

D08 发票域 · 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。 章节顺序理由：先数据模型（事实结构）→ 再核心流程（结构之上的运行路径）→ 最后接口契约（流程的对外投影）。

1. 定位与边界

定位：账期费用的收据化环节。回答三个问题——「这个账期该开多少」（账期费用汇总对量）、「开出来的是哪一张票」（发票实体与发票号）、「票开错了怎么退回去」（作废重开）。申请流与发票实体严格分离：申请是请求与处置的过程，发票是开票结果，一张申请可以被驳回零次或多次开票，也可以在作废后退回重开。

关键口径

概念	承载	语义
申请	<code>invoice_applications</code>	请求流：收票主体 + 账期 + 金额 + 票种 + 抬头；状态机主体
发票	<code>customer_invoices</code>	开票结果：发票号、开票时刻、抬头快照；不出现在申请列表以外
账期	<code>period_start</code> / <code>period_end</code>	日粒度闭区间，含首含尾
金额	<code>amount_cents</code>	分、整数；与跨域汇总源的元口径按 1 元 = 100 分换算
抬头	<code>title_info</code>	抬头与税号；缺省取组织企业资料
票种	<code>invoice_type</code>	仅两值：普票 / 专票
收票主体	<code>organization_id</code>	收票主体是组织；资金主体是钱包，二者一一对应
欠费态	<code>customers.overdue_since</code>	开票前置拦截的读侧判据（口径见 §4.3）

边界

- 不管计价与金额生成：申请上的金额由提交方给定并经人工核对，本域不做按定价规则的换算；账期费用汇总的读通道属计量与计费域。
- 不管资金流：开票不改现金余额、不动赠金、不落资金流水；退款与退款渠道对账属退款与对账域。
- 不管欠费判定本身：`overdue_since` 的置位与清除只在钱包变更事务末尾发生；本域只读该标记。
- 不管税控与票据实物：税控对接、纸质票寄送、税局回执不在本域。
- 不管门户自助入口的编排与展示：客户侧提交与查看由用户自助与门户域承载，详见 D12；本域提供申请人归属与抬头快照口径。
- 不管运营面端点的索引与权限载体：运营面端点语义细节在本域，统一索引登记在 D11。

归属口径：本域端点 = 内部运营链 `/internal/invoice-applications`（5 个）+ 运营面 `/api/account/ops/invoices/applications`（5 个）。两通道共用同一实现。

2. 角色与依赖

能力矩阵

能力	运营面	内部运营链（持运营令牌的调用方）	门户用户	计费路由与计量方
代客建申请	✓（scope saas:ops:invoice e）	✓	✗	✗
客户侧提交申请	✗	✗	✓（申请人落 applicant_user_id ）	✗
查申请台账（筛选 + 附组织名与发票号）	✓	✓	✗	✗
开票	✓	✓	✗	✗
驳回	✓	✓	✗	✗
作废重开	✓	✓	✗	✗

- 两条通道的操作人都是可空的（见 §4.7）：运营面取登录主体标识，内部运营链取请求头 X-Operator-Id。

上游依赖

上游	依赖内容	语义
租户与组织域	组织存在性、组织名、信用代码、实名等级、主体性质	收票主体与抬头缺省值、票种门禁的判定输入
钱包与账务域	由组织反查钱包、欠费态标记	发票实体挂钱包主体；欠费账期不可开
计量与计费域	账期用量与金额汇总	申请金额的核对数据源（人工核对）
实名与合规域	开票实名门禁判定	普票需主体达 L1、专票需企业达 L2

下游被引用

引用方	引用内容	语义
运营管理域	变更动作与操作人审计行	申请与发票的全部处置动作留痕，供运营台与合规追溯
用户自助与门户域	发票列表与票面字段	客户视角的发票展示（金额按分口径的展示规范详见 D12）
账期账单消费方	发票金额与账期区间	账期账单的「票面」一侧；金额与用量对量以计量与计费域口径为准

