

基于混合云架构的高可用实践

——云栖社区在线实时培训第八期

@YOH0! 李健



潮流有货
IN YOHO!

年轻人潮流购物中心

潮流随性 有货随行

 iPhone 下载

 Android 下载



一、为何选择混合云架构

为何选择混合云架构？

基于混合云的系统架构

服务化

自动化

一、为什么选择混合云架构

为什么使用公有云？为什么不完全使用公有云？

业务痛点

- 快速响应，灵活多变
- 活动频繁，弹性伸缩
- 流量突发增长，峰值应对

运维痛点

- CMDB流程繁琐
- 故障处理耗时耗力
- 服务器环境差异
- 硬件资源分配

成本控制

- 硬件成本
- 运维成本
- 迁移成本

安全控制

- 数据安全
- 网络安全
- 应用安全

二、基于混合云的系统架构

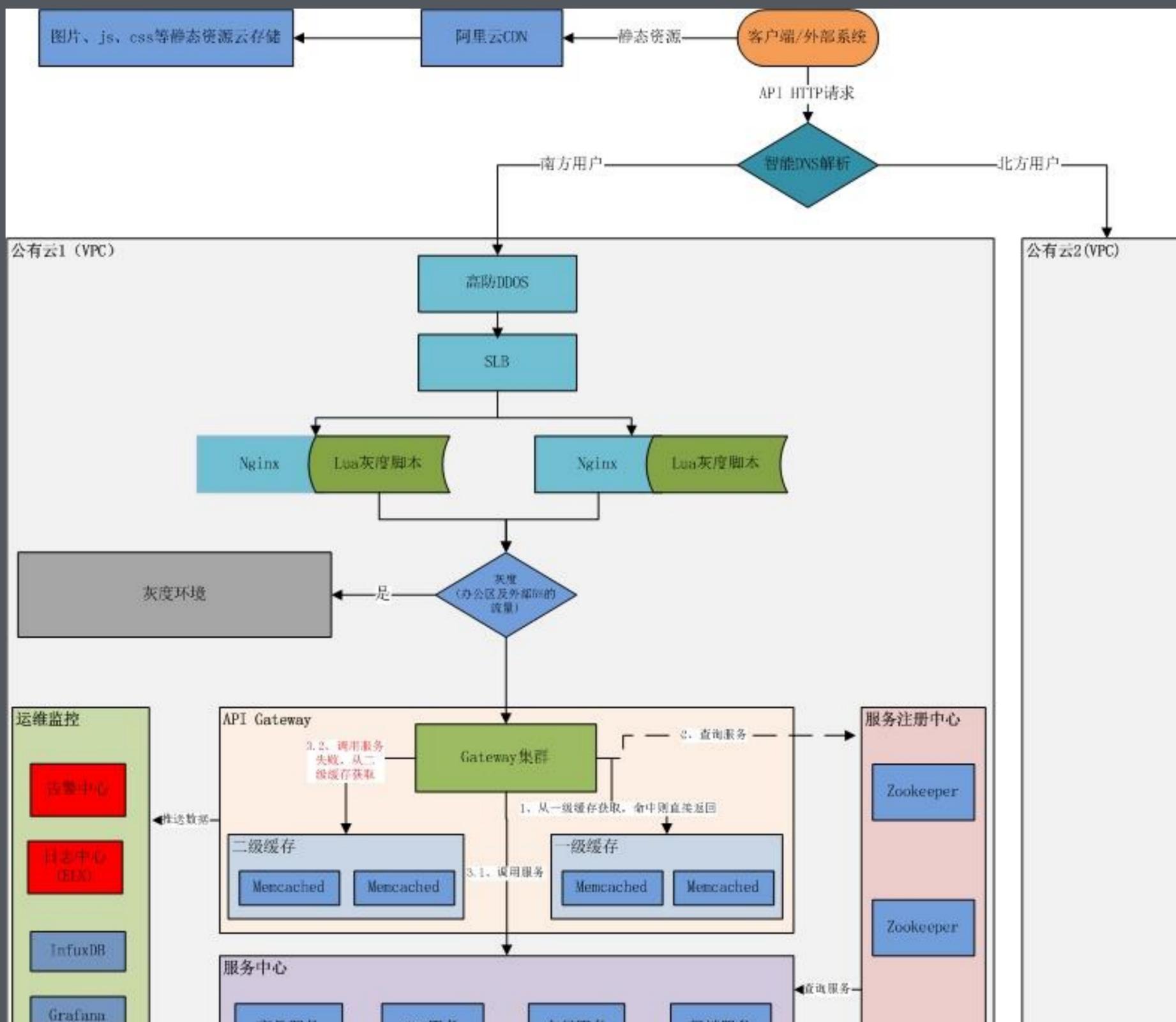
为何选择混合云架构？

基于混合云的系统架构

服务化

自动化

二、基于混合云的系统架构



二、基于混合云的系统架构

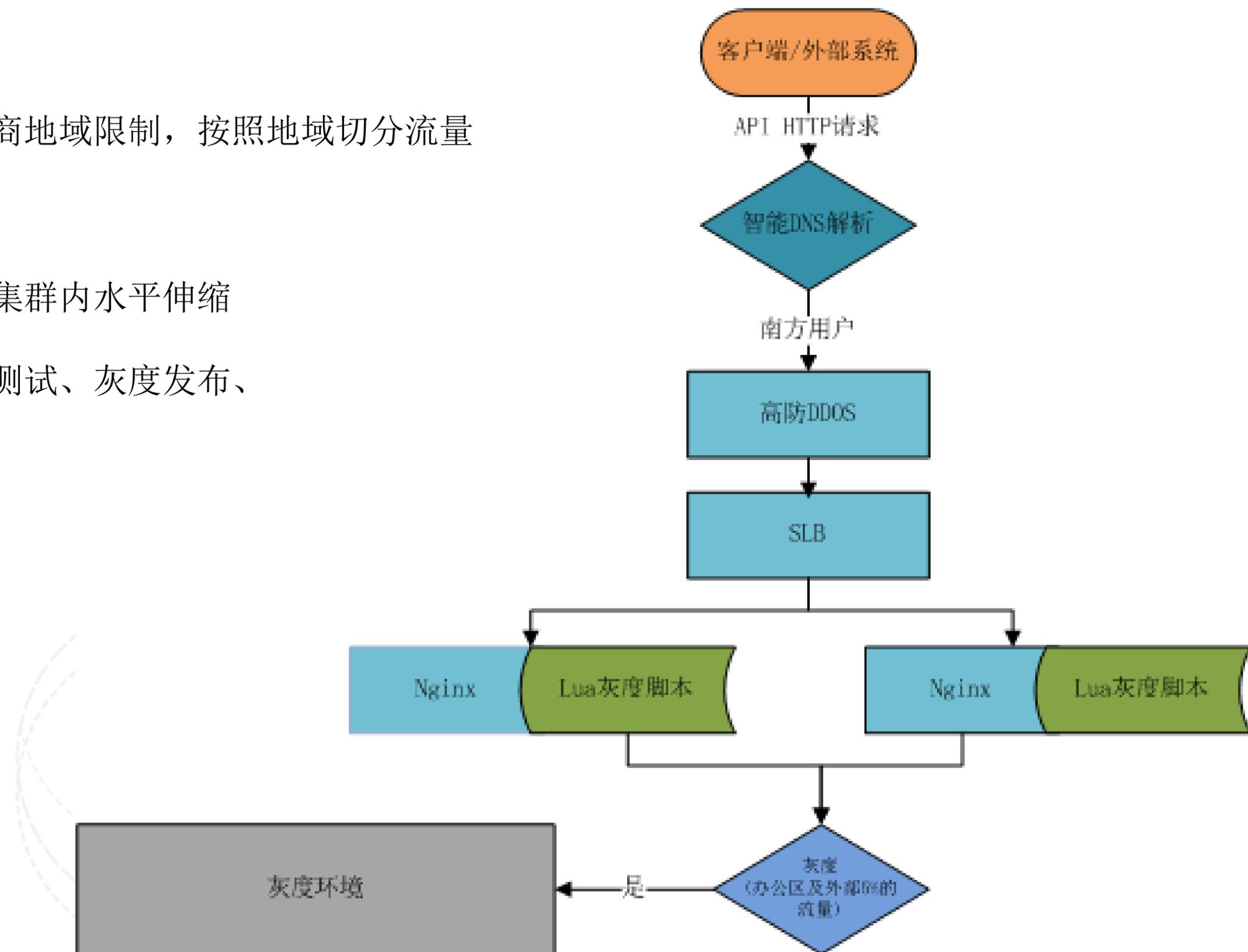


二、基于混合云的系统架构 —— 客户端

- 使用HttpDNS，解决LocalDNS的潜在问题
