

贝聊微服务架构实战历程

杨钦民
贝聊技术顾问

杨钦民

贝聊技术顾问

负责贝聊研发体系建设、团队建设、人才培养

构建贝聊微服务架构体系、平台架构体系、自动化运维体系



1 业务背景介绍

2 诞生期-技术架构选型

3 成长期-技术架构重构

4 爆发期-微服务架构

1 业务背景介绍

关于贝聊

- 贝聊成立于2013年，是中国幼儿园家长工作平台
- 致力于通过互联网产品及定制化解决方案，帮助幼儿园解决展示、通知、沟通等家长工作中的痛点，促进家园关系和谐

② 诞生期-技术架构选型

技术架构选型考虑点



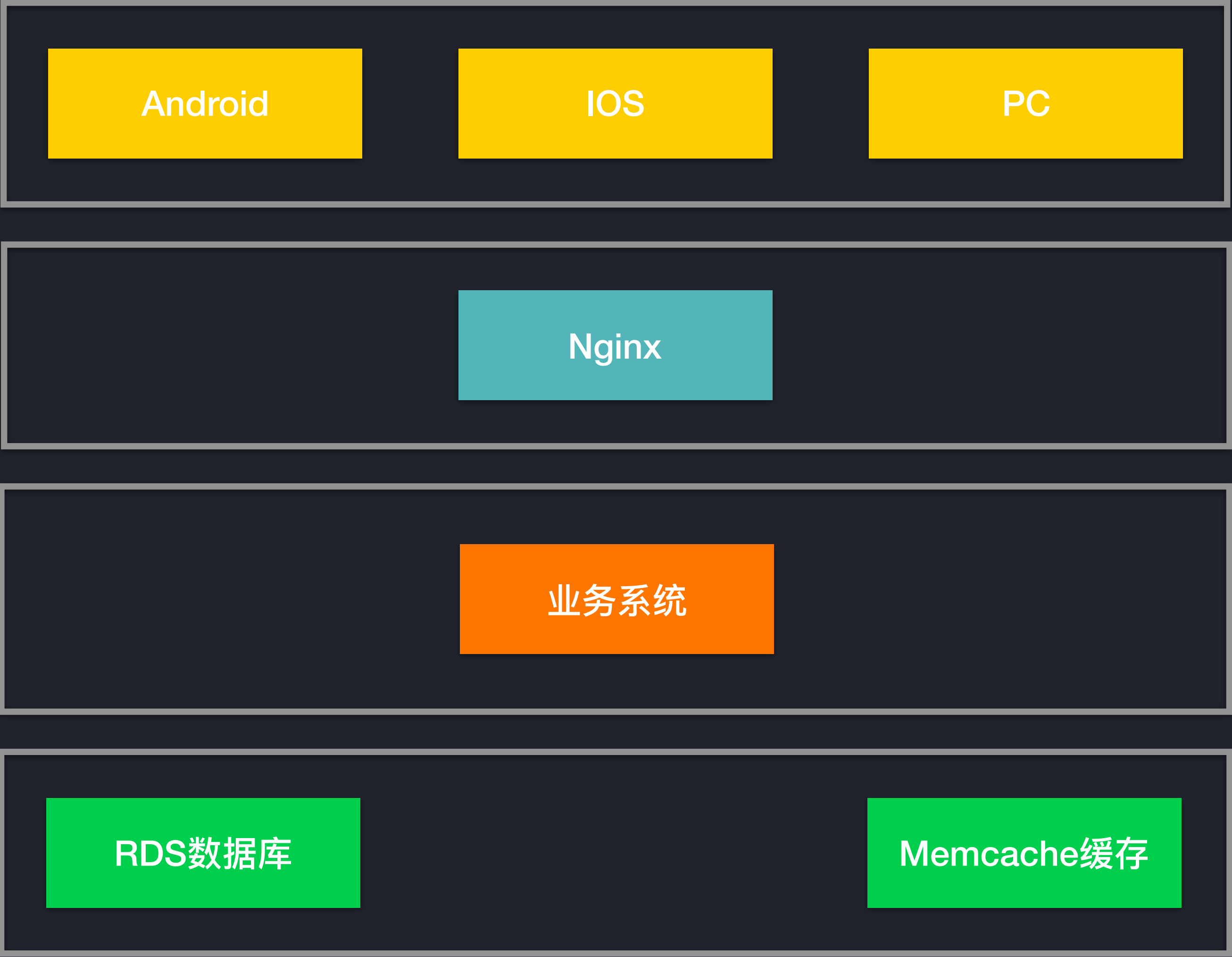
技术架构

3台ECS服务器

Nginx与系统部署在1台机器

RDS数据库1台机器

Memcache1台机器



用户层

接入层

系统层

存储层

技术架构特点



单体架构

快速研发

低成本

- 单体架构，架构简单，清晰的分层结构
- 可以快速研发，满足产品快速迭代需求
- 没有复杂的技术，技术学习成本低，同时运维成本低，无需专业的运维，节省开支

3 成长期-技术架构重构

技术架构重构契机

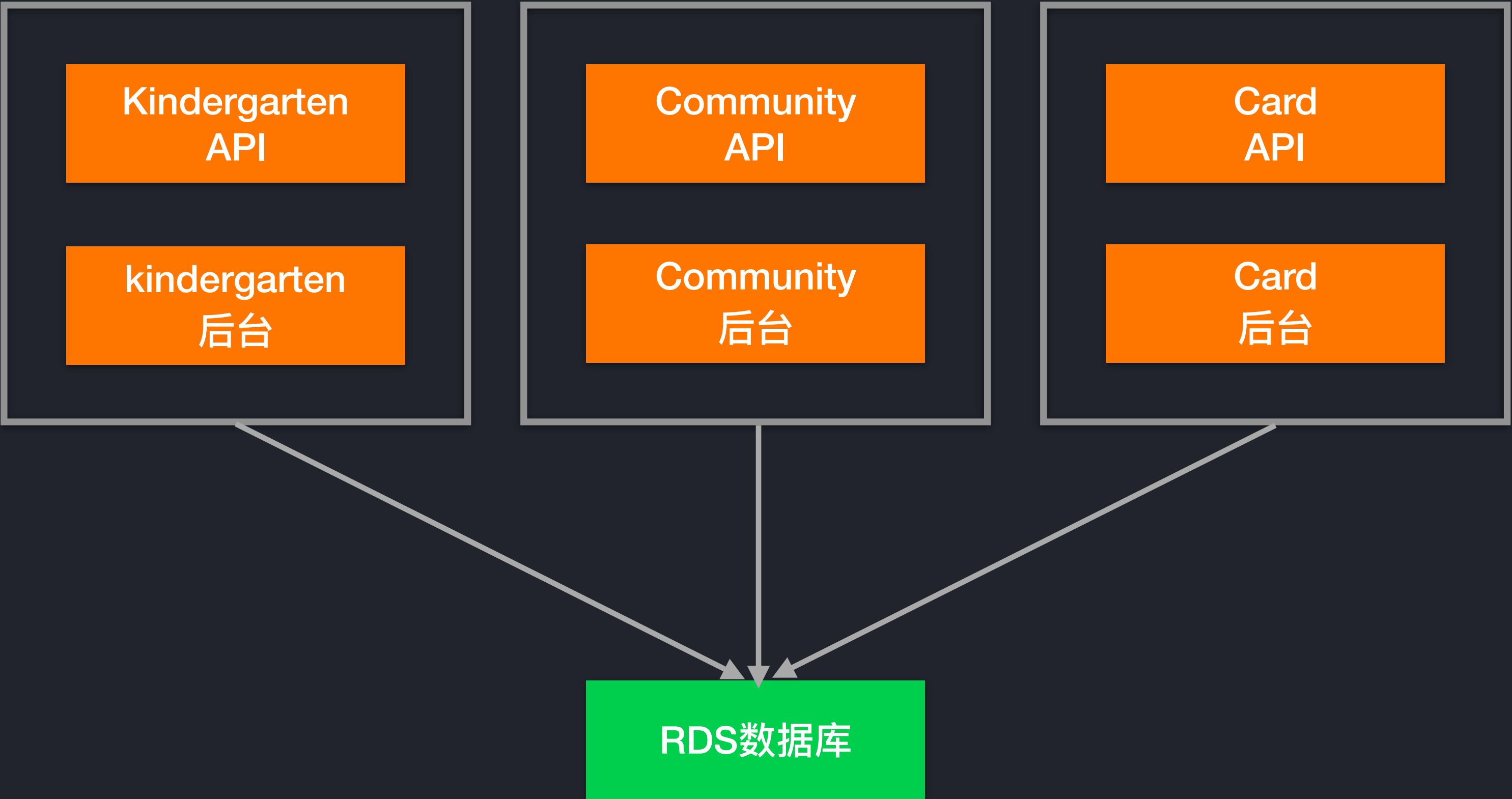


