



# 多云商环境 基于SpringCloud的微服务实践

Spring Cloud 社区深圳站分享

邓志辉

# 目录

—CONTENTS—

01

产品介绍

02

架构设计

03

云商们的使用

04

实践经验

05

总结

# 产品介绍



## 法力平台

以“打造最好的律师生态圈”为宗旨，  
为帮助律师实现业务运营移动化产品。  
基于社交、任务推动方式办理案件业务。  
平台提供个人注册，以及大客户独立部署提供服务。



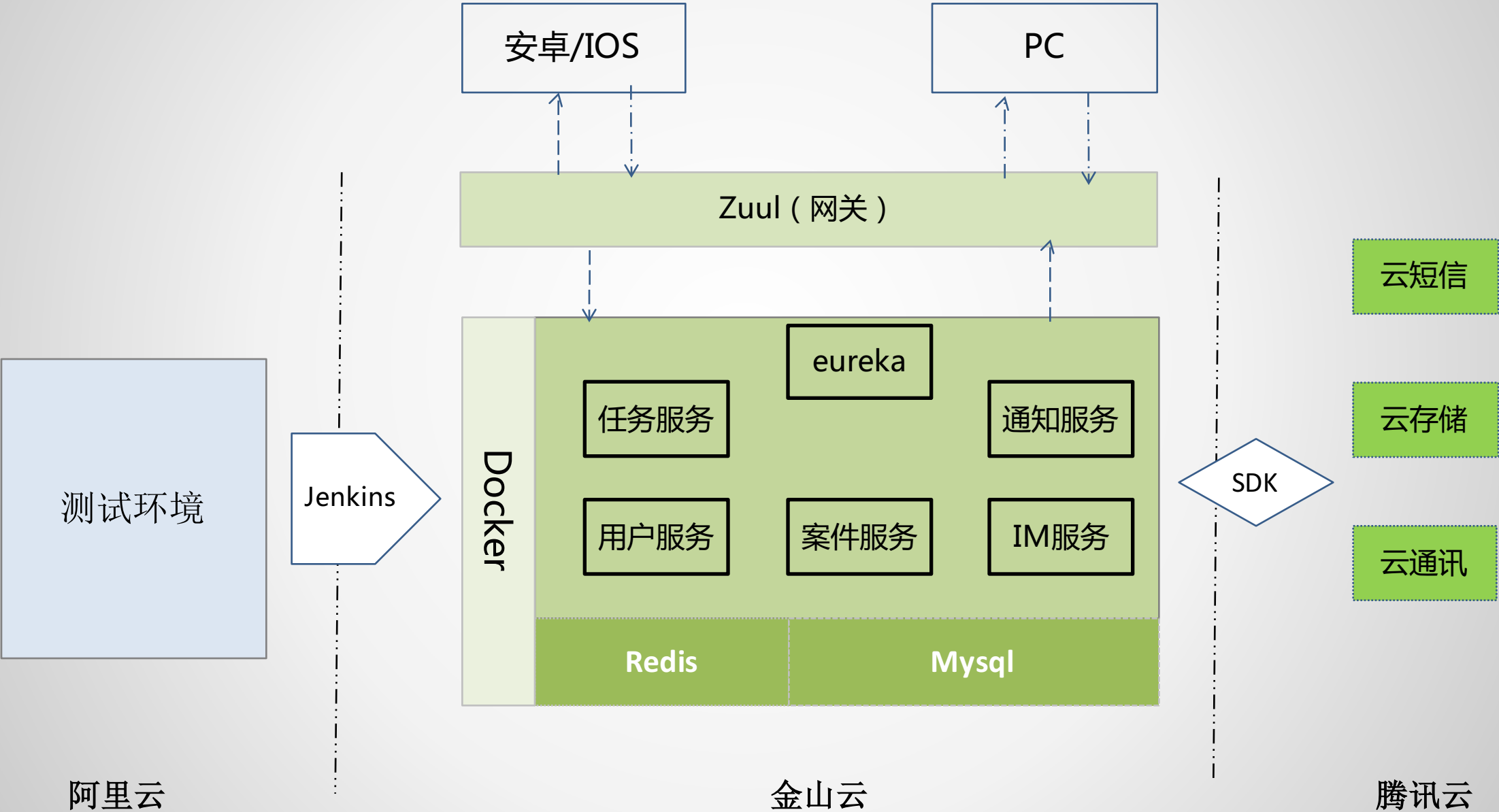


# 产品架构

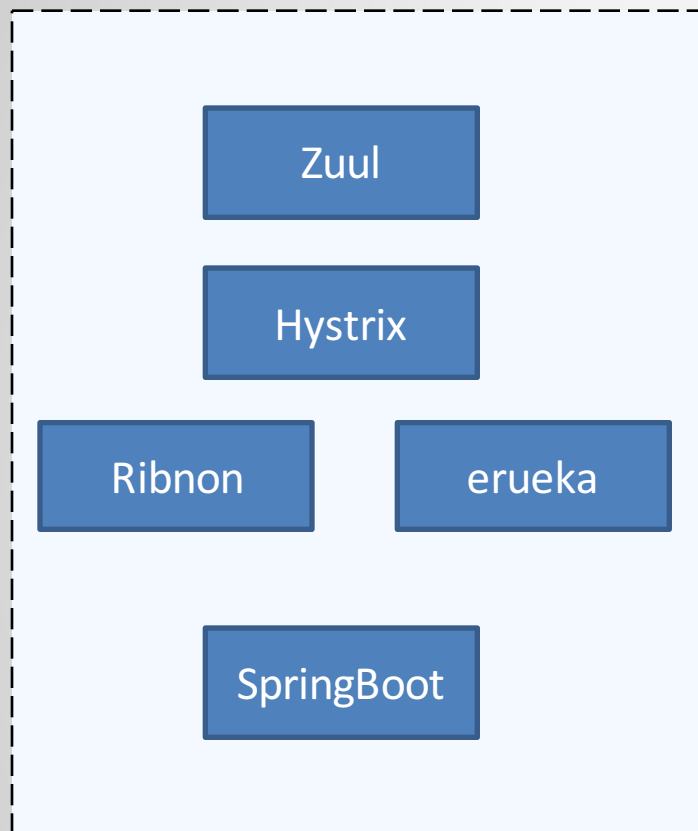
接入层

业务服务层

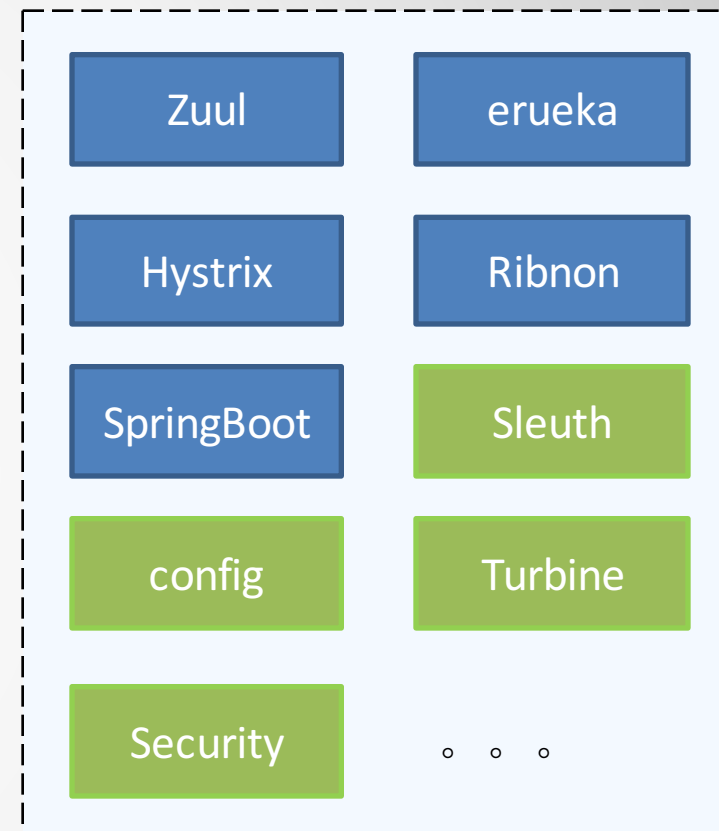
基础层



# 产品架构-最小化全家桶



逐步扩展



最熟悉技术，最简单组合，  
实施SpringCloud框架。  
确保充分学习组件，组件可控，  
迅速实现业务逻辑。

ps：本人在集成zipkin时，官方就出了sleuth，改造oauth2，官方出Security。不如躺着吃全家桶。

根据需求，团队掌握情况扩展组件  
应付业务需求

# 云商们的使用

	阿里云	金山云	腾讯云
性能 unixbench	100	126	120
磁盘IO	普通云盘连续读写50-60M	默认ssd	普通云盘300-400M
价格	100	105	98
控制面板	易用	困难	易用
产品功能	丰富	一般	丰富
特点	宽带便宜，综合最好，推荐使用	云直播，云存储，默认ssd盘	云通讯，云主机性能高



