51CTO WOT

World Of Tech 2024

WOT全球技术创新大会

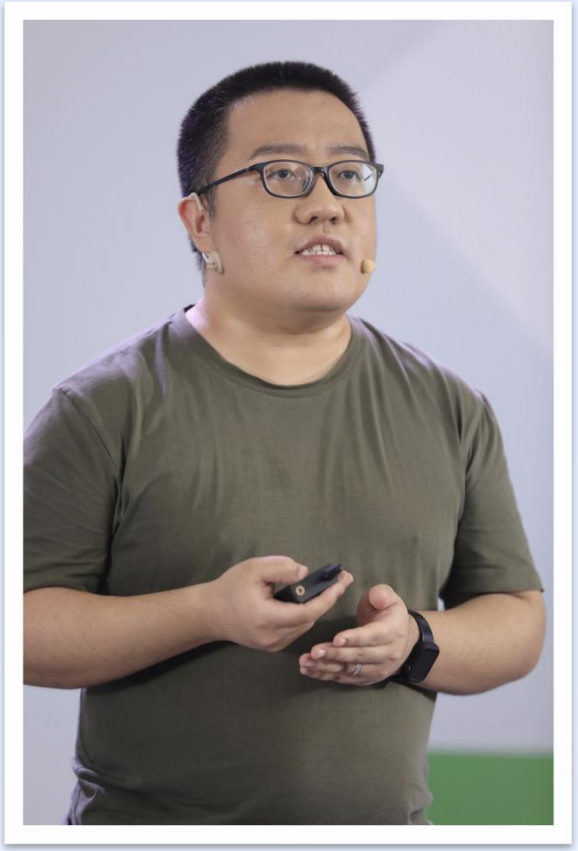
2024

智启新纪
慧创万物

51CTO WOT

去哪儿旅行机票售前客服 Agent&RAG实践

李佳奇
去哪儿旅行 技术总监



李佳奇 去哪儿旅行/机票目的地事业群 技术总监

- 10年机票业务研发工作经验
- 技术中心TC委员/机票TC主席/机票AIGC落地负责人
- 在业务架构/DDD落地/高并发高可用系统设计/技术团队建设等方面有丰富的经验
- 多次参与各种公司内外技术分享和行业大会分享

