Network Working Group
Internet-Draft
Intended status: Informational
Expires: 22 July 2026

X. Zinn
Independent

18 January 2026

# Identity Namespace Architecture
# draft-xzinn-idns-architecture-00

## Abstract

This document defines the Identity Namespace architecture.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 22 July 2026.

Table of Contents



## 1. Introduction

# Abstract

This document defines an Identity Namespace Architecture that establishes globally unique, cryptographically verifiable identifiers and their associated trust boundaries. The architecture specifies namespace structure, resolution semantics, and authority separation without prescribing specific transports or cryptographic algorithms. It is intended to serve as a foundational identity substrate for higher level protocols and enforcement mechanisms.

# Introduction

The Identity Namespace System Architecture (INSA) defines a cryptographically verifiable framework for the representation, publication, and resolution of human and institutional identities. INSA is designed to provide a durable identity substrate that is independent of transport protocols, application layers, and communication mechanisms.

Existing identity systems frequently conflate identity with authentication, messaging, directory services, or application-specific account models. This coupling introduces systemic fragility, centralization pressure, and opaque trust dependencies. INSA explicitly rejects this approach by isolating identity semantics and ownership from all higher-layer concerns.

At its core, INSA provides a mechanism by which an identity may be uniquely anchored to cryptographic material, published under a sovereign trust boundary, resolved without prior trust relationships, and independently verified without reliance on centralized intermediaries.

The architecture is intentionally minimal. INSA does not prescribe user interfaces, storage backends, transport protocols, or application workflows. Instead, it defines the smallest set of primitives required to support long-lived, verifiable identity in a globally distributed environment.

INSA distinguishes between canonical identities and subordinate identities. Canonical identities represent long-lived personhood and are anchored by cryptographic keys generated and controlled by the identity holder. Subordinate identities represent institutionally governed roles issued under explicit authorization and accountability structures. This separation enables institutions to assert role-based authority without subsuming or controlling the underlying human identity.

Identity information within INSA is expressed through signed manifests and published into Merkle-rooted registries operated by sovereign authorities. These registries allow relying parties to verify identity inclusion and integrity using cryptographic proofs rather than trust in online services or proprietary directories. Resolution is designed to be stateless, allowing independent verifiers to operate without synchronized databases or persistent connections to identity operators.

INSA deliberately excludes endpoint discovery, messaging, session establishment, and communication semantics. While identity identifiers may be used as stable references by higher-layer systems, the architecture itself does not define how communication occurs, how messages are routed, or how presence is established. These concerns are addressed in separate companion specifications.

The primary design goals of INSA are durability, verifiability, and sovereignty. Identities are intended to persist across decades, survive institutional change, and remain independently verifiable even when originating authorities are offline. By grounding identity in cryptographic proofs and explicit governance boundaries, INSA aims to provide an identity substrate suitable for civil, commercial, and personal use without reliance on centralized platforms.

This document specifies the architectural components, trust model, namespace structure, and resolution semantics of INSA. It does not define application profiles or operational policies beyond those required for interoperability and security.

# Terminology

The key words **MUST**, **MUST NOT**, **REQUIRED**, **SHALL**, **SHALL NOT**, **SHOULD**, **SHOULD NOT**, **RECOMMENDED**, **MAY**, and **OPTIONAL** in this document are to be interpreted as described in <<rfc2119>> and <<rfc8174>>.

The following terms are used throughout this document.

**Identity** A persistent representation of a human or institutional entity, anchored by cryptographic material and governed by explicit authority.

* **Canonical Identity** A long-lived identity representing a real human. A canonical identity is anchored by a public key generated and controlled by the identity holder and published under a sovereign namespace. Canonical identities persist independently of institutional roles, employment, or affiliations.

* **Subordinate Identity** An identity representing an institutionally governed role associated with a canonical identity. Subordinate identities are issued, renewed, and revoked by role authorities and are subject to institutional policy and accountability.

* **Sovereign Operator** An authority responsible for operating a sovereign identity namespace and publishing a Merkle-rooted registry of canonical identity manifests. A sovereign operator defines the trust boundary for canonical identity issuance and governance within its namespace.

* **Role Authority** An institution authorized to issue and manage subordinate identities within a sovereign namespace. A role authority asserts institutional accountability for actions performed under subordinate identities it issues.

* **Manifest** A signed, structured document describing identity attributes, cryptographic keys, governance metadata, and lineage information. Manifests are the primary data objects published and resolved within INSA.

* **Canonical Manifest** A manifest describing a canonical identity. Canonical manifests are published into sovereign identity trees and include the canonical public key, optional descriptive attributes, lineage metadata, and sovereign operator attestation.

