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-02

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 (-02 revision)

This memo is an Internet-Draft revision (-02) of draft-vandemeent-tibet-semantic-surface-manifest. The -02 revision adds Section 13 (Sealed Authentication Module Extension) following work on bounded credential-delegation as a complement to the claim-verified pattern in Section 12. The change set follows the four-W family naming agreed with the implementation team on 12 May 2026: tibet-vault (WHEN), tibet-keychain (WHERE/HOW), tibet-sam (WHY), tibet-gateway (WHERE-EXEC).

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. Claim-Verified Extension (Privacy-Preserving Policy Answers)

This section defines an OPTIONAL extension to the SSM that allows a sealed envelope to carry a privacy-preserving policy answer such as an age-gate verification, without disclosing the underlying personal data used to derive that answer.

The Claim-Verified Extension is intended for challenge-response flows where: a verifier (e.g. a merchant) MUST be assured that a rule is satisfied (e.g. "user is age 18 or older"); a responder (e.g. a wallet on a Secure Enclave smartphone) holds the personal data needed to evaluate the rule locally; and neither side wishes to transmit the personal data itself.

12.1. Design goal

A response envelope using this extension answers: did this challenge satisfy the requested rule. It explicitly does NOT answer: who the person is, what their exact birth date is, what document number they carry, or any other personal payload.

12.2. Challenge envelope fields

The verifier issues a challenge as a sealed envelope carrying the following fields. Required: intent (policy intent, e.g. "verify_age"); threshold (rule parameter, e.g. 18); reply_to (verifier address as AINS or JIS-DID); challenge_nonce (opaque per-challenge random value); challenge_hash (sha256 of canonical challenge material). Optional: valid_until (expiry timestamp, RFC3339); policy_domain (e.g. "age-gate"); jurisdiction (country or policy-scope identifier); evidence_class (e.g. "minimal-disclosure").

12.3. Response envelope fields

The responder issues a response as a sealed envelope carrying the following fields. Required: intent (the response form, e.g. "verify_age_response"); challenge_hash (sha256 copied from the challenge); claim_verified (symbolic claim, e.g. "age_over_threshold"); threshold (MUST match challenge); result (boolean true or false). Optional: issuer_anchor (trust anchor reference); valid_at (time of local verification); result_scope (e.g. "session-only"); receipt_requested (boolean).

12.4. Normative privacy posture

A response envelope using this extension MUST NOT include full legal name, birth date, passport number, citizen number, face template, or raw credential payload. It MAY include trust anchor reference for the issuing credential, policy domain reference, and challenge-bound correlation material such as challenge_hash, only where needed for downstream verification or audit.

12.5. Semantics

A claim_verified response asserts: "a trusted wallet answered this exact challenge with a valid claim_verified result against the parameter threshold". It explicitly does NOT assert "this is natural person X". This keeps the statement policy-bound, not identity-maximal. The cryptographic binding between challenge and response via challenge_hash prevents replay across distinct verifier sessions.

12.6. Audit and CBOM implications

Implementations MAY record the challenge-response pair in a Causal Bill of Materials (CBOM) to enable subsequent regulatory review. A compliant CBOM trace SHOULD show: challenge received with challenge_hash; local policy evaluation performed; response emitted with matching challenge_hash; response accepted or denied by verifier; receipt logged if receipt_requested is true. The CBOM trace MUST NOT cause the merchant or auditor to retain personal identity payloads. The cryptographic chain alone is sufficient evidence that the policy answer was produced by an identity-bound wallet for a specific challenge.

12.7. Relationship to broader claim-verification

This extension is intentionally minimal. The same shape generalizes to other policy domains: verify_age with threshold (age-gate); verify_residency with jurisdiction (region-gate); verify_kyc_class with level (tiered consent); verify_role with role (attestation-bound capability). In all cases the responder discloses only the symbolic claim and boolean result, never the underlying personal evidence.

13. Sealed Authentication Module (SAM) Extension

This section defines an OPTIONAL extension to the SSM that permits a sealed envelope to carry an intent-bound, one-shot authentication artifact, such that an agent may request execution of a specific action without ever holding the underlying secret material in cleartext.

The SAM extension is intended for credential-bearing flows where: an upstream caller (e.g. an automated agent) needs to cause a specific action to occur (e.g. publish a package, sign a transaction, deploy a container); the calling actor MUST NOT possess the raw credential needed to perform that action; and a bounded execution boundary (e.g. tibet-gateway) holds custody of the credential and performs the action on behalf of the caller under cryptographically enforced scope.

