

房源中台建设中的服务化实践

贝壳找房-窦圣伟

2018.10



自我介绍

- 个人经历
 - 百度：知识管理
 - 豆瓣：豆瓣电影
 - 美团：猫眼电影/演出/专业版
 - 贝壳：房源平台
- 专注于
 - 业务/应用/技术架构
 - 领域驱动设计



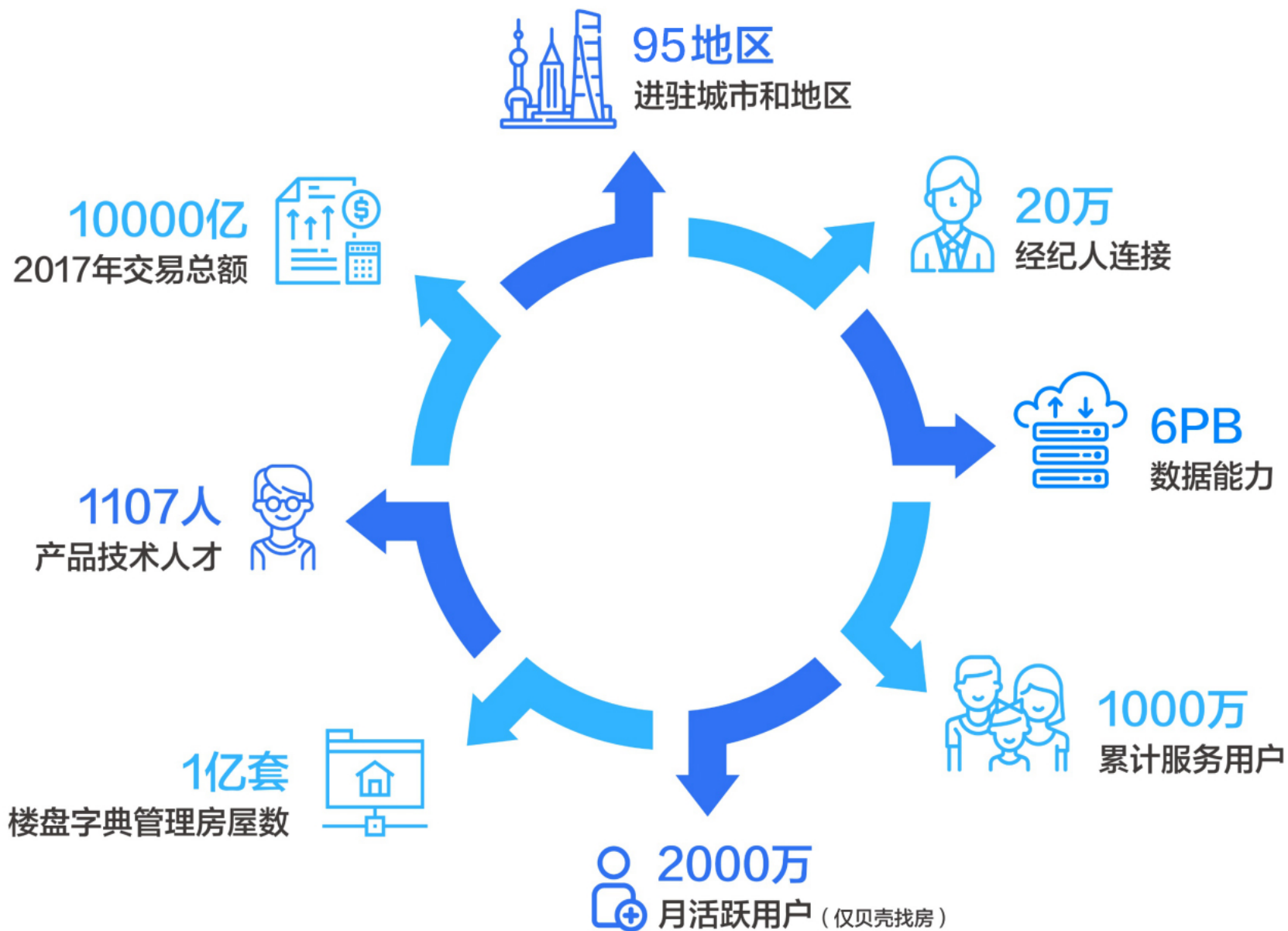
总览

- 关于“贝壳找房”
- 房源中台建设的理论支持
