

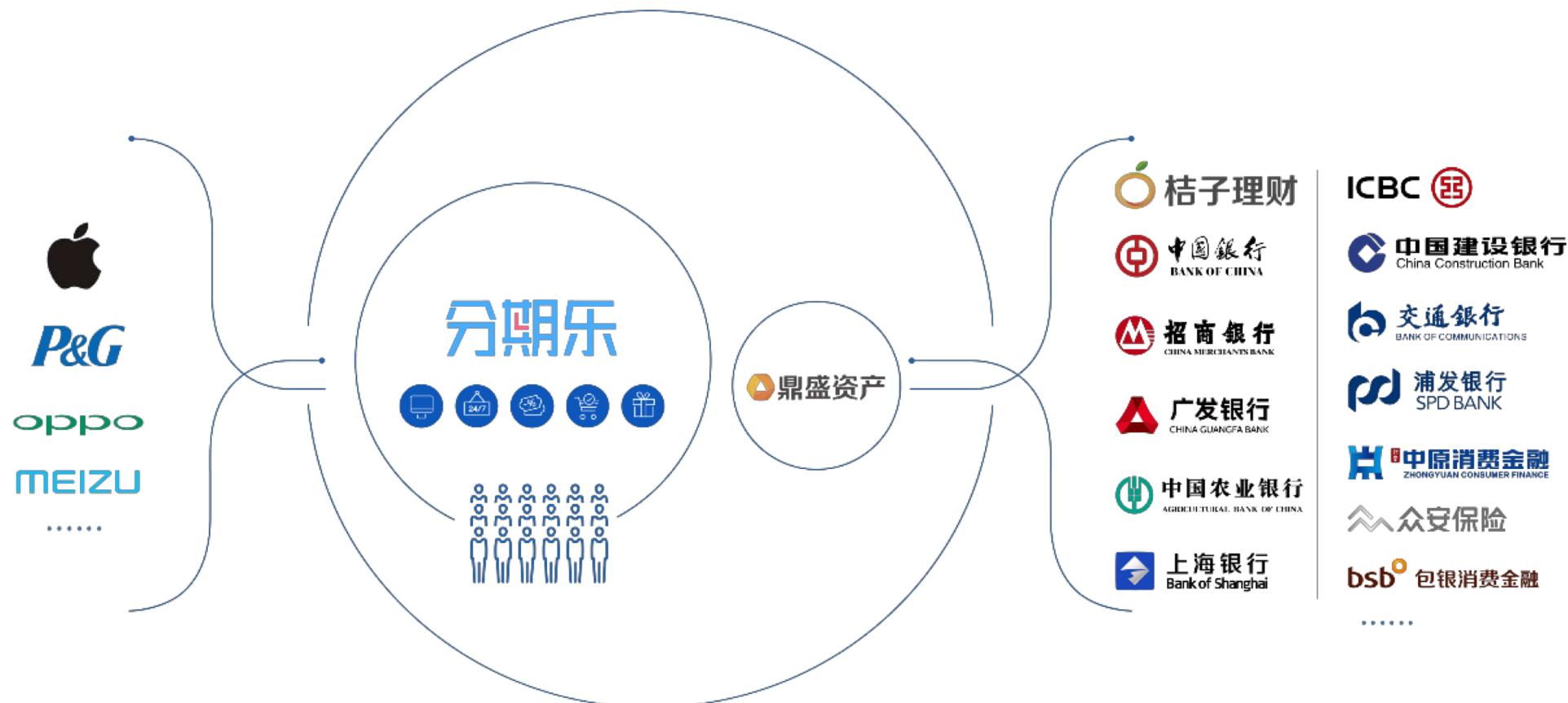


乐信的微服务之路

康彬 (robinkang)

信用成长成就品质生活

LEXIN乐信



LEXIN乐信

架构的组成与目标



高可用性

完善的容错容灾、监控告警、自动化运维

高可扩展性

易于水平伸缩、支持业务快速扩张

低成本

学习使用成本、维护成本、简单易用性

目 录

.....
content

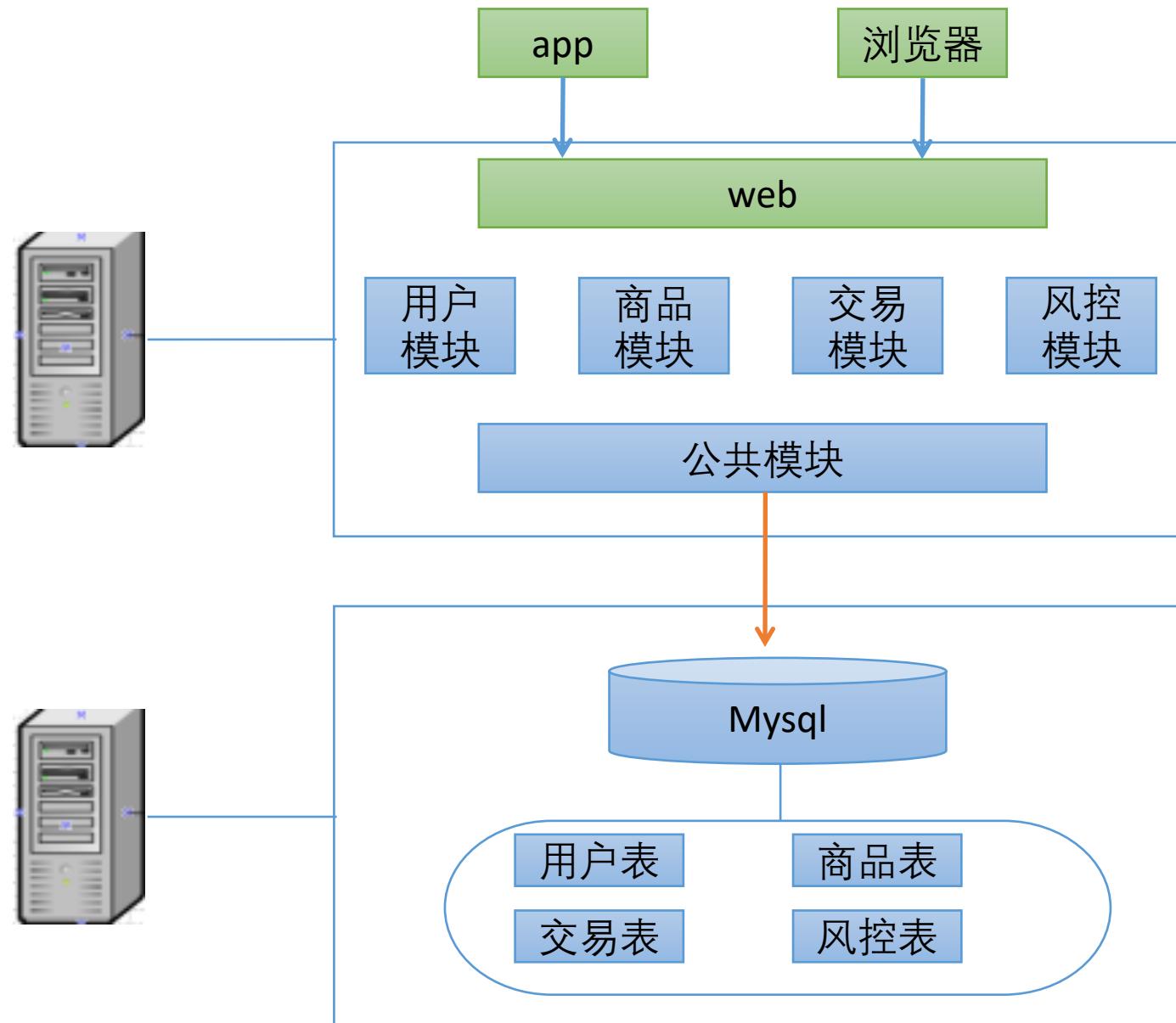
- 01 单体应用架构
- 02 微服务架构 (php+java)
- 03 混合云&同城多活



