

20241101-宋华堃-售卖业务串讲

目录

- 1.业务梳理
 - 1.1业务场景介绍
 - 1.2业务名词梳理
 - 1.3业务角色
 - 1.4 业务能力
 - 1.4.1业务能力概况
 - 1.4.2 下单场景
 - 1.4.3 支付场景
 - 1.4.4详细业务流程
- 2.系统梳理
 - 2.1应用架构
 - 2.2对外接口
 - 2.3数据模型
 - 2.4 状态机梳理
- 3.稳定性建设方面
 - 3.1业务告警
 - 3.1.1业务告警大盘
 - 3.1.2限流策略
 - 3.2资金安全建设
- 4.问题总结
 - 4.1业务问题
 - 4.2系统建设问题

1.业务梳理

1.1业务场景介绍

1、售卖中台解决了什么问题？

售卖下单业务场景和方式复杂多样、售卖业务系统稳定性难保证、业务要求更高的新场景接入效率、需要实时高效的系统性能监控能力、高频的业务数据分析诉求。

2、售卖中台的价值？

(1) 业务场景与下单方式的抽象化。通过设计能够涵盖PC、APP、小程序、H5等多种场景的通用的业务场景模型，定义能够支持端内、端外和联合售卖等多种下单方式统一的订单接口，整合爱企

查、商机宝、寻标宝和寻客宝等多个业务线，允许不同场景下的个性化需求在模型基础上进行扩展，提高系统的可扩展性和稳定性。

(2) **建立高效稳定的系统架构。**采用分布式架构，将售卖中台的各个功能模块拆分为独立的服务，并使用负载均衡和容错机制，确保系统在高并发和异常情况下的稳定性和可靠性。使用事务机制保障数据的一致性，提升在订单创建、支付、权益下发等关键环节的数据准确性。

(3) **提升新通道、新场景的接入效率。**标准化接口规范，明确新通道接入时需要遵循的协议和数据格式，降低新通道接入的技术难度和成本。采用持续集成和持续部署（CI/CD）流程，支持新需求的快速上线和迭代。

(4) **系统稳定性监控和业务数据分析。**建立性能监控体系，实时监控系统运行状况，及时发现并处理性能瓶颈。建立业务线间的数据统一管理机制，确保爱企查、商机宝、寻标宝和寻客宝等业务线的订单信息能够统一管理和处理。

通过构建高效、稳定的售卖中台，满足爱企查会员产品在不同业务场景和下单方式下的需求，提升新通道的接入效率，为业务的持续发展提供有力支持。

爱企查会员产品售卖依托于售卖中台来承载，售卖中台主要负责爱企查整套订单履约业务，用来管理从接收客户订单到为客户下发会员权益的全过程。

会员产品属于虚拟产品范畴，订单的完整生命周期涵盖了：从产品管理开始，到优惠营销的制定，用户下单，交易支付，再到权益下发，发票开票，以及财务入账等多个环节。

从场景上来看，订单的处理涉及多种场景。包括传统的pc端，移动端的app和小程序，以及轻量级的h5页面等，满足了不同用户在不同场景下的需求。

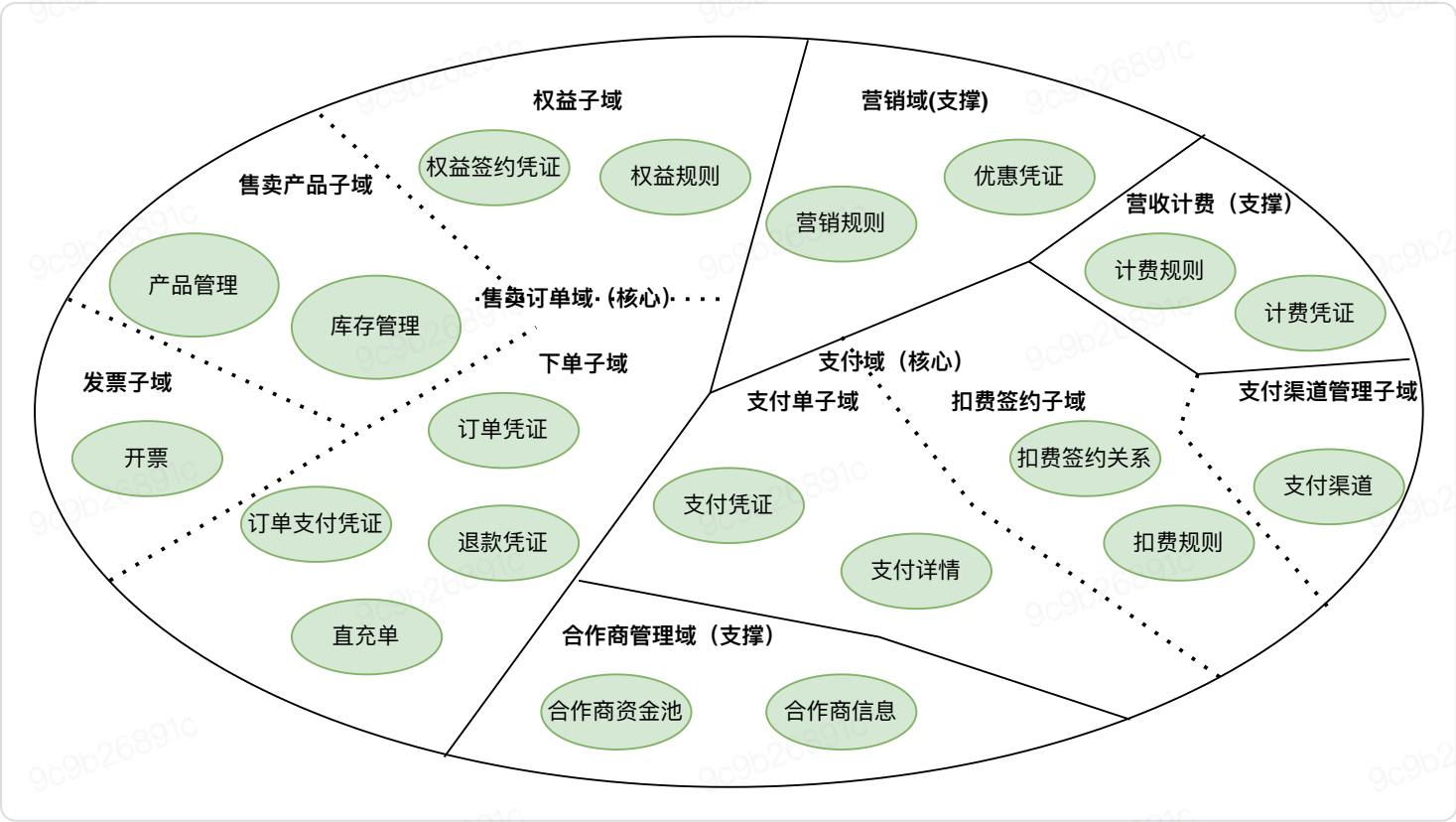
从下单方式上看，主要包括端内下单、端外直销和联合售卖三种。端内下单是指用户在百度平台或应用内直接下单购买。端外下单则是指用户在第三方平台或应用上购买会员产品，需要通过跳转或链接的方式完成购买。此外，联合售卖即与其他产品或服务进行捆绑销售，以满足用户的多样化需求。

