产生背景

随着业务不断发展，各个业务沉淀了大量的活动玩法。当前业务团队各自分工，产生了各种零散的活动配置，每次任 务数值调整得改代码、部署，不好统一管理，同时对开发同学来说又是重复造轮子。不同业务系统埋点数据分散在各 自的系统中，使得联合分析业务数据变得成本昂贵，数据孤岛现象愈发严重。 活动中台设计方向主打提升开发和运营的效率，收口用户行为埋点，提炼通用的模板和活动配置，从而减少各业务重 复性开发，降本增效。

2.设计理念

开发一套活动中台系统，统一配置公司活动信息&&任务内容&&兼备奖励下发。

统一管控，数据收口

事件定义 组成任务列表

活动列表

活动配置基础信息

活动配置任务玩法和奖励

各种玩法有各种配置和规则

触达配置 消息模板等

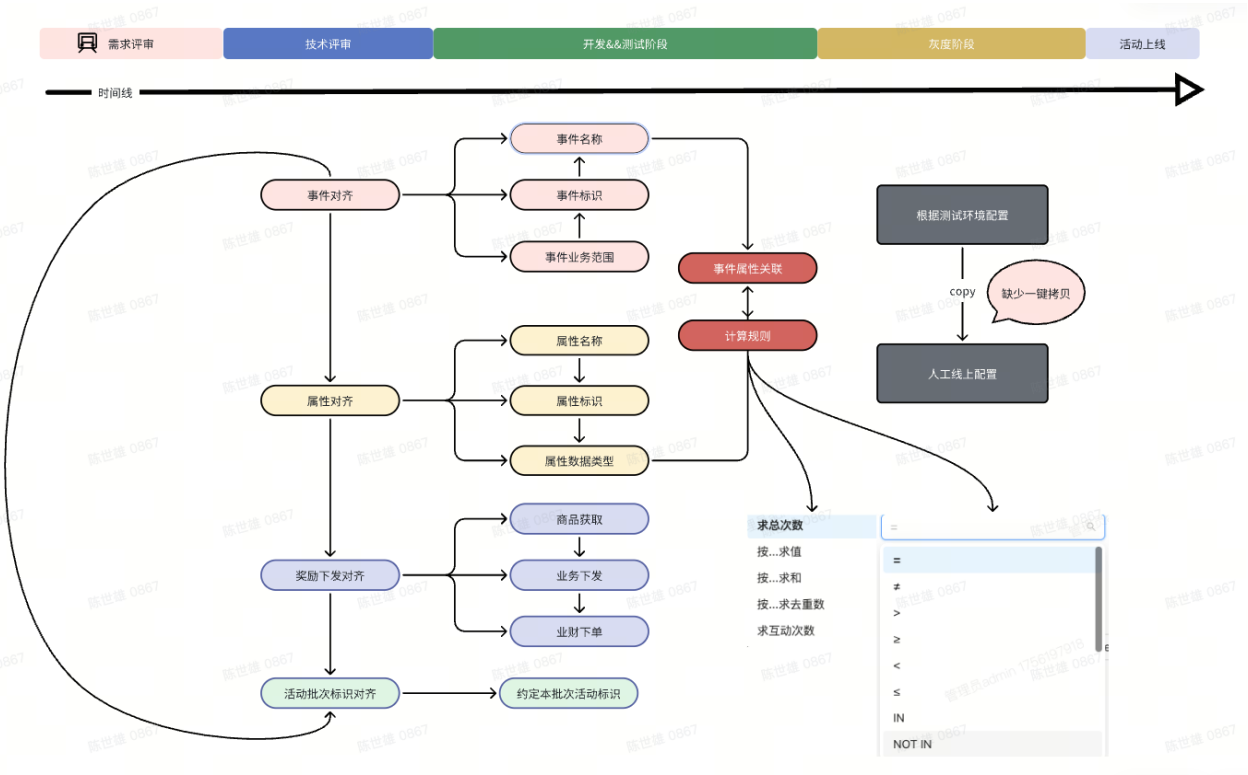
任务列表

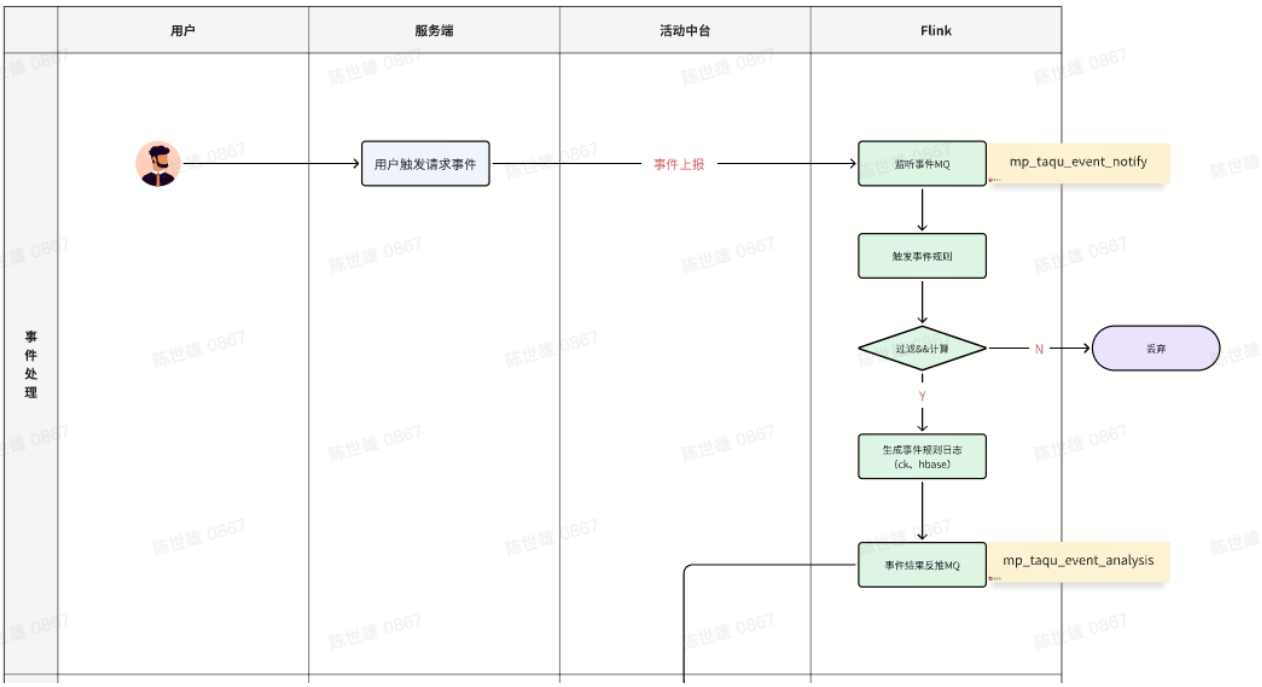
任务编辑配置

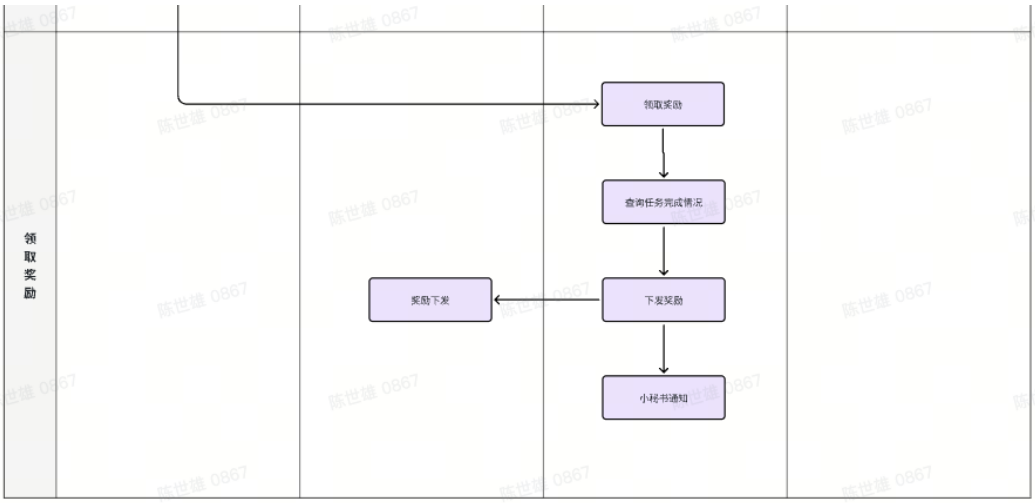
事件管理

名词术语

整体流程图







技术点

1. clickhouse【埋点日志、奖励明细】

每条埋点数据，都需要沉淀到日志表中。数据量大，且字段多，一天大致60kw。

该日志数据，用于后续排查问题数据链路追踪使用。

埋点数据来自消息队列可批量同步到ck中，发挥ck批量数据插入强项。同时OLAP的数据压缩效率高，存储资源成本更优解。

2. Hbase【任务进度】

各任务字段各式各样，HBase支持宽表和稀疏数据，适合存储多样化、动态变化的任务进度信息。