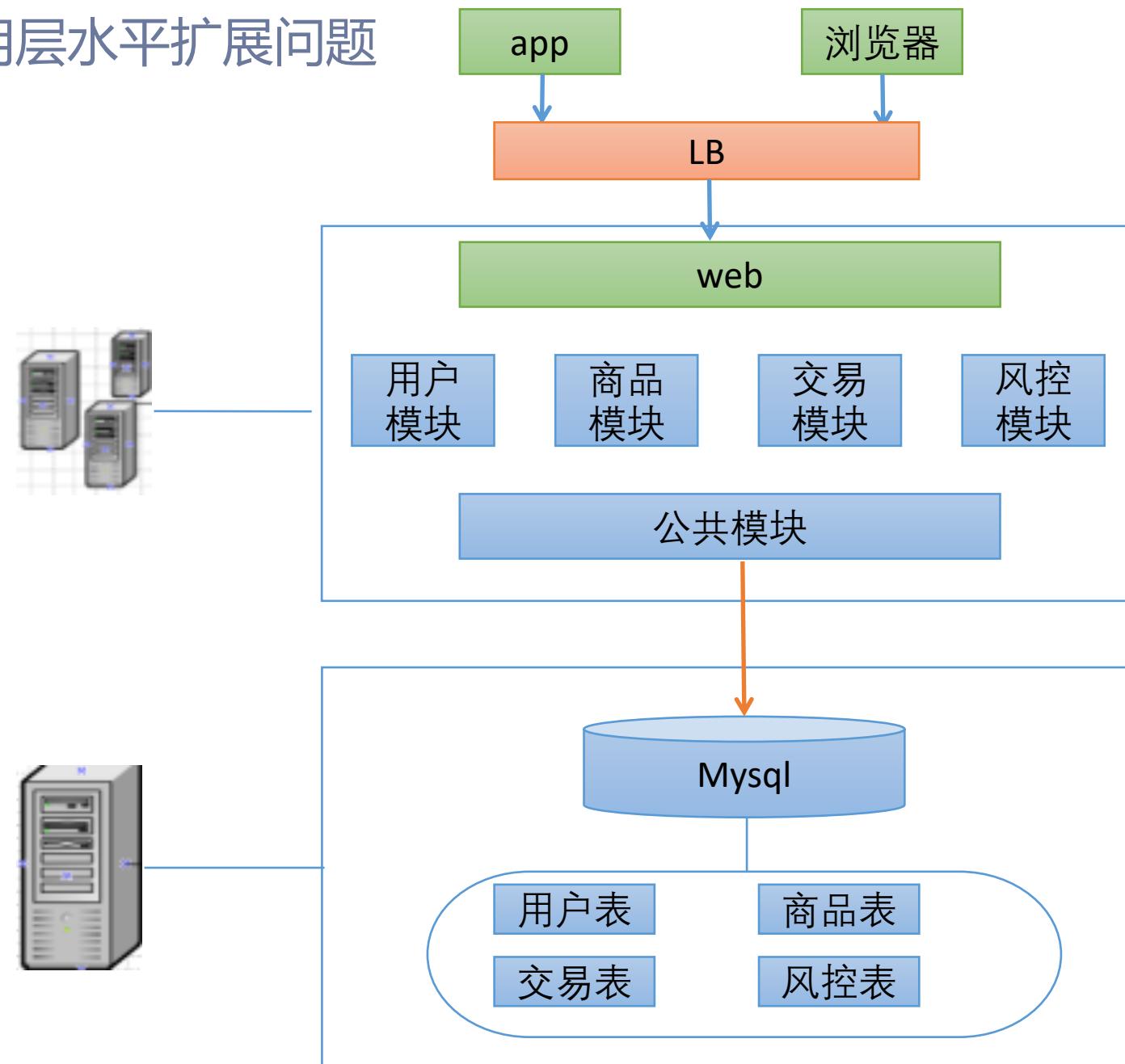
单体应用架构

业务验证初创期

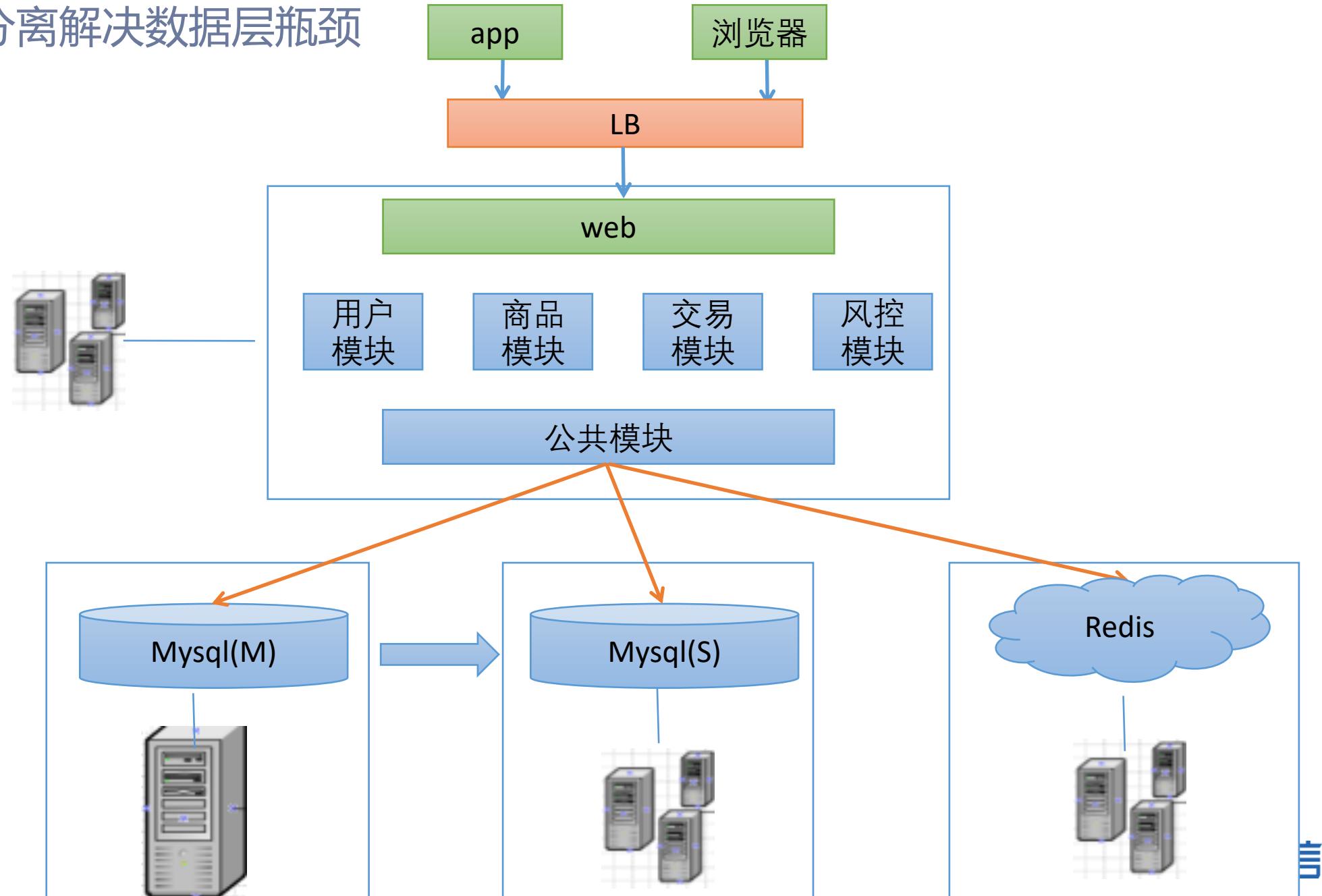
初创期的系统架构



反向代理解决应用层水平扩展问题



缓存+读写分离解决数据层瓶颈



组织规模与系统规模持续增长带来的痛点





微服务化

微服务架构演变

(php+java)

