

# 双11——淘宝下一代架构的成人礼

梁耀斌

全局架构技术专家 - 阿里技术保障

ArchSummit / 12月19日



阿里技术保障  
All Infrastructure Service

# 双11的印象



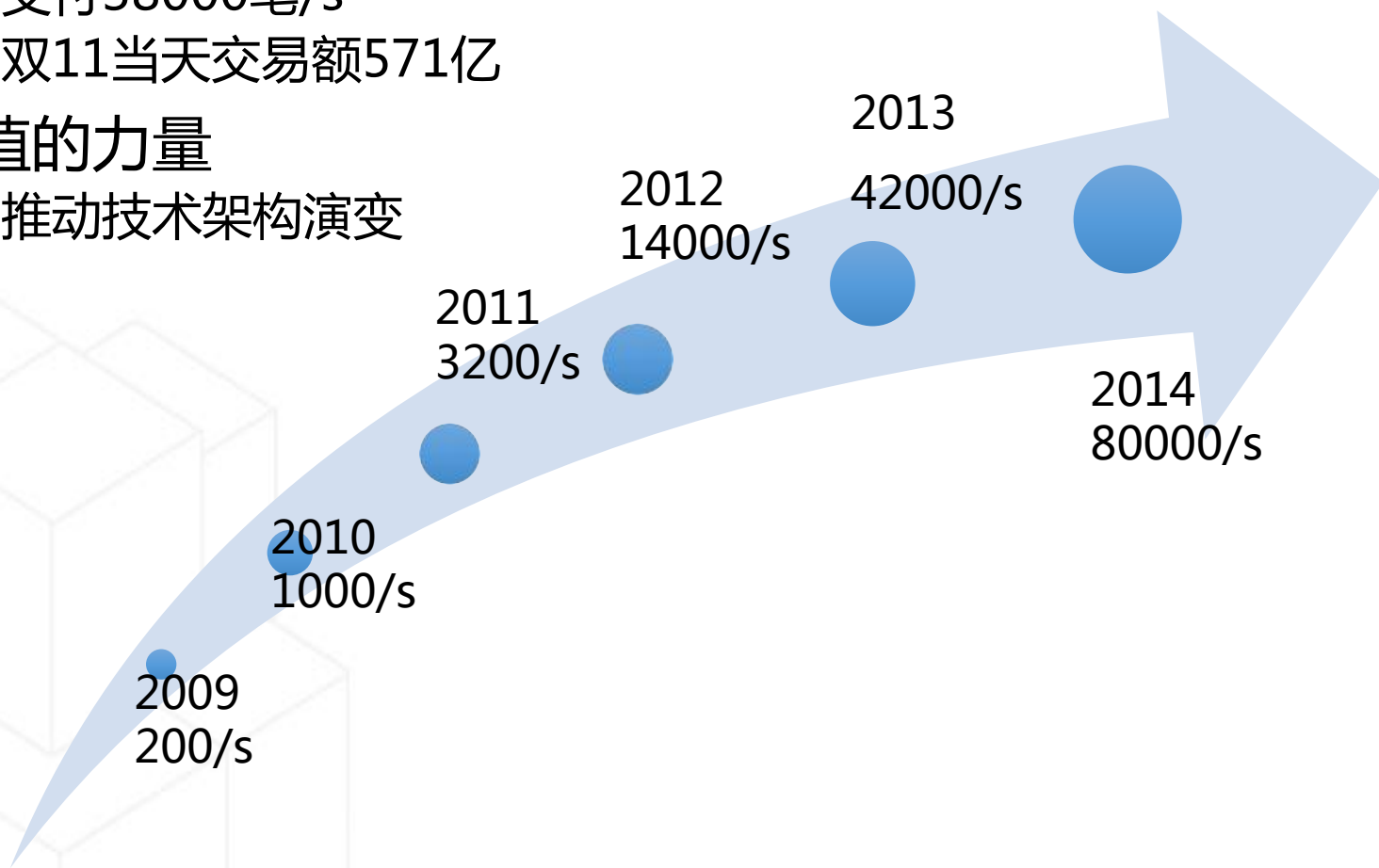
# 双11的印象

## ■澎湃，震撼

- 交易创建80000笔/s
- 支付38000笔/s
- 双11当天交易额571亿

## ■峰值的力量

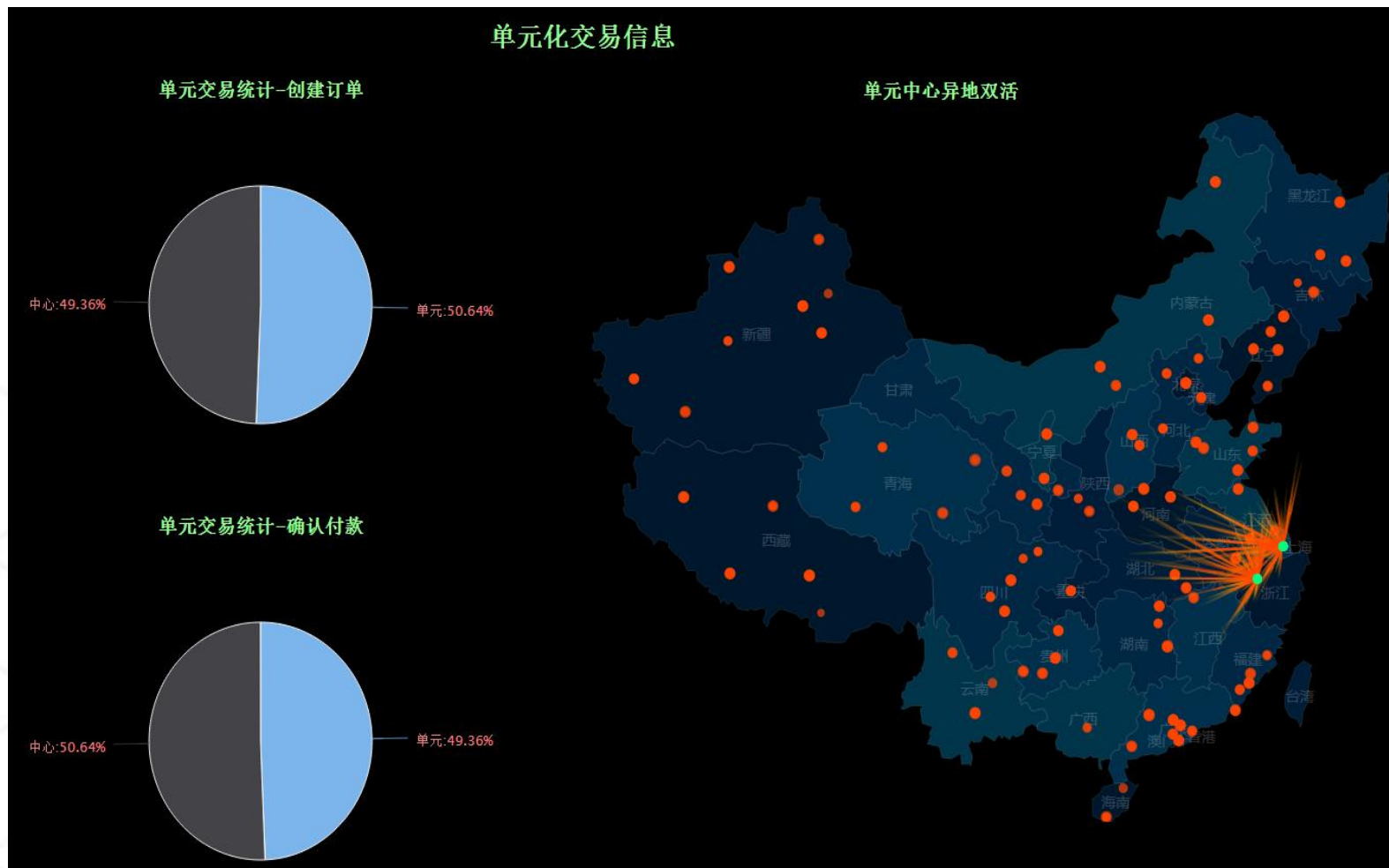
- 推动技术架构演变



# 架构工程师的烦恼



# 异地双活



# 过去的演变

## ■ 2.0时代(2007)

- 单应用
- 业务排期长
- 开发效率低
- 不能加机器，业务再增长就悲剧了





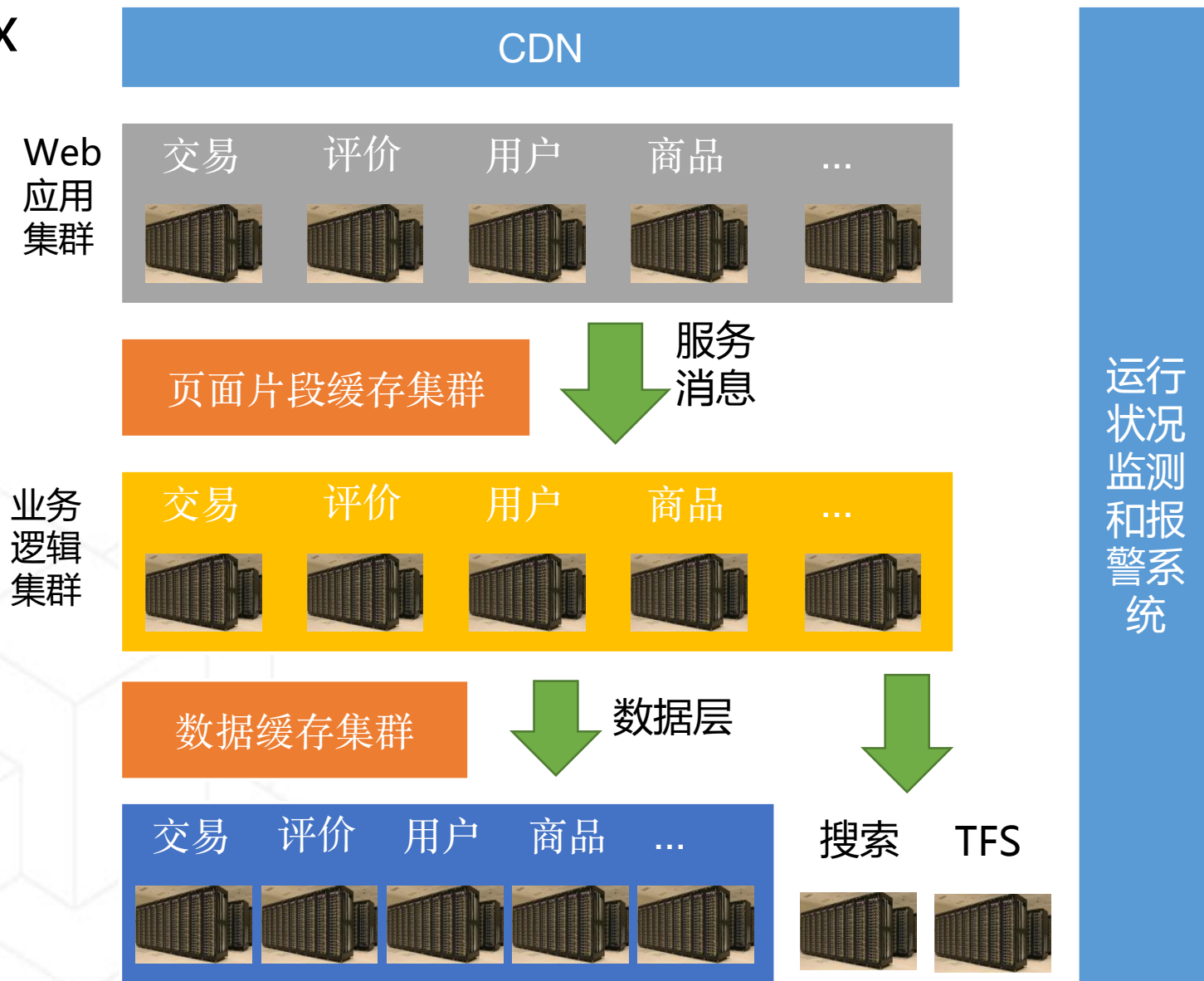
