Two Ways to Be Found Without DNS

Sunstone and the Nostr Community Conventions (NCC): a capability comparison, coexistence plan, and fault line map

Research article · 23 November 2026 · Sunstone Protocol v1.1 compared with NCC-02, NCC-05, NCC-06, and the Nostr Navigator implementation available at v0.43.0.

The short answer

Sunstone and NCC overlap, but they are not the same system wearing different names.

Sunstone is a released, transport-aware endpoint advertisement and resolver inside the MORP family. It says, in effect: this service key can be reached through these approved network doors, in this order. Its distinct strength is one signed address record that can describe several transport families—including MORP, PIPE, Tor, Reticulum, FIPS, LAN, and WebSocket endpoints—while enforcing privacy-persona separation.

NCC is a family of Nostr conventions. The pieces relevant here are **NCC-05**, which specifies identity-bound, short-lived service locators (primarily for dynamic IP-and-port reachability); **NCC-02**, which describes service identity and trust records; and **NCC-06**, which composes them into an opt-in service profile. **Nostr Navigator** is a browser implementation built around these conventions: it gives the user an `ncc://` address, resolves the service, and then fetches a web resource through the selected endpoint.

The clean coexistence strategy is **parallel records and an explicit client adapter**, not one schema pretending to be the other. Use NCC for the web-facing, browser-addressable service identity and its trust context; use Sunstone for MORP-family and other supported transport reachability. Publish both only when the owner intends the relationship to be public. If one side is anonymous and the other linkable, keep them on separate keys and do not cross-link them.

The hardest conflict is not “two DNS replacements.” It is **privacy policy**. NCC can present multiple endpoint families under one service identity and can pair a public trust anchor with a private locator. Sunstone’s M4 rule deliberately refuses to let one signing identity (or a shared TLS pin) straddle its anonymous and linkable transport personas. A bridge that casually merges those views could undo the very privacy separation Sunstone is designed to protect.

First, what does “NCC” mean here?

“NCC” is not one wire protocol. **Nostr Community Conventions** are documented usage patterns over existing Nostr primitives; NCC-05 explicitly says it standardizes client behaviour, not a change to Nostr or a new relay requirement. For this comparison, the key roles are:

1 **NCC-05 — locator:** a replaceable Nostr event, kind `30058`, authored by a service identity and distinguished by a `d` value. Its payload describes one or more reachable endpoints, preference, a freshness window, and optional capabilities. The convention's stated primary use is privately resolving dynamic `ip:port` endpoints.

1 **NCC-02 — service identity and trust:** a service record, kind `30059`, plus optional third-party certificate/transport-key attestations (`30060`) and their revocations (`30061`). It can bind a service identifier to an endpoint and a transport-key fingerprint.

1 **NCC-06 — participation profile:** a composition of NCC-02 and NCC-05. Its model is: validate the service identity/trust record, then use an NCC-05 locator for reachability where appropriate, while retaining the NCC-02 trust checks.

Those convention documents are not the browser. **Nostr Navigator** adds the user-facing `ncc://<npub>/<service>/<path>` address, endpoint resolution, a web proxy/navigation layer, transport security and a trust display. For reproducibility, I pin the product discussion to **v0.43.0**, whose release page timestamps it at 23 November 23:24 UTC; I do not treat later moving-branch changes as already released. If you are reading this before that same-day release time, v0.42.x was the available build at the earlier cutoff. The NCC-05 document identifies itself as **Draft** and NCC-02 as **Informal / Experimental**; a working application does not by itself change those status labels.

That distinction prevents an apples-to-oranges comparison: Sunstone v1.1 is a protocol and reference implementation, while NCC is a family of conventions plus an application ecosystem that includes a browser and separate publishing tools.