# 实践经验-选型

## ● 微服务框架技术选型的考虑点

### 框架成熟度

- 易于开发
- 易于迁移
- 多语言栈支持
- bug比较少

### 社区活跃度

- 文档丰富
- github issue活跃
- 易于招聘
- 未来发展性

### 学习曲线

- 适合团队知识体系
- 基于成熟技术构建

### 可维护性

- 易于扩展
- 是否开源
- 集成监控能力





# 实践经验-选型

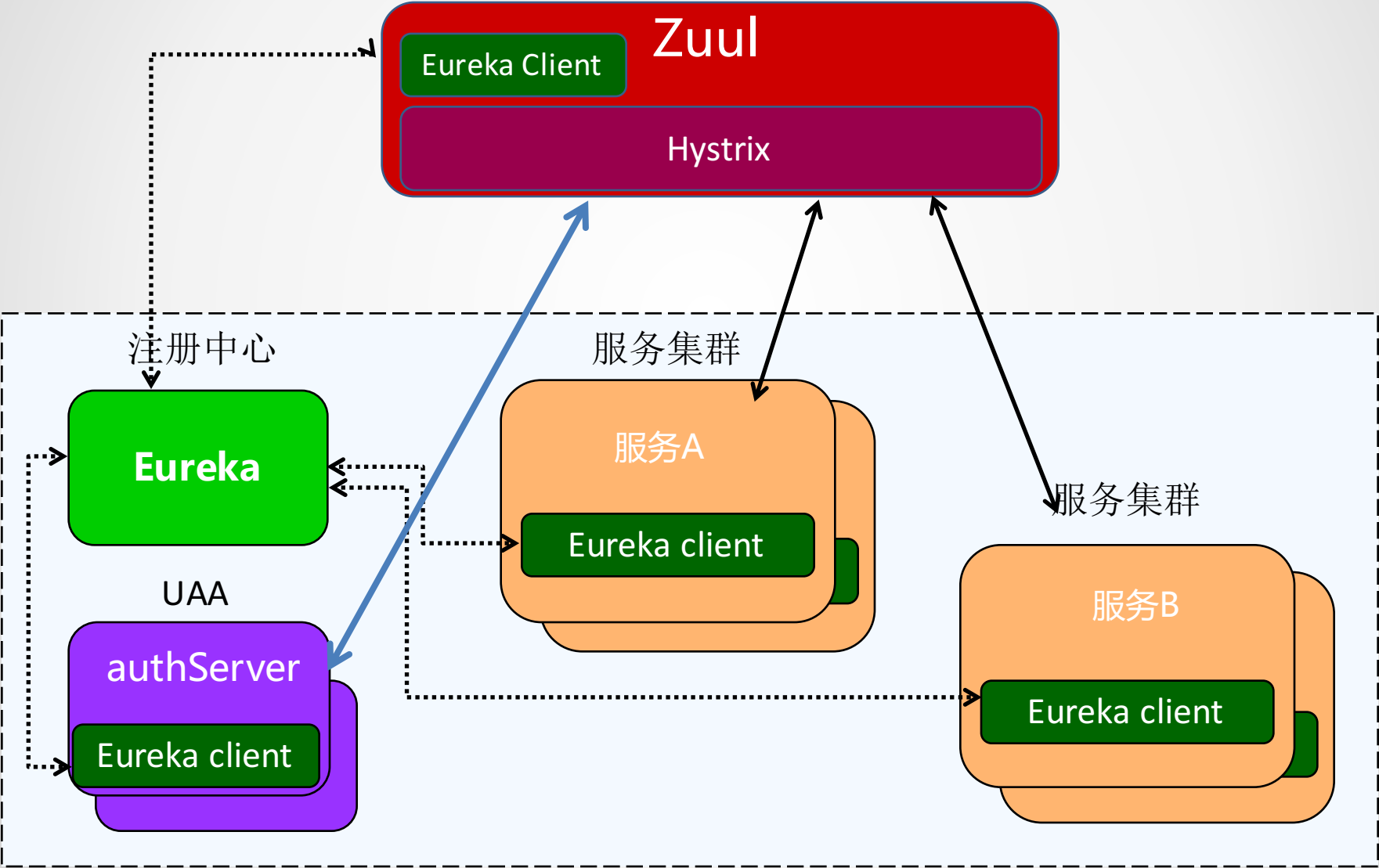
	SpringCloud	Dubbo	Motan	Thrift
定位	全家桶式微服务框架	服务框架	服务框架	Rpc框架
服务注册/发现	是	是	是	否
负载均衡	是	是	是	否
配置服务	是	否	是	否
高容错	是	是	是	否
调用方式	Rest/rpc(未来)	rpc	rpc	rpc
社区活跃度	高	弃坑	一般	一般
学习难度	低	中	低	高
文档丰富度	高	高	一般	一般