* **Subordinate Manifest** A manifest describing a subordinate identity. Subordinate manifests include role identifiers, capability bindings, validity periods, and dual attestations from both the canonical identity holder and the issuing role authority.

* **Capability** A cryptographically verifiable authorization describing a set of permitted actions. Capabilities are bound to subordinate identities and evaluated independently by relying systems.

* **Sovereign Identity Tree** A Merkle-rooted registry containing canonical identity manifests operated by a sovereign operator. Each tree version is committed by a signed Merkle root, enabling inclusion proofs and stateless verification.

* **Merkle Root** The cryptographic commitment representing the state of a sovereign identity tree at a specific point in time. The Merkle root is signed by the sovereign operator and published as authoritative metadata.

* **Inclusion Proof** A cryptographic proof demonstrating that a specific manifest is included within a given Merkle root. Inclusion proofs allow relying parties to verify identity membership without access to the full identity tree.

* **Resolver** A software component that retrieves identity manifests, Merkle roots, and inclusion proofs, and performs cryptographic verification of identity data. Resolvers operate without requiring trust in the source of retrieved data.

* **RATB (User-Controlled Trust Broker)** A user-controlled cryptographic component responsible for safeguarding private keys, performing signature operations, and enforcing local policy. The RATB MAY be implemented using hardware or software mechanisms and is not required to be network-accessible.

* **Namespace Suffix** A globally unique identifier designating a sovereign identity namespace, such as ‘ca.id‘. Namespace suffixes define trust boundaries and publication authority.

* **Resolution** The process of retrieving identity data and verifying its authenticity and integrity using cryptographic proofs and signed metadata.

* **Relying Party** An entity that consumes resolved identity information to make authorization, authentication, or policy decisions.

* **Companion Specification** A separate document defining functionality intentionally excluded from INSA, such as endpoint discovery, messaging, or communication profiles.

These terms are used normatively throughout this document to define the architecture and behavior of the Identity Namespace System Architecture.

# Architectural Overview

The Identity Namespace System Architecture is composed of a small set of layered components that together enable durable, verifiable identity publication and resolution. Each layer has a clearly defined responsibility and interacts with adjacent layers through explicit, cryptographically verifiable artifacts.

The architecture is intentionally non-interactive. No component is required to maintain session state, trust live network services, or participate in global consensus. All trust decisions are derived from verifiable data objects and signed commitments.

## Architectural Layers

INSA is organized into the following conceptual layers:

2. Problem Statement

* Sovereign Identity Tree Layer * Canonical Identity Layer * Subordinate Identity Layer * Capability Layer * Resolution Layer

Each layer MAY be implemented independently, provided that the required data formats and verification semantics are preserved.

## Sovereign Identity Tree Layer

The Sovereign Identity Tree Layer provides the authoritative publication mechanism for canonical identities. A sovereign operator maintains a Merkle-rooted registry containing canonical identity manifests.

Each update to the registry produces a new Merkle root. The sovereign operator MUST sign each Merkle root and make it available to relying parties. The signed Merkle root represents a cryptographic commitment to the complete set of canonical identities published at that point in time.

Sovereign identity trees do not require real-time availability. Relying parties MAY obtain Merkle roots and inclusion proofs from untrusted sources and verify them independently.

## Canonical Identity Layer

The Canonical Identity Layer defines the representation of long-lived human identities. Each canonical identity is anchored by a public key generated and controlled by the identity holder.

Canonical identity manifests describe the canonical public key, optional descriptive attributes, and lineage metadata required to support key rotation and continuity. Canonical manifests MUST be included in a sovereign identity tree to be considered valid.

The sovereign operator attests to inclusion and governance of canonical identities but does not control the private keys associated with them.

## Subordinate Identity Layer

The Subordinate Identity Layer enables institutions to issue role-based identities without assuming control of the underlying human identity.

Subordinate identities are represented by subordinate manifests issued under dual attestation. A subordinate manifest MUST be signed by both the canonical identity holder and the issuing role authority.

Subordinate identities MAY be renewed, revoked, or replaced independently of the canonical identity. Revocation of a subordinate identity does not affect the validity of the canonical identity.

## Capability Layer

The Capability Layer defines fine-grained authorization semantics associated with subordinate identities. Capabilities describe the actions a subordinate identity is authorized to perform and the conditions under which those actions are valid.

Capabilities are bound to subordinate identities and evaluated by relying parties during authorization decisions. INSA does not define a global capability language; it defines only the cryptographic binding and verification model.

## Resolution Layer

The Resolution Layer defines how identity data is retrieved and verified. Resolution consists of obtaining identity manifests, signed Merkle roots, and inclusion proofs, then performing cryptographic verification.

