

饿了么多活架构分享

陈忠克

zhongke.chen@ele.me

2017年6月23日



饿了么简介

- 愿景：以创新科技打造全球领先的本地生活平台
- 主营外卖、商超、即时配送、餐饮供应链等业务
- 外卖覆盖城市数1400+，餐厅数100万+，日订单900万+，日运单450万+，骑手300万+



用户浏览、下单、支付



商家接单、备餐、呼叫配送



骑手取餐、送餐

饿了么多活的背景

- 北京IDC: 4000+服务器
- 上海IDC: 建设中
- 峰值订单: 1000+/s, 规划容量4000+
- 微服务: 2000+ (Python, Java, Go)
- 灾备: 3000+云主机



饿了么为什么做多活?

单机房+灾备

- 容灾能力差
- 扩容受限于单机房容量
- 灾备资源巨大浪费



多活

- 提升容灾能力 (任何一个机房可挂, 流量可切)
- 多机房扩展能力 (不受限于任何一个机房的物理限制)
- 充分利用资源 (如果有n个同等大小的机房, 理论上利用率可达 $(n-1)/n$)



饿了么做多活的基本思路

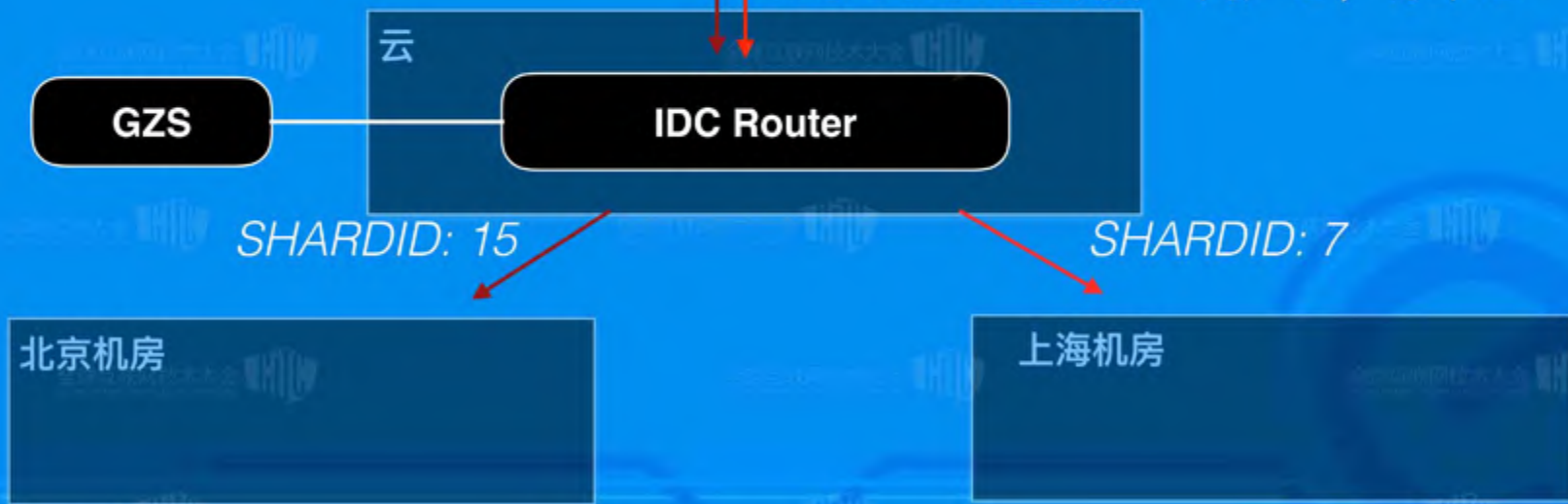
- 同城多活
 - 选择性数据同步
 - 一致性优先
 - 不同类型用户到不同机房，
一笔交易跨多个机房
- VS
- 异地多活
 - 全量数据同步
 - 可用性优先
 - 交易不跨机房