# 实践经验-权限认证

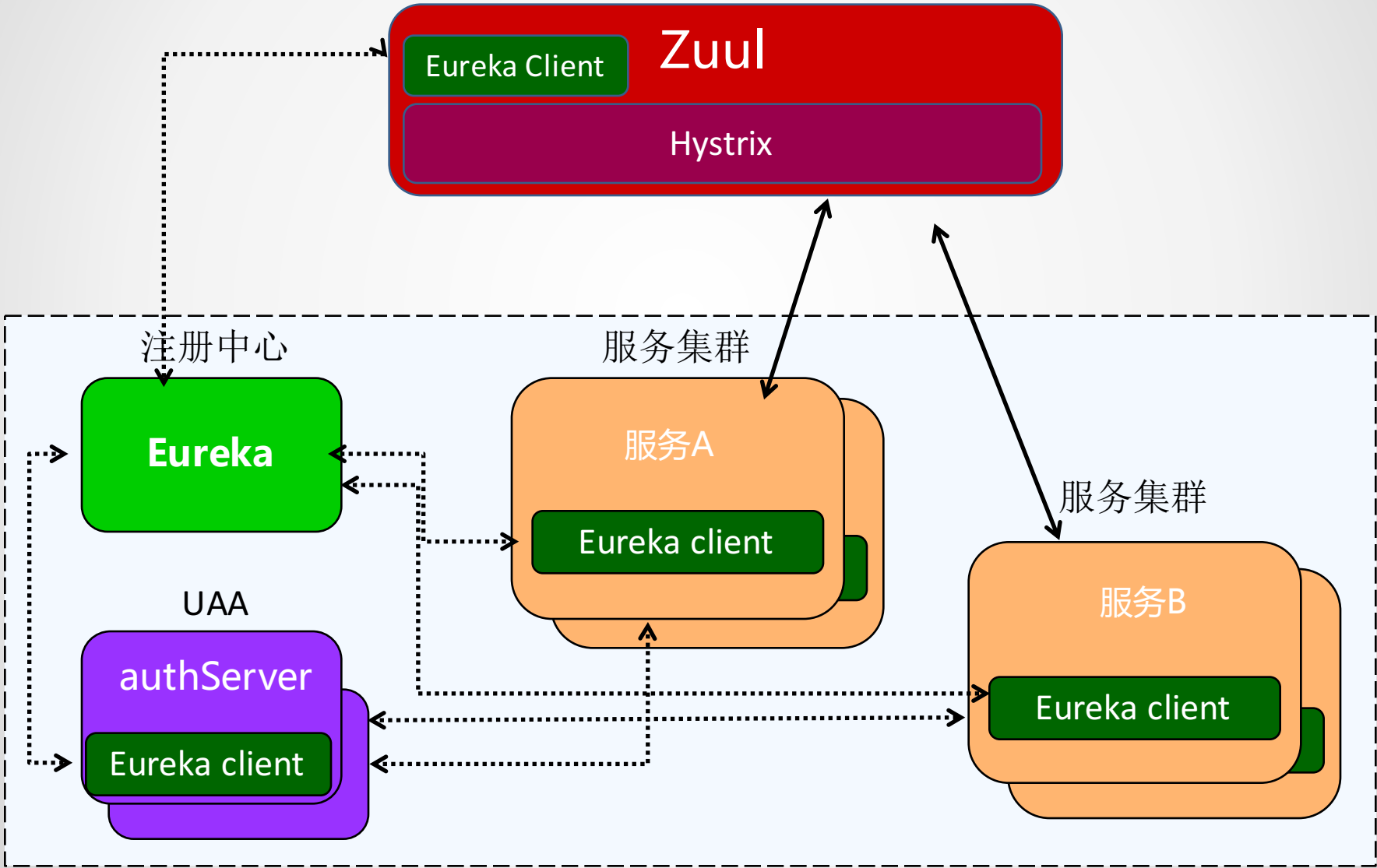
统一由ZUUL与UAA校验(<https://github.com/kakawait/uaa-behind-zuul-sample>)





