

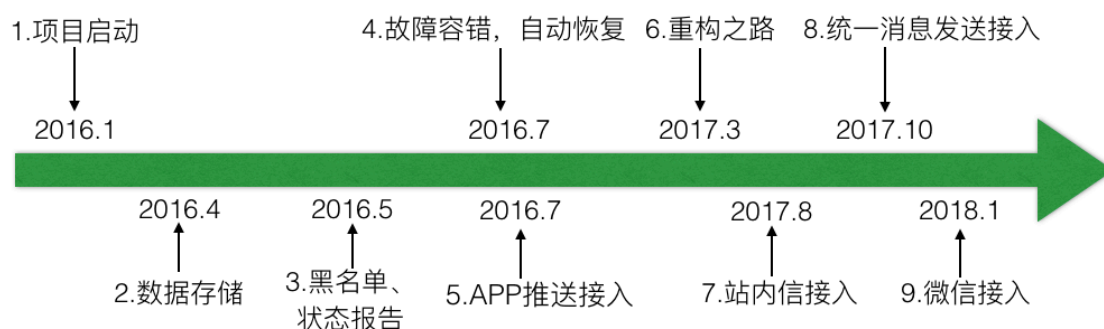
写在前头：

消息系统从0到1，经历了若干阶段，到现在相对成熟，特总结整理如下，希望能给大家一些启发。

“消息”这里指的是“短信、APP推送、站内信、微信”。

消息系统演进之路

项目里程碑（分九个阶段）：



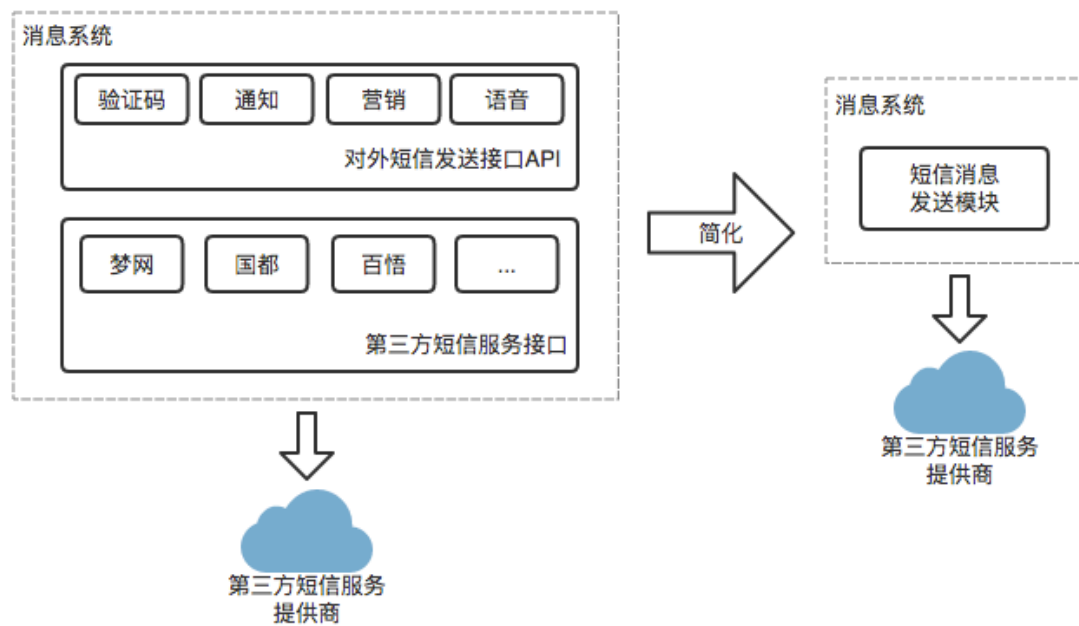
消息系统演进之一 - 初始化（统一第三方短信发送）

背景：

- 公司对接了一些第三方短信通道提供商，而这几个短信通道商的短信发送逻辑和接口方法都是独立开在不同的PHP文件，业务方使用混乱，没有统一规范，扩展性差。

方案：

- 恰逢当时正推行服务化项目，由周导牵头，决定将针对第三方短信发送的逻辑提取出来形成单独的系统，实现统一的短信发送接口API，封装第三方短信通道的发送细节，便于以后的扩展，此系统命名为“消息系统”。



