

# 图解支付系统的渠道路由设计

原创 隐墨星辰 隐墨星辰 2024-06-01 09:49 广东



**隐墨星辰**  
10余年顶尖支付公司架构设计经验，精通国内和跨境支付业务及系统架构设计  
35篇原创内容

公众号

大家好，我是隐墨星辰，今天和大家聊聊渠道路由设计。

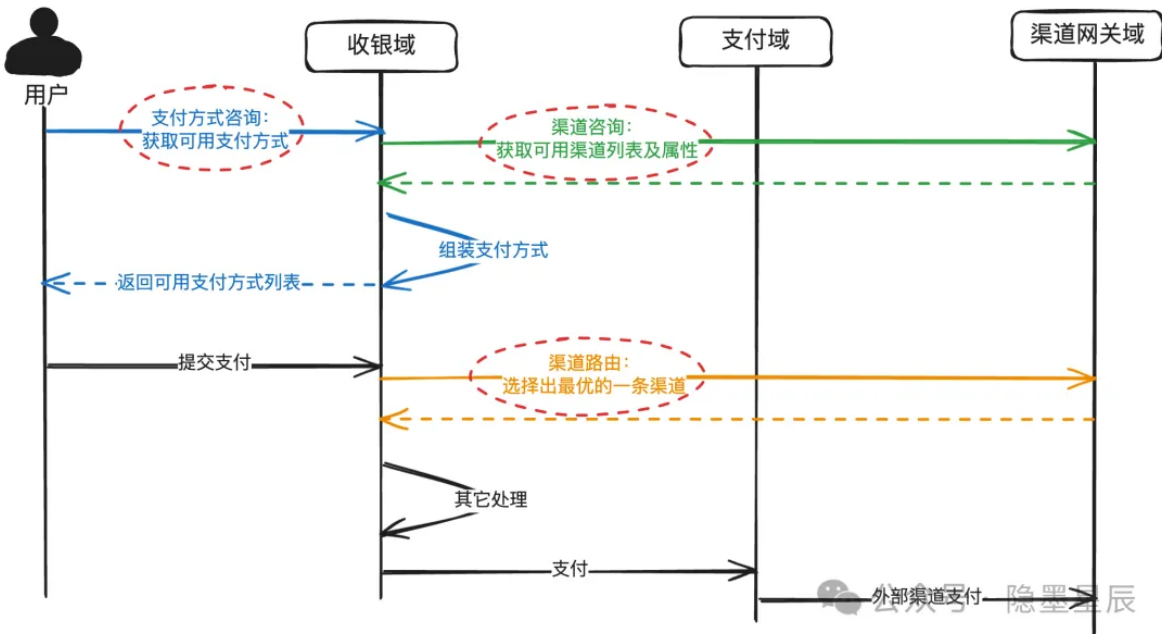
这篇文章主要讲清楚：渠道路由是什么，为什么需要渠道路由，渠道路由的几种形态，一个简洁而实用的基于规则的渠道路由设计。

注：有些公司称渠道为通道，都是一个意思，为方便起见，本文统称为渠道。

## 1. 一些背景知识

有些支付公司没有区分**支付方式咨询**、**渠道咨询**、**渠道路由**，而是**混在一起做掉**，这样的好处是简单而实用，缺点是扩展性不足。下面将以扩展性最好的拆分方式来讲解。

