

NiDD AI+ 研发数字峰会
AI+ Development Digital summit

第5届

AI驱动的智能工单系统建设和提效

李佳奇 | 去哪儿旅行

科技生态圈峰会 + 深度研习



—1000+ 技术团队的选择



 **K+峰会**  **敦煌站**

K+ 思考周®研习社

时间: 2025.08.29-30

 **K+峰会**  **上海站**

K+ 金融专场

时间: 2025.10.17-18

 **K+峰会**  **香港站**

K+ 思考周®研习社

时间: 2025.11.25-26



K+峰会详情



 **AiDD峰会**  **上海站**

AI+研发数字峰会

时间: 2025.05.17-18

 **AiDD峰会**  **北京站**

AI+研发数字峰会

时间: 2025.08.08-09

 **AiDD峰会**  **深圳站**

AI+研发数字峰会

时间: 2025.11.28-29



AiDD峰会详情



李佳奇

去哪儿旅行 技术总监/机票技术委员会主席

去哪儿旅行技术总监，技术中心TC委员、业务架构SIG负责人、机票研发TC主席，负责机票研发团队的AIGC落地和相关技术储备及人才培养。10余年OTA一线研发经验，在高并发高可用系统建设、DDD项目落地、业务域系统价值度量、线上系统防腐治理等领域有代表性作品。多次在各种技术峰会担任出品人和讲师，和AZone架构院、K+Talk等合作推出线上课程和直播。

目录

CONTENTS

1. 背景介绍
2. 方案设计
3. 成果展示
4. 总结与展望

PART 01

背景介绍

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



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

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

▶ 背景介绍

AI能力/业务机会	文本生成	图像生成	视频生成	逻辑推理	数据分析
搜索				✓	
预定	✓				
交易			✓		✓
出行		✓			
售后			✓		
用户体验				✓	
营销	✓				

AI*业务矩阵 (示例)