---

1. Capability comparison

Primary job	Advertise and resolve a service's ordered multi-transport endpoints.	NCC-05 resolves a service identity to current endpoint(s); NCC-02 supplies identity/trust context; Navigator turns that into a browsable service address.
Address/event model	Public UEA: Nostr kind `39530`, keyed by signing `pubkey` plus a `d` slot such as `sunstone:relay`. Private cards use kind `14` inside NIP-17/NIP-59 wrapping; offline `sunstone1:` Seed Cards are also defined.	NCC-05: parameterized-replaceable kind `30058`, selected by author and `d` (typically `addr`). NCC-02: kind `30059`, with optional `30060`/`30061` trust events. Navigator accepts `ncc://<npub>/<service>/<path>`.
What the record describes	A list of supported transport types, target, priority, and optional capabilities; it also fixes a role and freshness/withdrawal semantics.	A payload with `ttl`, `updated_at`, endpoints, and optional `caps`/`notes`; each endpoint has a URL and may carry priority, family, or `k`. The NCC-05 specification says dynamic IP-and-port is the primary case.
Transport range	Explicit MORP, PIPE, Tor, Reticulum, FIPS, WebSocket, and private LAN forms, subject to the spec's transport rules.	URL-oriented endpoints: NCC-05 centres on IP/port but allows tunnels, overlays, and onion endpoints as alternatives; NCC-06 treats onion endpoints as a supported deployment option. Navigator has direct/Tor handling for supported web endpoint forms. The convention set does not define native MORP, PIPE, Reticulum, or FIPS transport semantics.
Web navigation	Not a browser address space. Sunstone finds endpoints; it does not define page origins, HTTP path joining, browser cookies, redirects, or rendering.	A major Navigator capability. `ncc://` names a service and path; the browser resolves and fetches it while preserving per-service origins and handling navigation/proxy concerns.
Privacy model	Public kind `39530` records are public. Private endpoint sharing is a different path: wrapped private cards or offline Seed Cards. M4 rejects mixing anonymous and linkable transports for one signing identity across records.	NCC-05 requires encryption for private `ip:port` data by default, but allows plaintext when the endpoint is intentionally public. The `private` tag alone is only a marker, not privacy. NCC-02 records can disclose service existence, endpoint, and/or key information depending on what is published.
Trust	The event author's signature anchors the record; optional TLS SPKI pin `k` verifies a TLS endpoint. Sunstone has strict persona/pin-separation checks.	NCC-02 is explicitly a service trust layer: signature, expiry, `k`, and optional third-party attestation/revocation policy. NCC-06 composes this with locator resolution. Navigator exposes an anchor/trust view, but convention support and any particular client's evaluation policy must be checked separately.
Freshness	Every record has NIP-40 expiration, maximum TTL **90 days**, future-skew limit **24 hours**, and explicit `withdrawn` versus `retired` status.	NCC-05 freshness is the earlier of `expiration` and `updated_at + ttl` when both are present. The convention specifies bounded refresh behaviour but does not use Sunstone's same 90-day hard ceiling or its two-state slot lifecycle.
Endpoint failover	Valid endpoints are filtered by locally enabled transmission modes, then tried in ascending priority; rejected identity sets fail closed.	NCC-05 orders by priority and deterministic address family, tries in order, and recommends cooldowns. Navigator adds caching, endpoint health and retry/re-resolution behaviour.

Human-readable names	None are allocated or sold. The identity key and service slot are the lookup handle; Seed Cards solve out-of-band bootstrap.	`ncc://` is easier to type and share as a structured identity/service/path address, but it is still based on a Nostr public key and service id—not a global human name registry. NCC-05 itself lists public website discovery and human-readable naming as non-goals.
Payments	No payment fields, wallet operations, fee gates, or paid ranking in Sunstone resolution. This does not prohibit a separately supported Nostr event or payment tool from travelling over a connection.	NCC-05 explicitly has no dependency on payment/zap/wallet NIPs. Navigator may ship wallet features independently; those are product features, not NCC locator semantics.

