Web 7.0 Foundation Governance Council
Internet-Draft
Intended status: Informational
Expires: 27 September 2026

M. Herman
Web 7.0 Foundation

26 March 2026

Verifiable Trust Circles (VTCs) using VC Proof Sets
draft-herman-vtc-proof-sets-01

Abstract

This document specifies Web 7.0 Verifiable Trust Circles (VTCs), a generalized mechanism for expressing verifiable multi-party membership, belonging, and trust relationships using the W3C Verifiable Credentials (VC) Data Model 2.0 [W3C.VC-DATA-MODEL] and VC Data Integrity Proof Sets [W3C.VC-DATA-INTEGRITY]. VTCs extend the Partof Architecture Reference Model (PARM) to provide a universal credential pattern that subsumes prior pairwise constructs (Personhood Credentials (PHCs) and Verifiable Relationship Credentials (VRCs)) and additionally supports voting-based decision making, meeting requests, task forces, and digital societies.

Derivation Notice

This note is to be removed before publishing as an RFC.

This Internet-Draft is derived from the Web 7.0 Foundation specification "SDO: Verifiable Trust Circles (VTCs) using VC Proof Sets (Web 7.0)" [WEB70-VTC] authored by Michael Herman, published 26 March 2026, and from community discussion at the Trust over IP Foundation Digital Trust Graph Working Group (DTGWG) Credentials Task Force, GitHub Discussion #8 [DISCUSSION-8].

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

The Web 7.0 paradigm seeks to establish a decentralized, agent-centric, privacy-preserving digital society. Central to this vision is the ability of digital entities - people, organizations, and autonomous agents - to form verifiable groups: trust circles that are cryptographically provable, privacy-respecting, and composable.

Prior specifications in the Trust over IP (ToIP) ecosystem defined pairwise constructs - Personhood Credentials (PHCs) and Verifiable Relationship Credentials (VRCs) - to link pairs of entities. While useful, these constructs are insufficient to describe multi-party group membership, community affiliation, or collective decision-making.

This specification introduces Verifiable Trust Circles (VTCs), which generalize pairwise credentials into an N-party construct using the standard W3C VC Proof Set mechanism. A single VTC credential can represent a self-credential (N=1), a bilateral relationship (N=2), or any multi-member group (N>2), enabling a single, coherent model for all membership-like relationships.

> NOTE: Proof Sets are a normative feature of the W3C VC Data Integrity specification and are explicitly designed for scenarios in which the same data needs to be secured by multiple entities. VTCs leverage this mechanism rather than inventing new cryptographic primitives.

1.1. Motivation

The following observations motivate this specification:

* PHCs and VRCs both express a form of "belonging to" - they are specializations of the same universal pattern.

* The W3C VC Data Model 2.0 already provides Proof Sets as a standard mechanism for multi-party signing.

* A single, generalized VTC pattern - grounded in First Principles Thinking - can subsume both constructs and additionally support voting, community membership, digital governance, and inter-network trust.

* The SSC 7.0 Metamodel defines three controller layers (Beneficial, Intermediate, Technical) at which VTCs may apply, enabling rich composability.

1.2. Scope

This specification defines:

1. The VTC data model, including required and optional properties.

2. The roles of Initiator, Responder(s), and Notary within a VTC.

3. The lifecycle of a VTC Proof Set, from initial issuance through multi-party endorsement.

4. Use case profiles: self-credential, bilateral relationship, multi-party group, and voting scenario.

5. Privacy and security considerations specific to multi-party proof sets.

This specification does not define transport protocols, DID method requirements, or verifiable presentation formats, except where necessary to illustrate the VTC pattern.

2. Conventions 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.

Unless stated otherwise, terms have the meanings assigned in the W3C Verifiable Credentials Data Model 2.0 [W3C.VC-DATA-MODEL].

3. Terminology and Definitions

Verifiable Trust Circle (VTC): A Verifiable Credential whose credential subject identifies a multi-party trust relationship, and whose proof property contains a Proof Set with one proof contribution per participating member, plus the Notary’s initial proof.

Web 7.0 Verifiable Trust Circles (VTCs): The generalized name for the VTC pattern when applied to the broader class of MemberOf, PartOf, and CitizenOf relationships. A VTC is a Universal Membership Credential (UMC).

Proof Set: As defined in W3C VC Data Integrity [W3C.VC-DATA-INTEGRITY], a set of proofs attached to a single secured document where the order of proofs does not matter. Each proof is contributed by a distinct signer.

