



走进

盒子科技

金融科技 分布式缓存 支付 架构

2018年4月21日13:00 - 18:00

深圳市南山区软件产业基地5栋C座503-盒子科技





美的支付架构演进

李胜军 2018.04.21

- **Content**

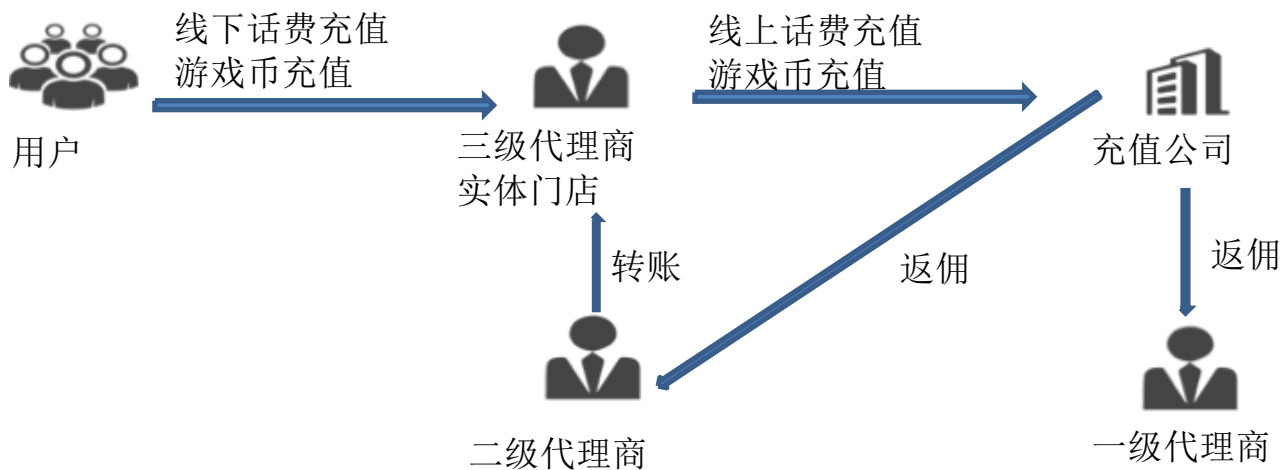
Part1 美的支付前身

Part2 美的支付架构

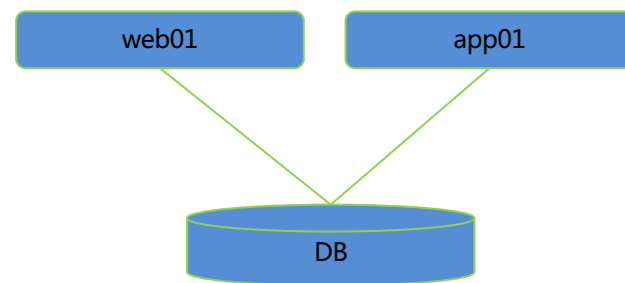
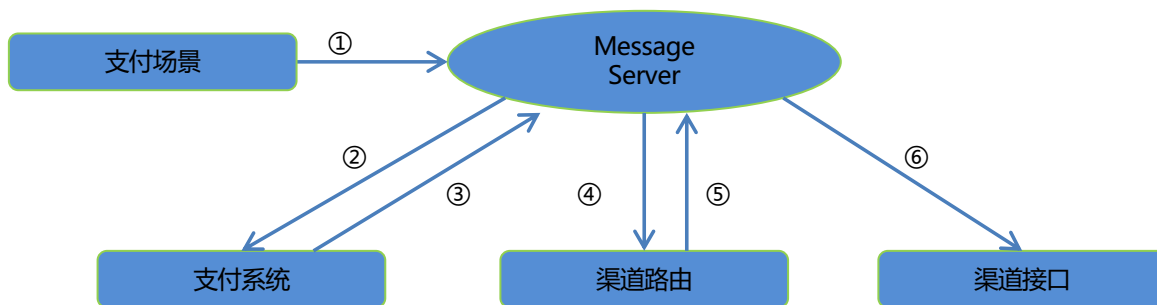
Part3 架构优化

