

面向零售和小微业务的数据处理

**技术方案 与 应用场景**

叶 亮

**目 录**

[1. 改版记录 2](#_Toc11261552)

[2. 背景 3](#_Toc11261553)

[3. 技术方案 4](#_Toc11261554)

[3.1 规则引擎 4](#_Toc11261555)

[3.1.1 缘起 4](#_Toc11261560)

[3.1.2 业务场景 4](#_Toc11261561)

[3.1.3 早期试水方案 4](#_Toc11261562)

[3.1.4 技术方案：规则引擎 5](#_Toc11261563)

[3.1.5 规则引擎核心组件 7](#_Toc11261564)

[3.1.6 规则引擎扩展组件 9](#_Toc11261565)

[3.1.7 监控与报警 10](#_Toc11261566)

[3.2 实时流处理平台 12](#_Toc11261567)

[3.3 日志采集系统 14](#_Toc11261568)

[3.4 客户标签和客户画像 16](#_Toc11261569)

[3.3.1 何为客户标签 17](#_Toc11261573)

[3.3.2 客户标签体系的构建 17](#_Toc11261574)

[3.3.3 客户标签应用平台 20](#_Toc11261575)

[3.5 微服务化和Docker容器化 22](#_Toc11261576)

[3.6 图数据库与图计算 22](#_Toc11261577)

[4. 应用场景 23](#_Toc11261578)

[4.1 客户行为实时分析 23](#_Toc11261579)

[1.1.1 客户规模与质量：活跃客户指标（部分） 25](#_Toc11261582)

[1.1.2 客户参与度分析：访问时段统计 26](#_Toc11261583)

[1.1.3 其他 27](#_Toc11261584)

[4.2 精准营销 27](#_Toc11261585)

[4.5.1 系统功能 27](#_Toc11261591)

[4.3 反欺诈 30](#_Toc11261592)

[4.4 反洗钱 30](#_Toc11261593)

[5. 建设路线（Roadmap） 31](#_Toc11261594)

[5.1 第一阶段 31](#_Toc11261595)

[5.2 第二阶段 31](#_Toc11261596)

[5.3 第三阶段 32](#_Toc11261597)

[6. 附录 33](#_Toc11261598)

[6.1 常见概念 33](#_Toc11261599)

[6.2 行业背景常识 33](#_Toc11261600)

# 改版记录

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 改版内容 | 改版人 | 改版日期 |
| 0.1 | 初版做成 | 叶亮 | 2019/01/18 |
| 0.2 | 增加图数据库应用场景 | 叶亮 | 2019/06/12 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 背景

TBD

# 技术方案

## 规则引擎



### 缘起

银行业务在离线场景下，已经得到了较为系统化的支持，通过对离线数据收集、挖掘，可对目标用户进行T+1触达，通过向目标用户发送/推送等多种方式，在一定程度上提高转化率。但T+1本身的延迟性会导致用户在产生特定行为时不能被实时触达，无法充分发挥数据的价值，取得更优的运营效果。

在此背景下，运营业务需要着手挖掘用户行为实时数据，如实时浏览、搜索、下单、付款、开户、销户、投诉等，对满足运营需求用户进行实时触达，最大化运营活动效果。

### 业务场景

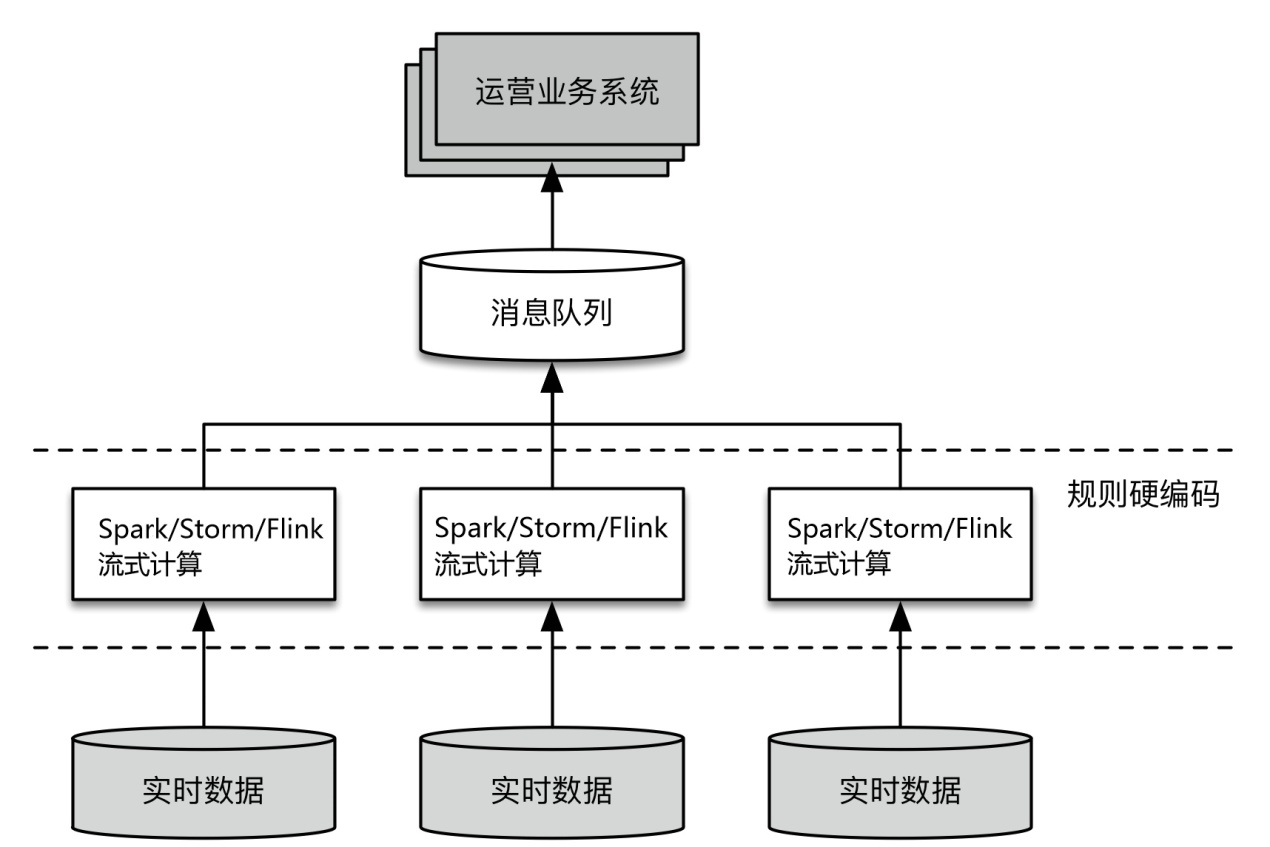
在客户运营实时触达需求中，存在如下具有代表性的业务场景：

* 用户在30分钟内发生A行为次数 ≥3次
* 用户为白金卡/钻卡/私行客户
* 用户为黑名单/灰名单客户
* 用户在A行为前48小时内发生过B行为
* 用户在A行为后60分钟内未发生B行为

下文以该典型实时运营场景为例，围绕如何设计出可支撑业务需求高效、稳定运行的系统展开。

### 早期试水方案

早期方案运营实时触达需求早期活动数量较少，我们通过为每个需求开发一套流式计算拓扑相关代码、将运营活动规则硬编码这一“短平快”的方式，对运营实时触达需求进行快速支持，如图1所示：



**早期方案的问题**

早期方案是一种Case By Case的解决方式，不能形成一个完整的系统。随着实时运营业务开展，相关运营活动数量激增，线上维护着多套相似代码，一方面破坏了DRY（Don’t Repeat Yourself）原则，另一方面线上维护成本也呈线性增长，系统逐渐无法支撑当前的需求。

**挑战**

为解决早期方案中出现的问题，对系统建设提出了以下挑战：

* 硬编码活动规则的方式产生了大量重复代码，开发成本较高，需求响应时间较长。
* 业务规则修改困难，调整运营活动条件需要修改代码并重启线上拓扑。
* 流式计算系统多套，资源利用率、系统吞吐量低，统一维护成本较高。
* 缺乏完善的监控报警机制，很难早于业务发现系统及数据中存在的稳定性问题。

