Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: 18 November 2026

J. van de Meent
Humotica

17 May 2026

IDDrop: Identity Drop and Acceptance Protocol
draft-vandemeent-iddrop-00

Abstract

This document specifies IDDrop, an identity-transfer protocol for moving identity claims, actor claims, and receiver-bound claim bundles across human-facing and machine-facing environments.

IDDrop defines two complementary operating modes: offer-first, intended for proximity and human-in-the-loop workflows, and request-first, intended for autonomous systems, daemon-to-daemon exchange, and policy-driven machine identity transfer.

IDDrop does not treat filenames or user-visible extensions as identity truth. Instead, it binds identity transfer to sealed carrier truth, semantic classification, actor provenance, receiver targeting, and causal validation. This document profiles the generic TIBET TAT handoff model for trustworthy 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 18 November 2026.

Copyright Notice

Table of Contents

1. Introduction 2
2. Terminology 3
3. Protocol Overview 4
4. Offer-First Mode 4
5. Request-First Mode 4
6. Validation Rules 5
7. Semantic Classification 5
8. Security Considerations 6
9. Transport and Carrier Bindings 6
9.1. Relationship to CEP and TAT 6
10. Examples 6
10.1. Human NFC Offer 7
10.2. System Request 7
11. Normative References 7
12. Informative References 7
Appendix A. Minimal JSON Shapes 7
Author's Address 8

1. Introduction

Identity transfer in distributed systems is frequently underspecified. Systems often know how to transport bytes, but not how to safely transfer the meaning of an identity claim.

Existing encrypted transport systems provide confidentiality and integrity for byte carriage, but they do not by themselves answer whether the transferred object is an identity claim at all, whether the claim was expected or solicited, whether the visible name matches sealed semantics, whether the claim is bound to a known actor registry, or whether the response belongs to a known causal request chain.

IDDrop fills this gap by defining a protocol for identity drop, acceptance, and validation across both human and system lanes.

This document is intended as an identity-transfer profile over the more general TIBET TAT transfer substrate and within the broader Continuity Envelope Protocol (CEP) model.

The key design principle is simple: identity MUST NOT arrive as a surprise commitment.

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.

IDDrop The protocol specified in this document for identity offer, request, response, validation, and acceptance.

Offer A temporary, visible, non-binding identity availability signal.

Request An explicit solicitation by a receiver for a specific claim or identity response.

Response A sealed and receiver-bound identity-carrying answer to either an accepted offer or a request.

Acceptance An explicit receiver-side act indicating consent to proceed from discovery into sealed identity transfer.

Materialization The act of committing a validated identity claim into a receiver runtime, registry, or operator-visible state.

Carrier Truth The byte-level truth established by sealed carrier properties such as magic bytes, authenticated encryption, signatures, and receiver binding.

Semantic Class The declared object class of a transfer unit, such as identity, code, document, command, or receipt.

Causal Binding The linkage between a request, offer acceptance, and later response, such that the receiver can verify that the identity answer belongs to a known prior step.

3. Protocol Overview

IDDrop operates across three conceptual planes: Discovery Plane, Sealed Transfer Plane, and Commitment Plane. Systems implementing IDDrop MUST NOT collapse these planes into a single implicit step.

IDDrop defines two operating modes:

* offer-first, for human, proximity, NFC, QR, and local discovery workflows

* request-first, for daemon, automation, cap-bus, and machine-to-machine workflows

4. Offer-First Mode

Offer-first mode is intended for NFC handoff, QR handoff, local discovery, AirDrop-like experiences, and home-agent or human-supervised device exchange.

The offer-first flow is:

1. sender enables a temporary offer

2. receiver discovers offer metadata

3. receiver chooses accept, challenge, or reject

4. if accepted, sender performs sealed transfer to receiver

5. receiver validates the claim

6. receiver MAY materialize the claim

An offer SHOULD contain at least ‘offer_id‘, ‘expires_at‘, ‘sender_pubkey‘, ‘claimed_aint‘, ‘claim_class‘, ‘semantic_type‘, ‘display_name‘, ‘preview_hash‘, and ‘ssm_class‘.

Offer metadata MUST NOT be treated as commitment truth.

5. Request-First Mode

Request-first mode is intended for daemon-to-daemon exchange, cap-bus identity requests, relay and automation lanes, and policy-driven system exchange.

