

# 异地多活单元化架构下的 微服务体系

时晖

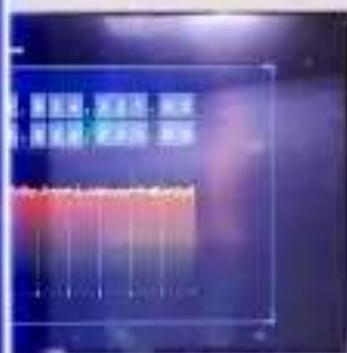
蚂蚁金服中间件 高级技术专家



# 从“挖光缆”到“剪网线”

# 三地五中心金融级高可用方案

FINANCIAL GRADE HIGH AVAILABILITY  
5-DATA CENTERS ACROSS 3-REGIONS

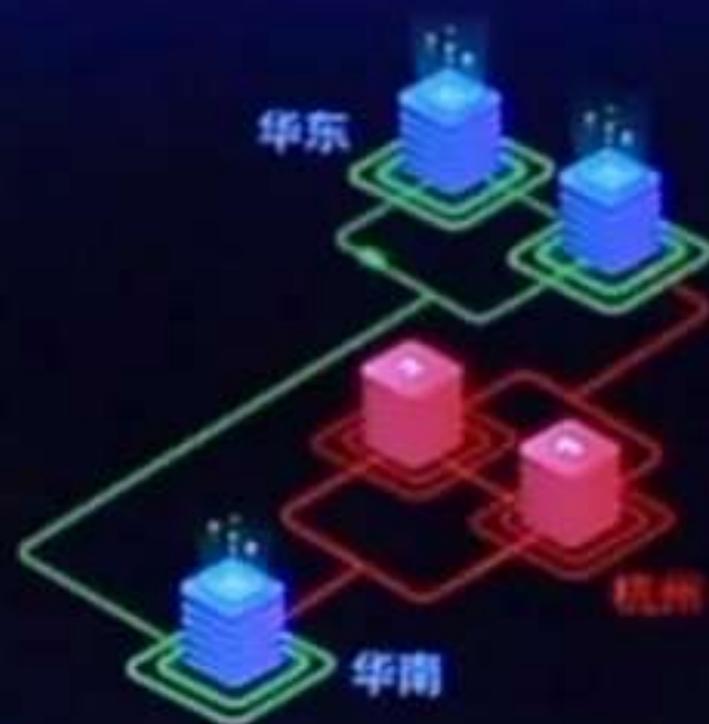


# 转账交易大盘 Transaction Dashboard

转入金额 ¥ 36,054,055.00  
转出金额 ¥ 36,054,055.00

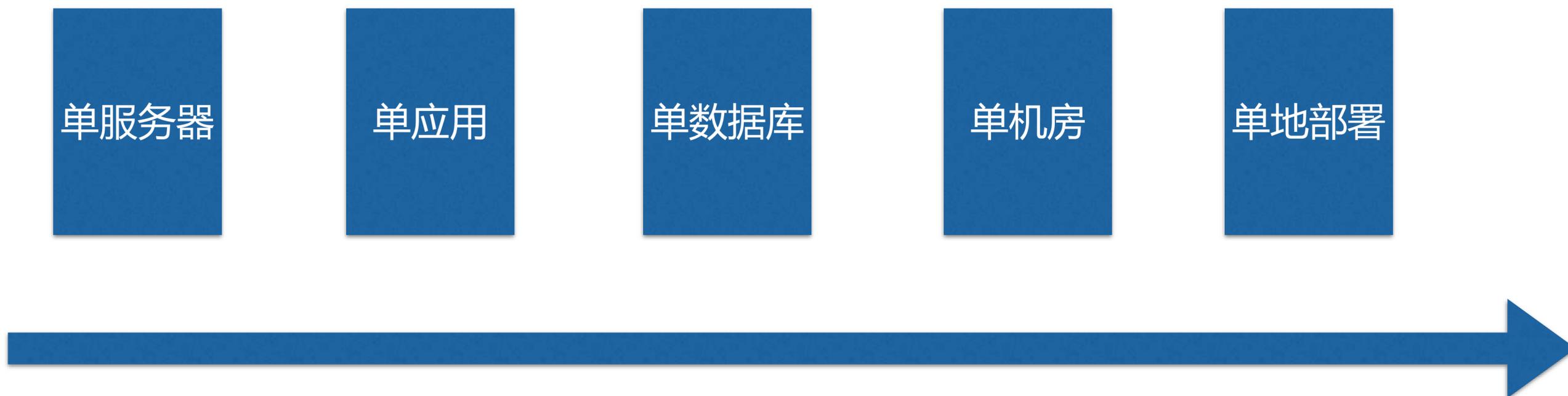


- 数据库切换 ●
- 缓存容灾切换 ●
- 多活规则切换 ●
- 中间件切换 ●
- 负载均衡切换 ●
- 域名解析切换 ●



# 去单点之路

# 各种层面的“单点”瓶颈



张翠花  
女，52岁  
个体工商户  
支付宝会员ID：54321233

绿色食品  
无明矾油条

王小明  