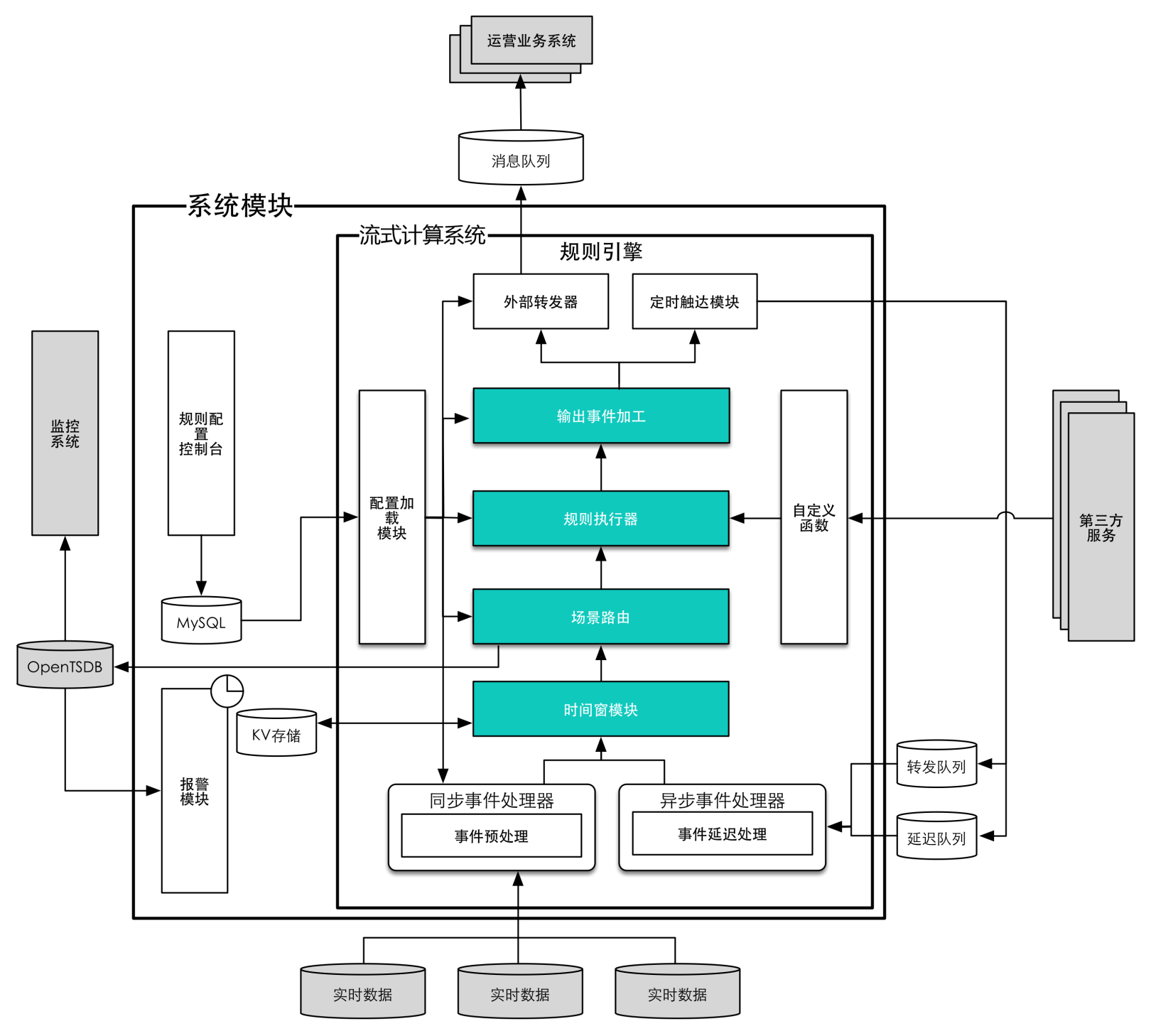
针对以上挑战，结合银行业务规则特点，我们设计了运营实时触达系统。

### 技术方案：规则引擎

确定引入规则引擎后，围绕规则引擎的设计开发成为了系统建设的主要着力点。

通过使用实时数据仓库（DW）或大数据平台中的用户实时行为数据，按业务运营活动规则，组合成有意义的复合事件，交由下游运营业务系统对事件的主体，也就是用户进行分析和触达。

将系统抽象为以下功能模块，如图所示：



总体上，系统组成模块如下：

* **规则引擎**：集成于流失计算系统中，执行运营活动条件转换成为的具体规则，作出对应响应。
* **时间窗模块**：具有可选时间跨度的滑动时间窗功能，为规则判定提供时间窗因子。
* **定时触达模块**：设定规则判定的执行时间，达到设定时间后，执行后续规则。
* **自定义函数**：在Aviator表达式引擎基础函数之上，扩展规则引擎功能。
* **报警模块**：定时检查系统处理的消息量，出现异常时为负责人发送报警信息。
* **规则配置控制台**：提供配置页面，通过控制台新增场景及规则配置。
* **配置加载模块**：定时加载活动规则等配置信息，供规则引擎使用。

对于时间触发类型的流计算规则，如日常的实时交易量统计，并发访问量，客户当日消费金额，我们使用spark streaming / Flink 等流失计算功能代替原本需要针对绝大多数数据源和规则逻辑开发的流计算组件，并将查询结果实时导入内存数据库，提供给规则引擎进行复杂逻辑处理。

规则引擎由核心组件构成的最小功能集及扩展组件提供的扩展功能组成。

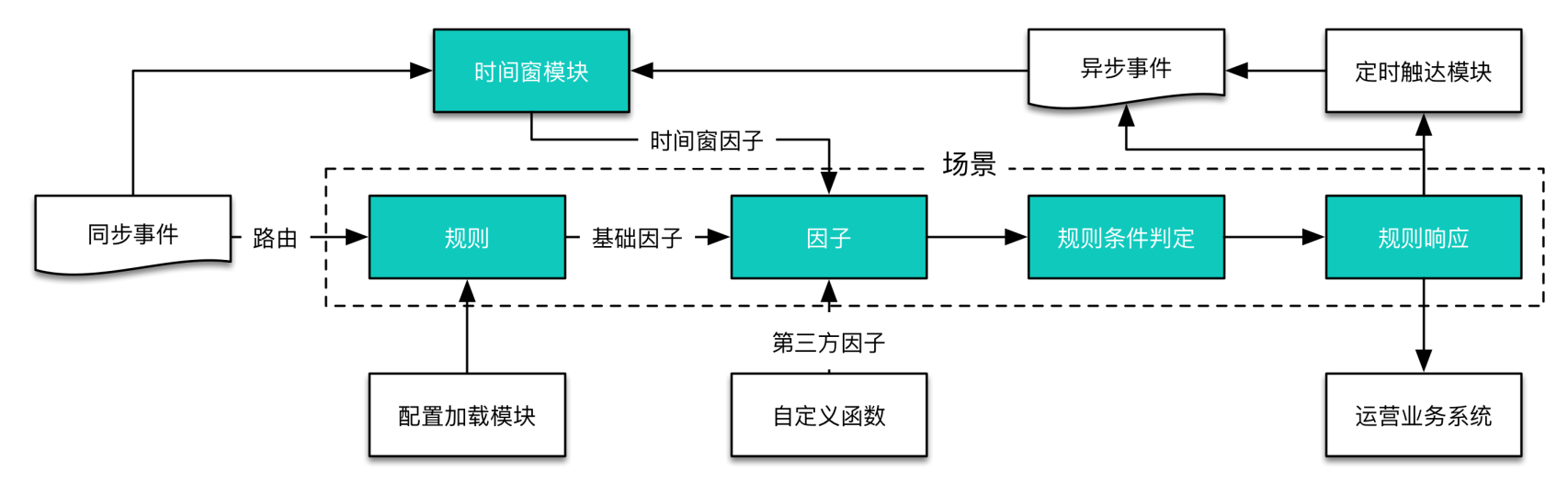
由于**规则引擎解耦了业务规则和系统代码**，使得实时数据在处理时变的抽象，对数据监控、报警提出了更高的要求。下面我们将从 **规则引擎核心组件、规则引擎扩展组件、监控与报警** 三方面进行介绍。

### 规则引擎核心组件

规则引擎核心组件为构成规则引擎的最小集合，用以支持完成基础规则判断。

**规则引擎核心流程**

引入规则引擎后，业务需求被转换为具体场景和规则进行执行，如图所示：



规则引擎在执行规则过程中，涉及以下数据模型：

* **场景**：业务需求的抽象，一个业务需求对应一个场景，一个场景由若干规则组成。用不同的规则组成时序和依赖关系以实现完整的业务需求。
* **规则**：规则由规则条件及因子组成，由路由至所属场景的事件触发，规则由规则条件、因子及规则响应组成。
* **规则条件**：规则条件由因子构成，为一个布尔表达式。规则条件的执行结果直接决定是否执行规则响应。
* **因子**：因子是规则条件的基础组成部分，按不同来源，划分为基础因子、时间窗因子和第三方因子。基础因子来源于事件，时间窗因子来源于时间窗模块获取的时间窗数据，第三方因子来源于第三方服务，如用户画像服务等。
* **规则响应**：规则执行成功后的动作，如将复合事件下发给运营业务系统，或发送异步事件进行后续规则判断等。
* **事件**：事件为系统的基础数据单元，划分为同步事件和异步事件两种类型。同步事件按规则路由后，不调用定时触达模块，顺序执行；异步事件调用定时触达模块，延后执行。

**时间窗模块**

时间窗模块是酒旅运营实时触达系统规则引擎中的重要构成部分，为规则引擎提供时间窗因子。**时间窗因子可用于统计时间窗口内浏览行为发生的次数、查询首次下单时间等**，下表中列举了在运营实时触达活动中需要支持的时间窗因子类型：

# 注：POI概念参见 [常见概念](#_常见概念)

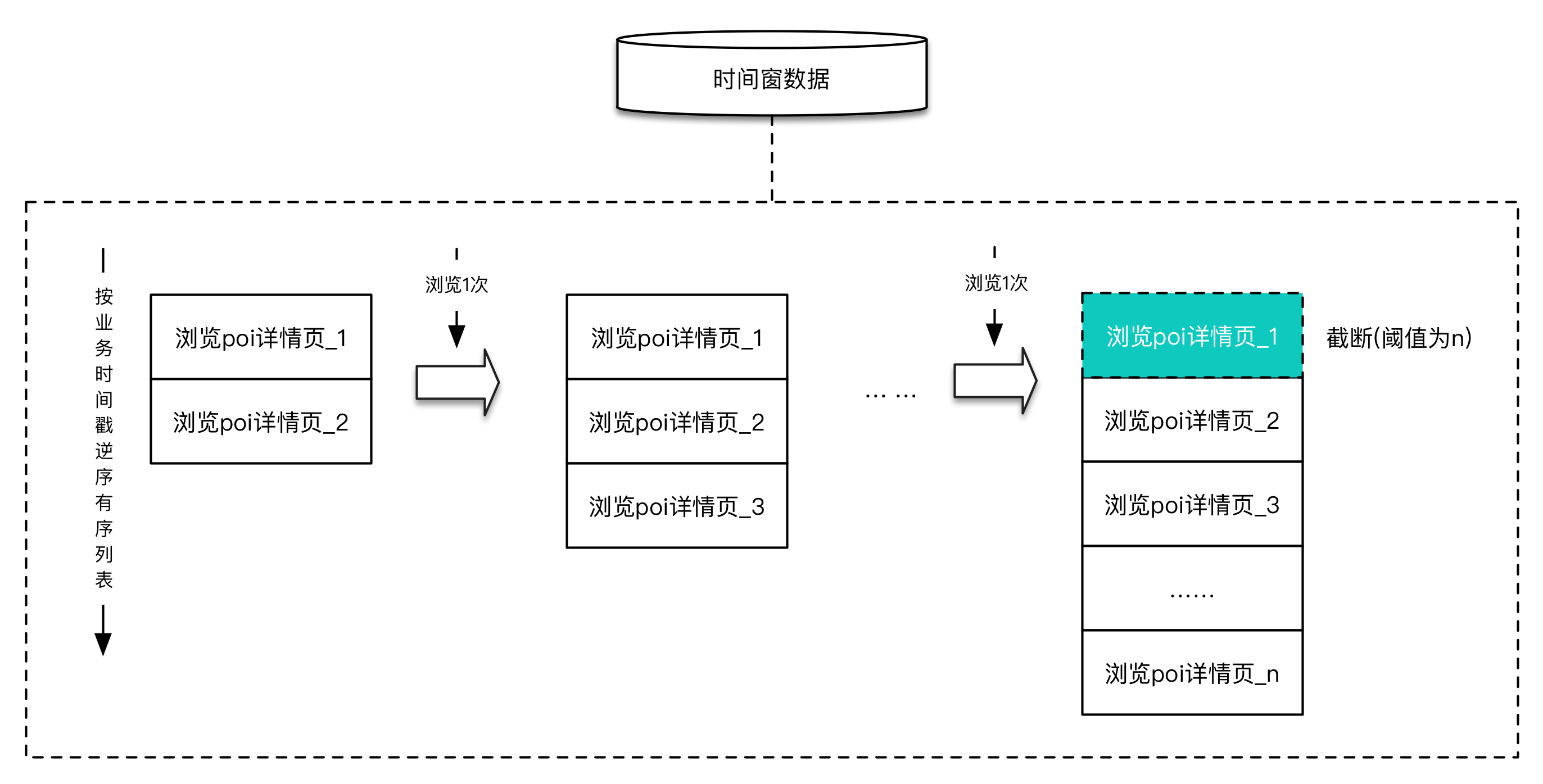
|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 类型 | 示例 | 因子构成 |
| 1 | **Count** | 近X分钟浏览POI大于Y次 | count(timeWindow(event.id, event.userId, X \* 60)) |
| 2 | **Distinct count** | 近X分钟浏览不同POI大于Y次 | count(distinct(timeWindow(event.id, event.userId, X \* 60))) |
| 3 | **First** | 近X天成交的首单产品 | first(timeWindow(event.id, event.userId, X \* 60)) |
| 4 | **Last** | 近X天最后一次搜索的产品 | last(timeWindow(event.id, event.userId, X \* 60)) |
| 5 |  |  |  |
| 6 |  |  |  |
| 7 |  |  |  |
| 8 |  |  |  |