The request-first flow is:

1. receiver emits a request

2. request carries ‘request_id‘ and ‘challenge_nonce‘

3. sender creates a receiver-bound response

4. receiver validates request/response linkage

5. receiver MAY materialize the claim

A request SHOULD contain at least ‘request_id‘, ‘challenge_nonce‘, ‘requested_claim_type‘, ‘receiver_pubkey‘, ‘requested_aint‘, ‘causal_parent‘, and ‘expires_at‘.

6. Validation Rules

The following six validation rules are REQUIRED.

1. *Carrier Truth*: the receiver MUST verify carrier truth before trusting object bytes.

2. *Semantic Class*: the receiver MUST determine the semantic class of the object before commitment. Byte validity alone is insufficient.

3. *Receiver Binding*: the receiver MUST verify that the response is bound to the intended receiver. A response not addressed to the receiver MUST be rejected.

4. *Claim Provenance*: the receiver MUST validate the claim against a provenance source such as AINS, JIS, or an equivalent actor registry.

5. *Causal Binding*: in request-first mode, the receiver MUST validate that the response belongs to the prior request by checking ‘request_id‘, ‘challenge_nonce‘, and any causal parent references. In offer-first mode, the receiver MUST validate that acceptance and later transfer belong to the same offer session.

6. *Commitment Discipline*: the receiver MUST NOT materialize or register an identity claim before Rules 1 through 5 have succeeded.

7. Semantic Classification

Implementations SHOULD distinguish at least the following semantic classes: identity, code, document, command, and receipt.

Implementations MUST NOT use filename extension alone as semantic truth.

8. Security Considerations

IDDrop forbids surprise identity commitment. Offer-first mode is safe only if offers remain temporary and non-binding until explicit acceptance and later validation. Request-first mode establishes intent and causal context, but the receiver MUST still validate the response and its provenance.

Implementations MUST treat user-visible names and extensions as advisory surfaces only. A man-in-the-middle or misleading sender MAY present a visually similar claim name. Provenance validation and receiver binding are REQUIRED to defeat such confusion.

Offers and requests SHOULD carry expiries. Responses SHOULD be bound to active requests or active acceptances to limit replay.

9. Transport and Carrier Bindings

IDDrop is transport-agnostic. It MAY be carried over NFC, QR-assisted local handoff, local network discovery, message relay, and cap-bus or command-substrate lanes.

When a sealed carrier is used, implementations SHOULD use a receiver-bound confidential carrier format rather than raw opaque payload carriage.

9.1. Relationship to CEP and TAT

IDDrop is not intended to redefine generic handoff mechanics.

CEP defines the umbrella continuity messaging model, TAT defines the generic consent-bound handoff and transfer flow, and IDDrop defines the identity-transfer profile over that flow.

When a TAT implementation is available, IDDrop SHOULD use it as the preferred transfer and anchoring substrate.

10. Examples

10.1. Human NFC Offer

A user enables identity offer mode for 60 seconds on a device. Another device taps via NFC, sees the claim metadata and sender fingerprint, and explicitly accepts. The sender then transfers a receiver-bound sealed response, which the receiver validates against AINS/JIS before materialization.

10.2. System Request

A daemon requests an identity claim for a specific actor and embeds a nonce and request identifier. The sender responds with a sealed, receiver-bound response referencing the original request. The receiver validates the request linkage and provenance before continuing the workflow.

11. 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>.

[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>.

12. Informative References

[AINS] van de Meent, J., "AINS Discovery", 2026.

[JIS] van de Meent, J., "JIS Identity", 2026.

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

[CEP] van de Meent, J., "Continuity Envelope Protocol", 2026.

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

[UPIP] van de Meent, J., "UPIP: Universal Process Integrity Protocol", 2026.

Appendix A. Minimal JSON Shapes

```
{
  "offer": {
    "offer_id": "offer-123",
    "expires_at": "2026-05-17T10:00:00Z",
    "sender_pubkey": "hex...",
    "claimed_aint": "jasper.aint",
    "semantic_type": "identity"
  },
  "request": {
    "request_id": "req-456",
    "challenge_nonce": "nonce-789",
    "receiver_pubkey": "hex...",
    "requested_claim_type": "identity"
  },
  "response": {
    "request_id": "req-456",
    "sender_pubkey": "hex...",
    "receiver_pubkey": "hex...",
    "claimed_aint": "jasper.aint",
    "semantic_type": "identity",
    "provenance_ref": "ains:..."
  }
}
```

Author’s Address

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