防止DNS劫持问题，避免LocalDNS缓存问题，避免域名解析慢或者解析失败，快速应对故障处理
- 使用阿里云CDN加速静态资源（图片、js、css、html）
阿里云CDN全球超过500+个节点，在国内，阿里云CDN节点绝大多数位于省会等一线城市，骨干网接入，能提供很好的网络和服务质量。
- 通过预加载和客户端缓存，实现离线化，大幅提升性能
App客户端通过Cache-Control HTTP头来定义自己的缓存策略
- 服务降级
客户端根据降级策略可在特定条件下对非关键业务进行降级，以保证核心关键业务的高可用
- 网络质量监测
根据用户在2G/3G/4G/Wi-Fi等不同网络环境下设置不同的超时参数，以及网络服务的并发数量。同时根据不同的网络质量设计不同的产品体验
- 业务异常监测
在客户端监测用户使用过程中的异常情况，实时上报异常数据，实时定位分析问题
- App出现故障时快速切换至H5模式
如果app出现大面积故障，可快速切换至web app模式

二、基于混合云的系统架构 —— 入口层

- 使用智能DNS分发流量至双中心，突破运营商地域限制，按照地域切分流量
- 使用阿里云高防DDOS产品防护DDOS攻击
- 使用阿里云SLB作为应用层负载均衡，实现集群内水平伸缩
- 使用Nginx+Lua做反向代理、分流限流、AB测试、灰度发布、故障切换、服务降级

