

# 巧房微服务实战面面观

巧房科技 / 刘朝刚

SHANGHAI



CNUTCon [上海]  
全球运维技术大会

主办方 Geekbang & InfoQ  
极客邦科技

# 50+ 年末充电<sup>⚡</sup>

## 开发&运维技术干货大盘点

容器

Kubernetes

DevOps

全链路压测

Serverless

自动化运维

Service Mesh

Elasticsearch

微服务

使用折扣码 「QCon」 优惠报名 咨询电话：13269078023



扫码锁定席位



# TABLE OF CONTENTS 大纲

---

- 背景
- 系统架构
- 持续集成/持续发布
- 典型问题剖析
- 未来规划

# 背景

# 业务背景

- ◆ 巧房主要是为**房产中介经纪人**提供一个安全、稳定、高效、智能的**SAAS**平台，主要涵盖如下几块功能：房源管理、客源管理、交易管理、财务管理、人事OA、运营分析等功能。
- ◆ 目前服务**4000**多家房产中介公司，**30W**房产经纪人，分布在全国20多个省份，100多个城市。当前有上海、广州两个IDC机房作为两个分区并在阿里云上面搭建了第三个分区。

# 单体架构



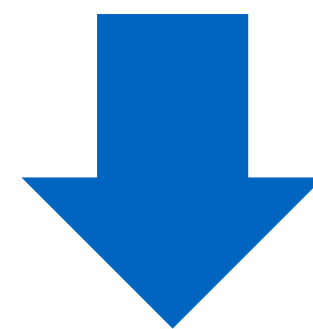
# 技术挑战

开发上线成本高周  
期长

牵一发而动全身

系统扩展、架构演  
进难度大成本高

应用扩容成本  
高、效率低



微服务

# 系统架构



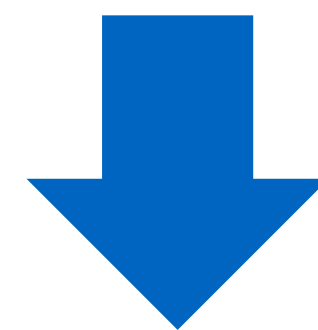
# 技术选型

## Spring Cloud

- 组件丰富，兼容性好
- 开发部署，简单快速
- 系统组件，扩展性强

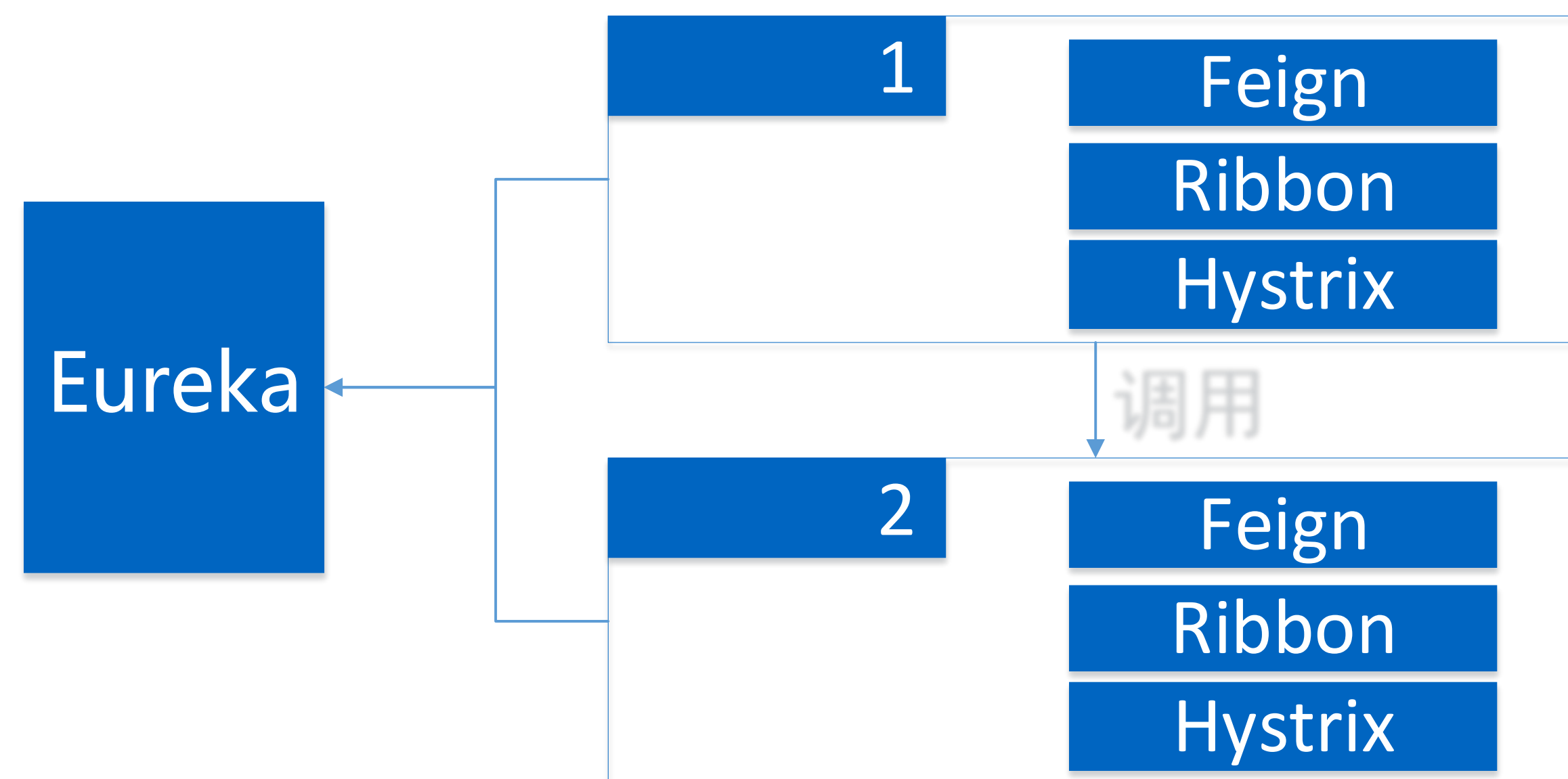
## Kubernetes

- 管理维护docker，功能强大
- 超强的扩容、缩容能力
- 系统稳定，容错能力强

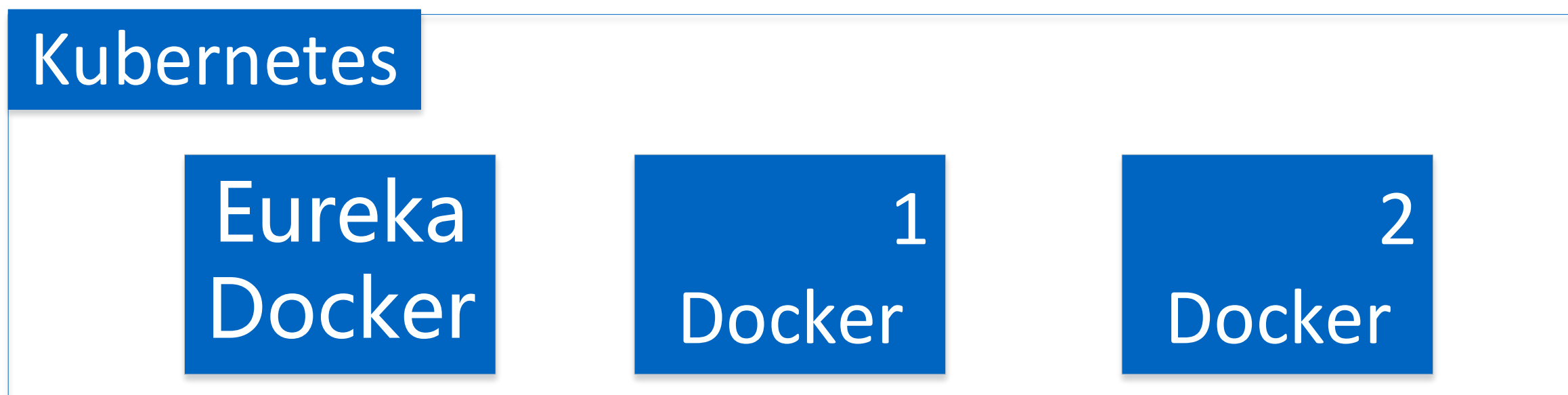


基于Spring Cloud和Kubernetes的微服务系统

# 微服务1.0架构核心



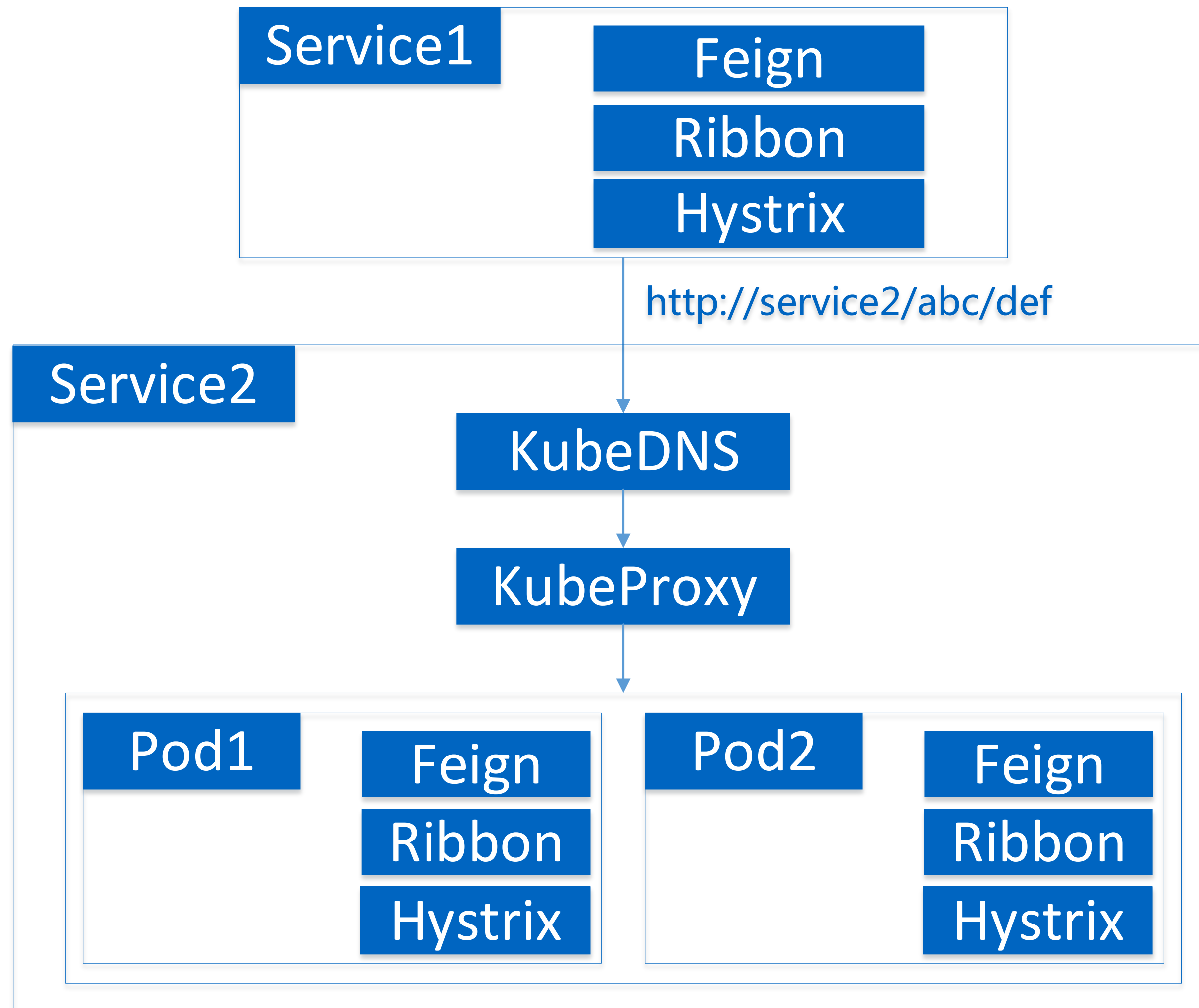
- 最大限度地实现解耦，开发效率大大提升、部署上线简单快速。
- 各个微服务高内聚，各司其职，接口设计更加轻量、合理。
- 降低了整个服务器的成本，扩容缩容简单易用



# 微服务1.0架构不足

- 服务上线与下线，调用服务需要一段时间后才从自己缓存的服务列表中移除。
- 在客户端做负载均衡，对开发来说需要感知，且微服务有点重。
- 无法兼容与使用Kubernetes在微服务运维与监控方面做的很好的一些功能，比如扩容缩容，灰度发布等等。

