



全球软件开发大会【上海站】

饿了么异地多活

架构和基础组件实现




极客时间

重拾极客精神 · 提升技术认知

每天10分钟,邀请顶级技术专家,为你传道授业解惑。



扫一扫,试读专栏

主办方 **Geekbang**  **InfoQ**
极客邦科技

ArchSummit

全球架构师峰会 2017

12月8-9日 北京·国际会议中心



APSEC 2017



Geekbang
极客邦科技

InfoQ

APSEC 2017

24th Asia-Pacific Software Engineering Conference
4-8 December 2017, Nanjing, Jiangsu, China

12月4-8日
中国南京



了解详情

AiCon

全球人工智能技术大会 2018

助力人工智能落地

2018.1.13 - 1.14 北京国际会议中心



扫描关注大会官网

第一部分：为什么要做多活 

 第三部分：多活 5 大基础组件

第二部分：多活的设计思路




致谢



阿里巴巴集团 毕玄



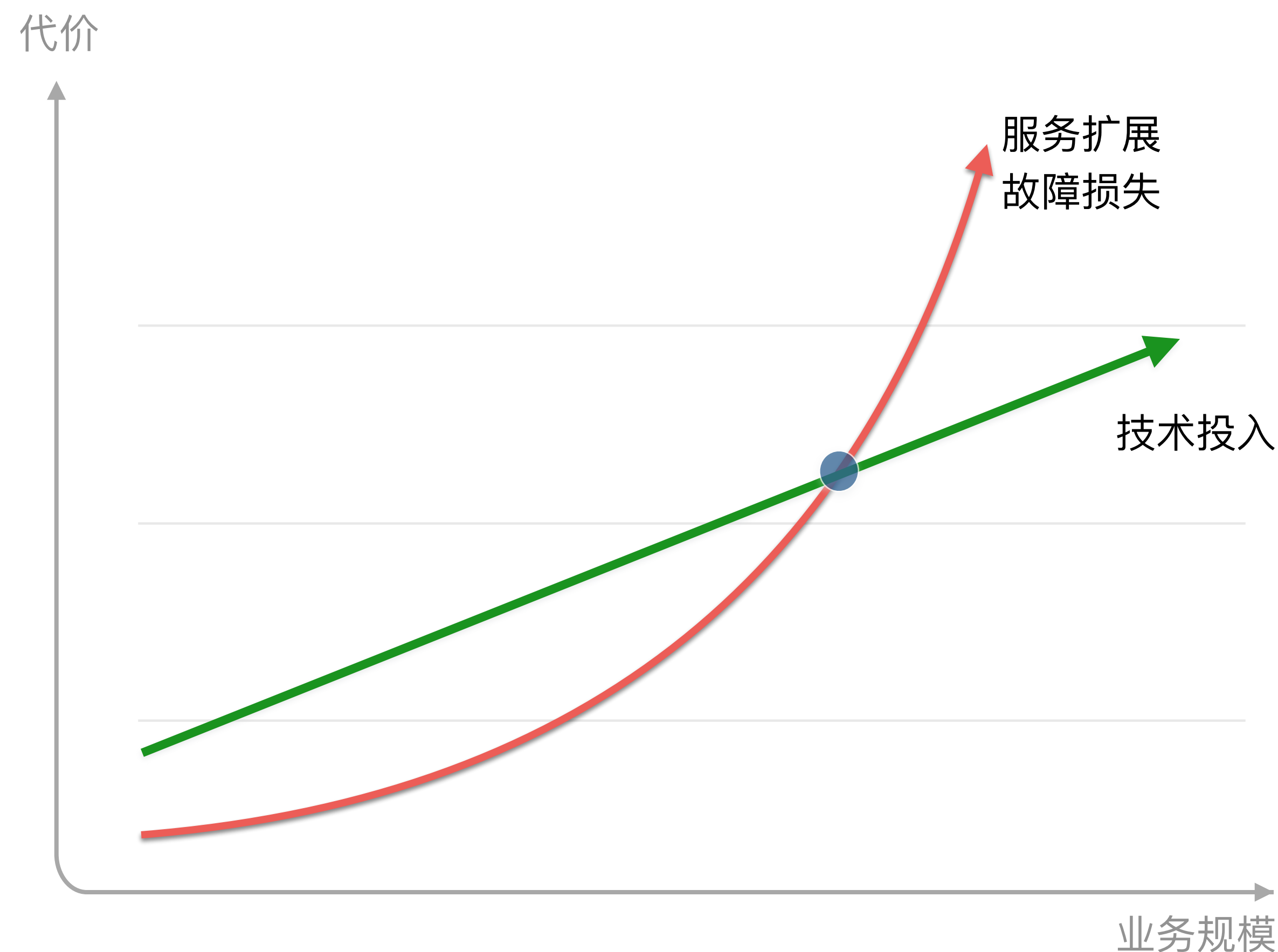
左耳朵耗子 陈皓

多活目前状态



饿了为什么要做多活?

- 扩展性** 1
服务可以扩展到多个机房
- 高可用** 2
能够应对整个机房级别的故障



多活的主要挑战

- 物理距离导致的传输延迟 (30ms)
- 这个小小的延迟决定了我们需要巨大的改造
- 跨地域网络的可靠性更低
- 定义机房的边界, 避免跨机房调用

L1 cache reference	0.5 ns	
Branch mispredict	5 ns	
L2 cache reference	7 ns	
Mutex lock/unlock	25 ns	
Main memory reference	100 ns	
Compress 1K bytes with Zippy	3,000 ns	= 3 µs
Send 2K bytes over 1 Gbps network	20,000 ns	= 20 µs
SSD random read	150,000 ns	= 150 µs
Read 1 MB sequentially from memory	250,000 ns	= 250 µs
Round trip within same datacenter	500,000 ns	= 0.5 ms
Read 1 MB sequentially from SSD*	1,000,000 ns	= 1 ms
Disk seek	10,000,000 ns	= 10 ms
Read 1 MB sequentially from disk	20,000,000 ns	= 20 ms
北京到上海的网络延时.....	30,000,000 ns	= 30 ms
Send packet CA->Netherlands->CA	150,000,000 ns	= 150 ms

同城多活 or 异地多活

	同城多活	异地多活
整体投入	高（机房投入 + 同城专线）	很高（机房投入 + 异地专线）
实现复杂度	低（依赖垮机房调用）	高（需要减少机房间的交互，清理调用边界）
可以扩展到多机房	中（只能在同城增加机房）	高（可以在全国选择机房，甚至扩展到全球）
服务可用性	低（降低现有可用性）	高（可以应对机房级故障）
对现有架构的影响	低（跨机房调用）	高（业务需要改造）
对服务质量的影响	无影响	无影响