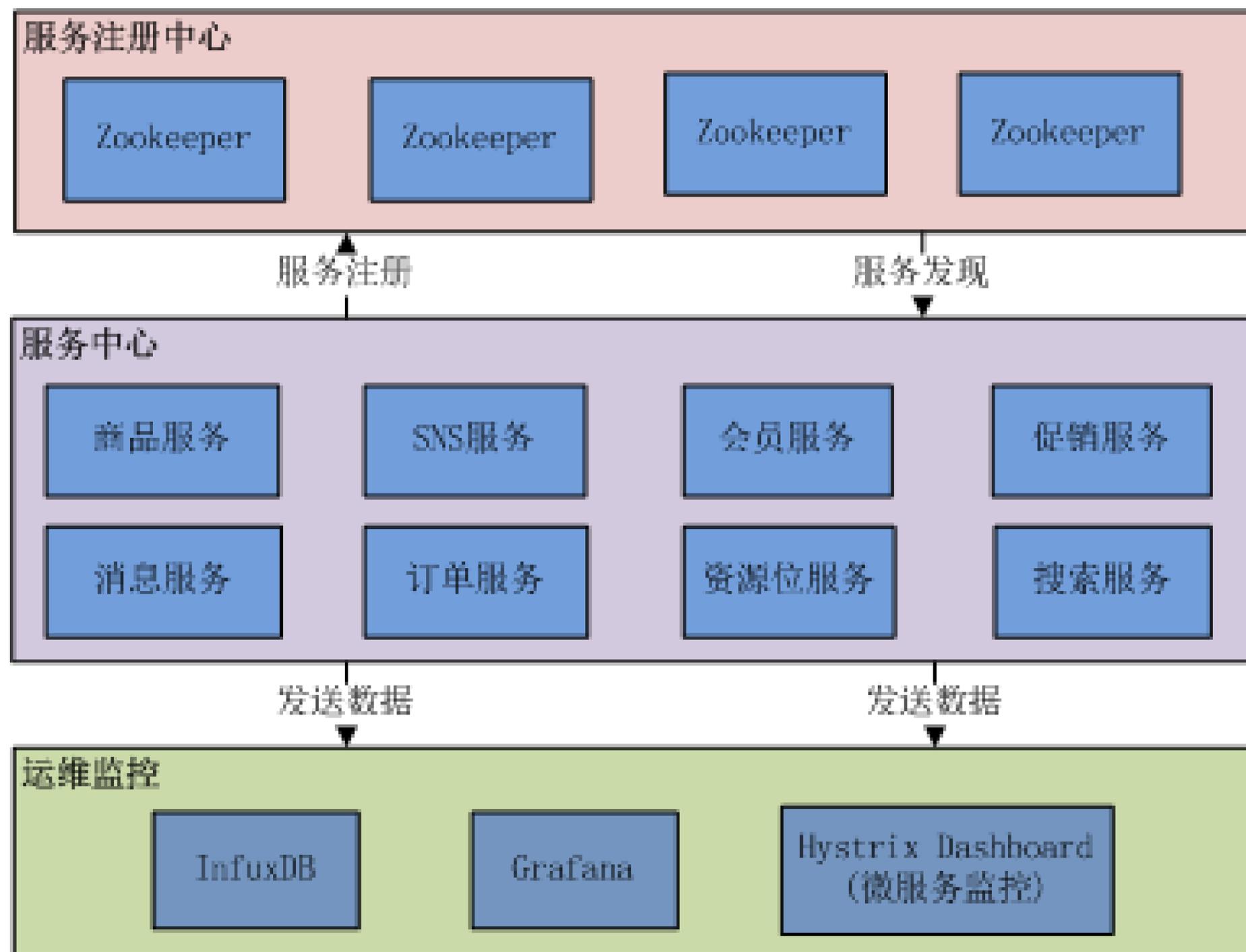


二、基于混合云的系统架构 —— 网关层

- 安全控制
在网关层统一完成客户端请求的身份认证，统一完成数据的加密解密
- 分流与限流
将流量按业务切分，路由至后端不同业务线的服务中心，以实现后端服务的实时动态水平扩展。
当流量超过预定阈值，系统出现瓶颈的时候自动限制流量进入后端服务，以防止雪崩效应。
- 服务降级
在系统出现瓶颈是，自动降级非关键业务，以保证核心业务的正常运转
- 熔断机制
根据后端服务的健康状况，自动熔断对服务的调用，以防止雪崩效应
- 异步化
网关异步化调用后端服务，避免长期占用请求线程，快速响应处理结果，快速释放线程资源
- 服务聚合
- 一级缓存用于加速热点数据
- 二级缓存用户容灾