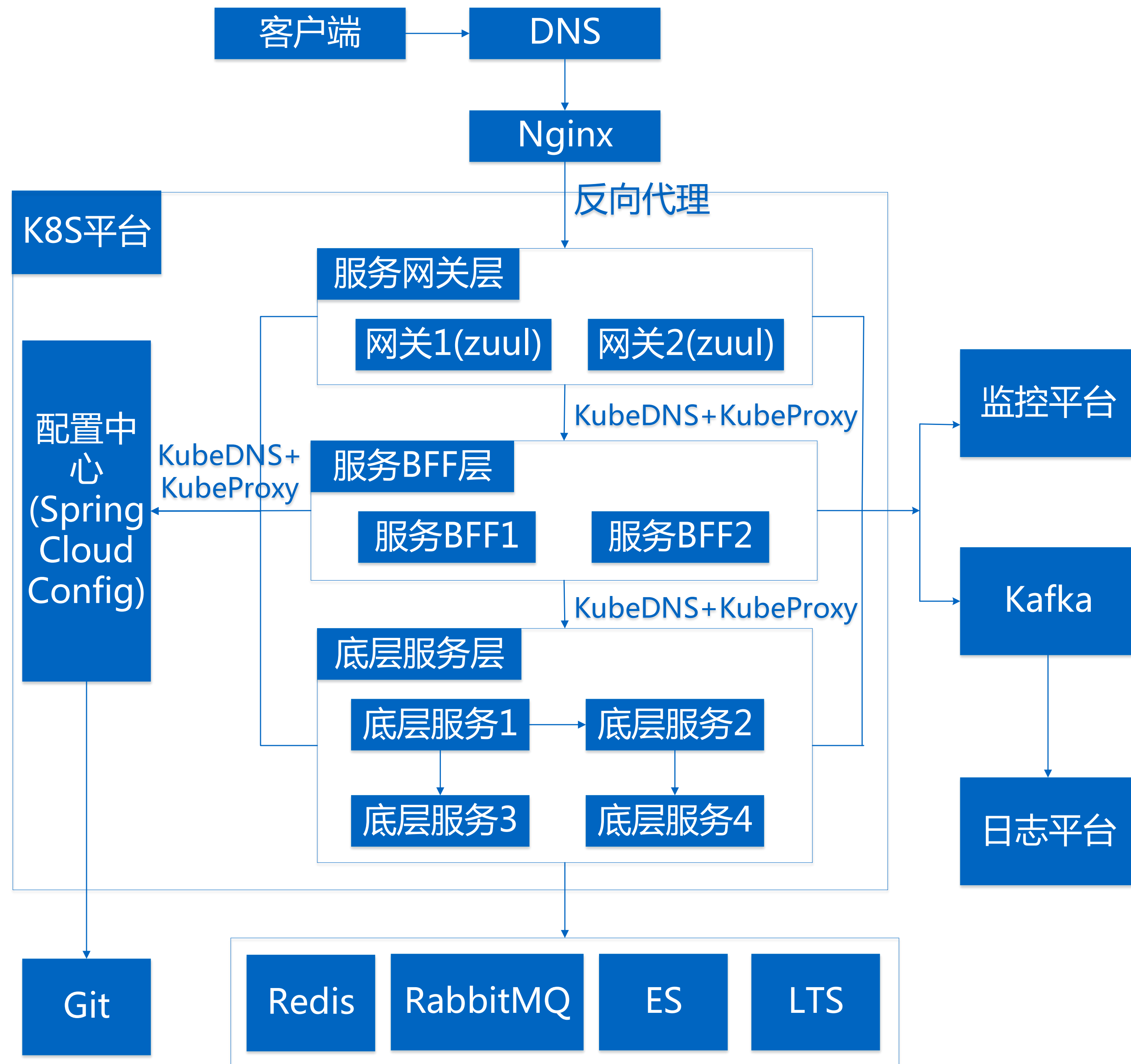
# 微服务2.0架构核心



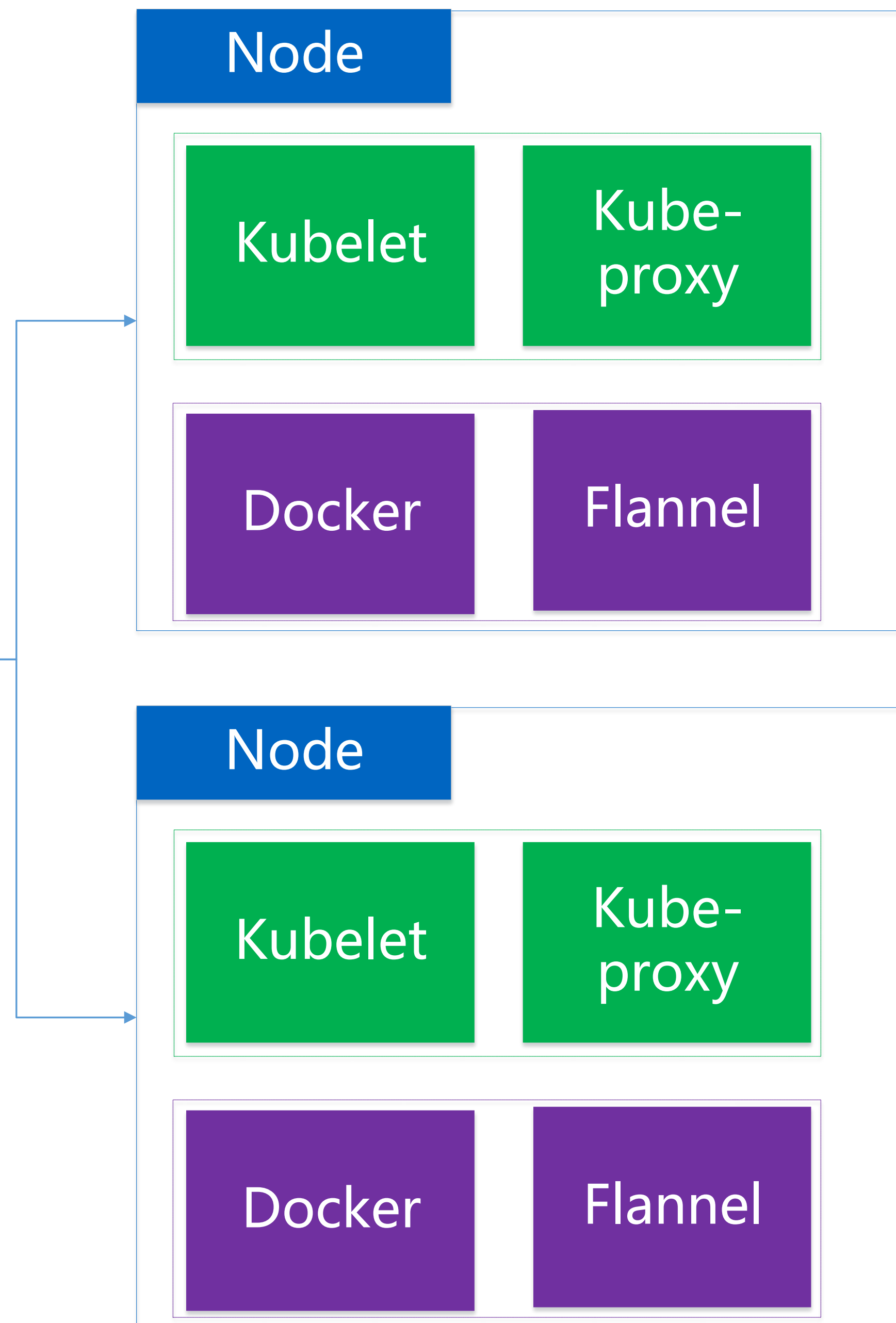
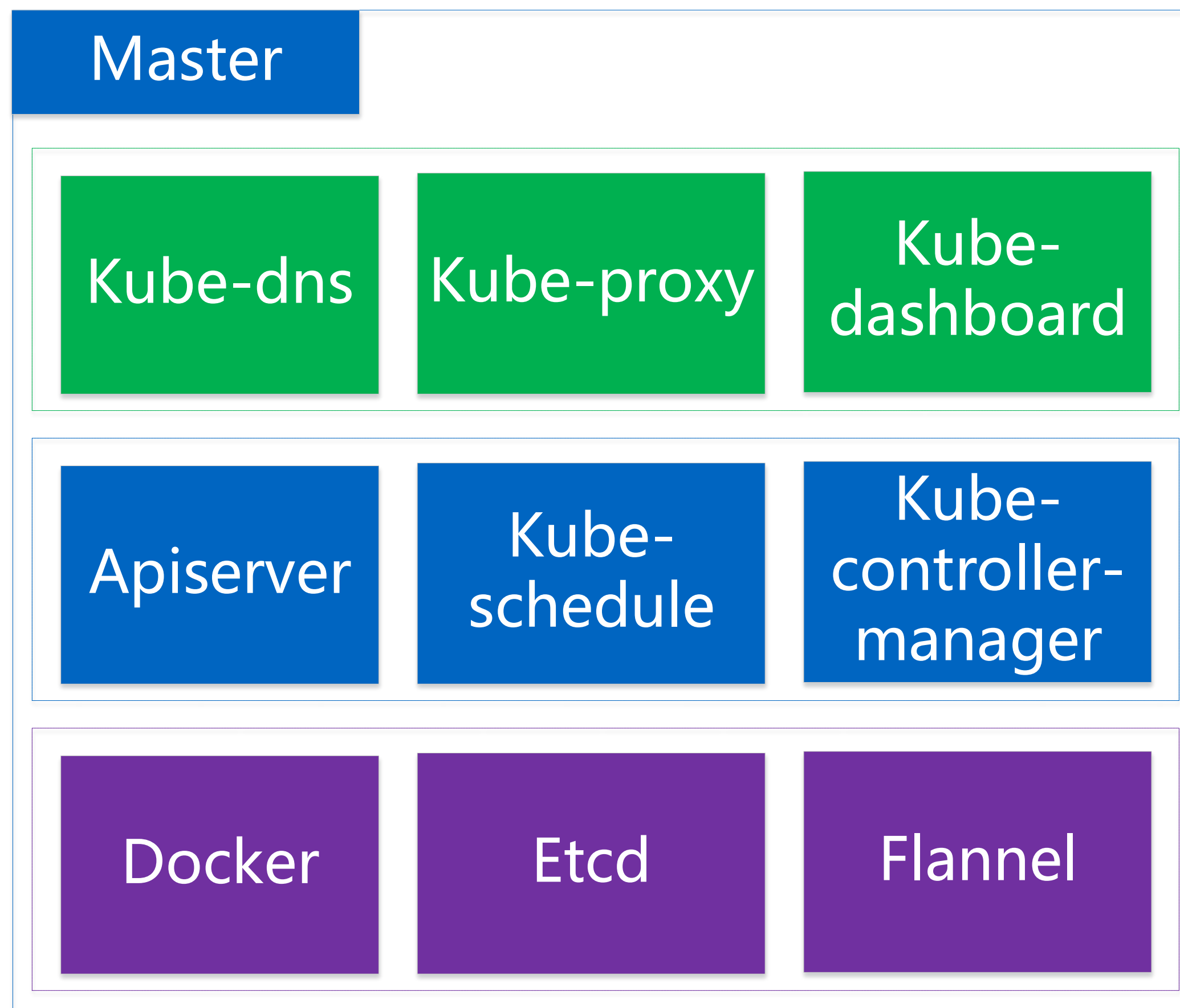
- 相对于之前的客户端负载均衡来说，这是典型的服务端负载均衡，此时客户端非常轻量，只做业务的处理。
- 使用Kubernetes代替Eureka做服务治理。