- 房源中台服务化演进实践

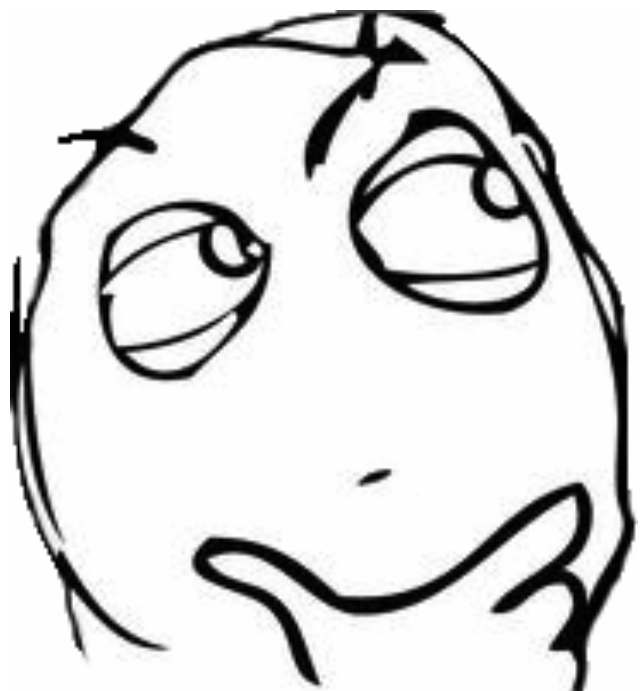
品牌架构



数说贝壳



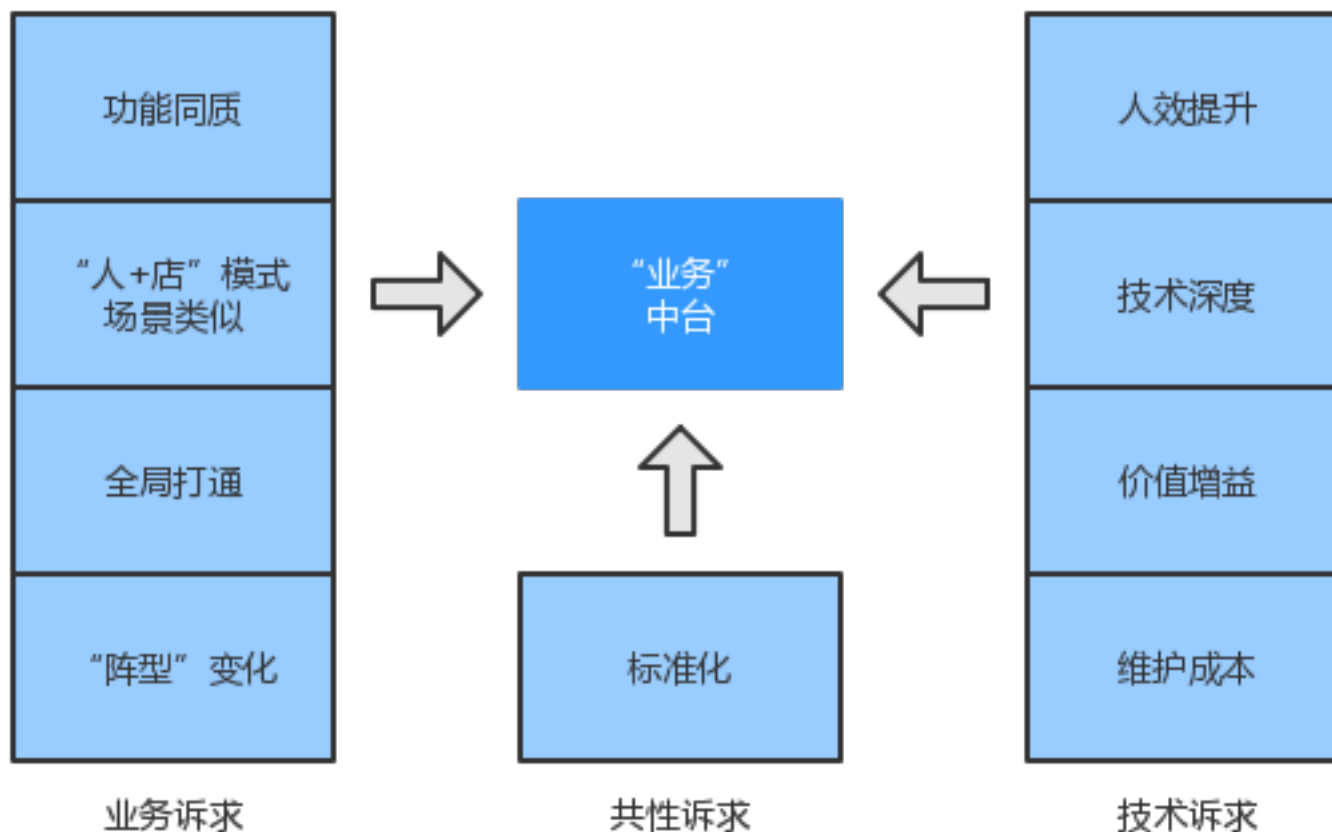
灵魂三问



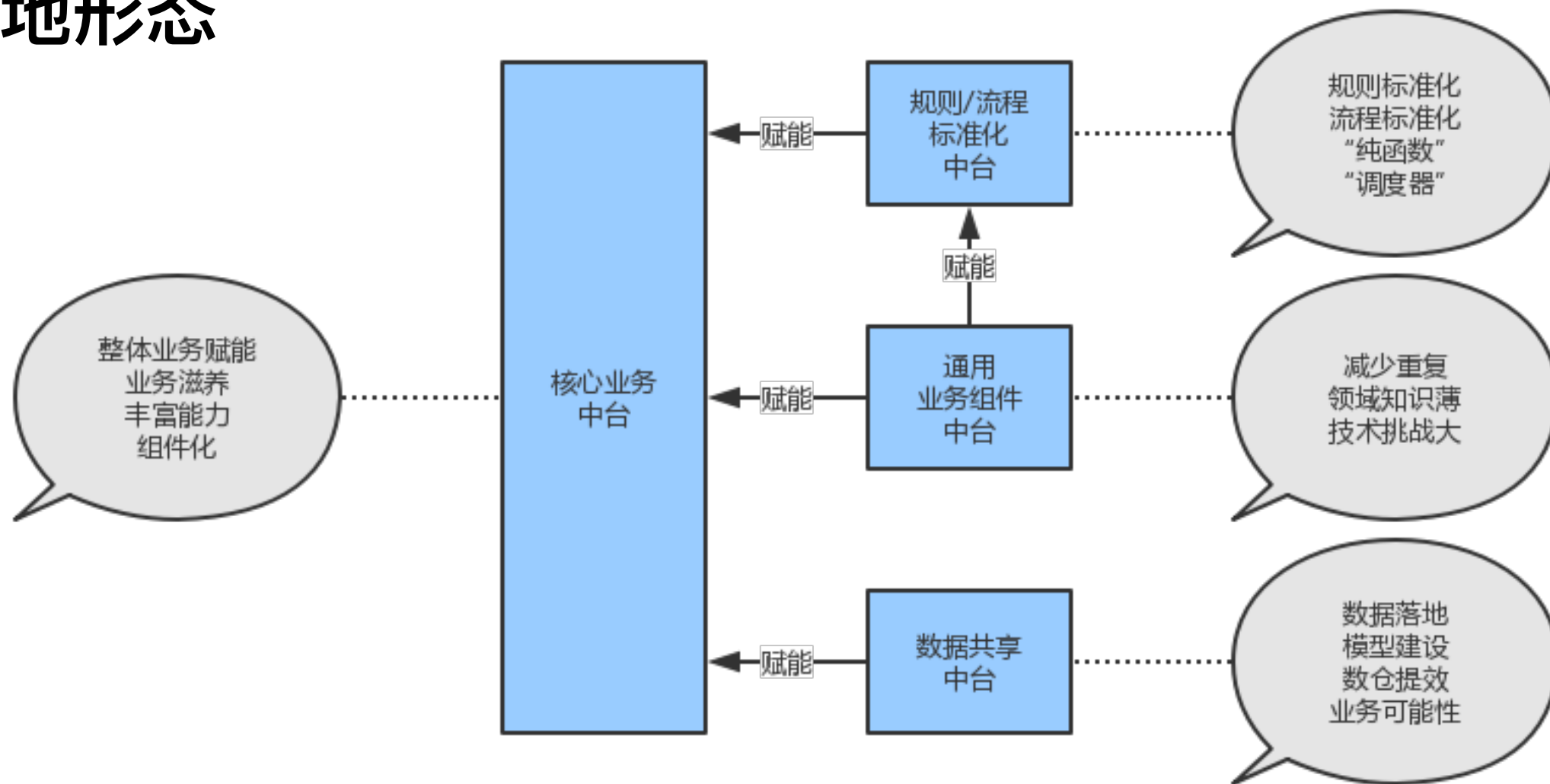
- “房源”+“中台”
 - 什么目标?
 - 怎么落地?
 - 从哪搞起?