微服务本质



- 01
- 02
- 03

前后端分离

web层与服务层分离解耦

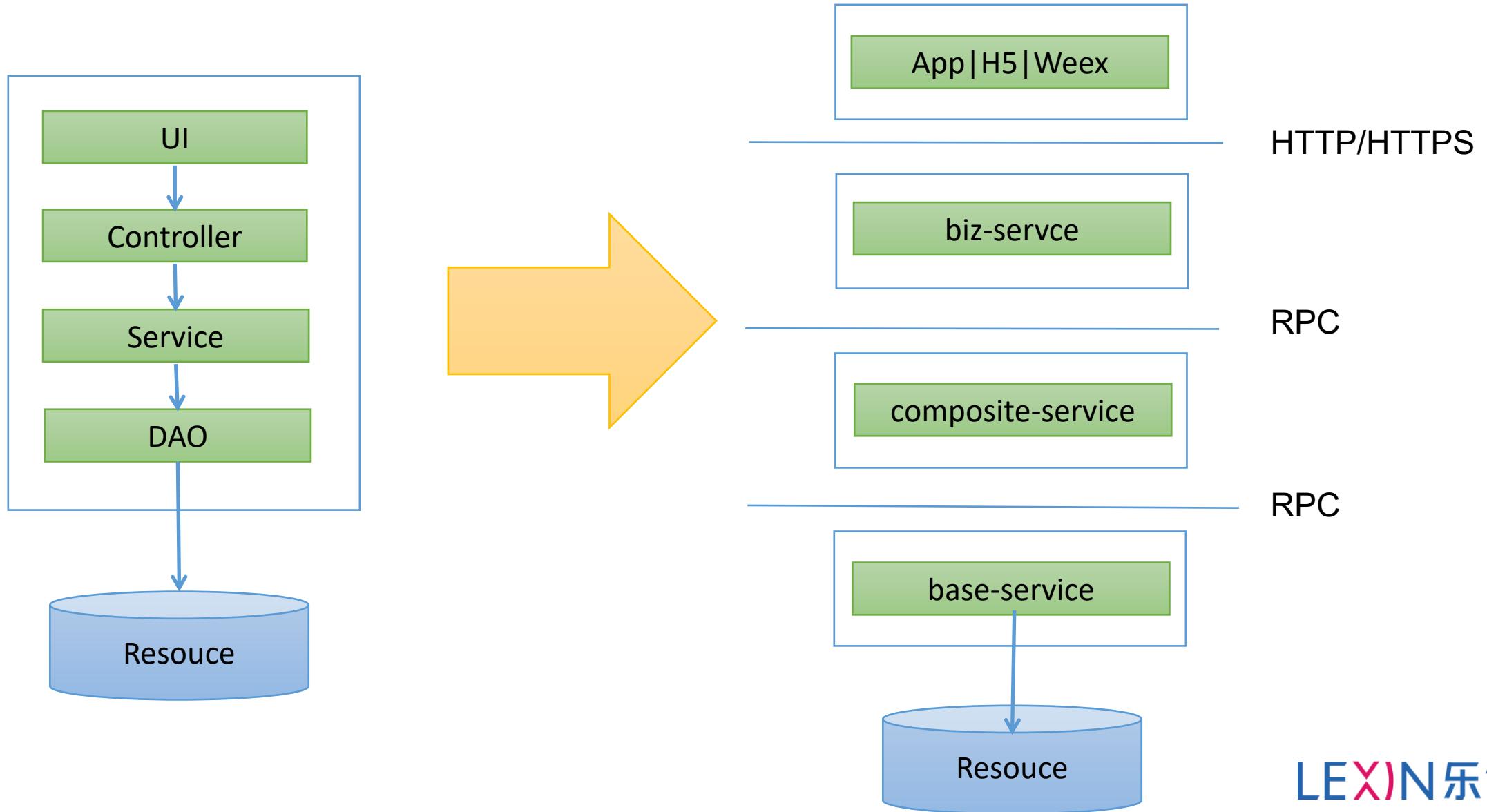
接口与实现分离

服务消费方与服务提供方分离解耦

应用和数据分离

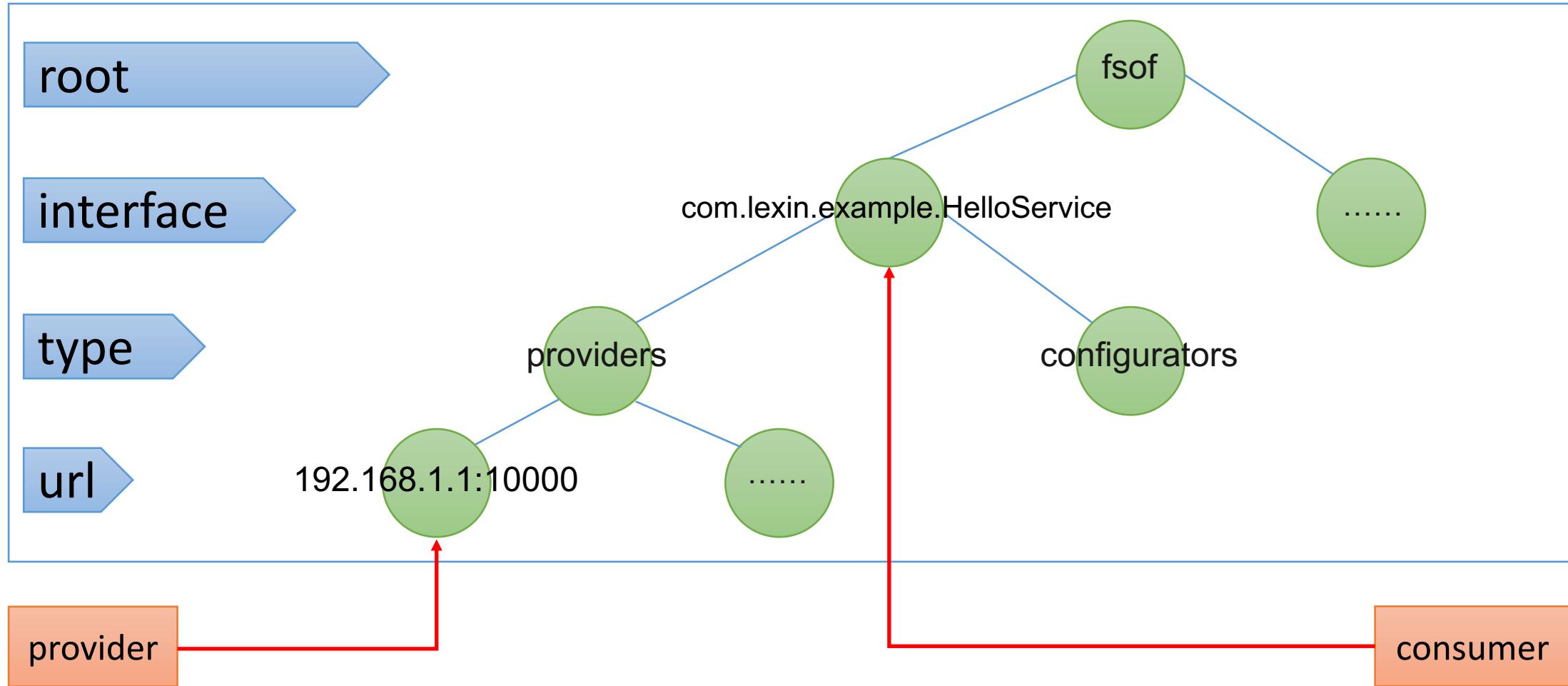
数据使用方与数据资源分离解耦，
由数据基础服务提供统一入口

php微服务拆分方式：先水平再垂直



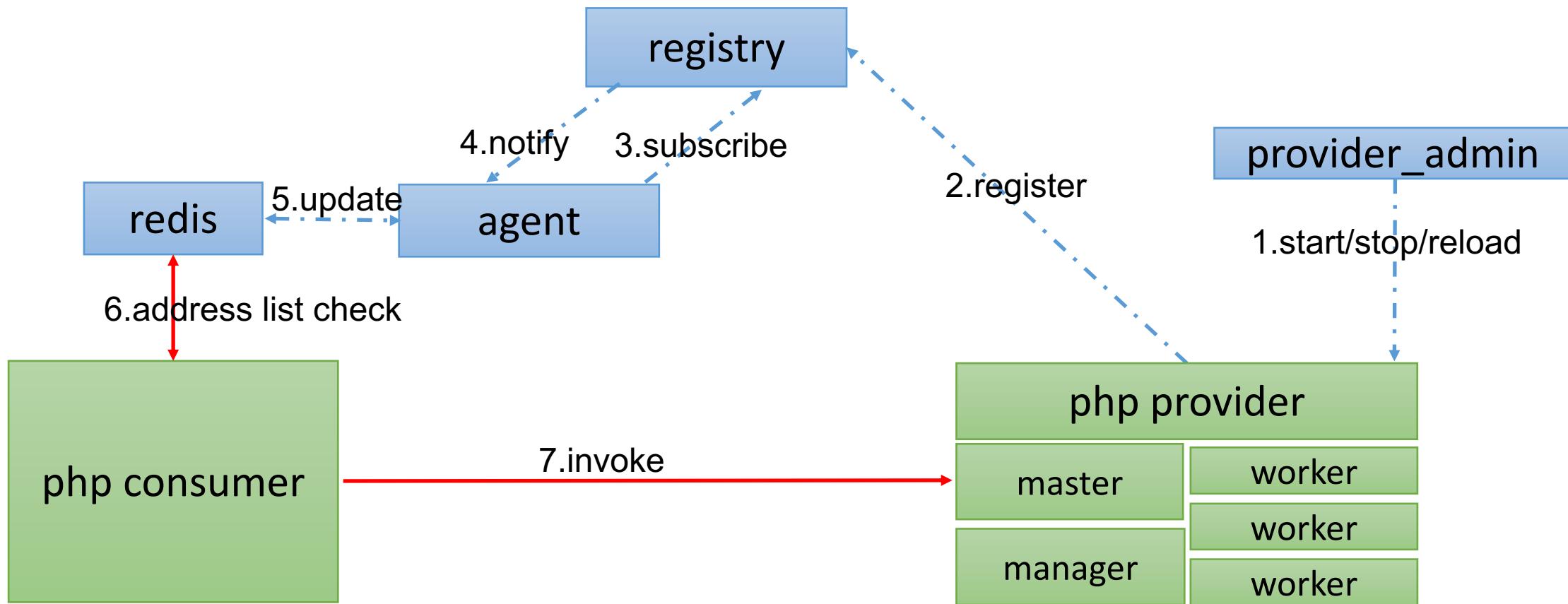
服务治理的粒度

应用 VS 接口

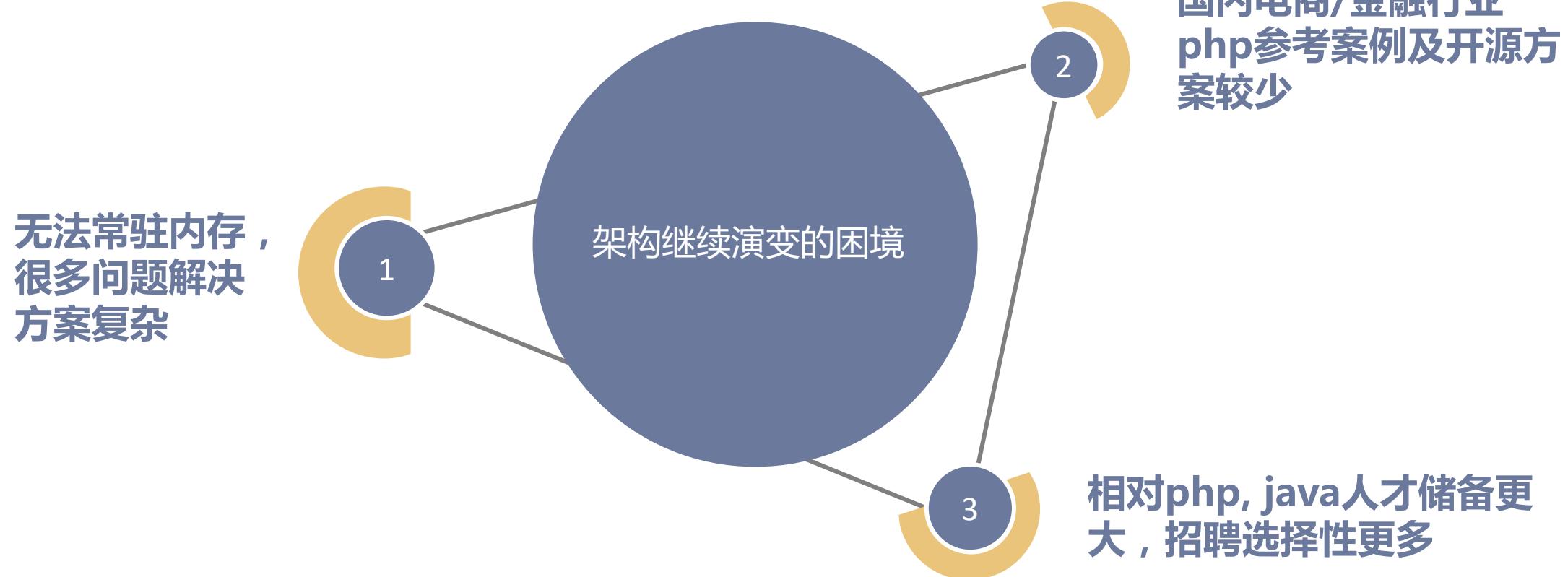


自研LSF-PHP微服务框架

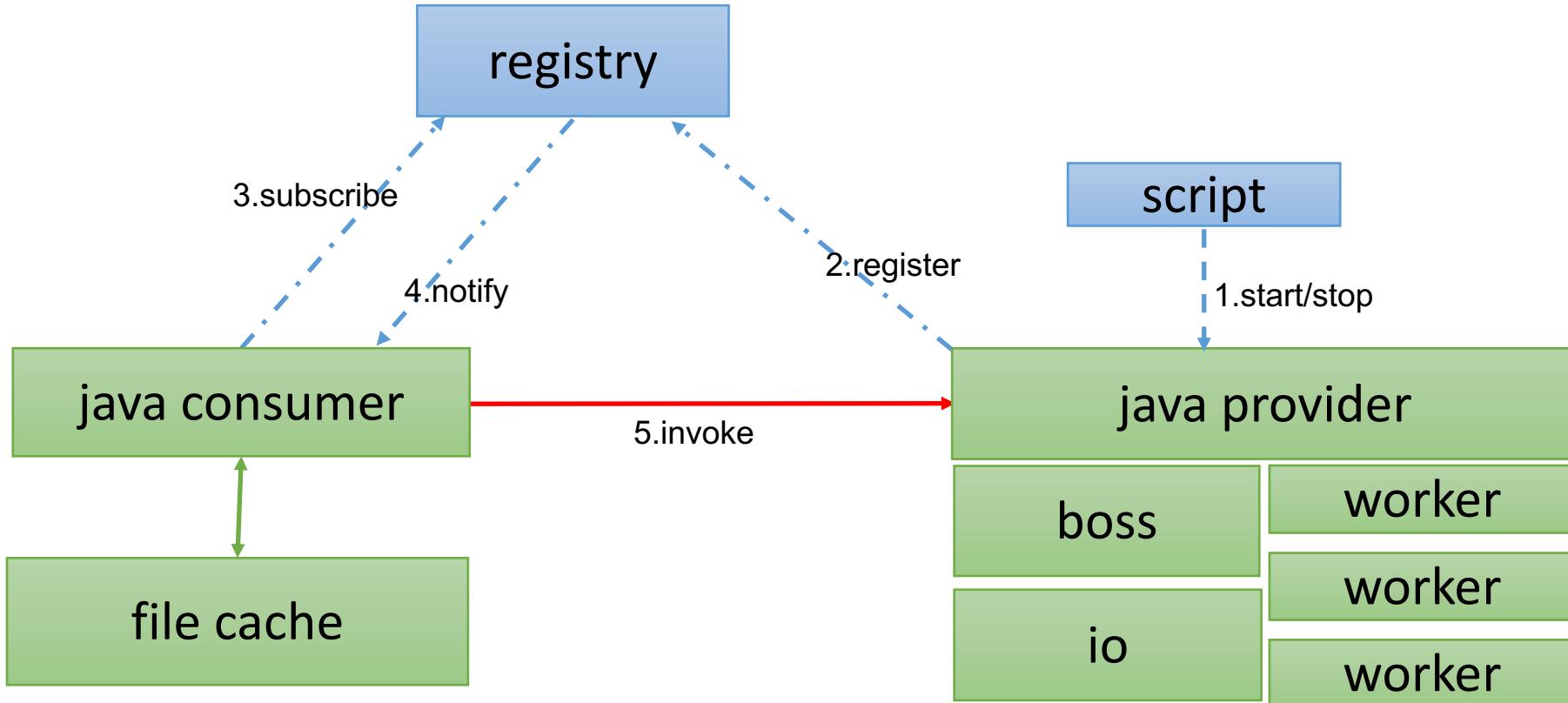
service性能提升十几倍



