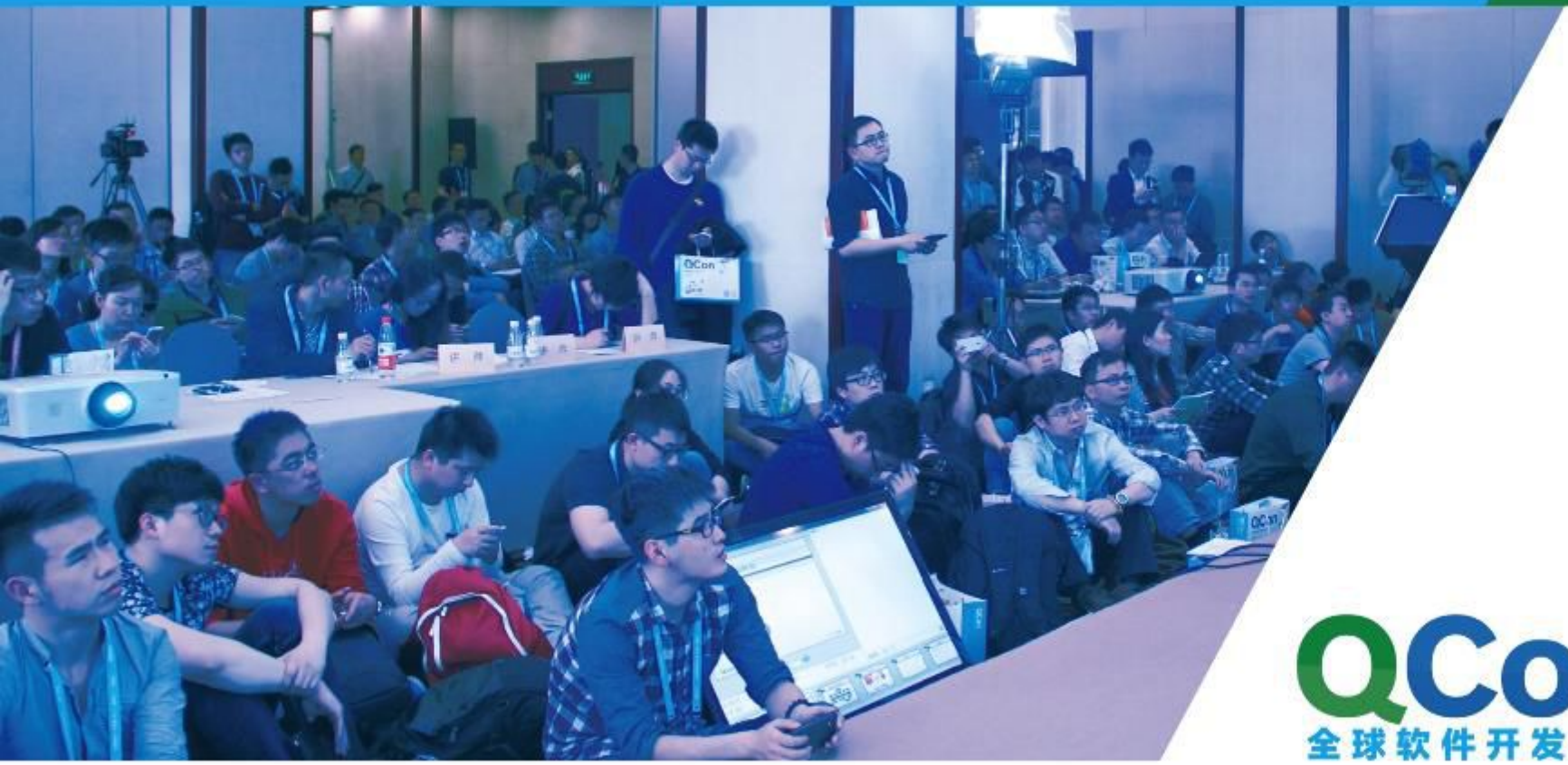
QCon全球软件开发大会

International Software Development Conference



QCon
全球软件开发大会

Geekbang.
极客邦科技

全球领先的技术人学习和交流平台

InfoQ
专注中高端技术
人员的社区媒体

EGO EXTRA GEEKS' ORGANIZATION NETWORKS
高端技术人员
学习型社交网络

StuQ
实践驱动的IT职业
学习和服务平台

扫我，码上开启新世界

Geekbang.
InfoQ | EGO NETWORKS | StuQ

InfoQ

促进软件开发领域知识与创新的传播

ArchSummit
全球架构师峰会

实践第一 案例为主

时间：2015年12月18-19日 / 地点：北京 · 国际会议中心

欢迎您参加ArchSummit北京2015，技术因你而不同

ArchSummit北京二维码

QCon
全球软件开发大会

[北京站]