团队提效

产运提效

报表助手

归因助手

商分助手

用分助手

研发提效

checklist生成

代码辅助

深度赋能

工单自动化

AI OPS



每周工单处理工时100+pd

每周工单量400+

可自动化处理率>40%

机票研发团队工单现状



大模型

- 1 问题识别
- 2 关联应用
- 3 处理策略

工具+知识库

- 1 日志
- 2 代码
- 3 知识库
- 4 精简/聚合

大模型

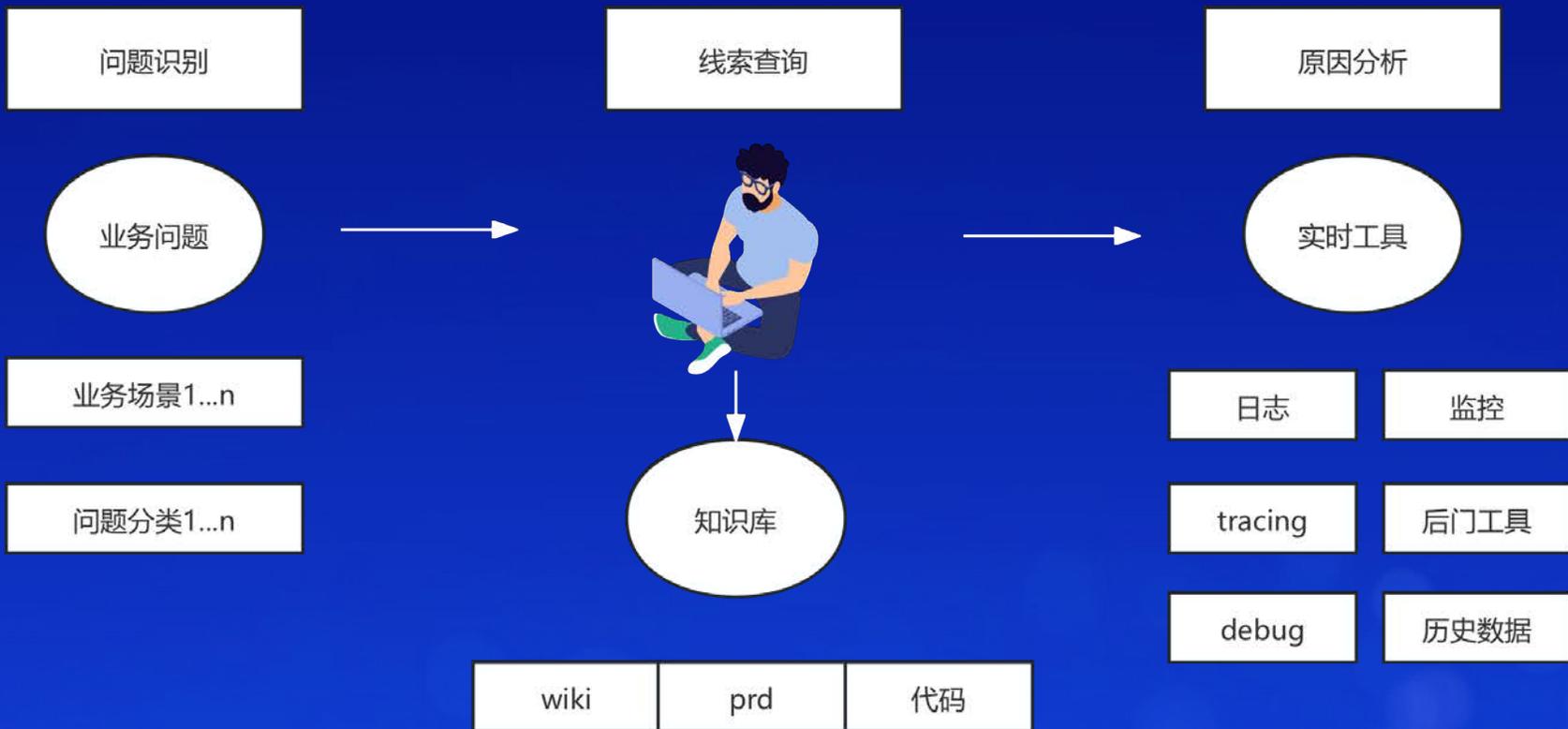
- 1 归因分析
- 2 生成结论

借助大模型建设工单自动化处理流程

PART 02

方案介绍

▶▶ 方案介绍-人工如何处理工单



人工处理工单的弊端

- 依赖人的知识和经验
- 占用人力成本高
- 工时投入产出比低
- 处理结果方差大

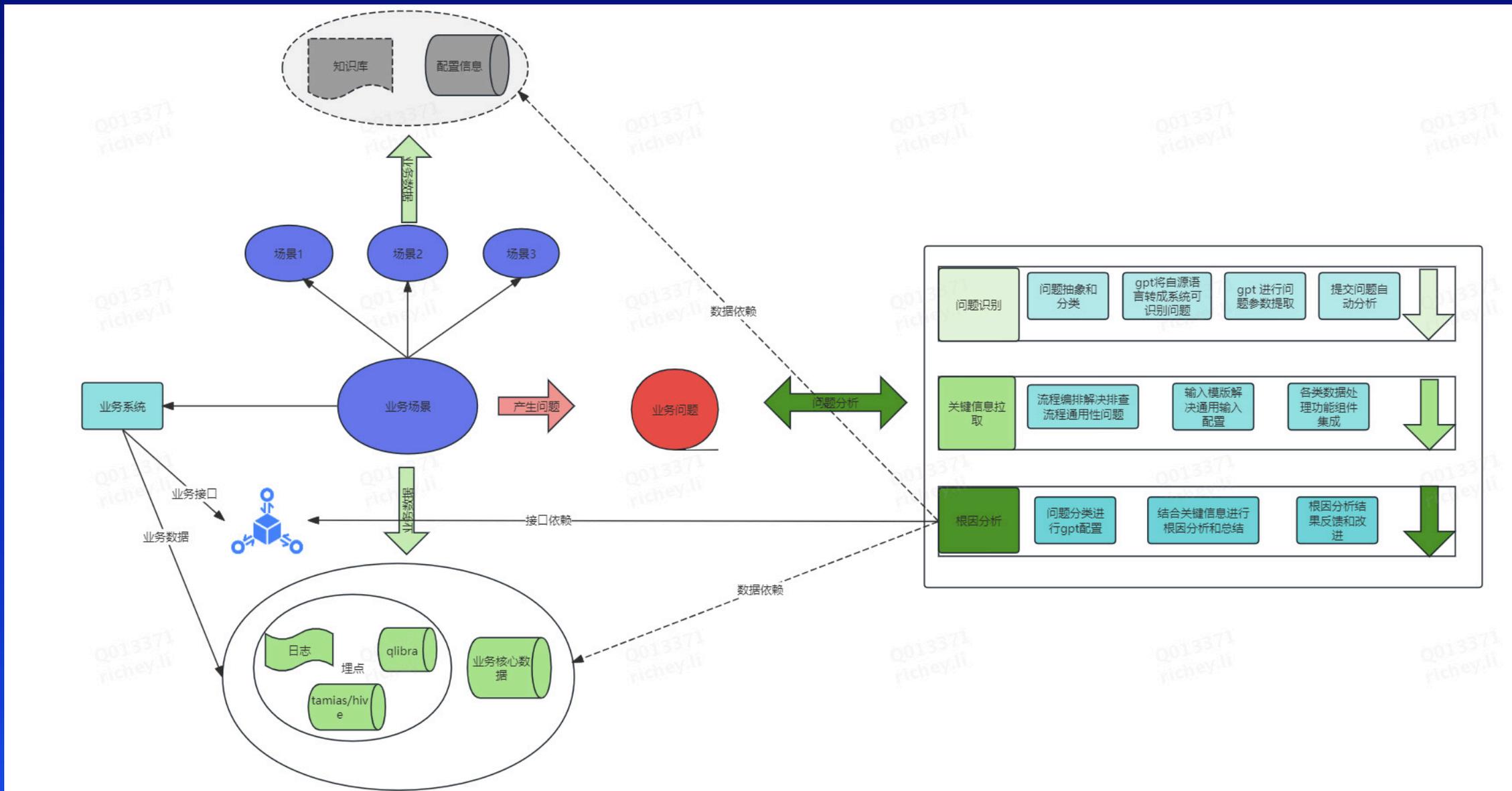