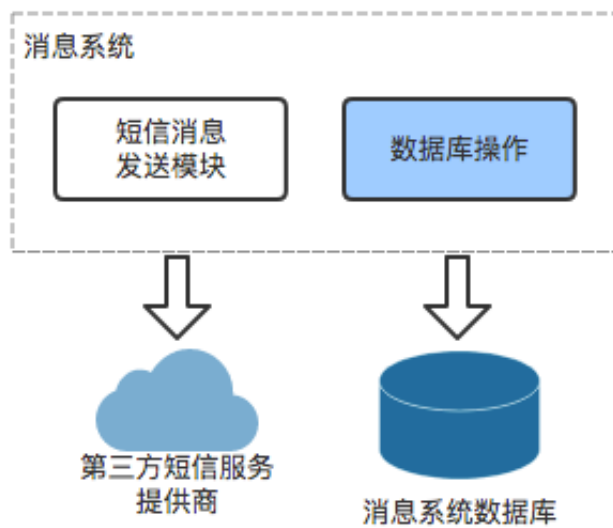
消息系统演进之二 - 数据存储（新增数据库）

背景：

- 每隔一段时间运营、财务都需要和第三方服务提供商核对短信发送量，结算费用。原始的结算方式是运维（叶晨阳）通过统计发送记录日志来确定发送量和第三方进行结算，如果日志记录丢失，则只能以第三方提供数据为准。
- 所有问题都只能通过日志查询，而日志系统不稳定，有时还会丢数据，给分析问题造成了极大的困扰。

方案：

- 新增消息系统数据库，保存消息发送记录，方便数据统计和问题分析。



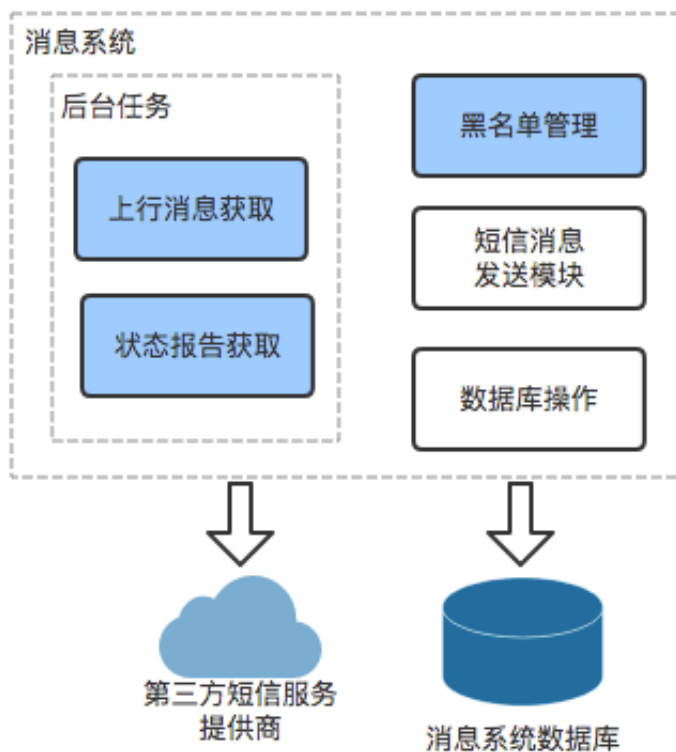
消息系统演进之三 - 黑名单和状态报告

背景：

- 有用户投诉“收到贝贝短信骚扰，回复TD退订短信无效”，之后依然会收到其它营销短信。
- 用户反馈收不到短信，排除方式只有查日志，查到具体发送的第三方服务提供方，然后再通过QQ的方式询问结果，耗时耗力。
- 多个第三方服务提供商，如何评估服务质量？

方案：

- 读取用户上行消息，针对回复TD的用户和通道，加入相应的黑名单，在短信发送之前进行黑名单检查。
- 读取状态报告，保存到用户发送记录及接收记录，后台方便查询用户发送和接收状态。有了状态报告就可能很明确的统计出各第三方渠道商的消息发送抵达时间和抵达率，有利于我们评估第三方服务提供商的服务质量。