目录

- 背景介绍
- 技术方案
- 成果总结

背景介绍

背景介绍

老板焦虑

是否会出现降维打击
如何继续保持领先
如何尽快拿出产品

团队迷茫

技术储备是否足够
是否会被AI代替
该如何顺应趋势

用户期待

杀手级应用何时出现
体验极大提升
智能效果提升

AIGC 时代来临我们该如何应对

我们的应对方法

打基建

开发基建、测试基建、运行基建

找机会

业务*AI矩阵

建团队

AI周会/培训/项目练兵

目前AIGC在去哪儿机票的落地现状

技术基建

01

Langchain4J Qunar框架

RAG

prompt/智能体平台

团队建设

02

覆盖团队20%成员的AIGC技术储备

AI周会/AI圆桌会

技术分享/技术交流

项目落地

03

业务侧：用户体验提升/收益提升/智能营销/归因分析

技术侧：工单自动化/SQL自动生成/checklist智能生成

售前客服项目介绍

项目背景

售前客服覆盖缺失

基于用户主流程的导购可行性验证

用户问卷调研需求

技术准备

RAG/Embedding基建准备

Agent设计落地技术准备

技术人员培养

达成路径

mvp版本入口小流量验证

封闭式问题/开放式问题

开源大模型/商用大模型

技术方案

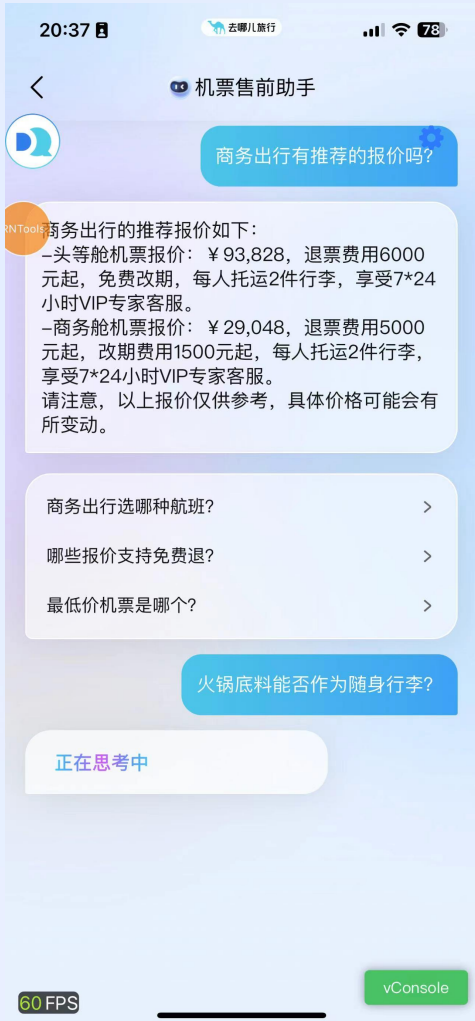
【开放式回答】初始版

静态/动态问题回答

【封闭式回答】文字版

问题分类

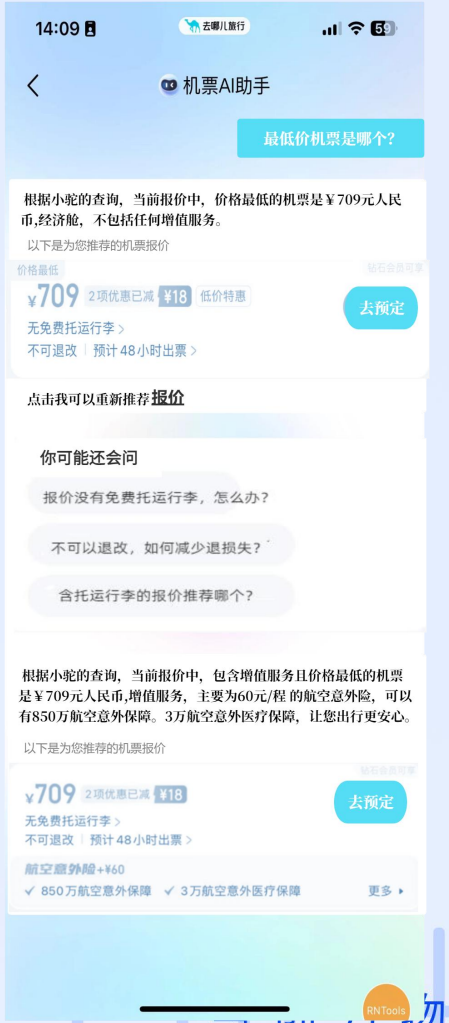
问题推荐



【封闭式回答】图文版

报价推荐

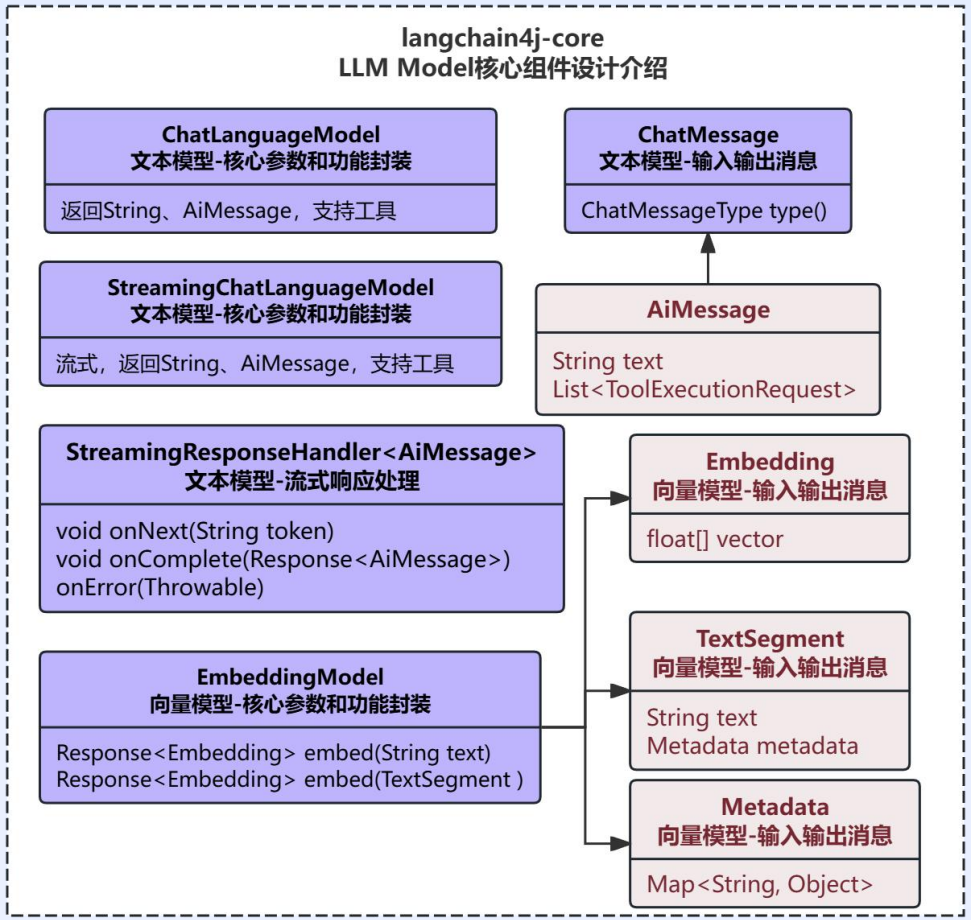
图文消息



51CTO WOT 技术方案

Models
- GPT3.5/4/4v DALL·E2/3
- Gemini
- 文心一言
- 通义千问
- ……

Model本质封装了和LLM供应商的HTTP API，文本、向量、图片等不同类型，API不同，封装用到的对象也不同。



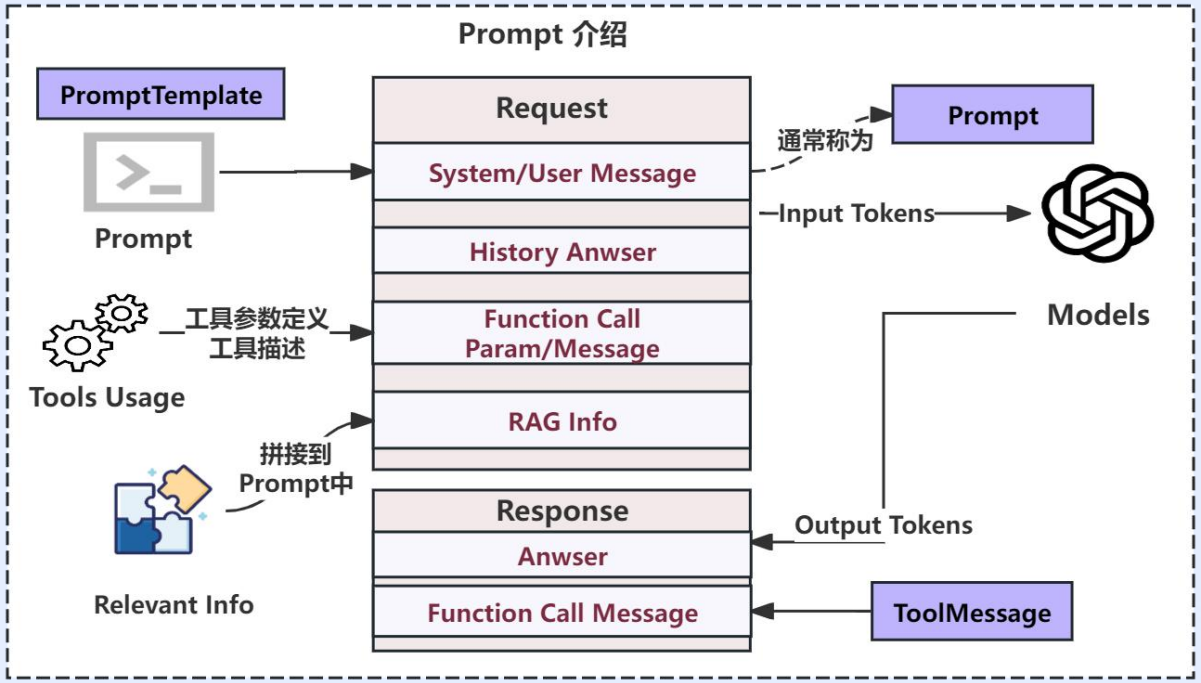
售前助手-代码示例

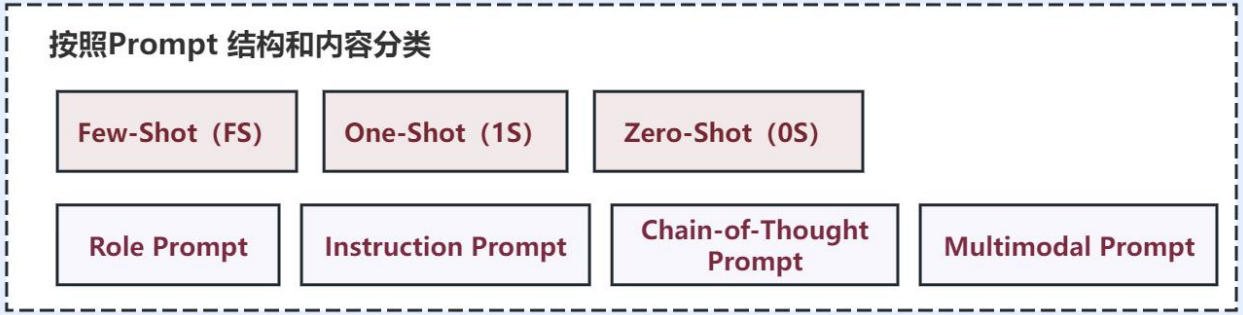
```
/**
 * 开源版本 <br/>
 * 智谱4 <br/>
 * 模型：glm-4<br/>
 */
[redacted]
@Bean
public ZhipuAiStreamingChatModel zhipuAiChatModel4(@Value("${glm4.api.key}") String apiKey) {
    return ZhipuAiStreamingChatModel.builder().model("glm-4").logRequests(true).logResponses(true).apiKey(apiKey)
        .build();
}
```

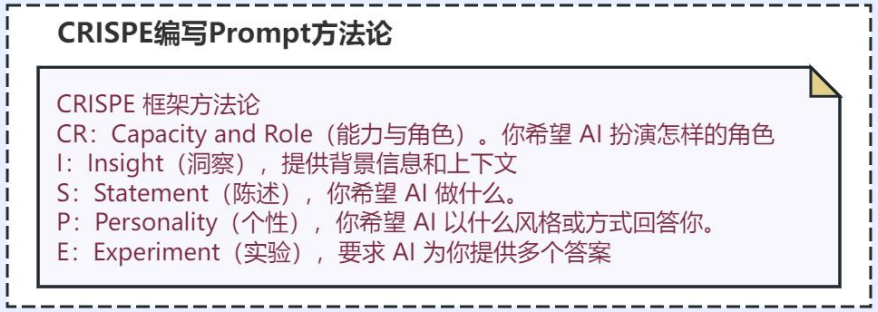
```
/**
 * 去哪儿算法组版本 <br/>
 * 算法组封装的llmModel，远程Http调用算法组封装的sdk接口，算法组实现了llmModel的统一的账号管理、风控拦截、llm供应商对接等。<br/>
 * 模型：gpt-4-turbo <br/>
 * 问题推荐Agent使用，输入和输出比较少，和回答Agent并行执行
 */
[redacted]1 *
@Bean
public QunarAiChatModel qunarAiChatModelGpt4(@Value("${gpt.api.key}") String key,
    @Value("${gpt.api.password}") String password) {
    DefaultQunarAiClient client = DefaultQunarAiClient.builder(key, password).logLevel(ClientLogLevel.ALL).build();
    ClientRequiredConfig requiredConfig =
        ClientRequiredConfig.builder().project("presale_ai_assist").password(password)
            .userIdentityInfo("fanmao.meng").build();
    ClientOptionalConfig optionalConfig = new ClientOptionalConfig();
    optionalConfig.setTopP(0);
    optionalConfig.setChatApiType(ChatComplexRequest.ApiType.GPT_4_TURBO);
    optionalConfig.setChatApiVersion(ChatComplexRequest.ApiVersion.V2023_12_01_PREVIEW);
    return QunarAiChatModel.builder().client(client).requiredConfig(requiredConfig).optionalConfig(optionalConfig)
        .build();
}
```

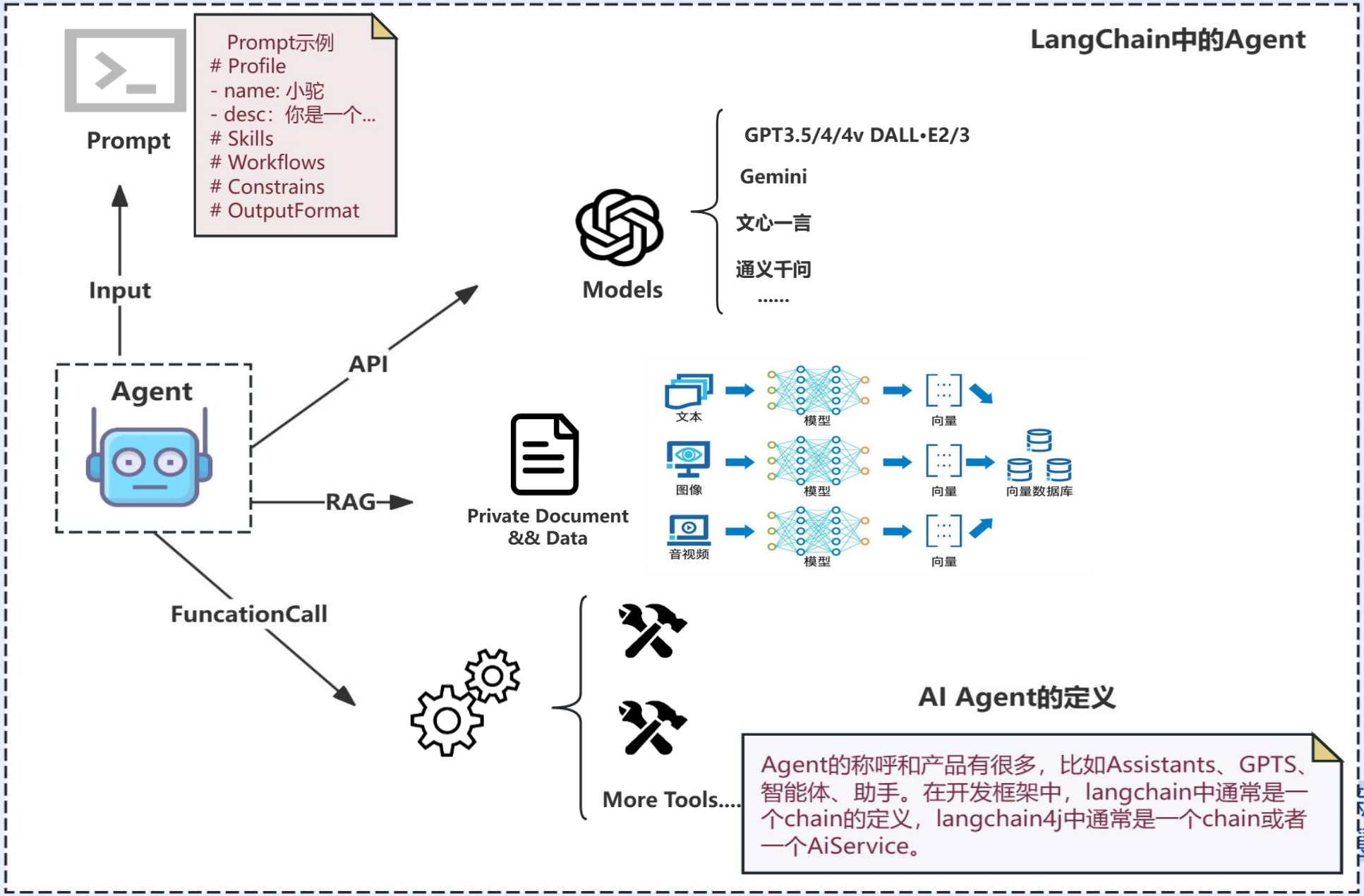
Prompt

Prompt示例
Profile
- name: 小驼
- desc：你是一个...
Skills
Workflows
Constrains
OutputFormat



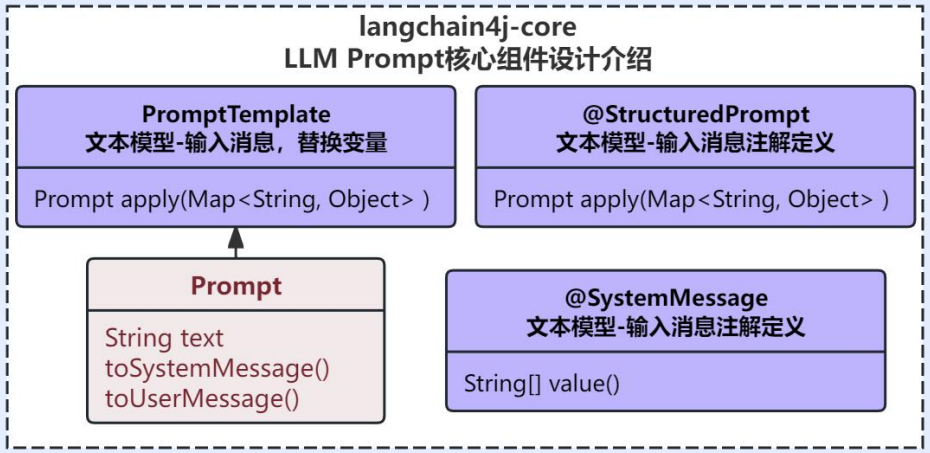





LangChain中的Agent
Prompt
Prompt示例
Profile
- name: 小驼
- desc：你是一个...
Skills
Workflows
Constrains
OutputFormat
Input
Agent
API
RAG
FuncationCall
Models
GPT3.5/4/4v DALL•E2/3
Gemini
文心一言
通义千问
.....
Private Document
&& Data
文本
图像
音视频
模型
向量
向量数据库
More Tools....
AI Agent的定义
Agent的称呼和产品有很多，比如Assistants、GPTS、智能体、助手。在开发框架中，langchain中通常是一个chain的定义，langchain4j中通常是一个chain或者一个AiService。

Prompt

Prompt示例
Profile
- name: 小驼
- desc：你是一个...
Skills
Workflows
Constrains
OutputFormat