开发是如何处理工单的

▶▶ 方案介绍-1.0版本自动化处理流程

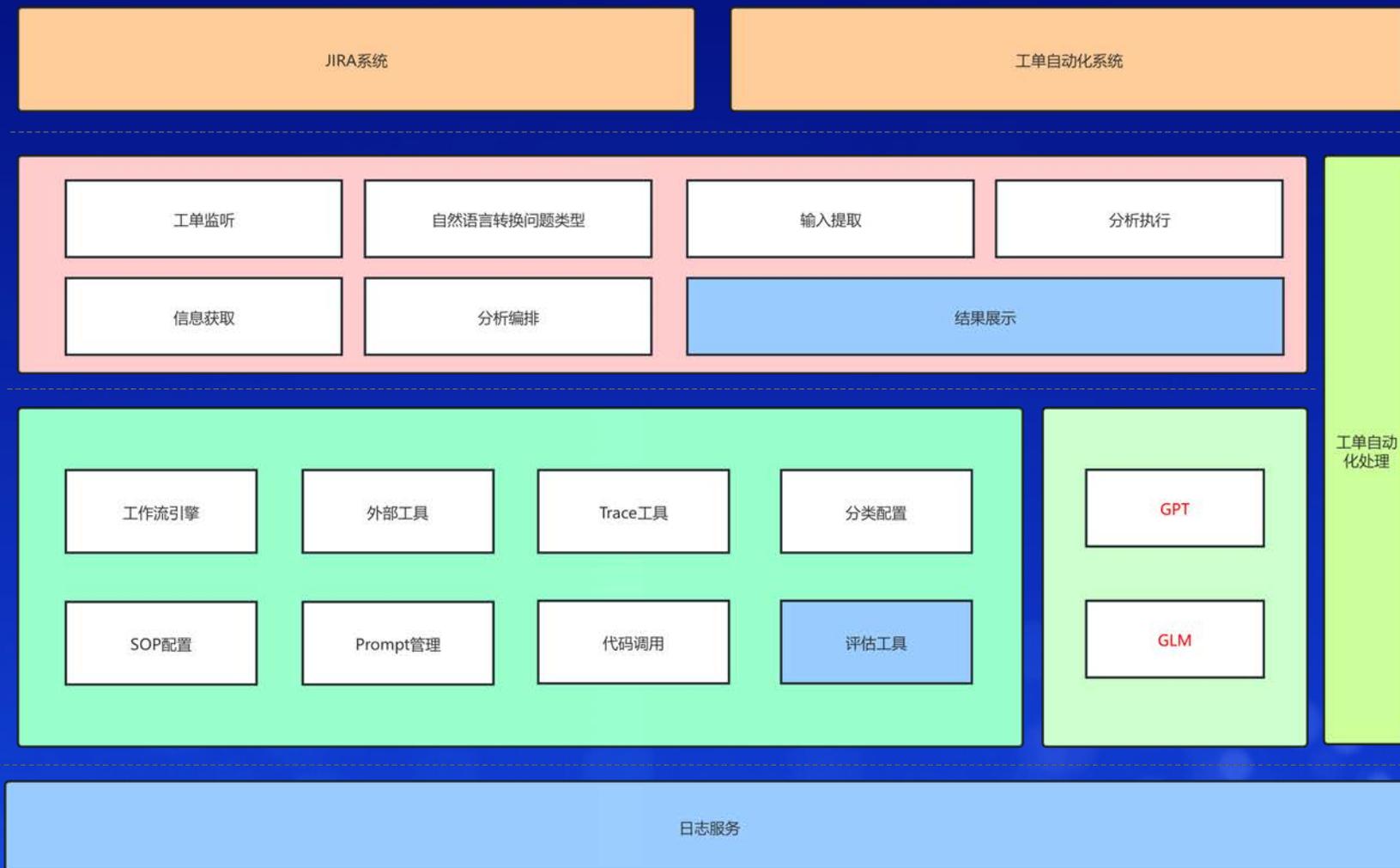


自动化处理步骤拆解

方案介绍-1.0版本自动化处理技术方案



▶▶ 方案介绍-1.0版本系统架构设计



1.0版本架构设计

▶▶ 方案介绍-1.0版本日志处理和LLM选型

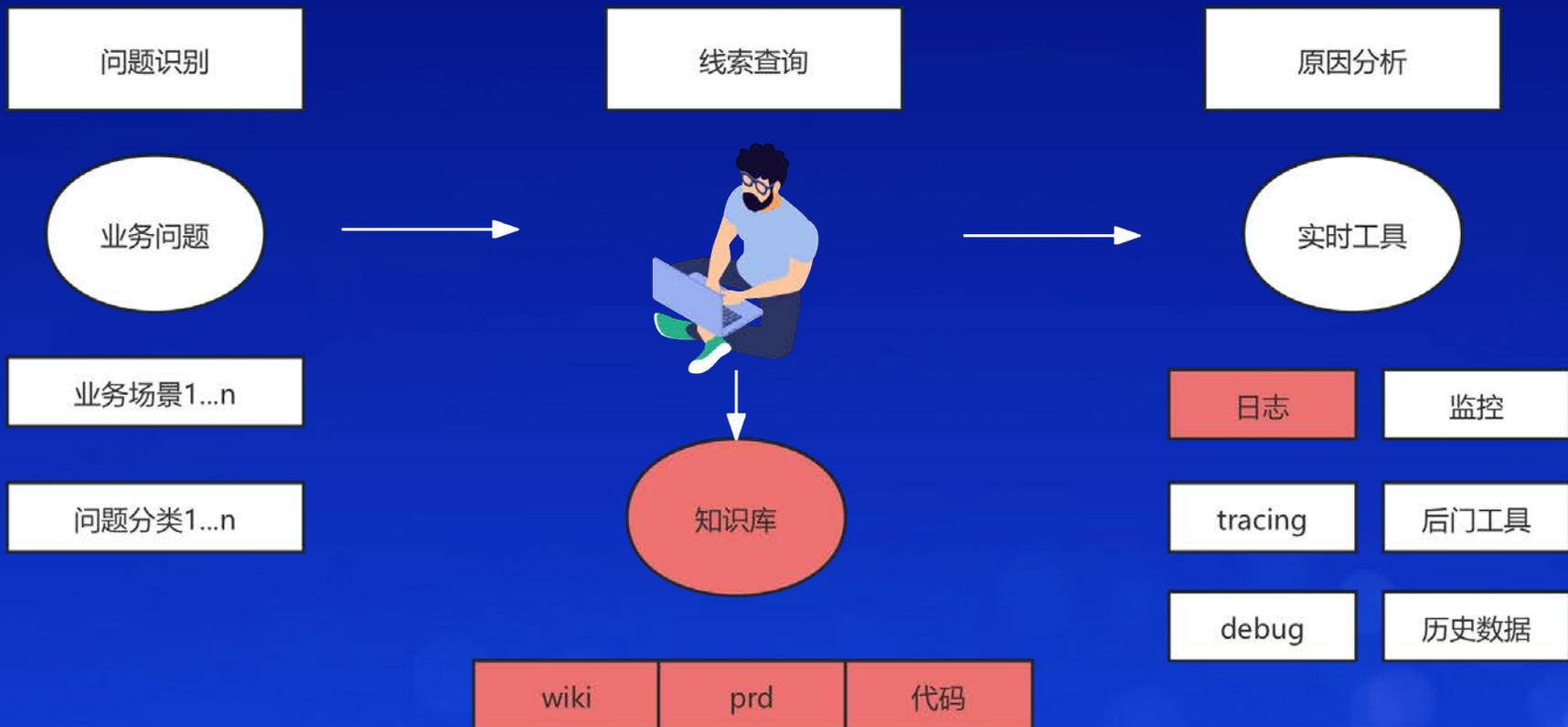


日志处理



LLM选型

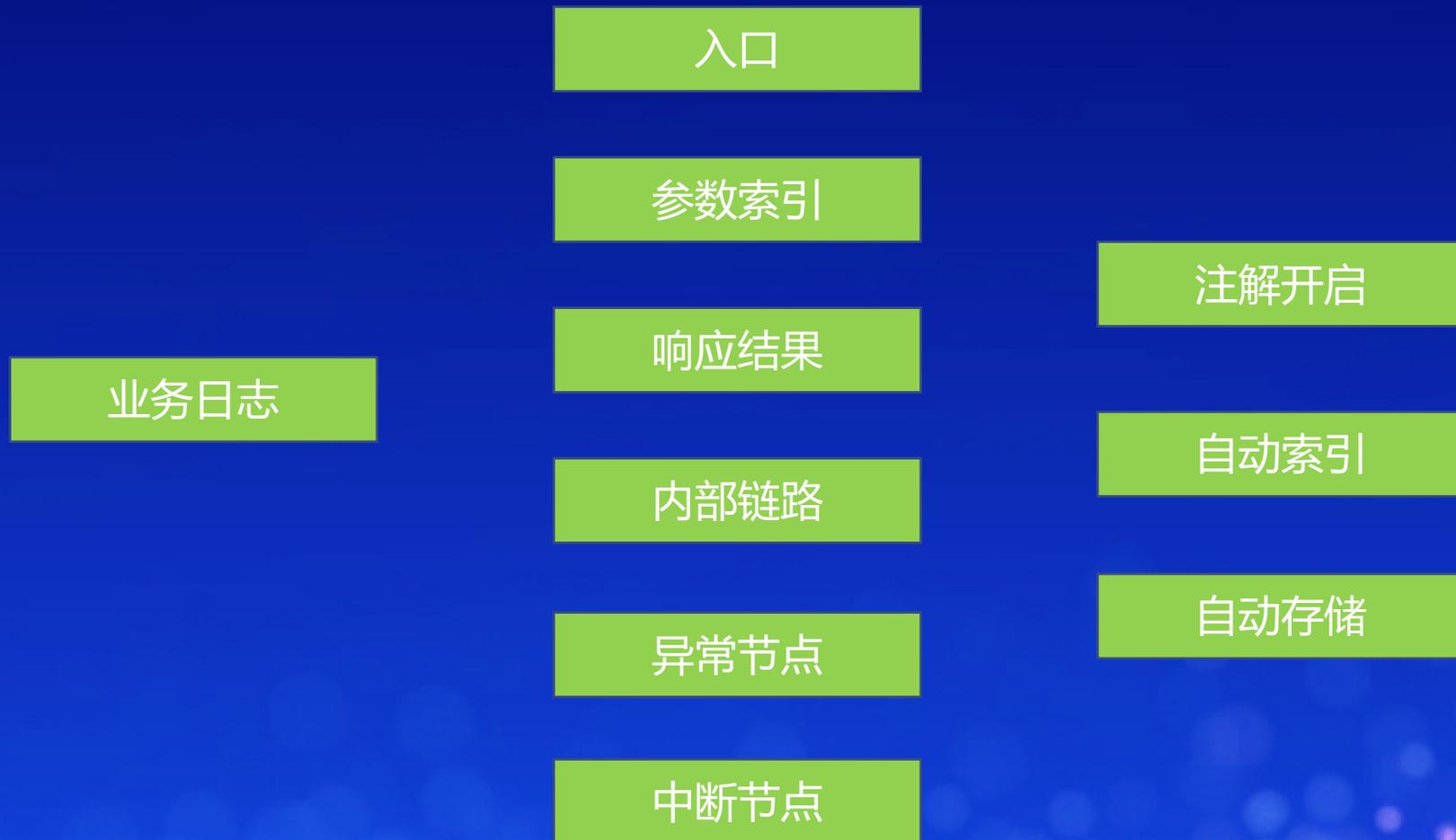
▶▶ 方案介绍-1.0版本知识不全的问题



解决工单需要更多的知识

- 需要把代码也纳入到工单自动处理
- 原有的系统日志结构化差，有效信息密度低

▶▶ 方案介绍-2.0方案之结构化日志埋点组件的功能



自动埋点Troy组件

▶ 方案介绍-2.0方案之结构化日志埋点组件的使用

如何使用注解开启一条链路?