# 实践经验-权限认证

每个微服务与UAA相互校验(<https://github.com/sqshq/PiggyMetrics>)



- 什么是RAP <https://github.com/thx/RAP>或<http://rapapi.org/org/index.do>

RAP是一个可视化接口管理工具 通过分析接口结构，动态生成模拟数据，校验真实接口正确性，围绕接口定义，通过一系列自动化工具提升我们的协作效率。

## RAP的特色

### 1. 轻松编辑与分享

可视化编辑，完善的版本控制，各种格式的导入导出。让前后端约定接口的工作变得十分简单

### 2. Mock服务

RAP会自动根据接口文档生成Mock接口，这些接口会自动生成模拟数据，支持复杂的生成逻辑。

### 3. 实力的保障

覆盖阿里几乎所有部门，有200+公司正在使用RAP，历经4年积累大量经验，可适应各种复杂的业务场景。

# 实践经验-RAP

R RAP v0.14.0 beta

主页团队文档数据教程

搜索项目名称

消息1

我的主

R RAP v0.14.0 beta

主页团队文档数据教程

搜索项目名称

消息1 yujiase

我的

新品研发里程碑任务 页面控制台

请求根路径  
mp.mesclouds.com:8090/mockjs/14

1.新增或修改里程碑任务 4.删除一个里程碑任务 2.根据id获取一个里程碑任务 3.根据新品项目id获取该项目所有的里程碑任务 5.获取我相关的里程碑任务

test123  
test123  
请求地址: /extapiserver/marubi/newproducttask/saveupdate  
taskinfo  
8

我加

资料  
5

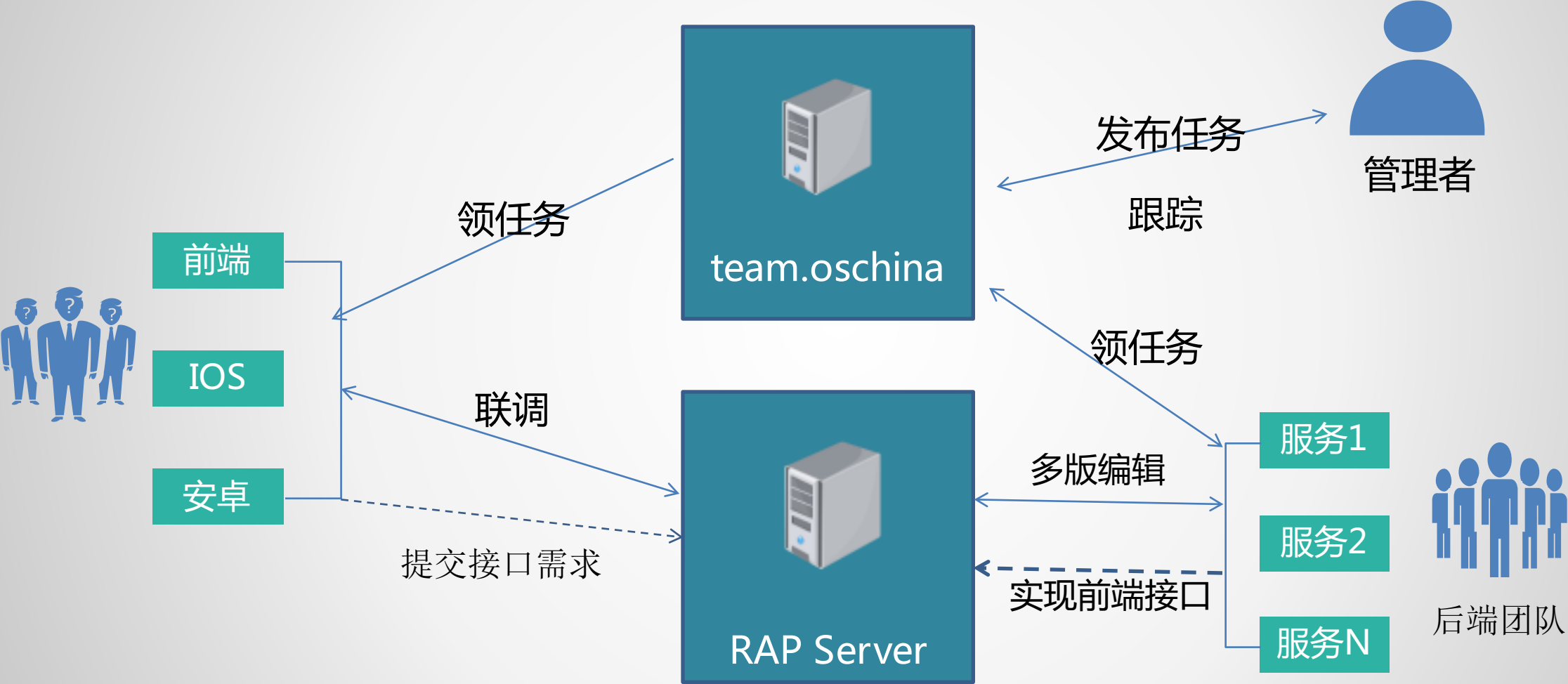
阿里爸爸喊你：  
提高效率，回家吃晚饭！

```
{  
  "code":200,"data":  
  {  
    "checkAndAcceptanceDepartmentId":"dasd232sadas232","checkAndAcceptanceDepartmentName":"营销部","checkAndAcceptancePersionId":"32423sdfssdas","checkAndAcceptancePersionName":"李xx","cooperateDepartment":[  
      {  
        "DepartmentId":"sdas234234sad","DepartmentName":"企划部","responsiblePersonId":"xxsda3423423sd32","responsiblePersonName":"杨xx"  
      }  
    ],  
    "description":"备注xxx要买","endDay":60,"id":"fsfgsdt45tdfge5tdferwsda3","isAllowDelete":0,"isMilestone":1,"isTimeOut":0,"milestoneTime":1484489922000,"parentId":"sda3sads343456sdsdsd","responsibleDepartmentId":"xx2323456sdasdas","responsibleDepartmentName":"BA3E","responsiblePersonName":"肖xx","seq":1,"startDay":30,"status":0,"taskName":"任务"  
  }  
}
```