从业务线上看，订单的处理涉及多个业务线。包括提供爱企查，商机宝，提供商标注册、寻标宝以及寻客宝等。

在产品类型上看，包括定期续费产品，用户需要按照约定的周期进行续费。以及一次性支付产品，用户只需支付一次费用即可获得相应的产品或服务。

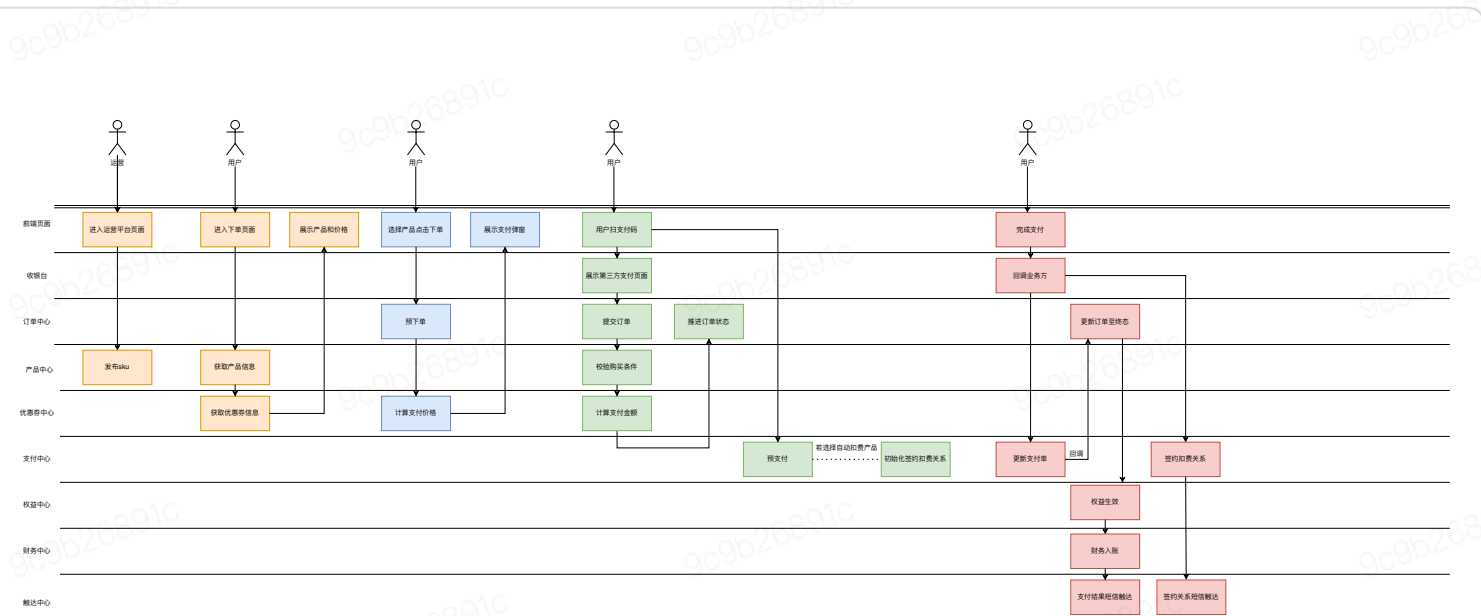
从登录方式上看，支持免登录和已登录两种支付方式。

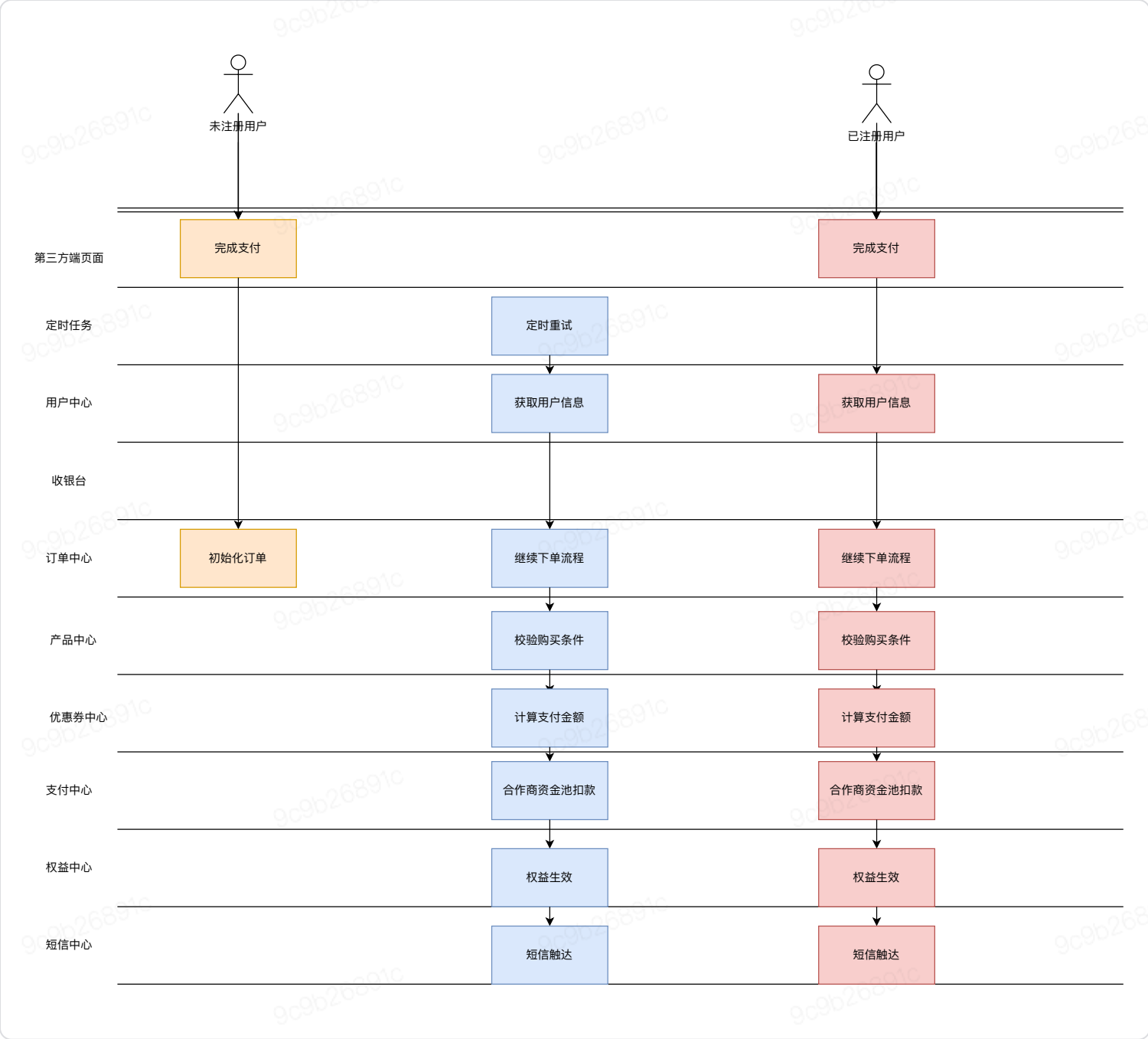
按照领域将订单履约业务进行切分如下：



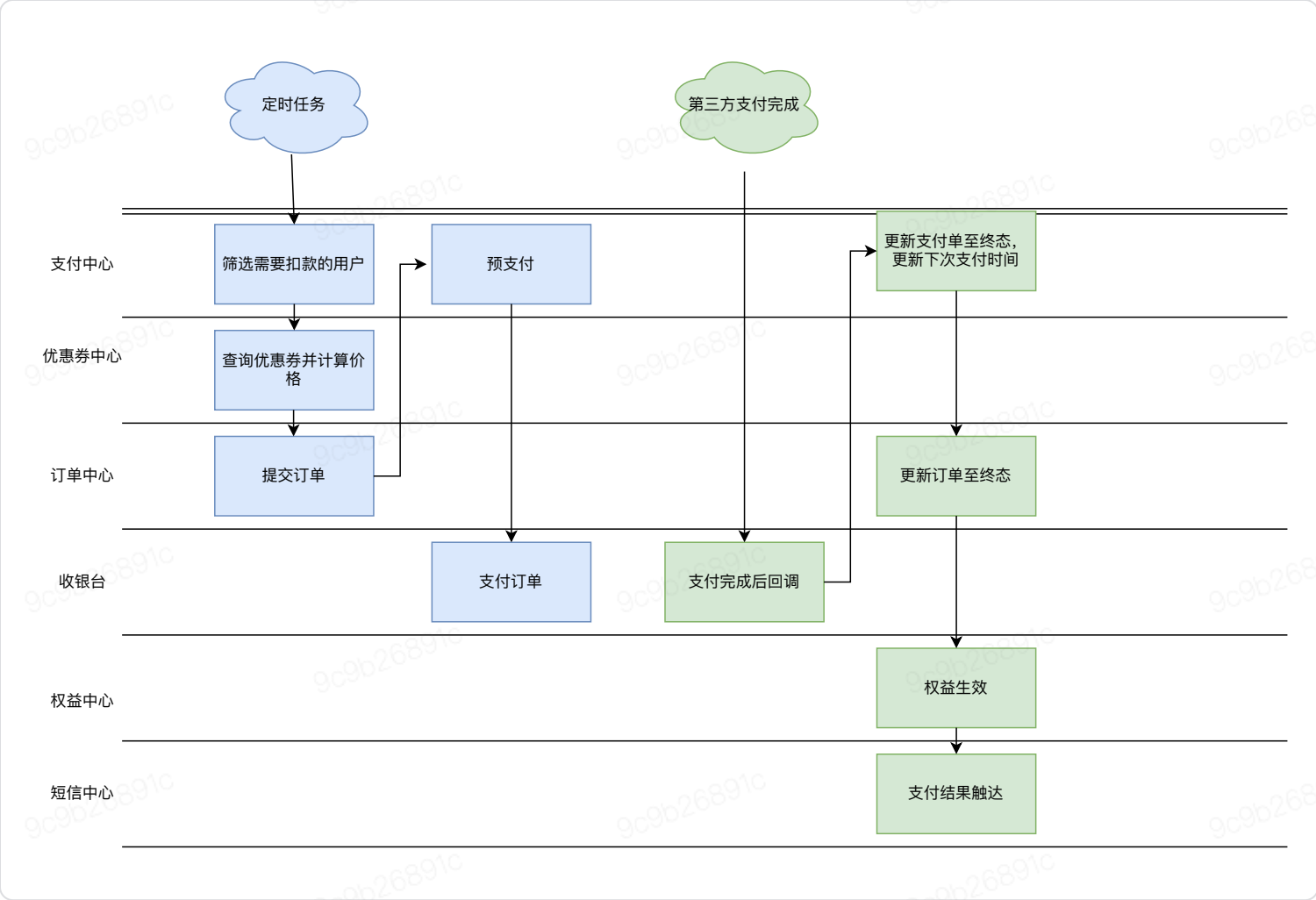
1.2业务名词梳理

1	名词	名词解释	备注
2	售卖商品	指代爱企查、商机宝、寻标宝和寻客宝等产品的会员卡，包括新人周卡、月、季、年卡以及连续包月卡等。	
3	权益	当用户购买了会员商品后，才能够享受对应产品的会员服务，比如解锁企业联系方式、空号检测等。	
4	扣费签约	当用户购买了连续包月卡后，会跟用户签约定时扣款协议，之后每月定时为用户续费对应产品。	
5	合作商	百度与外部企业合作，外部企业提前预支资金，用户可以直接获取爱企查和三宝产品权益，不需要走收银台逻辑，百度提供多种代销方式，例如新客、资金池和数量池，当前仅爱企查和商机宝支持直充下单，直充属于代销方式。	
6	直冲下单	合作商提前预支资金，用户可以在合作商的页面直接下单，百度侧在资金池扣除合作商资金，而无需走收银台支付链路。	
7	联合售卖	与其他产品或服务进行捆绑销售的方式，当前主要是同百度文库合作。	
8	抖音下单	对外的电商合作方式，有别与直充下单，是一种直营的方式，目前仅支持爱企查。	





