Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: 17 September 2026

A. Haberkamp
KH Sovereign, Inc.

March 2026

Intent Provenance Protocol (IPP)
draft-haberkamp-ipp-00

Abstract

This document specifies the Intent Provenance Protocol (IPP), a cryptographic infrastructure standard for carrying verified human intent through chains of autonomous artificial intelligence agent actions. IPP defines the Intent Token -- a signed, bounded, and tamper-evident data structure that travels with every agentic action, preserving an unbroken, auditable lineage from the originating human principal to each terminal action executed on their behalf.

As AI agents become primary actors in enterprise environments -- executing transactions, accessing sensitive data, orchestrating sub-agents, and operating across organizational boundaries -- the absence of a shared trust substrate creates systemic risk to organizational accountability, regulatory compliance, and legal liability attribution. IPP addresses this gap by establishing a protocol layer that operates above cryptographic authentication and below application logic, making human intent a first-class, verifiable primitive in agentic systems.

IPP introduces four foundational properties -- Lineage, Boundedness, Non-repudiation, and Interoperability -- enforced through a combination of Ed25519 digital signatures, Decentralized Identifiers (DIDs), and a Narrowing Invariant that prevents any derived token from exceeding its parent’s authorized scope. The protocol is framework-agnostic, cloud-agnostic, and designed for open implementation across the ecosystem of AI orchestration platforms.

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/.

Copyright Notice

Table of Contents

1. Introduction

1.1. Background and Motivation

For three decades, enterprise security has operated under a single foundational assumption: that every consequential digital action originates from, and is ultimately accountable to, a human being. Access control systems, identity governance frameworks, audit logging infrastructures, and regulatory compliance regimes all derive their validity from this assumption. A human authenticates. A human is authorized. A human acts. A human is accountable.

The emergence of autonomous artificial intelligence agents -- software entities capable of perceiving their environment, reasoning about goals, taking sequences of actions, and spawning additional agents to fulfill sub-tasks -- fundamentally invalidates this assumption. In agentic environments, the majority of consequential actions are executed by software entities operating without real-time human supervision. A human sets a goal; an agent, or a hierarchy of agents, determines and executes the means.

This creates a category of risk that existing security infrastructure is architecturally unequipped to address: not the risk of unauthorized access, but the risk of authorized-but-misaligned action -- agent behavior that is technically permitted by access control systems but inconsistent with the human intent that originally authorized the agent's deployment.

1.2. The Authorship Problem in Agentic Systems

Consider a scenario in which a Chief Financial Officer instructs an AI assistant to "optimize cash positions across all subsidiaries and move any idle balances over ten million dollars into short-term treasuries." The assistant spawns three sub-agents: one to query treasury balances, one to evaluate current rates, and one to execute transfers. The transfer agent, drawing on outputs from the other two, initiates eleven separate wire transfers totaling three hundred forty million dollars.

At the moment the eleventh transfer is executed, no existing system can answer the following questions with cryptographic certainty:

a. What specific human intent authorized this action?

b. What constraints bounded that intent?

c. What is the complete chain of delegation from the CFO’s instruction to this specific transfer?

d. Is this action consistent with what the CFO meant, or merely consistent with what the CFO said?

e. Who bears legal accountability if this action causes harm?

The Intent Provenance Protocol is designed to make all five of these questions answerable with cryptographic precision, in real time, without requiring a central authority, and in a form that is legally defensible across jurisdictions.

1.3. Scope

This specification defines: the structure and semantics of the Intent Token; the Genesis Seal mechanism for permanent authorship attribution; the cryptographic mechanisms underlying token signing and verification; the Narrowing Invariant governing token derivation; the revocation protocol; the provenance chain format; and the interoperability requirements for compliant implementations.

This specification does not define: specific AI agent architectures or orchestration frameworks; natural language processing mechanisms for intent parsing; application-layer authorization policies; or legal standards for accountability attribution (though it is designed to support such standards).

2. Terminology and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Agent:
: An autonomous software entity capable of perceiving inputs, reasoning about goals, taking sequences of actions, and potentially spawning additional agents. Agents are the primary non-human actors governed by this protocol.

Principal:
: A human being or legally accountable organization at the root of an intent chain. Every valid Intent Token chain MUST have a human Principal at its origin. There is no such thing as an agent-originated intent chain under this protocol.