■ 建设目标

- 堆砌基础设施？ NO！
- 仅做基础业务？ NO！
- 要做的
 - 进入业务纵深
 - 全局抽象打通
 - 支撑前台快跑
 - 输出标准规范



落地形态

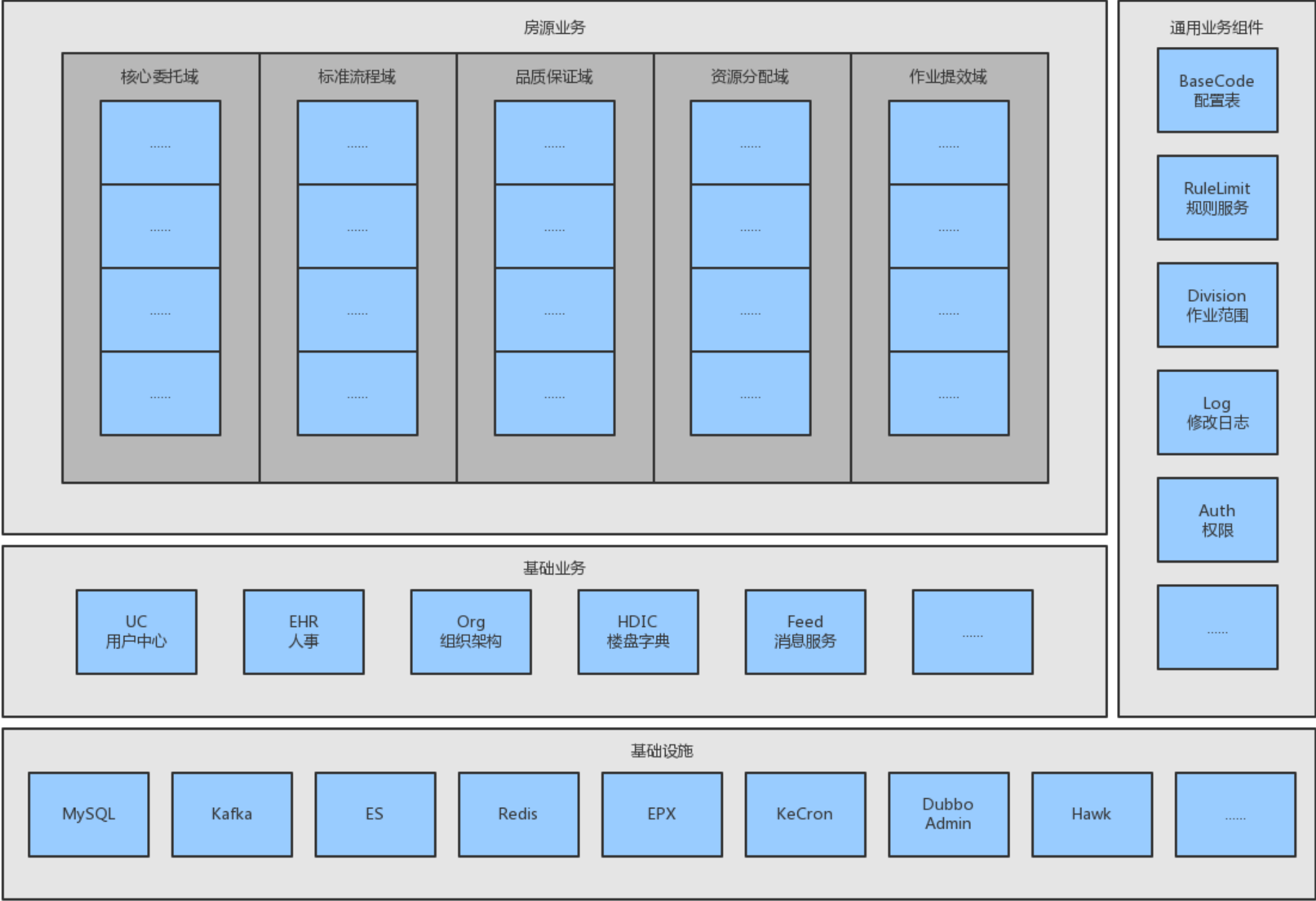


出发点

- 从服务能力/服务组件出发
 - 服务层是建设重点
 - OpenAPI, 领域层之上
- 从共性数据能力下沉出发
 - 剥离共性数据/主数据
 - 重数据落地, 轻业务规则
- 从已有系统前后分离/组件化出发
 - 服务化拆分+模型升级
 - 对既有业务的大力改造
- 我们的选择
 - 数据下沉+系统改造

■ 业务蓝本

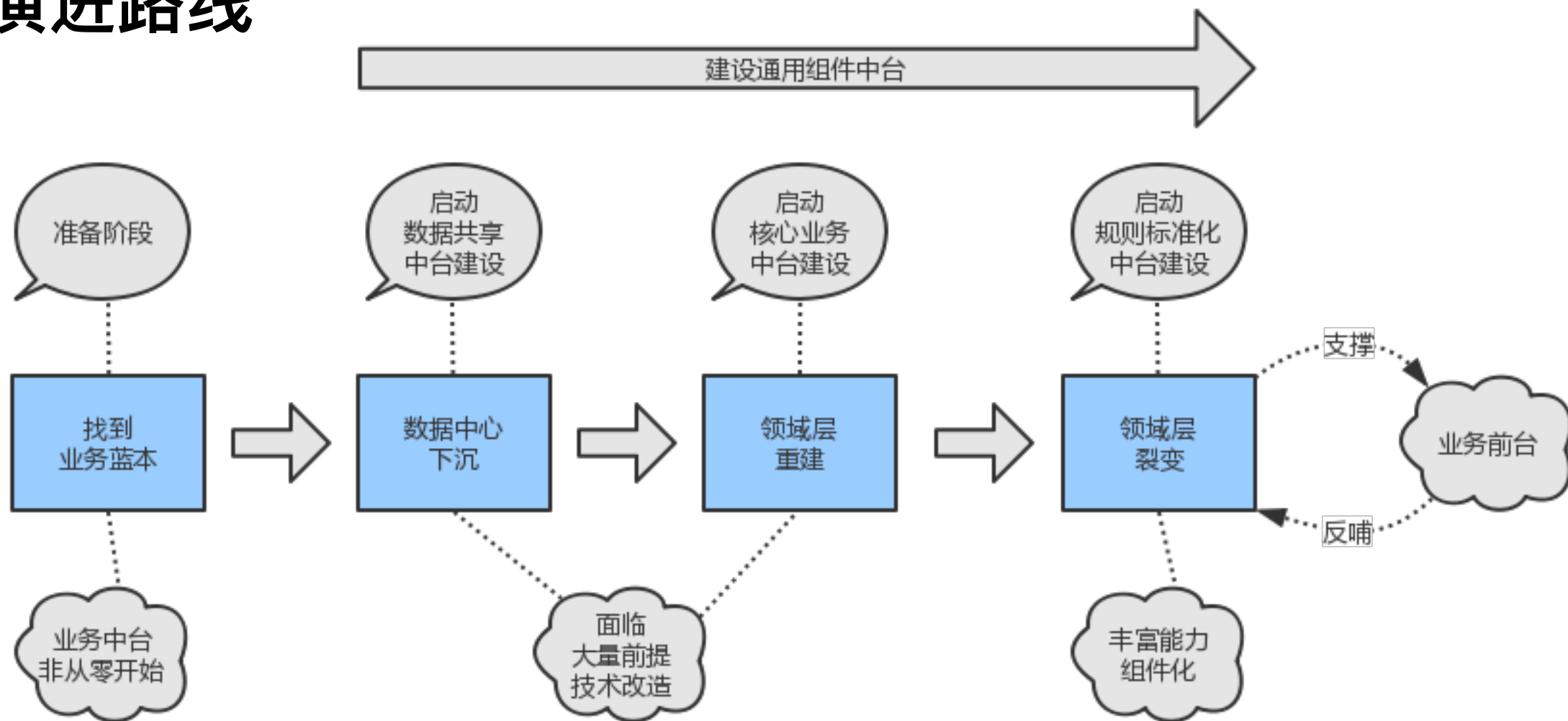
- 理想中的蓝本
 - 核心业务——找到抓手
 - 久经考验——稳定性
 - 模型完善——宏观思考
- 我们的选择
 - 直营房源系统