在方法上使用 `@TroyRoot` 注解, 见下表

字段	是否必填	含义	默认	备注
name	是	节点名称	无	
linkCode	是	链路唯一标识	无	不可重复
linkDesc	是	链路描述	无	
recordLinkResult	否	是否记录链路返回结果	不记录	
resultTag	否	方法记录数据类型tag	不记录	枚举类型
resultDesc	否	方法记录数据描述	空字符串	
converter	否	方法返回结果记录方式	对象引用	
exception	否	是否开启异常记录	不开启	
whiteListKey	否	白名单	空字符串	
interrupt	否	中断断言类	断言接口	需要实现接口

```
@TroyRoot(name = "订单航班查询请求入口", linkDesc = "查询订单对应航班动态与航变信息", linkCode = "queryFlightByOrderNo", recordLinkResult = true)  
public FlightStatusResult queryFlightByOrderNo(@LinkSearched QueryCondition queryCondition) {  
    // code...  
}
```

反馈

如何使用注解定义链路中的节点？

在方法上使用 `@TroyNode` 注解，见下表

字段	是否必填	含义	默认	备注
name	是	节点名称	无	
resultTag	否	方法记录数据类型tag	不记录	枚举类型
resultDesc	否	方法记录数据描述	空字符串	
converter	否	方法返回结果记录方式	对象引用	
exception	否	是否开启异常记录	不开启	
interrupt	否	中断断言类	断言接口	需要实现接口

```
@TroyNode(name = "根据订单号和ApCode查询航段信息", resultTag = RecordEnum.processData, resultDesc = "航段信息")
public FlightSegmentRes getFlightSegmentByOrderNoAndApCode(String orderNo, String depApCode, String arrApCode) {
    // code...
}
```

如何使用注解记录搜索项?

在参数上使用 `@LinkSearchItem` 注解, 见下表

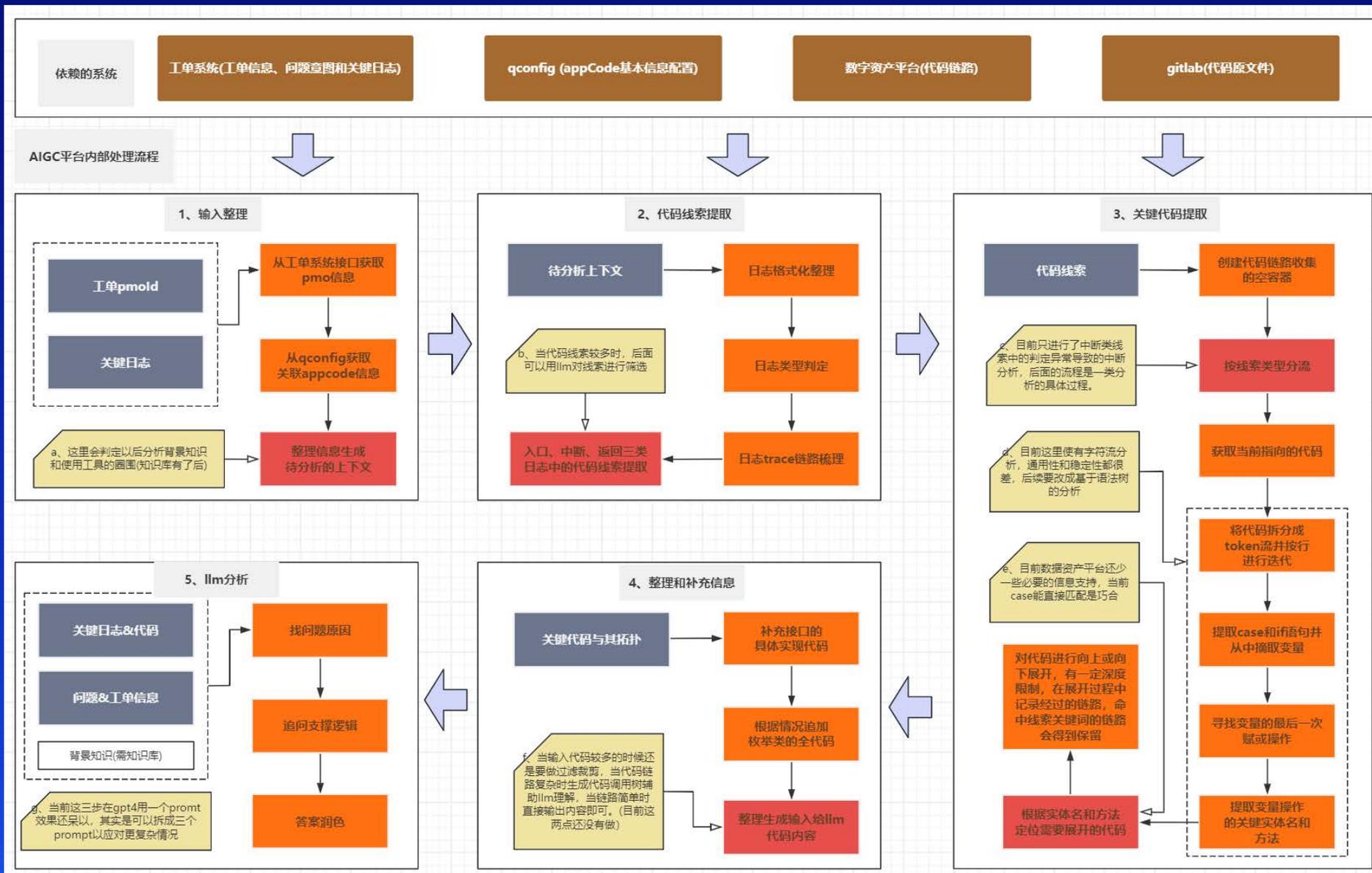
字段	是否必填	含义	默认	备注
value	否	搜索项	参数名	
converter	否	记录方式	对象引用	
extractorKey	否	自定义converter	空字符串	需要实现接口
extractor	否	自定义converter	接口	需要实现接口

```
public FlightStatusResult test(@LinkSearchItem("flightNo") String flightNo, @LinkSearchItem String depDate) {  
    // code....  
}
```

如果入参为一个对象, 则使用 `@LinkSearched` 注解, 然后在该类内部使用 `@LinkSearchItem`, 如:

```
public FlightStatusResult test(@LinkSearched QueryCondition queryCondition){  
    // code....  
}  
  
public class QueryCondition {  
  
    @LinkSearchItem("orderNo")  
    private String orderNo;  
  
    @LinkSearchItem  
    private String flightNo;  
}
```