业务特点



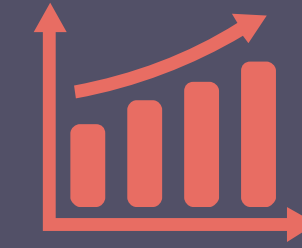
基本原则



服务划分



流量路由



第二部分：多活的设计思路



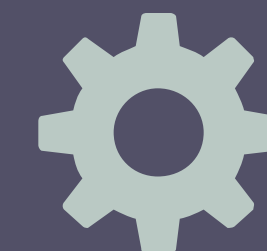
数据复制



一致性保证



业务改造

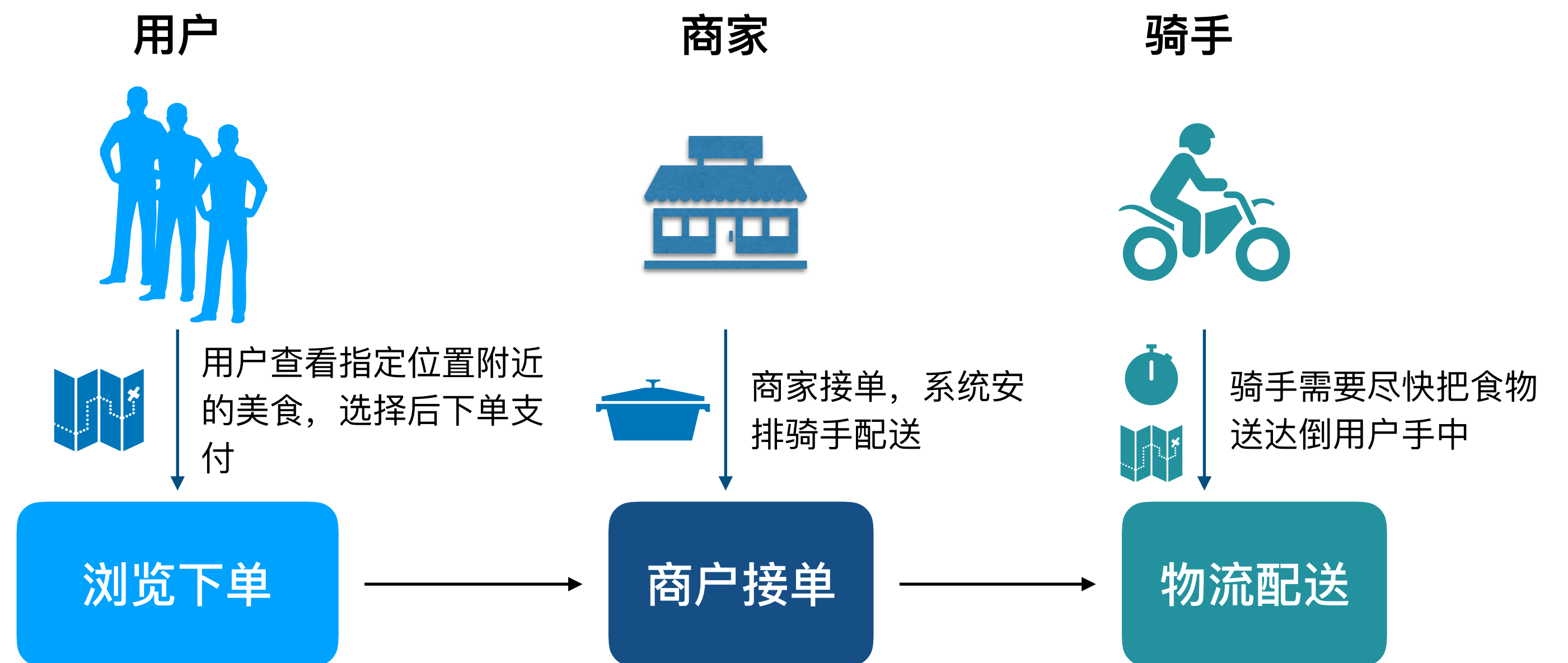


兜底和问题预防

饿了么的业务特点

饿了么的业务特点决定了异地多活的设计：

- 3个最重要的角色，用户、商家和骑手
- 用户就近找到食物，下单并支付
- 商家接单并开始制作食物
- 骑手到店取食物，并配送到客户手上
- 有严格的时间要求，必须在规定时间内完成
- 饿了么的业务有两个核心特性，**地域性**和**实时性**



饿了么异地多活的基本原则

1 业务内聚

- 一个订单的履单过程在一个机房中完成

4 业务可感

- 业务团队修改逻辑，能够识别出业务单元的边界，只处理本单元的数据
- 打造强大的业务状态机，发现和纠正错误

2 可用性优先

- 优先保证系统可用，让用户可以下单吃饭，
- 容忍暂时数据不一致，事后修复

3 保证正确性

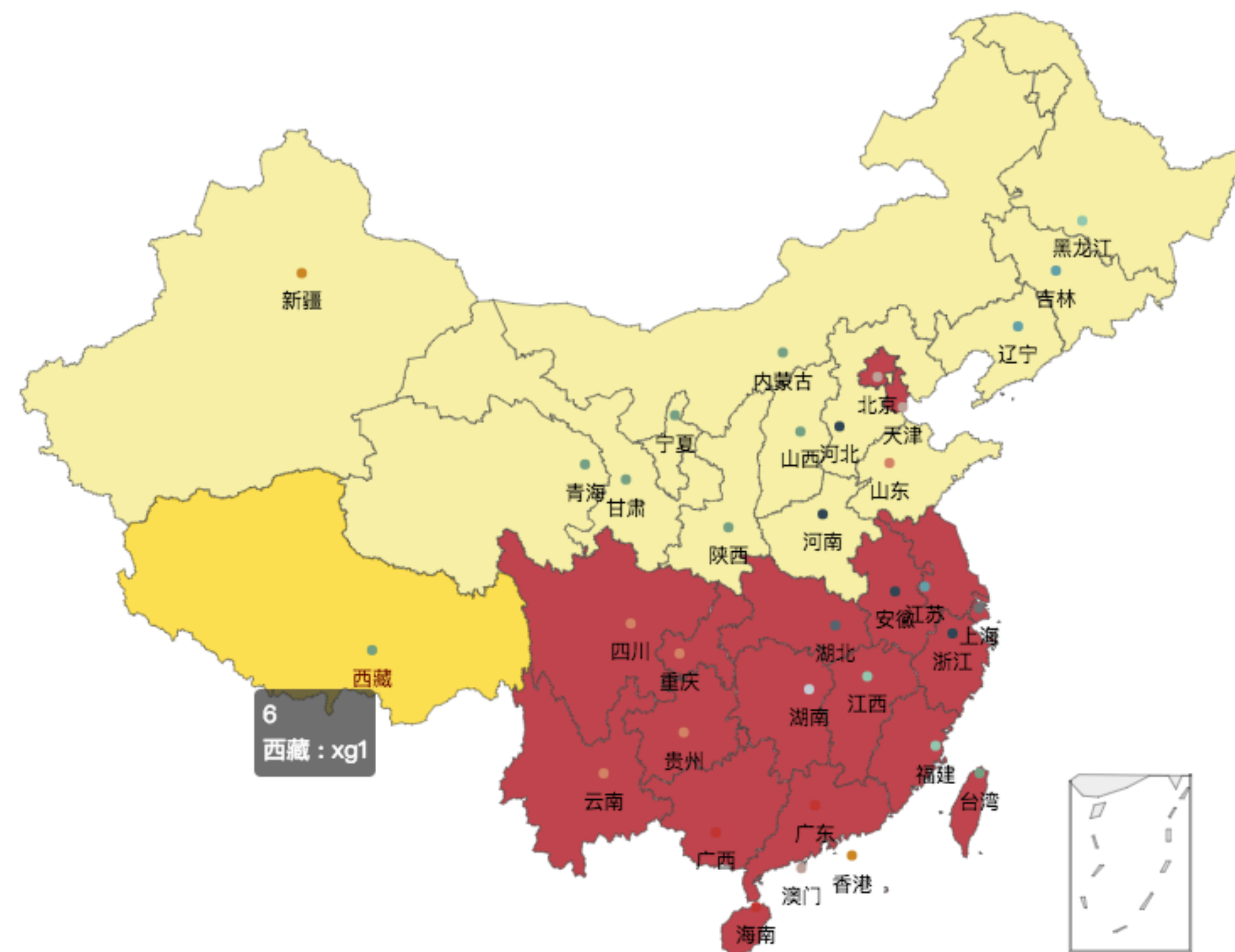
- 在确保可用的情况下，需要对数据做保护以避免错误



服务划分

■ wg1
■ xg1

- 对服务进行分区，让用户，商户，骑手能够正确的内聚到同一个 ezone 中
- 一个订单的履单流程在一个机房完成
- 基于地理位置作划分
- 每个分片（Sharding）有一个确定的地理围栏
- 保证时效，对网络故障不敏感
- 服务划分方法是我们方案中比较特殊的地方
- 特殊性来源于饿了么的业务特点