3. 数据模型

3.1 域内关联总览

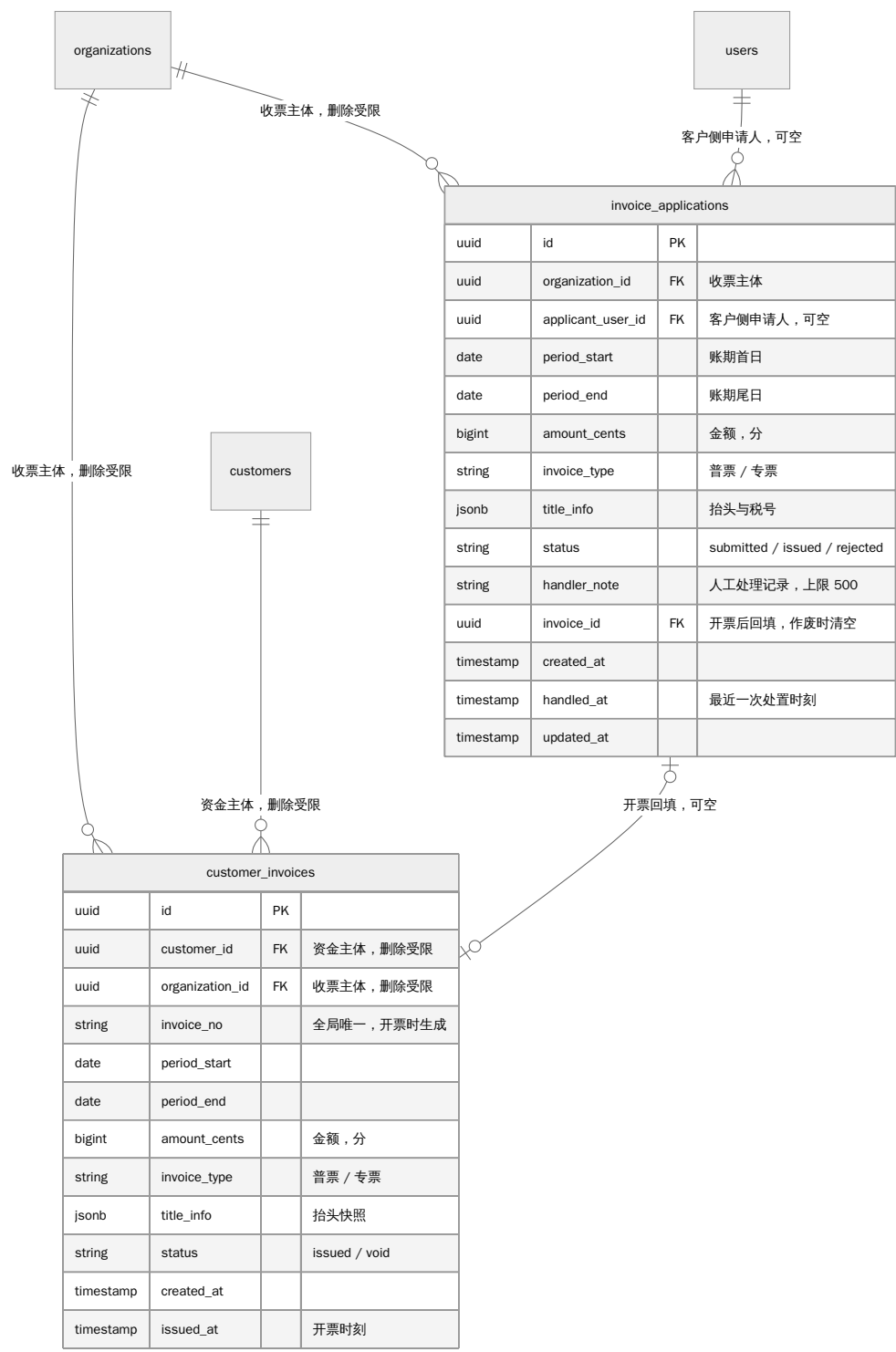


图 1

关联线是单向指针：申请可指向一张发票，发票不反向持有申请。作废重开时申请指针被清空并重新指向新发票，旧发票行原样留存供追溯。

3.2 域间引用

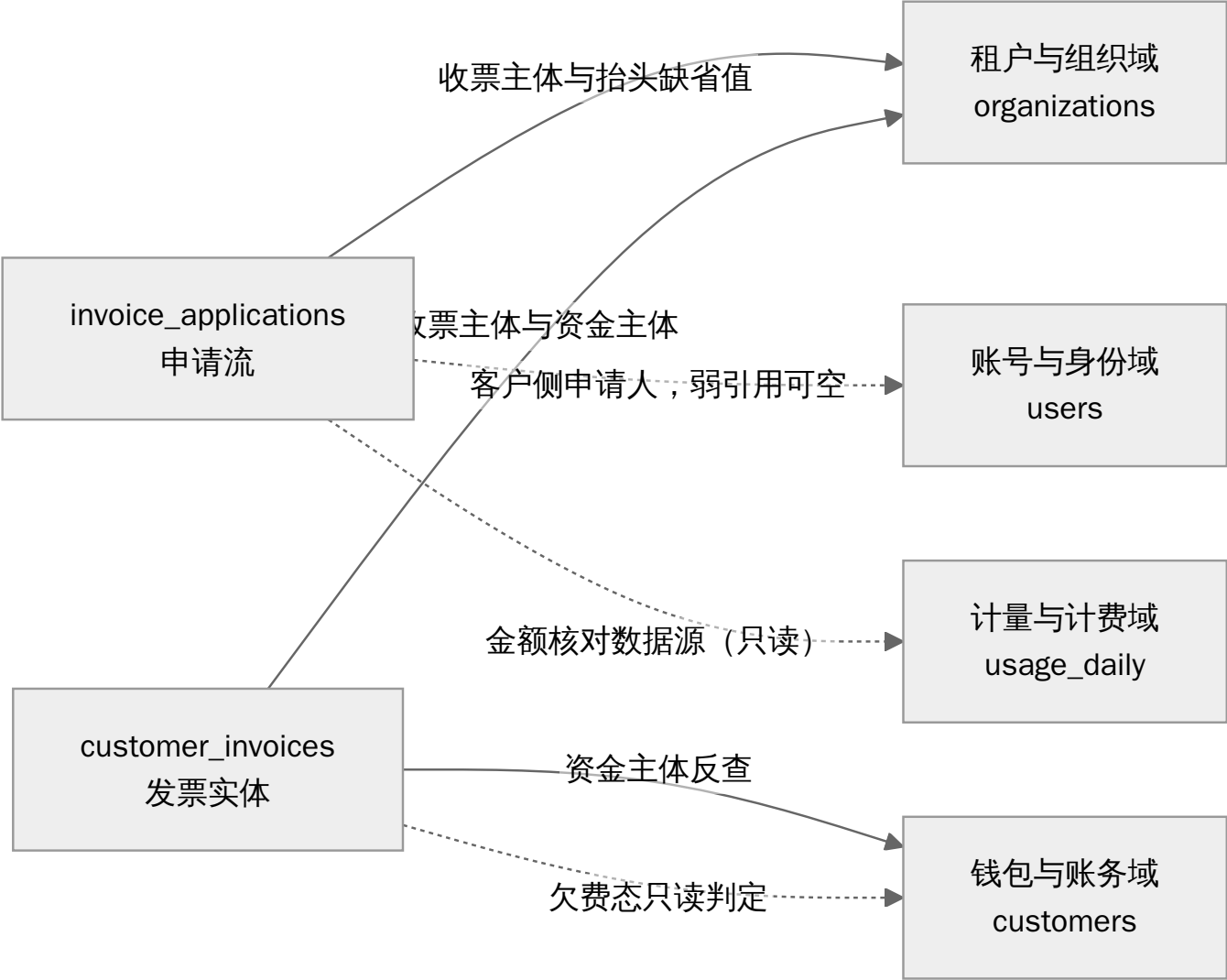


图 2

- 组织与钱包引用是实外键且删除受限：收票主体与资金主体不允许被物理删除而留下无主票据。
- 申请人对身份的引用是可空外键、置空删除：操作人不在本仓用户表中时该列留空，而操作人线索走审计（见 §4.7）。
- 账期汇总只作读数源，不建跨域外键。

3.3 表清单 (2 表)

表	职责	关键字段
invoice_applications	申请流与状态机主体：账期 + 金额 + 票种 + 抬头 + 处置结果	organization_id / applicant_user_id / period_start / period_end / amount_cents / invoice_type / title_info / status / handler_note / invoice_id / handled_at / updated_at
customer_invoices	开票结果：发票号 + 票面快照 + 生命周期状态	customer_id / organization_id / invoice_no / period_start / period_end / amount_cents / invoice_type / title_info / status / issued_at

3.4 设计约束

1. 金额为分整数：`amount_cents` 为分口径整数列，不接受小数；跨域汇总源为元、6 位小数，两者在开票前按 1 元 = 100 分对齐 (§4.8)。
 2. 发票号全局唯一：`invoice_no` 上建唯一约束，跨组织、跨账期、跨票种一律不重复；唯一约束是最终防线，生成规则只负责低撞号概率 (§4.3)。
 3. 发票号可空：`invoice_no` 在发票实体建立时即写入，可空仅为兼容历史基线缺省；业务上不存在「已开票而无号」的发票。
 4. 申请状态缺省为 `submitted`：建申请即入可处理态；两表的状态列都是字符串列，取值由服务层枚举把关，不依赖数据库层检查约束。
 5. 发票状态缺省为 `pending`：本域写入路径只产生 `issued` 与 `void` 两值；`pending` 是表级缺省的历史基线值，不表述业务态。
 6. 收票主体与资金主体删除受限：申请与发票对组织的引用、发票对钱包的引用均为受限删除；不允许删组织或删除钱包连带抹掉票据。
 7. 申请人对用户表置空删除、申请对发票置空删除：申请人离场不影响申请与票据；发票行被删（受控维护场景）时申请指针置空，不连带删除申请。
 8. 抬头是快照：`title_info` 为文档型列；申请上的抬头在开票时复制到发票实体，此后组织企业资料变更不影响已开发票票面。
 9. 申请不设账期唯一约束：同一收票主体、同一账期允许存在多条申请（换票种、换抬头、拆分金额），冲突由处置层的状态机与人工核对消解，而不由数据库唯一约束隐式拒绝。
 10. 处置时刻单一：`handled_at` 记录最近一次处置时刻，三个处置动作都会推进它；申请列表按 `created_at` 降序返回，行附组织名与已开发票号。
-