(3) 签约代扣



1.4.2 下单场景



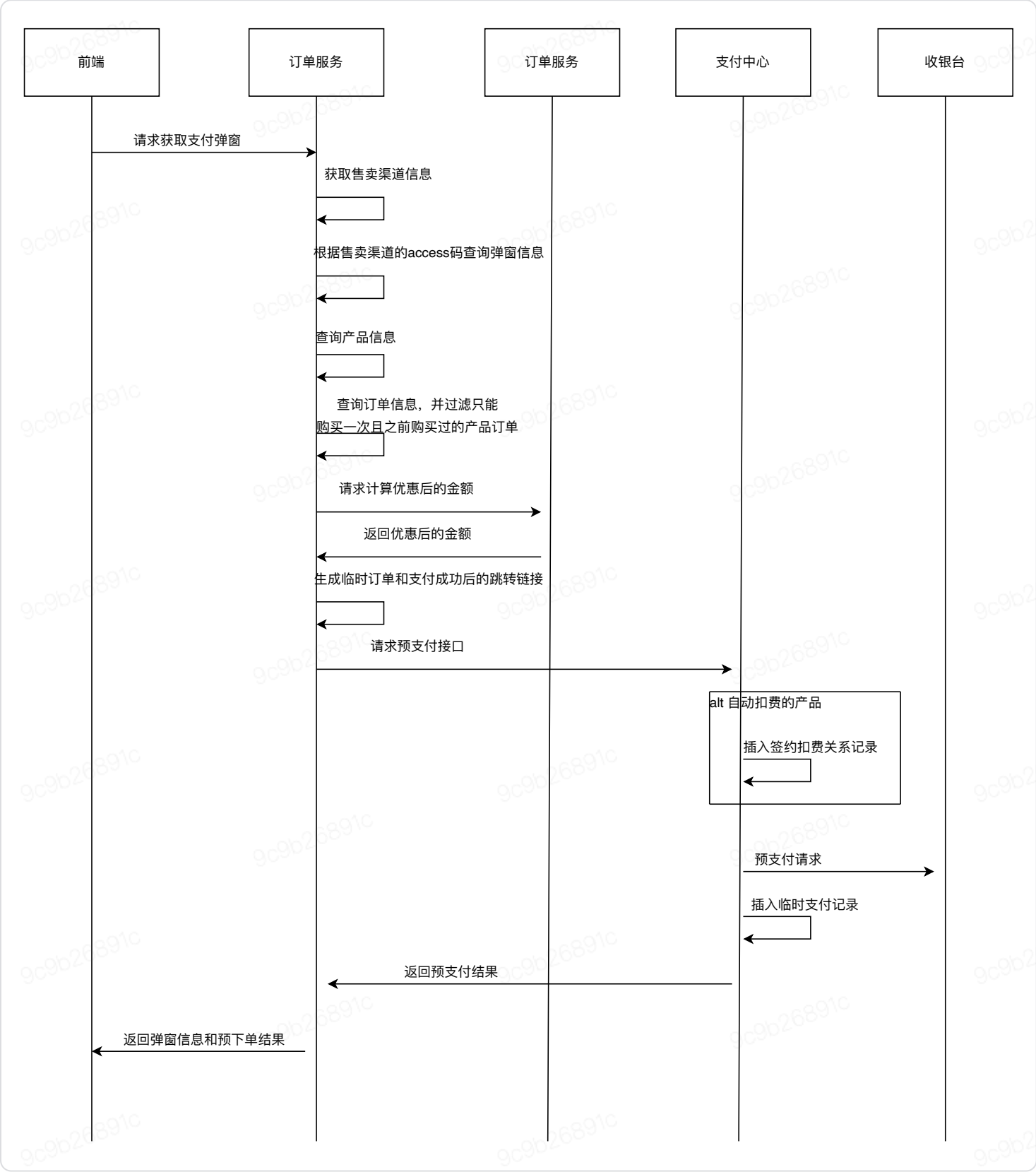
1.4.3 支付场景



1.4.4详细业务流程

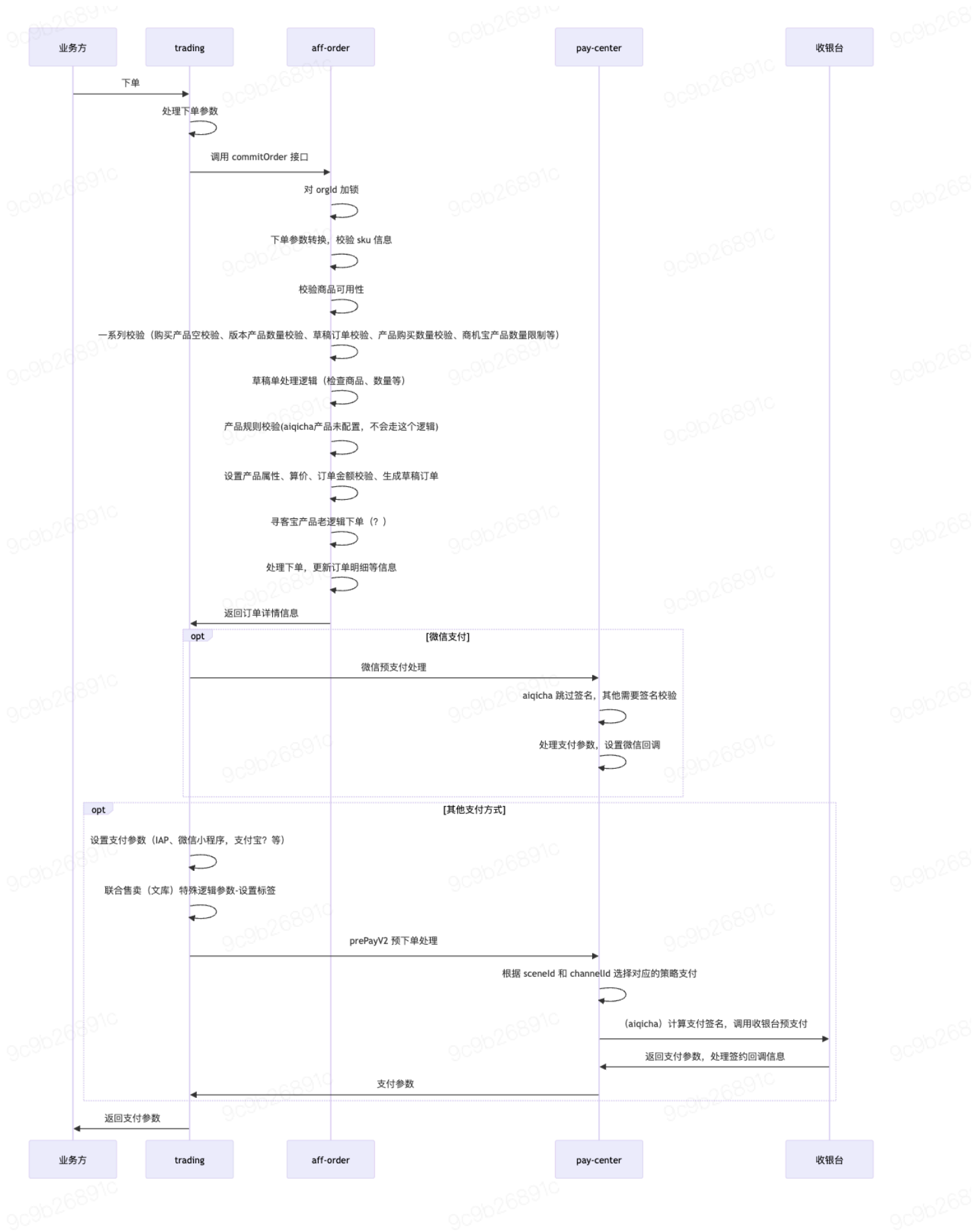
- 预下单（获取外部支付信息）

预下单的接口特别多，几乎每个场景都定义了一个接口，维护成本较高，但从职责上来看是比较单一的，建议对接口进行统一。



- 提交订单

