

D01 账号与身份域 · 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：终端用户身份与凭证的统一入口，同时承担平台统一登录（OIDC IdP）的身份提供方职责。回答两个问题——「你是谁」（身份）与「如何证明你是你」（凭证）。

边界

- 本域只管身份与凭证：账号、登录、密码、联系方式、会话、单点登录、人机验证门禁、协议同意留痕。
- 不管授权业务：组织、角色、成员归属租户与组织域；本域只告知「这个请求来自哪个用户」。
- 不管资金：本域不产生、不校验任何金额字段。
- 不做实名核验：实名属风控与合规域；本域仅承载其实名结果的回写位（ `identity_verified` / `identity_level` ），不实现核验。

登录方式口径

- 邮箱为主路径（验证码登录 + 密码登录），不受开关约束。
- 手机通道受能力开关控制：手机是否可用（验证码登录手机通道、短信找回、手机绑定）为单一数据源、DB 驱动、fail-closed——生产短信通道未签约时手机入口不出现。
- 微信扫码登录为一期纳入能力；真实凭据未就位时以联调模式运行，生产禁用联调模式。

2. 角色与依赖

角色	关系	说明
终端用户	主体	个人账号；企业注册人（注册即成为组织拥有者）
用户门户	消费方	承载注册/登录/账号设置界面；登录成功后跳单点登录回跳端点
统一登录客户端（RP）	消费方	经授权码 + PKCE 换取令牌； <code>scope</code> 含门户的客户端走浏览器原生兑换，控制台类客户端必须经集群内 BFF 兑换
通知与触达域	被调	发送短信 / 邮件验证码
运营管理域	被调	运营人员身份与权限范围（独立主体，不复用终端用户身份）
审计	被调	注册 / 登录 / 改密 / 换绑等动作落安全日志，门户可查近 90 天
全部业务域	依赖方	一切需要登录态的接口以本域签发令牌为唯一身份依据

令牌契约（对外提供）

- 门户访问令牌有效期 30 分钟，刷新令牌 7 天；签名算法 HS256 共享密钥（供统一网关复用校验）；载荷含 `sub / type / roles / exp / iat / jti / typ=portal`。
- 令牌区分类型：门户令牌（`typ=portal`）用于业务鉴权；过渡令牌（`typ=wechat_bind`）仅可用于绑定手机流程，业务鉴权一律拒绝。

3. 数据模型

3.1 域内关联总览

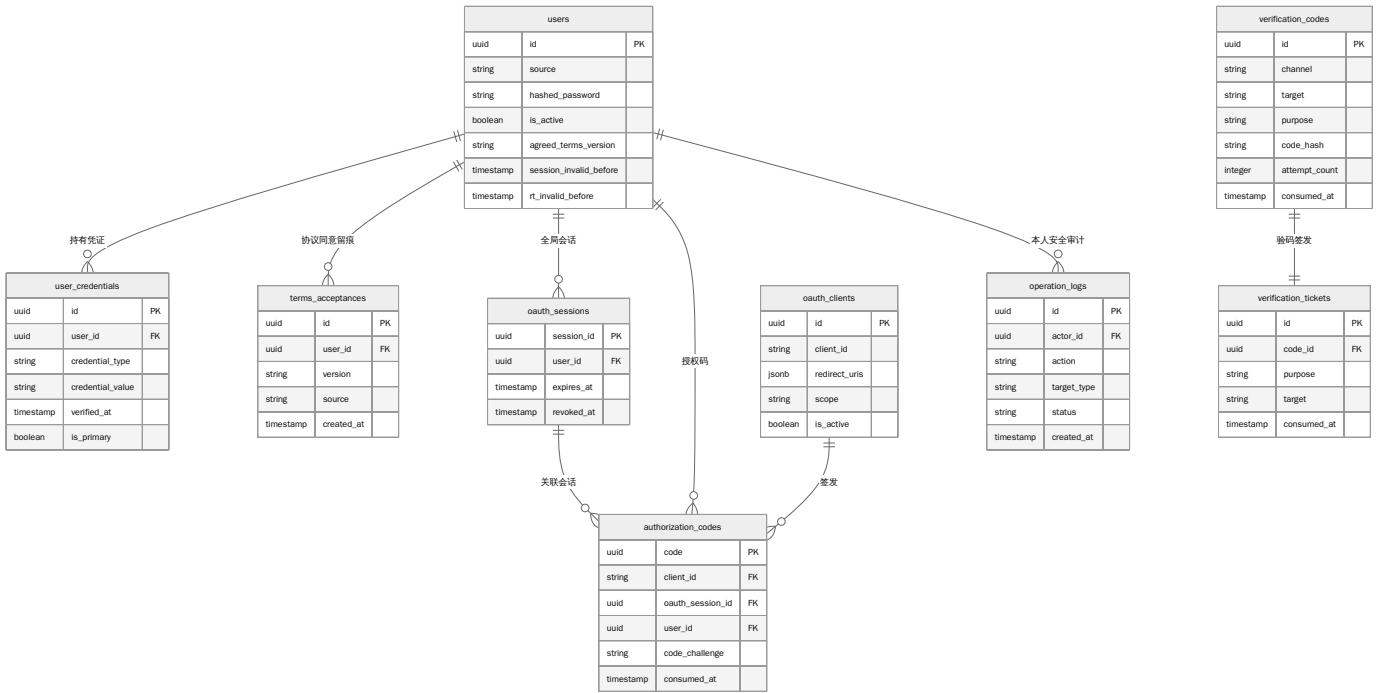


图 1

独立表（无外键，自持一次性令牌 / 按来源 IP 与身份串计数，随 TTL 或周期清理收敛）：`captchas` · `slider_captchas` · `login_throttles`