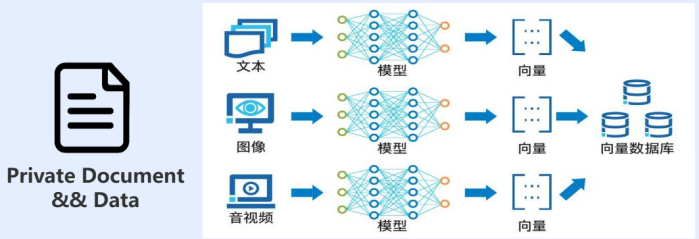
售前助手-代码示例

Prompt定义

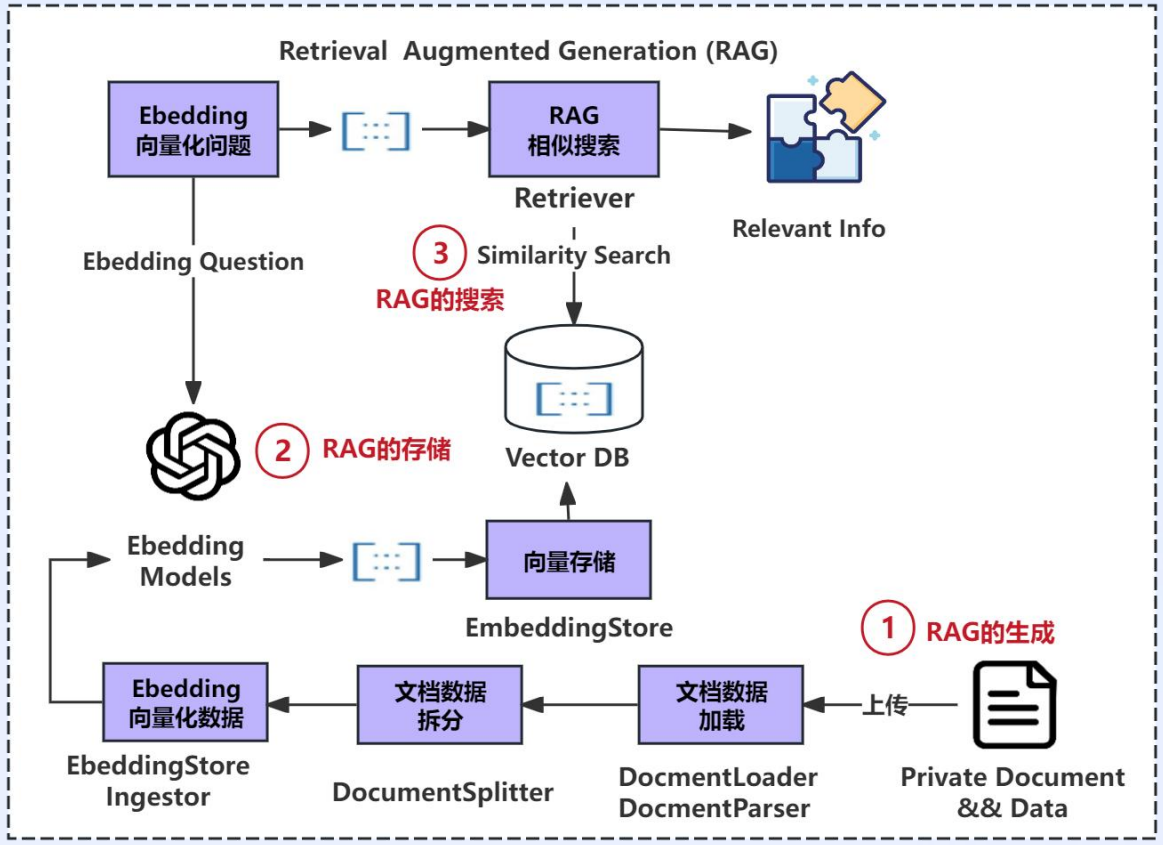
```
no usages  new *
@SystemMessage("""
        # Profile
        你是忠实的去哪儿网机票用户，你会收到一个问题，你的工作不是回答收到的问题，而是要思考，作为一个机票用户，问了这个问题之后，你还会问哪些问题
        # Skills
        ## 接下来要问的问题
        1. 你能够准确的理解给出的问题
        2. 你不会直接回答这个问题，而是从给出的**参考问题列表**中选择3个你接下来要问的问题
        3. 将你接下来要提问的问题整理成列表
        # Workflows
        1. 第一步：仔细理解给出的问题
        2. 第二步：接下来要问的问题
        3. 第三步：从第二步得到的问题列表中，挑选最多3个你接下来最想了解的问题，并给出
        4. 第四步：仔细思考，上一步得到的问题都是你接下来最想了解的问题吗
        5. 第五步：请注意：问题列表中的所有问题必须包含在**参考问题列表**中，如果有不存在的问题，请去掉该问题
        # Initializ
        **参考问题列表**: {{frequentlyAskQuestion}}
        # Constrains
        - 你不能直接回答给出的问题
        - 请严格参照回答格式进行回答，不要添加任何无关的信息
        - 最终得出的问题列表允许为空集合，但不允许包含与原始问题相同的问题
        - 最终得到的问题必须包含在**参考问题列表**中
        # OutputFormat
        你的回答必须是Json格式的问题列表，参考结构如下：
        {"questions":<问题列表>}
    """)
Response<AiMessage> questionGenerateX(@UserMessage String userMessage, @V("frequentlyAskQuestion") String frequentlyAsk,
    @UserName String username);
```

Prompt变量替换、转为请求LLM参数

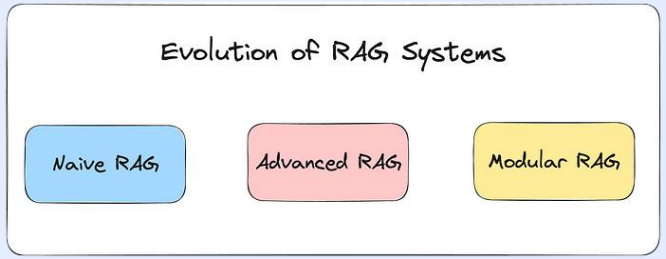
```
protected Optional<dev.langchain4j.data.message.SystemMessage> prepareSystemMessage(Method method, Object[] args) {

    Parameter[] parameters = method.getParameters();
    Map<String, Object> variables = getPromptTemplateVariables(args, parameters);

    dev.langchain4j.service.SystemMessage annotation = method.getAnnotation(dev.langchain4j.service.SystemMessage.class);
    if (annotation != null) {

        String systemMessageTemplate = getPromptText(
                method,
                type: "System",
                annotation.fromResource(),
                annotation.value(),
                annotation.delimiter()
        );

        Prompt prompt = PromptTemplate.from(systemMessageTemplate).apply(variables);
        return Optional.of(prompt.toSystemMessage());
    }

    return Optional.empty();
}
```

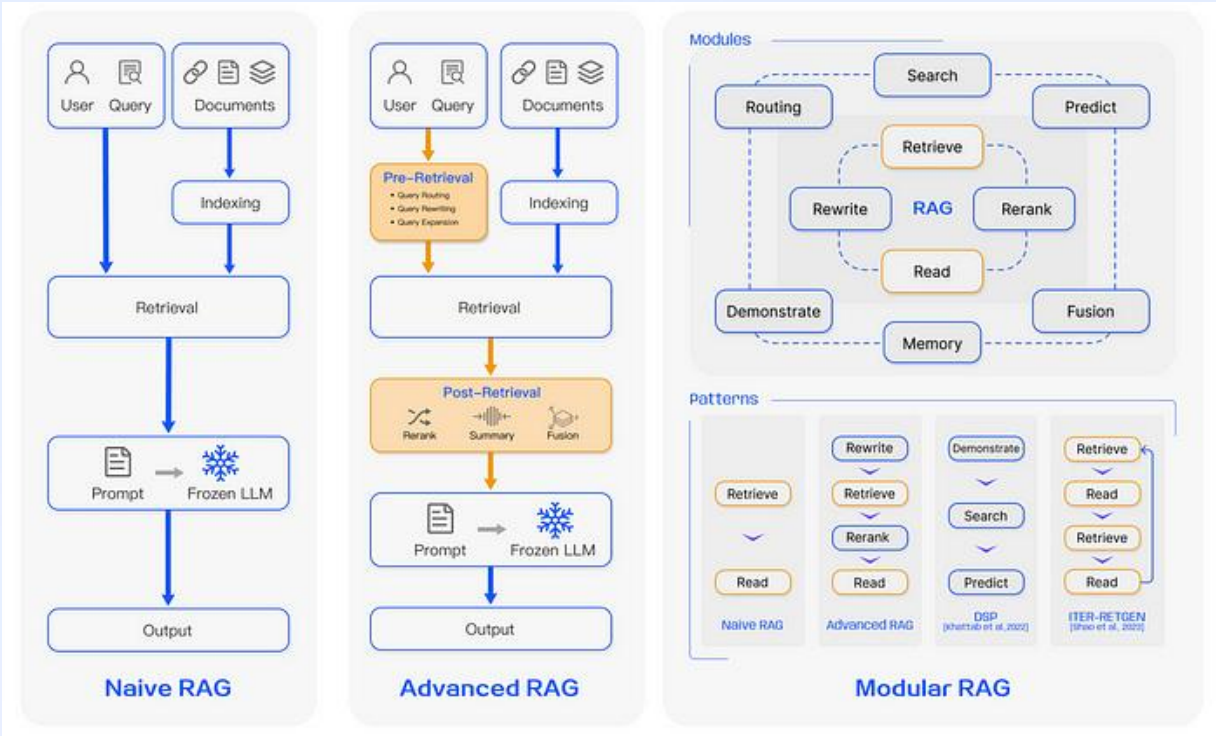


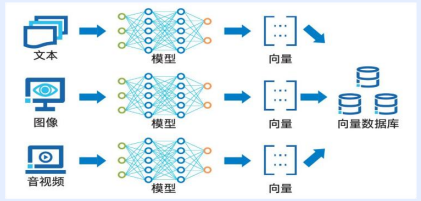
RAG生成和查询流程介绍

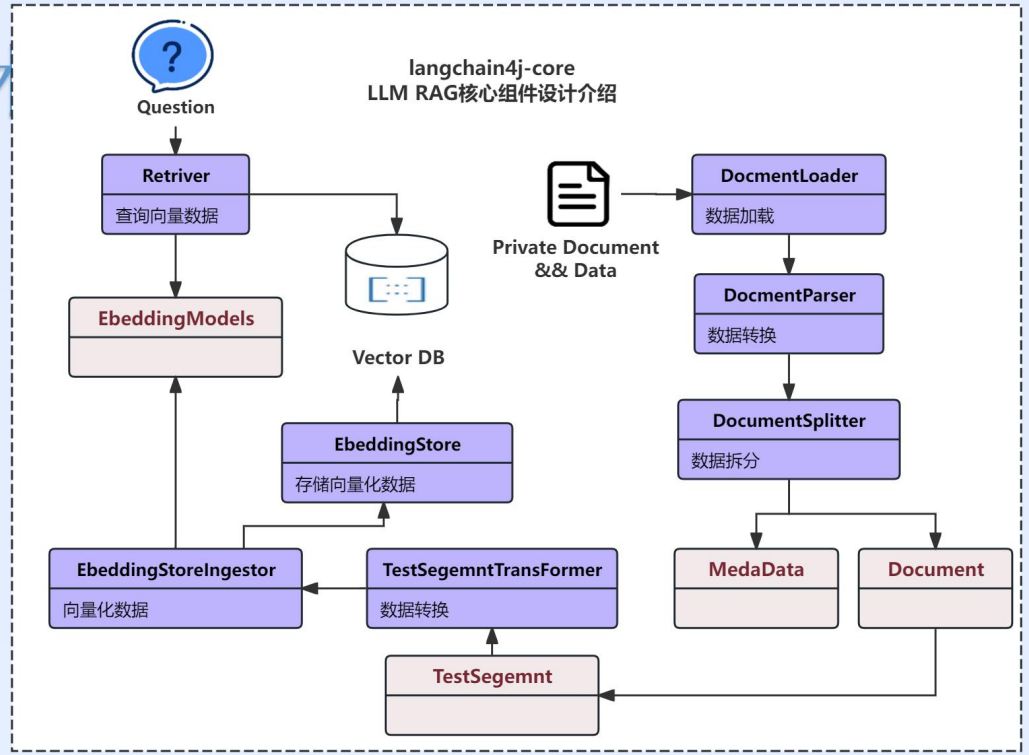


三种 RAG 范式介绍









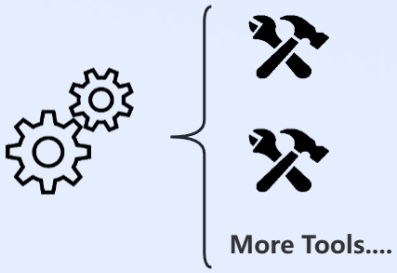

```
@Slf4j
public class QAPairTransform implements TextSegmentTransformer {

    no usages
    public static final QAPairTransform INSTANCE = new QAPairTransform();


    [redacted]
    @Override
    public TextSegment transform(TextSegment segment) {

        String text = segment.text();
        String[] qaParis = text.split( regex: "##");
        if (qaParis.length != 2) {
            log.info("解析问题对失败，text:{}", text);
            return segment;
        } else {
            return TextSegment.from(qaParis[0], segment.metadata());
        }
    }

}
```