第一步：双机房异地多活！

多活设计：分流

用户app、商户app、骑手app

X-SHARD: 经纬度 商户ID, 订单ID



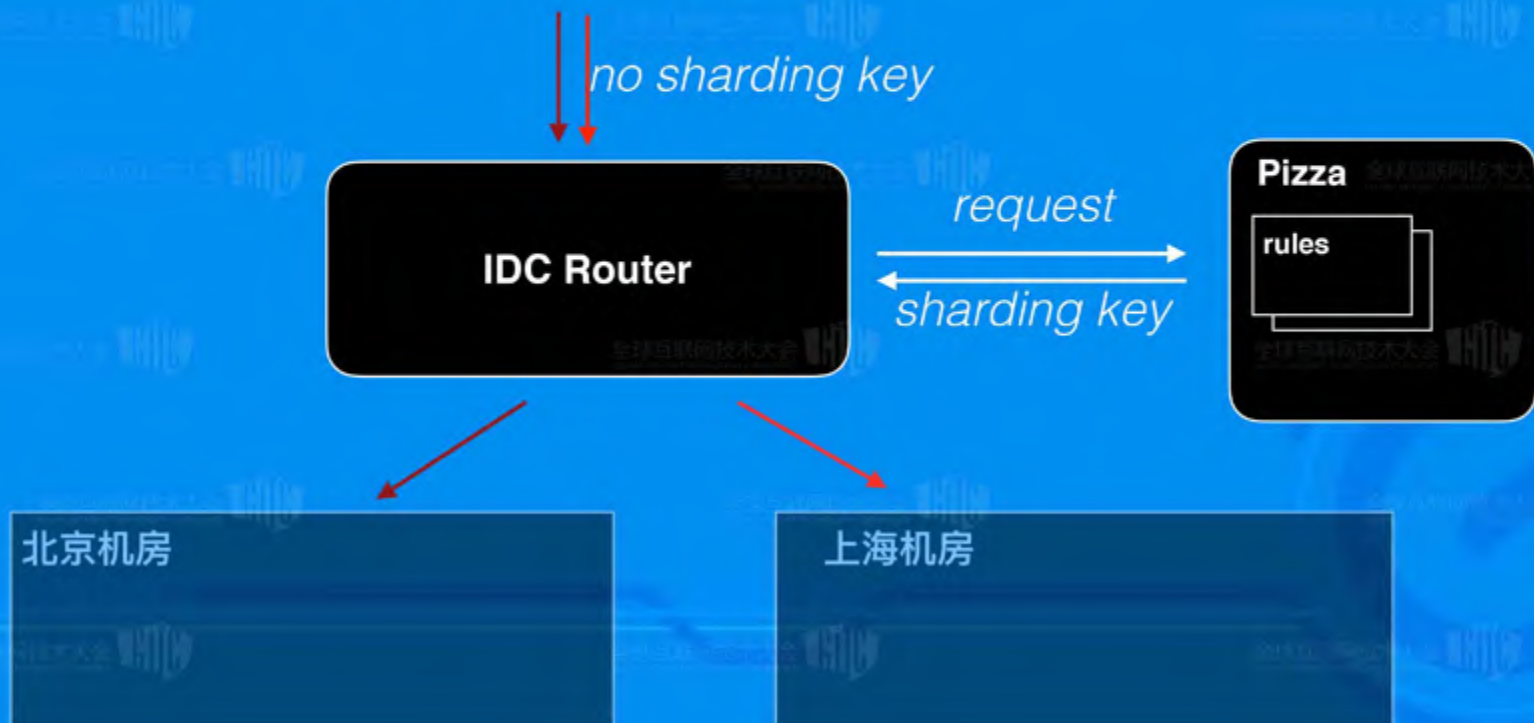
多活设计：分片规则

- 以省份（直辖市）为分片单位
- 南方省份划归南方机房
- 北方省份划归北方机房

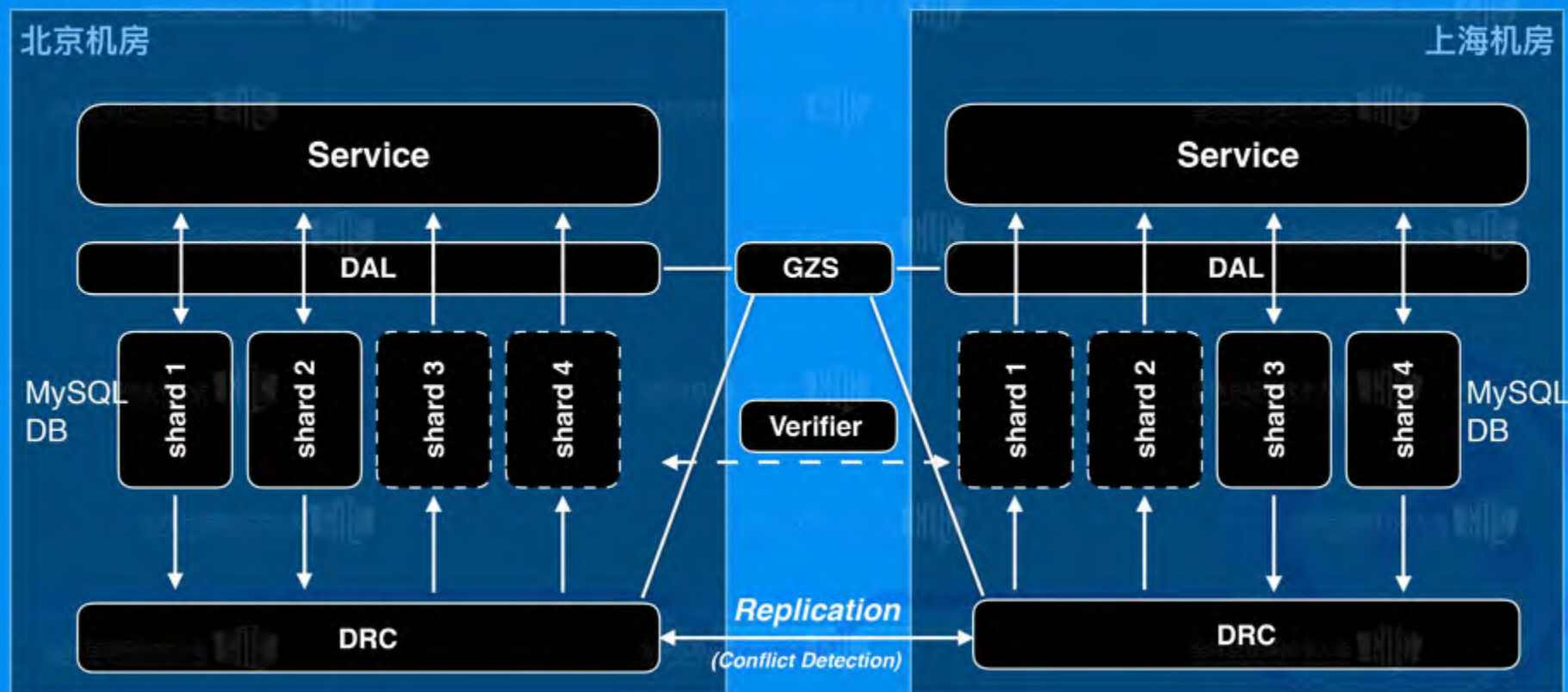


多活设计：老版本应用兼容

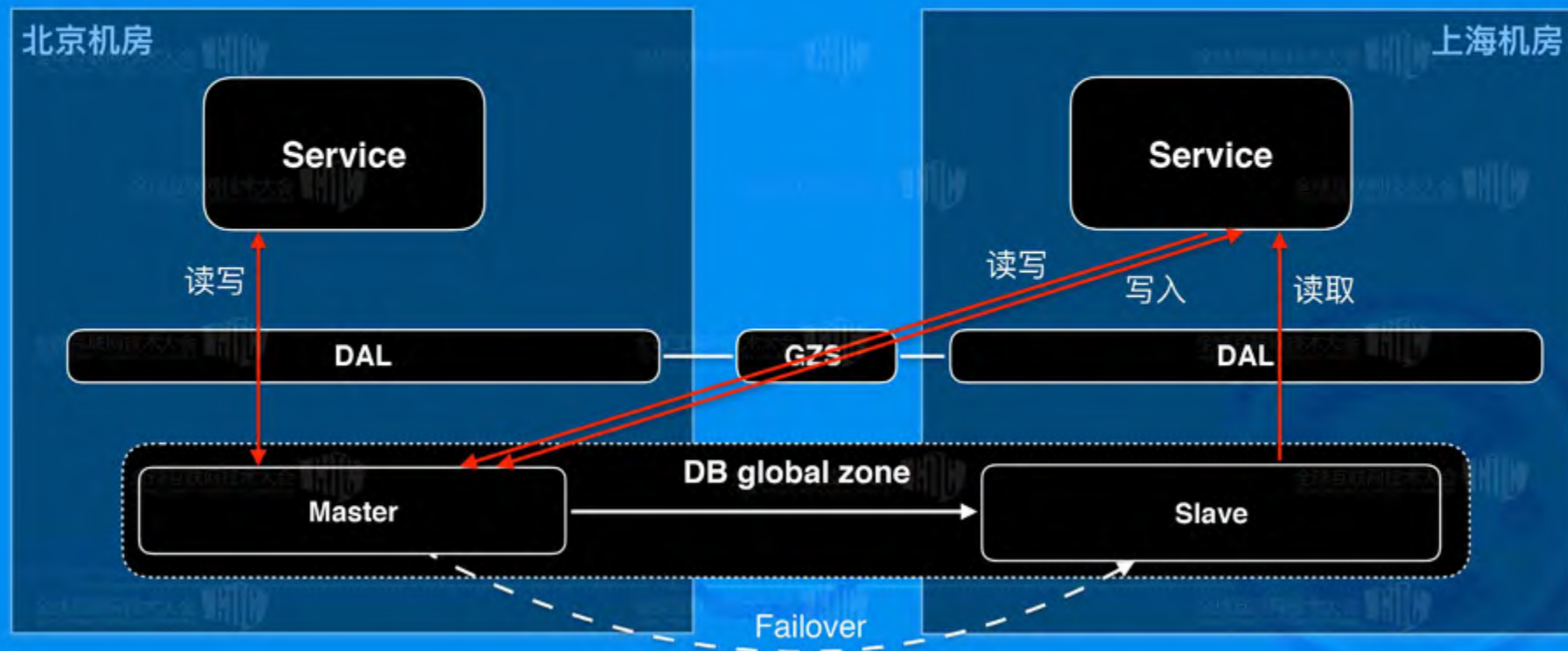
老版本应用，开放api应用



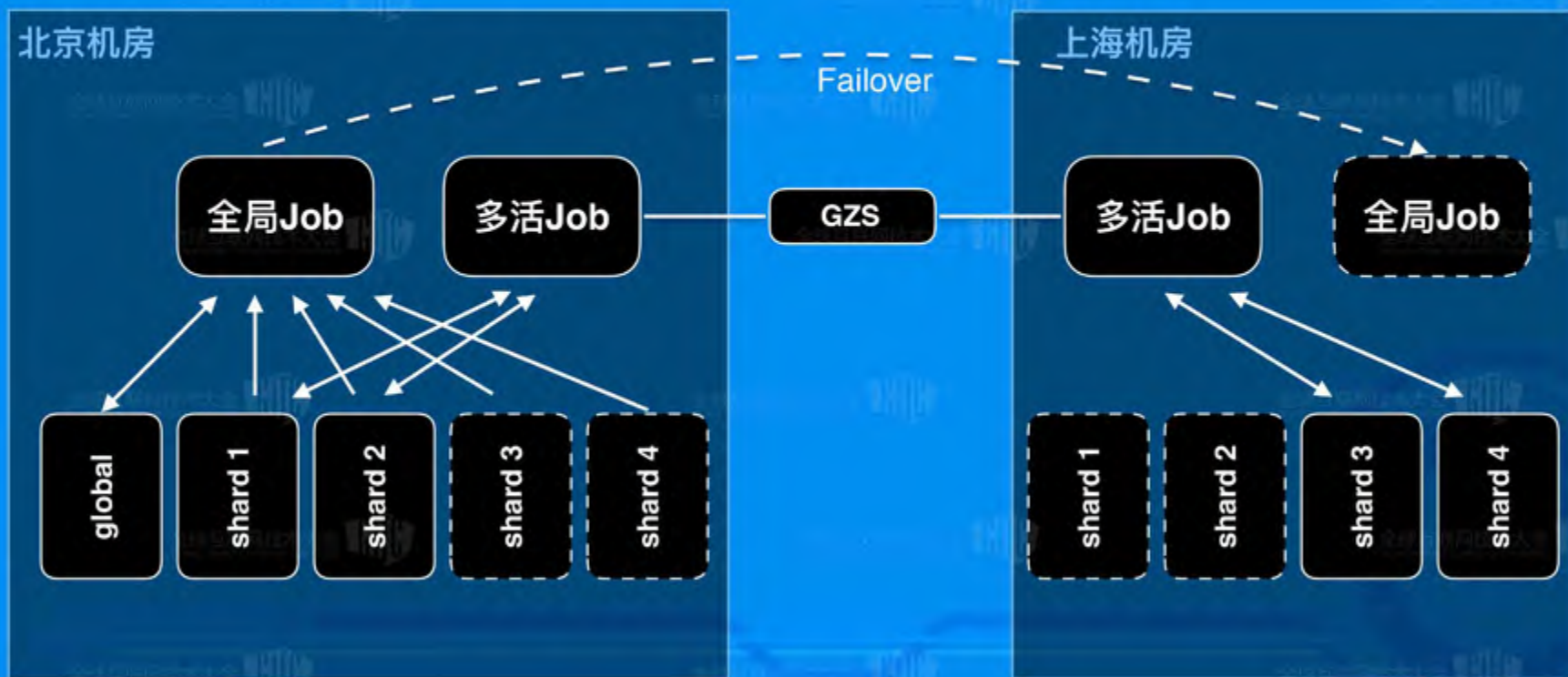
多活设计：数据库同步



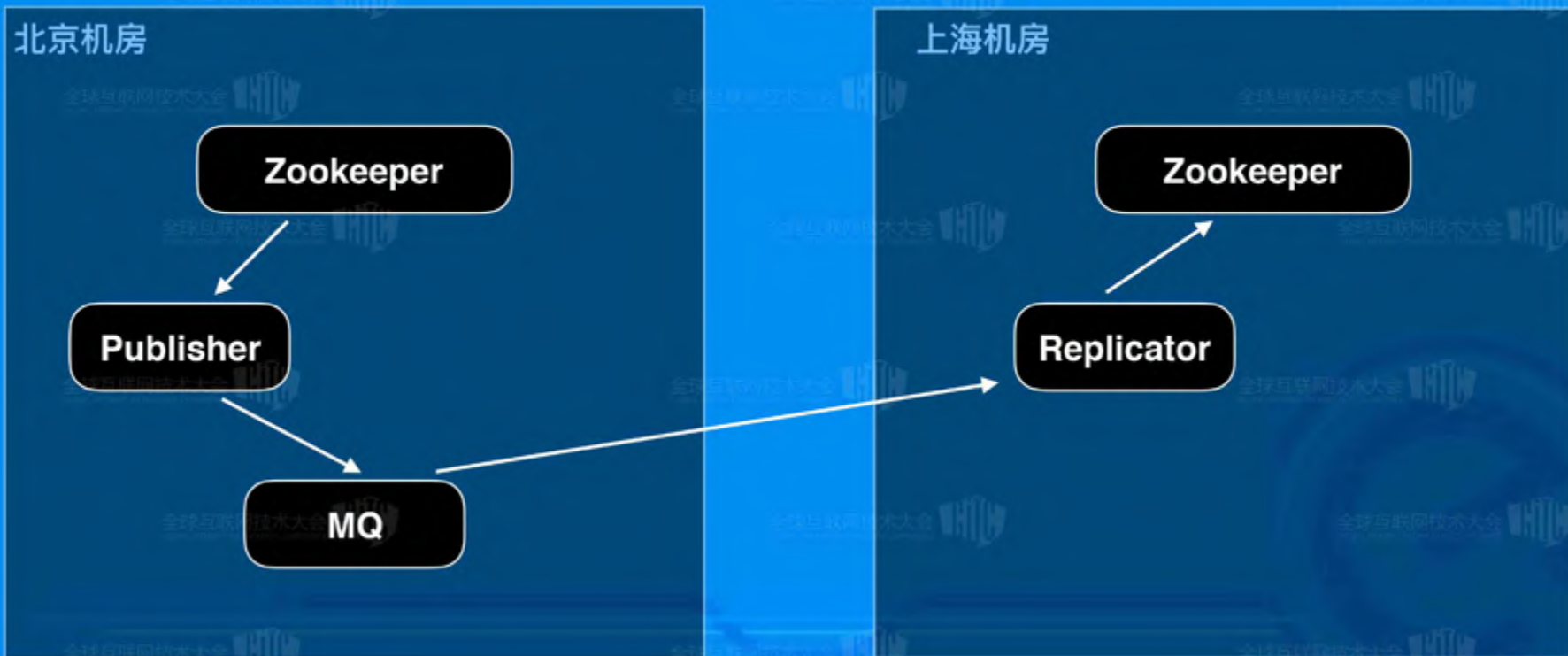
多活设计：全局强一致性业务



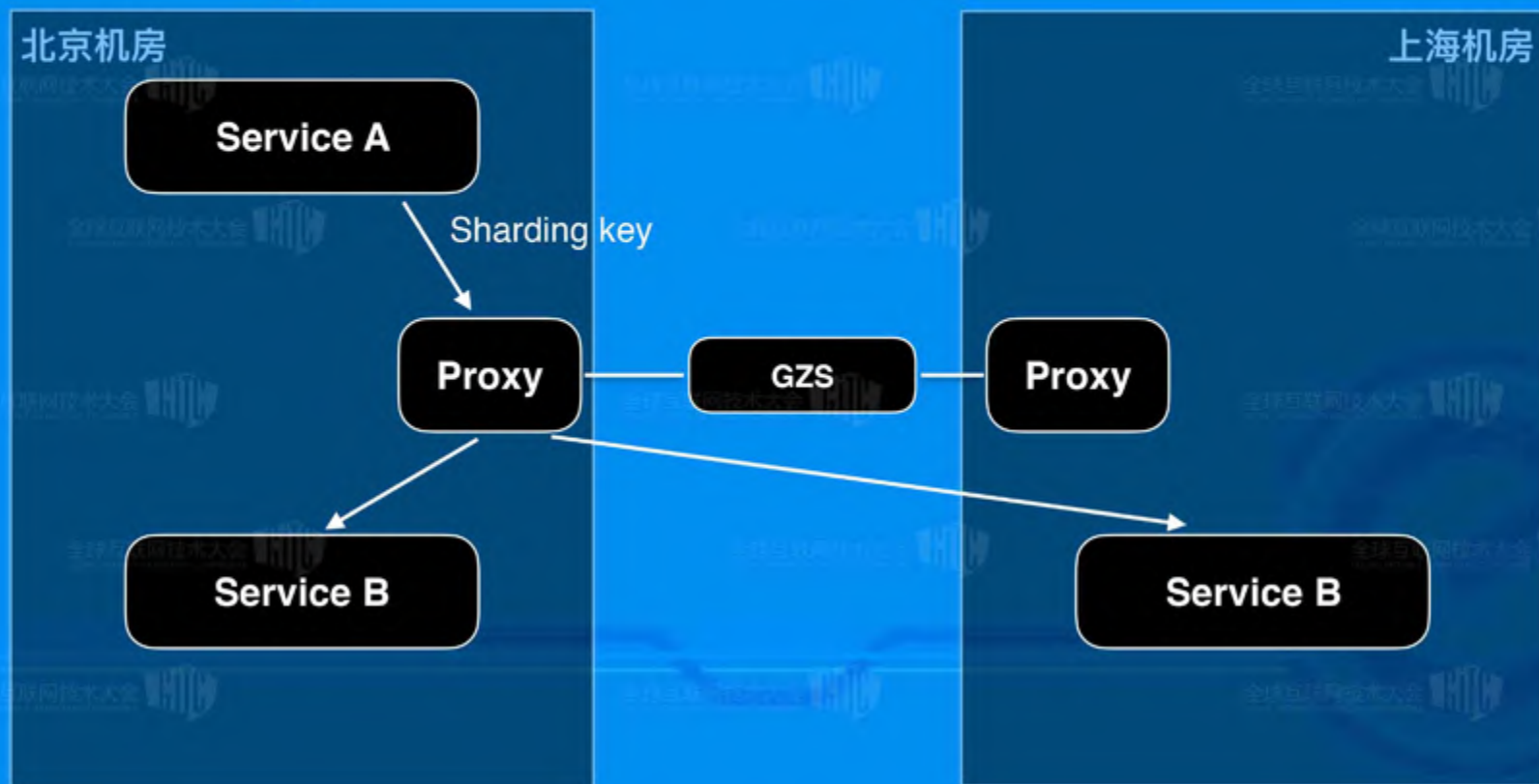