2016年04月21日-23日

关注InfoQ官方信息
及时获取QCon演讲视频信息

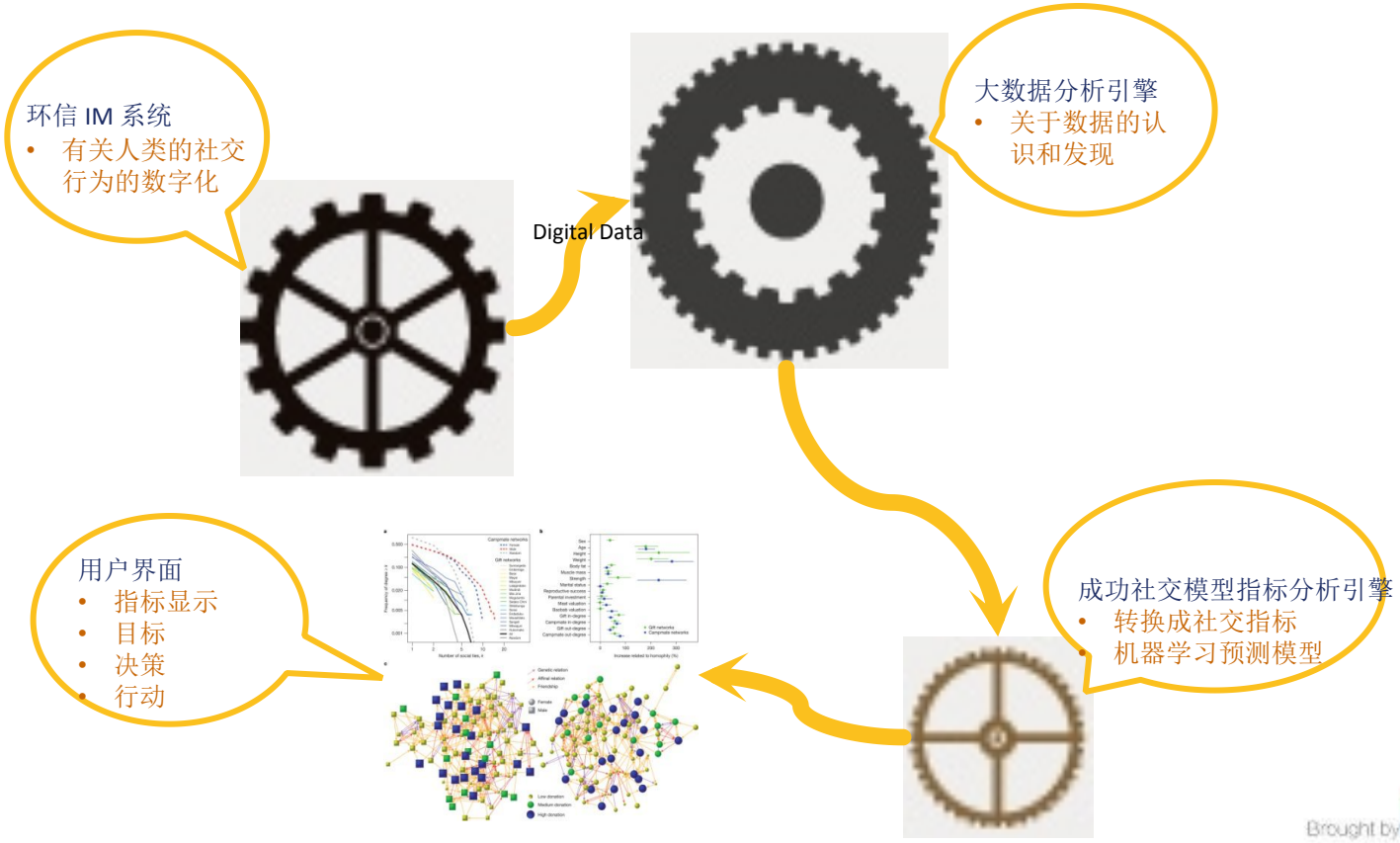
环信社交大数据挑战和实践

环信大数据团队

演讲主题

- 系统功能和逻辑架构介绍
- 社交大数据模型
- 性能和扩展性
- 重要的坑点难点

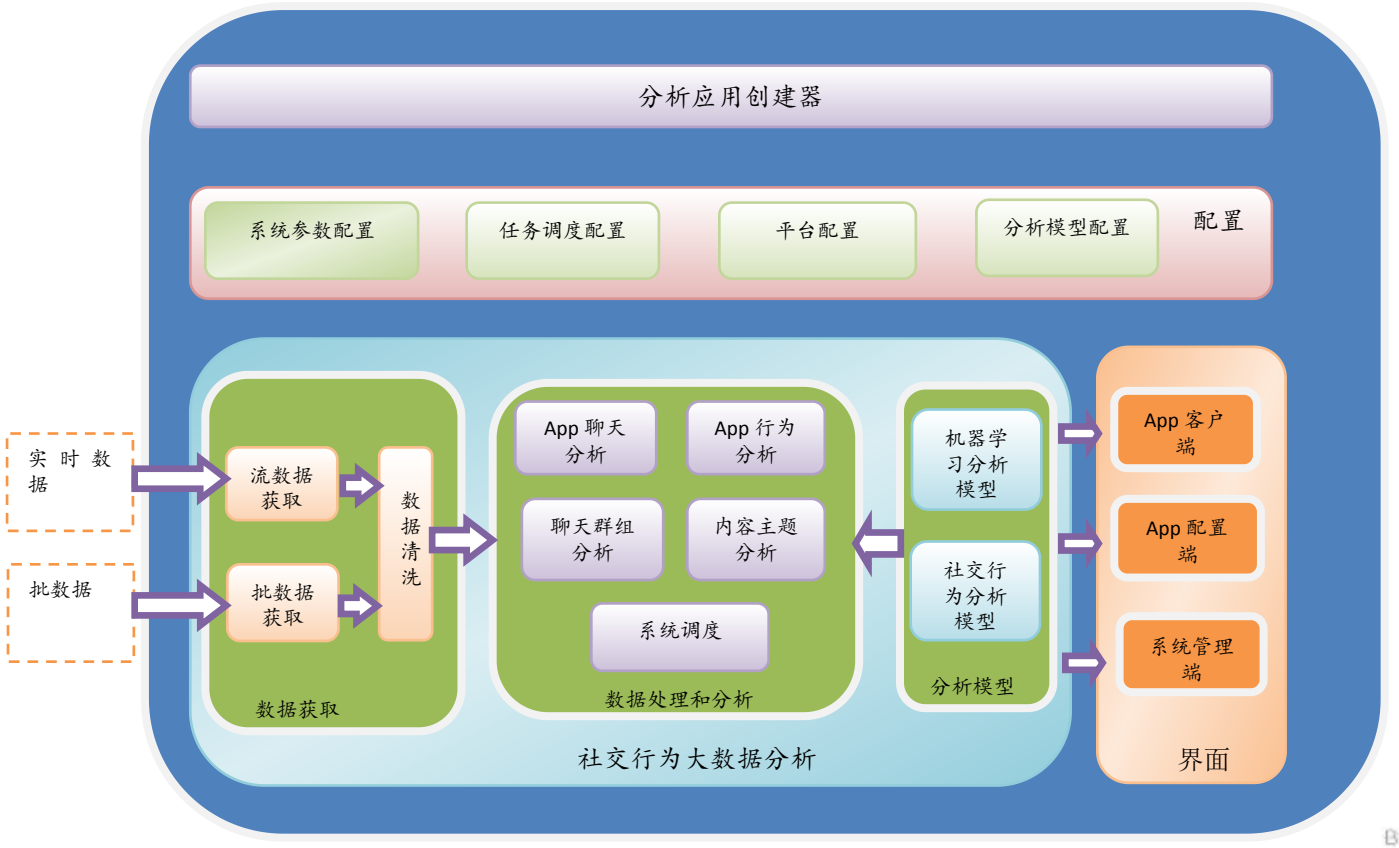
我们正在造什么?


环信 IM 系统
• 有关人类的社交行为的数字化
Digital Data
大数据分析引擎
• 关于数据的认识和发现
成功社交模型指标分析引擎
• 转换成社交指标
• 机器学习预测模型
用户界面
• 指标显示
• 目标
• 决策
• 行动

关键技术挑战和技术要点

- 基于云计算的社交大数据分析工具；
- 支持过亿数据分析，趋势挖掘；
- 完美的水平扩展性能；
- 实时数据分析快速准确；
- 多租户系统；
- 全自动的决策支持系统；
- 提供多种可定制的机器学习算法，方便用户建立模型进行预测预警

系统业务功能逻辑架构

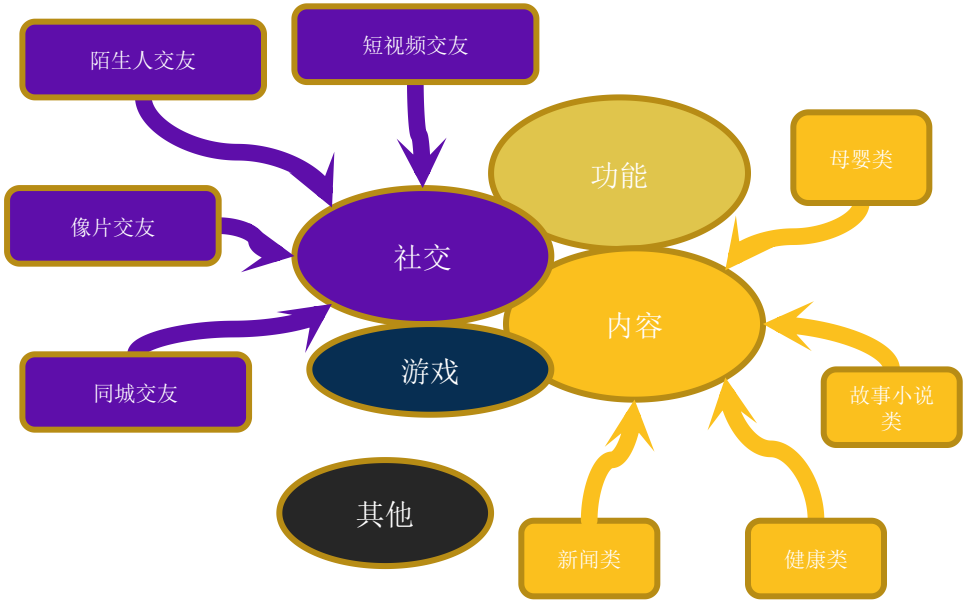

分析应用创建器
系统参数配置
任务调度配置
平台配置
分析模型配置
配置
实时数据
批数据
流数据获取
批数据获取
数据清洗
数据获取
App 聊天分析
App 行为分析
聊天群组分析
内容主题分析
系统调度
数据处理和分析
机器学习分析模型
社交行为分析模型
分析模型
App 客户端
App 配置端
系统管理端
界面
社交行为大数据分析