Intent Token:
: The atomic unit of this protocol. A cryptographically signed, bounded, and time-limited data structure that carries verified human intent through a chain of agentic actions.

Genesis Seal:
: A cryptographic artifact embedded in every Intent Token that permanently links the token to the original IPP specification and its author, Amanda Haberkamp. The Genesis Seal contains a hash of the specification document signed by the Founding Key.

Founding Key:
: The Ed25519 private key generated by Amanda Haberkamp during the Key Generation Ceremony and used exclusively to produce the Genesis Seal signature. This key is never used after the ceremony and is stored in offline cold storage.

Narrowing Invariant:
: The protocol rule that a Derived Token MUST be strictly less than or equal to its Parent Token in every dimension of scope, delegation depth, and expiry. Violations of this invariant are detectable by any participant without contacting a central authority.

Derived Token:
An Intent Token produced by an agent from a Parent Token. A Derived Token carries narrowed scope and reduced delegation depth, and is signed by both the issuing agent and linked to the Parent Token's signature.

Delegation Depth:
A non-negative integer field in the Intent Token specifying how many additional levels of sub-agent spawning are authorized. A value of zero means the token holder may act but may not spawn sub-agents.

Provenance Chain:
An append-only, cryptographically linked sequence of records within an Intent Token documenting every action taken under that token's authority.

Domain Taxonomy:
The hierarchical dot-notation vocabulary for classifying intent domains, maintained by KH Sovereign, Inc. and open for community contribution. Examples: financial.treasury, healthcare.records.read, infrastructure.compute.provision.

DID:
Decentralized Identifier. A W3C standard identifier that enables verifiable, self-sovereign identity without relying on a centralized registry. All Principal and Agent identities in this protocol are expressed as DIDs.

Revocation Registry:
A distributed service providing real-time token revocation status. Agents MUST check the registry before taking any action under a token.

3. Protocol Overview

3.1. Design Principles

IPP is designed according to the following principles:

* *Decentralized enforcement:* Token validity is verifiable by any participant without contacting a central authority. The cryptographic structure of the token itself makes violations detectable.

* *Human primacy:* Every intent chain MUST originate from a human Principal. This is not a policy choice -- it is a structural requirement enforced by the token schema.

* *Minimal footprint:* The SDK integration surface is intentionally small. Developers add governance capability to existing agents without rewriting their architecture.

* *Open interoperability:* The protocol is framework-agnostic, cloud-agnostic, and jurisdiction-agnostic. Any conformant implementation can interoperate with any other.

* *Permanent authorship:* The Genesis Seal embeds cryptographic attribution to the protocol's author in every token, making the authorship record tamper-evident and permanent by design.

3.2. The Four Foundational Properties

Every compliant IPP implementation MUST guarantee the following four properties:

Lineage:
: Every action taken by any agent under an Intent Token MUST be traceable, through an unbroken chain of cryptographic signatures, to a human Principal. The chain may pass through any number of intermediate agents, but the terminal node in the backwards trace MUST always be a DID resolving to a human or human-accountable legal entity.

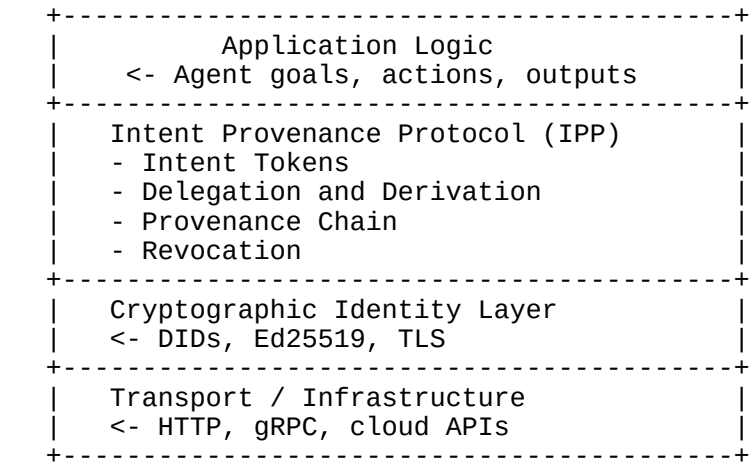
Boundedness:
: Every Intent Token MUST carry explicit, machine-readable constraints on the scope of authorized action. These constraints MUST be enforced by the SDK before any action is taken. Constraints travel with the token through every delegation level and MUST NOT be expanded by any intermediate agent.