Initiator (A): The entity that proposes or originates a VTC. Identified by a DID. Corresponds to the "from" role in VTC credential subject properties.

Responder (B, ..., Z): One or more entities that accept membership in a VTC by contributing their cryptographic proof to the Proof Set. Identified by DIDs. Corresponds to entries in the "to" array.

Notary (N): A trusted third party - trusted by both the Initiator and all Responders - that issues the initial credential shell and contributes the first proof. The Notary is assigned to the VC "issuer" role. In some use cases the Notary MAY be the Initiator or a Responder, provided they play both roles distinctly.

PARM: Partof Architecture Reference Model. The universal pattern underlying VTCs, encompassing MemberOf, CitizenOf, and PartOf relationships.

SSC 7.0 Metamodel: Self-Sovereign Control 7.0 Metamodel. Defines three controller layers - Beneficial Controller, Intermediate Controller (Agent), and Technical Controller (Agent) - at which VTCs may be anchored.

DTG: Digital Trust Graph. A graph of trust relationships between entities, each edge of which may be represented by a VTC.

PHC: Personhood Credential. A pairwise credential representing proof of personhood; a degenerate VTC where N=1.

VRC: Verifiable Relationship Credential. A pairwise credential representing a bilateral relationship; a degenerate VTC where N=2.

DID: Decentralized Identifier, as defined in [W3C.DID-CORE].

4. Design Principles

This specification adheres to the following design principles, consistent with the ToIP DTGWG Design Principles [DTGWG-DESIGN]:

4.1. As Simple As Possible But No Simpler

VTCs are grounded in existing W3C VC standards. No new cryptographic primitives or credential types are defined. The only structural addition is the deliberate use of the proof array (Proof Set) to carry per-member proofs alongside the Notary proof.

4.2. First Principles Thinking

PHCs and VRCs are recognized as specializations of a single underlying relationship pattern (PARM). Rather than defining multiple credential types for essentially the same concept, this specification derives one universal type that covers all cases by varying the cardinality of the "to" array and the composition of the Proof Set.

4.3. Privacy by Design

VTC credential subjects SHOULD use confidentialSubject semantics wherever selective disclosure is required. Members of a VTC should be able to prove membership to a verifier without unnecessarily revealing the full membership list. Zero-Knowledge Proof (ZKP) integration in Proof Sets is explicitly supported and encouraged.

4.4. Composability

VTCs compose at each layer of the SSC 7.0 Metamodel. A VTC at the Beneficial Controller layer expresses human-level trust relationships; one at the Intermediate Agent layer expresses agent-level relationships; one at the Technical Controller layer expresses device/key-level relationships.

4.5. Cross-Network Trust

The PARM model is network-agnostic. The same VTC pattern supports trust relationships across and between independent, distinct networks and ecosystems.

5. The Partof Architecture Reference Model (PARM)

The Partof Architecture Reference Model (PARM) provides the conceptual foundation for VTCs. It observes that a large class of real-world relationships - membership, citizenship, parthood, employment, and participation - share a common logical structure. Representative relationship types and examples are shown below:

Relationship Type	Example
MemberOf	Alice is a member of the Working Group Trust Circle.
PartOf	Bob is part of the study group.
CitizenOf	Carol is a citizen of the Digital Nation State of Sovronia.
EmployeeOf	Dave is an employee of Acme Corp (DID-identified).
ParticipantOf	Eve is a participant of the 09:00 meeting (a VC-based meeting request).
VoterFor	Frank has cast a vote for Candidate 1 by contributing his proof to that VTC.

Table 1

All of these reduce to the same credential structure: a VC whose credentialSubject.id identifies the group or decision entity (the "circle"), and whose proof array contains proofs from the Notary and each member who has accepted membership. PHCs and VRCs are degenerate cases of this pattern with N=1 and N=2 respectively.

6. VTC Data Model

6.1. Overview

A VTC is a valid W3C Verifiable Credential [W3C.VC-DATA-MODEL] with the following structural characteristics:

* The issuer property identifies the Notary (N).

* The credentialSubject (or confidentialSubject) object includes from, to, and optionally metadata properties that identify the Initiator, Responders, and relationship metadata respectively.

* The credentialSubject.id identifies the relationship or group itself, expressed as a DID.

