

AI 驱动 软件研发 全面进入数字化时代

中国·北京 08.18-19

AI+
software
Development
Digital
summit



AI+低代码在金融行业软件工程的实践

李方 普元信息技术有限公司

科技生态圈峰会 + 深度研习 —— 1000+ 技术团队的选择



2023K+
全球软件研发行业创新峰会
上海站

会议时间 | 06.09-10



2023K+
全球软件研发行业创新峰会
北京站

会议时间 | 07.21-22



2024K+
全球软件研发行业创新峰会
深圳站

会议时间 | 05.17-18



K+峰会详情



会议时间 | 08.18-19

AiDD AI+软件研发数字峰会
北京站



会议时间 | 11.17-18

AiDD AI+软件研发数字峰会
深圳站



AiDD峰会详情

▶ 演讲嘉宾



李方

普元金融解决方案研发中心首席方案专家

12年金融行业从业经验，曾就职于神州信息、德勤等多家金融科技服务或咨询公司。参与多家城市商业银行核心等业务系统的建设工作，以及某政策性银行新一代新核心业务系统的咨询工作；在业务领域，对银行的核心业务包括存贷支付等都有较为深入的了解；在技术领域，对目前分布式、云原生等在金融领域的应用也有一定的了解。目前主要从事金融相关解决方案的研发、售前等工作。

目录

CONTENTS

1. **AI为低代码的前进提供驱动力**
2. **AI+低代码在金融行业的实践**
3. **AI+低代码的未来展望**

PART 01

AI为低代码的前进提供驱动力

▶ 程序员失业了吗？

当托拉拽出现的时候，人们说程序员会失业？

当低代码出现的时候，人们说程序员会失业

Copilot出现的时候，人们说程序员会失业？

如今ChatGPT火爆的时候，人们说程序员会失业？



▶ “它” 是怎么看待AI+低代码的组合的

PH

AI+低代码



AI+低代码是一种强大的组合，将人工智能和低代码开发相结合，为应用程序开发带来了前所未有的效率和创新力。通过集成AI技术，低代码平台能够提供更智能化的开发工具和功能，进一步简化开发流程并提升应用程序的质量和功能。



AI在低代码开发中的应用有多方面的好处。首先，AI可以辅助开发人员自动生成代码、设计用户界面和优化应用逻辑，减少了手动编写代码的工作量。其次，AI能够分析大量的数据和用户行为，提供智能的建议和决策支持，帮助开发人员做出更准确的设计和优化选择。此外，AI还能够实现自动化测试、缺陷检测和错误修复，提高应用程序的质量和稳定性。

AI+低代码的组合还能够推动创新和个性化的应用程序开发。通过AI的能力，开发人员可以构建智能化的应用程序，实现自动化的推荐系统、智能搜索、语音识别和图像处理等功能。这使得应用程序能够更好地满足用户需求，提供个性化的体验和增强的交互性。

▶ AI在低代码开发中的定位

**Work with you,
not for you**

智能助手



合作伙伴



智能协同

交互变革带来软件革命：所有企业软件会使用人工智能方式重写

Windows带来的软件革命

- 交互：从文本字符到鼠标与GUI点击
- 生态：形成了开放的软件、硬件生态

互联网带来的软件革命

- 交互：从桌面端的GUI交互扩展为手机、平板等多模态交互
- 生态：围绕软件分层技术栈的开源和商业软件体系

智能手机带来的软件革命

- 交互：从鼠标点击到点击/滑动/缩放等触摸操作
- 生态：苹果形成了软件生态

企业软件还是以PC为主，影响不像Windows那样大

AIGC带来的软件革命

- 交互：以语言交流为主，触摸为辅
- 生态：围绕大模型的插件生态

▶ 金融企业应用低代码的复杂性

安全与开放

- 内外网络隔离
- 敏感信息保护
- 集成
- 被集成

合规与灵活

- 数据变更复核
- 操作审计
- 快速反应
- 事前预警

▶ AI不是简单交互，更需要将其无感知融入产品中



创建会话: 微软chatGPT 讯飞chatGPT

文档索引

微软服务

AI测试

第一个会话

Solr可以存储附件，但是存储附件需要使用Solr的Binary Uploading功能，将附件转换为二进制流并上传到Solr的字段中。为了从附件中提取文本并将其添加到Solr索引中，需要使用Solr的提取器 (Extractor) 功能或使用tika提取文本并发送到Solr。当需要使用Solr存储附件时，需要将Solr配置为具有合适的字段类型和提取器。

2023-05-25 09:37:11 0.046K

solr的中文分词插件如何选择和使用

2023-05-25 09:37:52 1.2K

Solr目前支持多种中文分词插件，常用的包括：

1. IK Analyzer: 性能较好，支持细粒度和智能分词，能够处理中文字符、日期、姓名等复合词。使用方法：下载IK Analyzer插件，将相关Jar包放入Solr的lib目录下，修改schema.xml配置中的FieldType，指定analyzer为IK Analyzer。
2. SmartCN Analyzer: Lucene内置的中文分词器，使用简单，性能一般。使用方法：Solr版本高于6.x时，无需额外安装和配置，只需修改schema.xml配置中的FieldType，指定analyzer为SmartCN Analyzer。
3. HanLP Analyzer: 支持智能分词、命名实体识别、关键词提取等多种NLP功能，但性能较差，适合针对场景对分词进行深度定制。使用方法：下载HanLP Analyzer插件，将相关Jar包放入Solr的lib目录下，修改schema.xml配置中的FieldType，指定analyzer为HanLP Analyzer。

选择对应的分词插件时，需考虑其性能、功能和易用性等因素，同时结合自己的应用场景进行选择。安装和使用插件的方式大多相同，主要是将相关Jar包存放至Solr的lib目录，然后在schema.xml中配置对应FieldType的analyzer即可。

2023-05-25 09:38:48 0.05K

选择IK Analyzer，请给出具体的配置示例

2023-05-25 09:39:19 0.869K

IK Analyzer是一个流行的中文分词器，可以在Elasticsearch、Solr等搜索引擎中使用。以下是一个基本的IK Analyzer配置示例：

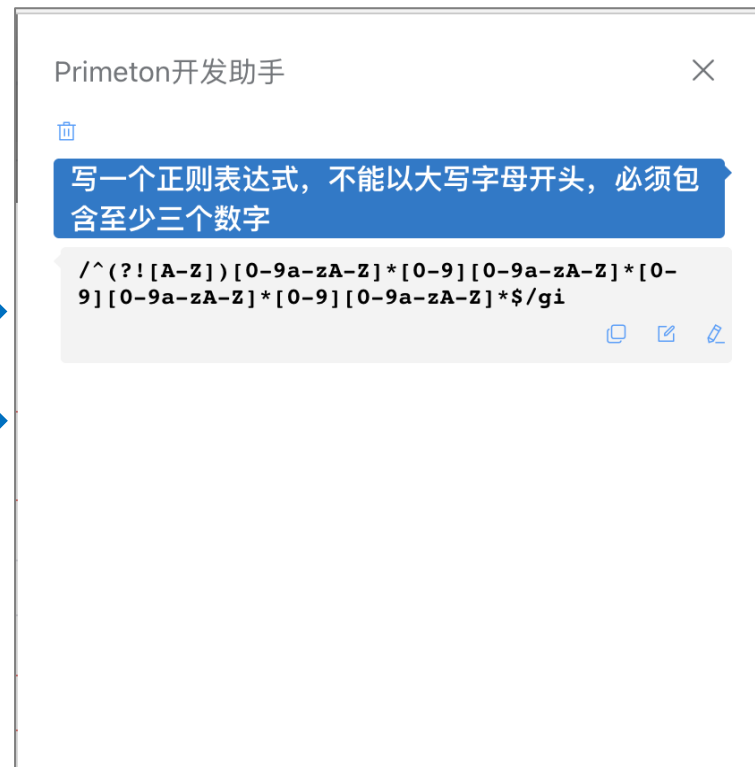
```
curl -XPUT http://localhost:9200/index_name -H 'Content-Type: application/json' -d '{
  "settings": {
    "analysis": {
      "analyzer": {
        "ik_analyzer": {
          "tokenizer": "ik_max_word"
        }
      }
    }
  }
}
```

开启上下文

Enter提交问题，Shift+Enter换行

每日最多可与小助手对话100次

每日与小助手对话内容限制为100K



Princeton开发助手

写一个正则表达式，不能以大写字母开头，必须包含至少三个数字

```
/^(?![A-Z])[0-9a-zA-Z]*[0-9][0-9a-zA-Z]*[0-9][0-9a-zA-Z]*[0-9]*/gi
```

探索未知

融入生产



知识融合

无感拥抱

▶ 几大AI在软件工程中的应用场景

AI生成需求

1

通过AI能力进行需求的分析以及需求的结构化拆分

AI进行工作量评估

2

基于拆分的需求进行需求的工作量评估，精准把控进度

AI生成测试用例

3

通过AI进行规则解析，生成全面的测试用例

AI处理页面

4

提取中文资源，自动国际化，增加特殊校验器

自助数据分析

5

测试数据准备，自助图表分析

其他功能浏览

6

流程、逻辑、脚本、权限等低代码开发高级功能说明

PART 02

AI+低代码在金融行业的实践

▶ 低代码使用适应不同场景的能力要求

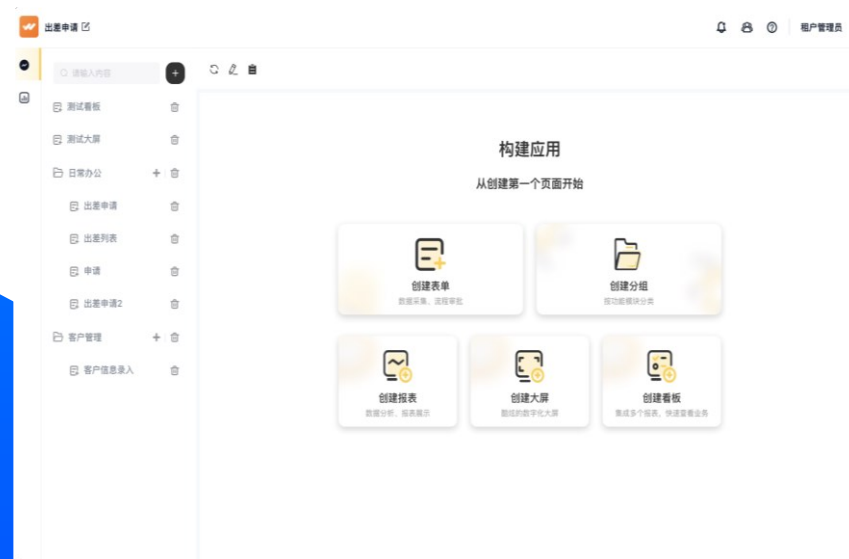
场景类型	零代码平台	低代码平台
轻应用系统	<ul style="list-style-type: none"> ✓ 信息收集, 如问卷调查 ✓ 数据管理, 如资产管理 ✓ 流程审批, 如办公采购 	<ul style="list-style-type: none"> ✓ 信息门户 ✓ 知识库 ✓ 业务人员使用
轻业务系统	<ul style="list-style-type: none"> ✓ 信息收集, 如问卷调查 ✓ 数据管理, 如资产管理 ✓ 流程审批, 如办公采购 	<ul style="list-style-type: none"> ✓ 信息门户 ✓ 知识库 ✓ 业务人员使用
轻业务系统	<ul style="list-style-type: none"> ✓ 页面集成, 如企业门户 ✓ 数据集成, 如CRM、HRM ✗ 服务集成, 如银行中间业务 	<ul style="list-style-type: none"> ✗ 流程中心 ✓ 应用独立部署
专业系统	<ul style="list-style-type: none"> ✓ 页面集成, 如企业门户 ✓ 数据集成, 如CRM、HRM ✓ 服务集成, 如银行中间业务 	<ul style="list-style-type: none"> ✓ 流程中心 ✓ 应用独立部署
专业系统	<ul style="list-style-type: none"> ✗ 数据模型驱动, 如MES、ERP ✗ 源代码生成 ✗ 专业代码工具融合 	<ul style="list-style-type: none"> ✗ 分布式架构 ✗ DevOps支撑 ✗ 应用治理
	<ul style="list-style-type: none"> ✓ 数据模型驱动, 如MES、ERP ✓ 源代码生成, 离线维护 ✓ 专业代码工具融合 	<ul style="list-style-type: none"> ✓ 数据模型驱动, 如MES、ERP ✓ 源代码生成, 离线维护 ✓ 专业代码工具融合



低代码平台的愿景

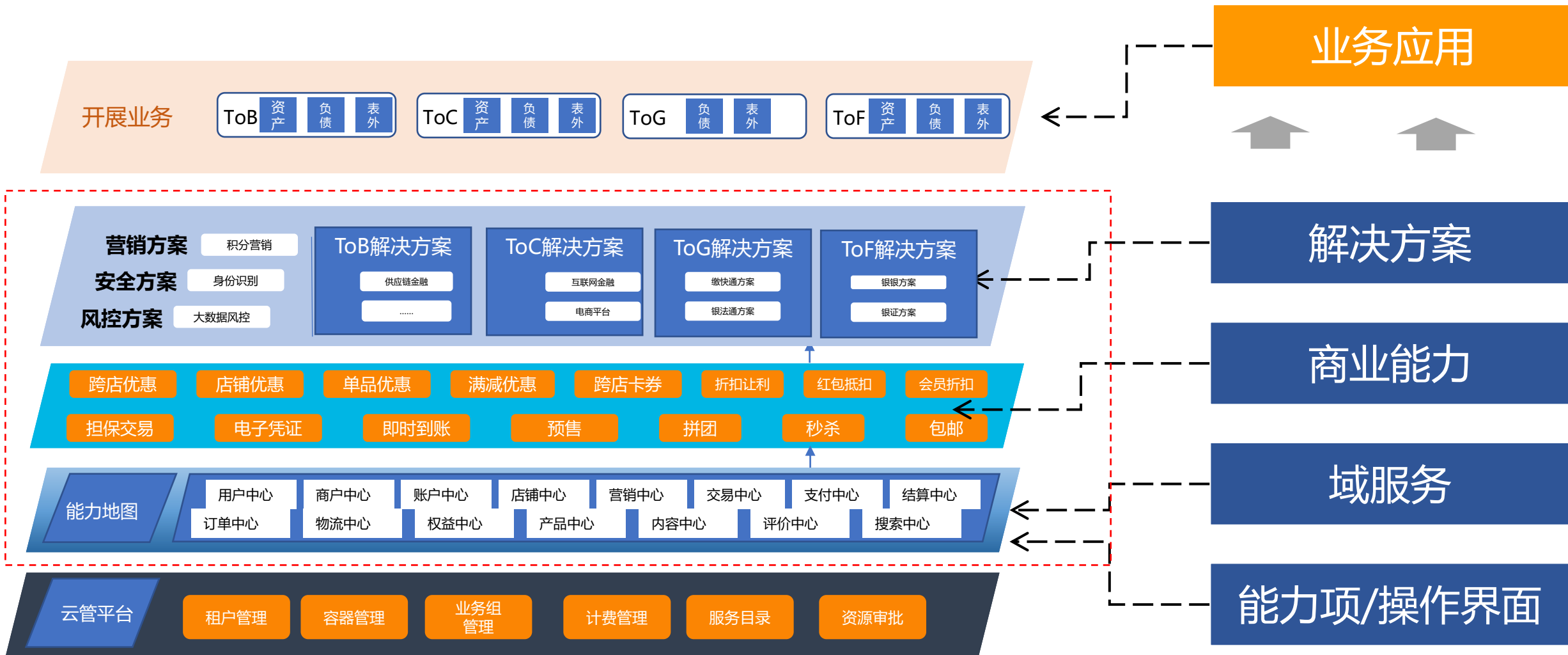
“模型驱动开发的理念，结合AIGC能力，帮助企业低成本高效建设复杂业务系统。”

“支撑行业零代码方案孵化，帮助合作伙伴聚焦领域零代码平台建设”



以低代码的方式，打造企业数字化工作场所，智能协作、贯通业务

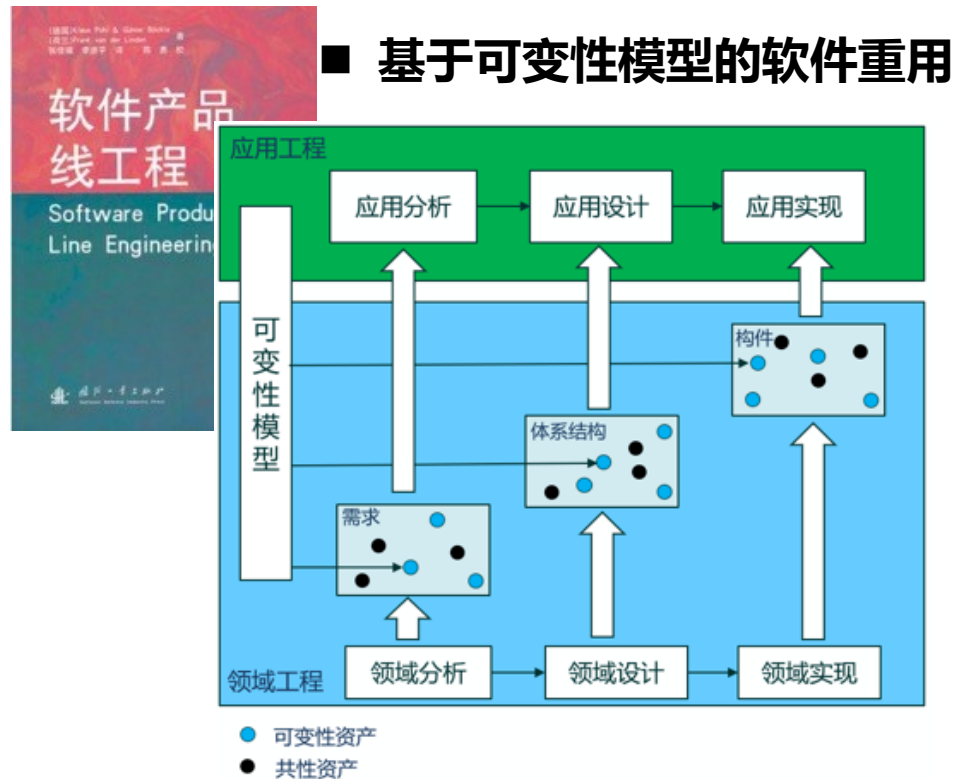
低代码的终局是“组装式应用”