Non-Repudiation:
: Every Intent Token MUST be cryptographically signed such that the signing Principal or Agent cannot credibly deny having issued the token. The signature MUST be verifiable by any third party using only the signer's public key, without requiring communication with the signer or any central authority.

Interoperability:
: Compliant implementations MUST be capable of consuming and producing Intent Tokens regardless of the AI framework, cloud environment, or programming language used. The token format is defined in JSON with a canonical serialization for signature purposes.

3.3. Protocol Layers

IPP operates as a distinct protocol layer between cryptographic identity infrastructure and application logic:



4. The Intent Token

4.1. Token Structure

The Intent Token is a JSON object with a canonical structure. The following is a complete example of a root Intent Token issued by a Principal:


```
{
  "$schema": "https://ipp.khsovereign.com/schema/v0.1/intent-token.json",
  "version": "0.1",
  "genesis": {
    "spec_hash": "sha3-256:a3f9c2d8e1b74f6a...",
    "author_did": "did:key:z6MkHaberkamp...",
    "author_name": "Amanda Haberkamp",
    "org": "KH Sovereign, Inc.",
    "genesis_sig": "ed25519:BASE64URL..."
  },
  "token_id": "ipp:tok:550e8400-e29b-41d4-a716-446655440000",
  "schema_version": "0.1",
  "created_at": "2026-09-01T14:32:00Z",
  "expires_at": "2026-09-01T22:32:00Z",
  "principal": {
    "did": "did:key:z6MkPrincipalXXX...",
    "name": "Amanda Haberkamp",
    "org": "KH Sovereign, Inc.",
```

```
    "legal_jurisdiction": "US-IL",
    "signature": "ed25519:BASE64URL..."
  },
  "intent": {
    "natural_language": "Optimize cash positions across subsidiaries",
    "domain": "financial.treasury",
    "resource_scope": ["subsidiary:*", "account_type:cash"],
    "quantitative_bounds": {
      "min_balance_threshold": 10000000,
      "currency": "USD",
      "max_single_transaction": 50000000,
      "time_window": "business_hours_CT"
    },
    "prohibited_actions": ["equity_purchase", "account_closure"]
  },
  "delegation": {
    "parent_token_id": null,
    "depth_remaining": 3,
    "depth_original": 3,
    "agent_id": "ipp:agent:langchain:treasury-optimizer-v1",
    "agent_framework": "langchain",
    "agent_version": "0.1.0",
    "spawned_by_principal": true
  },
  "revocation": {
    "registry_endpoint": "https://revoke.khsovereign.com/v1",
    "token_id_hash": "sha3-256:HASH...",
    "check_interval_ms": 5000
  },
  "provenance_chain": [],
  "token_signature": "ed25519:BASE64URL_SIGNED_BY_PRINCIPAL"
}
```


4.2. The Intent Scope Envelope

The intent field encodes the human's goal as a Structured Intent Envelope. The natural_language field is human-readable but NOT machine-enforceable. Enforcement is performed against the structured fields only.

The domain field value MUST be drawn from the IPP Domain Taxonomy maintained at https://ipp.khsovereign.com/taxonomy. Implementations using non-standard domain values MUST prefix them with "x." (e.g., x.mycompany.custom_domain) to avoid collision with future taxonomy additions.

4.3. Quantitative Bounds

The quantitative_bounds object is domain-specific. Standard bound fields for the financial domain are defined below. Domain-specific extensions are published in the IPP Domain Taxonomy registry.

```
"quantitative_bounds": {
  "min_balance_threshold": integer,
  "max_single_transaction": integer,
  "max_total_exposure": integer,
  "currency": "ISO 4217 code",
  "time_window": "named window or ISO 8601 interval",
  "geographic_restriction": ["ISO 3166-1 alpha-2 codes"],
  "counterparty_allowlist": ["DID or entity identifier"],
  "counterparty_blocklist": ["DID or entity identifier"]
}
```

5. The Genesis Seal

5.1. Purpose and Design

The Genesis Seal is a cryptographic artifact embedded in every Intent Token that permanently and irrevocably links the token to this specification and to its author, Amanda Haberkamp. The Genesis Seal serves three simultaneous functions:

* *Protocol versioning:* The spec_hash field uniquely identifies the exact version of the IPP specification under which the token was issued.