- 继续使用Spring Cloud中的其他组件，原来配置name来访问其他组件的地方，全部换成使用url来访问。



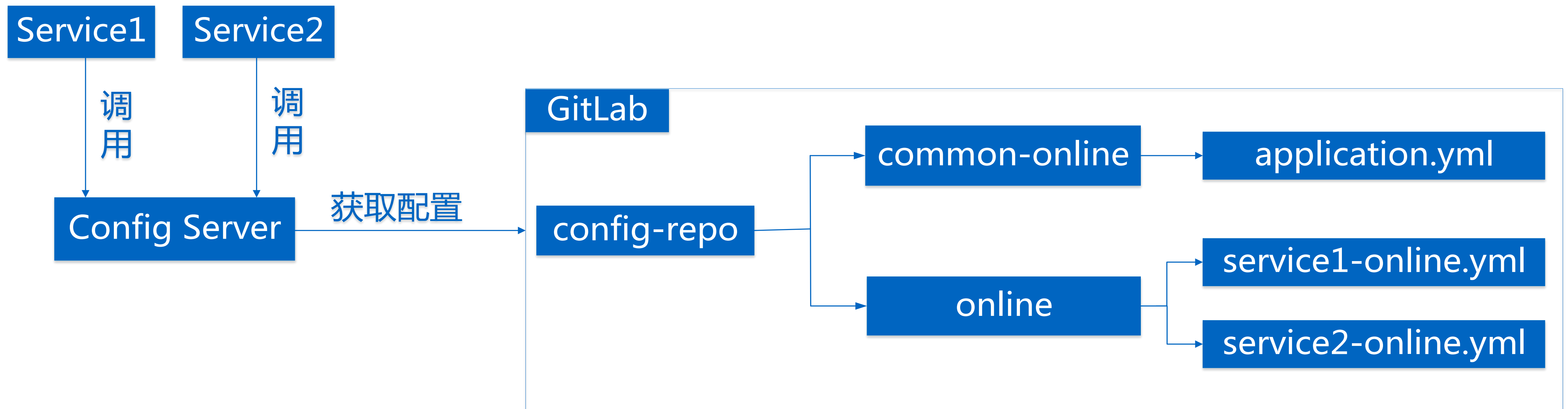
# （架构总览图）



# (Kubernetes)



# 架构详解~配置中心



- SearchPath中定义两个路径，分别存放服务具体配置与公共配置。
- 优先使用文件名与服务名完全一致的配置文件，如果没有再去 application\*.yml中查找并使用找到的配置值。