消息系统演进之四：流量控制、故障容错和自动恢复

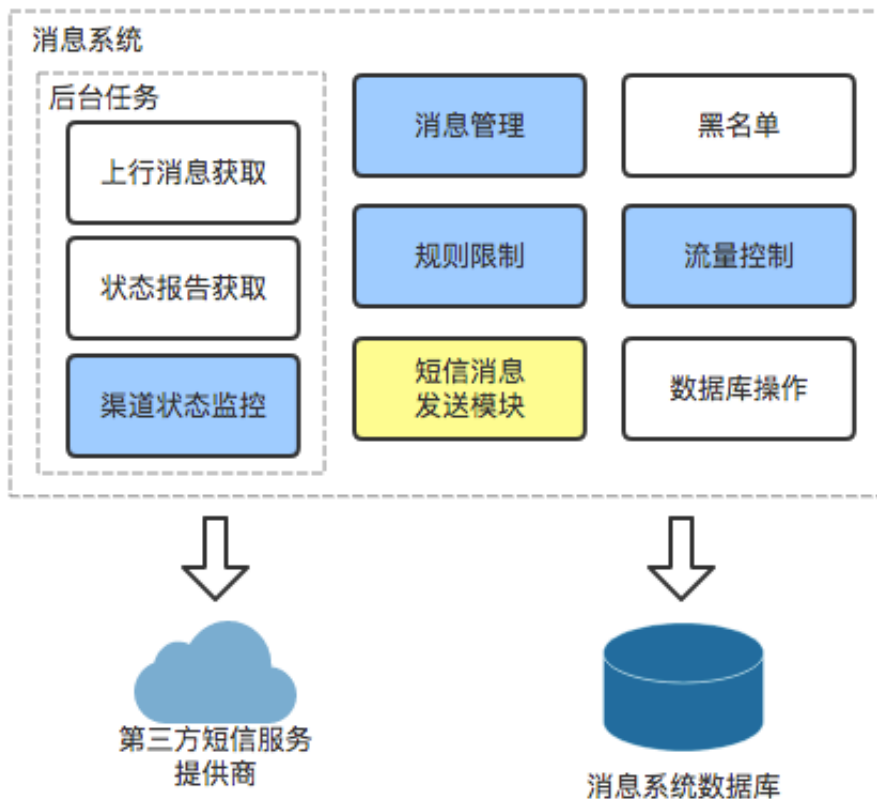
背景：

- 生产事故，国都（短信服务提供商）通道问题导致短信发送失败，直接导致半小时内注册数直接跌零。
- 某同事做短信测试时，将测试短信代码提交到线上，还好由于短信内容里带"测试"字样大部分被运营商拦截，未形成大的生产事故。
- 基于成本和运营灵活度各方面因素考虑，都希望能灵活控制各通道服务提供商的短信发送比例。

方案：

- 针对第三方服务的不稳定性，我们采用了两个方案。一是失败重试机制，发送失败后切换到另一家服务提供商重发。二是渠道状态监控，当渠道1分钟内失败次数达到10次以上，则将渠道状态设置为不可用，切换到其他渠道，每3分钟检查一次渠道状态，进行自动恢复。
- 对于消息发送内容的不确定性，也让我们倍感惶恐，因此我们做了消息管理，要求所有的消息都需要先进行配置，配置审核通过才能进行发送。
- 规则限制：为防止业务方使用失误，重复提交，因此我们也做了防重功能，当然