3.2 域间引用

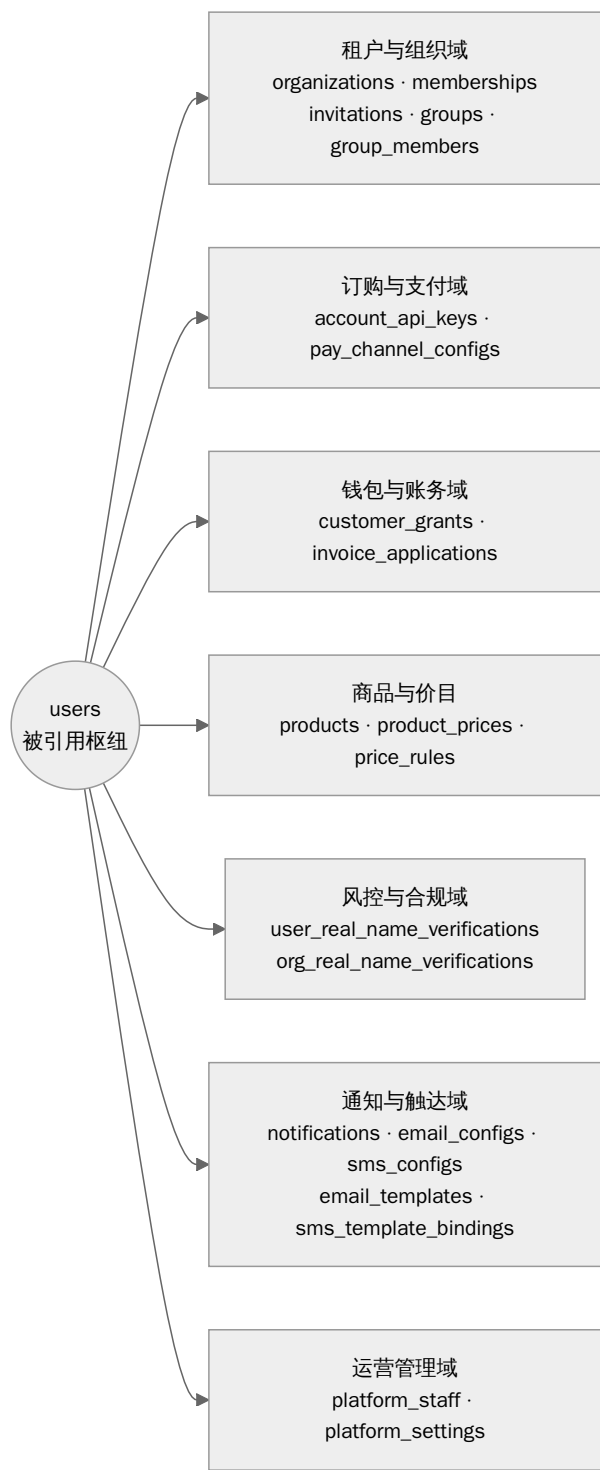


图 2

本域为最底层身份域，不引用其他域的表，对外只做被引用方；上图为 `users` 的下游引用分布。

3.3 表清单 (12 表)

表	职责	关键字段
<code>users</code>	登录身份主体 (一人一行)	<code>source</code> (注册来源)、 <code>display_name</code> 、 <code>hashed_password</code> (可空——外部身份账号无密码)、 <code>is_active</code> 、登录安全 (<code>failed_login_count</code> / <code>locked_until</code> / <code>last_login_at</code> / <code>last_login_ip</code> / <code>last_login_user_agent</code>)、会话水位 (<code>session_invalid_before</code> / <code>rt_invalid_before</code>)、协议留痕 (<code>agreed_terms_version</code> / <code>agreed_terms_at</code>)、实名回写位 (<code>identity_verified</code> / <code>verification_method</code> / <code>verified_at</code> / <code>identity_level</code>)
<code>user_credentials</code>	联系方式与外部身份统一凭证	<code>credential_type</code> (<code>phone</code> / <code>email</code> / <code>account</code> / 微信 <code>openid</code> / <code>unionid</code>)、 <code>credential_value</code> 、 <code>verified_at</code> (NULL = 未验证)、 <code>is_primary</code> ；约束：已验证凭证全局唯一、每用户每类型联系方式至多一行、账号名同人唯一、外部身份凭证建行即已验证
<code>terms_acceptances</code>	协议同意历史 (append-only)	<code>user_id</code> 、 <code>version</code> 、 <code>source</code> (注册 / 重新确认 / 微信首登)、 <code>ip</code> 、 <code>user_agent</code> 、 <code>created_at</code>
<code>verification_codes</code>	短信 / 邮箱一次性验证码	<code>channel</code> 、 <code>target</code> 、 <code>purpose</code> (注册 / 登录 / 换绑 / 重置密码)、 <code>code_hash</code> (哈希存储)、 <code>expires_at</code> 、 <code>consumed_at</code> 、 <code>attempt_count</code> 、 <code>ip</code>
<code>verification_ticks</code>	验证码校验通过后的一次性凭证	<code>code_id</code> 、 <code>purpose</code> 、 <code>target</code> 、 <code>expires_at</code> 、 <code>consumed_at</code> ； <code>purpose + target</code> 绑定防跨场景重放
<code>captcha_has</code>	图形验证码	<code>answer</code> 、 <code>expires_at</code> 、 <code>consumed_at</code> 、 <code>attempt_count</code> 、 <code>ip</code>
<code>slider_captchas</code>	滑块验证	<code>token</code> 、 <code>answer_x</code> / <code>answer_y</code> (仅服务端)、 <code>secret_key</code> 、 <code>expires_at</code> 、 <code>verified_at</code> 、 <code>failed_attempts</code> 、 <code>ticket</code> 、 <code>consumed_at</code> 、 <code>ip</code>
<code>login_throttles</code>	登录失败限流计数器	<code>scope</code> (<code>ip</code> / <code>identity</code>)、 <code>subject</code> (IP 或归一化身份串)、 <code>window_start</code> 、 <code>failure_count</code> ；(<code>scope</code> , <code>subject</code> , <code>window_start</code>) 唯一
<code>oauth_clients</code>	统一登录客户端注册	<code>client_id</code> 、 <code>client_name</code> 、 <code>redirect_uris</code> (精确匹配白名单)、 <code>frontchannel_logout_uri</code> 、 <code>scope</code> (含 <code>portal</code> 与否决定兑换形态)、 <code>is_active</code>
<code>oauth_sessions</code>	身份提供方全局会话	<code>session_id</code> 、 <code>user_id</code> 、 <code>expires_at</code> (服务端硬上限 7 天, 固定不滚动)、 <code>revoked_at</code>
<code>authorization_codes</code>	一次性授权码	<code>code</code> 、 <code>client_id</code> 、 <code>redirect_uri</code> 、 <code>code_challenge</code> + <code>code_challenge_method</code> (PKCE)、 <code>oauth_session_id</code> 、 <code>user_id</code> 、 <code>nonce</code> 、 <code>expires_at</code> 、 <code>consumed_at</code>