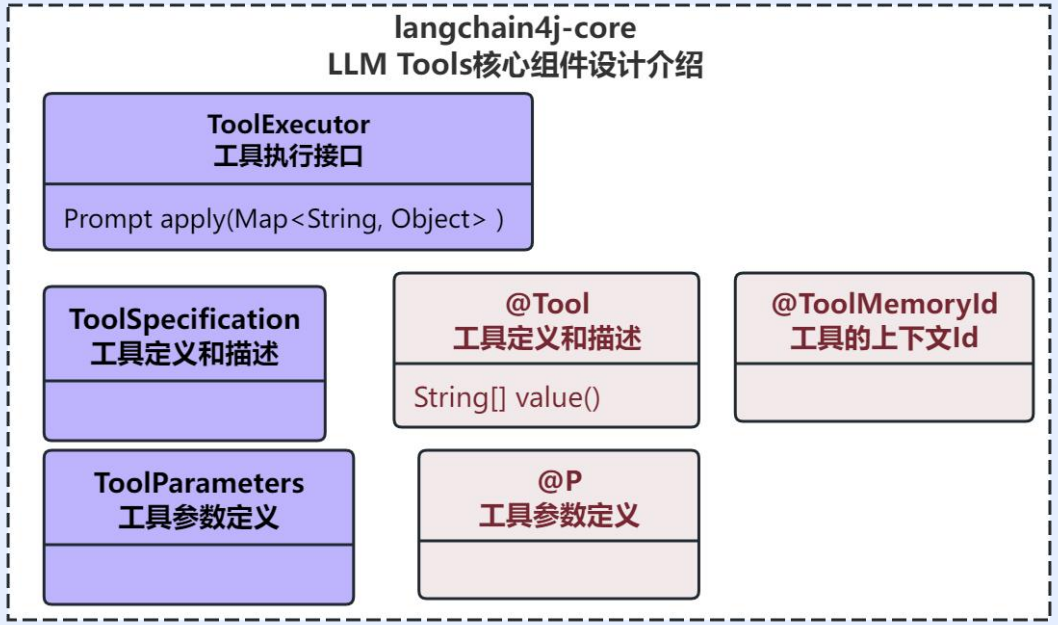
售前助手-代码示例

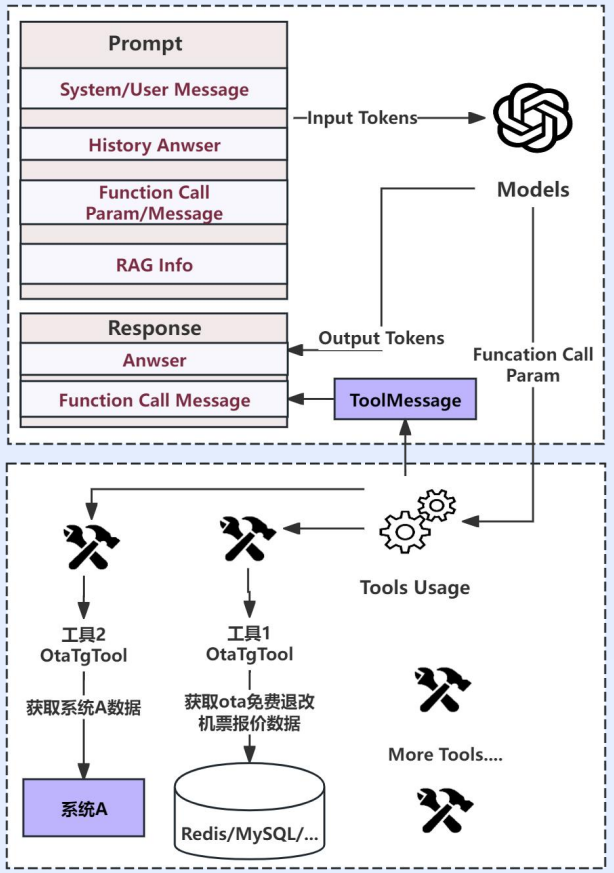
```
防晒隔离水能否带上飞机？##meta:{"category":"规则类,化妆品"}####
折叠式婴儿车能否进客舱？##meta:{"category":"规则类,儿童物品"}####
能否带十几条烟上飞机？##meta:{"category":"规则类,烟酒"}####
```

```
public class QuestionDocumentSplitter extends DocumentByHashTagSplitter {
    [redacted]
    protected TextSegment createSegment(String text, Document document, int index) {
        String[] elements = text.split( regex: "##meta:");
        String questionStr = elements[0];
        Map<String, String> metadataMap = Maps.newHashMap();
        metadataMap.putAll(document.metadata().asMap());
        return TextSegment.from(questionStr, Metadata.from(metadataMap));
    }
```

```
/**
 * 去哪儿封装的EmbeddingModel，远程Http调用算法组接口生成向量，算法组底层使用自有的模型
 */
[redacted]
@Bean
public EmbeddingModel embeddingModelQunar() {

    String key = modelAccountConfigProcess.getKey();
    String password = modelAccountConfigProcess.getPassword();
    String userIdentityInfo = modelAccountConfigProcess.getUserIdentityInfo();

    DefaultQunarAiClient client = DefaultQunarAiClient
            .builder(key, password)
            .build();
    ClientRequiredConfig requiredConfig = ClientRequiredConfig.builder()
            .project("presale_ai_assist")
            .password(password)
            .userIdentityInfo(userIdentityInfo).build();
    ClientOptionalConfig optionalConfig = new ClientOptionalConfig();
    optionalConfig.setEmbeddingApiType(modelAccountConfigProcess.getEmbeddingApiType());
    return QunarEmbeddingModel.builder().client(client).requiredConfig(requiredConfig).optionalConfig(optionalConfig).build();
}
```

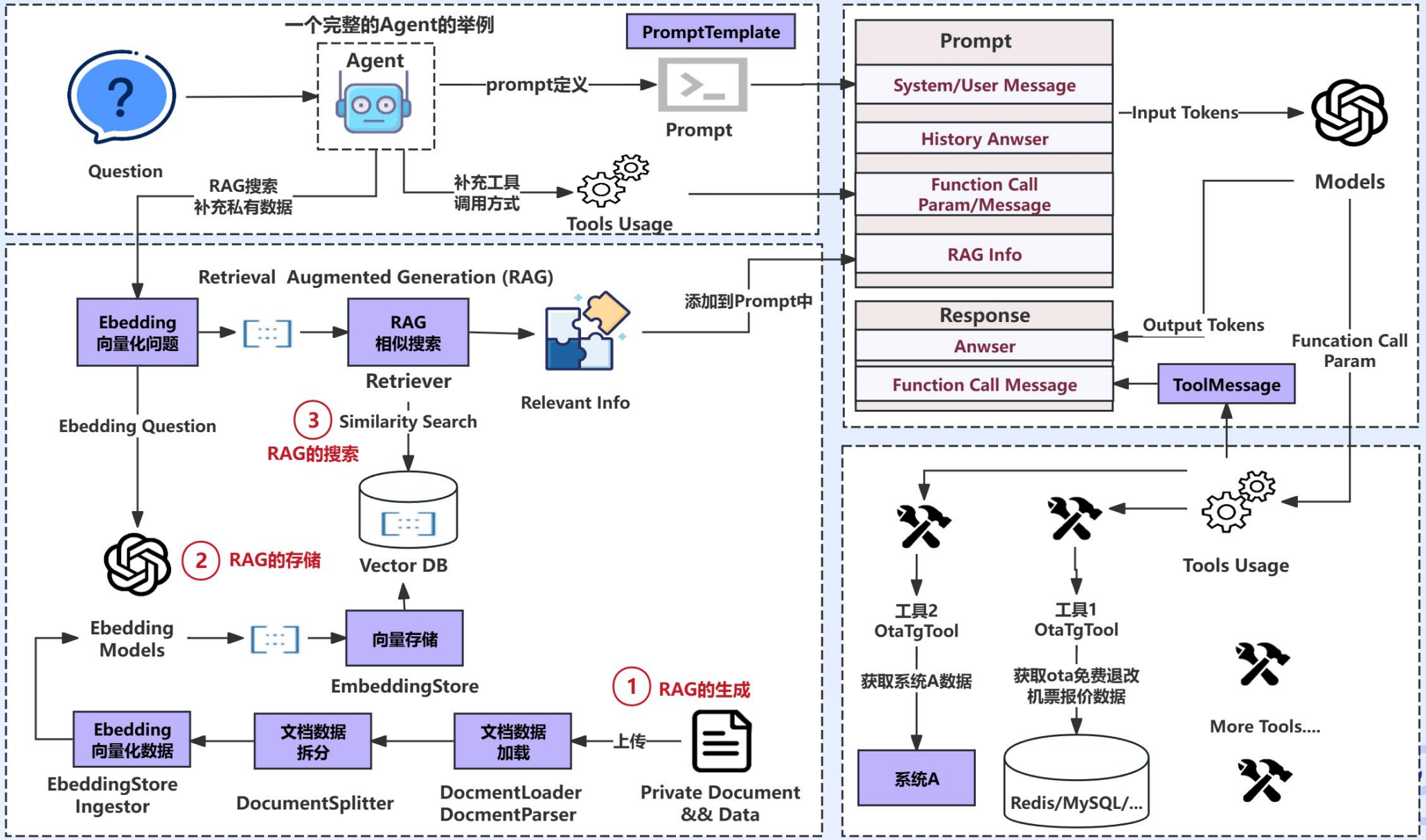






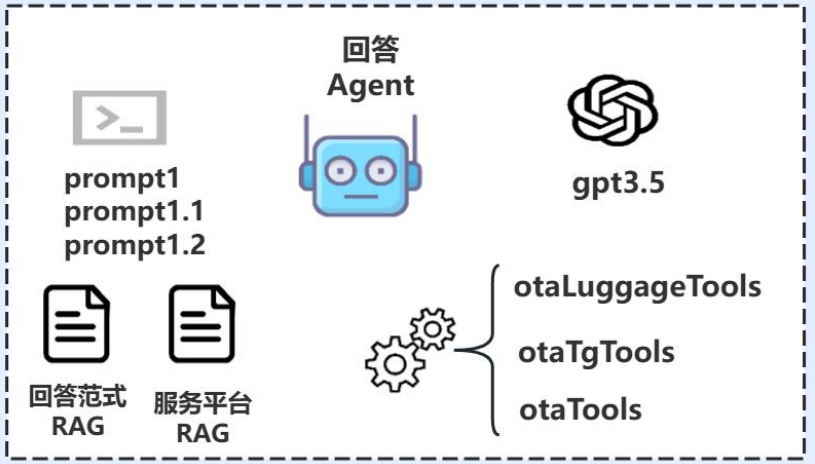
售前助手-代码示例

```
@Tool(name = "getBusinessCabinProduct", value = "可以通过这个工具查询商务舱、公务舱、头等舱的机票产品信息，返回的报价都是商务舱、公务舱、头等舱")
private String getBusinessCabinProduct(@P("ota报价缓存Key") String otaCacheKey) {
    log.info("商务舱、公务舱、头等舱报价查询工具，key:{}", otaCacheKey);
    String result = interRedisService.getKey(otaCacheKey);
    if (StringUtils.isNotEmpty(result)) {
        OtaData otaData = JsonUtil.fromJson(result, OtaData.class);
        return getBusinessCabinProduct(otaData);
    }
    return "查询结果为空，请引导用户购买别的机票产品";
}
```

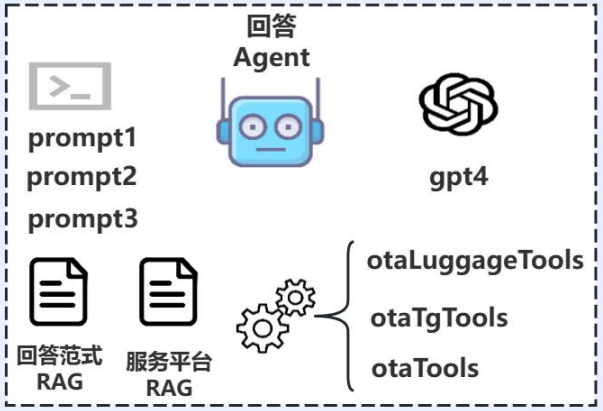

一个完整的Agent的举例
PromptTemplate
Agent
prompt定义
Prompt
Question
RAG搜索
补充私有数据
补充工具
调用方式
Tools Usage
Retrieval Augmented Generation (RAG)
Ebedding
向量化问题
RAG
相似搜索
Retriever
Relevant Info
添加到Prompt中
Ebedding Question
3 Similarity Search
RAG的搜索
Vector DB
2 RAG的存储
Ebedding
Models
向量存储
EmbeddingStore
1 RAG的生成
Ebedding
向量化数据
文档数据
拆分
文档数据
加载
上传
EbeddingStore
Ingestor
DocumentSplitter
DocmentLoader
DocmentParser
Private Document
&& Data
Prompt
System/User Message
History Anwser
Function Call
Param/Message
RAG Info
Input Tokens
Models
Response
Anwser
Function Call Message
Output Tokens
Funcation Call
Param
ToolMessage
Tools Usage
工具2
OtaTgTool
获取系统A数据
工具1
OtaTgTool
获取ota免费退改
机票报价数据
More Tools....
系统A
Redis/MySQL/...