4. 核心流程

4.1 申请创建（两条入口合一）

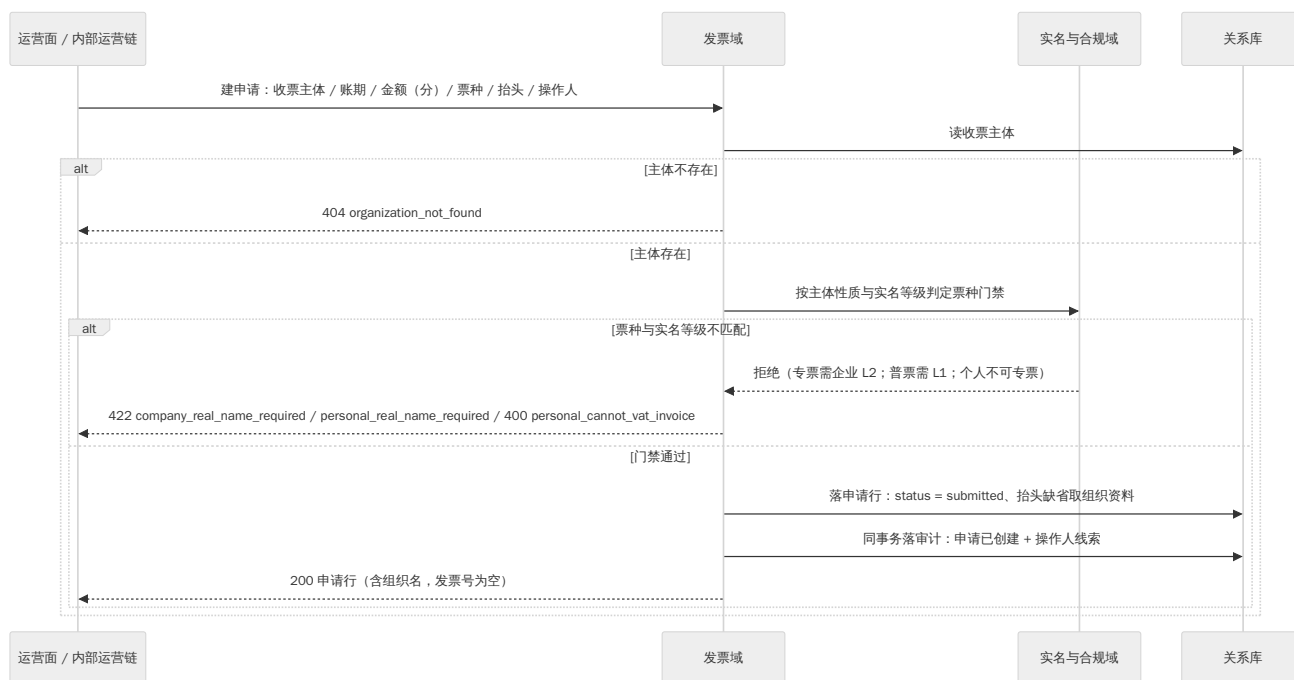


图 3

- 抬头缺省为组织名与信用代码，提交方可整体覆盖 `title_info`。
- 代客创建时申请人列置空——操作人属平台侧身份，不是客户侧申请人，线索走审计。
- 两条入口（运营面、内部运营链）共用同一实现与同一校验序，差别只在鉴权通道与操作人来源。

4.2 创建前置判定链

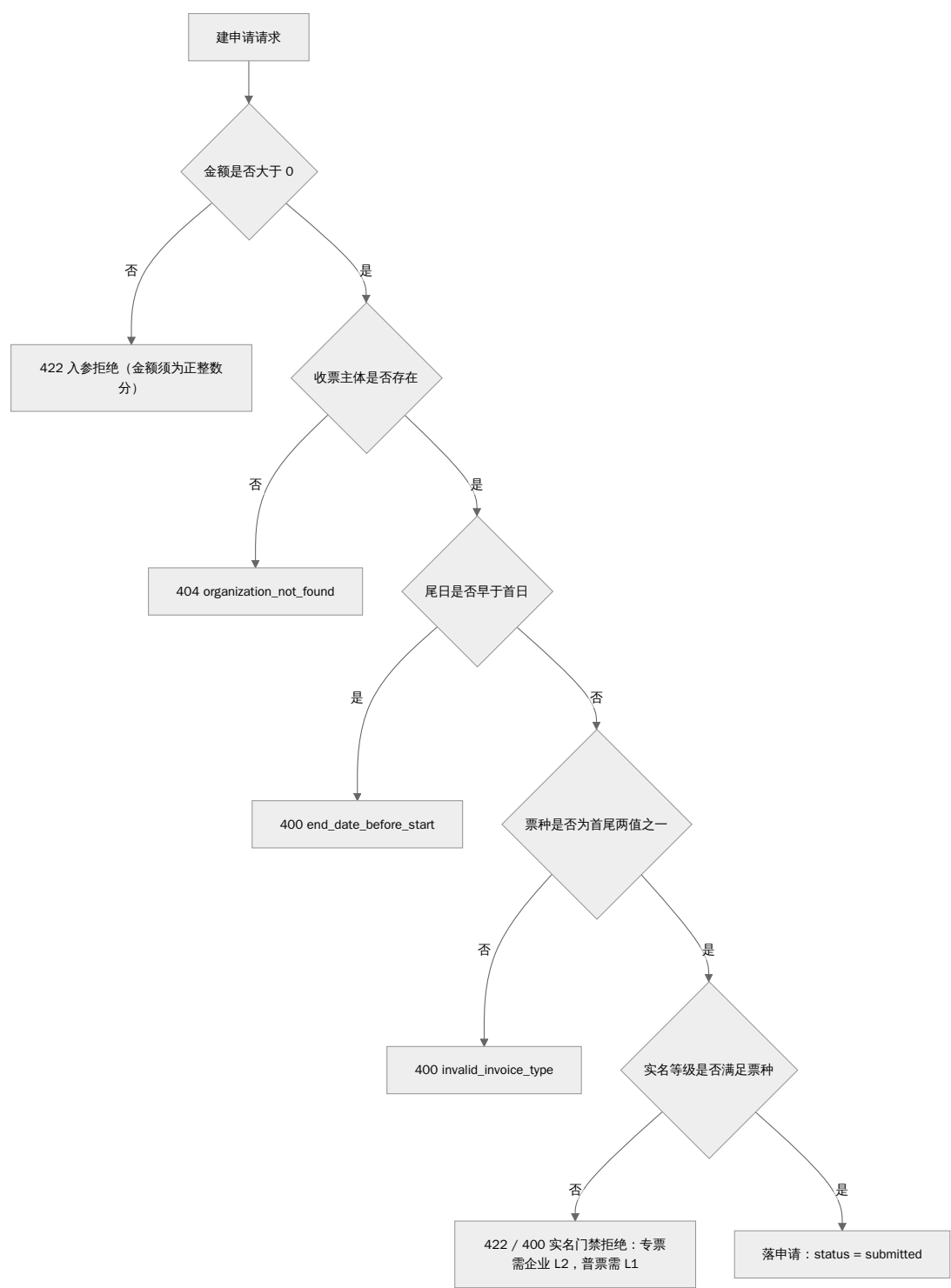


图 4

- 金额下界在入参层以正数约束先拦（0 与负值 422），服务层再兜一次「金额非正即 `invalid_amount`」，防绕过入参层的调用路径。
- 票种为受限两值，非枚举值一律拒；个人主体开专票恒拒（与实名等级无关）。
- 账期只校验顺序，不校验是否为整月或已封账历史区间——账期正当性由人工核对承担。