表	职责	关键字段
operation_logs	本人视角安全审计（与业务变更同事务写入）	actor_id、action、target_type、target_id、status、detail、operator_ip、operator_user_agent、created_at

3.4 设计约束

- 验证码答案、密码一律哈希存储；滑块缺口坐标仅存服务端，不下发。
 - 图形码 / 滑块 / 验证码状态落库（非进程内存），保证多副本共享；消费一律走原子抢占（`UPDATE ... RETURNING`），杜绝并发双消费。
 - `verification_tickets` 与协议同意表均为一次性 / 只增语义：前者消费即失效，后者永不覆盖（当前生效快照另存于 `users`）。
-

4. 核心流程

4.1 注册

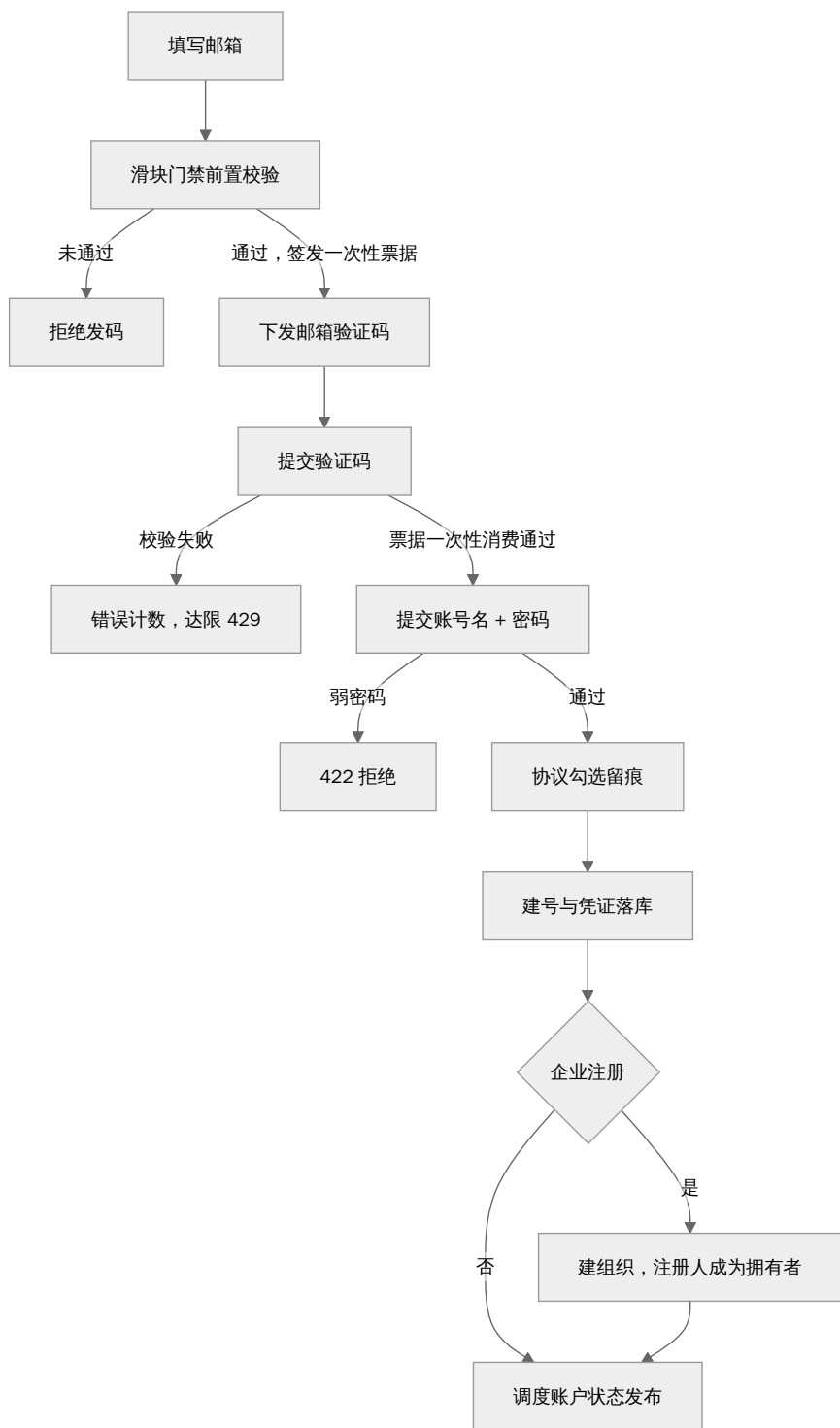


图 3

邮箱验证码校验通过（一次性票据）→ 提交账号名 + 密码（弱密码 422）→ 协议勾选留痕 → 建号。企业注册额外建组织并由注册人成为拥有者。注册成功即调度账户状态键发布（新账号准入语义见账户状态与准入域）。