* The proof property is an array (Proof Set), containing one proof per signer, ordered as: Notary first, then Initiator, then Responders.

6.2. Minimal Pairwise VTC (N=2, Alice and Bob)

The following non-normative example illustrates a bilateral VTC between Alice (Initiator) and Bob (Responder), notarized by a Notary entity:

```
{
  "@context": [
    "https://www.w3.org/ns/credentials/v2",
    "https://w3id.org/vtc/v1"
  ],
  "id": "did:envelope:1234",
  "type": [
    "VerifiableCredential",
    "VerifiableTrustCircle"
  ],
  "issuer": "did:example:notaryabcd",
  "validFrom": "2026-01-01T00:00:00Z",
  "credentialSubject": {
    "id": "did:vrc:2468",
    "from": "did:example:alice",
    "to": ["did:example:bob"],
    "metadata": {
      "label":
        "Alice-Bob Bilateral Trust Circle"
    }
  },
  "proof": [
    {
      "id": "did:example:notaryabcd",
      "type": "DataIntegrityProof",
      "...": "<notary-proof>"
    },
    {
      "id": "did:example:alice",
      "type": "DataIntegrityProof",
      "...": "<alice-proof>"
    },
    {
      "id": "did:example:bob",
      "type": "DataIntegrityProof",
      "...": "<bob-proof>"
    }
  ]
}
```

6.3. Multi-Party VTC (N=3+)

For groups with more than two members, the "to" array is extended to include all Responders, and the proof array gains one additional entry per additional Responder:

```
{
  "id": "did:envelope:5678",
  "type": [
    "VerifiableCredential",
    "VerifiableTrustCircle"
  ],
  "issuer": "did:example:notaryabcd",
  "credentialSubject": {
    "id": "did:vrc:9999",
    "from": "did:example:alice",
    "to": [
      "did:example:bob",
      "did:example:carol",
      "...",
      "did:example:zelda"
    ],
    "metadata": {
      "label": "Working Group Trust Circle",
      "policy": "did:policy:majority"
    }
  },
  "proof": [
    { "id": "did:example:notaryabcd",
      "...": "<notary-proof>" },
    { "id": "did:example:alice",
      "...": "<alice-proof>" },
    { "id": "did:example:bob",
      "...": "<bob-proof>" },
    { "id": "did:example:carol",
      "...": "<carol-proof>" },
    { "id": "did:example:zelda",
      "...": "<zelda-proof>" }
  ]
}
```

6.4. Self-Credential VTC (N=1, PHC Equivalent)

When "to" contains only the Initiator’s own DID, or when "from" and credentialSubject.id identify the same entity, the VTC degenerates to a Personhood Credential (PHC):

```
{
  "credentialSubject": {
    "id": "did:phc:alice-self",
    "from": "did:example:alice",
    "to": ["did:example:alice"],
    "metadata": {
      "label": "Alice Self-Attestation"
    }
  },
  "proof": [
    { "id": "did:example:notaryabcd",
      "...": "<notary-proof>" },
    { "id": "did:example:alice",
      "...": "<alice-proof>" }
  ]
}
```

6.5. Voting VTC

For a voting scenario, one VTC is created per candidate. Voters cast their vote by contributing their individual proof to the VTC of the candidate they support. The vote count is the number of valid member proofs in the Proof Set.

```
{
  "credentialSubject": {
    "id":
      "did:sovronia:election2026:district103:cand1",
    "from": "did:example:electionofficial",
    "to": [],
    "metadata": {
      "label":
        "Candidate 1 - District 103 - 2026"
    }
  },
  "proof": [
    { "id": "did:example:electionofficial",
      "...": "<official-proof>" },
    { "id": "did:example:voter001",
      "...": "<vote-proof>" },
    { "id": "did:example:voter002",
      "...": "<vote-proof>" }
  ]
}
```

> | NOTE: The "to" array MAY be populated in advance with eligible
> | voter DIDs, or it MAY be left empty and populated as votes are
> | cast, depending on the election policy and privacy
> | requirements.

