Network Working Group
Internet-Draft
Intended status: Informational
Expires: 15 August 2026

D. Condrey, Ed.
WritersLogic Inc

11 February 2026

Proof of Process (PoP) CDDL Schema
draft-condrey-rats-pop-schema-01

Abstract

This document provides the normative Concise Data Definition Language (CDDL) schema for the Proof of Process (PoP) protocol. The schema defines the CBOR-encoded wire format for PoP evidence packets, semantic editing event transcripts, time anchors, signed tool receipts, compact evidence references, and verifier-produced attestation results.

The schema is published separately to enable independent tooling, validation, and schema versioning decoupled from narrative specification text.

Status of This Memo

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

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."

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document contains the normative CDDL schema for the Proof of Process (PoP) protocol defined in [I-D.condrey-rats-pop-protocol].

The schema defines the CBOR-encoded structures used to represent PoP Evidence Packets (.pop), transcripts of semantic editing events, time anchors, signed tool receipts, compact evidence references, and Attestation Results (.war). Implementations that produce or verify PoP artifacts MUST conform to this schema.

The schema is provided as a separate document to facilitate independent tooling and validation and to support schema versioning separate from narrative specification updates.

1.1. Requirements Language

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] and [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Schema Information

This schema is specified using CDDL as defined in [RFC8610]. CDDL provides a notation for expressing CBOR [RFC8949] data structures with precise type constraints and extensibility support.

Schema Version: 1.5.1

Compatibility: Breaking changes increment the major version number. Minor version changes add optional fields or new enumeration values. Patch version changes are documentation or clarification only.

Deterministic Encoding: For any structure that participates in hashing or signing, implementations MUST use deterministic CBOR encoding as defined in RFC 8949, Section 4.2. Verifiers MUST reject signatures if the reconstructed payload bytes do not match the deterministic encoding of the signed structure defined in this document.

Label Space: Integer keys 1..99 are reserved for this schema. Integer keys 100..32767 are reserved for future IETF extensions. String keys are for vendor/private extensions and MUST NOT be required for interoperability. Verifiers MAY ignore unknown keys under policy.

Semantic Tags: Evidence Packets (.pop) use CBOR tag 1347571280 (Proof of Process Packet, "PPPP"). Compact Evidence References use CBOR tag 1347571281. Attestation Results (.war) use CBOR tag 1463894560 (Writers Authenticity Report, "WAR"). These tag values are registered in the IANA "CBOR Tags" registry [IANA.cbor-tags].

3. Signing and Hashing Surfaces

This section defines the byte-level payloads that MUST be used for hashing and signing in PoP artifacts. All payloads in this section MUST be encoded using deterministic CBOR encoding as defined in RFC 8949, Section 4.2.

Evidence Packet Signature Payload: The COSE_Sign1 payload in the evidence-packet structure MUST be the deterministic CBOR encoding of the evidence-packet fields excluding the signature itself. The signature field MUST NOT be included in the signed bytes.

Attestation Result Signature Payload: The COSE_Sign1 payload in an attestation-result (field 7) MUST be the deterministic CBOR encoding of fields 1-6 plus any optional fields 8-11. The signature field itself MUST NOT be included in the signed bytes.

Detached Payload: If a COSE_Sign1 structure uses a detached payload (i.e., the COSE payload is CBOR null), the verifier MUST reconstruct the payload bytes exactly as specified above and validate that the signature covers those bytes.

4. Complete CDDL Schema

The following CDDL defines the complete wire format for PoP artifacts and associated structures. This schema is normative and implementations MUST conform to it.

The schema constrains COSE_Sign1 structures to the array shape specified in [RFC9052] and defines unambiguous signing payloads (see Section 3). This schema does not reproduce the COSE algorithm registries; COSE processing rules remain as specified in [RFC9052].