针对各个业务线的定制化逻辑，使用类似责任链模式的设计，对不同业务线的业务流程进行编排，比较符合开闭原则，提高了系统的可维护性和可复用性。

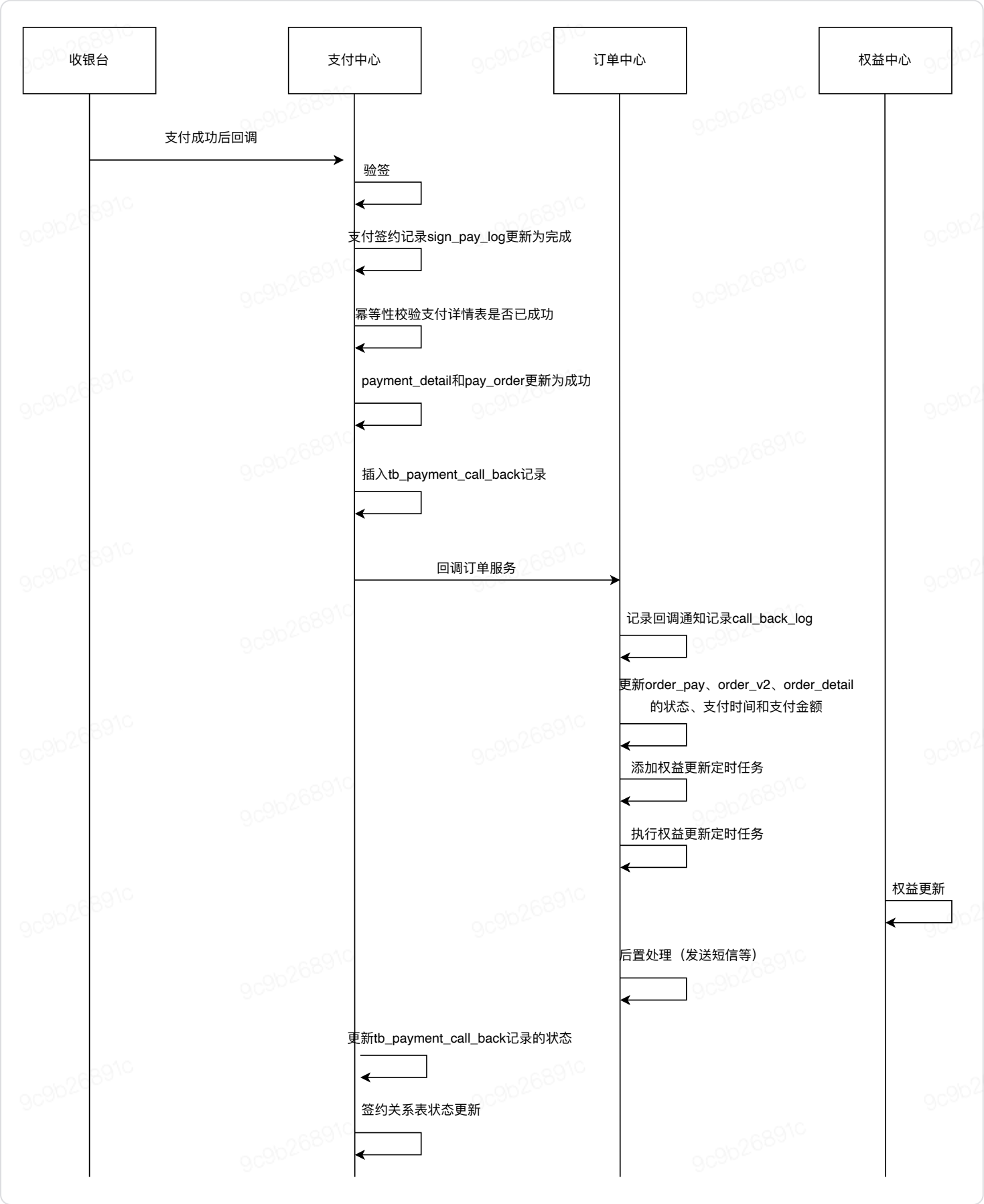


• 支付回调

使用策略+工厂的方式在一定程度上降低了系统复杂度，但把每个场景+通道当做一个策略，且都实现了一个子类，代码可复用性比较低，策略太多也比较难维护，建议使用模板模式对逻辑进行抽象，提升复用性和可

