

D04 钱包与账务域 · 设计文档

本域总览见《账号计费中心 · 结构大纲》。本文为该域完整设计：定位与边界 / 角色与依赖 / 数据模型 / 核心流程 / 接口契约 / 关键约束与验收标准。

1. 定位与边界

定位：全仓资金真相的唯一持有者。回答三个问题——「钱包里有多少钱」（现金余额与可用额度）、「每一分钱从哪来、到哪去」（只增流水）、「这个客户还能不能继续用」（欠款与欠费派生标记）。

核心概念

概念	承载	语义
现金余额	customers.cash_balance	元、6 位小数，可为负；负值即欠款，充值按符号算术自然清偿
赠金账本	customer_grants	授权记录而非余额计数器：剩余由流水聚合派生，不原地扣减
信用额度	customers.credit_limit	准入缓冲（可用额度的下限），默认 0；不拦截单笔扣费
可用额度	派生量 available	唯一公式 = 现金 + Σ (在营未过期赠金剩余) + 信用额度；准入真相
欠款	现金余额为负	唯一未偿债务真相
欠费标记	customers.overdue_since	派生标记（报表与清单用），非准入真相
资金流水	customer_transactions	账务真相源，只增不改

关键口径

- 金额口径：一律元、6 位小数、字符串传输，链路中不出现二进制浮点中间量。金额只入账不计价——应付金额由计量方按自身计价规则算出后上报，本域不按定价规则二次换算。
- 记账即事实：流水是真相源，余额是同事务缓存列；两者偏离由钱包对账暴露并告警，不靠肉眼核账。
- 扣费是钱包唯一入口：赠金优先、现金兜底的分配只在一处实现；不提供第二条无锁扣减原语。
- 透支是正常态：扣费不因余额不足拒付；熔断收敛在准入层（读侧契约在账户状态与准入域，本域只保证它读到正确的可用额度）。
- 欠款判定只认现金：未偿债务的唯一真相是现金余额为负；不读派生标记，不读准入额度。

边界

- 不管计价：不依赖定价规则做换算，不运行时拉取计价规则。
- 不管订单与渠道：下单、支付回调、渠道凭据、订单状态机在订购与支付域；本域只承载其入账与冲减的资金后果。
- 不管准入执行：状态键读侧判定与数据面拦截属账户状态与准入域。
- 不管退款审批与渠道退款对账（退款与对账域）、不管开票（发票域）、不管用量计量（计量与计费域）。

归属口径

- 运营面的客户账务端点（状态标记 / 信用额度 / 调整 / 宽限）语义细节在本域；运营面端点统一索引登记在 D11，不在两处重复展开。
- 组的生效限速读侧计算（档位默认值与降档减半）在本域；组资源与组额度配置属租户与组织域。

2. 角色与依赖

能力矩阵

能力	计费路由（router）与计量方	数据面网关	运营面	门户用户
上报批量扣费	✓（幂等键 = 事件标识）	✗	✗	✗
读余额与可用额度	✓	✓（经状态键）	✓	✓（本人钱包）
赠金入账	✗	✗	✓	✗
信用额度配置	✗	✗	✓	✗
调整流水（核销 / 修正）	✗	✗	✓	✗
宽限期设置	✗	✗	✓	✗
状态标记（正常 / 封禁）	✗	✗	✓	✗
按 ref 集合拉流水做同源比对	✓	✗	✓	✗（门户只读展示详见 D12）

- 计量链令牌与运营链令牌严格互斥（计量链不能赠金，运营链不能扣费），且均不经公网入口到达。
- 运营面写操作另行携带操作人标识，与业务变更同事务落审计。

上游依赖

上游	依赖内容	语义
订购与支付域	充值入账、退款冲减	现金增减的事实来源；以订单号为入账幂等键
计量与计费域	应付金额与用量	本域只入账不计价；金额随扣费请求上报
租户与组织域	工作区	钱包与工作区一一对应，是资金归属主体
账号与身份域	门户身份与令牌	本人钱包查询的身份来源
通知与触达域	站内信与告警投递	余额告警、欠费告知的送达（投递本身不在本域）

下游被引用

引用方	引用内容	语义
账户状态与准入域	现金余额、赠金剩余、信用额度、宽限截止	派生准入状态与状态键载荷的唯一账务来源
计量与计费域	按 ref 集合的流水只读通道	跨服务同源精确比对的数据来源
退款与对账域	现金扣费行、调整行、剩余可退额度	退款额度与「支付后消费」画像依赖支出类流水

引用方	引用内容	语义
发票域	账期费用构成	开票金额口径来自流水按类型聚合
风控与合规域	欠款账龄、封禁与宽限状态	分级提醒 / 催收 / 冻结的输入
运营管理	客户台账投影与审计	运营面只读展示与留痕索引