4.2 登录（含门禁）

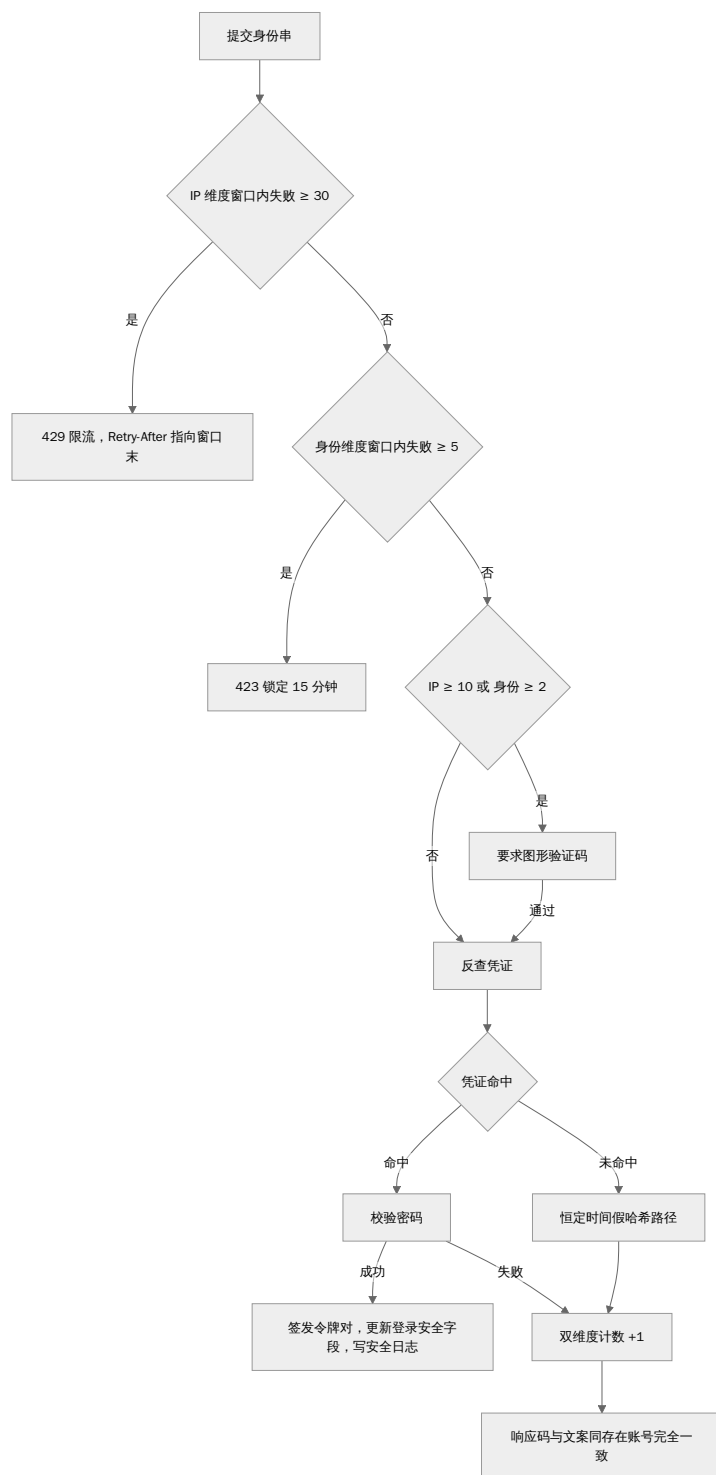


图 4

- 门禁判定在身份反查之前，按提交串计数；IP 维度与身份维度双维度独立计数，任一维达阈值即触发对应拦截。
- 反查凭证：命中则校验密码，未命中走恒定时间假哈希路径（耗时与命中一致）。
- 成功：签发令牌对、更新登录安全字段、写安全日志。
- 失败：双维度计数 +1（含「账号不存在」的尝试），响应码与文案与存在账号完全一致。
- 限流采用固定窗口（窗口边界对齐），计数行按 `(scope, subject, window_start)` 唯一、单条 UPSERT 原子自增；过期行由周期清理收敛（身份串完全由攻击者可控，无清理可被低成本撑爆）。

- 来源 IP 采信规则：仅当直连对端落在可信代理名单内才采信转发头，否则一律回落直连地址。

4.3 会话与令牌生命周期



图 5

- 访问令牌 30 分钟；刷新令牌 7 天、单活轮换——轮换以条件更新（CAS）原子抢占水位，仅当水位不晚于所呈令牌签发时间才放行；重放 / 他端已消费 一律 401。多端语义：最后轮换者存活，其余端待访问令牌自然过期。
- 双水位失效：`session_invalid_before`（批量失效访问 / 刷新令牌）与 `rt_invalid_before`（刷新令牌单活水位）。
- 触发失效的动作：修改密码、重置密码、换绑联系方式、注销、首次设置密码。

