Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: 10 November 2026

J. van de Meent
Humotica

9 May 2026

TIBET Semantic Surface Manifest
draft-vandemeent-tibet-semantic-surface-manifest-00

Abstract

This document defines the Semantic Surface Manifest, a human-readable and policy-matchable routing layer for identity-bound continuity containers and TBZ-based sealed bundles.

The Semantic Surface Manifest exposes limited dispatch metadata such as time fragment, context, profile, and priority without exposing sealed content. It is intended for use in systems where routing decisions may need to occur before deep inspection, while trust remains anchored in intrinsic bundle properties such as magic bytes, manifests, hashes, signatures, and causal references.

In short: address visible, content sealed.

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 10 November 2026.

Copyright Notice

Table of Contents

1. Status of This Memo

This memo is an Internet-Draft working document derived from operational architecture notes and prototype work in the Humotica / TIBET / TAT / ICC stack during May 2026.

The present -00 version captures the core routing model, visible syntax, mirrored-surface concept, and mismatch consequences needed for first public review.

2. Problem Statement

Sealed containers often provide strong integrity but weak dispatch semantics.

Systems therefore face a recurring tradeoff: either encrypt and seal everything, delaying routing and policy choice until deep inspection, or expose too much metadata, weakening privacy and creating new security ambiguities.

The Semantic Surface Manifest addresses this by providing a constrained, readable routing layer that supports dispatch without decrypting content, minimizes metadata exposure, does not replace cryptographic verification, and composes with existing sealed-container workflows.

3. Terminology

Identity-Bound Continuity Container: A cryptographically sealed bundle that combines identity binding, continuity semantics, and containerized payload transfer.

Semantic Surface Manifest: A human-readable routing surface associated with a bundle, typically expressed through filename or object-name structure and optionally mirrored into sealed manifest fields.

Intrinsic Truth: Properties established by the sealed object itself, such as magic bytes, manifest, signatures, hashes, and chain anchors.

Extrinsic Surface: Properties expressed outside the sealed object for dispatch and routing, such as time fragment, context, profile, and priority.

Surface-Integrity Event: A meaningful mismatch or anomaly involving visible routing surface and mirrored sealed routing fields.

4. Design Goals

The Semantic Surface Manifest is intended to remain human-readable, machine-parseable, bounded in disclosure, and composable with existing ICC or TBZ verification workflows.

It should support wildcard or policy matching, align with logs and audit ecosystems, and support mirrored sealed fields for consistency checks. It must not be treated as proof of identity or content, override manifest truth, or carry rich payload details.

5. Syntax

The normative external form is:

<time-fragment>.<context>.<profile>.<priority>[.<icc-ext>]

The Semantic Surface Manifest is intentionally flat and dot-delimited in version 1. The formal grammar uses ABNF as defined in [RFC5234].

Each segment is restricted to lowercase letters, digits, and hyphens. Segments must not contain spaces, slashes, underscores, nested dots, or uppercase letters.

