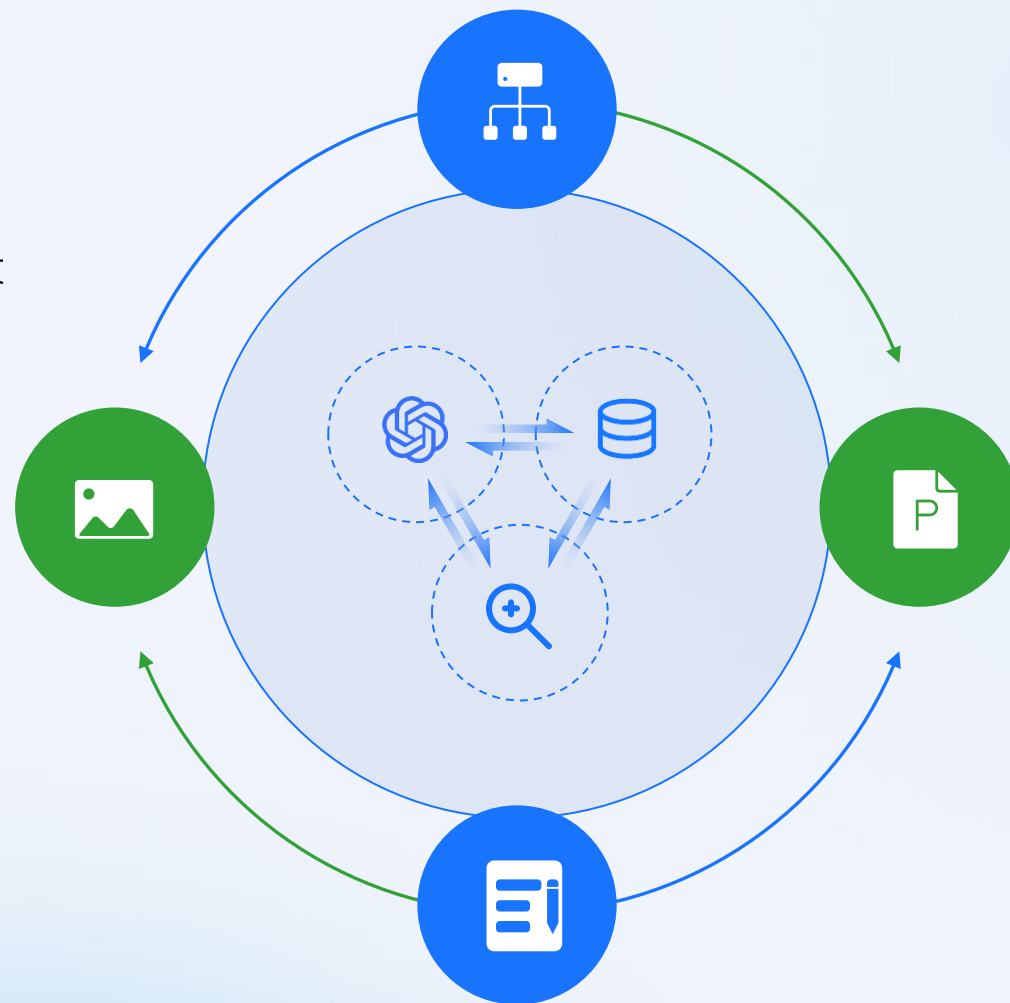
对于文档数据，我们使用基于Markdown文档结构与固定Chunk的两种方式结合，进行文字片段的抽取。上下文最多支持8K。

图片数据

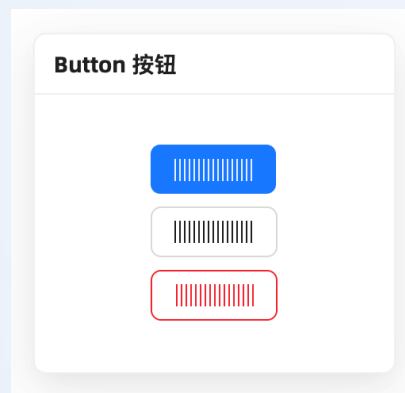
除了原始图片文件外，对于图片数据，我们额外使用多模态大模型的描述能力，抽取图片的文字内容和图片总结信息。这样图片数据可以在文本/多模态召回链路同时生效。