3. 数据模型

3.1 域内关联总览

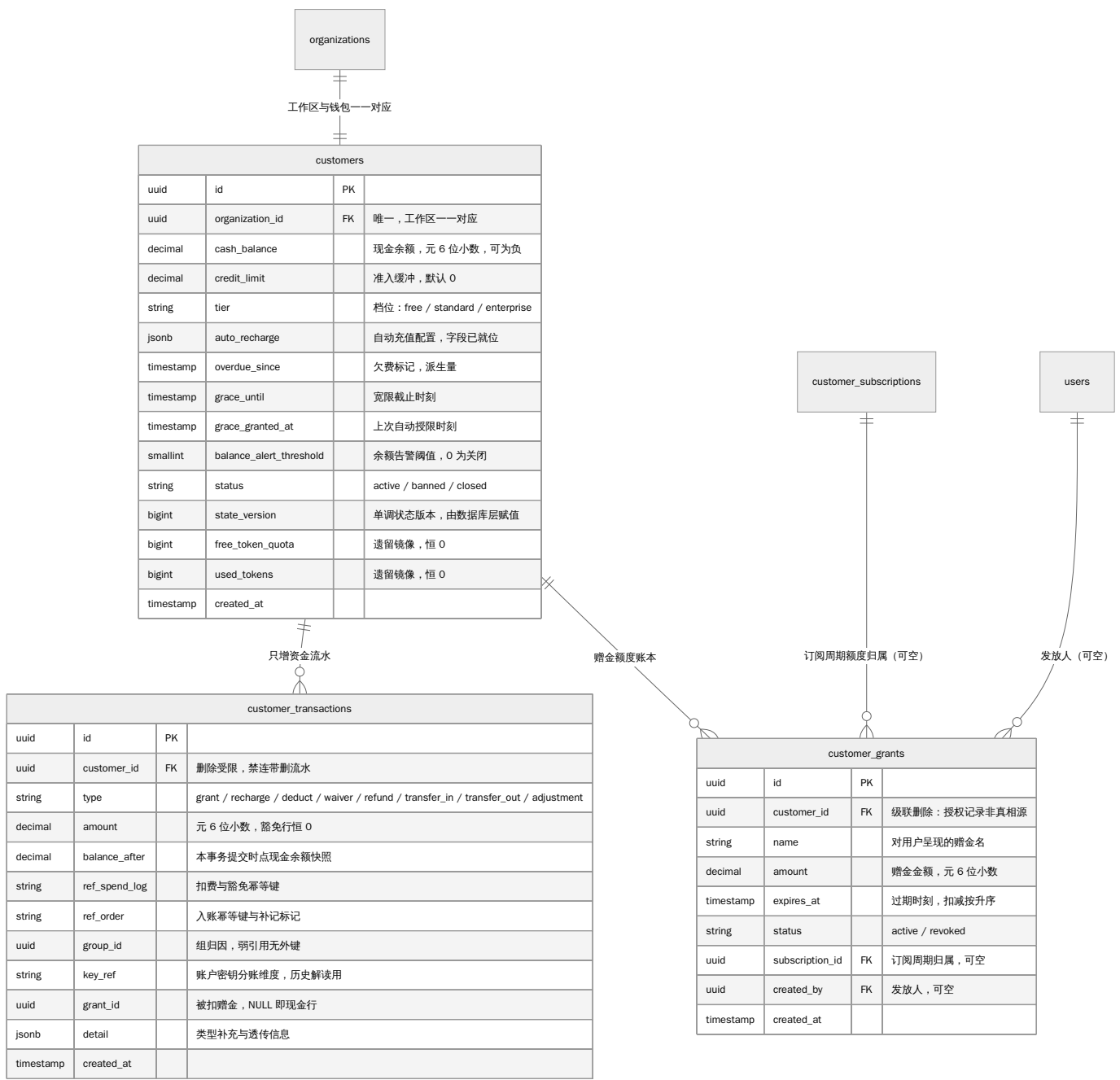


图 1

赠金不落余额列：剩余随时可用聚合派生（`amount` 减去被本赠金抵扣的扣费行金额之和），故不存在「计数器与流水不一致」这一类偏差。

3.2 域间引用

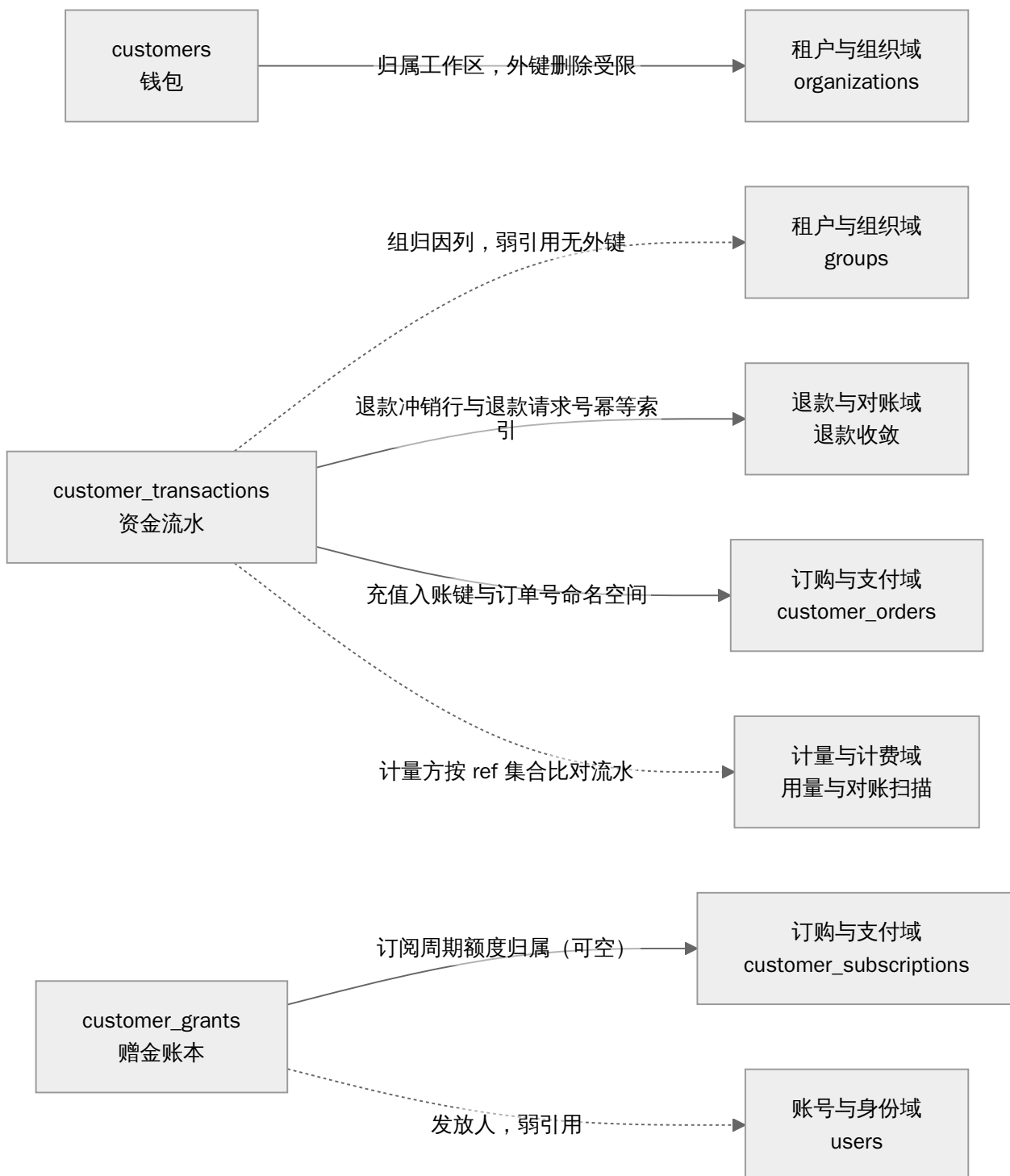


图 2

- 流水对组、对订单只落弱引用列：不建跨表外键，避免删除约束拖住高频写入路径。
- `users` 由本域弱引用（赠金发放人），不入本域表清单。

3.3 表清单（3 表）

表	职责	关键字段
customers	钱包主体与准入账务位，与工作区一一对应	organization_id / cash_balance / credit_limit / tier / overdue_since / grace_until / grace_granted_at / balance_alert_threshold / status / state_version / auto_recharge
customer_transactions	只增资金流水，全部资金事实的唯一真相源	customer_id / type / amount / balance_after / ref_spend_log / ref_order / group_id / grant_id / key_ref / detail / created_at
customer_grants	赠金额度授权账本，剩余由流水聚合	customer_id / name / amount / expires_at / status / subscription_id / created_by

3.4 设计约束

- 1. 金额精度：全部金额列为元、6 位小数；入参在请求体层限制到 6 位小数，写入前再归一一次（四舍五入兜底）；传输一律字符串。
- 2. 流水只增：行级触发器拒绝更新与删除，另挂语句级触发器拒绝整表清空——整表清空不触发行级触发器，两条缺一不可；违规一律按完整性约束冲突码拒绝。
- 3. 受控旁路须角色门槛：改写与清空的旁路开关须当前角色属于维护角色成员，判定用当前角色而非会话角色（否则临时切换角色即可借用身份）；角色缺失时一律拒绝（fail-closed），超级用户显式放行（否则连受控维护都无从执行）——故部署须确认服务账号非超级用户。
- 4. 扣费幂等键唯一：部分唯一索引按「客户 + 扣费幂等键 + 赠金归属」三元组建唯一性，且空值视为相同；同一次扣费每个赠金包至多一行、现金行至多一行，天然互异不误伤拆分。
- 5. 入账幂等键唯一：部分唯一索引按「客户 + 类型 + 入账引用」建唯一性，仅覆盖充值与非空引用的赠金两型——一次余额转让会落两行同引用（赠金资产对与现金资产对），纳入即误伤；调整的引用无「同引用即重复」语义。
- 6. 订单号命名空间跨类型互斥：另一部分唯一索引按「客户 + 入账引用」建唯一性且不含类型，收窄到订单号前缀，使同一笔订单号在同客户下至多落一行入账流水；其冲突语义是命名空间污染的脏写入，必须显式失败，不收敛为幂等。
- 7. 豁免留痕幂等：部分唯一索引按「客户 + 扣费幂等键」仅覆盖豁免行；豁免与扣费共享同一幂等键命名空间，同一事件只会是其一。
- 8. 钱包锁域唯一：同客户的钱包变更（建钱包 / 扣费 / 赠金入账 / 调整 / 补记）共用按客户标识的同一锁域；跨双钱包场景（余额转让）按标识升序加锁防死锁。
- 9. 归属与删除语义：钱包对工作区唯一（同一工作区至多一个钱包）；流水对钱包为受限删除（禁止物理删钱包连带删流水），赠金对钱包为级联删除；流水对组与订单只落弱引用列。
- 10. 状态值与版本：客户经营状态仅取 active / banned / closed 三值，closed 为终态、无退出路径；客户级单调状态版本由数据库层触发器在行锁内取全局序列赋值（版本序 = 提交序），应用层只读不写。
- 11. 赠金有效期非空：expires_at 为必填；过期是时间驱动的被动衰减，不落任何事务，故不以它作为债务或准入的判据。