系统拆分

大系统拆分成多个业务独立子系统

多系统继续共用同一个RDS数据库

不同业务系统之间不能互相访问表



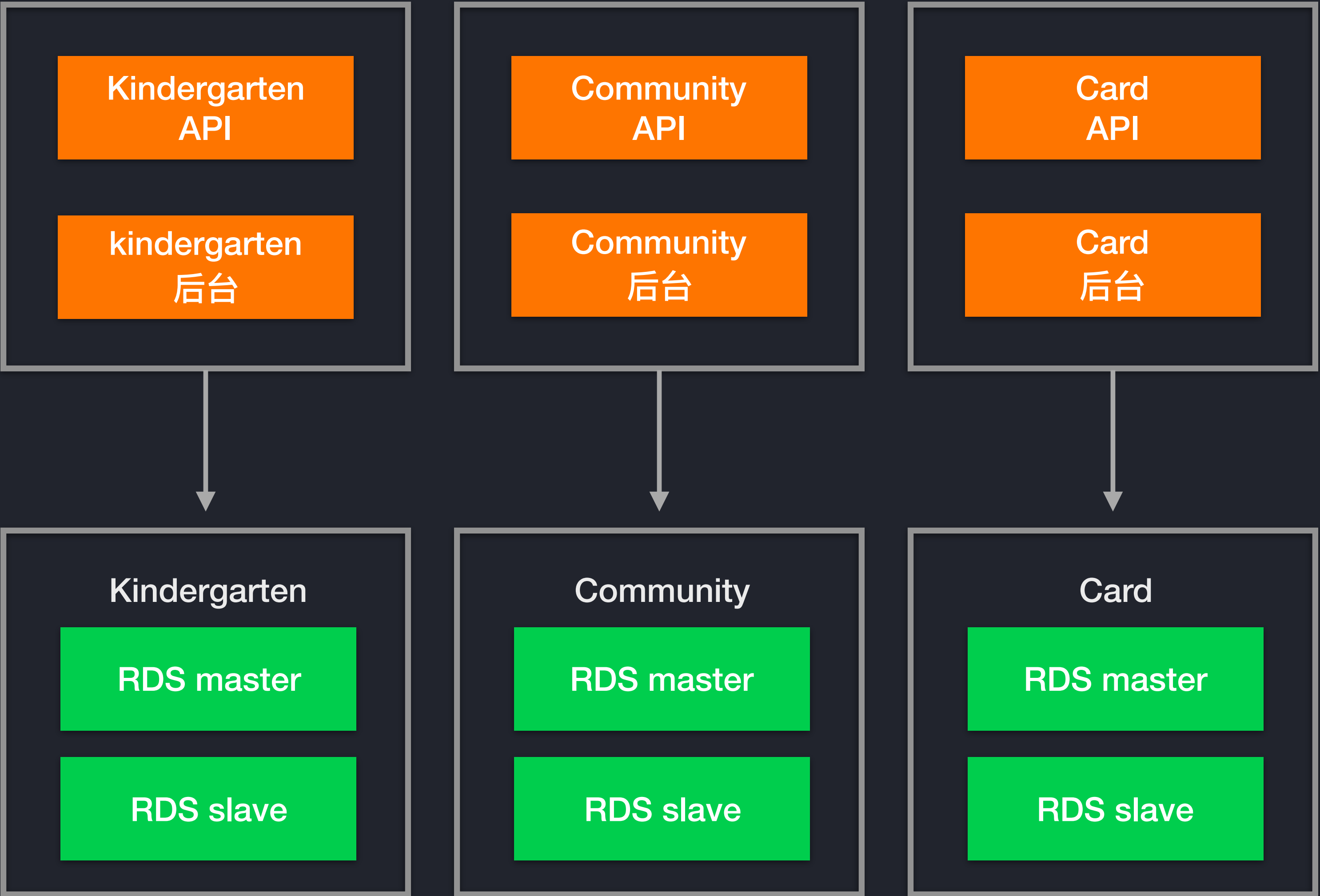
DB拆分

各子系统依赖的表独立拆分

独立RDS数据库

实现数据库主从分离

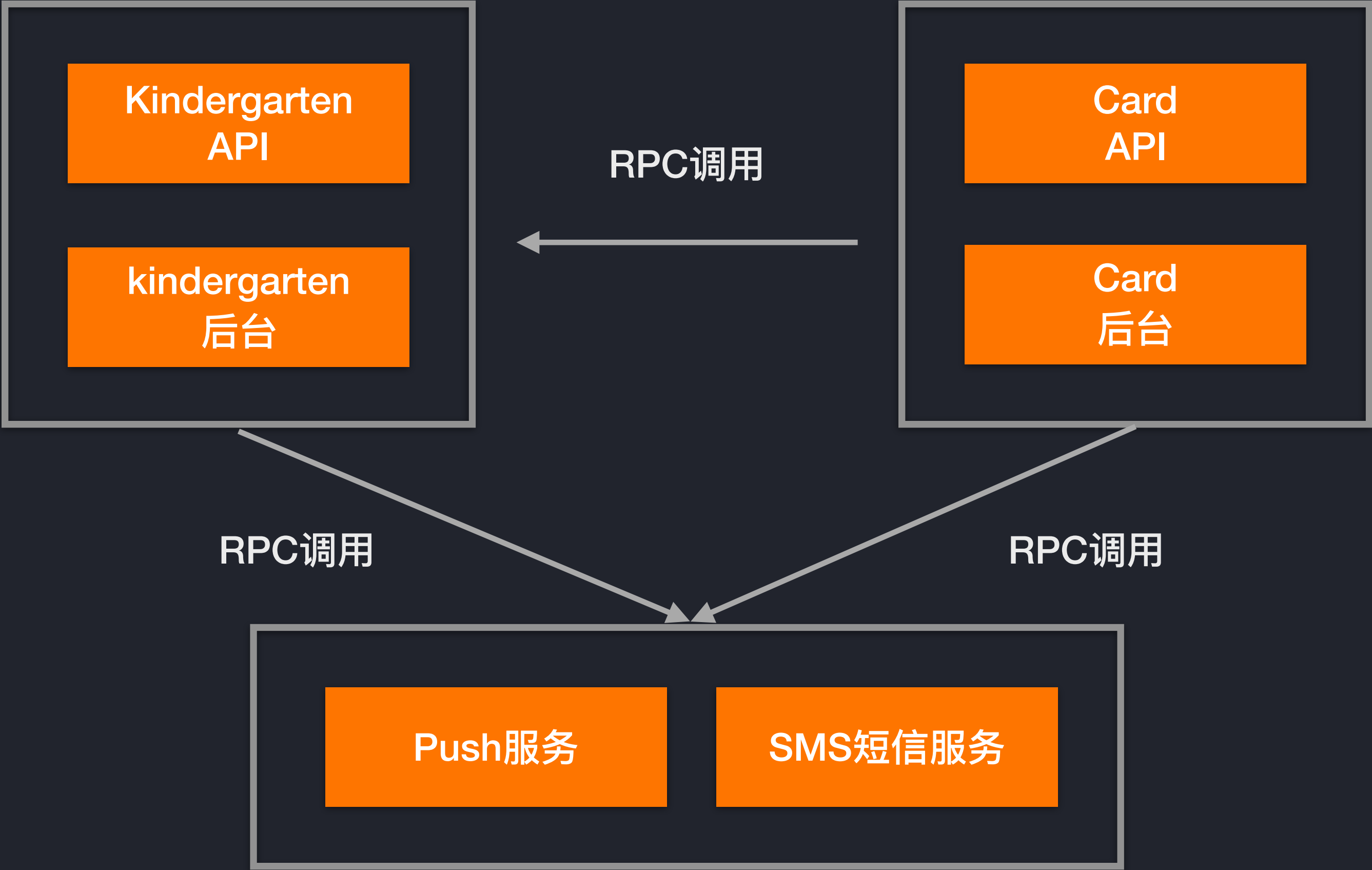
数据库物理隔离



初步服务化

Hessian实现简单RPC调用

独立第三方服务

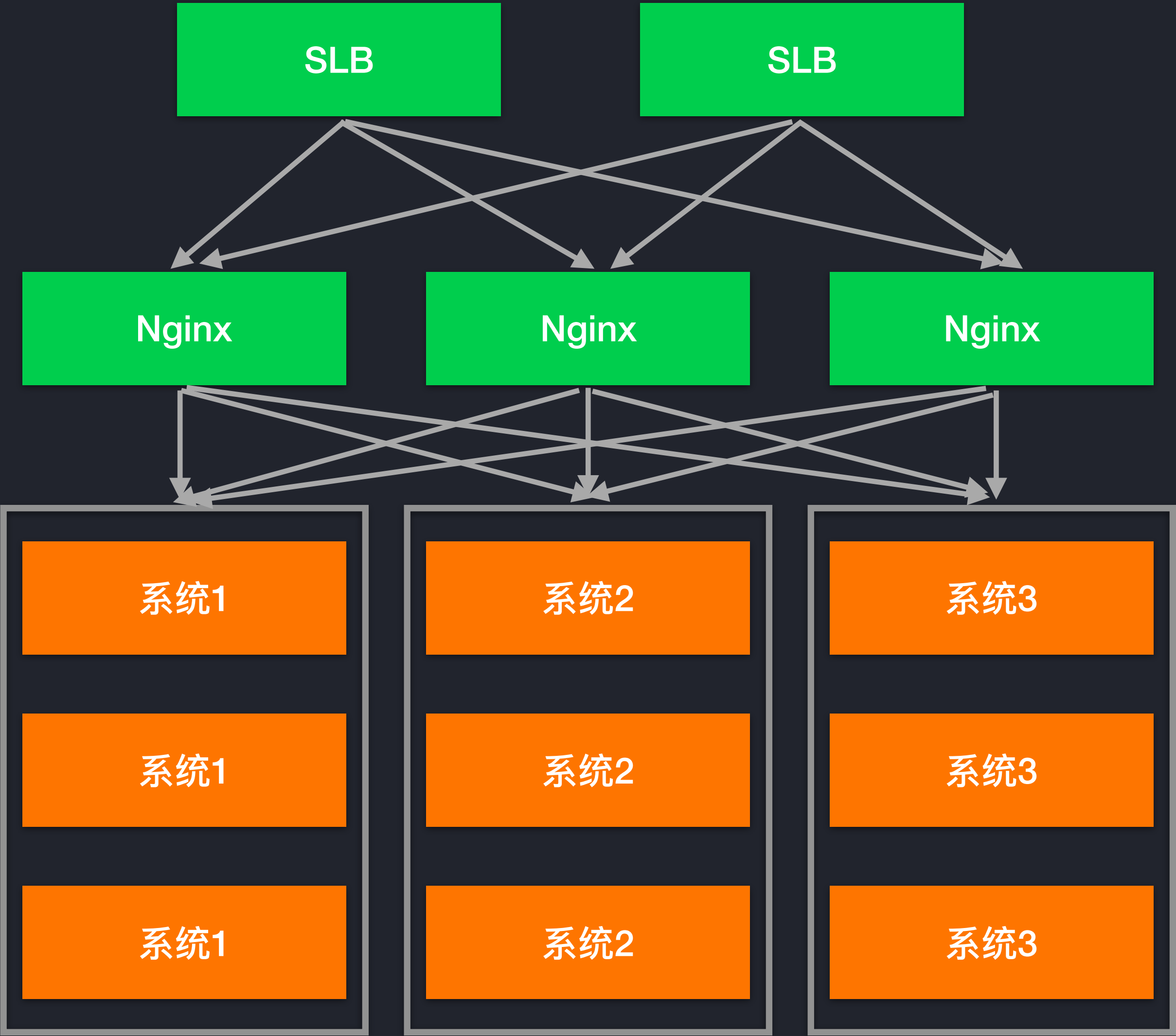


负载均衡

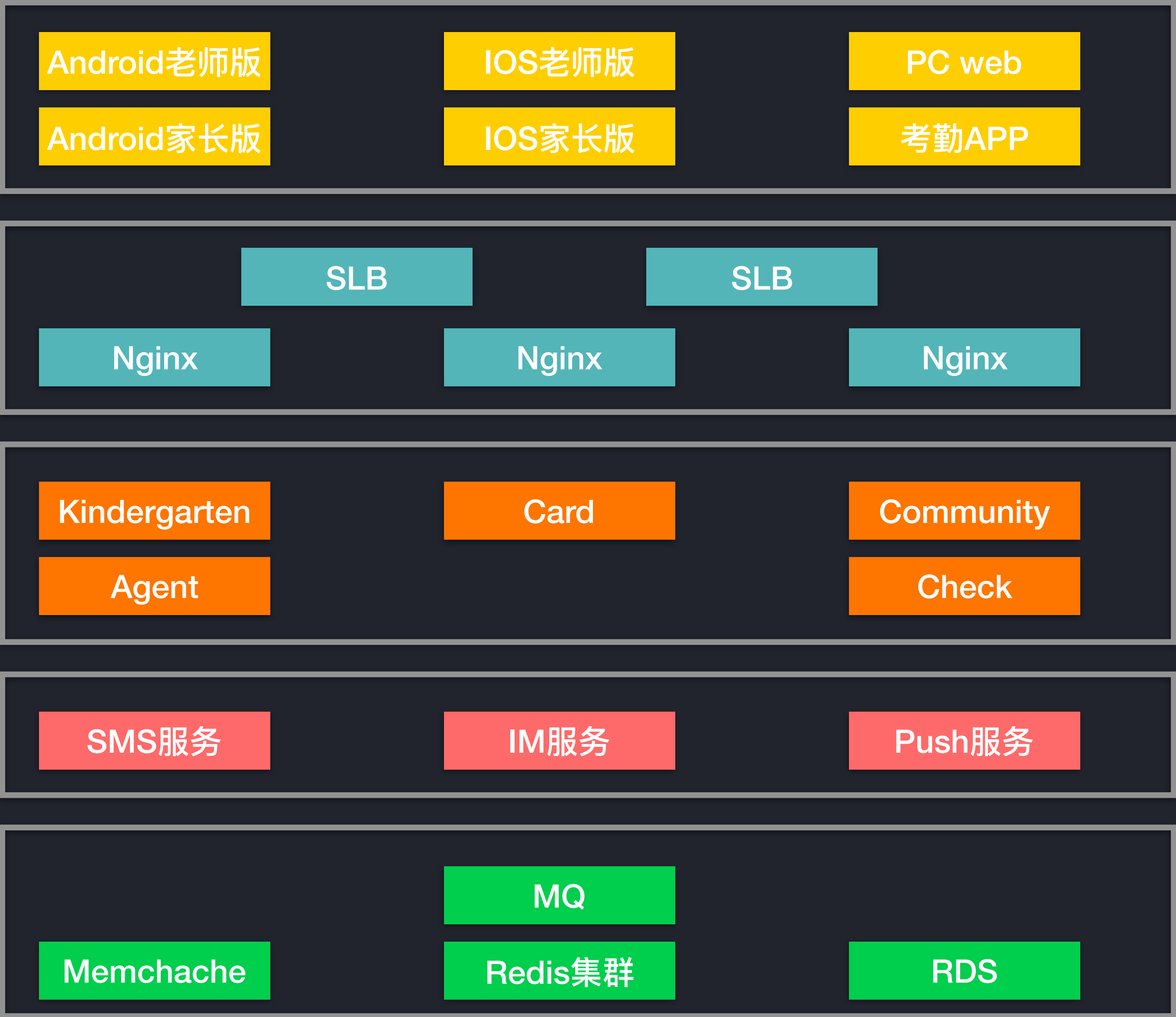
多SLB负载均衡

多Nginx反向代理负载均衡

避免单点故障



技术架构



用户层

接入层

系统层

第三方
服务层

存储层

技术架构特点

分布式
架构

系统级
拆分

初步服务
化

DB物理隔
离

MQ任务
异步化


SLB负载
均衡

4 爆发期-微服务架构

微服务技术架构实施契机



实施微服务的几个关键点



以服务为
中心

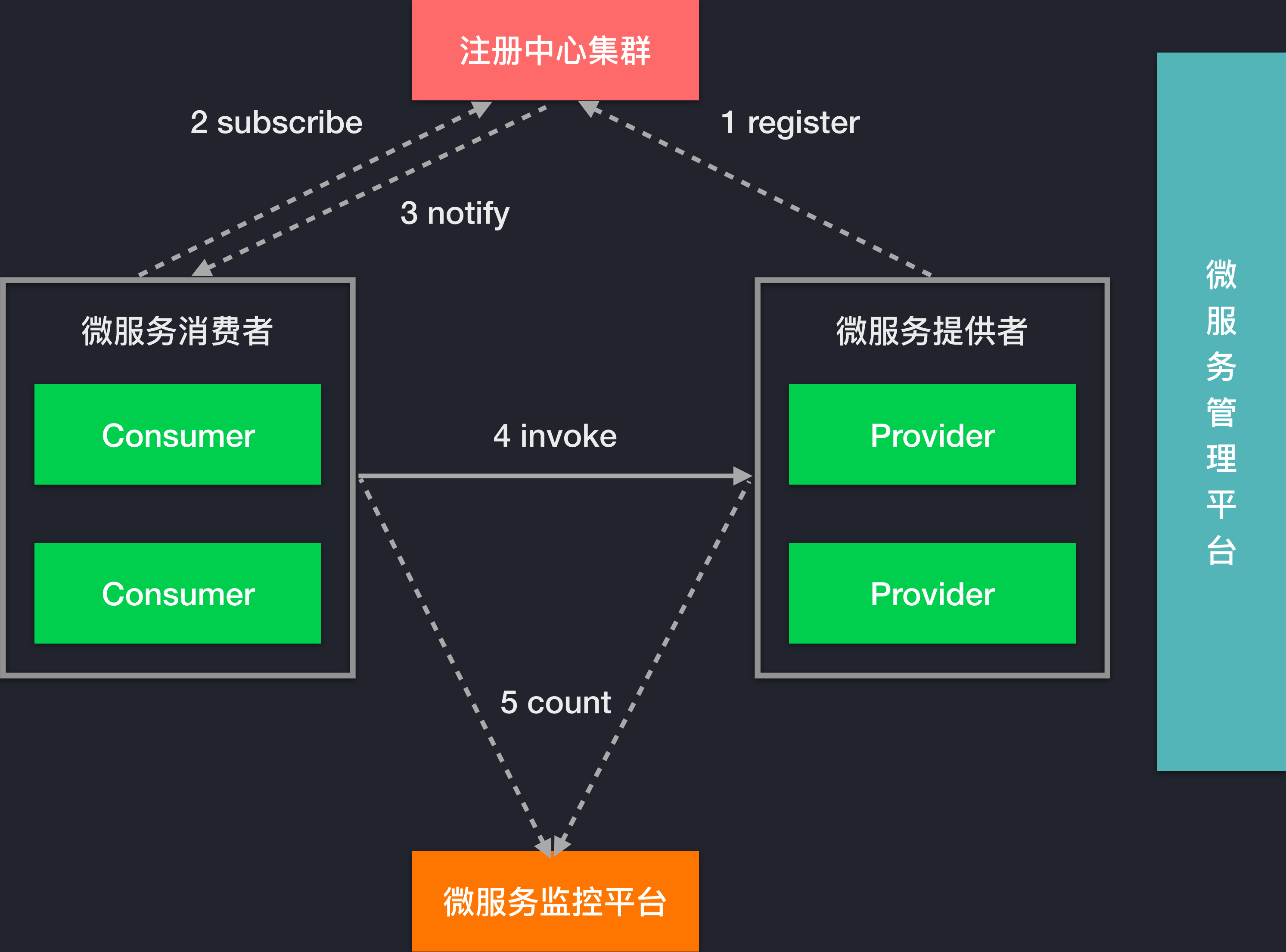