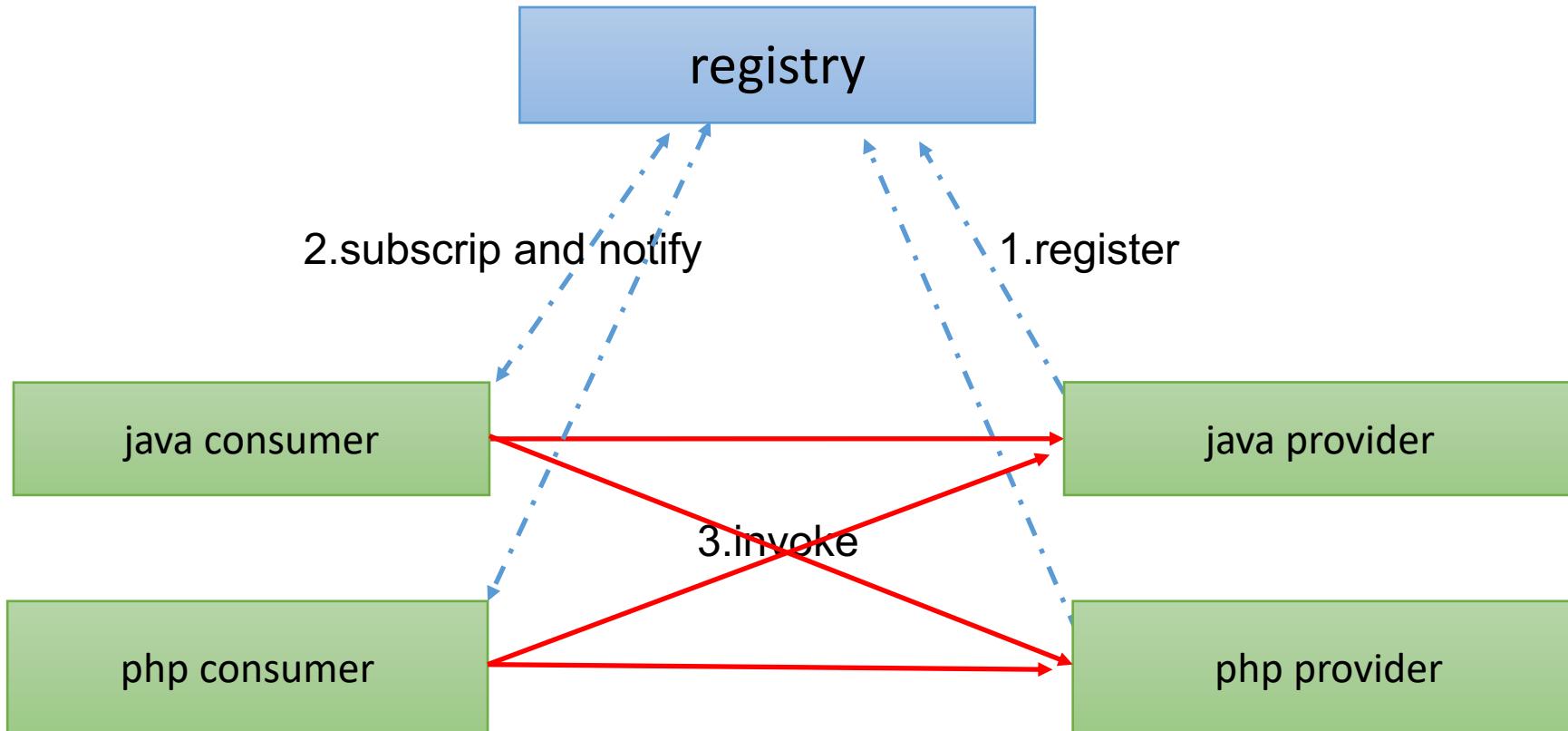
php技术栈的困境



融合dubbo,引入java技术栈



php&java微服务混用



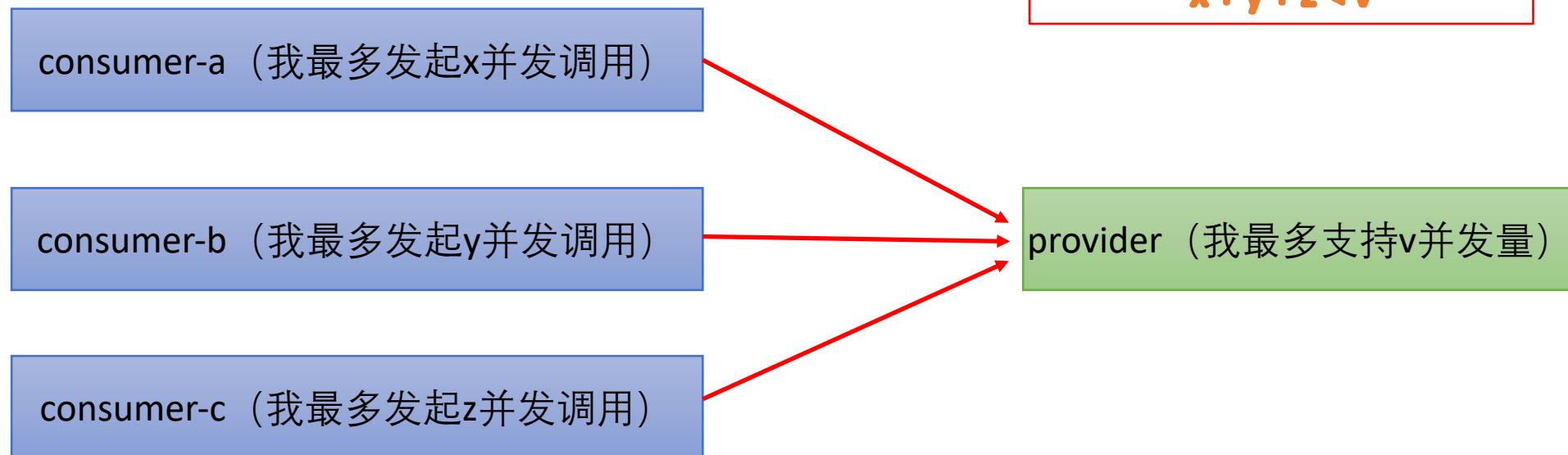
1.php和java之间以泛型方式互调

2.由于php是弱类型语言，且不支持方法重载，所以java provider
接口中禁止使用方法重载

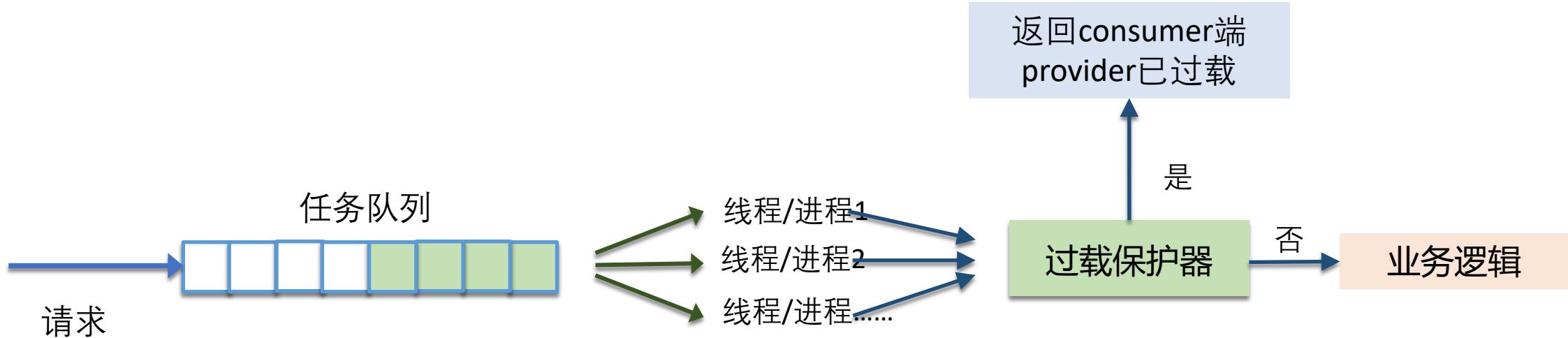
记一次系统雪崩



遵循SLA约定

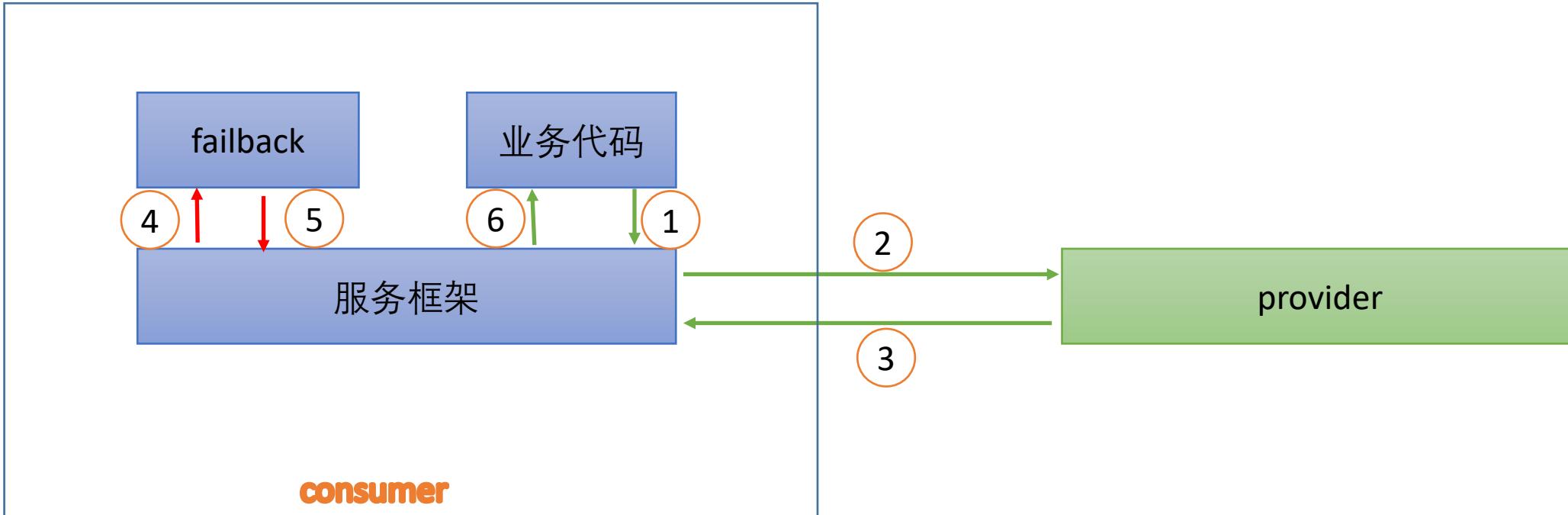


provider自我保护-过载保护



- 1. 队列等待超时策略**：请求在队列中等待时间超过设置值（默认1 s）
- 2. 丢包策略**：策略1连续触发x次，进入丢包模式，连续丢包y个恢复正常（x默认值5，y默认值20）