* *Authorship attribution:* The author_did and genesis_sig fields constitute a permanent, cryptographically verifiable record that Amanda Haberkamp is the originator of this protocol.

* *Ecosystem integrity:* Because the Genesis Seal is required in all tokens, any token that omits or falsifies it is detectable as non-compliant without reference to any external authority.

The genesis_sig is produced using the Founding Key -- an Ed25519 private key generated by Amanda Haberkamp and used ONLY ONCE, during the Key Generation Ceremony described in Section 5.3. The private component of this key is stored in offline cold storage and is never used again. The public component is published in this specification and in the IPP specification repository.

5.2. Genesis Seal Structure

```
"genesis": {
  "spec_hash":      "sha3-256:[HASH OF IPP SPECIFICATION DOCUMENT]",
  "author_did":     "did:key:z6MkHaberkampXXXXXXXXXXXXXXXXXXXXXXX",
  "author_name":    "Amanda Haberkamp",
  "org":            "KH Sovereign, Inc.",
  "founding_pubkey":"ed25519-pub:[BASE64URL PUBLIC KEY]",
  "genesis_sig":    "ed25519:[BASE64URL SIGNATURE]"
}

Where genesis_sig = Ed25519Sign(
  private_key = FOUNDING_PRIVATE_KEY,
  message     = SHA3-256(spec_hash || author_did
                         || author_name || org || timestamp)
)
```

Verification of the Genesis Seal MUST proceed as follows:

1. Retrieve the IPP specification from the canonical location: https://ipp.khsovereign.com/spec/v0.1

2. Compute SHA3-256 of the specification document and compare with spec_hash. If they do not match, the token was issued under a modified or forged specification.

3. Retrieve Amanda Haberkamp’s founding public key from the specification repository at https://ipp.khsovereign.com/keys/founding_public.pem

4. Verify the genesis_sig using the founding public key. If verification fails, the Genesis Seal is invalid.

5. If both checks pass, the token is certified as issued under the authentic IPP specification authored by Amanda Haberkamp.

5.3. Key Generation Ceremony

The Founding Key pair was generated according to the following procedure, designed to maximize security and establish a clear, witnessed chain of custody:

1. An air-gapped machine (disconnected from all networks) was used for all key generation operations.

2. The Ed25519 key pair was generated using OpenSSL 3.x:

```
openssl genpkey -algorithm ed25519 \
  -out amanda_haberkamp_founding_private.pem

openssl pkey -in amanda_haberkamp_founding_private.pem \
  -pubout \
  -out amanda_haberkamp_founding_public.pem
```

3. A DID was generated from the public key using the did:key method.

4. The Genesis Seal payload was constructed and signed using the private key.

5. The private key was printed to paper (two copies), placed in fireproof storage in separate physical locations, and deleted from all digital media.

6. The public key, DID, and Genesis Seal were committed to the IPP specification repository with a cryptographic timestamp.

SECURITY NOTE: The Founding Private Key is used exactly once. If the Founding Private Key is ever compromised, a Key Compromise event MUST be declared via a signed notice published at https://ipp.khsovereign.com/security, and a successor key ceremony MUST be conducted.

5.4. Authorship Attribution Requirements

Any implementation, derivative protocol, or software product that uses, embeds, or interoperates with Intent Tokens MUST include the following attribution:

Implements the Intent Provenance Protocol (IPP) v0.1,
authored by Amanda Haberkamp, KH Sovereign, Inc., 2026.
https://ipp.khsovereign.com

6. Cryptographic Mechanisms

6.1. Signature Algorithm

All signatures in this protocol MUST use Ed25519 as defined in [RFC8032]. Signature values are encoded as Base64URL without padding, prefixed with "ed25519:".

The canonical serialization for signing MUST be produced by: (1) removing the token_signature field, (2) sorting all keys lexicographically at every nesting level, (3) serializing to JSON with no extraneous whitespace, and (4) encoding as UTF-8.