6.6. Properties Reference

The following table defines the normative properties of a VTC credential:

Property	Req.	Description
id	REQUIRED	DID identifying the VTC credential itself. SHOULD use did:envelope or equivalent.
type	REQUIRED	MUST include "VerifiableCredential" and "VerifiableTrustCircle".
issuer	REQUIRED	DID of the Notary (N). The Notary MUST be trusted by all members.
credentialSubject.id	REQUIRED	DID identifying the relationship or group itself.
credentialSubject.from	REQUIRED	DID of the Initiator (A).
credentialSubject.to	REQUIRED	Array of Responder DIDs. MAY be empty for open voting VTCs.
credentialSubject.metadata	OPTIONAL	Arbitrary structured metadata (label, policy, expiry, etc.).
proof	REQUIRED	Array of proof objects (Proof Set). First proof MUST be from the Notary. Subsequent proofs are from the Initiator then Responders in any order.
proof[].id	REQUIRED	DID of the signer contributing this proof entry.

Table 2

7. VTC Proof Set Lifecycle

The VTC Proof Set lifecycle consists of the following phases. At each phase t, the VTC applies to the Notary and the first t members who have contributed their proof.

7.1. Phase 0 - Null VTC

The credential shell is created by the Notary with an empty or pre-populated "to" array. The Notary contributes the initial proof. No member relationships are yet verified. t = 0.

7.2. Phase 1..t - Progressive Endorsement

Each Responder, in any order, reviews the credential and - if they consent to membership - adds their individual proof to the existing Proof Set using the "add-proof-set-chain" algorithm defined in [W3C.VC-DATA-INTEGRITY]. The VTC becomes valid for those t members who have signed. Non-signing members are not yet bound.

7.3. Phase N - Complete VTC

All Responders listed in the "to" array have contributed their proofs. The VTC is fully executed and represents a complete, verifiable, multi-party trust relationship.

> | NOTE: Partial VTCs (0 < t < N) are valid credentials
> | representing the subset of relationships established so far.
> | Verifiers MUST check which proofs are present before asserting
> | full circle membership.

7.4. Adding a Proof

To add a proof to an existing secured VTC, implementors MUST follow the algorithm specified in W3C VC Data Integrity [W3C.VC-DATA-INTEGRITY], Section "add-proof-set-chain". The proof is appended to the existing proof array without modifying prior proofs.

7.5. Proof Ordering

Proof Sets are unordered by definition. However, this specification RECOMMENDED to follow the conventional ordering for readability and auditability: (1) Notary proof, (2) Initiator proof, (3) Responder proofs in the same order as the "to" array.

8. Roles and Participants

8.1. Notary (N) - Issuer

The Notary is the credential issuer. It MUST be trusted by both the Initiator and all Responders. The Notary is responsible for creating the credential shell, pre-populating the "to" array (or defining the voting policy), and contributing the first proof. In some use cases, the Notary MAY be the same entity as the Initiator or a Responder, provided that entity plays each role distinctly and the resulting credential satisfies all REQUIRED properties.

8.2. Initiator (A) - From

The Initiator proposes the trust circle. The Initiator's DID appears in credentialSubject.from. The Initiator contributes a proof to the Proof Set to signify their acceptance of the relationship.

8.3. Responders (B ... Z) - To

Each Responder is identified in the credentialSubject.to array. A Responder accepts membership by contributing their individual proof. A Responder who does not contribute a proof is proposed but not yet a verified member.

> | RULE: The cardinality t of verified members at any time equals
> | the number of valid member proofs (excluding the Notary proof)
> | present in the Proof Set.

9. Use Cases

9.1. Bilateral Trust Relationship (VRC Equivalent)

Alice and Bob wish to establish a verifiable bilateral trust relationship. A mutually trusted Notary issues a VTC with from = Alice and to = [Bob]. Both Alice and Bob contribute proofs. The result is a two-party VTC equivalent to a classic VRC.

9.2. Personhood Credential (PHC Equivalent)

Alice wishes to create a self-signed personhood credential. A Notary issues a VTC with from = Alice and to = [Alice]. Alice contributes her proof. The result is a one-party VTC equivalent to a PHC [PHC-PAPER].

9.3. Web 7.0 Foundation Governance Council

A task force of N participants is formed. A Notary (the working group chair or a community DID) issues a VTC with from = chair and to = [member1, ..., memberN]. Members join by contributing their proofs. The VTC provides a cryptographically verifiable roster.

9.4. VC-Based Meeting Request

An organizer issues a VTC with credentialSubject.id equal to the meeting DID, from = organizer, and to = [attendee1, ..., attendeeN]. Attendees RSVP by contributing their proofs. Attendance at the meeting is verifiable from the Proof Set.