V1.0版 单Agent协作 一个上下文

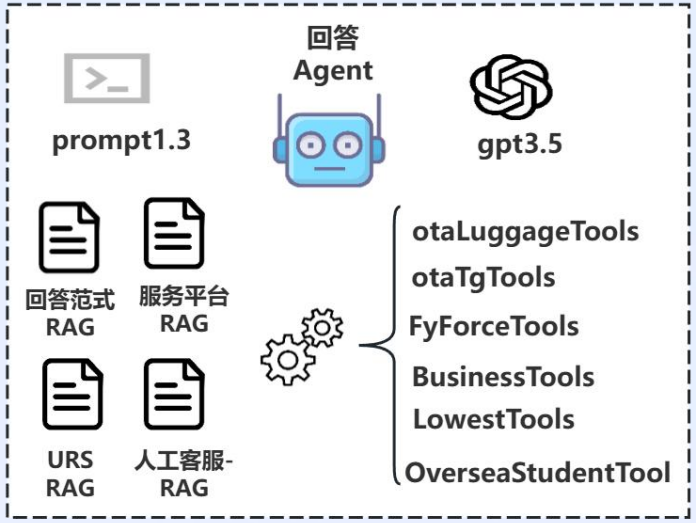
V1.1版 单Agent协作 扩展tools/优化prompt1/扩展RAG



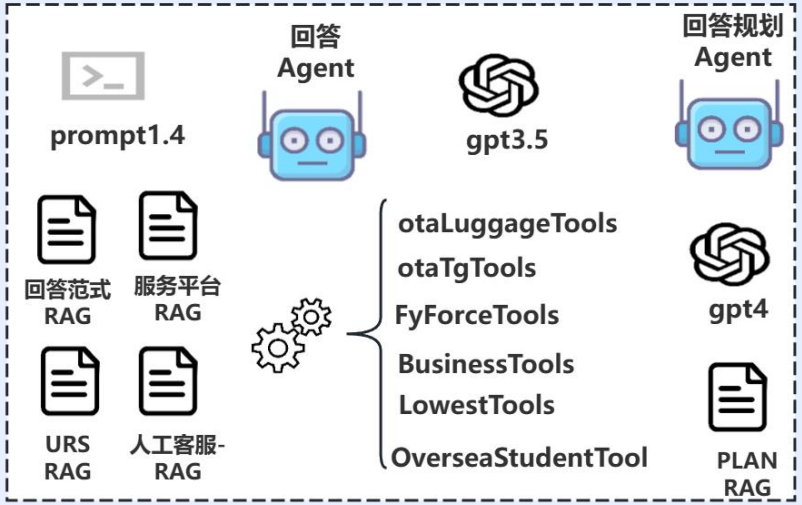
V1.2版 单Agent协作
Prompt合并+更换GPT4模型



V1.3版 单Agent协作
简化Prompt+增加知识覆盖+动静知识补充校准+Tools拆分和接入



V1.4版 单Agent协作
增加辅助自我反思Prompt+基于PLAN-RAG+隐式多轮



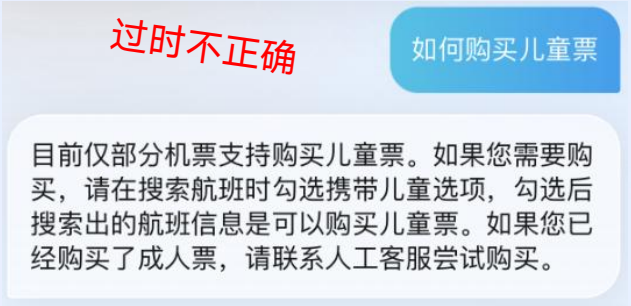
V1.0 版本问题

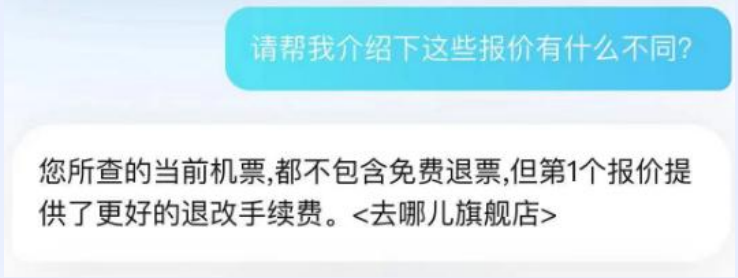
1）RAG数据少，答案过时不正确
2）Prompt长，gpt35不完全遵循规则
3）开放式回答准确率低
- 无意义、流水式回答
- 报价引导弱，无相关报价位置

V1.1 尝试解决

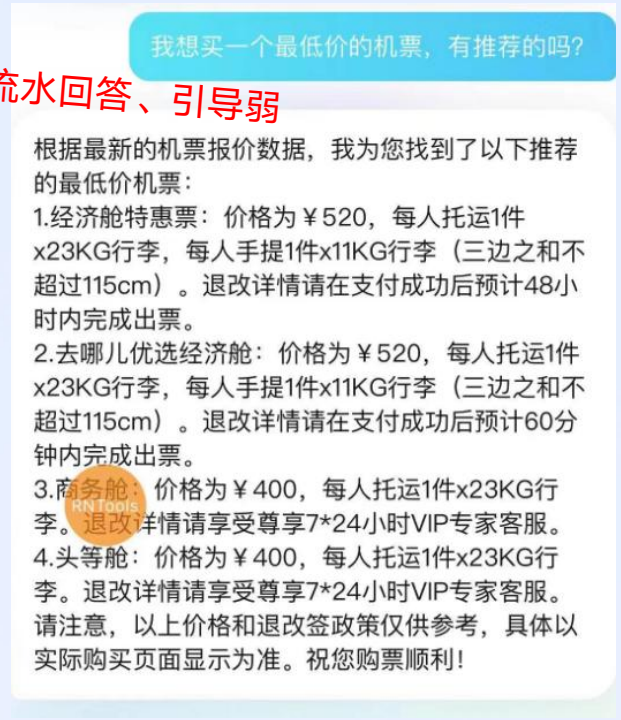
1）RAG校准、补充回答范式RAG知识
2）细化Tools 补充报价位置信息，拆分退改、行李等tools
3）优化Prompt，配置化