• 美的支付前身-业务

支付主要承接的商户业务：



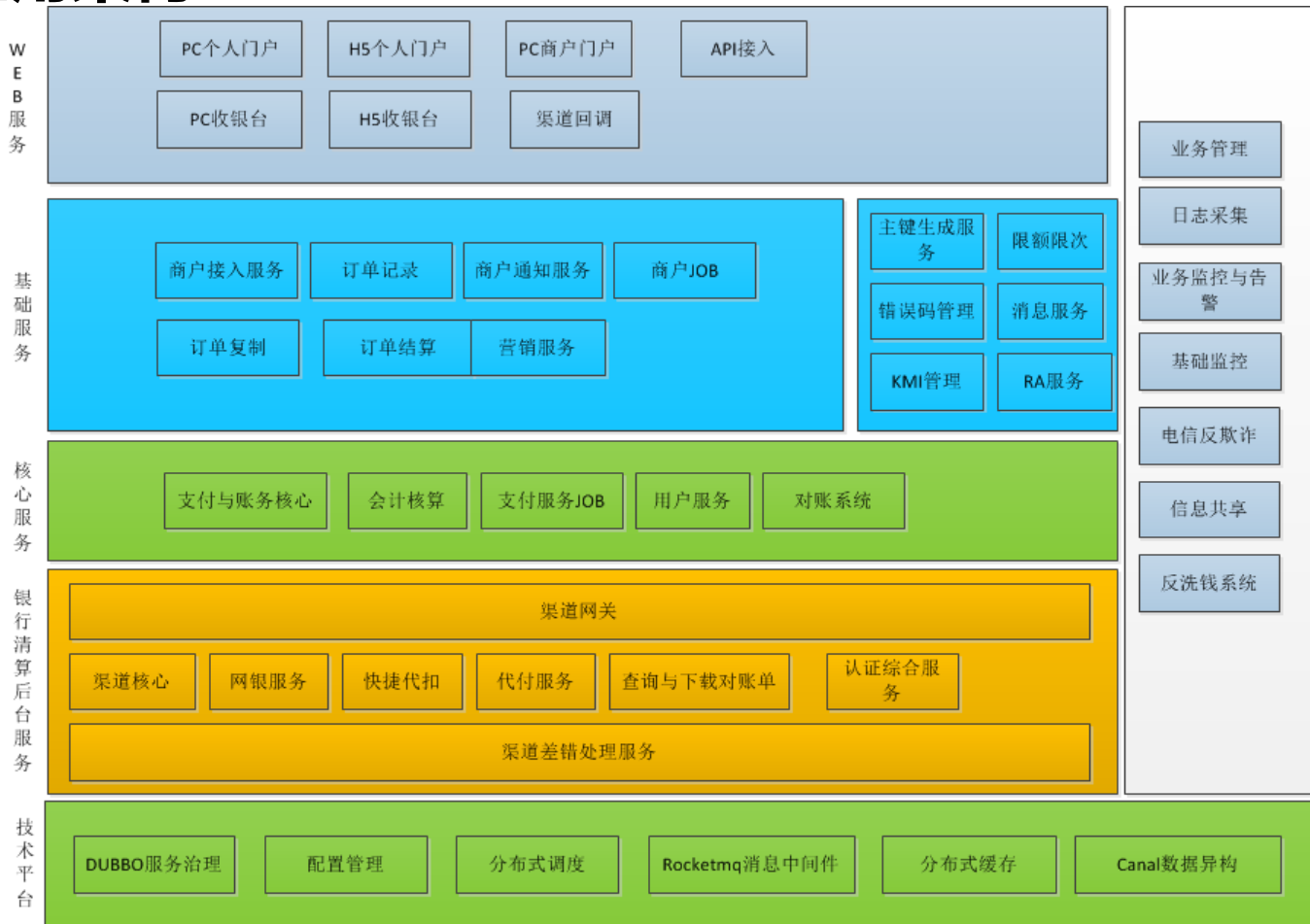
• 美的支付前身-系统



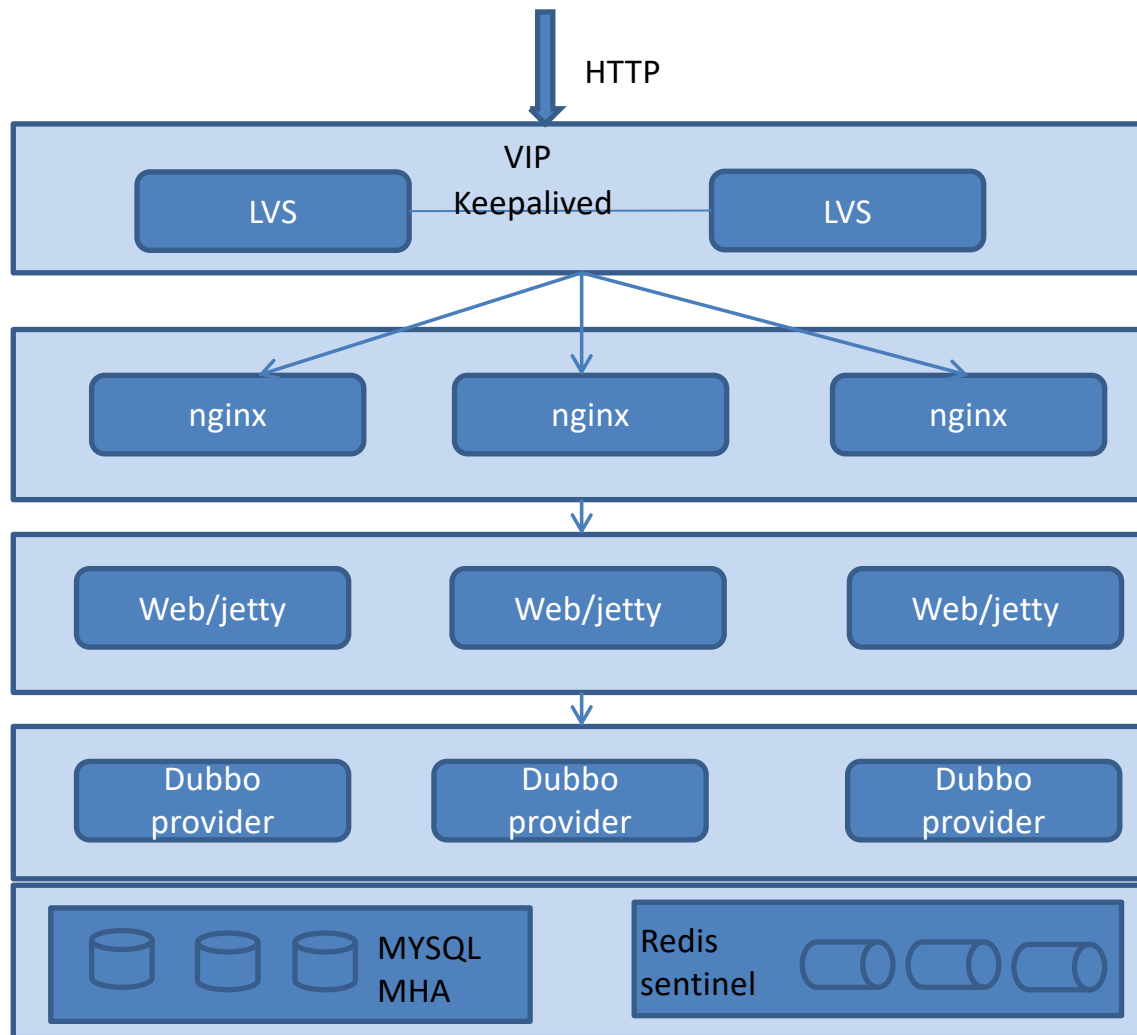
问题：

- 1、MessageServer单点：单点故障、负载无法均衡
- 2、缺少系统层次，模块职责不清晰
- 3、非独立系统，缺失标准商户接入接口
- 4、系统账务不清晰，资金流与会计科目不挂钩。
- 5、不便于系统业务及架构的扩展
- 6、数据库瓶颈

