# New Identifiers for new Webs

Framing the issues

SWICG Special Topic Call, 15 Sept 2026

# Terminology

For objects (on mutable webs):

- **Integrity** - Thing in hand is same thing signed/published/intended/etc
- **Identity** - If that thing is different, the delta is legitimate (provenance)

For identifiers:

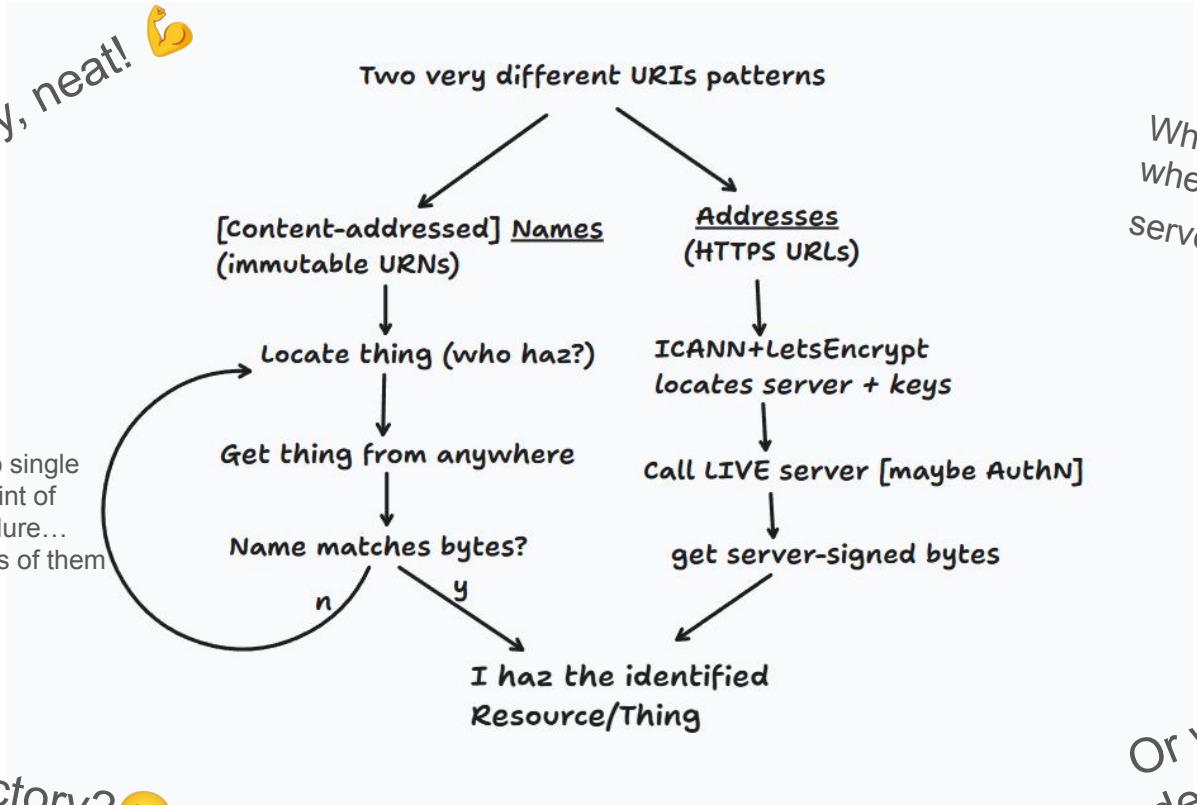
- **Resolve** - Translate pointers to current addresses, metadata, etc. if applicable
- **Dereference** - Get the thing identified from authoritative source
- **Validate** - Identifier is valid syntactically
- **Verify** - Object has been **dereferenced**; **integrity** & **identity** checked (if applicable)

# Conceptual Framing: Identifiers for what

- Attachments (most static, more valuable to deduplicate and make efficient)
  - CDNs and SRI are tablestakes in 2026… the “Hug of Death” is naïve HTTP resolution failing
- Activities
  - Activities not immutable/canonicalizable over time; each identifier is really a specific actor’s view at a point in time
  - Privacy problems with *non-repudiable circulation* beyond intended audience, particularly if activated retroactively
- Actors (hardest problems)
  - What does the spec mean by “the actor’s namespace”, or in 2017, “the user’s namespace?”
  - Who hosts Actor records independent of outbox server? A dumb sidecar or a distinct service?
  - Who witnesses migrations, if servers die suddenly or if servers defederate?
  - Client-side possibilities coming into focus (not just ActivityPub API but also Solid/Fedipod, ELFA, Wallet-Attached Storage, etc)