松耦合

高扩展

- 以服务为中心，一切都是服务，每个服务都针对单一业务进行封装，保证功能完整性和职责单一性
- 松耦合性，服务之间功能独立，能够独立部署，服务之间相互依赖
- 高扩展性，分散资源，团队协同工作，可无限扩展，更高的代码重用率

Dubbo微服务框架

- 成熟的高性能分布式框架，比较稳定
- 无缝接入Spring框架
- 具备服务注册、发现、路由、负载均衡、降级、权重调节
- 代码开源，可以定制化扩展功能



Dubbo微服务框架扩展之授权服务

为每一个服务分配一个APPID， consumer通过 attachment把appld传递给provider

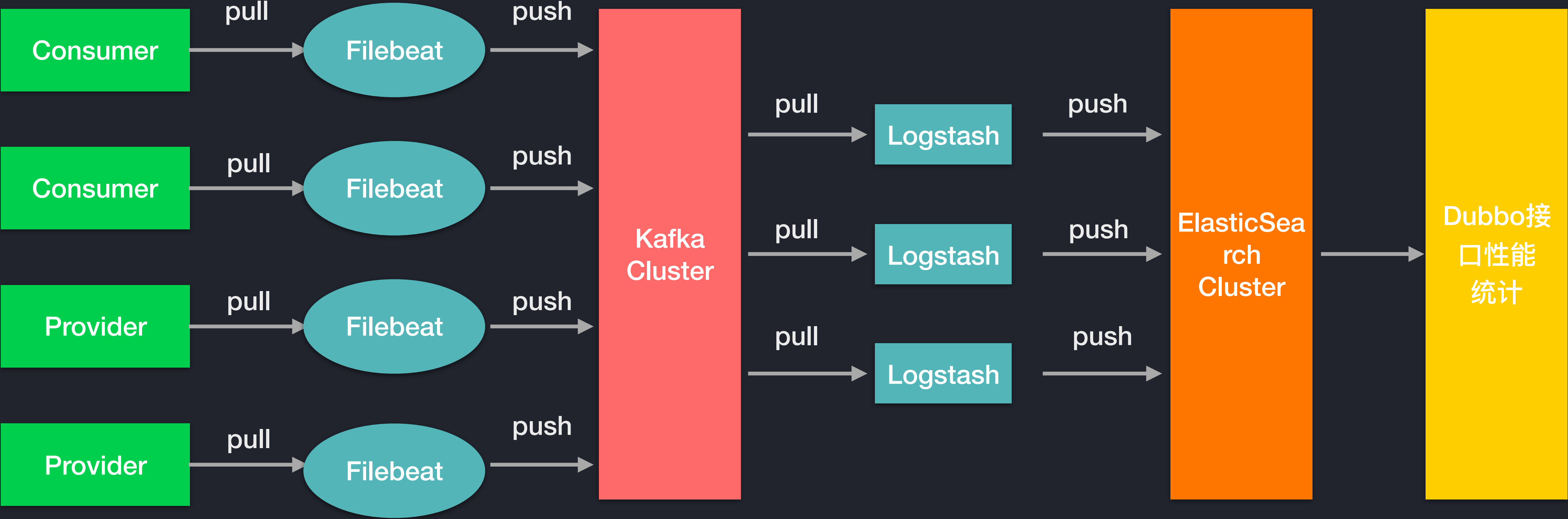
Consumer设置：
RpcContext.getContext().setAttachment(“appld”,123);

在provider读取appld：
String appld=invocation.getAttachment(“appld”);

源appld	要访问的appld	可以访问的接口
app1	app2	com.biz.UserProvider.getUser
app1	app2	com.biz.PushProvider.*