9.5. Voting-Based Decision Making

One VTC per candidate is issued by an election official (Notary). Eligible voters cast their vote by contributing their individual proof to the VTC of their chosen candidate. Vote tallying is performed by counting the number of valid member proofs in each candidate’s VTC. This supports maximum flexibility in vote-counting policies (simple majority, ranked-choice, threshold).

9.6. Verifiable Decentralized Registry

VC-based voting can be applied to implement a Verifiable Data Registry (VDR). Append operations to a distributed registry are authorized through a VTC whose members are the registry trustees.

9.7. Digital Society / Digital Nation State

A digital society (e.g., a digital community or nation state) is defined by a VTC whose members are the citizens. Governance operations - electing trustees, passing resolutions - are performed through subsidiary voting VTCs.

10. Conformance

A conforming VTC implementation:

* MUST produce VTC credentials that are valid W3C Verifiable Credentials conforming to [W3C.VC-DATA-MODEL].

* MUST use a proof array (Proof Set) as defined in [W3C.VC-DATA-INTEGRITY].

* MUST include the issuer property identifying the Notary.

* MUST include credentialSubject.id, credentialSubject.from, and credentialSubject.to.

* MUST use the "add-proof-set-chain" algorithm from [W3C.VC-DATA-INTEGRITY] when adding proofs incrementally.

* SHOULD include "VerifiableTrustCircle" in the type array.

* SHOULD implement selective disclosure mechanisms for credentialSubject properties.

* MAY extend the credentialSubject.metadata property with domain-specific claims.

11. Relationship to Other Specifications

11.1. W3C VC Data Model 2.0

VTCs are valid W3C Verifiable Credentials. All normative requirements of [W3C.VC-DATA-MODEL] apply. VTCs use the issuer and credentialSubject properties as defined therein.

11.2. W3C VC Data Integrity

VTCs rely on the Proof Set mechanism defined in [W3C.VC-DATA-INTEGRITY], specifically the "add-proof-set-chain" algorithm for incremental proof contributions.

11.3. ToIP DTGWG Design Principles

This specification is consistent with the ToIP DTGWG Design Principles [DTGWG-DESIGN], the DTG-ZKP Requirements [DTGWG-ZKP], and the VRC Design Proposals [DTGWG-VTC-13].

11.4. SSC 7.0 Metamodel

VTCs integrate with the Self-Sovereign Control 7.0 Metamodel [SSC-7]. VTCs may be anchored at the Beneficial Controller, Intermediate Controller, or Technical Controller layer.

11.5. Trust Spanning Protocol (TSP)

VTCs are compatible with the Trust Spanning Protocol [TSP] as a credential format for expressing channel-level membership and authorization relationships.

12. Privacy Considerations

12.1. Selective Disclosure

Implementations are strongly RECOMMENDED to use confidentialSubject semantics and selective disclosure proof mechanisms (e.g., BBS+ signatures) to allow individual members to prove their membership in a VTC without revealing the full membership list or metadata.

12.2. ZKP Integration

The Proof Set mechanism is compatible with zero-knowledge proof (ZKP) contributions. A member MAY contribute a ZKP as their proof entry, revealing only that they meet the membership criteria without revealing their DID. Implementations SHOULD define a profile for ZKP-based proof entries.

12.3. Privacy Budget and Reconstruction Ceiling

When multiple agents controlled by one First Person contribute to a shared VTC, care must be taken to ensure that the combined disclosure across proof entries does not exceed the privacy budget of the First Person. The reconstruction ceiling - the probability that an observer can reconstruct the First Person’s identity from the combined proof data - MUST be maintained below the threshold defined by the applicable trust framework.

12.4. Notary Trust

The Notary (issuer) occupies a privileged position: it issues the credential shell and contributes the first proof. Verifiers MUST independently verify that the Notary is trusted by all relevant parties. The Notary SHOULD be a well-known, community-governed DID with transparent governance.

13. Security Considerations

13.1. Voting Integrity

For voting VTCs, the following security properties MUST be considered:

* *Eligibility:* Only eligible voters can contribute proofs.

* *Anonymity:* Voter DIDs SHOULD be anonymized or pseudonymized.

* *Non-repudiation:* Each proof is cryptographically bound to the voter’s key.

* *Single-vote enforcement:* The "to" array or the Notary's policy SHOULD prevent duplicate proof contributions from the same voter DID.

13.2. Proof Integrity

Verifiers MUST validate each proof entry in a Proof Set independently against the secured document using the algorithm specified in [W3C.VC-DATA-INTEGRITY]. The presence of a valid Notary proof does not substitute for validating member proofs, and vice versa.