# 过去的演变

## ■ 2.0 -> 3.x (2007-2009)

- 单个应用 -> 大型分布式java应用服务化
- 分库分表
- 分布式cache
- 分布式文件系统
- 稳定性的关注



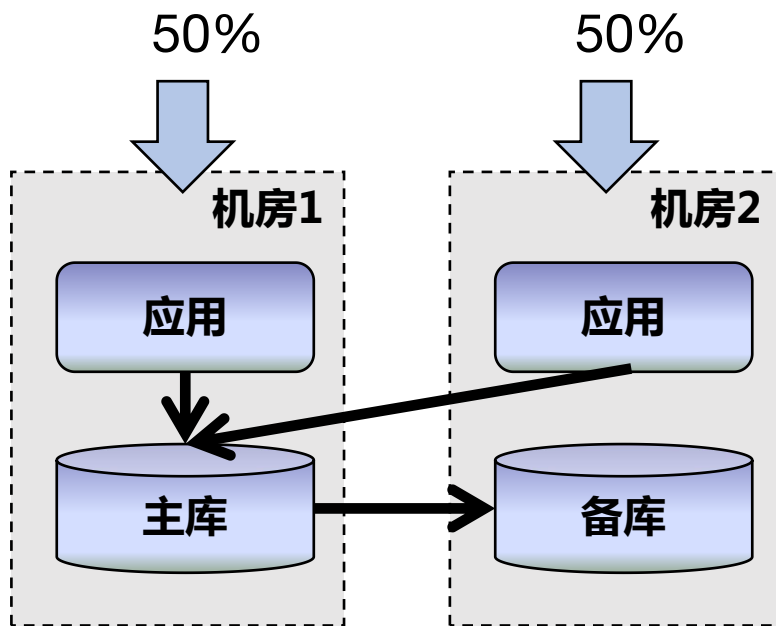
3.x





## 3.x时代容灾方面的一些改进

- 同城多机房的容灾
- 异地备份机房



# 问题又来了

## ■扩展性

- 系统水平伸缩

## ■资源限制

- 一个城市已经不能满足需求

## ■容灾

- 单地域机房风险

## ■业务需求

- 国际化

# 最大的挑战

## ■延迟











































































- 同一机房0.2ms
- 同一城市1ms
- 跨城市10ms~100ms

## ■对同步调用的影响

- 几百次的调用
- 并发的下降

## ■数据

- 多维度
- 实时性