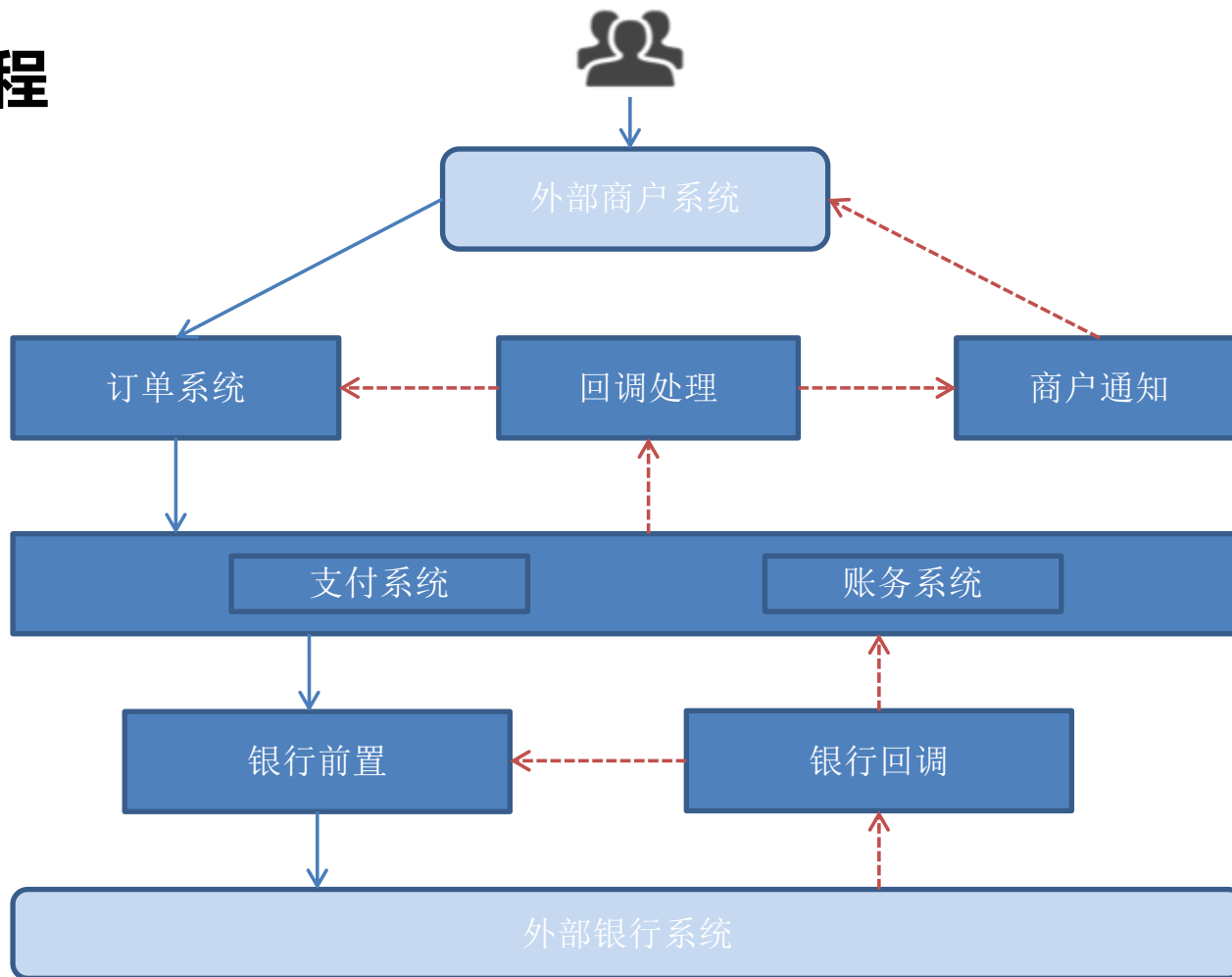
系统应用架构



容灾设计



• 核心流程



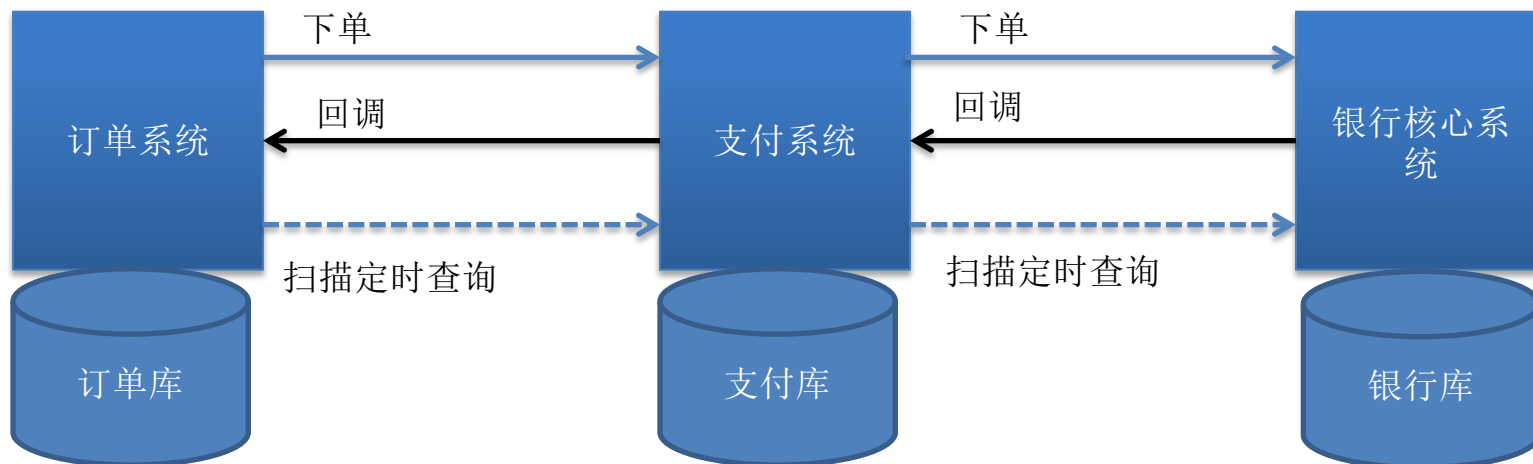
数据一致性

对外：最大努力通知、主动查询接口、手动通知

对内：主动推送、定时查询策略、人工介入

对账：系统内部对账，外部对账

进程内：柔性事务，拆分事务粒度，结合系统自动补偿





• 系统稳定性

重试机制:

幂等性：唯一业务单号，单号谁调用谁生成，抛出幂等异常、或者继续往下执行

代扣、代付业务：系统主动重试（只在系统异常、或者定时未返回状态时重试，一旦业务明确失败，则不再重试）

快捷支付：用户重试（同一笔单，用户可以通过不同银行卡支付多次）

dubbo接口重试

超时机制

多层调用情况下，超时时间设置，上层系统要大于下层系统

快慢接口分离，以免导致超时时间无法尽量设置准确

超时时间不能设置过大，避免资源耗尽

分库分表规则

根据业务特点按用户维度或者时间维度划分。

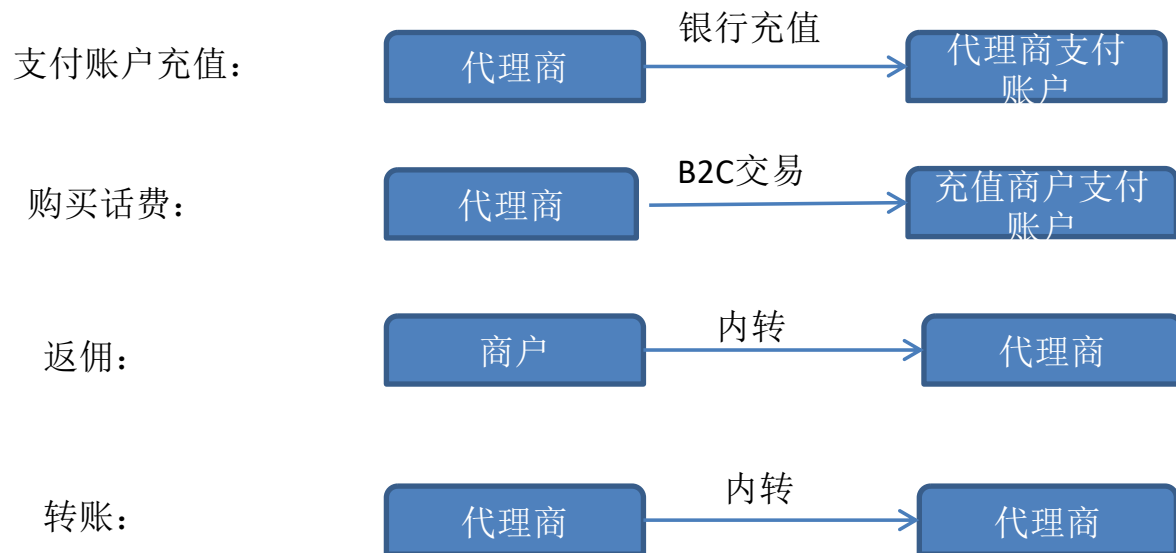
人工处理

一旦系统无法处理，后台留有入口

读写分离

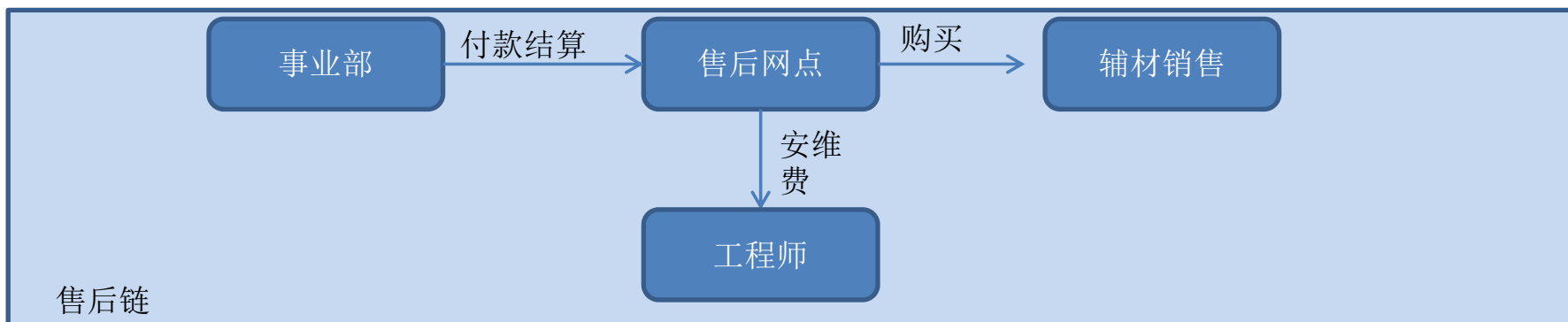
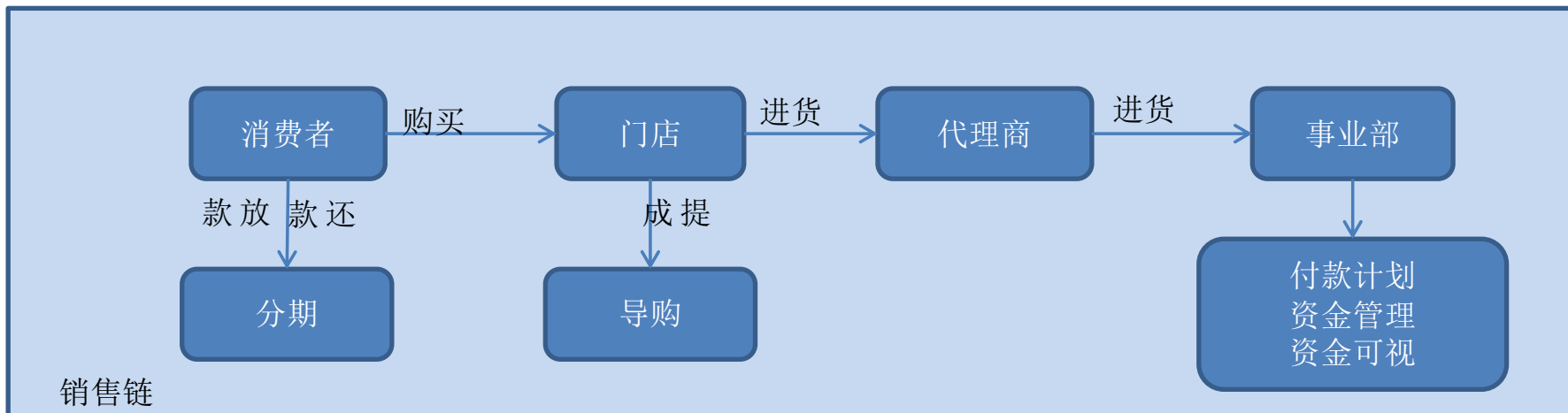
主从库的读写选择，交易链路读主库