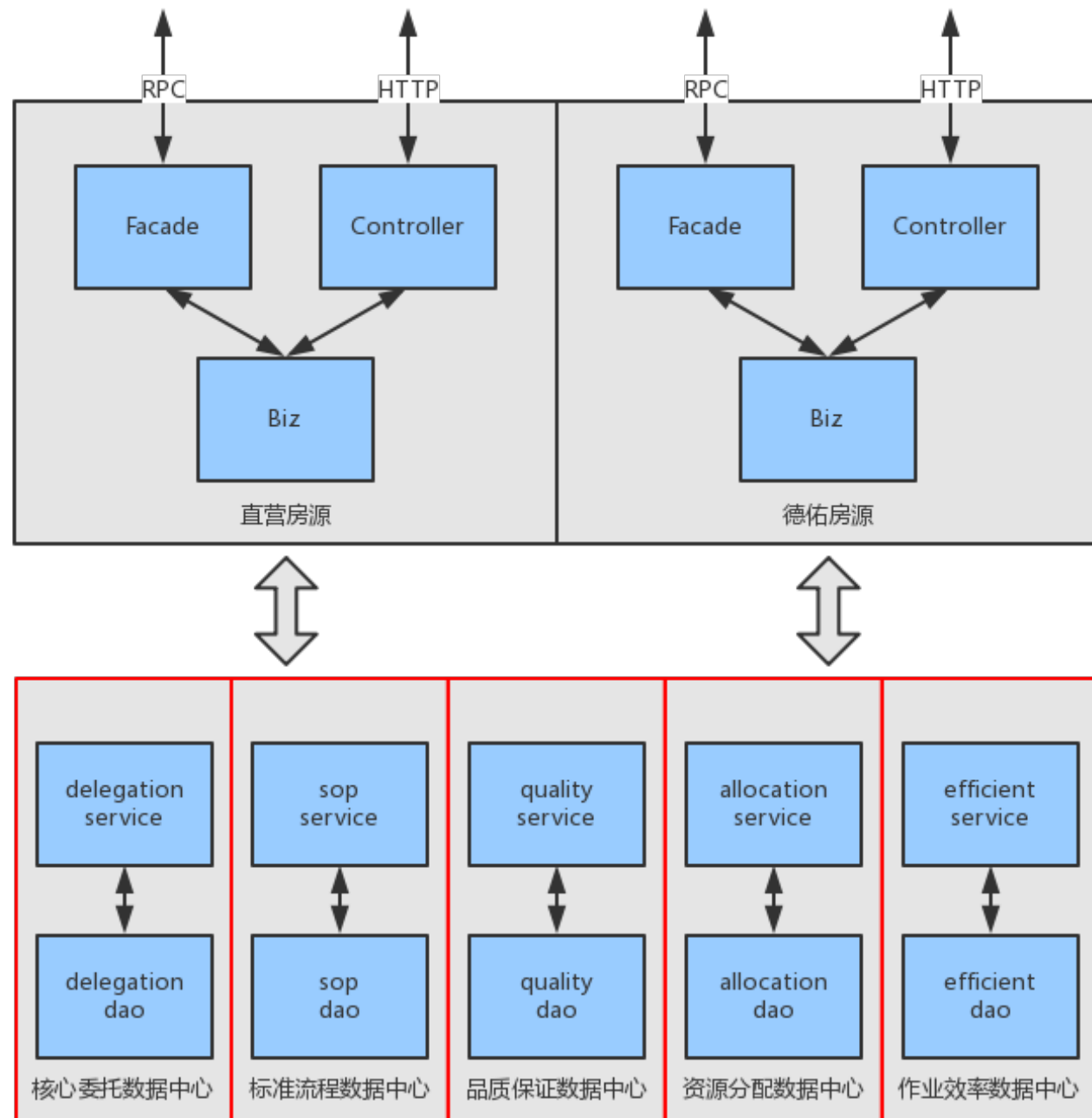
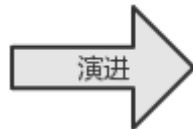
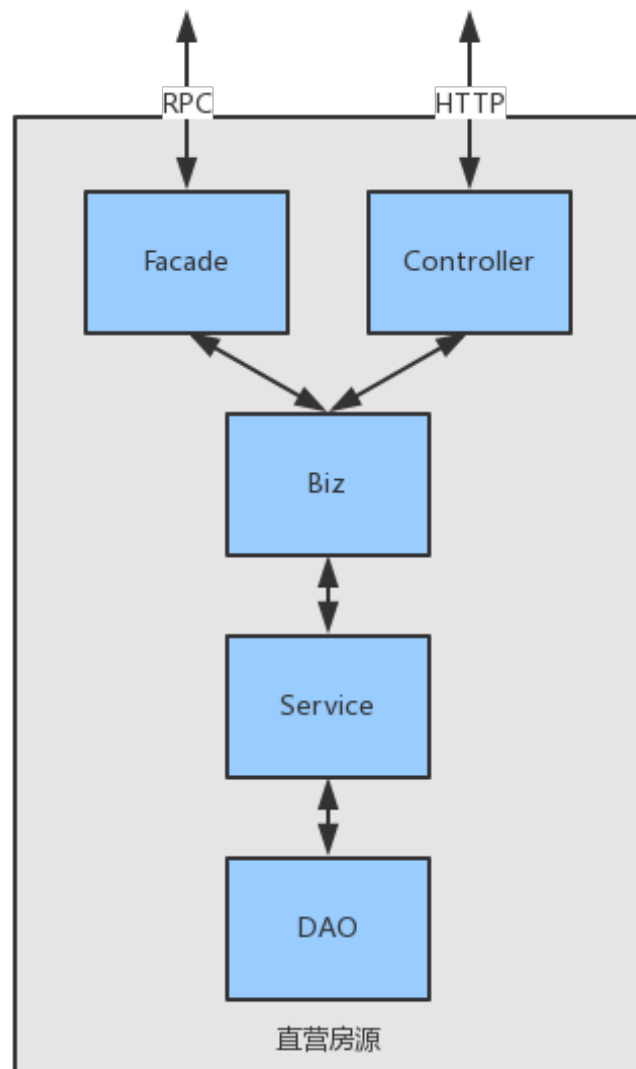
■ 面临挑战

- 技术+业务
 - 系统服务化不够，改造难度大
 - 底层性能优化欠缺，问题开始暴露
 - 近万家门店，百亿/月的交易额，可用性很重要
 - 逻辑复杂，领域层缺乏设计原则，面条代码多
 - 业务在跑，新需求不能停