▶ 低代码平台，覆盖企业软件智能生产的完整旅途



需求分析：基于可变性模型的软件重用



现在



配置化开发实现应用工程



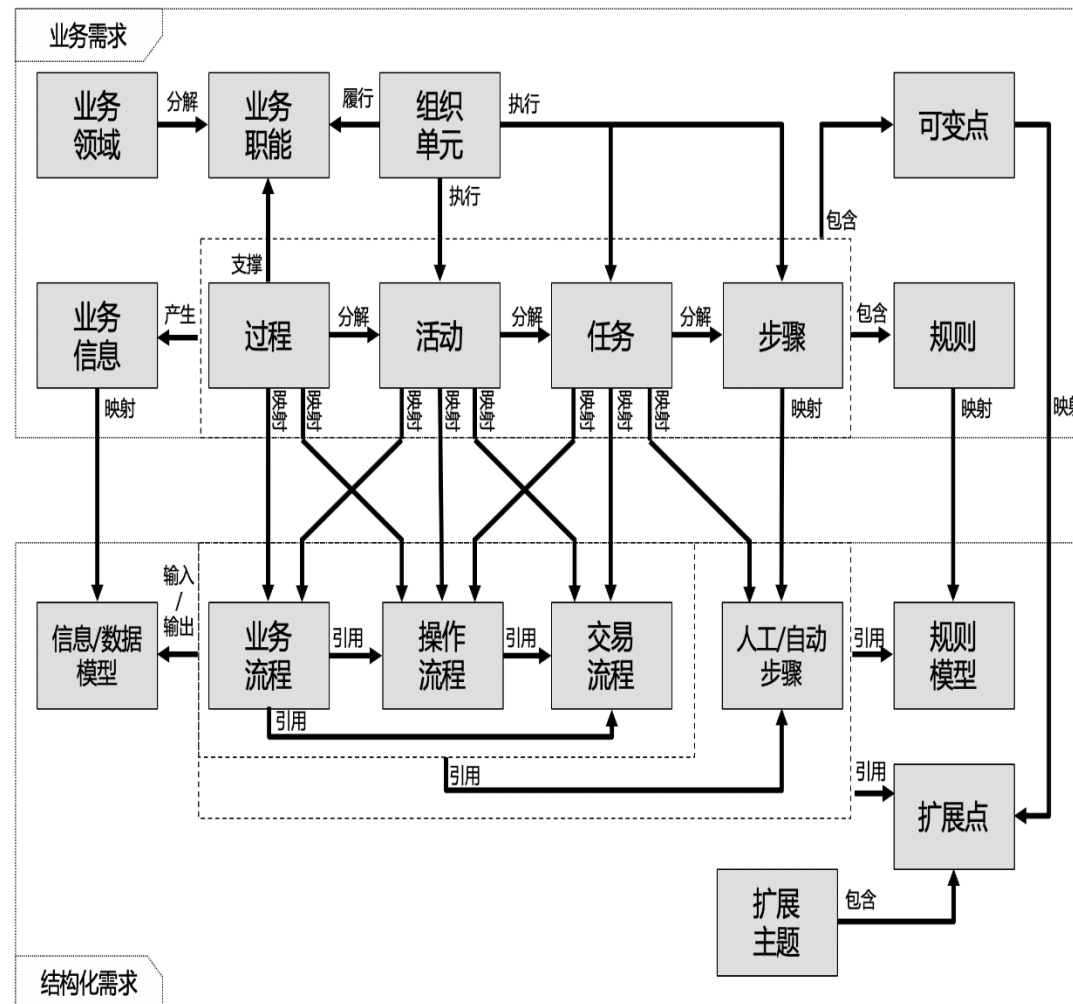
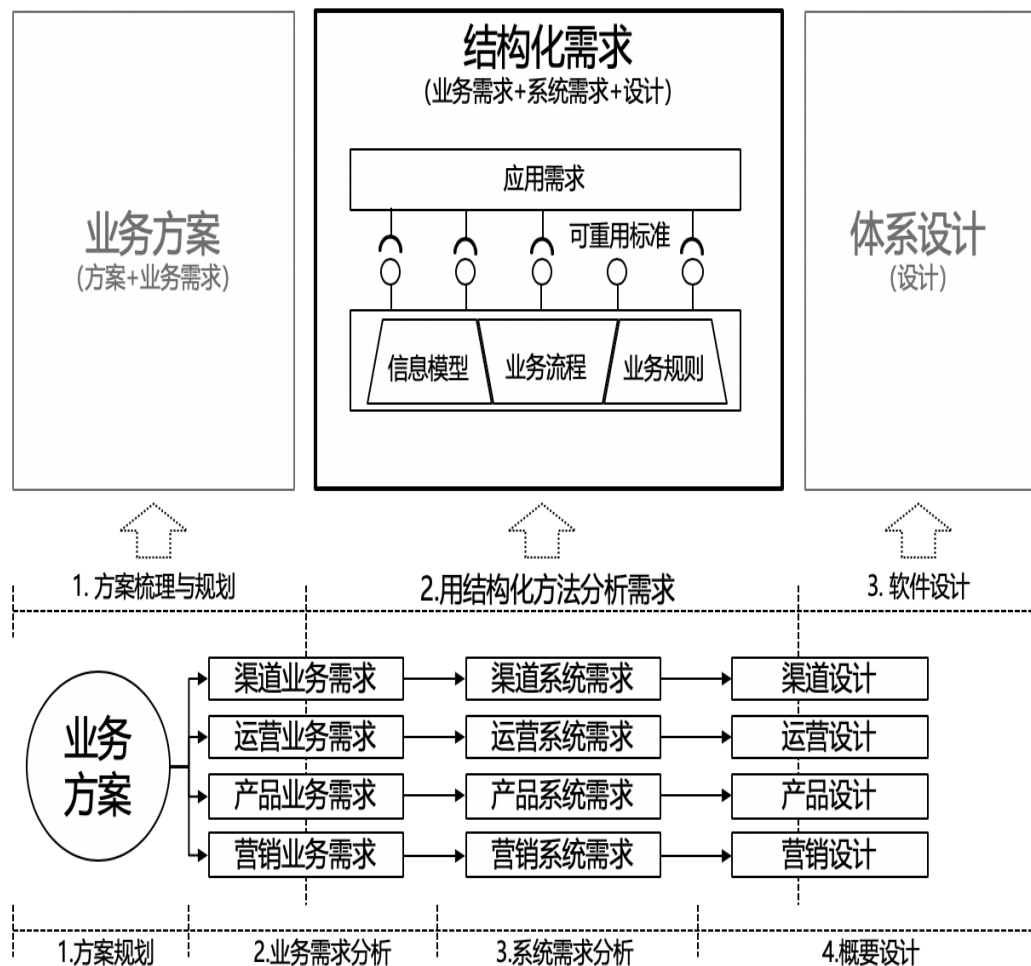
复用



代码开发实现领域工程

- 采用软件产品线工程理论实现软件可配置
 - ✓ 分为领域工程与应用工程，中台建设属于领域工程
 - ✓ 定义可变性
- 转变业务实现方式
 - ✓ 由开发人员完全代码开发实现转变为开发人员代码开发实现领域工程和业务人员配置化开发实现应用工程

需求分析：基于重用思想的需求结构化方法



需求分析：应用的组装和中台架构的构建

前台

线上客户
服务渠道应用

线下客户
服务渠道应用

合作伙伴
服务渠道应用

内部员工
应用渠道

中台

解决方案

解决方案1

解决方案2

业务流程



基础能力

数据



基础能力

数据



基础能力

数据



基础能力

数据



后台

账户服务

产品服务

支付/结算

风险管理

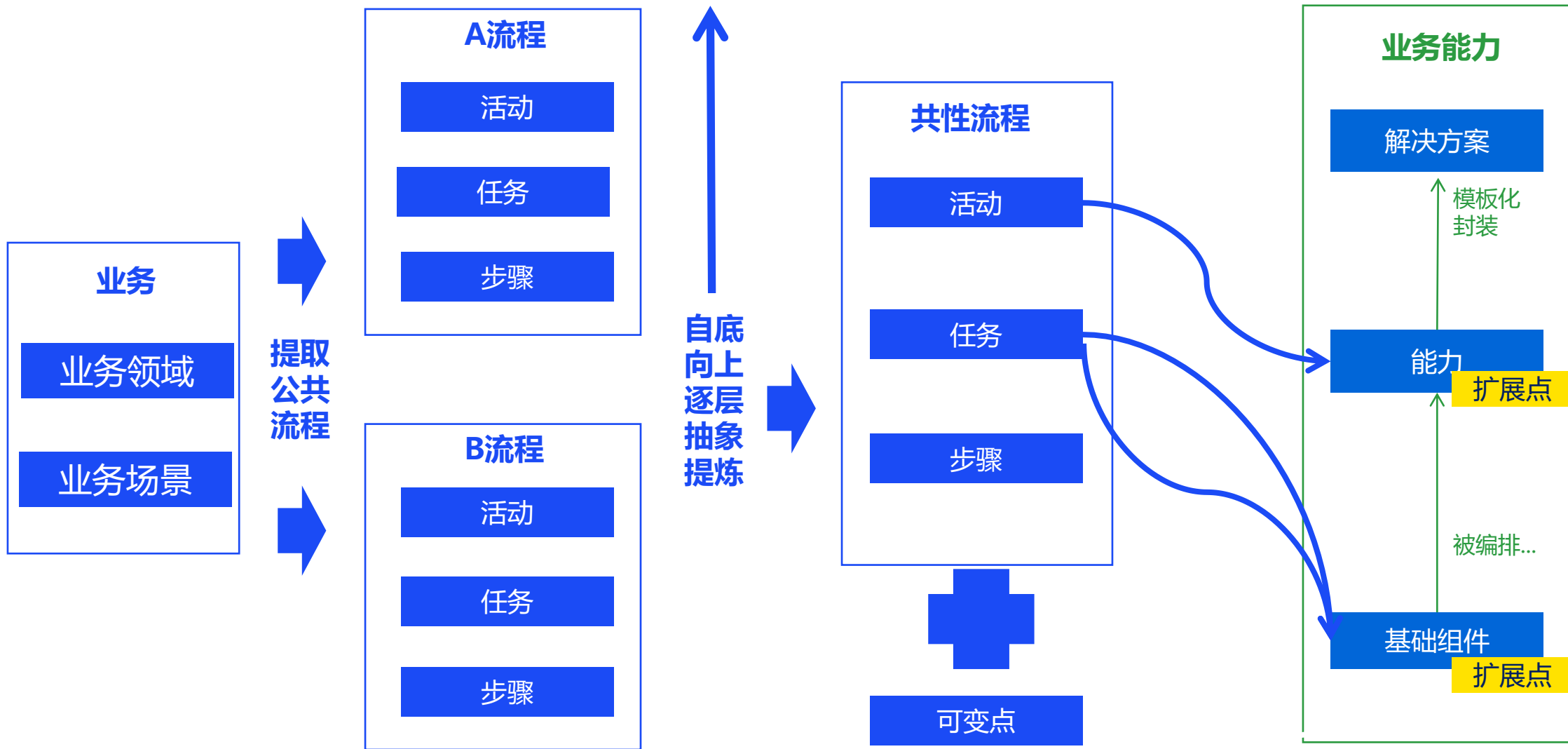
财务管理

监管

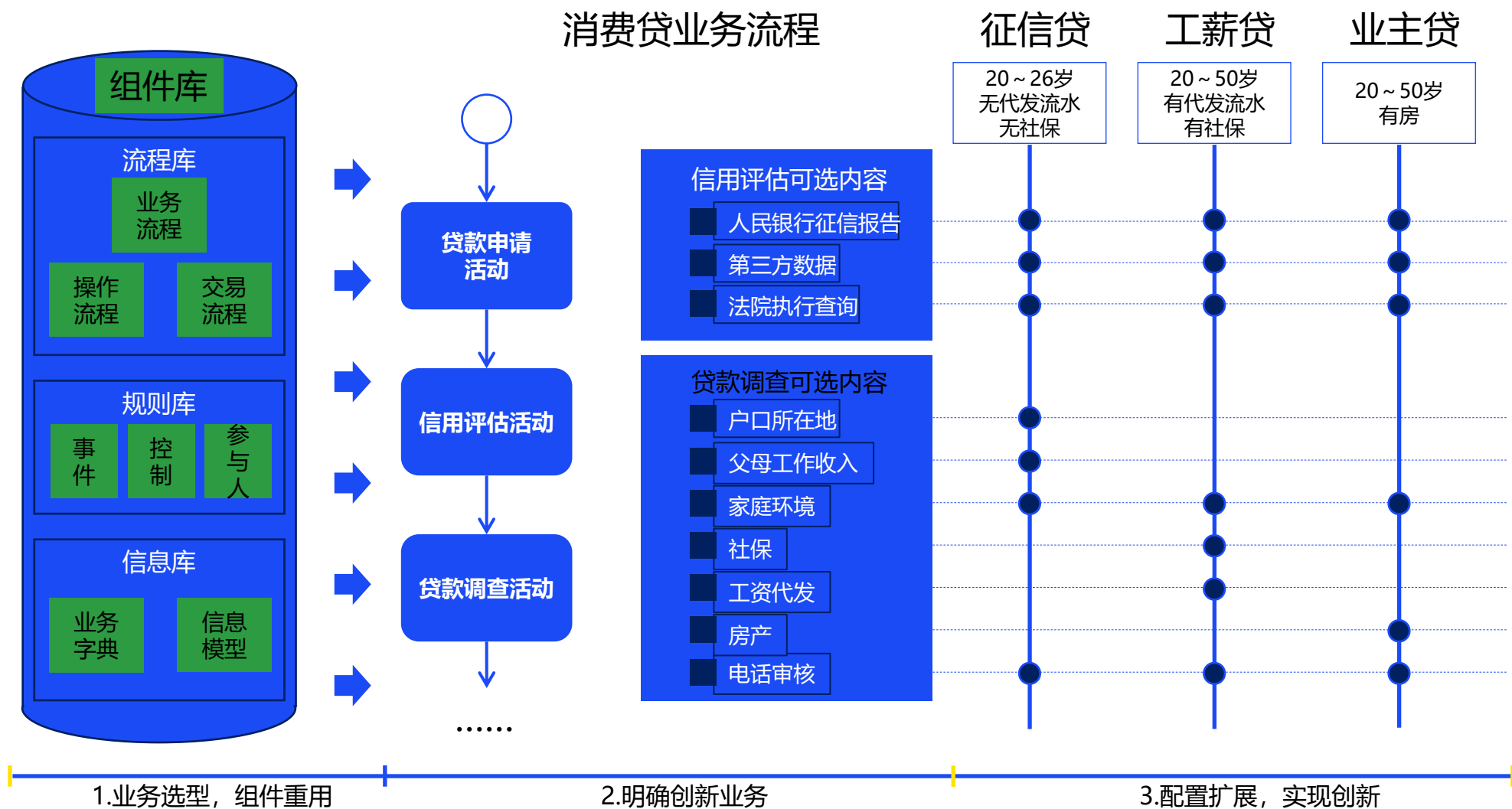
内部服务
支持

...