代码数据

除了官方框架文档的人工撰写的示例代码/代码片段外，对于企业内部依赖包，我们基于大模型自动完成了内部版Context7的复现，以实现更好的代码模块覆盖率。



前端组件录入



基础组件

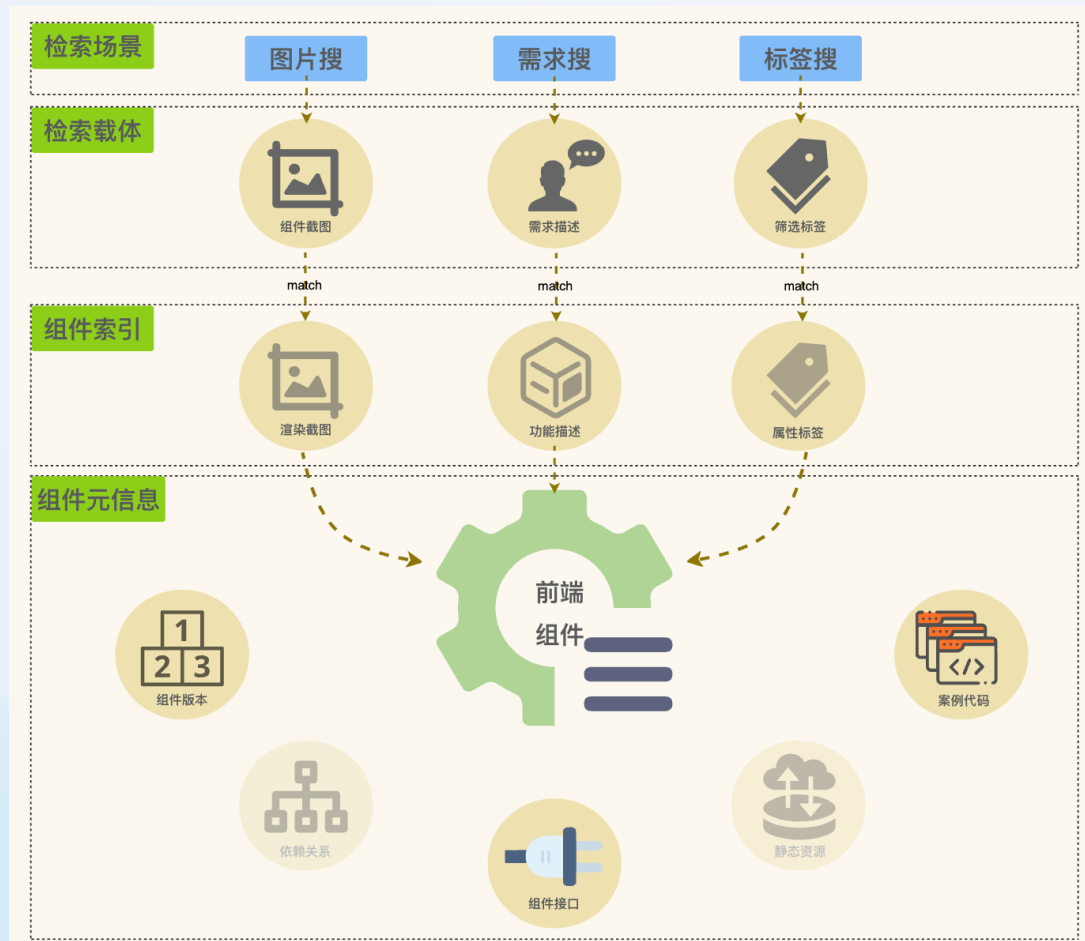


业务组件



复杂组件

前端组件录入



组件信息定义

索引侧：

- 渲染截图：组件代码运行后，可视化的截图结果。
- 功能描述：描述组件的功能、适用场景。由人工或者大模型总结得到。
- 属性标签：版本、组件名称、所属组件库、业务域等标签。

元信息侧：

- 组件版本：组件版本信息。
- 依赖关系：依赖的其他组件/库的信息。
- 组件接口：包含接口名、接口值、接口自然语言描述，描述由人工总结/大模型总结得到。
- 静态资源：组件 cdn 等信息。
- 案例代码：demo代码，展示如何使用该组件。

