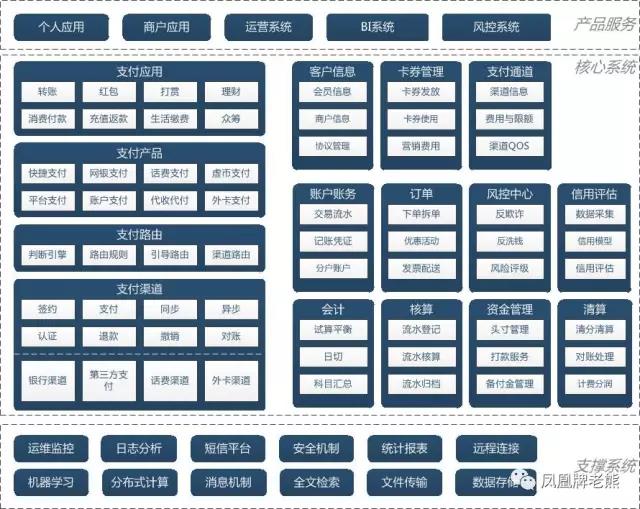
# 支付系统设计总结

### 系统架构

一个完整的支付系统架构包括如下模块：



以笔者所在的公司系统为例：

* 产品服务层为个人应用。
* 核心系统层中，支付应用包含转账，理财购买，生活缴费，消费付款。
* 支付产品主要是系统内部建立的电子账户-二类账户作为支付。
* 支付路由主要用于判断付款账户及收款账户走什么渠道。
* 支付渠道包括二代支付和银联，帮付宝等渠道。（系统无法对接所有银行，所以通过银联等中间协调，完成对各个银行卡渠道的支持。）
* 与支付相关的模块应包括账户模块，核算模块，支付模块，对账模块。

接下来，从数据库设计的角度，来描述各个模块如何设计。

### 模块设计

#### 2.1 账户模块

账户系统的设计分为客户账户和内部账户。客户账户主要是针对每一个用户的持有资产。而内部账户是为了记账而开立的属于系统使用的账户，比如申购归集户。

* 设计客户账户，最基本的应该有一个账户表和账户流水表。其中账户表t\_usr\_acount记录客户的账户号，用户号，账户余额，账户状态等。t\_acount\_list账户流水表登记交易时间，交易金额，借贷标识（借表示从该账户支出，贷表示向该账户入账），付款账户和收款账户等等。
* 设计内部账户t\_inner\_account，应包括账户，余额，上日余额。用于核算的科目序号等。对应的也有一个内部账明细表 t\_inner\_account\_list，记录内部账的交易时间和交易金额。

实际的系统设计可能会更为复杂，包括：1.可能业务需要账户冻结，或者监管要求系统可以对账户进行不可支出的控制，因此还要设计账户的冻结，止付状态。并且有对应的止付，冻结记录表。 2.某些监管会要求账户一天或者一次支出金额的限制，需要为账户设计限额表。3.对于内部账户，可能某个业务所有用户账户都需要和一个内部账户交易，而数据库的读写性能有瓶颈，还需要设计用于异步入账的内部户影子分户。

实际的账户系统设计远不止这些，根据不同的系统设计和业务场景，会要求更多字段和数据库表。在以上的基础上，当一个支付接口调用账户模块时，可能就会涉及到：

账户状态的检查——账户冻结和止付的检查——账户限额检查——账户余额检查等。

在多项检查通过后，才有后续的记账等。

如果是为商户服务，还会为商户设置有“可用余额”账户、“待结算”账户。用户付款给商户，钱打到待结算账户（平台收一点手续费）。然后周期的吧待结算账户中的钱转入可用余额账户（商户随时可提走）。

#### 2.2 核算模块

这个模块是一个专门用于记账的模块。系统所有内部的账户变更，都统一调用此模块进行记账。它要求对用户的余额变更是同步的（一般使用数据库行锁保证单个用户的账户余额查询+更新原子操作）。