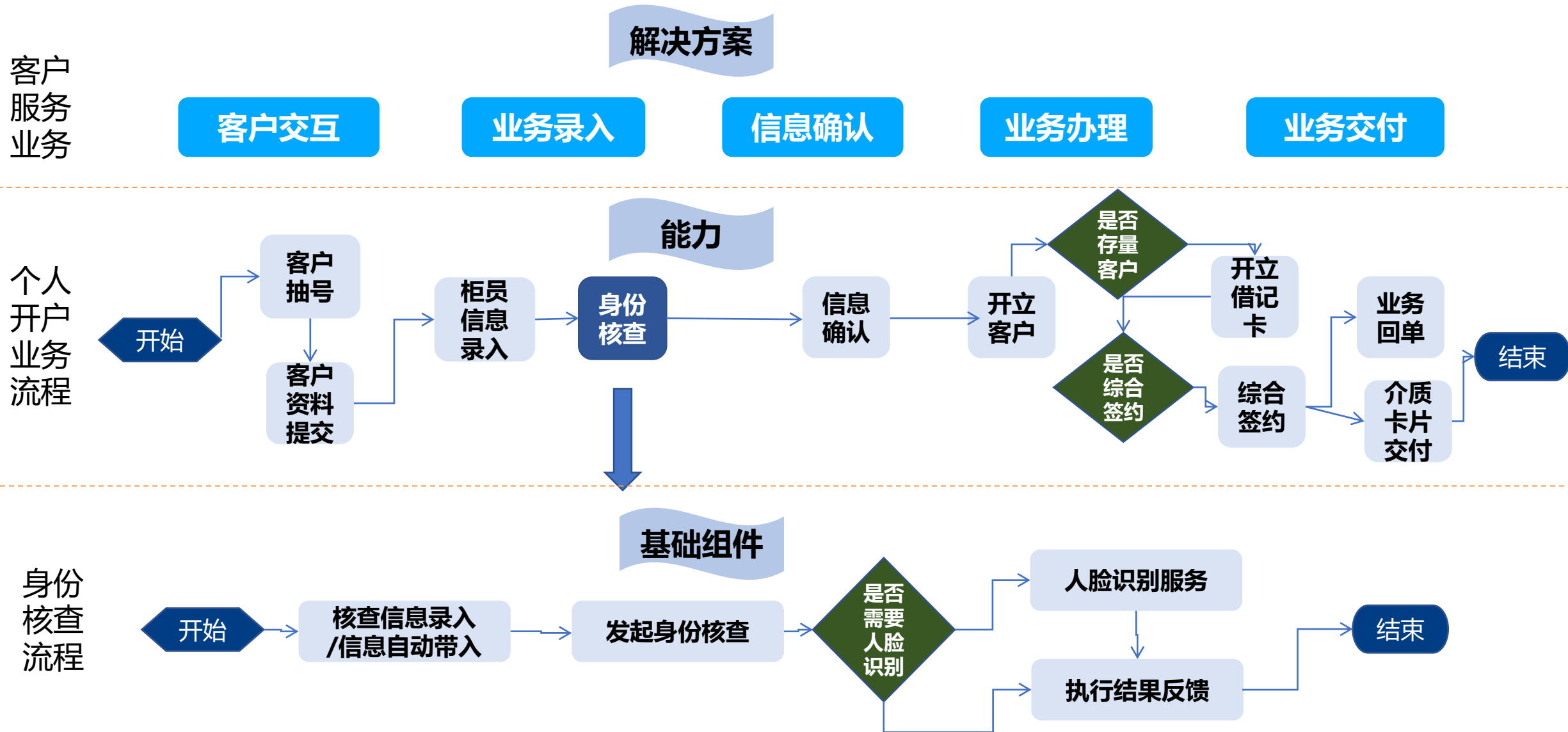
需求分析：核心在于共用组件的抽取



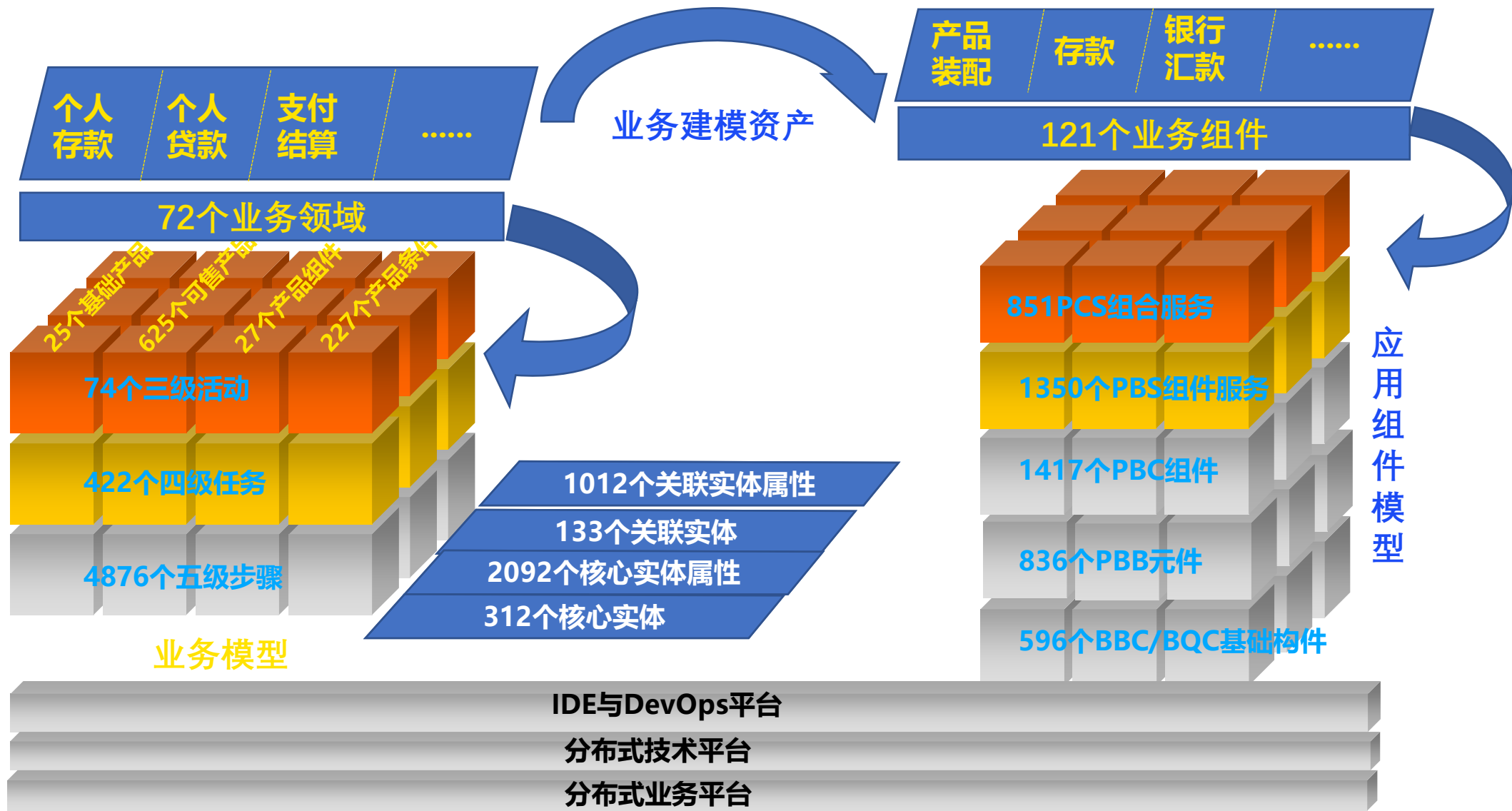
需求分析：基于可变性分析的消费贷业务流程



需求分析：业务组装示例

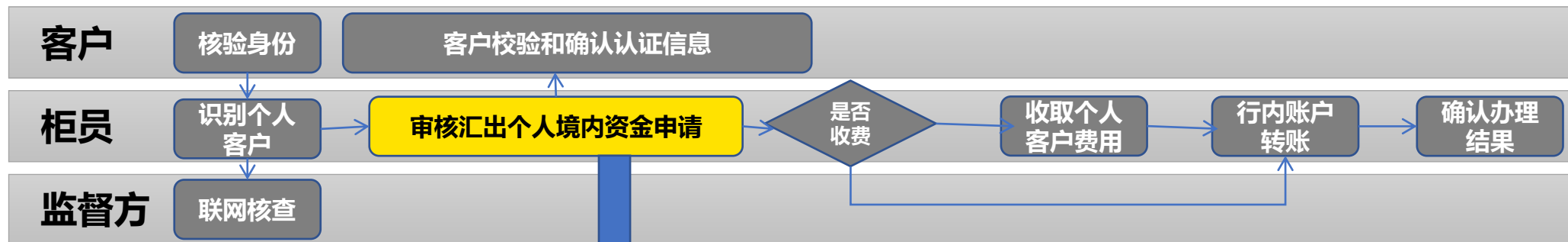


某银行通过组装式方式实现核心业务系统的建设

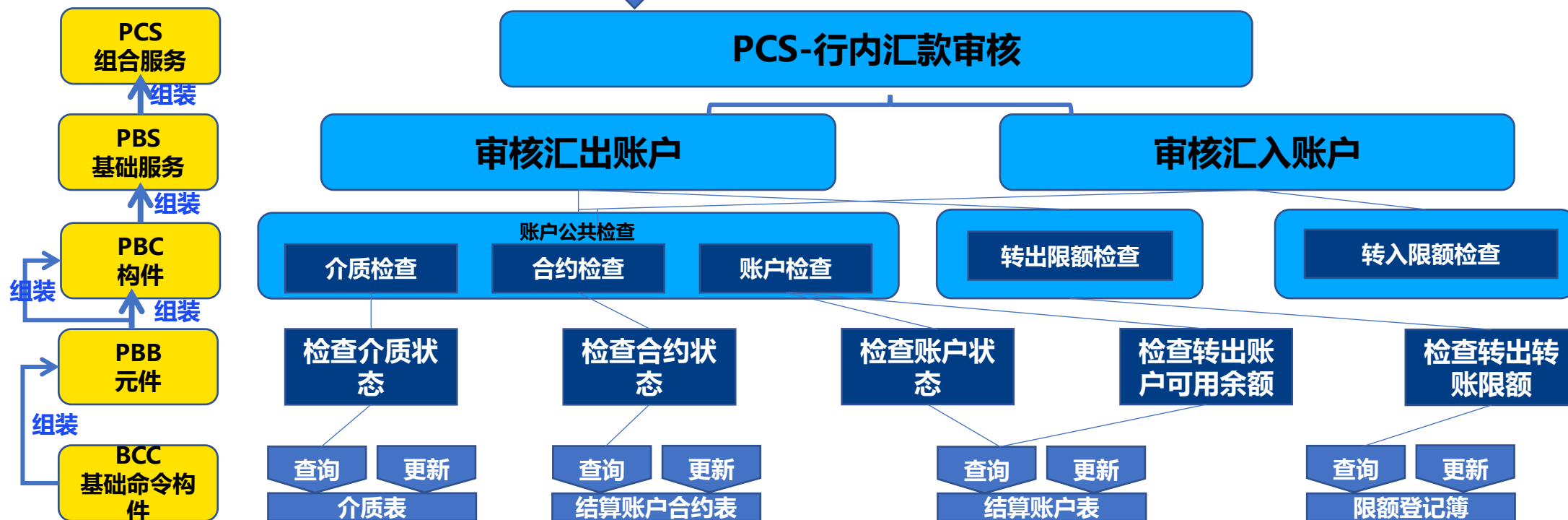


个人结算账户行内汇款流程示例

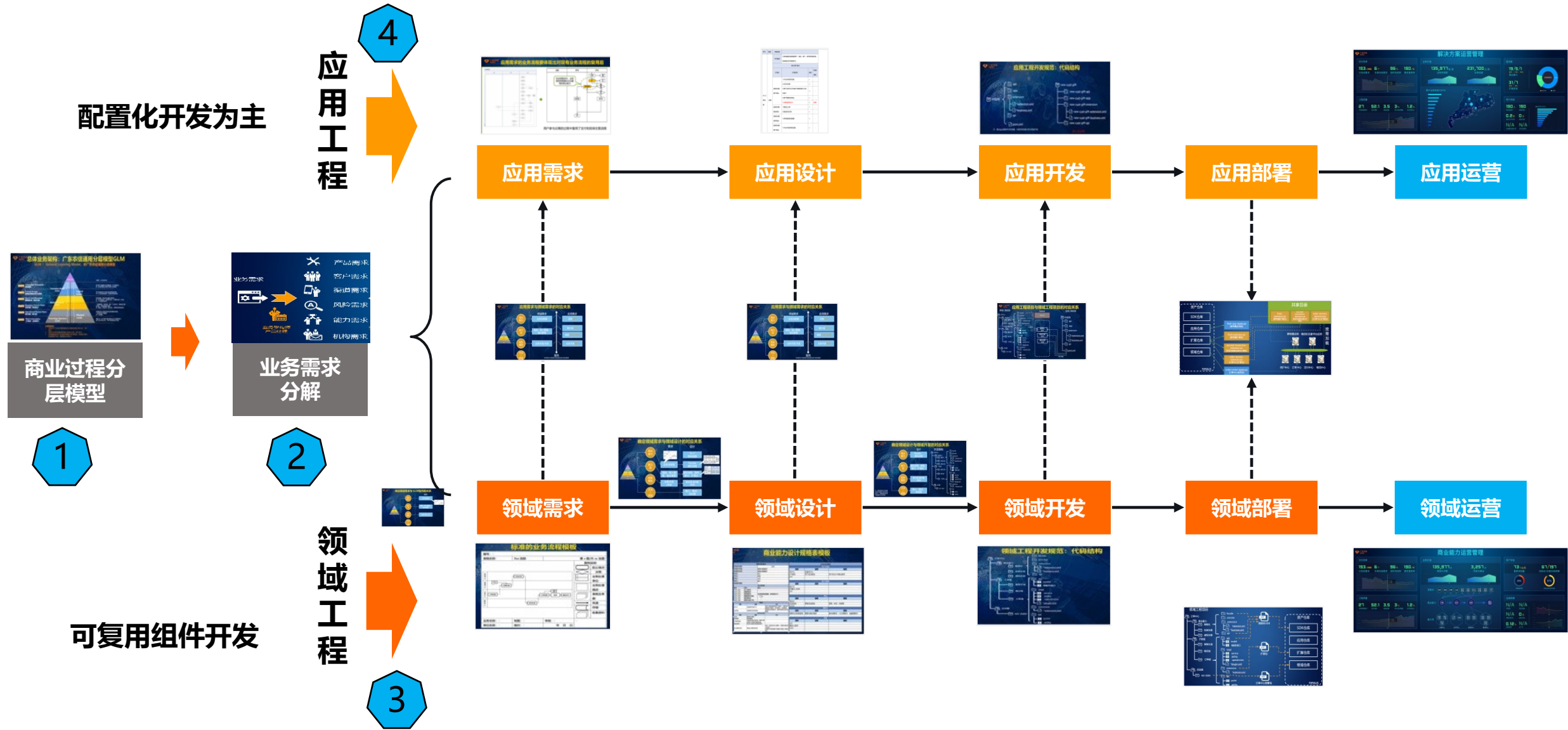
流程示例



应用组装示例



需求分析：生产从代码开发到领域 + 应用的开发模式



需求分析：AI助力需求文档的智能化生成

访谈时间：2023-03-21 21:30-23:30

访谈对象：浦发银行北京银行理财经理 Ruby，女，从事银行理财工作有 8 年之久，招行 6 年，浦发 2 年。

访谈方式：线上微信语音

访谈内容：

一. 总体用户体验上，Ruby 对招行的系统感受很好，对浦发的系统感受很差。

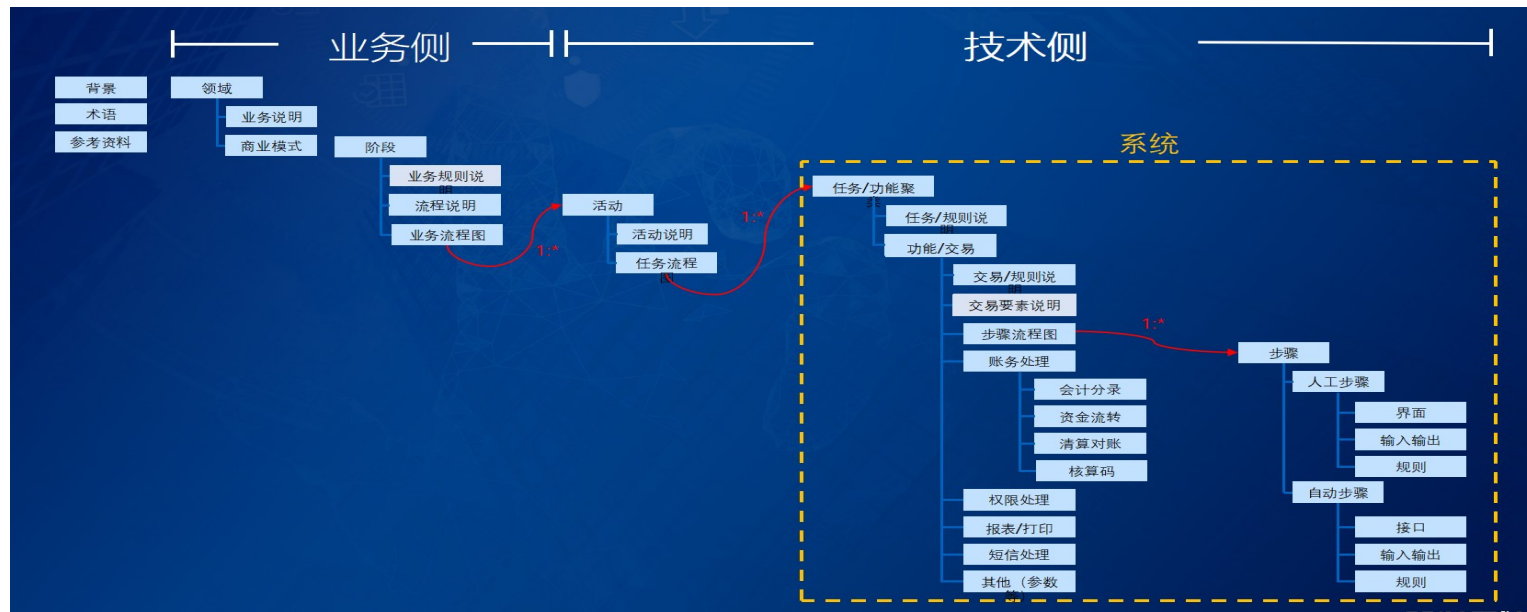
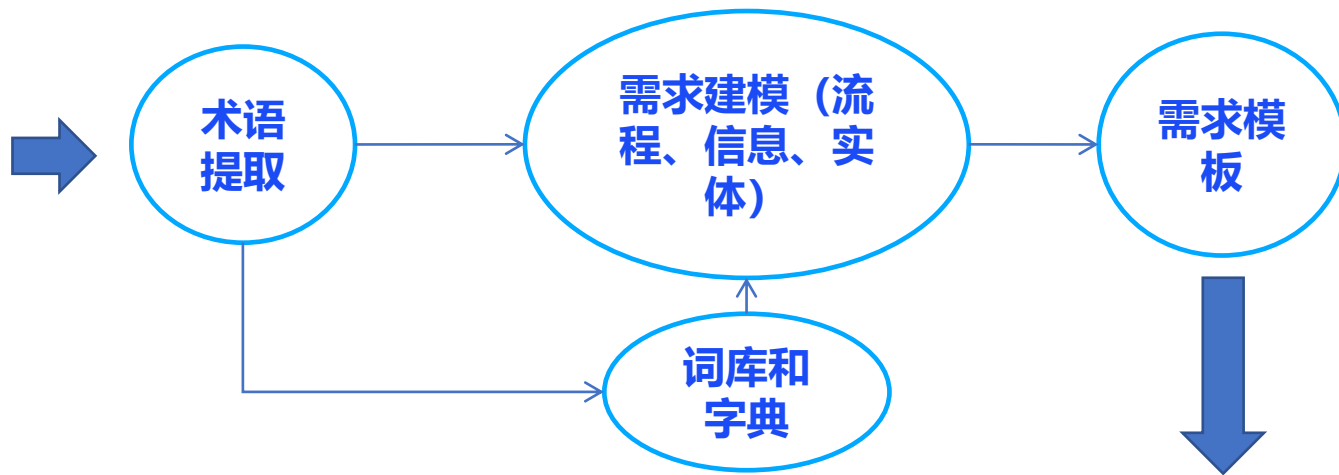
1. 对招行的系统感受很好。一是体现在招行的系统数据相对集中，能够满足理财经理日常工作需要。Ruby 介绍理财经理的日常工作中，一部分是客户性工作，占比很大，一部分是碎片性工作，琐碎零散又必须要做的事。招行的客户性工作体验很好，每天上班第一件事，打开系统就可以看到当天需要处理的所有客户性事务，进入每个客户信息界面，都有该客户的全部信息，有待处理，已处理，客户画像等，理财经理在处理客户事务时可以更新该客户画像，也可以查看其他渠道反馈的客户更新历史信息。手机和电脑端展现形式符合理财经理工作需要预期，招行员工普遍反响很好。二是体现在招行的系统访问不会卡顿，很顺畅。7年前会有卡顿，但近几年基本没有卡顿现象出现，一般在 5 秒内都会响应完成。