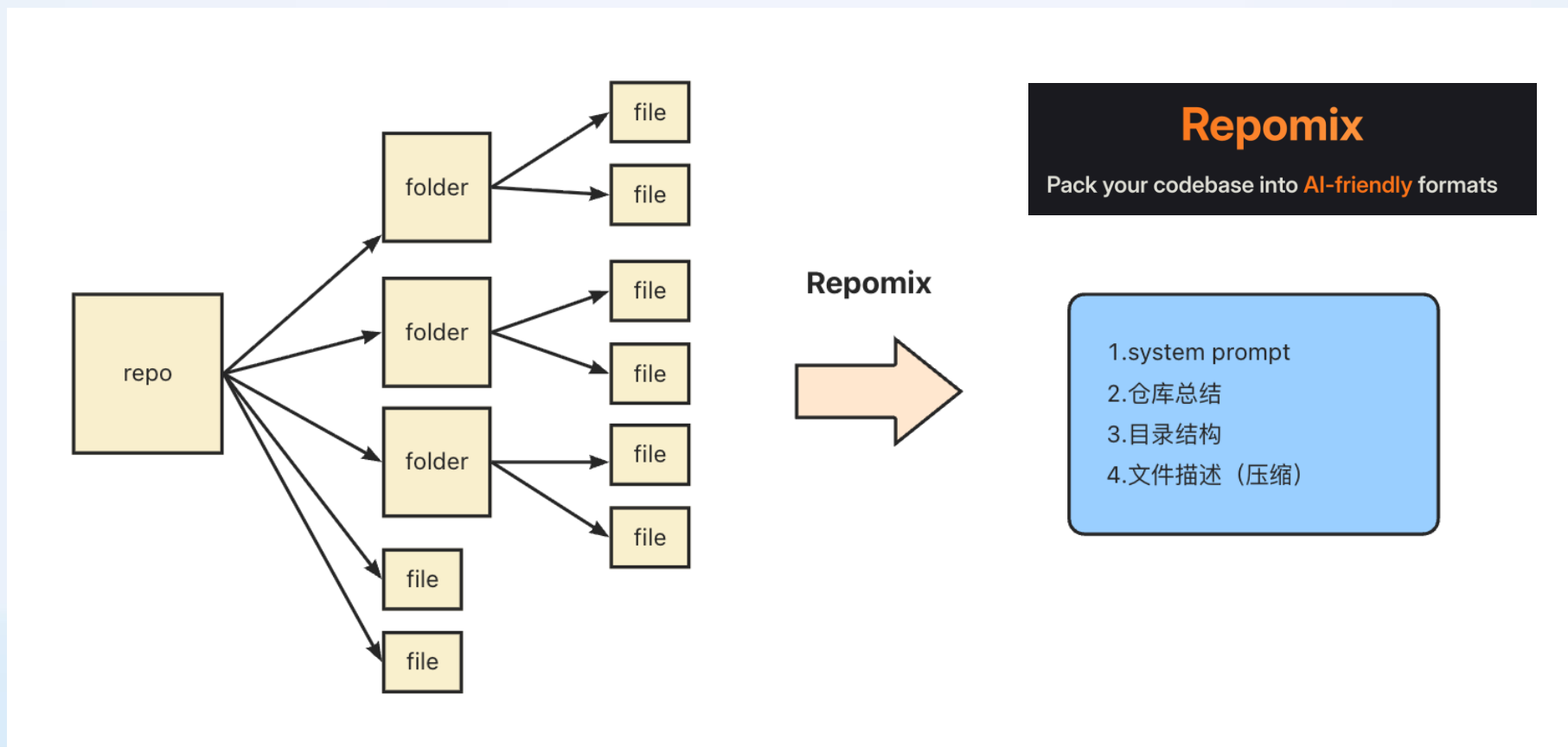
02

仓库描述新范式

智能仓库描述——仓库Wiki



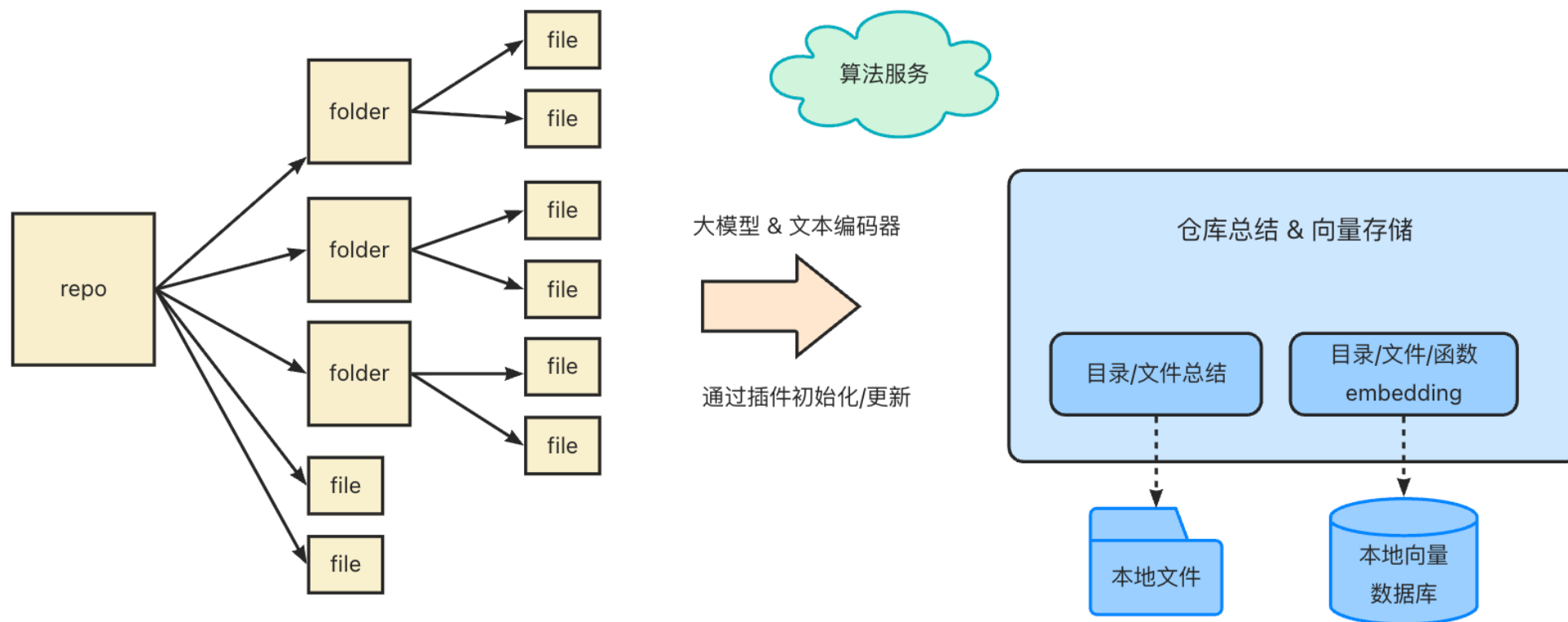
智能仓库描述——AI Repomix



上下文过长,
注意力不集中

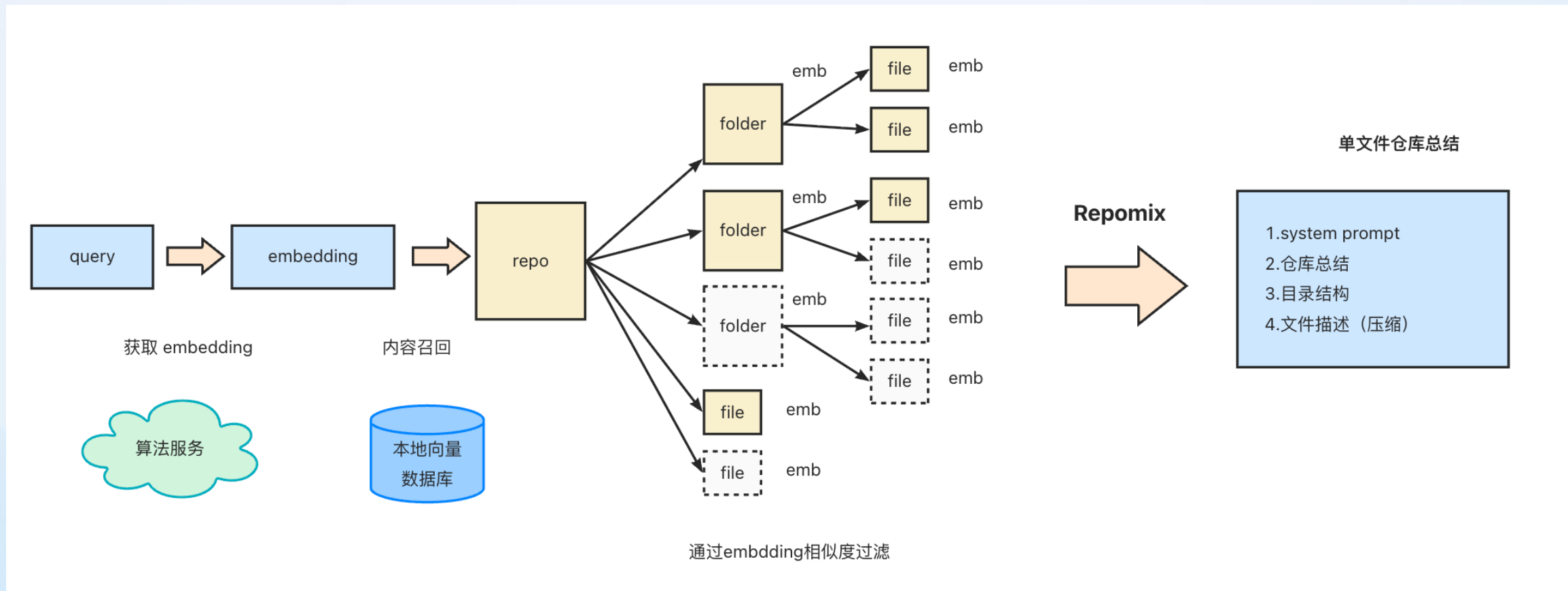
Repomix: 打包描述一个代码仓库

智能仓库描述——AI Repomix



AI Repomix 异步录入

智能仓库描述——AI Repomix



AI Repomix 运行时

端云结合的上下文构造

云端：前端领域知识库

企业级的系统化前端技术体系文档，包括通用框架级别的代码规范、基础知识、最佳实践等。

本地：仓库Wiki

对本地仓库进行大模型总结，描述仓库的目录结构，功能说明等基本信息。

云端：最新仓库依赖信息

基于内外部Context7 和文档等工具，获取指定仓库依赖对应版本的使用说明和示例代码。

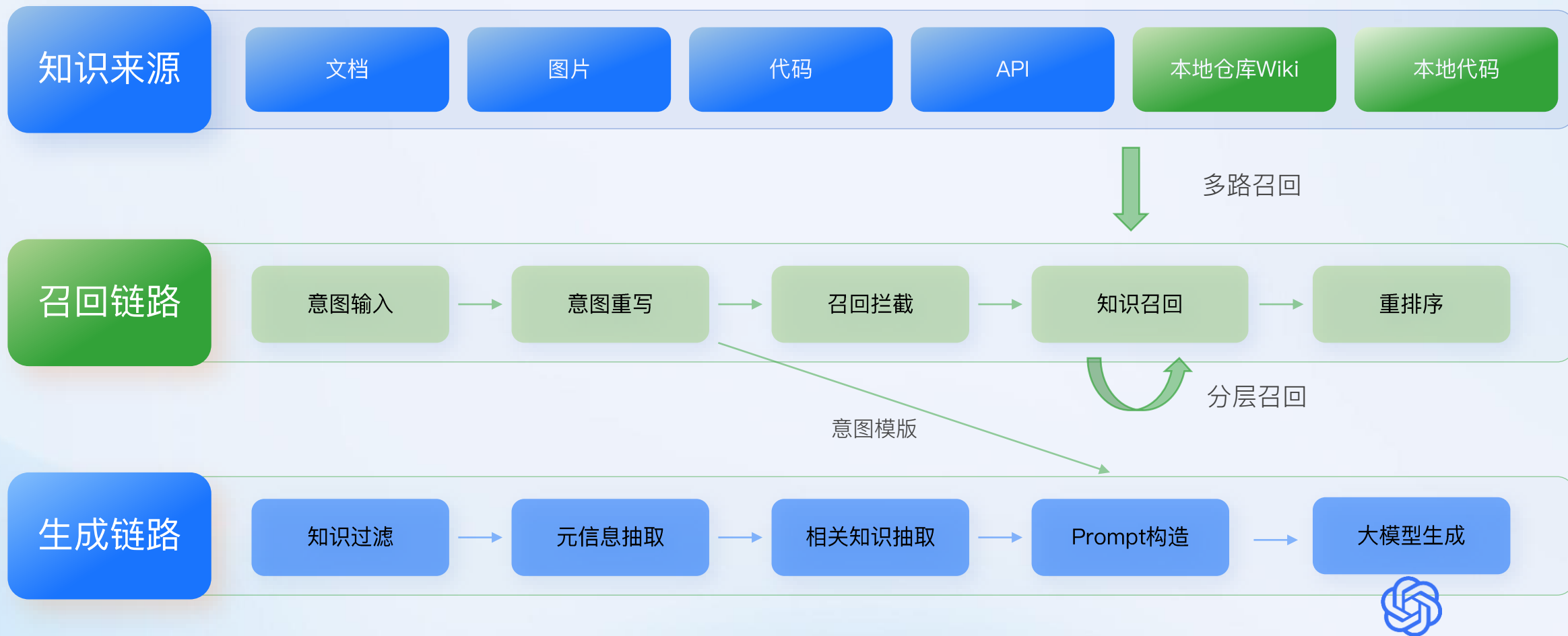
本地：代码文件

本地化的、可潜在复用的代码片段，以及与需求相关的功能代码集合。本地代码文件也间接描述了研发者的代码风格和研发习惯。

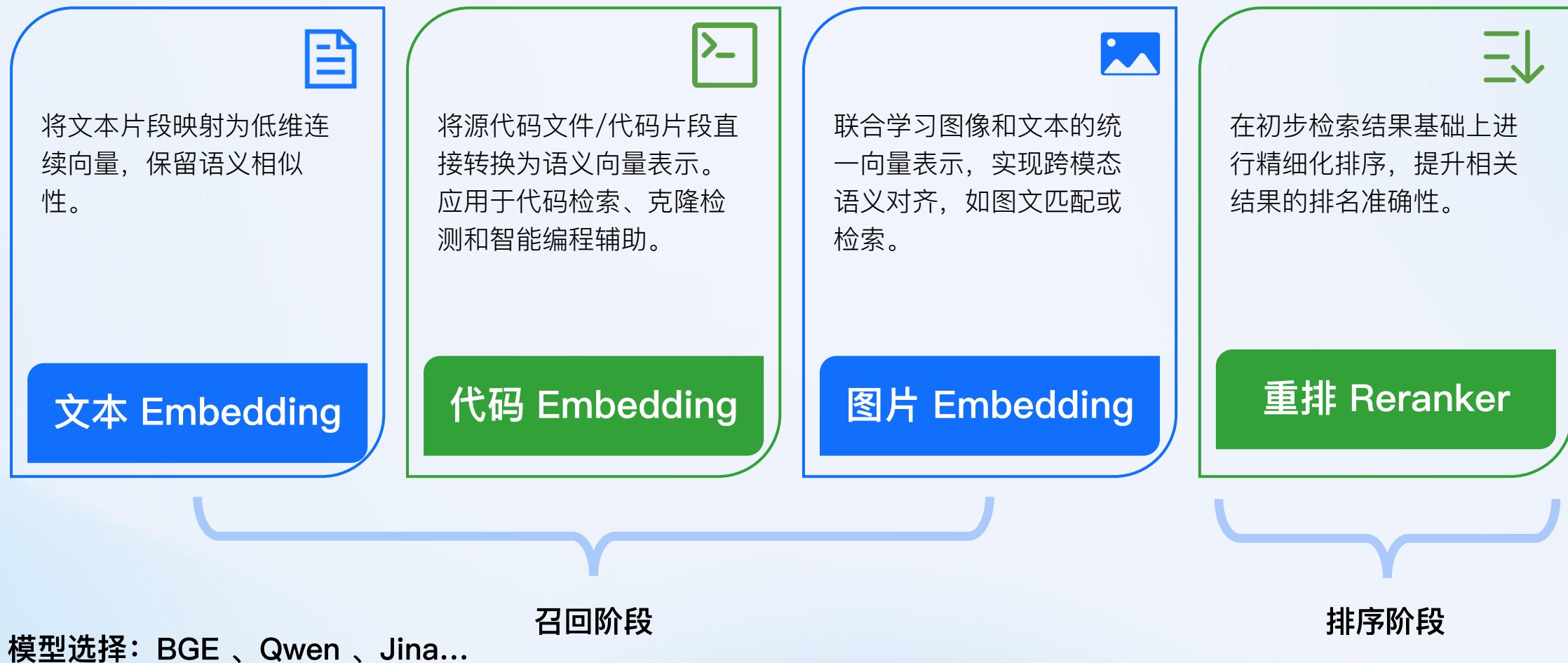