• 架构演进一：原有业务特点

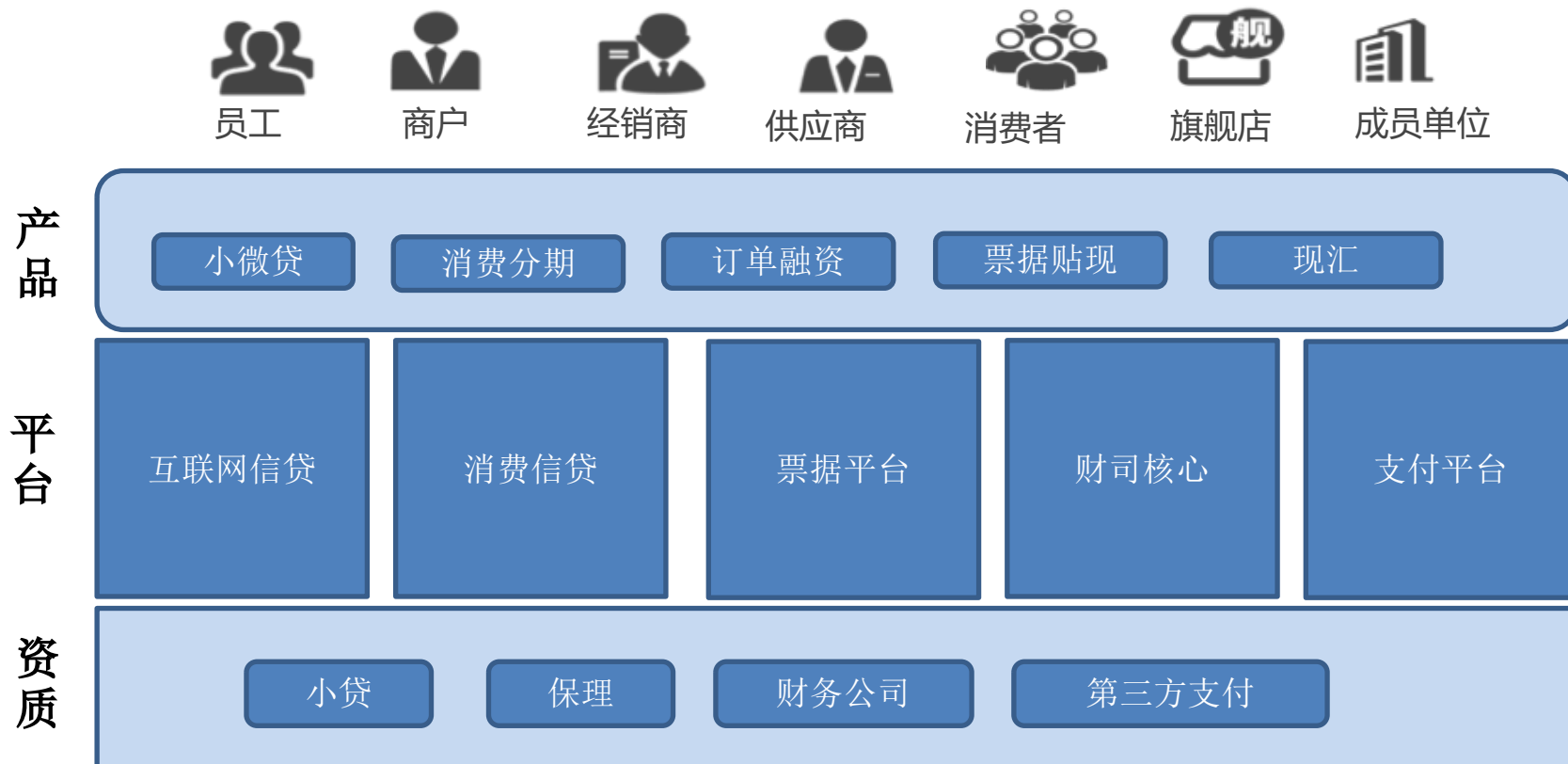


业务特点：业务单一，银行充值频率低，内转交易单日高峰百万笔

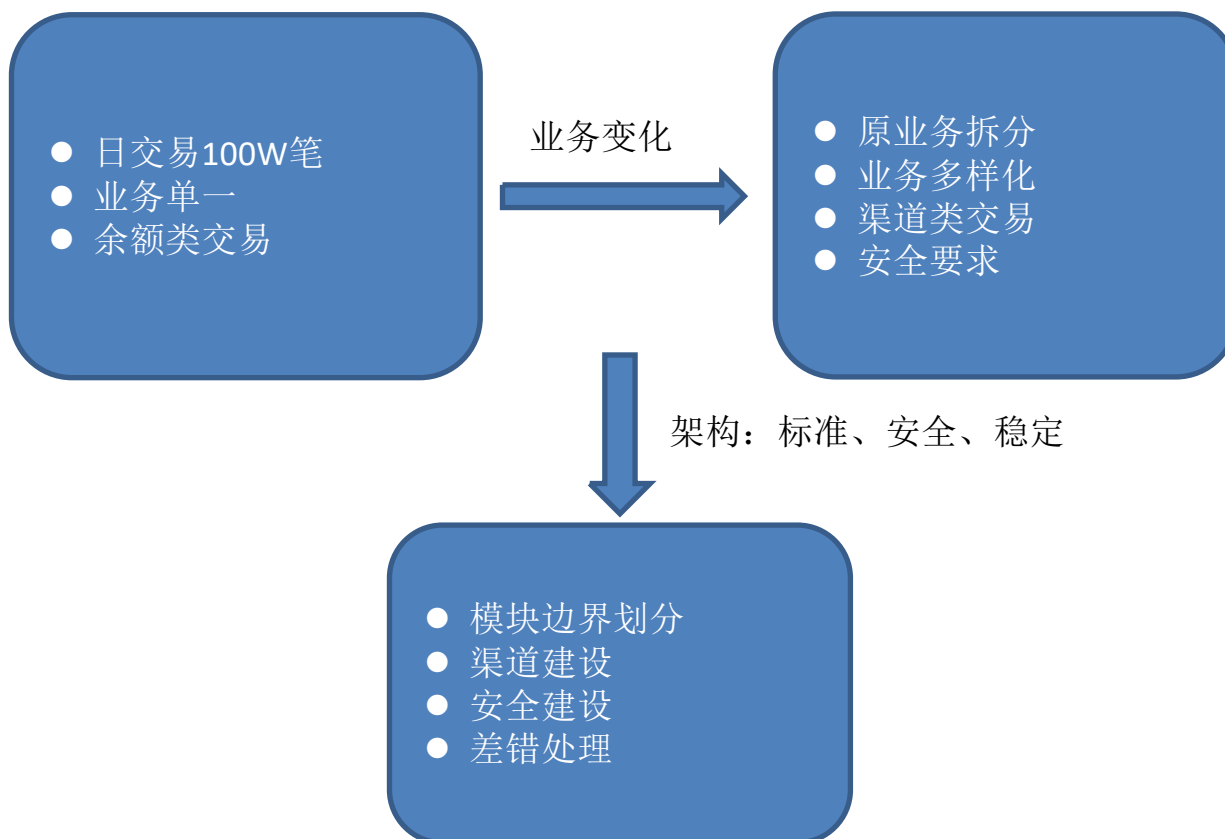
架构演进一：美的业务场景



架构演进一：美的金融概况