What Sunstone does better for MORP's job

Sunstone's differentiator is **transport breadth with a single privacy-aware endpoint model**. A service can advertise, for example, a Tor endpoint, a MORP route, a PIPE lane, or an RNS/FIPS route and let the client choose among the locally enabled options. Its M4 rule is not a UI warning; it rejects a signing identity that appears in both anonymous and linkable endpoint sets, including across separate records. That makes Sunstone much closer to a transport-aware MORP address layer than a website directory.

It also has protocol-specific lifecycle features that are unusually explicit: 90-day maximum record lifetime, 24-hour future clock-skew tolerance, and a distinction between a reopenable "offline for now" withdrawal and permanent retirement of a slot. Those semantics matter to a resolver implementing Sunstone exactly, but they should not be assumed to exist in NCC.

What NCC/Navigator does better for a web service

NCC's advantage is **service identity plus usable web navigation**. NCC-02 gives an application a place to express service identity and trust material; NCC-05 gives it a moving endpoint; NCC-06 explains how to use both. Navigator makes that composition practical for a human: enter an `ncc://` address, resolve the service, fetch a page or application path, and inspect the trust/anchor information in the browser.

The resulting experience is much more than a locator record. Navigator handles the browser-specific tasks Sunstone intentionally does not: path and redirect handling, web origin separation, cookies/storage boundaries, HTTPS policy, and a signer interface. That is not a protocol superiority claim; it is simply a different scope.

---

2. How the systems can coexist

The simple architecture: two records, one deliberate service relationship

A service owner can publish:

1. **An NCC-02 service record** for the web-facing or Nostr-facing service identity and trust policy.

2. **An NCC-05 locator** for Navigator-compatible endpoint lookup and address rotation.
3. **A Sunstone record** for the same service's MORP-family, Tor, RNS, FIPS, PIPE, or other Sunstone-supported access paths—provided the identity and privacy rules genuinely match.

These are separate Nostr events with separate schemas. They can refer to one service without being interchangeable. A client that understands only NCC does not automatically understand kind `39530`; a Sunstone resolver does not automatically know how to validate NCC-02 attestations or apply a browser's trust policy.

For a **public, linkable service**, the owner might publish a Navigator-friendly HTTPS service and a Sunstone `wss` or other compatible endpoint under an intentionally public service identity. A client integration could present both paths: "open the web UI" through NCC and "connect using MORP" through Sunstone. Where the same TLS key and endpoint are deliberately shared, the publisher should verify that the exact pin bytes and trust scope mean the same thing in both systems; a tag named `k` is not by itself proof of semantic equivalence.

For a **private service**, NCC-05 can carry an encrypted locator to authorized recipients while Sunstone uses a private wrapped card or Seed Card. The two access mechanisms can coexist, but their recipient sets, endpoint freshness, and trust decisions must remain separately auditable. A public NCC-02 record may still reveal that a service exists even when its endpoint is hidden; do not treat "encrypted locator" as "no metadata."

A bridge should translate intent, not event bytes

A good integration is a **resolver adapter** or shared interface:

1. For `ncc://` navigation, run NCC-02/05 validation and browser-origin rules.
1. For a Sunstone identity/slot, run Sunstone validation, M4 checks, mode filtering, pin checks, and endpoint ordering.
1. Let the user or calling application choose which namespace and trust policy applies.
1. If a service publishes both records, link them only when the owner has explicitly chosen to make that relationship discoverable.

A naïve bridge that copies an NCC endpoint into a Sunstone record (or vice versa) can lose critical semantics: NCC `d` naming, TTL representation, anchor identity, recipient encryption, or Sunstone transport/persona rules. A bridge should create a new, valid record in each convention, not relabel one event as the other.

Where the stack could meet

A practical MORP integration could use NCC for a browser-friendly public front door and Sunstone for mesh/private reachability. A user might open a documentation page through Navigator, then discover that the service also offers a MORP endpoint through a MORP-aware client. A community index could show both choices for public services. The NCC browser would remain the web client; MORP Box, Sovereignty, or Base would remain the MORP transport products.