# 架构详解~服务网关





# 架构详解~服务调用（1）

## Service1 TestClient

```
@FeignClient(url="${feign.url.service1}")
public interface TestClient {
    @RequestMapping(value = "/testabc")
    public testabc();
}
```

调用接口

## Service2 TestFacade

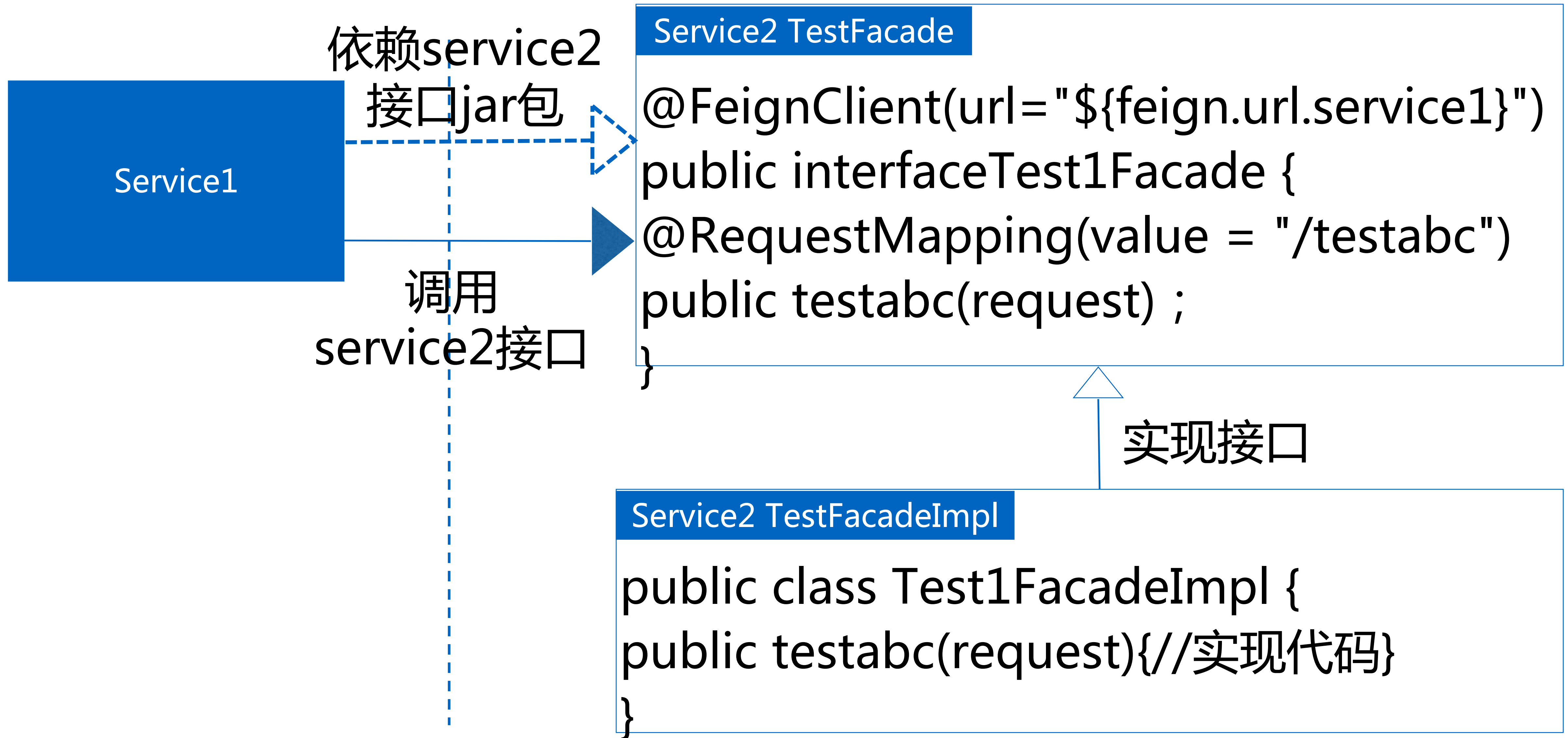
```
public interface Test1Facade {
    public testabc(request) ;
}
```

实现接口

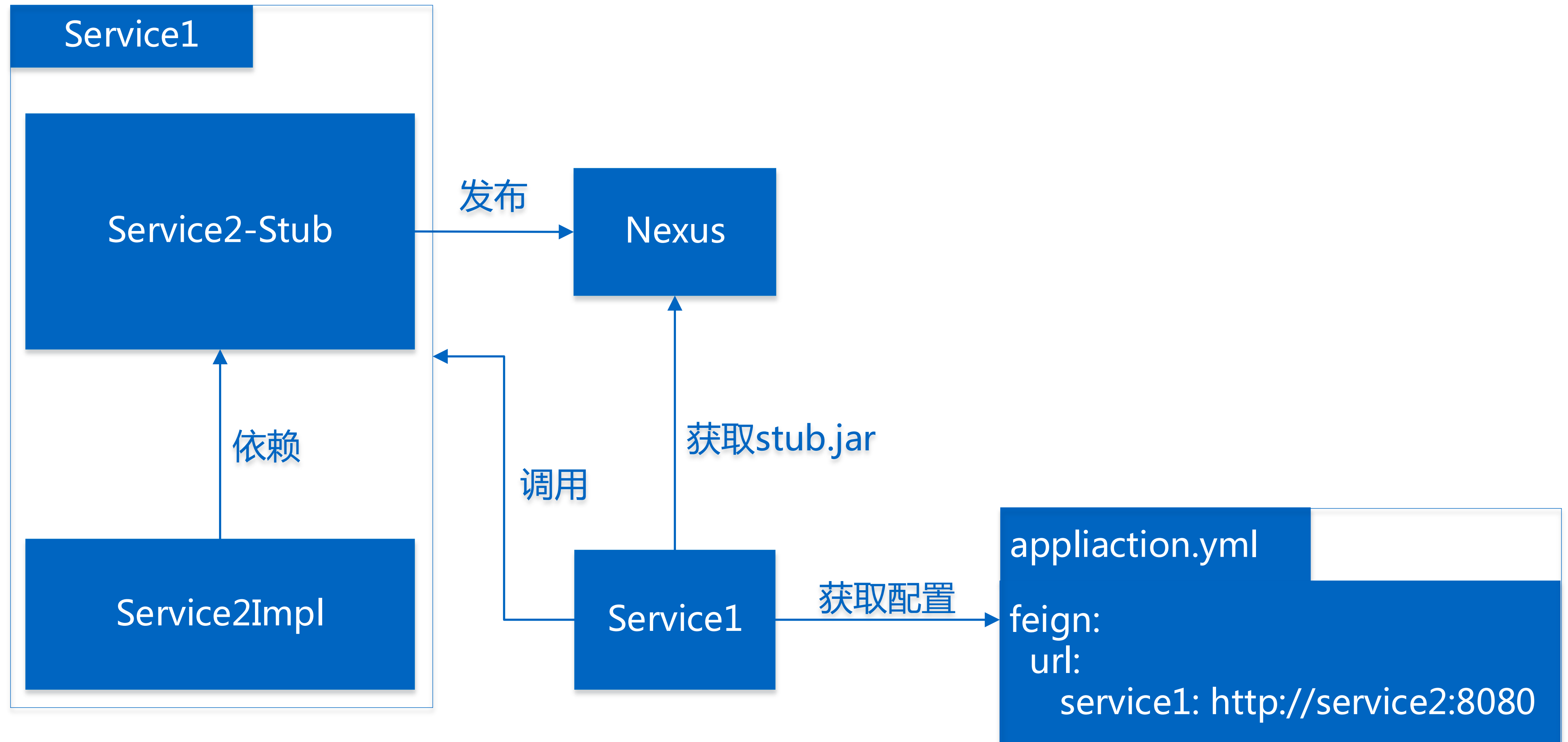
## Service2 TestFacadeImpl

```
public class Test1FacadeImpl {
    @RequestMapping(value = "/testabc")
    public testabc(request){//实现代码}
}
```