核算其实就是记账，它的表包括：核算科目表t\_sys\_sub，用户和内部账户都有自己的科目属性。而核算科目表就记录了这些科目的核算代码，余额借贷方向，核算代码类别（资产类还是负债类），是否红字（允许账户为负）。t\_event记账事件表，每一种记账事件，都配置在事件表中，比如用户消费购物，用户购买理财产品这都是一个记账事件。一次记账事件可能包含多个记账调用。因此需要设计一个记账参数表t\_event\_call规定某一个事件下有多少个调用。而每一条记账参数有一个对应的记账引擎表t\_call\_detail的复式记账规则。（比如用户付款给商户100，平台收取6毛手续费，商户收取99.4元，这就是一次记账调用）。那么这个例子就需要三条数据库记录分别描写同一个记账代码下，关联的三个账户的借贷关系。

另外还有一个t\_call\_detail\_list，即核算流水流。记录每一笔借贷的详细流水。也就是发生一次记账，至少都会有两笔记录，一借一贷，借贷相等。

核算模块的重点是：1.通用可配置，即任务应用场景都可以调用该模块进行记账，只要配置了记账事件，以及该事件的记账引擎账号分录。然后调用的接口传入记账事件，金额和账户，就可以进行记账。 2.事务隔离，即对同一个账户的余额变动必须保证是同步的，并发的查询+修改可能引入脏数据。一般通过数据库的行锁保证。

#### 2.3支付模块

账户和核算模块其实都是为支付模块服务的（其实核算模块还包括结息，计提等业务。但是非银行的系统一般不会给用户账户算利息，或者都是虚拟货币）。

支付系统的设计需要哪些表，跟业务有很大的关系。比如一个普通的app，可能通过集成支付宝和微信的SDK就可以充值提现了。但是类似淘宝或者京东，它允许用户通过银行卡去支付。而不同的银行卡的支付接口都是通过银联统一的，因此当一个用户选择使用工行的银行卡支付时，可能流程如下：

获取银行卡——获取卡bin，查找支付通道——根据指定的支付通道调用支付网关。

所以后台系统并不需要对接所有银行，只需要对接银联或者二代支付，帮付宝这样的第三方支付系统。所以我们的数据库设计如下：

1. 订单表，记录某笔交易的订单金额，名称，状态等。
2. 银行卡bin与通道对应表。
3. 支付通道表
4. 支付流水记录表

#### 2.4 对账系统

对账系统的存在，是因为实时的接口调用经常会因为网络不稳定，而无法获取支付结果而设计的。双方通过文件交互，告知对方：我收到了多少笔你发给我的请求，时间和金额是多少。然后对方系统在自己的系统中进行逐笔核对。当出现双方的状态不一致的记入差错表。以便人工处理。