斩断故障传递链条-consumer的熔断与服务降级

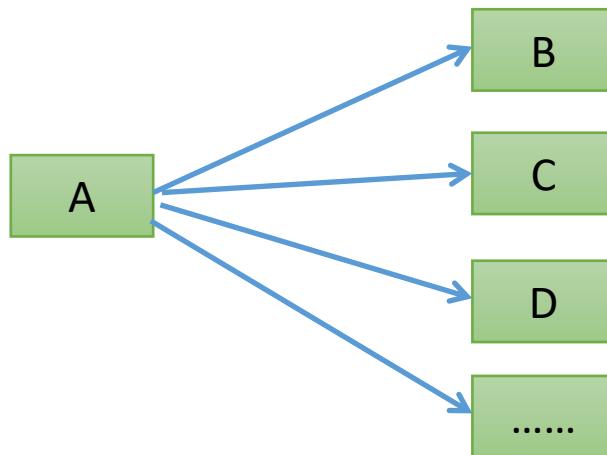


正常流程: 1-2-3-6
熔断流程: 1-2-3-4-5-6

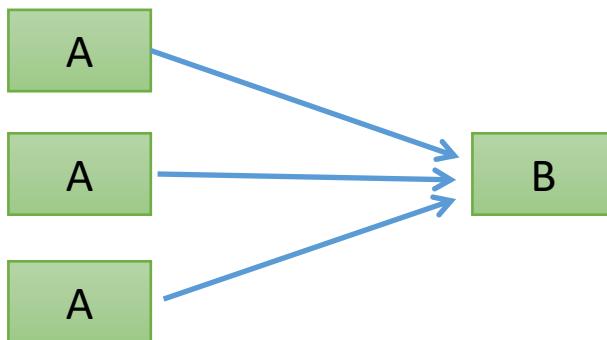
业务系统继续发展带来的新问题



系统上下游强
耦合

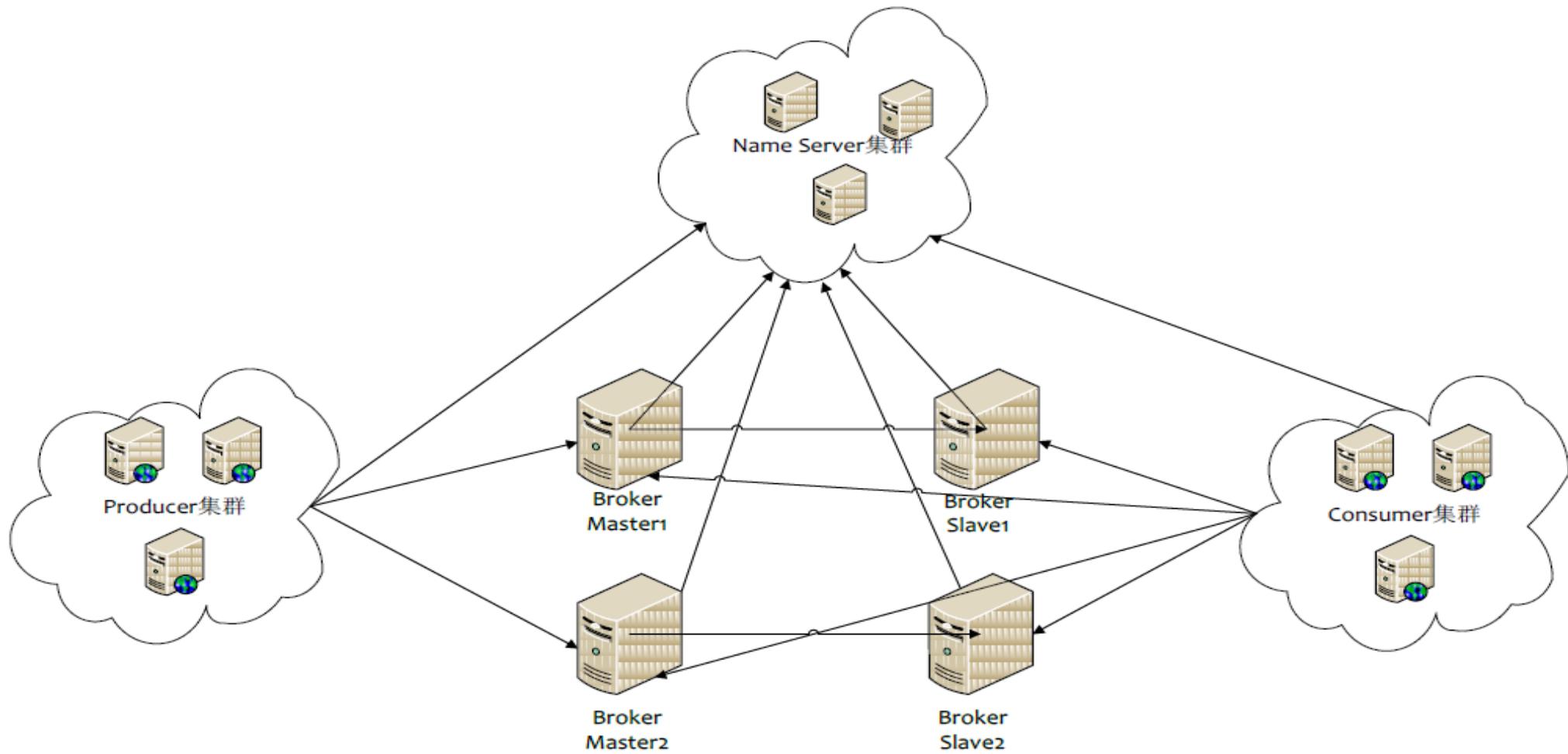


广播通知



上下游处理能
力不对等

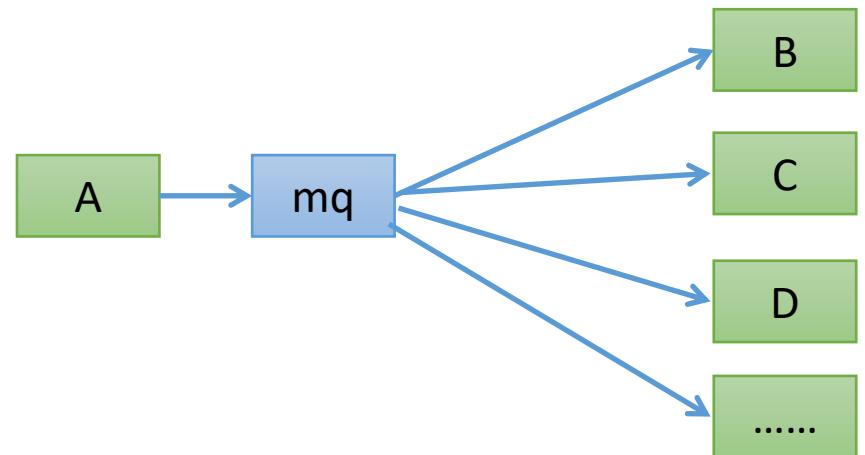
引入rocketmq



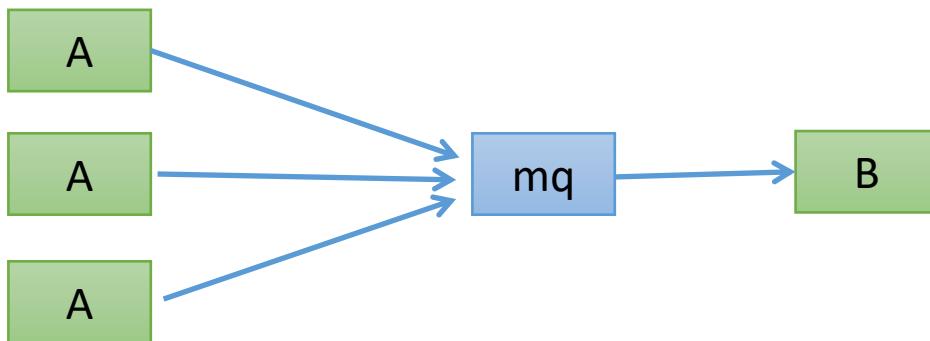
通过mq来解耦、削峰



系统解耦