男，27岁  
互联网公司职员  
支付宝会员ID：12345666

# 单体应用



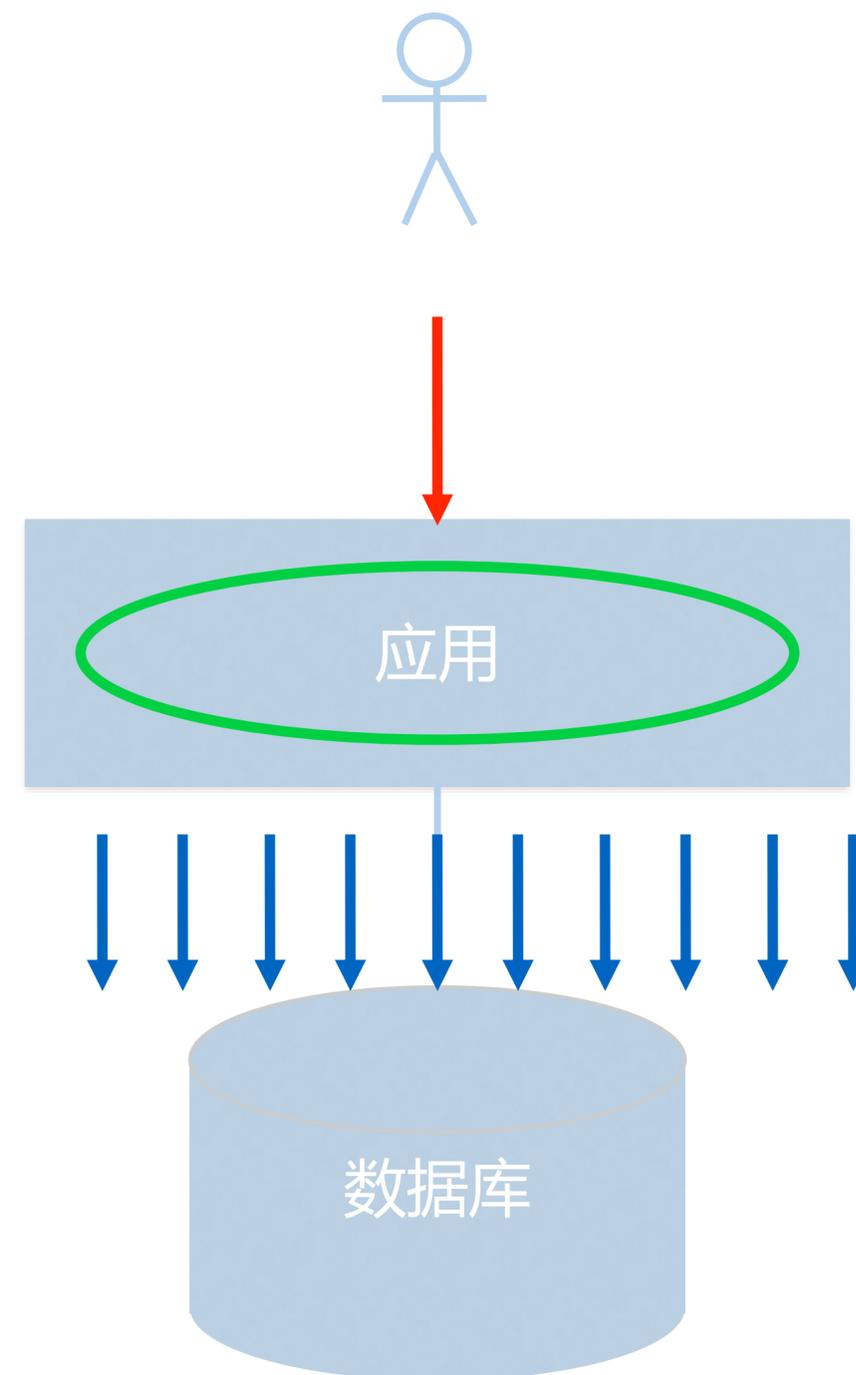
# 单体应用

异地请求 1 次

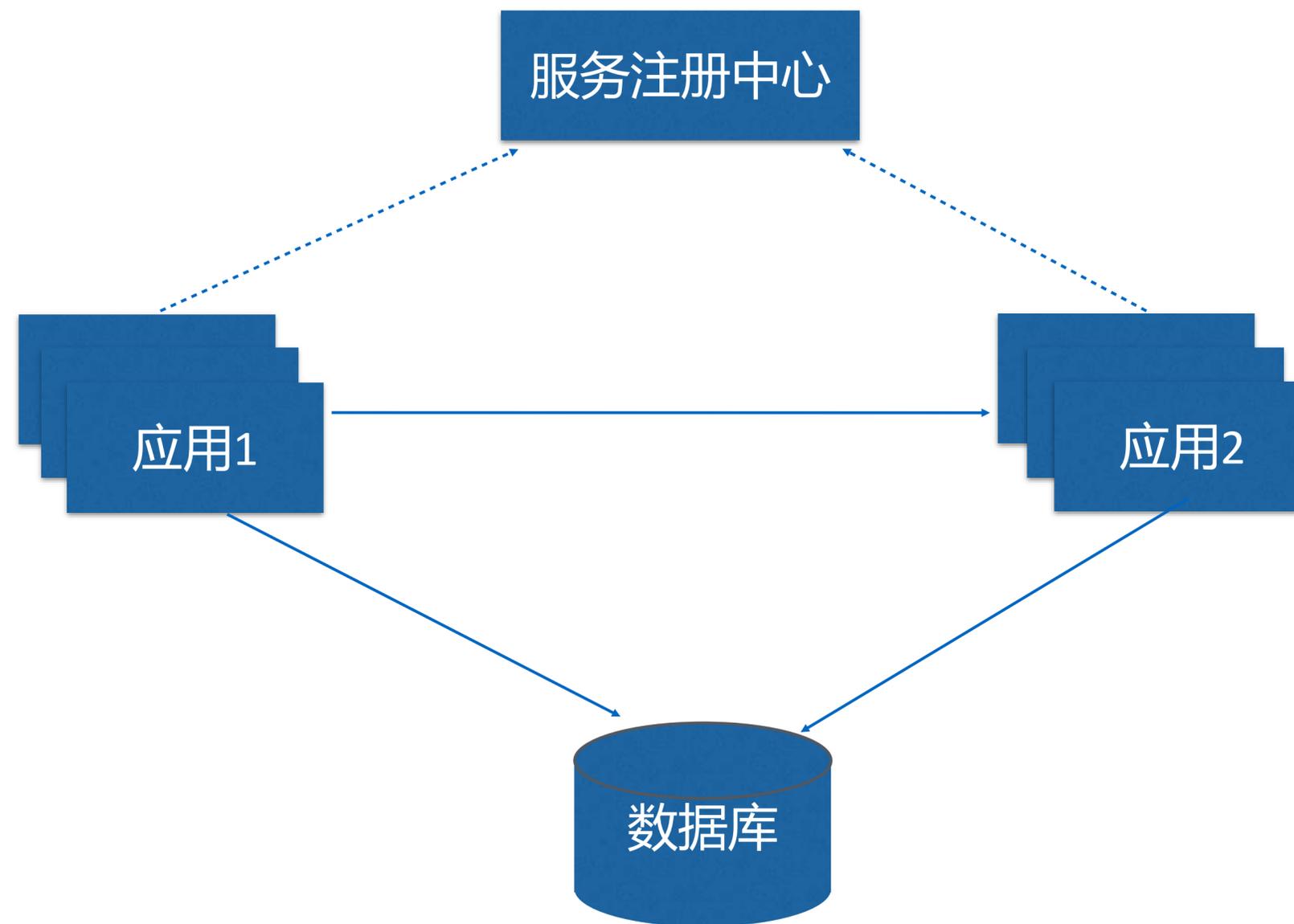
进程内调用 N 次

数据库访问 10 次

此数据为概略示例，仅供定性分析问题



# 单机房服务化



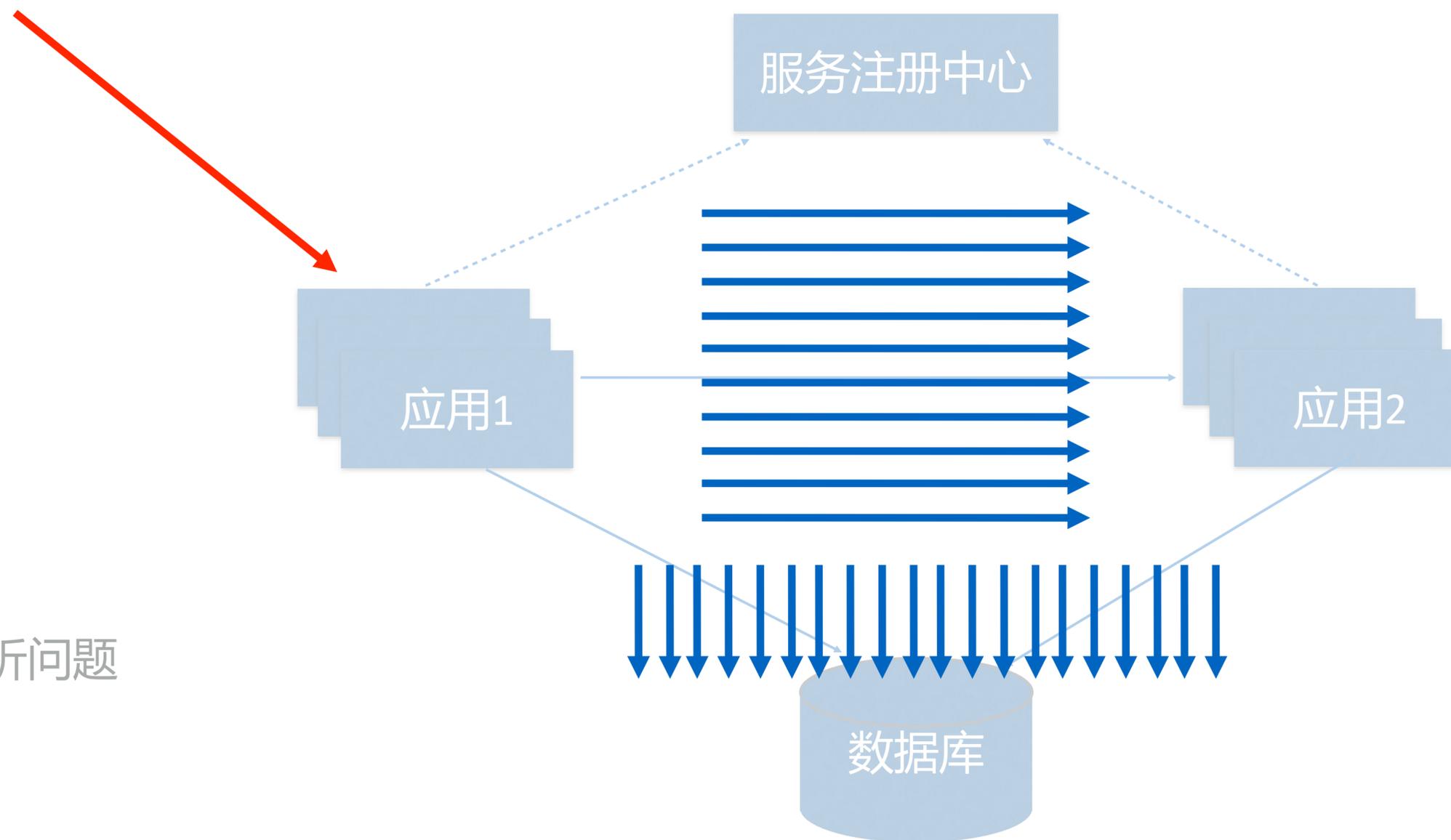
# 单机房服务化

异地请求 1 次

RPC调用 10 次

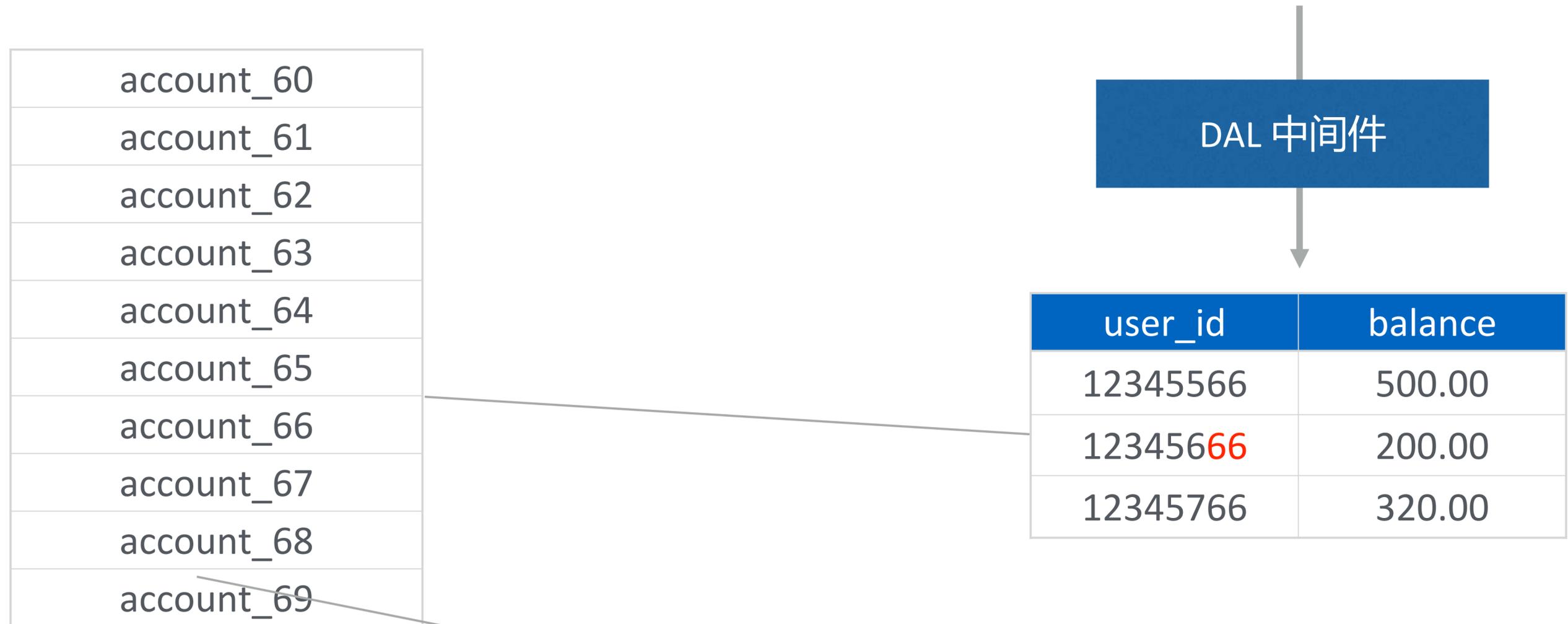
数据库访问 20 次

此数据为概略示例，仅供定性分析问题

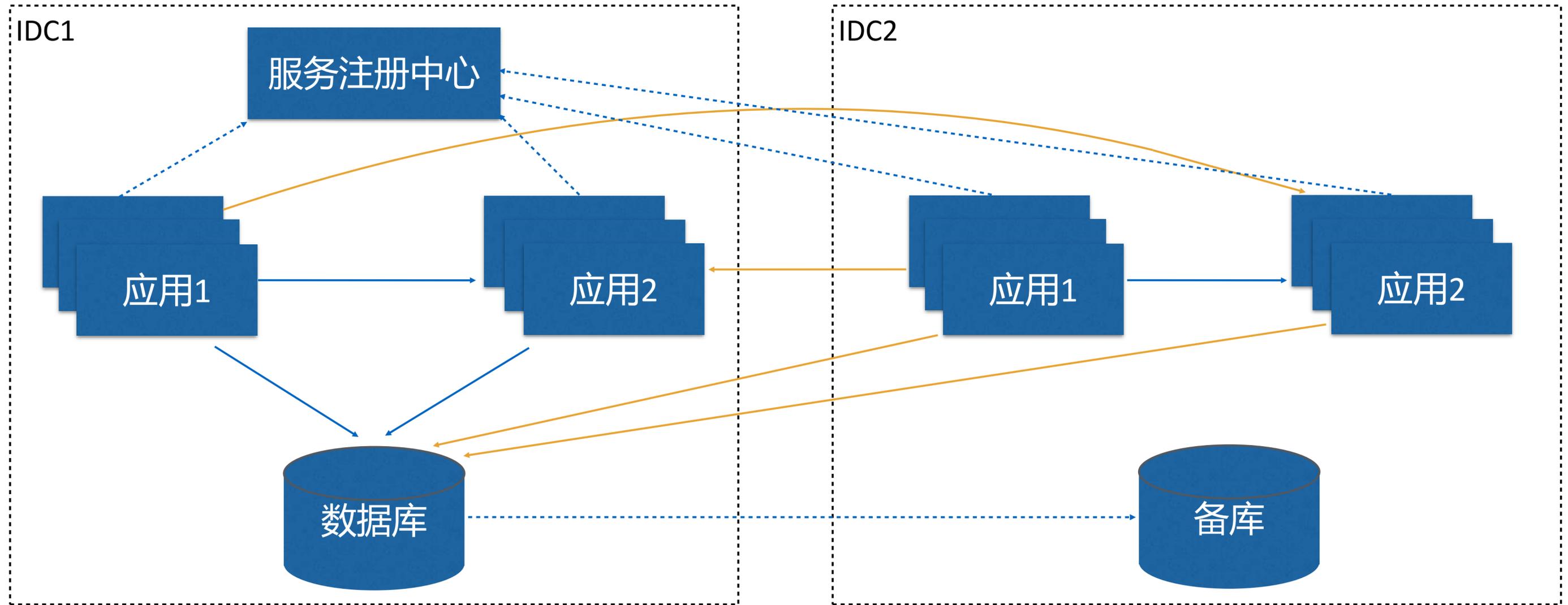


# 分库分表

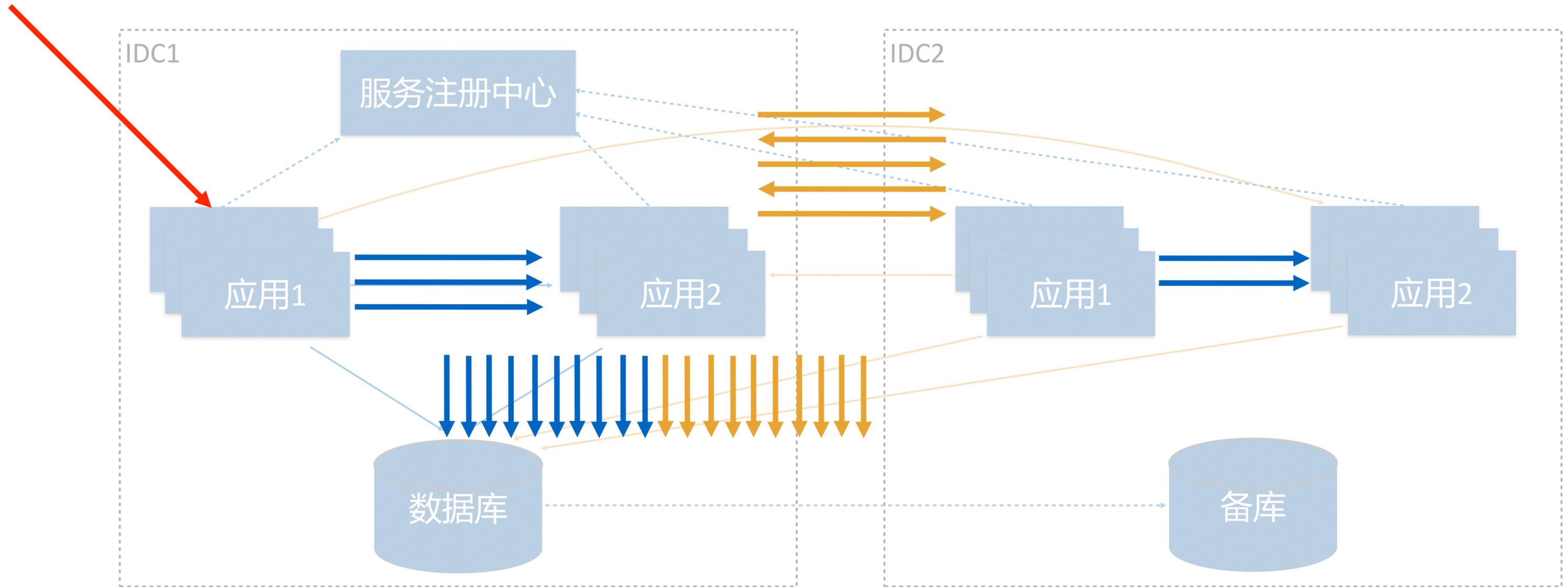
```
SELECT * FROM account WHERE user_id = '12345666';
```