13.1. Design goal

A SAM envelope answers: which single bounded act is this artifact permitted to authorise, by which actor, for which target, until when. It does NOT carry the secret material in a form the holder can read; the encrypted credential block can only be unsealed inside the trusted execution boundary that materialised the SAM.

This is the natural complement to the Claim-Verified Extension (Section 13): claim-verified responses say "this rule is satisfied"; SAM envelopes say "this single bounded act is now authorised".

13.2. SAM envelope fields

A SAM envelope carries the following fields in its sealed manifest. Required: intent (semantic label for the bounded act, e.g. "upload_package"); secret_id (opaque identifier of the underlying credential in the issuing keychain); target_action (the action endpoint or contract address); actor_id (JIS-DID of the actor permitted to present this envelope); valid_until (RFC3339 expiry timestamp); ephemeral_id (single- use identifier tied to this materialisation).

Optional: constraints (array of key-value pairs that further bound the act, e.g. package="tibet-zip", registry="pypi"); policy_lane (which policy regime authorised this SAM); receipt_required (boolean, whether a sealed receipt is expected after execution); supersedes_sam_id (link to a prior SAM this one replaces, in rotation chains).

13.3. Normative posture

A SAM envelope MUST NOT include the underlying secret material in cleartext form. The encrypted credential block, if present in the sealed payload, MUST be decryptable only by the execution boundary identified in the manifest (typically identified by its public key or AINS reference).

An execution boundary that materialises a SAM envelope MUST: validate the manifest signature against the issuing keychain; check that the actor_id matches the entity presenting the envelope (via TIBET-SIG-V1 or equivalent identity proof); confirm that valid_until has not elapsed; enforce that the intended action matches target_action and all constraints; and destroy the ephemeral session after execution completes, whether successful or not.

An execution boundary MUST NOT permit a SAM envelope to be replayed: the ephemeral_id is single-use and MUST be recorded in a non-replay log for at least the duration of valid_until plus the conservative network-clock-skew margin.

13.4. Causal chain implications

SAM materialisation is itself a causally-anchored event. The issuing keychain SHOULD record a secret-proxied event in its custody timeline (see related tibet-keychain specification), with action_id and parent_action_id linking the SAM materialisation to the originating intent request.

Upon execution, the execution boundary SHOULD emit a provenance-sealed response that references the SAM envelope’s sam_id and ephemeral_id, so that downstream audit walkers (e.g. tibet-cbom) can construct the full chain: intent request - SAM materialisation - execution - response. Each step is Ed25519-signed by the actor responsible for that step.

13.5. Example

Agent intent: "upload tibet-zip v1.0.3 to PyPI". Keychain materialises a SAM envelope with the following manifest fields:

intent = "upload_package", secret_id = "sec_pypi_001", target_action = "/upload/pypi", actor_id = "jis:humotica:codex", valid_until = "2026-05-12T10:34:18Z", ephemeral_id = "eph_d7fad2165e13", constraints = [package="tibet-zip", registry="pypi"].

The agent receives the SAM as a sealed .tza envelope; the agent cannot extract the underlying PyPI token. The agent forwards the envelope to a tibet-gateway endpoint, which validates the manifest, breaks the seal in its own runtime, executes the upload against PyPI, emits a provenance-sealed response back to the agent, and destroys the ephemeral session. The agent never possessed the PyPI token at any point.

13.6. Generalisation across credential domains

The same envelope shape generalises to any credential- bearing action where bounded delegation is required: publication to package registries (PyPI, crates.io, npm, Maven), source-code platforms (GitHub PATs, GitLab tokens), cloud APIs (AWS service accounts, GCP service credentials), signing operations (smart-contract transactions, code-signing keys, TLS-certificate issuance), and database access where the agent should not see the connection string. In all cases the envelope captures intent + scope, and the gateway performs the act under cryptographic enforcement.

13.7. Position in the four-W family

The SAM extension defines the WHY primitive in a four-part family of credential-life-cycle artifacts: a temporal-trigger layer (WHEN, see related tibet-vault specification) decides when a sealed result may be disclosed; a custody-and-timeline layer (WHERE/HOW, see related tibet-keychain specification) records where the secret lives and who touched it; the SAM layer (WHY, this section) authorises a single bounded act without releasing the secret; and an execution-boundary layer (WHERE-EXEC, see related tibet-gateway specification) safely performs the act. The four layers compose into one causally verifiable credential workflow.

14. 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.

15. 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.

16. 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.

17. 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.

18. 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

19. 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

20. 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.

21. 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/