Resolvers MUST verify:

* the signature on the Merkle root * the validity of the inclusion proof * the signatures on resolved manifests * the binding between canonical and subordinate identities

Resolvers do not require direct trust in sovereign operators, role authorities, or distribution infrastructure. All retrieved data MAY be cached, mirrored, or transported through untrusted channels.

## Separation of Concerns

INSA explicitly separates identity from authentication protocols, messaging systems, transport mechanisms, and application logic. While identity identifiers may be used by higher-layer systems, INSA does not define how identities are authenticated in sessions or how messages are exchanged.

This separation allows INSA to serve as a stable substrate across diverse technical and institutional environments without imposing application-level constraints.

## Extensibility

INSA is designed to be extensible through companion specifications. Additional documents may define endpoint discovery, communication profiles, application bindings, or sector- specific policies without modifying the core architecture defined in this document.

Such extensions MUST NOT alter the semantics of canonical identities, subordinate identities, or sovereign identity trees as defined herein.

# Sovereign Identity Namespace Model

INSA organizes canonical identities within sovereign identity namespaces. A sovereign identity namespace defines the trust boundary under which canonical identities are issued, governed, and published. Each namespace is operated by a sovereign operator and identified by a globally unique namespace suffix.

The namespace model is designed to support decentralization, jurisdictional diversity, and explicit governance without introducing global coordination requirements.

## Namespace Suffixes

Each sovereign identity namespace is identified by a namespace suffix, such as ‘ca.id‘. Namespace suffixes are globally unique identifiers that establish authority over canonical identity publication within their scope.

Namespace suffixes do not imply naming uniqueness at the human-readable level. Human-readable labels MAY collide across namespaces or within a namespace without affecting cryptographic identity uniqueness.

## Authority and Trust Boundaries

A sovereign operator is authoritative only within the namespace suffix it operates. Trust in a sovereign operator does not imply trust in any other operator.

Relying parties MAY choose to trust multiple sovereign operators concurrently or restrict trust to a specific set of namespaces based on policy, jurisdiction, or application context.

INSA does not define a global root authority or hierarchical trust chain across sovereign operators.

## Canonical Identity Allocation

Within a sovereign namespace, canonical identities are allocated according to policies defined by the sovereign operator. These policies MAY include identity proofing, registration requirements, or governance constraints.

# 3. Threat Model

INSA does not mandate any specific identity proofing process. The architecture assumes only that canonical identity manifests published within a sovereign identity tree are governed by the operator responsible for that namespace.

## Name Ambiguity and Human Readability

Human-readable labels associated with canonical identities are descriptive attributes and are not treated as unique identifiers. Multiple canonical identities MAY share identical human-readable names without ambiguity at the cryptographic level.

Relying parties MUST NOT assume uniqueness of human-readable labels. Cryptographic keys and manifest identifiers are the sole authoritative identity anchors.

## Delegation to Role Authorities

Sovereign operators MAY delegate authority to institutions to issue subordinate identities within their namespace. Such delegation does not transfer control over canonical identities.

Delegated role authorities operate under the governance constraints of the sovereign operator and are accountable for subordinate identities they issue.

## Namespace Evolution

Namespace suffixes and governance policies MAY evolve over time. Changes to namespace policies MUST NOT invalidate previously issued canonical identities unless explicitly revoked under sovereign authority.

INSA requires that sovereign operators provide continuity guarantees for canonical identities across namespace evolution events.

## Cross-Namespace Resolution

INSA does not require resolvers to possess prior knowledge of all namespace suffixes. Resolvers MAY obtain namespace metadata dynamically or through local policy configuration.

Cross-namespace resolution does not imply cross-namespace trust. Each resolved identity MUST be evaluated independently according to the relying party’s trust policy.

# Canonical Identity

Canonical identities represent long-lived human identities within INSA. A canonical identity is intended to persist across institutional affiliations, employment changes, and application contexts. Canonical identities form the foundational layer upon which all subordinate identities and institutional roles are built.

Canonical identities are governed within sovereign identity namespaces but are not owned or controlled by institutions.

## Canonical Public Key

Each canonical identity is anchored by a canonical public key. The canonical public key serves as the primary cryptographic identifier for the identity.

The corresponding private key MUST be generated and controlled by the identity holder. Sovereign operators and institutions MUST NOT generate or retain private keys associated with canonical identities.

Loss of control over a canonical private key constitutes loss of control over the associated identity and therefore represents a critical security event.

## Canonical Manifest

Canonical identities are described by canonical manifests. A canonical manifest is a signed data structure that includes, at a minimum:

* the canonical public key * a manifest identifier * optional descriptive attributes * lineage metadata * a validity interval or issuance timestamp * sovereign operator attestation

Canonical manifests MUST be signed by the canonical private key and attested by the sovereign operator through inclusion in a sovereign identity tree.