同时Hbase支持高吞吐量的并发读写操作，适合活动系统这种“高并发、低延迟、持续更新”的业务场景。
hologress适合全文搜索 特别适合需要同时处理实时数据和历史数据，并且对查询性能有较高要求的场景

3. Flink

数据量大，消费能力要求最大化，同时支撑公司各业务埋点，支持业务数据扩展。
Flink是专业的分布式流处理引擎，轮子丰富，其专业能力远非一个业务系统内的简单消费程序可比。
举个例子：“状态管理”，原生强大支持，开箱启用。可轻松实现精确一次（Exactly-Once） 语义，处理窗口聚合、去重、关联等复杂逻辑时状态可靠。数据清洗和聚合往往需要状态（如判断是否重复、session 统计），Flink的状态管理是内置的、持久化的、可容错的。如果是业务系统自研消费脚本，通常依赖数据库，难以保证一致性。实现一个可靠的“5分钟窗口聚合”都非常复杂且容易出错。
未来扩展：今天的需求是清洗和聚合。明天的需求可能是：

- 实时计算用户完成任务的趋势图。
- 将清洗后的数据实时写入下游多个系统（如ClickHouse、Elasticsearch、HDFS）。 这些需求在Flink中只需增加一个Job或一个输出链路即可，架构无需改动。而如果写在业务系统里，每增加一个需求，都需要去修改和发布核心业务系统，风险高、周期长。

活动例子

活动后台定义

- **属性定义 标识和字段类型 （理解成扩展属性的控制）**

- **原始事件定义（任务行为） 理解成命名一个事件行为 只是名称方便后面关联**

- **自定义事件、事件加工（埋点的计算规则：进行聚合或过滤计算）**

由原始事件组成 进行加工

配置对应活动的任务 （用到上面配置的事件）