根据时间窗因子类型可以看出，时间窗因子有以下特点：

* 时间窗存储中需要以List形式保存时间窗详情数据，以分别支持聚合及详情需求。
* 时间窗因子需要天粒度持久化，并支持EXPIRE。
* 时间窗因子应用场景多，是许多规则的重要组成因子，服务承受的压力较大，响应时间需要在ms级别。

对于以上特点，在评估使用场景和接入数据量级的基础上，我们选择KV存储服务Redis存储时间窗数据，经测试其在5K QPS请求下，TP99能保证在2ms左右，且存储方面性价比较高，可以满足系统需求。

在实际运营活动中，对时间窗内用户某种行为次数的判断多数在5次以内，结合此业务场景，同时为避免Value过大影响读写响应时间，在更新时间窗数据时设置阈值，对超出阈值部分进行截断。时间窗数据更新及截断流程如下图所示：



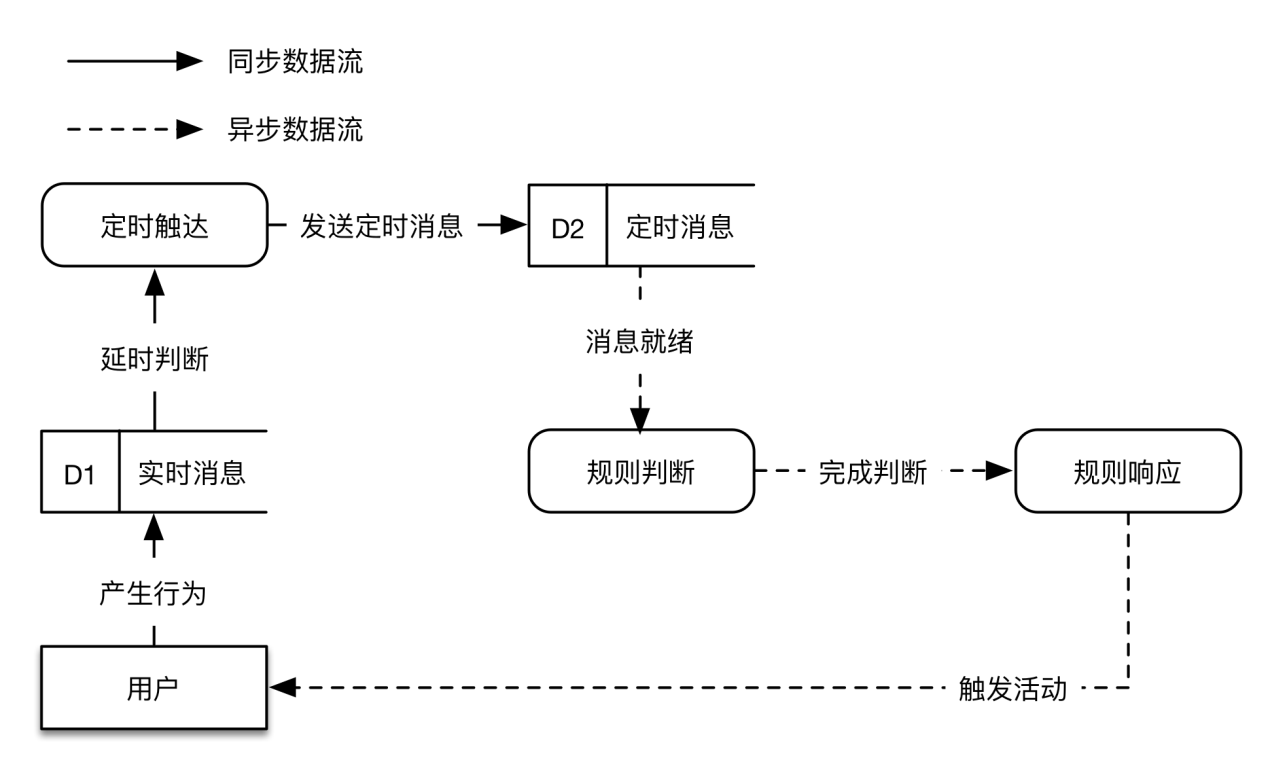
### 规则引擎扩展组件

**定时触达模块**

定时触达模块支持为规则设定定时执行时间，延后某些规则的执行以满足运营活动规则。

最前面背景中提到的业务场景，在 **用户在A行为后60分钟内未发生B行为** 条件中，需要在A行为发生60分钟后，对用户是否发生B行为进行判定。

定时触达模块涉及的数据流图如图所示：



**消息中间件Kafka支持消息粒度延迟**，非常贴合我们的使用场景，因此采用此特性，实现定时触达模块。

**自定义函数**

自定义函数可以扩充功能，**规则引擎可通过自定义函数执行因子及规则条件**，如调用用户画像等服务。系统内为支持运营需求扩展的部分自定义函数如下表所示：

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 示例 | 含义 |
| 1 | **Equals** | equals(message.orderType, 0) | 判断订单类型是否为0 |
| 2 | **Filter** | filter(browseList,‘source’,‘dp’) | 过滤点评侧浏览列表数据 |
| 3 | **poiPortrait** | poiPortrait(message.poiId) | 根据poiId获取用户画像数据，如等级属性 |
| 4 | **userPortrait** | userPortrait(message.userId) | 根据userId获取用户画像数据，如用户常住地城市、用户新老客属性 |
| 5 | **userBlackList** | userBlackList(message.userId) | 根据userId判断用户是否为黑名单用户 |
| 6 |  |  |  |
| 7 |  |  |  |
| 8 |  |  |  |

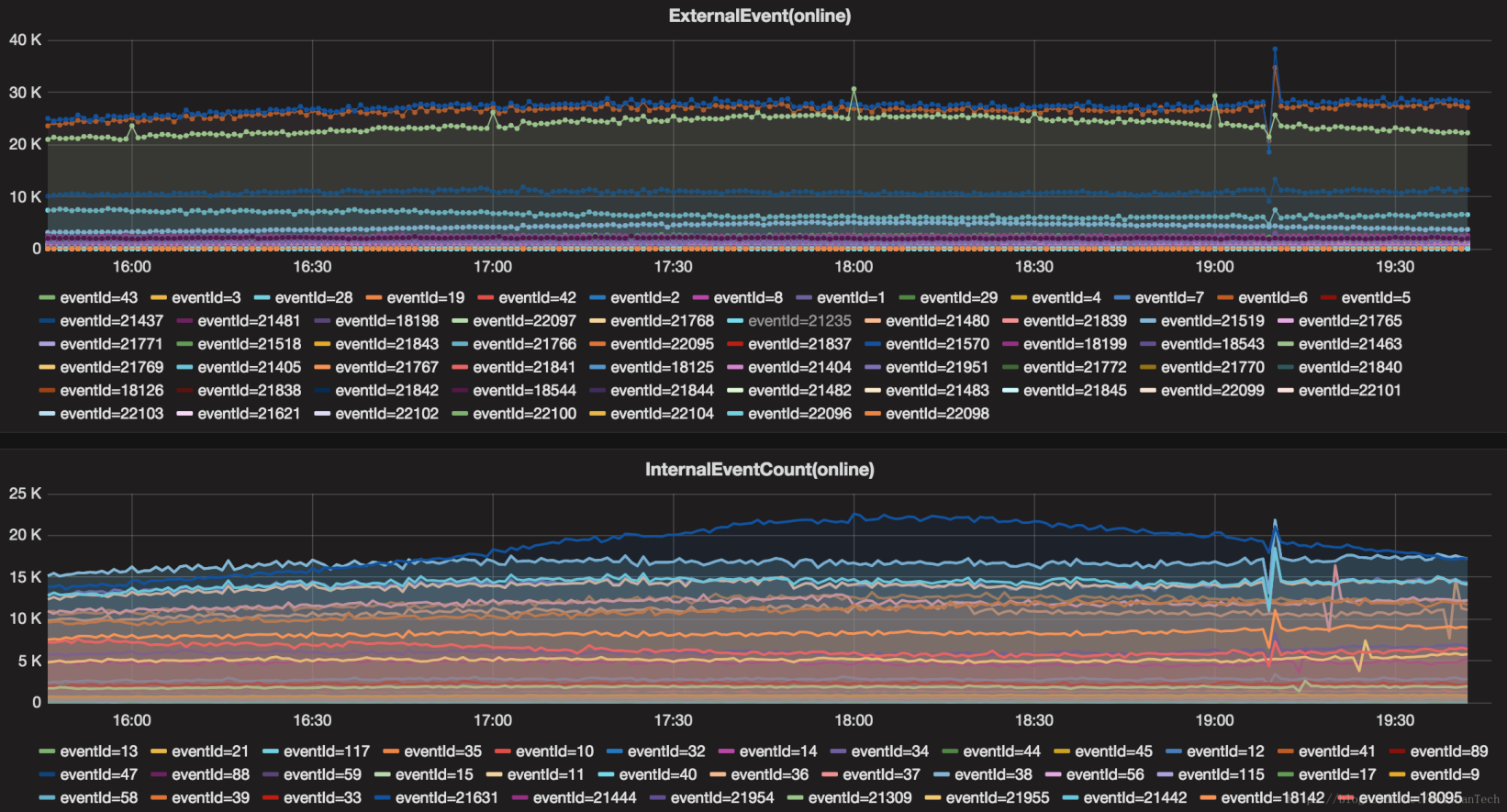
### 监控与报警

对比离线数据，实时数据在使用过程中出现问题不易感知。

由于系统针对的运营活动直接面向C端，在出现系统异常或数据质量异常时，如果没有及时发现，将会直接造成运营成本浪费，严重影响活动转化率等活动效果评估指标。针对系统稳定性问题，我们**从 监控与报警 两个角度入手**解决。