## Descriptive Attributes

Descriptive attributes associated with canonical identities MAY include human-readable names, photographs, or other metadata. Such attributes are non-authoritative and MUST NOT be treated as identity anchors.

Descriptive attributes MAY change over time without affecting the continuity of the canonical identity.

## Key Rotation and Continuity

Canonical identities support key rotation to mitigate key compromise and cryptographic obsolescence. Key rotation is achieved through issuance of a new canonical manifest that references the previous canonical key via lineage metadata.

Lineage metadata MUST cryptographically bind successive canonical keys to preserve identity continuity. Relying parties MUST validate lineage chains when resolving rotated canonical identities.

Rotation of a canonical key MUST NOT require issuance of a new identity unless explicitly mandated by sovereign policy.

## Revocation and Invalidity

Revocation of a canonical identity is an exceptional event and is governed by sovereign operator policy. Revocation MAY occur in cases of fraud, legal mandate, or explicit identity holder request.

Revocation of a canonical identity invalidates all subordinate identities derived from it.

## Sovereign Operator Role

Sovereign operators are responsible for governing canonical identity issuance, revocation, and publication. Sovereign operators attest to canonical identity inclusion but do not participate in authentication or usage of canonical identities.

Sovereign operators MUST publish signed Merkle roots and associated metadata sufficient to allow independent verification of canonical identity inclusion.

## Independence from Institutions

Canonical identities are independent of institutional systems and roles. Institutions MUST NOT require transfer or escrow of canonical private keys as a condition of service or employment.

This separation preserves human identity sovereignty while enabling institutional accountability through subordinate identities.

# 4. Identity Pressure

# Subordinate Identity

Subordinate identities represent institutionally governed roles associated with a canonical identity. Subordinate identities enable organizations to assert authority, responsibility, and accountability for actions performed by individuals without assuming control over the underlying human identity.

Subordinate identities are explicitly scoped, time-bounded, and revocable.

## Role-Based Identity Model

A subordinate identity corresponds to a specific role, function, or capacity within an institutional context. Examples include employment roles, professional certifications, delegated authorities, or operational permissions.

Subordinate identities are not intended to persist indefinitely and MAY be replaced, renewed, or revoked according to institutional policy.

## Dual Attestation

Creation of a subordinate identity requires dual attestation. A subordinate manifest MUST be signed by:

* the canonical identity holder, indicating acceptance of the role * the issuing role authority, indicating institutional authorization

Dual attestation establishes explicit accountability for both the human actor and the issuing institution.

## Subordinate Manifest

A subordinate manifest is a signed data structure that includes, at a minimum:

* a reference to the associated canonical identity * a role identifier * one or more bound capabilities * validity constraints such as time limits or conditions * signatures from both the canonical identity holder and the role authority

Subordinate manifests MAY include additional metadata defined by institutional policy.

## Renewal and Revocation

Subordinate identities MAY be renewed by issuance of a new subordinate manifest. Renewal does not require modification of the canonical identity.

Revocation of a subordinate identity MUST be performed by the issuing role authority and MUST be discoverable by relying parties during resolution.

Revocation of a subordinate identity does not affect the validity of the associated canonical identity.

## Accountability and Attribution

Actions performed under a subordinate identity are attributable to both the canonical identity holder and the issuing role authority.

Relying parties SHOULD record both attestations when logging or auditing actions performed under subordinate identities.

## Separation from Canonical Identity

Subordinate identities do not confer ownership or control over the canonical identity. Institutions MUST NOT require disclosure of canonical private keys or enforce key escrow as a condition of subordinate identity issuance.

This separation preserves individual sovereignty while enabling institutional governance.

## Expiration Semantics

Subordinate identities SHOULD include explicit expiration semantics. Expired subordinate identities MUST be treated as invalid even if cryptographic signatures remain valid.

Relying parties MUST verify validity intervals during resolution.

# Capability Assignment

Capabilities define the actions a subordinate identity is authorized to perform within a given context. Capabilities provide fine-grained authorization semantics that are evaluated by relying parties independently of identity resolution.

INSA does not define a universal capability language. Instead, it defines the cryptographic binding and verification model required to associate capabilities with subordinate identities in a verifiable manner.

## Capability Semantics

A capability represents permission to perform a specific class of actions under defined conditions. Capability semantics are interpreted by relying systems according to local policy and application requirements.

Capabilities MAY encode scope, constraints, limits, or contextual parameters. INSA does not impose semantic meaning beyond the requirement that capabilities be cryptographically verifiable.