4.3 开票

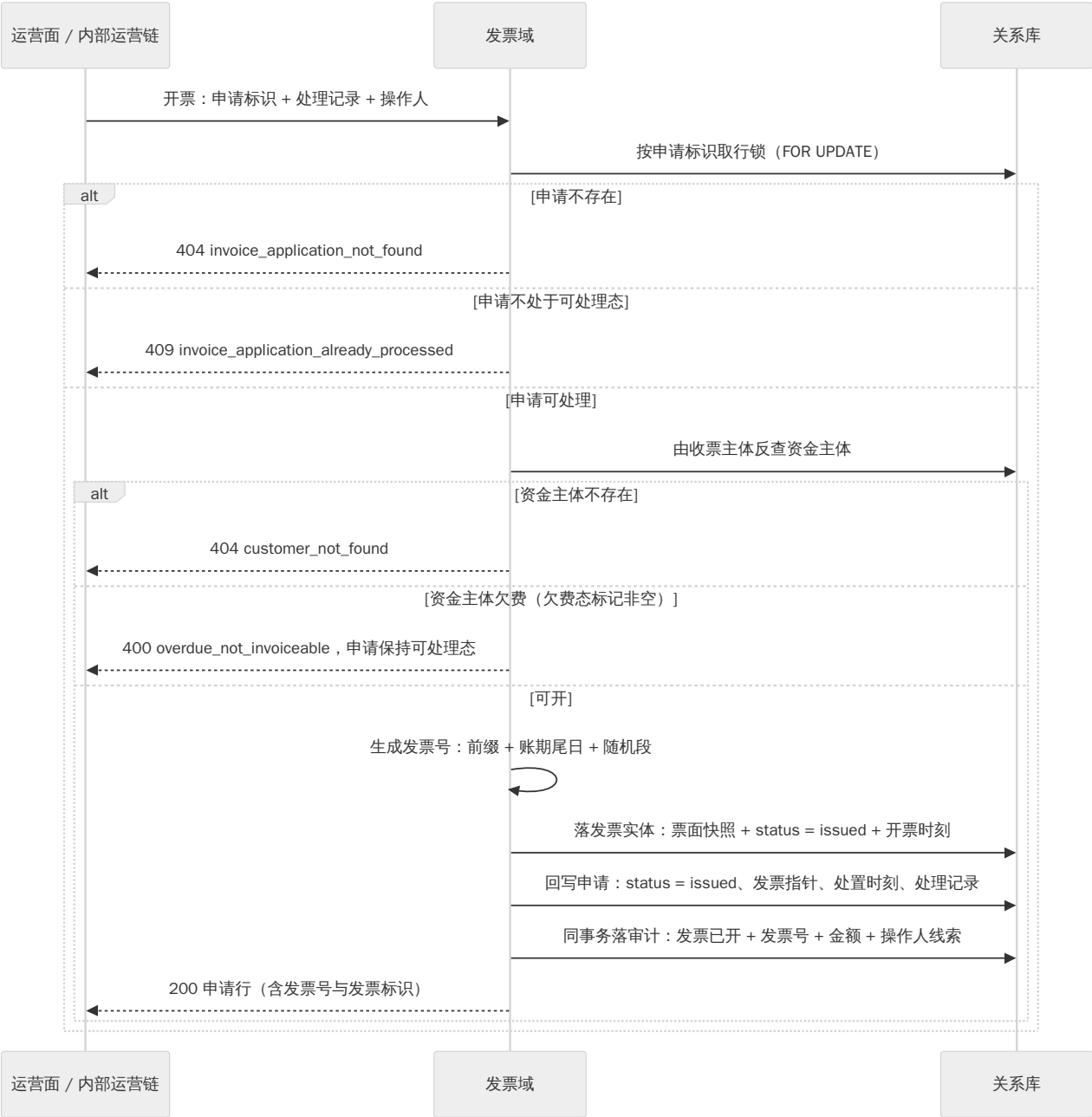


图 5

- 欠费判据：读 `customers.overdue_since` 非空即拦（400 `overdue_not_invoiceable`），拦截失败后申请保持可处理态、不产生任何发票实体。该标记是派生标记（只在钱包变更事务末尾重算），此处取的是「客户当前处于欠费态」，即开票时点的服务状态口径；这与生命周期出口的债务判据（只认现金余额为负）是两个不同用途：出口拦的是未清偿债务，开票拦的是账期服务状态，故此处按标记判定，不改写为现金判据。
- 发票号规则：`INV` + 账期尾日的年月日八位 + 8 位十六进制随机段（大写），定长 19 字符；尾日段使同账期票据在号面上可归堆，随机段使同秒并发不撞号。全局唯一由 `invoice_no` 上的唯一约束兜底，跨组织不重号；随机段撞号（概率量级 $1/2^{32}$ ）会由唯一约束冲突显式失败，不静默改号。
- 票面快照与同事务：账期、金额（分）、票种、抬头一并从申请复制到发票实体，此后申请不可再改（唯一可改字段是处理记录）；开票、回写与审计同事务一次提交，不存在「发票已建而申请未回写」的中间态。

4.4 申请状态机

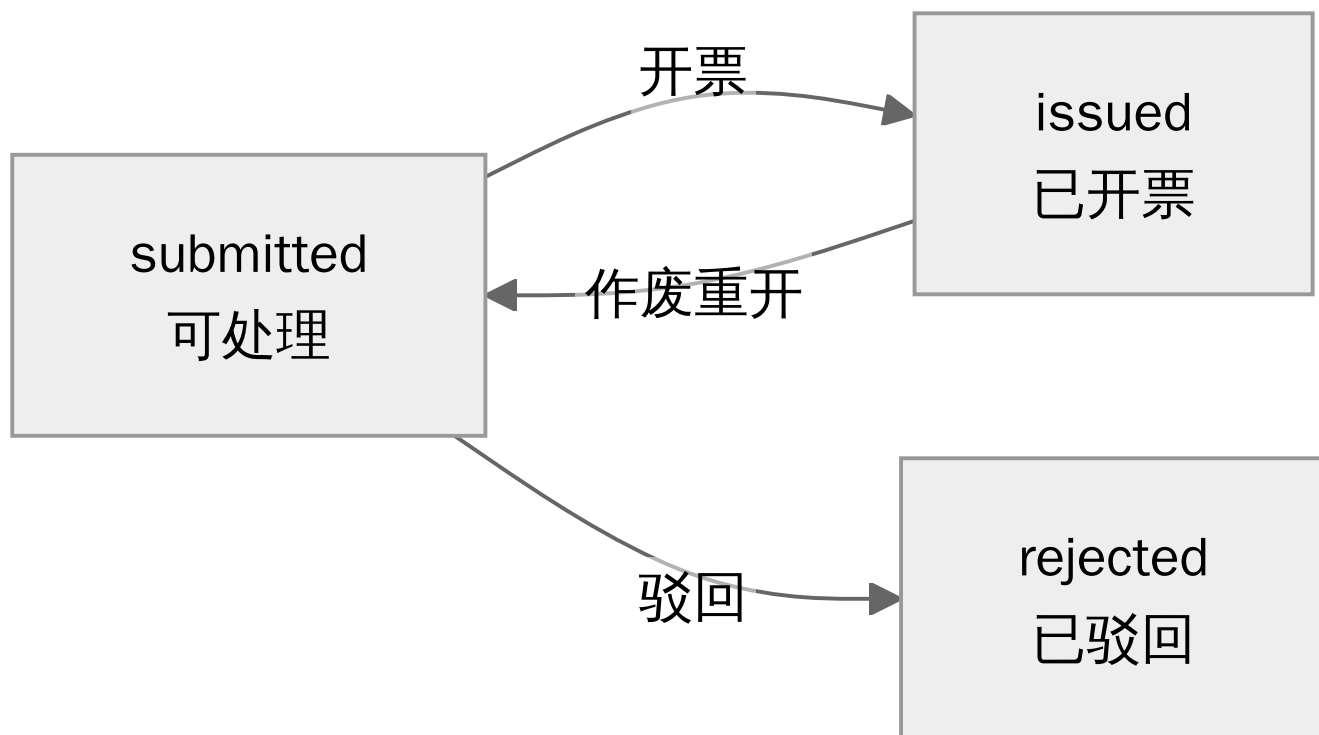


图 6

- 三个状态字符串：`submitted` / `issued` / `rejected`；`issued` 是唯一可作废的状态，作废后退回 `submitted`（回环）。
- `rejected` 在处置语义上为终态：已驳回的申请既不能再开票也不能再驳回（均 409），需要重新开票时另建申请。
- 作废不回退 `rejected`：作废只能对自己开出的票发起。