读性

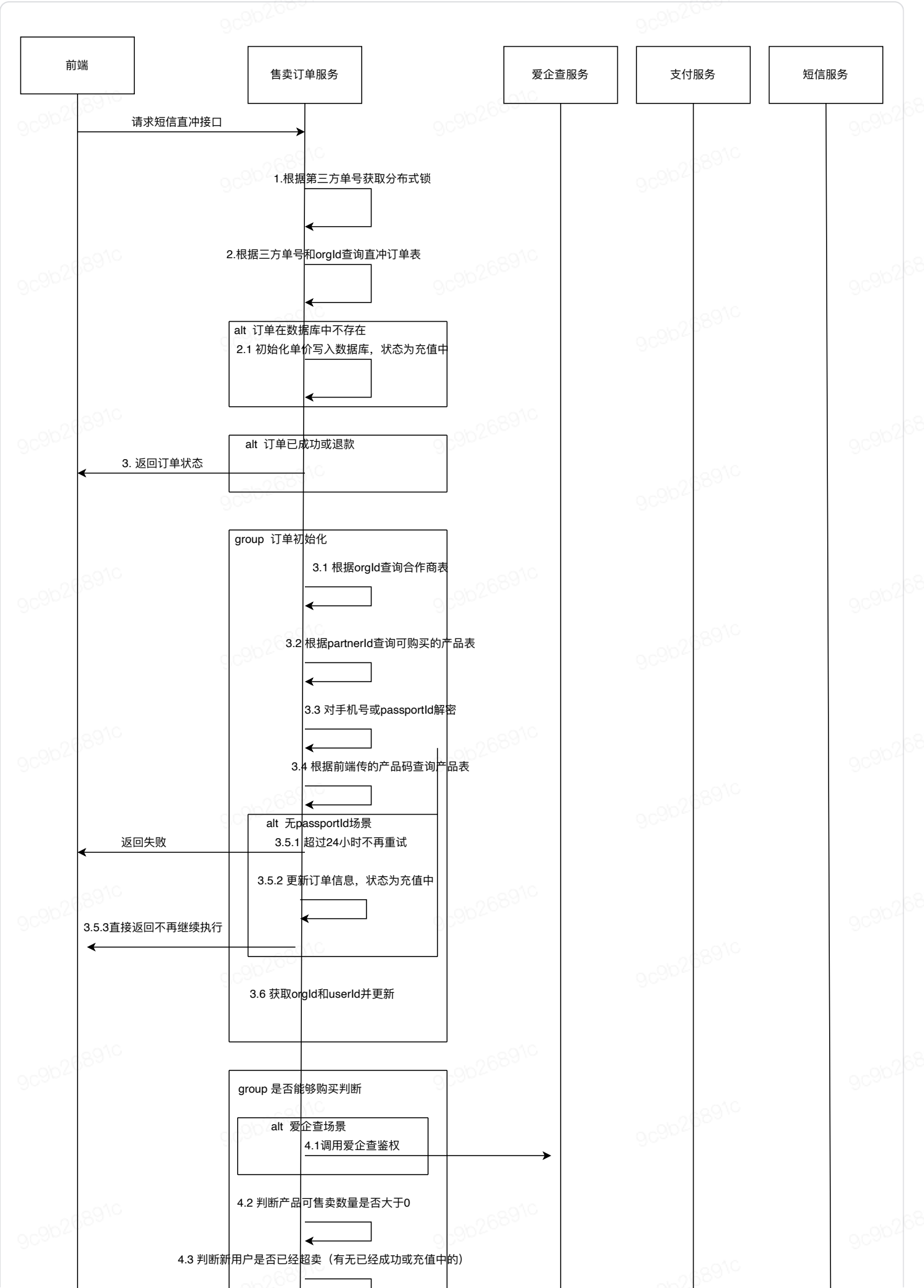
支付回调有做幂等控制，但缺少分布式锁之类的并发控制手段，比较容易出现重复调用的问题，比如最近在做的重复支付问题排查。



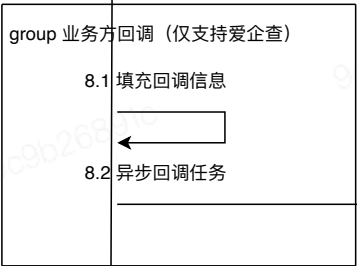
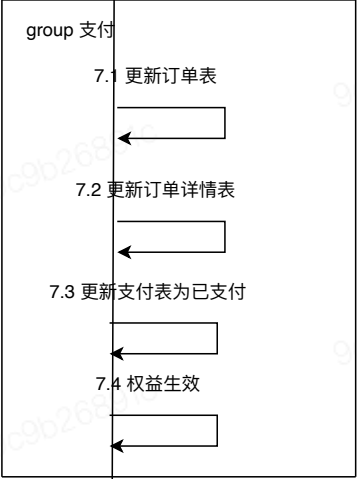
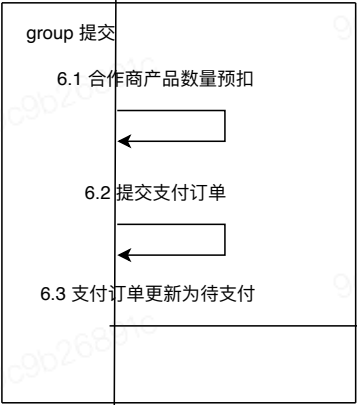
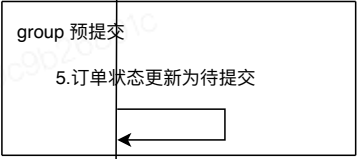
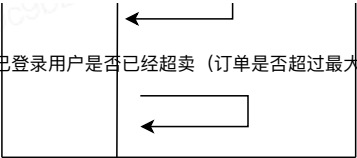
- 直充下单