# Content-addressed versus Authority-addressing

Ambient Authority, neat! 💪



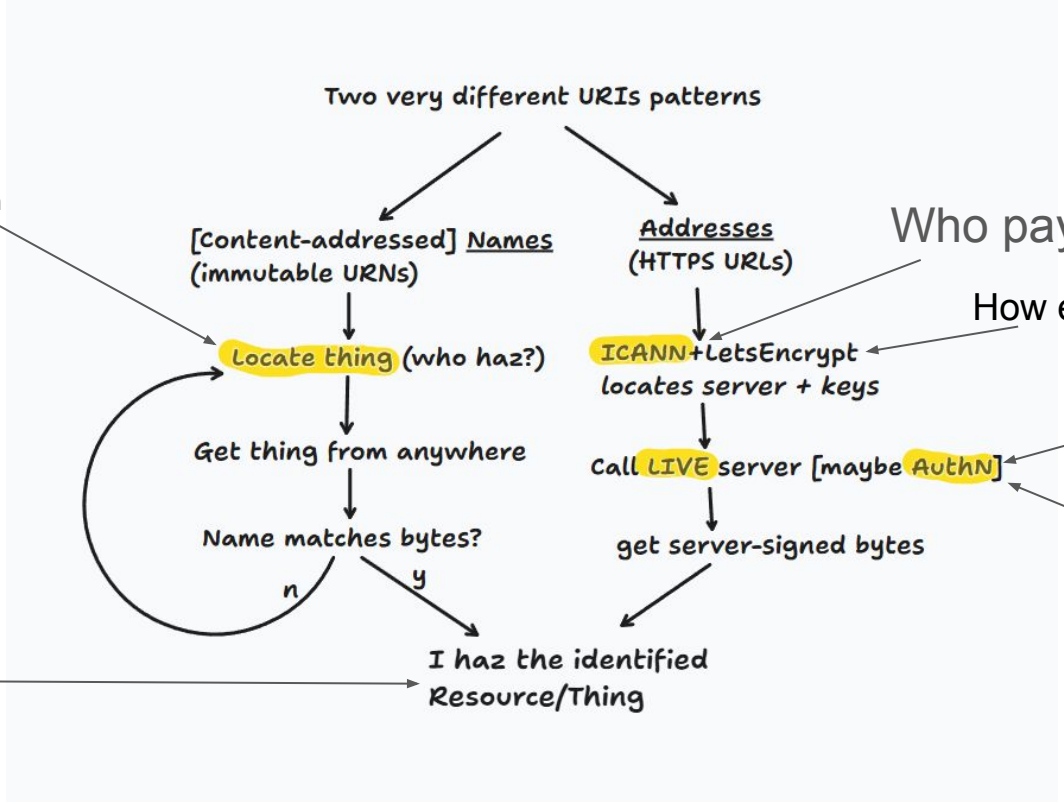

No single point of failure… lots of them 😎

What happens when users & their servers fight? ⚔️

where's the directory? 😬

Or when servers defederate? 🐵

# Apples-to-Apples Comparisons: Where options differ




How do you know you've found the freshest version of a mutable resource?

Who pays that rent?

How else confirm keys?

Someone **else's** AuthN is always the weakest link!

Can we authenticate data independently from actors and outbox servers?

How do I check a signature in the past? How do I find **historical keys,** if those are mutable too?