方案介绍-2.0方案之引入代码来增强工单处理效果

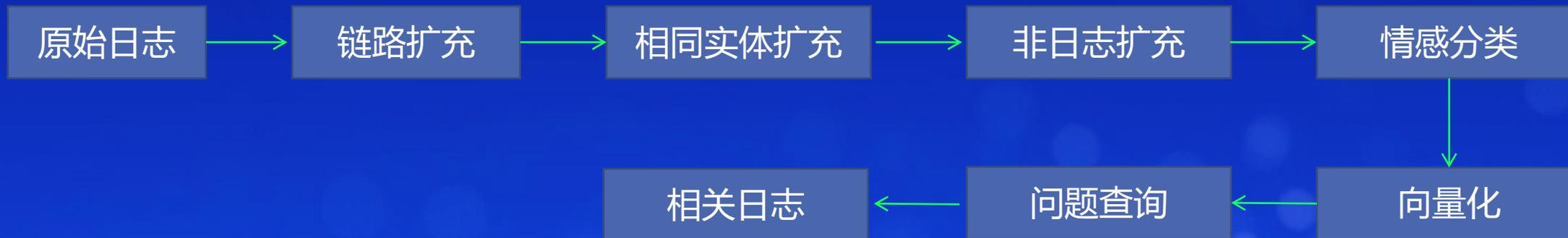


2.0版本架构设计

▶▶ 方案介绍-2.0版本语义搜索



代码语义化



日志向量化

▶▶ 方案介绍-更多可能性



做完以上这些足够了吗

▶▶ 方案介绍-基于会话的3.0智能工单处理方案

基于会话的端到端的解决方案是智能工单的理想形态



会话+Agent+RAG

- 产品、运营直接以会话的形式和工单智能体进行交互
- 将处理工单所需要的日志、代码、知识等内容构造为知识库，使用RAG技术进行检索，提高检索结果的有效信息密度

3.0版本智能工单的形态

▶▶ 方案介绍-AIGC开发框架基建

Langchain4J Qunar

LlmaIndex4J Qunar

支持单Agent开发

支持RAG全流程

支持权限管理

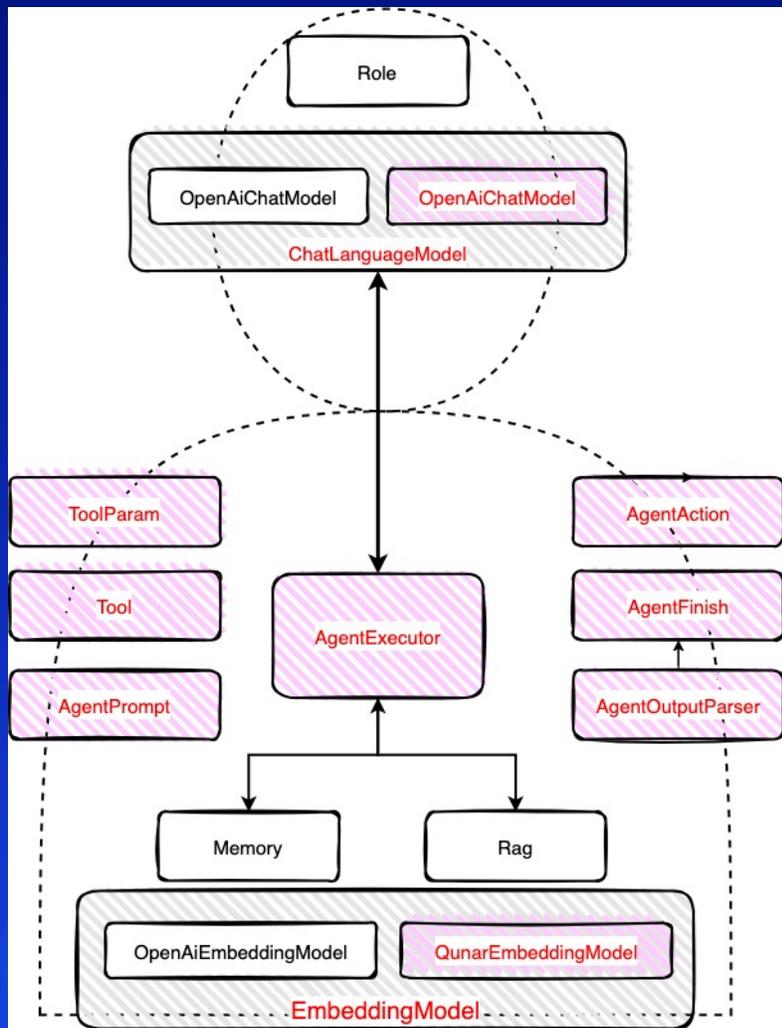
针对代码RAG增强

支持Prompt管理

支持多Agent开发

Agent和RAG Java开发框架建设

► 方案介绍-LangChain4J Qunar



Java版：文本补全助手
CompletionAssistant

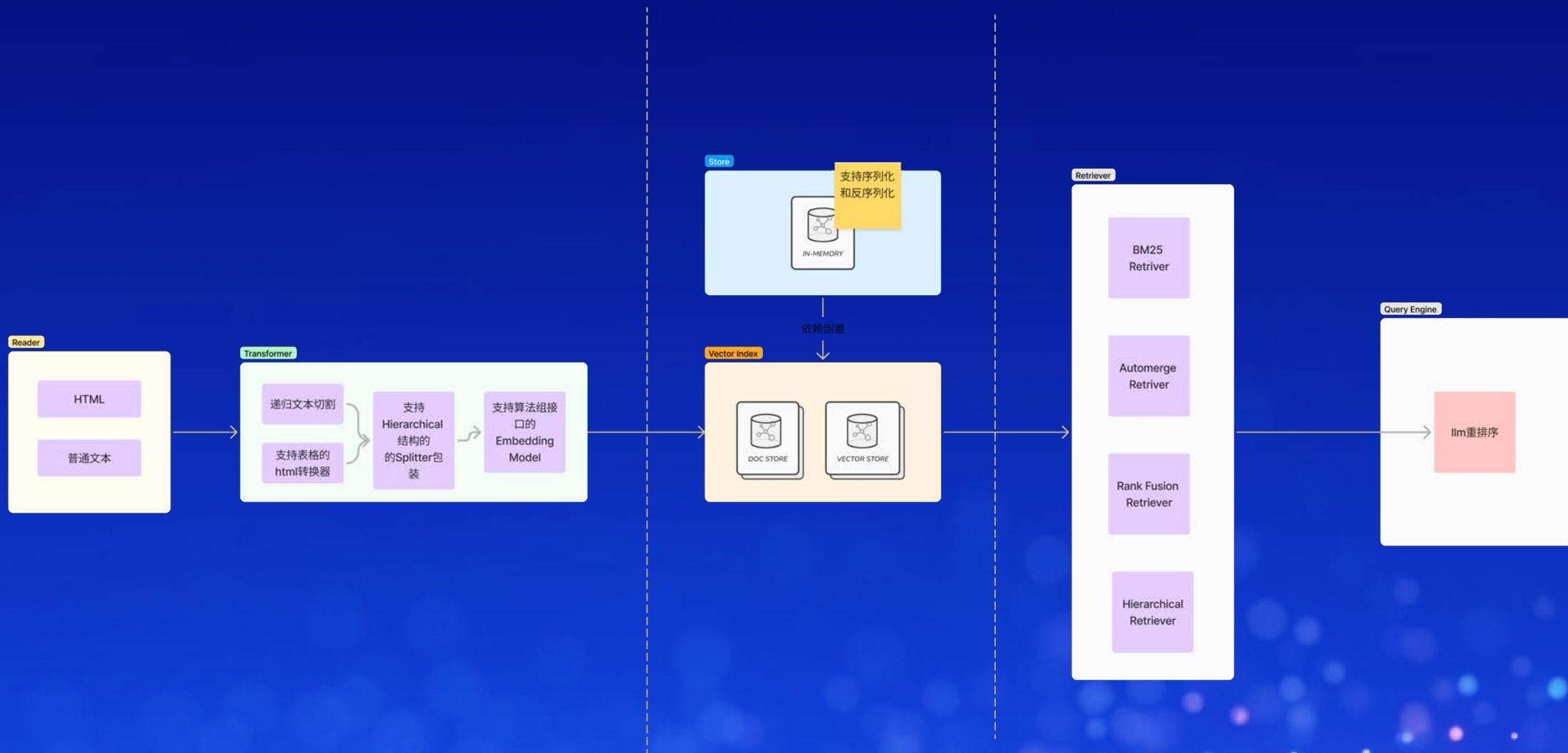


Java版：聊天对话助手
ChatAssistant, 相比文本补全新增了：
聊天角色, system、user、assistant
历史会话记忆: ChatMemory
相关内容获取: ChatMessageRetriever

