## 技术架构调研与分析

## 一、需求分析

考虑到当前平台服务的支持要求，集成平台需要满足单点登录、服务路由，权限控制等基本能力；支持当前能力的服务架构分为两类：ESB总线型服务与微服务框架。下面是对这两类服务架构的优缺点分析：

## 二、技术架构调研

### 2.1微服务架构

优点：

1、可维护性强，架构调整与技术更新代价低，适应敏捷开发；

2、系统对接协议轻量级，效率更高；

3、降低系统复杂度。

4、局部修改很容易部署，有利于持续集成和持续交付

5、故障隔离，一个服务出现问题不会影响整个应用

缺点：

1、采用分布式部署，存在固有的复杂度；

2、在多个业务处理不同数据时，数据一致性不容易处理；

典型框架：spring cloud/dubbox

典型架构比较：

spring cloud由于产商与社区的支持，未来会有越来越多的工具支持，同时也能与spring框架本身做很好的融合，是微服务未来的方向。如果产期考虑可以选在spring cloud ；

dubbox的文档较完整与稳定，国内受众较多，在短期内，我们可以选择 dubbox；

### 2.2 ESB总线型架构

优点：

1、为人所熟知：现有的大部分工具、应用服务器、框架和脚本都是这种应用程序；

2、IDE友好：像NetBeans、Eclipse、IntelliJ这些开发环境都是针对开发、部署、调试这样的单个应用而设计的；

3、便于共享：单个归档文件包含所有功能，便于在团队之间以及不同的部署阶段之间共享；

4、易于测试：单体应用一旦部署，所有的服务或特性就都可以使用了，这简化了测试过程，因为没有额外的依赖，每项测试都可以在部署完成后立刻开始；

5、容易部署：只需将单个归档文件复制到单个目录下。

缺点：

1、可维护性差，项目复杂度难以控制，产品透明度低；

2、不能针对某特定业务做集群或负载均衡处理；

3、妨碍持续交付

典型框架：ServiceMix ESB 、Apache Camel、Mule ESB、Spring Integration

典型框架分析：

当前总线型的框架中，服务路由的能力都是具备的，对于ServiceMix ESB也是集成Apache Camel达到具备服务路由的能力，当前框架会更重一点，Mule ESB集成了一些连接器，比如：SAP, Tibco Rendevous, Oracle Siebel CRM等，对于这些连接有特殊需求的，可以考虑引入；对于Apache Camel，本身比较轻量，对动态路由的支持相对较好，同时他可以集成到各种常用框架中，比如spring；针对Spring Integration，它的主要特点在于消息分发的能力；

## 三、技术分析与选型

1、在考虑微服务架构时，我们考虑后续的可扩展性与支持情况，建议使用spring cloud；

2、当考虑使用ESB总线架构时，apache Camel对于我们产品已经足够，同时框架比较轻量，后续组件服务可以自由组合。

3、在当前阶段，我们在做集成平台时，最紧急的是单点登录与授权，后续的版本才考虑服务治理的功能；使用spring cloud存在业务分散后，数据库不能共享的问题，对于系统后续数据统计与分享影响较大，同时，spring cloud在对接第三方异构系统时，只能通过增加适配子系统的方式进行接入，开发成本相对较高，另外一点新技术的学习成本比较高，综合考虑来讲，在性能压力不大的情况下，我们使用apache camel比较合适，后续开发调整的自由度较高，**建议采用spring mvc+apache camel满足第一版本需求；**