**监控**

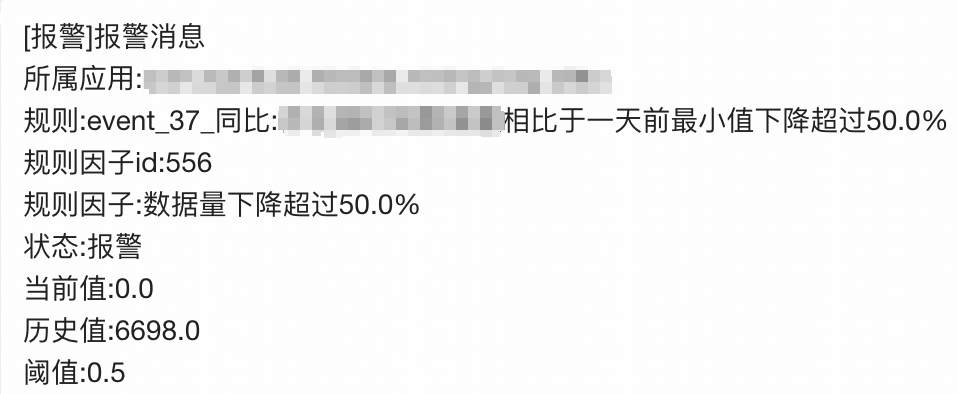
利用公司数据平台现有产品，对系统处理的实时事件按其EventID上报，以时间粒度聚合，数据上报后可实时查看各类事件量，通过消息量评估活动规则和活动效果是否正常，上报数据展示效果如图所示：



**报警**

监控只能作为Dashboard供展示及查看，无法实现自动化报警。由于用于监控所上报的聚合数据存储于 **时序数据库OpenTSDB**中，我们基于OpenTSDB开放的REST API，定制报警模块，定时调度、拉取数据，对不同事件，按事件量级、活动重要性等指标，应用**环比、绝对值等不同报警规则及阈值**。超出设定阈值后，通过消息平台及时发送报警信息。

如下图所示，该事件环比出现数据量级下降，收到报警后相关负责人可及时跟踪问题：



目前，大部分实时规则的计算时间从数据触发到计算结果输出的时间都控制在60ms以内。

## 实时流处理平台

客户行为实时分析、实时营销、实时风控等系统的核心是对数据的**实时采集、实时计算和实时查询**服务。站在全行大数据建设角度看，因此首先需要构建基础的大数据实时流处理平台。通过实时流处理平台的建设，构建全行统一的实时数据存储与应用平台，实现对行内多项业务的全方位支持。以此平台为基础，可以**为各种关注不同业务主题的应用提供实时流处理服务**。

大数据实时流处理平台的整体逻辑架构如下图所示：



平台整体逻辑架构划分为：**数据源层、平台层、流计算层、分布式实时数据总线服务层、应用层** 五个逻辑层次。其中：

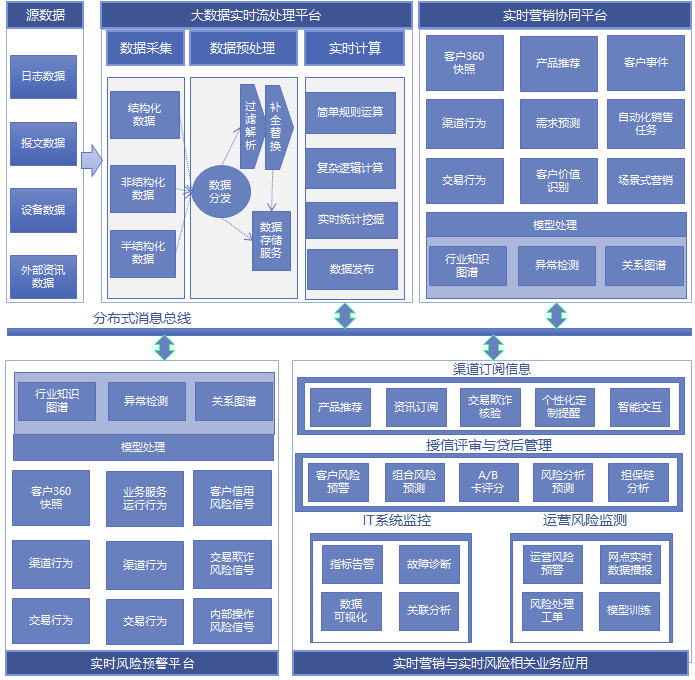
* **数据源层**：提供多种形式的数据实时采集功能，向平台推送数据。
* **平台层**：一方面对所有组件做了模板化封装，以方便灵活部署在Docker容器或VM虚拟机上，使平台具备弹性扩展能力，另一方面实现与恒丰银行大数据平台无缝对接，将结果数据写入大数据平台或者从大数据平台实时获取数据。
* **流计算层**：**是整个平台的核心层**, 底层基于SpringCloud和Spark Streaming开发框架构建，除实现了基本的流处理计算功能外，还构建了Streaming SQL模块，分析人员可以通过该模块直接在流上执行SQL语句，进行流上的数据关联查询；Streaming MLlib模块提供了机器学习模型在流上的植入功能，可以将构建的数据挖掘模型随时运行在流平台之上；Streaming Cube模块可以通过流实时构建多维度数据分析的数据立方体，进而在Cube上进行数据下钻、上卷等多维度切片分析。综上所述，流处理平台即支持OLTP的实时类业务应用，也支持近实时OLAP分析类业务应用，且支持数据挖掘类业务应用。
* **分布式实时数据总线服务层**：为流处理平台对外提供基础服务层，每个服务均以微服务的方式部署发布，具有弹性扩展和高可用特性。
* **应用层**：该层主要体现流处理平台在恒丰银行现阶段支撑的业务应用。

流处理平台使用的主要技术组件：

* Zookeeper集群，主要负责配置流处理平台管理和任务队列管理功能；
* Flume Collector集群，主要负责流处理平台文件类数据的接入工作；
* Streaming集群，为流处理平台的核心组件集群，主要负责数据流处理功能；
* Kafka集群，主要负责消息订阅与消息传输缓存功能；
* Redis集群，主要负责消息订阅，数据分析及数据查询功能；
* 大数据平台TDH集群，主要负责数据的存储、同步和数据分析挖掘等功能。

同互联网公司相比，商业银行在业务种类、交易模式、监管要求方面都大大不同，针对行内业务需求设计了以下功能架构：流处理平台提供基础的数据采集、接收、过滤解析、实时规则计算、存储和分析挖掘功能，以此为基础构建实时营销平台、实时风险预警平台进行业务逻辑加工，行内的各个渠道系统、信贷系统、IT运营监控系统、运营风险监测通过订阅方式获得实时处理后的数据，满足IT运营实时监控、客户行为分析、交易反欺诈、授信评审与贷后预警、运营风险监测等方面的业务需求。

如下图：



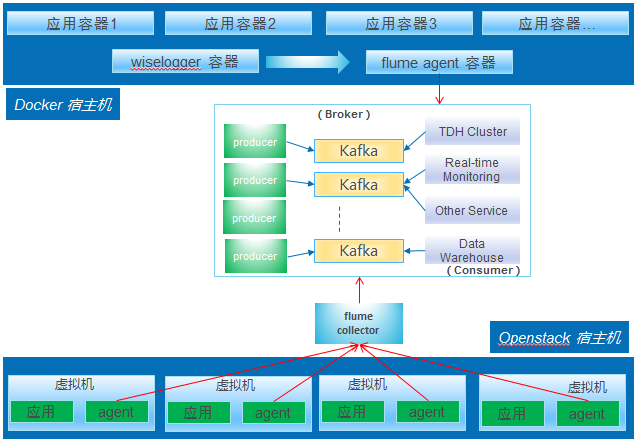
## 日志采集系统

当前的实时数据源主要包括业务系统的应用日志，企业消息总线关联交易日志，第三方数据公司实时推送数据，网络设备、操作系统、中间件日志，数据库日志，外部网站爬虫信息，流处理平台要负责实现以下目标：