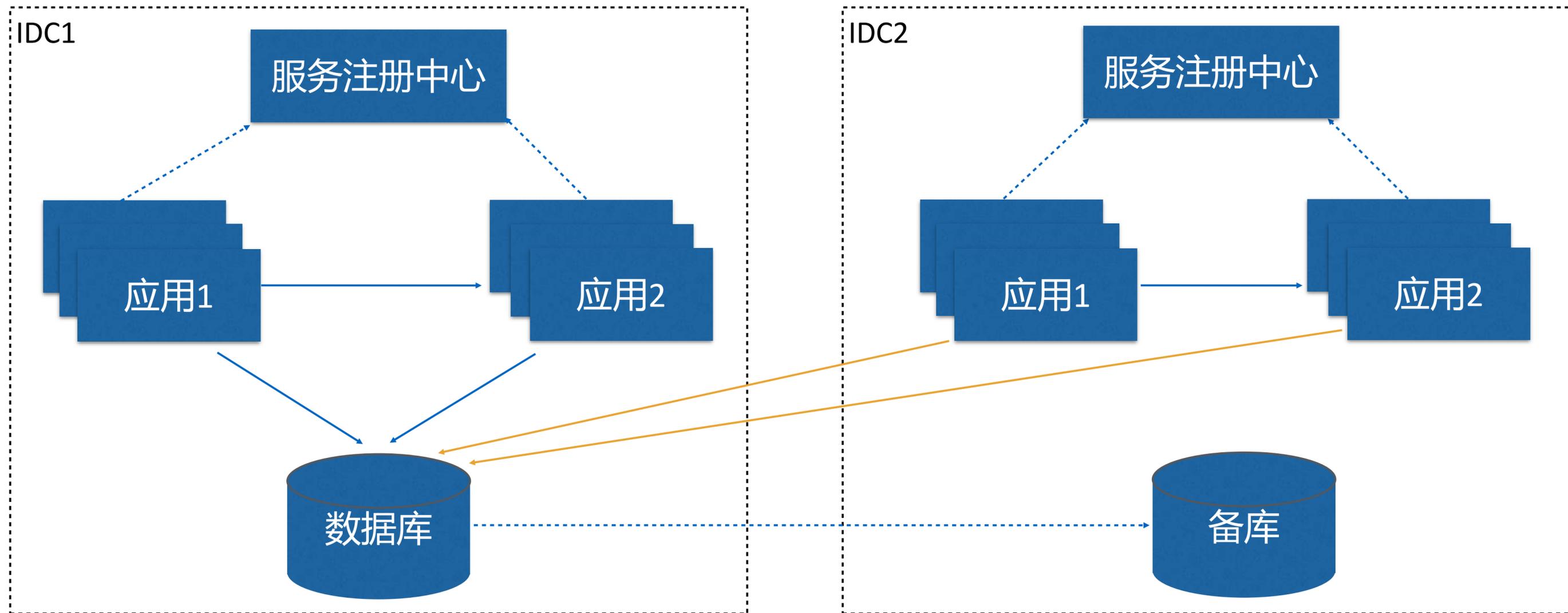
# 同城多机房阶段一



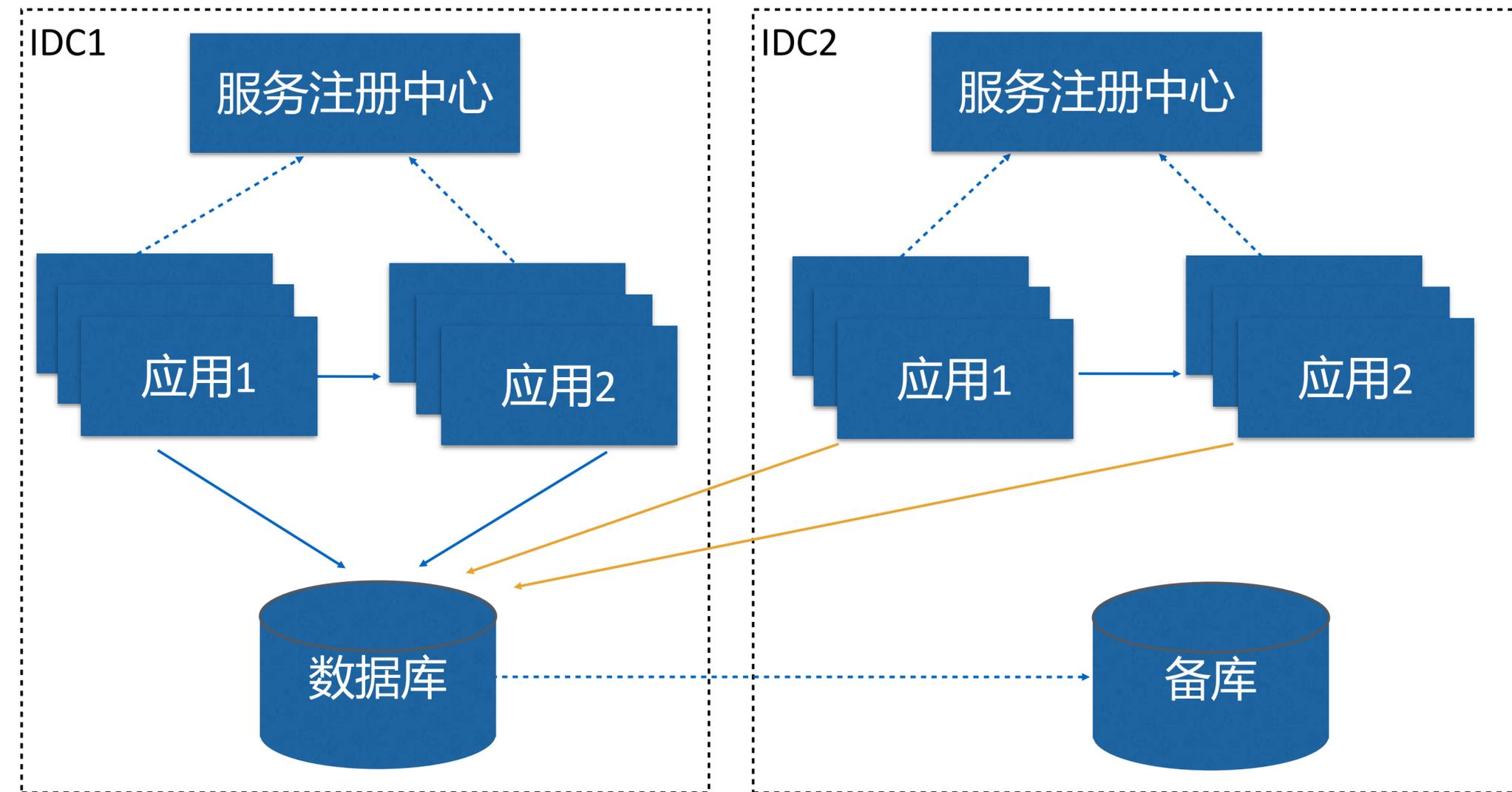
# 同城多机房阶段一



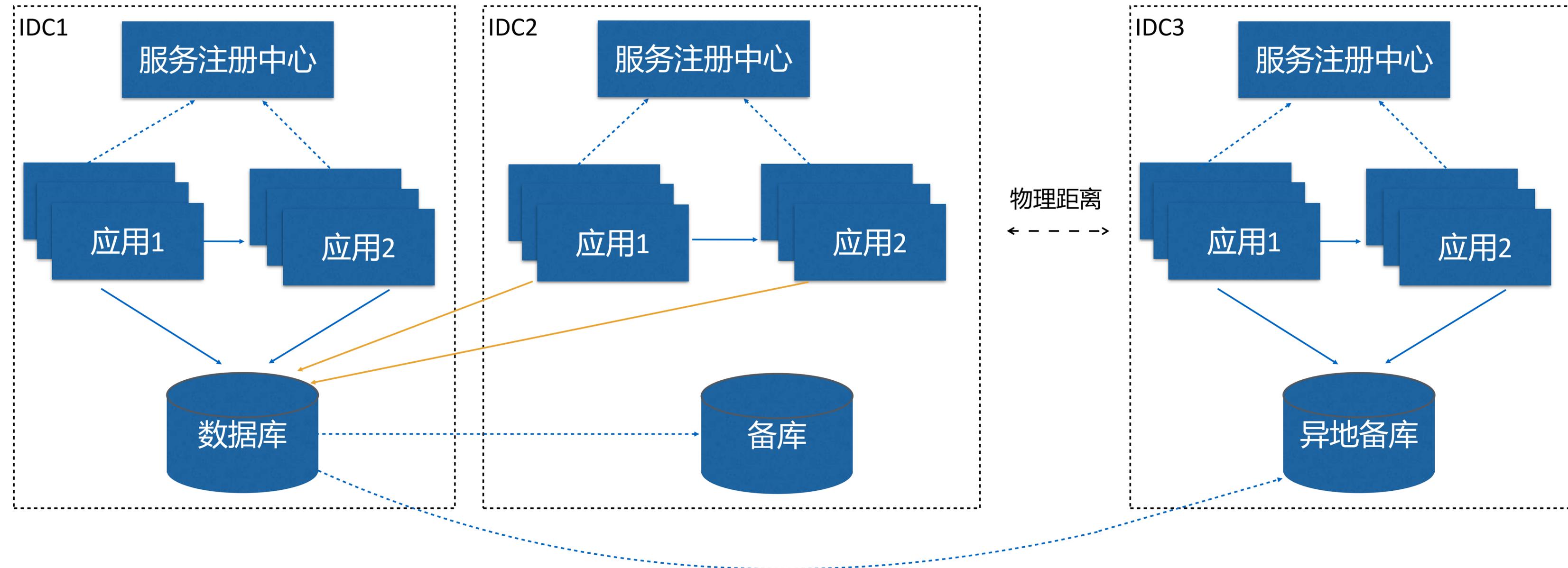
# 同城多机房阶段二



# 两地三中心（实际未采用）



# 两地三中心（实际未采用）



# 小结

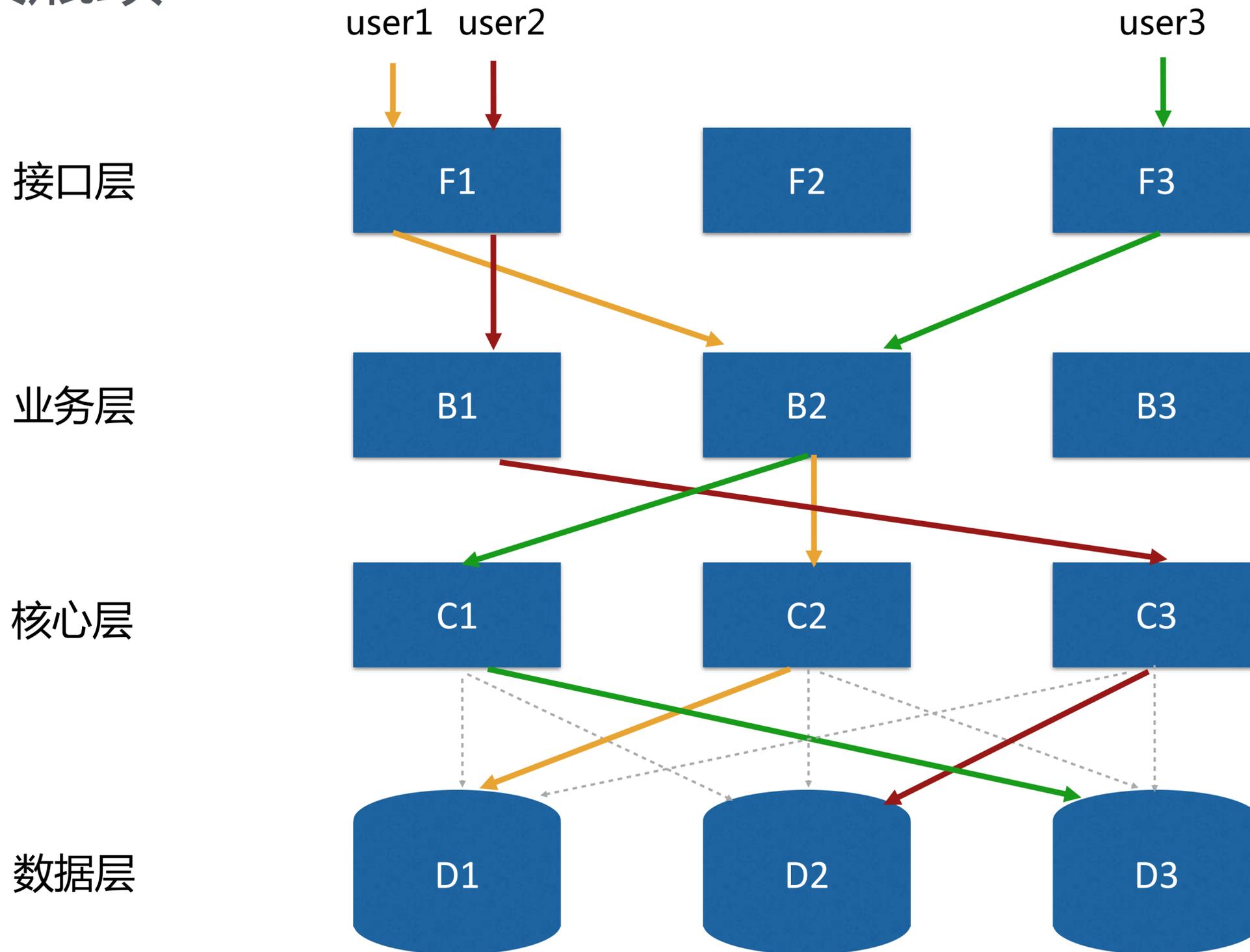
架构类型	优势	问题
单体应用	网络开销小	扩展性差，维护困难
单机房服务化	解耦，可扩展	容量受限，机房级单点
同城多机房阶段一	突破单机房容量瓶颈	<b>非必要的</b> 跨机房网络开销大
同城多机房阶段二	非必要的跨机房网络开销小； 机房级容灾能力	城市级单点
两地三中心	(理论上的) 异地容灾能力	网络耗时与数据一致性的矛盾