2. 对浦发的系统感受很差，一是体现在客户相关数据分散，不够集中。客户相关数据丰富度相差不大，该有的数据银行都有，就是日常处理客户性工作，需要登陆不同的系统或进入不同的菜单页面，查找所需要的客户数据，需要手工计算理财收益率等指标数据，比如系统有提供 1 个月、3 个月、6 个月的收益率，遇到 9 个月就需要手工来计算。二是浦发的系统访问经常卡顿，一个页面有时要 5 分钟才能打开，极大影响工作效率和心情。

二. 工作产量上，招行日处理客户数 15 个很轻松，浦发 10 个客户是极限。

三. 工作产能上，招行理财经理一年可为银行在中间业务收入创收 150-200 万，浦发达不到 100 万。

四. 业绩排名上，招行的业绩排名是按分行进行滚动，浦发是按全行进行滚动，前者给理财经理压力很大，后者基本无压力。



需求分析：符合行业特点规则模型提炼

规则

约束型

约束是对业务事件行为的强制约束或建议约束。

如：客户一次的未结订单不得超过10个。客户订单的总金额不得大于客户的单订单信用额度金额。

指导方针型

对测试条件为真或不为真的情况表示警告。

如：客户一次的未结订单不应超过10个。

启动业务活动型

测试条件为真时启动另一个业务事件、消息或其他活动。

如：如果客户订单有效，则启动下单流程。如果客户是高风险客户，则通知客户服务经理。

计算型

提供了一种算法，用于计算一个项的值，这种算法可能包括和、差、积、商、计数、最大值、最小值、平均值。

如：订单的应付总金额计算为订单的行项目金额加上税费之和。

推论型

测试条件为真时，确定一个新事实的真相。

如：如果客户没有未付发票，则该客户为首选状态。如果客户处于首选状态，则该客户的订单有资格享受20%的折扣。

待继续探究。

添加规则

业务规则一(条件规则)

左值	<input type="text" value="预约起始日期"/>	比较运算符	<input "="" type="text" value="<="/>	右值	<input type="text" value="预约"/>
----	-------------------------------------	-------	---	----	---------------------------------

业务规则二(文本规则)

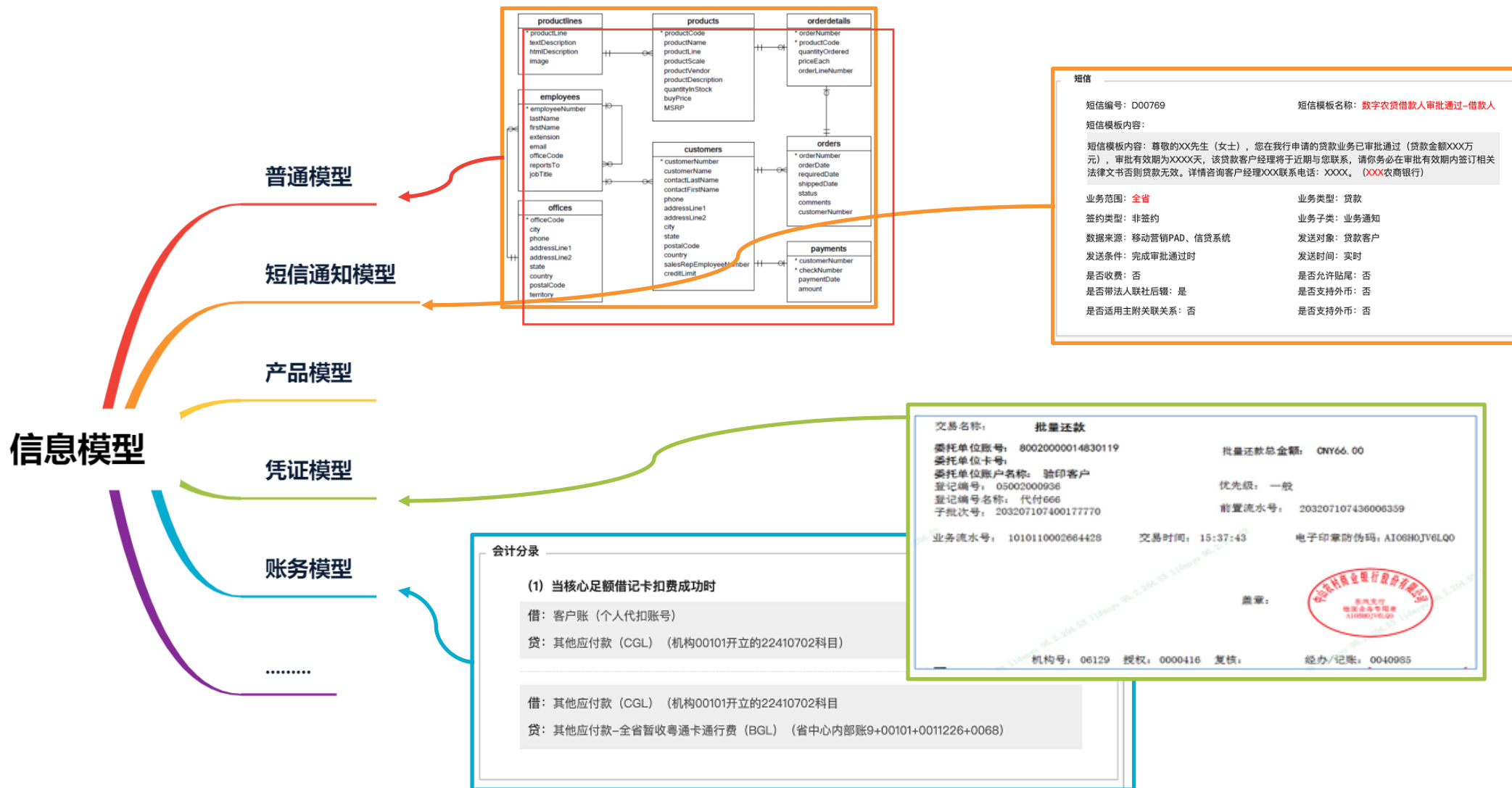
规则文本

业务规则三(事件规则)

启动条件	<input type="text" value="银行久账户存取款"/>	参与者	<input type="text" value="异地客户经理"/>	事件动作	<input type="text" value="异地"/>
------	---------------------------------------	-----	-------------------------------------	------	---------------------------------

编号	功能	借方	贷方	触发条件
1	理财资金归集	各类存款	代理业务负债-本金23140101	产品成立日，进行解冻并扣款；产品开放日的申购，在确认日进行的扣款或解冻并扣款。非保本自营非保本、代销所有
		各类存款	贷：结构性存款-单位活期结构性存款-本金20080101 (对公) 或 结构性存款-个人活期结构性存款-本金20080101 (个人) 或 同业存放款项-同业买入表内理财产品-本金20171001 (同业)	产品成立日，进行解冻并扣款；产品开放日的申购，在确认日进行的扣款或解冻并扣款。保本自营保本

需求分析：普通模型基础上提炼专有模型

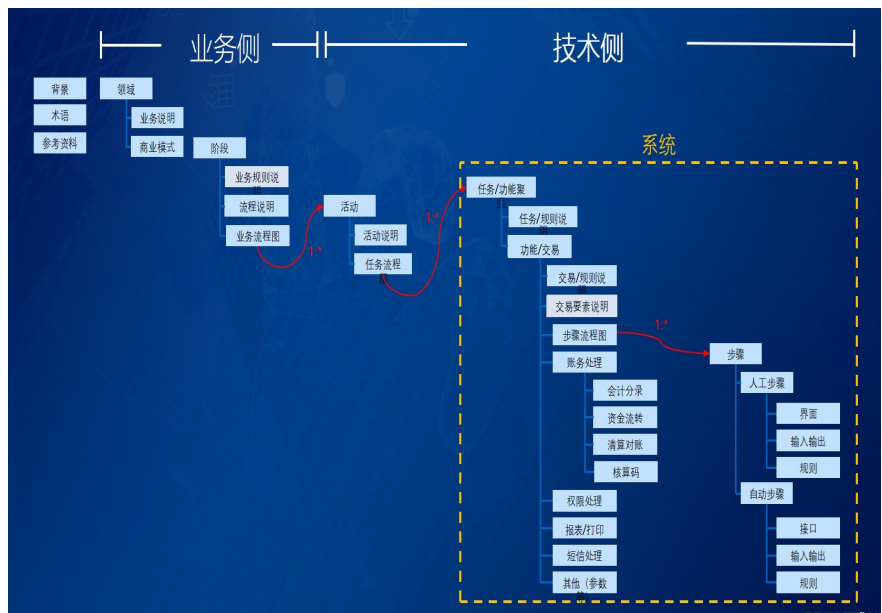


信息模型是经过抽象化的标准模型，在需求实施过程中，我们需要不断地细化和完善，以适应多样化的业务需求。

需求分析：功能点评估以及需求排期的智能化方法

为此国家制定了
了多项规范

国家标准《软件工程 软件开发成本度量规范》GB/T 36964-2018
工信部行业标准《软件研发成本度量规范》SJ/T 11463-2013
国家标准《软件测试成本度量规范》GB/T 32911



推算投产版本

功能点数量:

开发开始时间:

开发开始时间: 2021-08-30 测试完成时间: 2021-09-10 验收完成时间: 2021-09-20

推荐投产版本	版本类型	捞版日期	冻结日期	投产日期	操作
NO1	大版本	2021-09-15	2021-12-22	2021-10-21	复制投产日期
NO2	小版本	2021-09-25	2021-12-30	2021-09-22	复制投产日期
NO3	普通版本	2021-09-26	2021-12-28	2021-11-10	复制投产日期

架构设计：组件化和脚手架

框架

Spring、MyBatis、
Hibernate、Struts、
Django、Vue、Angular

框架是提供基础功能的半成品： 已经对通用组件、工具进行了代码实现和封装，开发者使用框架可以省去很多代码编写，从而提高工作效率和开发速度。

框架是规范： 框架规范应用的体系结构。通过框架规范，最大限度地满足某些方面的特性，如可读性、可靠性、可扩展性、安全性。同时有使得代码更为清晰及更易维护，降低工程维护成本。

脚手架

项目搭建工具： 基于开发框架、组件，自动创建项目结构、基础代码、管理依赖、生成示例代码，为开发者提供更简单、更高效的项目构建体验，并实现项目开发规约。

工程化 ≠ 某个具体工具

手工构建工程

新建工程

添加依赖

添加配置

编写测试代码

运行测试

开源框架

企业内部
公共组件

基础应用框架

基于开源脚手架

新建工程

添加依赖

添加配置

编写测试代码

运行测试

脚手架
工具

项目工程
实例

基于定制脚手架

新建工程

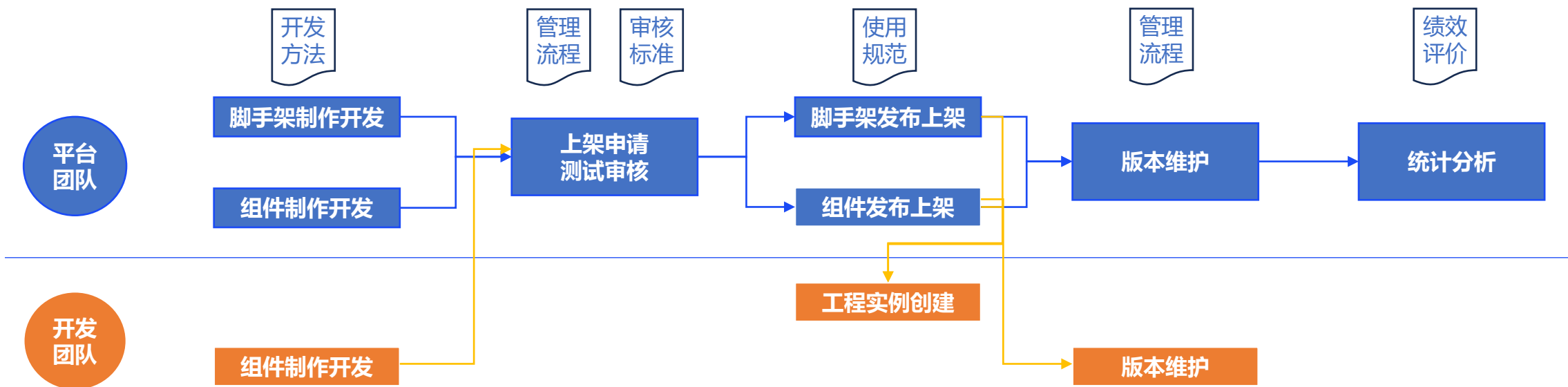
添加依赖

添加配置

编写测试代码

运行测试

架构设计：脚手架&组件建设目标



基础功能建设

- 平台基础功能建设，实现脚手架模板、组件上架发布、使用、运维迭代、统计分析等基础功能；

脚手架模板、组件沉淀

- 基于行内对技术框架规范要求、各业务组实际开发需求，按类型丰富脚手架模板、组件商店，提升开发效率和规范性；
- 基于平台基础功能，形成开发体系模板组件共享机制；

体系规范

- 建立整合相关体系规范及知识库，如技术架构规范要求、模板/组件管理流程、模板/组件制作及审核标准、模板/组件使用规范等；

推广应用

- 从试点到全面推广，推进效能提升、架构治理、能力共享；

架构设计：基于系统的分级分类智能化的脚手架推荐

分级分类体系

- ✓ 系统采集信息标签，包括总体规划、业务类、应用类、技术类、数据类、安全类标签等。
- ✓ 架构管控等级计算规则，涉及10+个维度
- ✓ 技术实施难度指标计算规则，涉及15+个维度

标签计算规则

标签名称	一级标签项	二级标签项	计算表达式	操作
			}	
	T+0内完成交易或数据处理(B015-1)	暂无数据	switch (value) { case 'B015-1': return 8	
	T+1及以上完成交易或数据处理(B015-2)	暂无数据	case 'B015-2': return 5	
异步及批量服务的时效性要求(B015)			case 'B015-3':	编辑 ...

基础建设类系统加权评估模型

管控强度 技术复杂度

管控强度计算

管控强度	分数值	操作
一级	0-10	编辑 ...
二级	10-20	编辑 ...
三级	20-30	编辑 ...

系统分级分类模型

技术
复杂度

管控
等级

不同架构管控流程
不同架构设计模板
不同应用脚手架
...

架构设计：丰富的标签助力系统建设的精确定位

架构标签

B001、系统所在应用架构层级：渠道门户层/产品服务层/....
B002、系统所在应用架构域：渠道接入域/行业应用域/....

业务标签

B003、系统功能定位：

- ◆ 交易系统
 - 动帐类
 - 非动帐类
 - ◆ 数据分析系统
 - 监管报送
 - 业务分析
 - 其它
 - ◆ 办公系统
 - ◆ 基础技术平台
- B004、界面面向操作者
- ◆ 面向客户
 - 对公和对私
 - 对私
 - 对公 (含商户等)
 - 其它
 - ◆ 面向柜员、内部员工
 - 全行营运机构使用
 - 领域线条使用 (例如信贷)
 - 单独部门
 - ◆ 无界面

业务标签 (续)

- B005、业务类日均交易量
- 大于100万笔
 - 大于10万笔
 - 大于1万笔
 - 小于1万笔
 - 不涉及 (非业务系统)
- B006、服务时间窗口
- 7*24
 - 7*12
 - 每周服务时间40小时及以上
 - 每周服务时间40小时以下
- B007、风险监管的要求
- 事前风险监控
 - 事中风险监控
 - 事后风险监控
 - 无
- B008、业务恢复时间目标RPO
- 20分钟内
 - 2小时以内
 - 2小时以上
 - 无
- B009、是否和行外第三方信息交互
- 有账务信息交互
 - 非账务的信息交互
 - 无
- B010、对账
- 和第三方对账, 以本银行为准
 - 和第三方对账, 以第三方为准
 - 只和核心对账
 - 无对账

系统功能定位(B001)计算规则验证

×

管控强度

技术复杂度

管控强度计算

+ 新增

管控强度	分数值	操作
一级	0-50	编辑 删除
二级	51-80	编辑 删除
三级	80-100	编辑 删除

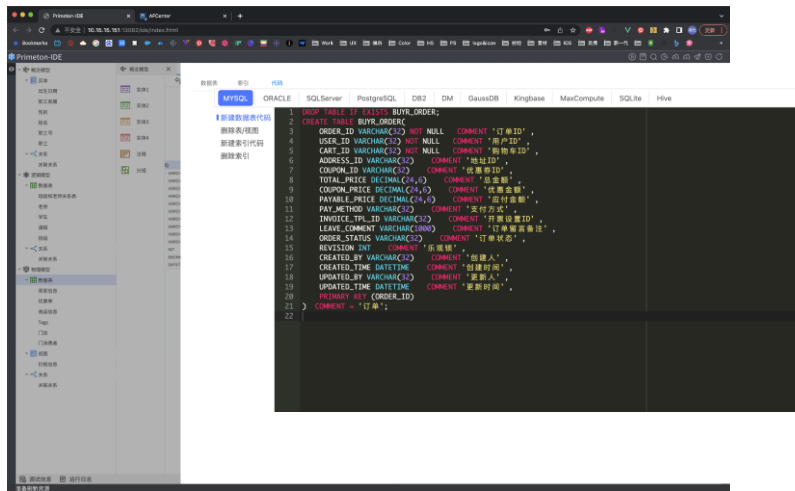
标签计算规则

标签名称	一级标签项	二级标签项	计算表达式	操作
日均交易量(B002)	高频(B002-1)	暂无数据		编辑 验证
	中频(B002-2)	暂无数据		
	低频(B002-3)	暂无数据		
	summers(B001-summer)	s(B001-summer-s)		
		s1(B001-summer-s1)		
		s10(B001-summer-s10)		