因此数据库设计如下：

1. 对账流水表
2. 差错明细表。

### 交互方式

实时接口通过网关交互，一般充值，提现，购买基金等的成功场景都是通过该接口完成。

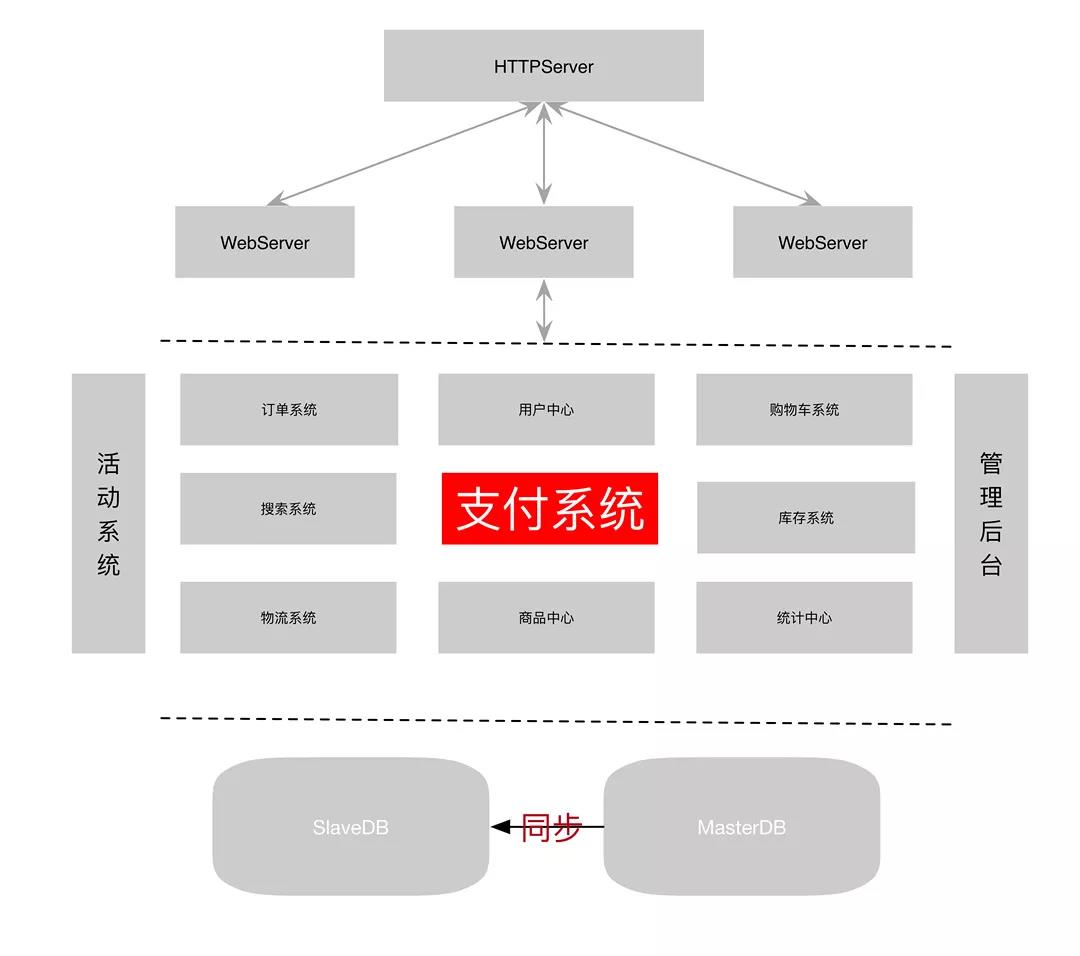
对账和一些业务如长期理财产品通过批量的对账文件或确认文件来确认交易的结果。比如购买理财时，仅仅是冻结了用户的金额，并没有产生分录，等第二天对方返回我确认购买成功时，才调用记账引擎给客户扣款。或者对于充值提现等，因为超时无法确认结果时，通过回调接口或者批量轮询接口查询订单状态。并且在晚上生成对账文件和对方确认金额一致后，再将内部户中的钱划给对方。