4. 核心流程

4.1 批量扣费（钱包唯一入口）

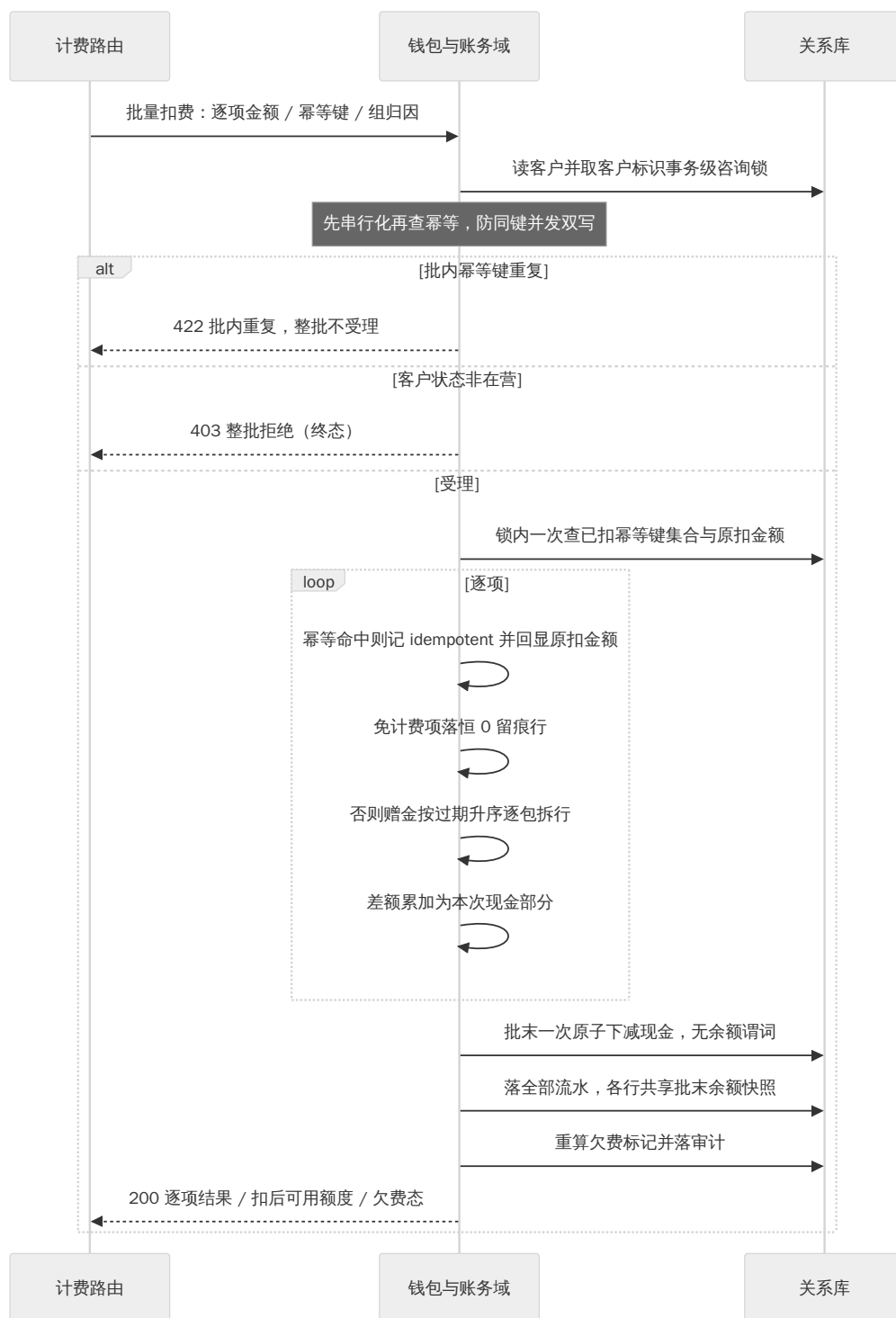


图 3

- 单批上限 200 项；幂等键长度 8~128；金额下界 0、精度 6 位。
- 扣减序固定为：状态门禁 → 赠金（过期升序、逐包拆行）→ 现金（差额无条件下减，可为负）。
- 整批同事务，全成或全不成；幂等命中项按 **idempotent** 回显、不回扣。
- 幂等预查同时命中豁免行——同一事件无论当初上报为豁免还是扣费，重放都返回幂等，不重复留痕也不重复扣费。