# 蚂蚁金服单元化实践

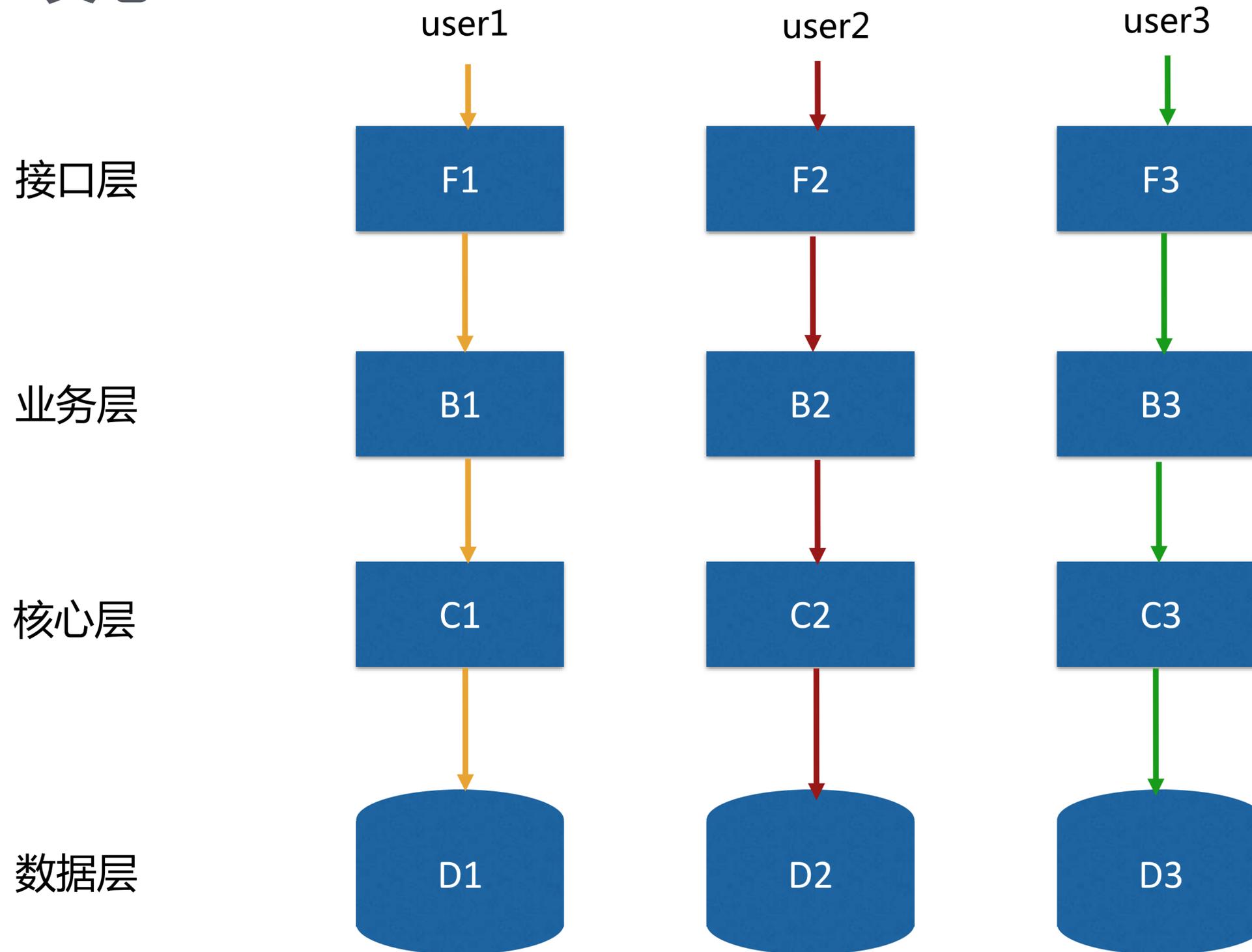
# 原始驱动力

- 异地多活容灾需求带来的数据访问耗时问题，量变引起质变
- 数据库连接数瓶颈制约了整体水平扩展能力，危急存亡之秋

# 数据库连接瓶颈



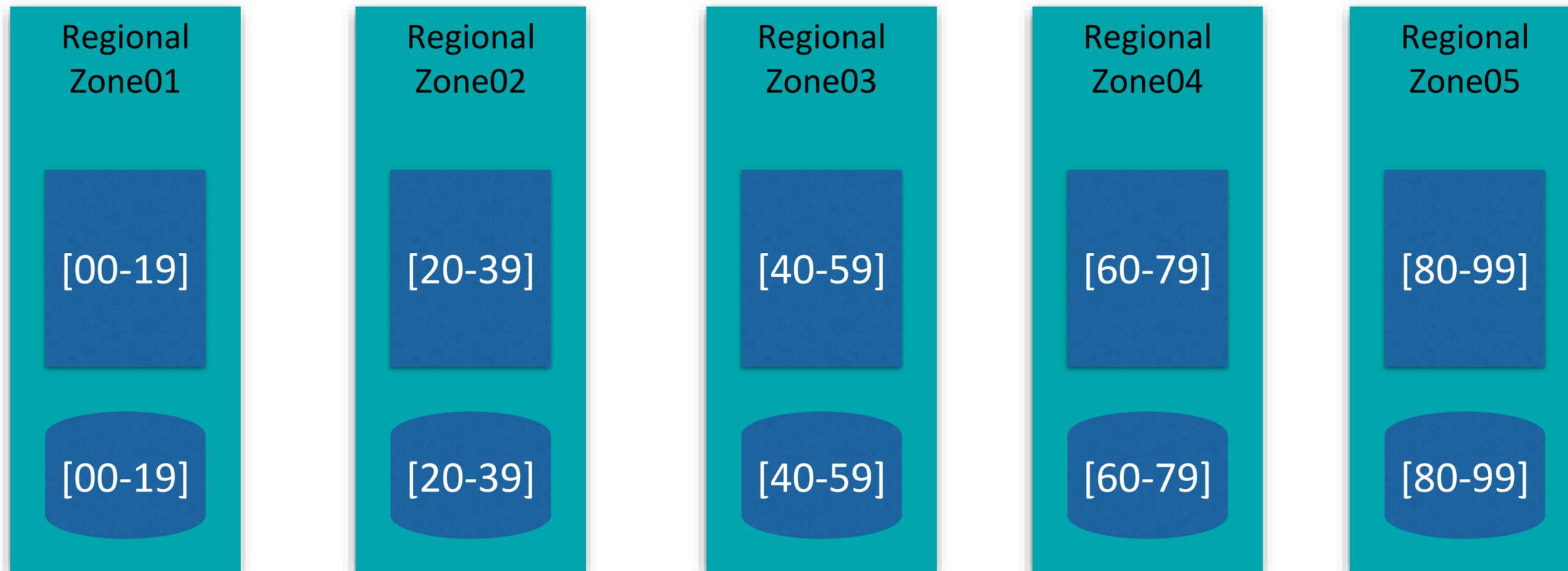
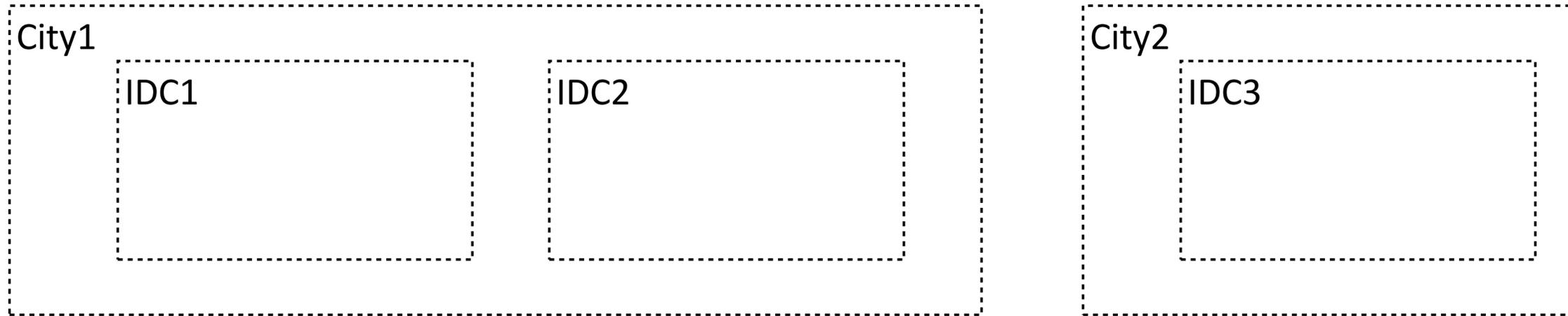
# “单元化”设想