取消

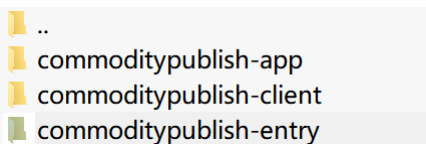
保存表达式

架构设计：通过低代码能力自动生成数据库脚本和代码



数据库 DML 语句

同时根据技术端选定的ORM框架(MyBaits等等)自动生成对应的DAO层工具和相关配置文件



自动生成应用包结构

```
package com.userdemo.businesscapability.commoditypublish.app.service;

import com.userdemo.businesscapability.commoditypublish.app.api.SpuService;
import com.userdemo.businessdomain.commodity.client.api.dto.SpuListDTO;
import java.lang.Override;
import java.util.Collections;
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Component;

/**
 * 产品服务商业能力服务实现类
 *
 * @author : 租户管理员
 * @date 2022.11.19
 */
@Component("CommodityPublishSpuServiceBS")
public class SpuServiceImpl implements SpuService {

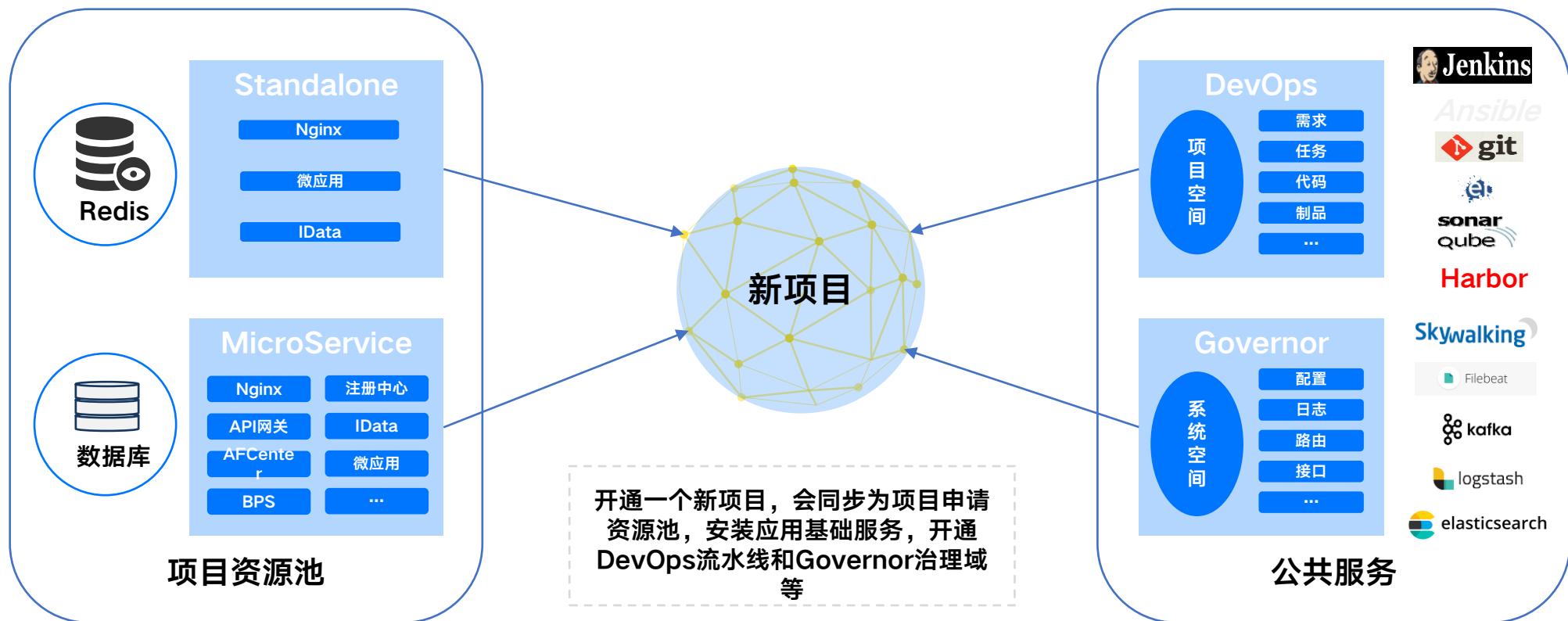
    @Autowired
    private com.userdemo.businessdomain.commodity.app.api.SpuService spuService;

    @Override
    public String createSpu(SpuListDTO spu) {
        return spuService.createSpu(spu);
    }

    @Override
    public List<SpuListDTO> listSpu(List<String> spuIds) {
        return spuService.listSpu(spuIds);
    }
}
```

JavaBean代码

架构设计：私有化APaaS，一键开通开发/运行/运维/治理环境



▶ 代码开发：通过Prompt智能推荐，标准化企业应用的数据设计



结合数据资产Prompt



计算资产关联度
给出推荐



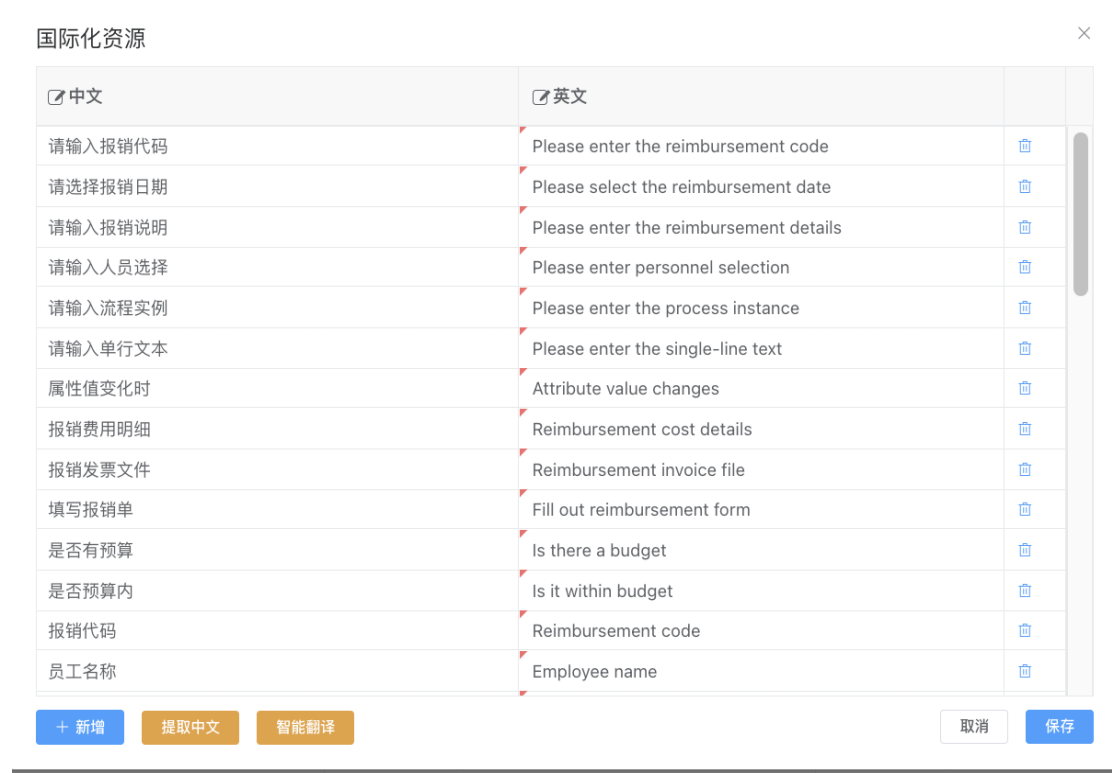
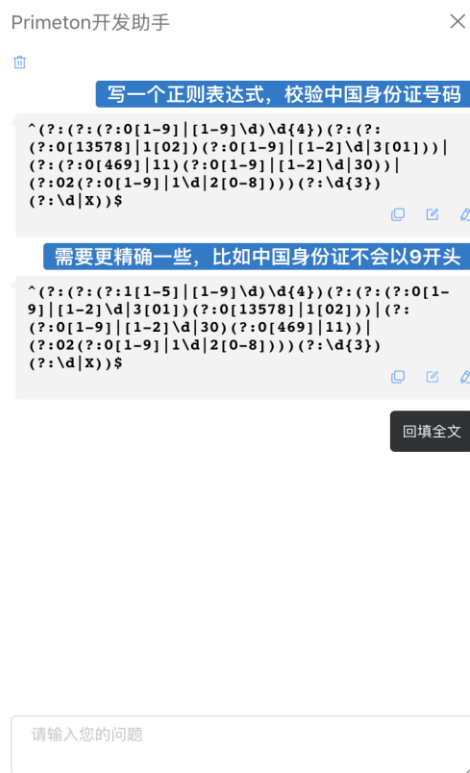
选择相关资产进行
模型设计



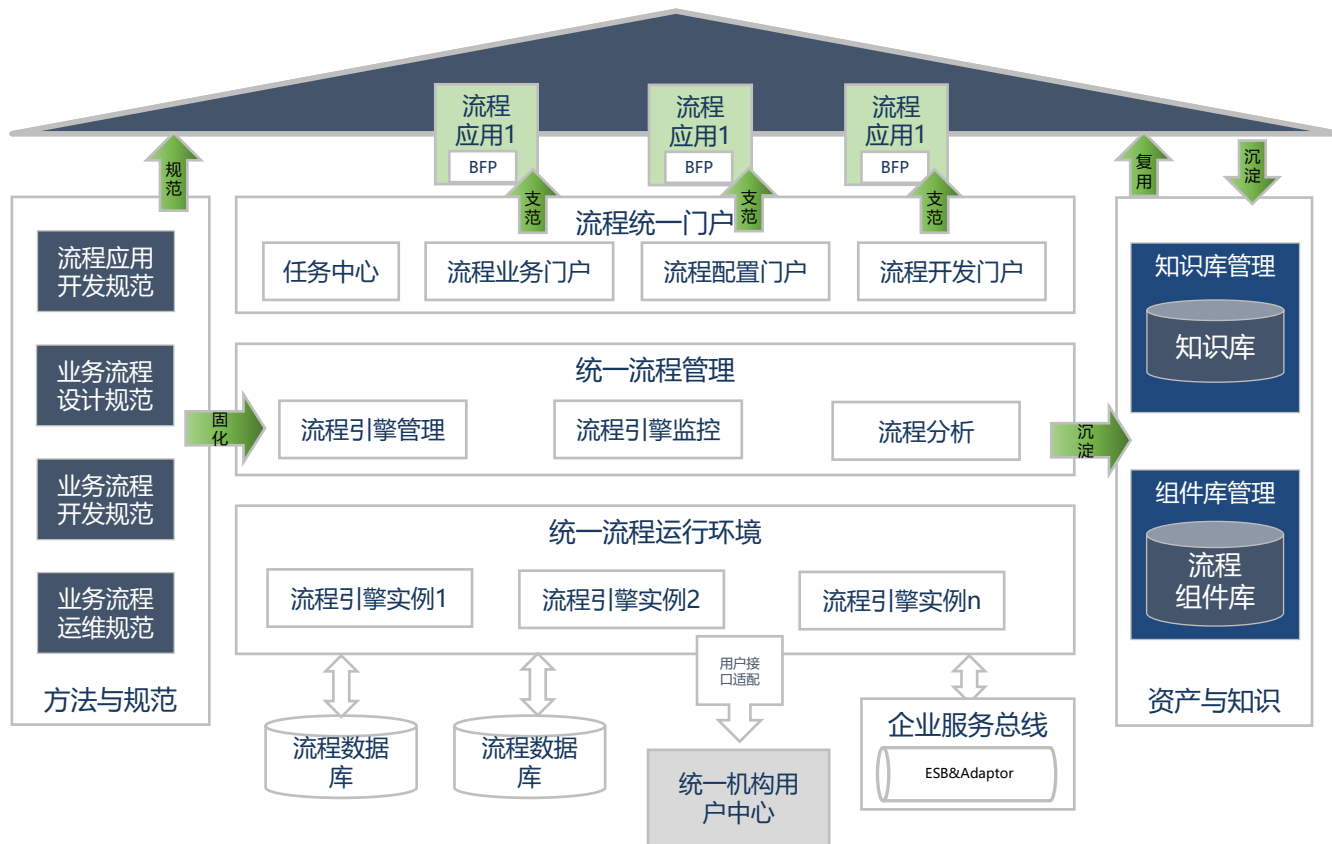
确认字段、关系等
模型提交

▶ 代码开发：表单开发智能辅助，多语言智能翻译等场景

- 结合AI能力实现**智能代码生成**，辅助生成正则校验、VUE、HTML、JS等代码逻辑片段；
- 国际化多语言场景：智能化自动提取中文字段，智能辅助翻译为英文内容



▶ 代码开发：提供统一流程中心，支撑集团型企业特色流程业务



区别于简单审批流，提供的是业务流程的建设能力，同时覆盖审批流所有能力。

企业建立统一流程中心，并非简单的集权管理，核心是打通分公司、子公司的业务和信息，持续进行工作优化、业务创新。

充分考虑国内诸多特色需求，如特事特办、自由流转、多重聚合等等

▶ 代码开发：提供服务编排引擎，满足系统间多模集成与业务创新

根据统计，对于ERP、MES这类大型企业级复杂系统，页面驱动的零代码平台只能实现20%-30%的功能，模型驱动的低代码平台，可覆盖50%-60%的功能，服务编排的纳入，则使得低代码平台可通过在线配置覆盖80%-90%的功能

源值	目标值	操作
TENANT_ID	TENANT_ID	更多
ID	APP_ID	更多
NAME	FUNCGROUPNAME	更多
ISOPEN	ISLEAF	更多
CODE	FUNCGROUPID	更多

```
1 def alist = context.a1;
2 def blist = context.a2;
3 def reslist = new ArrayList();
4
5 reslist.addAll(alist.findAll().addAll(blist.findAll()))
6
7 context.ret = reslist
8
```

对象映射

数据聚合

可靠集成

任务调度

代码开发：提供智能助手服务，降低平台内生能力的扩展门槛



提供包括VUE、JS、Ansible、Java、Groovy等代码辅助，用于扩展包括**表单组件、业务规则、网关插件、部署脚本、在线图表**等平台可扩展或开发的插件。结合基础数据的Prompt，结合普元产品及接口，更准确地提供代码生成能力。



表单组件

以微前端的方式，挂载到表单设计器中

业务规则

```
1 @ParticipantRuleFunctionClass("流程发起人默认参与者规则")
2 public class DefaultProcessStarterParticipantFunctions {
3
4     @ParticipantRuleFunction(value = "流程发起人")
5     public WParticipant getProcessStarter(Map<String, Object> relativeData, Proc
6         WParticipant participant = new WParticipant();
7         participant.setId(processContext.getProcessCreator());
8         participant.setName(processContext.getProcessCreatorName());
9         participant.setTypeCode(PartyTypeAdapter.EMP.getCode());
10    }
11
12
13     @ParticipantRuleFunction(value = "流程发起人所在部门")
14     public WParticipant getProcessStarterDepartment(Map<String, Object> relative
15         WParticipant participant = new WParticipant();
16         IAFCClient afcClient = AFCClientFactory.getInstance().createAFCClient();
17         Application app = afcClient.getApplicationAPI().getAppByClientCodeAndTene
18         Employee emp = afcClient.getEmployeeAPI().findEmployeeById(processContext
19         Org org = afcClient.getOrgAPI().queryOrgByEmployeeCode(emp.getCode()), Str
20         participant.setId(org.getId());
21         participant.setName(org.getName());
22         participant.setTypeCode(PartyTypeAdapter.ORG.getCode());
23         return participant;
24    }
25
26
27     @ParticipantRuleFunction(value = "流程发起人直属领导")
28     public WParticipant getProcessStarterDirectLeader(Map<String, Object> relat
29         IAFCClient afcClient = AFCClientFactory.getInstance().createAFCClient();
```



代码开发：依托智能助手，辅助更灵活的复杂图表和大屏设计

- 平台支持常用图表的无代码配置，还支持通过AI生成更复杂的**图表Option**，通过绑定实际的**数据服务**，完成图表视图的在线开发
- 平台支持大屏的在线设计，用户可提交相关UI素材，支持通过**AI生成JS代码**，对大屏上的图表进行**高级属性和操作配置**

The screenshot shows the Newform1 development environment. On the left, a pie chart titled "公司各事业部合同总额" (Total Contract Amount of Company Divisions) is displayed. The chart is divided into five segments representing different divisions: 银行军团 (Banking Group), 信创军团 (IT Innovation Group), 央企事业部 (Central Enterprise Division), 东大事业部 (East University Division), and 东南事业部 (Southeast Division). The data source is listed as "数据来源: 某公司" (Data Source: A Company).

In the center, there are configuration options for the chart, including width (600px) and height (500px). A "图表Option" (Chart Option) button is visible.