# Comparison: Major Identifier Schemes

| type | URI scheme | resolution+deref algorithms | normref | pros | cons |
|---|---|---|---|---|---|
| ni:// | yes, IANA-registered, simple /.well-known/ system | HTTPS GET on authority-hint with /.well-known/ query on known caching servers as fallback | IETF RFC 6920 | low complexity, zero-dep implementable | limited range of hash functions |
| ipfs:/// | yes, but very complex and likely needs to be profiled via regexp to reduce security surface for AP use cases | many round-trips, some unknowable timing | no hope of normref | "kitchen-sink", can handle many different use-cases and extensions, e.g. virtual file system, toxic/DMCA content blocklist built-in, etc. | quite complicated, requires separate DHT and/or trustful HTTPS routing to dereference, an additional chatty protocol to host, content-type must be passed out of band for type safety |
| magnet:// | yes, stable and versioned | special side-protocol on standard endpoint on authority-hint(s) provided in query params with known indexing servers as fallback | stable community spec and impls but no normref | less complicated, diverse implementations | requires separate DHT protocol to dereference and/or host, content-type much be passed out of band for type safety, no built-in affordances for tracking/blocklisting toxic or DMCA payloads |
| eris | URN-only | straightforward urn→deref algo | no but spec stable since 2022 and community tooling | interesting deduplication/tracking-avoidant feature ("convergence secret", i.e. arbitrary hash salt to produce new identifiers for tracked content); DNSLink system like IPFS; has an SSB-like append-only mode per publisher; has its own RFC Canonicalization for content-addressing RDF recrods | no way of preserving or tracking content-type; no identity-awareness or built-in metadata layer |
| UUIDv5 | no | IETF RFC 9562 | no | takes a URL as input and turns it into an opaque UUID (but can be verified if you know the original URL) | not a *content*-identifier, just a one-way commitment to the URL itself |
| digestMultibase | no | no | stable, W3C normref | low complexity, zero-dep implementable; works more like SRI than rest of this list, i.e. provides a stable syntax for passing hash alongside URL as part of link object | limited range of hash functions |
| hashlink | yes | deref as normal but with checksum in hand at the end | stable, but unlikely to get normative status | very simple syntax for expressing hash in query parameter of URLs | query parameters not ideal in every use-case, can get lost in transport, etc. |

# Prioritizing (and Parallelizing) Use Cases

1. Efficiencies of Scale - SRI-like ways to reduce the “Hug of Death” (FEP-cd47)
2. Discontinued/dead servers part 1: “Rescuing” content from archived/dead servers, and the semantics of “data recovery” (Sutty coöp, FEP-1042)
3. Discontinued/dead servers part 2: User migration (FEP-73cd#3&4, cipub)
    a. The dreaded “Third Server” problem - where do Actors live beyond Outbox servers?
4. Protocol Experimentation - Backwards-compatibility and graceful failures when weird URLs and URNs creep into objects you see (FEP-fffd, FEP-1042)
5. Bridging - What to do when the rel=canonical URL is not an HTTPS URL
    a. IMHO, this might best be specified in a CG Report like Discovery!

# Different Questions for Different Constituents

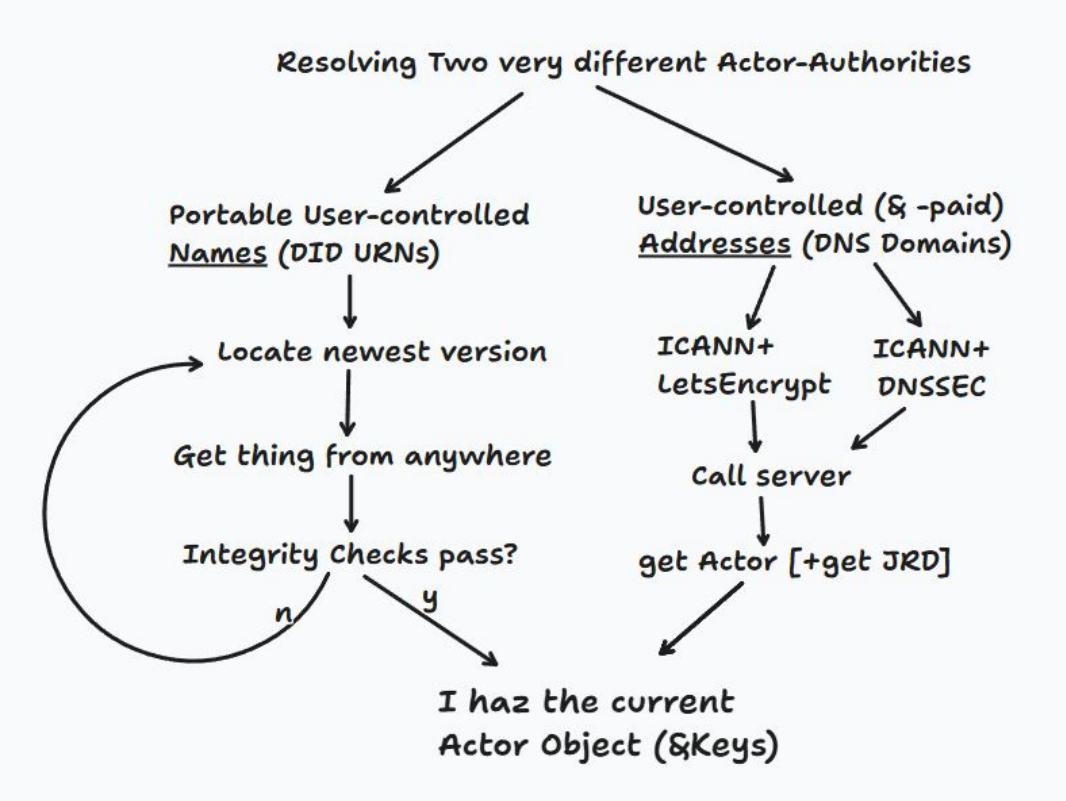
1. (for Implementers) For each use-case or approach, *who* needs to support *what* for it to work? Who needs to buy-in before it's worth designing, whether in the open or in private?