多活设计：后台任务



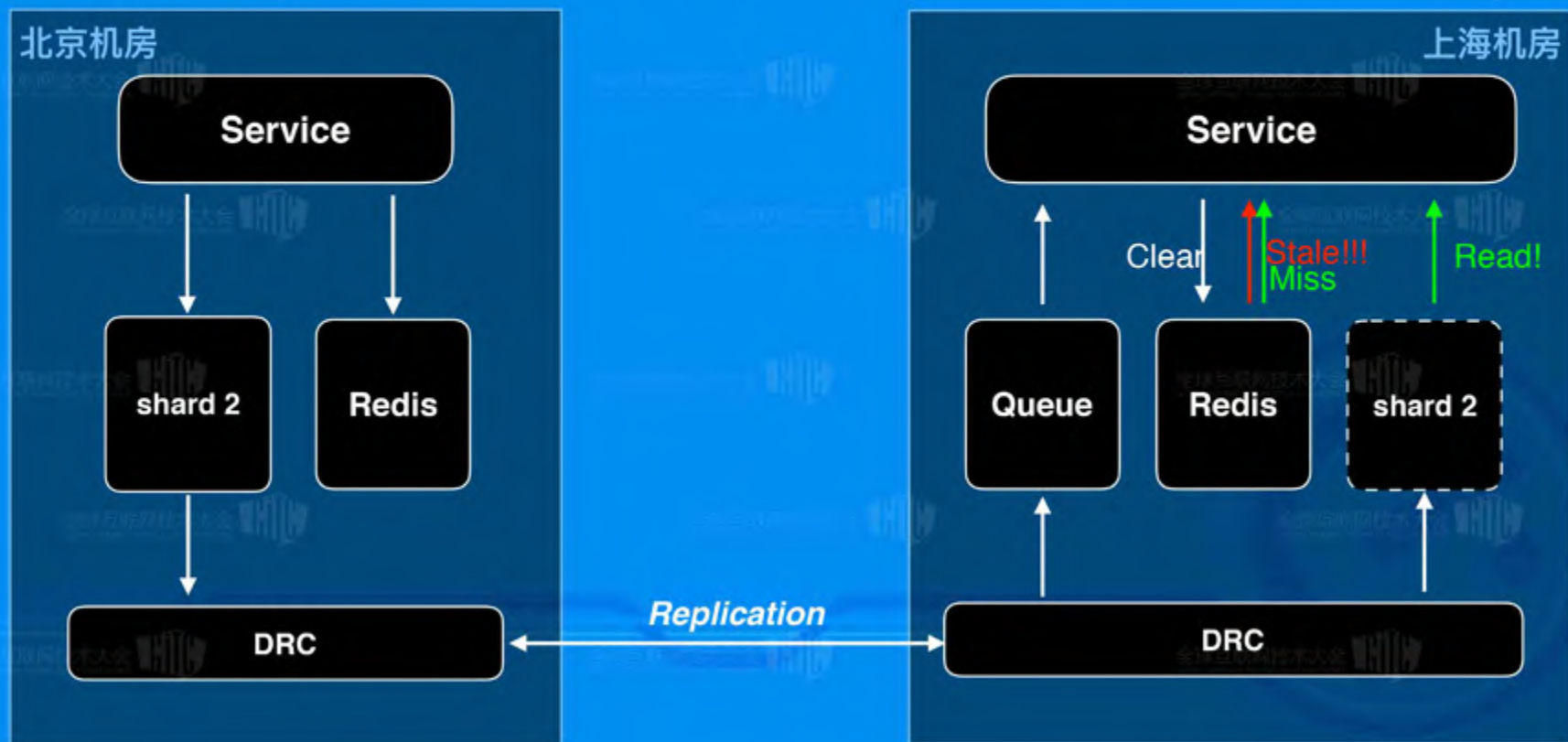
设计方案：Zookeeper同步



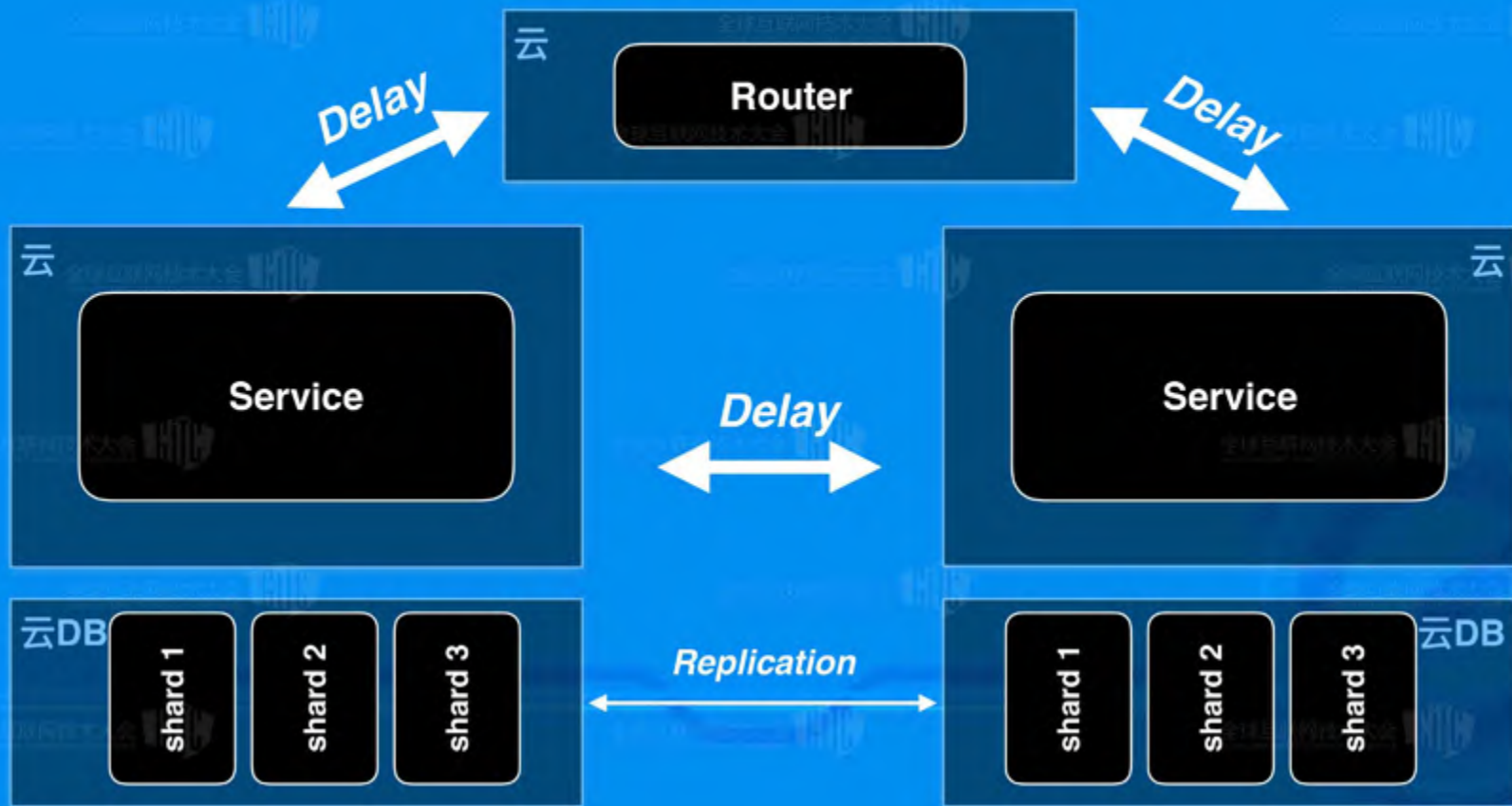
设计方案：跨机房服务调用



多活设计：缓存同步



多活设计：测试



多活设计：基础组件、工具

- 改造已有组件和工具
 - DAL
 - PIZZA
 - Python框架
 - Java框架
 - Trace
 - Metrics
 - 发布系统
- 新增组件和工具
 - IDC Router
 - GZS
 - DRC
 - SOA Proxy
 - DB比对工具
 - 一键部署工具Eone
 - 多活控制台gzs console

回顾与展望

- 前期准备（半年）
 - 方案调研、设计
 - 基础组件开发
- 一期（3个月）
 - 主要业务改造，测试
 - 双机房部署上线
 - 灰度切换、灾难演练
- 二期（待定）
 - 其他业务多活
 - 基础设施和工具进一步改进
 - 三机房、四机房部署

饿了么多活架构分享

陈忠克

zhongke.chen@ele.me

2017年6月23日



饿了么简介

- 愿景：以创新科技打造全球领先的本地生活平台
- 主营外卖、商超、即时配送、餐饮供应链等业务
- 外卖覆盖城市数1400+，餐厅数100万+，日订单900万+，日运单450万+，骑手300万+



用户浏览、下单、支付



商家接单、备餐、呼叫配送



骑手取餐、送餐

饿了么多活的背景

- 北京IDC: 4000+服务器
- 上海IDC: 建设中
- 峰值订单: 1000+/s, 规划容量4000+
- 微服务: 2000+ (Python, Java, Go)
- 灾备: 3000+云主机



饿了么为什么做多活?