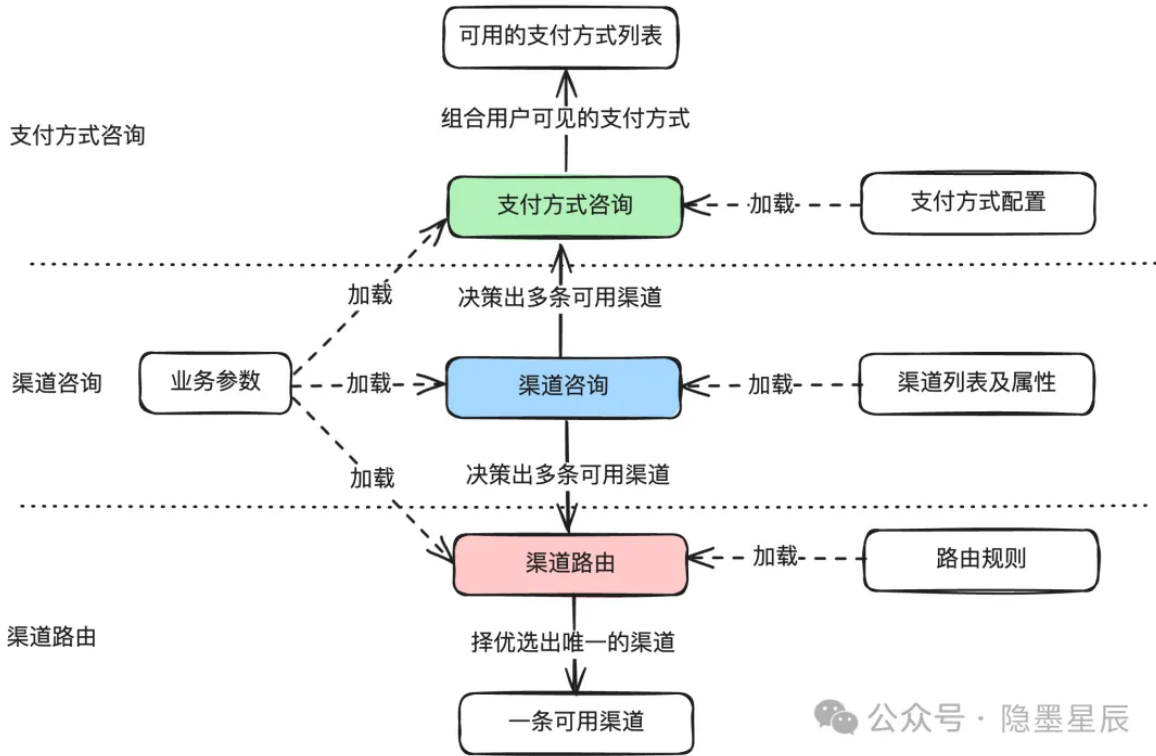
下面是三者之间的简单关系图：



说明：

- 支付方式咨询**：根据用户的请求，组装**可用支付方式列表**返回给用户。由收银域提供服务。
- 渠道咨询**：根据用户的请求，组装可用的**渠道列表**和**渠道属性**返回给收银域，再由收银域转换为支付方式返回给用户。由渠道网关提供服务。