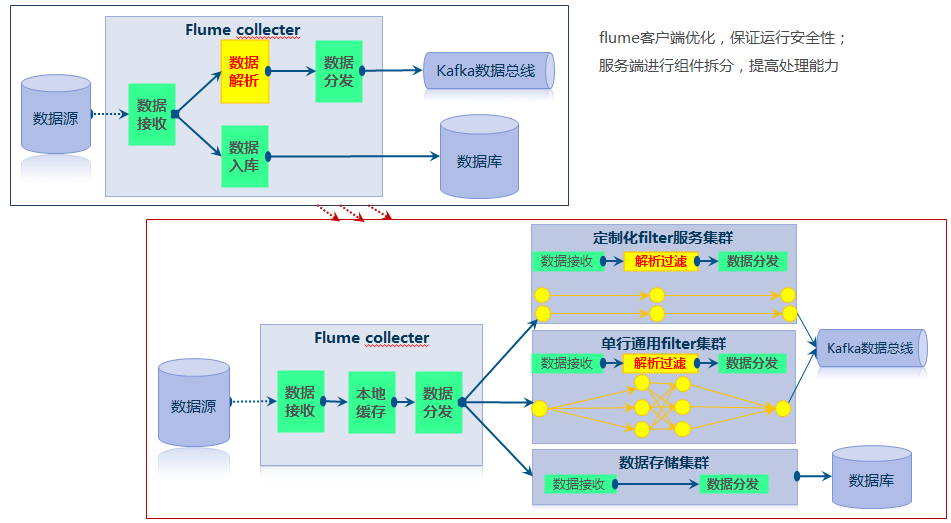
* 各类基础数据的实时采集，过滤解析；
* 根据业务应用需要提供基础数据实时加工功能；
* 同时支持简单和复杂业务逻辑规则模型，支持基于流数据的实时分析；
* 便于与异构系统集成，实现数据共享，要包含与主流的流计算框架、各类数据库、前端框架、消息中间件设施、主流接口协议；
* 海量数据的持久化存储和快速检索；
* 提供平台级别的数据管理功能，包括数据脱敏、用户权限、数据时效管理和分级存储等方面功能。

采用Java语言开发，在高可用、资源隔离、二次开发方面具有优势的flume作为我们的日志采集组件。

在日志收集流程方面，针对不同云环境设计了不同的采集流程，对于部署在openstack云环境和部分物理机上的应用，通过在应用服务器直接部署flume agent实时采集每个节点配置的日志文件、syslog、进程状态信息，实时发送后端的flume 服务端，服务端完成原始数据入库和初步的过滤解析并发送到kafka消息中间件；对于部署在docker容器环境下应用，因为已经做了日志规范化，宿主机上的flume直接利用宿主机提供的接口读取对应镜像所属路径下的日志文件，直接发送到kafka消息中间件。



此外，根据实际需要，对flume的客户端进行开发，增加了source种类，覆盖目前所有应用系统的日志打印类型，修改了客户端软件的一些缺陷，设计并实现了不同策略，控制异常情况下对系统资源的占用，修改了agent与zookeeper之间更新配置的方式，利用心跳机制实现对flume agent状态的监控，利用Jenkins、puppet等工具支持进行大规模客户端推送、部署。



为了解决flume服务端收集数据的性能问题，我们对flume服务端进行微服务化拆分，将原来集中在服务端的接收、解析、入库工作拆分出来，原有的flume服务端用其对loadbalance、failover以及与客户端之间发送接收事务的支持进行数据接收，构建微服务组件集群实现解析和入库的功能，并根据不同级别、不同类型的解析、入库需求分配不同的组件。

除此之外，为了补充采集一些既有系统交易数据和爬取外部网站一些数据，平台提供探针组件和爬虫组件，并基于zookeeper实现了上述客户端程序组件的注册、注销、实时任务分配，实现了高可用和水平扩展。

## 客户标签和客户画像

客户标签体系是构建用户画像精准营销、产品推荐、反欺诈、社交化等各种业务的数据基础。



### 何为客户标签

给客户贴标签是大数据营销中常用的做法，诸如“商务人士”、“育婴妈妈”、“在校学生”、“奢侈品粉丝”等客户标签早已在互联网企业中建立，借助客户标签，互联网企业可实现广告营销活动的精确推送。商业银行也开始尝试通过深入的数据分析和挖掘，洞察客户行为、喜好，给客户打上各种类型的标签。

合理准确的客户标签的背后是银行对客户全方位信息的深入理解与认知。在这个过程中，银行可以发现哪些潜在客户对营销活动响应度高；哪些客户接受新产品困难，只钟情于传统业务；哪些客户信用度低、风险高或存在欺诈可能。准确勾勒客户轮廓需要结合银行内部数据、社交媒体数据、外部公共数据等多维度数据，深入分析、挖掘后获得潜在客户知识，并依据业务目标对客户进行分类细化，采用类自然语言方式对客户进行描述。

如：客户标签“手机消费达人”=统计周期[当年（自然年）01月01日至今]+渠道偏好特征[通过手机银行渠道办理业务]+服务偏好特征=[缴费交易笔数≥4]+交易对手特征[支付宝商户]。

一个客户标签通常是一个或多个客户特征的集合，构成集合的特征也称为业务特征规则，是表达客户标签规则的原子组成。

### 客户标签体系的构建

对商业银行而言，基于客户特征集合形成的客户标签有成百上千、甚至成千上万个，这些标签在构建时的业务目的和适用场景各有不同。随着应用标签的场景越来越丰富，商业银行也会逐渐形成一套完整的客户标签体系。

基于行业特性，客户标签有着不同的分类方法。如业内常见的传统分类方法，将人口统计、兴趣爱好、社会属性和金融特征作为客户的二级标签，在金融特征下细分出产品偏好、渠道偏好、交易偏好等三级标签。与大数据结合后，分类方法增加了基于互联网和外部数据的客户生活标签，在这个标签下细分出生存需求、关系需求、成长需求和互联网等多个三级标签，在互联网标签中又细分出访问偏好、内容偏好、行为特征等四级标签。本文重点讨论在大数据环境下支持商业银行进行精准营销的客户标签体系建设。

商业银行的数据不仅包括传统交易系统的结构化业务数据，还包括来自银行自身电子商务网站、网上银行、手机银行和外部社交媒体网站、即时聊天工具、微博、微信等渠道的半结构化数据以及日志等文本信息和来自客户服务渠道的各种语音、图片、影像等非结构化数据。建立客户标签体系可对不同渠道、不同口径、不同来源、不同结构的客户信息进行统一分类管理。根据商业银行客户的特性，结合大数据思维，客户标签体系可分为五大类：**人口统计、兴趣爱好、社会属性、金融特征和互联网特征**（如图所示）。



1. **人口统计维度**

根据人口统计学知识定义的客户基本信息，包括性别、年龄、收入、人生阶段、子女情况、父母情况、婚姻情况、宗教信仰、民族、国籍、籍贯、教育水平等。

1. **兴趣爱好维度**

兴趣爱好是指一个人表现出的对现实的态度、行为方式以及心理特征，可分为娱乐偏好、生活偏好、文化偏好、性格情绪和消费心理五个方面。其中，娱乐偏好是指对娱乐项目的喜爱，如音乐、戏曲等；生活偏好是指对日常生活事物的喜好，如宠物、家居等；文化偏好是指对文化信息方面的喜爱，如阅读、摄影等；性格情绪是指个性和心理方面的特点，如有爱心、重感情、急躁等；消费心理是指在消费购物方面的行为特点，如网购偏好、促销偏好、假日旅游偏好等。

1. **社会属性维度**

社会中不同类别的人会以家庭、邻里、朋友等群体形式生活，社会属性维度就是描述社会群体的一些特性，主要包括生活特征、工作特征和社交特征。其中，生活特征是指居住区域（如市中心、郊区）、是否购车、是否购房等；工作特征是指工作区域、工作性质、行业类别、工作职务等；社交特征是指交友情况（友人众多、宅男等）、社交圈属性（如高端知识分子、篮球爱好者等）、人群归属（如大学生群体、恋爱群体等）。

1. **金融特征**

金融特征是指客户与商业银行交互过程中的银行资产、产品持有、渠道使用、投资偏好等与资金账务、交易往来相关的金融特征信息，这类数据可以来自一线业务、汇总信息或挖掘分析等多种渠道。金融特征可分为资产信息特征、收入贡献特征、产品偏好特征、消费行为特征、渠道偏好特征和生命周期特征。其中，资产信息特征包括客户AUM、各类产品的日均值和时点值；收入贡献特征指客户通过贷款、中间业务收入等给银行带来的利润贡献，可分为高价值客户、中高价值客户、低价值客户等，可与不同产品进行交叉组合；产品偏好特征包括产品持有及投资、服务偏好特性，如定期持有、生活缴费偏好、银行理财偏好等；消费行为特征包括消费地理位置、消费时段、品牌偏好、广告营销偏好和消费热点偏好等；渠道偏好特征是指客户与银行交互往来的渠道特性，如柜台偏好、网银偏好、手机银行偏好等；生命周期特征是指客户与银行从接触到销户的不同阶段，包括获取期、提升期、成熟期、衰退期和流失期。

1. **互联网行为特性维度**

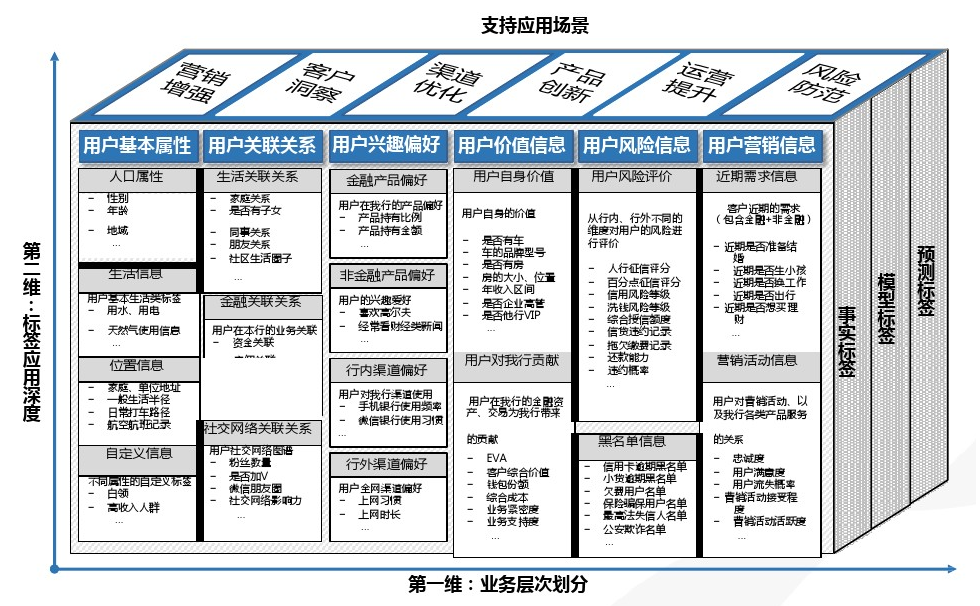
互联网行为特性维度是指客户在互联网或移动终端的访问内容、操作行为的相关信息。这类信息包括银行自有网站、手机APP的动态数据收集、从第三方机构（如蚂蚁金服、京东、携程）购买的客户互联网交易或操作行为数据。一般可分为访问终端特征（如终端类型、终端品牌等）、内容偏好（如偏爱访问信用卡频道、理财频道等）、操作行为偏好（如目标直达型、悠闲型等）。