也还有用户的每日发送次数限制，营销短信发送时间限制等，防止过度打扰用户。



消息系统演进之五：APP推送接入

背景：

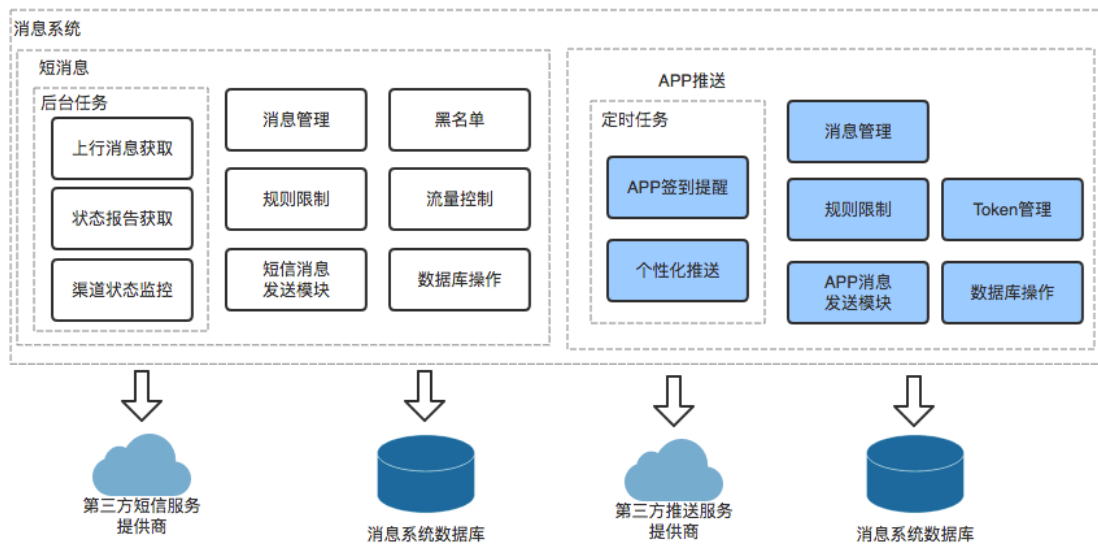
- APP推送消息接入也经历了前面的数据库，规则限制，消息管理的几个阶段，当然APP推送消息也有其独特的地方。
- 消息管理和规则限制：生产事故，生活帮代码逻辑问题导致对同一用户调用了N次推送接口，造成部分用户同一时间收到数十条相同的APP推送消息。
- APP推送底层接口是靠第三方服务的token来推送的，而token是由客户端上报的，所以想要使用我们的APP推送服务，必传参数是第三方服务的token，这使得业务方使用非常困惑，不能直接通过uid进行APP推送吗？
- APP推送虽然抵达率不高，但是是一种低成本带来高收益的手段，如何利用好我们APP推送促进DAU和GMV？

方案：

- Token管理：用户打开APP时获得第三方服务商提供的token和设备device的关

系，当用户注册登录时获取到device与uid的关系（统一的API Session管理），最终将APP、uid和token关系保存起来。

- 个性化推送：与大数据合作，千人千面，精准营销，带来了不错的GMV提升。
- APP签到提醒：会员俱乐部承载了APP百分之八十的日活，而APP签到是会员日活的基础保障，APP签到提醒对全站DAU提升也有不错的表现。