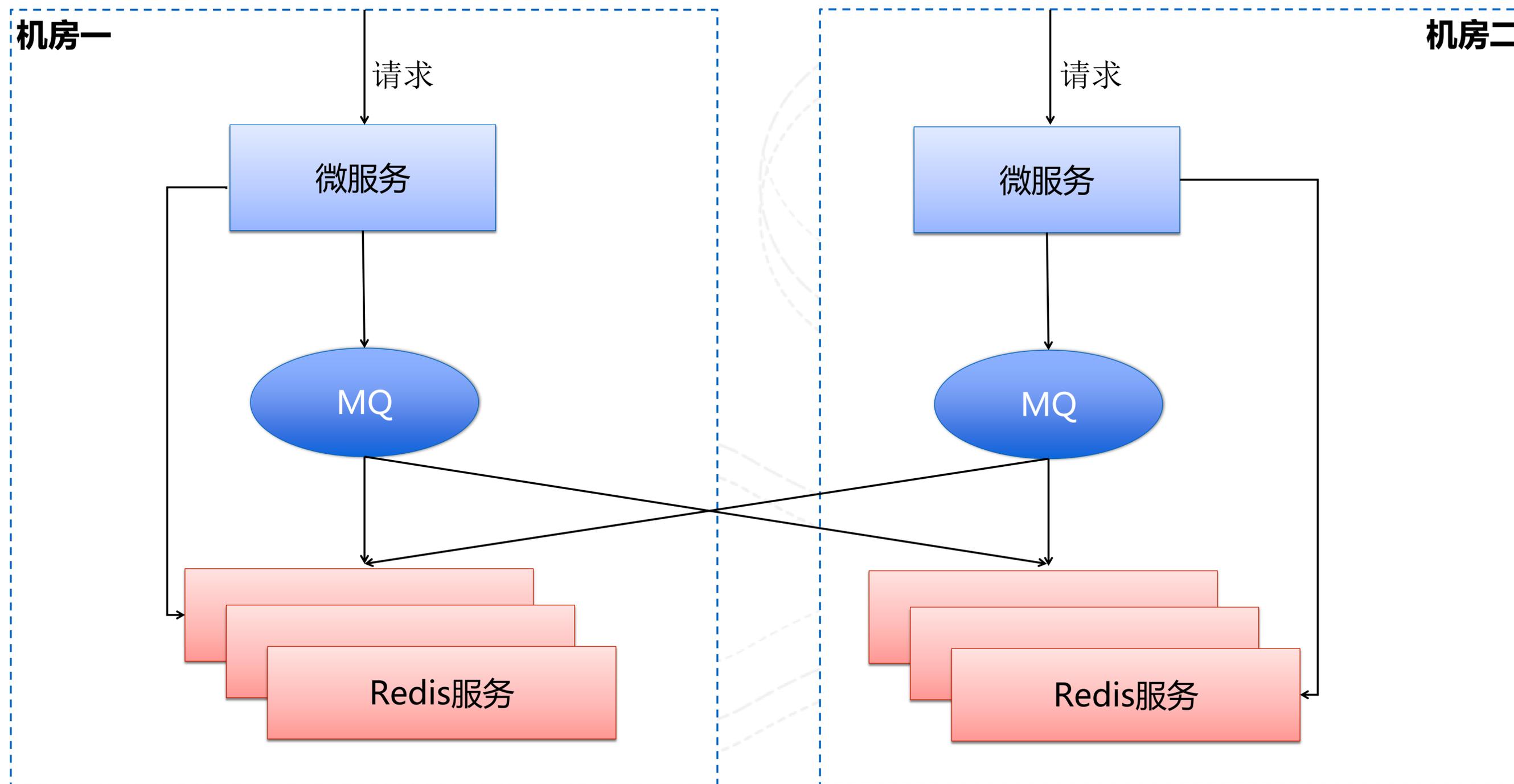
二、基于混合云的系统架构 —— 服务层

- 服务注册
- 服务发现
- 服务调用
- 负载均衡
- 服务监控
- 性能优化
- 调用链分析
- 服务降级
- 流量控制



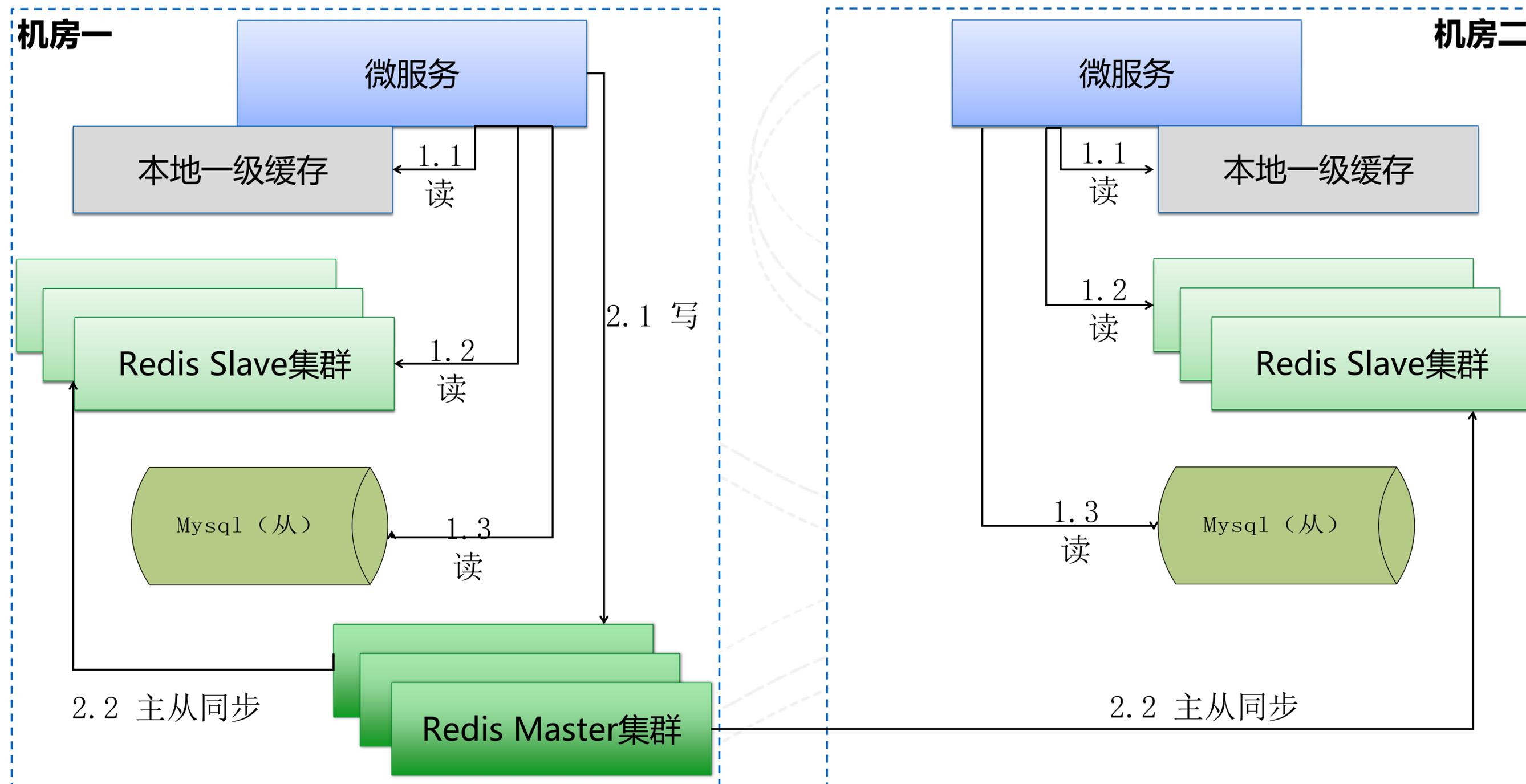
二、基于混合云的系统架构 —— 缓存层

场景一：用户数据持久化（如收藏夹、浏览记录）



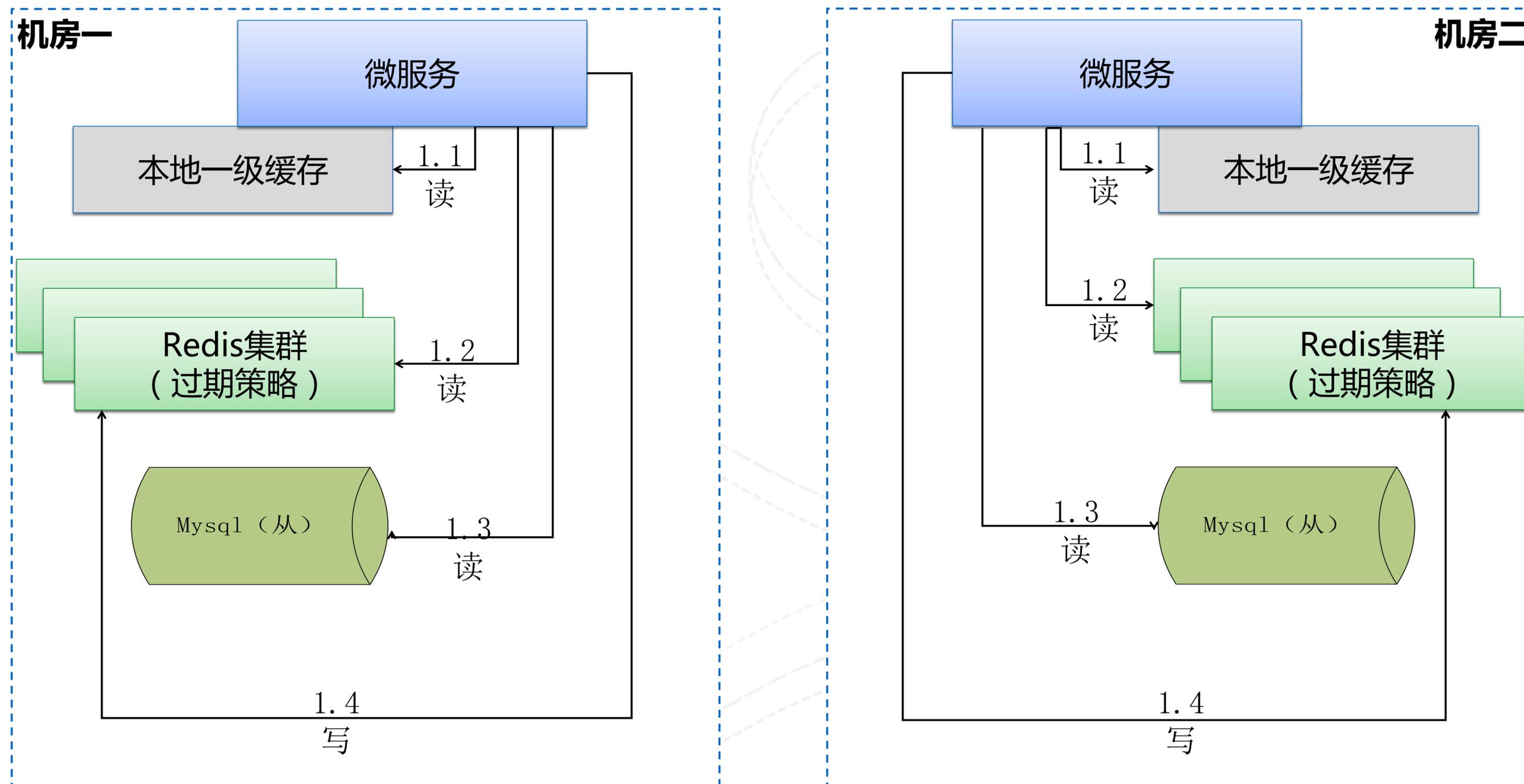
二、基于混合云的系统架构 —— 缓存层

场景二：共享数据加速，强数据一致性（如订单数据）



二、基于混合云的系统架构 —— 缓存层

场景三：共享数据加速、不考虑数据一致性（如商品数据）



二、基于混合云的系统架构 —— 数据层

- 同城双活
由业务层来控制数据的实时性和最终一致性，而不是通过数据同步来保证实时性和一致性。
- 业务层双写，数据异步分发至两个数据中心
任意机房写入的数据通过异步消息的方式分发到另一个机房，以此来保证两个机房数据的最终一致性
- 业务层通过二级查询保证数据的实时一致性
由于业务层双写只能保证数据的最终一致性，无法保证实时一致性，因此，针对具有实时一致性要求的业务场景，我们通过业务层的二级查询来保证。
- 通过重复写入应对单机房故障
当任意机房出现故障时，如果写入的数据还没有分发至另一个机房，则由业务层在可用机房重复写入数据，通过算法来生成相同的id
- 通过failover库为高可用提供双重保险
针对流水型业务数据，在数据库故障时，需要进行主从切换，此时通过业务层将所有数据的读写切换至failover库，主库恢复以后再将流量切回主库。
- 垂直拆分 & 水平拆分