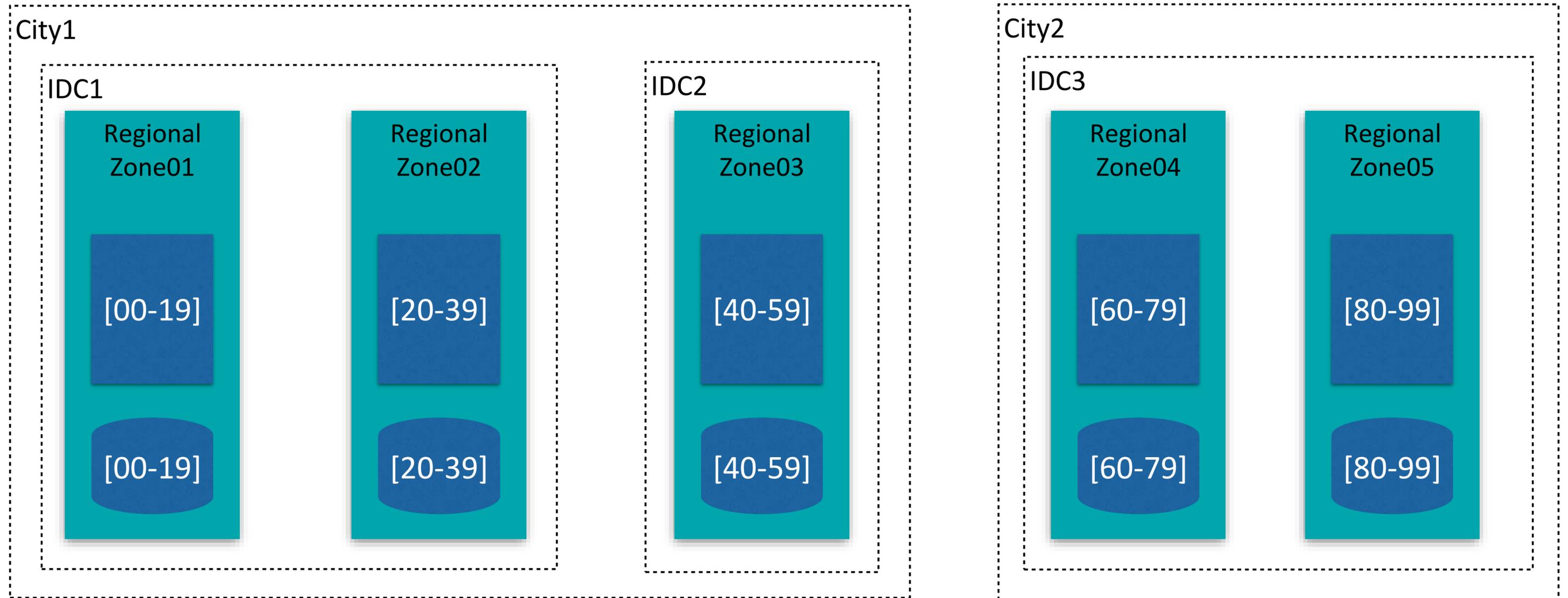
# 单元化设计原则

- 核心业务单元化
- 保证核心业务单元分片均衡(UID)
- 核心业务尽量自包含（调用封闭）
- 面向逻辑分区设计，而不是物理部署

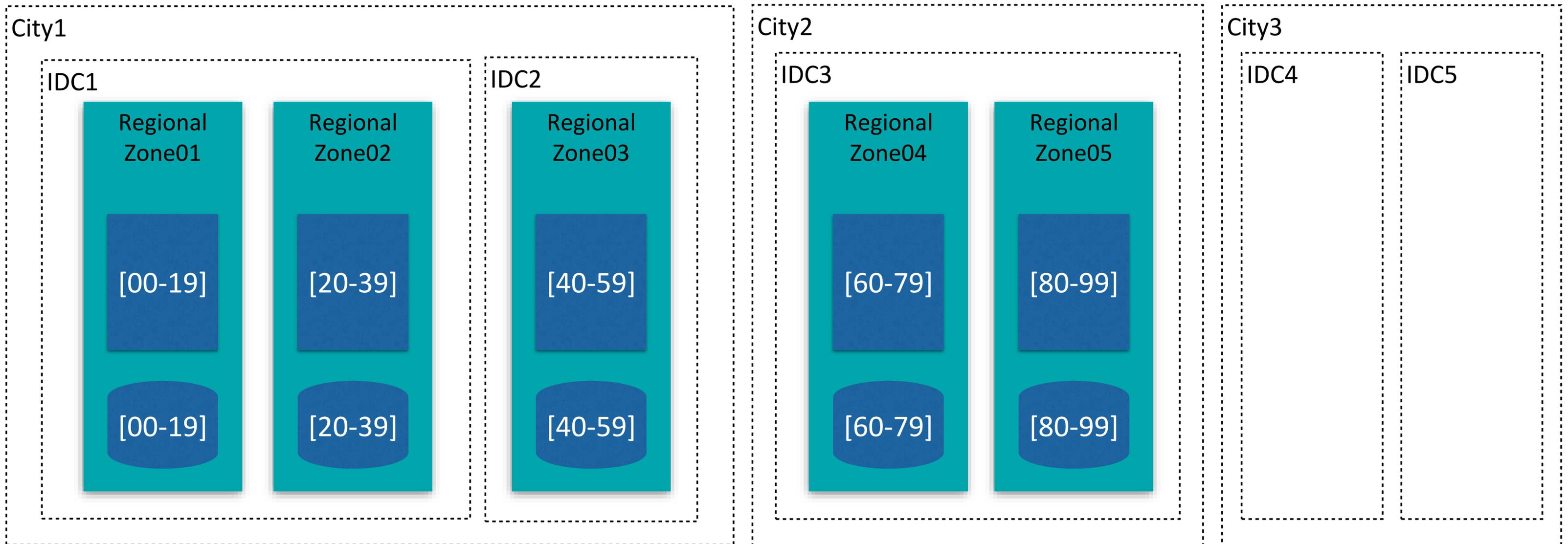