### 支付中的坑

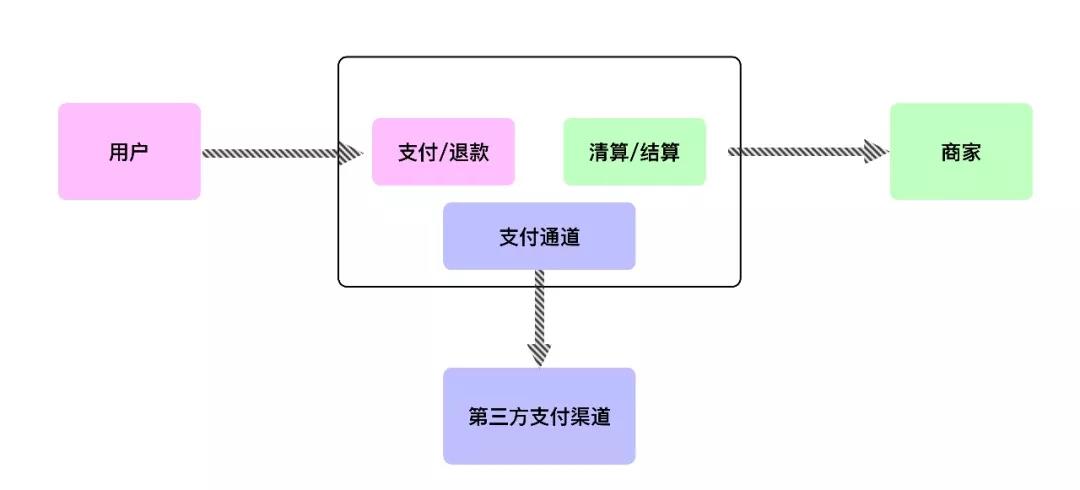
支付系统中遇到过的一些坑，如果解决不好可能会对资金产生影响，产生长款，短款等。

* 重复记账：一般前端会控制按钮置灰，防止用户多次提交支付请求。但是如果前端没有做控制，那么后台分布式服务并发时，（或单机并发时）会发生重复记账的问题。一般可以通过数据库行锁+订单状态控制，如果有跨原子的事务，可能会需要用到分布式锁。
* 接口超时：再调用扣款时，应该先记账再调第三方接口，并且对方无响应导致超时的时候，不能将订单置为失败。应该等待回调或者轮询批量查询到对方返回的真实订单状态。
* 加签和验签可以保证报文是指定系统发过来的。RSA算法签名方使用私钥加签，接收方使用公钥验签。（接收返回时使用另一个自己的私钥）。MD5算法双方使用约定的秘钥签名，并且通过对比返回的签名结果判断报文是否被篡改。MD5算法即使别知道了秘钥也没关系。因为要保证参数不同，签名相同时非常困难的。RSA私钥被知道了，会有风险嘛？（RSA加密使用公钥，解密使用私钥。RSAUtils.java提供了生成公钥私钥对，以及加密解密的方法）。