演进路线



■ 数据中心下沉



■ 数据中心下沉

- 拆分粒度考量

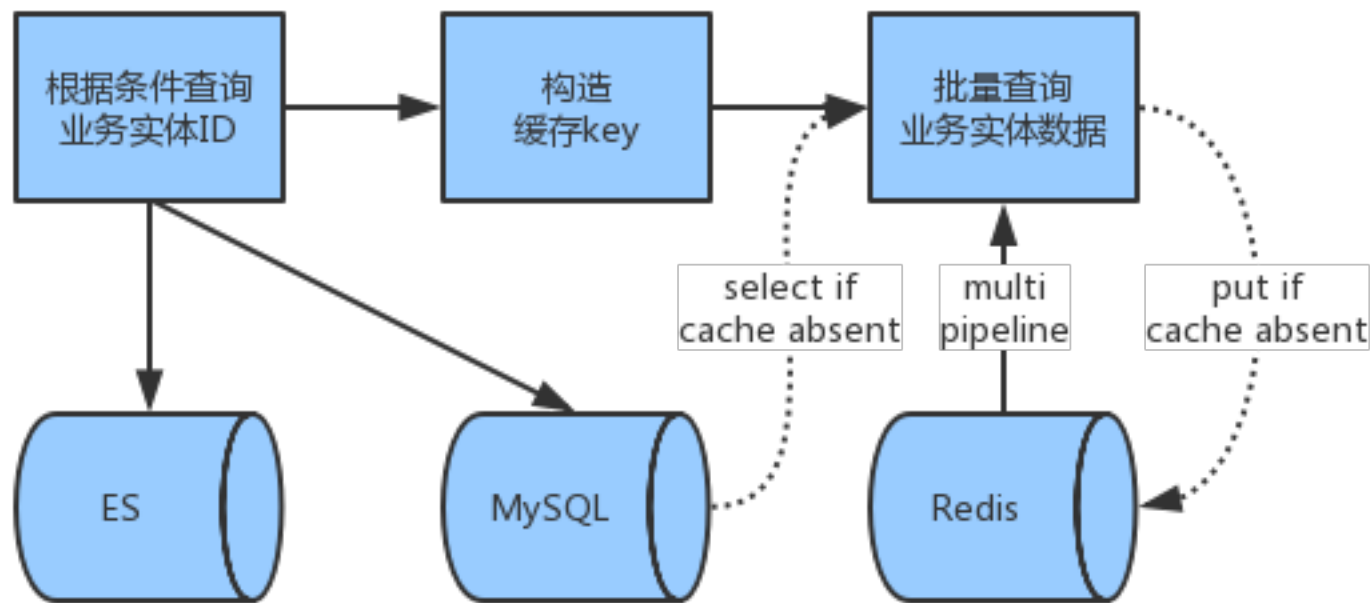
- 领域边界
- 基础设施
- 团队人员

- 框架代码 (Service)

- 事务能力
- 缓存集成
- 领域约束

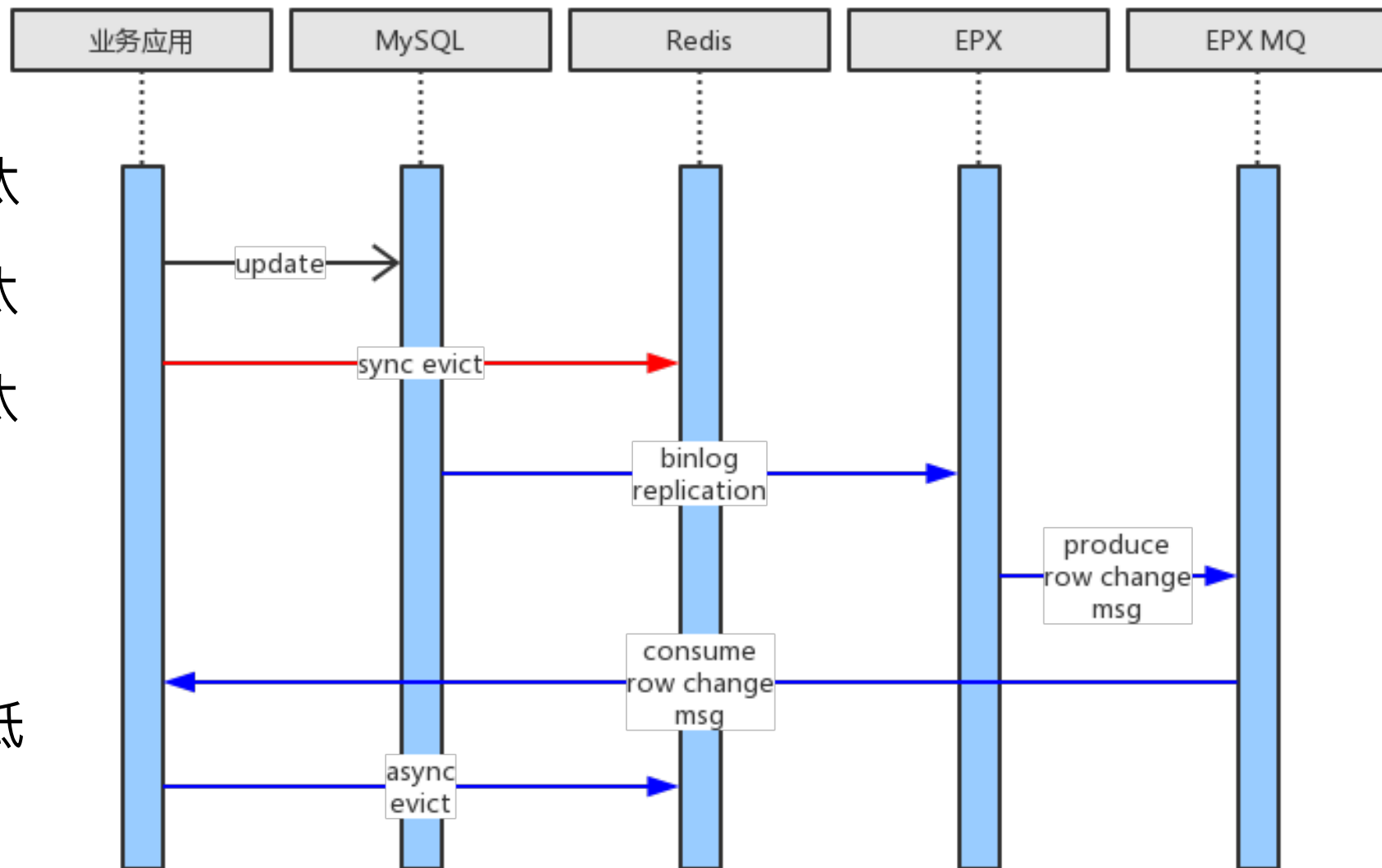
性能攻坚

- 缓存融入基础框架
 - Cache: ID -> Entity
 - Cache-Aside
- 全局收益
 - ES: no store
 - MySQL: 索引覆盖



