Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: 15 November 2026

J. van de Meent
Humotica

14 May 2026

Continuity Envelope Protocol
draft-vandemeent-continuity-envelope-00

Abstract

This document defines the Continuity Envelope Protocol (CEP), a transport-agnostic protocol model for identity-bound messaging systems where delivery alone is not sufficient.

CEP separates a visible envelope surface from sealed internal object truth and defines how receivers determine whether a delivered object may safely and legitimately continue. The protocol distinguishes between control-plane notification, data-plane object carriage, and decision-plane continuation handling.

CEP is the umbrella model for companion drafts such as TIBET TAT, which specifies generic sealed handoff and transfer flow, and IDDrop, which profiles that handoff model for identity transfer.

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 15 November 2026.

Copyright Notice

Table of Contents

1. Introduction

Classical messaging protocols are primarily optimized for delivery. For cross-host agents, sealed artifact exchange, and resumable automation, delivery is necessary but insufficient.

CEP defines how visible surfaces, sealed truth, and continuation decisions relate across transport fabrics.

1.1. Scope

This document defines the separation between control-plane notification, data-plane carriage, and decision-plane continuation handling; the role of the Continuity Envelope as a visible routing surface; the distinction between envelope reference, presence, verification, and continuation approval; layered time semantics and drift handling; and a common vocabulary for causality and continuation outcomes.

This document does not define a single mandatory wire format, a single mandatory transport binding, a specific cryptographic primitive suite, or a universal trust-scoring algorithm.

Instead, this document provides the architectural frame into which companion drafts fit: TIBET TAT defines the generic touch-and-transfer and relay handoff protocol, while IDDrop defines an identity-transfer profile over that handoff model.

2. Terminology

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.

Continuity Messaging A messaging model in which the exchanged unit is a continuity step rather than delivery alone.

Continuity Envelope The visible routing and operator-facing surface associated with an object handoff.

Envelope Reference A control-plane pointer to an object, without implying byte possession or verification.

Sealed Truth The normative internal object state carried through manifests, signatures, mirrored fields, and causal claims.

3. Architectural Model

3.1. Control Plane

The control plane carries notification and lightweight exchange metadata. Control-plane data MUST NOT be interpreted as proof that the receiver possesses or has verified the underlying object bytes unless such proof is explicitly encoded.

3.2. Data Plane

The data plane carries the continuity-bearing object itself. Examples include local file drop, relay-carried sealed object, peer-to-peer artifact transfer, and object fetched by reference.

3.3. Decision Plane

The decision plane evaluates whether continuation is legitimate and safe. The decision plane MUST treat sealed truth as authoritative where conflict exists.

4. Problem Statement

In mailbox-centered systems, a successful delivery event is frequently treated as the end of the transport problem. In continuity-bearing systems, delivery is only one stage.

Several failure classes appear after arrival, including divergence between transport metadata and sealed truth, visible names that overclaim object identity, control-plane references without local bytes, disagreement between time fragments across layers, stale or replayed continuity, and missing or invalid causal parentage.

5. Continuity Envelope Semantics

The Continuity Envelope provides a visible routing surface for operator readability, sorting, grouping, routing hints, freshness hints, priority hints, and continuity diagnostics.

The envelope surface is not the final source of truth. CEP therefore adopts the principle: name is hint, content is truth.

5.1. Surface Shape

<time-fragment>.<context>.<profile>.<priority>[.<extension>]

5.2. ABNF

The following ABNF, using the notation of [RFC5234], defines a conservative baseline grammar for version 1 envelope surfaces:

```
surface-name    = time-fragment "." context "." profile "." priority
                  [ "." extension ]

time-fragment   = compact-time / dashed-time / date-only
compact-time    = 8DIGIT "t" 6DIGIT "z"
dashed-time     = 4DIGIT "-" 2DIGIT "-" 2DIGIT "t"
                  2DIGIT "-" 2DIGIT "-" 2DIGIT "z"
date-only       = 4DIGIT "-" 2DIGIT "-" 2DIGIT

context         = 1*48(segment-char)
profile         = 1*48(segment-char)
priority        = 1*32(segment-char)
extension       = 1*48(segment-char)

segment-char    = LCALPHA / DIGIT / "-"
LCALPHA         = %x61-7A
DIGIT           = %x30-39
```

5.3. Character and Length Policy

For portability and operator safety, implementations SHOULD prefer lowercase ASCII, dot-delimited segments, and per-segment characters from [a-z0-9-]. Implementations SHOULD reject or normalize spaces, shell metacharacters, empty segments, and path separators.

Implementations SHOULD also apply conservative operational limits. Representative values are a total surface length of 180 characters or less, hard rejection above 200 characters, a preferred segment length of 32 characters or less, and a segment upper bound of 48 characters.

6. Sealed Object Truth

The sealed object carries the normative internal state. This MAY include manifest data, mirrored surface fields, integrity digests, signature data, sealed time fields, causal parentage fields, and schema or version identifiers.

If the envelope surface and sealed truth disagree, the sealed truth MUST be treated as authoritative for object identity, while the mismatch SHOULD be surfaced as a continuity signal.