三、服务化

为何选择混合云架构？

基于混合云的系统架构

服务化

自动化

为何要服务化？

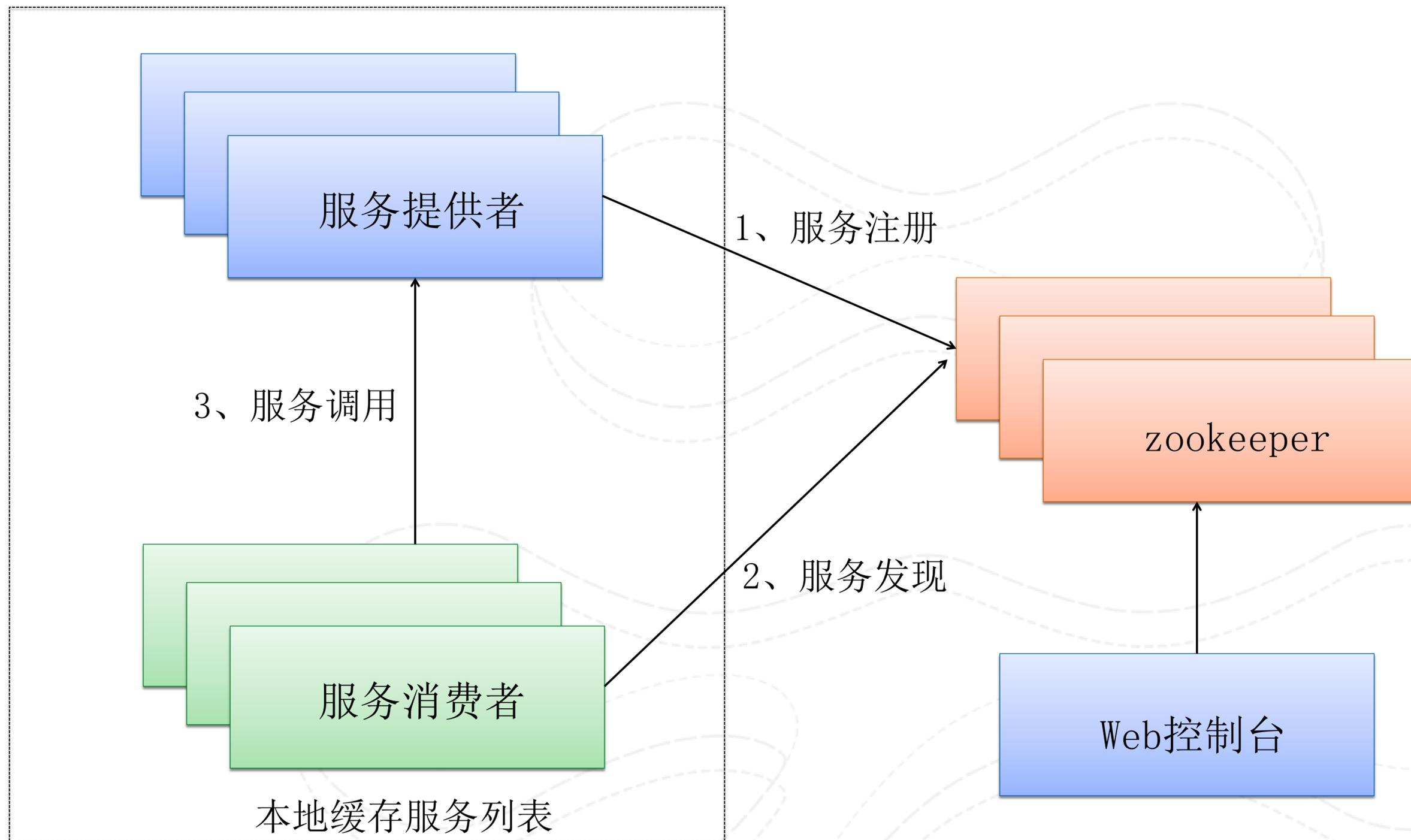
- 系统高度耦合，牵一发而动全身，直接影响到系统可用性
- 业务相互影响，系统很难维护
- 系统逻辑过于耦合，很难进行水平扩展
- 无法通过流控、降级等手段保障系统的可用性
- 由于系统的高度耦合，极易产生雪崩效应