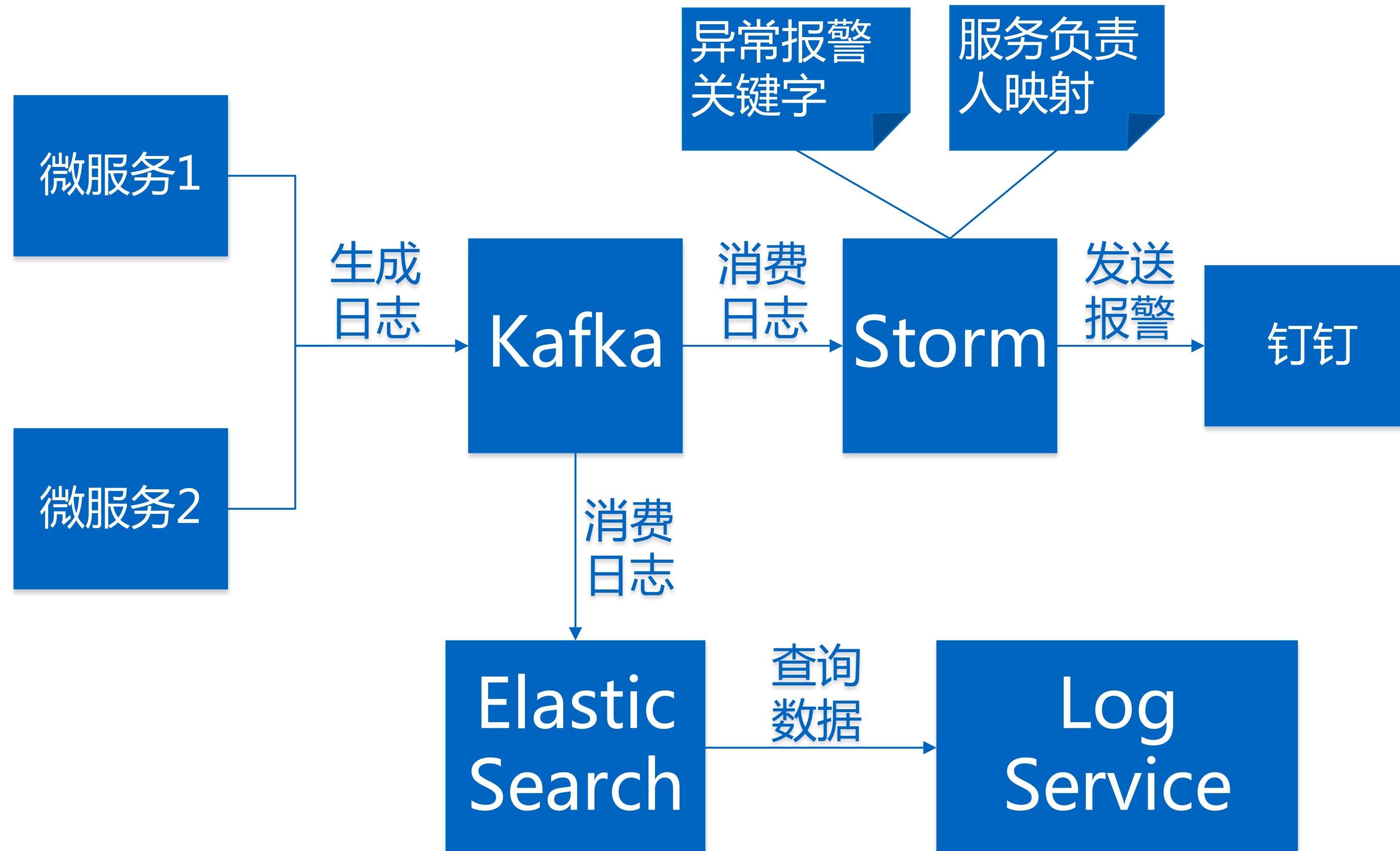
# 架构详解~服务调用（2）



# 架构详解~服务调用（3）



# 架构详解~日志处理（1）





# 架构详解~日志处理（2）

日志查询

日志汇总 > 服务日志查询

时间范围

2015/00/00 00:00:00

服务名称

x ▼

日志等级

全部 ▼

链路ID

链路ID

IP地址

IP

类名

类名

日志信息

日志信息

thrown

异常信息

线程名称

线程名称

用户UUid

用户UUid

公司UUid

公司UUid

日志环境

全部 ▼

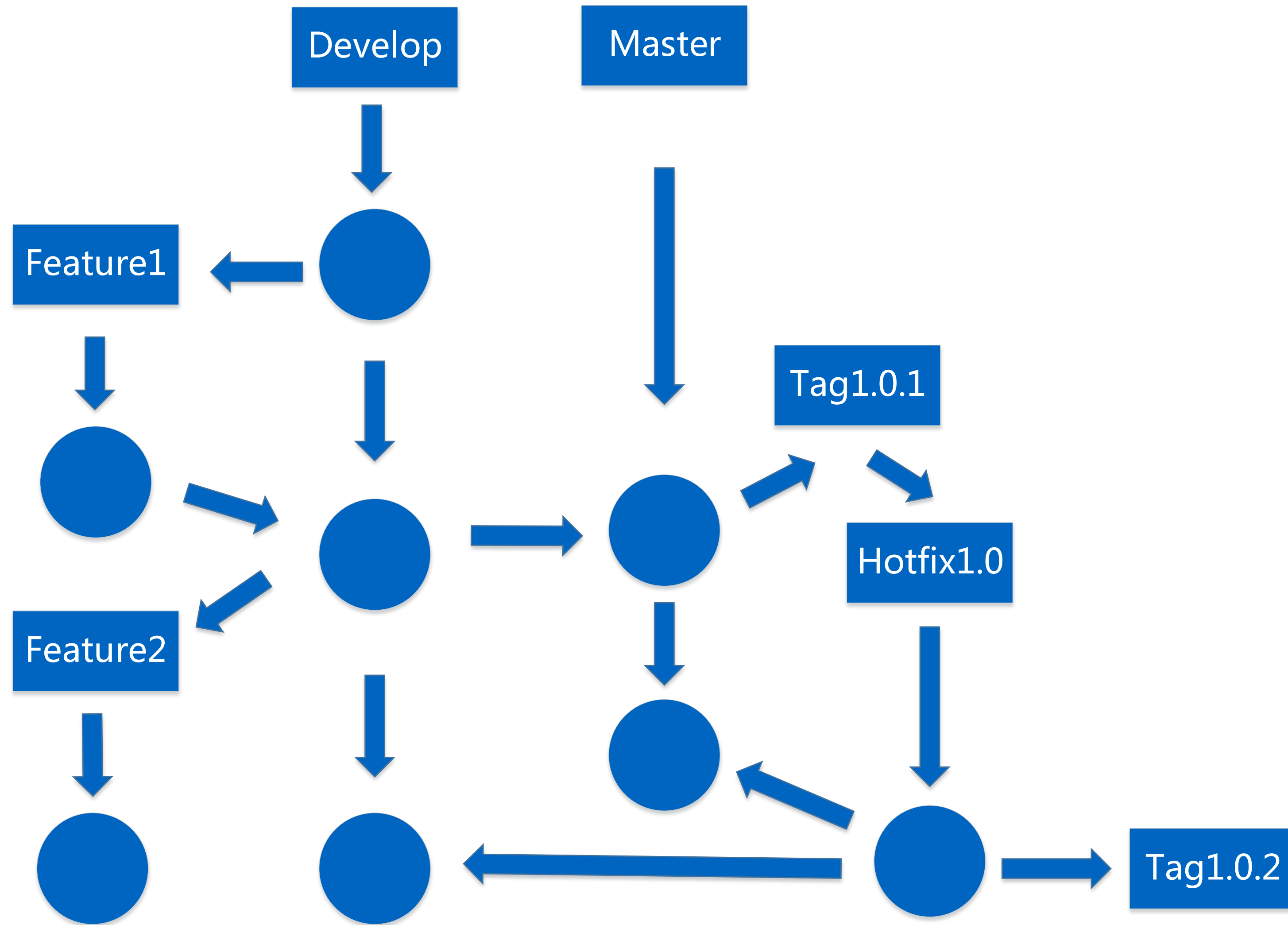
Q 查询

每页显示10条

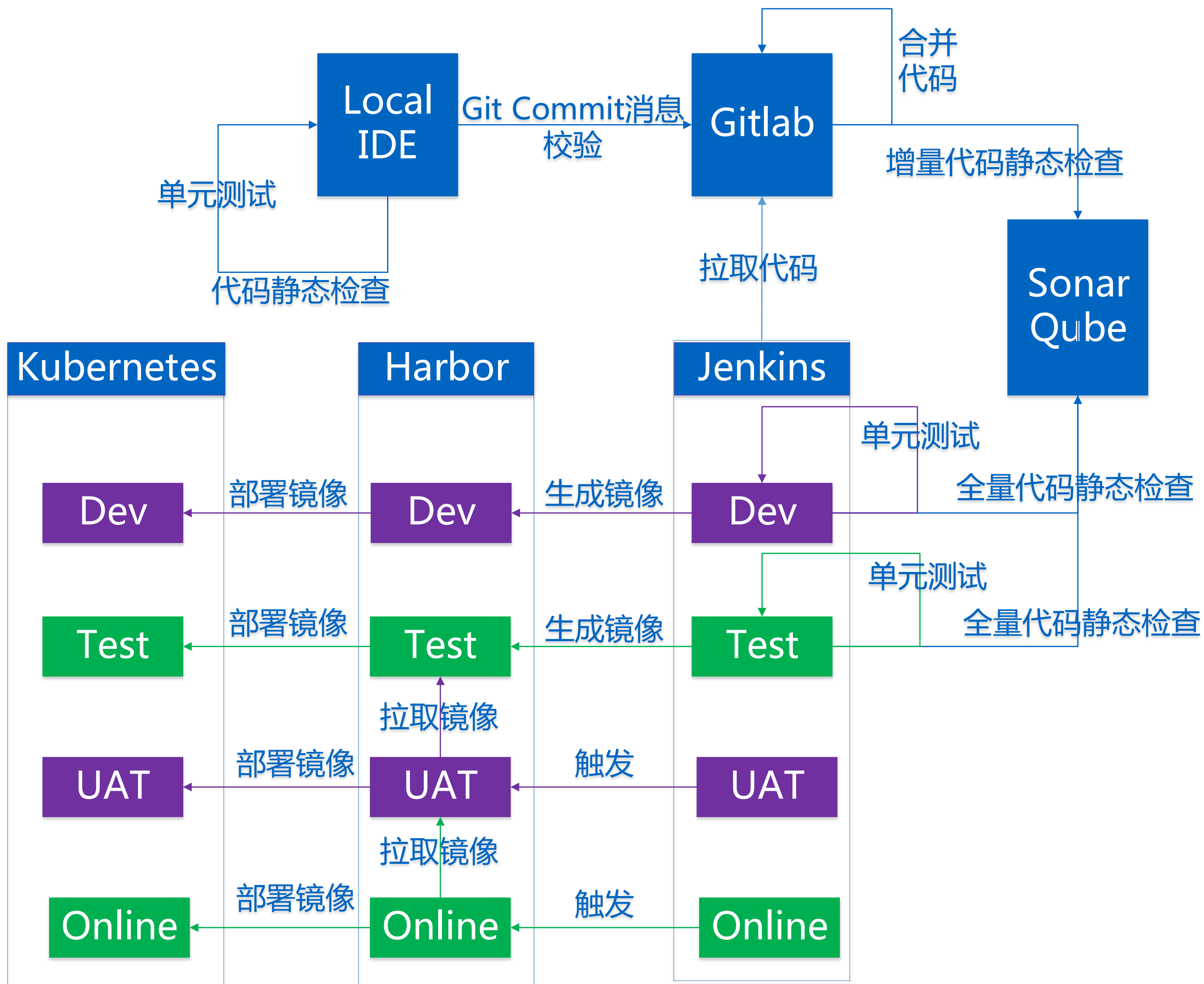
AppName	时间	链路ID	日志等级	IP地址	公司ID	用户ID	日志名称	日志信息	操作
	2015-00-01 21:00:00	711238d9c...	INFO				g.U...er. cachon...i	getting permission from redis...	详情