3. **渠道路由**：当有多个渠道可用时，选择出**最优的一个渠道**。由渠道网关提供服务。



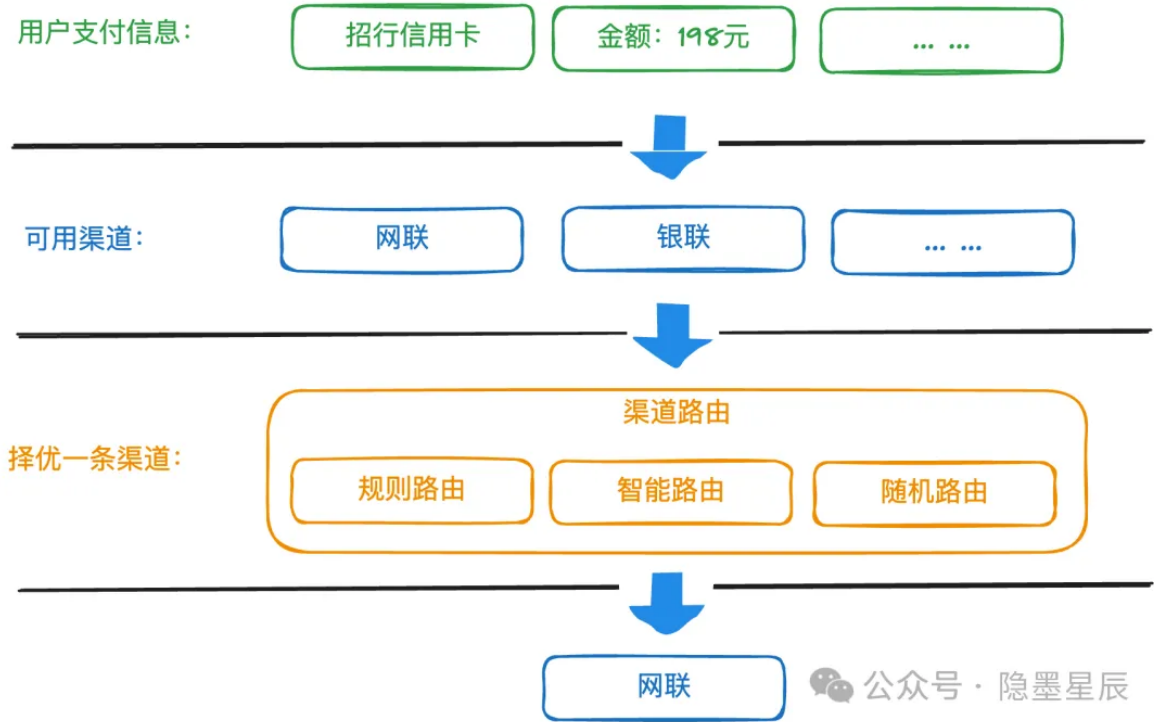
再详细一点，如下：

**支付方式咨询**：解决用户可以使用哪些支付方式，比如余额、招行借记卡、招行信用卡等。比如虚拟商品不能使用信用卡，这种支付方式的管理就是支付方式咨询的职责。

**渠道咨询**：解决是否有渠道可以支持当前的支付行为。比如用户绑定了招行借记卡，但招行当前正在维护无法提供服务，那就将渠道状态设置为不可用，收银域对应的支付方式会置灰。

**渠道路由**：解决最优渠道问题。需要综合支付成功率、支付成本、用户体验、渠道状态等多种因素挑选出最优的一条渠道。

## 2. 渠道路由核心作用



渠道路由核心作用是当有多个渠道同时满足业务诉求时，综合**支付成功率、支付成本、用户体验、渠道状态**等多种因素**挑选出最优的一条渠道**。具体如下：

- 1. **提高支付成功率**：通过选择最合适的渠道，可以提高支付的成功率，减少支付失败带来的用户流失。原因在于不同的渠道在其内部的风险偏好是不一样的，同一个请求在A渠道会失败，但在B渠道会成功。
- 2. **优化成本**：不同渠道的费用可能不同，通过合理的路由，可以降低支付成本。一些渠道还有阶梯收费，需要通过分流不同的渠道，保持整体成本最优。
- 3. **提升用户体验**：快速、稳定的支付体验能增强用户的满意度和忠诚度。用户如果经常在A渠道支付，新的请求过来后，仍然发给A渠道支付的成功率往往会更高。
- 4. **负载均衡**：将支付请求合理分配到不同的渠道，避免某个渠道过载，提升整体系统的稳定性。

举几个渠道路由应用的小场景：

- 1. 用户使用招行信用卡支付，支付平台同时对接了网联和银联，而网联和银联都支持招行信用卡，那么就需要渠道路由挑选一个渠道。
- 2. 做实名认证，平台对接了多个实名认证通道，通过渠道路由挑选一个认证渠道。

由上面可以看到，除了支付路由外，还可能有**信息类渠道路由**，比如实名认证类。

那退款有没有路由？显示没有。在银联做的支付，只能去银联退款。特殊的渠道也没有路由，比如用户选择使用支付宝支付，因为支付宝只能在支付宝做支付，所以无需路由。

### 3. 渠道路由的设计原则

渠道路由作为支付系统的核心模块，需要满足以下几个设计原则：

- **灵活性**：路由规则需要支持动态调整。从业务的角度出发，有些场景考虑成本，有些场景考虑成功率，都能方便支持。

- **扩展性**：设计时要考虑到未来可能新增的支付渠道，新增的决策因子，这些都不能在代码中写死，确保系统易于扩展。
- **高可用性**：路由系统本身需要具备高可用性，确保在高并发和故障情况下仍能稳定运行，哪怕内部报错，仍然能返回一条可用的渠道。
- **性能**：在保证准确性的前提下，路由决策需要快速，不能成为支付流程的瓶颈。