本地研发
上下文依赖

03 智能召回体系

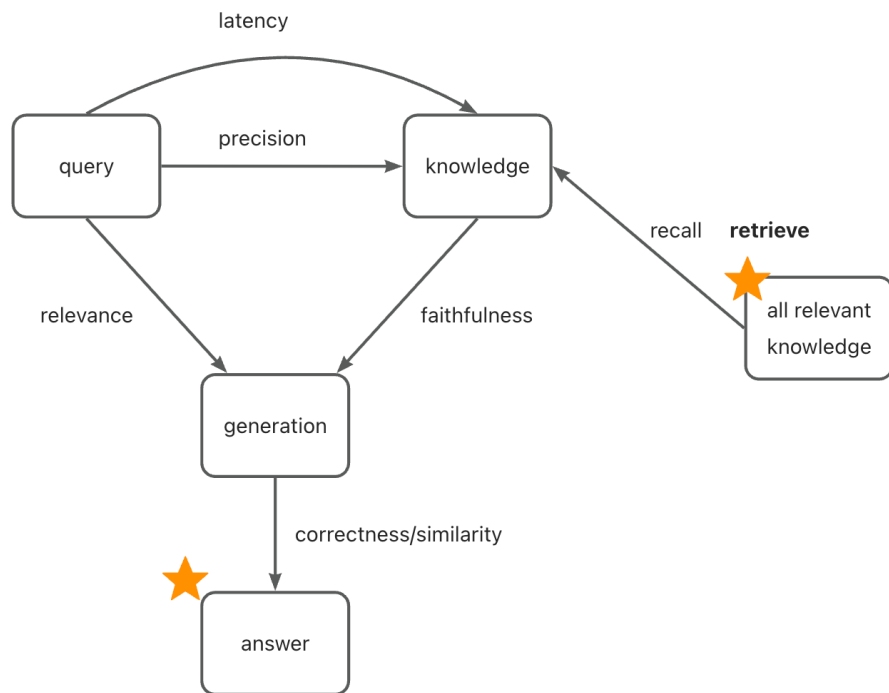
智能召回生成链路



链路中的算法模型



RAG自动化评测体系



基于Ragas框架的评测

评测指标说明

Context Precision: 衡量检索到的上下文中与答案相关的信息占比，反映检索结果的精确性。

Context Recall: 评估检索到的上下文覆盖真实答案所需信息的程度，体现检索的完整性。

Faithfulness: 衡量生成答案在多大程度上忠实于检索到的上下文，避免幻觉或捏造内容。

Factual Correctness: 评估答案在事实层面上的正确性，通常依赖外部知识或标注数据判断。

Answer Relevancy: 衡量生成答案与用户问题的相关程度，反映答案是否切题。

Answer Similarity: 通过与标准答案的语义相似度来评估生成答案的质量，常使用嵌入向量计算。

RAG评估测试

不同的召回策略实验对比，以提升整体效果

	Context Precision	Context Recall	Faithfulness	Factual Correctness	Answer Relevancy	Answer Similarity
纯向量召回	0.6235	0.5959	0.6396	0.2891	0.6680	0.7210