- 一致性

应用名	类型	状态	大小	单元化	服务/方法	
	TRACE	OK	-			194ms
	TAIR	OK	318B	单元内		1ms
	TAIR	OK	268B	单元内		0ms
	TAIR	OK	242B	单元内		0ms
	TAIR	OK	1.6KB	单元内		1ms
	TAIR	OK	140B	单元内		1ms
	TAIR	OK	589B	单元内		0ms
	TAIR	OK	288B	单元内		0ms
	TAIR	OK	3.0KB	单元内		1ms
	TAIR	OK	248B	单元内		1ms
	TAIR	PARTSUC	1.9KB	单元内		1ms
	TAIR	OK	589B	单元内		0ms
	TAIR	OK	288B	单元内		0ms
	TAIR	OK	140B	单元内		0ms
	TAIR	OK	517B	单元内		0ms
	TAIR	OK	264B	单元内		0ms
	TAIR	OK	3.1KB	单元内		1ms
	HSF	OK	654B	单元内		1ms
	HSF	OK	639B	单元内		0ms
	TAIR	OK	1.6KB	单元内		1ms
	TAIR	OK	140B	单元内		0ms
	TAIR	OK	161B	单元内		1ms
	HSF	OK	1.1KB	单元内		2ms
	TDDL	OK	-	单元内		1ms
	HSF	OK	1.1KB	单元内		2ms
	TDDL	OK	-	单元内		0ms
	HSF	OK	6.8KB	单元内		18ms
	TAIR	OK	130B	合理跨单元		0ms
	TDDL	OK	-	单元内		0ms
	TAIR	OK	517B	单元内		1ms
	TAIR	OK	242B	单元内		0ms
	TAIR	OK	589B	单元内		1ms
	TAIR	OK	288B	单元内		0ms
	TAIR	OK	255B	单元内		0ms
	TAIR	OK	264B	单元内		1ms
	TAIR	NOTEXSI	77B	合理跨单元		0ms
	TAIR	OK	2.9KB	单元内		1ms