社交模型 – 功能概括和目标

- 社交模型涉及到300个以上的社交统计分析指标；
- 为智能化深度挖掘分析打下了良好的基础；
- 不同类型的App对应不同的数据分析模型；
- 客户可对自己的App模型进行定制化操作，得到最想要的分析结果

目标：清晰描述用户关系，深度挖掘App问题，划分用户圈子，
辨别用户质量

社交模型 – 社交指标分类简介

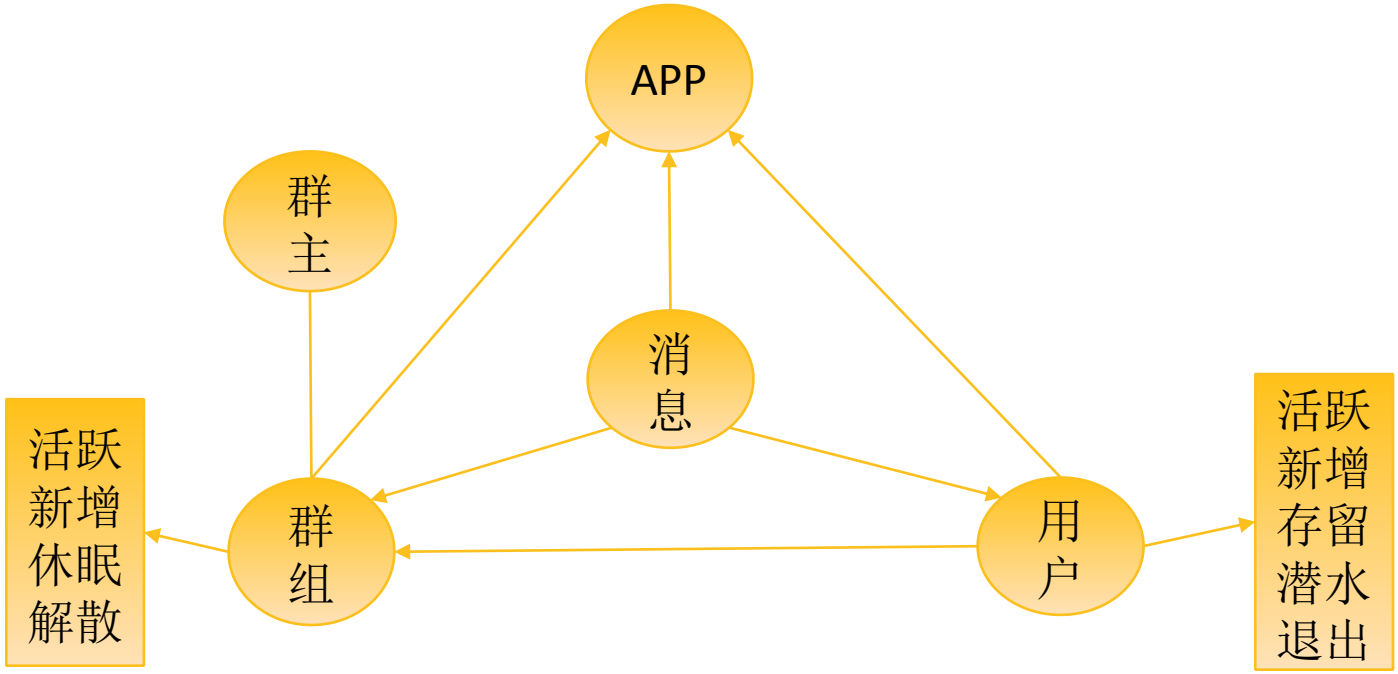


根据一个app类型，它的社交行为和社交环境的发展阶段的不同，我们为他们定义不同的社交指标模型，帮助他们提升社交质量，最终拥有一个成功的产品.