+ Reranker重排	0.7806	0.6334	0.7026	0.3131	0.6886	0.7323
+ 关键词	0.8156	0.6477	0.7238	0.3203	0.7082	0.7373
+ Summary总结	0.8248	0.6703	0.7205	0.3276	0.7161	0.7414
+ 知识图谱	0.8467	0.6704	0.7283	0.3232	0.7333	0.7422

服务MCP化

Agent + MCP



选择是否使用知识库MCP

通过Agent进行意图识别，根据任务难度决策是否要使用知识库MCP服务。

筛选检索知识库范围

根据意图和上下文背景，选择合适的知识来源和需要检索的内容数据。

智能决策最佳参数

通过预设的不同场景知识召回最佳实践，选择提供给知识库服务的最佳参数。

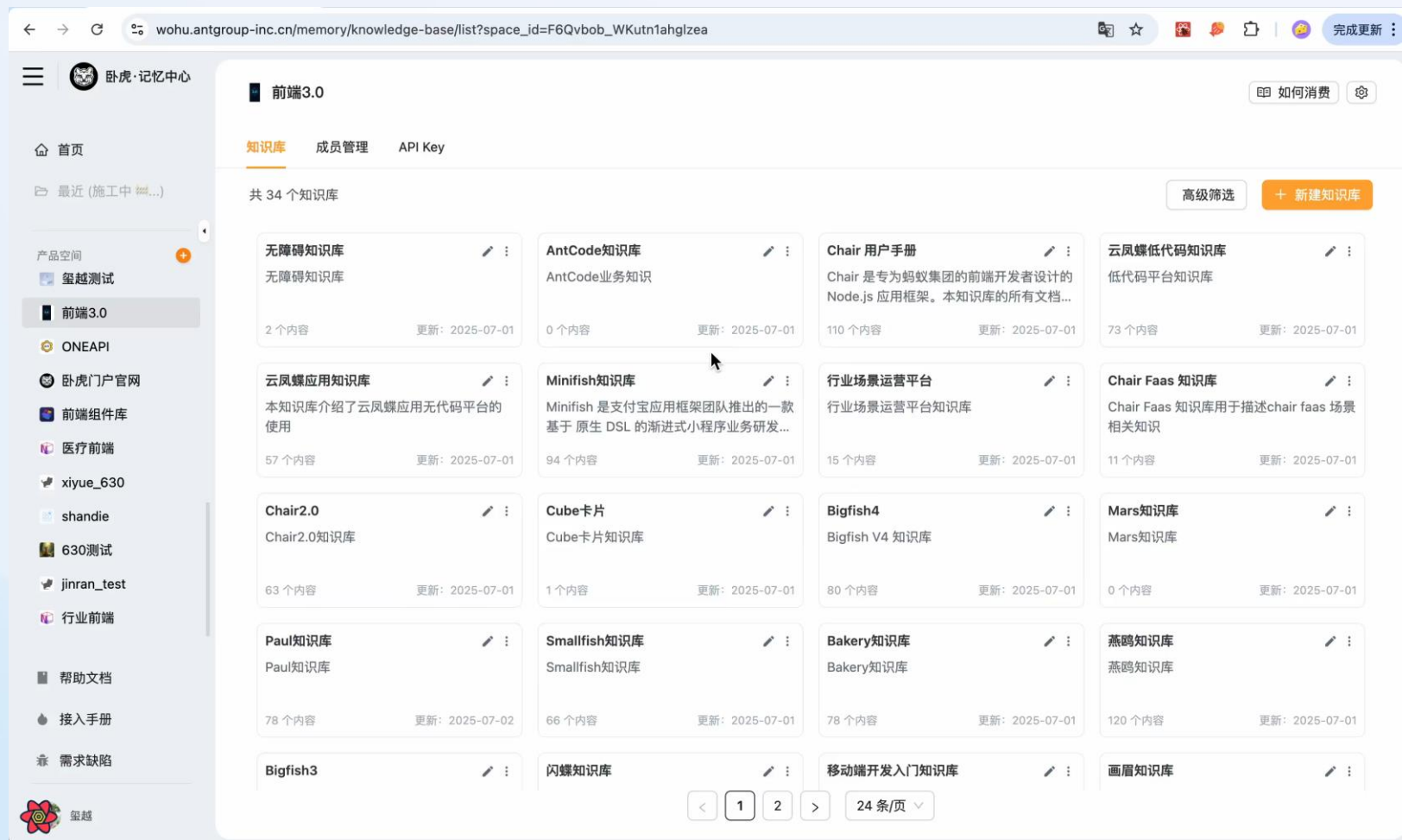
发起MCP调用

发起MCP调用，获取知识库的结果，拼接到Agent的上下文中，进行后续的大模型生成。

04

知识辅助研发案例

辅助问答研发助手



Ragas框架评测

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

人工评测准确率

前端研发平台 82%

前端研发框架 57%
(提升40%)