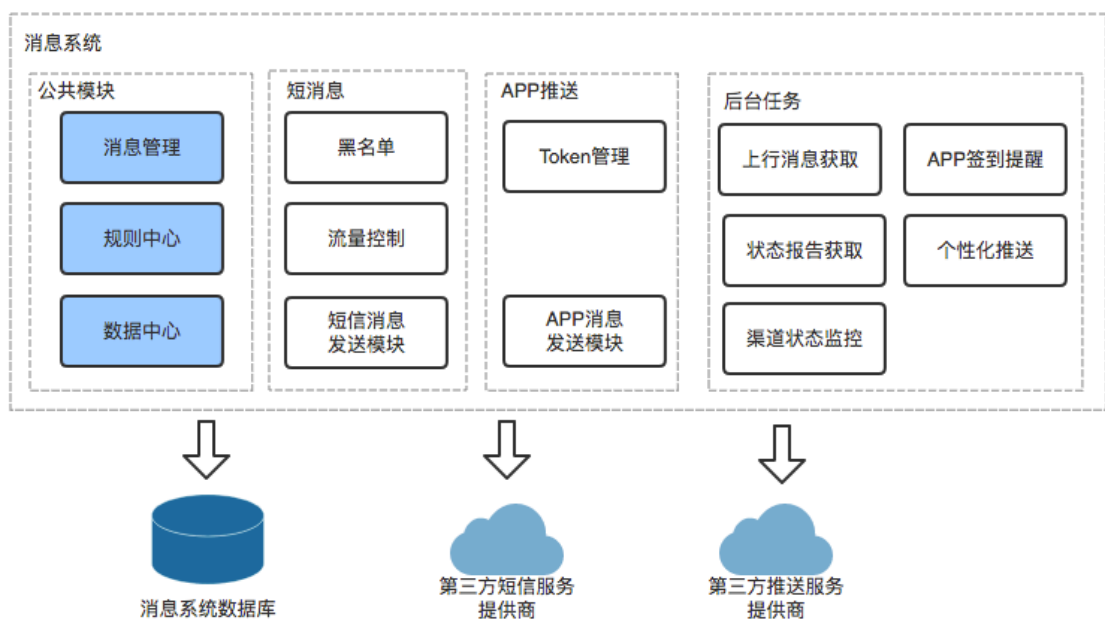
消息系统演进之六：重构之路

背景：

- 重构往往是系统演进中的必经之路，经历了很多功能的增加和迭代，老的代码结构已经不利于我们功能的扩展，维护成本变高，最显著的就是重复代码，相似的逻辑没有很好的整合。