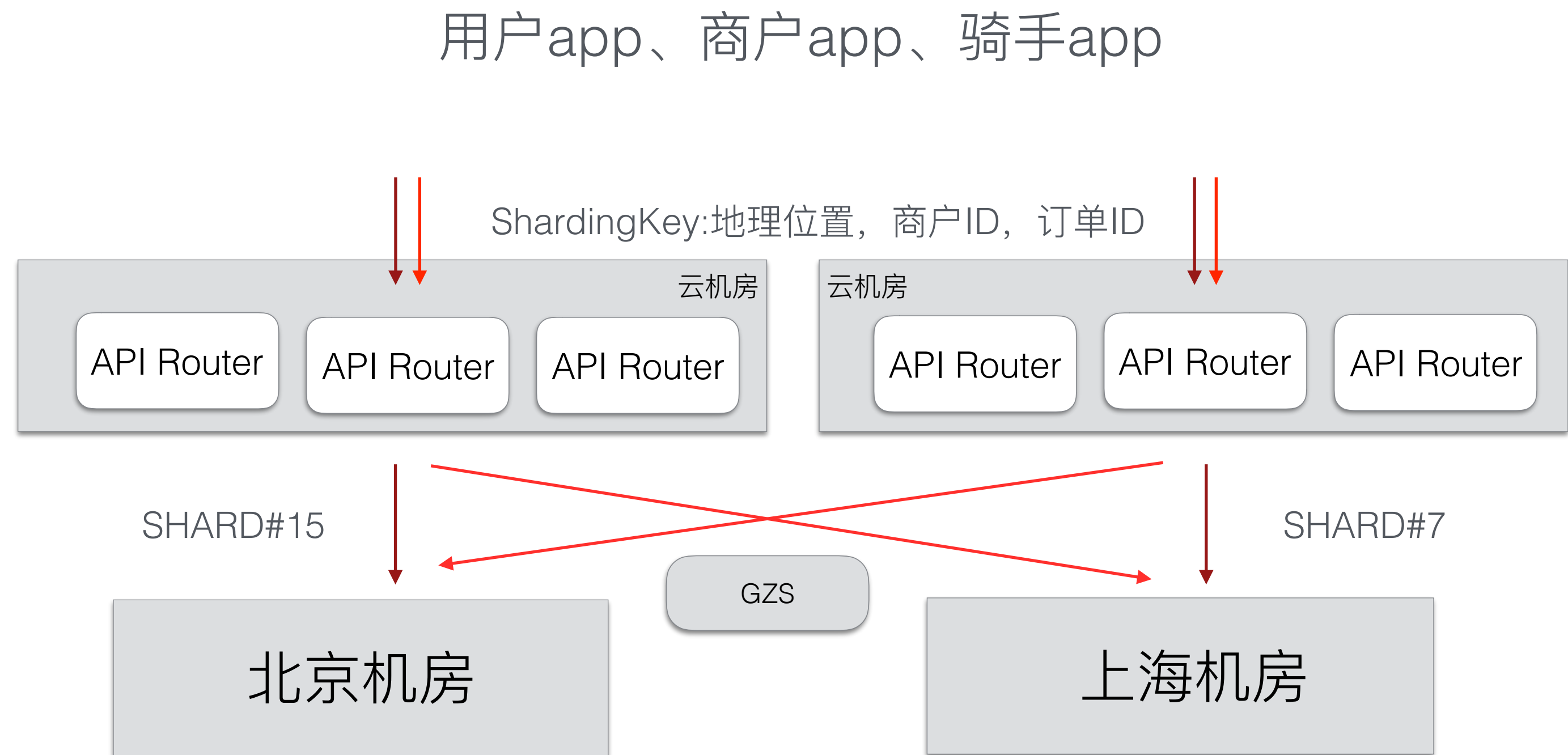
地理划分的常见问题

- 问题一：按照什么地理规则划分，才能保证让地理上接近的用户被划分到同一个机房？大体基于行政区划，结合现有数据分析，做一些局部调整
- 问题二：用户是会动的，如果用户从北京到了上海，那么划分规则应该怎么应对？各个机房全量数据，用户的数据在另外一个机房只有1s的延迟，不受用户移动影响
- 为什么不简单点，按照用户的ID来切分？参考饿了么业务两个核心特性，地域性和实时性，用ID划分，则难以保证这两点。



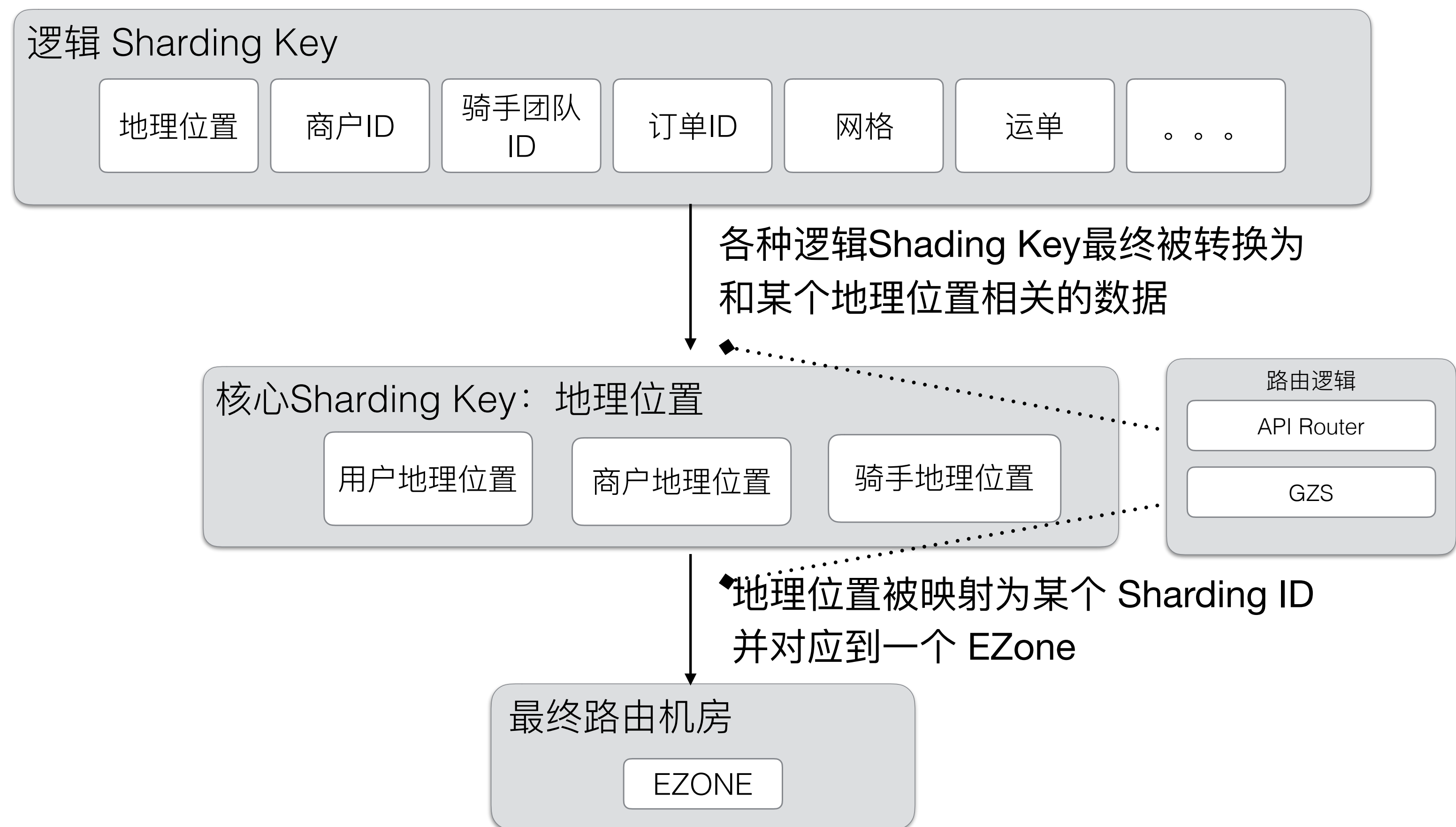
流量路由

- 为每个请求都带上了分流标签 (Sharding Key)
- API Router 把Sharding Key转换为对应的 Shard ID
- Shard ID 被映射到对应的机房
- 由GZS (Global Zone Service) 统一维护地理围栏, 映射关系, 切换动作通知...