# 逻辑单元与物理数据中心



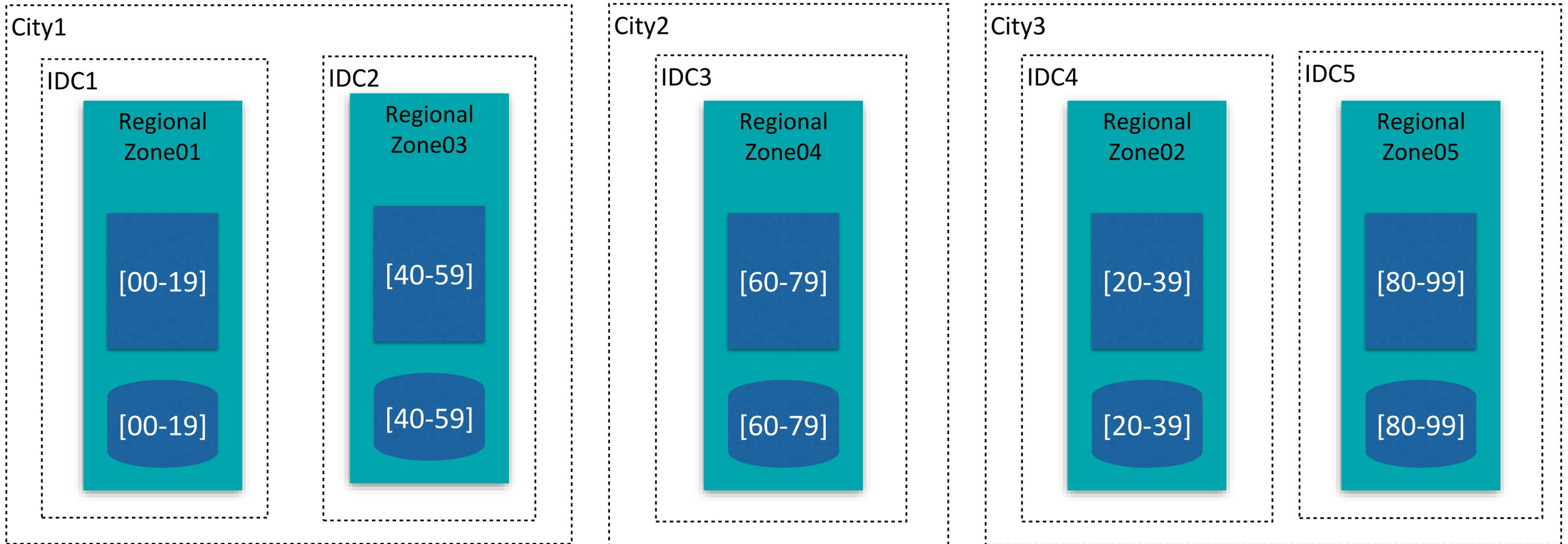
# 两地三中心的一种典型部署



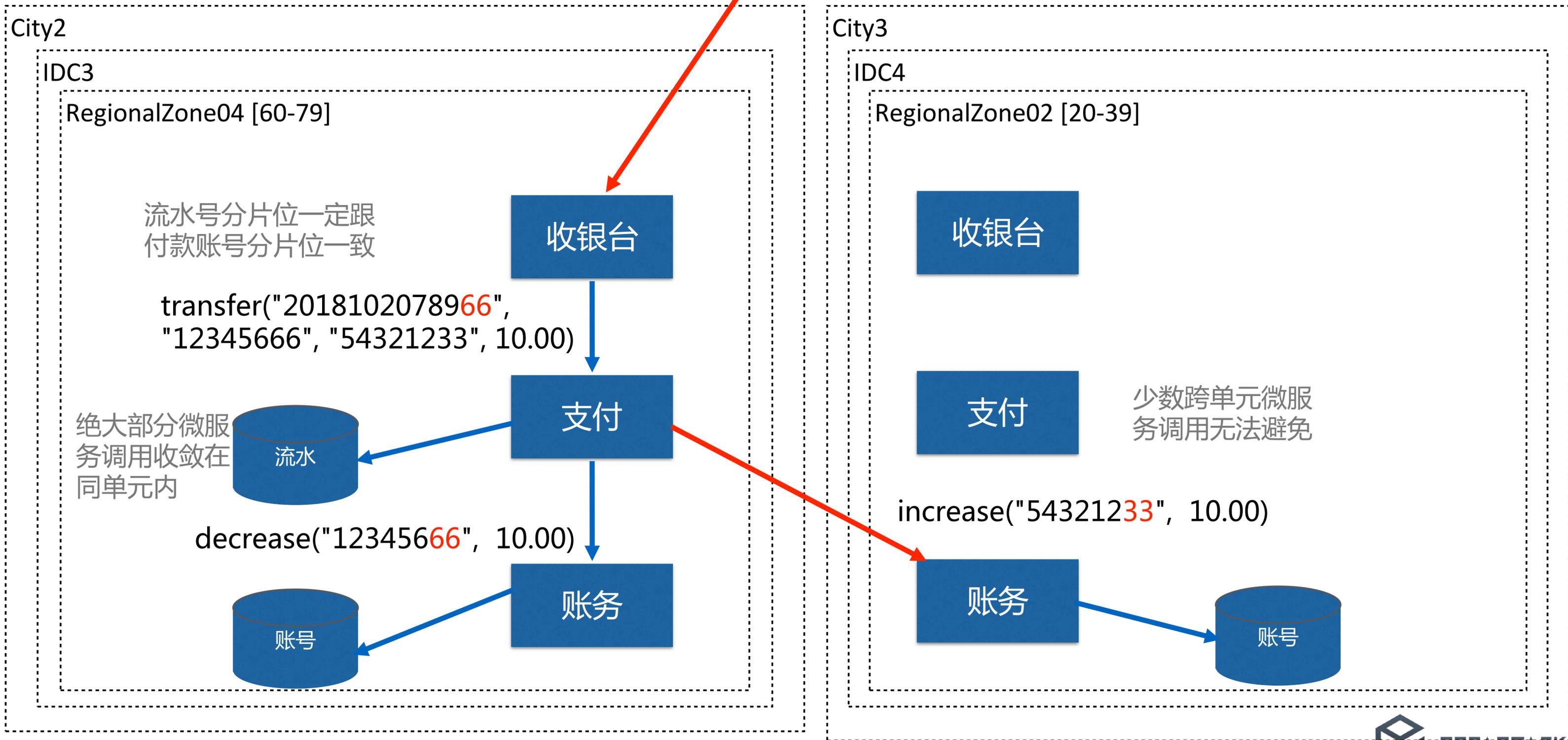
# 向三地五中心演进

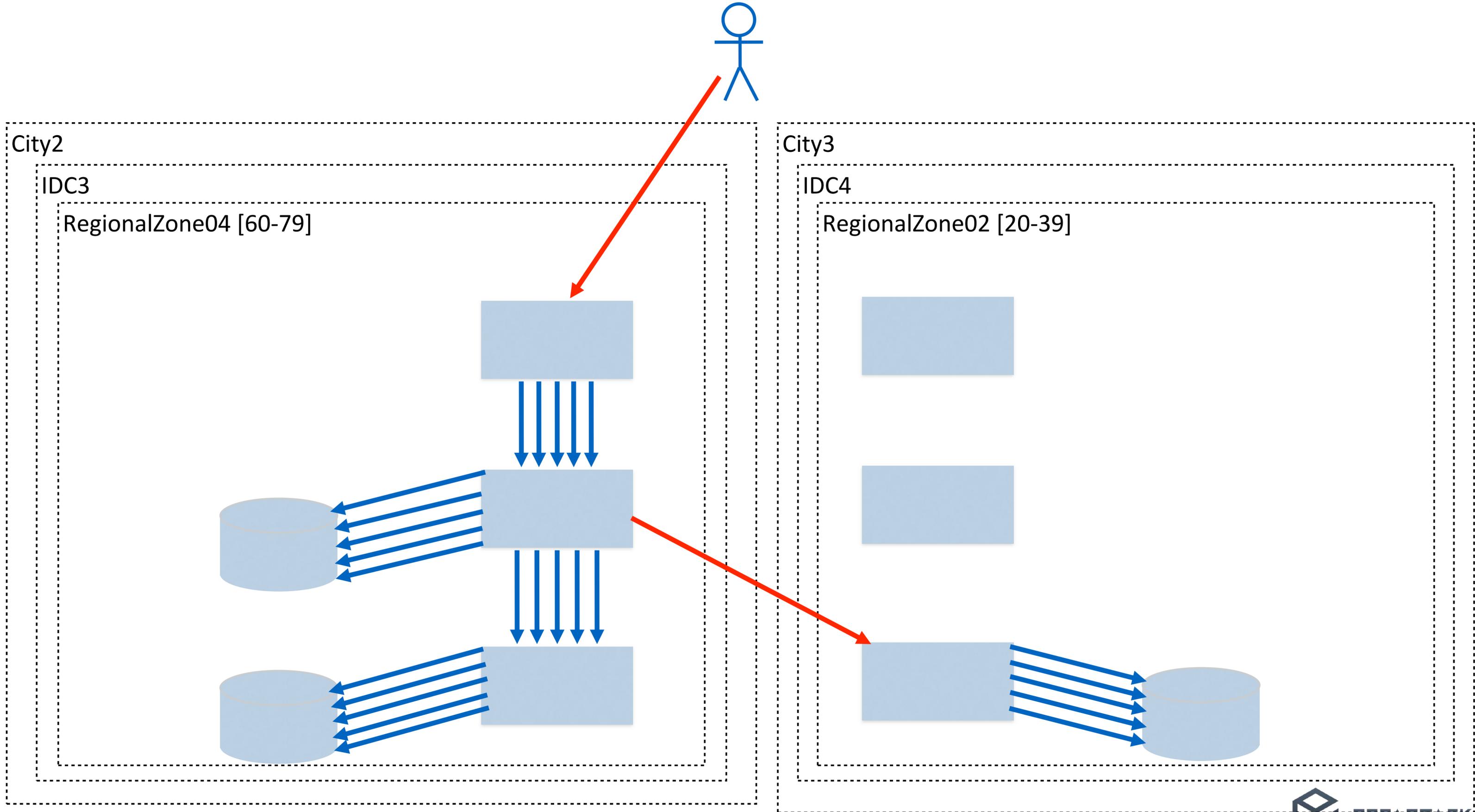