## 4. 业界常见的几种路由形态

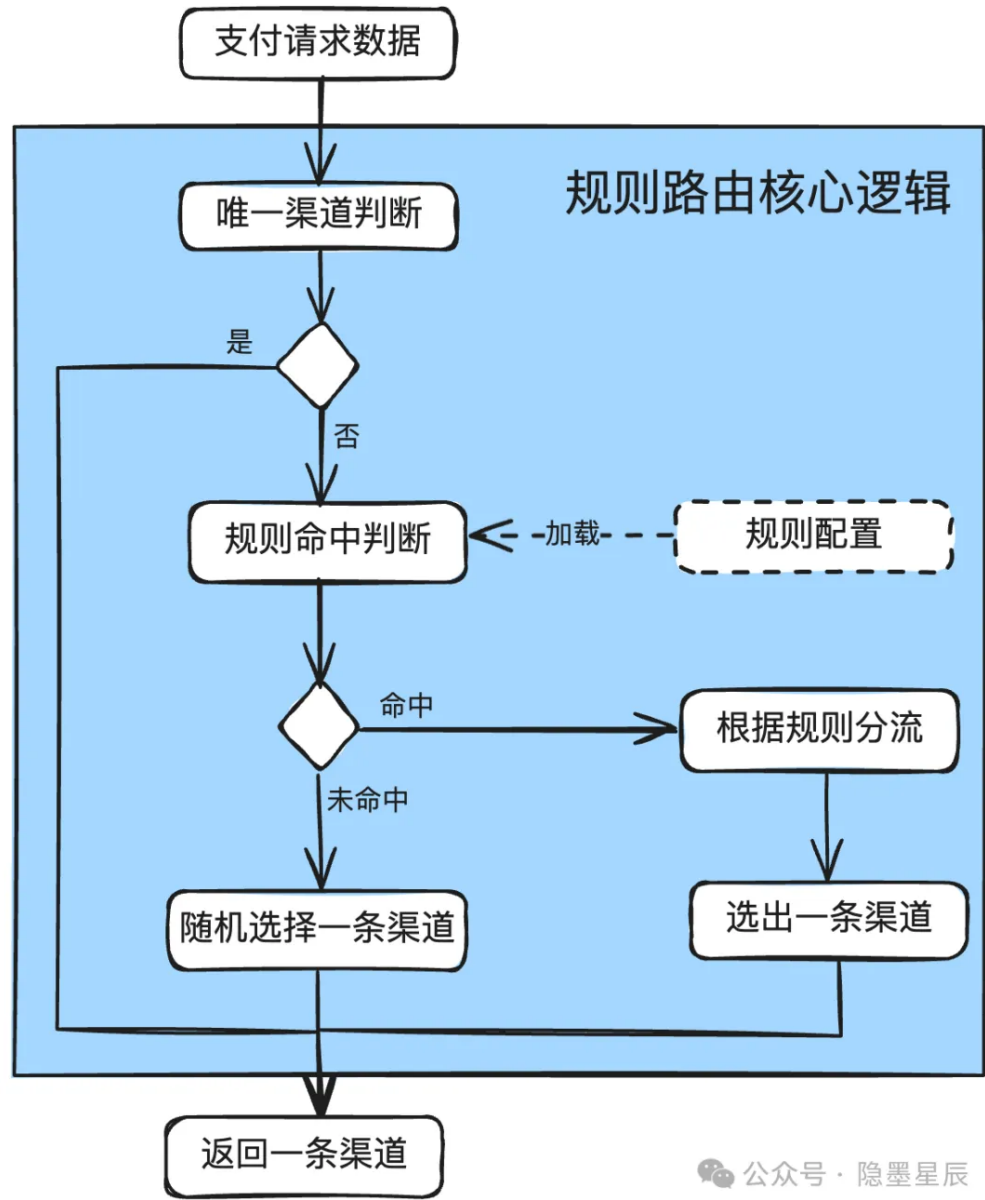
根据业务的需要，通常有以下几种路由形态：

1. **硬编码取第一个**。在项目上线初期，为了赶进度，同时渠道也不多，常常在代码中写死取第一个，先保证支付主流程能走通。
2. **基于规则的路由**。通过预定义的规则提高灵活性和可扩展性。
3. **智能路由**。利用机器学习和大数据分析，根据历史数据和实时状态，智能地选择最佳渠道。

## 5. 一种典型的基于规则的渠道路由设计

基于规则的渠道路由是最常见的设计。简单地说，就是符合什么条件就执行什么样的分流逻辑。比如：支付平台对接了网联和银联，招行信用卡全部走网联，工行信用卡500块以内的走网联，500块以上的走银联。

### 5.1. 核心流程设计



公众号 · 隐墨星辰

说明：

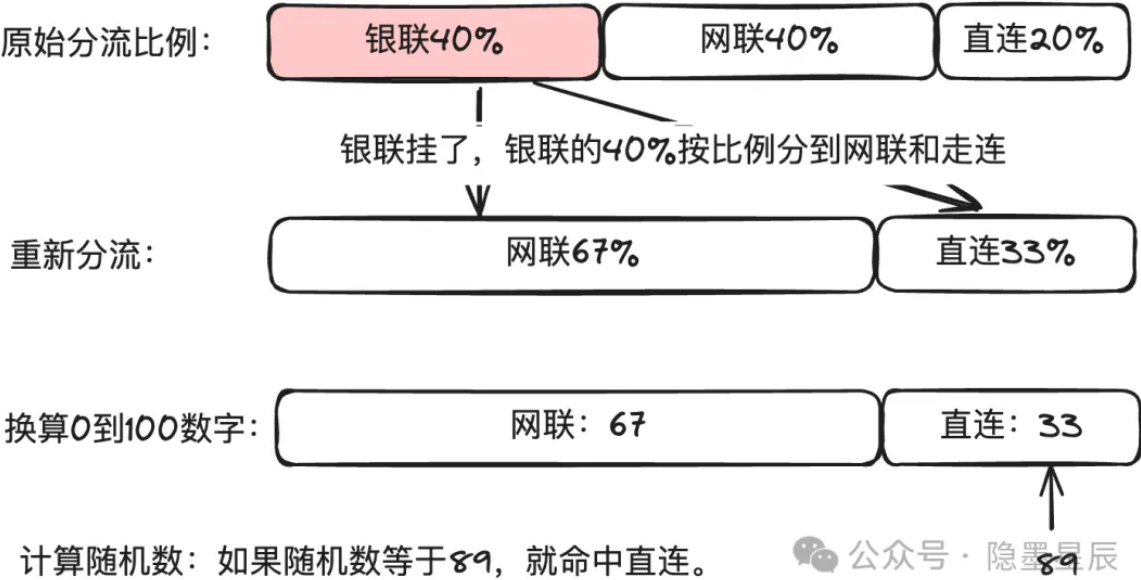
1. 先进行唯一渠道判断，如果只有一条渠道，直接返回。
2. 判断规则，如果规则没有命中，那就从可用渠道中随机挑选一条。
3. 如果命中规则，再根据规则中的分流逻辑进行分流。
4. 最后返回唯一的一条渠道。