标签还可以按照处理过程、标签获取的方式进行划分，分为**事实标签、模型标签、预测标签**。不同类别的处理方式不一样。

**事实标签**：直接从原始数据中提取，例如性别、年龄、住址、婚否、职业等

**模型标签**：需要建立模型进行计算，例如可支配收入、风险偏好度、是否频繁转账、是否归属灰名单

**预测标签**：通过预测算法挖掘，例如最近是否有借款需求，是否存在账户资金风险、



例如，建设银行将个人客户画像标签归纳为七大类：

**基础信息、信用与风险、金融行为、非金融行为、联系情况、渠道特征、客户分析**

共计2229种标签。

### 客户标签应用平台

从构建一个客户标签到形成一套客户标签体系，需要搭建客户标签应用平台。

客户标签应用平台将客户标签体系（常用标签信息）与360度客户信息视图的基本数据有机整合，可基本满足业务部门日常数据分析需求。客户标签应用平台还提供自助信息获取和调用经验规则获取信息的功能，并可将规则进行知识沉淀后复用；此外，营销反馈的结果也被纳入客户标签应用平台作为标签体系、规则优化的补充。

客户标签应用平台分为 **客户标签获取、客户标签库、客户标签应用** 三个层次（如图）。



**客户标签获取层**

负责客户标签的数据收集，主要通过调查研究、一线人员收集（如支行网点、电话服务热线或客户经理等）、客户接触（包括业务办理接触及营销活动接触等）、网络标签规则（如维基百科、电商标签库）、业务营销经验、数据仓库信息层、数据分析及挖掘建模等渠道或技术手段，积累客户特征规则。

**客户标签库层**

负责客户属性、特征规则和客户标签的存储。尽管借助客户标签可以快速进行营销定位，但由于标签是一个浓缩的信息项，若要展开客户画像和分析挖掘，则略显不足，因此还需结合360度客户信息视图的明细数据，以确保客户信息的完整性。此外，由于外界环境和客户行为信息不断变化，为确保客户标签数据的准确性、时效性和高质量，应当做好客户标签的基础管理工作，建立标签创建、编辑、审批、发布、执行、评估和下线的生命周期管理流程，实现标签管理的体系化。

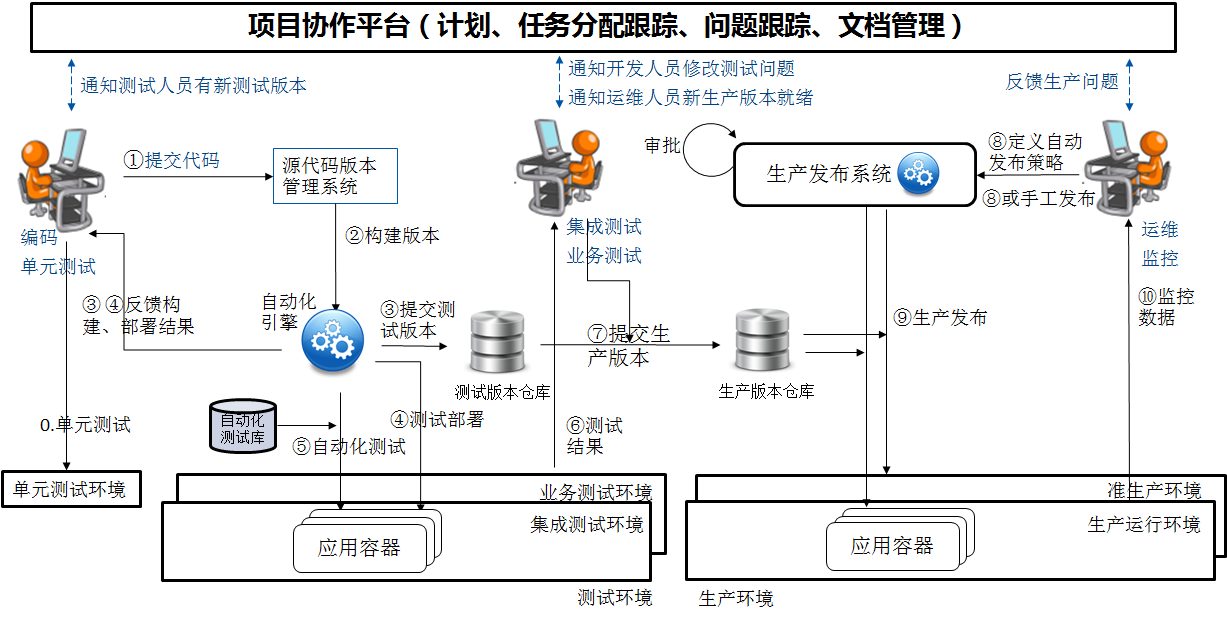
**客户标签应用层**

是利用客户标签实现客户画像、客户群细分、精准营销、客户价值提升、分析挖掘和信息快速推送，其应用对象包括二线的产品经理、营销人员、数据分析人员及一线的客户经理、网点柜台等。

## 微服务化和Docker容器化

客户行为实时分析系统还做到了微服务化和docker容器化。我们将SpringCloud微服务架构的集群负载，服务监控、故障恢复与弹性部署能力结合银行数据中心PaaS云平台采用的docker容器技术对应用级负载、监控、弹性资源分配以及快速部署能力相结合，对客户行为实时分析系统的Xitrum、JDBC、 Redis 等功能组件进行了拆分，实现了整个应用的docker容器化。

流处理平台使用的flume、kafka、zookeeper等开源组件也可以实现docker容器化并使用DevOps工具服务，从开发、构建、测试到版本发布的全流程自动化，中间提供包括计划、任务分配跟踪、问题跟踪、文档管理、版本发布全过程的项目协作支持。



## 图数据库与图计算

# 应用场景

既有为当前业务运营发展提供数据支持的客户规模和质量、渠道运营分析模块、实时状态跟踪、分析中心等功能模块，也有为未来业务营销发展打下数据基础的个体行为分析、客户属性分析等功能模块，还有从提升客户体验角度出发，为行内应用优化方向提供数据证据的有客户参与度分析、功能分析 等模块。具体介绍如下：

* **整体运营**
* **客户规模与质量**：通过分析某些功能模块的客户使用情况，了解客户产品偏好，分析客户结构和质量信息，从而为银行业务决策打下数据基础。
* **渠道运营分析**：联动多个应用的使用情况，提供整体的运营分析情况。
* **实时状态跟踪**：通过实施检测当前客户使用应用的情况，可实时发现例如服务异常等突发事件情况，从而做到及时预警，为事件预警和处理赢得时间。
* **分析中心**：为业务定制化输出各类分析报告，满足业务日常运营需求。
* **实时营销**
* **客户个体实时行为分析**：把单个客户的行为分析单独统计分析，从而为精准营销、交叉营销和即时服务等提供数据基础。
* **客户属性分析**：通过分析客户常用应用终端、客户常用地域等，分析客户基础属性信息，从而为营销服务打下数据基础。
* **客户体验提升**
* **客户参与度分析**：通过获取客户的使用时长、访问深度、应用各功能模块的使用等情况，分析客户应用使用情况信息，从而为应用优化等打下数据基础。
* **功能分析**：分析某个应用的客户最常用路径，了解客户常用行为，从而为优化路径提供数据说明。

## 客户行为实时分析

客户行为实时分析系统 通过对客户基本信息和行为数据的监测追踪、收集整合、评估分析，为业务人员决策业务策略时提供更全面、更准确、更有价值的信息。从不同的角度出发，客户行为实时分析系统需要达到不同的目标。