4.4 换绑与第二因子

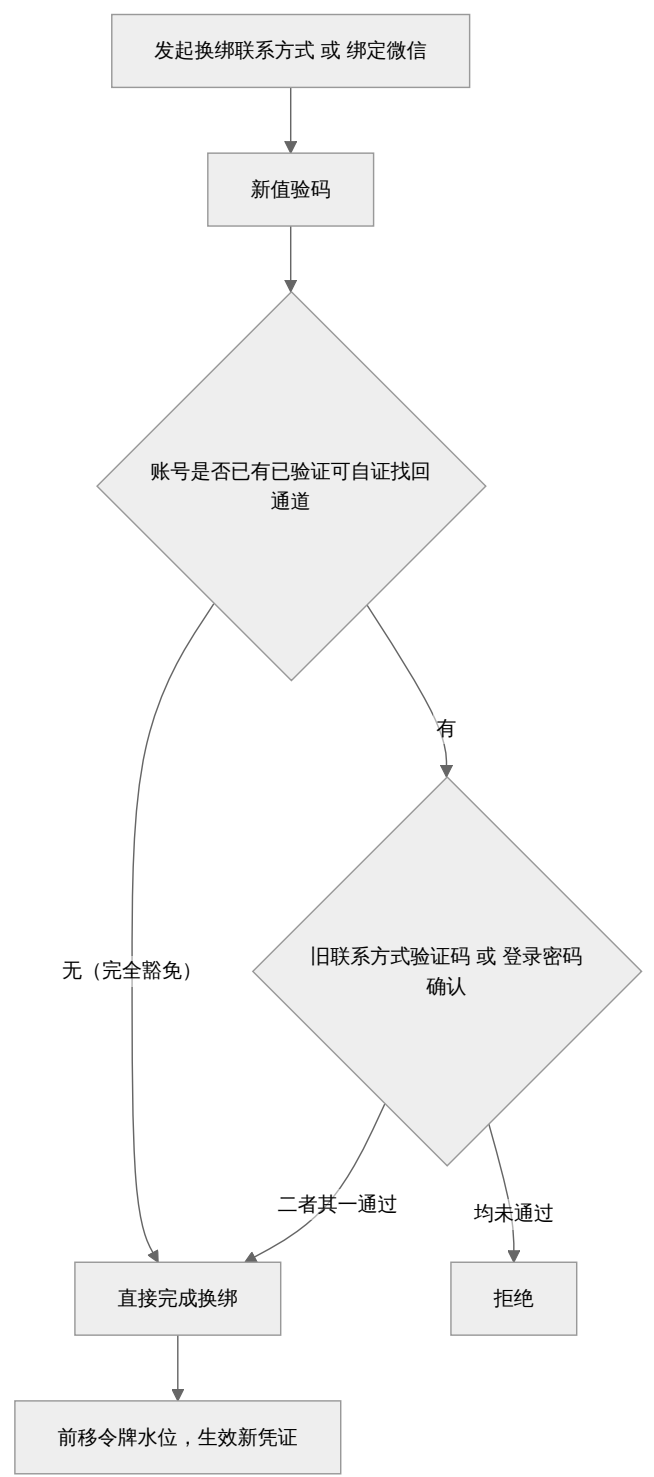


图 6

换绑邮箱 / 手机需新值验码 + 旧凭证占有因子二选一（旧联系方式验证码，或登录密码确认）。授权依据是「账号是否已有可自证的找回通道」，而非「该类型有无已验证旧值」——仅当账号不存在任何已验证联系方式时才豁免第二因子；否则会给攻击者开一条免第二因子的找回通道生产线。绑定微信同为新增登录入口，口径与此一致。

4.5 单点登录（授权码 + PKCE）

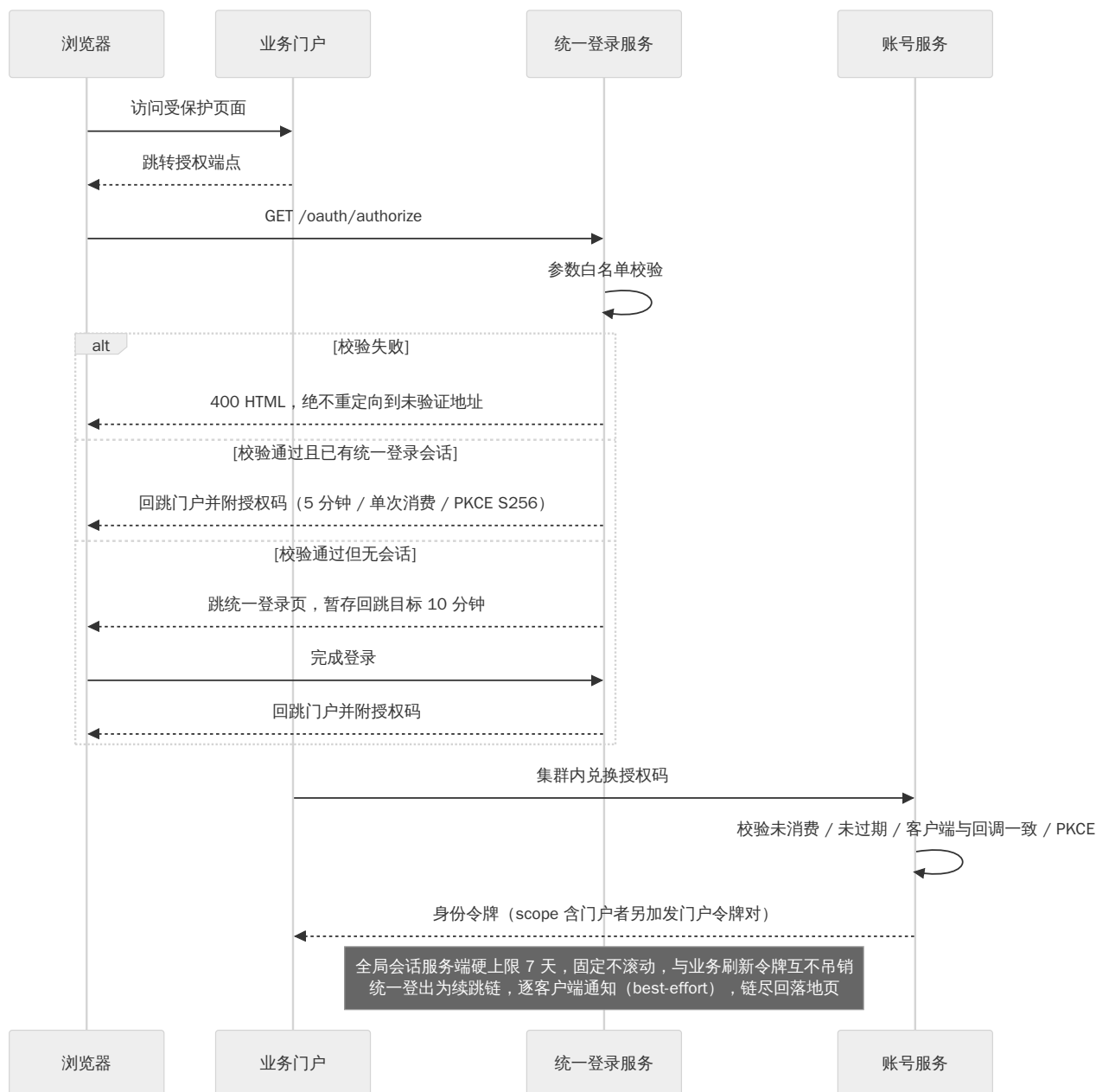


图 7

- 授权端点：参数白名单校验，失败一律 400 HTML，绝不重定向到未验证地址。
- 兑换在集群内完成：校验授权码未消费、未过期、客户端与回调一致、PKCE。
- `scope` 含门户的客户端，兑换时另加发门户令牌对。
- IdP 全局会话服务端硬上限 7 天、固定不滚动，与业务刷新令牌完全独立、互不吊销。
- 统一登出为续跳链：逐客户端通知 (best-effort)，链尽回落地页；无会话同样走清 cookie 与续跳（幂等）。
- 撤销时机：统一登出、重置密码、修改密码、首次设置密码均撤销该用户全部未撤销 IdP 会话——只前移门户水位不撤销 IdP 会话时，失窃会话持有者仍可经授权端点静默续出客户端令牌，改密自救失效。