app2	App3	*



Dubbo微服务框架扩展之统计中心



Dubbo微服务框架扩展之统计中心报表

接口耗时排行

类型	服务	方法	耗时>=100ms	总数	百分比
PROVIDER	com.ibeiliao.aaa.bbbProvider	test	1000	10000	10%
Consumer	com.ibeiliao.ccc.ddd	get	100	10000	1%

接口失败排行

类型	服务	方法	失败数	总数	百分比
PROVIDER	com.ibeiliao.eee.fffProvider	add	50	10000	0.5%
Consumer	com.ibeiliao.ee1.ff2	test	20	10000	0.2%

Dubbo微服务框架扩展之统计中心报表

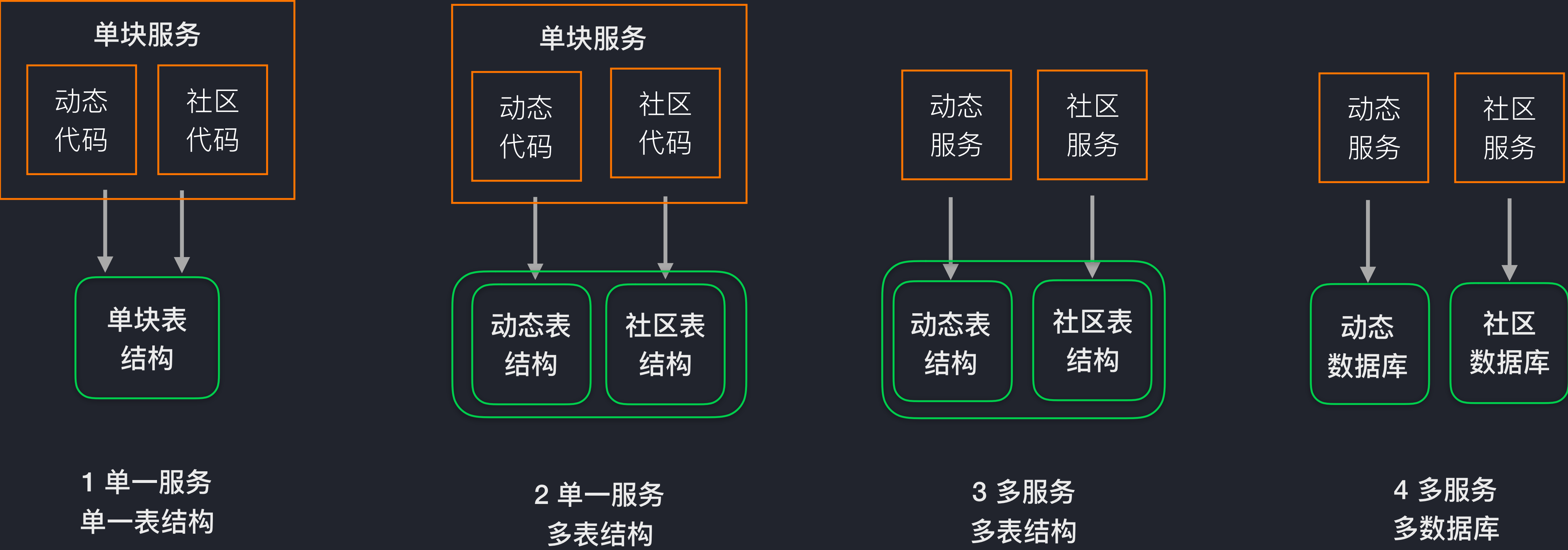
系统访问量排行

系统名称	访问量	对比昨天	最高QPS	最高QPS时间	最高QPS时处理时间>1s的数量
系统1	10000000	+20%	3000	2018-02-18 18: 00: 00	20
系统2	3000000	-10%	900	2018-02-18 19: 00: 00	10