6.2. Decentralized Identifiers (DIDs)

All Principal and Agent identities MUST be expressed as DIDs conforming to the W3C DID Core specification [W3C-DID]. This specification RECOMMENDS the did:key method for its self-sovereign properties. Implementations MAY additionally support did:web, did:ion, or other conformant DID methods provided they satisfy the following requirements: (a) the DID is resolvable without requiring communication with the token issuer; (b) the resolved DID document contains a verificationMethod with the public key material; and (c) the DID is persistent.

6.3. Hash Functions

All hash operations in this specification use SHA3-256 [FIPS202]. Hash values are represented as lowercase hexadecimal strings prefixed with "sha3-256:".

6.4. The Narrowing Invariant

For any Derived Token D with Parent Token P, the following MUST hold:

```
D.expires_at            <= P.expires_at
D.delegation.depth      <  P.delegation.depth
D.intent.domain         is a sub-domain of P.intent.domain
D.intent.resource_scope  ⊆ P.intent.resource_scope
D.intent.quant_bounds    ⊆ P.intent.quant_bounds
D.intent.prohibited      ⊇ P.intent.prohibited_actions
```

Any receiving agent MUST independently verify this invariant before accepting a Derived Token. Verification failure MUST result in token rejection.

7. Delegation and Derived Tokens

7.1. Derivation Rules

When an agent derives a child token from a parent, it MUST:

1. Set parent_token_id to the token_id of the parent.

2. Reduce depth_remaining by exactly one.

3. Set the intent scope to a subset of the parent’s scope, satisfying the Narrowing Invariant.

4. Set expires_at to be no later than the parent’s expires_at.

5. Set the agent_id to a unique identifier for the spawned sub-agent.

6. Compute and embed a valid narrowing_proof.

7. Sign the new token with the parent agent's private key.

8. Propagate the genesis field unchanged from the parent token. The Genesis Seal is never re-derived.

7.2. Delegation Depth

The depth_remaining field represents the number of additional sub-agent spawning levels authorized. An agent holding a token with depth_remaining = 0 MAY act under the token but MUST NOT issue Derived Tokens.

7.3. Sub-Agent Spawning

An agent spawning a sub-agent MUST provide the Derived Token to the sub-agent as its authorization credential. The sub-agent MUST NOT accept instructions that are inconsistent with its token's intent scope, even if those instructions come from the spawning parent agent. The token is the authority -- not the parent agent.

CRITICAL SECURITY PROPERTY: An agent MUST refuse instructions from any source -- including its parent agent -- that would require it to act outside its token's authorized scope.

8. Revocation

8.1. Revocation Registry

The Revocation Registry accepts token revocation notices from authorized principals and returns revocation status for token IDs. Revocation is keyed on token_id_hash -- the SHA3-256 hash of the token_id -- to preserve privacy while maintaining revocability.

8.2. Polling Requirements

Agents MUST poll the revocation registry at the interval specified by check_interval_ms before taking any action. The RECOMMENDED default interval is 5000 milliseconds. If the revocation registry is unreachable, agents MUST NOT proceed with actions unless an offline_grace_period_ms is specified and has not been exceeded.

8.3. Mid-Chain Revocation

Revocation of a parent token MUST propagate to all derived tokens. The registry MUST maintain the full ancestry tree of token derivation so that when a root or intermediate token is revoked, all descendant tokens are simultaneously marked as revoked.

9. Provenance Chain

9.1. Append-Only Structure

The provenance_chain is an append-only array. Agents MUST append a record to the chain before taking any action under the token. Records MUST NOT be modified or removed once appended. Each record is signed by the acting agent, creating a cryptographically linked sequence of accountability records.

9.2. Provenance Record Format

```
{
  "record_id":      "ipp:pr:uuid4",
  "token_id":       "ipp:tok:parent-token-id",
  "agent_id":       "ipp:agent:uuid4",
  "timestamp":      "ISO8601_UTC",
  "action_type":    "financial.treasury.transfer",
  "action_summary": "Human-readable description of action",
  "resource_id":    "Identifier of resource acted upon",
  "outcome":        "success | failure | partial | blocked",
  "within_bounds":  true,
  "agent_sig":      "ed25519:BASE64URL"
}
```

10. Interoperability

10.1. Framework Integration

The IPP SDK MUST provide integration adapters for the following AI orchestration frameworks at minimum: LangChain (Python and JavaScript), AutoGen (Microsoft), CrewAI, Microsoft Semantic Kernel, and Google Vertex AI Agent Builder. Integration MUST NOT require rewriting existing agent logic.

