聚合支付相关

功能需求:

1. 多通道统一接入api
2. 每种通道隶属场景分类
3. 每种通道各自费率
4. 商户资料代理
5. 多通道自动切换
6. 统一账务清算

1.清算账户;2.清算通道

1.支付加密;2.用户支付;3.回调确认

1.商户; 2.场景;3.可用通道;4.通道配置;5.通道账户;

Repuest API

1. 用户角色

商户与通道,相互透明, 在支付实际请求中, 均以子账号形式请求( 依赖与通道是否支持, 不支持则自建用户信息管理, 并在提交支付的每个阶段记录流水与支付状态) , 在清算日按按商户对应的子账号进行清算 ( 我们与通道清算之后, 再分别计算每个子账号的金额, 初期可能人为计算 )

商户的配置 , 包含但不限于相关资料 , 开通的并使用的通道 ( 基于各个通道的情况不一致, 有的通道更敏感 , 选择性的开通更好 )

1. 多通道的选择

每个支付请求, 都需要区分是有那个子账号( 商户 )发起的, 并在当前的场景分类下( 安卓设配 和 pc 显然支付流程是不一致的 )的可用通道中选择其一进行后续支付 ( 需保正,任意支付通到的用户体验是完全一致的 ) ;

通道选择的策略, 子账号 ( 商户 )必须是开通的对应通道( 这里的开通是指人为审核 ), 按照预定策略选择 ( 伐值型 , 权重型 , 业务型 , 复合型等 ) , 每种都应有单独的数据库关系来表示, 而最终的选策略 ( 复合型 ) , 则应由更多一张数据表表示;

1. 清算

我们与各个通道清算得到总值 , 我们与各个子账号清算得到子值

1. 容错

每个支付请求需要流水记录 , 每个支付请求需要加解密合法方能通过 ,

需要统计子账号( 商户 )的支付行为, 包含但不限于时间, ip , 金额, 描述

1. 流量控制

当一个请求过来, 先进入令牌桶容器 , 一来避免网络流量攻击(ddos)造成系统信息紊乱, 二来在用户激增的情况下防止非预期的请求对系统压力过大而引起的系统瘫痪，当流量过大时，可以采取拒绝或者引流等机制.

1. 兼容性

当接口因为功能增加,业务调整等原因,是的当前接口无法满足需求时, 应该通过,接口版本号来控制接口升级, 而非使用另一个接口 ; 同样的, 当接口的需要对接多个上下游通道时, 应该始终保证接口以及参数的统一 , 参数可以冗余而不能缺少;

7.