On the right, a "代码生成" (Code Generation) window is open, showing a prompt: "写一个Echarts的option, 实现一个极坐标柱状图, 展示公司的各个事业部的销售额" (Write an Echarts option to implement a polar bar chart showing the sales of various divisions). Below the prompt, there is a code block for the ECharts option:

```
option = {
  title: {
    text: '公司各事业部销售额',
    subtext: '数据来源: 某公司',
    x: 'center'
  },
  tooltip: {
    trigger: 'axis',
    axisPointer: {
      type: 'shadow'
    }
  },
  legend: {
    data: ['销售额'],
    x: 'left'
  },
  polar: {
    radius: '60%'
  },
  angleAxis: {
    type: 'category',
    data: ['事业部1', '事业部2', '事业部']
  },
  legend: {
    data: [
      {
        name: '销售额'
      }
    ],
    x: 'left'
  },
  polar: {

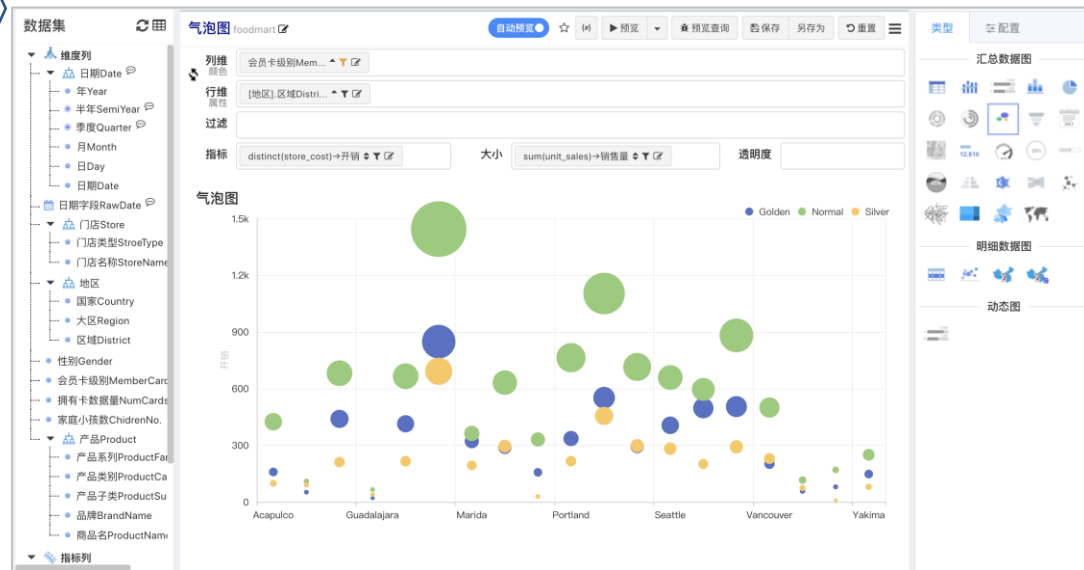
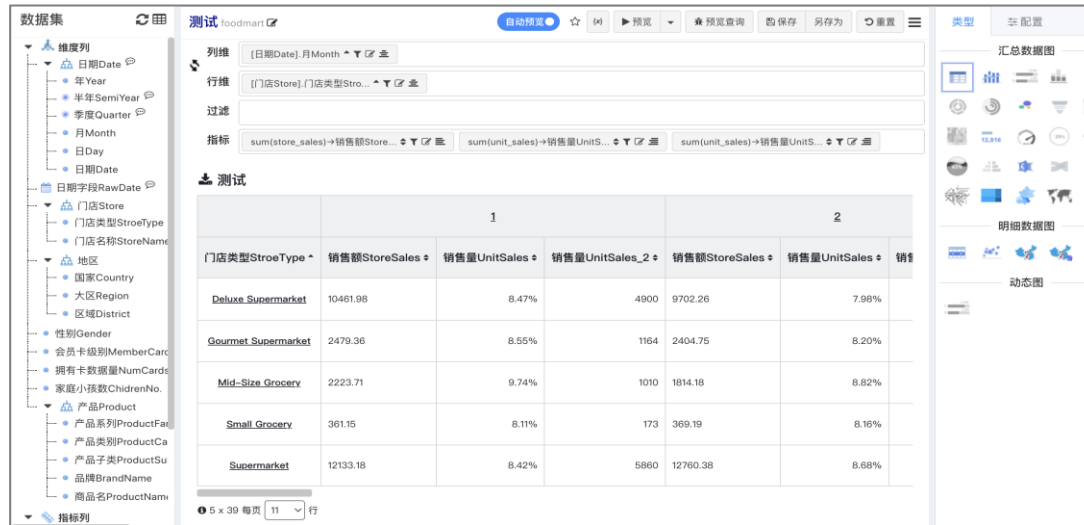
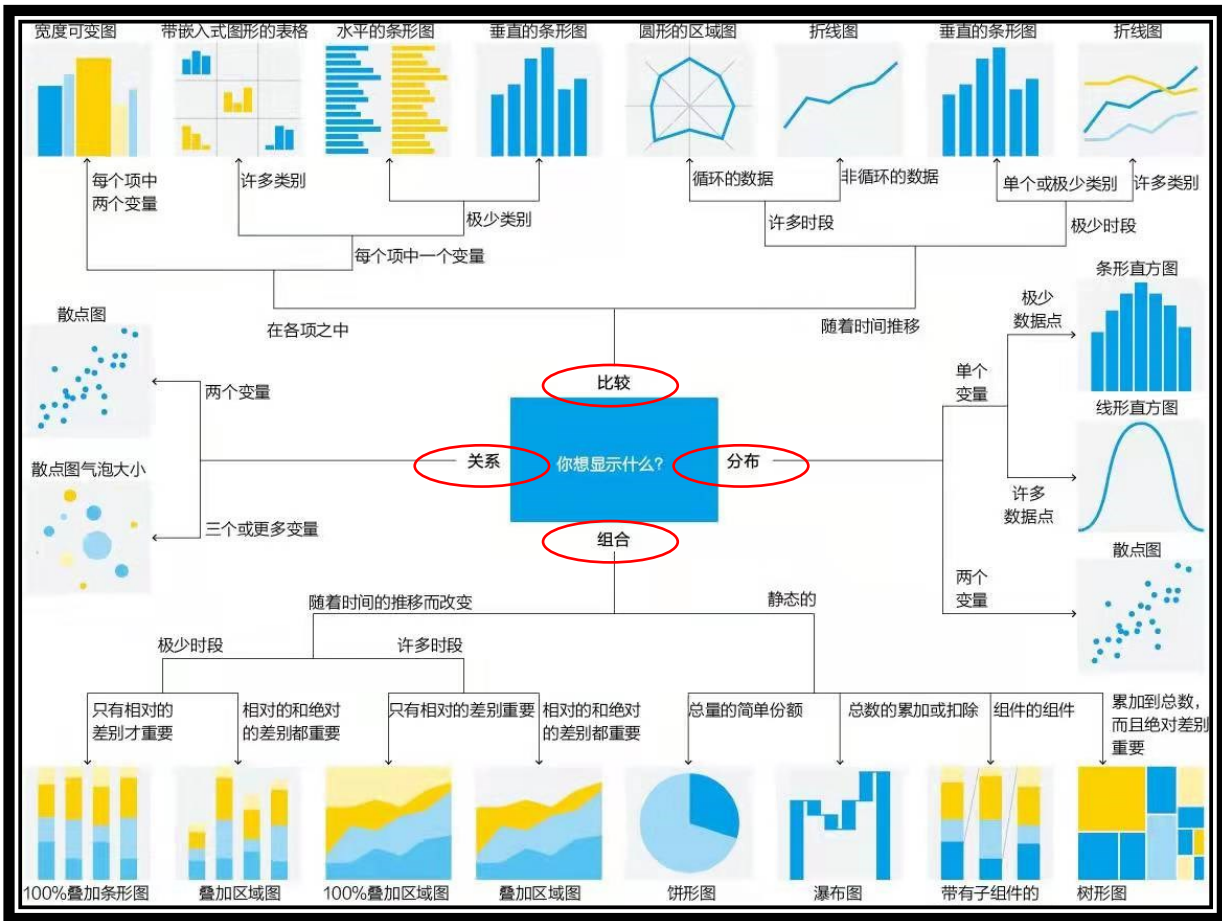
```

The screenshot shows a large data dashboard titled "企业大屏演示" (Enterprise Large Screen Demonstration). The dashboard contains several charts and data points:

- 销售排名** (Sales Ranking): A horizontal bar chart showing sales ranking points.
- 销售地图** (Sales China Map): A map of China showing sales data by region, with a total of 9,045.29元 and a 115.73% increase.
- 销售进度** (Sales Progress): A gauge chart showing sales progress at 45.54%.
- 销售总额** (Total Amount): A gauge chart showing the total amount of 9,045.29元.
- 漏斗分析** (Funnel Analysis): A funnel chart showing sales analysis.
- 销售分区排行** (Sales Ranking by District): A horizontal bar chart showing sales ranking by district.
- 投入产出比** (ROI): A gauge chart showing the ROI.
- 旭日图** (Sunburst Chart): A sunburst chart showing hierarchical data.

On the right side, there is a configuration panel with tabs for "基础" (Basic), "样式" (Style), and "组件列表" (Component List). The "样式" tab is active, showing various configuration options for the charts, such as "通用配置" (General Configuration), "标题" (Title), "颜色" (Color), "布局" (Layout), "图例" (Legend), "单项样式" (Single Item Style), "轴工具栏" (Axis Toolbar), "轮播提示设置" (Carousel Prompt Settings), "图例" (Legend), "数字标签" (Digital Labels), and "其他配置" (Other Configurations). A "代码助手" (Code Assistant) button is visible at the bottom right, with a note "通过JS代码调整图表配置" (Adjust chart configuration through JS code).

代码开发：模型助力数据可视化



▶ 代码开发：高低开融合助力复杂应用场景

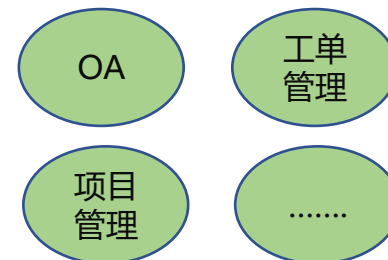
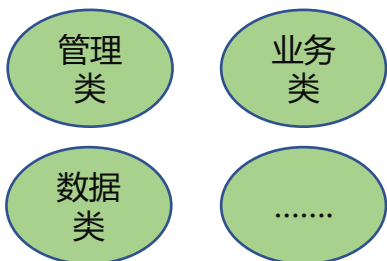
开发灵活

复杂场景

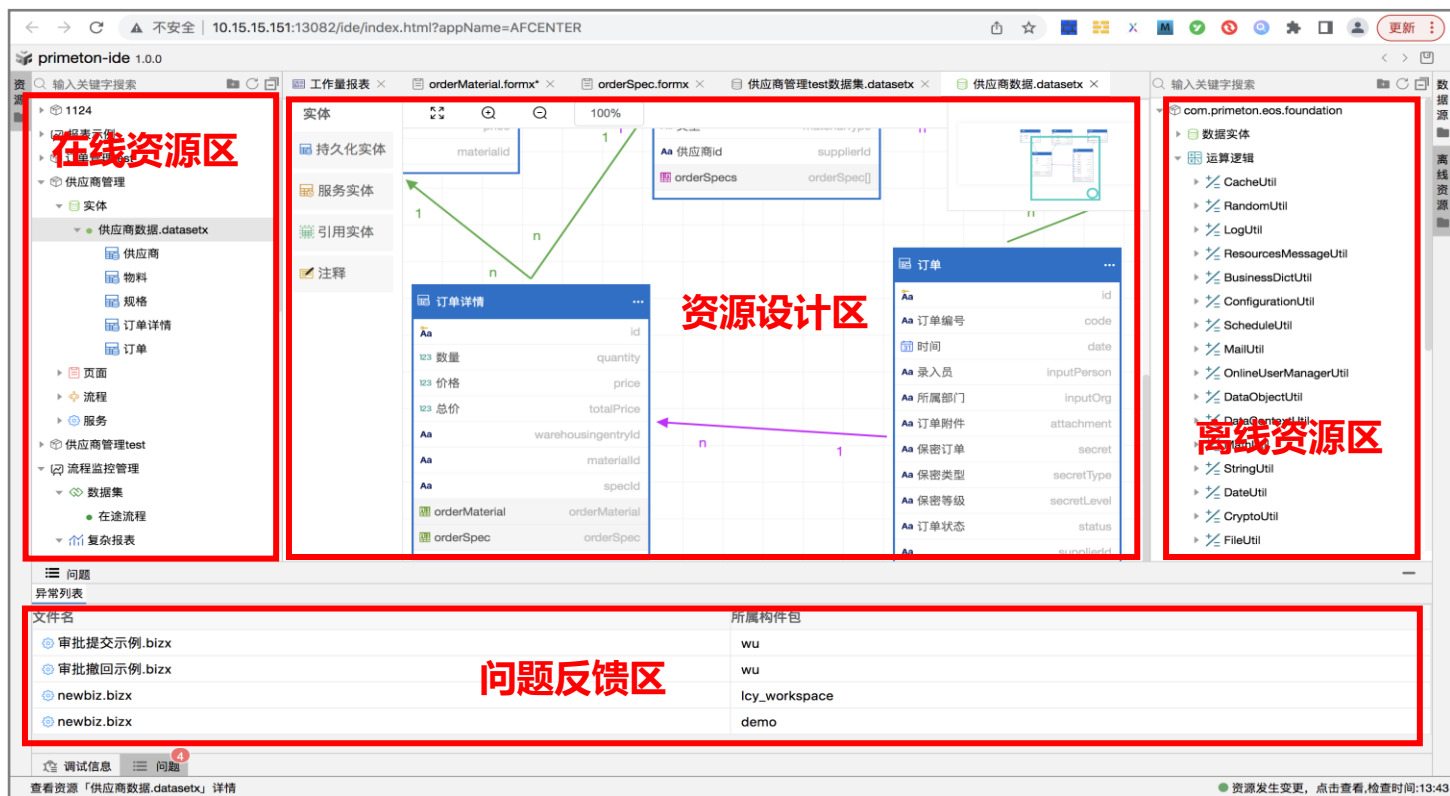
- 高开的资源如何在低开平台中无缝使用？
- 什么能力放高开、什么能力用低开？
- 大规模微服务架构下，低开平台如何适应？

快捷推出

固化模式

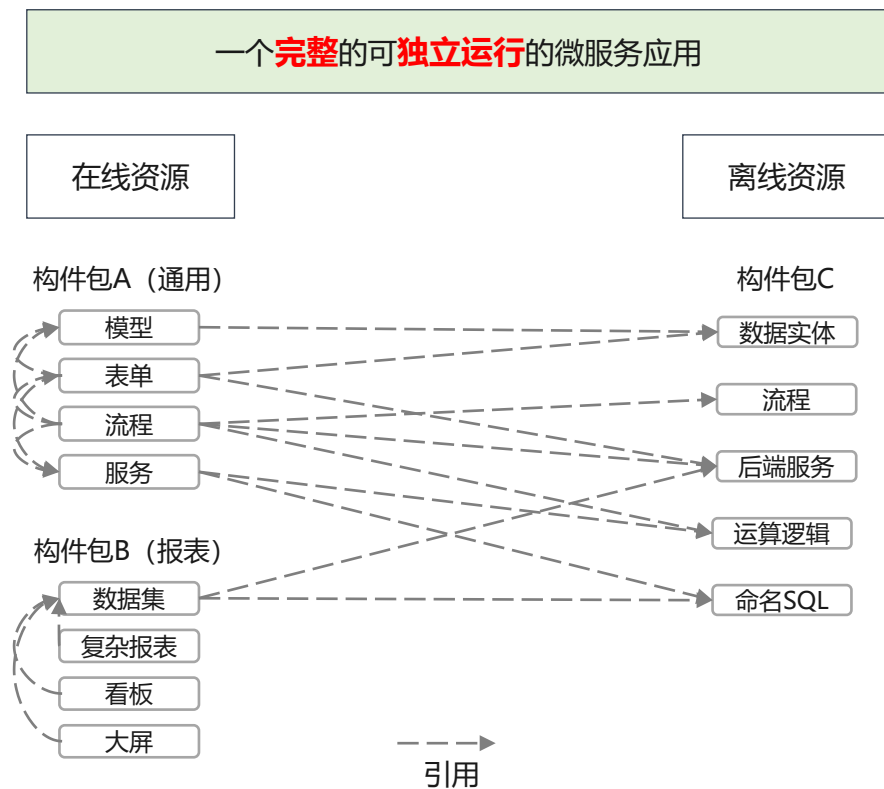


代码开发：高低开融合助力复杂应用场景



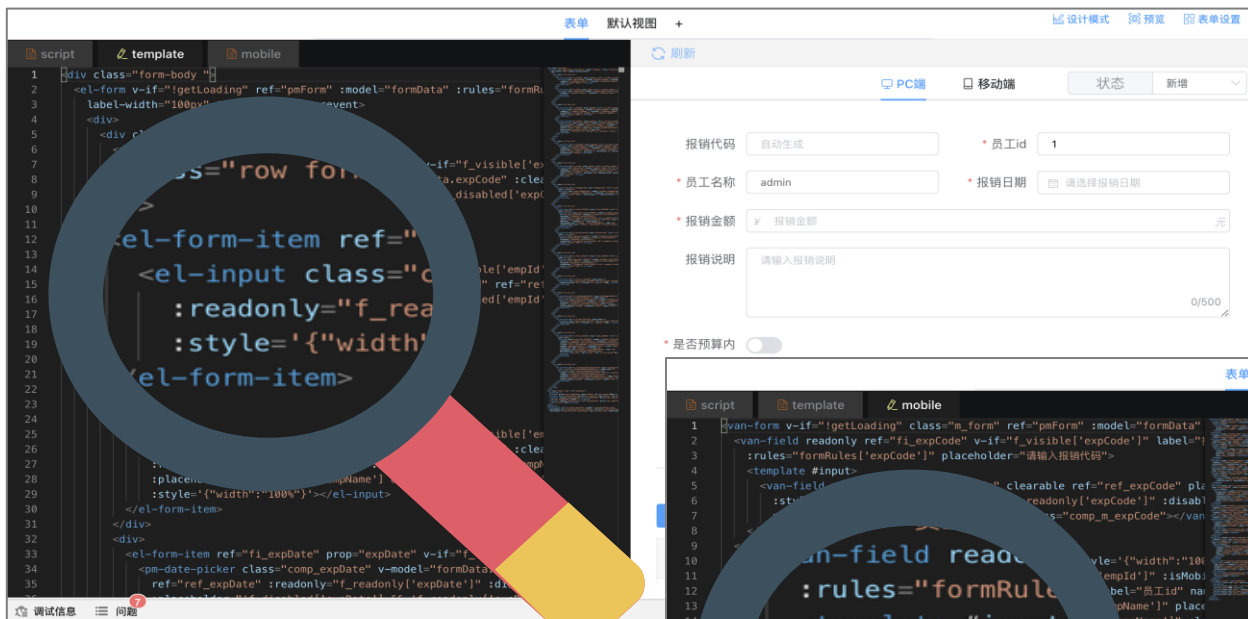
- 仿照离线开发工具模式，形成以资源开发为核心的在线低代码开发环境
- 右边展示离线资源，支持对离线资源的快速引用
- 支持问题定位，在线开发时可对具体资源进行编译检查，根据问题信息定位到具体资源内部元素