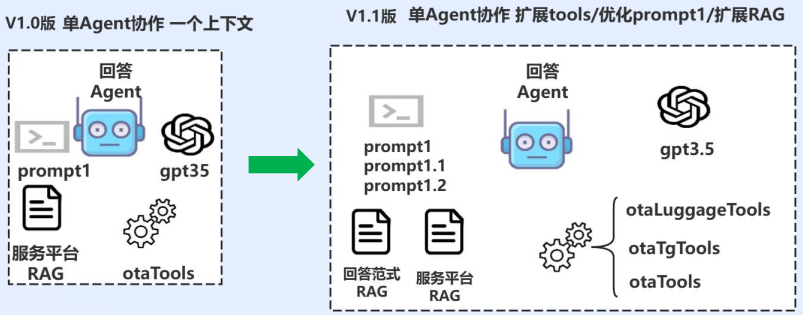
过时不正确





流水回答、引导弱





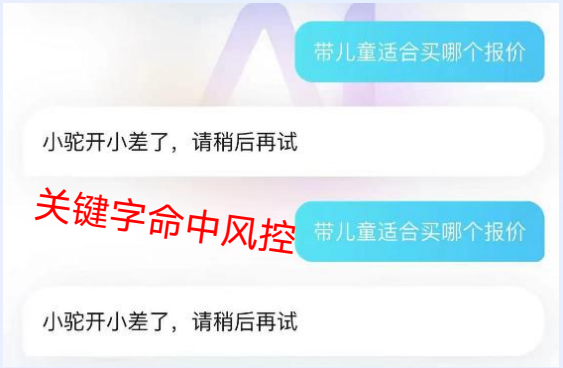
V1.1 版本问题

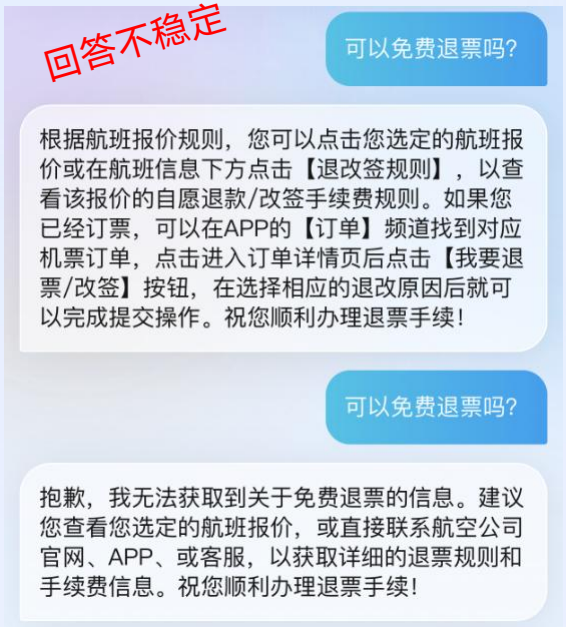
1）默认推荐问题，小孩关键字命中风控
2）回答不稳定

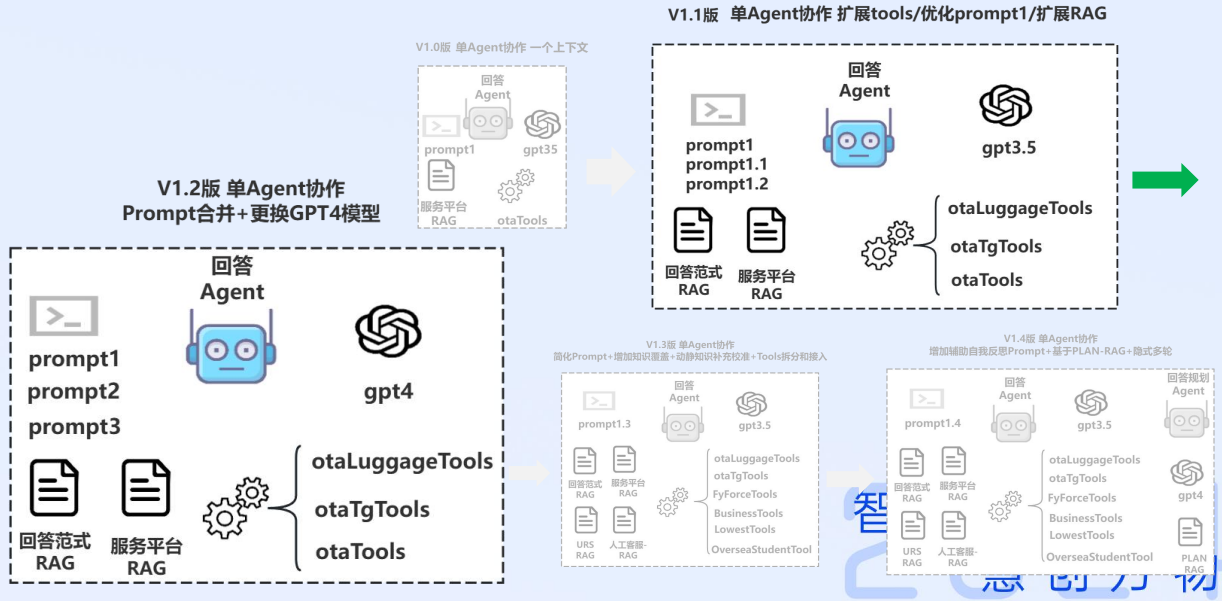
V1.2 尝试解决

1）Prompt问题同义词默认知识增加
2）升级模型到GPT4

售前助手-问题示例







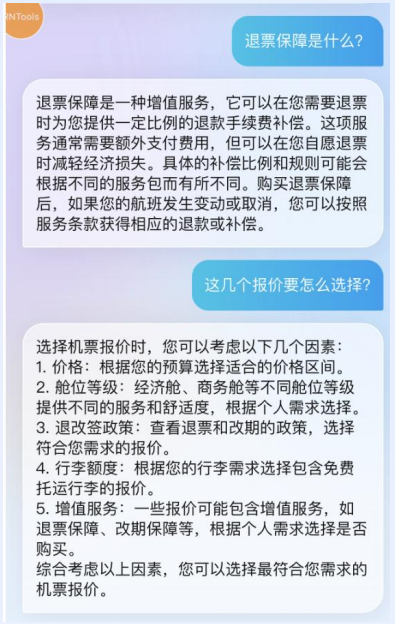
V1.2 版本问题

1）GPT4耗时慢 平均35s
2）RAG参考回答质量不高，回答不够用户

V1.3尝试解决

1）降级回GPT 3.5，拆分Agent职责，多Agent并行
2）RAG拆分和补充，Q-RAG/ QA-RAG/URS-RAG/人工客服RAG
3）RAG分类和校准，【静态答案】校准、【动态参考回答】优化
4）Tools继续拆分

售前助手-问题示例

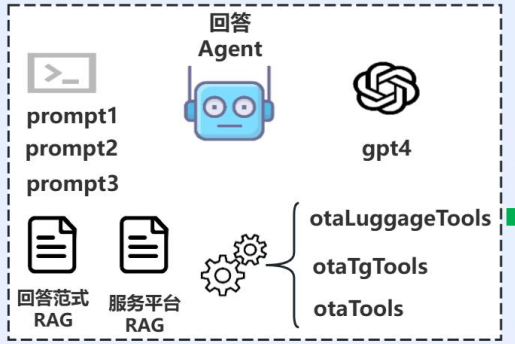


无工具，有RAG
35s返回

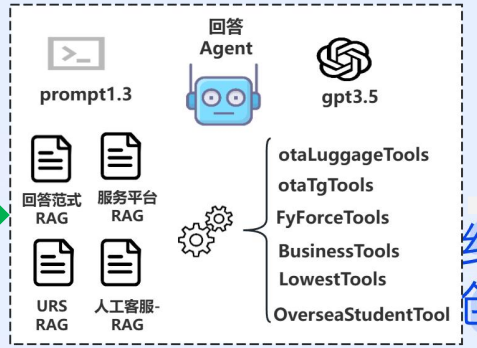
有RAG、有工具回调
52s返回

回答不够用户

V1.2版 单Agent协作
Prompt合并+更换GPT4模型



V1.3版 单Agent协作
简化Prompt+增加知识覆盖+动静知识补充校准+Tools拆分和接入



纪创万物

V1.3 版本问题

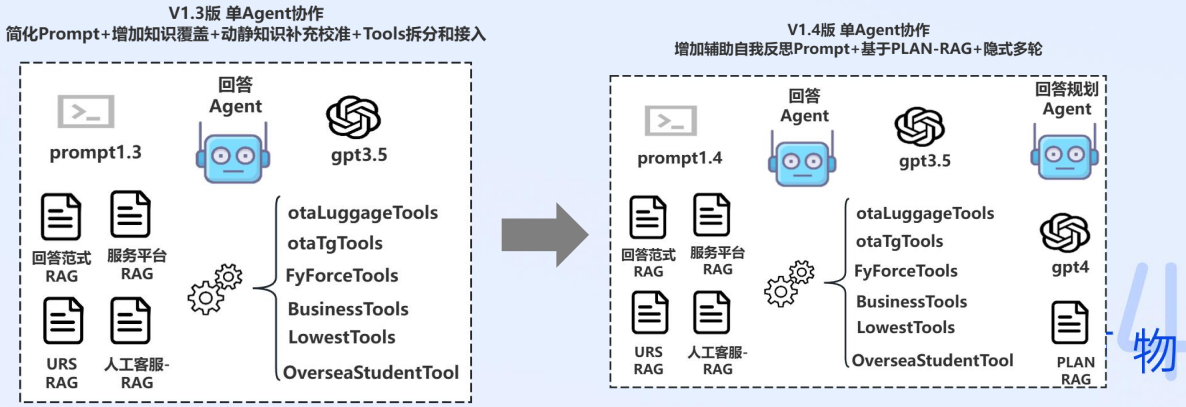
1） 【动态参考回答】优化成本高、效果慢
2）开放式问题回答准确率、一致率低

V1.4 尝试解决

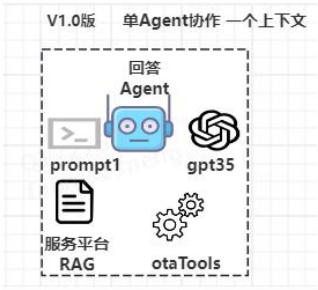
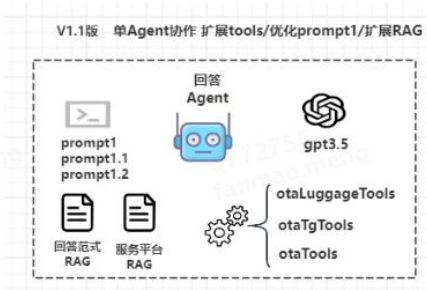
1）测试其他模型性能， 探索可能性
2）增加回答规划Agent，生成PLAN-RAG
3）Prompt增加自我反思流程
4）尝试隐式多轮，假设性问答
5）增加回答复盘Agent

售前助手V1.4-计划中

【开放式回答】初始版--->【开放式回答】优化版



Agent测试方案版本迭代记录

单Agent版本	v1.0	v1.1	v1.2	v1.3
版本功能	1）基于RAG数据回答 2）基于OAT实时数据回答 3）打招呼功能，默认推荐3个问题	1）RAG校准、补充回答范式RAG知识 2）细化Tools 补充报价位置信息，拆分退改、行李等tools 3）优化Prompt，并且QConfig配置化	1）Prompt问题同义词默认知识增加 2）升级模型到GPT4	1）降级回GPT 3.5，拆分Agent职责，多Agent并行 2）RAG拆分和补充，Q-RAG/ QA-RAG/URS-RAG/人工客服RAG 3）RAG分类和校准，【静态答案】校准、【动态参考回答】优化 4）Tools继续拆分
版本问题	1）RAG数据少，答案过时不正确 2）Prompt长，gpt35不完全遵循规则 3）回答准确率低 3.1 无意义、流水式回答 3.2 报价引导弱，无相关报价位置 4）默认问题推荐不合理	1）默认推荐问题，小孩关键字命中风控 2）回答不稳定	1）耗时慢 平均35s	
Agent组件图				

成果总结

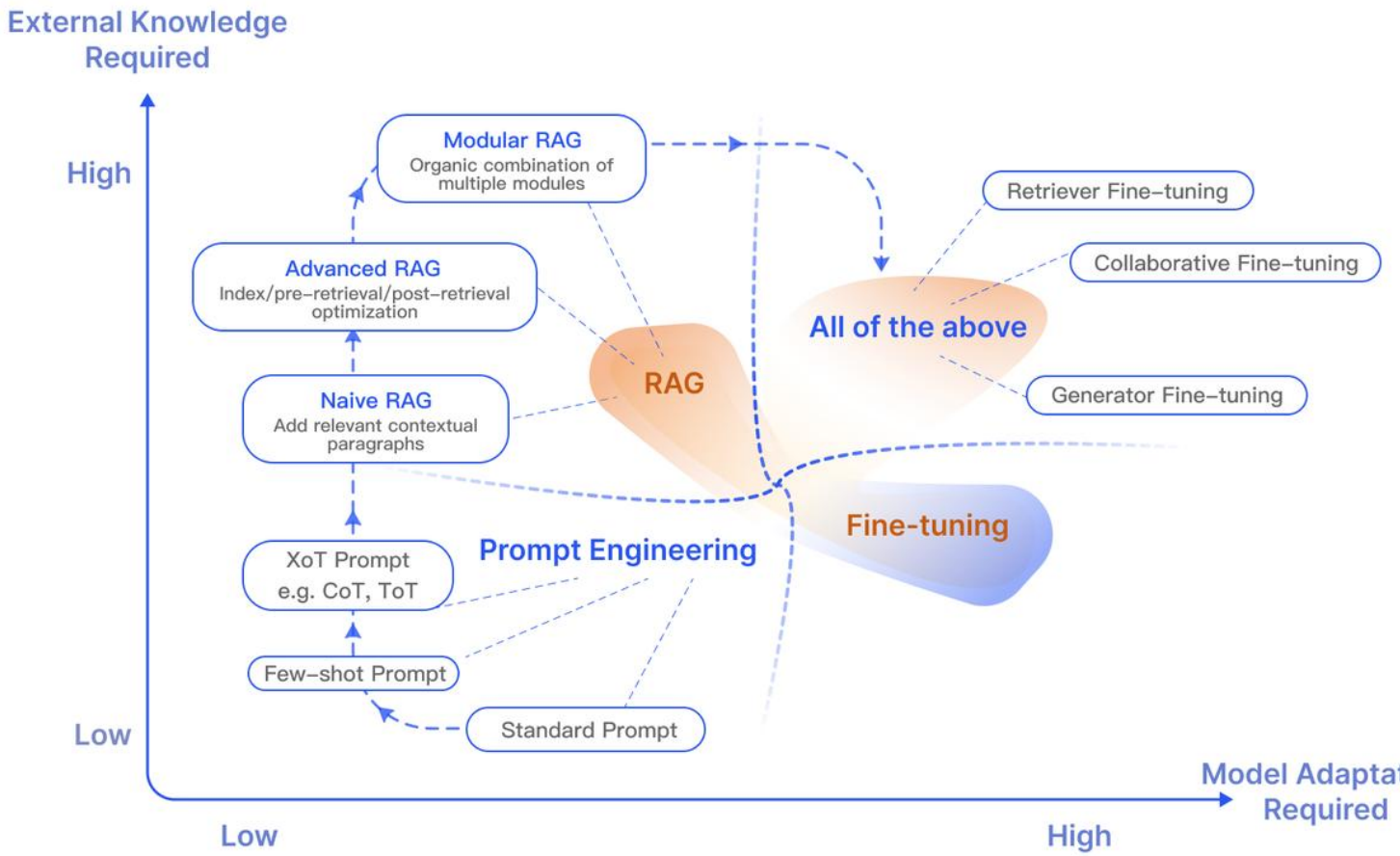
Agent优化的四个方向

一、优化Prompt

二、优化RAG

三、优化Tools

四、Model和Agent切换设计



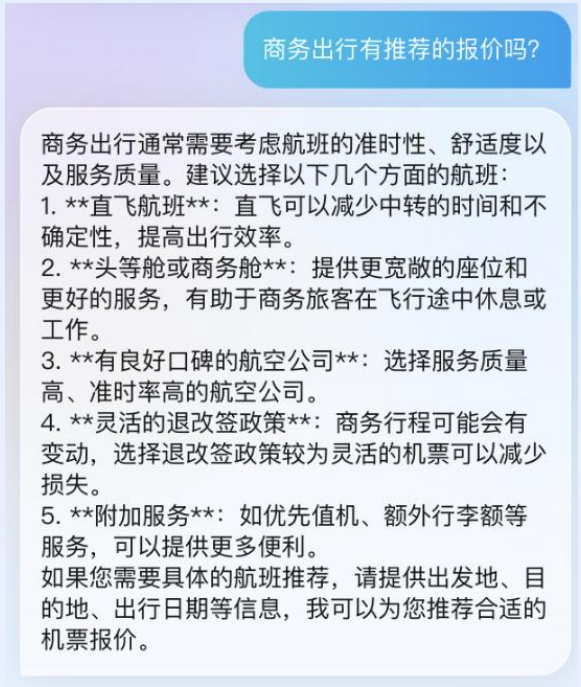
Prompt 从Zero-Shot到Few-Shot

1）默认通用知识、同义词知识补充

```
## 默认知识
    - **什么是增值服务**：退票保障、改期保障等服务介绍，帮助用户了解增值服务的价格和作用。
    - **小孩的含义**：在中文语意中，小孩、孩子、儿童是同等语意，可以互换使用
```