三、服务化



三、服务化

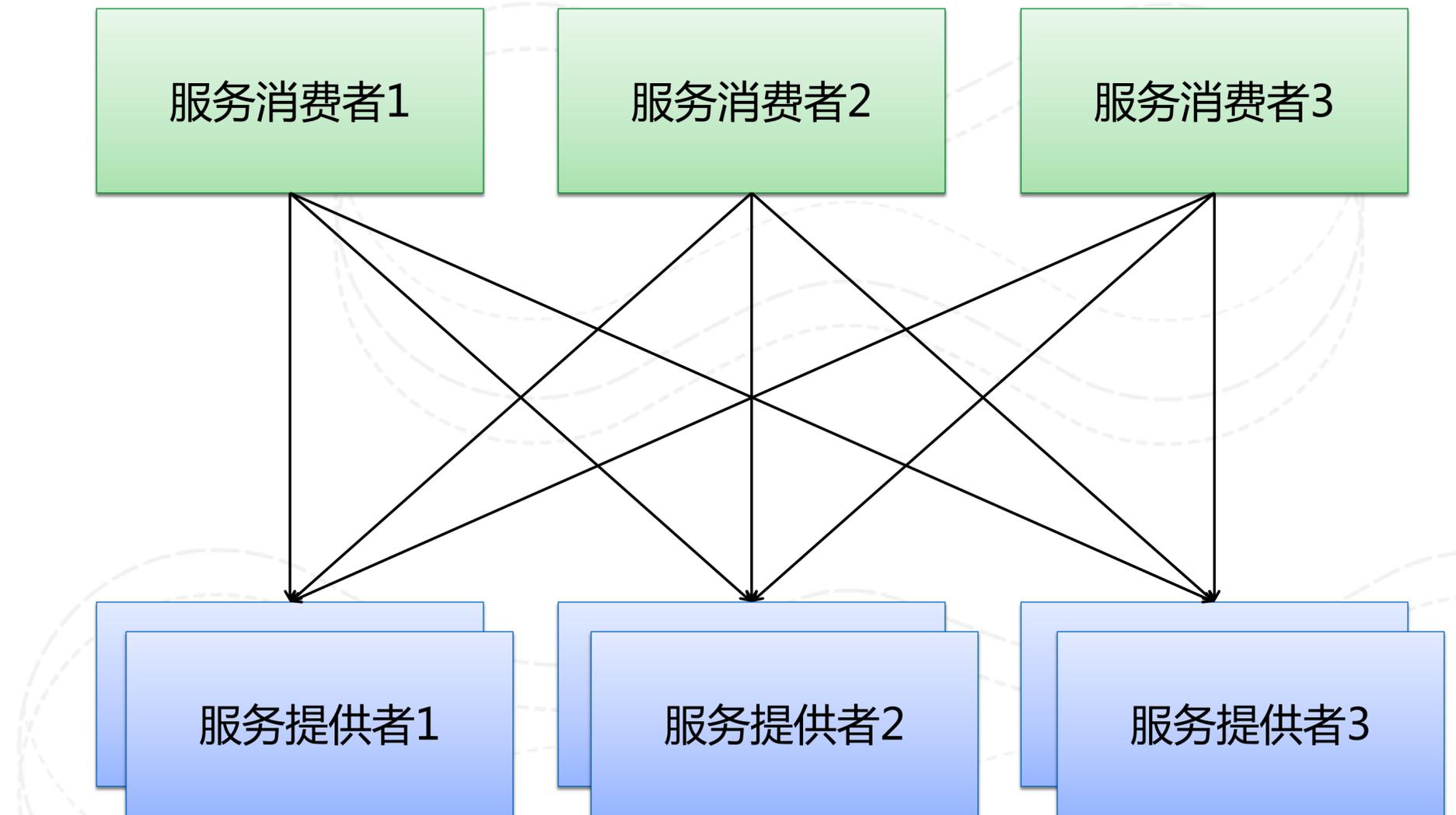
- 服务注册
- 服务发现
- 服务调用



三、服务化

➤ 负载均衡

- 一个服务至少部署两个以上实例
- 基于注册中心的客户端寻址访问模式
- 服务提供者集群动态调整不影响客户端
- 服务消费者负责对服务提供健康检查



➤ 服务降级

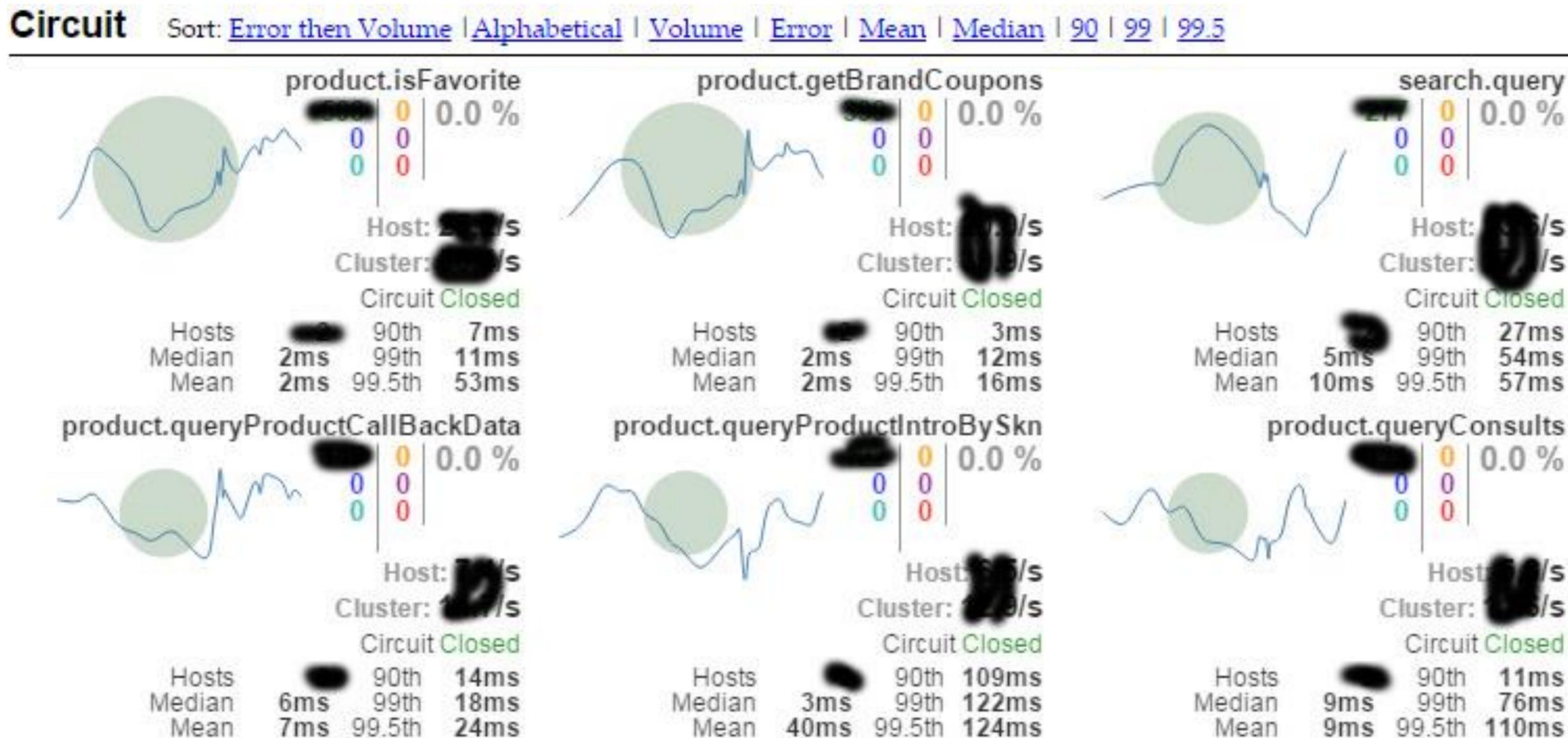
- 可配置服务调用超时时间，当服务调用超时时，直接返回或执行fallback逻辑。
- 服务调用结果分：成功，失败，超时，线程拒绝，短路。请求失败(异常，拒绝，超时，短路)时执行fallback(降级)逻辑。
- 提供熔断器组件，可以自动运行或手动调用，停止当前服务一段时间
- 以上主要是基于Hystrix来完成