4.2 免计费豁免

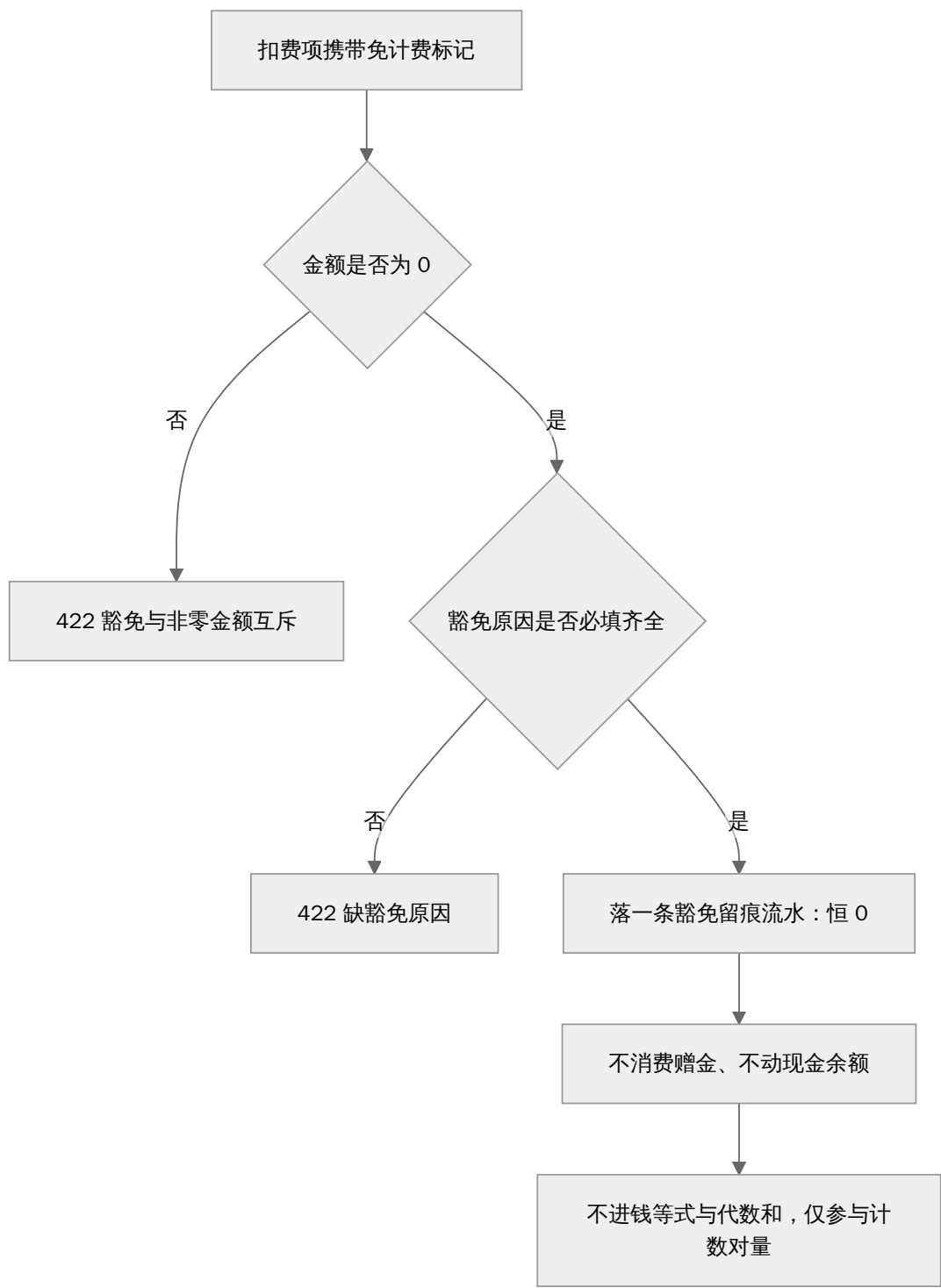


图 4

- 豁免原因是受限枚举（运营促销 / 内部调用 / 未定价），非枚举值即拒。
- 裸零（未标豁免）一律拒绝：防「应免未标」静默空扣——账上既无扣费也无留痕。

4.3 单笔扣费兼容入口

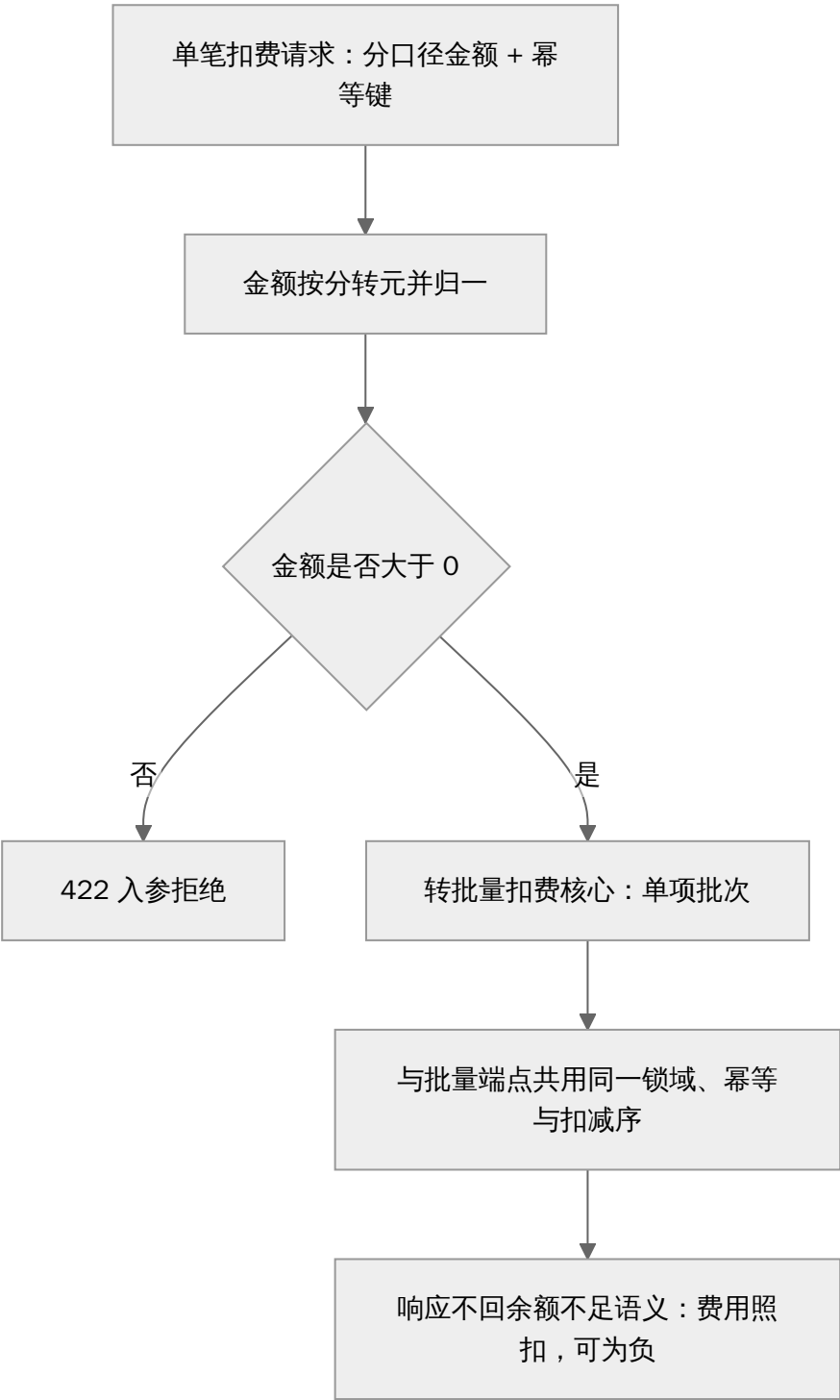


图 5

- 该入口为存量调用方的兼容形态，不新增独立扣减逻辑；新增扣费能力一律扩展批量端点。
- 响应字段与批量端点对齐（是否幂等 / 实扣金额 / 扣后可用额度 / 欠费态 / 状态）。

4.4 赠金入账（运营赠予）

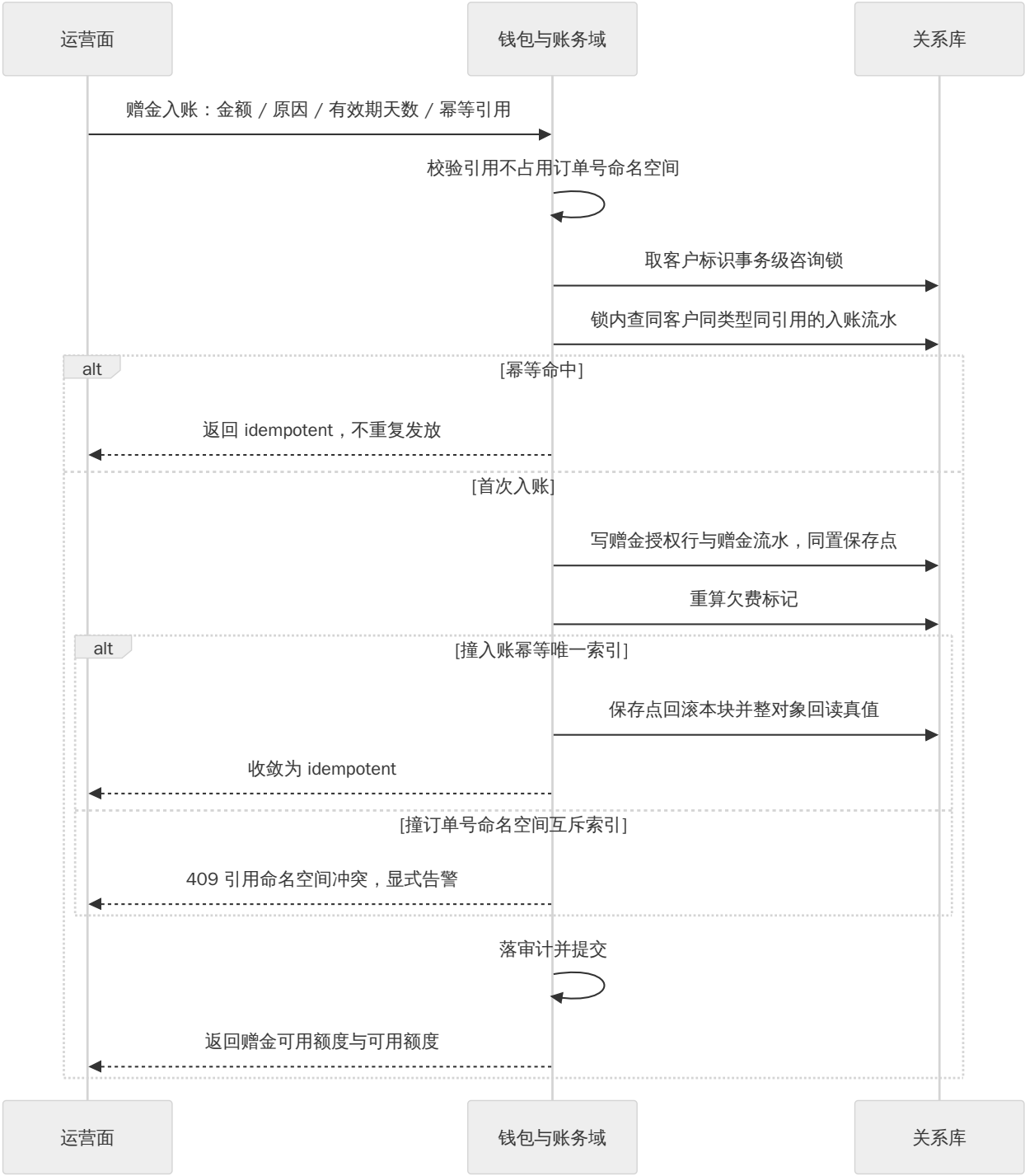


图 6

- 有效期缺省取运营赠金默认时长；幂等引用不得以订单号前缀开头，违者入参即拒。
- 赠金提升可用额度（可恢复服务）但不抵扣现金负债：解封与清债是两个概念。
- 幂等只认同类型入账流水：否则一条同引用的充值流水会让赠金静默不发（运营以为已发、实际没有）。