2）RAG回答范式补充Prompt

售前助手-示例

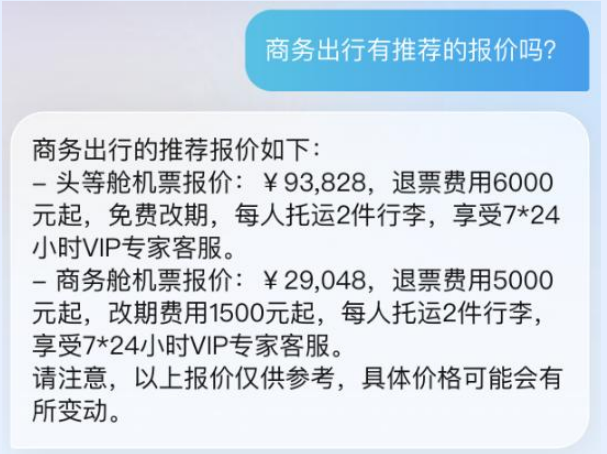


优化前

VS

RAG回答范式补充Prompt

```
## 参考回答格式：
问题：商务出行有推荐的报价吗？## 回答：您好，为了更好地旅行体验，我们推荐您考虑【头等/公务/商务舱】机票。 <br>
一个<商务舱/公务舱的机票报价产品信息> <br>
一个<头等舱的机票报价产品信息>。
```



优化后

RAG模块和运算符选择

1、附加元数据Metadata Attachment

可以使用元数据信息来丰富块，例如分类、特征、摘要等，随后根据该元数据过滤检索，从而限制搜索范围。

2、尝试使用元数据过滤

相似 ≠ 相关，根据块中的关键字和元数据进行过滤以缩小搜索范围,增加知识相关性

3、层次索引|Hierarchical Index

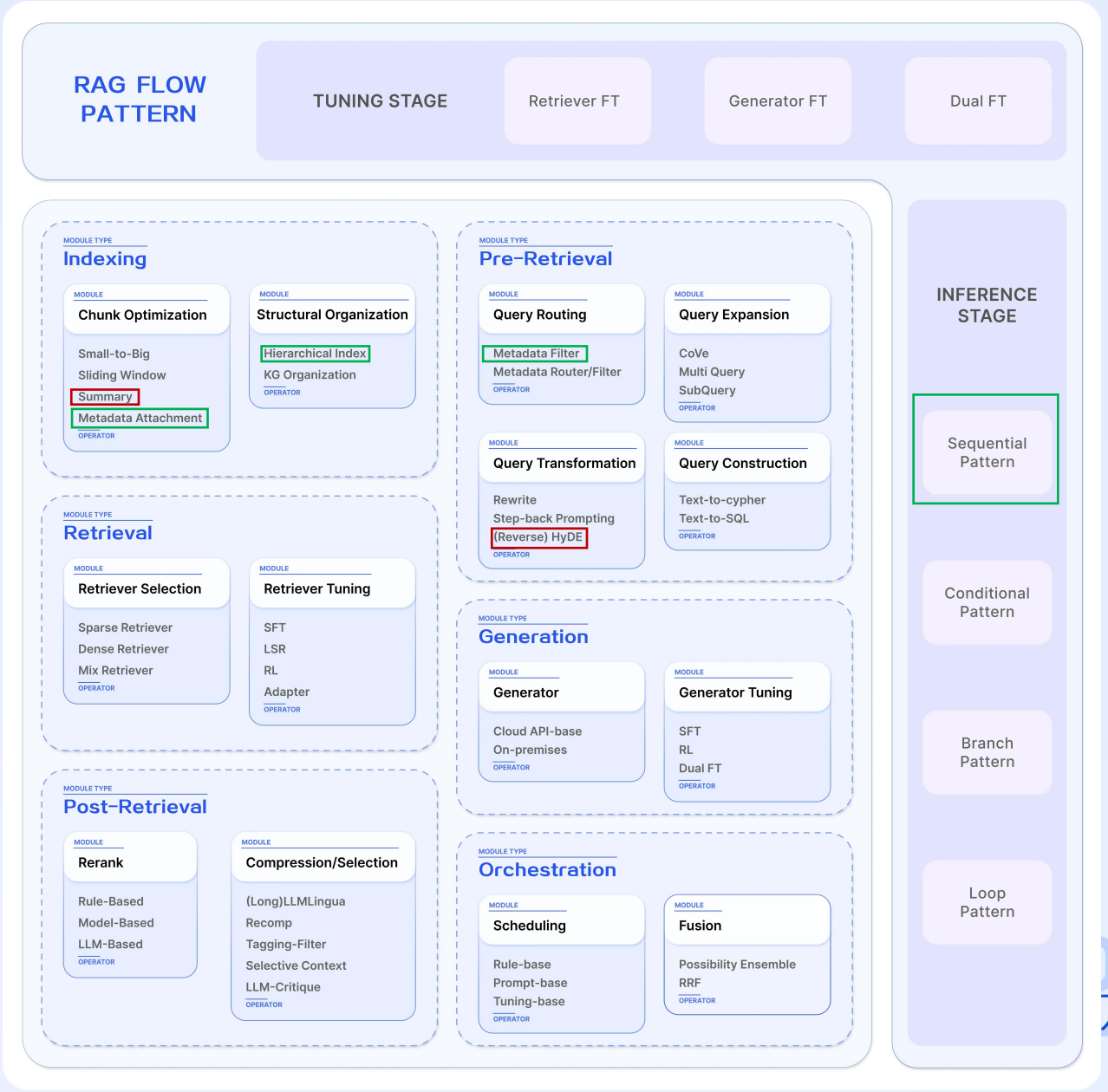
在文档的层次结构中，节点以父子关系排列，块链接到它们，增加知识的递进性。

RAG Flow流程选择

微调阶段：无
推理阶段：顺序模式。

以模块化 RAG 的思路优化

(以下的模块RAG，包含 6 种模块类型、14 个模块和 40 多个运算符，即优化方式)