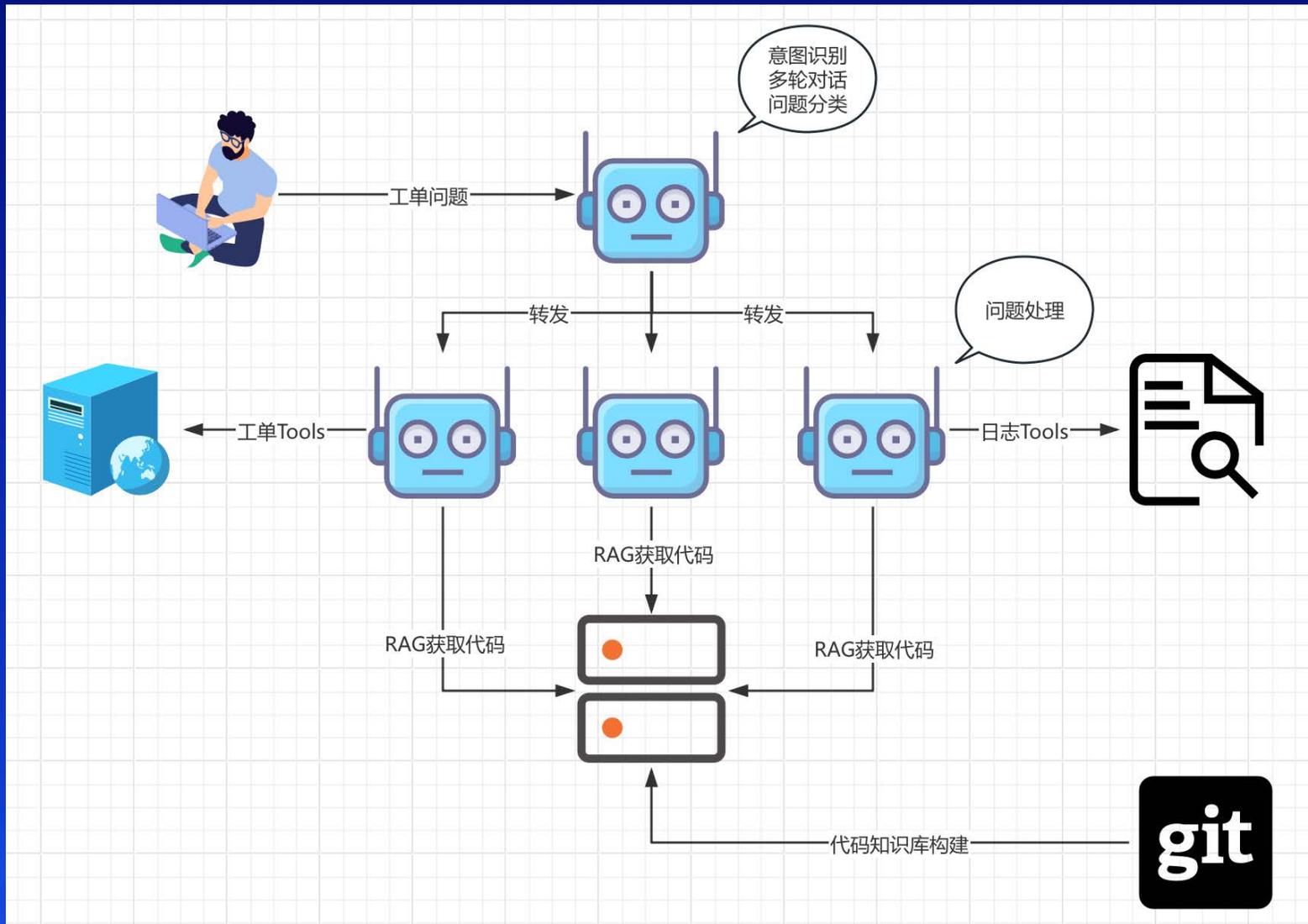


Java版：Agents
Agent(代理) - 大脑
Tool(工具) - 各种方法函数等
Agent_Executor(代理执行器)
代理执行器是代理的运行时。这是实际调用代理并执行其选择的动作的部分

▶▶ 方案介绍-LlamaIndex4J Qunar



▶ 方案介绍-3.0方案之工单智能体工作流程



Agent流程

▶ 方案介绍-3.0方案之工单智能体的核心提示词

```
ChatComplexResponse (data=ChatComplexResponse.ChatComplexResponseData (prompt=ChatComplexResponse.Prompt (messages=[Message (role=system, name=system1, content= ## 角色:
- 工单问题排查和业务咨询助手, 资深的工单问题处理专家。

## 背景:
- 将要分析的是某OTA平台机票交易系统工单。

## 目标:
- 根据给定的问题、问题相关日志和相关代码片段, 回答产生问题的根因

## 限制:
- 你的回答必须基于客观事实、日志内容、代码逻辑和对OTA平台的了解, 回答内容要以不太了解这块业务的能也能看懂为标准
- 生成追问的时候只能生成一个追问

## 技能:
- 理解用户问题和意图的能力
- 分析日志能力
- 分析代码能力
- 结合日志和代码排查问题的能力

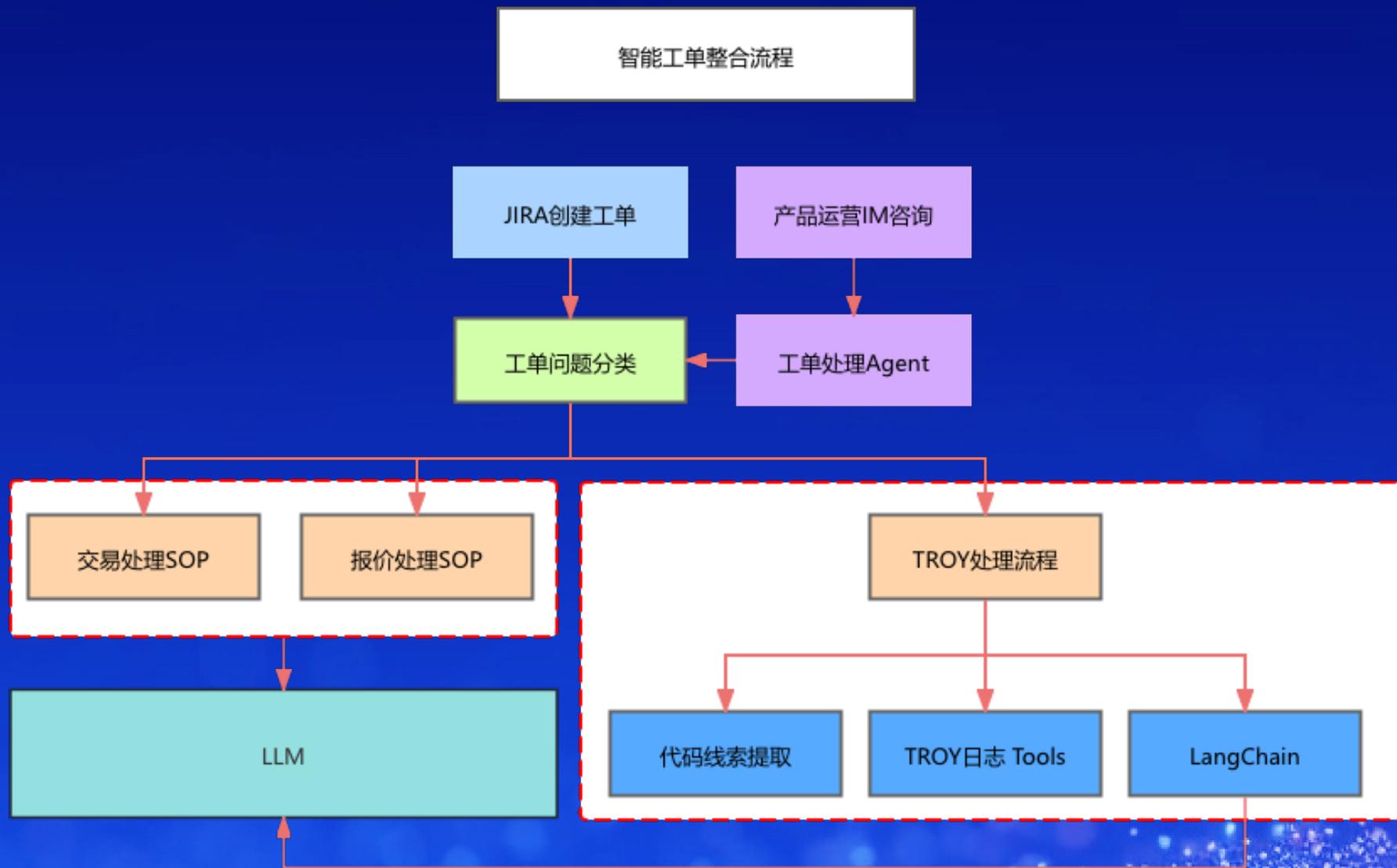
## 工作流:
1. 充分理解用户所提出的问题。
2. 关注日志中与问题相关的重点信息。
3. 理解给出的代码片段逻辑, 尽可能将这些片段逻辑联系起来思考, 这个过程注意参考代码中的注释。
4. 综合日志和代码排查问题, 在组织答案时尽可能参考代码中的注释, 以使回答的结论容易被普通用户理解。
5. 对回答行评估判定此回答普通用户是否好理解, 如不好理解则进行修订, 直至产出满意答案为止, 至此生成了初步答案。
6. 根据给定的原问题、问题的初步答案生成进一步的追问, 生成的追问是为了能进一步找到隐藏在代码或注释中的根因逻辑, 以便给原问题生成更好的回答。新产生的问题的答案必须也是原问题的答案, 同时也是初步答案的因。
示例:
  示例1
  ...
  原问题: 小明今天为什么没有参考早会?
  初步答案: 早会是8点, 小明9点才到公司。
  新生成的问题: 小明为什么9点才到公司导致他没来得及参加早会?
  ...
  示例2
  ...
  原问题: 为什么看不到ADT这家代理商的报价?
  初步答案: 这个代理商的报价不符合平台设定的规则。
  新生成的问题: 平台有什么规则导致看不到ADT这家代理商的报价?。
  ...

7. 结合原问题答案、代码逻辑和代码中的注释在兼顾原问题时尽可能对生成的追问作答, 最后整理所有内容按输出格式输出。

## 注意:
- 日志与代码关联的关键词, 从关键字开始按代码的执行流程向上分析。
- 给出的可能是多个代码片段, 它们一般是调用与被调用的关系, 你可以根据方法签名来理解它们之间的联系。
- 注意代码注释中的内容, 结合这些注释内容可以更好的以人类好理解的方式组织答案。

## 输出格式
分析完以JSON格式回答, 不要含有JSON外的其它内容, 好好检查一个JSON格式, 特别是引号和各种类似括号的对应关系, JSON属性包括以下内容
- foundReason: 是否已经找到问题的原因, 确定找到返回true, 未找到返回false, 没有相关日志时也返回false, 暂时没有完全确定但发现了线索也返回false;
- conclusion: 分析原结论, 组织结论的时候多多注意代码注释中的内容, 结合代码注释中提到的信息可能回答的内容更可能被用户所理解;
- foundClue: 是否找到线索;
- notice: 补充说明或注意事项, 可以为空;
- probeQ: 思考过程中产生的追问
- probeA: 追问对应的答案
- evidence: 如果isFinish为true, 此处填写日志中的相关证据, 指明出处, 如果isFinish为false, 此处归纳总结有意义的线索
, functionCall=null), Message (role=user, name=user1, content=## 问题:
```