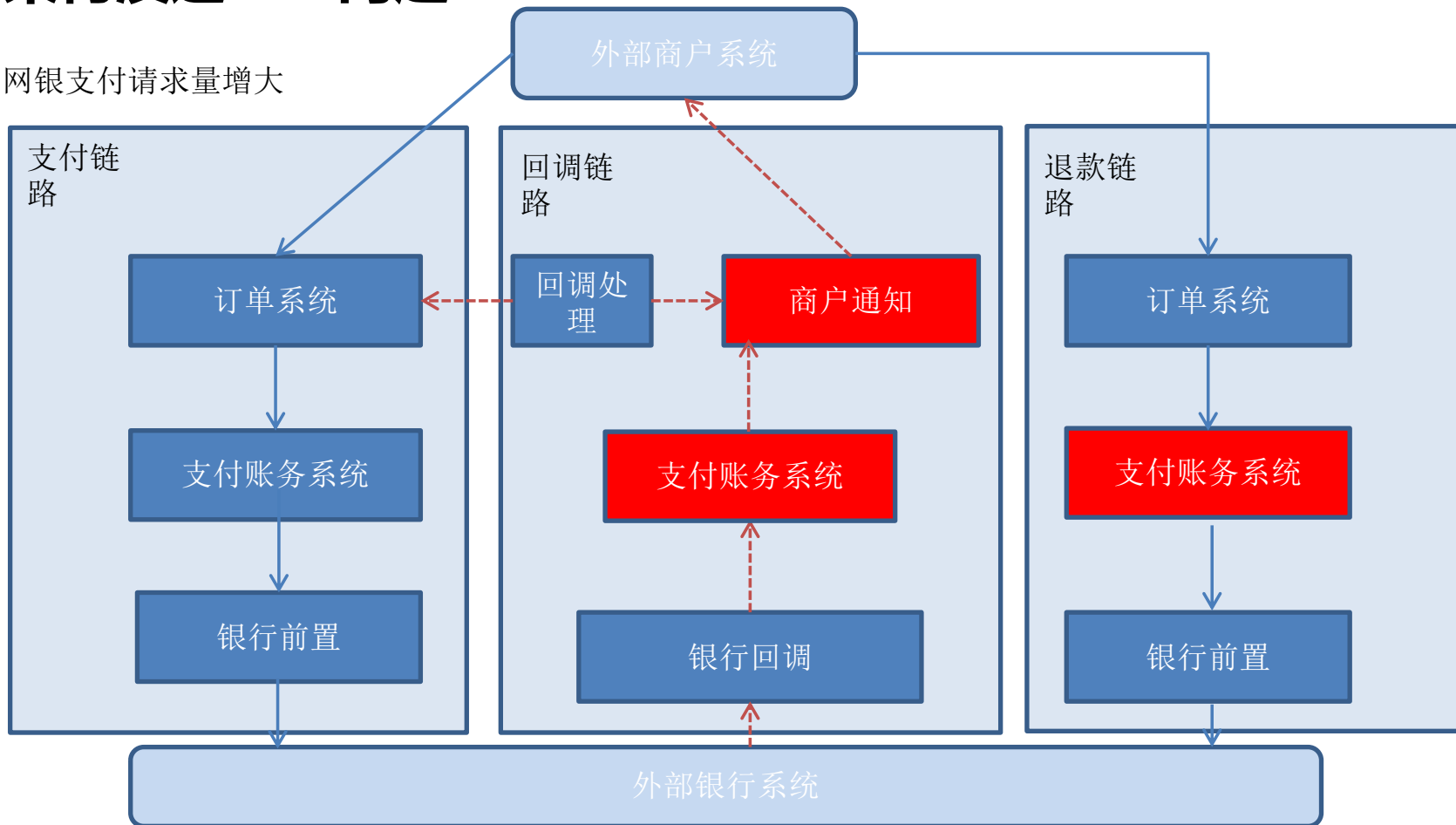
• 架构演进一



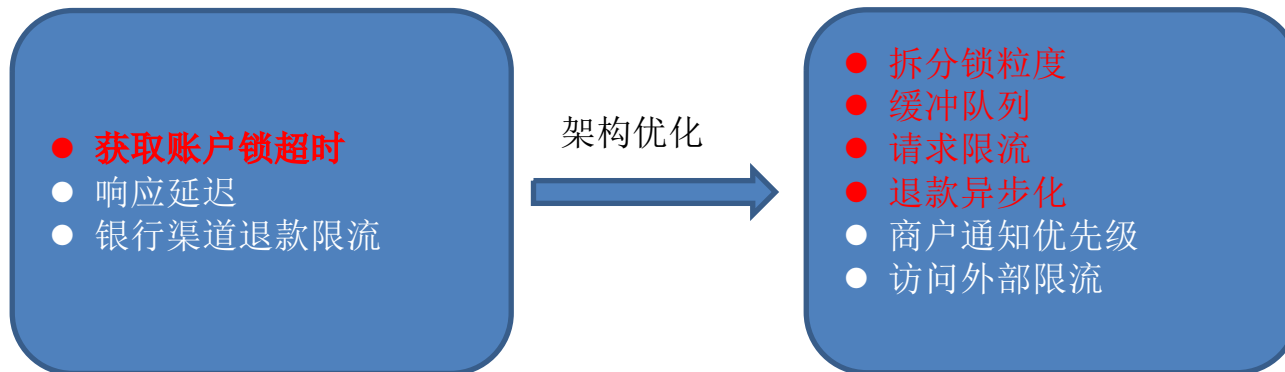


架构演进二：问题

网银支付请求量增大



• 架构演进二：热点账户



回调：事务一： `select ** where act_no='中转账户' for update;` (账户+)