# 三地五中心的一种典型部署

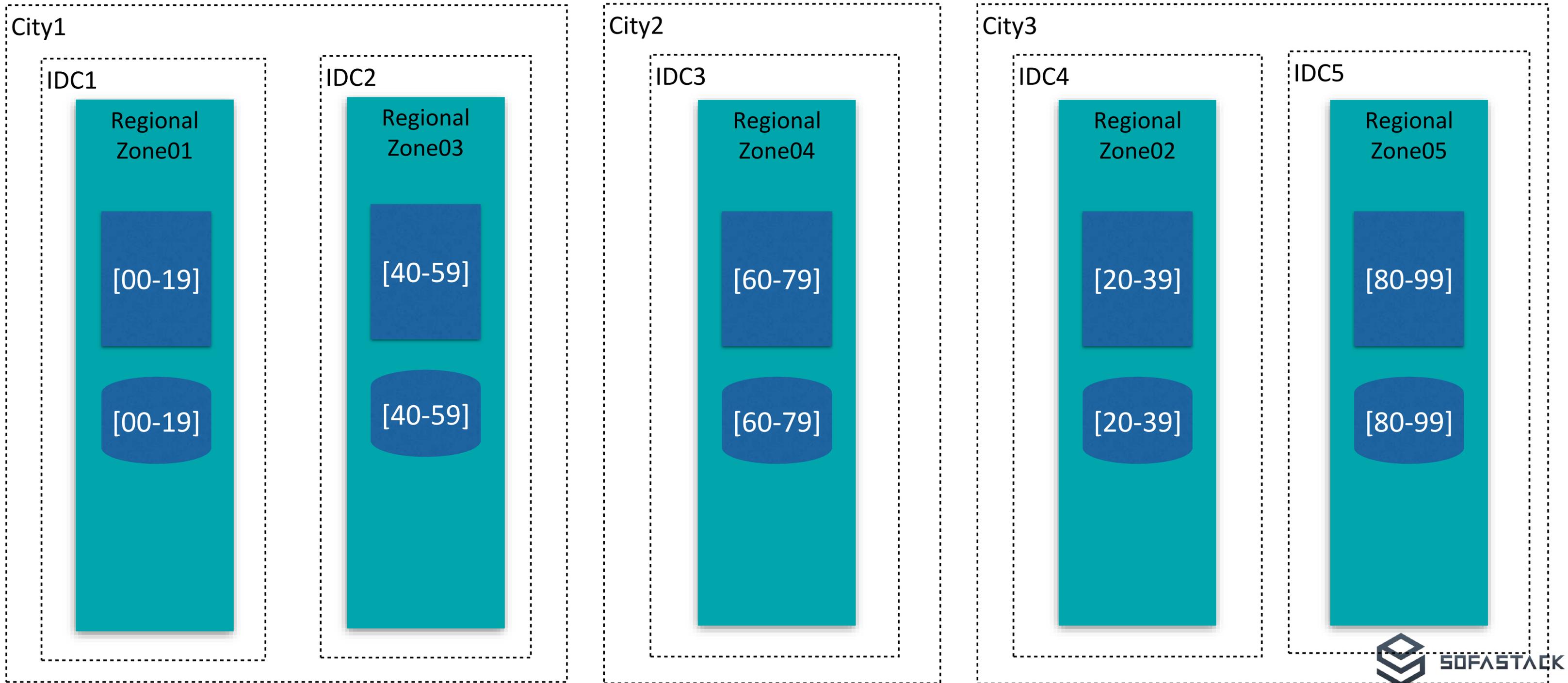


当前用户 12345666

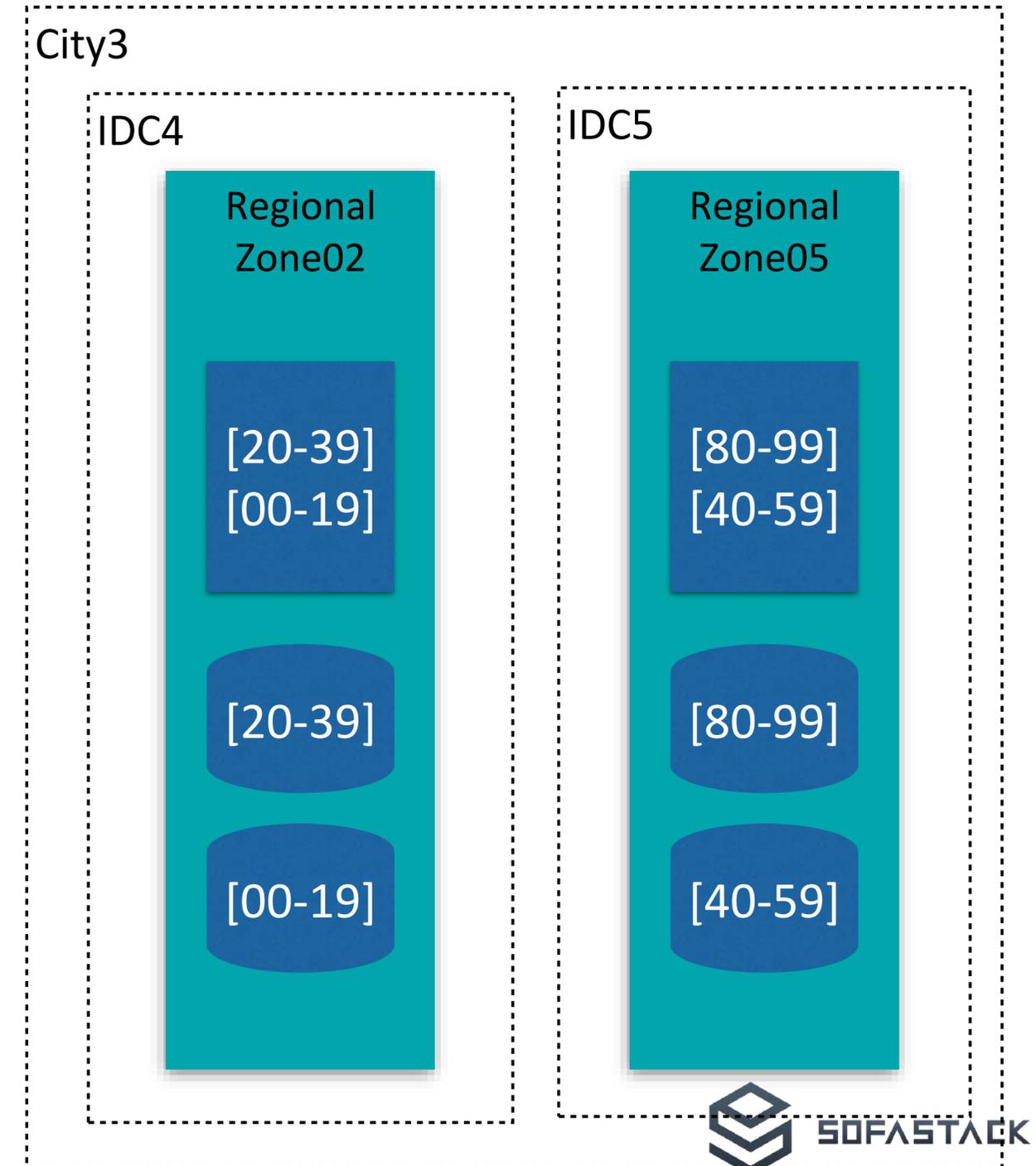
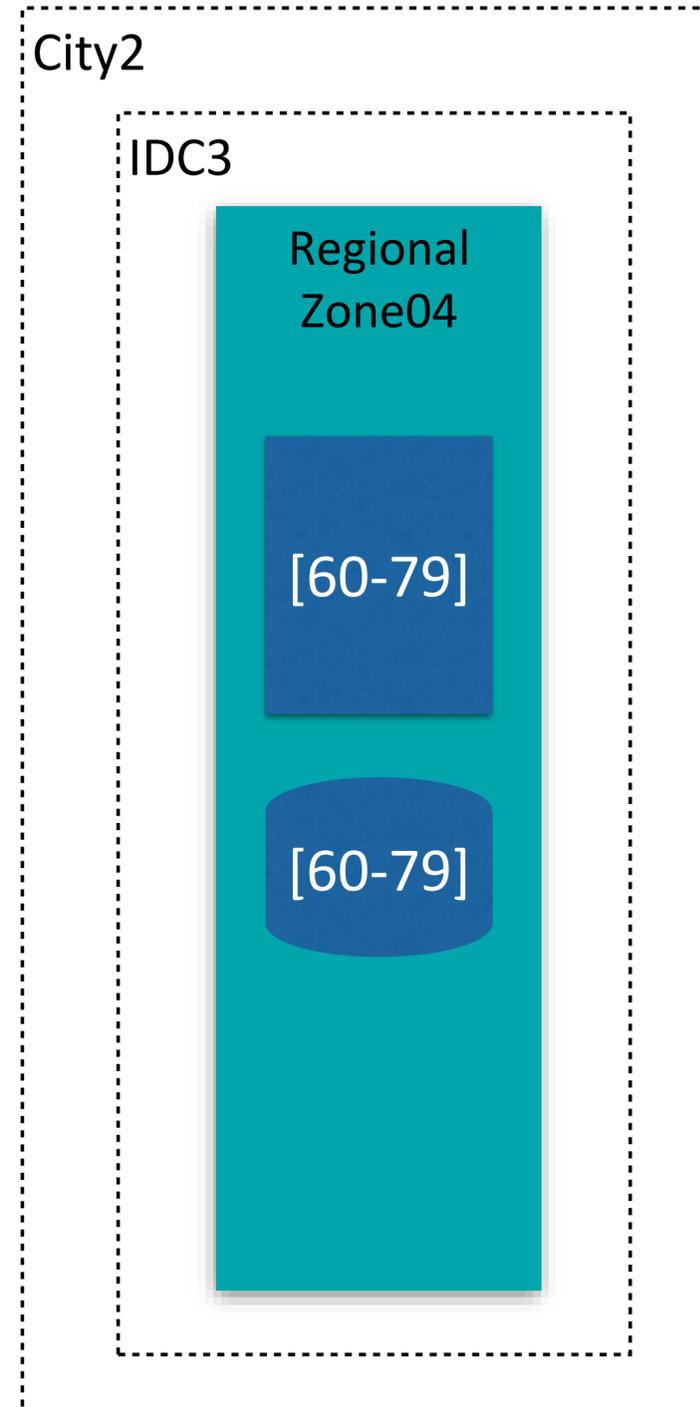
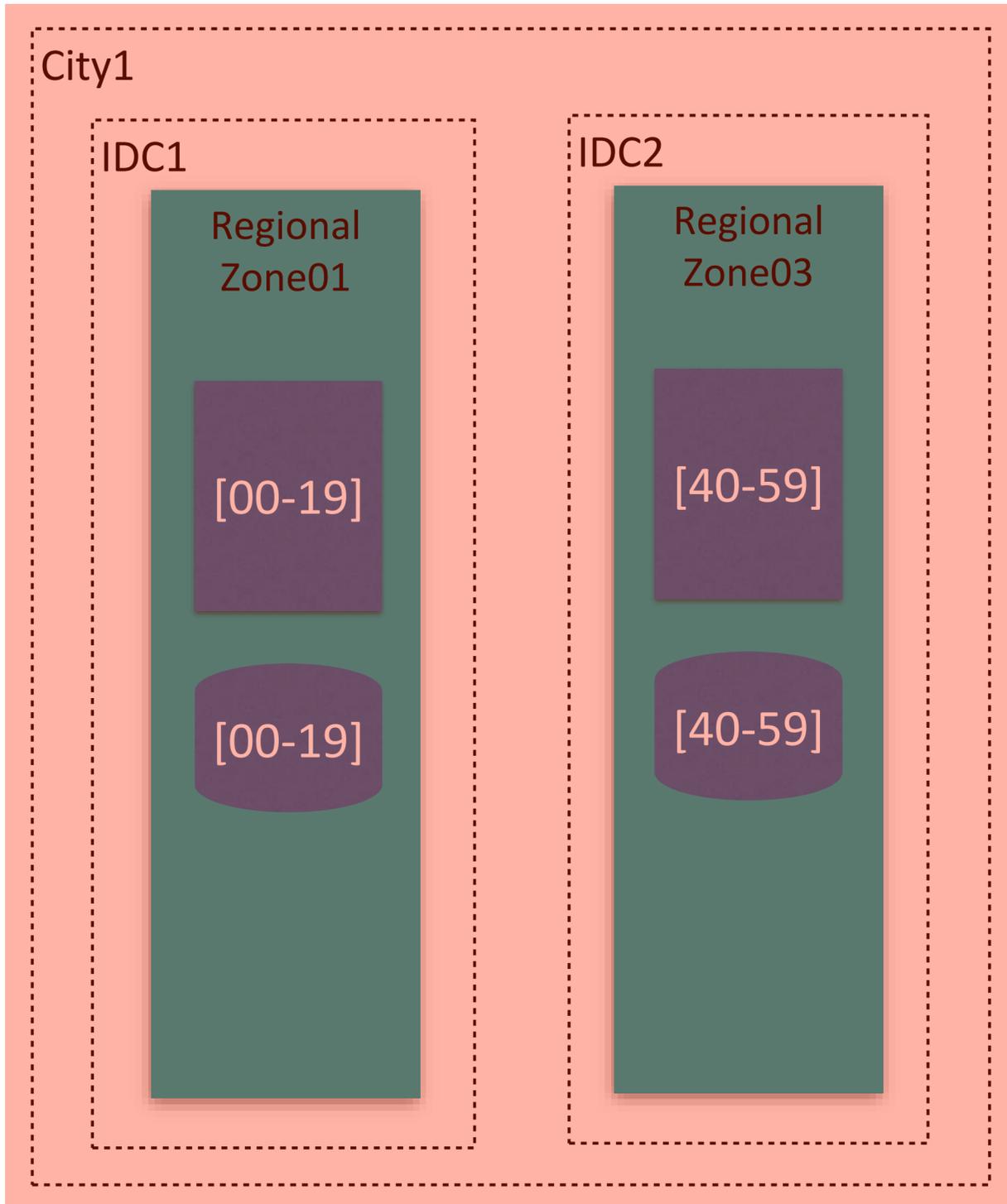