## 怎么拆？

### ■关键是数据

#### ■单点写

#### ■数据拆分

### ■单元的定义

#### ■交易链路

#### ■中心

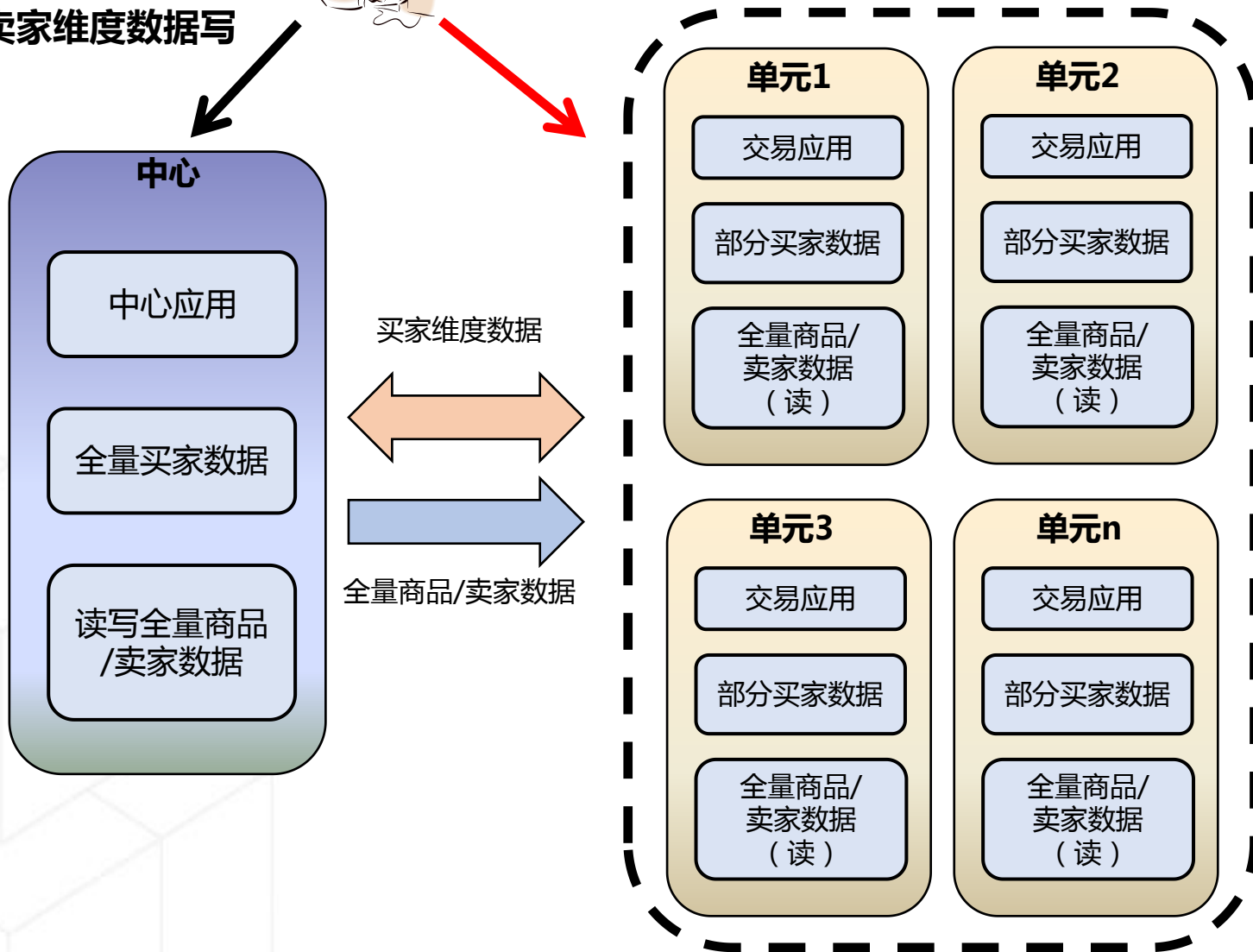
### ■最大原则——单元封闭



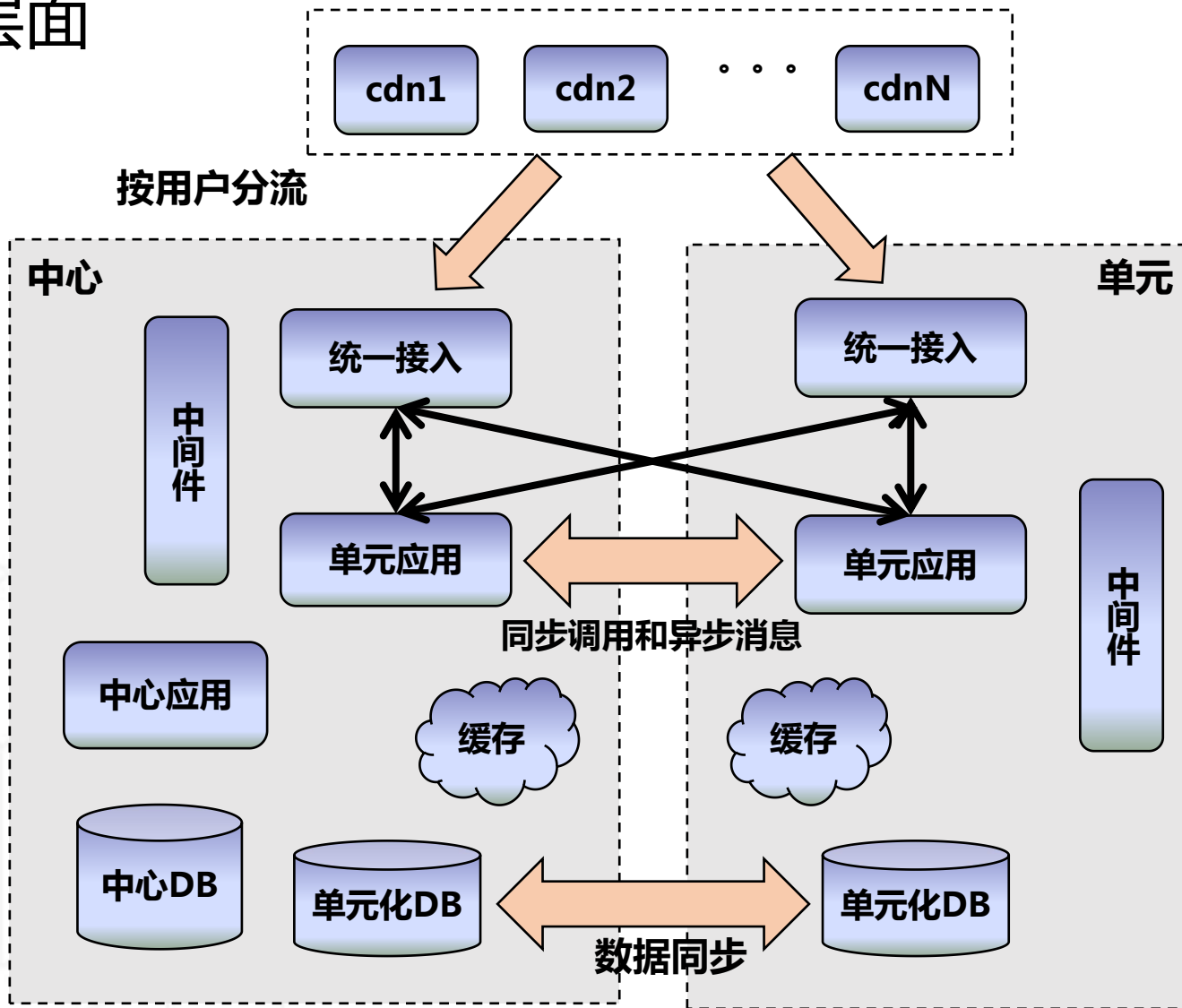
# 业务层面

卖家维度数据写

买家交易在单元内完成读写