--- Top3结果包含全量知识内容

多模态知识召回

组件召回



蚂蚁理财社区 App... 2025/7/14 17:06:45

蚂蚁理财社区 App 界面总结

以下是基于提供的参考资料对蚂蚁理财社区 App 界面的总结分析：

一、界面整体布局

1.1 主色调与视觉风格

- 主色调：以白色为主，搭配橙色、蓝色等辅助色，整体风格简洁、清新。
- 图表颜色：涨跌用红色（跌）和绿色（涨）区分，视觉清晰，便于快速识别市场趋势。
- 界面设计：信息层次分明，布局合理，没有过多装饰，突出内容传达。

二、主要功能模块

图文混合案例

19 个内容 | 11.2 MB

该知识库用于展示图文混合的多模态能力

知识库总结 知识库内容 召回信息展示

共召回 4 条知识片段

更新于 2025-07-14 17:04



匹配度 97.46%

更新于 2025-07-11 17:12

查看知识

知识库总结

知识库内容

召回信息展示

共召回 2 条知识片段

组件描述 ##### 1. 组件标识与类型 - **组件名称**: hand-guide - **组件类型**: 提示引导组件 ##### 2. 核心功能 - **主要功能**: 通过小手动画或Lottie动画来引导用户进行特定的操作。 - **解决的问题**: 在用户首次使用应用或需要特别指导的情况下，提供直观的操作提示，帮助用户快速理解和掌握产品的功能。 - **适用场景**: 引导教育类界面，如新手教程、功能介绍等。 ##### 3. 视觉呈现 - **样式特点**: - 支持多种动画效果，包括点击、滑动以及以根据不同的业务需求选择不同的小手样式（产品功能型或活动运营用绝对定位确保小手可以出现在页面的任何位置。 - 动画元素通内动画效果。 - **响应式表现**: - 组件支持不同尺寸屏幕下的显示。 - 通过CSS动画和Lottie动画两种方式提供了良好的用户体验。

查看知识

UI 图片召回



案例：知识辅助前端需求生码

System Prompt

背景

你是一个前端领域专家，熟悉 react 等前端工程开发。正在根据用户问题和一些召回出来规范进行编码工作。

工作流程

- 深度理解用户需求。
- 理解召回出来可能有所帮助的信息。
- 根据用户需求，结合有用的额外信息进行编码。
- 直接输出遵循前端工程 react 编码规范不要有额外信息。

用户输入

{需求query}

组件描述信息

...(召回的知识信息)

可参考的示例demo

...(召回的知识信息)

```
{  
  "component_desc": "组件的基本描述，使用组件的场景",  
  "component_api": "组件属性及事件回调接口",  
  "how_to_install": "安装组件的命令",  
  "how_to_import": "引入组件并注册使用的配置方法",  
  "reference_url": "组件源码的参考链接"  
}
```

```
{  
  "component_demo": "组件使用及行内布局用法 Demo"  
}
```

案例：知识辅助前端需求生码

Query

写一个支持多维筛选的页面，用来过滤景点数据。数据维度包含：城市、星级、门票价格，城市默认值为杭州、星级默认五星，价格默认50元以内。生成mock的景点数据来填充，加一些布局元素来做视觉上的分割，只返回单一tsx代码。

```
import React, { useState, useMemo } from 'react';  
import { TagFilter } from '@alipay/tech-ui';  
import styled from 'styled-components';
```

杭州 上海 北京

5星 4星 3星

50元以内 50-100元 100元以上

西湖

城市: hangzhou

星级: 5星

价格: 30元

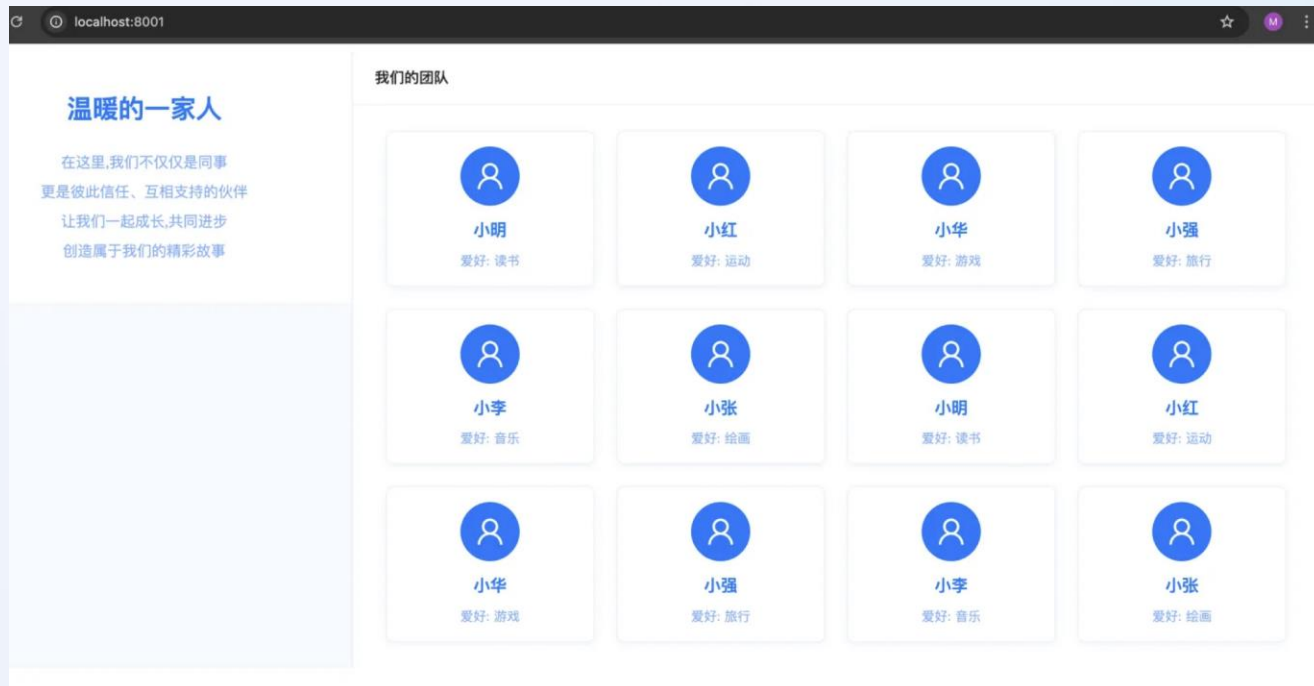
江南好景致

案例：知识辅助前端需求生码

Query

写一个页面，用来展示每位员工的信息。采用左右分栏的布局。左栏写一些温暖的话。右栏是卡片区域，包含员工的头像、姓名、爱好，分三行四列。生成mock的员工信息填充，整体文字样式花哨一点，页面整体风格一致，使用支付宝蓝色。头像使用AntD里的icon填充。

```
Import { ProCard } from '@alipay/tech-ui';  
Import { UserOutlined } from '@ant-design/icons';
```