## Binding Capabilities to Subordinate Identities

Capabilities MUST be bound to subordinate identities through subordinate manifests. A subordinate manifest MAY bind one or more capabilities.

Capability bindings MUST be covered by the signatures of both the canonical identity holder and the issuing role authority.

## Capability Evaluation

Relying parties evaluate capabilities when making authorization decisions. Evaluation includes verification of:

* the subordinate manifest signatures * the validity interval of the subordinate identity * the integrity of bound capabilities * any application-specific constraints

INSA does not require relying parties to contact issuing authorities during capability evaluation.

# 5. Document Pressure

## Revocation and Update of Capabilities

Capabilities MAY be revoked or modified by issuing a new subordinate manifest. Revocation of a subordinate identity implicitly revokes all associated capabilities.

Relying parties MUST treat revoked or expired subordinate identities as having no valid capabilities.

## Least Privilege

Issuing role authorities SHOULD apply the principle of least privilege when assigning capabilities. Capabilities SHOULD grant only the minimum permissions necessary to perform the intended role.

## Independence from Authentication Protocols

Capability evaluation is independent of authentication protocols, session management, or transport mechanisms. INSA defines only the authorization data model, not how capabilities are exercised within applications.

## Capability Portability

Capabilities MAY be evaluated across different systems and applications provided that the capability semantics are understood by the relying party. INSA does not restrict capability interpretation to a single platform or vendor.

# IDNS Resolution

IDNS resolution is the process by which a relying party retrieves and verifies identity information using cryptographic proofs rather than trusted online services. Resolution is designed to be stateless, deterministic, and verifiable using data obtained from untrusted sources.

Resolution does not require prior trust relationships between the resolver and sovereign operators, role authorities, or distribution infrastructure.

## Resolution Inputs

An IDNS resolution operation takes the following inputs:

* an identity reference, such as a canonical or subordinate identifier * a namespace suffix identifying the sovereign identity namespace * locally configured trust policy defining acceptable sovereign operators

Resolvers MAY accept additional policy inputs defined by the relying party.

## Resolution Outputs

Successful resolution produces a verified identity result that includes:

* the resolved canonical manifest * zero or more resolved subordinate manifests * verified Merkle inclusion proofs * verified signed Merkle root metadata * validation status and error indicators

Resolvers MUST distinguish between verification failure and absence of identity data.

## Canonical Identity Resolution

To resolve a canonical identity, a resolver performs the following steps:

1. Retrieve the canonical manifest from any available source. 2. Retrieve the corresponding Merkle inclusion proof. 3. Retrieve the signed Merkle root for the relevant sovereign identity tree. 4. Verify the signature on the Merkle root. 5. Verify the inclusion proof against the Merkle root. 6. Verify the signature on the canonical manifest. 7. Apply local trust policy to the sovereign operator signature.

Failure at any step MUST result in resolution failure.

## Subordinate Identity Resolution

To resolve a subordinate identity, a resolver performs the following steps:

1. Resolve the associated canonical identity. 2. Retrieve the subordinate manifest. 3. Verify the subordinate manifest signatures. 4. Verify validity intervals and revocation status. 5. Verify capability bindings.

Resolvers MUST verify that subordinate manifests reference a valid canonical identity.

## Stateless Verification

Resolvers MUST NOT rely on synchronized databases or real-time connections to sovereign operators or role authorities. All retrieved data MAY be cached, mirrored, or relayed through untrusted intermediaries.

Stateless verification ensures that resolution remains robust under network partition, operator outage, or adversarial distribution conditions.

## Caching Semantics

Resolvers MAY cache verified manifests, Merkle roots, and inclusion proofs to improve performance. Cached data MUST be revalidated according to local policy and expiration constraints.

Caching MUST NOT override verification requirements.

## Failure Modes

Resolvers MUST provide explicit failure indicators for the following conditions:

* invalid signatures * invalid Merkle proofs * expired or revoked identities * unsupported namespace suffixes * policy rejection by the relying party

Resolvers SHOULD avoid ambiguous failure states.

## Multiple Namespace Support

Resolvers MAY support multiple sovereign namespaces concurrently. Resolution within one namespace MUST NOT imply trust in any other namespace.

# 6. Institutional Pressure

Trust decisions across namespaces are governed solely by local policy.

## Offline Resolution

Resolvers MAY perform resolution using previously obtained data when network access is unavailable. Offline resolution MUST apply the same verification rules as online resolution and MUST respect validity intervals and revocation information.

# Merkle Tree Construction and Update Semantics

INSA uses Merkle trees as cryptographic commitment structures for publishing canonical identity manifests. Merkle trees provide efficient inclusion proofs, tamper evidence, and stateless verification without requiring global consensus or synchronized replicas.