单机房+灾备

- 容灾能力差
- 扩容受限于单机房容量
- 灾备资源巨大浪费



多活

- 提升容灾能力 (任何一个机房可挂, 流量可切)
- 多机房扩展能力 (不受限于任何一个机房的物理限制)
- 充分利用资源 (如果有n个同等大小的机房, 理论上利用率可达 $(n-1)/n$)



饿了么做多活的基本思路

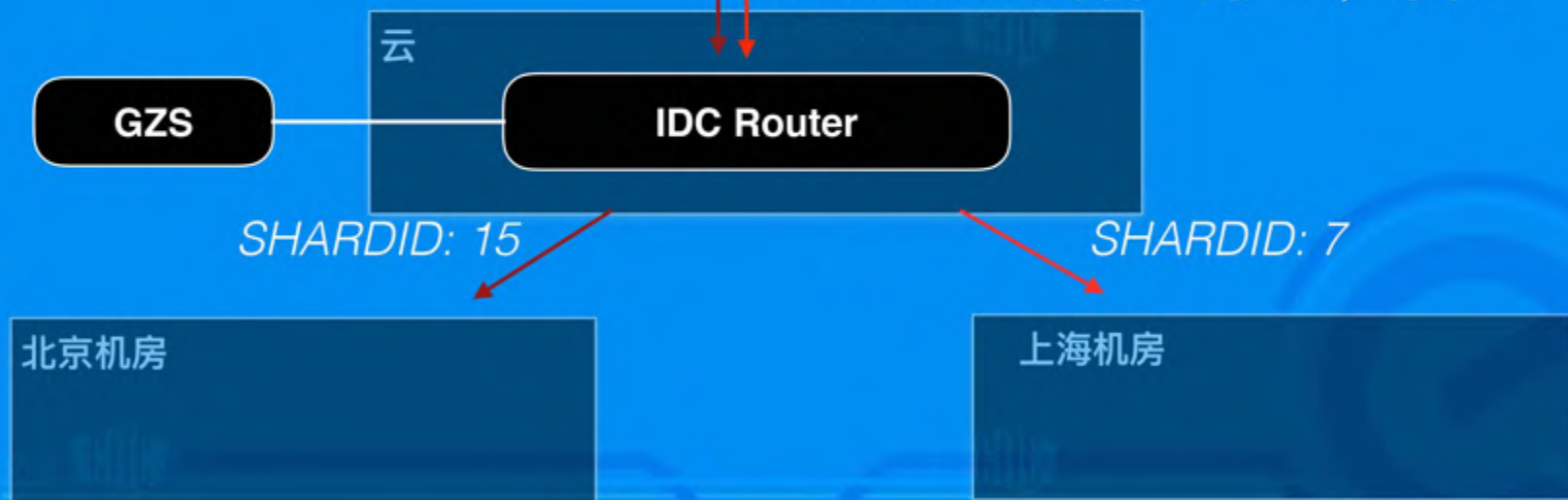
- 同城多活
 - 选择性数据同步
 - 一致性优先
 - 不同类型用户到不同机房，
一笔交易跨多个机房
- VS
- 异地多活
 - 全量数据同步
 - 可用性优先
 - 交易不跨机房