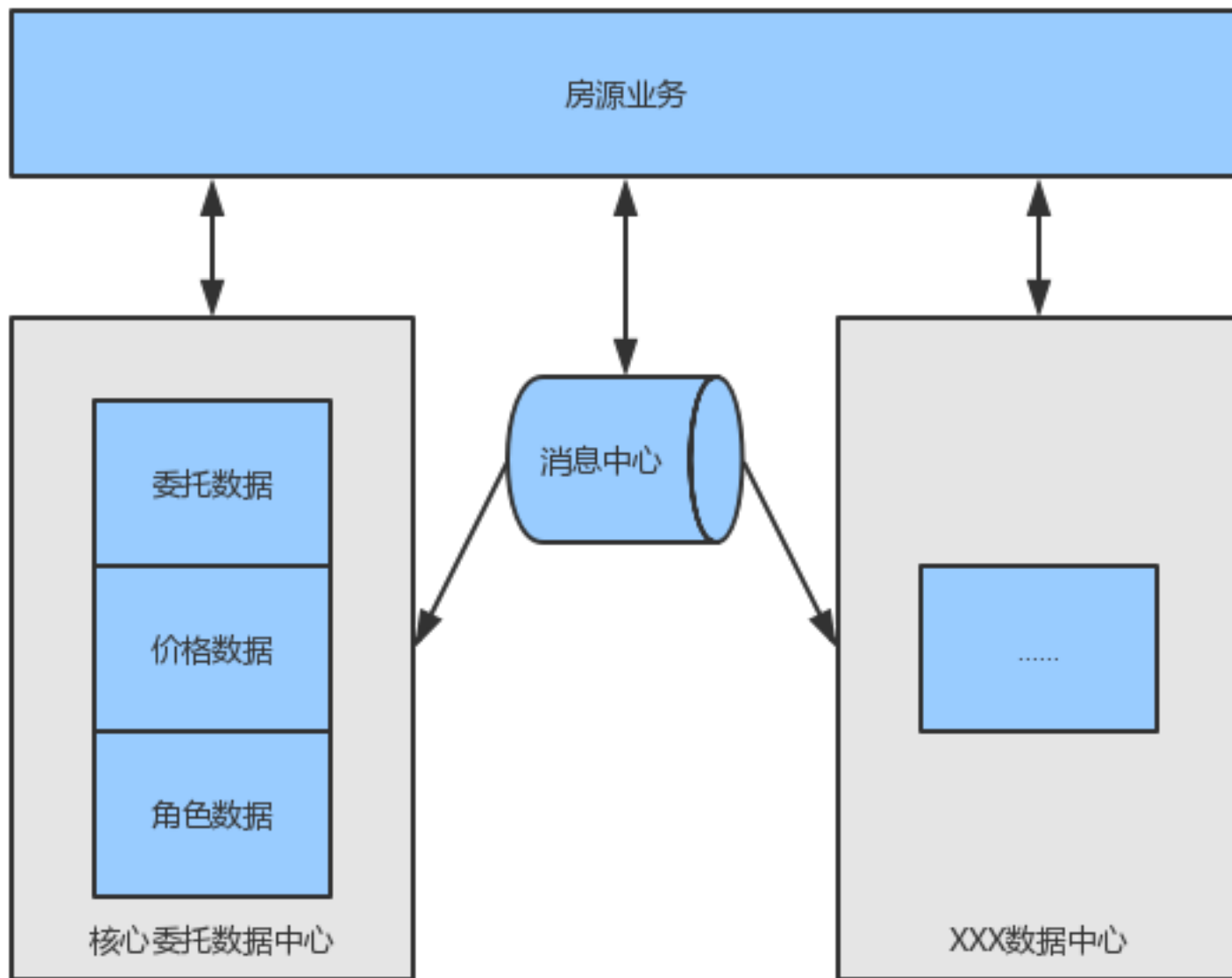
性能攻坚

- 一致性保证/主动淘汰
 - 关键场景同步淘汰
 - 最终一致异步淘汰
 - 降低心智负担
- 效果
 - CPU负载显著降低
 - IO负载下降70%

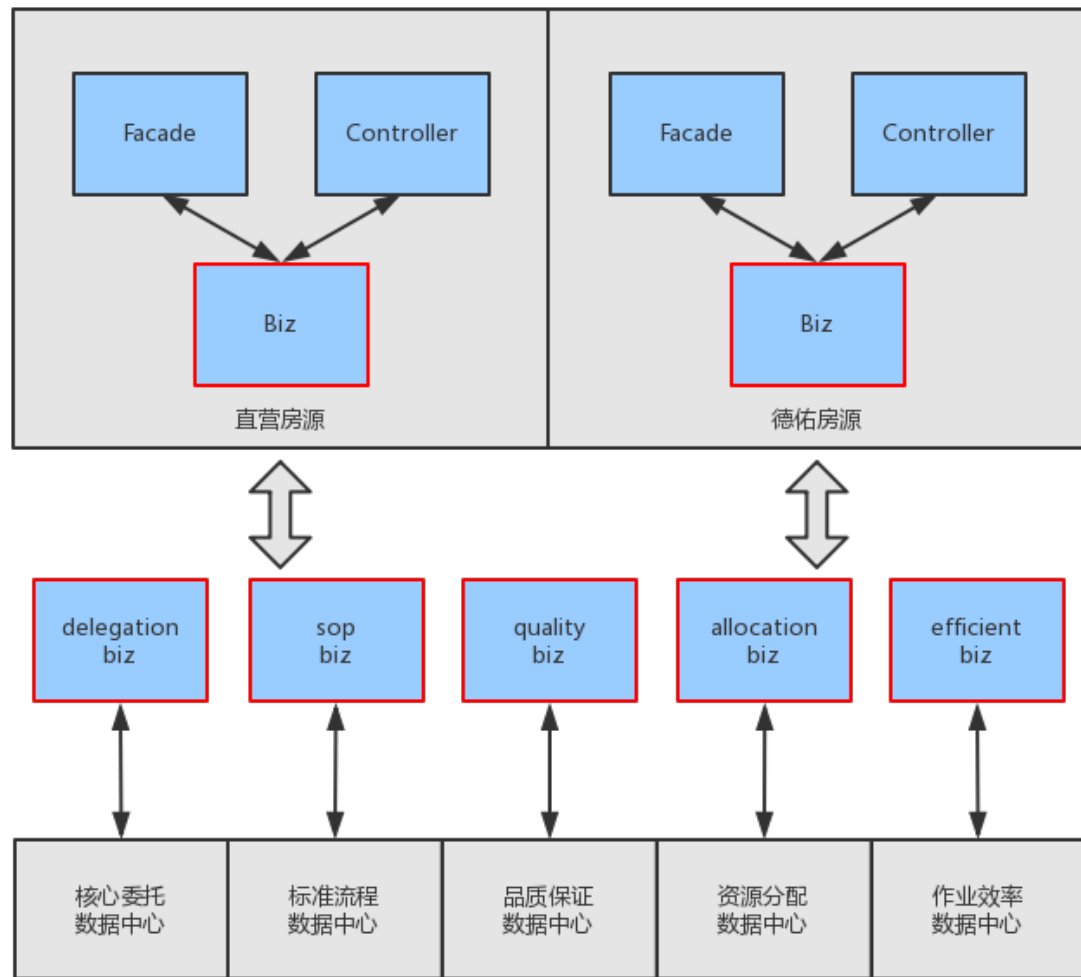
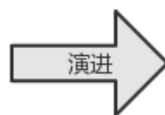
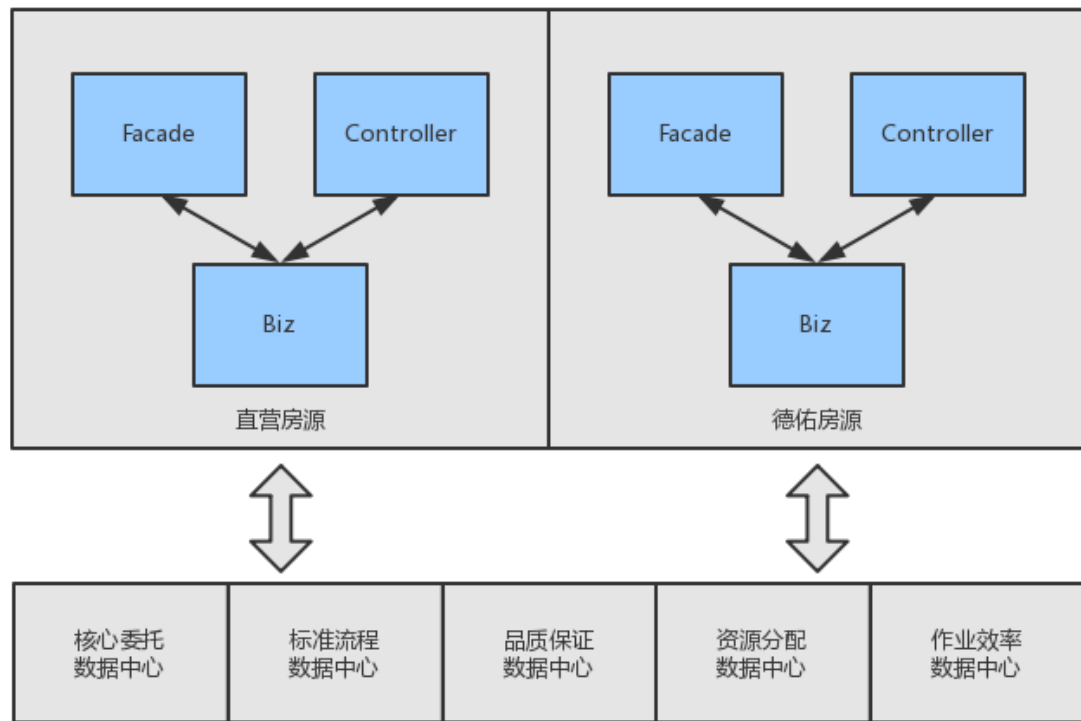