方案：

- 提取公共模块（消息管理，规则中心，数据中心），对于短信息和APP的独特功能也做到模块化，高内聚，低耦合。



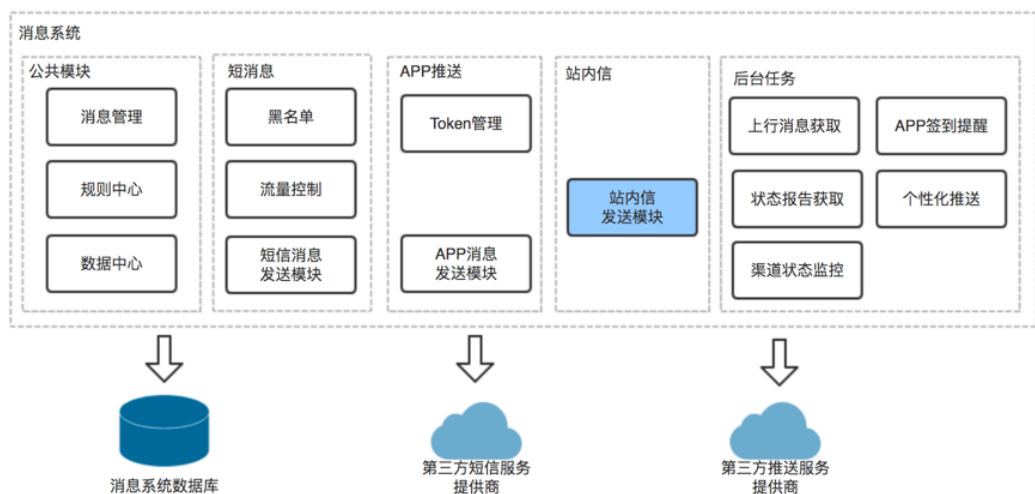
消息系统演进之七：站内信改版接入

背景：

- 消息中心（站内信）没有根据公司业务发展做相应的服务升级。诸多新业务如买啥、问大家、闲置，优惠促销等用户唤醒功能，消息中心并不能做很好的承接。

方案：

- 站内信作为消息的一种，顺理成章的纳入到系统系统中，统一消息管理和规则中心和数据中心都能保证站内信快速接入。



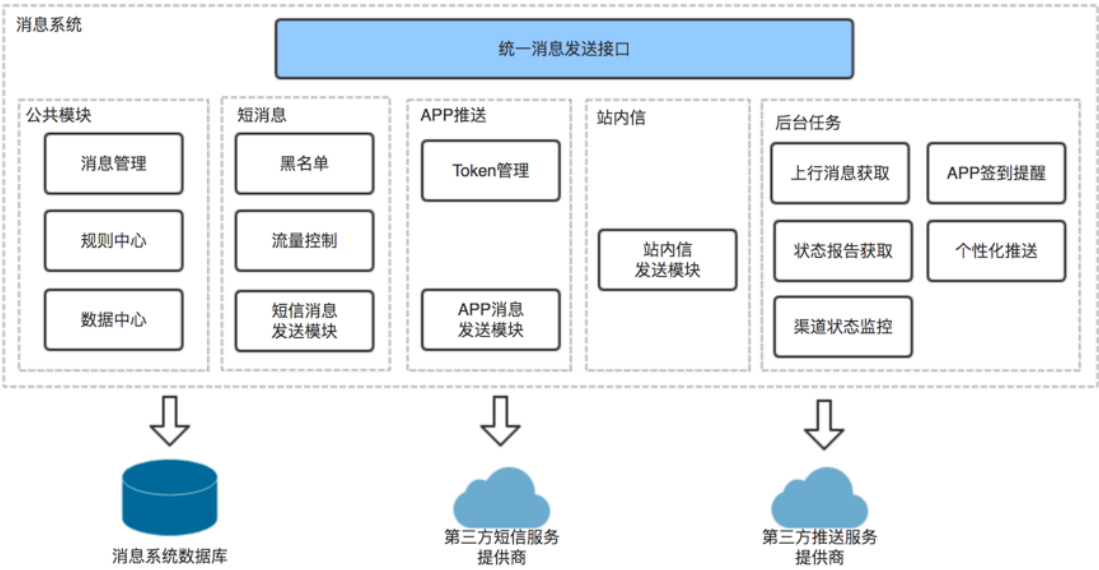
消息系统演进之八：统一消息发送接口

背景：

- 站内信、APP推送、短信发送接口太多，业务方使用成本比较高，很多人私下询问具体的接口调用方式。

方案：

- 统一的消息发送，减少了使用方的接入成本。



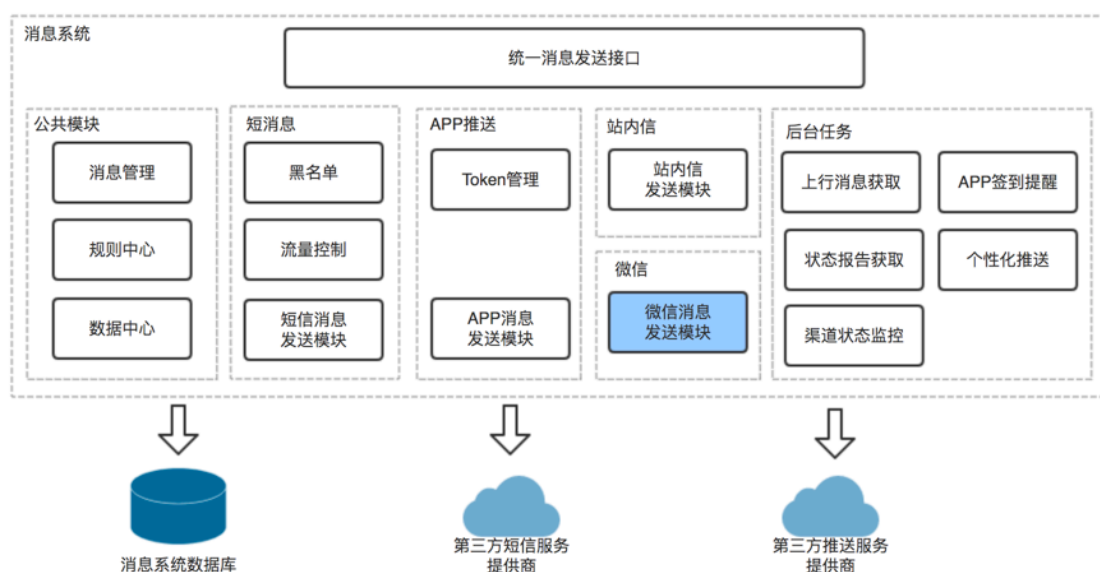