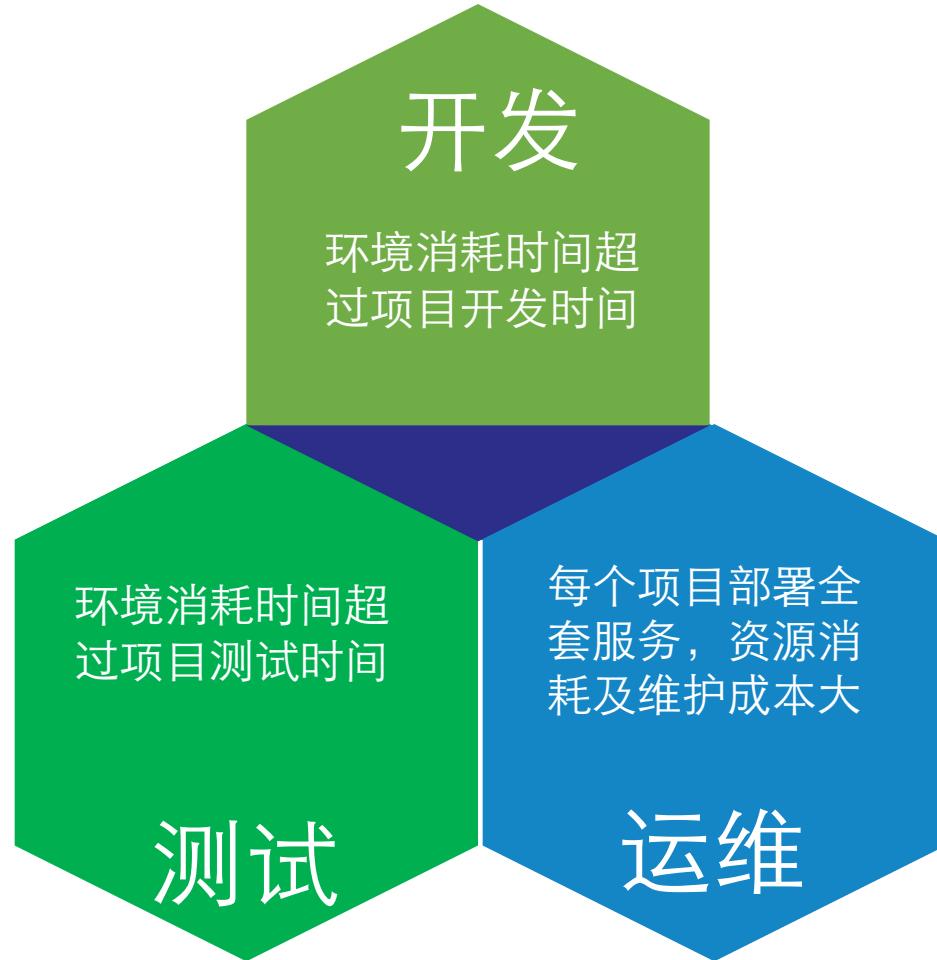


业务解耦

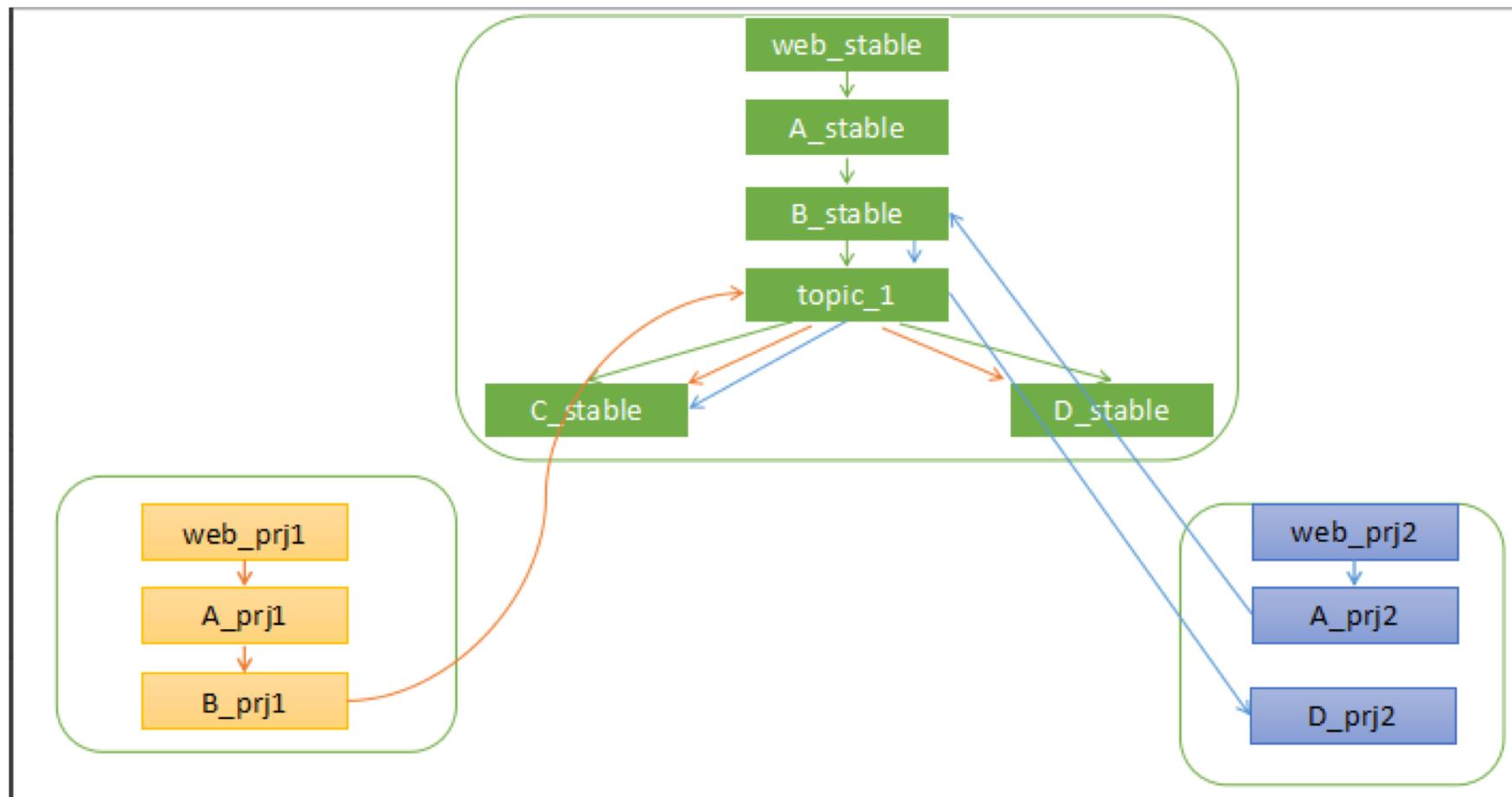


削峰

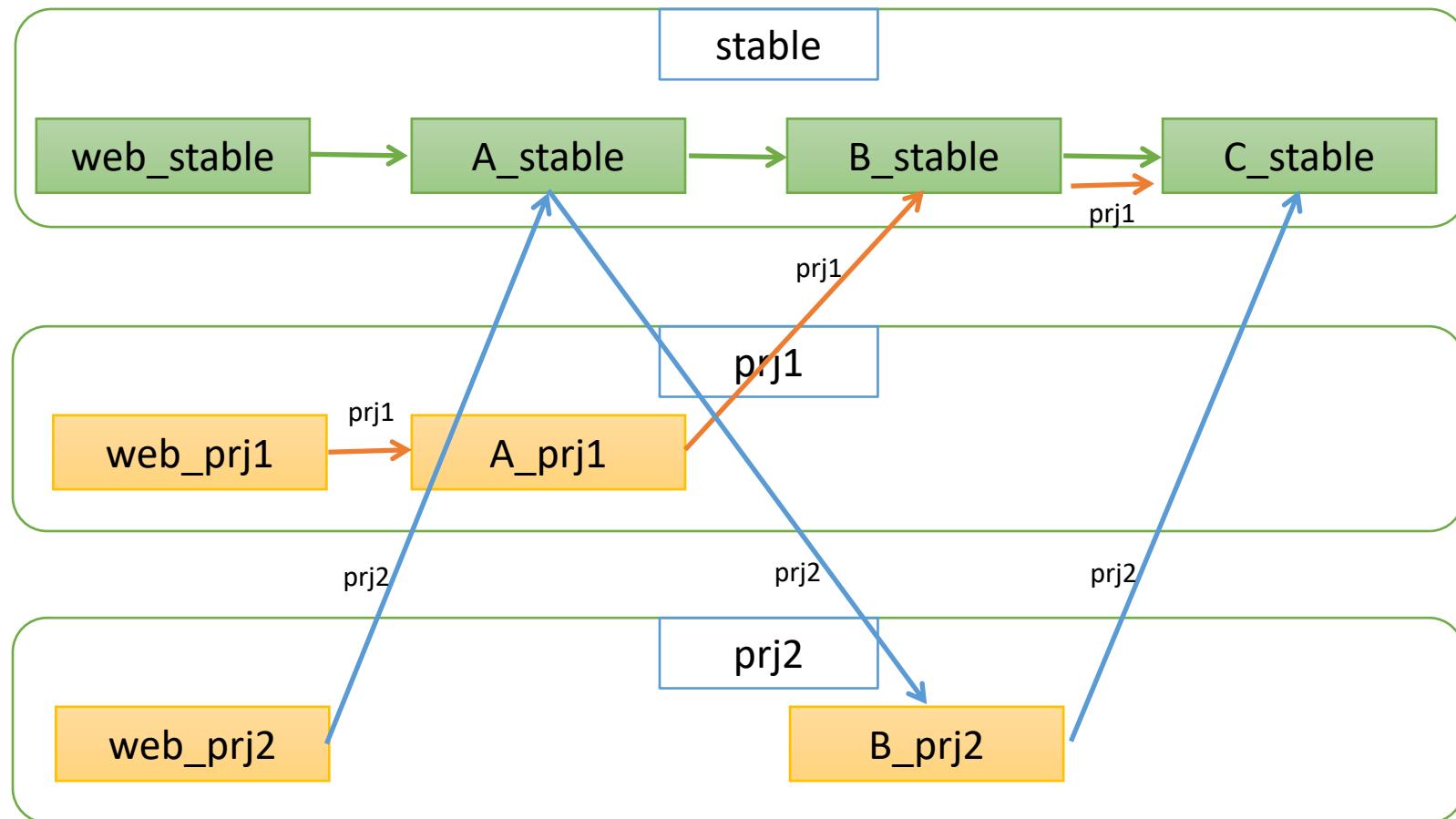
环境问题爆发



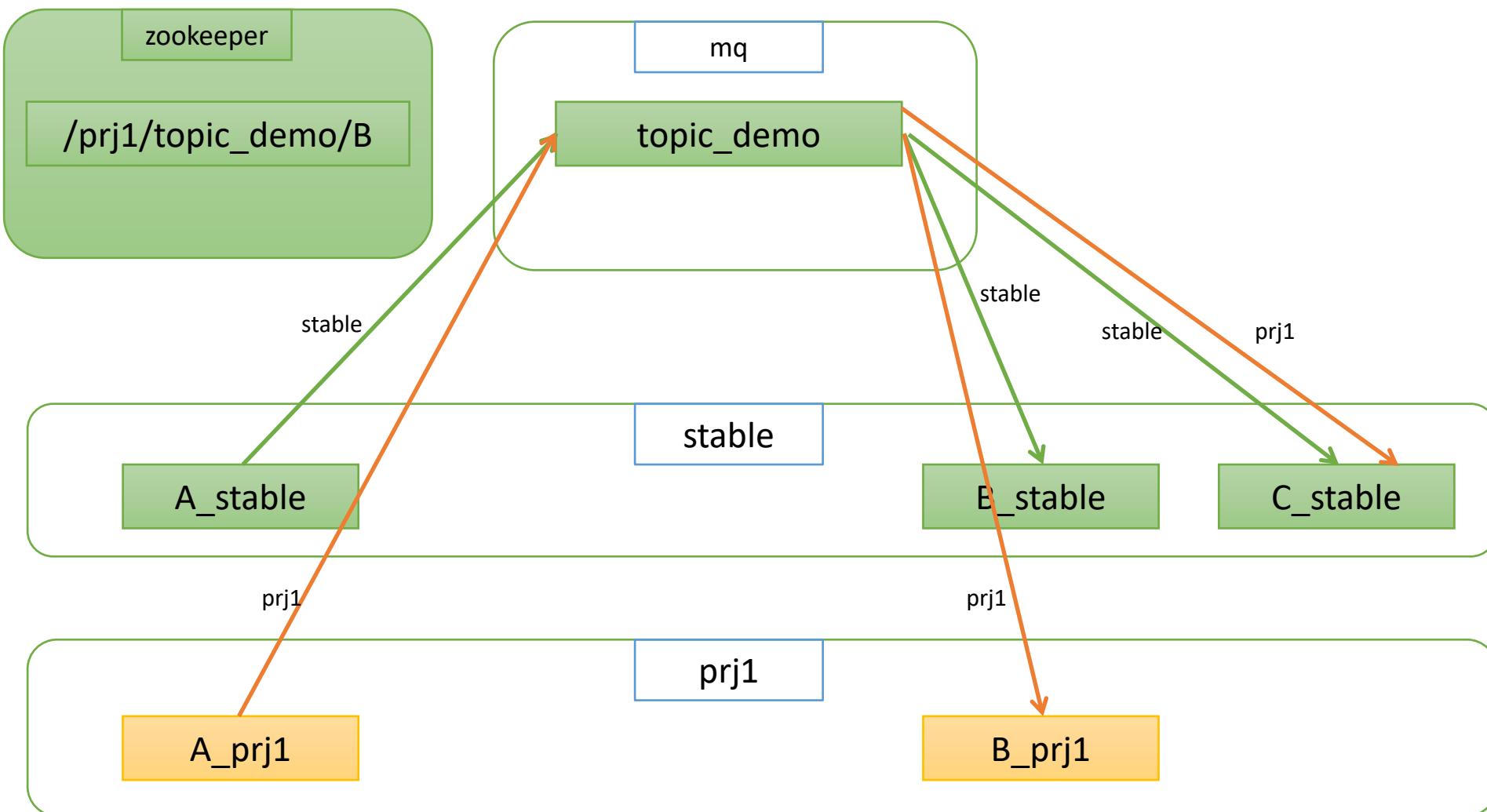
stable环境方案



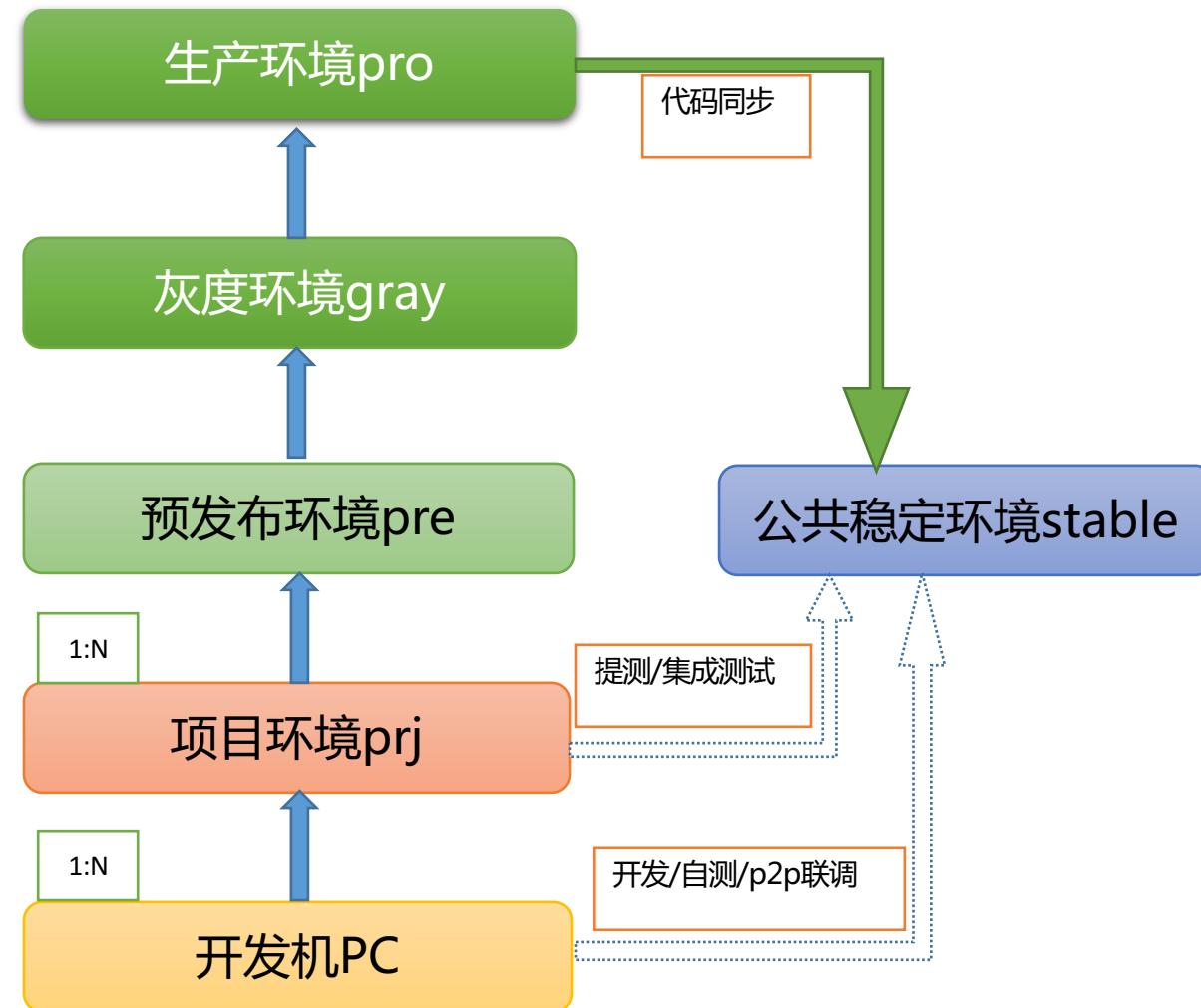
服务化框架环境隔离方案



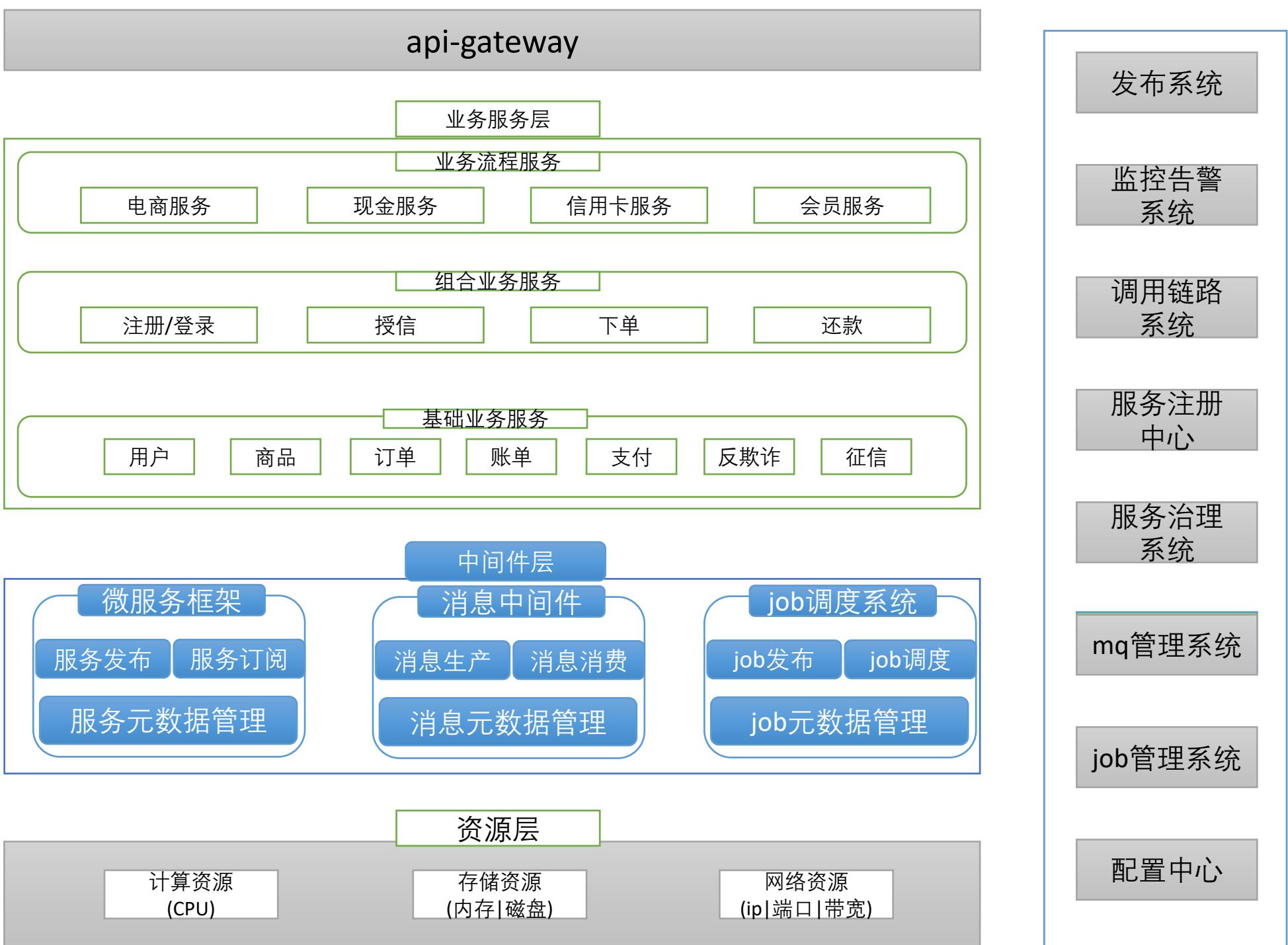
mq环境隔离方案



基于stable方案的研发流程



微服务架构体系全景图





混合云&同城多活

新一代技术架构探索