直充订单有个比较容易让人混乱的业务点，未注册百度账号流程只能走到充值中，后边有个定时任务不断地重试。

未注册、重试任务、已注册充值，三个场景混杂在一块导致直充接口逻辑特别复杂，这种情况就建议拆分接口或者在controller做分流了，接口的职责有点太多了。

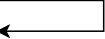


4.4 判断已登录用户是否已经超卖（订单是否超过最大数量）



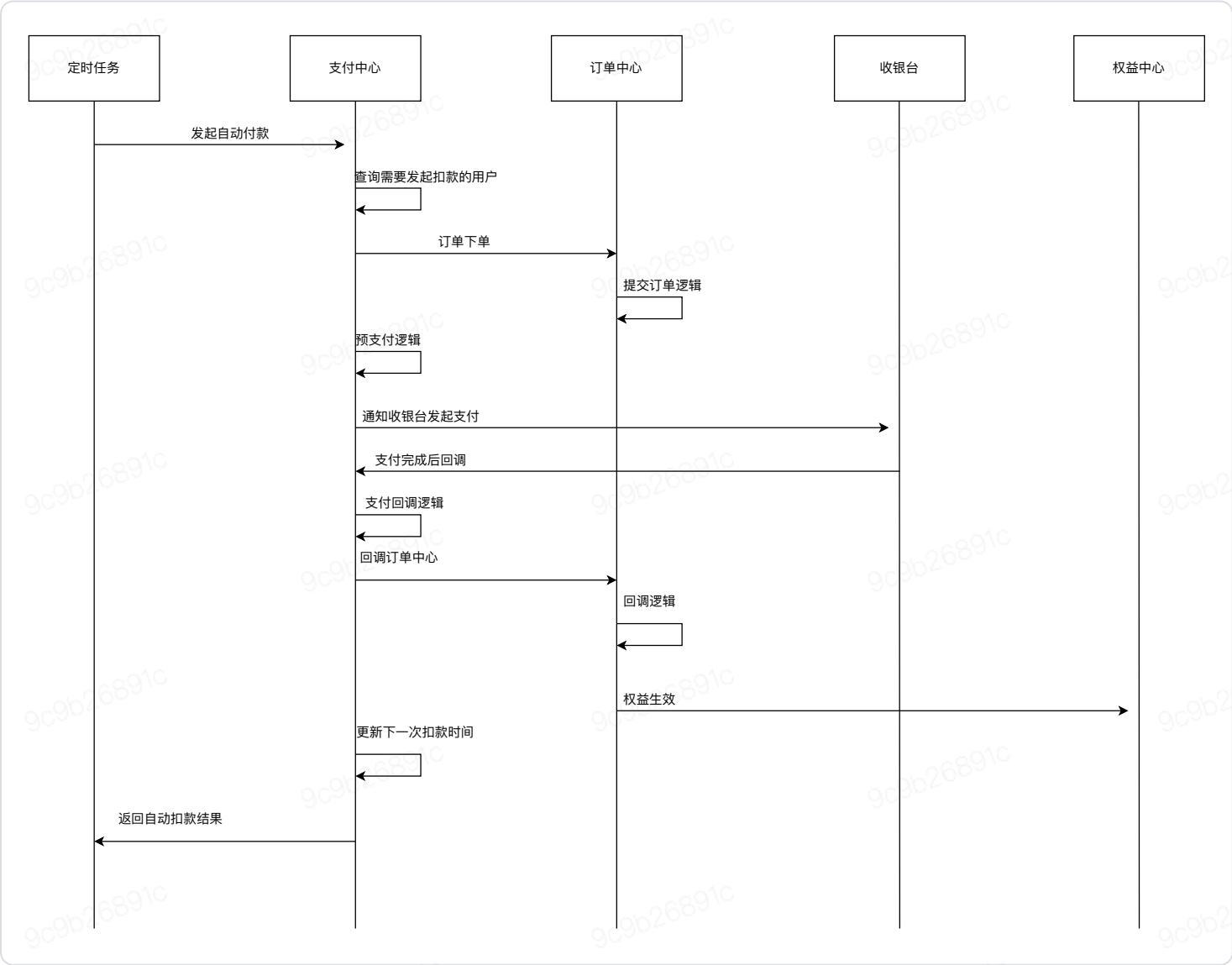
9. 短信通知

10. 状态更新为充值成功

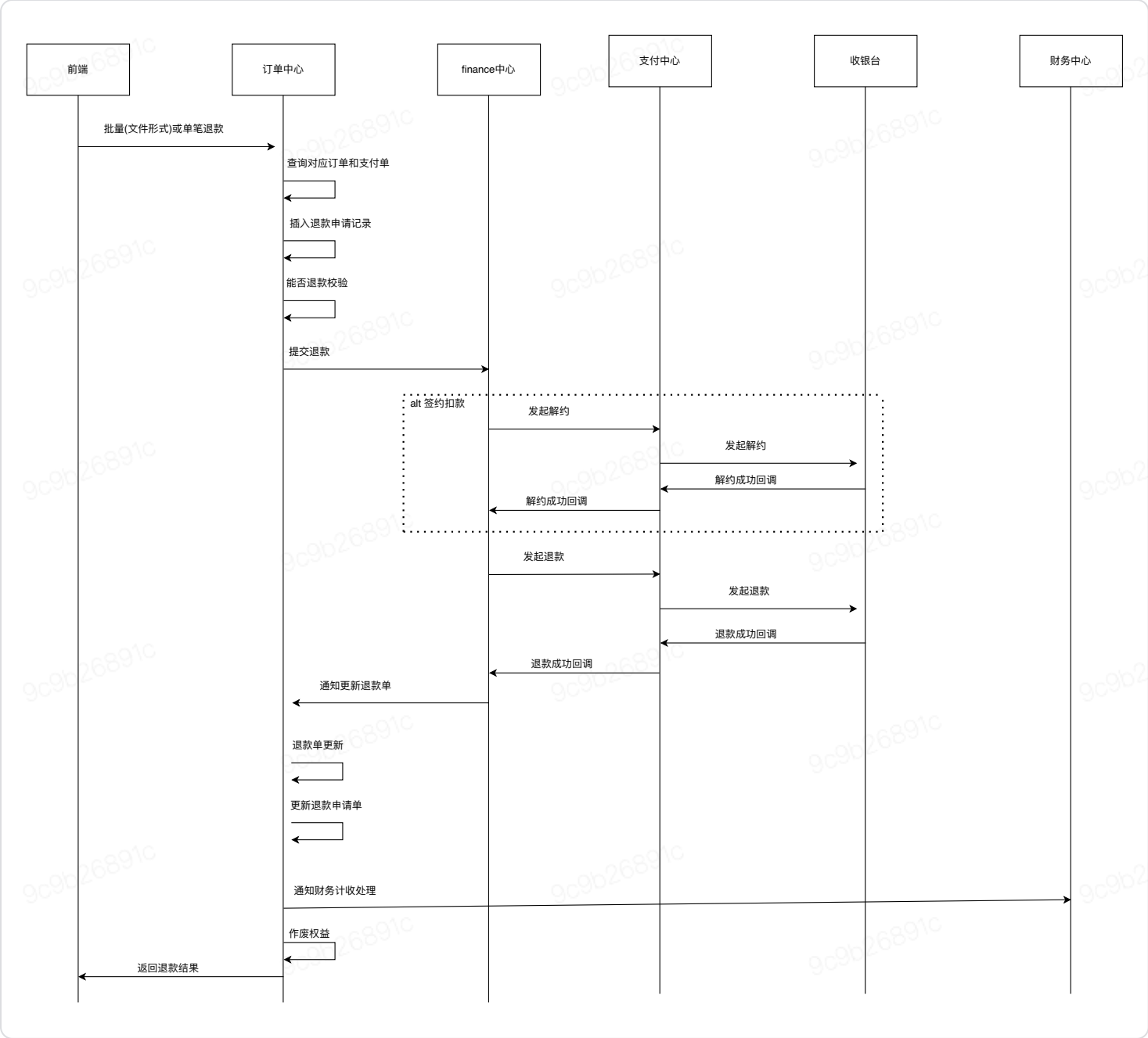


返回结果

• 定时自动扣款

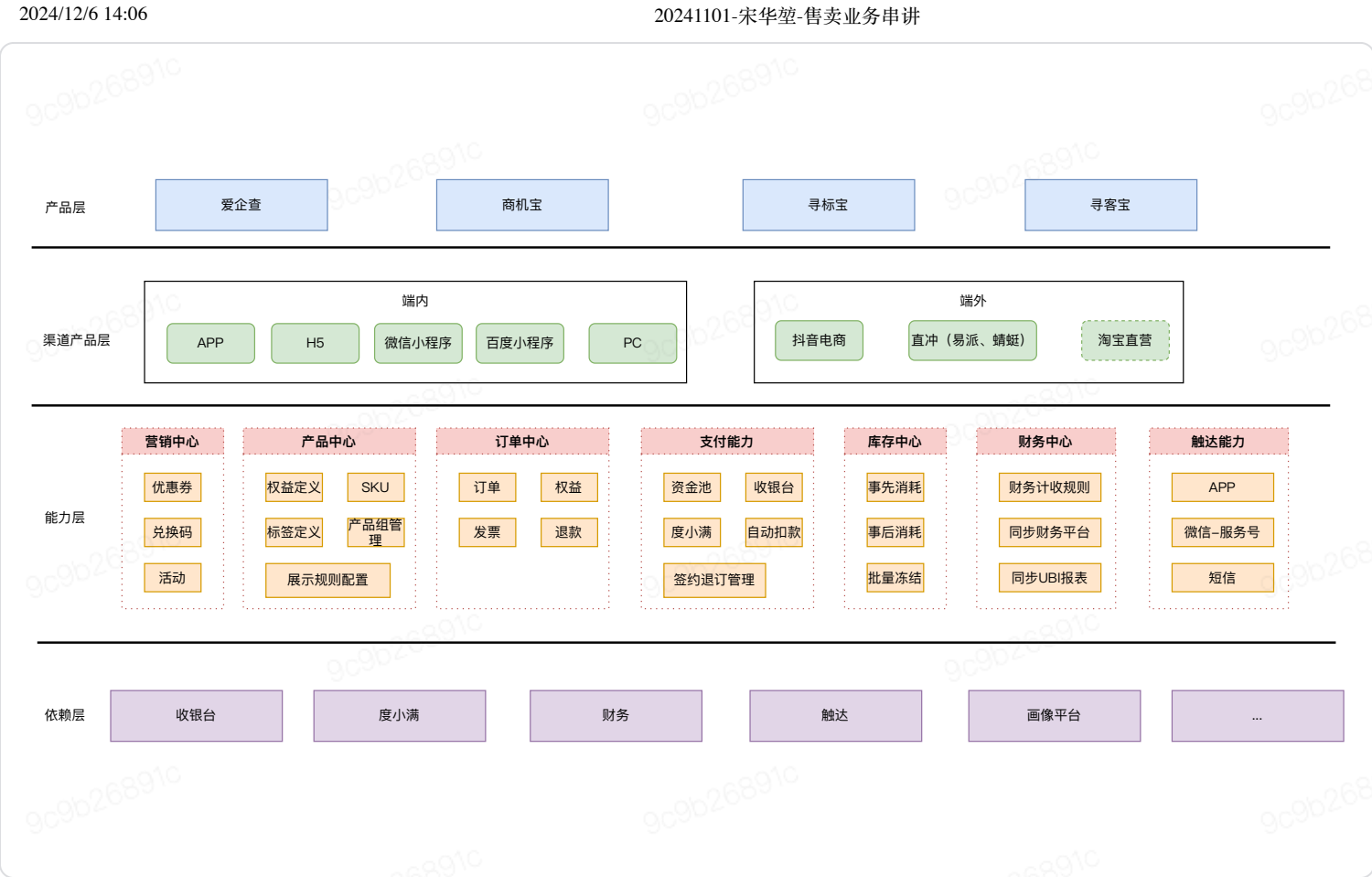


• 退款



2.系统梳理

2.1应用架构



15		/payNotify/syncSignPay	订单服务	支付中心	签约扣款关系