4.5 作废重开

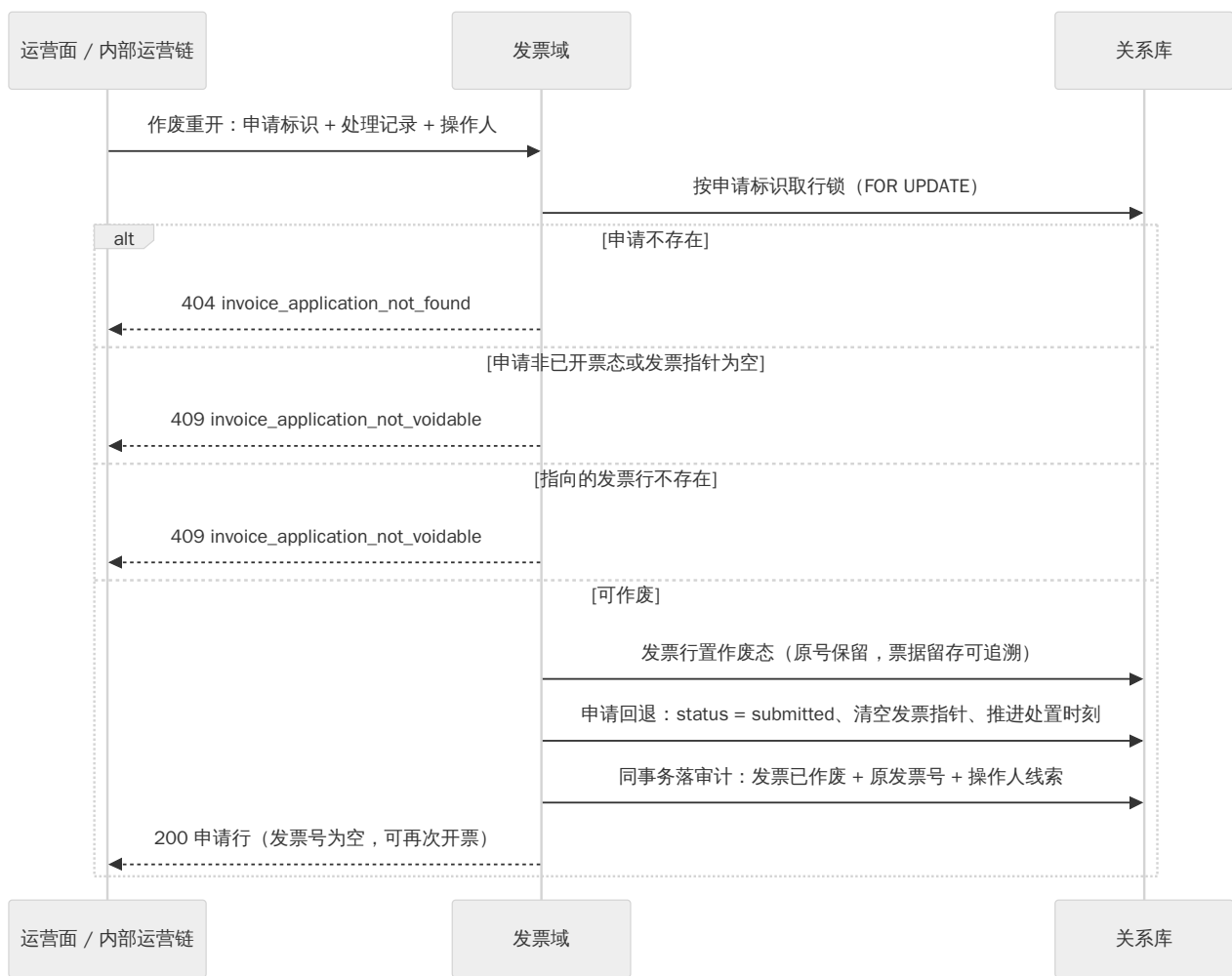


图 7

- 两步同事务：发票置作废与申请回退必须在同一事务内提交。任何一步失败即整体回滚——否则会出现「票已废而申请仍指向废票」（无法重开）或「申请可重开而旧票仍有效」（一账期两张有效票）。
- 重开生成新号：退回 `submitted` 后再走 §4.3，新发票号为新建实体的号，与已作废的旧号必然不同；旧票行状态保持作废、号不被复用。
- 作废不校验欠费：作废是把错票收回，不是开出新票；欠费拦截只发生在开票动作上。
- 作废是否退费不属本域：票据作废引起的资金后果（退款、核销）在退款与对账域与钱包与账务域处理。

4.6 驳回

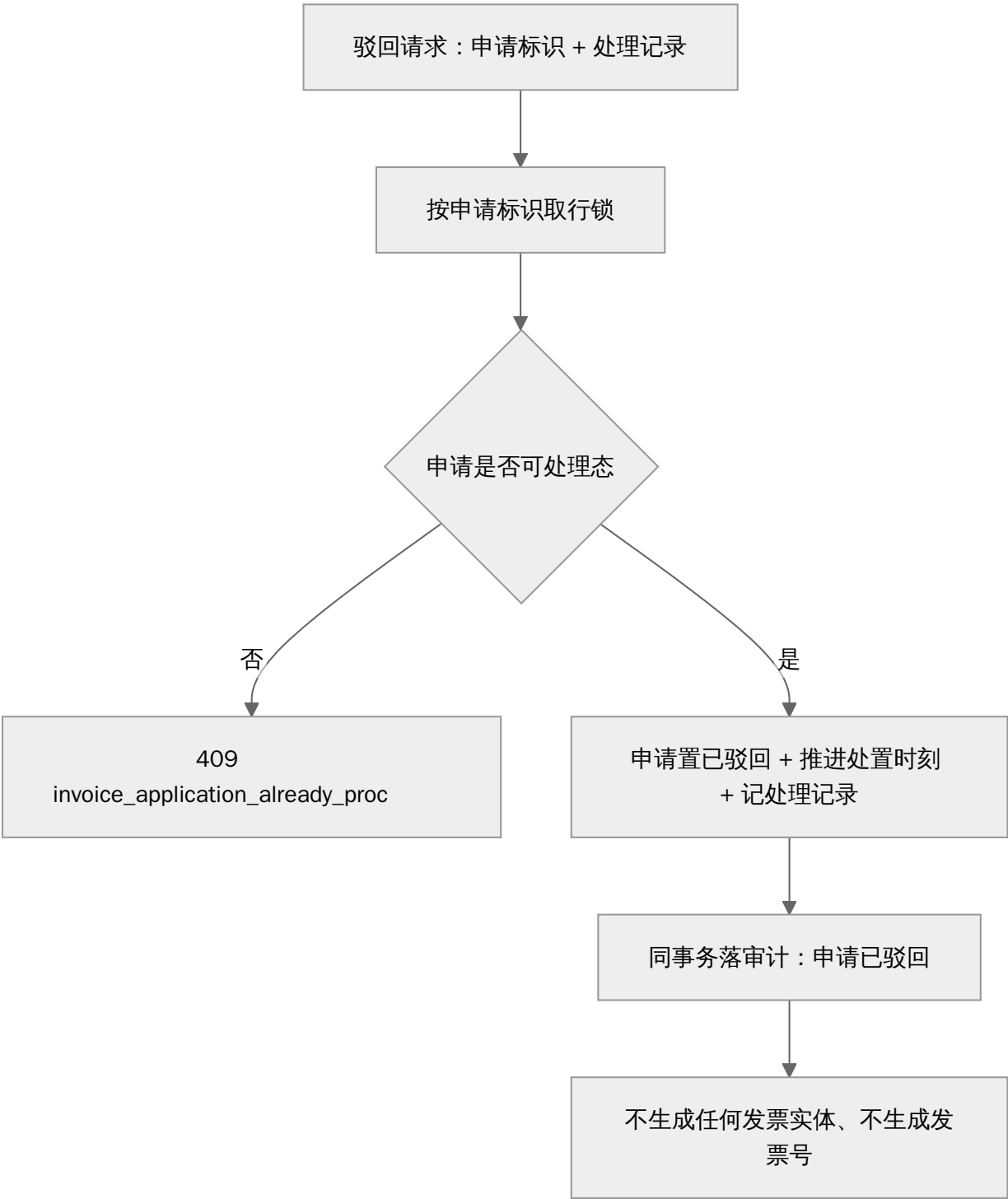


图 8

- 驳回是「不受理」语义：无票据、无号，不留半成品；后续如仍要开票须另建申请。
- 驳回原因写入处理记录（上限 500 字符），是台账上唯一的驳回说明载体；驳回与开票共用同一取单入口与行锁，并发同申请仅一方成功 (§4.9) 。