## 5.2. 分流算法设计

如果一个请求既可以走银联，也可以走网联，还可以走直连，有以下几种情况：

1. 没有命中任何一条规则，随机选择一条渠道。
2. 有多条规则可以命中，选择优先级最高的。
3. 命中的路由规则里，银联和网联分别是40%和40%，直连20%，根据规则分流。如果当前银联挂了，把银联按比例分配到网联和直连。

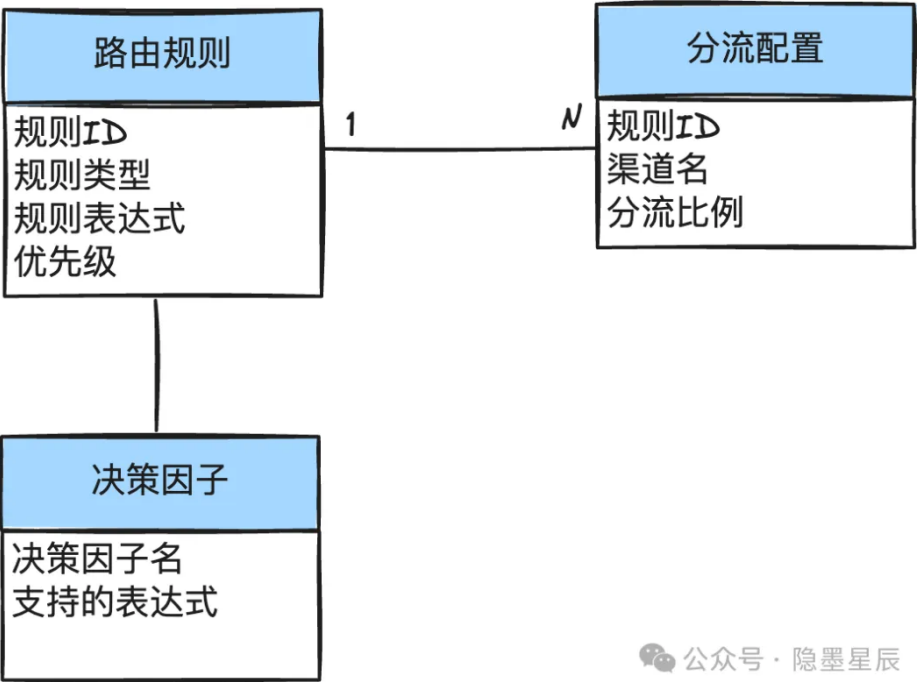
常见的分流算法是先把各渠道的分流比例换算成0-100区间的数字，从大到小排序，再根据用户ID取模、请求单号取模或生成一个随机数，再看这个数落在哪个区间，对应的渠道就是命中的渠道。



伪代码如下：

```
int random = 用户ID取模（或请求单号取模，或生成随机数）；
for (int i = 0; i < 分流集合.size(); i++) {
    if (random % 分流集合[i] <= 0) {
        return 命中的渠道；
    }
}
```

5.3. 路由规则配置模型



说明：

- 1. **路由规则用于规则引擎运算是否命中**。核心字段包括：规则ID、规则类型、规则表达式、优先级。实际实现时可根据各公司内部情况加字段。

- a. 规则ID：用于分流配置做关联；
  - b. 规则类型：用于区分支付、实名认证等。
  - c. 规则表达式：用于规则引擎运算；
  - d. 优先级：用于排序，如果有多个规则都符合，以优先级最高的为准；
2. **分流配置用于规则命中后，如何进行分流。**核心字段包括：规则ID、渠道名、分流比例。
- a. 规则ID：用于与路由规则进行关联。
  - b. 渠道名：表示要分流去的渠道。
  - c. 分流比例：说明有多少流量要分过去。
3. **决策因子定义用于决策的条件。**比如卡BIN，卡品牌，金额等。