4.6 微信扫码

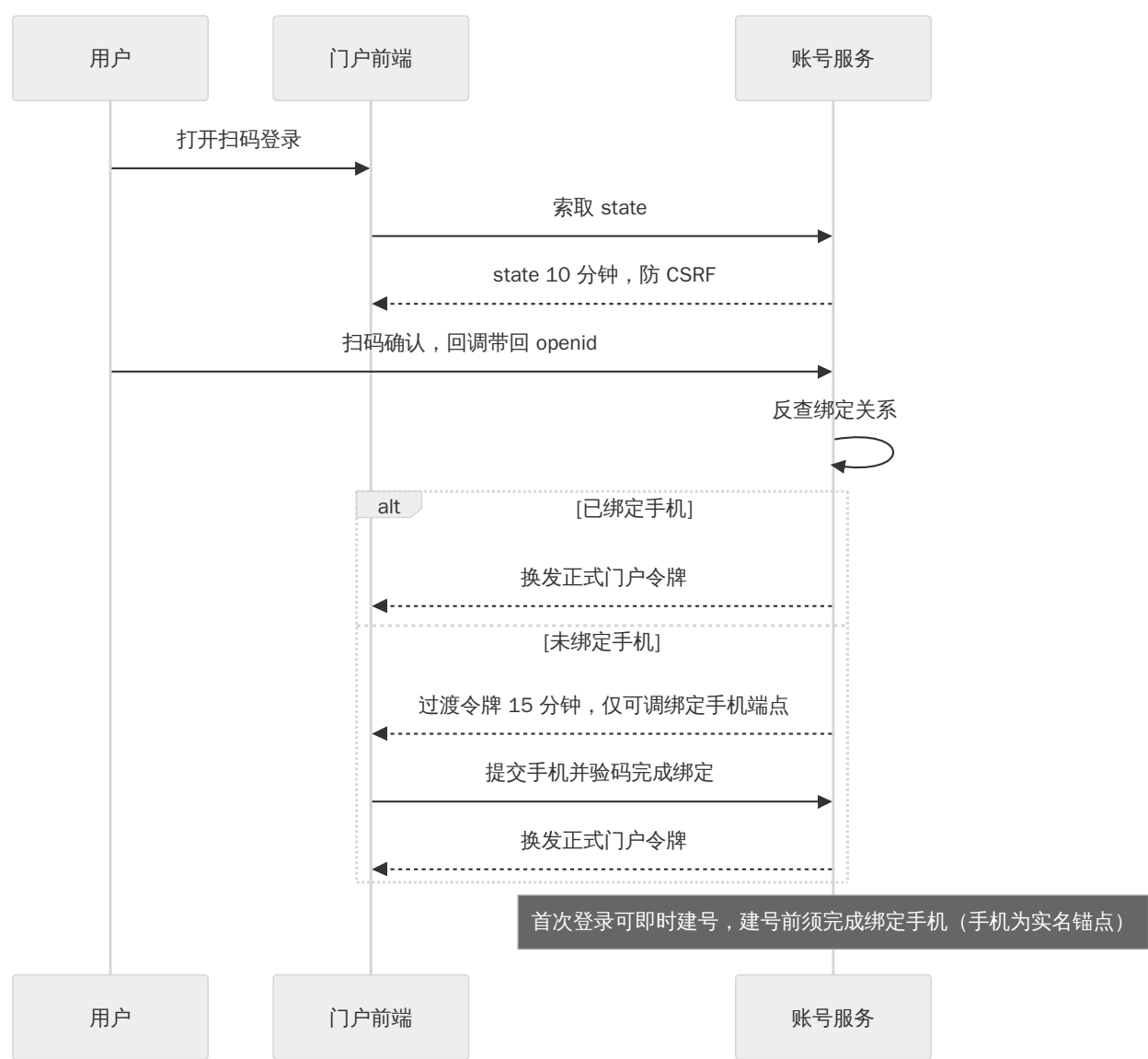


图 8

签发 `state` (防 CSRF, 10 分钟) → 回调换 `openid` → 已绑定手机直接登录; 未绑手机发过渡令牌 (15 分钟, 仅可调绑定手机端点) → 绑定手机后换发正式门户令牌。首次登录可即时建号, 建号前须完成绑定手机 (手机为实名锚点)。