4.7 操作人归属与外键安全

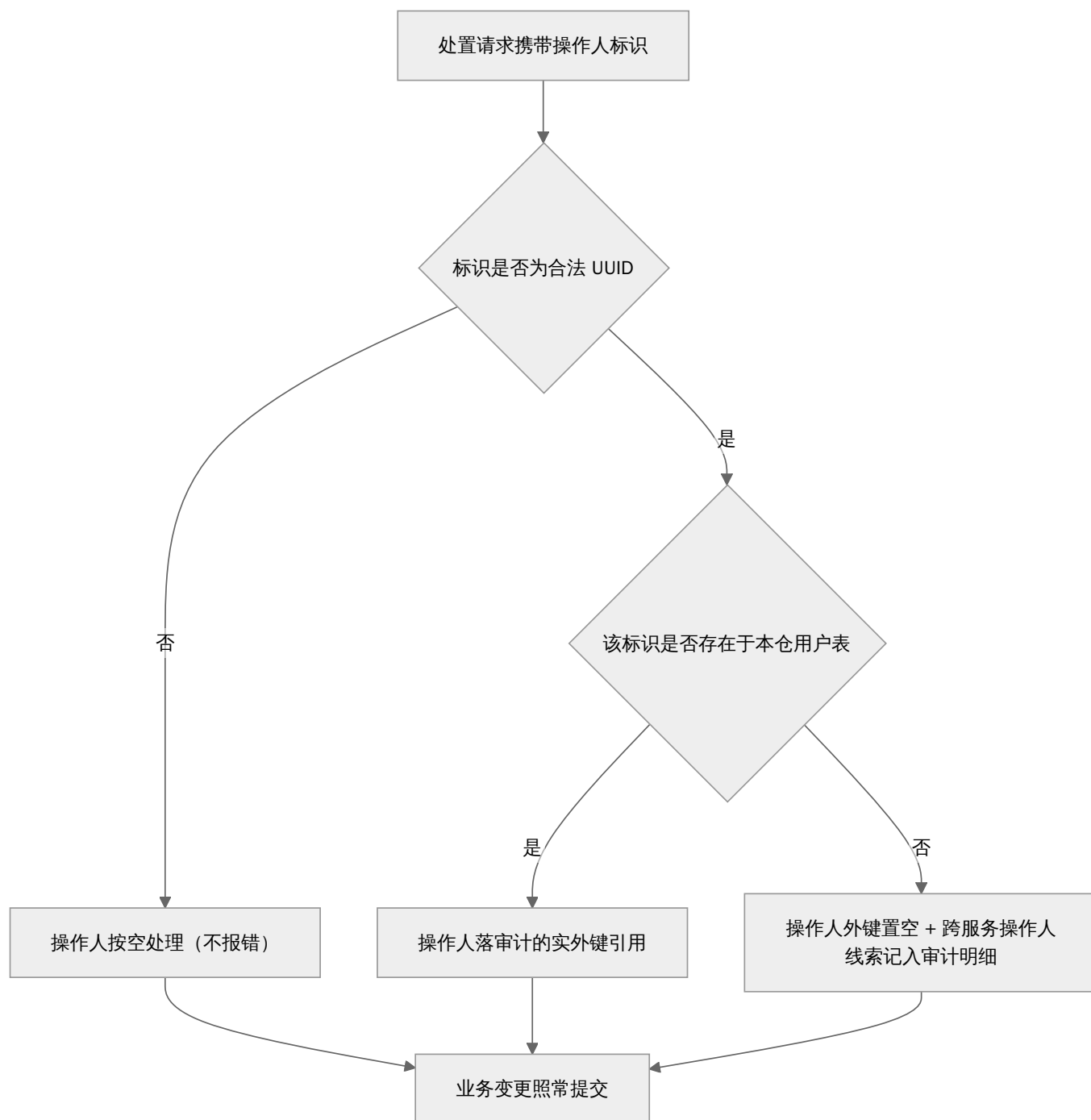


图 9

- 为什么置空不报错：运营侧身份来自平台侧账号体系，本仓用户表里往往没有对应行；若直接把它写入审计的操作人外键，外键约束会把一次正常的开票动作打成 500。
- 审计怎么留：动作、目标类型、目标标识、处理明细（含发票号与金额）随业务同事务落审计；跨服务操作人线索以文本形式记入明细，运营台可按该线索回溯外部操作者。
- 审计层另有类型守卫：操作人实参若非 UUID 类型则降级为空并告警——降级方向是「审计缺操作人，但主流程不崩」。

4.8 账期费用汇总作为开票数据源

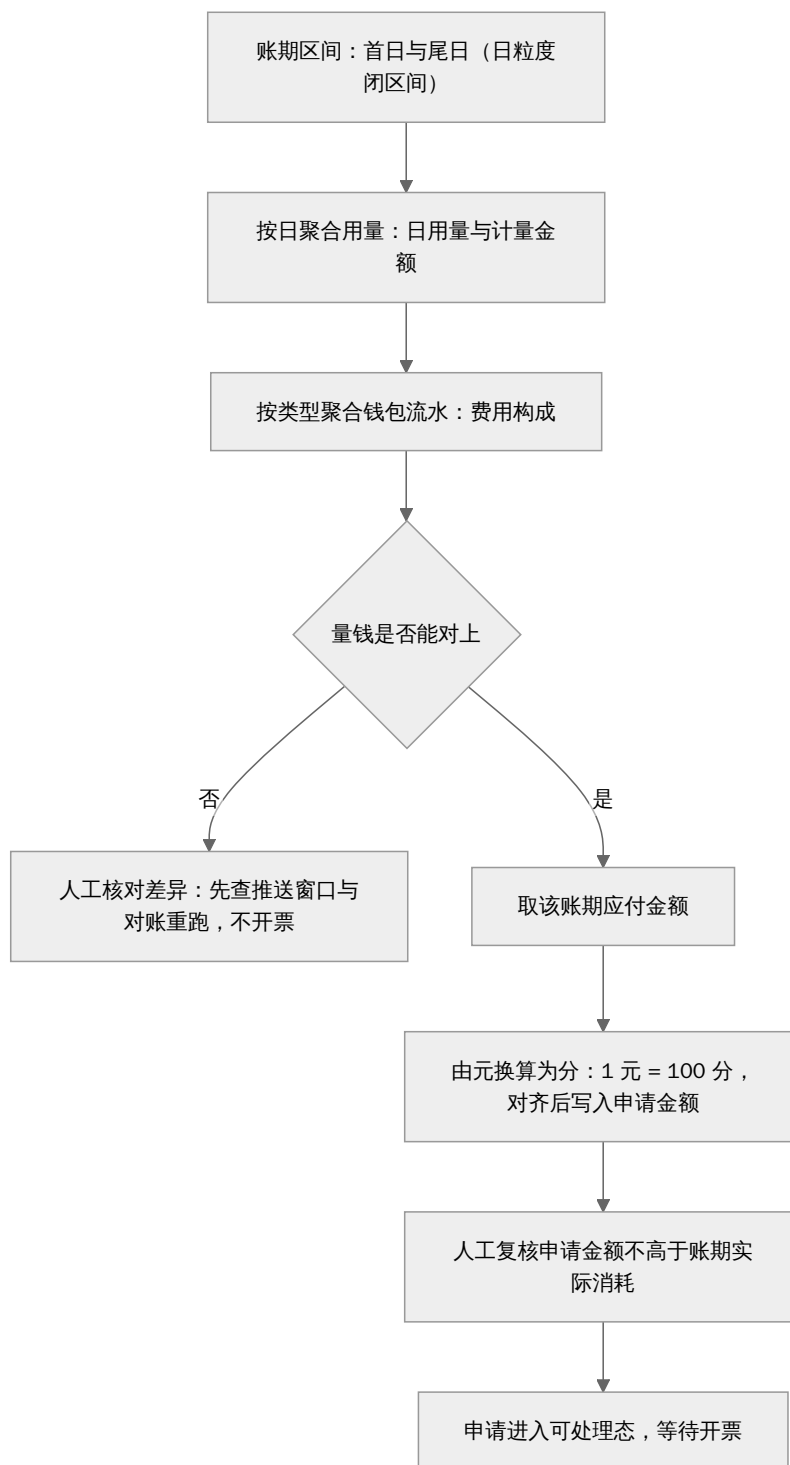


图 10

- 账期区间如何划定：以申请上的首日与尾日为准；用量侧按同一区间做日聚合，归日为统一业务时区，写入侧只接受落在回溯窗口内的日期，历史账期不被迟到推送污染。
- 金额如何取：以计量金额（用量侧）与按类型聚合的流水为两侧对量依据；「量可重算、钱不可重放」，所以差异先查推送与对账，不靠改申请金额抹平。
- 以什么为单位：汇总源为元、6 位小数；申请与发票金额列为分整数，两者按 1 元 = 100 分对齐，换算结果按四舍五入取整到分。