5.4. 规则引擎选择

业务的规则引擎有很多，比如大名鼎鼎的Drools等，也可以选择自研，各公司可以根据自己的技术生态来选择。

我个人推荐QIExpress。推荐理由：简单实用。因为路由规则都非常简单，没有过于复杂的运算，不需要引入一些很重的规则引擎。

关于QIExpress的资料，可参考官网介绍。

后面会有QIExpress的规则示例。

5.5. 决策子选择

决策因子就是路由规则匹配的条件，一般有以下几种：

- 1. 金额：比如小于某个金额，或大于某个金额。
- 2. 卡品牌：VISA、MASTER、UPAY等。
- 3. 发卡行：CMB、ICBC等。
- 4. 卡类型：借记卡、信用卡等。
- 5. 卡BIN：某个号段的卡。
- 6. 业务场景字段：各公司自定，比如线下场景，线上场景等。

5.6. 路由规则示例

规则的编写和规则引擎强相关。下面以QIExpress做个示例。实际落地时，需要根据自己的选择的规则引擎做改造。

**假设：**支付平台对接了网联和银联，要求：

- 1) 招行信用卡全部走网联。
- 2) 工行信用卡500块以内（不包含）的40%走网联，60%走银联。
- 3) 工行信用卡500块以上的走银联。

**一些基本的变量定义：**

银行名称: bankName

支付方式: paymentMethod

卡类型: cardType

金额变量: amount

网联: NUCC

银联: UPAY

招行: CMB

工行: ICBC

定义规则:

- 1. 规则1: paymentMethod='card' && cardType='credit' && bankName='CMB'  
分流: NUCC: 100
- 2. 规则 2 : paymentMethod='card' && cardType='credit' && bankName='ICBC'  
and amount < 500.00  
分流: NUCC: 40, UPAY: 60
- 3. 规则 3 : paymentMethod='card' && cardType='credit' && bankName='ICBC'  
and amount >= 500.00  
分流: UPAY: 100

### 5.7. 界面配置示例

下面只是示意一个简单的路由配置。如果是多层次的与和或关系，需要产品经理做一些稍微复杂一点的界面。



规则名称: CARD\_CREDIT\_ICBC\_LESS\_500

规则配置:

支付方式

=

卡

⊖ ⊕

&&

卡类型

=

信用卡

⊖ ⊕

&&

开户行

=

工商银行

⊖ ⊕

&&

金额 (元)

<

500

⊖ ⊕

分流配置:

网联

:

40

%

银联

:

60

%

备注: 工行信用卡小于 (不包含) 500元时的分流规则

取消

保存

公众号 · 隐墨星辰

说明:

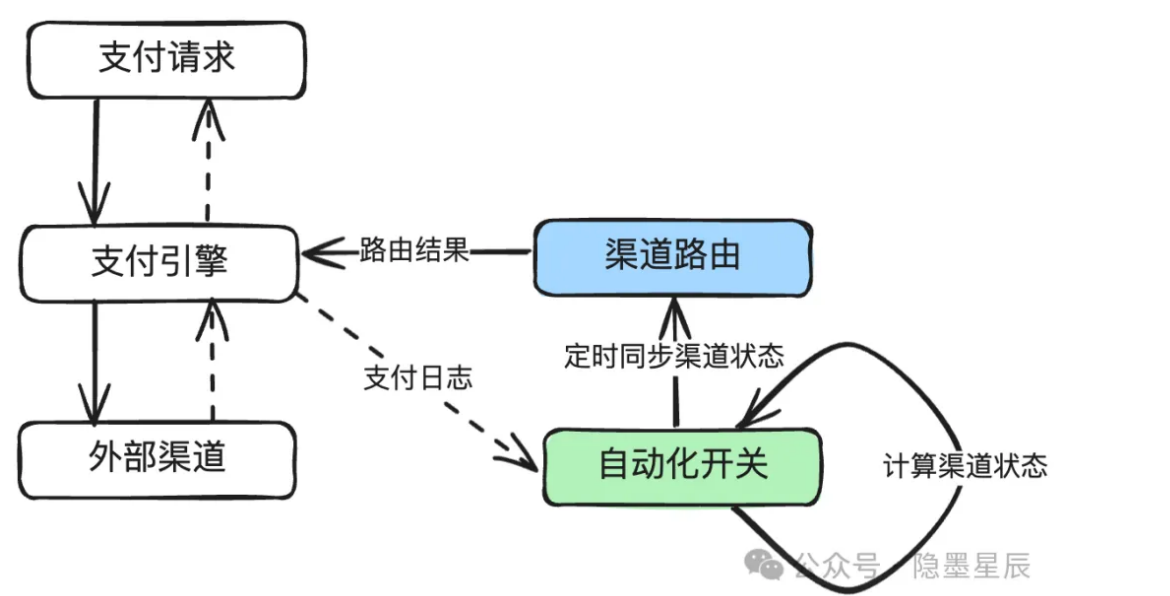
1. 示例规则的业务诉求: 工行信用卡500块以内 (不包含) 的40%走网联, 60%走银联。
2. 后台保存的规则为: "paymentMethod='card' && cardType='credit' && bankName='ICBC' and amount < 500.00", 分流有2两条, 分别是: "NUCC: 40", "UPAY: 60"。
3. 用于决策因子的元数据, 需要提前定义好, 包括这些字段的运算规则 (比如开户行就不能使用大于、等于)。