分布式事务

- 本地事务能力
 - 数据中心合理分组
- 跨数据中心
 - 可靠消息通信
- TCC/2PC/3PC
 - 代码侵入性大
 - 开发成本高

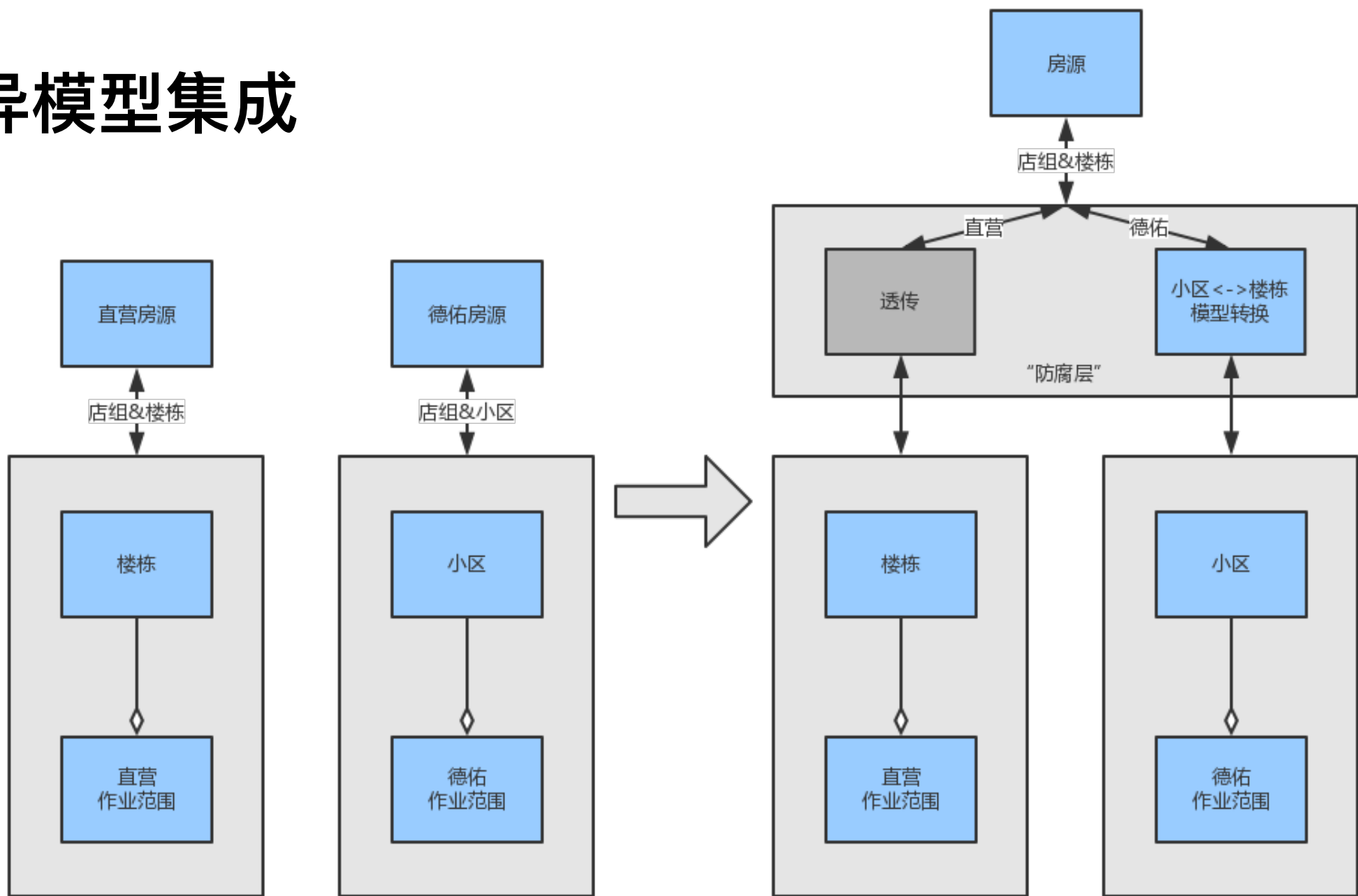


构建新的领域层

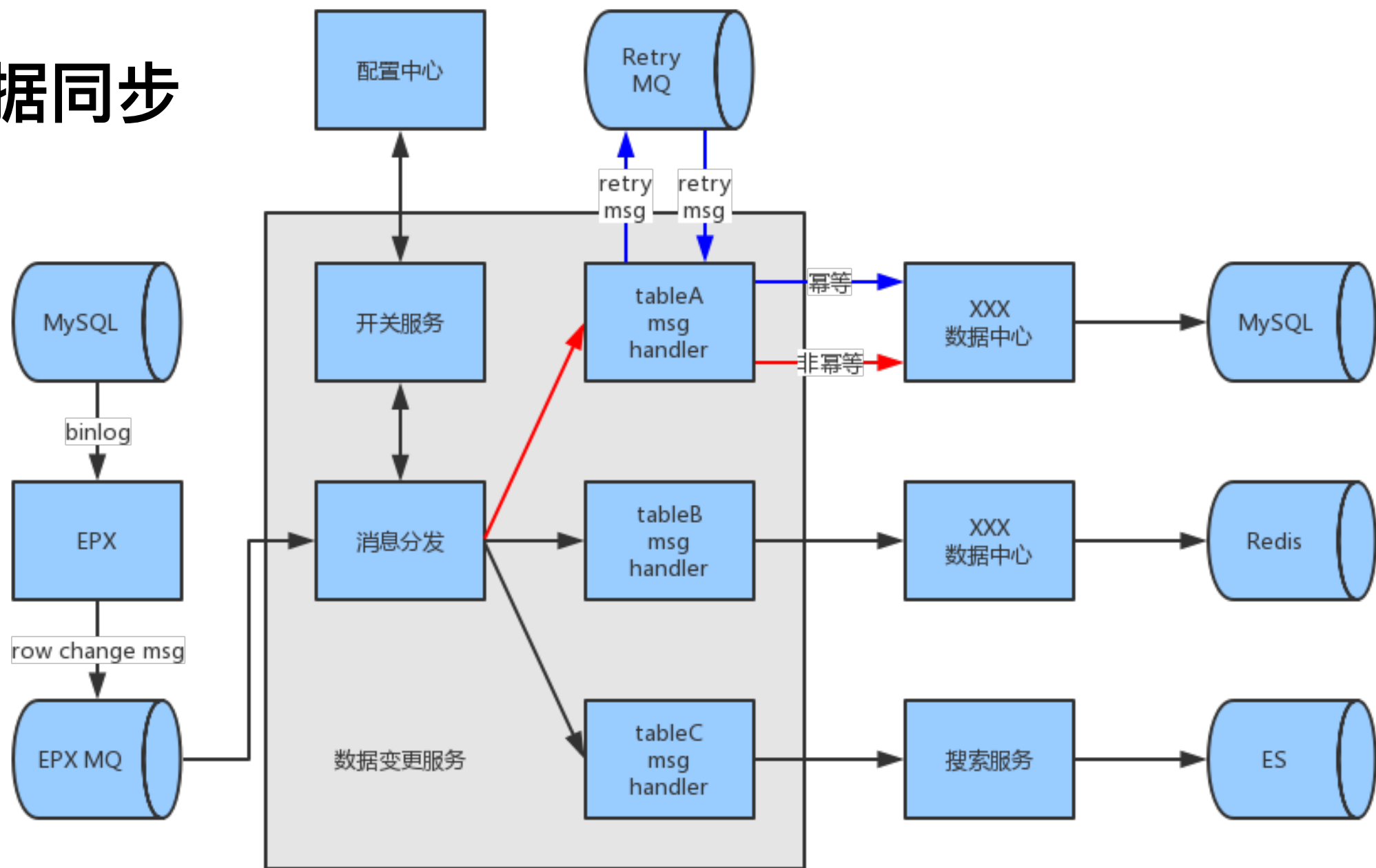


- 新领域层服务**独立部署**
- “肉烂在一口锅里，不要烂到瓢里碗里勺里”