▶▶ 方案介绍-3.0方案整合结果



PART 03

成果展示

▶ 成果展示-1.0版本的工单自动处理和结果通知

The screenshot displays a web interface for a ticket management system. The main content area is titled "全链路工单排查工具" (Full-link ticket troubleshooting tool). It shows the following information:

- Header: 机票目的地事业群 / FD-242327 异步生单失败原因
- Buttons: 编辑, 备注, 分配, 更多, 重新打开, 验证关闭, 变更申请
- Text: 是什么意思?
- Attachment section: 附件
- Structure section: 活动
- Activity list: 所有, 备注, 工作日志, 改动记录, 活动, Transitions, 流程跳转
- Activity details: transitioned issue - 2024-04-01 20:30
- Work action name: 问题 - 2024-04-01 20:30
- User action: skynet-pmo 添加了备注 - 2024-04-01 20:30

The central part of the interface shows the output of the "全链路工单排查工具":

- Header: 全链路工单排查工具
- Text: pmo (工单分析结果(工单自动分析任务执行正确, 请及时关闭工单))
- Text: pmo
- Text: 识别问题类型: ["生单拦截"]
- Text: 分析任务总数: 2
- Text: 分析任务状态: 总数: 2, 执行完成数: 1, 已提交数: 1, 失败数: 0
- Text: 识别分析结果详情
- Text: [issueKey=FD-242327](#)
- Text: [maintenance/pmo/processor/pmoProcessDetail?](#)
- Text: 手动查询分析结果链接: [maintenance/index#/overview/questionQuery?pmoId=FD-245088](#)
- Text: 手动提交分析链接: [maintenance/index#/overview/questionSubmit](#)
- Text: 消息来自于 appcode : f_pangu

At the bottom of the screenshot, there are two red boxes with arrows pointing to them:

- Left box: 工单结果回填 (Ticket result backfill)
- Right box: 工单处理通知 (Ticket processing notification)

工单结果回填

工单处理通知

成果展示-2.0版本的代码提取和工单回答

```
工单类型: 线上问题排查
业务线: 机票-国内交易
...
## 相关日志
...
[2024-06-17 12:16:21.105 QTraceId[ops_slugger_240617.121620.10.95.133.30.4140947.6898438399_1]-QSpanId[1.1.1] ... its.ttsord ... 5
[2024-06-17 12:16:21.105 QTraceId[ops_slugger_240617.121620.10.95.133.30.4140947.6898438399_1]-QSpanId[1.1.1.1] ... 5
[2024-06-17 12:16:21.112 QTraceId[ops_slugger_240617.121620.10.95.133.30.4140947.6898438399_1]-QSpanId[1.1.1] ... 1
[2024-06-17 12:16:21.105 QTraceId[ops_slugger_240617.121620.10.95.133.30.4140947.6898438399_1]-QSpanId[1.1.1] ... 5
...
## 相关代码片段
...
@QInterfaceDoc(
    ...
    url = "RouteOrderCreateService.createOrder"
)
@Override
public CreateOrderResponse createOrder(CreateOrderRequest createOrderRequest) {
    Stopwatch stopwatch = Stopwatch.createStarted();

    CreateOrderResponse createOrderResponse = CreateOrderResponse.failed(BASE_FAILED);
    SourceType sourceType = SourceType.MAP;
    FlightType flightType = FlightType.UNKNOWN;
    RouteOrderRequestContext requestContext = null;
    try {
        log.info("生单请求参数为: {}", JsonUtils.serialize(createOrderRequest));

        //获取最原始参数,风控用
        CreateOrderRiskOriginParam createOrderRiskOriginParam = buildCreateOrderRiskOriginParam(createOrderRequest);
        String preValid = routeRequestValidator.isValid(createOrderRequest);
        if (StringUtil.isEmpty(preValid)) {
            log.error("生单接口参数前置校验失败: {}", preValid);
        }
        ...
        ...
    } catch (Exception e) {
        log.error("生单接口参数前置校验失败: {}", e.getMessage());
    }
}

public String isValid(CreateOrderRequest createOrderRequest) {
    if (createOrderRequest == null) {
        return "生单请求参数不能为空";
    }

    if (createOrderRequest.getUserClientInfo() == null) {
        return "客户端信息不能为空";
    }

    if (createOrderRequest.getBookingInfo() == null) {
        return "预定信息不能为空";
    }

    int flightType = createOrderRequest.getBookingInfo().getFlightType();
    if (FlightType.UNKNOWN == FlightType.fromCode(flightType)) {
        return "航程类型不合法";
    }

    if (CollectionUtils.isEmpty(createOrderRequest.getPassengerInfos())) {
        return "乘机人信息不能为空";
    }
}
```