事务二：

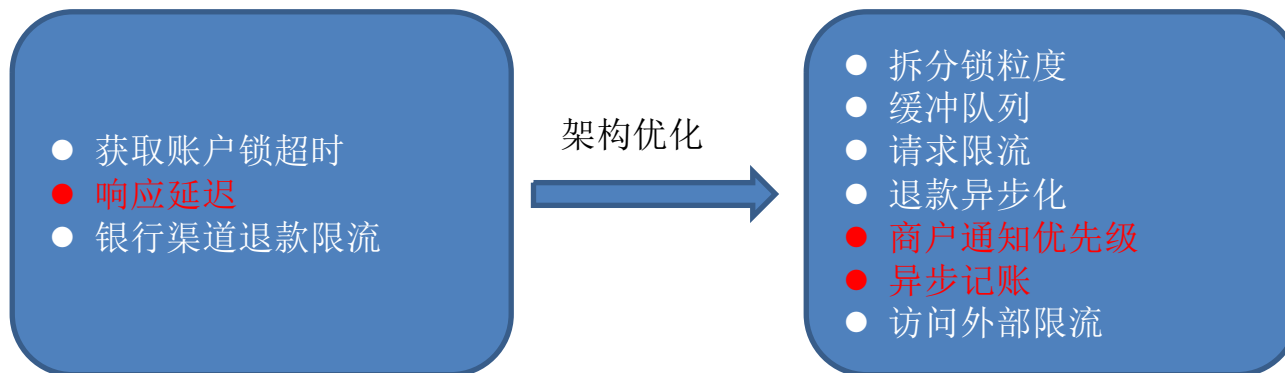
`select ** where act_no='中转账户' for update;` (账户-)

`select ** where act_no='商户账户' for update;` (账户+)

退款： `select ** where act_no='商户账户' for update;`

`select ** where act_no='中转账户' for update;`

• 架构演进二：响应延迟

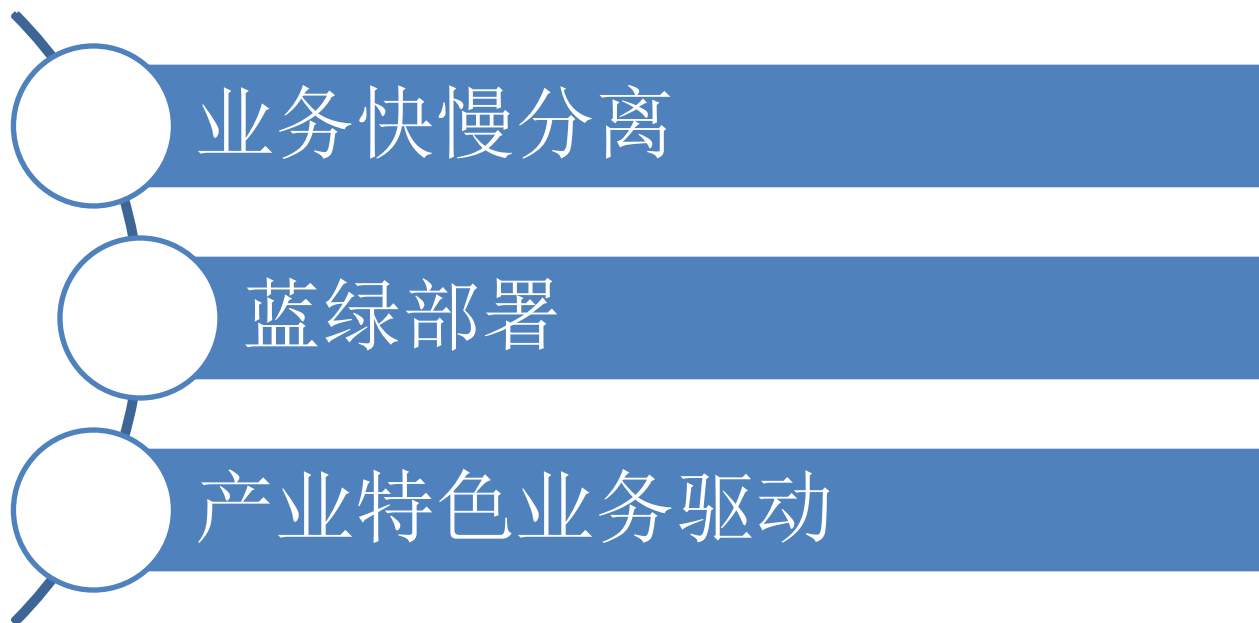


改进前通知：job扫描数据库待通知单，逐步拉大通知时间间隔通知

改进后通知：消息中间件及时发送通知，优先通知最近时间通知单

异步记账：先返回业务成功，再入账成功。

- 架构演进三阶段



Q&A

谢谢大家的聆听！