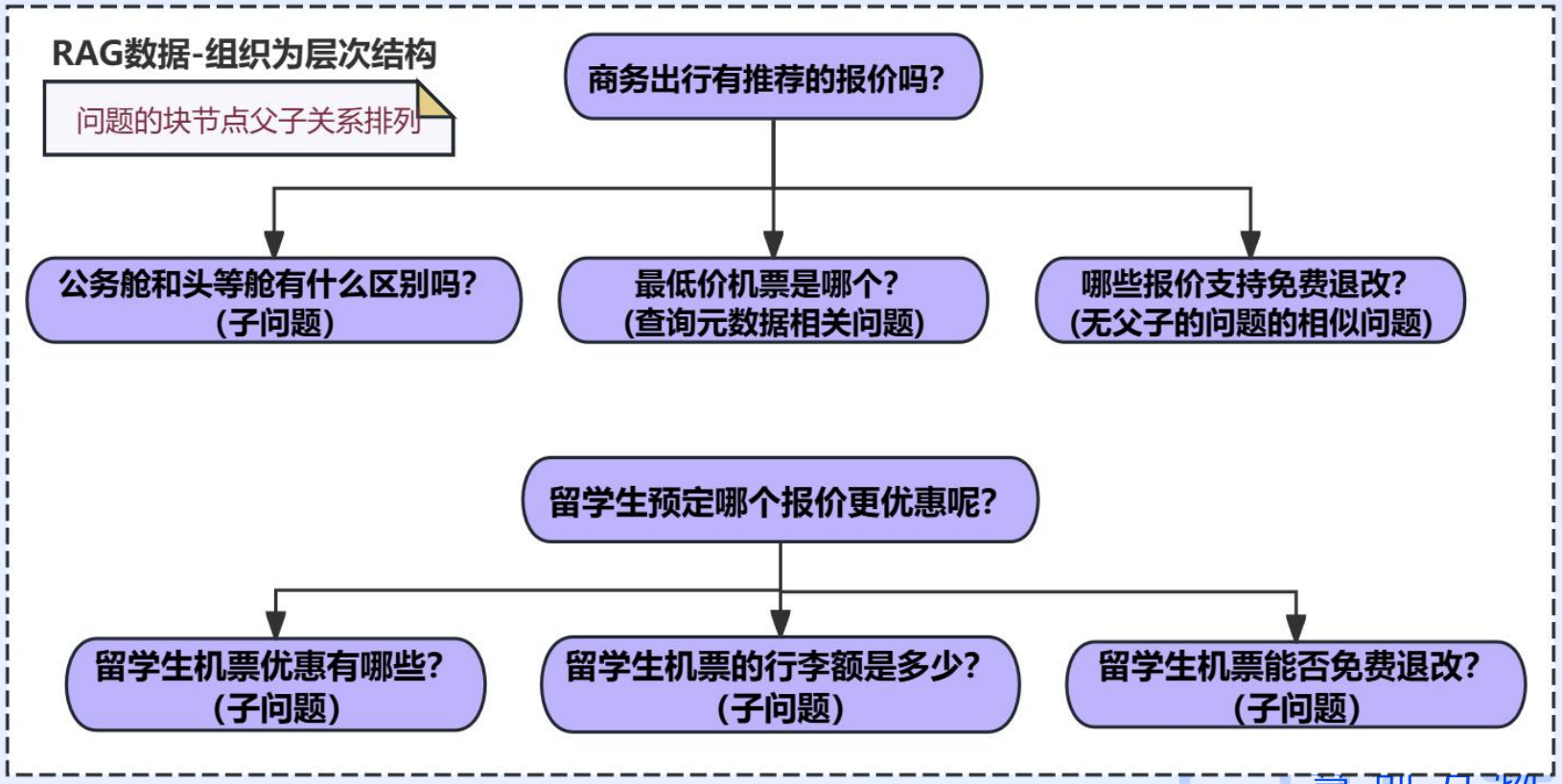


RAG模块和运算符选择

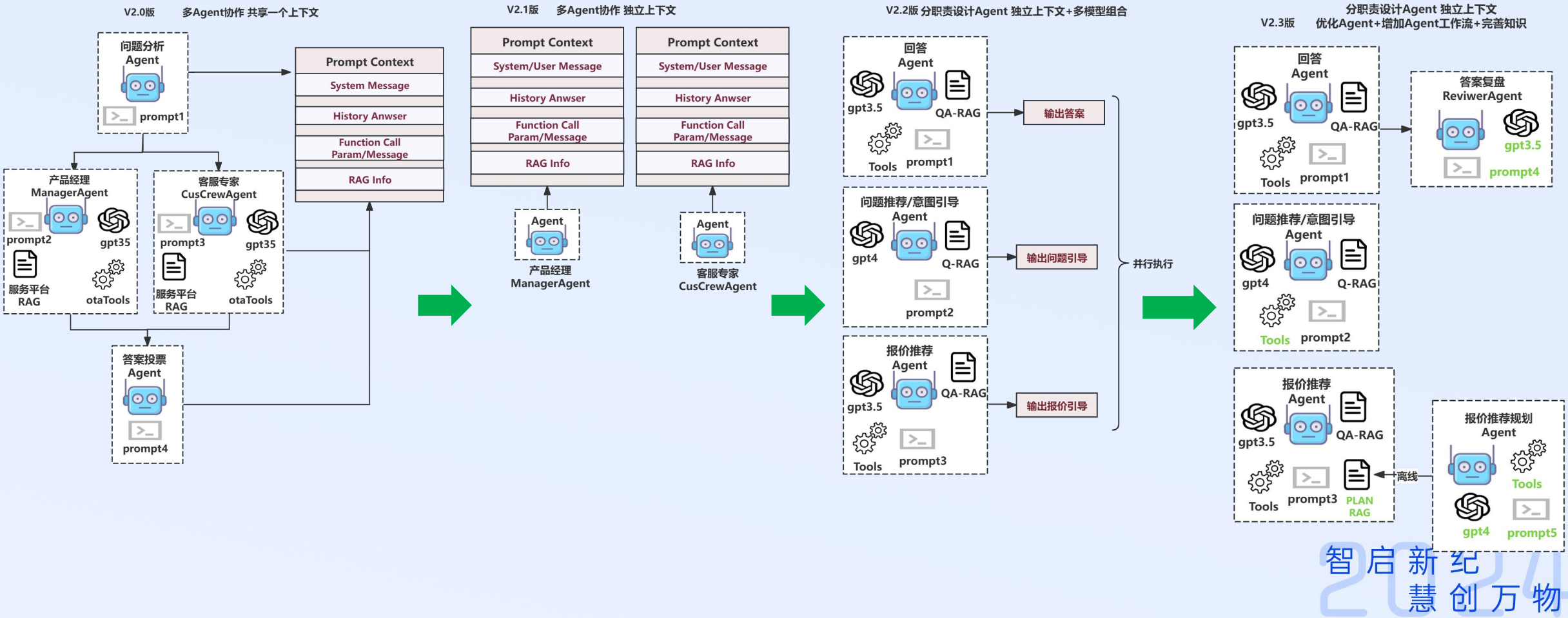
层次索引|Hierarchical Index

在文档的层次结构中，节点以父子关系排列，块链接到它们，增加知识的递进性。

question_embedding表的id 和 question_relation表的id进行关联



多Agent

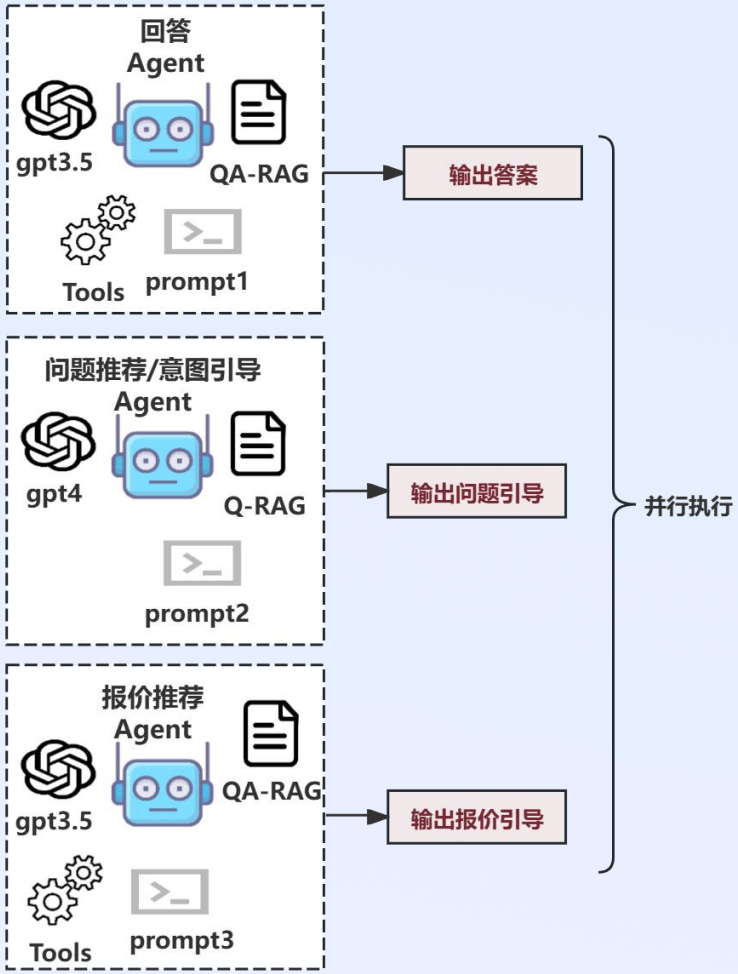


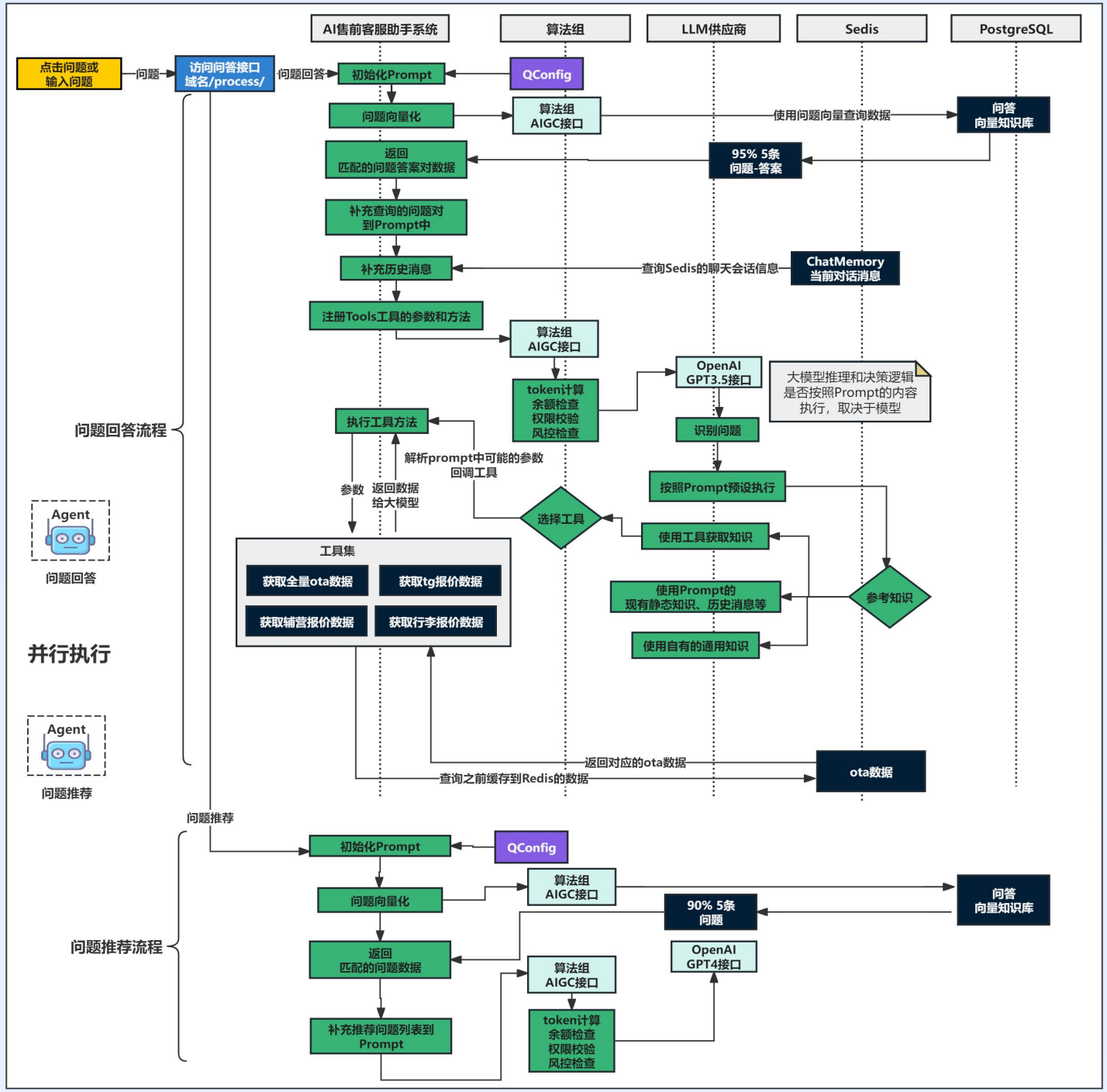
51CTO WOT 成果总结

多Agent

售前助手-示例

V2.2版 分职责设计Agent 独立上下文+多模型组合






16:02
去哪儿旅行
成都 — 首尔
分享
直飞 07-17 周三 成都 — 首尔 全程 3小时 25分
降价提醒
08:30 成都天府国际机场 T1
12:55 仁川国际机场 T1
中国国航CA401 | 空客321(中)
【建议持有返程票】持短期签证如商务、旅游等类型，建议购买或持...
Hi~ 钻石会员 | 下单享PLUS体验价~
出行公告(1条)
享立减 · 可享受最高减¥34优惠
经济舱 ¥1162起
头等/公务舱 ¥4069起
¥1172 惊喜已减 ¥34 降价必赔
预订
托运行李 1件x23KG
不可退票,改期300元起 | 预计12小时出票
买贵退差保障+¥68
机票价格波动大，买贵最高退¥500
最高25元国际机票
更换
国际酒店代金券
60元国际酒店代金券
¥1162 2项优惠已减 ¥44
预订
托运行李 1件x23KG
不可退票,改期300元起 | 预计12小时出票
航空意外险+¥60
850万航空意外保障
3万航空意外医疗保障
更多
¥1220 中国国航 国际直营旗舰店
选购
托运行李 1件x23KG
不可退票,改期300元起 | 实时出票
价格最低
¥1172 惊喜已减 ¥34 金牌产品
选购
托运行李 1件x23KG
不可退票,改期300元起 | 预计12小时出票


16:02
去哪儿旅行
机票AI助手
飞行时长有多久?
商务出行有推荐的报价吗?
留学生应该预定哪个报价呢?
价格最低的机票是哪个?
您好，根据小驼的查询，当前报价中，价格最低的机票位于第4位，机票价格是1172元人民币。
以下是为您推荐的机票报价:
¥1172 惊喜已减 ¥34 金牌产品
预订
咨询
托运行李 1件x23KG
不可退票,改期300元起 | 预计12小时出票
相关问题
价格最低的机票退改签政策是怎样的?
价格最低的机票行李托运是多少?
机票余量不足，应该怎么办?


16:03
去哪儿旅行
填单页-->选服务-->支付
客服
直飞 成都 — 首尔 07-17周三 08:30
经济舱 | 成人含税价¥1206 儿童含税价¥1060
婴儿购票说明
不可退票,改期300元起 | 托运行李 1件x23KG
选购升级服务，行程变动也省钱
退票补贴 ¥65/程
最高省70%手续费
机票降价补差价¥65/份
买贵最高补¥500
改签补贴
选择乘机人
+更多乘机人
旅途多重保障 家人更放心
航空意外险 ¥60/程
航意身故保障¥850万，航意医疗保障¥3万
国际组合险 ¥80/程
航意身故保障¥500万，到达延误最高赔¥1000
真划算！本单已享34元立减
¥1172 1206
明细
下一步

51CTO WOT

谢 谢 观 看

THANKS

智启新纪
慧创万物
2024