ODNCA-STD-011 · WEB3-ACTIONS-1: The URI Scheme Family

sns: and opns: for payments · ord:// for content · ordmail: for web3 mail — one click model, one resolution ladder

Status: v0.5 — Draft (adoption together with STD-010; supersedes ODNCA-STD-005 upon adoption)
Companion documents: ODNCA-STD-001 (resolution & normalization), ODNCA-STD-003 (SNS-NAME-1), ODNCA-STD-004 §6 (state commitments), ODNCA-STD-010 (content discovery), ODNCA-STD-005 (superseded — see §6a)

This document is the **single scheme specification** for the family: it defines both syntax and behavior for all four schemes. Upon adoption it supersedes **ODNCA-STD-005 "URI Schemes" v1.0**; STD-005 then carries a superseded-by notice. IANA registrations reference this document; no other ODNCA document defines these schemes' syntax.

This document specifies **intended, testable behavior**. Its normative language (RFC 2119) binds implementations that claim conformance; nothing in it claims current deployment. §9 states, per scheme, what is live today. The document leaves DRAFT when the payment schemes reach Live.

§1 · The model

One principle: **the scheme tells the user which world they are acting in and what a click does.** Web3 actions and web2 actions are never mixed — clients that support both worlds keep them apart by scheme, as a security property, not a branding choice.

Scheme	Role	Click action	Web2 counterpart
`sns:name.tld`	**pay** an SNS name	wallet → Send, prefilled	—
`opns:name`	**pay** an OpNS name	wallet → Send, prefilled	—
`ord://name[/path]`	**load content**	browser → render	`https:`
`ordmail:prefix@domain.tld`	**web3 mail**	mail client → compose	`mailto:`

`ord://` is namespace-agnostic: a name with a dot is SNS, without a dot OpNS (per SNS-NAME-1) — one locator for both naming systems, with no opinion on payments. Content loaded via `ord://` may itself contain payment links; the origin rule in §2 governs those.

§1a · Scheme syntax (normative)

```
sns:[<mailbox>@]<name.tld>[?amount=&label=&message=&ref=]
opns:<name>[?amount=&label=&message=&ref=]
ord://<name.tld>[/<path>]
ord://<txid>[/<vout>]          (immutable form; vout defaults to 0)
ordmail:<prefix>@<domain.tld>[?subject=&body=]
```

The `sns:` address grammar and normalization are those of ODNCA-STD-001 §4.1/§4.2. A `<name>` with exactly one dot is an SNS name and MUST satisfy SNS-NAME-1 (ODNCA-STD-003); a `<name>` without a dot is an OpNS name (per the OpNS rules of ODNCA-STD-003). `ord://` keeps both STD-005 forms first-class: the name form is a mutable reference (content follows the name↔content coupling); the 64-hex txid form is a permanent reference that always loads that exact inscription. UTF-8 throughout; no whitespace or control characters; percent-encoding per RFC 3986; no IDNA/punycode — names are not DNS labels. `<path>` starts with `/`; absent path equals `/`.

§2 · Click behavior (normative)

- **Payment schemes (`sns:`, `opns:`):** the click opens the wallet's Send flow with the name as recipient. Before anything is signed the wallet MUST display: the input name, the **resolved address**, the resolution source (§3), the attestation check result (§4), and the **origin** of the invocation. A click MUST NOT sign or send by itself; parameters (§5) only prefill fields.
- **Origin rule (the primary attack vector):** content rendered from `ord://` is third-party and can contain payment links whose visible text misrepresents the target name. Therefore: invoking a payment scheme from rendered content REQUIRES a user gesture (browsers MUST block redirects, scripts and auto-navigation into payment schemes), and the confirmation screen MUST show the origin ("requested by ord://site/path") next to the actual recipient name and resolved address. The user judges both.
- **The wallet is the last line of defense:** the browser-side blocking above binds conforming clients, but the same link can reach the wallet through an OS handler from software this standard does not govern. The wallet-side requirements therefore hold independently: the wallet cannot know whether the invoker was conformant, so it MUST always show the full confirmation screen — and when no origin information is available, it displays "origin: external/unknown" rather than omitting the line.

- **`ord://`:** dereference and render. Never actions, never payments, never signing.
- **`ordmail:`:** opens compose with the recipient filled in; sending is always an explicit user action.

§3 · The resolution ladder (normative)

Every client that resolves recipient input applies the same ladder, top-down:

1. **Bare address** (base58check-valid) → use directly; source = "address".
2. **`prefix@domain`** → mail-directory lookup → wallet address + optional published public key; source = "mail".
3. **Name with a dot** → registry whois (owner-sync follows the chain) → holder address; source = "domain".
4. **Name without a dot** → OpNS index, exact match required → holder address; source = "opns".

Rules: **the first match decides — no fall-through**; if the matched rung's resolution fails, the input fails (for money there is exactly one deterministic path). **Zero matches are a hard reject**: each rung validates its input strictly (base58check for addresses; SNS-NAME-1 for names — exactly one dot for SNS, none for OpNS, no whitespace or control characters); syntactically invalid input (empty, spaces, multiple dots) is rejected before any network call — nothing resolved, nothing sent. The resolved address is always shown before signing (resolution never sends). "Unknown name" is strictly distinguished from "source unreachable": on unreachable, nothing is sent and the client does not pretend the name does not exist. The source label is always displayed.