差异模型集成



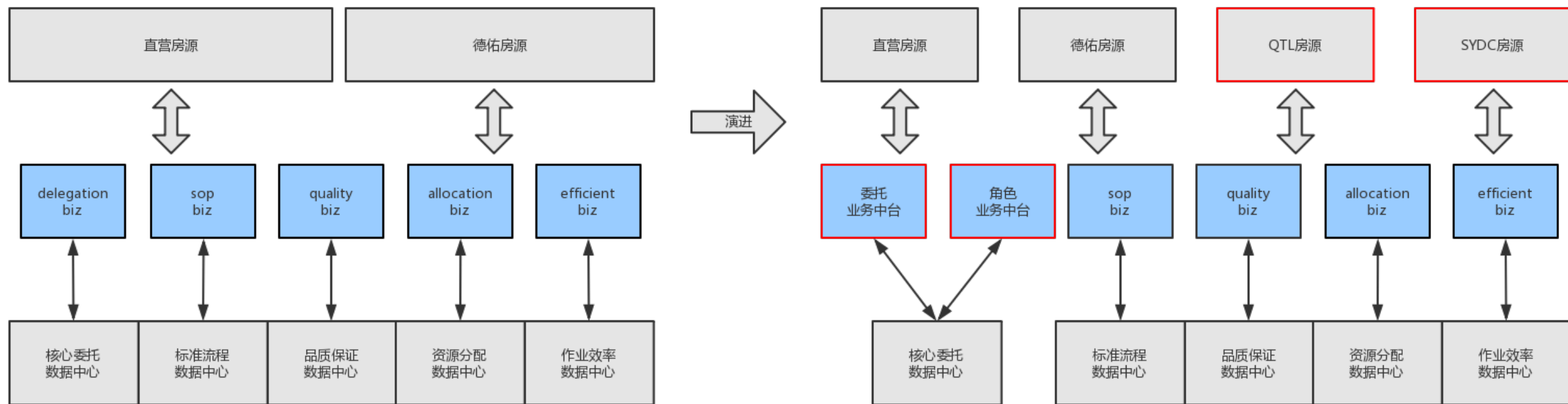
■ 数据同步



■ 数据变更服务

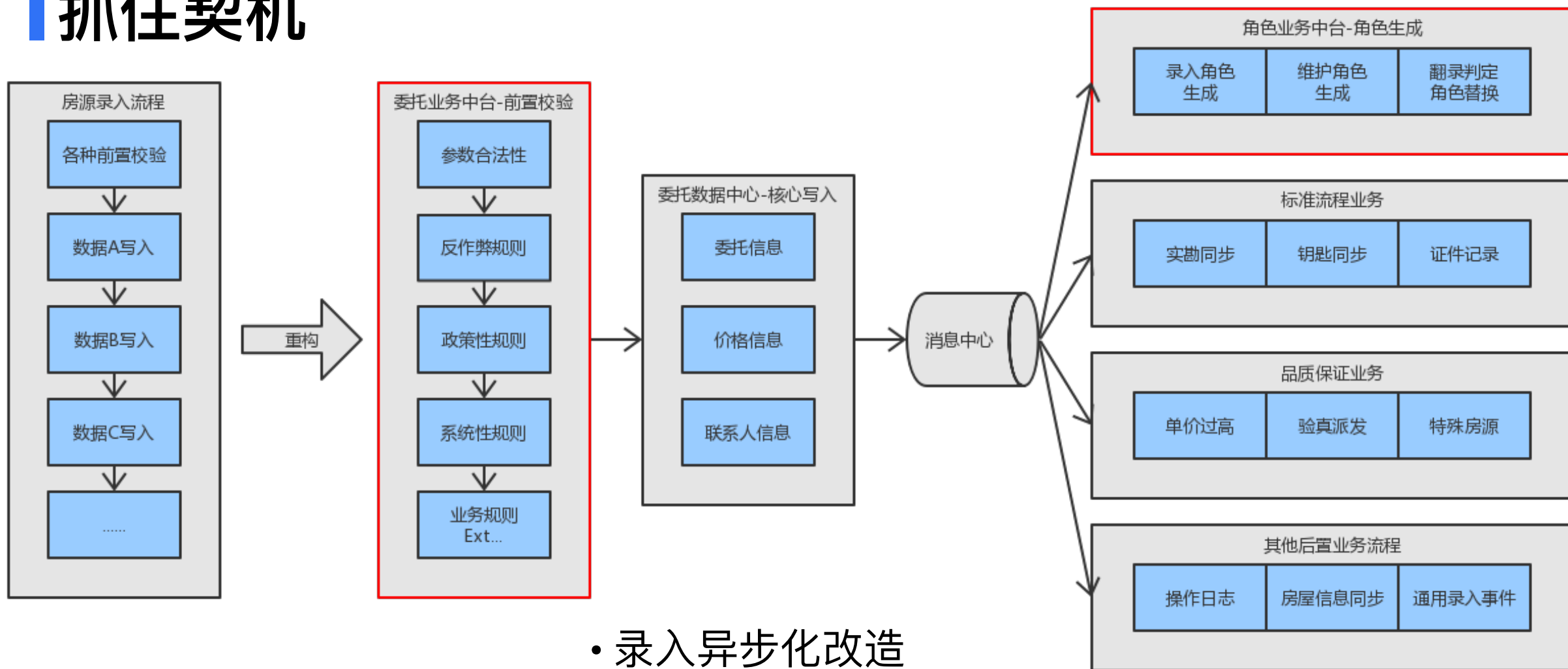
- 建设初衷
 - 保证数据中心之间无耦合
 - 希望领域层相对“纯净”
 - 统一开关能力
 - 统一对接基础设施
- 非幂等调用
 - 常规流程
 - 有序性
- 幂等调用
 - 重试机制
 - 服务可用

领域层裂变



• 拆分粒度：数据中心 > 领域层微服务

抓住契机



- 录入异步化改造
- 孵化出2个中台组件

■ 中台的边界

- 判断原则

- 可复用

- 侵入性

- 人员能力

- 业务规划

- 决策机制

- 产品+技术+业务

- “多数服从少数”

- 协作模式

- 数据子集，非超集

- 功能可编排或默认版本

- 标准输出

■ 经验总结

- 代码防腐治理
 - 微观框架控制
 - 宏观架构约束
 - 基础设施支撑
- 中台建设演进
 - 核心业务抓手
 - 不停演进迭代
 - 合适的就是最好的
 - 明确职责边界



个人微信-窦窦



非你不“壳”