案例：AI IDE 中的代码研发

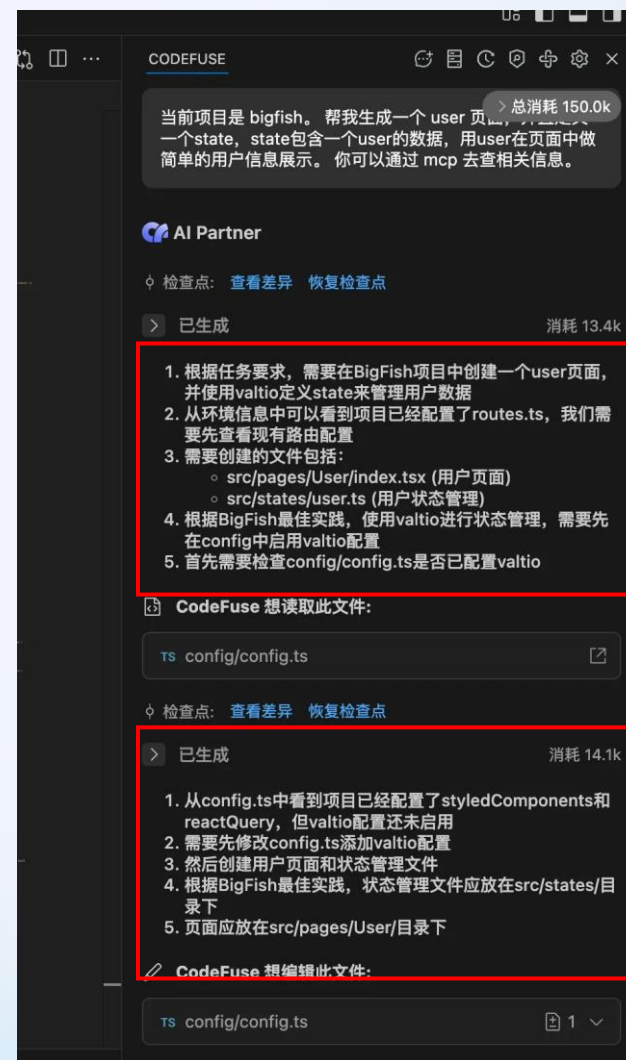
Query

帮我生成一个 user 页面，并且定义一个 state，
state包含一个user的数据，用user在页面中做简单的用户信息展示。

```
CodeFuse Settings  index.tsx 1, U X
src > pages > User > index.tsx > UserPage
1 import React, { useState } from 'react';
2 import { PageContainer } from '@ant-design/pro-components';
3 import { Card, Descriptions } from 'antd';
4
5 const UserPage: React.FC = () => {
6   // 定义用户state
7   const [user, setUser] = useState<API.UserInfo>({
8     id: '1',
9     name: '张三',
10    nickName: '三儿',
11    email: 'zhangsan@example.com',
12    gender: 'MALE'
13  });
14
15  return (
16    <PageContainer header={{ title: '用户信息' }}>
17      <Card title="基本信息">
18        <Descriptions column=1>
19          <Descriptions.Item label="姓名">{user.name}</Descriptions.Item>
20          <Descriptions.Item label="昵称">{user.nickName}</Descriptions.Item>
21          <Descriptions.Item label="邮箱">{user.email}</Descriptions.Item>
22          <Descriptions.Item label="性别">
23            {user.gender === 'MALE' ? '男' : '女'}
24          </Descriptions.Item>
25        </Descriptions>
26      </Card>
27    </PageContainer>
28  );
29 };
30
31 export default UserPage;
32
```