- 「申请金额不高于账期实际消耗」的自动校验依赖计价落定，当前口径为人工核对并在处理记录中留痕（见 §6.3）。

4.9 幂等与并发收敛

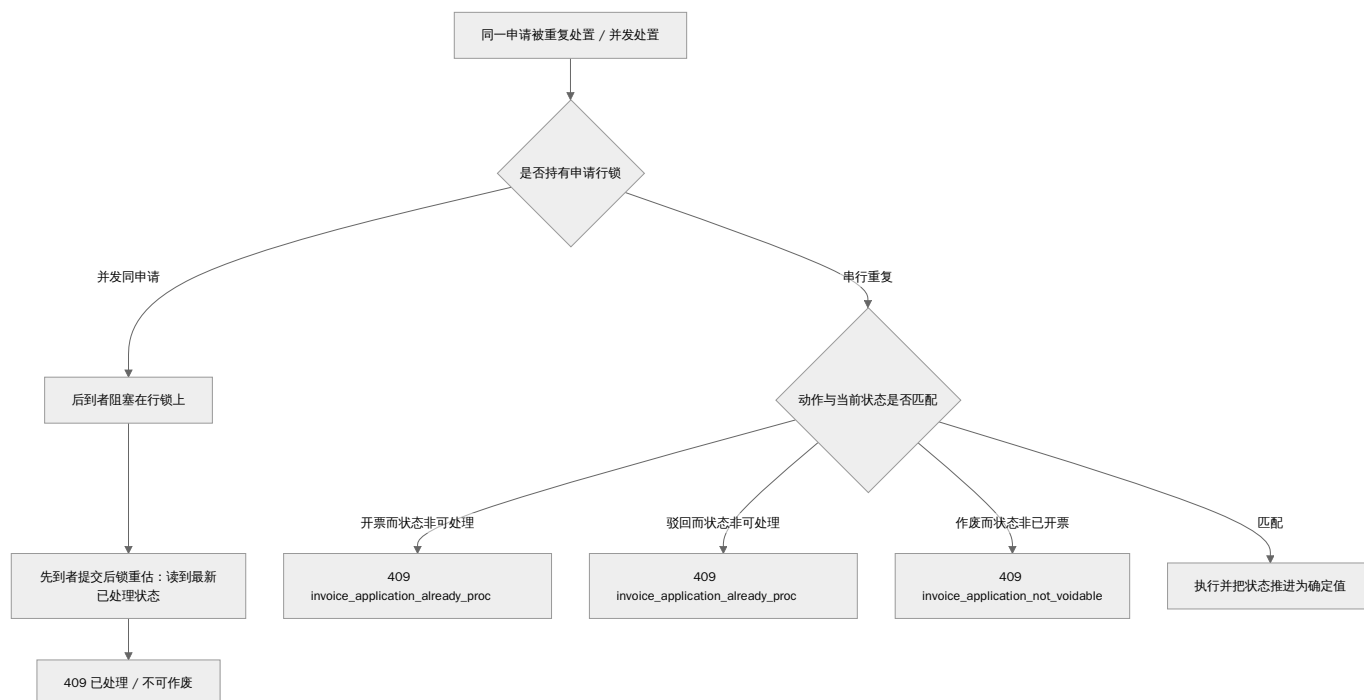


图 11

- 重复开具收敛为显式失败而非静默幂等：第二次开票得到 409，不生成第二张票，也不复用原票号回复——「一申请一有效票」下，重复调用是运营动作失误，需要被看见而不是被吞掉。
- 并发双开由申请行锁串行化：取单即加行锁，后到者在锁上等待，先到者提交后锁重估读到 `issued`，进入 409 分支；从机制上不可能出现一申请两票。
- 重复作废得到 409（非已开票态即不可作废）；作废后重开是第一次开票，产生新号，不与原动作视为重复。
- 开票失败不留半成品：欠费拦截与状态不匹配都发生在落库之前，申请与发票实体都保持原样。

4.10 抬头与开票主体

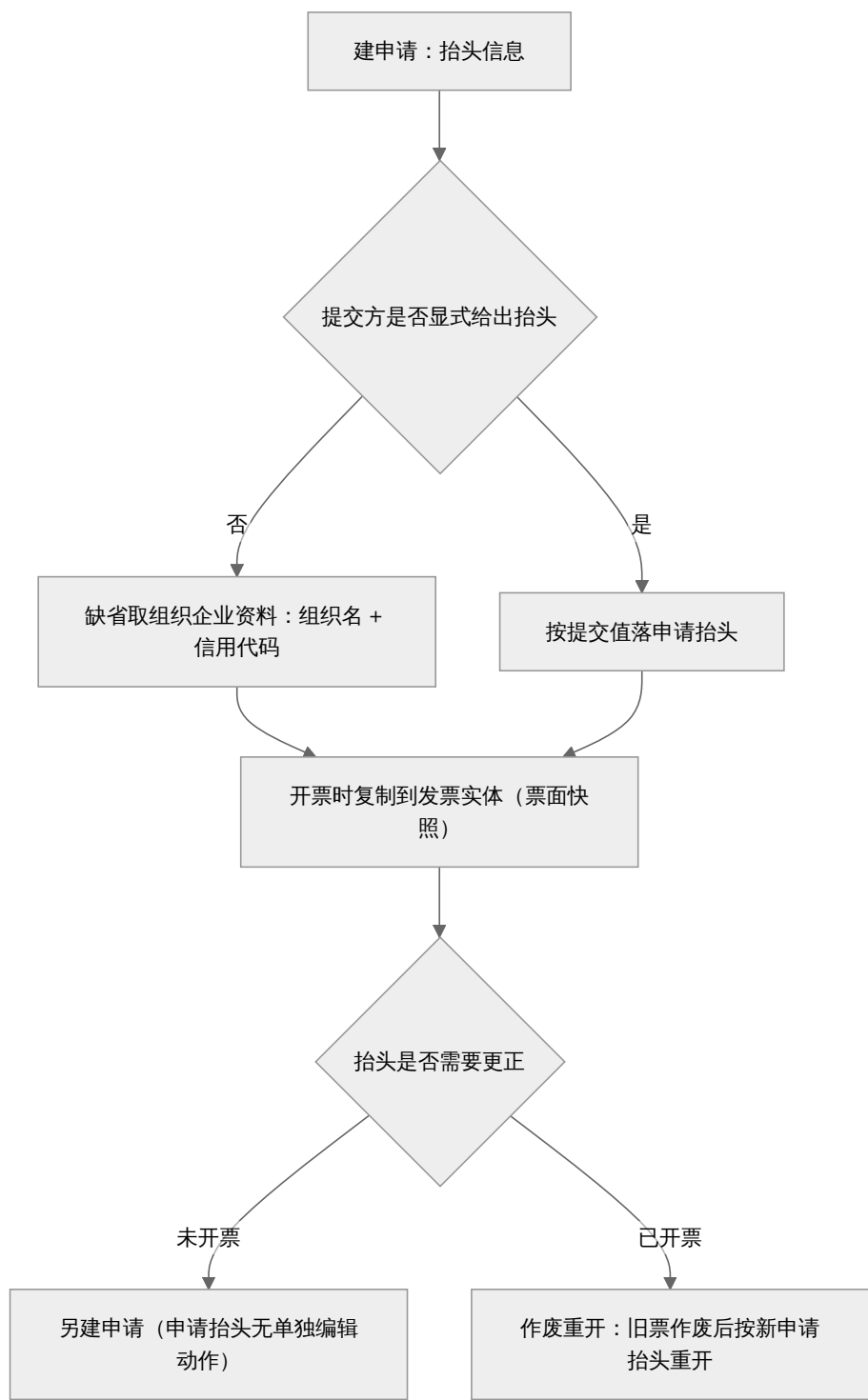


图 12

- 抬头存在哪里：先落申请的文档型抬头列，开票时复制到发票实体的抬头列；发票实体持有票面快照，不随组织资料后续变更而变。
- 与开票主体的关系：收票主体是组织（决定票种门禁与抬头缺省值），资金主体是钱包（由组织反查落票）。抬头可覆盖缺省，但不能改变收票主体。
- 变更口径：申请抬头无独立编辑动作；未开票时改抬头即另建申请，已开票时须走作废重开，避免同一发票号下票面被改写。