2. (For designers) How to backwards-compatibly integrate new kinds of Actor, service, and/or identifier?
   a. FEPs: 1042 (static & P2P addressing), ef61 (ap:// scheme), cd47 (dedup use cases), (signing), 73cd (migration user stories), e3e9 (detached actor service)

3. (For businesses): How to get shared infrastructure off the ground (Key Transparency Log, Portability Witnessing Service, toxic-data scanning, etc)?

4. (For CG) How to negotiate interop profiles between distinct profiles of AP?

# Bonus Round: Resolving Actors beyond [one] outbox



# Comparison: URI schemes for Activities or Actor+Activities

| type | URI scheme | resolution+deref algorithms | normref | pros | cons |
|---|---|---|---|---|---|
| secure scuttlebutt log | urn only, no authorities in TCP sense | locating is the tricky/OOB part, dereferencing deterministic once you have the bytes | unlikely | per-publisher append-only non-repudiable log | key rotation not possible; much discovery is OOB/underspecified by web standards |
| at:// | current scheme is URL-spec non-conformant | very straightforward, similar to DID URL default behavior (dereference DID, transform rest of URL to PDS-based to dereference) | iterating (slowly) in IETF | uses individual user's DIDs as "authority" component, for now at least | requires trustful and/or complex resolution (merkel tree walking), non-normative CBOR serialization, all events fully-public and non-repudiable |
| DASL | rasl spec | NIH-like URN format | unlikely | more generic superset of ATP, very simple/NIH-like | requires serialization as bespoke CBOR profile for which no normref is possible |
| ap:// | current scheme is URL-spec non-conformant | similar to NIH, a /.well-known/ gateway can be tried on any server suspected of having a copy; DID support and DID Doc resolution unspecified | unlikely unless APWG adopts it as a work item with strong CG consensus | rightly defers authenticity and integrity to DI signatures | malicious server ejecting tenant case is unspecified, DID resolution is unspecified, and case of two valid conflicting signatures from the same key on different servers also |
| FEP-e3e9 | not needed | not needed | unlikely | no backwards-compatibility issues | just moved server-dependency to an independent server (at additional cost to user) |
| DID URLs (in general) | per-method URN scheme + base/default DID URL pathing behavior | per-method (resolve DID Doc before deref path/params) | DID URL is ratified, DID Resolution (debatably on critical path) still unstable | portable | resolution, security all depend on the specific method used |
| DID URLs (webvh) | yes, specified URN scheme + a simple /.well-known/ translation for URL fallback | clearly specified in method spec | v1.0 at DIF, on shortlist for W3C DID Methods | low complexity, zero-dep implementation | some complexity around witnessing to be 100% tamper-evident against malicious servers (requires something like soatak's key transparency system) |
| did:dns | just a URN scheme, no URL scheme per se | vanilla DNS records→DID Doc | not in scope for DID Methods WG, but could be v1.0'd as a CCG spec (equiv to SWICG report) | allows DNS records to be the "independent authority" | can ONLY set keys and service endpoints, not arbitrary properties (e.g. "inbox" and "outbox" would have to be set as serviceEndpoints, not top-level properties) |