# 持续集成/持续发布

# CICD – 代码分支管理



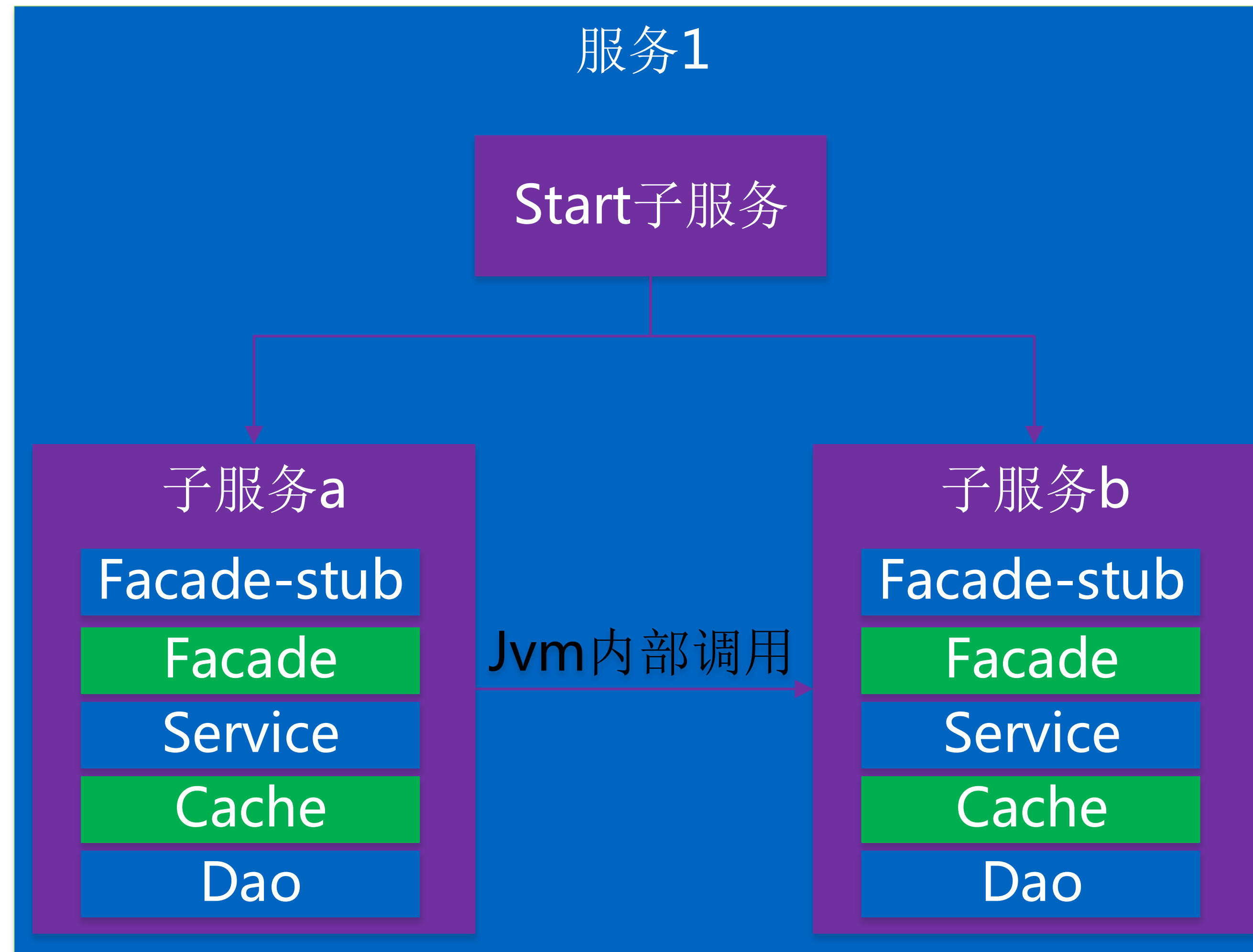
# CI/CD 流程图



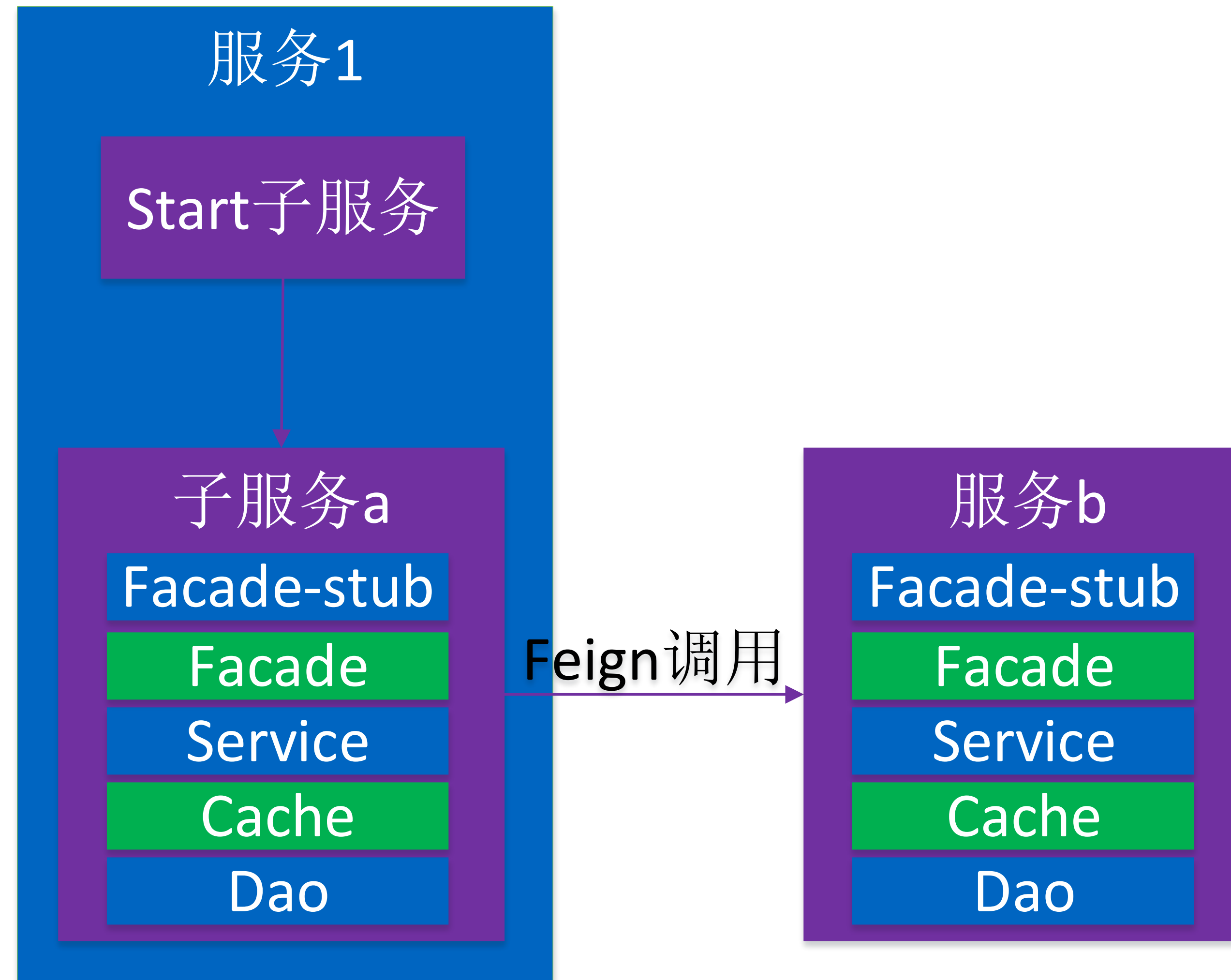


# 典型问题分析

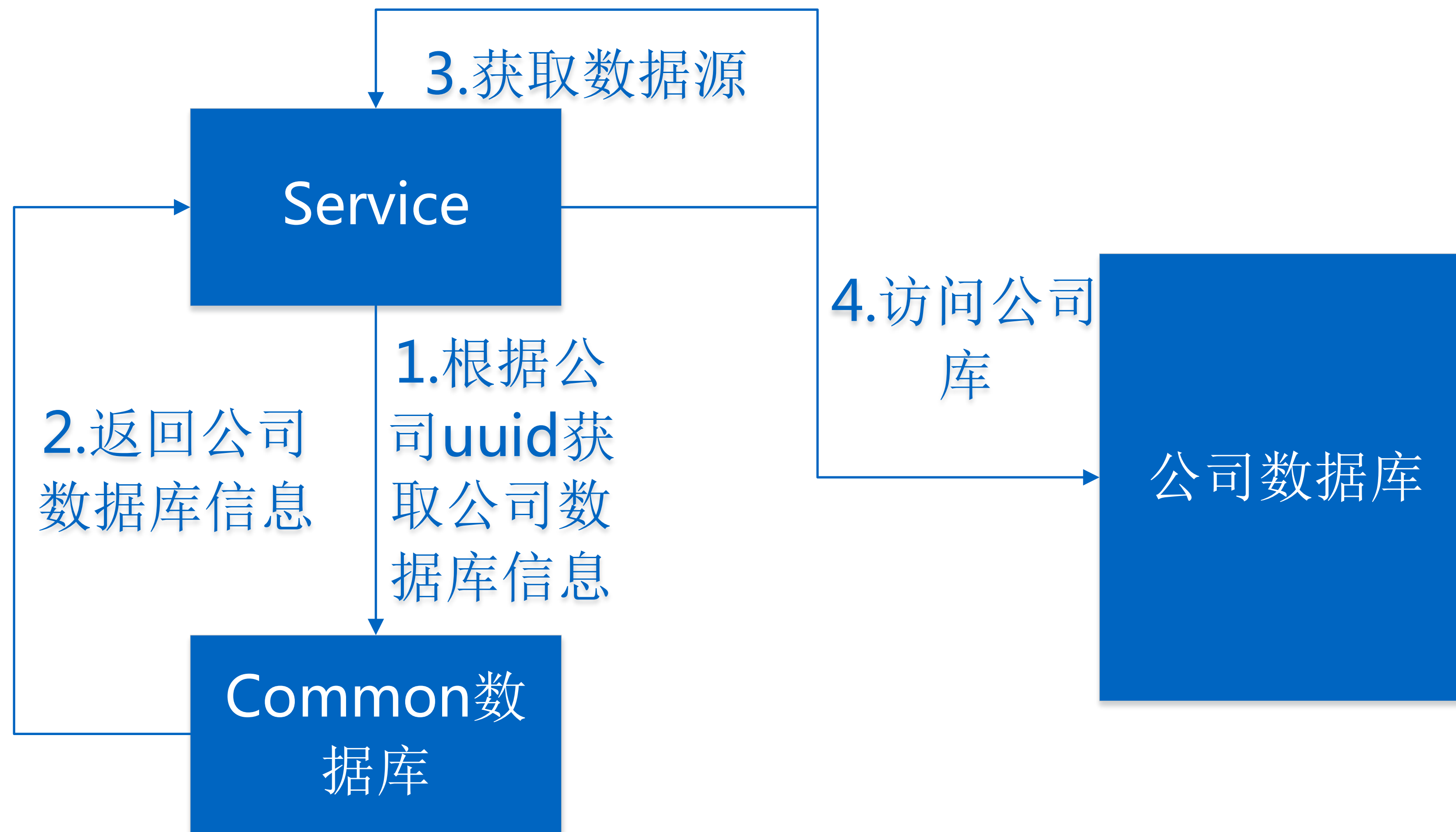
# 服务拆分错了怎么办（1）



# 服务拆分错了怎么办（2）



# 数据库连接不够问题（1）



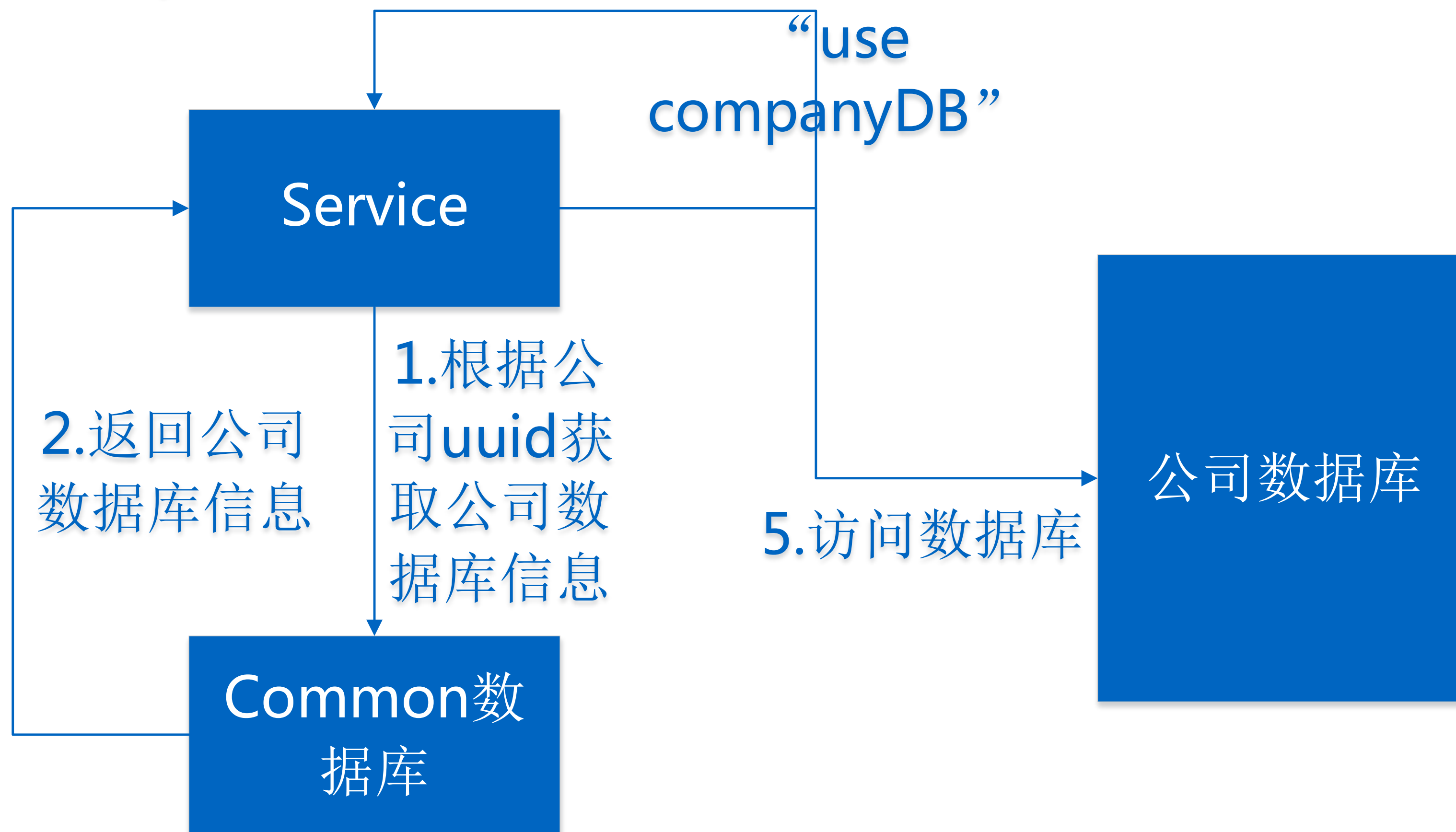
单体：20（连接数）\*10  
（服务实例数）\*1（服务数量）\*100（公司数） =  
20000连接数

微服务：10（连接数）\*5  
（服务实例数）\*50（微服务数量）\*100（公司数）  
=250000 连接数

一台数据库服务器所能接受的连接数也就是几万，微服务架构下远远超过这个值。

# 数据库连接不够问题（2）

- 3.以数据库serverId为key，创建数据源      4.Mybatis拦截器来修改SQL，添加