接口访问量排行

接口名称	访问量	对比昨天	最高QPS	最高QPS时间	最高QPS时处理时间>1s的数量
接口1	2000000	-1%	800	2018-02-18 19: 00: 00	5
接口2	400000	+0%	100	2018-02-18 20: 00: 00	1

微服务架构平滑演进

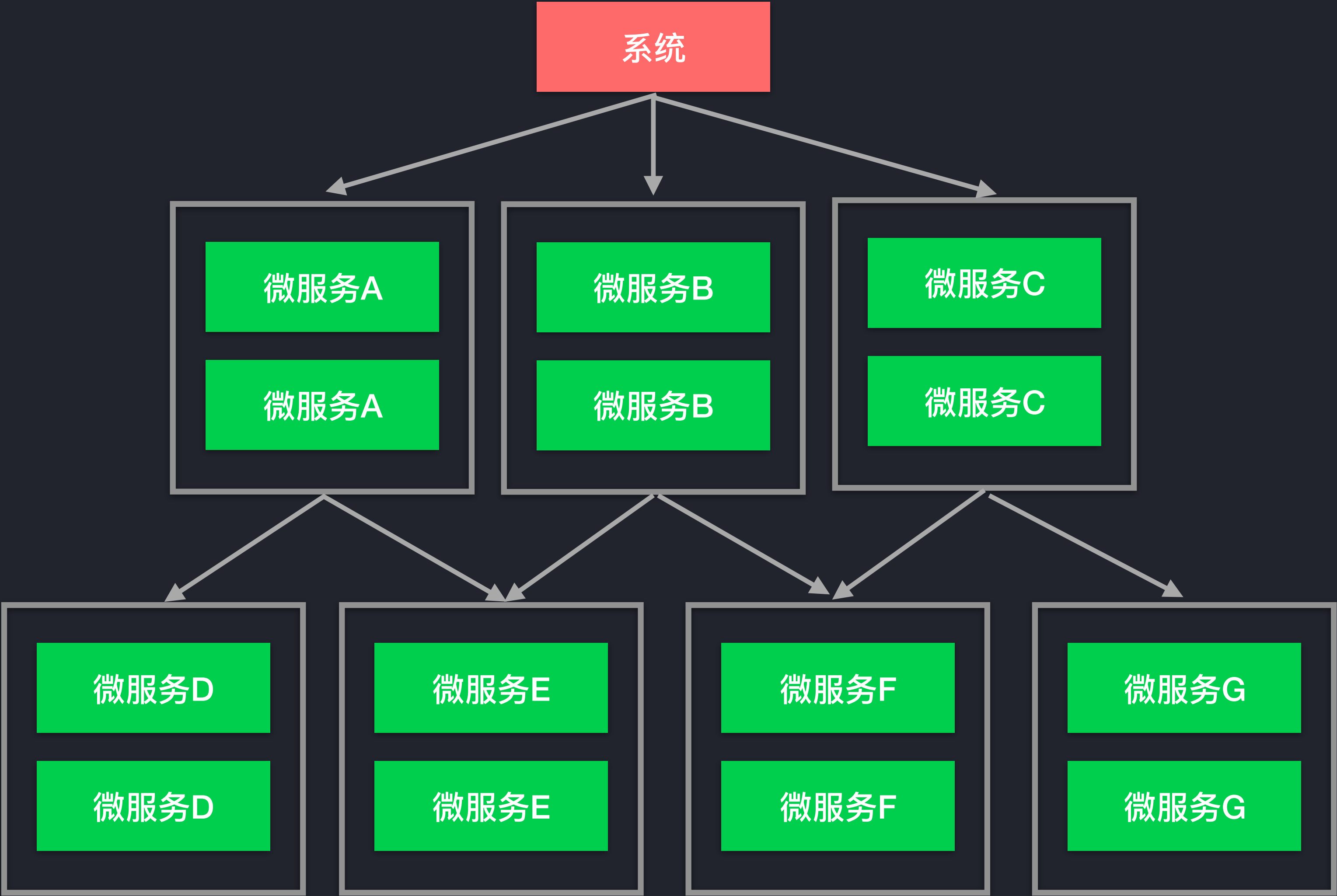


微服务架构实施

独立业务拆分微服务，独立部署

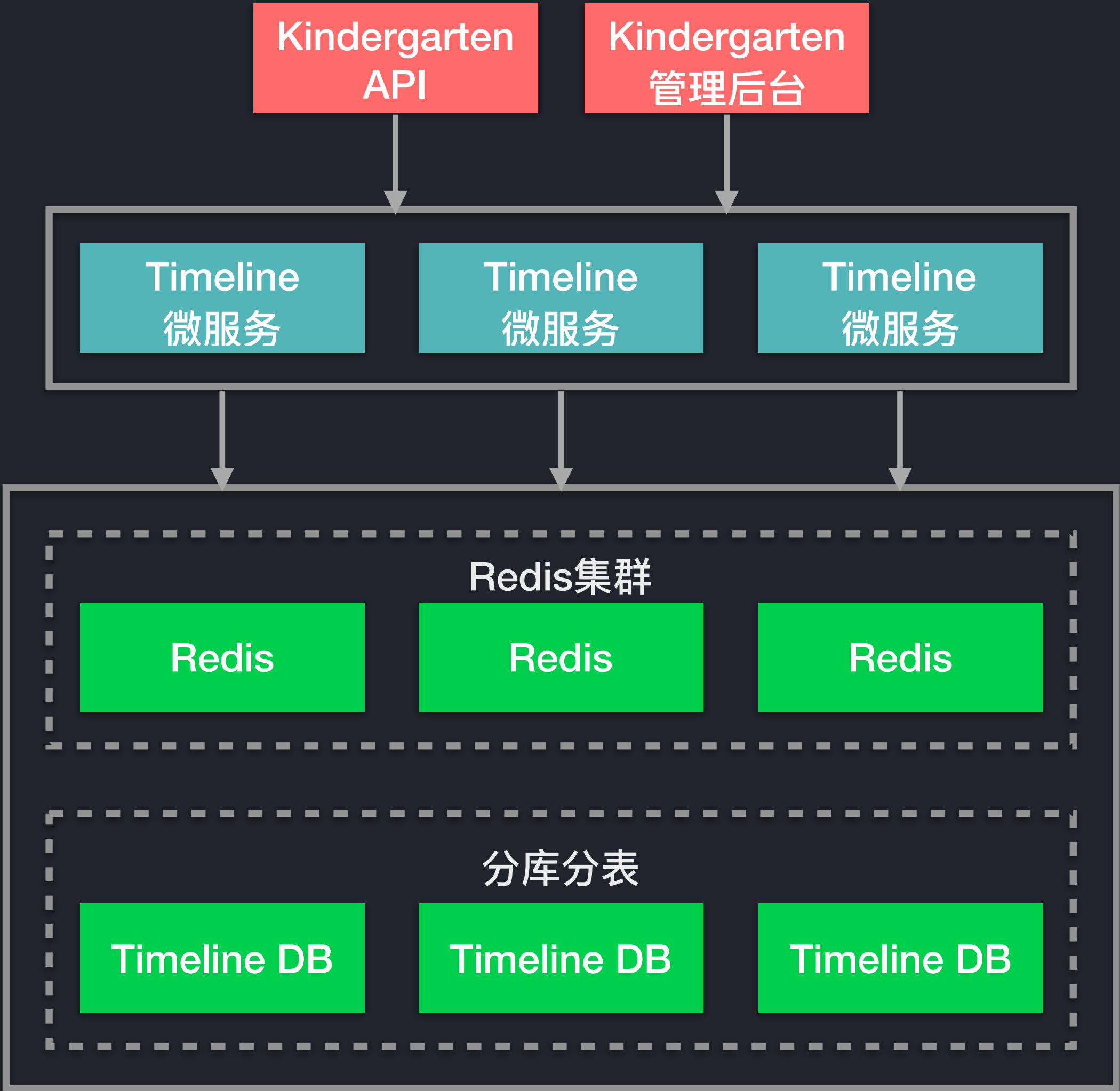
系统功能全部通过调用微服务实现，系统不能直接访问DB

每个微服务都维护独立的DB



微服务架构案例—班级动态微服务

- 班级动态微服务对业务调用方透明，业务方无需关注实现细节
- 代码复用性，独立微服务组件，业务系统不再散落班级动态业务逻辑代码
- 实施了分库分表，解决了单数据库数据量大、数据处理能力有限的瓶颈问题，接口的整体性能提升了几倍