4.7 人机验证门禁

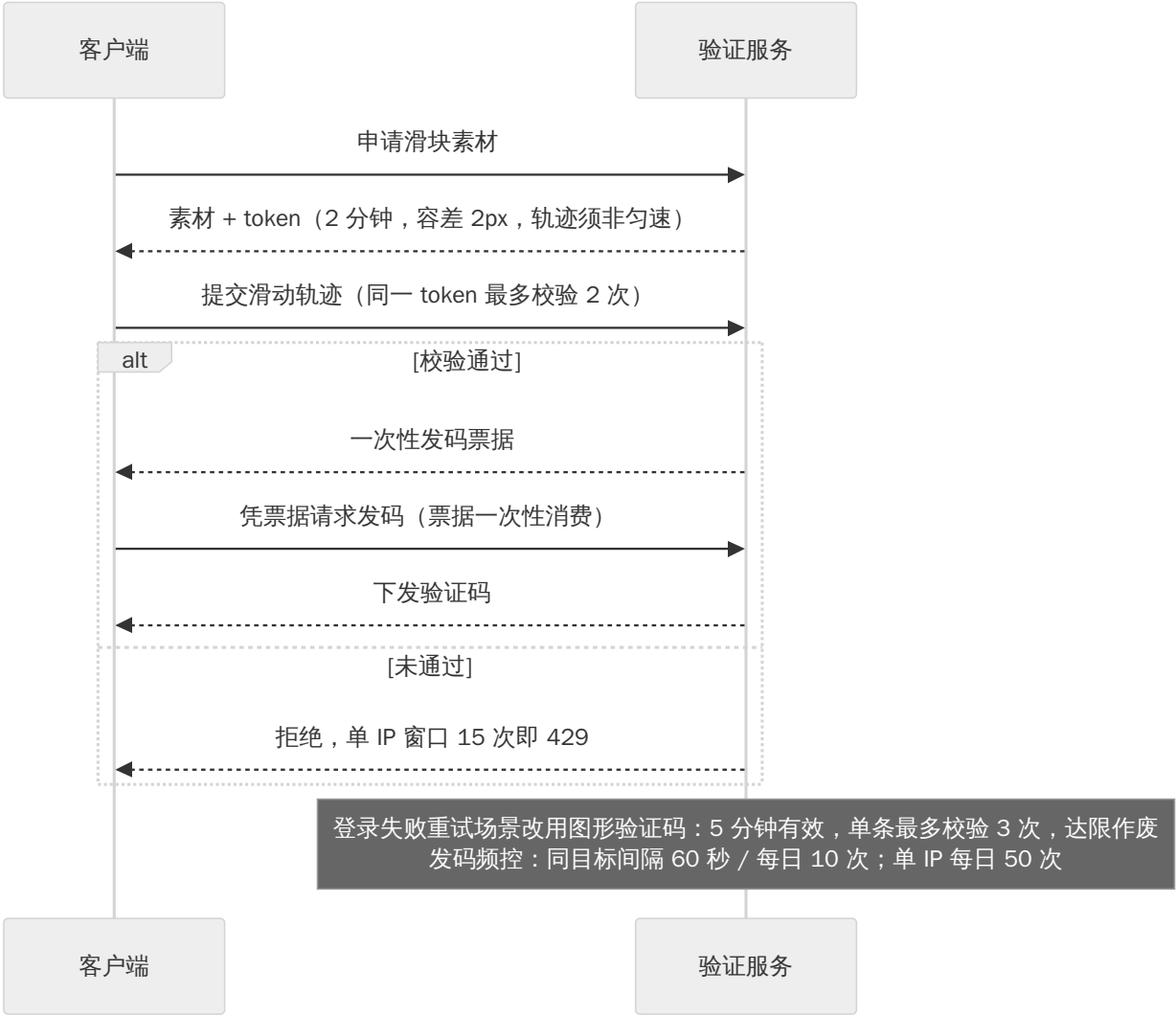


图 9

- 图形验证码：用于登录失败重试场景；5 分钟有效、单条最多校验 3 次，达限作废。
- 滑块验证：验证码发码前的前置门禁；`token` 2 分钟有效、横向容差 2px、轨迹须非匀速、单 `token` 最多校验 2 次；校验通过签发一次性票据，发码凭票据一次性消费。
- 阈值设计目标：单 `token` 通过率 < 5% (容差 × 尝试数 / 取值域)。
- 发码频控：同 `target` 重发间隔 60 秒、每日 10 次；单 IP 每日 50 次；素材生成单 IP 每日 200 次。
- 失败频控：图形码单 IP 窗口内错误 20 次 → 429；滑块单 `token` 失败上限 2 次、单 IP 窗口内 15 次 → 429。

4.8 协议同意与门禁

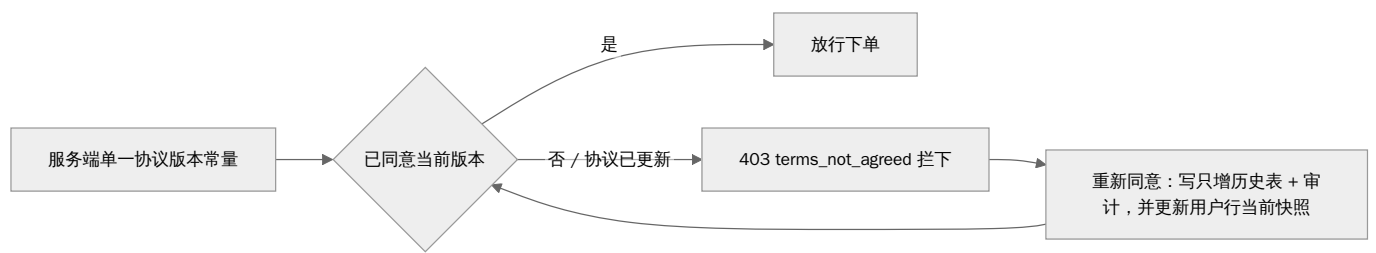


图 10

协议版本由服务端单一常量定义；同意写入只增历史表 + 审计，并在用户行留「当前生效」快照。版本不一致（未同意 / 协议已更新）时，下单被 `403 terms_not_agreed` 拦下——门禁开关可按环境关闭（灰度 / 演示）。

5. 接口契约

5.1 门户认证 (`/api/account/auth`)

方法	路径	语义
GET	<code>/terms/{kind}</code>	协议文本（用户协议 / 隐私政策 / 服务等级协议），公开可读（注册页未登录即需展示）
GET	<code>/captcha</code>	签发图形验证码；联调开关开启时附答案（生产禁用）
GET	<code>/login-options</code>	登录页可用登录方式（公开）：手机通道可用性与微信登录可用性各自独立判定，任一开关来源不可用时该入口不下发
POST	<code>/verification-code/send</code>	发码：落库 + 渠道发送；返回真实发送结果（渠道未配置时不谎报成功）；联调开关开启时附明码
POST	<code>/verification-code/verify</code>	校验验证码 → 签发一次性票据（供注册 / 换绑等后续提交）
POST	<code>/login</code>	密码登录（任一已验证联系方式或账号名）；成功签发门户令牌对
POST	<code>/login-or-register</code>	验证码登录：已注册 → 等价验证码登录；未注册 → 404 <code>contact_not_registered</code> （验证码已证明联系方式所有权，无枚举风险），前端引导跳独立注册页
POST	<code>/reset-password</code>	忘记密码：凭一次性票据（ <code>purpose=reset_password</code> ）反查用户； <code>target</code> 取自票据不信任前端；重置后旧令牌全部失效
POST	<code>/refresh</code>	刷新令牌（单活轮换）：消费即失效，重放 401
POST	<code>/sso/exchange</code>	单点登录授权码兑换（浏览器原生形态）：仅服务注册 <code>scope</code> 含 <code>portal</code> 的客户端；PKCE S256 校验 + 授权码单次消费
GET	<code>/me</code>	当前登录用户（门户鉴权）

5.2 注册与账号管理 (/api/account)

方法	路径	语义
POST	/register	个人注册 (账号名 + 密码 + 协议门禁)
POST	/register/organization	企业账户注册 (注册人自动成为组织拥有者)
POST	/register/invite	邀请注册
POST	/me/accept-terms	同意当前协议版本：写同意历史 + 审计；版本以服务端当前值为准 (不接受客户端传旧版本)
GET / PATCH	/profile	读取 / 更新资料 (昵称即改即存；头像仅接受预置头像路径)
GET	/preset-avatars	预置头像列表
POST	/password	修改密码 (需旧密码)
POST	/password/set	首次设置密码 (仅无密码账号；凭验证码，target 服务端解析)
POST	/username/set	首次设置账号名 (一期不可改)
POST	/contact/change	换绑 / 首次绑定邮箱或手机 (新值验证码 + 旧凭证第二因子)
POST	/contact/verify	重新验证已绑定的联系方式
GET	/security-logs	安全日志 (近 90 天登录 / 改密 / 换绑 / 注册 / 组织动作)
DELETE	/account	注销账号