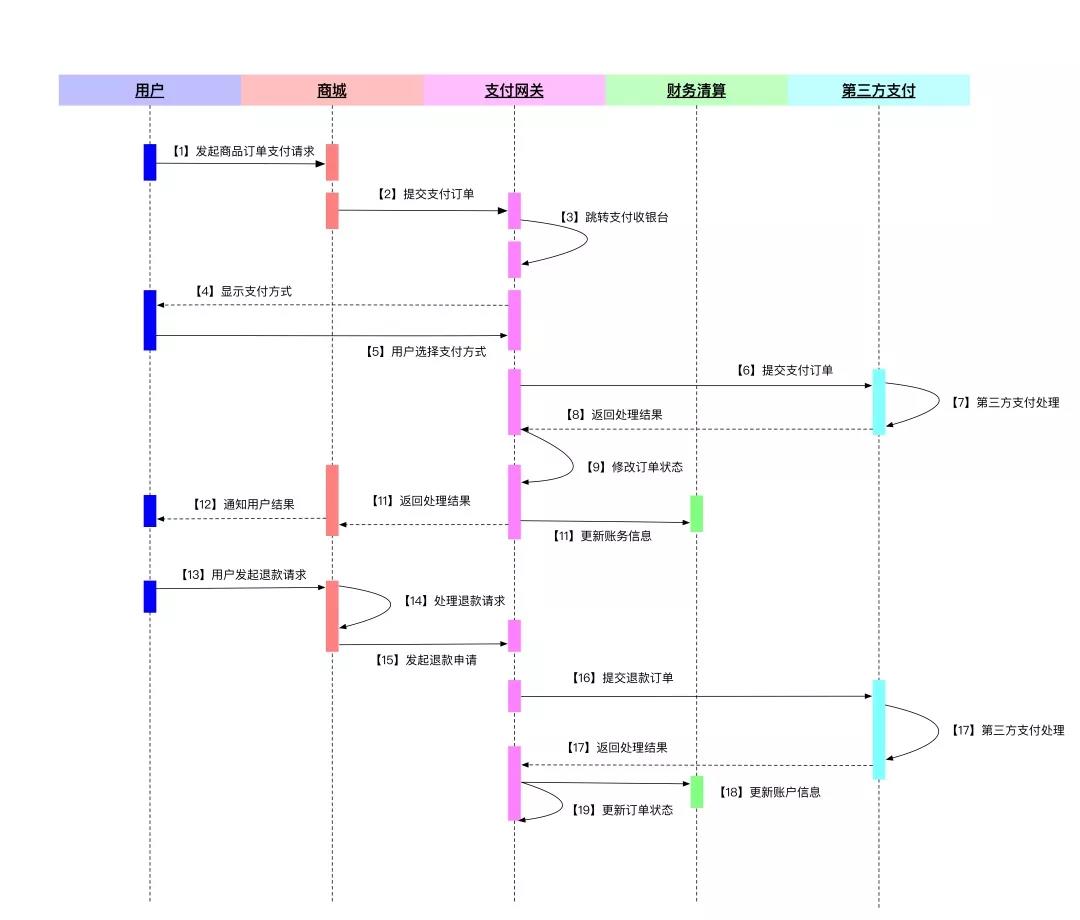
### 支付系统图示



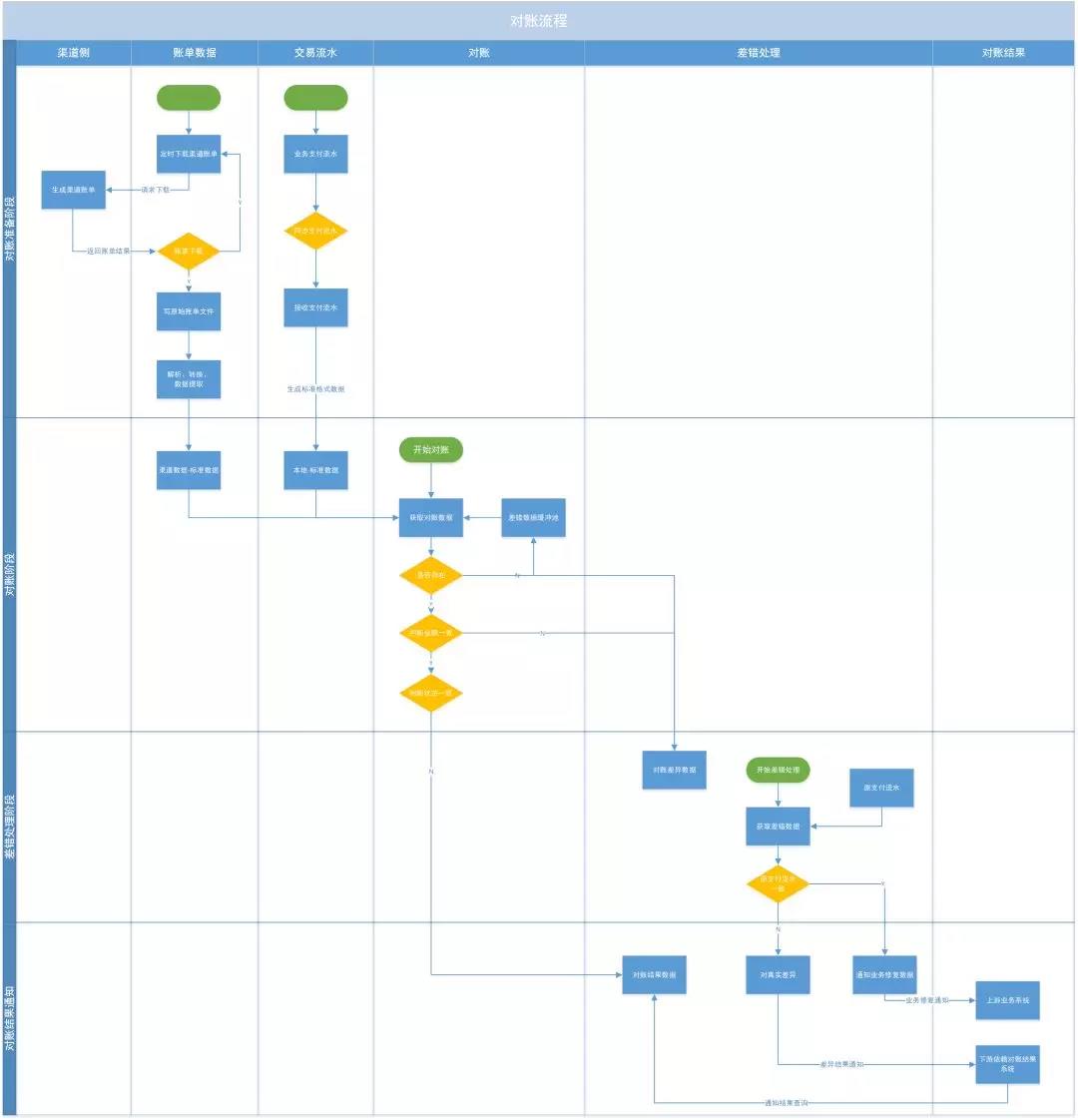
表格 1电子商务网站以支付系统为核心



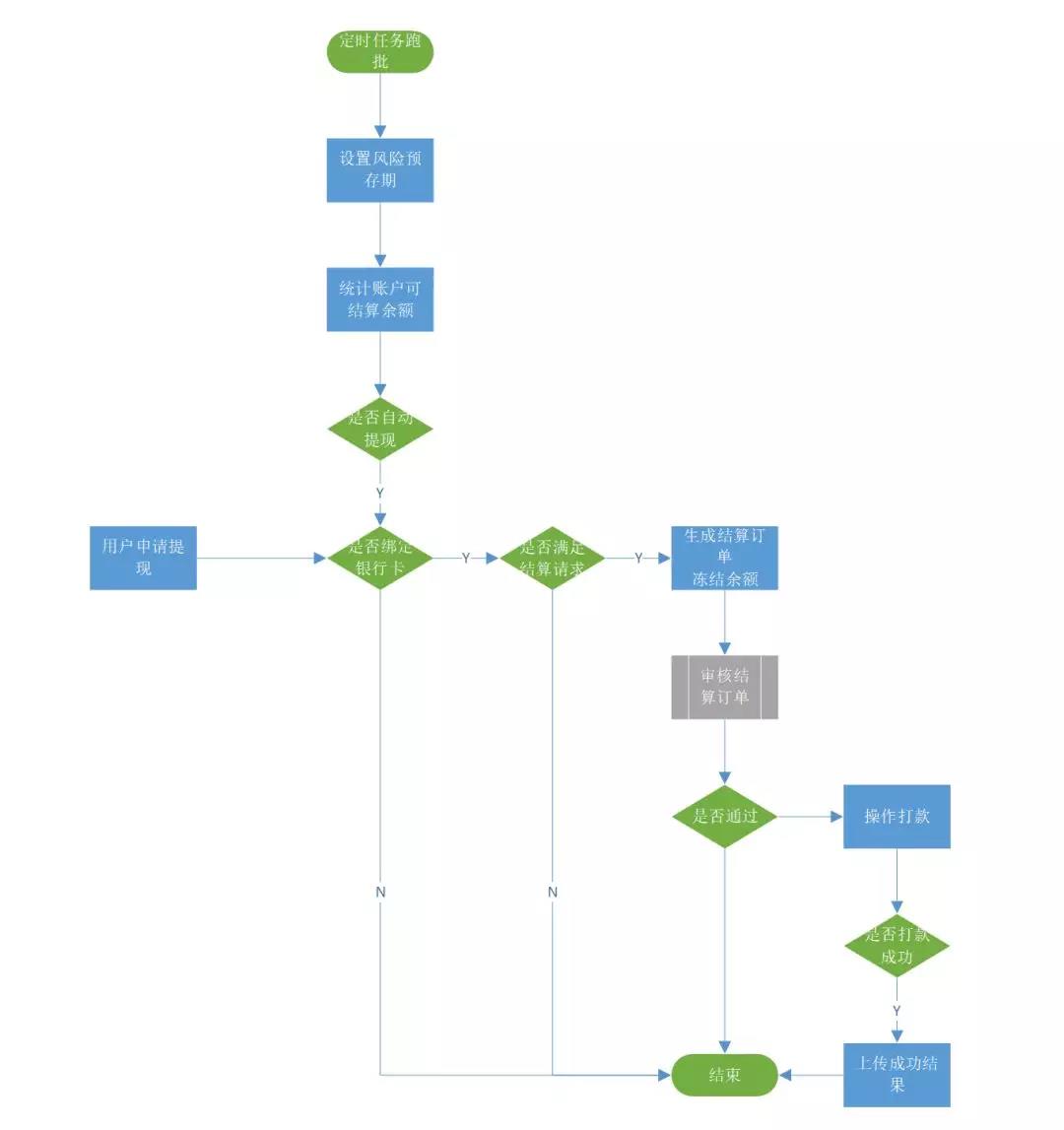
表格 2系统和商家清算/结算



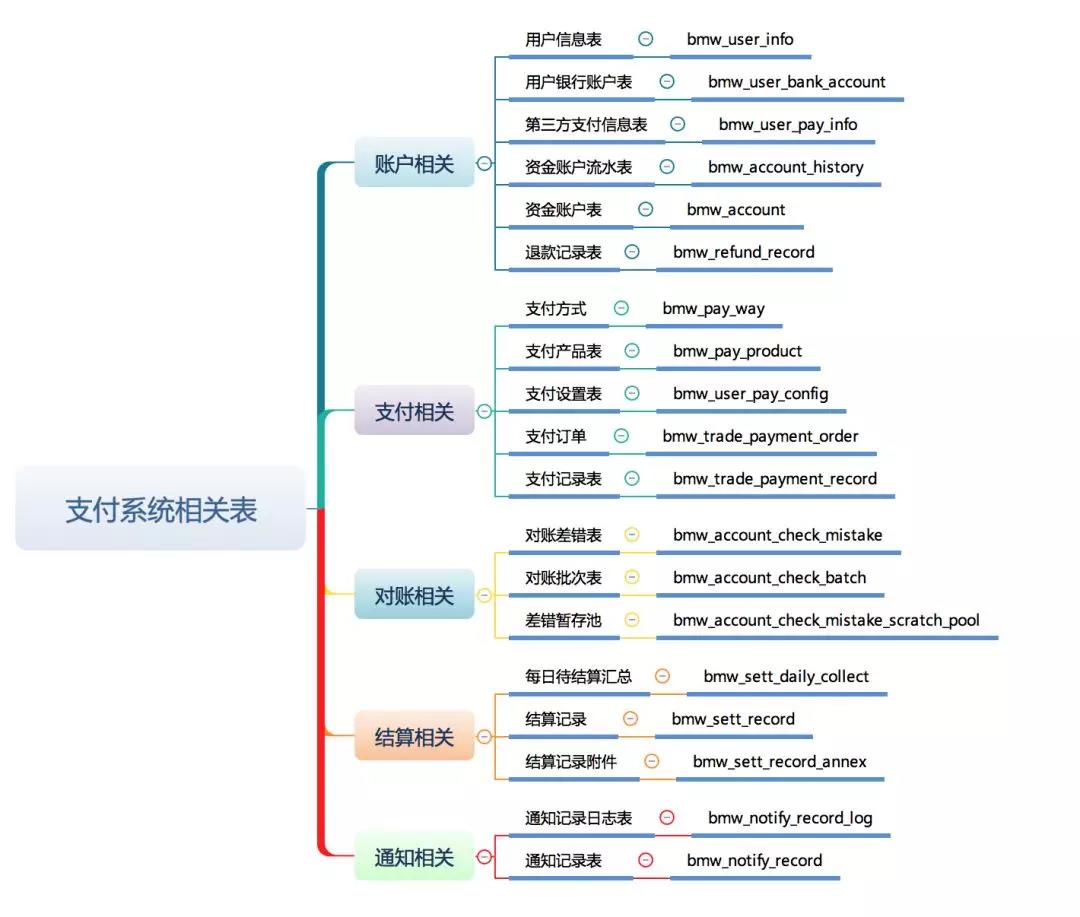
表格 3支付和退款时序图



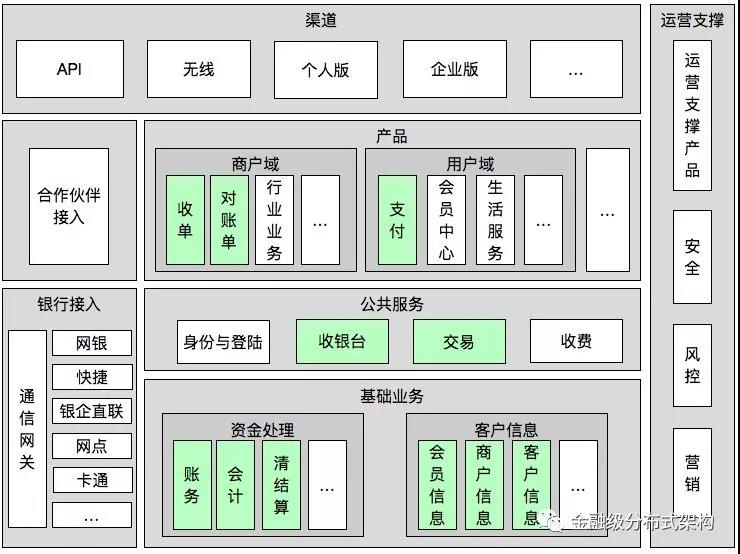
表格 4对账



表格 5结算流程



表格 6表格设计



表格 7支付系统的应用架构

#### 5.1交易流水和记账

每一笔交易都需要记录流水，并登记到个人和机构的分户账户上，统计和分析也需要根据交易流水来更新相关数据。 而个人和机构账户总额更新、交易流水记录以及库存的处理，更是需要事务处理机制的支持。 从性能角度， 可以弱化了事务处理的要求，采用消息机制来异步化和交易相关的数据处理。