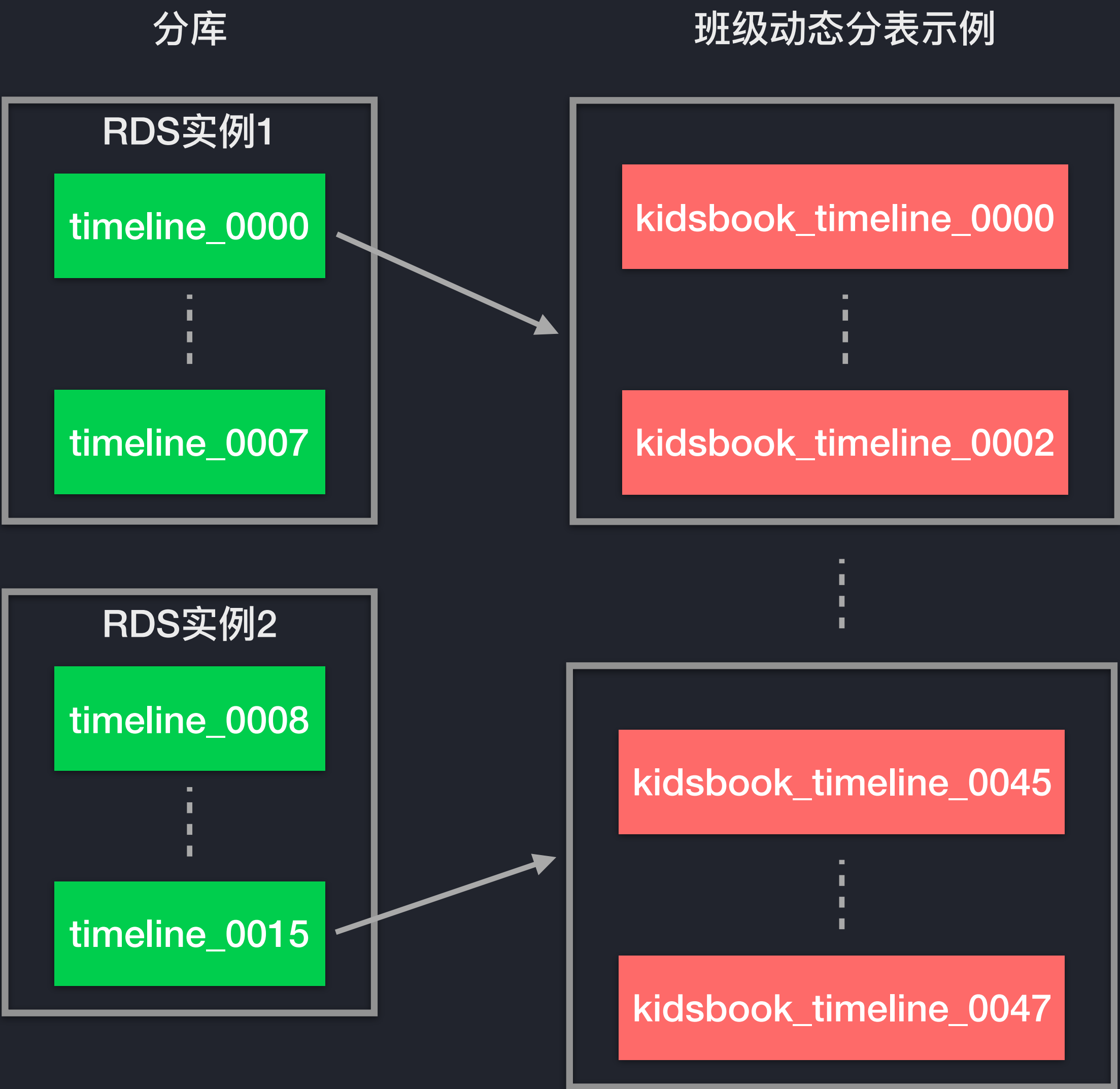
微服务架构案例—班级动态微服务Redis集群

Redis3.x版本做集群cluster
6台机器，3个主节点，3个从节点



微服务架构案例—班级动态微服务分库分表

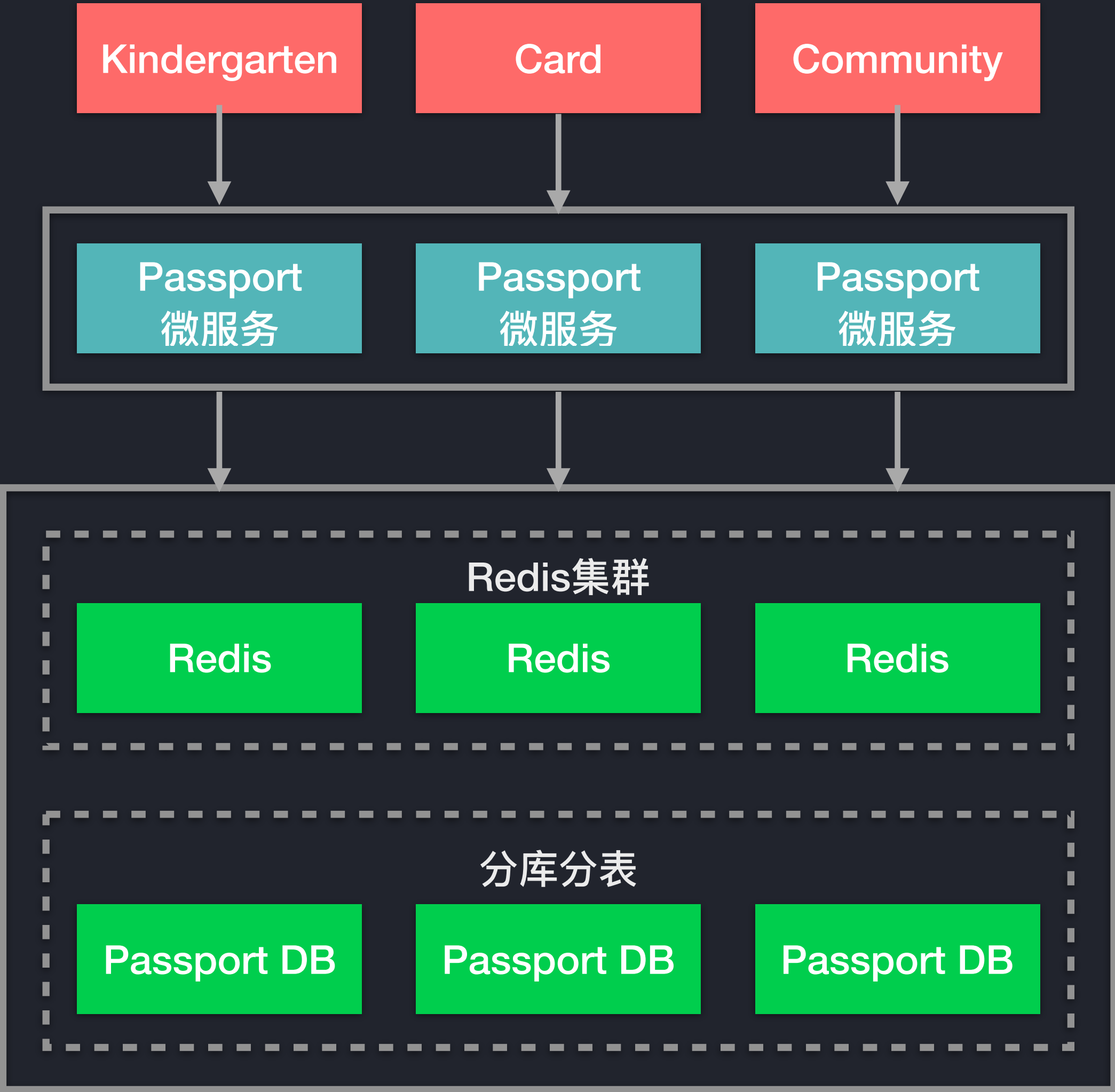
- 分库键按照kindergartenid进行HASH取模切分
- 分表键按照kindergartenid进行HASH取模切分
- 2台RDS数据库实例， 每台RDS切分8个分库
- 班级动态表每个分库切分3张表
- 动态点赞表每个分库切分8张表
- 动态回复表每个分库切分3张表