工具型 → 平台型 → 云生态类型

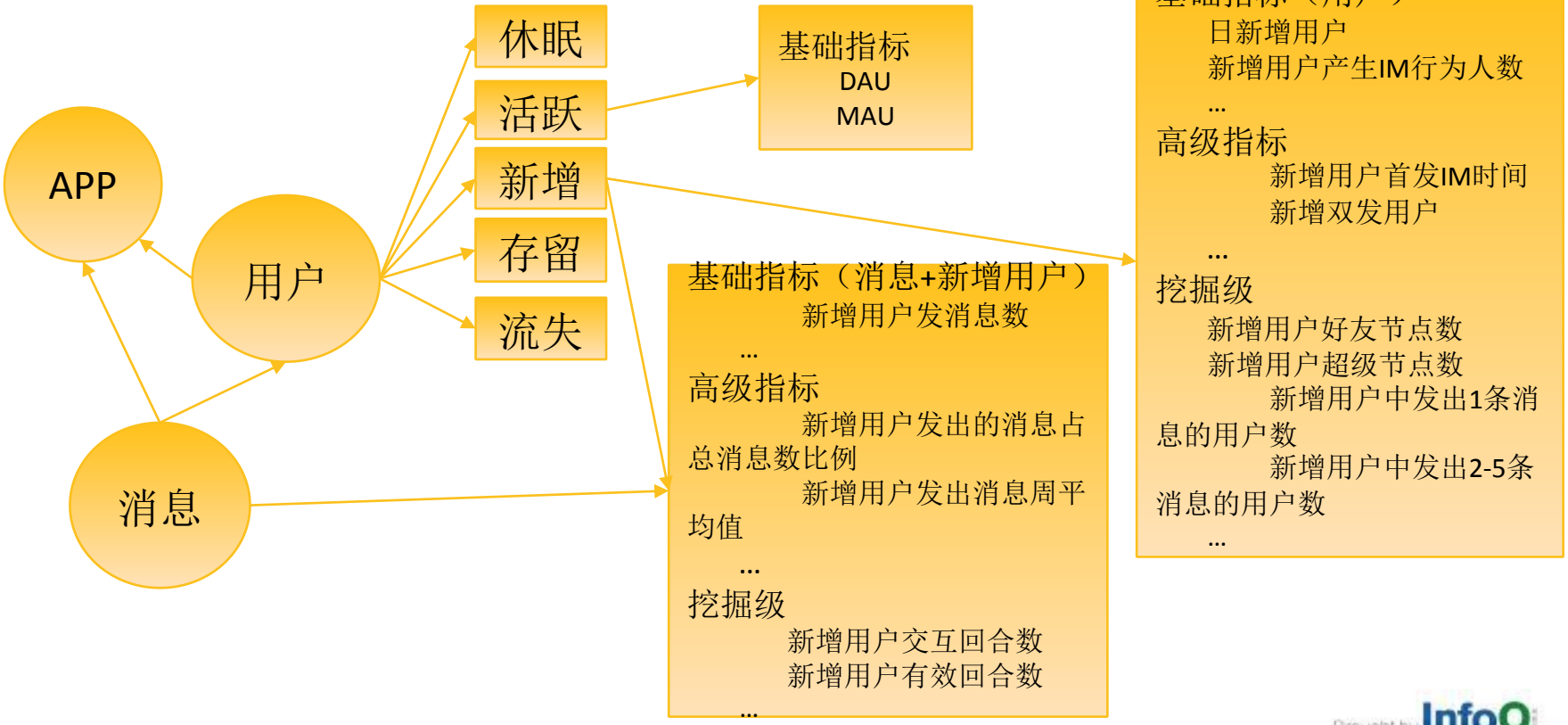
QCon 全球软件开发大会

社交模型 – 社交实体关系及其状态



Brought by InfoQ

社交模型 – 新增流量分析指标构造



社交模型 –新增流量分析模型

基础指标
- DAU
- MAU
- 日新增用户
- 新增用户产生IM行为人数
- 新增用户发消息数
- …

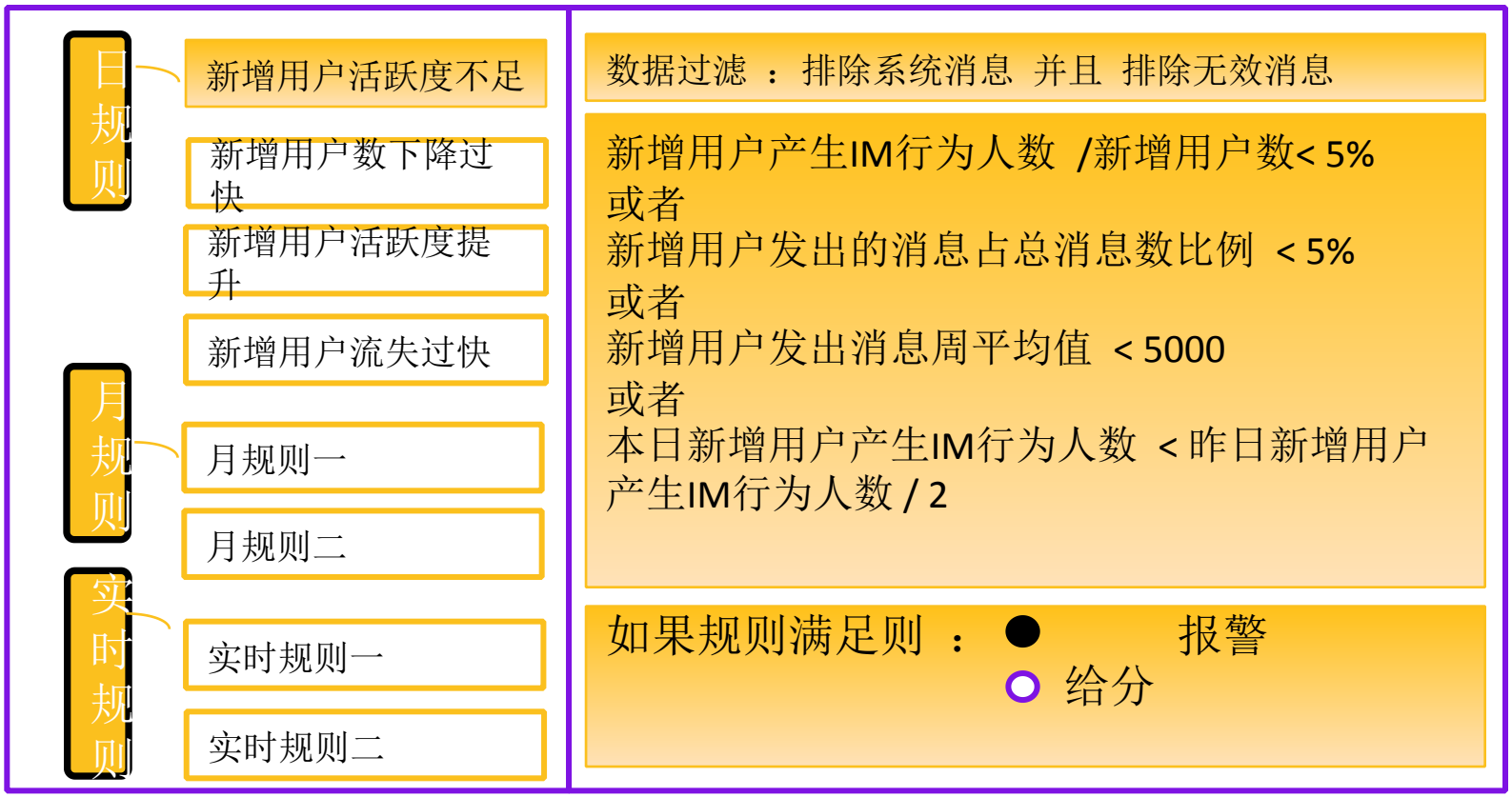
高级指标
- 新增用户首发IM时间
- 新增双发用户
- 新增用户发出的消息占总消息数比例
- 新增用户发出消息周平均值
- …

挖掘级
- 新增用户中发出1条消息的用户数
- 新增用户中发出2-5条消息的用户数
- 新增用户IM流量时间分布图
- 新增用户好友节点数
- 新增用户超级节点数
- 新增用户交互回合数
- 新增用户无效回合数
- 新增用户有效回合数
- …