This section defines the construction, ordering, hashing, and update semantics required for interoperable Merkle tree implementations within INSA.

## Hash Function Requirements

Merkle trees used in INSA MUST employ a cryptographic hash function that provides preimage resistance, second-preimage resistance, and collision resistance.

The selected hash function MUST be specified by the sovereign operator and published as part of namespace metadata. Hash function agility SHOULD be supported to accommodate cryptographic evolution.

## Leaf Encoding

Each leaf node in a sovereign identity tree represents a canonical identity manifest.

Leaf nodes MUST be constructed by hashing a canonical, byte-for-byte serialization of the canonical manifest. The serialization format MUST be deterministic and unambiguous.

Implementations MUST NOT include transport-specific or presentation-specific data in leaf encodings.

## Canonical Ordering

Canonical identity manifests MUST be ordered deterministically prior to Merkle tree construction. Ordering MAY be based on manifest identifiers, canonical public keys, or other deterministic attributes defined by the sovereign operator.

The ordering rule MUST be published and stable across tree updates.

## Tree Construction

Merkle trees MAY use binary or higher-arity branching. The branching factor MUST be consistent within a given sovereign identity tree and documented in namespace metadata.

Interior nodes MUST be constructed by hashing the concatenation of their child node hashes. The concatenation order MUST follow the canonical ordering of child nodes.

## Merkle Root Commitment

The Merkle root represents a cryptographic commitment to the complete set of canonical identity manifests at a specific point in time.

Sovereign operators MUST sign each Merkle root using an operator signing key. The signed Merkle root MUST be published along with sufficient metadata to allow independent verification.

## Inclusion Proofs

An inclusion proof consists of the minimal set of sibling hashes required to reconstruct the Merkle root from a given leaf.

Inclusion proofs MUST be verifiable without access to the full tree and MUST be valid only for the specific Merkle root to which they correspond.

Resolvers MUST reject inclusion proofs that do not validate against the signed Merkle root.

## Tree Updates

Updates to a sovereign identity tree occur when canonical identity manifests are added, updated, or revoked.

Each update produces a new Merkle root. Previous roots MUST remain verifiable and MUST NOT be retroactively modified.

Sovereign operators SHOULD retain historical Merkle roots to support auditability and long-term verification.

## Key Rotation and Tree Updates

Canonical identity key rotation results in issuance of a new canonical manifest and a corresponding tree update. The new manifest MUST reference prior lineage information to preserve identity continuity.

Tree updates resulting from key rotation MUST preserve the validity of historical roots.

## Transparency Compatibility

Merkle trees used in INSA are compatible with transparency log architectures. Sovereign operators MAY publish Merkle roots to external transparency systems to enhance auditability.

INSA does not require global transparency logs or blockchain-based consensus.

## Error Handling

Resolvers MUST detect and reject malformed trees, invalid node hashes, or inconsistent ordering rules. Failure to validate Merkle semantics MUST result in resolution failure.

## Security Considerations for Merkle Trees

Merkle tree integrity depends on correct ordering, deterministic encoding, and secure hash functions. Sovereign operators MUST ensure that tree construction rules are clearly specified and consistently applied.

7. Temporal Pressure

Ambiguity in leaf encoding or ordering MAY enable substitution or equivocation attacks and MUST be avoided.

# Security Considerations

INSA is designed to minimize implicit trust and reduce reliance on centralized services. Security is achieved through cryptographic verification, explicit governance boundaries, and deterministic resolution semantics. This section analyzes relevant threat models and defines required mitigations.

No trust relationship is implied by co residency, shared infrastructure, deployment environment, organizational ownership, or administrative domain.

## Threat Model

INSA assumes an adversarial environment in which network infrastructure, storage systems, and data distribution channels may be compromised or malicious. Adversaries MAY control distribution endpoints, attempt data substitution, replay stale information, or induce confusion across namespaces.

INSA does not assume compromise of cryptographic primitives or secure key generation processes.

## Substitution Attacks

An attacker may attempt to substitute identity manifests, Merkle roots, or inclusion proofs.

Resolvers MUST verify signatures on Merkle roots and manifests and MUST validate inclusion proofs against the signed Merkle root. Failure to validate any cryptographic element MUST result in resolution failure.

Substitution attacks are mitigated by cryptographic binding of manifests to Merkle roots and sovereign operator signatures.

## Replay Attacks

An attacker may attempt to replay outdated but previously valid identity data.

Resolvers MUST verify validity intervals, revocation status, and lineage metadata. Resolvers SHOULD apply local policy to determine acceptable staleness of Merkle roots.

Sovereign operators SHOULD publish root metadata frequently enough to limit replay windows.