大促的烦恼



机器准备周期长，
紧急情况无法应对

大促后机器闲置率高，
资源浪费巨大



上云的约束条件

01

服务上云，数据
不上云

02

流量大的只读服
务上云，写服务
留在自建idc

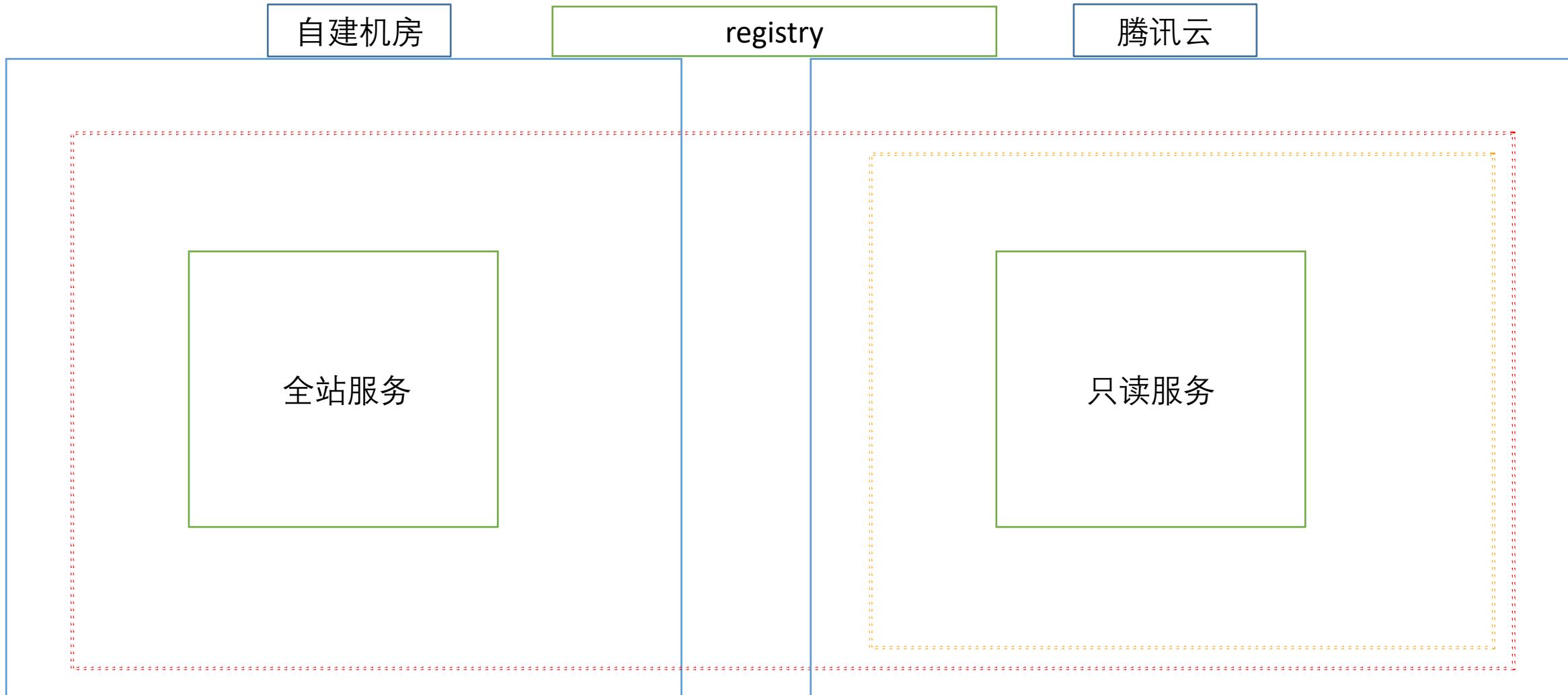
03

接入层支持快速
的uri粒度流量调
度

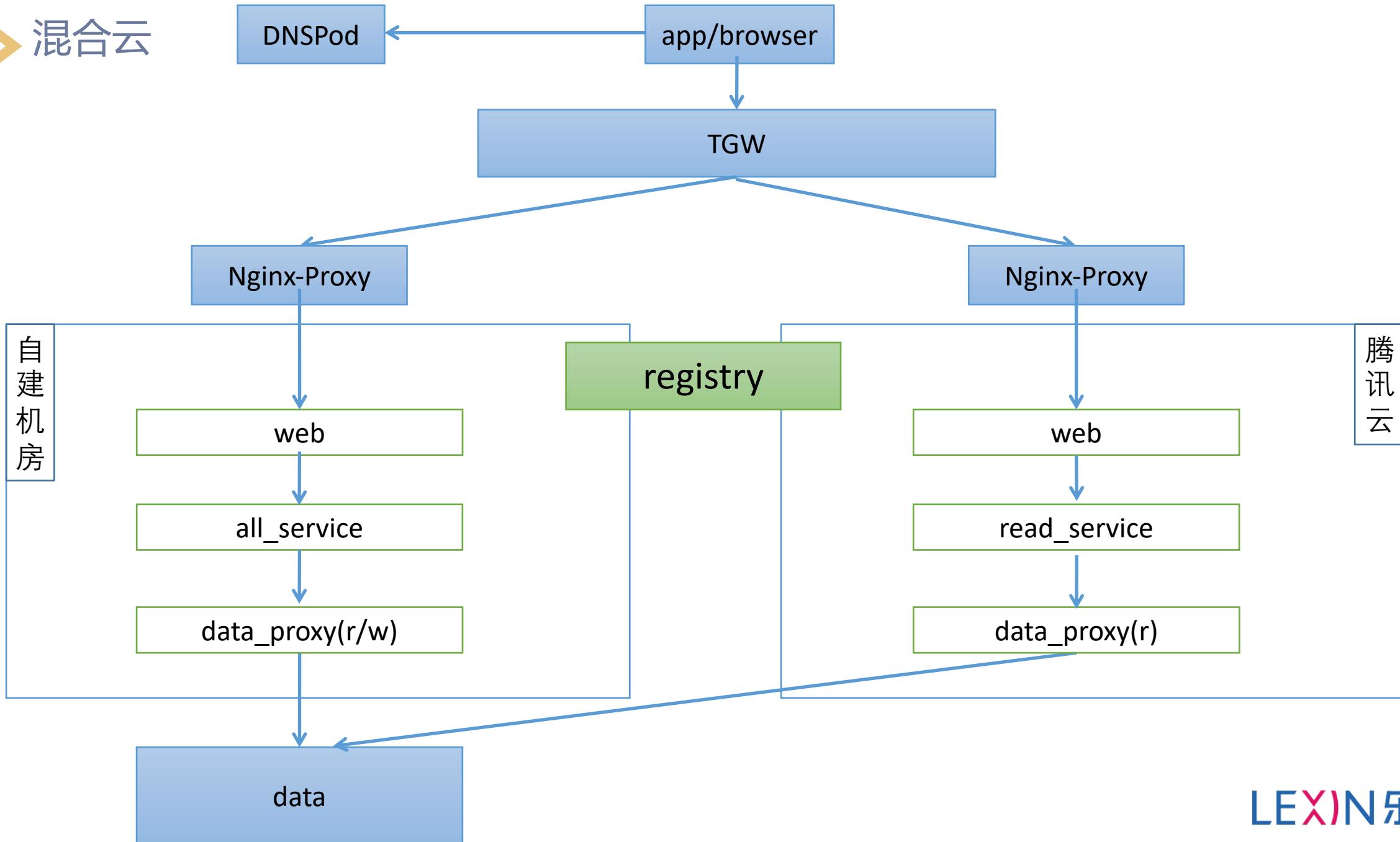
04

服务层优先保证
同一机房内服务
调用，减少跨机
房调用延迟影响

服务框架set路由示意图



混合云



未来的担忧

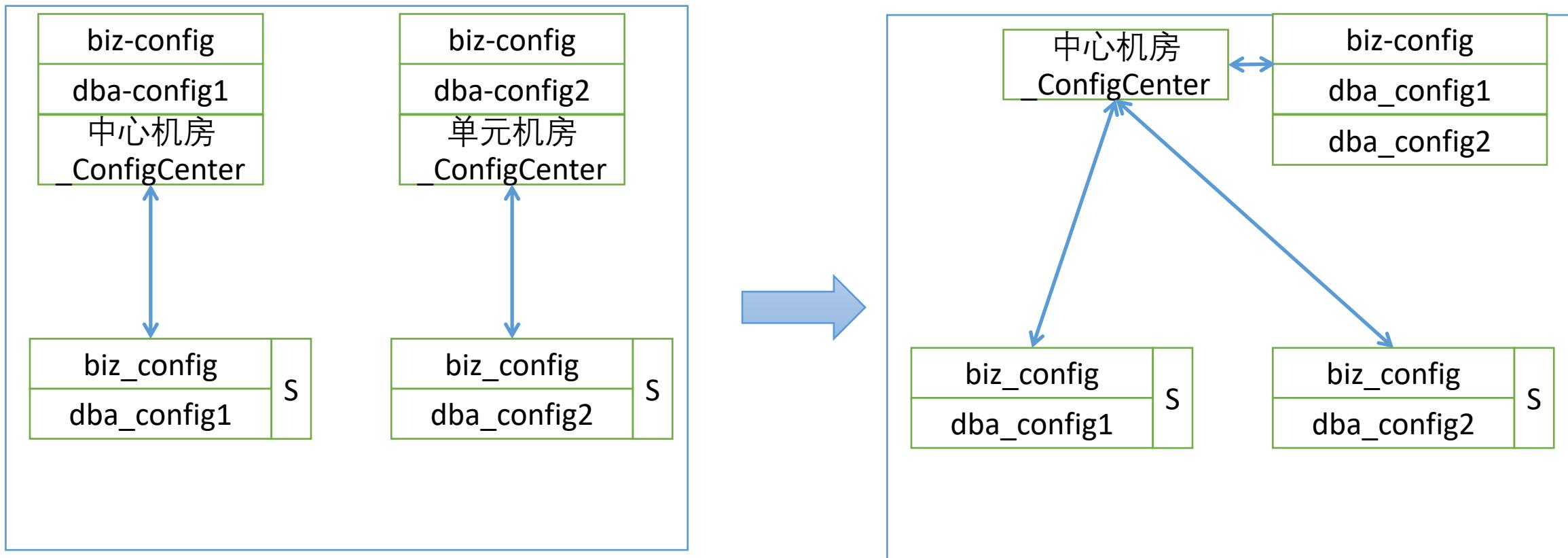


如果发生机房级的
灾难怎么办？

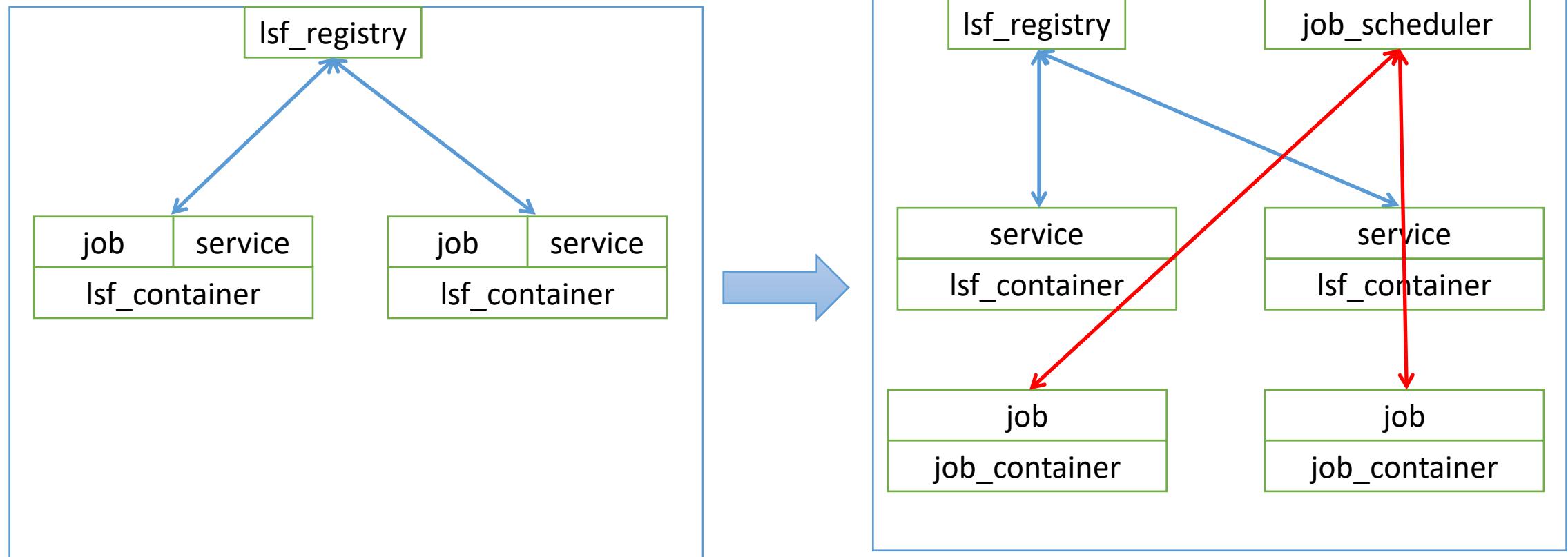