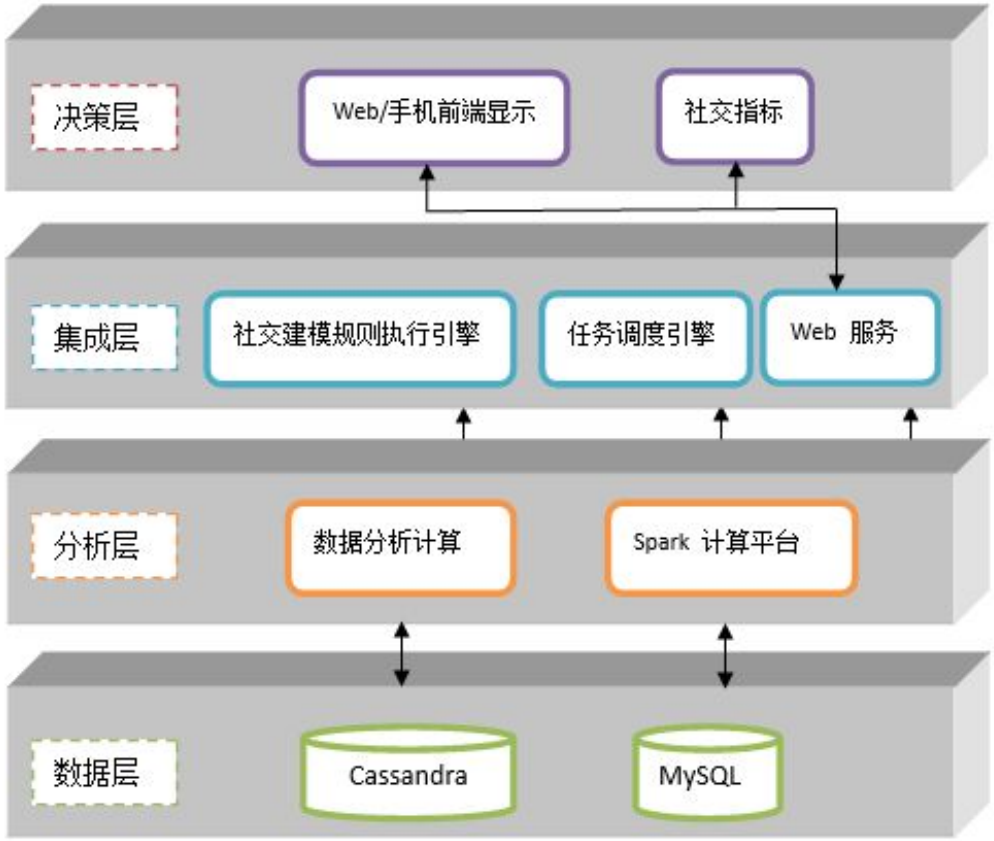
社交模型 – 数据挖掘型指标举例

新增用户交互回合数	当日新增用户发出消息，被回复，为一个回合，求总数（这是一个对数，多少对）。同一用户发送多条后得到回复视为一个回合。
新增用户IM流量时间分布图	以时间为横轴，新增用户产生的流量为纵轴的折线图，精确到KB。
新增用户好友节点	相互有过IM行为的3个用户形成节点，其中包含至少一个新增用户的。
新增用户超级节点	相互有过IM行为的3个用户形成节点，其中包含至少一个新增用户的，其他用户每周登录3次以上，周累计单人IM超过30条。

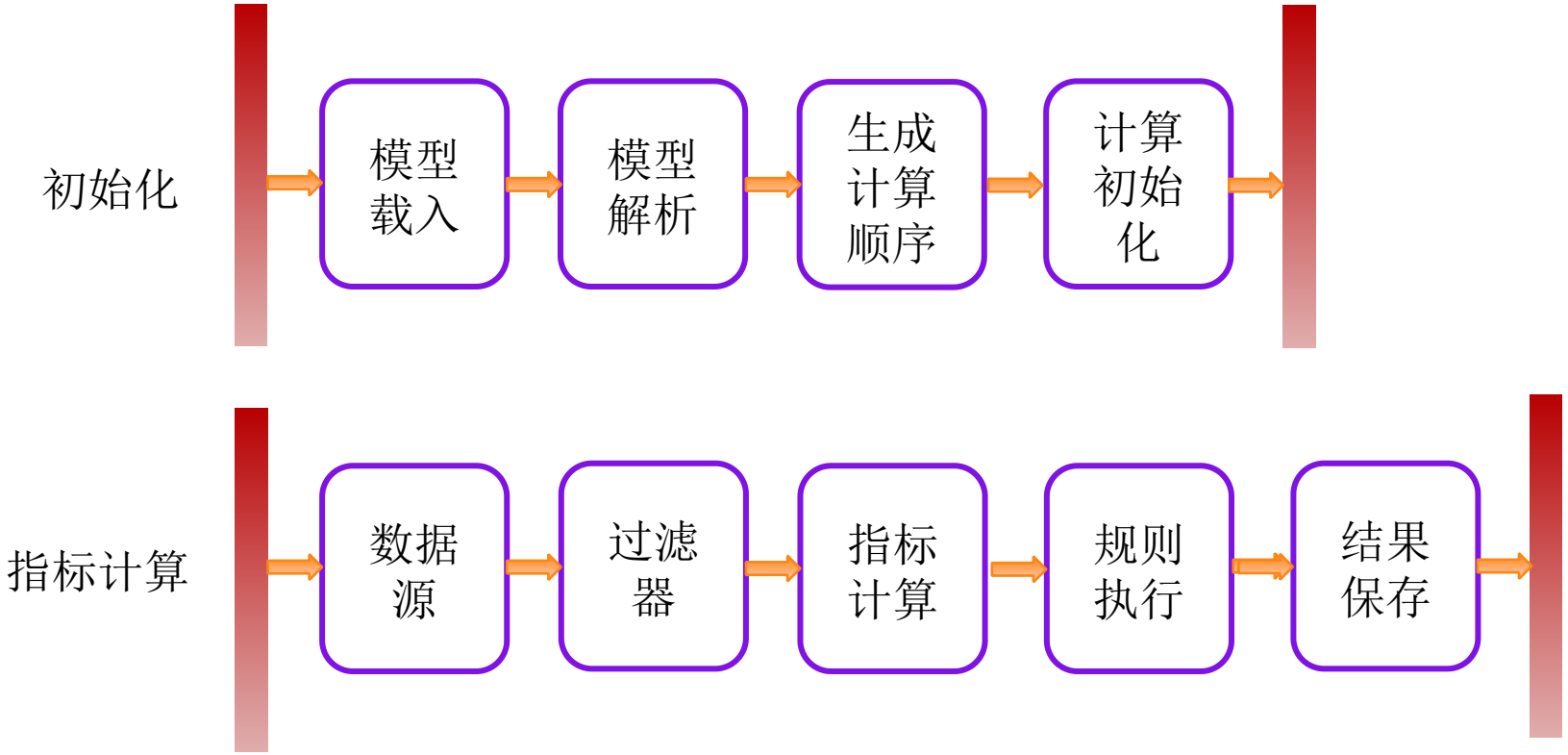
社交模型 – 规则引擎


日规则
新增用户活跃度不足
新增用户数下降过快
新增用户活跃度提升
新增用户流失过快
月规则
月规则一
月规则二
实时规则
实时规则一
实时规则二
数据过滤 ： 排除系统消息 并且 排除无效消息
新增用户产生IM行为人数 /新增用户数< 5%
或者
新增用户发出的消息占总消息数比例 < 5%
或者
新增用户发出消息周平均值 < 5000
或者
本日新增用户产生IM行为人数 < 昨日新增用户产生IM行为人数 / 2
如果规则满足则 ：
报警
给分