在全行的大数据建设规划角度出发，客户行为实时分析系统首先要实现两个目标：

* **构建面向全行客户的群体客户行为数据基线**
* **可以实时的发现和处理个体客户的差异性行为数据**

前者是基础，后者则是价值挖掘对象。

**从业务角度出发**，客户行为实时分析系统 主要需要达到或支持以下三个角度的业务要求：

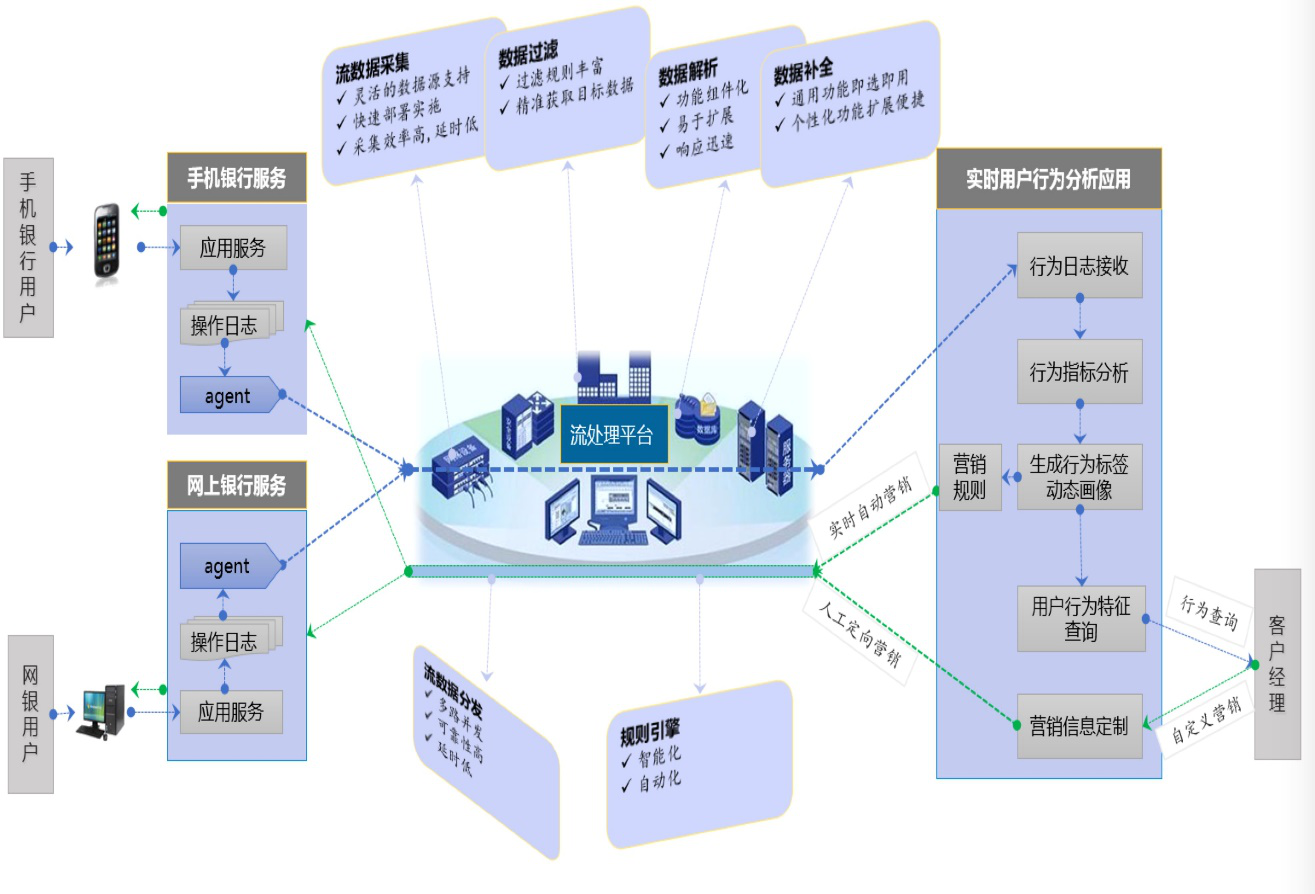
* + - * 从营销角度，可以深度了解和分析客户在持有、购买、放弃产品及应用的前、中、后期行为特征，从而为客户偏好、360度画像、数据挖掘提供更精准的数据依据，最终实现精准营销。
      * 从风控角度，可以深度了解和分析客户的活动范围、生活轨迹、交易支付习惯等，从而为信贷风控提供有力证据。
      * 从客户体验角度，一方面可以深度了解和分析银行现有应用的运行状况、应用中各个功能模块的触达频率、客户访问路径长度等情况，从而为银行应用设计和优化提供数据支持，让银行应用产品更符合客户群体期望和个体期待；另一方面为客服系统、CRM、移动柜员等前端应用提供更加精准的客户信息，方便一线业务人员为客户带来更加个性化、定制化的服务体验，提升银行客户粘性。

**从技术角度出发**，面对全量客户海量行为数据，客户行为实时分析系统要达到如下目标：

* 数据要能够实时的采集和存储
* 数据要能够实时计算和分析
* 实时数据服务能够实时展现和查询

前端应用（如手机银行、网银等）客户操作日志是文件类数据，对于这类文件类数据流处理平台主要采用Flume client & server 模式，因此前端业务系统需要部署agent实时采集数据，Flume Collector 负责收集各agent数据，再将数据发送至流处理平台，随后由流处理平台负责数据过滤、数据解析、数据补全等实时处理工作，最后客户行为实时分析系统负责对实时性指标进行实时分析，对于非实时性指标则进行离线分析，并最终将分析结果以图表形式展现给。

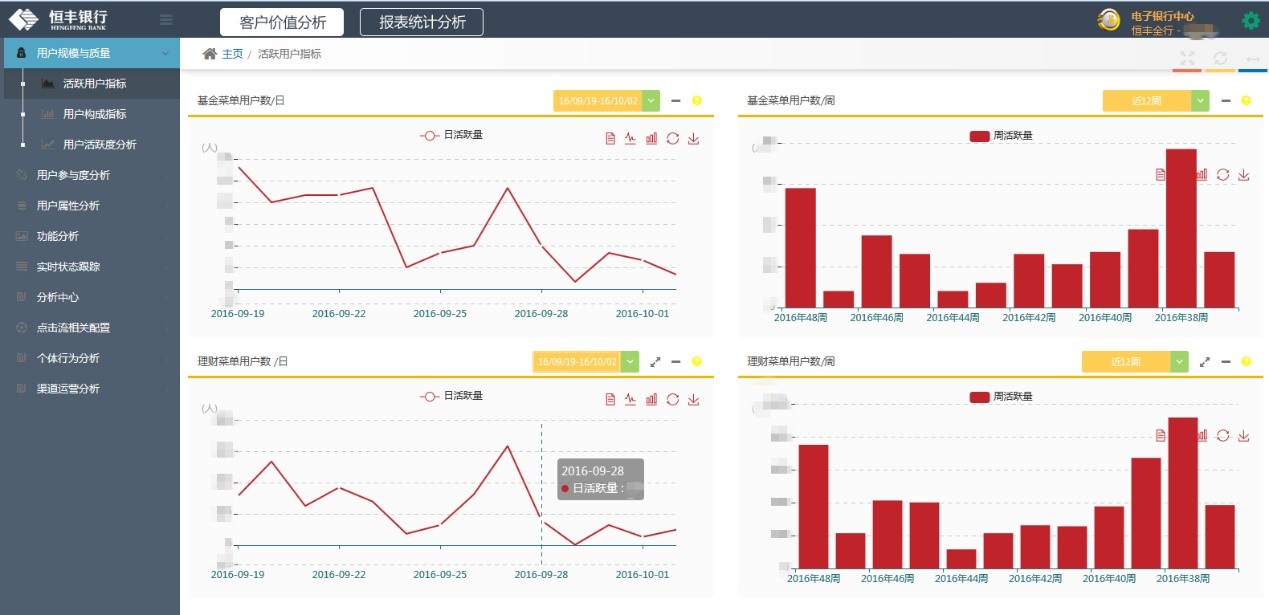
客户行为实时分析系统整体逻辑架构如下图所示：



客户行为实时分析系统填补了银行在客户行为分析方面的空缺，业务人员能够直面感受和了解客户真实的应用使用行为状态。例如：



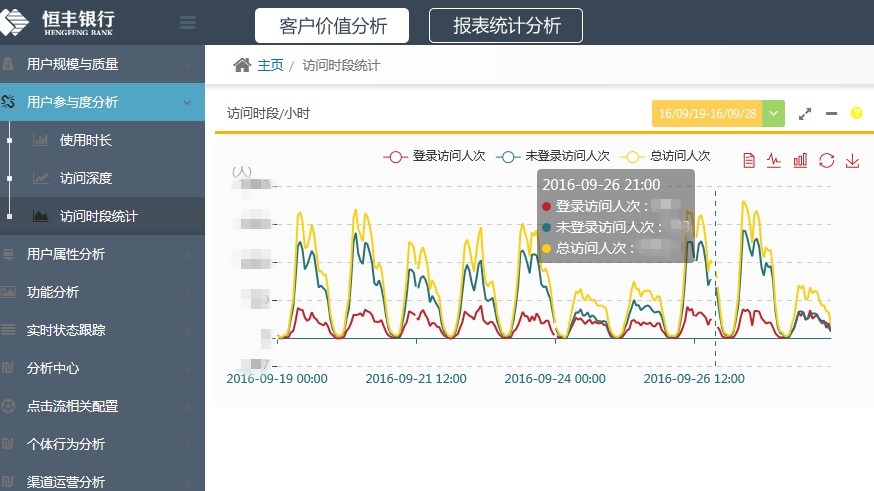
### 客户规模与质量：活跃客户指标（部分）



活跃客户指标之基金、理财模块客户每日活跃、每周活跃情况

通过分析理财、基金等功能模块的客户活跃指标，观察银行理财偏好客户在全行占比情况，以及理财、基金产品的关注情况，从而更好的了解银行客户的规模和质量。

### 客户参与度分析：访问时段统计



客户访问时段分布之按小时分布

该功能呈现选定时间段内的登录客户、匿名客户、客户整体访问时段分布情况。如果按小时统计，通过分析可以明显看出每天都会呈现三个客户活跃高峰，分别对应早、中、晚的某个特定时段，其中每天早间时段客户最为活跃。如果按天观察，可以看出，周一、周二客户最为活跃，周六、周日客户最为不活跃。基于此，可以让业务更好的制定或优化营销策略，例如加大周一、周二早间时段的渠道营销力度，可能会获得更佳营销效果。

综上，借助客户行为实时分析系统，从全渠道运营上，渠道业务人员能够随时了解渠道系统客户的地域分布、访问时段分布、交易类型分布、关注产品分布、终端使用分布信息等，从而判断产品运营情况，提高产品运营水平；从精准营销上，渠道业务人员能够随时了解单一客户的理财产品偏好、基金产品偏好、地域偏好、功能偏好、访问时段偏好等行为画像信息，从而为后续精准营销、反欺诈等打下坚实数据基础。