第一步：双机房异地多活！

多活设计：分流

用户app、商户app、骑手app

X-SHARD: 经纬度 商户ID, 订单ID



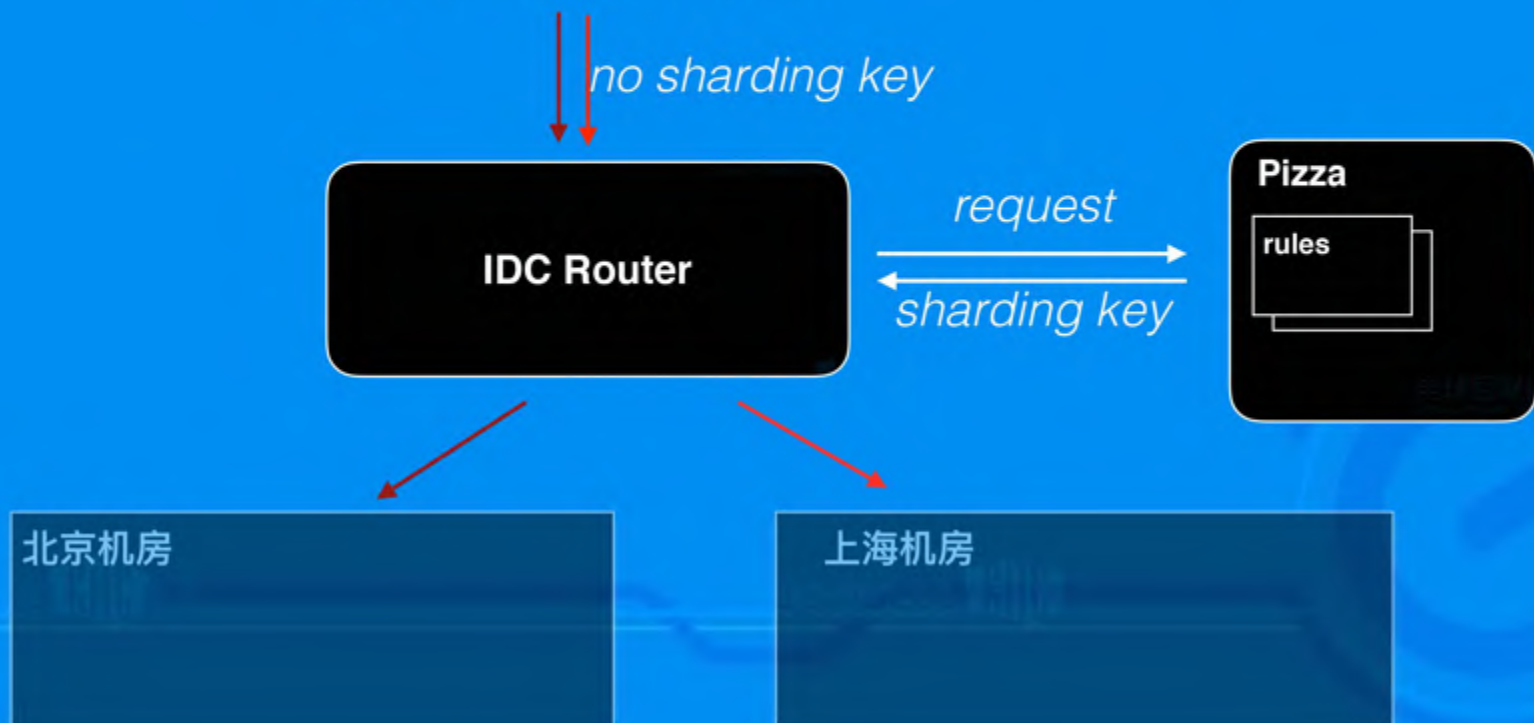
多活设计：分片规则

- 以省份（直辖市）为分片单位
- 南方省份划归南方机房
- 北方省份划归北方机房

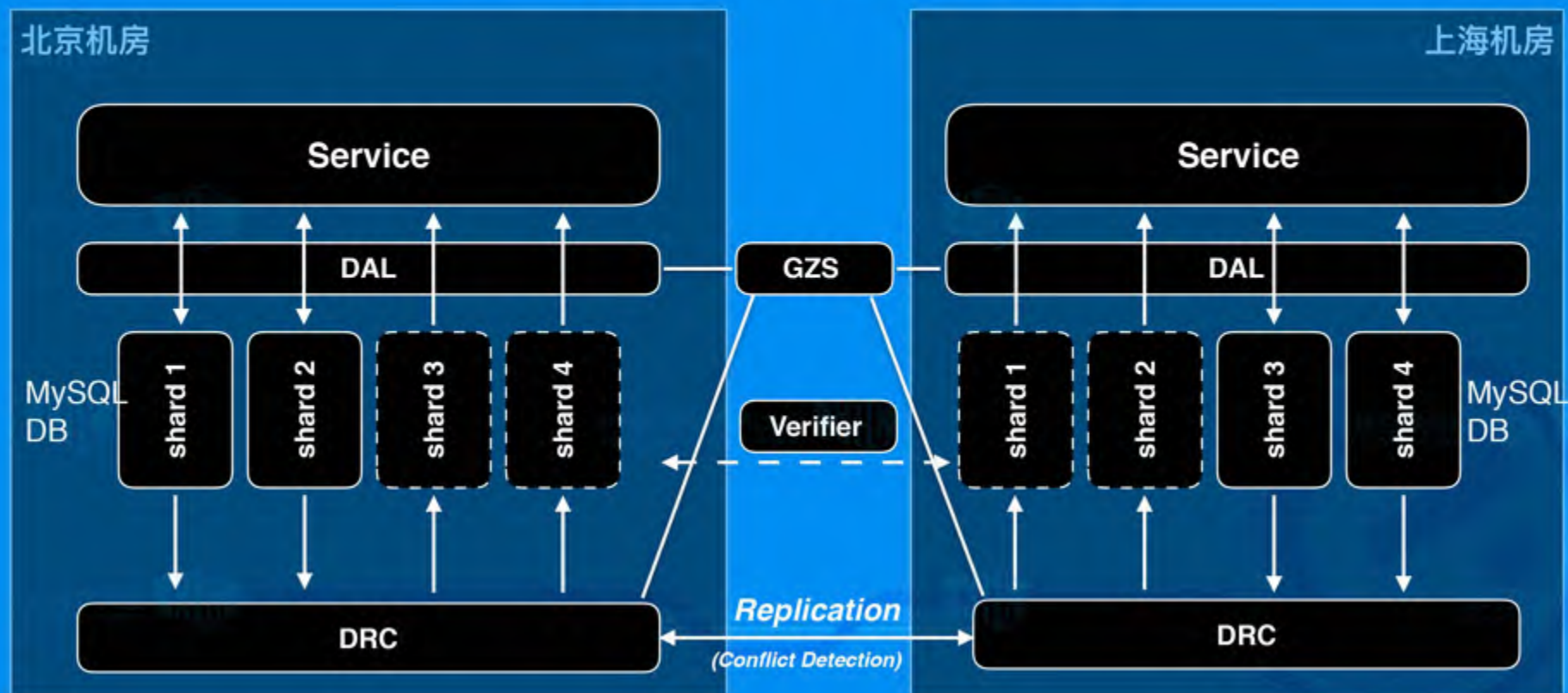


多活设计：老版本应用兼容

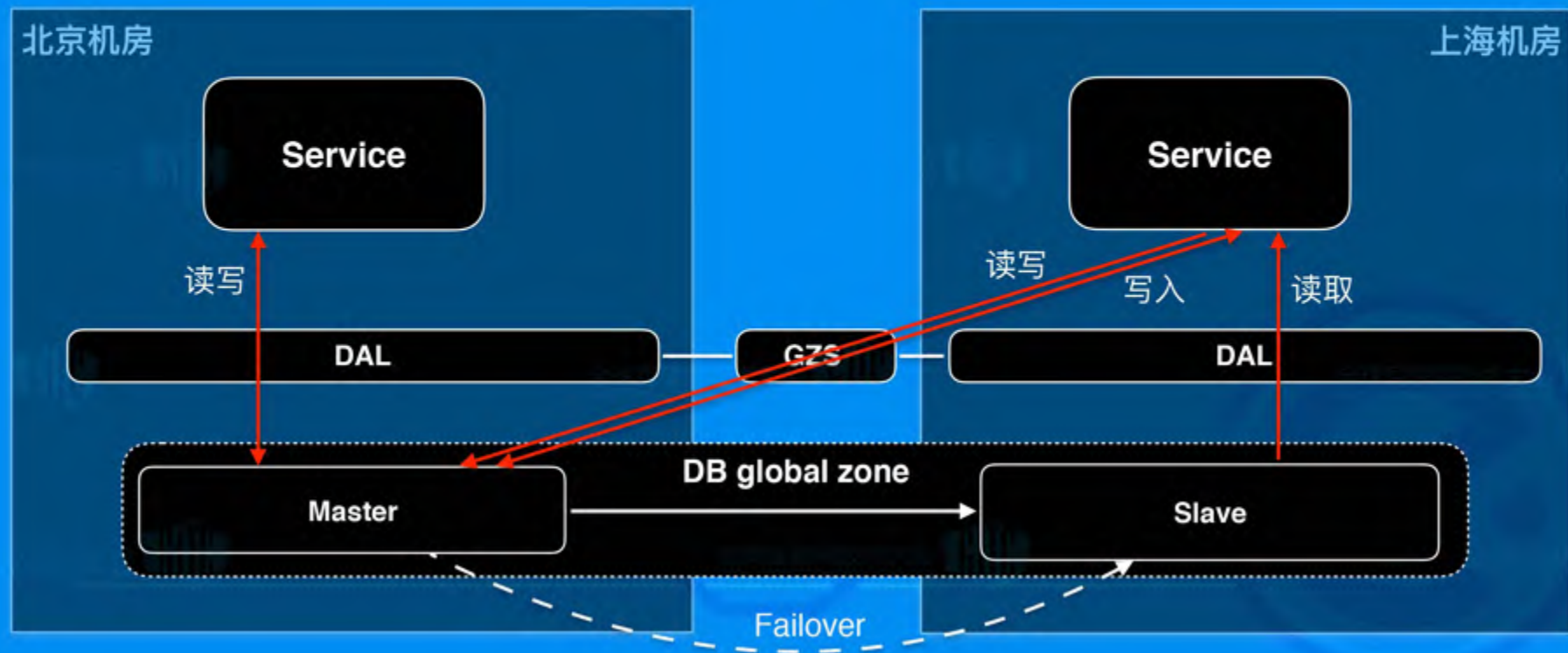
老版本应用，开放api应用



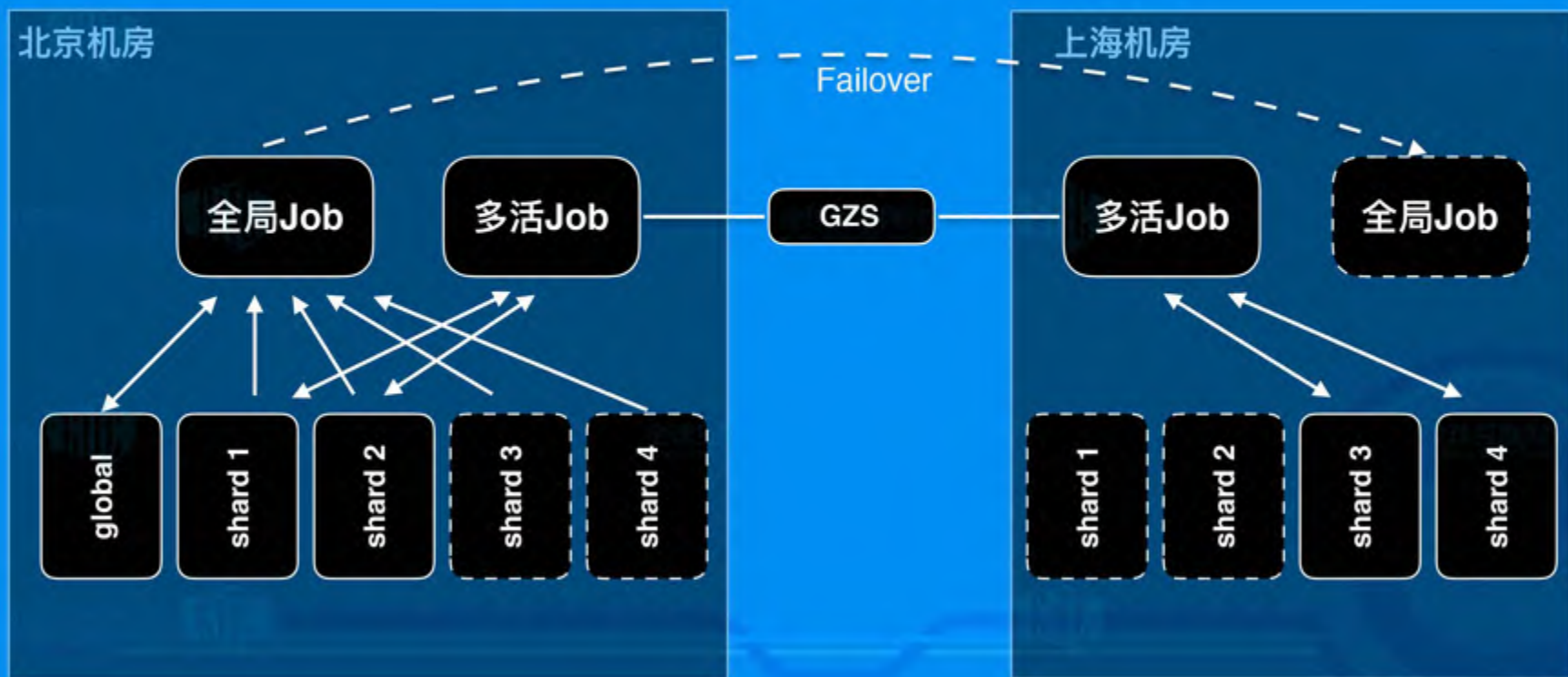