社交模型 – 社交指标分析逻辑架构



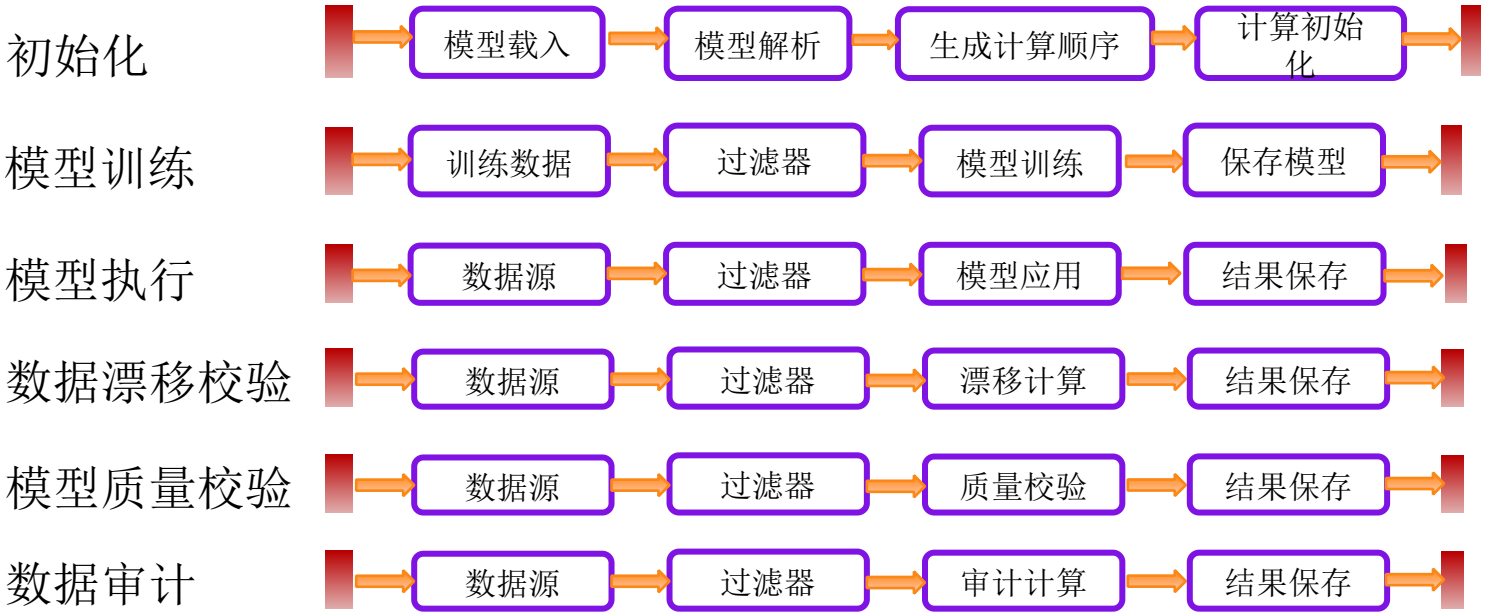
社交模型 – 指标计算组件执行序列



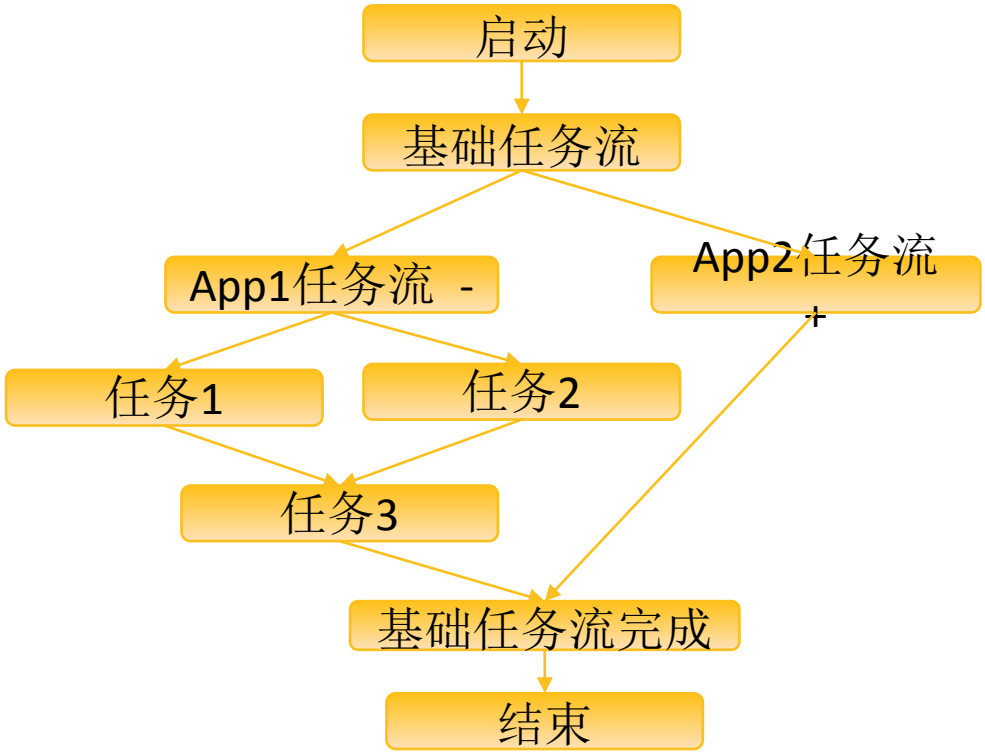
社交模型 – 机器学习模型

- **目标**：智能化社交模型
- **方法**：大数据+计算能力+社交模型+高效算法
- **问题**：
  - **分类问题**. 用户，群组行为分类
    - 算法：有监督/无监督/半监督分类，聚类算法
    - 模型例子：朴素贝叶斯分类模型（比如预测一个新增用户将来活跃与否）
  - **关键绩效指标相关的预估问题**：活跃率，新增率，存留率，流失率，解散率（群组）
    - **算法**：各类回归算法例如logistic regression, Cox regression
    - **模型例子**：用户流失风险模型，群组解散风险模型

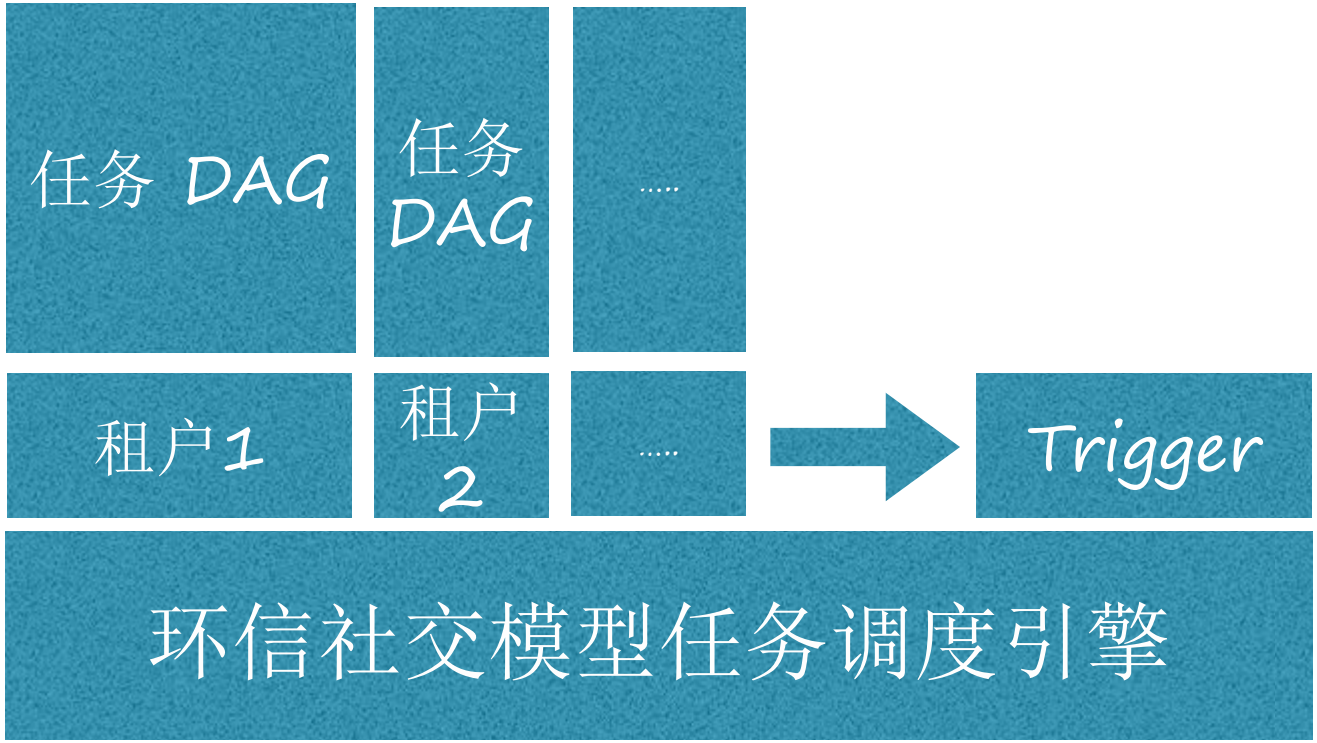
社交模型 – 机器学习组件执行序列



社交模型 - 任务调度



社交模型 - 任务调度逻辑架构



社交模型 - 应用构造

- 应用构造指的是针对不同类型app的分析需求进行抽象后的一个模板类型的具体实现。这个实现体现在根据该类型app的需求所定义的配置模板。该模板主要包括三大块：
  - 系统配置
  - 模型配置
  - 界面配置