▶ 代码开发：高低开融合下资源分类和依赖关系



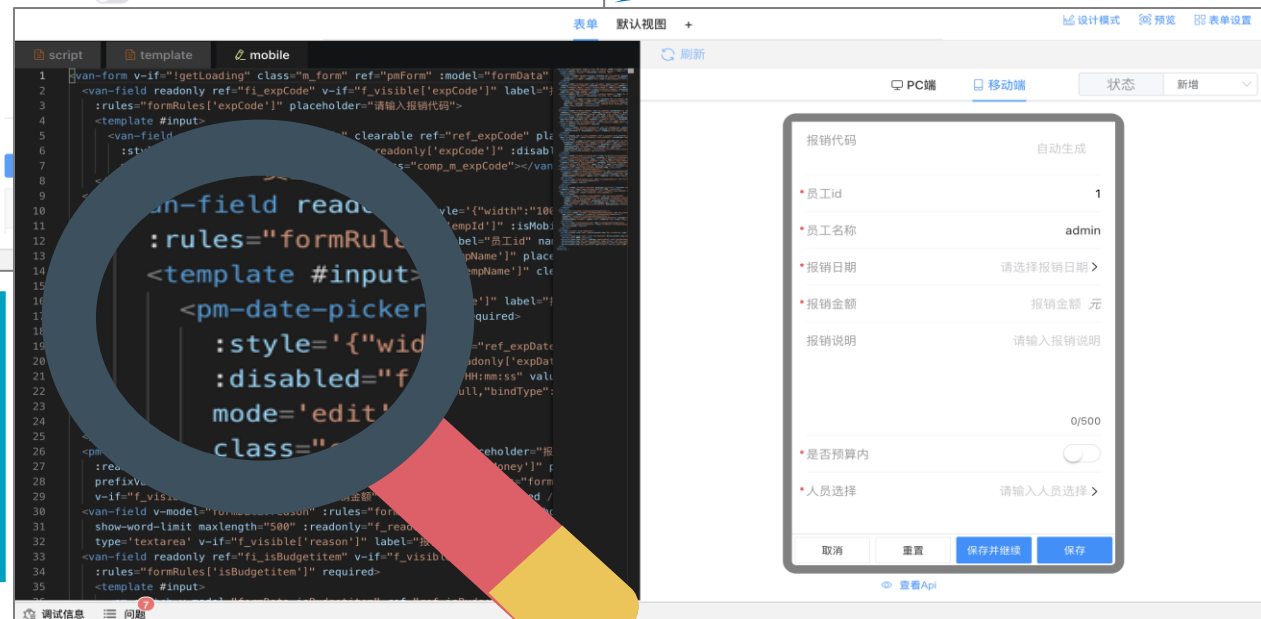
- 一个微服务应用下有多个构件包 (模块)，在线模块分为**通用类型和报表类型**，这些模块共同组成一个完整的应用
- 通用构件包中，分**模型、表单、流程、服务**四类资源
- 报表构件包中，分**数据集、复杂报表、看板、大屏**四类资源
- 在线构件包开发时，可以引用该应用内的各类离线资源，比如在线服务编排使用离线提供的本地运算逻辑

▶ 代码开发：源码而非中间模型执行，应用性能提升30%+



一些行业有着迫切的源码自主掌控要求，即使是低代码，也需要有一份真实源码留痕，且可快速线下打包运行。

无论前端UI设计还是后端逻辑编排，都是采用设计器 -> 中间模型 -> 源代码的模式，提升最终渲染或执行性能。



▶ 用户测试：基于结构化需求，测试案例的自动生成

功能需求

- 功能需求是指在业务需求书中直接体现具体需要提供的功能和内容提升点，以用例的形式呈现；
- 功能需求是验收测试的核心测试对象，是当前版本的必测内容，必须有测试案例直接覆盖

非功能需求

- 非功能需求是指为满足用户业务需求，系统需要达到的一些特性；
- 在需求说明书中明确写出非功能需求，也是当前版本的必测内容，应当针对性的设计测试案例

紧密关联业务功能

- 和业务功能需求紧密关联的业务功能，通常包括业务变动可能影像的存量数据、其他地区、其他业务模型、其它接口调用等，以及主动识别出来非功能测试需求，如安全、性能、兼容性等

核心业务功能测试范围

- 核心业务功能是指营业必备的存贷汇兑等基础业务功能，通常进行周期性测试，是版本改造项目测试单位的有效补充

需求说明书

基本信息：需求名称、描述、产品信息等

业务流程与规则

一、活动任务图

二、.....

三、前置条件

四、流程以及规则描述

1、任务A：

业务流程图

步骤：

2、任务B

.....

五、错误处理

六、后置条件

输入与输出

输入信息定义

输出信息定义

.....

IT架构结构化设计平台

用户测试：测试用例的生成是对规则模型的解析

需求说明书

基本信息：需求名称、描述、产品信息等

业务流程与规则

- 一、活动任务图
- 二、.....
- 三、前置条件
- 四、流程以及规则描述
 - 1、任务A：
 - 业务流程图
 - 步骤：
 - 2、任务B
 -
- 五、错误处理
- 六、后置条件

输入与输出

- 输入信息定义
- 输出信息定义
-

功能名称	需求内容	前置条件	操作步骤	预期结果	实际结果
对私子户信息查询	界面内容测试		进入对私子户信息查询界面	界面加载正常	
	查询方式选择“内部查询”域后测试		查询方式选择“内部查询”，输入账号，点击查询	回显对私子户信息，打印按钮可用，点击打印按钮，调用外设打印文件	
	查询方式选择“内部查询”域后测试		1.查询方式选择“客户查询”； 2.输入账号，调用外设获取密码，密码正确后输入证件信息； 3.输入证件信息，检查证件信息与账户留存的证件信息是否相符。不符则提示“客户证件号码与预留不相符”相符则继续输入 4.点击查询	回显对私子户信息，打印按钮可用，点击打印按钮，调用外设打印文件	

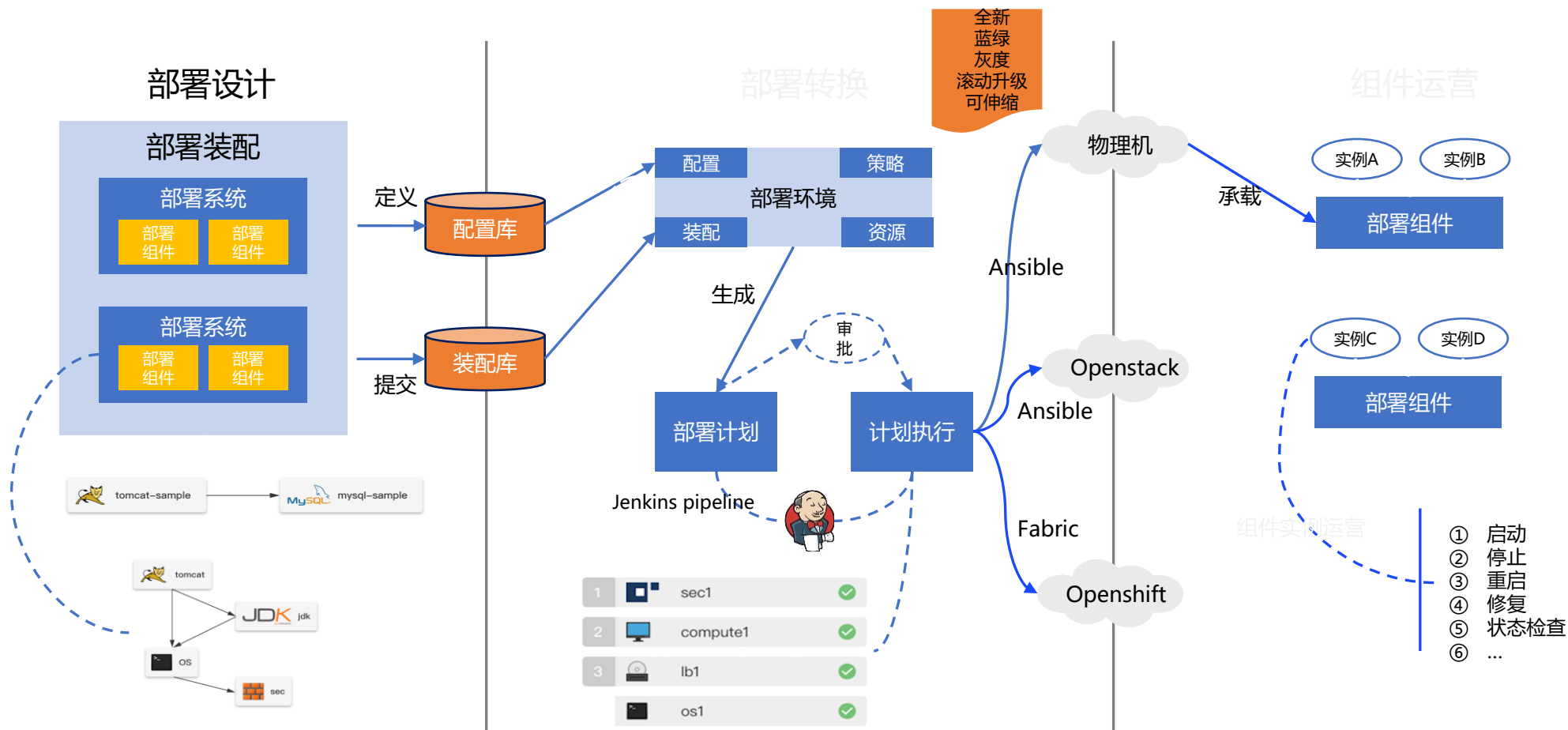
对应部分反例

接口测试案例

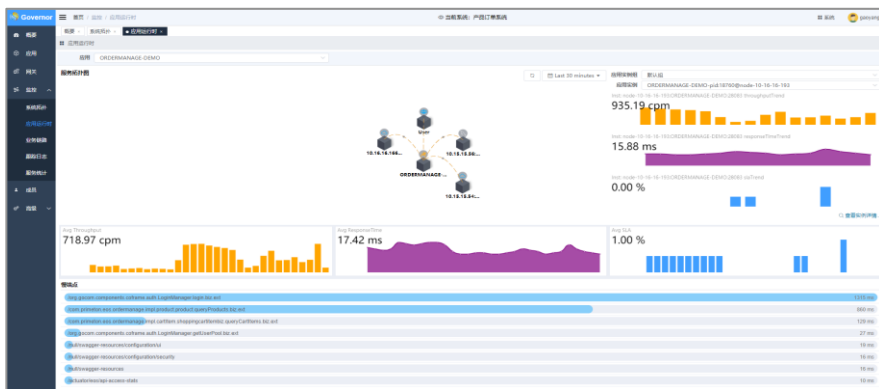
测试用例

测试用例的自动生成主要依赖于前期对需求结构化的描述，描述的内容可以根据模型自动生成对应的测试案例，比如对于规则模型可以根据规则生成对应的正例和反例，对于描述性需求可以直接按照步骤生成对应的测试用例。在智能化时代也可以借助机器学习和自然语言等技术进行一定程度上的智能化处理

集成和发布：涵盖多策略的部署能力，实现云上云下应用自动发布



运营监测：依托全方位运行监控与链路回溯，保障系统的可靠运行

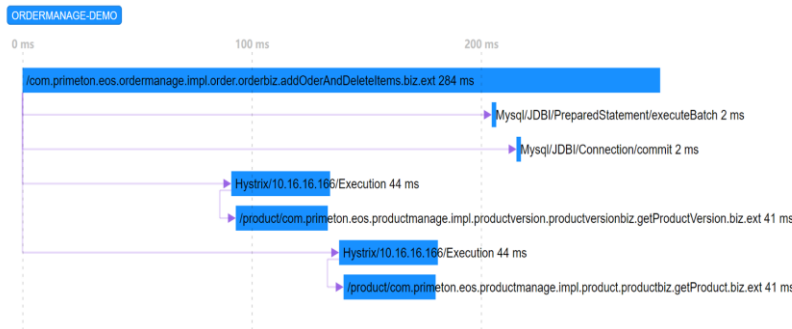


关注各个应用内的资源运行情况，统计服务运行信息：

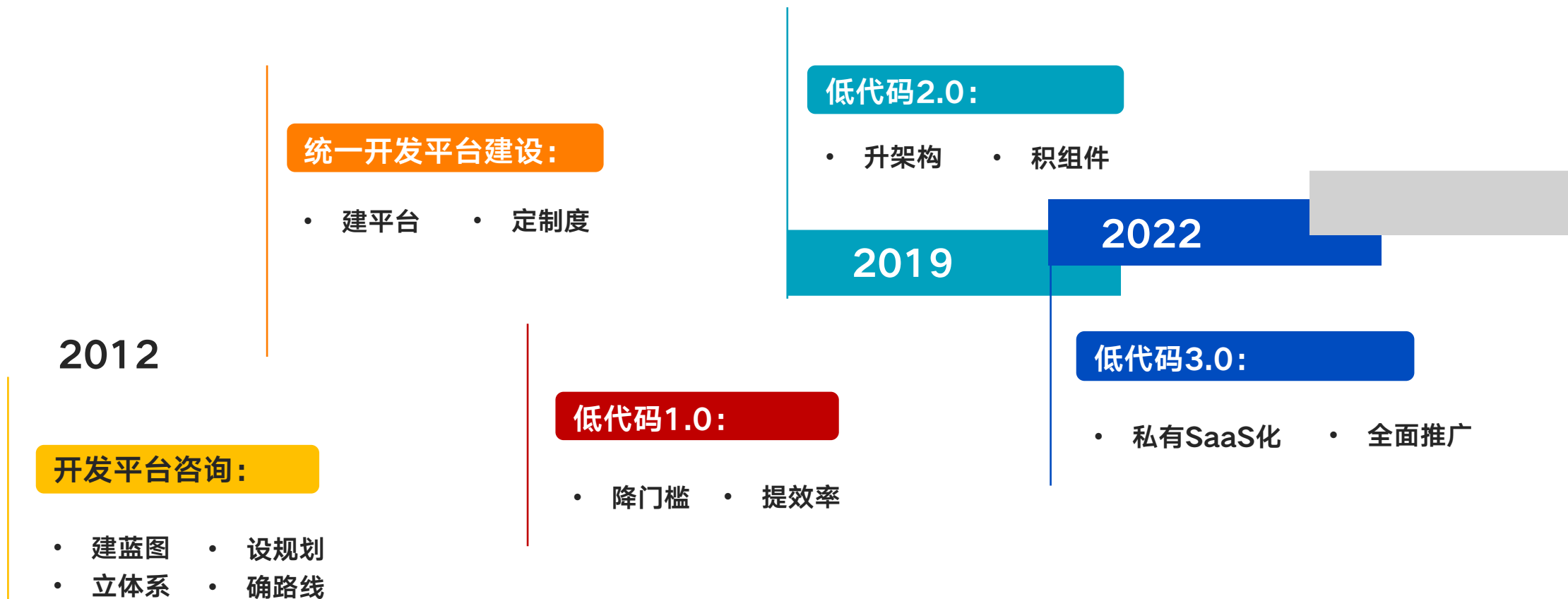
- 1、慢服务、热点服务
- 2、慢sql、热点sql
- 3、服务成功、失败等
- 4、机器CPU、内存等
- 5、.....

从交易的角度，观测每笔交易的调用链路，支持系统间、微服务间、前后端等各类链路信息回溯：

- 1、全局流水号、本地Span
- 2、支持与APM采样信息关联
- 3、支持与遗留系统的关联
- 4、.....