消息系统演进之九：微信消息接入

背景：

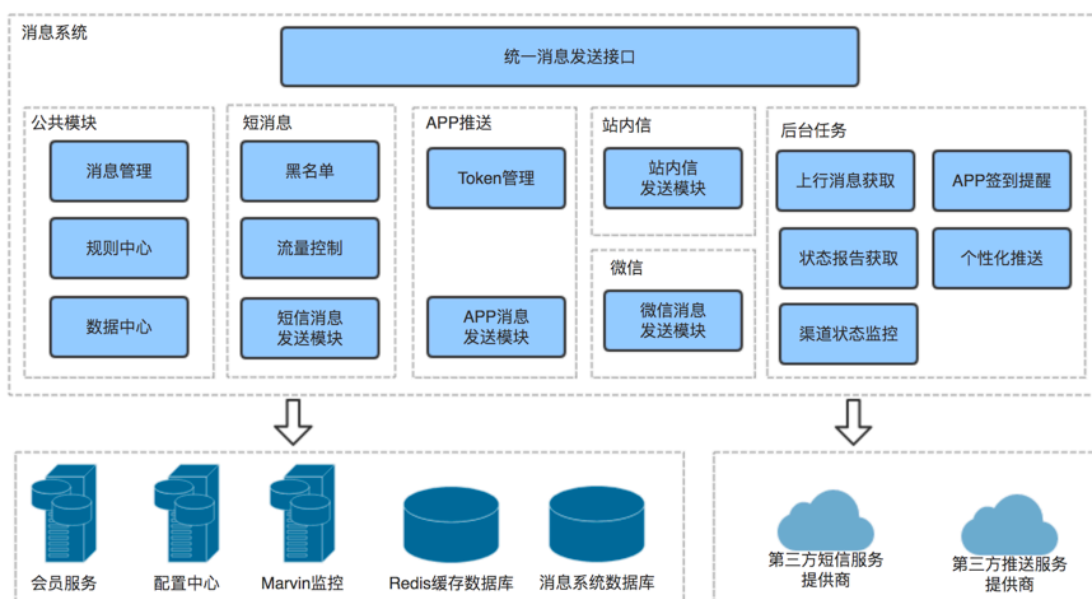
- 社群电商到微信电商，微信消息重要性显著增加，因此将微信消息纳入统一消息发送接口就变得非常有意义。

方案：

- 消息管理里增加微信消息配置，消息系统中增加相应的实现即可。



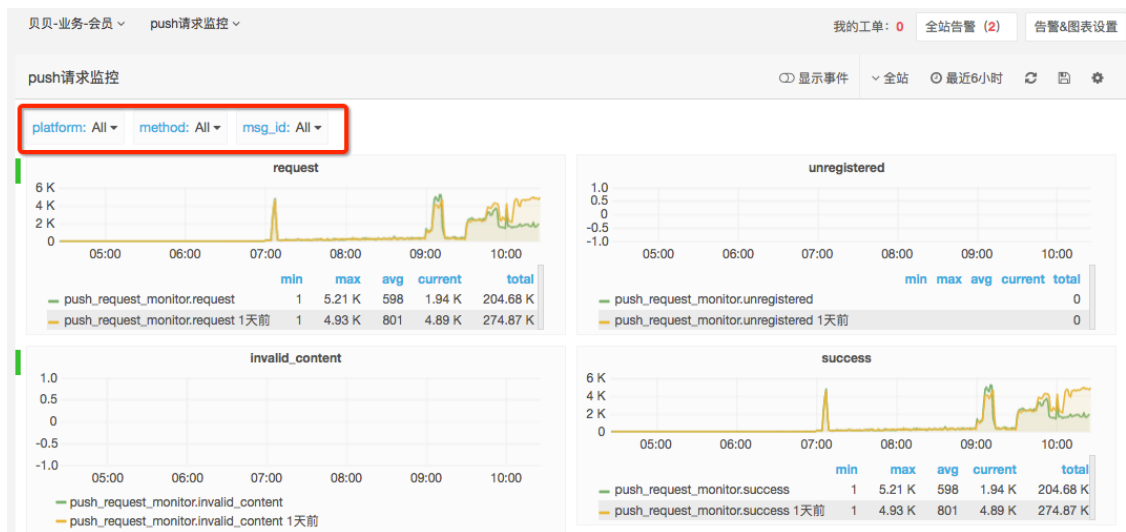
目前完整的消息系统（加入第三方依赖后）



完善的监控体系

消息系统演进过程中，监控体系也是在不断的完善，以APP推送监控为例：

基于Marvin数据打点监控，按平台，发送方式（通知，营销），消息ID维度打点监控，所有通过消息系统发送的消息都将被监控下来，未注册（审核）的消息及消息内容非法的都将被拦截并告警通知我们。



消息系统接入指南

为方便业务方的快速接入，提供消息系统接入指南如下：

<http://doc.husor.com/pages/viewpage.action?pageId=87864814>

END