基于上述数据积累，我们还在流失预警、客户分群、消费周期模型、理财产品销量预测模型方面也进行了一系列模型挖掘研究。

### 其他

产品关注度、服务关注度、首页广告、用户反馈四大功能模块

**产品关注度**

## 精准营销

应用目标是希望依托大数据平台的分析结果在线获取客户的各类信息，再通过模型分析、客户个性化需求、不同产品的特点等，在客户与产品之间建立精准的对应关系。

在业务操作上，还应当能帮助客户经理对客户做出准入判断、提供营销方案、实现限额管理、定价指导等智能决策信息。这些应用目标可以拆分成如下应用需求：

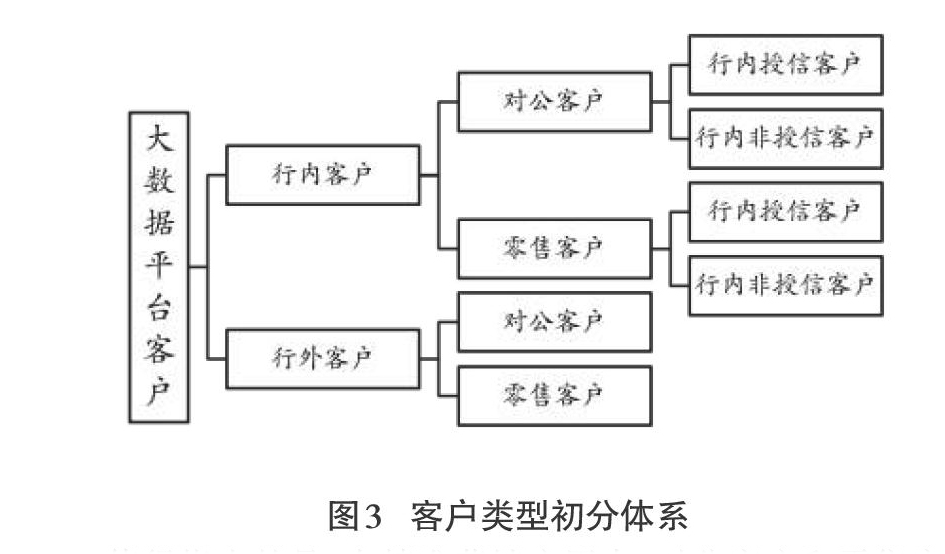
* **客户挖掘**。精准营销的首要目的就是寻找目标客户，只有寻找到精准的营销对象才可能实现最终的精准营销结果。客户挖掘功能应当从不同角度、不同渠道进行挖掘，最大程度发掘潜在客户群。
* **智能决策引擎**。在获得客户挖掘结果后，精准营销应用的下一步功能应当是针对具体客户，提供一整套智能决策方案，包括客户准入判断、产品推荐、定价指导等。对于集团客户、上下游客户或关联方客户等，还应当能够提出整体的营销计划。
* **业务统一作业平台**。精准营销系统除了可以提供精准营销的方案、计划等决策结果，还应配备完整的业务操作平台完成业务实现。业务统一作业平台可以实现整个精准营销从客户挖掘直至业务完成的各工作环节，最大化程度缩短业务操作流程、减少贷前调查复杂程度，实现精准营销的“一站式”操作。
* **全生命周期的营销计划**。精准营销的应用不应只以单次营销为目标，而是应当从纵向和横向两个方向提供持久的精准营销计划。纵向的可持续是指跟随客户的成长过程，在客户不同时期提供适时提供客户需要的产品和服务；横向的可持续是指通过集团客户、上下游客户和其他关联方客户不断挖掘目标客户，扩大精准营销范围。



### 系统功能

下面针对应用需求分析，从系统实现上把应用需求分解到系统功能点上进行详细描述。

* **客户类型初分**。对于不同类型的目标客户，精准营销模型应当给出具有针对性的营销方案。因此需要对大数据平台里的所有客户进行类型的区分。建立合理的客户类型初分体系是精准营销的基础。客户类型初分可以按照图中给出的体系进行划分。



在精准营销应用中，对公客户和零售客户并非完彼此独立。完备的精准营销应用应当具备从对公客户中挖掘零售客户或从零售客户中挖掘对公客户的功能。

* **准入客户筛选**。客户准入筛选是对客户质量进行把关的工具，良好的客户准入筛选标准是确保客户质量、引导客户结构的保障。实践中考虑根据大数据平台的**黑白灰名单**对客户的准入进行判定：原则上白名单客户属于精准营销判定的推荐客户；黑名单客户则是禁止准入的客户；灰名单客户可设为风险提示类客户。
* **客户信息整合及验证**。利用大数据平台打通内外部数据、不同业务数据、不同结构数据之间的壁垒，形成以客户为中心的“一户一条”数据记录。

为确保精准营销结果的准确性，还应建立严密的客户信息验证机制。客户信息校验包含两方面：

* 错误信息的验证。如同一企业在不同系统中的行业分类存在差异，大数据平台将设置校验规则判定哪一来源数据正确，并自动用正确数据覆盖错误数据；
* 数据时效的验证。如大数据平台有多个来源提供某个企业的营业执照有效期，系统将自动获取最新的日期作为改企业的营业执照有效期。
* **客户及场景标签设定**。根据精准营销的不同角度设定不同类别的标签，标签应当具有灵活性，可以随着业务发展和精准营销场景的变换随时增减或改变。标签的设定主要包含以下几类：
  + 客户标签：主要区分客户的属性特征，如职业信息可设定诸如企业高管、普通白领、职场新人、全职妈妈、小企业主等。
  + 事件标签：主要根据客户生活场景或者银行业务场景设定标签，如对于客户生活场景可能出现的标签包括买房、买车、装修、出国等。
  + 行为标签：主要针对客户日常规律的行为情况设定标签，如对客户上下班的出行情况可以区分成地铁族、打车族、开车族、走路族、骑车族等。
  + 偏好标签：主要对于客户的兴趣爱好等设定标签，如网购达人、资深吃货、旅游爱好者、大牌控等等。
* **精准营销模型设计**。利用SAS等建模工具设计算法，建立精准营销应用模型。精准营销模型可拆分为多个子模型，部分子模型属于精准营销内部模型，如客户准入模型、行为分析模型、业务推荐模型等；还有部分子模型是调用其他系统的模型结果，属于精准营销调用的外部模型，如定价模型、客户评级模型、限额模型等。

整体的精准营销模型是结合客户、产品、事件、行业及发展情况等各方面，按需调用不同的子模型进行进一步计算，获得最终的精准营销智能决策方案。

* **业务统一工作平台**。根据应用目标中所述的业务统一工作平台应用目标，该模块的开发包含但不限于以下功能：
  + **营销推荐**。营销推荐功能应当从两个角度提供推荐信息：一是根据产品特点推荐适合的客户；二是根据客户偏好推荐适合的产品。无论哪个角度的推荐，系统推荐的都应当以精准为评判标准。
  + **客户准入判定**。对于客户经理自主开发的客户，精准营销系统应当可以提供客户准入的判定。客户准入的判定不仅包含客户的黑白灰名单信息，还应包含客户评级信息等其他可作为客户准入条件的信息。
  + **O2O的调查报告**。对于可准入的客户，系统可自动线上（on-line）生成标准模板的调查报告。如报告中需要获取的信息通过线上可自动采集，则直接生成在调查报告中，其余系统无法获取的内容则可由客户经理线下（off-line）调查核实后补充至调查报告中。这种O2O的调查报告生成方式可使数据的自动取数率高达80%以上，大大减轻了客户经理的工作量，同时也保证了报告内容的准确程度。
  + **客户跟踪管理**。为实现可持续的精准营销计划，系统应当提供客户跟踪管理模块，以便记录客户的业务偏好信息、成长变化曲线、贡献度信息、关联关系图谱等。
  + **精准营销高级应用**。除了上述功能外，为更全面和深入的利用大数据平台和客户画像结果，精准营销还应站在更高视角。目前设想的高级应用可以包括：行业整体的精准营销方案、产业链精准营销方案、公司联动营销计划等。

客户实时营销系统是运营/营销活动中十分重要的环节，起到承上启下的作用，在系统处理能力及对业务贡献方面取得了较好的效果：

* 平均日处理实时消息量近2000万。
* 峰值事件QPS 2K。
* 帮助大零售、信用卡中心等业务线开展了丰富的运营活动。
* 对转化率、GMV、拉新等指标促进显著。

当前系统虽然已解决了业务需求，但仍存在一些实际痛点：

* 实时数据接入非自动化。
* 规则引擎能力需要推广、泛化。

场景及规则编辑注册，未对业务/运营人员开放，只能由技术人员完成。

## 反欺诈

## 反洗钱

# 建设路线（Roadmap）

秉承“基础设施决定上层建筑”的观点，在数据仓库/集市、大数据平台、数据驱动优化运营的建设过程中，需要逐级逐步的“先修路，后开车”的层层递进建设，夯实软硬件基础平台，做好基础的数据治理，才能更好的为上层应用和运营奠定坚实的基础。

## 第一阶段

* 数据中心软硬件基础设施的建设
  + 基于x86或ARM架构（如：华为泰山服务器）的服务器集群、网络设备、防火墙、堡垒机等机房建设；
  + 基于MySQL/Oracle等关系数据库的传统数据库集群；
  + 基于MongoDB的NoSQL（Key-Value）数据库集群；
  + 基于Redis的数据缓存集群；
  + CDH/HDP/TDH/FusionInsight等开源或商业版本的Hadoop大数据平台；
  + 基于Teradata/Greenplum/LibrA的MPPDB数据仓库/集市；
  + 基于 Neo4J/OrientDB的图数据库集群；
* 数据源清单整理和基础数据导入
  + 整理数据源，包括行内数据源和外部数据源，注意其性质、结构、内容、容量的区别；
  + 选择合适的存储模式和物理节点，设计数据模型；
  + 分批将基础数据通过ETL处理导入对应的数据节点；
  + 对于online的流式数据源（如通过API接口、ESB、消息队列传播的数据源）
* 冒烟式的基础应用场景建设
  + 多维综合查询
  + 统一日志分析与运维监控平台

## 第二阶段

* 面向不同场景的应用系统建设
  + 用户画像与产品画像
  + 用户行为分析系统
  + 精准营销系统

## 第三阶段

# 附录

## 常见概念

|  |  |  |
| --- | --- | --- |
| 序号 | 概念 | 说明 |
| 1 | **POI** | Point Of Interest. 兴趣点。在业务系统中，POI可以是某种产品或服务。 |
| 2 | **堡垒机** | <https://baike.baidu.com/item/堡垒机/2196549> |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |
| 10 |  |  |

## 行业背景常识

1. **规则引擎（Rule Engine）**：
   1. 国内商用：
      1. 旗正：<http://www.flagleader.com>
      2. 锐道（URule）：<http://www.bstek.com/>
   2. 国外商用：
      1. Esper：<http://www.espertech.com/esper/>
   3. 开源产品：
      1. Drools：<http://www.drools.org>

本系统中如无特殊说明，采用的规则引擎为：基于URule社区开源版本上自行开发的本司自主分支版本；

1. **图数据库（Graph Database）**
   1. 国外产品：
      1. Neo4J：<https://neo4j.com/>
      2. TitanDB：<http://titandb.io>
      3. OrientDB：<https://orientdb.com/>
   2. 国内产品（尚不成熟）：
      1. 阿里云GDB：<https://www.aliyun.com/product/gdb>
      2. 蚂蚁金服GeaBase：<https://tech.antfin.com/products/GEABASE>
2. 其他问题，可联络 叶亮（15319967068）