## Rogue Sovereign Operators

A sovereign operator may behave maliciously or negligently within its namespace.

INSA does not attempt to prevent malicious sovereign behavior. Instead, it enables relying parties to apply explicit trust policy when selecting acceptable namespaces.

Relying parties MUST NOT assume correctness or benevolence of sovereign operators outside their configured trust policy.

## Compromised Role Authorities

Role authorities may issue unauthorized subordinate identities or capabilities.

Dual attestation mitigates this risk by requiring canonical identity holder signatures. Relying parties SHOULD verify both attestations and apply institutional trust policy when evaluating subordinate identities.

## Canonical Key Compromise

Compromise of a canonical private key enables impersonation of the associated identity.

Canonical key holders MUST protect private keys using appropriate safeguards. Key rotation mechanisms enable recovery from compromise but cannot prevent misuse prior to detection.

Loss of canonical key control represents a severe security event.

## Subordinate Key Compromise

Compromise of subordinate identity keys enables misuse of institutional roles.

Institutions SHOULD limit subordinate identity validity periods and apply least-privilege capability assignment. Revocation mechanisms MUST be discoverable during resolution.

## Namespace Confusion Attacks

An attacker may attempt to exploit human-readable name ambiguity or namespace similarity.

Relying parties MUST treat namespace suffixes and cryptographic identifiers as authoritative and MUST NOT rely on human-readable labels for security decisions.

## Downgrade Attacks

Attackers may attempt to induce resolvers to accept weaker cryptographic parameters.

Resolvers MUST enforce minimum cryptographic requirements and reject unsupported or deprecated algorithms according to local policy.

## Denial of Service Considerations

INSA does not require interactive services during resolution, reducing exposure to denial-of-service attacks. However, large inclusion proofs or excessive resolution requests may impact resolver performance.

Resolvers SHOULD implement reasonable resource limits and caching strategies.

## Auditability and Accountability

Retention of historical Merkle roots and manifests enables post-event audit and forensic analysis. Sovereign operators SHOULD provide sufficient data to support independent audits.

# 8. Systemic Pressure

## Security Summary

INSA's security model relies on explicit trust selection, cryptographic verification, and stateless resolution. Relying parties bear responsibility for trust policy configuration and enforcement.

# Privacy Considerations

INSA is designed to support verifiable identity while minimizing unnecessary disclosure and correlation. Privacy protection is achieved primarily through architectural separation, selective disclosure, and reliance on cryptographic verification rather than centralized query services.

## Separation of Identity Contexts

INSA distinguishes between canonical identities and subordinate identities to reduce cross-context correlation. Canonical identities represent long-lived personhood, while subordinate identities represent scoped, institution-specific roles.

Relying parties SHOULD prefer subordinate identities when interacting in institutional or role-based contexts. Use of canonical identities SHOULD be limited to cases where long-term personhood is explicitly required.

## Minimal Disclosure

Canonical and subordinate manifests MAY include optional descriptive attributes. Such attributes SHOULD be minimized and limited to information strictly necessary for the intended purpose.

Relying parties MUST NOT assume the presence of descriptive attributes and MUST NOT require attributes beyond those necessary for verification and authorization.

## Correlation Risks

Repeated resolution of the same canonical identity across unrelated contexts may enable correlation by observers of resolution activity.

INSA mitigates this risk by enabling stateless resolution and allowing manifests, Merkle roots, and inclusion proofs to be retrieved from untrusted or offline sources. Resolvers MAY cache identity data to reduce observable network activity.

## Namespace-Level Observability

Sovereign operators may observe publication events within their namespace but do not participate in identity resolution. Resolution does not require contacting the sovereign operator in real time.

This design limits operator visibility into relying party behavior and identity usage patterns.

## Offline and Delegated Resolution

Resolvers MAY perform resolution using locally cached data or delegated resolution services. Delegated resolvers MUST NOT be trusted implicitly and SHOULD be treated as untrusted distribution points.

All verification MUST occur locally at the relying party.

## Attribute Evolution

Descriptive attributes associated with identities may change over time. Attribute updates MUST NOT invalidate identity continuity and SHOULD be treated as non-authoritative metadata.

## Relationship to Companion Specifications

Endpoint discovery, messaging, and communication profiles introduce additional privacy considerations beyond the scope of this document. Such considerations are addressed in companion specifications and MUST NOT alter the privacy guarantees described herein.

## Privacy Summary

INSA provides structural privacy protections through separation of concerns, stateless verification, and minimization of required disclosure. Privacy outcomes ultimately depend on relying party policy and implementation choices.

# IANA Considerations

This document defines architectural elements that require coordinated registration to ensure interoperability across independent implementations. This section specifies the IANA registries required by INSA and the policies governing their maintenance.