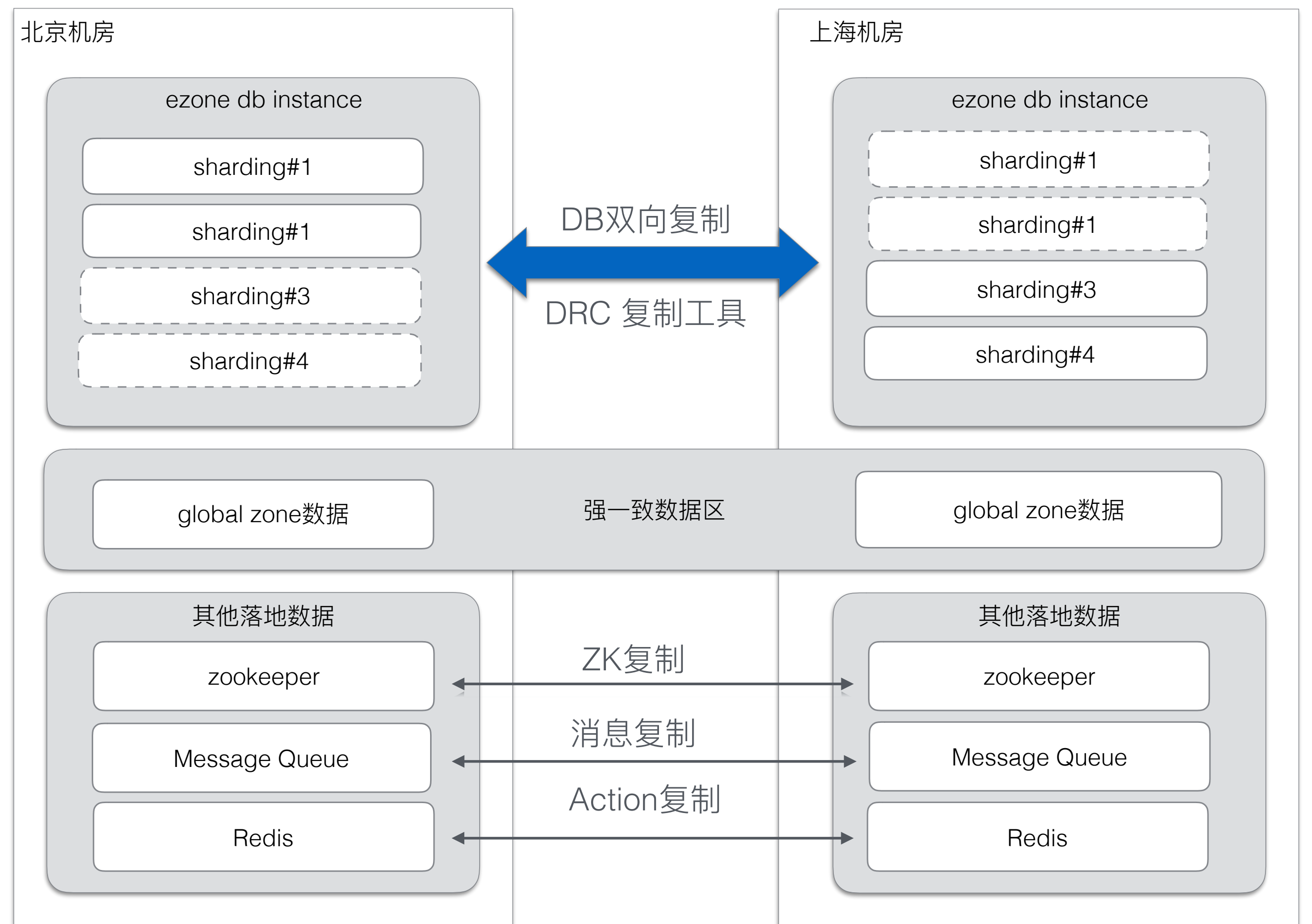
灵活的逻辑路由方案

- 并不是所有的调用都能直接关联到某个地理位置上
- 减少业务改造成本
- 中间件支持逻辑Key到物理Key的转换



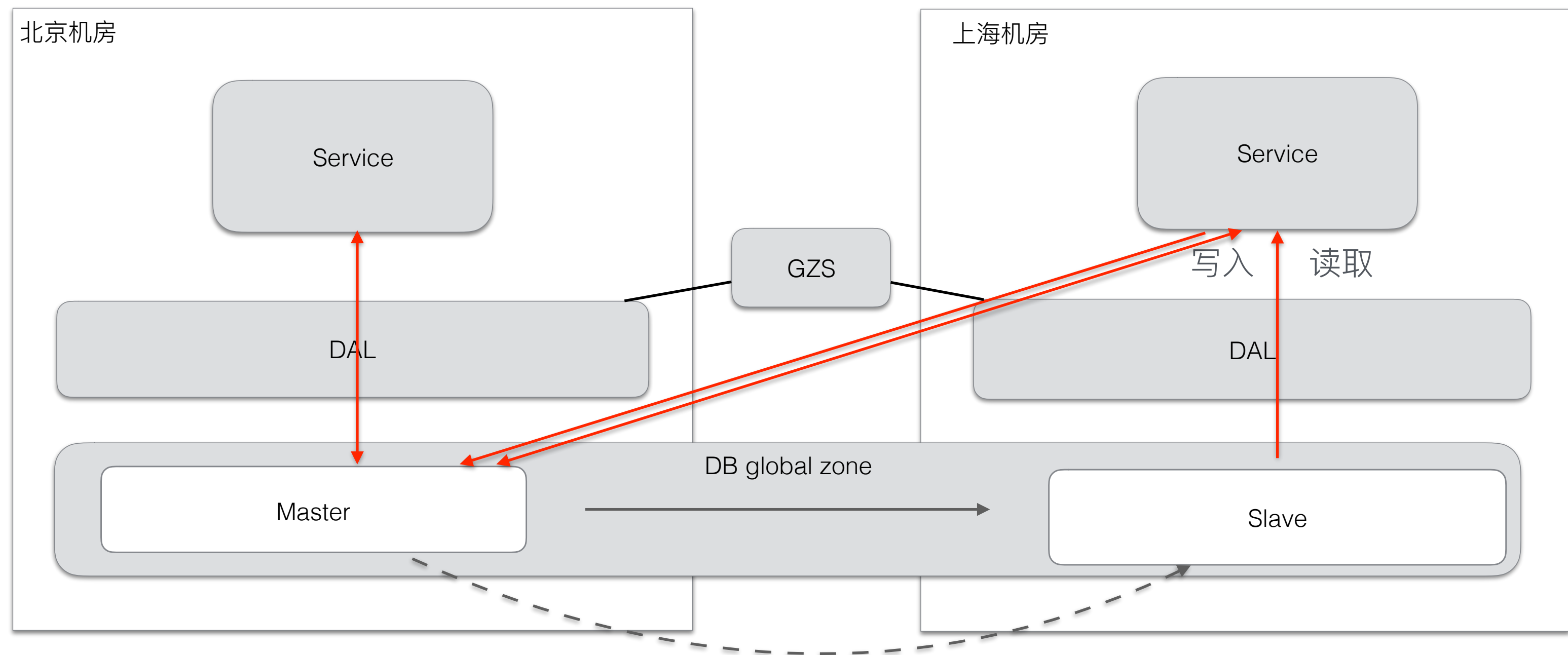
数据复制

- MYSQL 数据实施双向复制 (DRC)
- ZooKeeper 双向复制
- 消息队列复制

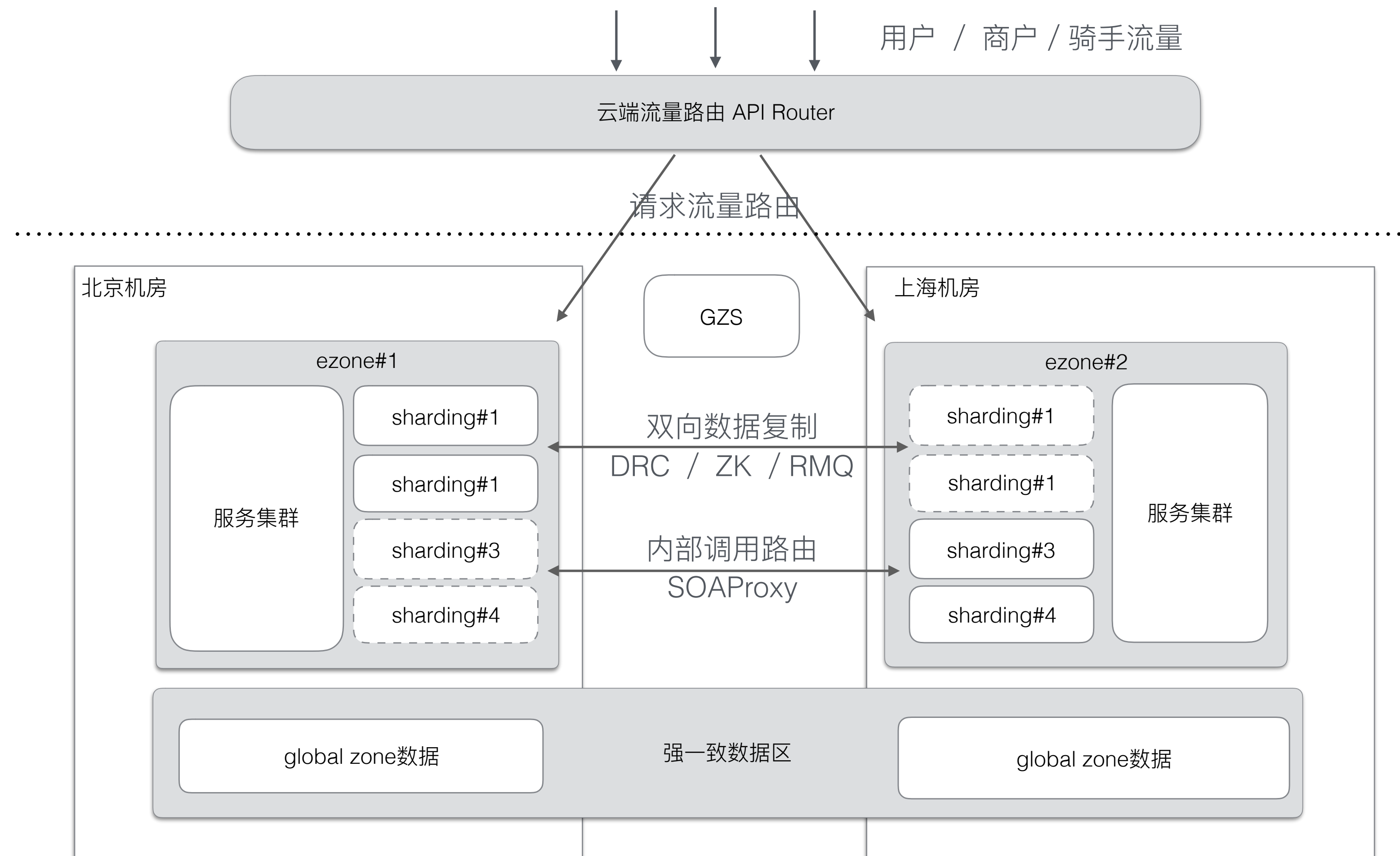


强一致保证 (Global Zone)

- 个别一致性要求很高的场景