4.5 建钱包与注册赠金

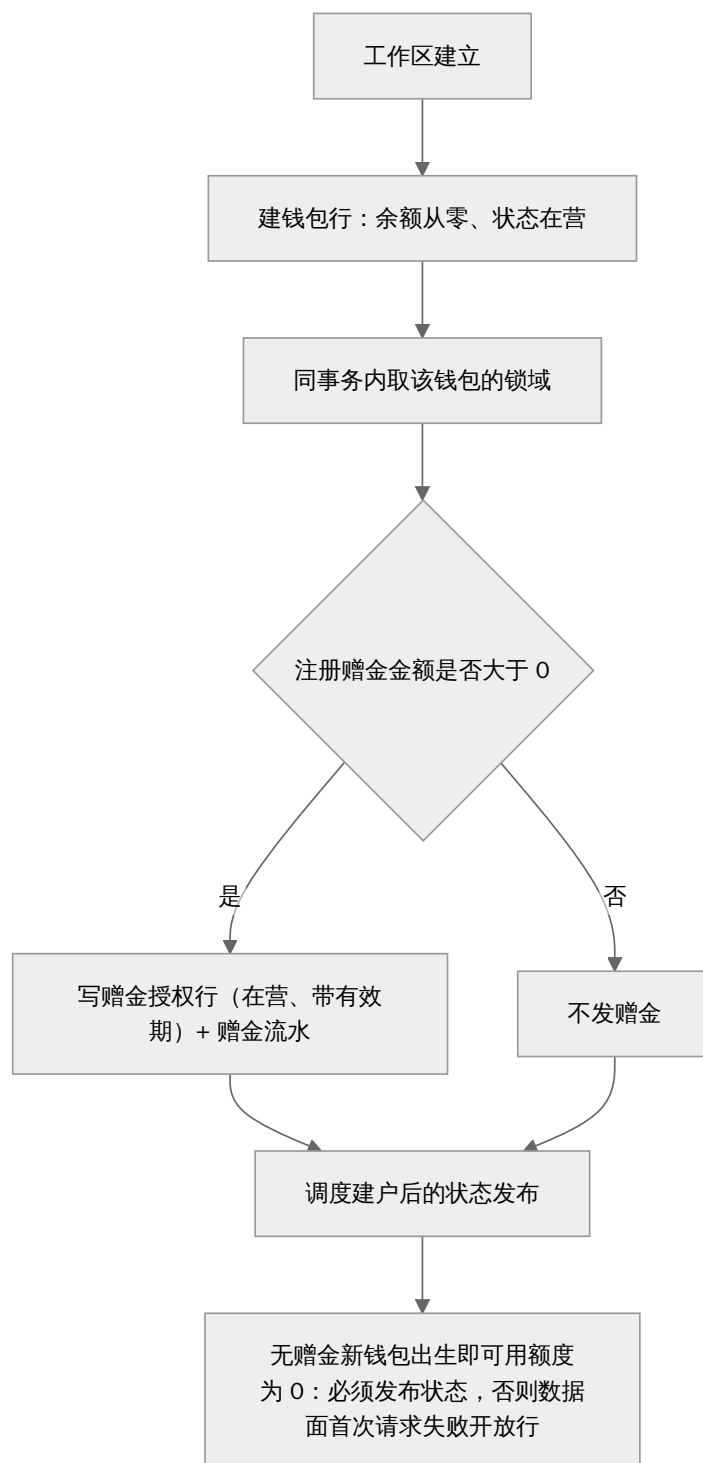


图 7

- 建钱包即取锁域：并发扣费阻塞至建户事务提交（随后能读到赠金）或回滚，杜绝「新钱包的赠金被跳过、费用全额落现金」。
- 锁的持有者是创建方而非调用方——调用方容易漏传或绕过，创建者是唯一汇聚点。

4.6 欠费标记的单一置位与清除

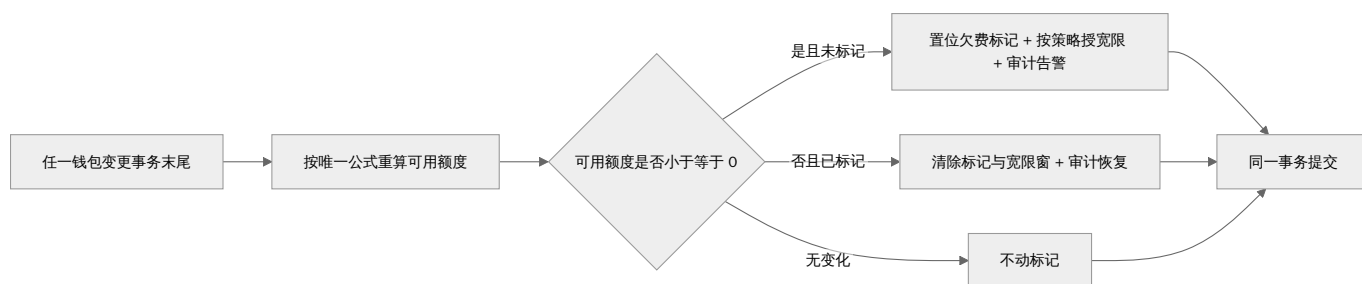


图 8

- 变化路径覆盖扣费、充值、赠金入账、调整、转让、补记；只有这一处置位或清除。
- 已废除两条旧路径：「任何扣费成功即清标记」（在途小额扣费会错误解封仍欠费客户）与「入账无条件清标记」（不足额充值会错误解封）。
- 赠金过期是时间驱动的被动衰减，不落事务：标记滞后到下一次钱包变更或查询，但准入与生命周期出口校验都不读该标记，故滞后不产生误判。