COSE Header Requirements: The protected header of each COSE_Sign1 structure MUST decode to a CBOR map that includes the alg header parameter. Either the protected or unprotected header map MUST provide key identification sufficient for verification (for example, kid or an X.509 chain) per deployment policy.

```
; ============================================================
; Proof of Process (PoP) — Normative CDDL Schema
; Schema Version: 1.5.1
; ============================================================
;
; Notes:
; - This schema constrains COSE_Sign1 to its array shape per RFC 9052.
; - COSE header registries and algorithm processing remain in RFC 9052.
; - Deterministic CBOR encoding (RFC 8949 §4.2) is REQUIRED for any
;   structure that participates in hashing or signing.
;
; ============================================================
; Label Space / Extensibility
; ============================================================
;
; Integer labels 1..99: reserved by this schema (normative).
; Integer labels 100..32767: reserved for future IETF extensions.
; Text labels: vendor/private extensions; MUST NOT be required for interop.
;
; ============================================================
; Top-Level Objects
; ============================================================

; IANA-registered CBOR tags:
; - 1347571280: Proof of Process Packet (PPPP)
; - 1347571281: Compact Evidence Reference
; - 1463894560: Writers Authenticity Report (WAR)

tagged-evidence-packet    = #6.1347571280(evidence-packet)
tagged-evidence-reference = #6.1347571281(evidence-reference)
tagged-attestation-result = #6.1463894560(attestation-result)

; ============================================================
; Core Scalar Types
; ============================================================

; UUID as 16-byte string (RFC 9562)
uuid = bstr .size 16

; Timestamp as CBOR epoch-based date/time (tag 1)
pop-timestamp = #6.1(number)
```

```
; Hash value (SHA-256, SHA-384, or SHA-512)
hash-value = bstr .size 32 / bstr .size 48 / bstr .size 64

; Fixed-point type definitions for compact encoding
confidence-millibits = uint .le 1000   ; 0-1000 representing 0.000-1.000
ratio-millibits = uint .le 1000        ; generic 0.0-1.0 ratio
entropy-decibits = uint .le 640        ; 0-640 representing 0.0-64.0 bits

; ============================================================
; Evidence Packet (.pop)
; ============================================================
;
; The primary Evidence artifact produced by the Attester.
; Contains all cryptographic proofs and behavioral evidence.

evidence-packet = {
  1 => uint,                        ; version (1)
  2 => vdf-structure,               ; VDF proof
  3 => jitter-seal-structure,       ; Jitter Seal (mandatory in v1.1+)
  4 => content-hash-tree,           ; Merkle tree for segments
  5 => correlation-proof,           ; Spearman Correlation
  6 => error-topology,              ; Fractal Error Pattern
  7 => hardware-attestation,        ; Hardware Assurance Binding
  8 => process-metrics,             ; Raw Process Measurements
  * tstr => any,                    ; extensions
}

vdf-structure = {
  1 => bstr,                        ; input: H(DST_CHAIN || content || jitter_seal)
  2 => bstr,                        ; output
  3 => uint,                        ; iterations
  4 => [* uint],                    ; rdtsc_checkpoints (continuous calibration)
  5 => bstr,                        ; entropic_pulse: HMAC(SK, T ^ E)
  * tstr => any,
}

jitter-seal-structure = {
  1 => tstr,                        ; lang (e.g., "en-US")
  2 => bstr,                        ; bucket_commitment (ZK-Private)
  3 => uint,                        ; entropy_millibits
  5 => int .within -100..100,       ; pink_noise_slope_decibits (-10.0..10.0)
  * tstr => any,
}

content-hash-tree = {
  1 => bstr,                        ; root
  2 => uint .ge 20,                 ; segment_count
  * tstr => any,
```

```
}

correlation-proof = {
  1 => int .within -1000..1000,     ; rho (scaled: -1000..1000 = -1.0..1.0)
  2 => uint,                        ; threshold (e.g., 700 = 0.7)
  * tstr => any,
}

error-topology = {
  1 => bstr,                        ; fractal-signature commitment
  2 => ratio-millibits,             ; pattern-score
  ? 3 => bstr,                      ; stark-proof (optional ZK proof)
  * tstr => any,
}

hardware-attestation = {
  1 => tstr,                        ; attestation-type ("tpm2.0" / "secure-enclave")
  2 => bstr,                        ; attestation-data
  ? 3 => [* bstr],                  ; certificate-chain
  * tstr => any,
}

process-metrics = {
  1 => ratio-millibits,             ; linearity-score
  2 => ratio-millibits,             ; structural-edit-ratio
  3 => int,                         ; hesitation-phase-offset (signed millibits)
  4 => ratio-millibits,             ; revision-clustering
  5 => ratio-millibits,             ; fatigue-slope
  6 => uint,                        ; checkpoint-count
  7 => uint,                        ; total-duration-ms
  ? 8 => [+ ratio-millibits],       ; per-checkpoint-conformity-scores
  * tstr => any,
}

; ============================================================
; Compact Evidence Reference (Tagged)
; ============================================================

evidence-reference = {
  1 => uint,                        ; version
  2 => uuid,                        ; packet-id (matches evidence-packet)
  3 => hash-value,                  ; content-hash
  ? 4 => pop-timestamp,             ; created timestamp
  ? 5 => forensic-assessment,       ; verdict (if available)
  ? 6 => confidence-millibits,      ; confidence (0-1000)
  * tstr => any,
}
```

```
; ============================================================
; Attestation Result (.war)
; ============================================================
;
; The Verifier's assessment of an Evidence packet.
; Implements a witnessd-specific profile of EAR.

attestation-result = {
  1 => uint,                        ; version
  2 => uuid,                        ; reference-packet-id
  3 => pop-timestamp,               ; verified-at
  4 => forensic-assessment,         ; verdict
  5 => confidence-millibits,        ; confidence (0-1000 = 0.0-1.0)
  6 => [+ result-claim],            ; verified-claims
  7 => cose-signature,              ; verifier-signature
  ? 8 => tstr,                      ; verifier-identity
  ? 9 => verifier-metadata,         ; additional info
  ? 10 => [+ tstr],                 ; caveats
  ? 11 => source-consistency-analysis, ; Verifier's interpretation
  * tstr => any,
}

; Forensic assessment enumeration
forensic-assessment = &(
  not-assessed: 0,
  manual-composition-consistent: 1,
  manual-composition-likely: 2,
  inconclusive: 3,
  automated-assisted-likely: 4,
  automated-insertion-consistent: 5,
)

result-claim = {
  1 => uint,                        ; claim-type
  2 => bool,                        ; verified
  ? 3 => tstr,                      ; detail
  ? 4 => confidence-millibits,      ; claim-confidence
  * tstr => any,
}

verifier-metadata = {
  ? 1 => tstr,                      ; verifier-version (software version)
  ? 2 => tstr,                      ; verifier-uri (service endpoint)
  ? 3 => [+ bstr],                  ; verifier-cert-chain (X.509 DER)
  ? 4 => tstr,                      ; policy-id (appraisal policy used)
  * tstr => any,
}
```

```
source-consistency-analysis = {
  1 => tstr,                          ; detected-pattern
  2 => ratio-millibits,               ; aggregate-consistency (0-1000)
  ? 3 => [+ uint],                    ; deviation-checkpoint-indices
  ? 4 => tstr,                        ; verifier-policy-id
  * tstr => any,
}

; ============================================================
; COSE Signatures
; ============================================================

cose-signature = [
  protected : bstr,
  unprotected : { * int => any, * tstr => any },
  payload : bstr / nil,
  signature : bstr
]
```