§4 · The dual attestation check for payments (normative)

For **money**, the wallet checks two independent read paths onto the same chain:

- **Path 1 — service layer:** registry/index (owner-sync; fast and rich, single operator).
- **Path 2 — ODNCA anchor:** `GET odnca.org/verify/<name>` → compare `holder_address` with the address resolved via path 1. Independently verifiable through the state-commitment backbone of ODNCA-STD-004 §6.

Behavior: - **Match:** display "✓ matches committed state @ block N". - **Mismatch:** block the payment, show both addresses, and explain (possible causes: a transfer after the last commit, or an index discrepancy). A mismatch right after a genuine transfer is *correct* protective behavior — with an honest caveat: the window lasts until the next commit lands **and** the wallet has fetched it, so up to ~24 hours plus propagation, during which a new holder cannot receive payments by name. The mismatch screen therefore offers an exit: **"view the chain proof"** — an advanced route showing the current on-chain outpoint and holder, after which the user may proceed, informed and explicitly. - **Why mismatch is softer than unreachable (deliberate):** the mismatch exit moves the user toward *better* evidence — the chain proof is the chain itself, the highest authority above both read paths. An exit on unreachability would rest on the *absence* of evidence. Softer when knowing more, stricter when knowing nothing. - **ODNCA unreachable:** this is where an attacker would aim — disabling path 2 must not silently disable the check. Below an app-defined threshold, proceeding is an **explicit user choice** (a decision screen, not a dismissible notice); **above the threshold the payment is blocked** until the check is available again.

`opns:` payments reach full conformance only once OpNS state is part of the ODNCA commitments; until then clients apply path 1 with the explicit single-source choice.

§5 · Parameters

Payment schemes: `?amount=` (BSV, decimal, at most 8 fractional digits — satoshi granularity), `?label=`, `?message=`, `?ref=` (order reference). All optional, prefill-only; unknown parameters MUST be ignored. `amount` MUST be parsed as an exact decimal (integer satoshis or decimal string arithmetic), never as binary floating point; a value that fails exact parsing or exceeds 8 fractional digits MUST be treated as absent (field left empty) — never rounded. **Under single-source resolution** (an `opns:` recipient before OpNS commitments, or the explicit single-source choice of §4 when the ODNCA check is unavailable) the wallet MUST NOT prefill `amount`: it displays the requested amount as information and the user enters it manually — a prefilled amount on a single-source recipient is the fastest click-through this standard otherwise works to prevent. `ordmail:`: `?subject=`, `?body=` (compose prefill).

§6 · Form rationale (recorded once)

- **Payment and mail schemes are colon-only** (`sns:`, `opns:`, `ordmail:` — the bitcoin:/mailto: family): deliberate. The `sns:` and `ordmail:` addresses carry an `@`, and in an authority form every RFC 3986 parser would read it as userinfo@host; without `//`, everything after the colon parses as path, so the userinfo risk exists only in the form not chosen. This also preserves STD-005's adopted emission rule for `sns:`. Parsers SHOULD accept a sloppy double-slash on input and strip it (STD-005 §2); emitters MUST NOT produce it.
- **`ord://` with authority:** the content locator; the authority+path structure carries the route semantics of STD-010, and the txid form gives a permanent address.

§6a · Supersession of ODNCA-STD-005 (explicit)

This document supersedes STD-005 without breaking it. What changes and what carries over:

- **`sns:` semantics narrow, its form does not.** STD-005 defined `sns:` as "the name — action from context". This document fixes the click action: the default handling of an `sns:` URI is **payment** (§2); mail composition has its own scheme (`ordmail:`), and `sns:mailbox@name.tld` prefills a payment to the mailbox's resolved holder. The address form is unchanged from STD-005; all previously emitted `sns:` URIs remain valid, and STD-005's input leniency (accept-and-strip a sloppy `//`) is retained.
- **`ordns:` remains reserved** exactly as in STD-005 §3: parsers accept and strip it on input; applications MUST NOT emit it until a future standard defines it. The reservation carries over undiminished.
- **Both `ord://` forms carry over first-class** (name form mutable, txid form permanent), including STD-005's load-time status surfaces per ODNCA-POL-003 and the lookalike display rules of ODNCA-STD-001 §10.

§7 · Security considerations (summary)

Payment schemes never auto-sign; origin is always displayed for content-initiated invocations; resolution is deterministic with hard rejection of invalid input; the dual check anchors recipients in an independently verifiable committed state; homograph risks are addressed by client display policy; web2 and web3 actions are separated by scheme.

§8 · Relationship to other standards

ODNCA-STD-001 (address grammar, normalization, lookalike display), ODNCA-STD-003 SNS-NAME-1 (name validity), ODNCA-STD-004 §6 (the `/verify` state-commitment anchor), STD-010 (ord:// canonical content resolution), ODNCA-STD-005 (superseded — see §6a). This document is the scheme specification itself; IANA registrations reference it directly.

§9 · Implementation status (honest, updated per release)

Scheme / mechanism	Status today
`ord://` content loading	Live in ORDnet browser clients
`sns:` payment handler	Designed (this document); not yet shipped
`opns:` payment handler	Designed; awaits OpNS commitments for full conformance
`ordmail:` handler	Client in testing; handler not yet shipped
Dual attestation check (§4)	Designed; ODNCA `/verify` is live, wallet integration pending
Resolution ladder (§3)	Partly implemented (ORD/mail client resolver); §3 strict validation and zero-match rules pending

Status upgrade rule: a row moves to **Live** when a shipping client implements §2–§4 for that scheme; the document status leaves DRAFT when both payment schemes are Live.