随着流量越来越大，
集群规模越来越大，
怎么破？

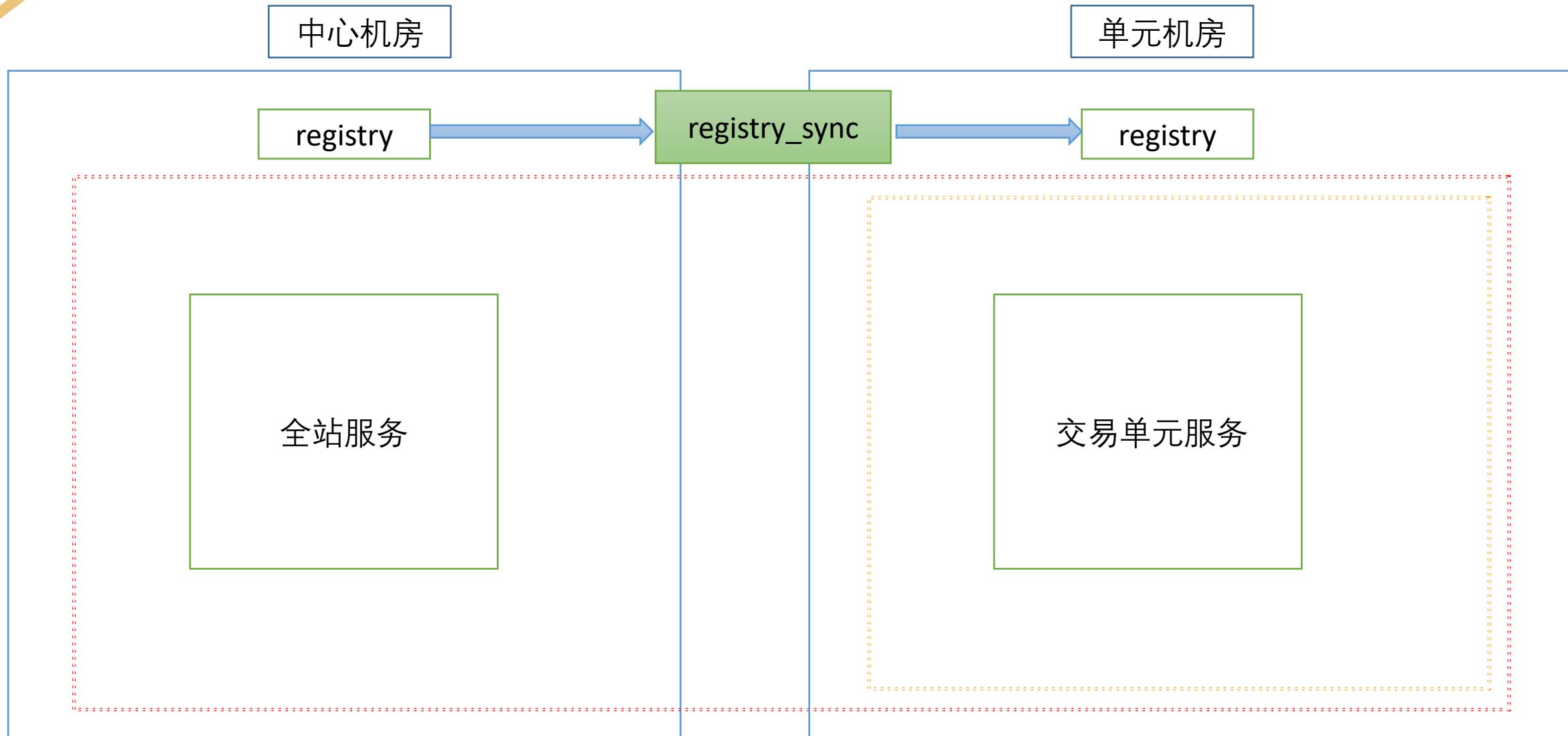
跨机房的配置中心



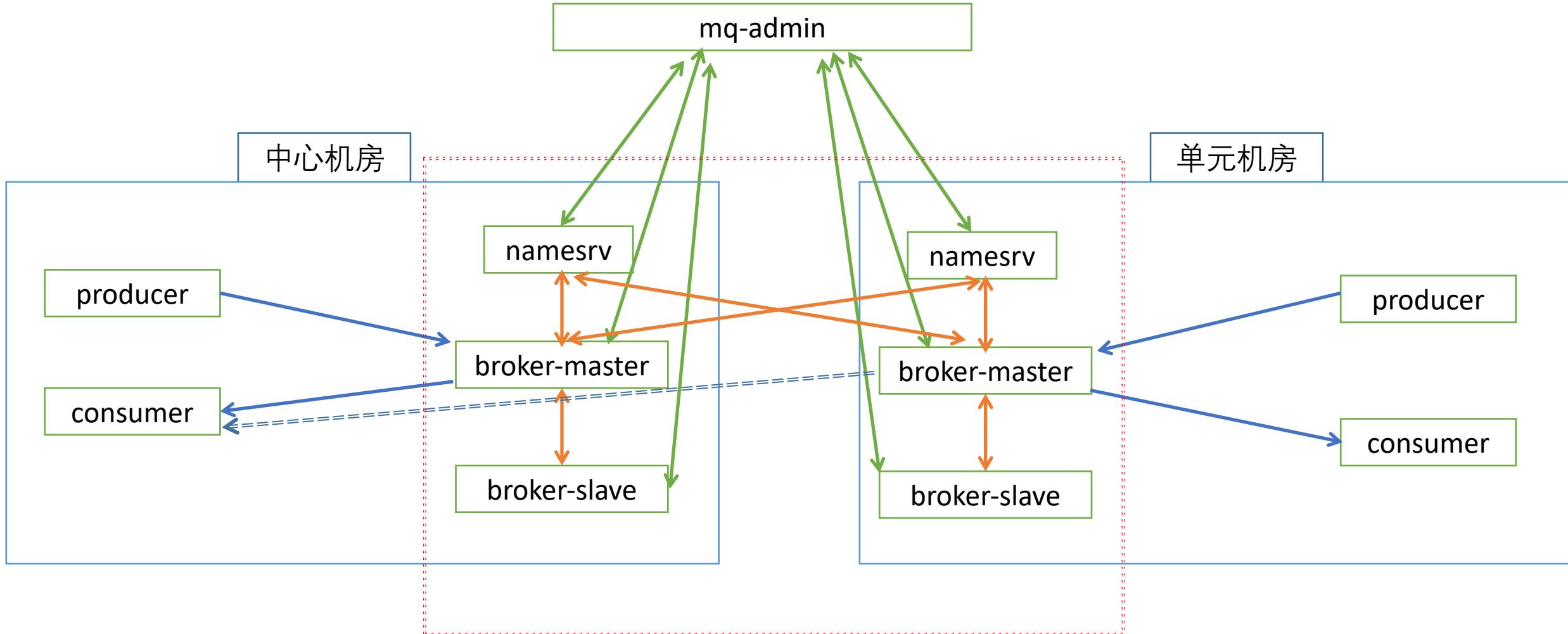
java的job和服务分离



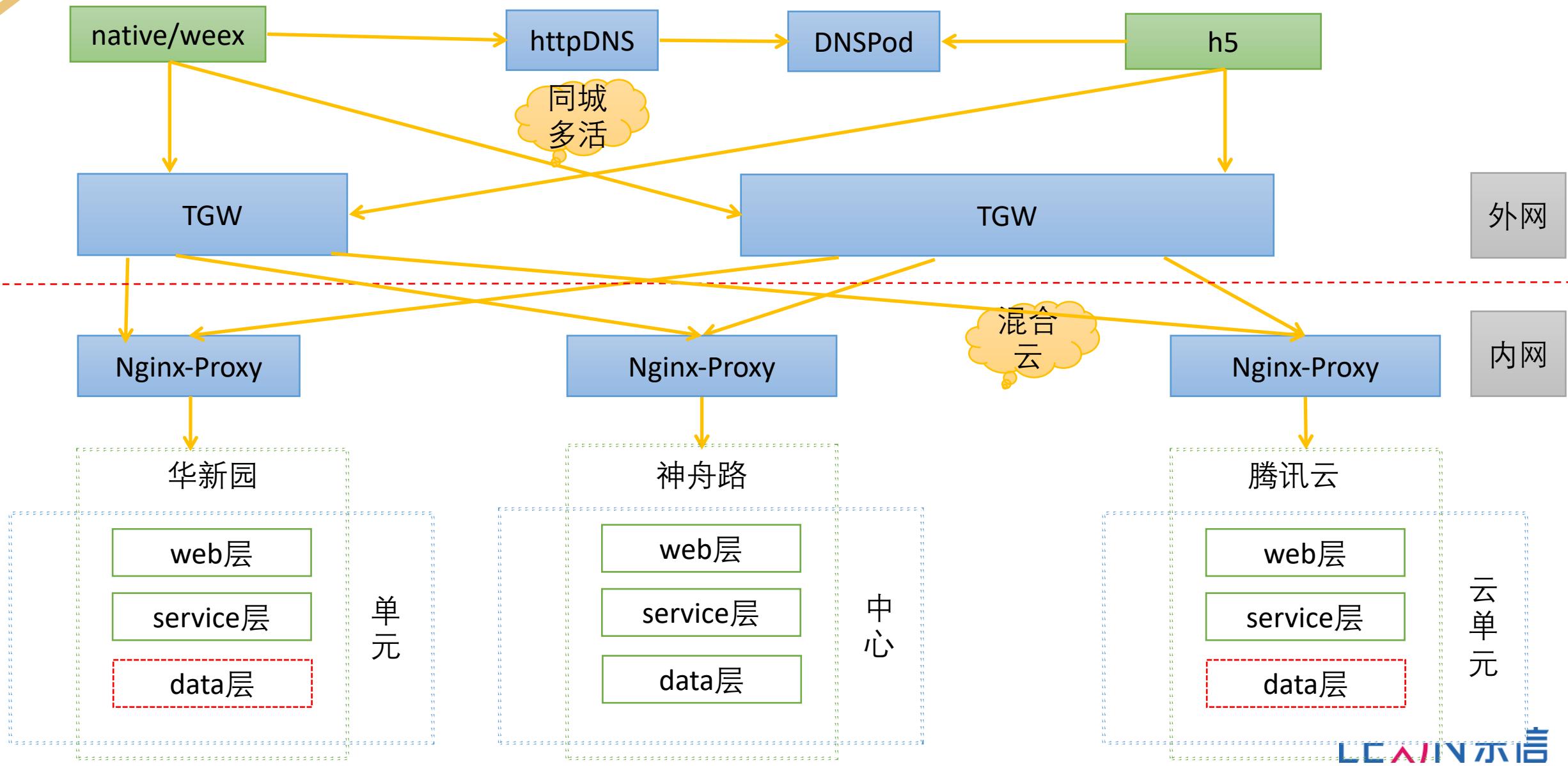
服务化框架单元化



mq的单元化



混合云&同城多活





经验和教训



架构是演变而来的，不是设计出来的



没有最好的架构，只有最合适的架构



慎用临时方案规避技术债，否则将来会付出更大的代价来还债



一开始就考虑静默升级问题，否则将来必将成为最大的绊脚石

[DUBBO]



LEXIN乐信