13.3. Partial VTC Assertions

Verifiers MUST NOT assert full circle membership based solely on the presence of the Notary proof or a subset of member proofs. Assertions about membership MUST be scoped to the verified set of proof contributors at the time of verification.

14. IANA Considerations

This document has no IANA actions. The "VerifiableTrustCircle" credential type identifier is defined within the W3C VC ecosystem and SHOULD be registered in the W3C VC Extensions Registry upon advancement of this specification.

15. References

15.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[W3C.DID-CORE] Sporny, M., Guy, A., Sabadello, M., and D. Reed, "Decentralized Identifiers (DIDs) v1.0", W3C Recommendation, July 2022, <https://www.w3.org/TR/did-core/>.

[W3C.VC-DATA-INTEGRITY] Sporny, M. and D. Longley, "Verifiable Credential Data Integrity 1.0", W3C Recommendation, 2024, <https://www.w3.org/TR/vc-data-integrity/>.

[W3C.VC-DATA-MODEL]
Sporny, M., Longley, D., and D. Chadwick, "Verifiable Credentials Data Model v2.0", W3C Recommendation, 2024, <https://www.w3.org/TR/vc-data-model-2.0/>.

15.2. Informative References

[DISCUSSION-8]
Herman, M., "Web 7.0 Verifiable Trust Circles (VTCs)", GitHub Discussion #8, trustoverip/dtgwg-cred-tf, 2025, <https://github.com/trustoverip/dtgwg-cred-tf/discussions/8>.

[DTGWG-DESIGN]
Trust over IP Foundation, "DTGWG Design Principles", GitHub Discussion #11, trustoverip/dtgwg-cred-tf, 2025, <https://github.com/trustoverip/dtgwg-cred-tf/discussions/11>.

[DTGWG-VTC-13]
Trust over IP Foundation, "VRC Design Proposals", GitHub Discussion #13, trustoverip/dtgwg-cred-tf, 2025, <https://github.com/trustoverip/dtgwg-cred-tf/discussions/13>.

[DTGWG-ZKP]
Trust over IP Foundation, "DTG-ZKP Requirements", GitHub Discussion #12, trustoverip/dtgwg-cred-tf, 2025, <https://github.com/trustoverip/dtgwg-cred-tf/discussions/12>.

[PHC-PAPER]
Crites, B., "Personhood Credentials", arXiv preprint 2408.07892, 2024, <https://arxiv.org/pdf/2408.07892>.

[SSC-7] Herman, M., "Self-Sovereign Control (SSC) 7.0 Metamodel", Hyperonomy, December 2025, <https://hyperonomy.com/2025/12/10/ self-sovereign-control-ssc-7-0-metamodel/>.

[TSP] Trust over IP Foundation, "Trust Spanning Protocol", 2025, <https://trustoverip.org/>.

[WEB70-VTC]
Herman, M., "SDO: Verifiable Trust Circles (VTCs) using VC Proof Sets (Web 7.0)", Licensed under Creative Commons Attribution-ShareAlike 4.0 International Public License, March 2026, <https://hyperonomy.com/2026/03/26/ sdo-verifiable-trust-circles-vtcs-using-vc-proof-sets-web-7-0/>.

Appendix A. VTC Cardinality and Credential Type Mapping

The following table summarizes the mapping between VTC cardinality (size of the "to" array) and the equivalent prior credential construct:

N (members)	VTC Type	Prior Equivalent	Proof Count
0	Null VTC	None	1 (Notary only)
1	Self-Credential	PHC	2 (Notary + A)
2	Bilateral	VRC	3 (Notary + A + B)
N > 2	Multi-Party	None (new)	N+2 (Notary + A + B...Z)
Open	Voting VTC	None (new)	1 + votes cast

Table 3

Acknowledgements

This specification was derived from community discussion contributions by Michael Herman (mwherman2000), talltree, adamstallard, mitchuski, peacekeeper, GraceRachmany, and other participants of the Trust over IP Foundation DTGWG Credentials Task Force. The editors gratefully acknowledge all contributors to GitHub Discussion #8 [DISCUSSION-8].

Author’s Address

Michael Herman
Web 7.0 Foundation
Bindloss Alberta
Canada
Email: mwherman@gmail.com
URI: https://hyperonomy.com/about/