4 request end in:412ms.  
3 request starting, url: http://mp.mesclouds.com:8090/mockjs/14/extapiserver/marubi/newproducttask/saveupdate?taskinfo=&t=



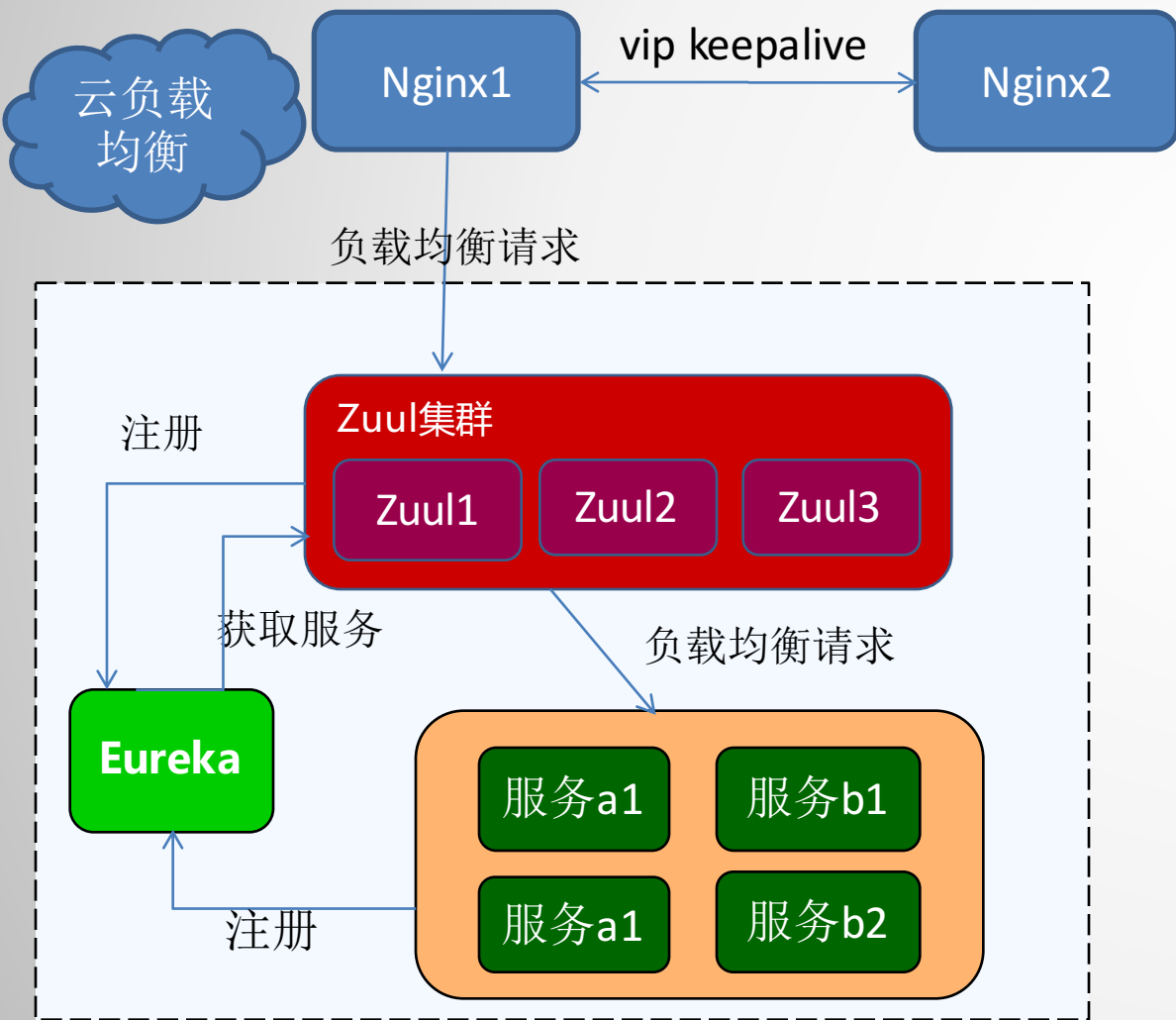
# 实践经验-前后端联调



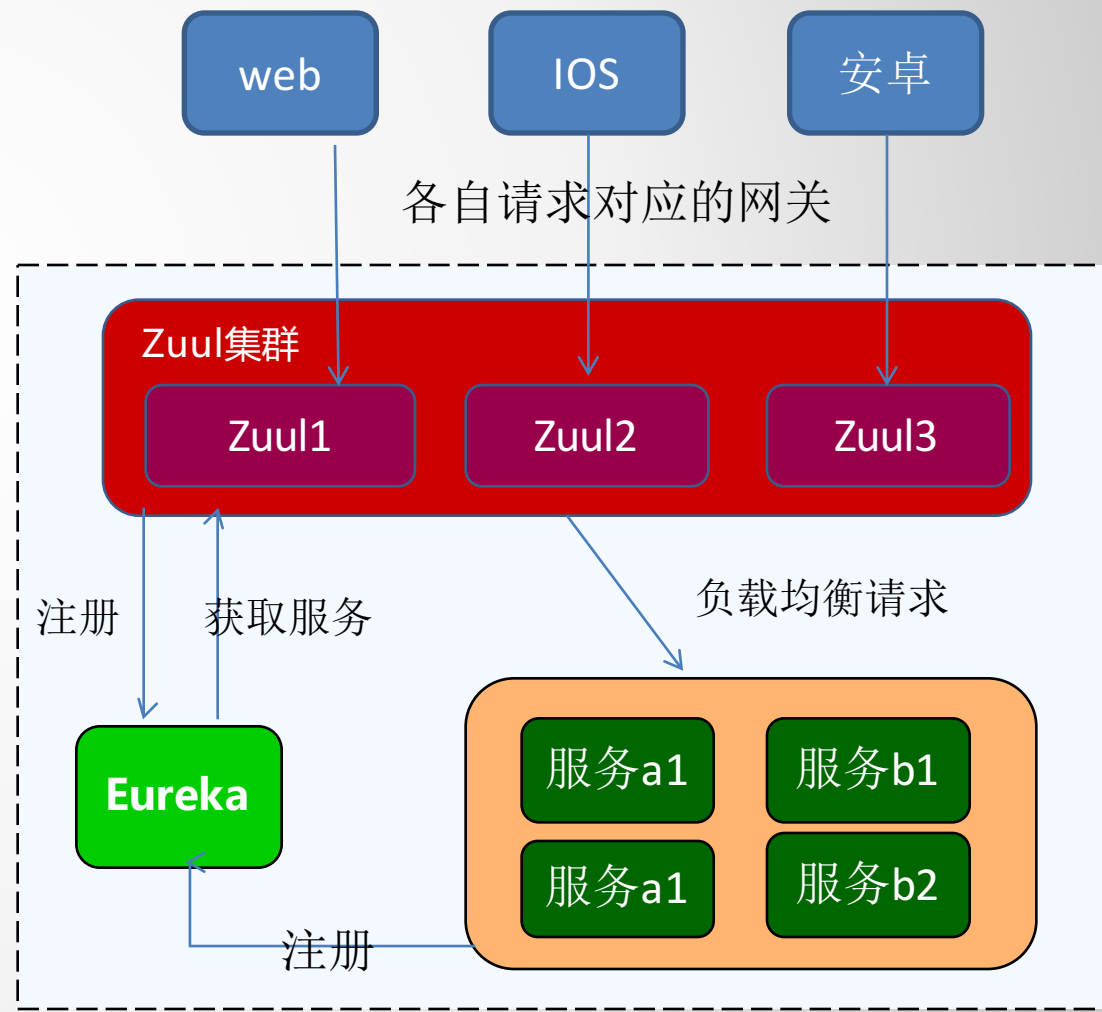


# 实践经验-Zuul高可用

高可用方案A



高可用方案B



# 总结-常见问题

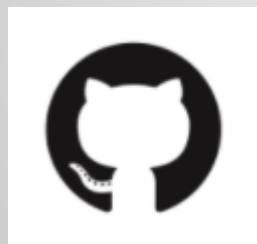