Figure 1: Proof of Process (PoP) CDDL Schema (Version 1.5.1)

5. Security Considerations

This document defines a data format schema for PoP artifacts. Security considerations for the PoP protocol, including verification requirements and threat models, are specified in the companion protocol document [I-D.condrey-rats-pop-protocol].

Disclosure Risk: Transcripts may reveal sensitive intermediate content, including deleted text and draft iterations. Deployments SHOULD define disclosure policies that minimize unnecessary exposure, and MAY use selective disclosure proofs where supported by the companion protocol specification.

Auxiliary Data: Implementations MUST treat auxiliary fields and vendor extensions as potentially sensitive. Verifiers SHOULD ignore unknown extensions unless explicitly allowed by policy.

Receipt Trust: A valid receipt signature only attests that a tool produced an output commitment; it does not itself establish that the tool behaved honestly. Deployments SHOULD define trust and revocation policies for tool keys and SHOULD surface receipt flags to users and verifiers.

Replay and Rebinding: Implementations SHOULD bind receipts and anchors to a session (directly or via commitments) to prevent replay in unrelated documents or sessions, as specified by the companion protocol document.

6. IANA Considerations

This document has no IANA actions. The PoP-related CBOR tags referenced by this schema are already registered in the IANA "CBOR Tags" registry [IANA.cbor-tags].

7. 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/info/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/info/rfc8174>.

[RFC8610] Birkholz, H., Vigano, C., and C. Bormann, "Concise Data Definition Language (CDDL)", RFC 8610, DOI 10.17487/RFC8610, June 2019, <https://www.rfc-editor.org/info/rfc8610>.

[RFC8949] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", STD 94, RFC 8949, DOI 10.17487/RFC8949, December 2020, <https://www.rfc-editor.org/info/rfc8949>.

[RFC9052] Schaad, J., "CBOR Object Signing and Encryption (COSE): Structures and Process", STD 96, RFC 9052, DOI 10.17487/RFC9052, August 2022, <https://www.rfc-editor.org/info/rfc9052>.

8. Informative References

[I-D.condrey-rats-pop-protocol] Condrey, D., "Proof of Process (PoP): A Verifiable Process Transcript Format", Work in Progress, Internet-Draft, draft-condrey-rats-pop-protocol-00, <https://datatracker.ietf.org/doc/html/draft-condrey-rats-pop-protocol-00>.

[IANA.cbor-tags] IANA, "Concise Binary Object Representation (CBOR) Tags", <https://www.iana.org/assignments/cbor-tags/cbor-tags.xhtml>.

Acknowledgments

The author thanks the RATS community and early implementers for review feedback on schema determinism, COSE signing surfaces, and CBOR tag interoperability.

Author’s Address

David Condrey (editor)
WritersLogic Inc
United States
Email: david@writerslogic.com