- Global Zone是一种跨机房的读写分离机制
- 基于数据库访问层 (DAL)
- 业务无感知



多活整体结构



业务改造

1

后台任务

后台任务只处理本 ezone 的数据。

2

调用链改造

改造调用链，让 Sharding 信息能够在整个调用过程传递

3

切换感知

可以在发生切换时，执行特定的逻辑，触发一些动作

4

兜底修复

业务需要准备一些数据修复逻辑，在万一发生不一致时，手工或者自动纠正数据。

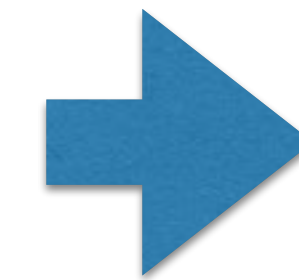
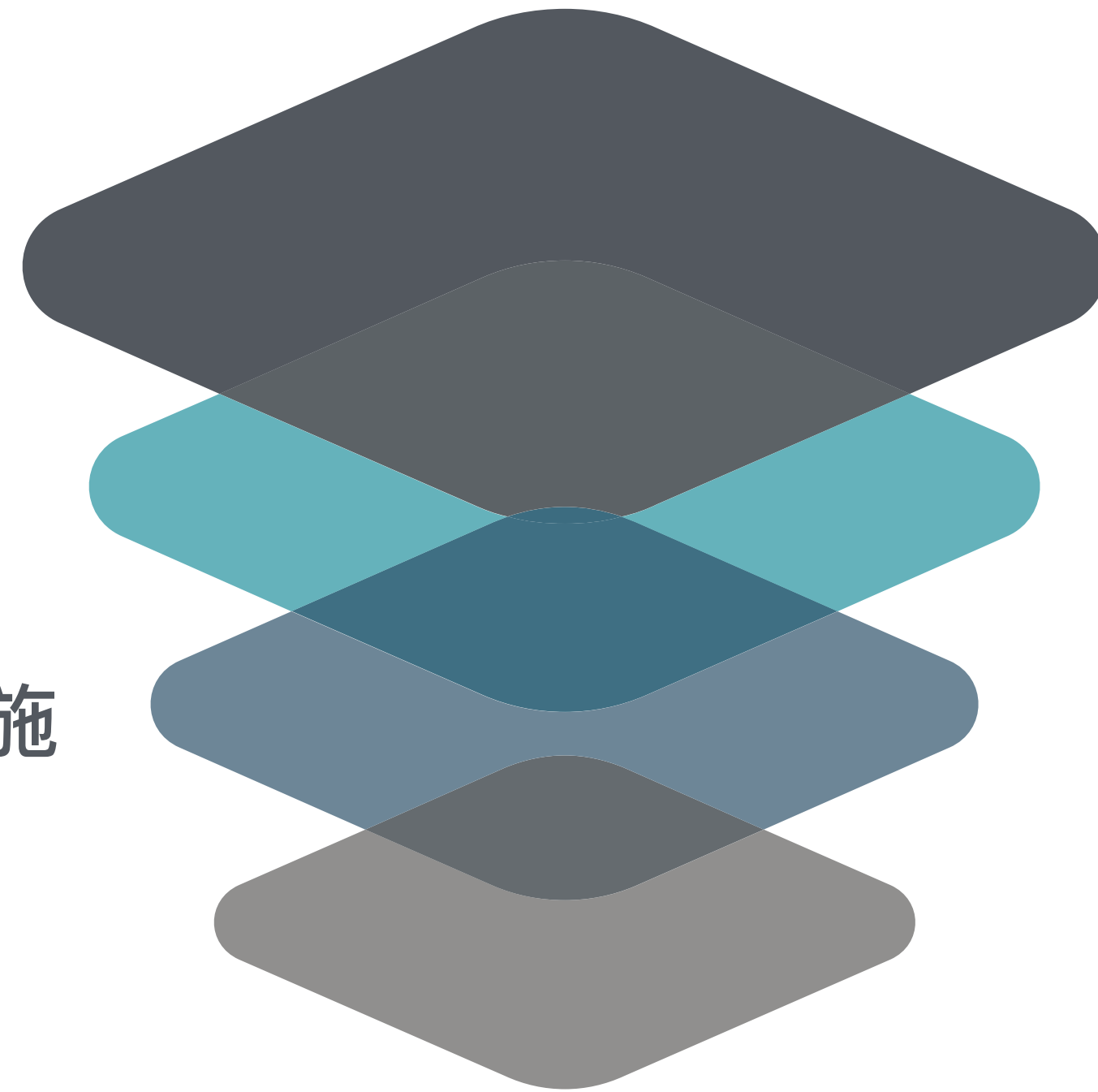