6.1. Schema Identity

Control-plane metadata and sealed manifest schema identifiers SHOULD be consistent for the same object handoff. An implementation receiving a mismatch MAY triage the object, downgrade trust, request resynchronization, or continue under local policy if the mismatch is understood and bounded.

7. Verification States

CEP requires verification to be layered. These states MUST NOT be collapsed into a single generic "verified" signal when their meanings differ.

State	Meaning
referenced	Control plane carries a pointer only
present	Receiver possesses bytes locally
digest-matched	Observed digest matches expected digest
container-verified	Container structure and integrity verified
manifest-verified	Manifest semantics and mirrored fields validated
continuation-approved	Object may safely continue under policy

Table 1: Initial Verification States

8. Causality and Continuation

CEP treats causal legitimacy as distinct from delivery. An object MAY be successfully delivered but still fail continuation checks if causal parentage is missing, points to an unknown or invalid predecessor, policy forbids progression from the observed lineage, or drift exceeds acceptable thresholds.

Outcome	Meaning
continue	Progression is allowed
triage	Further inspection is required
fork	Progression continues along a branch
quarantine	Object is isolated pending policy review
resync	Fresh alignment is required before continuation

Table 2: Recommended Continuation Outcomes

9. Time and Drift

CEP treats time as layered rather than singular. Envelope time SHOULD be treated as a grouping hint, freshness hint, operator readability signal, and early drift indicator. It MUST NOT be treated as the sole authority for continuity legitimacy.

Sealed internal time fields carry the object’s normative time claims. Causal order is authoritative for continuation legitimacy.

10. Transport Bindings

CEP is transport-agnostic. Possible bindings include local file drop, peer-to-peer delivery, relay-based object transfer, control-plane notification over message protocols, object fetch by reference, and future bridge transports.

10.1. Companion Transfer and Application Drafts

CEP is intended to be read together with companion drafts when a concrete transfer or application profile is needed.

In particular, TIBET TAT defines how a consent-bound sealed handoff is executed across proximity, relay, or local network paths, and IDDrop defines how identity claims are offered, requested, validated, and materialized over that handoff substrate.

CEP therefore acts as the umbrella model: CEP says how continuity-bearing messaging is structured, TAT says how a sealed transfer happens, and IDDrop says how identity transfer happens over TAT.

11. Examples

11.1. Example Control-Plane Notification

```
{
  "from_agent": "jasper",
  "to_agent": "codex",
  "poll_type": "PUSH",
  "metadata": {
    "envelope_name": "20260514t064148z.message.jasper.normal.tza",
    "envelope_ref": "/var/lib/tibet/inbox/20260514t064148z.message.jasper.normal.tza",
    "schema": "aint-send-v2",
    "verification_state": "referenced"
  }
}
```

11.2. Example Sealed Manifest Fields

```
{
  "schema": "aint-send-v2",
  "surface_time_fragment": "20260514t064148z",
  "surface_context": "message",
  "surface_profile": "jasper",
  "surface_priority": "normal",
  "sealed_created_at": "2026-05-14T06:41:48.207135+00:00",
  "content_digest": "sha256:b5ca11489ffadc4438a3815a0d7ea3bc88836dbc2f5951b1e9d7800d78
4aa675",
  "causal_parent": null
}
```

12. Interoperability Considerations

To remain interoperable, CEP implementations SHOULD preserve predictable envelope grammars, separate reference from verification states, expose mismatches rather than hiding them, converge schema identity across control and sealed planes, and support both operator readability and machine normalization.

13. Security Considerations

A receiver MUST NOT interpret a control-plane reference as proof of object integrity unless the referenced object has been acquired and verified under local policy.

14. IANA Considerations

This version of the document makes no IANA requests.

15. Normative References

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

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008, <https://www.rfc-editor.org/info/rfc5234>.

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

16. Informative References

[AINS] van de Meent, J. and R. AI, "AINS: AInternet Name Service", 2026.

[JIS] van de Meent, J. and R. AI, "JTel Identity Standard", 2026.

[TIBET] van de Meent, J. and R. AI, "Traceable Intent-Based Event Tokens", 2026.

[SSM] van de Meent, J., "TIBET Semantic Surface Manifest", 2026.

[TAT] van de Meent, J. and R. AI, "TIBET TAT: Touch-and-Transfer Protocol", 2026.

[IDDROP] van de Meent, J. and R. AI, "IDDrop: Identity Drop and Acceptance Protocol", 2026.

Appendix A. Initial Verification-State Registry

The following values are RECOMMENDED starting points for implementation alignment:

* referenced
* present
* digest-matched
* container-verified
* manifest-verified
* continuation-approved

Future versions of this document MAY elevate these values into a formal registry.

Appendix B. Changes from -00

This is the initial -00 version.

Appendix C. Acknowledgments

This draft emerged from continuityd, TBZ/TZA, AInternet, and JIS/TIBET design work across the HumoticaOS stack, including real operational feedback from early cross-host continuity handoffs.

Author’s Address

Jasper van de Meent
Humotica
Netherlands
Email: info@humotica.com
URI: https://humotica.com/