ODNCA-STD-004 · The Root Registry

Canonical namespaces, TLD activation, and the ownership-proof backbone

Status: v1.0 — Adopted
Launch scope: BSV / SNS. The format is chain-agnostic by design; additional chains (ambition: BTC, DOGE) join under §8 without format changes.
Companion documents: ODNCA-STD-001 (Resolution), ODNCA-STD-002 (Parameters), ODNCA-STD-003 (SNS-NAME-1), ODNCA-POL-001 (TLD Fair-Use).
Normative reference implementation (certificate layer): ODNCA State Commitments v1.2.

1. Purpose and the three layers

The Root Registry answers one question with authority: **for a given TLD, which registry on which chain is canonical?** It is the collision-protection layer — the reason `name.web3` means one thing in the coordinated ecosystem even if look-alike systems exist elsewhere.

It is deliberately **not a new database.** The ecosystem already has its layers, and the Root Registry is the thin, signed coordination sheet on top:

LAYER	LIVES AT	ROLE
Fact layer — the indexer	search.ordnet.io	Knows every claim: what, by whom, when. Neutral; indexes everything.
Coordination layer — the Root Registry	this document + the signed root list	Which namespaces and TLDs are recognised, with their anchors and activation facts.
Service layer — registrar registry	whois.ordnet.io	Commercial services: txid↔domain coupling, premium tiers, content measures (POL-003).

Everything in the coordination layer is either an announced decision (anchored on-chain) or derivable from the fact layer given those decisions. There is no third kind of entry.

2. Namespace entries

A **namespace** is a (chain, protocol) pair with published, frozen rules. Its identity is not a name or a number but its **ruleset fingerprint**:

```
namespace-id := <chain> ":" <protocol> ":" <ruleset-hash>
e.g.            bsv:sns:<sha256 of the frozen SNS ruleset document>
```

A registry *is* its rules; its rules are therefore its fingerprint. Change the rules and it is demonstrably a different system. This is not aspiration but deployed practice: every state commitment the ecosystem publishes already embeds the ruleset hash (§6), binding every ownership certificate to the exact rules under which it is true.

A namespace entry records: chain, protocol, ruleset hash, the ruleset document location, the conformance vector set, and the operating indexer/resolver endpoints. At launch there is one namespace: **BSV / SNS**.

3. TLD entries and the activation rule

3.1 Birth is a decision, not a discovery

A TLD's validity does not begin with its first claim — it begins with its **activation**. The rule, normative for every conformant indexer and resolver:

> **A claim under a TLD is valid only at or after that TLD's activation height.** Claims inscribed before activation are void and confer no rights, regardless of first-seen-wins — first-seen-wins operates only among valid claims.

Worked example: `.claude` activates at height 975,000. A claim of `fable.claude` at height 963,000 is void; a claim at 975,001 is the first valid claim and wins. Nobody can squat a TLD before it exists.

3.2 Activation is an on-chain fact

Each activation is inscribed by ODNCA as a signed **activation record**: TLD, activation height, category (per POL-001), effective status. The transaction id of this inscription is the TLD's birth certificate, and the basis of its registry-id:

```
tld-id := <chain> ":" <protocol> ":" <activation-txid>
e.g.      bsv:sns:<txid of the .claude activation inscription>
```

The five TLDs active at adoption (`web3, bitcoin, crypto, blockchain, ordnet`) and the retired pair (`bsv, bitcoinsv`) receive retro-activation records at adoption of this standard, carrying their factual effective heights. Category changes (retire, withhold-release, POL-001 §5) are inscribed the same way, each referencing its predecessor — decisions form their own auditable chain.

3.3 Derived anchors

Given a published activation height, the **first-valid-claim txid** under a TLD becomes an objective, recomputable fact via the indexer rules — it is recorded in the TLD entry as an informational anchor. Decisions first, facts derived after; anyone can check the derivation.

3.4 TLD entry fields

`tld`, `tld-id` (§3.2), `namespace-id` (§2), `category` (Active/Retired/Withheld, POL-001), `activation_height`, `activation_txid`, `first_valid_claim` (derived, §3.3), `resolver` and `registry` endpoints, changelog references.

---

4. Collision rule

Where the same TLD label exists in multiple systems or on multiple chains, **the recognised entry is canonical for the coordinated ecosystem**; all others are Unrecognised (POL-001 §2) and resolve nowhere in it. Recognition follows POL-001's fair-use screening — including the possibility that the fair answer is to recognise an *external* party's system as the canonical home of a label rather than compete with it (the `.1sat` principle at TLD scale).

---

5. Publication

The root list is published in two forms from one source:

- **Machine form:** signed JSON at the root endpoint, using the same key infrastructure and rotation procedure as ODNCA-STD-001 §7.2. Resolvers already load their TLD sets live from the registry (single source); the root list is the authority those sets are derived from.
- **Human form:** the registry pages on the ODNCA site, generated from the same data.

Every change is a dated changelog entry in ODNCA-STD-002, and — per §3.2 — anchored on-chain. A silent edit to the root list is thus detectable three ways: signature, changelog, chain.

---

6. The ownership-proof backbone (state commitments)

Recognition tells you *which* registry is canonical; the commitment layer proves *what that registry says*, compactly, to anyone. Operated under the ODNCA flag (ODNCA publishes the rulesets and the keys; the math validates, ODNCA coordinates):

1. **Commit chain.** The indexer's canonical state is periodically frozen as a merkle tree; the root is inscribed on-chain as a commit record carrying the state height, name count, the **ruleset hash** (§2), and a `prev` link to the preceding commit — commitments chain into an auditable history per protocol (`sns-commit`, and `opns-commit` when OpNS state lands).
2. **Certificates.** Any holder obtains a **Certificate of Ownership**: their leaf (name, origin, current outpoint, holder script, state height), the merkle path to the committed root, and the commit's on-chain anchor. Verification is offline and dependency-free with the published reference verifier.
3. **Minted certificates.** The proof JSON can itself be inscribed (`sns-cert`) onto the holder's address — an on-chain, self-contained ownership attestation.
4. **Pinned constants.** Leaf and node hashing, key order, and the commit format are frozen by golden-vector tests; changing any of them invalidates every certificate in the wild and therefore constitutes a new namespace (§2).

---

7. Conformance

An implementation conforms to this standard when it (a) rejects claims below their TLD's activation height and accepts the published activation records as the source of those heights, (b) derives the same first-valid-claim anchors from the same chain and parameters, and (c) reproduces the commitment layer's golden vectors. As everywhere in ODNCA: run the vectors, match the answers.

---

8. Adding a chain

A new namespace (e.g. BTC or DOGE, when their time comes) is admitted when it brings: deterministic claim-validity rules published as a frozen, hashed ruleset; a conformance vector set; at least one operating indexer implementation exposing canonical state; and a commitment path per §6. The Root Registry format needs no change — a new namespace is one new entry with its own fingerprint, and its TLDs follow §3 under its own activation records.

Annex — Implementation notes

- **Per-TLD activation enforcement.** The activation rule (§3.1) is enforced today by the fact that all active TLDs predate all indexed claims under them. Before the first *new* TLD activates, the indexer's rule engine must support per-TLD minimum heights — the mechanism exists in spirit (the BSVmap minimum-height rule) and needs to become per-TLD configuration. This is a tracked work item, not a spec gap: the rule is normative now, the enforcement wiring lands with the first activation that needs it.
- **OpNS commits.** The commitment service is multi-protocol; OpNS commit chains start when the indexer exposes OpNS canonical state (genesis-lineage rules) — until then, SNS-only operation is the configured mode.