业界常用异地多活方案

基本原则

全公司各团队协作实施

各种异常 Case 处理



饿了么多活

第三部分：多活5大基础组件



API Router
流量路由和分发



GZS
多活信息的全局协调器



SOA Proxy
SOA 调用路由



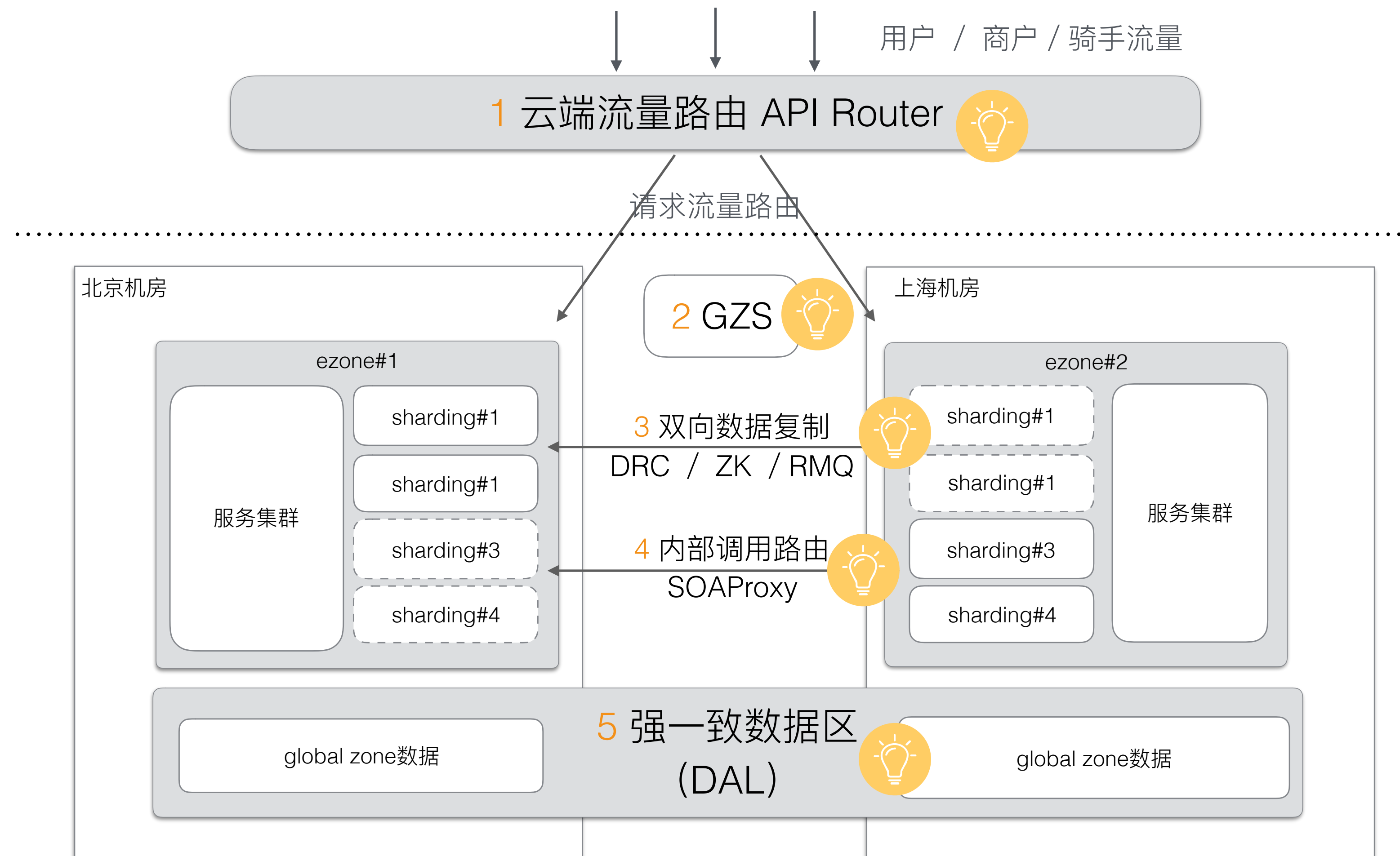
DRC
数据复制工具



DAL
数据访问和兜底

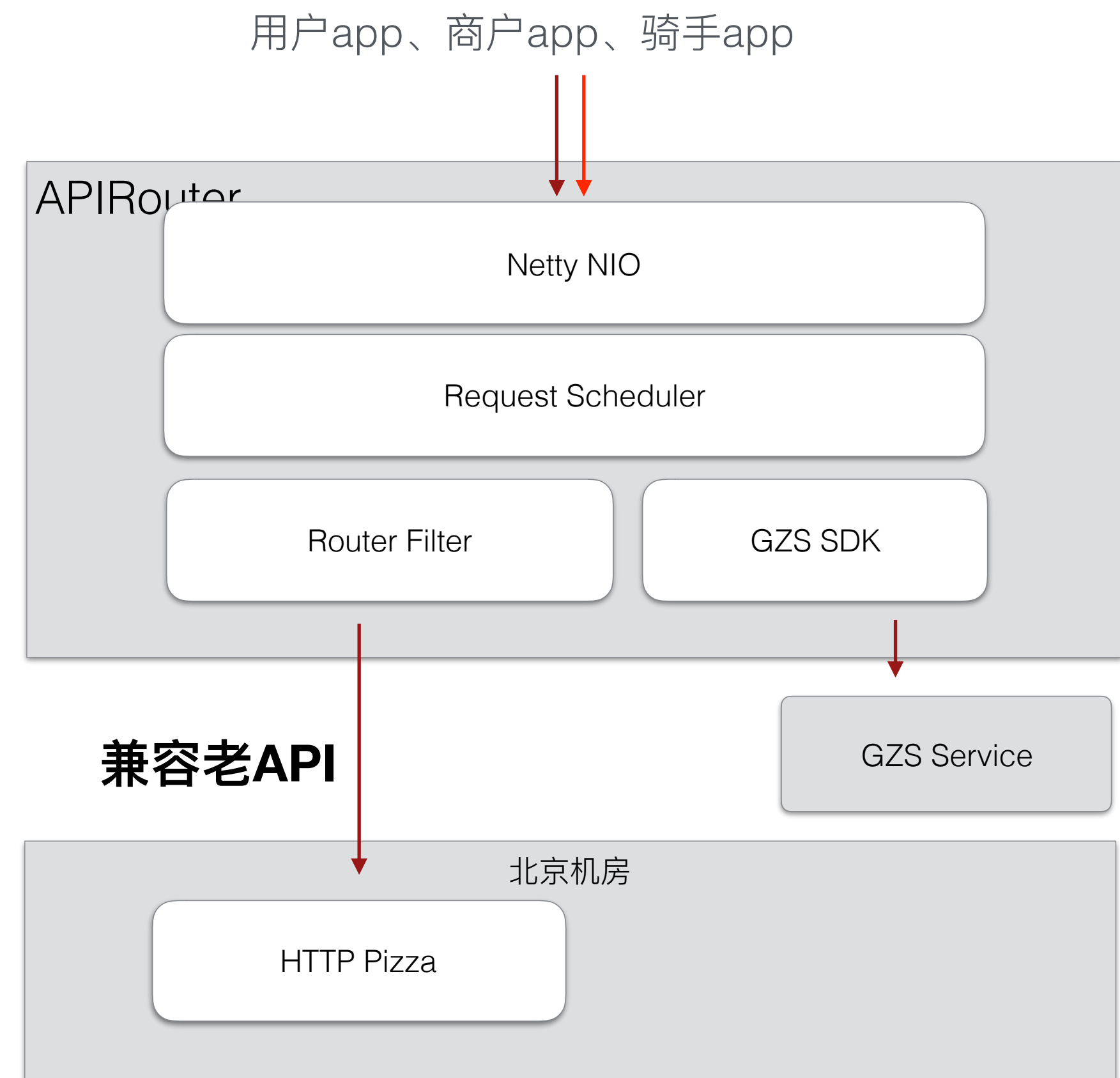


多活5大基础组件



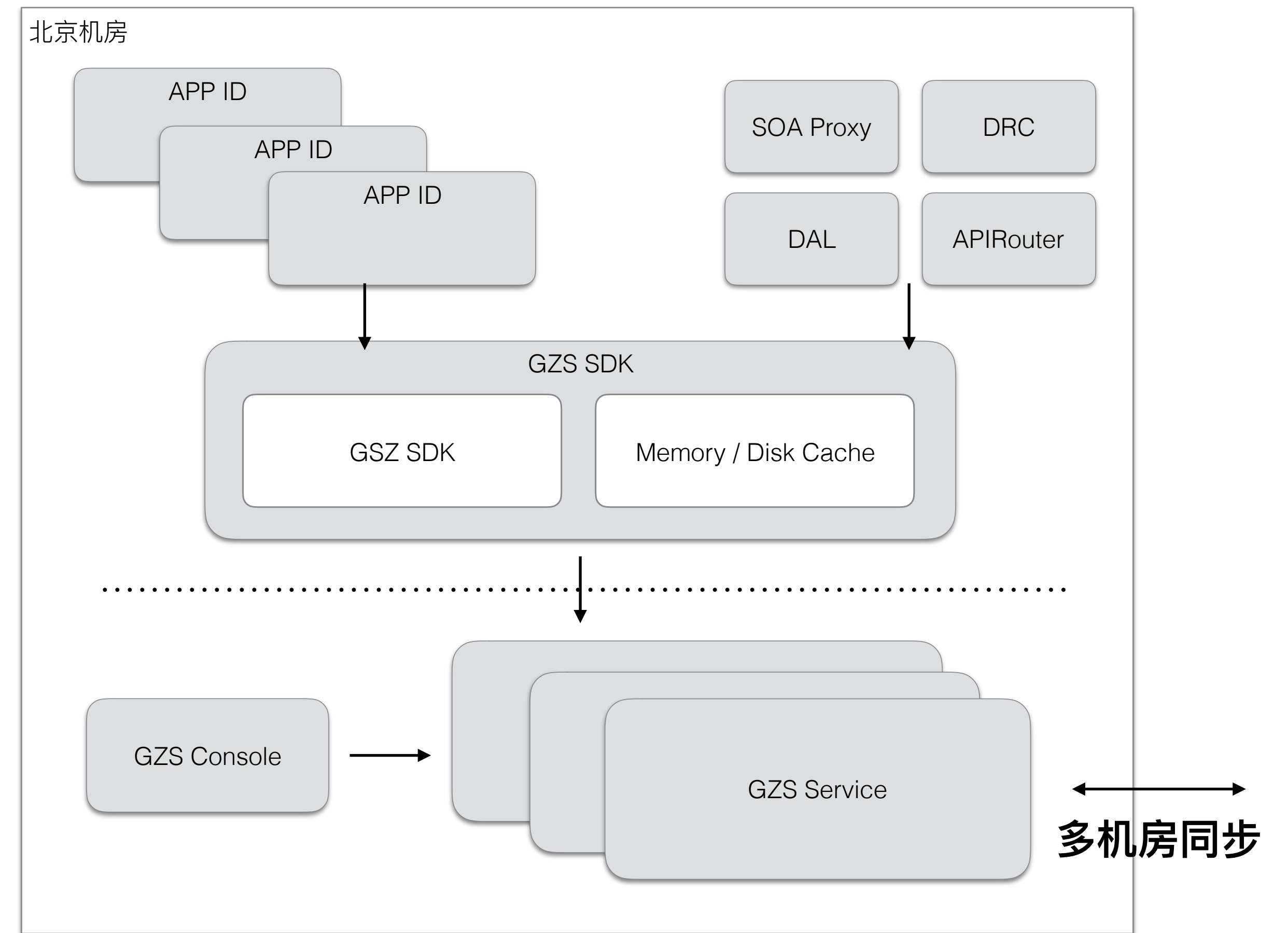
API Router