This is a **complementary stack**, not one protocol taking over the other. The handoff must be explicit: a browser does not become MORP-aware simply because a page's endpoint was found by NCC, and a Sunstone client does not become a browser because one endpoint happens to serve a website.

3. Where the systems are at odds

A. One identity cannot always represent both privacy personas

This is the principal architectural conflict.

NCC-05 allows an ordered endpoint set. NCC-06 expressly discusses multiple paths, including clearnet and onion. Nothing in the NCC locator model imposes Sunstone's M4 identity split. That flexibility is useful for resilience—but if the same Nostr key advertises both an anonymous transport and a linkable one, Sunstone's resolver must reject that identity's combined record set.

So a single NCC service identity that says "here is my public HTTPS address and my Tor/MORP address" **cannot simply be imported as one Sunstone identity**. It would contradict the Sunstone privacy invariant. The safe choices are:

1. keep that service wholly linkable and do not publish anonymous transports under that key;
1. use distinct keys for the anonymous and linkable personas, with no public cross-link if unlinkability is a goal; or
1. consciously decide the identities are not meant to be unlinkable, document that choice, and still comply with Sunstone's required separation rules rather than bypassing them.

The last option is not "turn M4 off." If the same key has both transport classes, Sunstone rejects it. And using a common TLS SPKI pin across the two personas can recreate the linkage even when the signing keys differ.

B. NCC-02 trust anchoring is not Sunstone trust anchoring

NCC-02 can incorporate third-party attestations and revocations, with client-selected certifiers and trust levels. Sunstone does not define those event kinds or that trust policy. Sunstone's record signature and optional SPKI pin answer different questions: "who signed this endpoint record?" and "does this TLS endpoint present the pinned key?"

A client may use both systems, but it must keep the two verdicts distinct. An NCC-02 attestation is not automatically a Sunstone signature, and a Sunstone `k` pin is not automatically an NCC-02 attestation. If an application wants to require both, it needs a documented composition rule and must fail safely when one system's evidence is missing or conflicts.

C. The data models and addresses do not line up

NCC's `ncc://<npub>/<service>/<path>` includes a browser navigation path. Sunstone's `(pubkey, d)` slot resolves a service to ordered endpoints and capabilities; it does not define a page-origin model. The event kinds differ (`30058`/`30059` versus `39530` and private kind `14`), as do tags and payload schemas. A generic NCC URL or TCP locator is not automatically a Sunstone `transport`/`target` pair, and a MORP or PIPE route is not automatically a valid NCC browser endpoint.

This is a clean interoperability boundary: **same underlying Nostr relays, different application conventions**. Matching keys or similar fields do not make the records wire-compatible.

D. Private-by-default versus a public DNS-like record

NCC-05 is private by default for `ip:port` data, but has an intentional-public plaintext exception. Sunstone's public record is public by design; its private-sharing mechanisms are separate. That is a policy difference, not a contradiction: each owner can choose the appropriate disclosure mode. But a translator must not turn a private NCC locator into a public Sunstone endpoint record, nor treat an NCC `private` marker as encryption.

Likewise, a public NCC-02 service record can reveal service existence and possibly key material even if NCC-05 conceals the endpoint. If the purpose is to hide the relationship between a person and a service, adding a public anchor can defeat that purpose.

E. Withdrawal and permanent retirement have no exact NCC equivalent

Sunstone distinguishes `withdrawn` (zero endpoints, reopenable) from `retired` (zero endpoints, permanent tombstone for the `(pubkey, d)` slot). NCC-05 primarily models latest replaceable locator state plus TTL/expiration; NCC-02's revocation kind is for a certifier's attestation, not a Sunstone-style permanent service-slot tombstone.

A bridge can stop publishing a locator, expire it, or publish whatever empty-state behaviour its NCC client supports. But it cannot honestly claim that this is identical to Sunstone `retired`: an NCC identity may later publish a new locator, and older caches or relay state follow NCC's own rules. Retirement must be recorded and enforced separately if the permanence guarantee matters.