▶ 某大型商业银行的低代码演进之路



低代码加速金融企业数字化转型



轻应用

- 资产盘点
- 财务报销
- 问卷调查

轻业务系统

- 代缴代付
- 员工数字化工作台
- 对客服务与营销

专业系统

- 风险控制
- 综合柜员
- 场景银行

▶ 低代码在该行的应用取得显著效果

覆盖广

- 涵盖多个业务场景，包括：零售、风险、企金、运管、内部管理、中间业务，共建设303个系统。
- 推广至43家分行，5个研发中心
- 低代码应用覆盖全行6万名员工

效率高

- 开发效率提升300%
- 1~2天的业务流程处理，降低至1~2小时
- 平均每个分行，节省一线员工工作量800+人月

成本省

- 平均每家分行减少配备20+位专职IT人员
- 平均每个IT系统建设项目，节省人力成本30%

▶ 打造适合金融行业的数智化研发体系

智能化数字应用

数字化

智能化

◀ 金融低代码 ▶

生态化

集约化



全面的
方法论
以及规范
支撑



业务需求 产品需求 项目需求

企业业务建模平台

能力维



架构设计

企业IT架构
建模平台



开发实现

高低开融合
开发工具



部署上线

DevSecOps



运维



交付用户



数字
化科技
运营



价值
维度

知识小蜜蜂

数据智能小蜜

项目智能管家

外包智能管家

科技风险智能管家

采购智能管家

信息安全底座

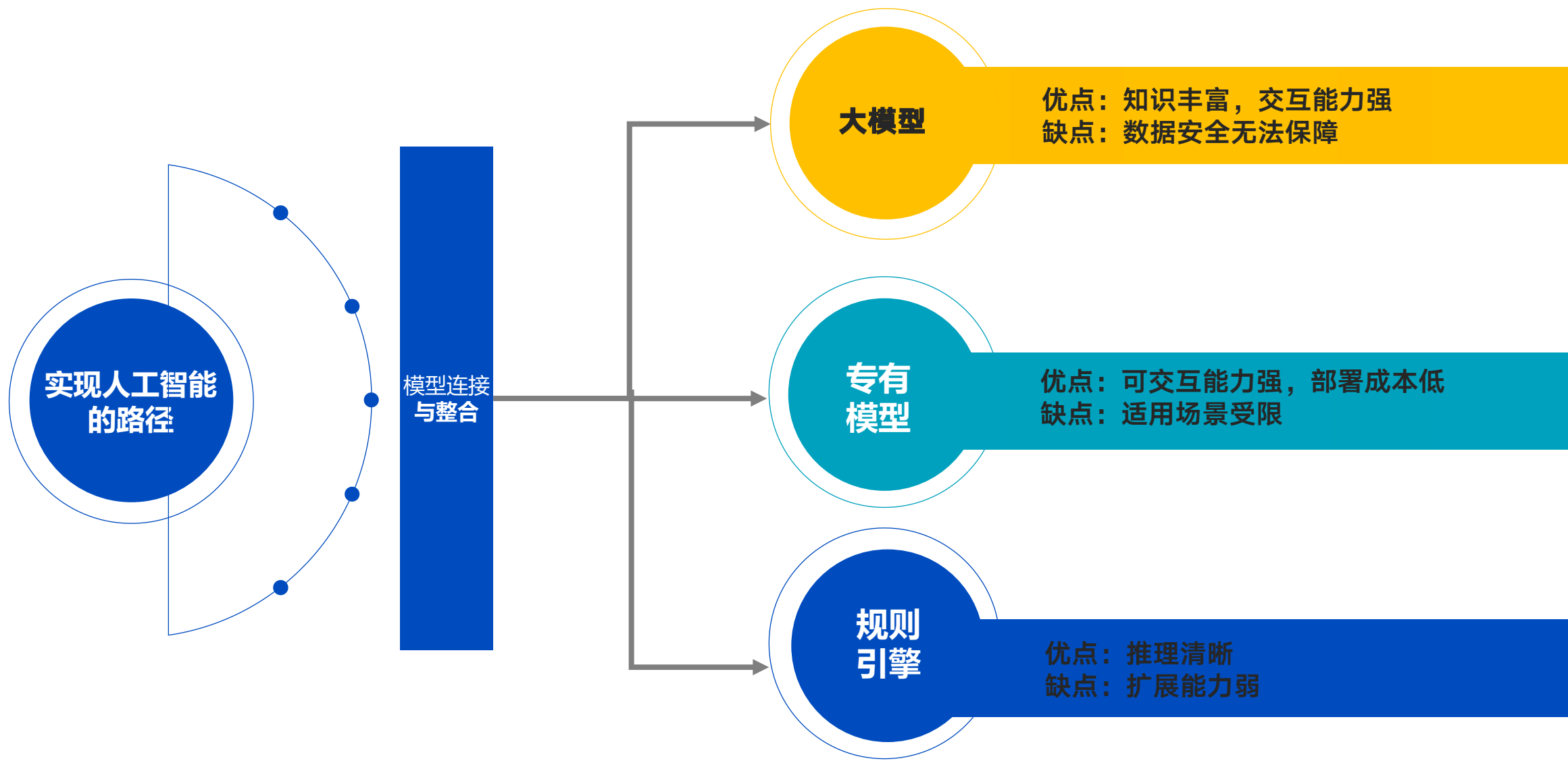
AI驱动软件研发全面进入数字化时代

 AI+ 软件研发数字峰会
AI+ software Development Digital summit

PART 03

AI+低代码的未来展望

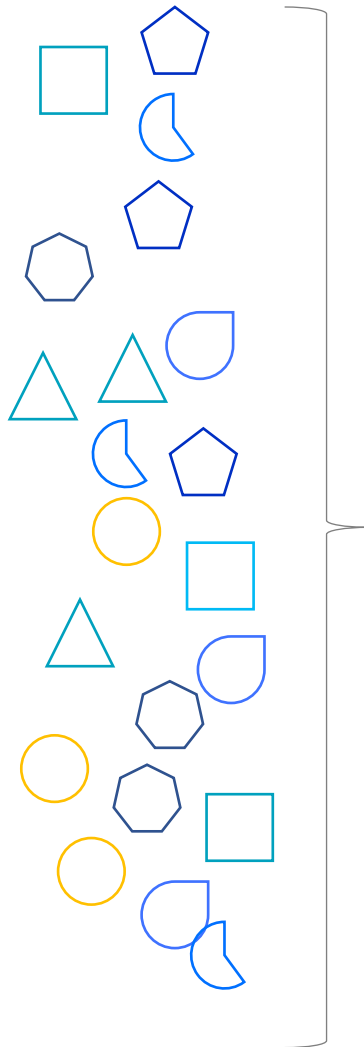
▶ 人工智能化的路径并非只有AIGC



▶ 智能化加持的低代码开发平台：为更广泛用户群提供更高的价值



▶ 5000+大客户, 100000+应用打造低代码开发专有模型



业务最佳实践的知识沉淀

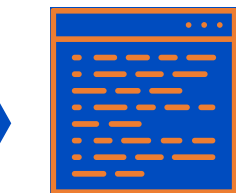
训练
微调

专有模型

规则引擎

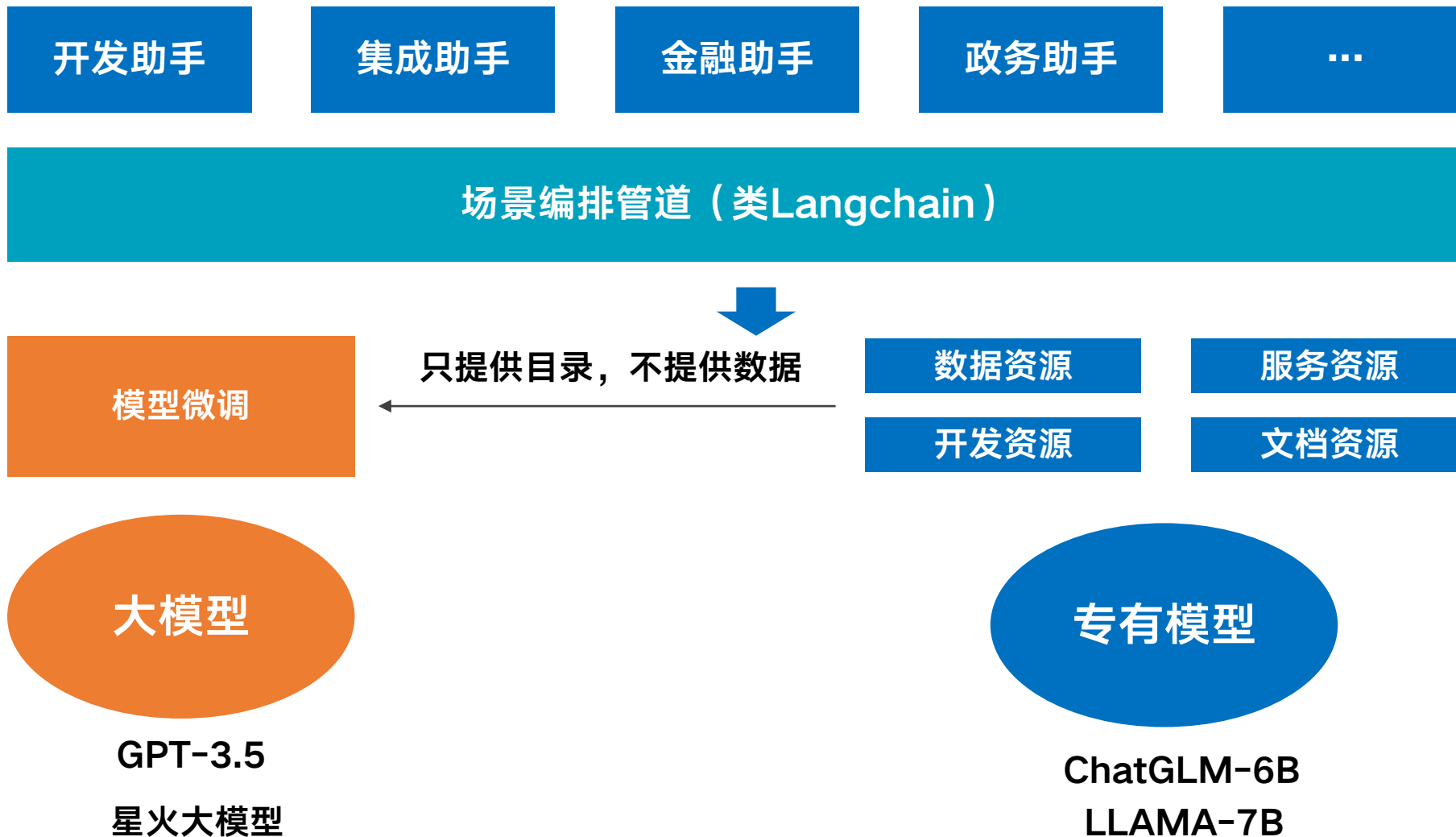
大模型

模型连接与整合

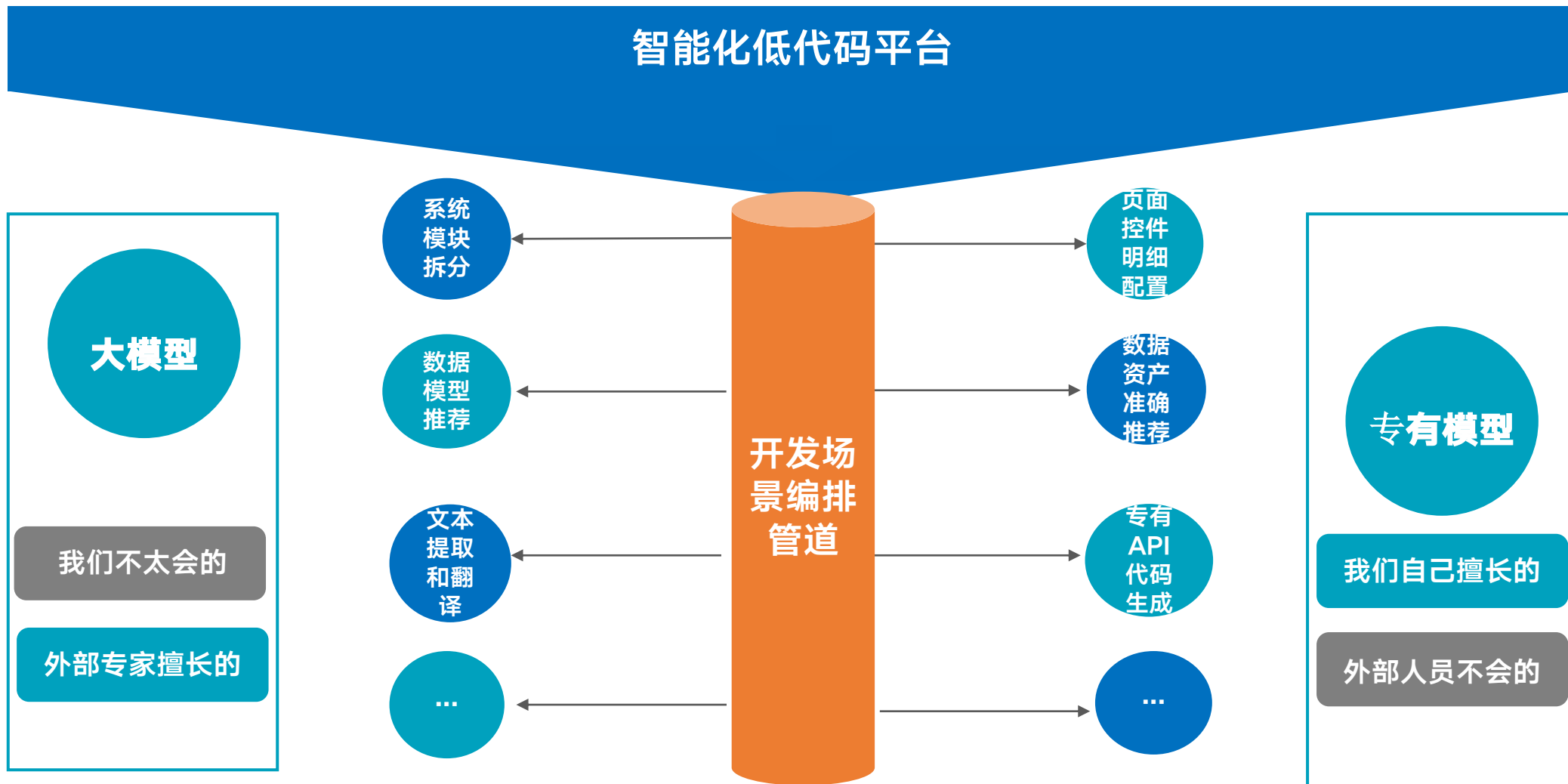


代码生成

▶ 提供多款AI大模型与专有模型的应用能力



▶ 审视开发过程，专有模型和大模型明确分工



低代码产品战略路线图



低代码+大模型

支持企业构建智能化应用开发能力

低代码+大模型+专有模型

支持企业构建的员工数字化工作平台

低代码+大模型+专有模型+智能研发服务

支持企业构建知识驱动的数字化能力

低代码+大模型+专有模型+智能研发服务+业务场景

提供面向行业的智能研发 SaaS 服务

AI驱动软件研发全面进入数字化时代

AI+ 软件研发数字峰会
AI+ software Development Digital summit

▶ 未来，AI加持的低代码在金融领域大有可为

1

自动化交易

利用低代码平台和AI技术，可以构建自动化交易系统，实现快速、准确的交易执行和风险管理。

通过低代码平台和AI算法，可以构建欺诈检测系统，帮助金融机构及时发现和防止欺诈行为。

欺诈检测

2

3

风险评估

利用低代码平台和AI模型，可以进行风险评估和预测，帮助金融机构制定有效的风险管理策略。

1.通过低代码平台和AI技术，可以构建智能客服系统，提供个性化的客户服务和支持。

客户服务

4

5

投资决策

利用低代码平台和AI算法，可以进行投资决策分析和预测，帮助投资者做出更明智的投资决策。

▶ AIG 还有些需要解决的问题

解决专业知识+数据量
不足问题

专用模型与 知识积累

- 沉淀**业务组件**，形成知识有效积累问题，**利用专家**人工进行知识积累，**不需要大数据量训练**
- 基于**企业已有数据资产**形成**知识图谱**，训练专用模型

解决与模型
精准对话问题

Prompt Engineering (提示工程)

- 多种对话模式
- 实现长时间记忆
- 协助意图推理
- 判定对话语气
- ...

解决调用问题

自动化调用

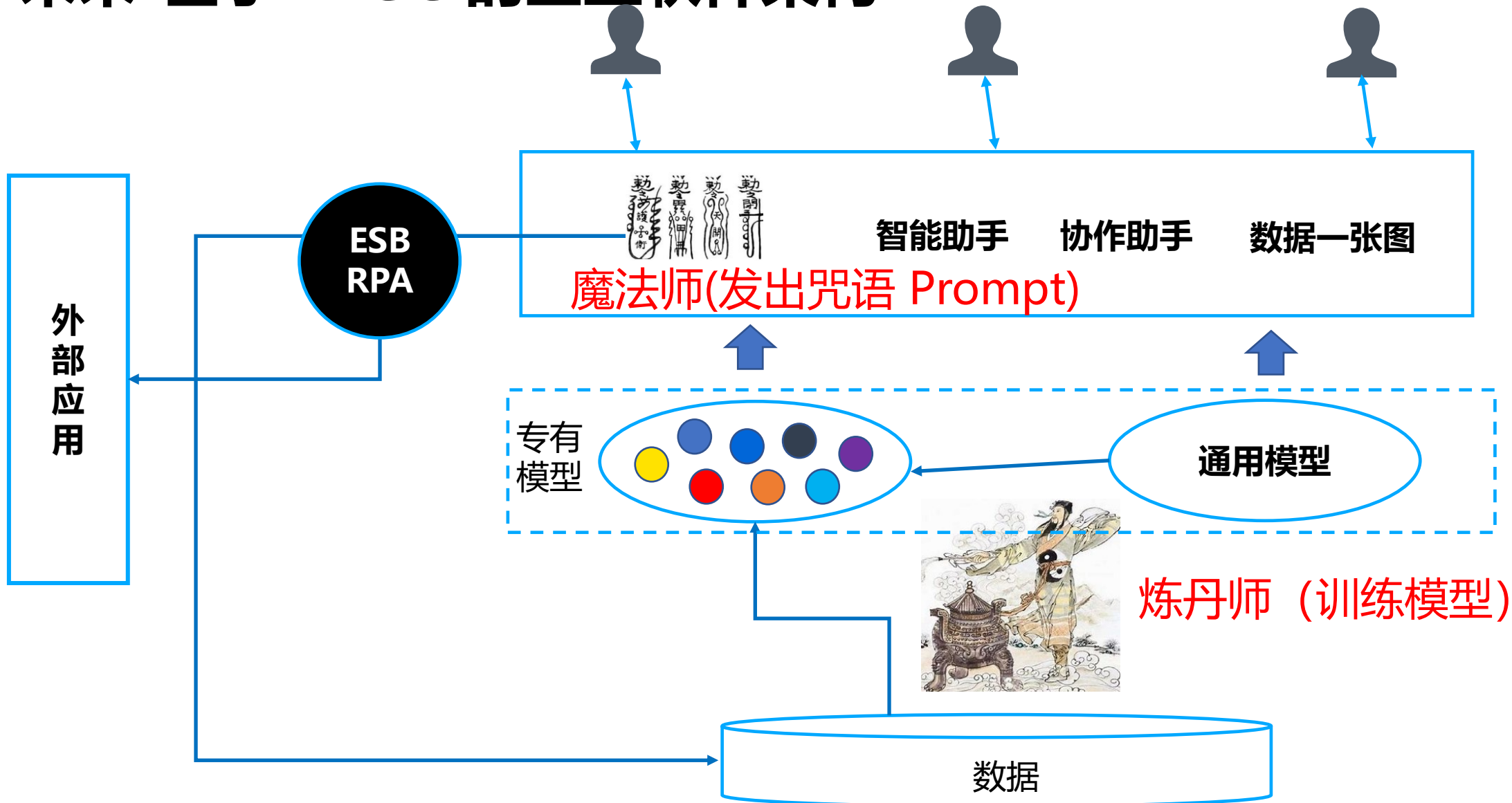
- 业务组件
- ESB、RPA
- 模型插件
- 搜索引擎整合

解决算力与安全问题

专用模型的选择与 私有化部署

- 选择开源小模型训练专用模型
- 适配不同大模型

▶ 未来-基于 AIGC 的企业软件架构



感谢聆听