- API Router部署在云环境中
- 作为HTTP API流量入口
- 识别归属shard，并转发到对应的 ezone
- 兼容老的无法修改的API
- API Router支持多种路由键



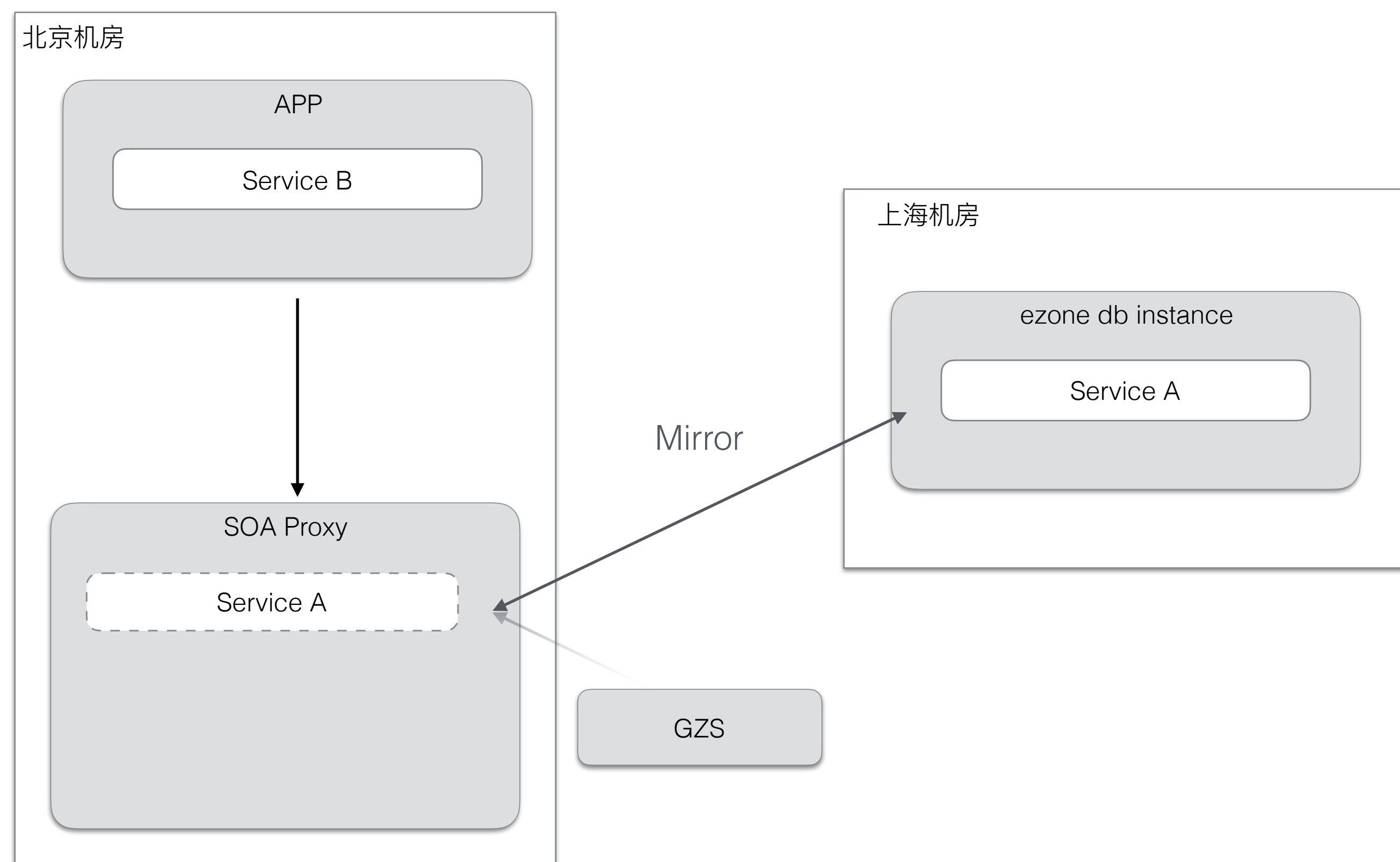
Global Zone Service (GZS)

- GZS 维护着整个多活的路由表
 - 地理围栏信息,
 - shard 到 ezone 的归属信息,
 - 商铺ID / 订单ID 等路由逻辑层到的映射关系
 - . . .
- GZS 通过在 SDK 端建立 Cache
- 映射关系发生变化实时推送
- 切换机房的操作也在 GZS 控制台中完成