16		/pay-center/bizcrm_pay_center_sign_pay_minute	定时任务	支付中心	发起自动扣款
17	退款	/refundApply	前端	订单服务	申请退款
18					
19					

2.3数据模型

- i

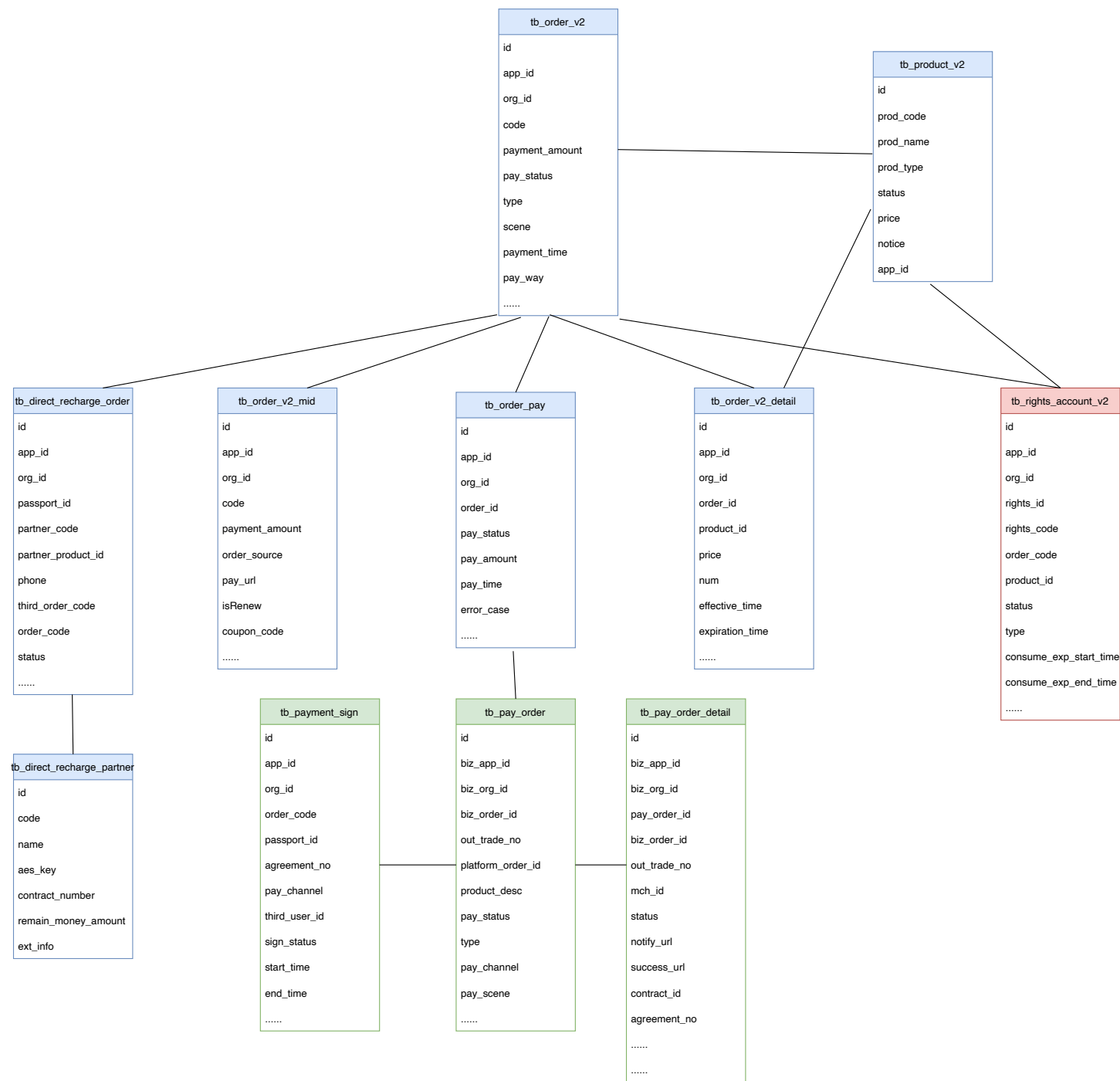
当前DB存在的问题：
- 1、

详情信息表和订单表的字段信息混杂。理论上每条订单实体代表一次业务行为，只需要记录关键的业务字段即可（如状态、用户、场景、金额、产品等）。而详情表记录的是业务快照信息，作用类似于日志流水，后续新增的冗余信息应该放在详情信息表。
- 2、

存在多个微服务共用同一个数据库，不符合微服务的隔离性，建议对数据库做拆分。
- 3、

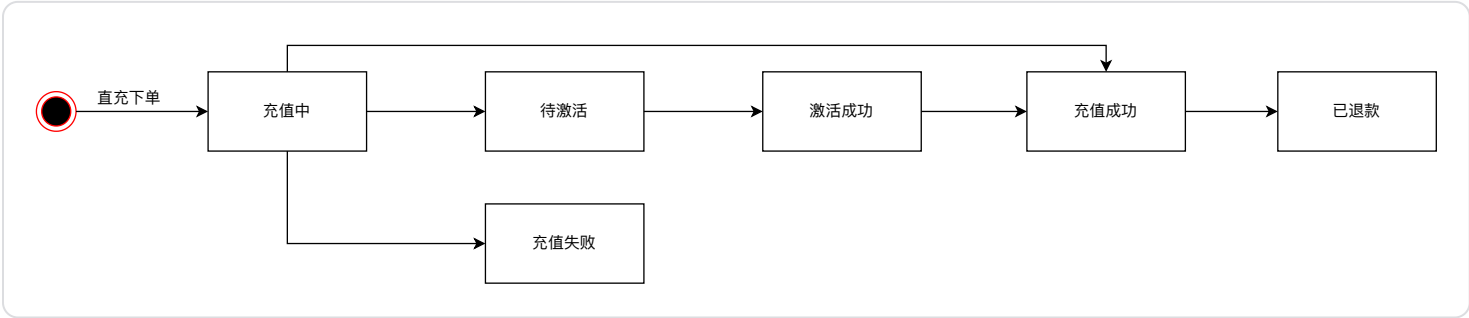
tb_order_pay从职责上看，主要用于记录发起支付和支付完成后的信息，职责与order_v2重合，可以被合并到order_v2。