- 应用构造是形成不同类型app行业解决方案的基础

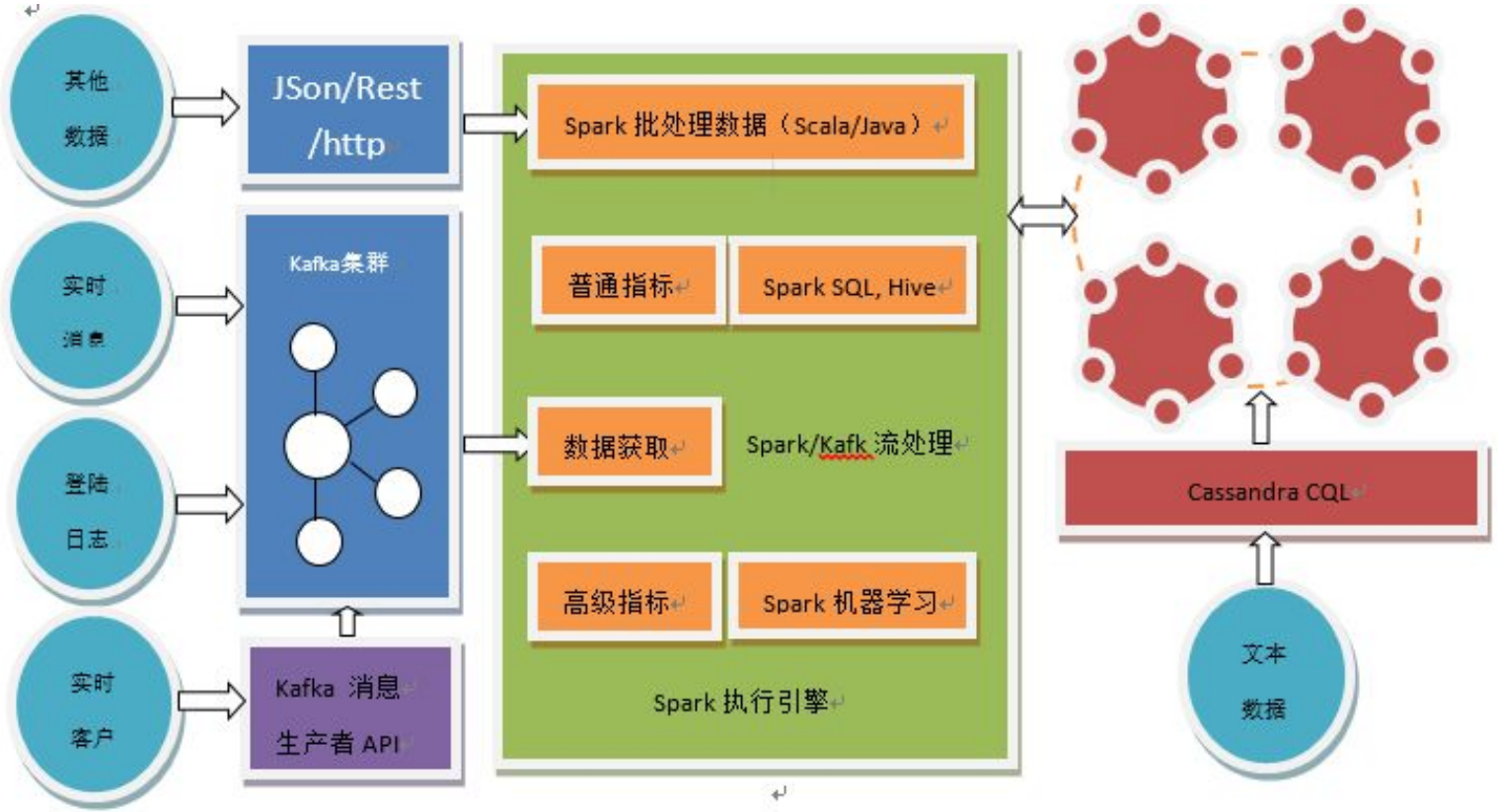
性能与扩展性 – 问题描述

- 截止2015年上半年数据规模
  - SDK覆盖用户数高达2.51亿
  - 日均发送消息量

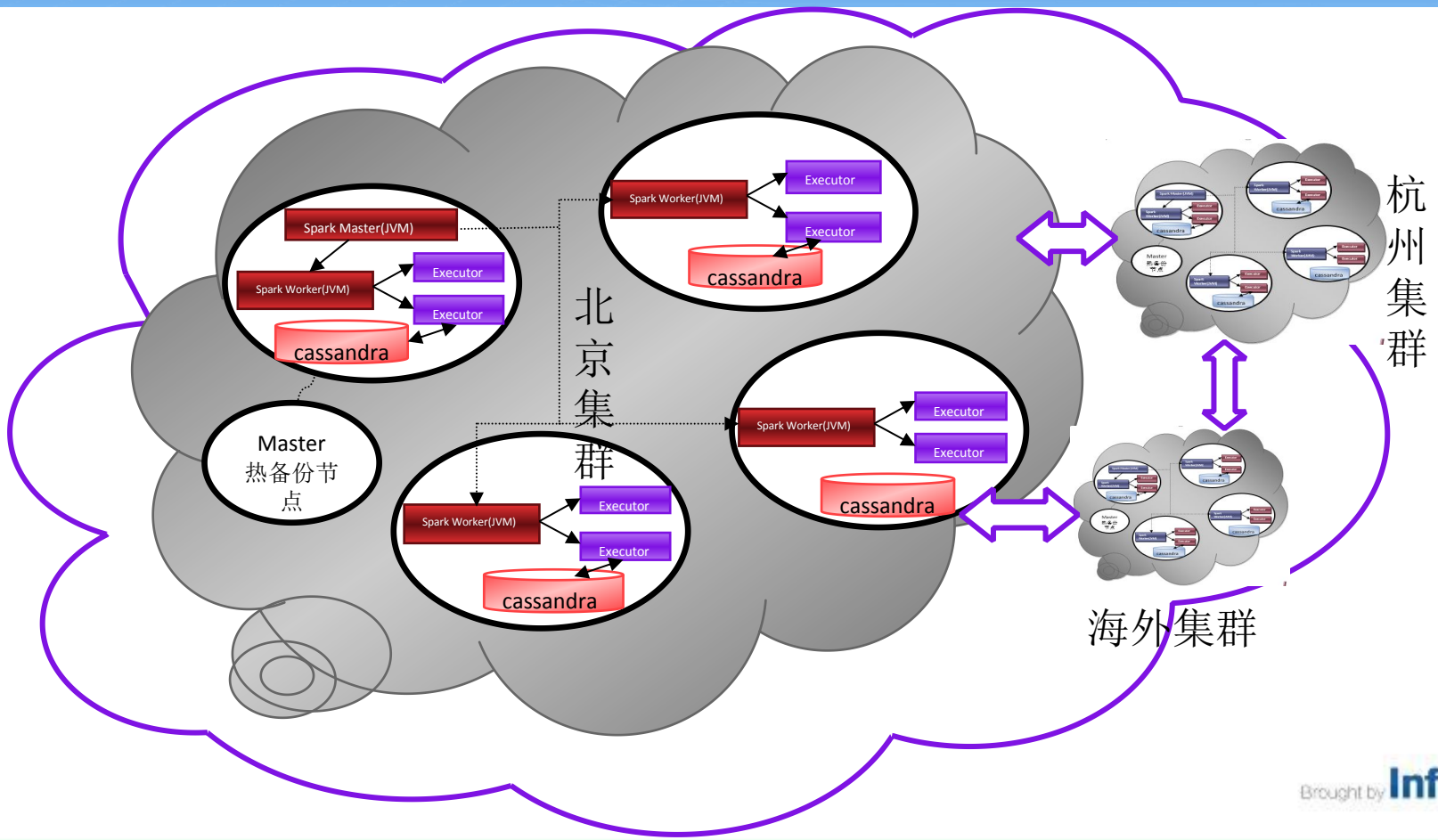
    每天1亿+
  - 环信共服务注册App

    23062
- 要求
  - 实时处理 + 批处理
  - 完全的水平扩展性

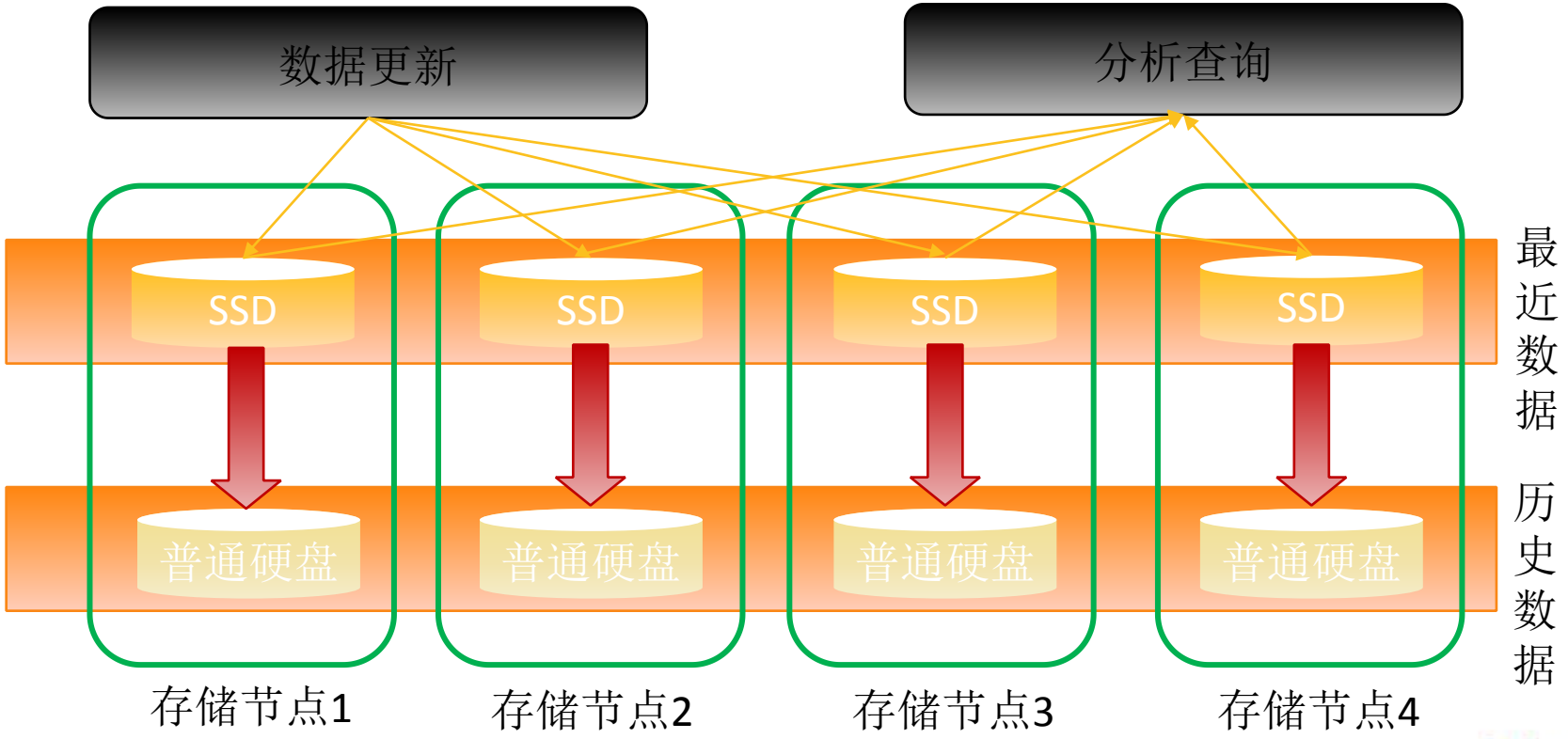
性能与扩展性 - 系统环境


其他数据
实时消息
登陆日志
实时客户
JSon/Rest /http
Kafka集群
Kafka 消息生产者 API
Spark 批处理数据（Scala/Java）
普通指标
Spark SQL, Hive
数据获取
Spark/Kafk 流处理
高级指标
Spark 机器学习
Spark 执行引擎
Cassandra CQL
文本数据

性能与扩展性 – 水平扩展


Spark Master(JVM)
Spark Worker(JVM)
Executor
Executor
cassandra
Spark Worker(JVM)
Executor
Executor
cassandra
Master
热备份节点
北京集群
Spark Worker(JVM)
Executor
Executor
cassandra
Spark Worker(JVM)
Executor
Executor
cassandra