# 架构层面





# 实现要点

## ■ 链路梳理

- 调用依赖
- 单元封闭



# 实现要点

- 统一路由管理
  - 统一接入层
  - 去中心化rpc框架
  - 异步消息



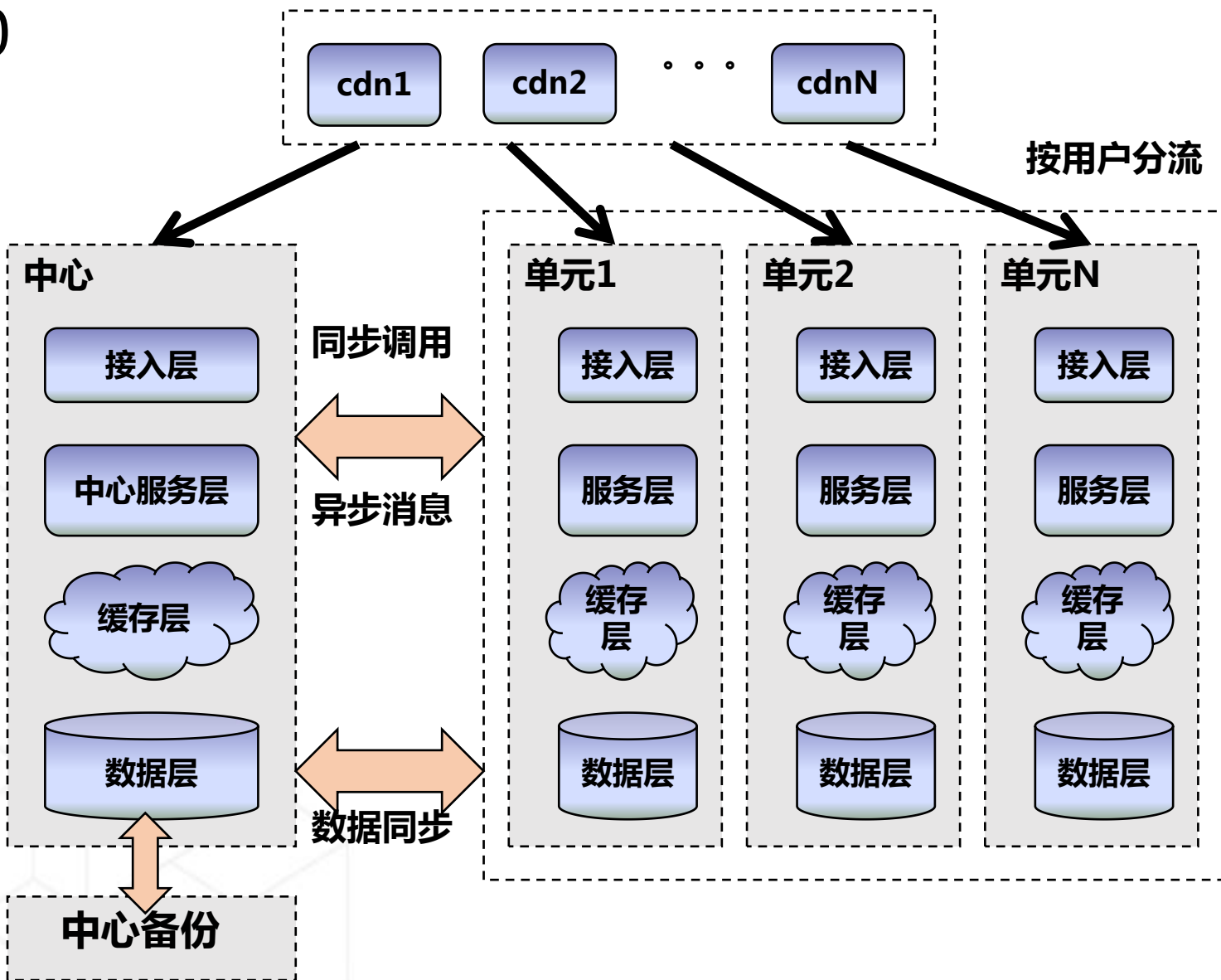
# 实现要点

## ■数据同步

- 跨地域数据同步工具
- 数据全量和增量一致性校验
- 数据同步延迟监控



# 4.0



# 小结

## ■数据拆分

- 按一个维度拆分数据

