ODNCA-STD-007 · SNSP — SNS Registration Protocol

The registrar claim flow: check, prepare, inscribe, complete

Status: v1.0 — Adopted

Normative reference implementation: ORDnet Registry v16.2x (`/check` , `/claim/prepare` , `/claim/complete` , order flow, owner-sync webhook).

Layer note: this protocol describes how a *registrar service* assists registration. The chain itself needs none of it — a raw conformant inscription (ODNCA-STD-003) is always a valid path. SNSP is what makes a registrar's assisted path safe, fair, and verifiable, and what a second registrar implements to interoperate.

1. Flow overview

```
GET  /check?name=…              is it available, and why not if not
POST /claim/prepare             gates + price + signed claim record
        ↓ wallet inscribes the record (1-sat, OP_RETURN tag)
POST /claim/complete            { claim_id, txid } — registrar books it
        ↓ chain → indexer → owner-sync            (~10 min propagation)
resolver serves the name        (no_holder during propagation, STD-001)
```

2. Availability (`/check`)

Answers with a machine reason, never a bare no: `tld_retired` , `reserved` , `flagged` , taken (registry), taken on-chain, or available with its price. Two properties are normative:

1. **Portal validation is registrar policy.** The reference portal enforces `a–z0–9` , 1–63 characters — deliberately stricter than chain validity (ODNCA-STD-003 §6) per the layer rule. A second registrar MAY choose otherwise within chain validity.
2. **The on-chain gate is fail-safe.** Availability is confirmed against the chain before any claim is signed; if chain verification is unreachable the answer is an honest `503 verification_unavailable` — never an optimistic yes.

3. Preparing a claim (`/claim/prepare`)

The registrar runs the gates in order — retired TLD, reserved, flagged, already registered, on-chain taken — then handles payment state, then signs.

3.1 The claim lock

One name, one active claim at a time: a prepared claim locks the name for a TTL. The same wallet asking again receives **the same signed record back (idempotent)**; a different wallet receives `name_locked` . No race, no double-sell, and an abandoned claim simply expires.

3.2 Payment shapes

- **Free names:** rate-limited (reference: 3 free registrations per wallet per 24h) — anti-drain, not gatekeeping.
- **Order flow:** pay first (order reaches `paid_awaiting_inscribe`), then prepare; the claim binds to the order and **only the order owner's wallet** may have the record signed (`wallet_mismatch` otherwise).
- **Direct flow:** the payment rides as a P2PKH output **in the same transaction as the inscription** — no order, no webhook, no trust window; reconciliation verifies payment on-chain after the fact, because the transaction itself contains both the claim and the money.

3.3 The signed claim record

The registrar signs a registration record binding **the name AND the destination wallet** — a signature over `{op:"reg", name, recipient}` by the published issuer key. Binding both is what makes the record replay-proof: it cannot be lifted and inscribed for another name or to another owner. The prepare answer hands the wallet everything needed to inscribe: the record JSON as payload, content type, the service-fee output, and the `ORDnet.io` OP_RETURN tag. Signing runs fail-safe: if the issuer key is not loaded and verified against the expected public key, the endpoint answers 503 rather than issuing unsigned records.

4. Completing (`/claim/complete`)

The wallet reports `{claim_id, txid}` ; the registrar books the registration **chain-first** — the on-chain fact leads, the administrative rows follow. From here the pipeline is ODNCA-STD-001 §4.4: chain → indexer → owner-sync → resolver, with the honest `no_holder` answer during propagation (~10 minutes) instead of a fabricated one.

5. Verification: registered vs. verified

Two distinct facts, deliberately separate:

- **Valid** (chain layer): the inscription satisfies ODNCA-STD-003. True for any conformant claim, assisted or raw.

- **Verified** (coordination layer): the claim carries a correct issuer signature per §3.3, checked independently by the indexer and recorded alongside — never instead of — validity.

Verification marks registrar-issued claims and underpins services that need an administered counterparty (custody semantics, premium services, the owner-sync scope). It never affects validity or first-seen-wins: an unverified raw claim that is first still wins the name.

---

6. Second-registrar conformance

A registrar interoperates when it: implements §§2–4 with its own issuer key (published per the operator listing and rotatable per ODNCA-STD-001 §7.2); honours the gates of §3 including POL-001 TLD categories; keeps its own-policy restrictions honestly labelled per POL-002; and produces claims that the reference indexer marks valid — and, under its published key, verified. The claim lock is per-registrar; cross-registrar races are settled where they always are: on the chain, first seen wins.