杭州集群
海外集群

性能与扩展性 – 存取能力扩展



坑点难点分析 - 数据库

- 坑点1 - 主键定义
  - Rowkey + columnkeys
  - 问题一：数据丢失
  - 分析：Rowkey和columnkey相同的数据会被覆盖，新数据将替换旧数据。
  - 解决方案：使用时间和随机数作为主键之一
  - 问题二：数据集中存储在一处
  - 分析：Rowkey设计不合理，数据无法partition
  - 解决方案：重新设计rowkey，使得数据可以打散分布在不同的节点。
- 坑点2 -大量删除数据系统压力大的时候容易报连接错
  - 问题：cassandra不直接删除数据，只是标志为要被删除。真正开始删除的时候往往因为要删除的数据量太大引发系统问题
  - 解决方案：大数据量删除尽量避免采用sql删除功能

坑点难点分析 – Spark 数据处理

- 坑点一：Spark sql 复杂查询速度极慢
  - 问题分析：即便是主键，非相等比较的范围查询不走索引
  - 解决方案：按索引建立的顺序，使用rowkey和columnkey做简单的相等比较查询查出数据，再利用scala强大的数据处理能力完成复杂sql的功能
- 坑点二：scala/kafka streaming实时插入数据极慢
  - 问题分析：二级索引导致大量数据更新操作执行慢
  - 解决方案：取消二级索引
- 坑点三：scala/kafka streaming 内存outofmemory
  - 问题分析：一次性读取大量kafka数据，写入数据量太大
  - 解决方案：限制一次性读取写入数据量

THANKS