## ■单元封闭

- 链路梳理

## ■全局路由

- 统一管理

## ■数据保障

- 延迟和一致性监控

# 收益

## ■扩展性

## ■容灾

## ■稳定性

### ■部分发布

### ■小规模验证

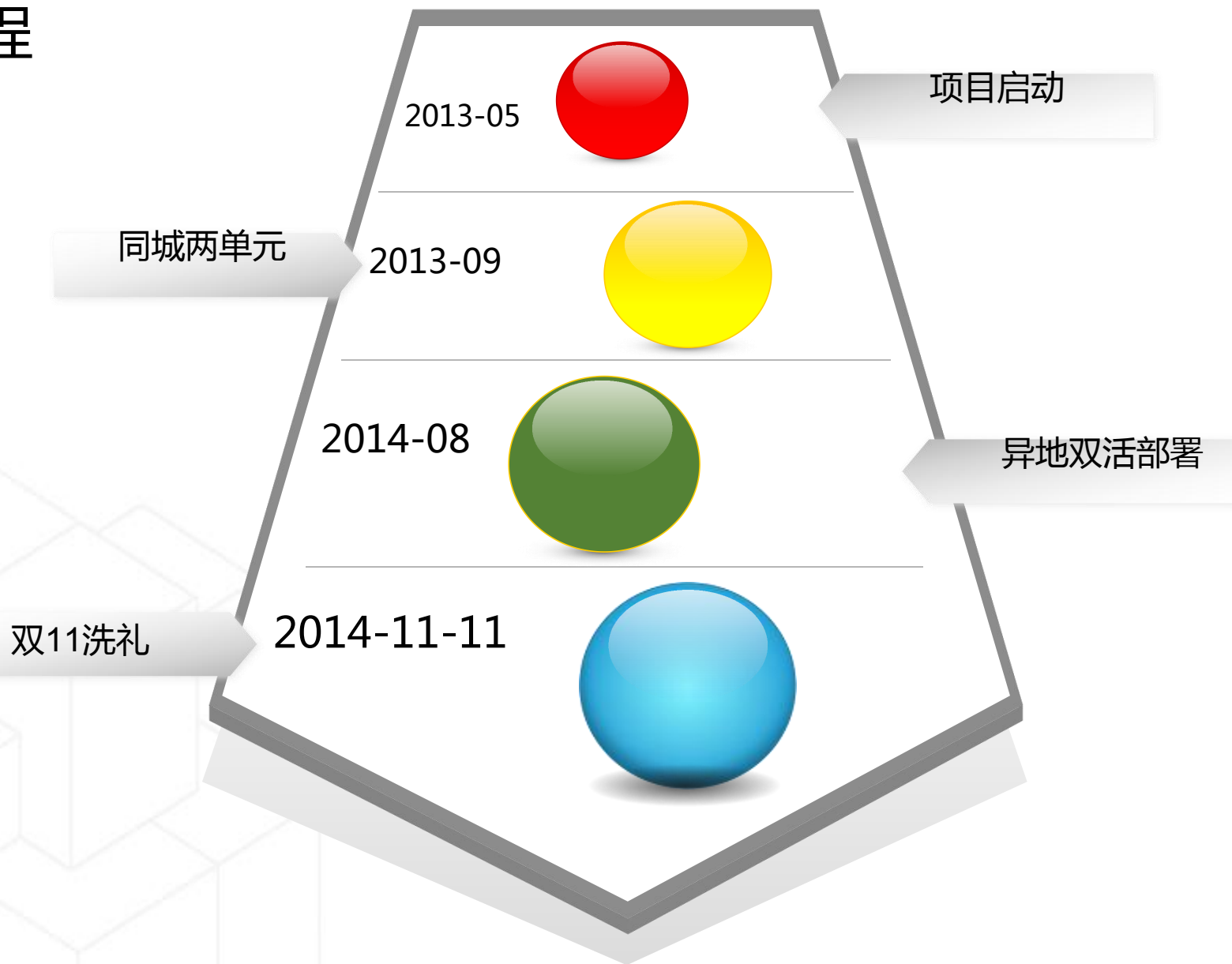
## ■易伸缩

### ■摆脱机房的限制

### ■简化容量规划



# 历程



# 双11备战

## ■链路分析

- 0点峰值行为的分析
- 减少跨单元调用
- 强一致需求

## ■容量预估

- 不同单元的机器机型不同
- 不同单元的机器数不同

## ■核心监控

- 核心业务数据
- 调用链路延迟
- 数据同步延迟
- 数据校验

# 双11备战

## ■容灾预案

- 机房故障
- 单元故障
- 跨地域网络故障

## ■全链路压测

- 8次模拟双11峰值模型的压力测试



# Thank You



@ 阿里技术保障