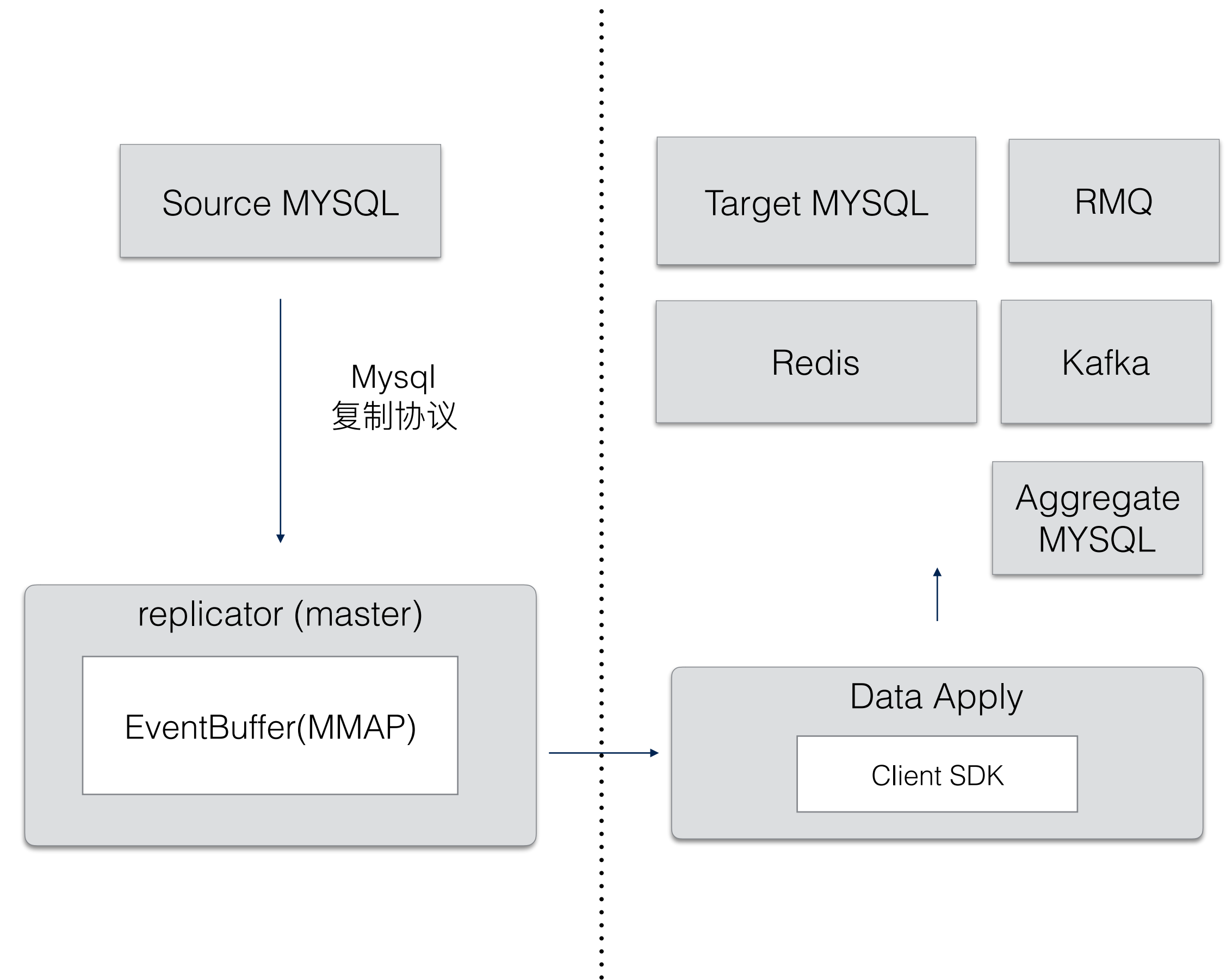
SOA Proxy

- SOA Proxy 实现了对 SOA 调用的路由，执行和 API Router 相似的逻辑，但只用在机房之间进行通信的场景。
- 业务使用 SOA Proxy 需要对代码做一些修改，把路由信息加入到调用的上下文中。



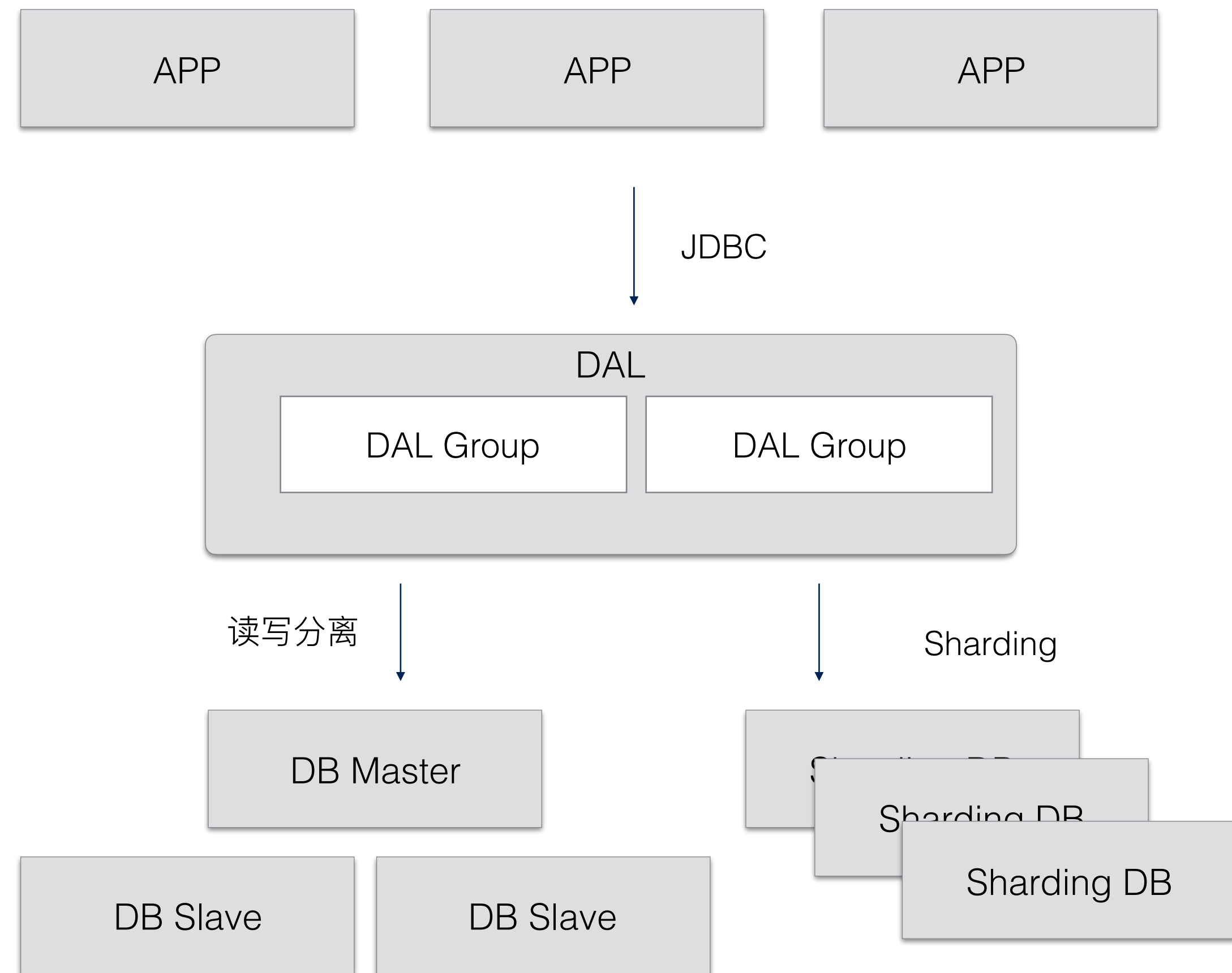
Data Replication Center (DRC)

- 实时双向复制Mysql 数据
- 跨机房延时在1s以内
- 提供了基于时间的冲突解决方案
- 还对外提供了数据变更的通知
- 除了DRC，我们还有 ZK复制工具
- RMQ 复制工具
- Redis复制工具



Data Access Layer (DAL)

- DAL 支撑了 Global Zone 功能
- 路由错误兜底，保证数据正确性
- 写入错误保护





关注QCon微信公众号
获得更多干货!

Thanks!

INTERNATIONAL SOFTWARE DEVELOPMENT CONFERENCE

主办方: **Geekbang**  **InfoQ**
极客邦科技