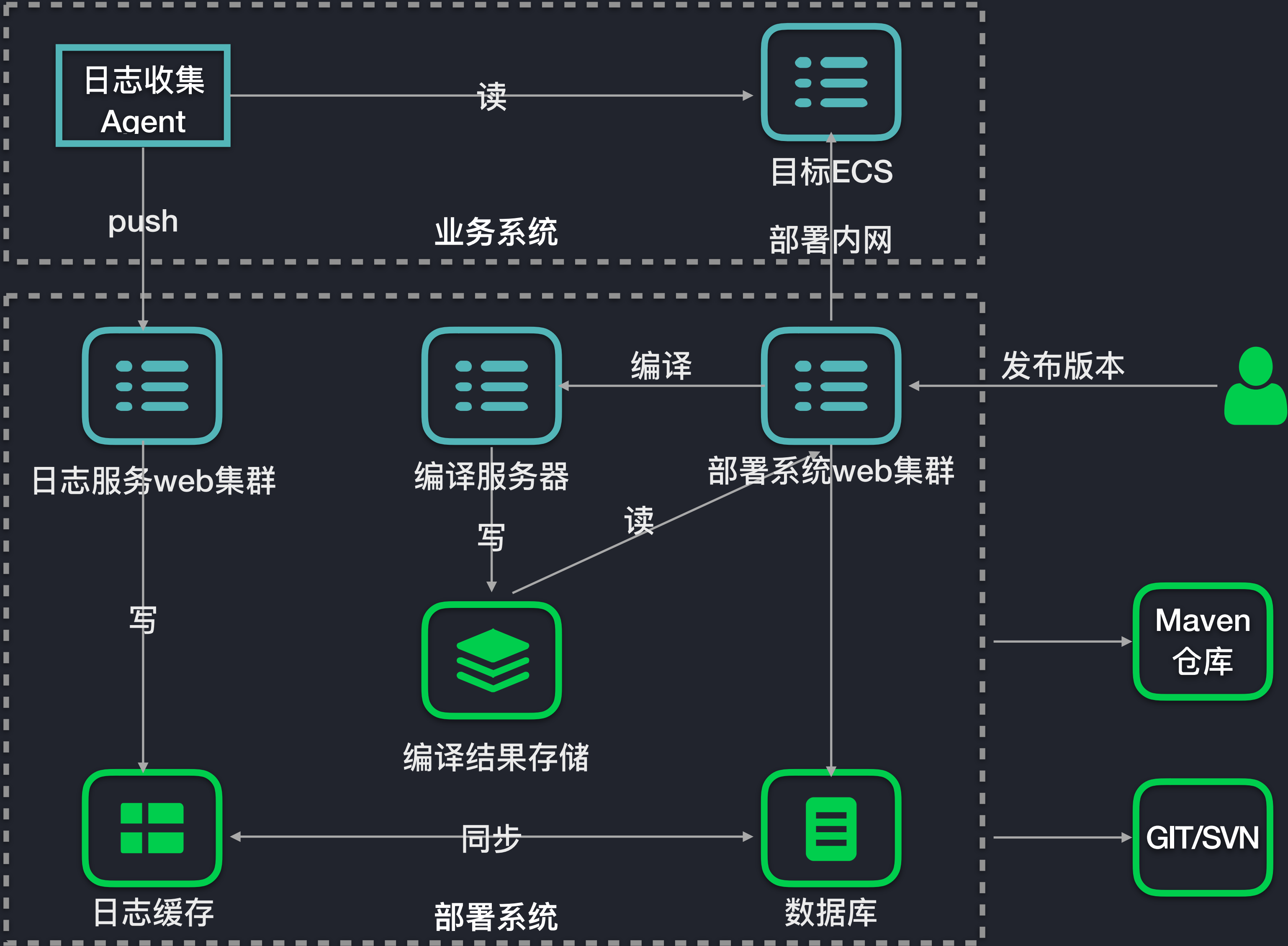
微服务架构案例—用户通行证微服务

用户数据一致性，所有跟用户逻辑相关的功能，都由用户通行证微服务提供

用户数据解耦，用户数据库独立部署，方便扩容以及性能优化



版本发布系统架构



版本发布系统特性

项目配置

一键部署

灰度发布

部署过程
日志

发布异常
处理

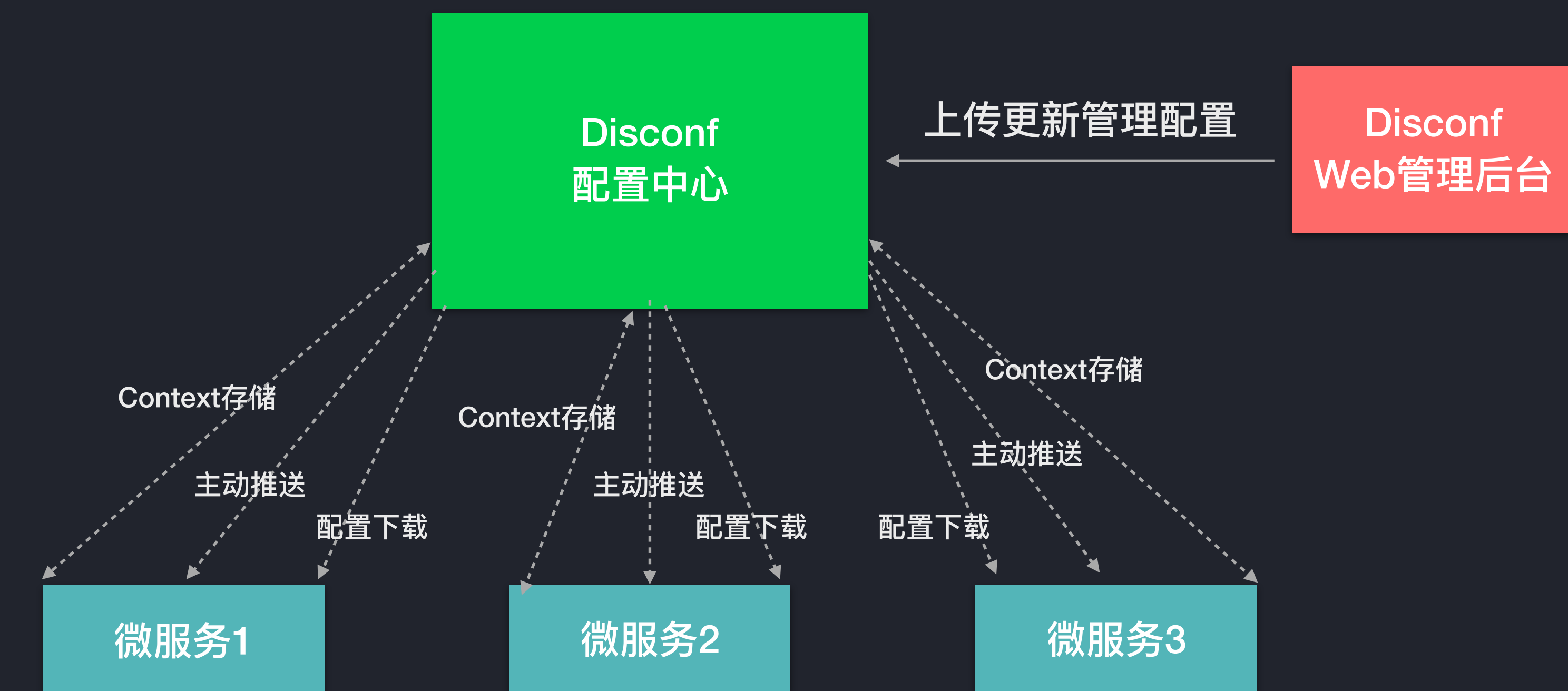
快速回滚

开发测试发布部署



- 开发环境，供研发人员在开发、调试阶段使用
- 测试环境，QA测试验收的环境
- 预发布环境，只允许内网访问，在上线生产环境前，确认是否还存在BUG，不会影响生产环境的用户，确保上线生产环境成功
- 生产环境，即线上环境，面向用户的线上环境