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

time-fragment   = date-frag [ "t" time-frag "z" ]
date-frag       = 4DIGIT "-" 2DIGIT "-" 2DIGIT
time-frag       = 2DIGIT "-" 2DIGIT
context         = 1*32(segment-char)
profile         = 1*16(segment-char)
priority        = 1*16(segment-char)
icc-ext         = 1*16(segment-char)
segment-char    = LCALPHA / DIGIT / "-"
LCALPHA         = %x61-7A
DIGIT           = %x30-39
```

6. Time Fragment Format

This document prefers an ISO8601-style fragment over compact local date forms because it is lexicographically sortable, readable across jurisdictions, aligned with logs, and supports both coarse and fine routing granularity.

Two forms are recommended in version 1:

2026-05-08
2026-05-08t18-38z

7. Vocabulary Registries

7.1. Profile Registry

Initial profile values include claude, gemini, gpt, kit, iddrop, parentattest, capsule, and tza. These values describe semantic class, not vendor authenticity.

7.2. Priority Registry

Initial priority values include urgent, normal, background, and sealed.

7.3. Context

The context field remains open-text in version 1 but is expected to be short, low-leakage, and ABNF-conforming.

8. Processing Model

8.1. Surface Parse

A compliant implementation may parse the semantic surface before opening the bundle in order to choose a queue, handler, retention policy, or operator lane.

8.2. Type Sniff

An implementation should verify container type using intrinsic signals such as TBZ magic bytes before deep handling.

8.3. Deep Verification

Before trust-sensitive operations, an implementation must verify the sealed container according to its intrinsic integrity rules.

8.4. Surface-to-Manifest Consistency

If the sealed manifest contains mirrored surface fields, the implementation should compare them against the external semantic surface. Meaningful mismatch should be treated as a surface-integrity event leading to triage, quarantine, or policy review rather than silent acceptance.

9. Mirrored Manifest Fields

This document defines optional mirrored manifest fields such as surface_time_fragment, surface_context, surface_profile, and surface_priority.

If both an external semantic surface and internal mirrored fields are present, the mirrored fields are authoritative for triage classification and deep semantic handling.

10. Example

Example external surface:

2026-05-08.redspecter-review.claude.urgent

Processing may route to an urgent queue, classify as a candidate profile, verify TBZ magic bytes, inspect the manifest, verify signatures and hashes, compare visible and sealed surface, and only then hand to a profile-aware handler if consistent or policy-approved.

11. Mismatch Classes

11.1. Cosmetic Mismatch

A visible label changes while sealed truth remains intact. Recommended disposition is triage with manifest semantics prevailing.

11.2. Routing-Risk Mismatch

A visible profile and a sealed profile differ in a way that creates significant misrouting risk. Recommended disposition is triage or quarantine, not auto-materialization.

11.3. No Mirrored Fields

Legacy bundles may provide visible routing only, yielding a reduced-assurance mode because no sealed-surface comparison is possible.

12. Security Considerations

The Semantic Surface Manifest is not a source of trust. Implementations must assume that external names can be changed and visible routing labels can be misleading. The sealed container remains the only strong source of truth.

Routing may depend on SSM, but trust must not depend on SSM alone.

13. Privacy Considerations

The Semantic Surface Manifest intentionally exposes limited metadata. Implementers should keep context low-sensitivity, avoid direct secrets or detailed personal data, and prefer naming for dispatch rather than disclosure.

14. Interoperability Considerations

The SSM is designed to compose with TBZ, ICC-based continuity containers, TIBET Drop or TAT flows, session-state bundles, attestation bundles, sealed capsules, local storage, transport objects, attachments, queues, and router decisions.

15. Relationship to JIS, TIBET, TAT, and ICC

This document does not replace JIS identity semantics, TIBET causal ordering, TAT transfer flow, or ICC sealed object semantics. It adds a visible routing surface above them.

A clean split is that JIS decides who is acting, TIBET decides causal truth, TAT decides transfer flow, ICC decides sealed object class, and SSM decides visible dispatch semantics.

16. Future Work

* richer but still bounded registries for profile
* explicit mirrored-surface validation modes
* MUX or SNAFT routing integration
* UI conventions for displaying safe routing metadata
* signed or policy-bound surface-to-manifest binding hints

17. Questions for Future Revisions

The following topics are non-blocking for the present -00 version and are recorded here to guide later discussion and interoperability work.

* whether profile should remain open-text or move to a tighter registry
* whether the optional suffix should be preserved, normalized, or ignored by parsers

* whether seconds-level time fragments should be allowed in version 1

* whether some domains should escalate all mismatch to quarantine while others allow low-risk auto-continue

18. IANA Considerations

This document requests two registries: a Surface Profile Registry and a Surface Priority Registry.

Registration policy for both is Expert Review as described in [RFC8126]. Initial profile values are claude, gemini, gpt, kit, iddrop, parentattest, capsule, and tza. Initial priority values are urgent, normal, background, and sealed.

No registries are requested for time-fragment, context, or icc-ext in version 1.

19. References

[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>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

Appendix A. Acknowledgements

The author thanks the Humotica team for editorial assistance, RFC outline preparation, mismatch class formalization, and the operational tooling that made the surface consistency model concrete.

The author also thanks Richard Barron of Red Specter Security Research for adversarial framing that helped sharpen the address visible, content sealed principle and the rename-attack perspective.

Author’s Address

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