概念模型

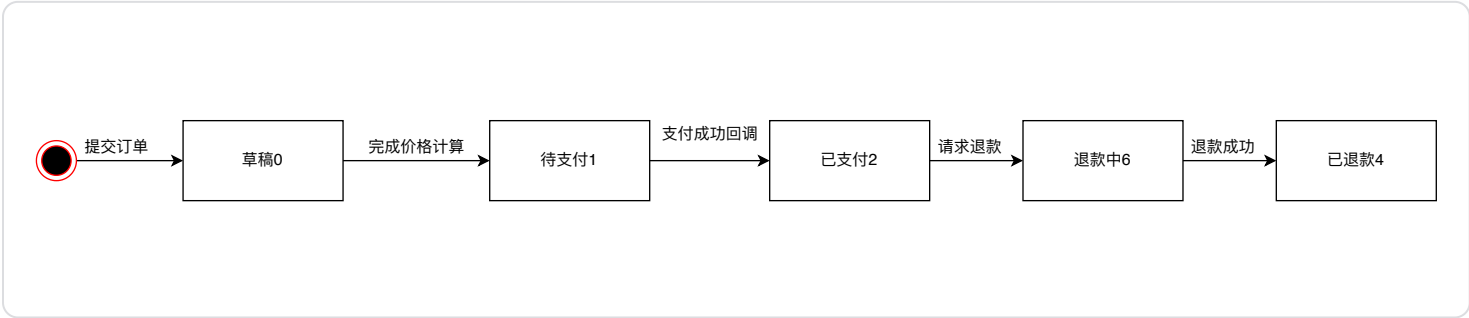


2.4 状态机梳理

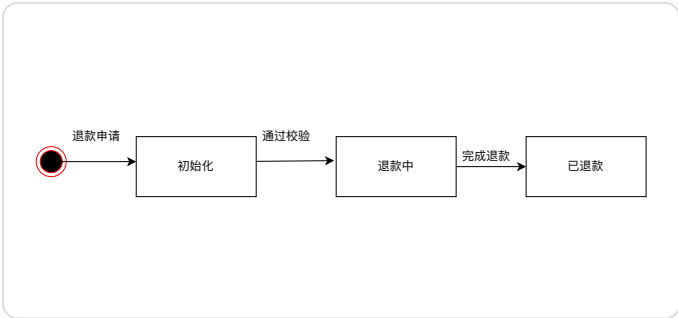
1、直充订单



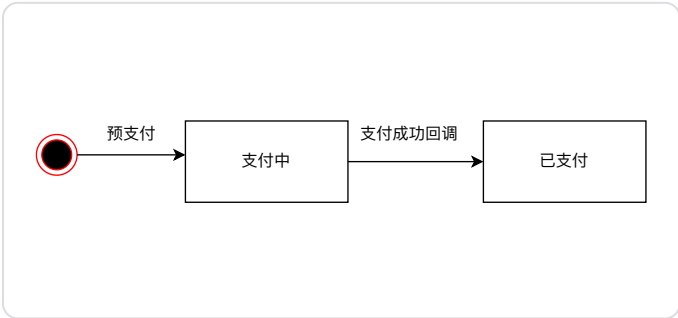
2、orderV2订单和orderPay订单



3、退款申请订单



4、支付中心订单



3.稳定性建设方面

3.1业务告警

1	业务场景	监控项	监控目标	类别	埋点方式	告警策略	告警内容
2	aff-order服务下所有接口	接口耗时	服务稳定性	超时		10分钟内中至少3分钟p95时间超过10000ms	cce-aff-or
3	aff-order服务下所有接口	业务异常	接口可用性	报错		60秒内超过100次	【鲁班】[9能有非预
4	aff-order下单核心链路告警	业务日志异常	接口可用性	报错		1分钟内出现2次	订单核心
5	aff-order下单接口(queryToFramePay)超时	接口耗时	接口可用性	超时		1分钟4次	订单接口
6	[融合会员]-自动续费订单监控&异常监控	业务日志异常	业务异常	报错		1分钟2次	自动续费
7	第三方支付回调获取明文验签失败	业务日志异常	接口可用性	报错		30秒内存在1次	cce-pay-c调获取明
8	支付中心接口可用性不达99.99%	接口耗时	服务稳定性	超时		1分钟内服务稳定性不达99.99%	cce-pay-c99.99%
9	更新权益时间失败	业务日志异常	接口可用性	报错		1分钟内存在1次失败	cce-aff-ric间失败
10	权益中心p95超3000ms	接口耗时	服务稳定性	超时		10分钟内中至少3分钟p95时间超过3000ms	cce-aff-ric

3.1.1业务告警大盘

<https://aff.baidu-int.com/app/grafana/alerting/list?view=list>

3.2.2限流策略

3.2资金安全建设

1	风险点	风险评估	当前策略	状态	解决思路
---	-----	------	------	----	------

2024/12/6 14:06		20241101-宋华堃-售卖业务串讲			
2	订单系统、支付系统和收银台之间单据状态、支付金额等关键信息的一致性	高	月度和天维度的对账		业界通用的一 一、技术设计 1、幂等处理和 2、分布式事务 3、补偿或重试 二、数据核对 1、实时核对： 中使用可配置 对，例如 链接 统 2、离线核对： 定时任务扫描 量核对
3	已支付订单和用户权益的一致性，如何保证用户支付后用户权益生效时间符合预期	高	定时任务重试对失败的订单进行补偿		
4	订单和扣款签约关系的一致性，如何保证用户下单连续包月产品后，扣款签约关系符合预期	高			
5	签约扣款的用户未定时扣款	高	通过sugar对数据做监控		
6	防止用户薅羊毛行为	中	通过sugar对数据做监控		
7	售卖系统和公司财务系统之间的数据核对	高	自动对账平台，每月初对账		

4.问题总结

4.1业务问题

- 1、运营端退款流程复杂且长，理解和改动成本高，可以对退款流程做简化。
- 2、在多个下单环节（预下单、提交订单、预支付、支付回调）中存在与优惠券服务的多次交互，我理解是否只需要预下单和支付回调两个环节与优惠券服务做交互就可以了，建议对这块逻辑做简化减少业务依赖。
- 3、线下退款依赖手动插入db和手动调用接口，存在操作风险，建议建设相关运营平台能力。

4.2系统建设问题

- 1、在业务流程推进中，系统大量使用回调和领域事件的方式，在问题排查过程中查找对应日志非常麻烦，建议当发生回调和领域事件时在日志中能更明显地标注出来，并说明具体的消费接口，或者优化链路追踪系统在领域事件的消费者中增加生产者的trace，减少排查成本。
- 2、实时性的资金安全建设有缺失，不利于及时发现和定位问题，造成月初对账压力大，建议先用定时任务扫描db的短期方案快速补齐监控能力。
- 3、aff-order旧的代码逻辑未完成下线，导致服务内部业务代码较混乱。
- 4、业务监控不完善且比较分散，缺少集中统一管理，查看监控效率比较低。