F. "DNS replacement" overstates what either system does

Both systems reduce dependence on DNS for services addressed by Nostr identity. Neither automatically supplies a global, human-readable, universally resolvable namespace for every website on the ordinary Internet. NCC-05 itself says public website discovery and human-readable naming are non-goals. Sunstone similarly assigns no names for sale; it resolves a key and slot. Navigator's `ncc://` URI is a client convention, not an ICANN-style global domain root.

Both are better understood as **identity-based service discovery and endpoint resolution**. That is powerful, but narrower—and more honest—than claiming DNS has disappeared from all web navigation.

What I would recommend for MORP

1. **Keep Sunstone as the MORP-family endpoint layer.** Its transport range and M4 rule solve problems NCC-05 does not attempt to solve in its core locator model.
2. **Treat NCC as an optional web/service interoperability profile, not a replacement.** NCC can be useful for browser-friendly service access and a distinct service trust record.
3. **Do not merge records or trust states.** Add an adapter only when a client needs to consult both, and preserve which system supplied each endpoint and each trust decision.

4. **Separate linkable and anonymous service identities before publishing.** Do not reuse the signing key or TLS SPKI across an anonymous and a linkable persona; avoid public cross-links if the goal is unlinkability.
5. **Use separate freshness/lifecycle logic.** Enforce Sunstone's 90-day ceiling and D15 tombstones for Sunstone; enforce NCC's TTL/expiration and trust rules for NCC. Any mapping between them must be explicitly lossy and documented.
6. **Keep the Owner's existing scope ruling in force.** MORP's D12 allowed borrowing only the light-touch TLS SPKI-pinning concept for Sunstone v1.1; it did not authorize wholesale adoption of NCC or NCC-Server. This comparison is analysis, not authorization to replatform, merge, or start a new implementation.

Bottom line

Sunstone is the stronger fit when the question is **"Which of my supported network paths can reach this service, while preserving MORP's privacy-persona boundaries?"** NCC/Navigator is the stronger fit when the question is **"How can I open a Nostr-identified service in a browser, with a service trust record and ordinary web navigation?"**

They can coexist at the service owner's discretion. The safe design is to publish independent records and make the relationship visible only when intended. They are at odds wherever a bridge assumes one shared schema, one trust model, one lifecycle, or one key can safely represent both anonymous and linkable paths.

The practical takeaway is not "pick a winner." It is: **Sunstone for transport-aware MORP reachability; NCC for NCC-aware service discovery and browser navigation; a carefully governed adapter between them, never a silent schema substitution.**

Sources and comparison basis

NCC sources

1. NCC-05: Identity-Bound Service Locator Resolution — Draft, source snapshot available by 23 Nov 2026
2. NCC-02: Pubkey-Owned Service Discovery and Trust — Informal / Experimental, pre-23-Nov source snapshot
3. NCC-06: NCC-02 / NCC-05 Service Profile — source snapshot including 15-Nov changes
4. Nostr Navigator v0.43.0 README and v0.43.0 release (released 23 Nov 2026).

Sunstone source basis

1. Sunstone Protocol **v1.1** release artifact: `docs/SUNSTONE_SPEC_v1.1.md`, tag `v1.1` → specification commit `58d4cc90d43c3d05c5a981aa86fdd85f512d5d7d`; release evidence at repository head `be6029d70c5f35550ed02a34fd5b497c3c8a3adc`.
1. The comparison focuses on specification semantics, not an unverified claim that all MORP product integrations are complete. Box's MORP-NERD testing and later Sovereignty/Base integration remain separate milestones.

[1] **Metadata note:** the v1.1 specification's front-matter still contains old "release candidate" wording even though the v1.1 tag and release evidence establish that it was released. This comparison treats the Owner-approved tag and release evidence as authoritative; the stale sentence is a documentation inconsistency, not a different technical release.

This is a comparative analysis, not a normative protocol specification. NCC documents can change; implementation behaviour is version-specific. Check the linked release/source snapshot before implementing an adapter.