## URI Scheme Registration

This document requests registration of the ‘id‘ URI scheme in the IANA URI Schemes registry in accordance with <<rfc3986>>.

### Scheme Name

‘id‘

### Status

Permanent

### Scheme Syntax

The general syntax of an ‘id‘ URI is:

id://‘<identifier>‘

The ‘<identifier>‘ component is interpreted according to INSA resolution rules and MAY include namespace suffixes, canonical identifiers, or subordinate identifiers.

### Scheme Semantics

The ‘id‘ URI scheme identifies an INSA identity reference. It does not imply transport, location, or communication semantics. Resolution of an ‘id‘ URI yields identity data and verification artifacts, not network endpoints.

### Encoding Considerations

## 9. Architectural Implications

The ‘id‘ URI scheme conforms to the encoding requirements defined in <<rfc3986>>.

### Security Considerations

Security considerations for the ‘id‘ URI scheme are described throughout this document, including Sections on resolution, Merkle verification, and trust policy.

---

## Sovereign Namespace Suffix Registry

This document requests the creation of an IANA registry for Sovereign Identity Namespace Suffixes.

### Registry Purpose

The registry records namespace suffixes corresponding to sovereign identity namespaces. Each entry represents a trust boundary operated by a sovereign operator.

### Registry Contents

Each registry entry MUST include:

* Namespace suffix (for example, ‘ca.id‘) * Sovereign operator name * Reference to operator policy documentation * Cryptographic metadata endpoint or reference * Contact information for operational issues

### Registration Policy

Registration of namespace suffixes MUST follow a policy of **Expert Review** as defined in <<rfc8126>>.

Expert review ensures that suffixes are unique, well-formed, and associated with a clearly identified sovereign operator.

### Update and Revocation

Registry entries MAY be updated to reflect changes in operator metadata. Revocation or deprecation of a namespace suffix MUST be clearly indicated and MUST NOT invalidate previously issued canonical identities unless explicitly stated by sovereign policy.

---

## Extension Registries

Future specifications MAY define additional IANA registries related to INSA, including but not limited to:

* identity manifest schema identifiers * capability schema identifiers * resolution metadata formats

Such registries MUST be defined in companion specifications and MUST NOT modify the core registries defined in this document.

---

## No Global Root Registry

INSA explicitly does not require a global root authority or hierarchical registry beyond the namespace suffix registry defined above. The absence of a global root registry is a deliberate design choice intended to preserve decentralization and sovereign autonomy.

---

## IANA Summary

The IANA actions requested by this document are limited in scope and designed to support interoperability without introducing centralized control over identity issuance or resolution.

# Summary

The Identity Namespace System Architecture defines a durable, verifiable, and sovereign identity substrate grounded in cryptographic proofs rather than centralized services. By separating canonical human identities from institutionally governed subordinate roles, INSA enables accountability without compromising individual sovereignty.

INSA’s use of signed manifests, Merkle-rooted registries, and stateless resolution allows relying parties to independently verify identity information obtained from untrusted sources. Explicit trust boundaries defined by sovereign namespaces replace implicit global trust assumptions, enabling decentralized governance and jurisdictional diversity.

This document intentionally limits its scope to identity representation, publication, and resolution. Endpoint discovery, messaging, and communication semantics are addressed in separate companion specifications to preserve architectural clarity and long-term stability.

Together, these design principles position INSA as a foundational identity layer suitable for civil, commercial, and personal use in a globally distributed environment.

# References

## Normative References

<<rfc2119>> Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

<<rfc8174>> Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, May 2017.

<<rfc3986>> Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

<<rfc8259>> Bray, T., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, December 2017.

## Informative References

<<rfc6962>> Laurie, B., Langley, A., and E. Kasper, "Certificate Transparency", RFC 6962, June 2013.

<<rfc8446>> Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, August 2018.

<<merkle1987>> Merkle, R. C., "A Digital Signature Based on a Conventional Encryption Function", Advances in Cryptology CRYPTO 87, Lecture Notes in Computer Science, vol. 293, 1987.

<<fips180-4>> National Institute of Standards and Technology, "Secure Hash Standard", FIPS PUB 180-4, August 2015.

<<fips186-5>> National Institute of Standards and Technology, "Digital Signature Standard (DSS)", FIPS PUB 186-5, 2023.

<<nist-pqc>> National Institute of Standards and Technology, "Post-Quantum Cryptography Standardization Project", https://csrc.nist.gov/projects/post-quantum-cryptography.

<<idns-messaging>> REDACTED, "Identity Endpoint Discovery and Communication Profiles", Work in Progress (companion document).