5.8. 一些调优思路

1. **用户最近支付成功记录优先, 提高成功率。**比如用户可以用银联, 也可以使用网联, 如果5天内最近一次使用了银联是成功的, 那么为了成功率, 可以考虑再次路由到银联去。因为从渠道风控的角度, 已经成功的前提下, 再次成功的可能性更大 (余额不足除外)。
2. **根据用户ID做分流,**而不是当前订单号做分流, 或者完全随机数。也就是使用伪随机。好处是同一个用户更大概率走到同一个渠道, 有利于提高成功率, 进而提升用户体验。
3. **适当使用缓存,**以提高运算速度。

6. 加入自动化开关的渠道路由

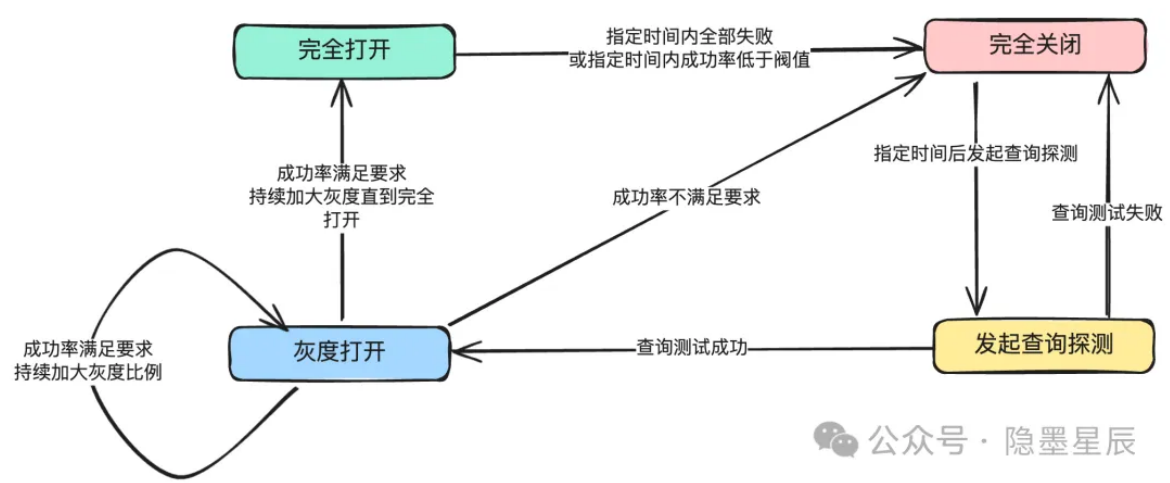
外部渠道服务经常不稳定，通过自动化开关模块监听支付引擎或渠道网关的支付结果消息，实时计算渠道的状态，在渠道出现问题后自动关闭，并推送给渠道路由。



说明：

1. 由自动化开关模块监听支付引擎的支付结果消息，每秒计算一次渠道的状态，在渠道出现问题后，自动关闭，并把结果推送给渠道路由。
2. 渠道关闭后，再发起探测服务，探测成功后，灰度打开渠道。

下图是渠道自动化渠道开关的示意图。有多种技术手段可以实现，基本原理就是基于滑动时间窗口算法来做，具体落地可以使用时序数据库，或者自己通过redis实现。



说明：

1. 初始是完成打开。
2. 指定时间内全部失败或指定时间内成功率低于阈值，关闭渠道。
3. 指定时间后，发起查询，如果查询渠道失败，持续关闭。
4. 如果查询成功，就灰度打开，如果灰度打开后的成功率不满足要求，就继续关闭。
5. 如果灰度打开后的成功率满足要求，就持续加大灰度，直到完成打开。

后面会单独起一篇文章来讲自动化渠道开关的设计与实现。

## 7. 高阶的智能路由

一些有实力的公司，通过算法和机器学习来做智能路由。所谓智能路由，就是不仅是根据路由规则来计算路由，而是根据当前的请求参数和渠道数据，综合成功率、成本、用户使用习惯、地域等多因子计算出最优的一条渠道。