在支付网关前置的主流程中，仅记录交易流水，即将当前的请求保存到数据库中。

完成数据记录后，发送MQ出来，记账、统计、分析，都是接收MQ来完成数据处理。

涉及到本地资金支付，比如钱包支付，会需要分布式事务处理，扣减账号余额，记账，扣减库存等，每个操作失败，都要回滚。阿里有很不错的分享，这里不详细描述。

当交易量上来后，需要考虑交易表的分表分库的事情。分表分库有两个策略，按照流水号或者交易主体id来走。后者可以支持按用户来获取交易记录。我们用的是前者。后者可以走elastic，确保数据库专用。风控，信用和统计所需要的数据，通过MQ同步到历史库里面。作为支付系统最有价值的数据，在存储上做到专库专用，无可厚非，毕竟存储成本还是廉价的。

#### 5.2支付路由

支付路由是一个复杂的话题。对支付系统来说，能支持的支付方式越多越好，不能由于支付方式的不支持断了财路。现实中的支付方式多得难以置信。用户随时甩出一张你听都没听说过的卡。如果一个银行卡只有几个用户在用，那针对这个卡开发个对接有点得不尝失。现在第三方支付的爆发，确实给开发支付系统省了不少事。但是公司不可能只对接一个第三方支付，如果这个渠道出问题了，或者闹矛盾了，把链接给掐了，老板还不欲哭无泪。总之，得对接多个渠道。对于交易量大的银行，还得考虑直联

#### 5.3渠道接入

对于支付渠道，首先考虑的是接入哪些渠道。要对接的渠道按优先级有：

第三方支付，对大部分应用来说，支付宝和微信支付都是必须的，一般来说，这两者可以占到90%以上的交易量。用户不需要绑卡，授权后直接支付就行。各种平台都支持，性能和稳定性都不错。对于一些特殊业务，比如游戏，企业支付，可以查看一些专用的第三方支付平台。

银联，它的存在，极大方便了和银行的对接。和第三方支付主要不同在两个地方一是需要绑卡，也就是用户先把卡号，手机，身份证号提供出来。这一步会折损不少用户。绑卡后，以后的支付操作就简单了，用户只需要输入密码就行。手机客户端不需要像第三方支付那样安装SDK，都在服务器端完成。当然，这是针对快捷支付。网银支付还是挺麻烦的。银联接入也需要ADSS认证。

银行：2018年2月9日银监会公布了最新权威数字：一共【4549家】开发性金融机构1家：国家开发银行；政策性银行2家：进出口银行、农业发展银行；5大国有银行：工、建、农、中、交；邮储银行1家；全国性股份制商业银行12家：招行、中信、兴业、民生、浦发、光大、广发、华夏、平安、浙商、渤海、恒丰；金融资产管理公司4家：信达、华融、长城、东方四大AMC；城商行134家；住房储蓄银行1家；民营银行17家，如网商银行；农商行1262家；农村合作银行33家；农村信用社965家；村镇银行1562家；贷款公司13家；农村资金互助社48家；外资法人银行39家；信托公司68家；金融租赁公司69家；企业集团财务公司247家；汽车金融公司25家；消费金融公司22家；货币经纪公司5家；其他金融机构14家。一般对接一个银行预计有3周左右的工作量，大部分银行需要专线接入，费用和带宽有关，一年也得几万费用。不同银行对接入环境有不同要求，这也是成本。

手机支付：比如苹果的In-App支付， 三星支付、华为支付等， 这些支付仅针对特定的手机型号， 支持NFC等，根据业务需要也可以接入。