5. 接口契约

5.1 内部运营链（内部令牌的运营范围，5 个端点）

方法	路径	语义
GET	<code>/internal/invoice-applications</code>	申请台账：按创建时刻倒序，可按收票主体与状态筛选，分页上限 200；行附组织名与已开发票号
POST	<code>/internal/invoice-applications</code>	建申请（代客）：校验主体 / 金额 / 账期 / 票种 / 实名门禁 → <code>submitted</code>
POST	<code>/internal/invoice-applications/{application_id}/issue</code>	开票：欠费账期不可开；同事务落发票实体（ <code>issued</code> + 发票号）并回写申请
POST	<code>/internal/invoice-applications/{application_id}/reject</code>	驳回： <code>submitted</code> → <code>rejected</code> ，不产生发票实体
POST	<code>/internal/invoice-applications/{application_id}/void</code>	作废重开：原发票置 <code>void</code> ，申请退回 <code>submitted</code> ，可重新开票

- 鉴权：内部令牌须持有运营范围，缺失即 403 `forbidden`；操作人经请求头 `X-Operator-Id` 传入，可空。
- 请求体：建申请带收票主体、首尾日、金额（分，正数）、票种（缺省普票）、抬头（可空）；处置体只带处理记录（上限 500 字符，可空）。
- 响应统一为「成功标记 + 申请行」，申请行含发票号与发票标识（未开票为空）。

5.2 运营面（`saas:ops:invoice` 权限范围，5 个端点）

方法	路径	语义
GET	<code>/api/account/ops/invoices/applications</code>	发票申请台账（与内部通道同源同实现），行附组织名与发票号
POST	<code>/api/account/ops/invoices/applications</code>	代客建申请（运营代客户发起 → <code>submitted</code> ）
POST	<code>/api/account/ops/invoices/applications/{application_id}/issue</code>	开票（欠费账期不可开）
POST	<code>/api/account/ops/invoices/applications/{application_id}/reject</code>	驳回（无发票实体产生）
POST	<code>/api/account/ops/invoices/applications/{application_id}/void</code>	作废重开（发票作废 + 申请退回）

- 权限范围独立于客户账务与信用额度范围：只持信用额度范围的运营角色访问发票台账即 403（能力隔离在权限范围内落地）。
- 两条通道共用请求体模型与同一服务实现，差异仅有鉴权通道与操作人来源；运营面端点的统一索引与权限载体登记在 D11。

5.3 客户侧入口（归属说明）

方向	承载	语义
客户提交	申请行上的客户侧申请人列	客户侧提交的申请携带申请人；代客申请该列为空，操作人在审计
客户查看	用户自助与门户域	发票列表与票面字段的展示由门户承载（展示规范详见 D12）

- 客户侧不为本域新增第二条写入路径：申请创建、状态机与发票实体全部复用同一套实现，客户侧只多带一个申请人归属。

6. 关键约束与验收标准

6.1 约束

1. 申请与发票分离：申请是流，发票是结果；发票实体只由开票动作创建，被驳回的申请永不产生发票实体。
2. 可处理态唯一：仅 `submitted` 可被开票或驳回；已开票的申请不可再驳回，已驳回的申请不可再开票或再驳回（一律 409）。
3. 一申请一有效票：开票与回写同事务，发票指针同事务建立；作废时指针与发票状态同事务一并变更。
4. 作废可追溯：作废只改发票状态，发票行与发票号整体留存；重开必然生成新号，旧号不复用、不覆盖。
5. 发票号全局唯一：号面 = 前缀 + 账期尾日 + 随机段；唯一约束兜底，撞号显式失败。
6. 欠费不开票：欠费态标记非空的客户，其申请开票被 400 拒，申请保持可处理态；作废与驳回不受此约束。
7. 金额单位为分：申请与发票金额列一律为分整数；与元口径汇总源换算按 1 元 = 100 分取整到分，链路中不出现二进制浮点中间量。
8. 抬头开票即冻结：抬头随开票复制为票面快照；更正抬头只能另建申请或作废重开。
9. 处置串行化：开票、驳回、作废共用同一取单入口并施加行锁；并发同申请只有先到者成功，后到者得到 409。
10. 操作人可空但不丢线索：操作人不在本仓用户表时外键置空、线索入审计明细，绝不因外键约束把正常处置打成 500。
11. 变更必留审计：建申请、开票、驳回、作废四类动作与业务变更同事务落审计（动作、目标类型、目标标识、明细）。
12. 主体引用受限删除：收票主体与资金主体的引用为受限删除；申请人对用户表、申请对发票为置空删除。
13. 门户与运营面同源：客户侧展示与运营台账消费同一份数据，不存在两套发票事实。
14. 票种受实名门禁：专票需企业实名达 L2，普票需主体达 L1；个人主体恒不可开专票。

6.2 验收标准

#	验收口径
1	建申请后立即可见：状态为可处理态、发票号与发票标识为空，且审计中有一条「申请已创建」
2	收票主体不存在时建申请：404；金额为 0 或负数：入参 422 拒绝；尾日早于首日：400；票种非枚举值：400
3	主体实名等级不足时建申请：专票需企业 L2（否则 422），普票需 L1（否则 422），个人开专票恒拒

#	验收口径
4	开票成功后：发票实体状态为 <code>issued</code> 且带发票号，申请状态为已开票、发票指针非空、处置时刻被推进
5	发票号满足「前缀 + 账期尾日 + 8 位随机段」，且跨组织不重复；人为构造同号写入被唯一约束拒绝
6	欠费态标记非空的客户开票：400，申请保持可处理态，且不产生任何发票实体
7	同一申请重复开票：第二次 409，发票实体总数不增加
8	并发对同一申请发起两次开票：恰好一次成功、另一次 409，发票实体恰好一张
9	驳回成功后：申请为已驳回、处理记录落库，该组织下发票实体数不增
10	已驳回的申请再开票或再驳回：均 409
11	非已开票态发起作废：409；已开票发起作废：原发票置作废且号不变，申请退回可处理态且发票指针清空
12	作废后重开：新票号与作废票号不同，同一组织下发票实体为两张（一作废一已开），作废票状态不被改写
13	作废事务中任一步失败：发票状态与申请状态一并回滚，不出现半废或半开态
14	外部操作人（不在本仓用户表中）发起开票：返回 200 且正常出票，审计操作人外键为空、明细中留有跨服务操作人线索
15	内部运营链无令牌或令牌不含运营范围：全部 5 个端点均 403
16	运营面权限范围不匹配（如只有信用额度范围）：发票台账 403
17	台账筛选：按主体与状态筛选后条数正确，行内含组织名；已开票行带发票号，未开票行为空
18	抬头口径：未显式给抬头时申请抬头等于组织名与信用代码；开票后修改组织名，已开发票的票面抬头不变
19	金额口径：申请金额以分为单位落入申请与发票两表，元到分的换算按 1 元 = 100 分取整到分，无中间浮点误差
20	已开发票不因欠费态变化而被改写：欠费只在开票动作上拦截，已开票的申请状态保持已开票

6.3 待后续设计细化的开放项

1. 金额自动校验：「申请金额不高于账期实际消耗」的自动判定需等计价口径落定，当前为人工核对与处理记录留痕。
2. 客户侧提交入口的编排：客户侧提交与查看的位置、权限与频次限制归用户自助与门户域，本域只提供申请人归属与抬头快照口径。
3. 抬头编辑语义：是否允许在未开票状态下更正申请抬头（而非另建申请），以及抬头字段的规范化与校验（税号格式、抬头长度）。
4. 重复申请的去重口径：同一主体同一账期存在多条未处置申请时的合并或提示规则；当前不设账期唯一约束。
5. 开票数据源的自动化：账期汇总自动发起开票的条件（封账时点、对账通过阈值、异常人工介入），以及分口径与元口径对账的全自动闭合。
6. 票据外部状态的回接：外部开票处理端的开票结果、拒开与红冲结果如何落回本域（当前本域只承载人工开票的结果记录）。