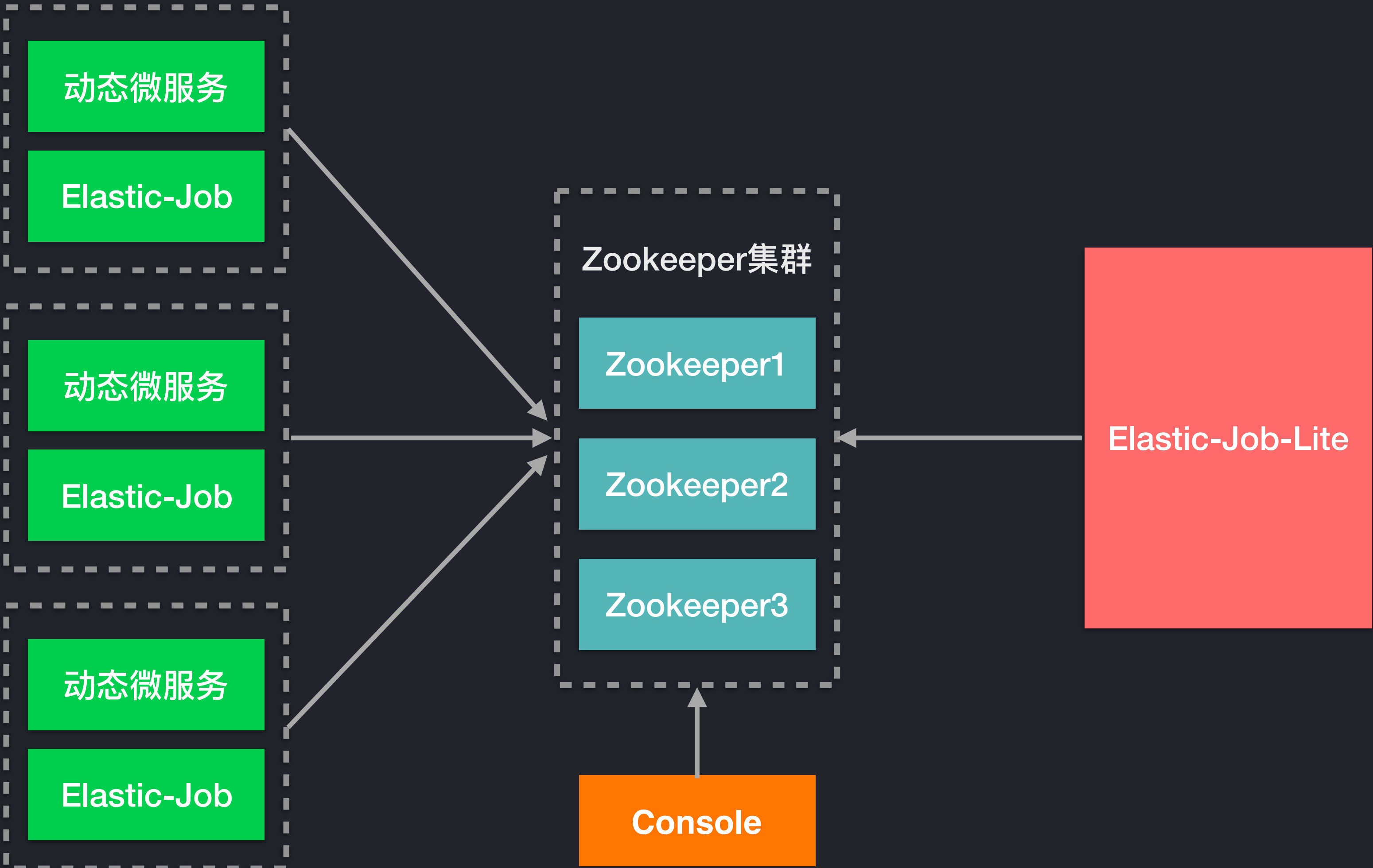
微服务治理之分布式配置中心



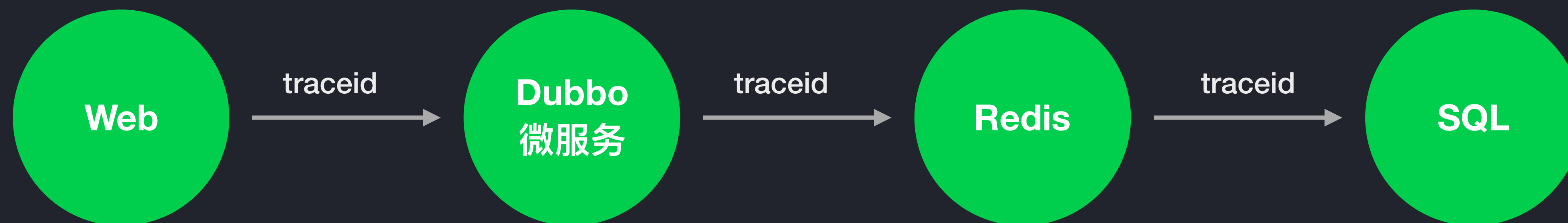
- 配置发布统一化
- 配置信息存储在云端
- 在线发布、更新配置信息
- 定制化研发，配置信息加密，避免账号、密码等敏感信息泄露

微服务治理之分布式任务调度平台

微服务的定时任务全部接入Elastic-Job分布式任务调度平台



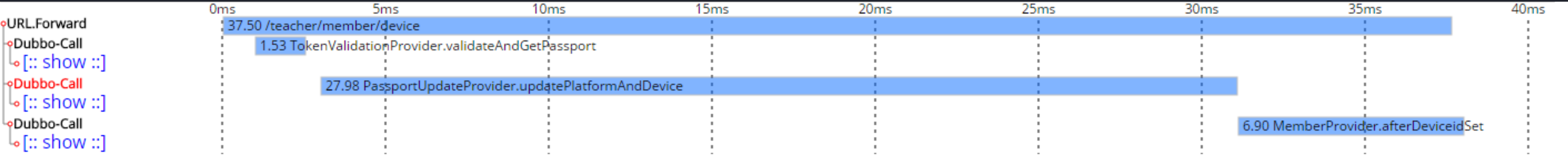
全链路跟踪



- 采用Cat实时监控系统
- 通过Traceid将全链路日志串起来
- 全链路监控web请求、Dubbo微服务、Redis缓存、SQL调用耗时

全链路跟踪示例

t14:29:32.486	URL.Forward	/teacher/member/device		
E14:29:32.486	URL.Forward	URL.Forward.Method	HTTP/POST /teacher/member/device?clientid=1&platform=1&version=4.15.0	
t14:29:32.487	Dubbo-Call	TokenValidationProvider.validateAndGetPassport		
[:: show ::]				
T14:29:32.488	Dubbo-Call	TokenValidationProvider.validateAndGetPassport	1.53ms	
t14:29:32.489	Dubbo-Call	PassportUpdateProvider.updatePlatformAndDevice		
[:: hide ::]				
RootLogview				
t14:29:32.489	Dubbo-Provider	PassportUpdateProvider.updatePlatformAndDevice		
A14:29:32.490	Jedis	JedisCommands.get	0.34ms	
t14:29:32.490	SQL	PassportDetailDao.updatePlatformAndDeviceAndToken		
E14:29:32.490	SQL-stat	update	update ibl_passport.t_passport_detail_53 set device_token=?, platform=?, device_id=? where member_id=?	
T14:29:32.494	SQL	PassportDetailDao.updatePlatformAndDeviceAndToken	4.27ms	
t14:29:32.495	SQL	DeviceIdIdxDao.delete		
E14:29:32.495	SQL-stat	DELETE	DELETE FROM ibl_passport.t_device_id_idx_7 WHERE device_id=? AND platform=?	
T14:29:32.500	SQL	DeviceIdIdxDao.delete	5.21ms	
t14:29:32.500	SQL	DeviceIdIdxDao.getByDeviceIdPlatform		
E14:29:32.500	SQL-stat	SELECT	SELECT * FROM ibl_passport.t_device_id_idx_7 WHERE device_id=? AND platform=?	
T14:29:32.502	SQL	DeviceIdIdxDao.getByDeviceIdPlatform	2.37ms	
t14:29:32.503	SQL	DeviceIdIdxDao.insert		
E14:29:32.503	SQL-stat	insert	insert into ibl_passport.t_device_id_idx_7 (id,member_id,device_id,platform) values (?, ?, ?, ?) ON DUPLICATE KEY UPDATE member_id=?	
T14:29:32.507	SQL	DeviceIdIdxDao.insert	4.83ms	
A14:29:32.515	Jedis	JedisCommands.hset	0.32ms	
A14:29:32.515	Jedis	JedisCommands.hincrBy	0.28ms	
A14:29:32.516	Jedis	JedisCommands.hset	0.24ms	
A14:29:32.516	Jedis	JedisCommands.del	0.25ms	
T14:29:32.516	Dubbo-Provider	PassportUpdateProvider.updatePlatformAndDevice	27ms	
T14:29:32.516	Dubbo-Call	PassportUpdateProvider.updatePlatformAndDevice	27ms	
t14:29:32.517	Dubbo-Call	MemberProvider.afterDeviceidSet		
[:: show ::]				
T14:29:32.523	Dubbo-Call	MemberProvider.afterDeviceidSet	6.90ms	
T14:29:32.523	URL.Forward	/teacher/member/device		



微服务技术架构



微服务技术架构特点



分布式
架构

微服务

高可用

- 完全实现了分布式部署架构，系统与微服务组件都非常容易扩展
- 以服务为中心，全面构建了微服务组件
- 系统、微服务组件、缓存、MQ消息队列、DB等均无单点故障

极客邦企业培训与咨询

帮助企业与技术人员成长



精品课程

Excellent Course

- ✓ 《互联网大规模分布式架构设计与实践》
- ✓ 《基于大数据的企业运营与精准营销》
- ✓ 《大数据和人工智能在金融领域的应用》
- ✓ 《区块链应用与开发技术高级培训》
- ✓ 《通往卓越管理的阶梯》



扫码关注官方微信服务号
了解更多课程详细信息

Geekbang
极客邦科技