4.7 宽限窗与状态派生

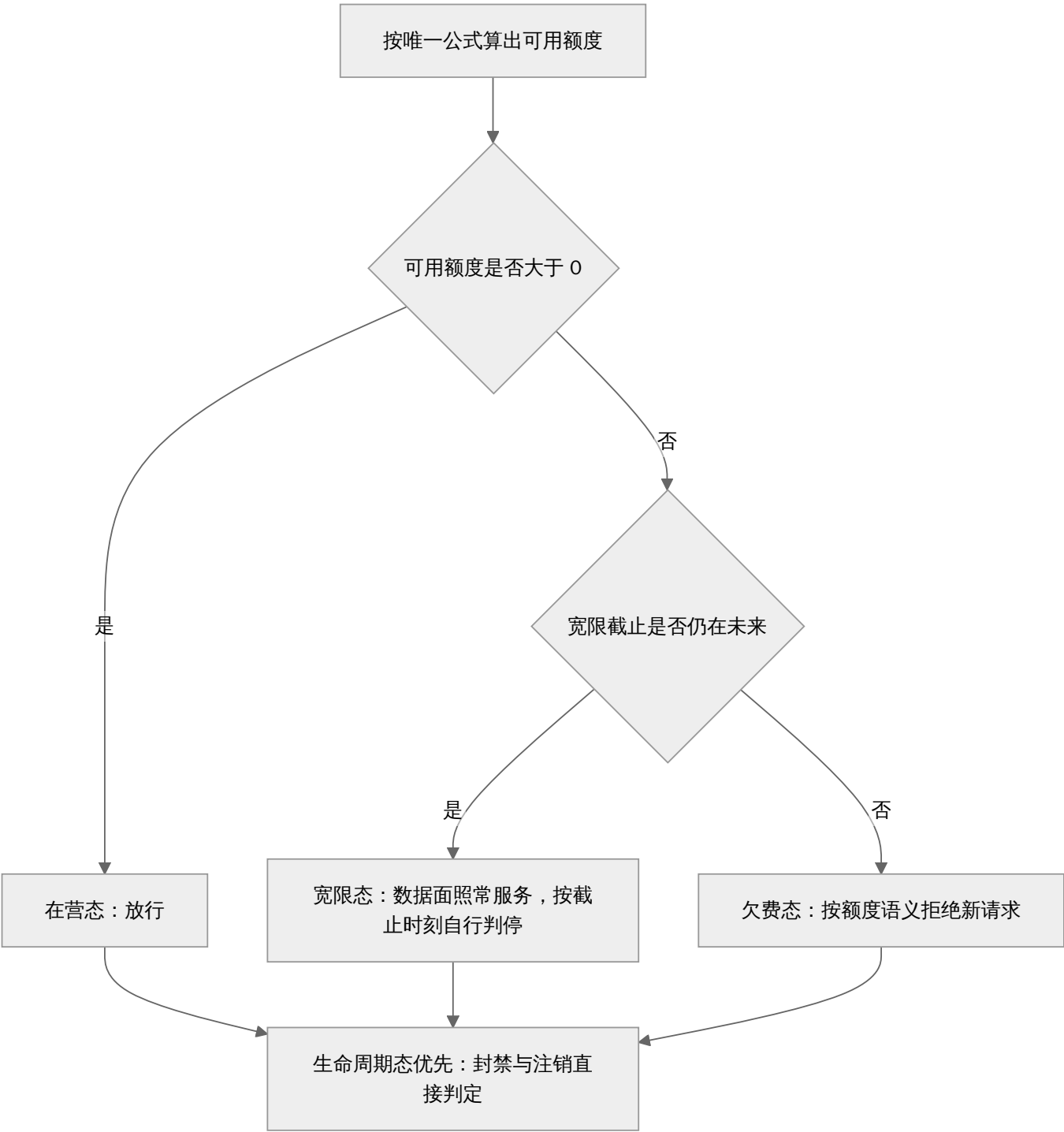


图 9

- 宽限默认关闭（时长配置为 0）：关闭时宽限态不出现，行为与无宽限一致。
- 开启后进入欠费即按策略自动授信，且设冷却期（默认 30 天）防「最小充值循环刷宽限」；运营预设的宽限优先，不被自动授信覆盖。
- 恢复（可用额度转正）即清宽限窗，下次欠费按规则重新授信。
- 状态键的读侧判定与票面语义属账户状态与准入域，详见 D06。

4.8 生命周期出口的债务拦截

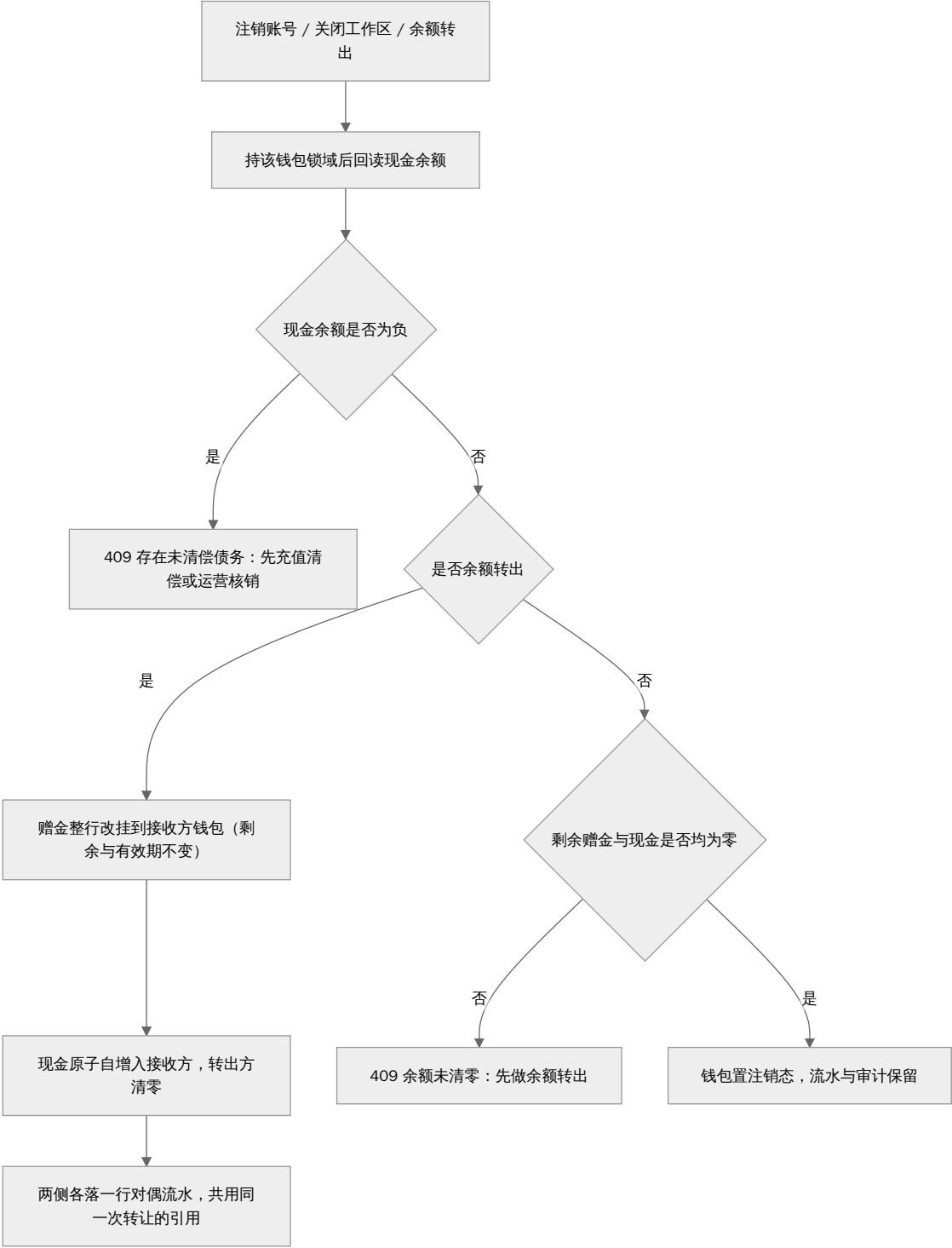


图 10

- 债务判据只有一条：现金余额为负。明确不读残留欠费标记（派生量、只在钱包变更事务末尾重算，赠金过期纯被动漂移会让它滞后，「现金为正、只是赠金刚过期」的有钱客户会被误拦为欠债）；也明确不读实时可用额度（可用额度是准入额度而非未偿债务，「现金为零、无赠金、无信用」的干净客户可用额度为零，按小于等于零判定会被误拦；平台授予的消费额度耗尽不构成债务）。
- 堵死的逃逸口是「刷到负余额再注销让债务蒸发」；充值清偿即现金回正，放行。
- 对偶流水是必须的：流水是资金真相源，转出方资产不能凭空消失，迁出与迁入两侧都要各自可解释。