5.3 人机验证 (/api/account/slider-captcha)

方法	路径	语义
GET	/generate	生成滑块素材 (随机背景 + 缺口)；返回 token + 图片 + secret_key
POST	/verify	校验拖动 (坐标容差内 + 轨迹非匀速) → 一次性票据；失败累计达上限即作废该 token

5.4 微信扫码 (/api/account)

方法	路径	语义
GET	/auth/wechat/qrcode	签发扫码参数 (含防 CSRF 的 state)；联调模式返回内置授权页二维码，真实模式返回授权跳转地址
GET	/wechat/qrcode-status	PC 端轮询扫码状态 (联调模式)：待扫 / 已扫 / 已确认 / 已过期
GET	/wechat/mock-auth	联调授权页 (仅联调模式)
POST	/wechat/mock-auth/scan	联调：置为已扫

方法	路径	语义
POST	<code>/wechat/mock-auth/confirm</code>	联调：确认 / 取消
POST	<code>/auth/wechat/callback</code>	授权回调：已绑定 → 直接登录；未绑手机（含首次登录建号）→ 发过渡令牌
POST	<code>/auth/wechat/bind-phone</code>	绑定手机（短信验证码）后换发正式门户令牌
POST	<code>/wechat/bind</code>	已登录账号绑定微信；该微信已被占用 → 409；需过第二因子

5.5 统一登录（IdP，非 `/api/account` 前缀）

方法	路径	语义
GET	<code>/oauth/authorize</code>	授权端点：客户端 / 回调地址 / PKCE 白名单校验 → 会话判定 → 发码或跳登录页
POST	<code>/oauth/token</code>	令牌端点：JSON 请求，仅集群内可达（不进公网入口）；错误统一标准 OAuth 错误体
GET	<code>/.well-known/jwks.json</code>	公钥集合；客户端缓存，遇未知 <code>kid</code> 强制刷新一次
GET	<code>/oauth/logout</code>	统一登出（幂等）：清 cookie + 续跳链
GET	<code>/oauth/logout/continue</code>	登出续跳中转：验签 → 弹出已处理端点 → 重签跳下一个
GET / POST	<code>/login</code>	IdP 登录页（服务端渲染最小实现）：账密 + 回跳恢复；已登录直接续跳
GET	<code>/oauth/returnto</code>	单点登录回跳恢复：门户登录成功后整页跳本端点，服务端签发并验签的目标地址，无开放跳转面
GET	<code>/loggedout</code>	统一登出落地页

6. 关键约束与验收标准

6.1 约束

- 1. 不得泄露账号存在性：门禁判定先于身份反查，且不存在账号与存在账号走完全相同的计数与响应路径；不存在用户路径跑恒定时间假哈希，防时序侧信道。
- 2. 凭证不可逆与最小化：密码、验证码答案哈希存储；证书 / 密钥类凭据经环境注入，不入库、不回显。
- 3. 状态多副本共享 + 原子消费：图形码 / 滑块 / 验证码状态落库；消费一律 `UPDATE ... RETURNING` 抢占，并发双消费恒至多一次成功。
- 4. 令牌单活：刷新令牌消费即失效、重放 401；改密 / 换绑 / 注销须同时前移门户水位并撤销 IdP 会话，二者缺一即构成自救失败。

- 5. 开放跳转面为零：授权端点参数错误一律 400 HTML，不 302；回跳目标由服务端签发并验签，不接受客户端传入。
- 6. 令牌端点不可公网直达：令牌兑换端点仅集群内可达，浏览器全程不直连。
- 7. 登录页无注入面：服务端渲染登录页不回显任何请求输入，错误统一静态文案。
- 8. 能力开关 fail-closed：手机通道、微信登录、单点登录任一能力未就位时其入口不下发；联调模式凭据在生产一律拒绝（白名单判定为「环境 = 开发」，非「≠ 生产」）。
- 9. 限流覆盖不存在账号：两维度计数均覆盖「账号不存在」的尝试，否则攻击者可对任意账号试 2 次绕过图形码与锁定。

6.2 验收标准

项	验收口径
注册闭环	邮箱验证码注册成功；弱密码 422；未勾选协议不可注册；企业注册后注册人角色为拥有者
登录四通道	密码（联系方式 / 账号名）、验证码、一键注册登录、微信扫码各自可登录并签发门户令牌
门禁分级	IP 达 10 次失败→要求图形码，达 30 次→429 且带 Retry-After；身份达 2 次→要求图形码，达 5 次→423 且 15 分钟后自动解锁
防枚举	存在 / 不存在账号在门禁各档位返回完全相同的状态码与文案；响应耗时无可利用差异
令牌单活	同一刷新令牌连续两次使用，第二次 401；多端登录仅最后轮换者存活
改密自救	改密后旧访问 / 刷新令牌全部 401，且原 IdP 会话不可再换出客户端令牌
换绑门禁	已有任一已验证联系方式时，仅凭新值验证码换绑必须被拒（要求第二因子）
单点登录	授权码 5 分钟内单次消费；重放 / PKCE 不匹配 / 回调地址不符一律拒绝；登出链逐客户端生效且幂等
人机验证	滑块坐标超容差或匀速轨迹必失败；单 token 超尝试上限即作废；发码须凭未消费票据
频控	同 target 60 秒内重发被拒；超日限发码被拒；单 IP 超素材生成日限被拒
协议门禁	未同意当前版本时下单 403 terms_not_agreed；同意后放行；同意历史可查且不可改写
注销	注销后该身份全部令牌失效、不可再登录，关联凭证按级联清理

6.3 待后续设计细化的开放项

- 账号名一期不可改，后续是否开放改名（含改名冷却与占用释放）。
- 手机通道开放后，手机与邮箱同时存在时的主联系方式重算规则。
- 微信扫码在真实凭据下的多应用（公众号 / 开放平台）账号归并口径。