## 云主机部署

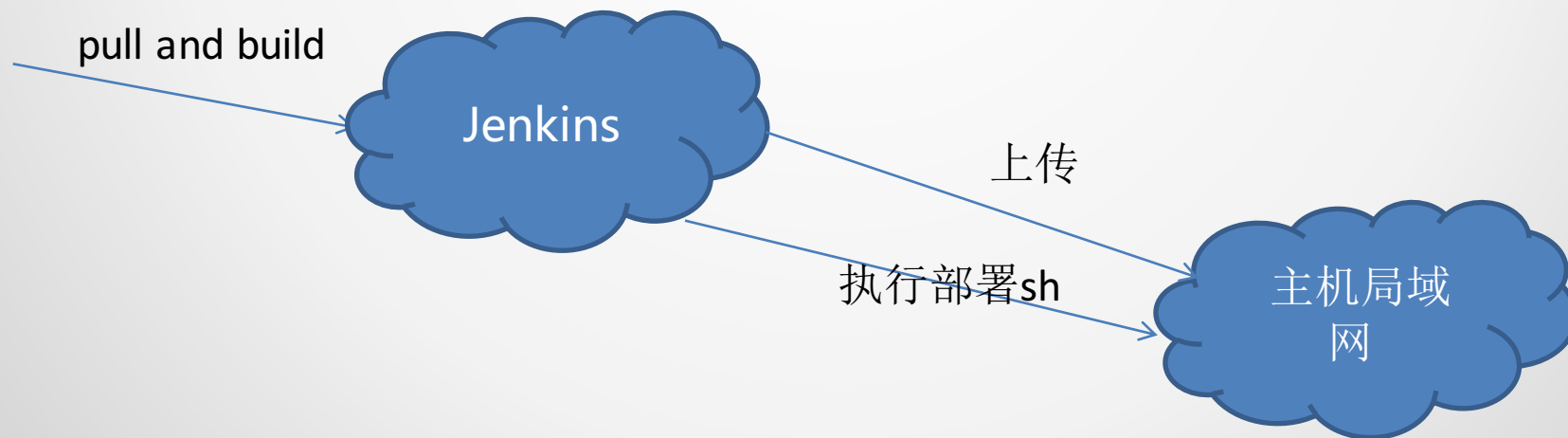
主机宽带有限，jar文件过大，部署更新较慢  
不是每一台主机具备独立ip



## 建议



git







## 是否使用JWT ( JSON WEB TOKEN )

- 1、JWT服务端签名认证性能稍慢
- 2、与设计认证方式冲突 ( 基于cookie或者session )



## 建议

- 前后端分离设计和集成Springcloud Security，有一站式解决方案，可以使用。
- 参数不要太多或者只带基本参数，其他参数明文传递。
- 非前后分离设计，使用传递认证方式(cookie或session)更容易实施起来



## 各云主机性能不一致，如何充分利用资源

服务集群因压力不同，横向伸缩加入新主机，而新主机与正运行主机性能有差异（云主机有升级周期，不同时期的逻辑核心，内存，IO性能 不尽相同）



## 建议

- 尽量使用docker方式部署，高性能主机运行多个docker实例注册到erueka，使用Springcloud默认策略调度
- 扩展实现权重策略  
参考<http://xujin.org/sc/sc-ribbon-demoted/>



# 感谢您的聆听

Talents come from diligence,  
and knowledge is gained by accumulation.