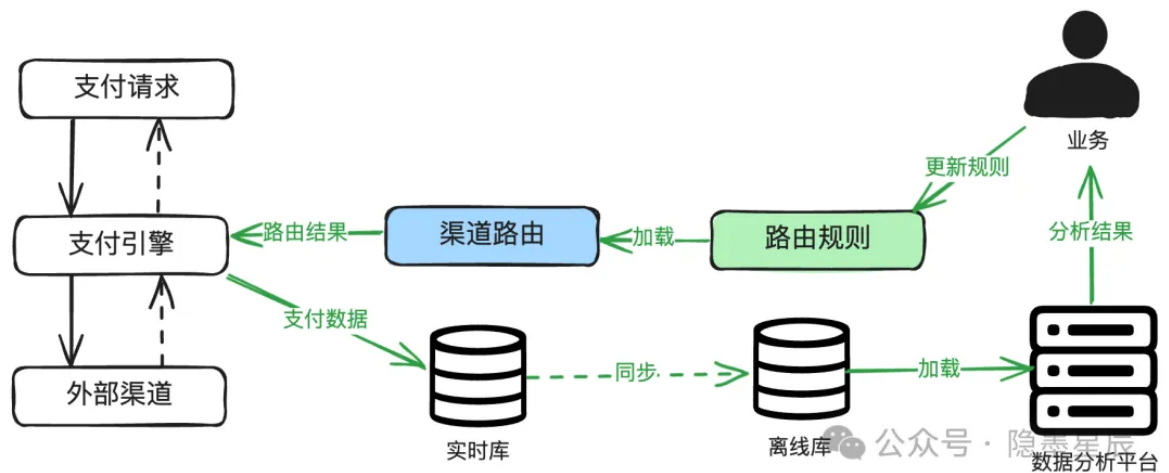
这个方案有几个难题不好解决。

首先是公司实力足够强。有人才来做算法，且这些算法同学需要懂一点业务；

其次是经验不好总结。比如成功率提升了2%，是因为什么原因提升了？有一些不可解释性。

最后业务无法直接操作调优。比如有些场景下业务希望保成功率，有些场景下业务希望保较低的成本，智能路由如何调参？

我个人更倾向于【规则路由 + 离线数据分析】的组合。其中离线数据分析平台可以引入一些算法来分析各因子对成功率的影响，供业务人员决策，并调整路由规则。



说明：

1. 通过分析数据，找到影响成功率成本的因子。
2. 更新路由规则。
3. 重复第1步。

## 8. 结束语

渠道路由在现代支付系统中扮演着至关重要的角色，一个高效、灵活的渠道路由设计能够显著提升支付成功率，优化成本，并改善用户体验。通过本文的介绍，希望大家在实际项目中设计和实现渠道路由提供一些有益的参考。

这是《百图解码支付系统设计与实现》专栏系列文章中的第（25）篇。欢迎和我一起深入解码支付系统的方方面面。

## 精选

《百图解码支付系统设计与实现》电子书\_V20240525

**专栏地址：**[百图解码支付系统设计与实现](#)

**领域相关（部分）：**

[支付行业黑话：支付系统必知术语一网打尽](#)

[跟着图走，学支付：在线支付系统设计的图解教程](#)

[图解收单平台：打造商户收款的高效之道](#)

[图解结算平台：准确高效给商户结款](#)

[图解收银台：支付系统承上启下的关键应用](#)

[图解支付引擎：资产流动的枢纽](#)

[图解渠道网关：不只是对接渠道的接口（一）](#)

[源码讲解：JAVA版低代码报文网关-通过简单配置快速对接银行渠道](#)

**技术专题（部分）：**

[交易流水号的艺术：掌握支付系统的业务ID生成指南](#)

[支付系统的核心：简洁而精妙的状态机设计与核心代码实现](#)

[揭秘支付安全：为什么你的交易无法被篡改](#)

[金融密语：揭秘支付系统的加解密艺术](#)

[支付系统日志设计完全指南：构建高效监控和问题排查体系的关键基石](#)

[避免重复扣款：分布式支付系统的幂等性原理与实践](#)

[一文搞懂支付安全](#)

[用极简的工具画超帅的架构图](#)

[资损防控：搞定交易系统中金额处理规范](#)

专栏系列文章PDF合集，以及电子书《图解支付系统设计与实现》（已写9万字左右）不定期更新，欢迎关注我的公众号获取：隐墨星辰。

有个小群不定时解答一些问题或知识点，有兴趣的同学可先加微信（yinmon\_sc）后进入，加微信请备注：666。

上一篇 · 《图解支付系统设计与实现》电子书\_V20240525

喜欢此内容的人还喜欢

一文搞懂支付安全

隐墨星辰



征信上的账户数怎么消除，太多网贷记录不好看？

星岩



不看征信的借钱口子非常少！征信不佳，可以适当参考这几个借钱口子！

花户小助手