多活设计：数据库同步



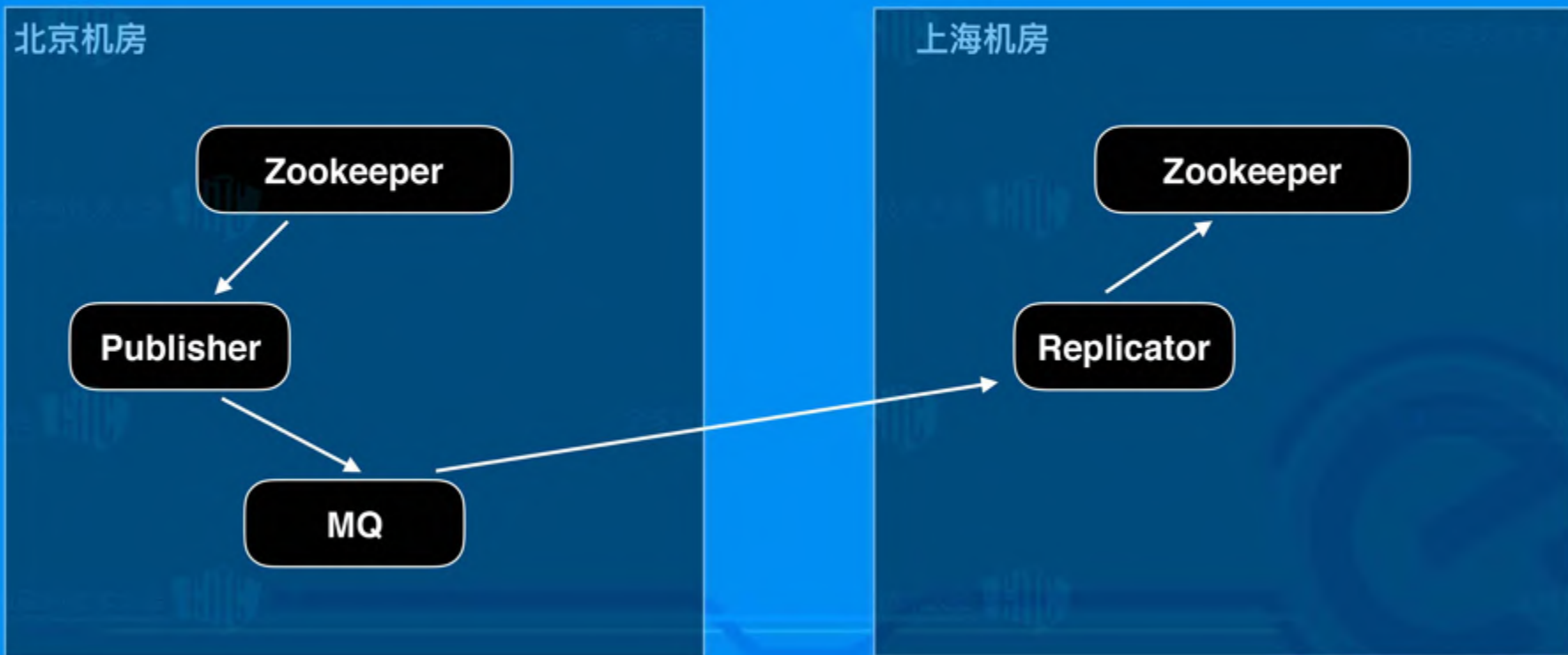
多活设计：全局强一致性业务



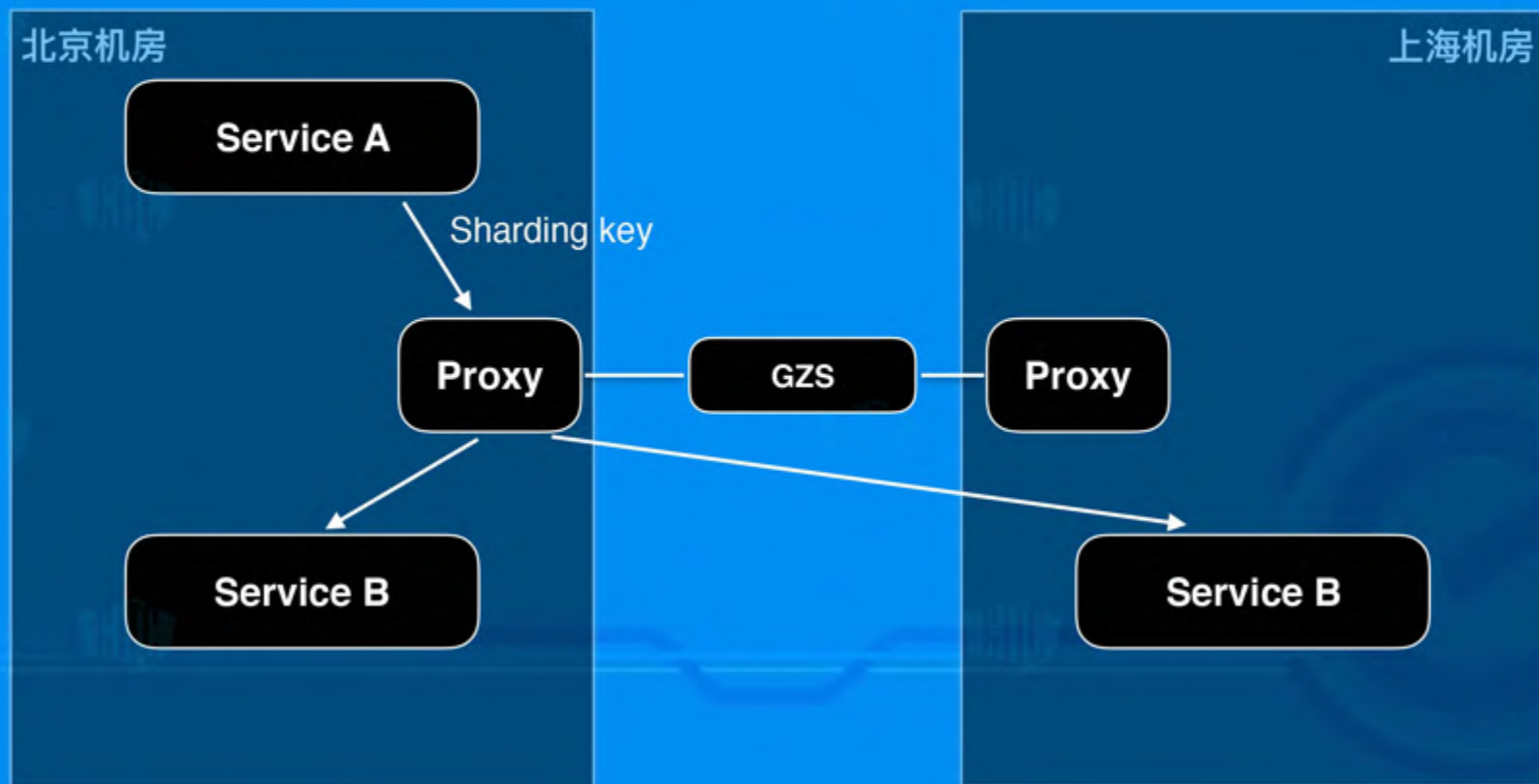
多活设计：后台任务



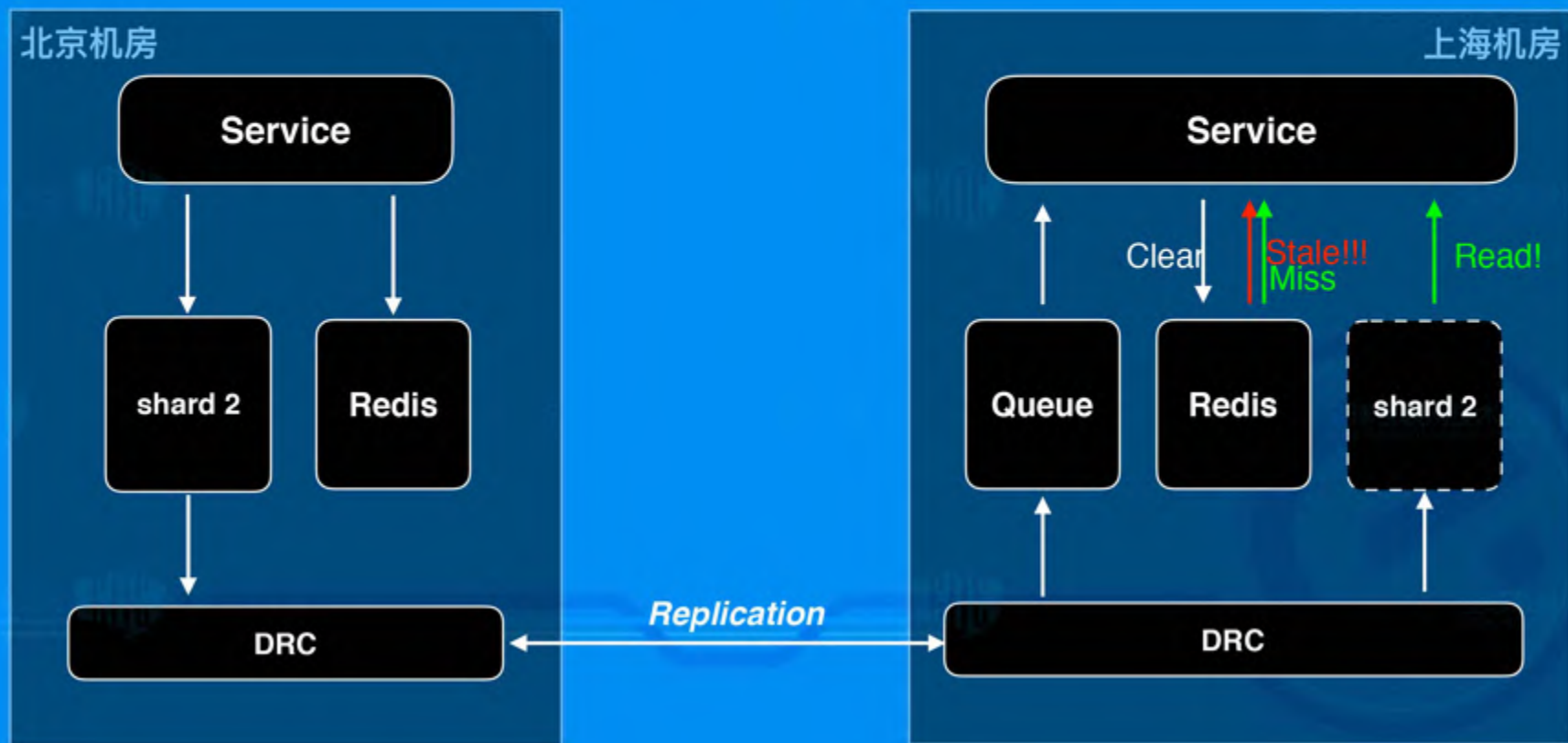
设计方案：Zookeeper同步



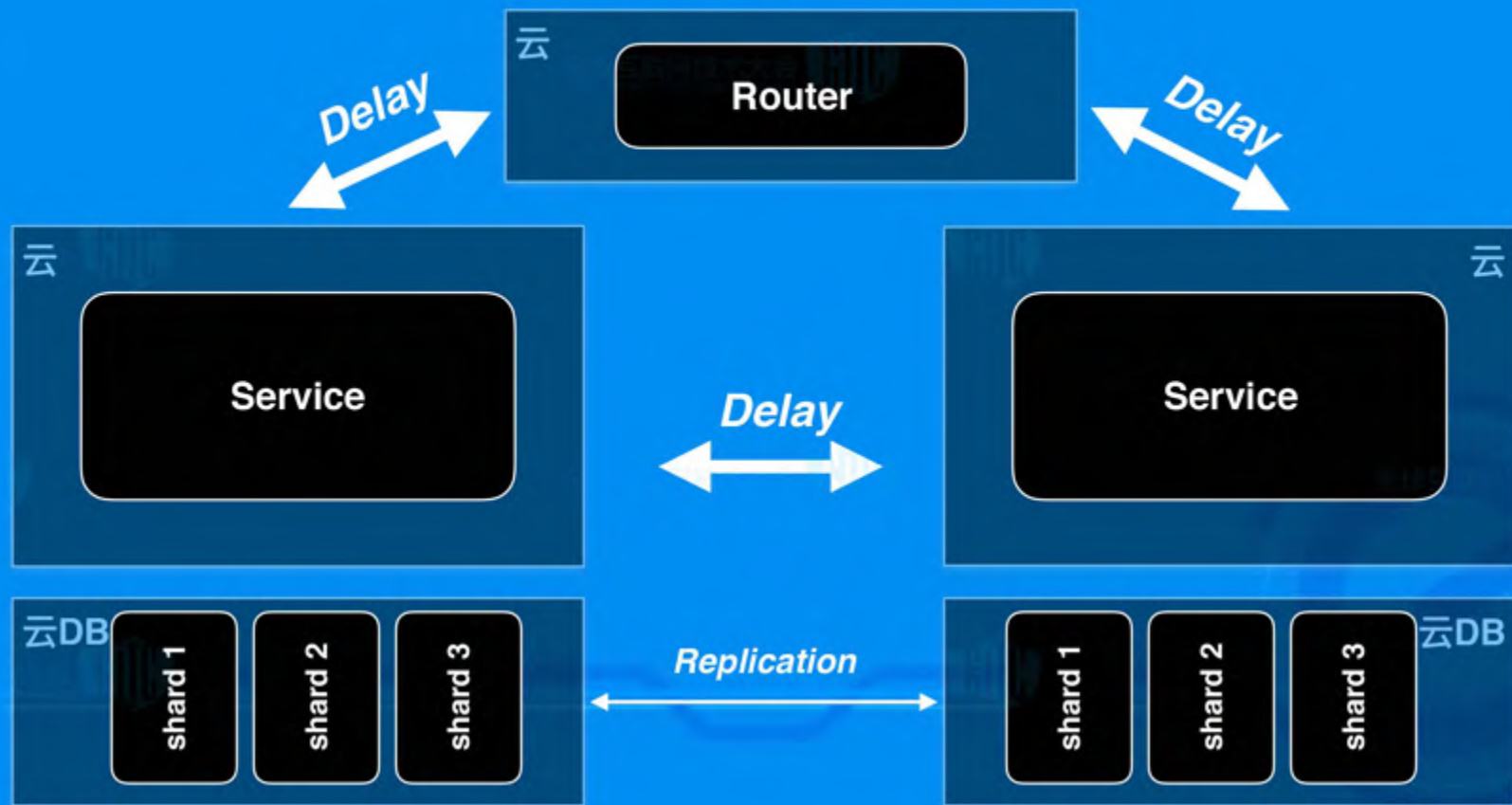
设计方案：跨机房服务调用



多活设计：缓存同步



多活设计：测试



多活设计：基础组件、工具

- 改造已有组件和工具

- DAL
- PIZZA
- Python框架
- Java框架
- Trace
- Metrics
- 发布系统

- 新增组件和工具

- IDC Router
- GZS
- DRC
- SOA Proxy
- DB比对工具
- 一键部署工具Eone
- 多活控制台gzs console

回顾与展望

- 前期准备（半年）
 - 方案调研、设计
 - 基础组件开发
- 一期（3个月）
 - 主要业务改造，测试
 - 双机房部署上线
 - 灰度切换、灾难演练
- 二期（待定）
 - 其他业务多活
 - 基础设施和工具进一步改进
 - 三机房、四机房部署

The End

欢迎关注饿了么技术社区