# 城市级容灾



# 城市级容灾



# 单元化设计原则

- 核心业务单元化
- 保证核心业务单元分片均衡(UID)
- 核心业务尽量自包含（调用封闭）
- 面向逻辑分区设计，而不是物理部署

# 技术组件



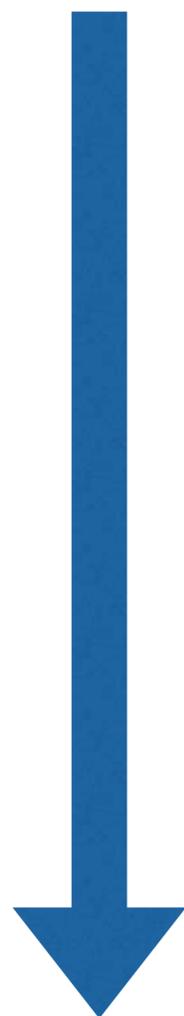
# 单元化流量管控

多层防线，迷途知返

<https://cashiergtj.alipay.com/standard/lightpay/lightPayCasl>

支付宝 ALIPAY | 我的收银台

付款-转账



DNS层

多域名技术

反向代理层

代理转发

网关/WEB层

HTTP拦截器转发

服务层

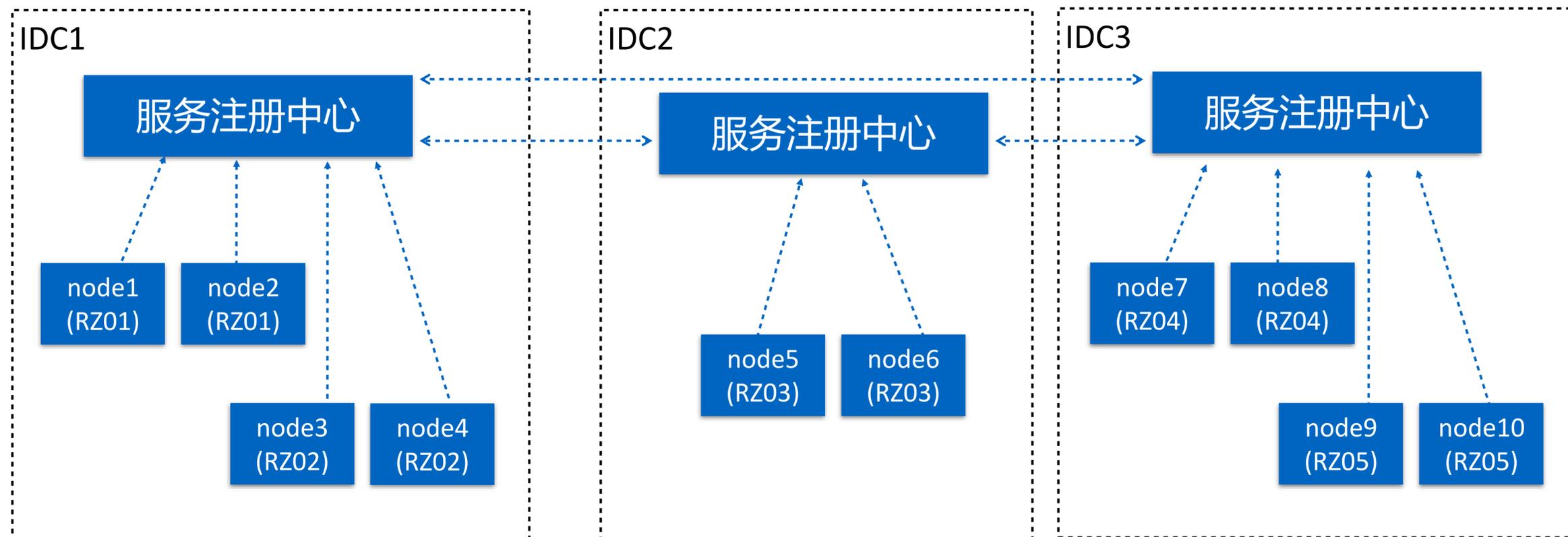
服务路由

数据访问层

最后兜底确保数据正确

# 全局服务注册中心

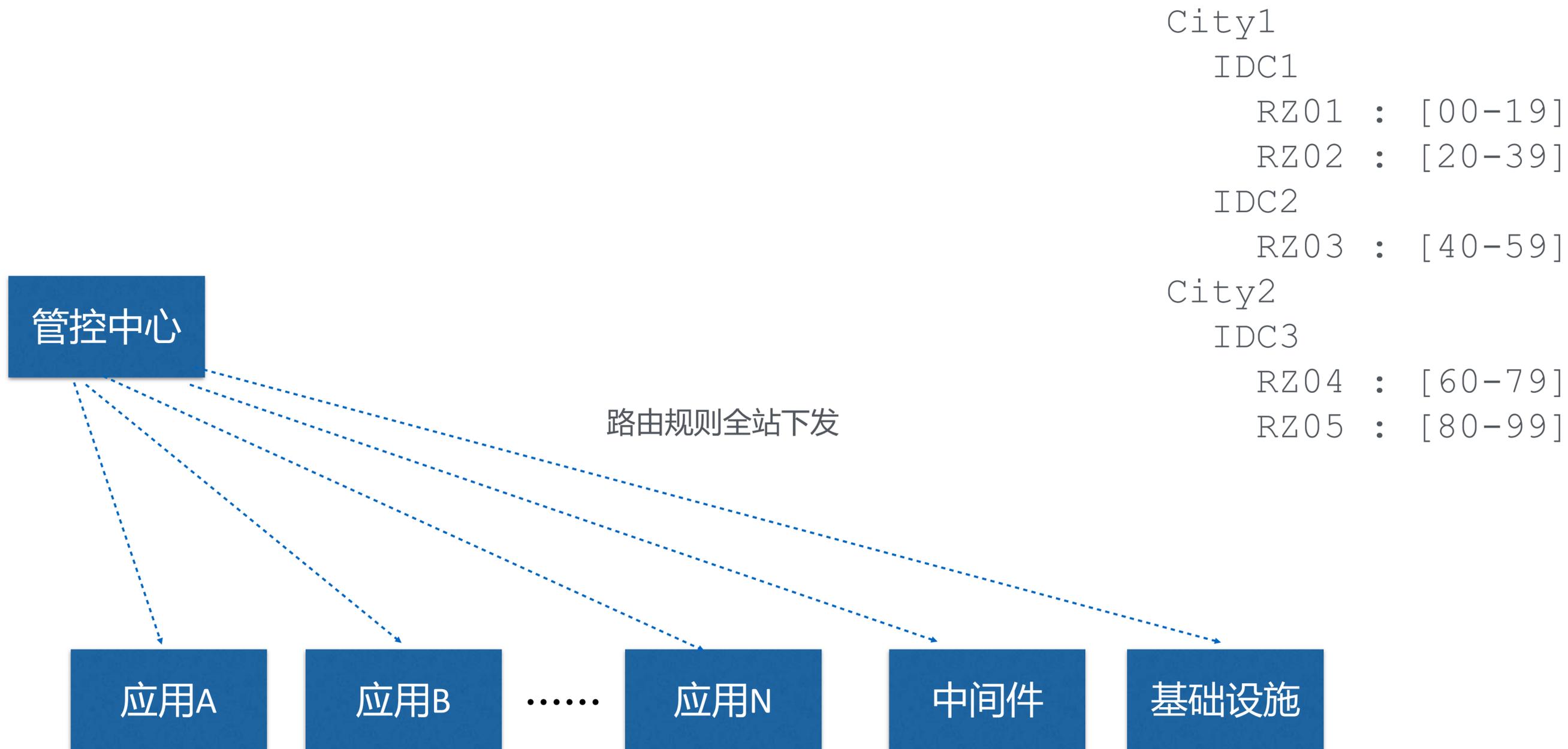
层次化服务注册 ( Tag实现亦可 )



Service1 :

- RZ01
  - node1
  - node2
- RZ02
  - node3
  - node4
- RZ03
  - node5
  - node6
- RZ04
  - node7
  - node8
- RZ05
  - node9
  - node10

# 统一路由规则



# 注解驱动的 RPC 路由

```
public interface TradeService {
    @ZoneRoute(uidGenerator = "com.alipay.trade.util.TradeUidGenerator")
    Result transfer(String tradeNumber, String buyerId, String sellerId, Double balance);
}

public class TradeUidGenerator implements UidGenerator {
    public String generateUid(Method method, Object[] args) {
        //参数校验及异常处理略.....
        String tradeNumber = (String) args[0];
        //从交易号截取末两位作为分片ID
        return tradeNumber.substring(tradeNumber.length()-2);
    }
}
```

# 注解驱动的 RPC 路由

```
@ZoneRoute(uidGenerator = "com.alipay.account.util.AccountUidGenerator")
public interface AccountService {
    //增加
    Result increase(String userId, Double balance);
    //扣减
    Result decrease(String userId, Double balance);
}

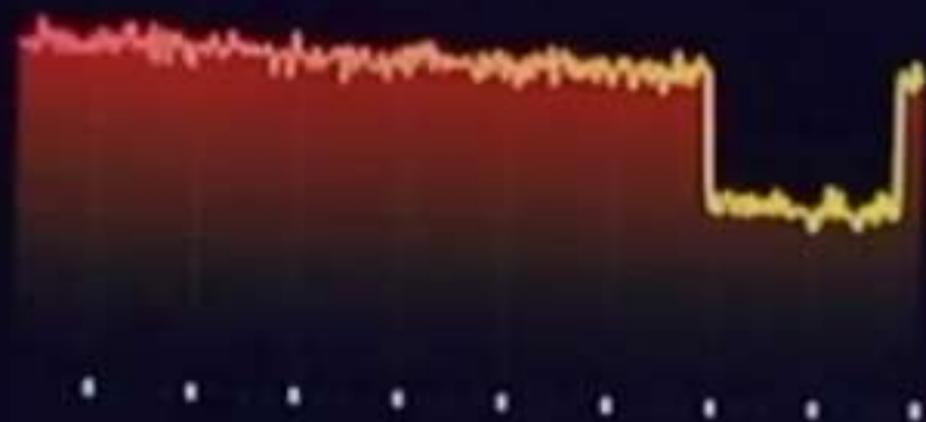
public class AccountUidGenerator implements UidGenerator {
    public String generateUid(Method method, Object[] args) {
        //参数校验及异常处理略.....
        String userId = (String) args[0];
        //从用户ID截取末两位作为分片ID
        return userId.substring(userId.length()-2);
    }
}
```

# 小结

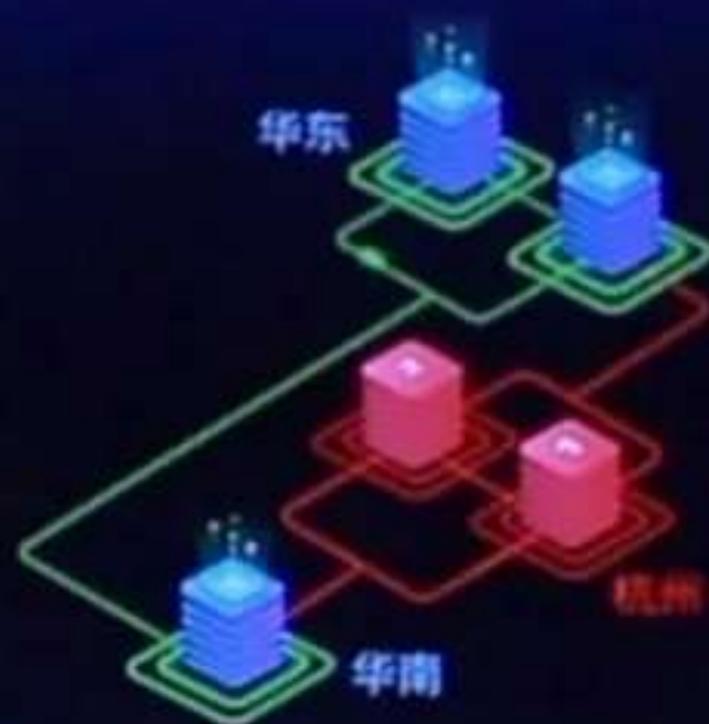
- 多层防线，迷途知返，尽可能早进入正确的单元
- 单元粒度的服务发现
- 全局统一路由规则
- 业务透明的单元化 RPC 框架
- DAL层最后兜底，确保数据正确性

# 转账交易大盘 Transaction Dashboard

转入金额 ¥ 36,054,055.00  
转出金额 ¥ 36,054,055.00



- 数据库切换 ●
- 缓存容灾切换 ●
- 多活规则切换 ●
- 中间件切换 ●
- 负载均衡切换 ●
- 域名解析切换 ●



为世界带来微小而美好的改变

# THANKS!

绿色食品

无明矾油条

拓展阅读

公众号：金融级分布式架构

<http://www.sofastack.tech>