10.2. Cross-Organization Trust

When an agent from Organization A must interact with an agent from Organization B, the inter-organization trust handshake proceeds as follows:

1. Organization A's agent presents its Derived Token to Organization B's agent.

2. Organization B's agent verifies the Genesis Seal, validates the token signature chain back to a Principal DID, verifies the token has not expired, and checks revocation status.

3. Organization B's agent verifies that the requested interaction falls within the scope defined in the presented token.

4. If all checks pass, Organization B's agent may respond and optionally issue its own Derived Token for downstream actions.

5. Both agents append provenance records to their respective tokens for the interaction.

10.3. Domain Taxonomy

The IPP Domain Taxonomy is maintained at https://ipp.khsovereign.com/taxonomy as an open standard with community contribution via pull request. KH Sovereign, Inc. serves as the taxonomy steward. Current top-level domains include:

```
financial.*      - Financial operations (treasury, payments, trading)
healthcare.*     - Healthcare data and clinical operations
infrastructure.* - Cloud and system infrastructure operations
analytics.*      - Data analysis and reporting
communications.* - Email, messaging, external communications
hr.*             - Human resources and personnel operations
legal.*          - Legal document and compliance operations
security.*       - Security monitoring and response operations
x.*              - Private/experimental namespace (not standardized)
```

11. Security Considerations

Implementers MUST be aware of the following security considerations:

Private Key Compromise:
: If a Principal's private key is compromised, an attacker can issue Intent Tokens on behalf of that Principal. Revocation of all tokens issued under the compromised key MUST be performed immediately.

Replay Attacks:
: Intent Tokens are time-bounded by the expires_at field. Implementations MUST reject expired tokens. The token_id SHOULD be checked against a short-term cache of recently used token IDs to prevent replay of valid, non-expired tokens.

Scope Creep:
The Narrowing Invariant prevents scope expansion in derived tokens. However, implementations MUST verify the invariant independently -- they MUST NOT trust the narrowing_proof field without cryptographic verification.

Registry Availability:
The Revocation Registry is a critical dependency. Implementations MUST design for registry unavailability and MUST fail safe when the registry is unreachable and no offline grace period applies.

12. Privacy Considerations

Intent Tokens contain sensitive information about organizational operations. Implementations MUST transmit tokens only over encrypted channels (TLS 1.3 minimum). The Revocation Registry uses token_id_hash rather than token_id to prevent the registry operator from building a map of active tokens. Implementations MUST NOT send the full token to the revocation registry.

13. IANA Considerations

This document requests registration of the URI scheme "ipp:" for use as the prefix for Intent Token identifiers (ipp:tok:*), Agent identifiers (ipp:agent:*), and Provenance Record identifiers (ipp:pr:*). This document also requests registration of the media type application/ipp+json for serialized Intent Token payloads.

14. References

14.1. Normative References

[FIPS202] NIST, "SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions", FIPS PUB 202, August 2015, <https://doi.org/10.6028/NIST.FIPS.202>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC3161] Adams, C., "Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)", RFC 3161, August 2001, <https://www.rfc-editor.org/rfc/rfc3161>.

[RFC8032] Josefsson, S. and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)", RFC 8032, January 2017, <https://www.rfc-editor.org/rfc/rfc8032>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[W3C-DID] Sporny, M., "Decentralized Identifiers (DIDs) v1.0", July 2022, <https://www.w3.org/TR/did-core/>.

14.2. Informative References

[RFC6749] Hardt, D., "The OAuth 2.0 Authorization Framework", RFC 6749, October 2012, <https://www.rfc-editor.org/rfc/rfc6749>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, August 2018, <https://www.rfc-editor.org/rfc/rfc8446>.

Author's Address

Amanda Haberkamp
KH Sovereign, Inc.
Chicago, Illinois
United States
Email: amanda@khsovereign.com
URI: https://ipp.khsovereign.com