在代码中有一个检查逻辑，确保如果预定信息中存在非成人乘客并且对应的'bookingMultipleBookingTag'字段被填写，则该订单会被拒绝。这是为了避免系统风险，因为可能存在法律或政策上的限制，不允许未成年人不在成人陪同下进行机票购买。

▶ 成果展示-3.0版本工单智能体会话



用户体验后服务AI助手

-----哈喽,我是用户体验与后服务AI助手,我可以为你提供以下服务-----

- 1.聊天对话
- 2.文本向量提取
- 3.用户行为分析
- 4.qc往返劣势分析
- 5.sql自动生成
- 6.航班动态问题处理



用户体验后服务AI助手

如果咨询有关用户航班的问题, 需要提供订单号/用户名/Qtrace/航班号, 以及用户的访问时间 (因为用户可能存在多次访问记录, 所以尽量精确到以小时为维度的时间区间)。#比如: 订单号333745996113, 他在6月20日19点-20点访问了航班动态页, 用户是否展示了起飞时间19点09分, 展示的数据源来自于哪里

消息来自于appcode: f_poseidon_aigc_server



航班信息查询 订单号333745996113, 他在7月7日19点-20点访问了航班动态页, 用户是否展示了值机柜台, 展示的数据源来自于哪里



用户体验后服务AI助手

正在处理, 请耐心等待

消息来自于appcode: f_poseidon_aigc_server



用户体验后服务AI助手

用户在7月7日19点至20点访问航班动态页时, 系统显示了值机柜台信息。值机柜台的数据显示为"A09", 这个信息来自于通过dubbo接口查询到的国际航班动态信息, 其数据源是基础数据的dubbo接口。

消息来自于appcode: f_poseidon_aigc_server

工单自动化整体提效数据

WORK TARGET



**下降
45%**

值班工时

机票研发整体值班
工时下降



38%

自动化处理率

无开发参与的工单
处理占比



**最高下
降100%**

工单量

部分业务线工单清
零

PART 04

总结和展望



数据重要性

企业自身数据治理水平影响大模型的效果上限



从总结到推理

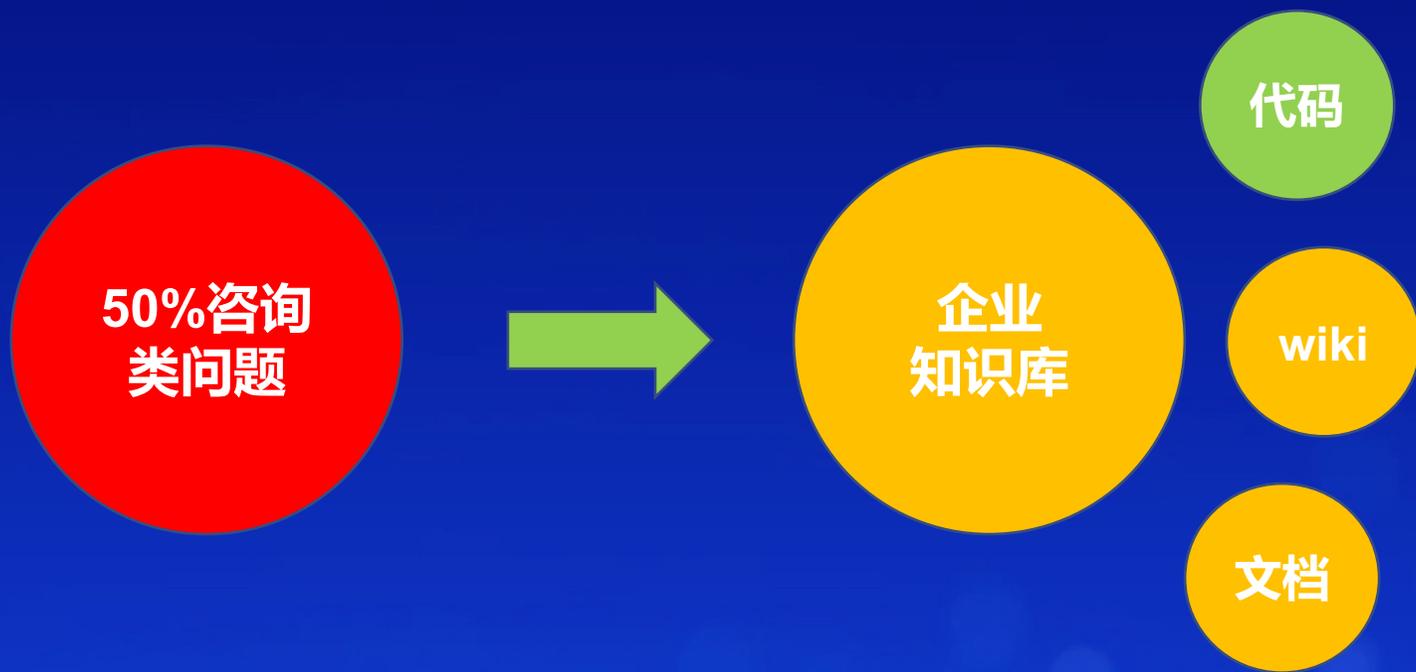
对大模型的能力使用由浅入深，参数从小变大



从工具到助手

将工具助手化，降低使用门槛和成本

工单自动化实践总结



更大占比的咨询类工单
需要
更完善的企业知识库建设



业务研发团队转型AIGC的准备

☰ 目录

- Java工程师步入AIGC开发指南-知识图谱
- Java工程师步入AIGC开发指南-LlamaIndex
- Java工程师步入AIGC开发指南-GPT-4o
- Java工程师步入AIGC开发指南-Tools优化**
- Java工程师步入AIGC开发指南-RAG
- Java工程师步入AIGC开发指南-memory
- Java工程师步入AIGC开发指南-Agent框架介绍
- Java工程师步入AIGC开发指南-LLM长思考方案介绍
- Java工程师步入AIGC开发指南-微调技术介绍
- Java工程师步入AIGC开发指南-GPT-4V提示词指南（下）
- Java工程师步入AIGC开发指南-GPT-4V提示词指南（上）
- Java工程师步入AIGC开发指南-LangSmith
- Java工程师步入AIGC开发指南-langchain
- Java工程师步入AIGC开发指南-提示词工程

去哪儿机票研发内部培训资料

相较于C端落地AIGC，在B端落地有更高的确定性，也更容易看到效果和收益。

科技生态圈峰会 + 深度研习



—1000+ 技术团队的选择



 **K+峰会**  **敦煌站**

K+ 思考周®研习社

时间: 2025.08.29-30

 **K+峰会**  **上海站**

K+ 金融专场

时间: 2025.10.17-18

 **K+峰会**  **香港站**

K+ 思考周®研习社

时间: 2025.11.25-26



K+峰会详情



 **AiDD峰会**  **上海站**

AI+研发数字峰会

时间: 2025.05.17-18

 **AiDD峰会**  **北京站**

AI+研发数字峰会

时间: 2025.08.08-09

 **AiDD峰会**  **深圳站**

AI+研发数字峰会

时间: 2025.11.28-29



AiDD峰会详情



利用AI技术深化计算机对现实世界的理解

推动研发进入智能化时代