4.9 钱包对账与余额补记

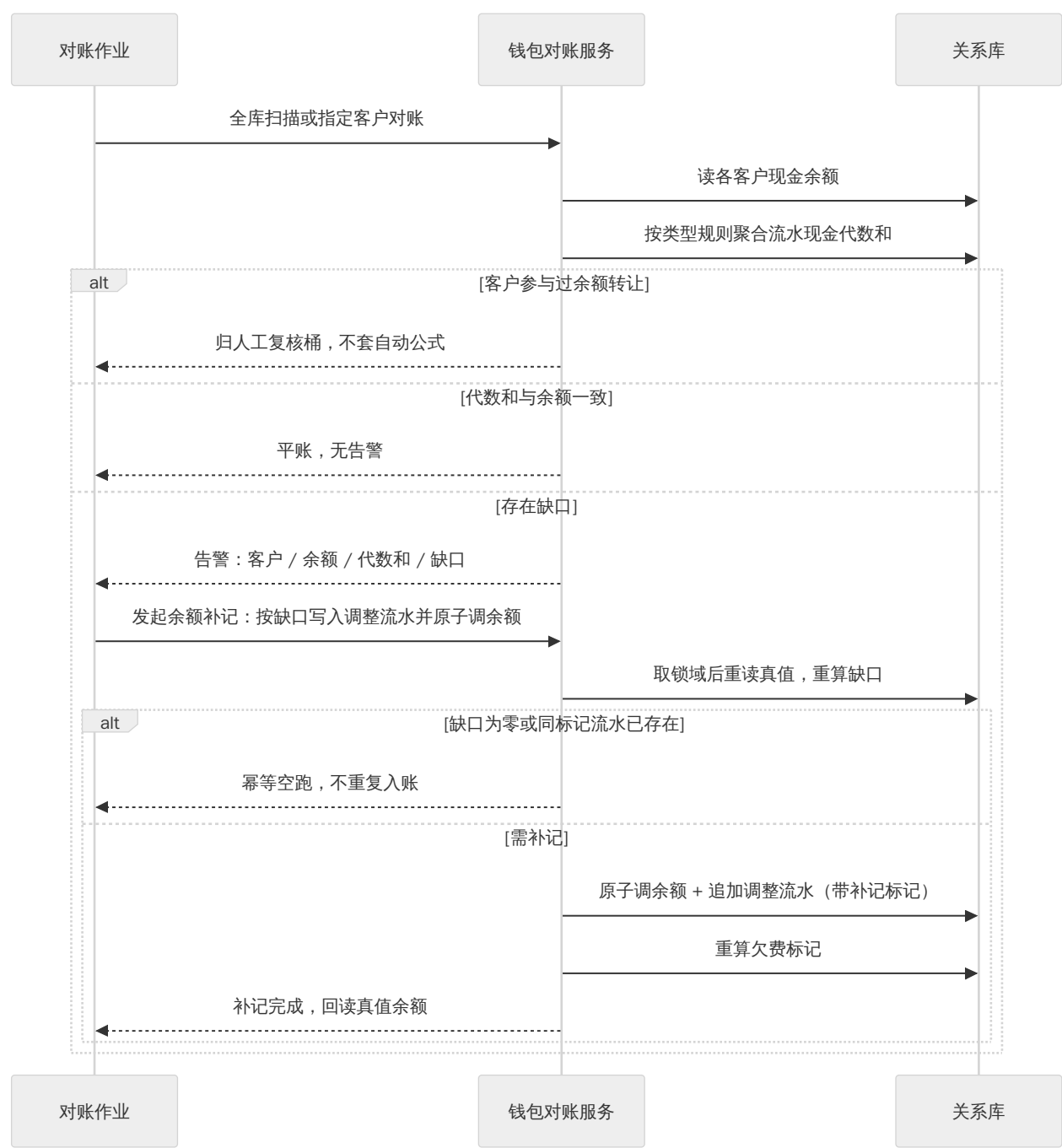


图 11

- 对账口径：现金代数和 = 充值 + 调整 + 退款 + 转入 - 现金扣费行 - 转出；赠金抵扣行（被扣赠金非空）与补记自身行都不进代数和（补记是把缓存拉回代数之和的修正动作，计入即二次账不平）。
- 显式指定补记金额时必须与缺口一致，防错补造成二次账不平；缺口为零或同标记流水已存在即空跑。
- 参与过余额转让的客户拒绝自动补记（对偶行同类型同引用、仅金额不同，自动公式无法区分），由人工核转让记录；注销态钱包拒绝补记。

4.10 档位与生效限速

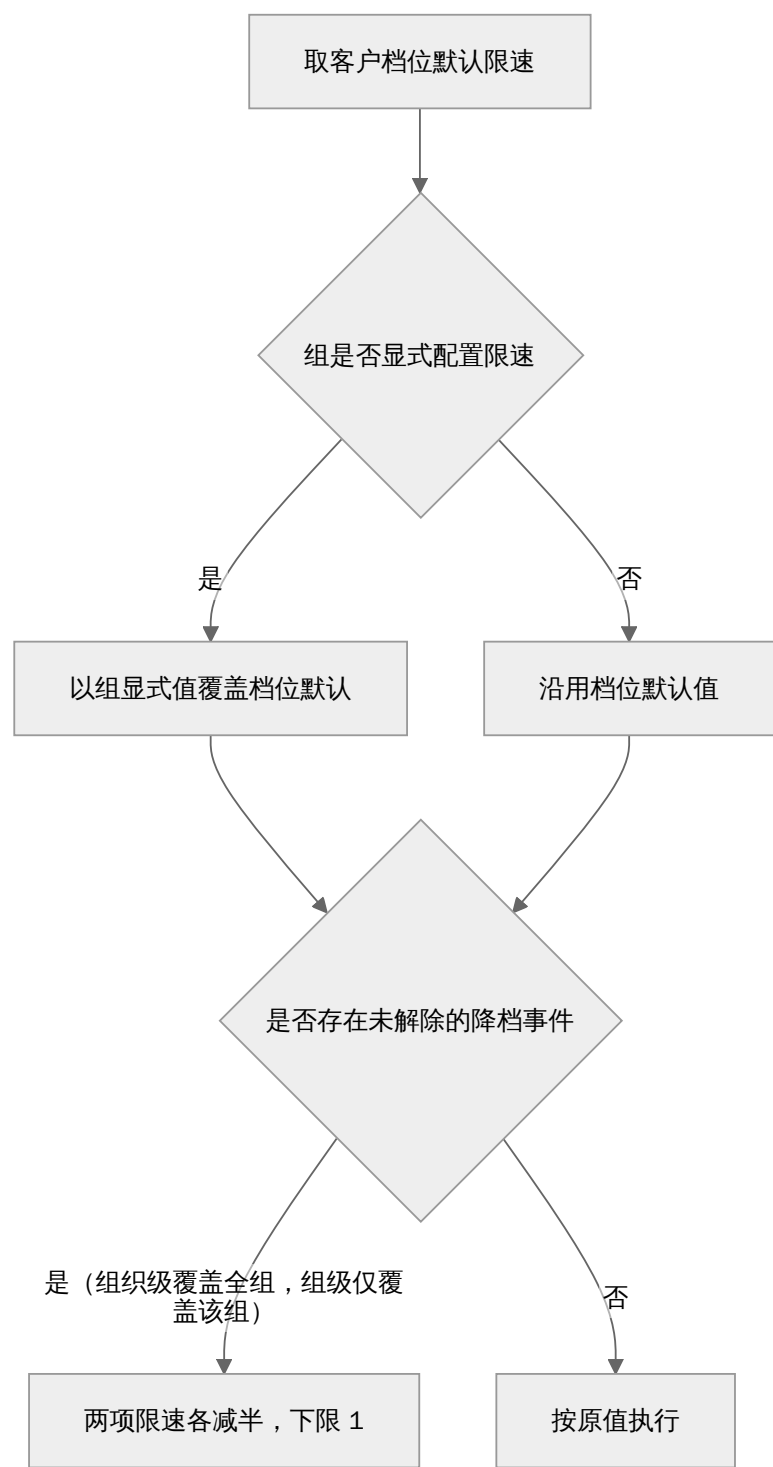


图 12

- 档位取值：`free`（评估档，新注册默认，限速从严）、`standard`（完成首笔支付或转为企业工作区后自动升档）、`enterprise`（预留，仅供手工设置）。
- 取值为「组显式值优先，否则档位默认」；未知或空档位按评估档兜底。降档期间两项限速各除以 2（下限 1）。
- 档位与信用额度是两个正交旋钮：档位决定默认限速，信用额度是准入缓冲（默认 0），二者互不推导。

5. 接口契约

5.1 门户钱包（`/api/account`）

方法	路径	语义
GET	<code>/wallet</code>	本人钱包：现金余额（可为负，负值即欠款）、赠金额度与明细、信用额度、可用额度、派生态与宽限截止、支付模式标记