➤ 流量控制

- 通过计数器服务限定单位时间内当前服务的最大调用次数（比如600次/分钟）
- 为每个服务提供一个小的线程池，如果线程池已满，调用将被立即拒绝，默认不采用排队. 加速失败判定时间。

三、服务化

- 服务监控
- 性能优化
- 调用链分析



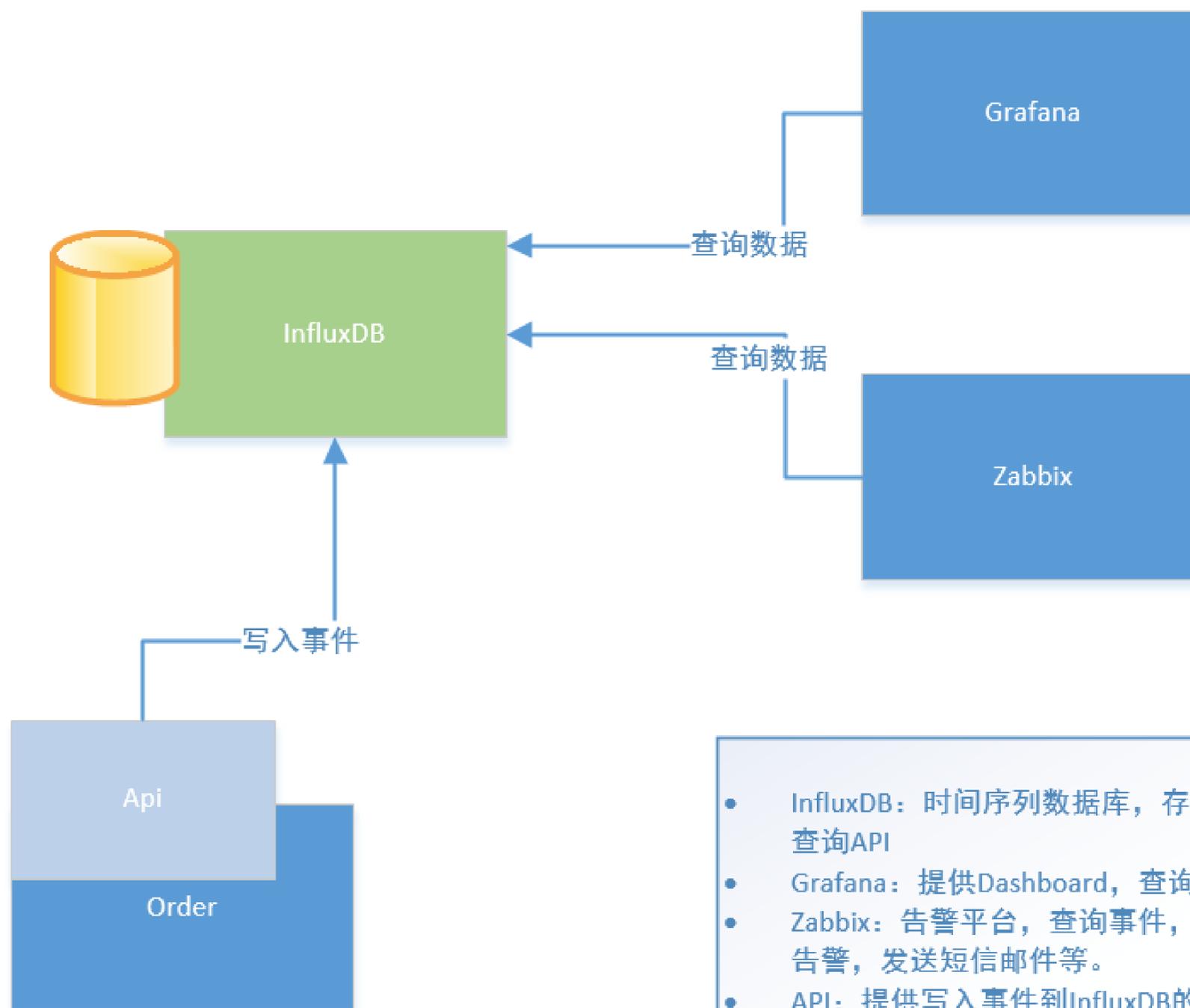
三、服务化

- 服务监控
- 性能优化
- 调用链分析



三、服务化

- 服务监控
- 性能优化
- 调用链分析



- InfluxDB: 时间序列数据库, 存储事件, 并且提供查询API
- Grafana: 提供Dashboard, 查询事件
- Zabbix: 告警平台, 查询事件, 配置trigger, 触发告警, 发送短信邮件等。
- API: 提供写入事件到InfluxDB的api



四、自动化

为何选择混合云架构？

基于混合云的系统架构

服务化

自动化

四、自动化

- 1、自动发布与回滚
- 2、业务实时数据监控
- 3、应用运行状态监控
- 4、系统故障监控
- 5、第三方服务质量监测
- 6、故障快速恢复
- 7、系统容量规划



Thanks!



yq.aliyun.com

云栖社区，我们的IT江湖