问题：

1. 未理解 Bigfish 项目目录
2. 未在路由中配置
3. 未使用 Valtio



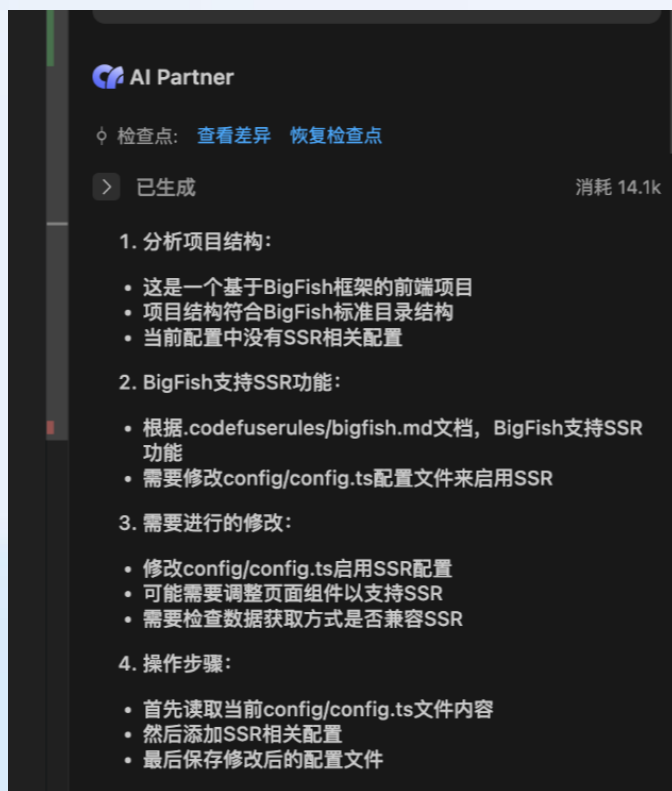
领域代码生成

配置生成

案例：AI IDE 中的代码研发

Query

帮我把这个Bigfish项目改成 SSR。

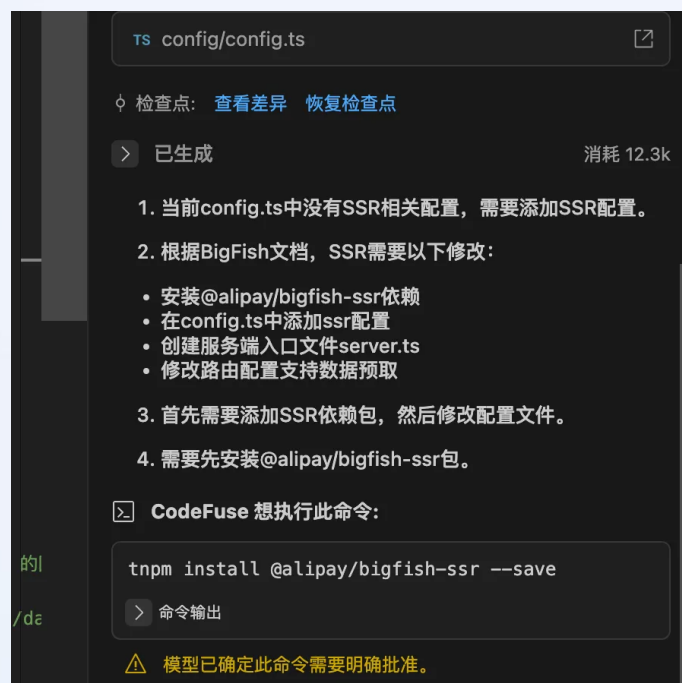


AI Partner

检查点: 查看差异 恢复检查点

> 已生成 消耗 14.1k

1. 分析项目结构:
 - 这是一个基于BigFish框架的前端项目
 - 项目结构符合BigFish标准目录结构
 - 当前配置中没有SSR相关配置
2. BigFish支持SSR功能:
 - 根据.codefuserules/bigfish.md文档, BigFish支持SSR功能
 - 需要修改config/config.ts配置文件来启用SSR
3. 需要进行的修改:
 - 修改config/config.ts启用SSR配置
 - 可能需要调整页面组件以支持SSR
 - 需要检查数据获取方式是否兼容SSR
4. 操作步骤:
 - 首先读取当前config/config.ts文件内容
 - 然后添加SSR相关配置
 - 最后保存修改后的配置文件



TS config/config.ts

检查点: 查看差异 恢复检查点

> 已生成 消耗 12.3k

1. 当前config.ts中没有SSR相关配置, 需要添加SSR配置。
2. 根据BigFish文档, SSR需要以下修改:
 - 安装@alipay/bigfish-ssr依赖
 - 在config.ts中添加ssr配置
 - 创建服务端入口文件server.ts
 - 修改路由配置支持数据预取
3. 首先需要添加SSR依赖包, 然后修改配置文件。
4. 需要先安装@alipay/bigfish-ssr包。

CodeFuse 想执行此命令:

```
tnpm install @alipay/bigfish-ssr --save
```

> 命令输出

⚠ 模型已确定此命令需要明确批准。

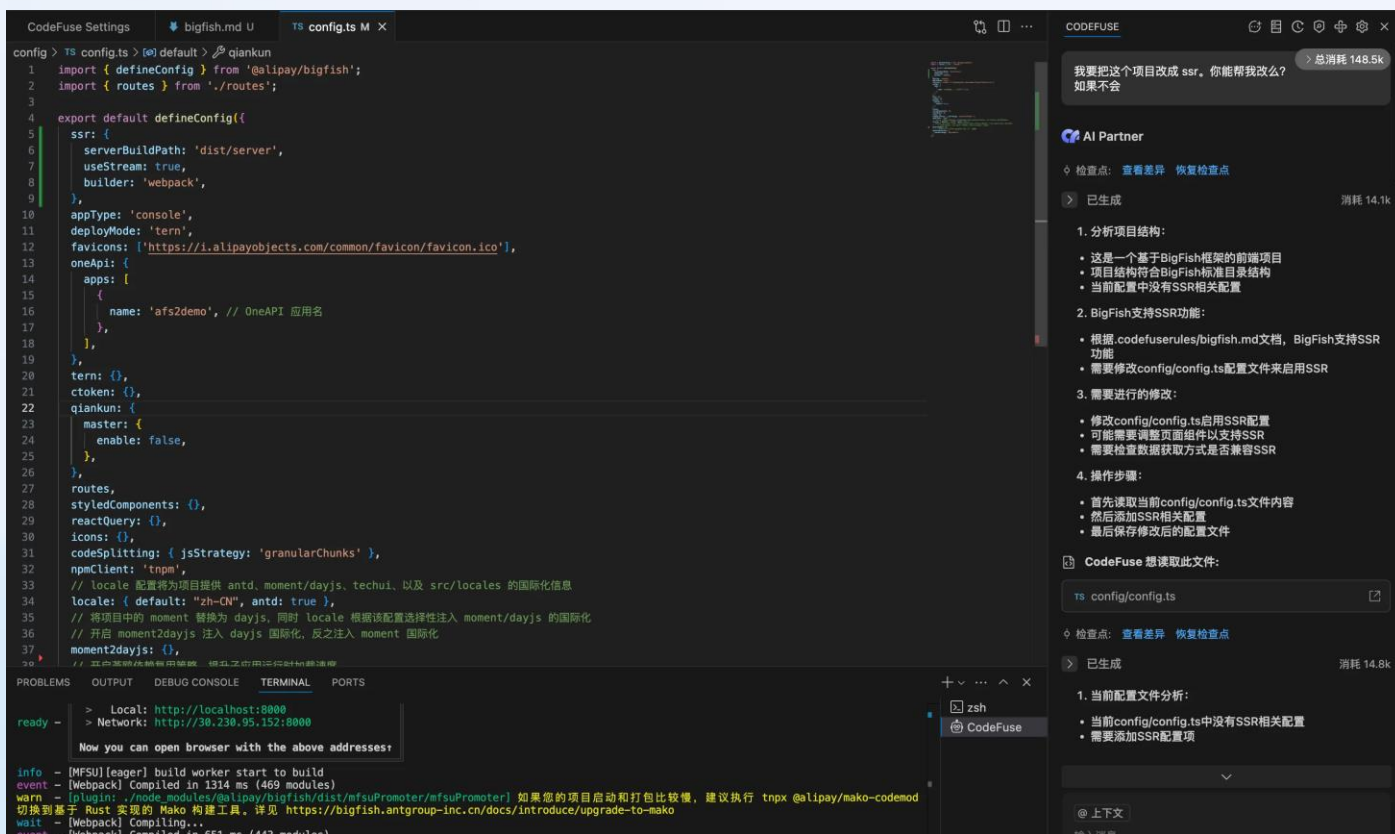
Before:

无法理解 Bigfish、无法知道 Bigfish 的框架信息。解决方案出现了极大的幻觉。

案例：AI IDE 中的代码研发

Query

帮我把这个Bigfish项目改成 SSR。



After:

能正确理解 Bigfish 的 config。知道如何开启 SSR。知道如何关闭微前端。

05

总结与未来展望

总结与未来展望



📖 更丰富的数据来源

🔗 更精细化的知识管理

👍 更准确的知识召回

📍 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

06

Q & A