- 只读展示，不做任何钱包变更；充值入账走订购与支付域的下单与回调路径（详见 D03）。
- 金额一律元字符串；状态与可用额度取派生态唯一来源，端点不自算。

5.2 计费对接（`/internal`）

方法	路径	语义
POST	<code>/internal/customers/{customer_id}/deduct-batch</code>	批量扣费（1~200 项）：幂等键去重、赠金优先、现金兜底；计量链令牌
POST	<code>/internal/customers/{customer_id}/deduct</code>	单笔扣费兼容入口（分口径），内联同一核心；计量链令牌
GET	<code>/internal/customers/{customer_id}/balance</code>	余额全量：现金 + 信用额度 + 赠金明细 + 可用额度（准入真相）；运营链令牌
POST	<code>/internal/customers/{customer_id}/credit</code>	赠金入账（运营赠予），按引用幂等；运营链令牌
GET	<code>/internal/transactions</code>	按日 / 类型 / 客户 / 组 / ref 集合拉流水（跨服务同源比对数据通道）：返回全条数、截断标记与全集金额和；计量链令牌

- 扣费响应含逐项结果（幂等标记与实扣金额）、扣后可用额度、欠费态与状态；状态非在营时整批拒绝。
- 流水查询按中国标准时间归日（含结束日当日）；至少传日期 / 客户 / ref 集合之一，防全表拉取；ref 集合上限 500。
- 轻量准入状态端点归账户状态与准入域（详见 D06），此处不重复展开。

5.3 运营客户台账（`/internal`，运营链令牌）

方法	路径	语义
GET	<code>/internal/customers</code>	客户列表：按工作区与负责人投影，支持按档位 / 经营状态 / 欠费筛选与关键词搜索，附余额与可用额度派生值
GET	<code>/internal/customers/{customer_id}</code>	客户详情：钱包账务位 + 赠金明细 + 最近流水 + 用量摘要 + 成员与组摘要
PATCH	<code>/internal/customers/{customer_id}/status</code>	状态标记（正常 ↔ 封禁）：欠费是自动派生态不可手工设置，注销态为终态不可改写

5.4 运营面客户账务 (/api/account/ops)

方法	路径	语义
PATCH	/api/account/ops/customers/{customer_id}/status	状态标记（同上语义，运营面入口，落审计并同步发布状态）
PATCH	/api/account/ops/customers/{customer_id}/credit-limit	信用额度配置：即时影响可用额度（提高可解封欠费客户，下调收紧准入）；不触碰现金与赠金
POST	/api/account/ops/customers/{customer_id}/adjustment	运营调整：坏账核销（正）/ 手工修正（正负）：必带原因，同事务落调整流水并重算欠费标记
PUT	/api/account/ops/customers/{customer_id}/grace	宽限期设置：延长或清零（0 即清除、欠费即时停服）；仅动宽限截止，不动账务与欠费标记

- 运营面端点的统一索引与鉴权分组登记在 D11；此处语义细节为本域主语义，索引不重复展开。
- 调整不设幂等键（人工操作、必带原因、逐笔可审）；手工充现金走对公转账订单人工确认，本端点只承载核销与修正。
- 客户列表与详情的运营面读入口与 5.3 同源共用同一实现。

5.5 归因与限速读穿 (/internal ， 计量链令牌)

方法	路径	语义
GET	/internal/groups/{group_id}	按组反查归属工作区与钱包：计量归因与准入的读穿入口
GET	/internal/groups	全部在营组的生效限速（组显式值优先，否则档位默认；降档事件生效期间减半），分页返回并带截断标记

- 组资源本身属租户与组织域，此处只承载「生效限速」的读侧计算；组额度配置不在本域。

6. 关键约束与验收标准

6.1 约束

- 金额零浮点：全链路元、6 位小数、字符串传输；取整只在计量方一次完成，本域不二次取整。
- 扣费不因余额不足拒付：唯一拒绝条件是客户状态非在营（整批）与系统级错误；费用照扣，现金可为负。
- 幂等三层：锁内幂等预查 + 批内幂等键去重 + 数据库部分唯一索引兜底；漏锁双写是显式失败，不是静默双扣。
- 扣减顺序不可变：赠金（按最快要到期优先，逐包拆行）→ 现金（差额，无下限拦截）；赠金剩余永不为负。
- 整批同事务：一次批量扣费全成或全不成；异常即整批回滚，由调用方按同一幂等键重试。
- 欠费标记单一规则：唯一置位与清除点是钱包变更事务末尾；不存在「扣费成功清标记」与「入账无条件清标记」两条旧路径。

7. 债务判据只有一条：现金余额为负；不读残留欠费标记（派生且滞后），也不读准入额度（额度耗尽不构成债务）。
8. 解封不等于清债：赠金入账提升可用额度但不抵扣现金负债；清债只有充值或运营调整。
9. 资金更正只追加反向流水：数据库层拒绝改写与删除既有流水；运营调整与对账补记都以新增调整行为唯一手段。
10. 补记幂等且拒争议客户：缺口为零或同标记流水已存在即空跑；参与过余额转让的客户与注销态钱包拒绝自动补记；显式金额须等于缺口。
11. 余额快照口径统一：新流水一律写「本事务提交时点现金余额」，整批扣费共享批末余额，赠金入账行快照为入账前现金；存量行为空、不回填。
12. 生命周期出口先清债：注销、关户、余额转出遇负现金余额一律拦截；关户另需剩余赠金与现金均为零。
13. 钱包锁域单一：同客户的建钱包与全部钱包变更共用同一把按客户标识的咨询锁；双钱包场景按标识升序加锁。
14. 运营面写操作留痕：状态、信用额度、调整、宽限四类变更均与业务变更同事务落审计，并同步发布状态（含解封方向）。

6.2 验收标准

#	验收口径
1	同一幂等键重复提交扣费：第二次返回幂等标记且不重复扣减，流水行数不增加
2	批内出现重复幂等键：整批 422 拒绝，无任何流水落库
3	客户状态非在营时提交扣费：整批 403 拒绝，无部分落库
4	现金不足时提交扣费：费用全额照扣、现金转负，响应不返回余额不足语义
5	一次扣费跨越两个赠金包：流水为「每包一行 + 现金行一行」，各行共享同一幂等键与批末余额快照
6	赠金按最快要到期优先消耗；某赠金扣满后剩余恰为 0，不出现负剩余
7	并发两次同客户扣费：现金余额为两次扣减之和，无丢失更新（对账无缺口）
8	数据库层直接更新或删除任一流水行被拒；整表清空语句同样被拒；旁路开关在普通角色下打开仍被拒
9	同客户同类型同引用重复入账：返回幂等，赠金不重复发放
10	赠金引用以订单号前缀开头：入参即拒（422）；绕过入参层写入的跨类型同引用行撞互斥索引时显式 409，不收敛为幂等
11	可用额度转正或转负后，欠费标记在同一事务内被清除或置位
12	「现金为正、唯一赠金刚过期」与「现金为零、无赠金、无信用」的客户发起注销或余额转出：均放行，不被残留标记或准入额度误拦
13	「现金为负」的客户发起注销 / 关户 / 余额转出：409 存在未清偿债务；现金清偿后放行

#	验收口径
14	余额转出：转出方现金清零、接收方原子自增；两侧各有且仅有一行对偶流水且引用相同；重复提交自然空跑不二次入账
15	全库对账：任一客户流水代数和与现金余额不等即告警并列出缺口；补记后缺口归零，同客户重跑不再写入
16	参与过余额转让的客户：对账归人工复核桶、补记被拒
17	免计费项：金额必须为 0 且带枚举原因，落一行恒 0 留痕流水、不动现金与赠金；裸零（未标豁免）被 422 拒绝
18	信用额度调高使可用额度转正后，欠费标记被清除、状态转为在营；下调不触碰现金与赠金
19	降档事件生效期间，组生效限速恰为原值的一半（下限 1）；事件解除后恢复原值
20	门户钱包响应中金额均为元字符串，现金余额为负时如实返回负值

6.3 待后续设计细化的开放项

- 1. 自动充值（配置字段已就位）的触发阈值、扣款幂等口径与失败重试策略。
- 2. 余额告警阈值的扫描节奏与去重窗口：字段与阈值语义在本域，扫描与投递在通知与触达域。
- 3. 信用额度的档位预设集合与提额审批口径：现为运营逐客户设置，档位模板待定。
- 4. 流水的分区与归档策略、导出上限与保留期。
- 5. 订阅周期额度发放与赠金账本的耦合口径（发放动作在订购与支付域，账本为本域）。