微服务：100（连接数）\*5（服务实例数）\*1（数据库服务器）\*50（微服务数量）=25000 连接数



# 数据库连接不够问题 ( 3 )

com.microsoft.sqlserver.jdbc.SQLServerException: The prepared statement handle 6 is not valid in this context. Please **verify that current database**, user default schema, and ANSI\_NULLS and QUOTED\_IDENTIFIER set options are not changed since the handle is prepared.

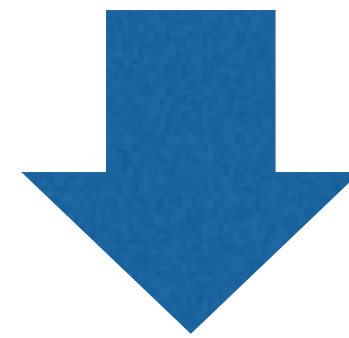
```
USE dbName ;
```



```
connection.setCatalog(dbName)
```

# unable to create new native thread

Caused by: java.lang.OutOfMemoryError:  
unable to create new native thread

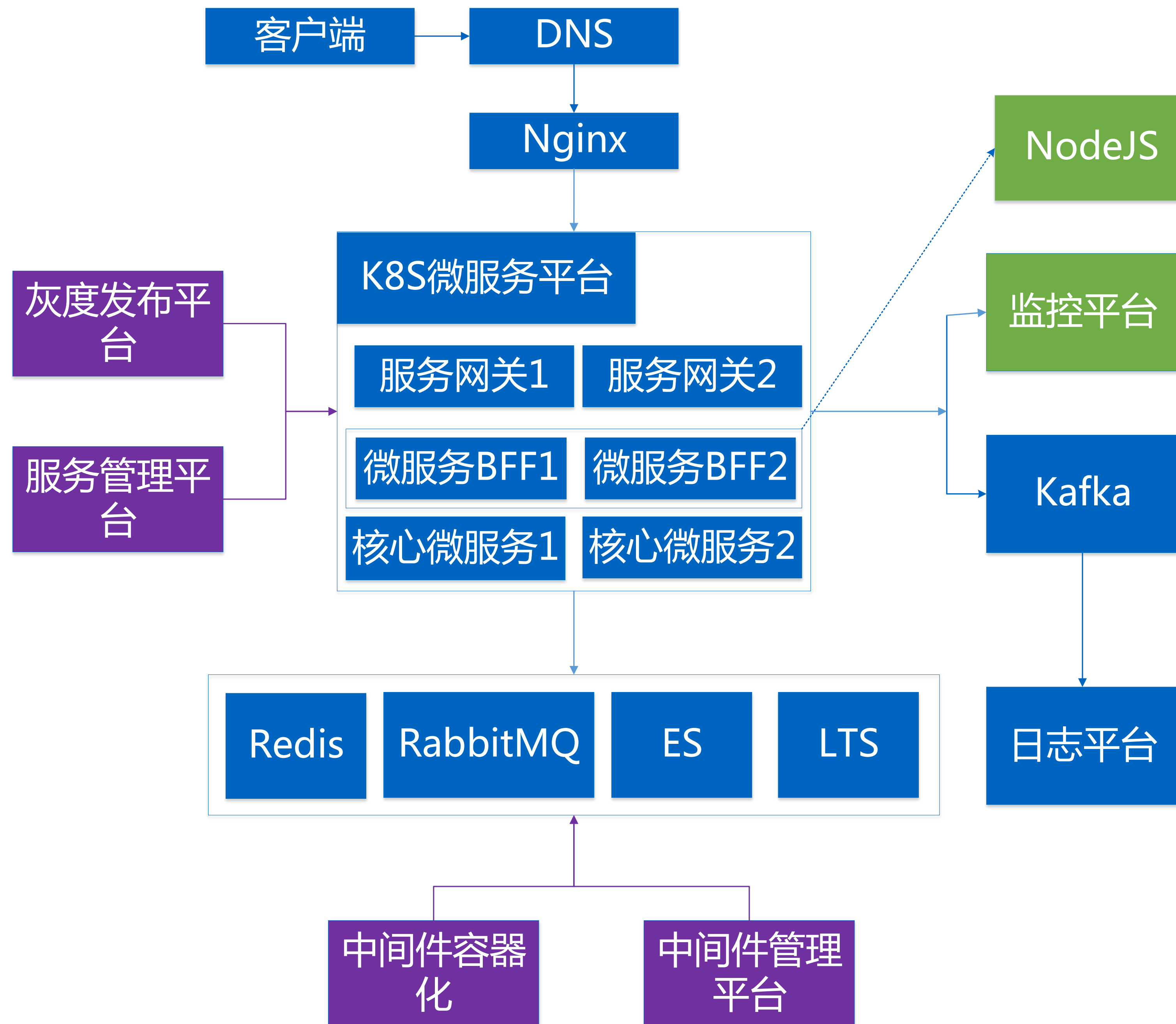


```
[root@GZ-P-K8S-node-1 60045]# ps -eLf | wc -l  
65168
```

```
[root@GZ-P-K8S-node-1 60045]# cat /proc/sys/kernel/pid_max  
65535
```

# 未来规划

# （架构图规划图）





# 极客时间VIP年卡

每天6元, 365天畅看全部技术实战课程

- 20余类硬技能, 培养多岗多能的混合型人才
- 全方位拆解业务实战案例, 快速提升开发效率
- 碎片化时间学习, 不占用大量工作、培训时间





# QCon

## 全球软件开发大会

### 北京·2019

更多技术干货分享，北京站精彩继续  
提前参与，还能享受更多优惠

识别二维码  
查看了解更多

[2019.qconbeijing.com](http://2019.qconbeijing.com)



# THANKS!

## SHANGHAI