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

J. van de Meent
Humotica

9 May 2026

TIBET Causal Time Substrate
draft-vandemeent-tibet-causal-time-02

Abstract

This document describes the TIBET Causal Time Substrate, a forward-only causal ordering model for identity-bound distributed systems.

TIBET does not treat wall-clock time as the primary ordering primitive. Instead, it uses a cryptographically bound logical-time structure encoded through append-only linkage, monotonic generation counters, and signed causal references. External wall-clock sources, including NTP, RFC 3161 timestamping services, Roughtime, GNSS, or public ledger timestamps, are treated as auxiliary alignment anchors rather than as the constitutive source of event order.

The core claim is simple: TIBET is a forward-only causal substrate that enables recovery and reversibility without rewriting history.

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

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Status of This Memo

This memo is an Internet-Draft revision (-02) of draft-vandemeent-tibet-causal-time, derived from operational architecture notes, prototype implementations, and forensic delivery work produced in the Humotica / TIBET stack during May 2026. The -01 revision added Section 12.4 (Filename-Surface Contract) following external peer review by Red Specter Security Research (RS-2026-001). The -02 revision adds Section 13 (Position in the Four-W Vault Family) following the 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 Causal Time Substrate underpins all four.

It is intended to capture and formalize an already deployed structural property of TIBET rather than to introduce a greenfield timing model.

This document is an initial public framing and substrate document intended to align distributed-systems theory, identity-bound execution, off-grid or degraded-network operations, and forward-only recovery semantics.

2. Problem Statement

Many distributed systems continue to over-privilege wall-clock time. They assume that safe ordering, freshness, replay defense, and reversibility can be grounded primarily in synchronized UTC.

That assumption is fragile under intermittent connectivity, NTP outage or misconfiguration, GNSS disruption, clock drift across edge nodes, compromised or ambiguous time authorities, and adversarial replay after restore or rollback.

This document argues that causal order should be primary, wall-clock time auxiliary, recovery forward-only, and history non-rewritable.

3. Terminology

Causal Time: The ordering of events by their dependency and sequence relationships, rather than by globally synchronized wall-clock timestamps.

Forward-Only Causal Substrate: A substrate in which valid state evolution occurs only by appending new causally linked events.

Logical Counter: A monotonic counter associated with event

generation and ordering. Within TIBET this corresponds to the generation field.

External Time Anchor: An observation of an external time-bearing source, recorded into the causal substrate as a signed event.

Drift Record: A signed record describing observed offset between local time and an external anchor or between two time-bearing participants.

Triage Fork: A forward-causal isolation path created in response to anomaly, mismatch, or uncertain continuity.

4. Design Goals

The TIBET Causal Time Substrate is intended to preserve causal ordering without dependence on absolute time, support cryptographic identity binding of events, permit recovery and revocation without history rewriting, and remain meaningful under offline or degraded-network conditions.

It is also intended to integrate with external time anchors, support replay-sensitive freshness checks, and surface time uncertainty honestly.

5. Model Overview

TIBET encodes causal order through a set of existing structural primitives including append-only linkage, signatures, generation counters, and parent references.

These map naturally onto a Lamport-style logical ordering model: happened-before pointers, tamper-evident order proof, authenticated event origin, logical counters, causal predecessors, and event-line roots.

The important consequence is that TIBET already behaves as a causally ordered logical-time substrate. This document formalizes that fact.

6. Relation to Lamport and Related Work

This document is directly aligned with Lamport’s logical-time model and the later vector-clock tradition associated with Fidge and Mattern.

TIBET extends that tradition by binding logical ordering to cryptographic identity, append-only lineage, and explicit fork, merge, and tombstone semantics.

7. Forward-Only Property

The defining property of TIBET causal time is not merely that events are ordered, but that valid evolution occurs only by moving forward.

Restore becomes fork rather than rewind, revocation becomes successor event rather than mutation, correction becomes amendment rather than overwrite, and cancellation becomes compensating action rather than erasure.

8. Event Classes

* ordinary application or system action events
* fork or snapshot-reference events
* merge or transfer-pair events
* tombstone events
* triage or quarantine events
* external time-anchor events
* drift-record events

These classes are not all necessarily already standardized in TIBET registries, but they align structurally with the substrate described here.

9. External Time Anchors

External time anchors provide auxiliary alignment between local causal order and broader time-bearing reference systems such as NTP, timestamping authorities defined by [RFC3161], Roughtime, GNSS, PTP, public ledger timestamps, or observed environmental anchors.

External time anchors may improve alignment. They must not redefine already established causal order.

10. Drift and Alignment

Clock drift is expected in real systems, especially edge systems and off-grid nodes. TIBET treats drift as a recordable condition, not as a collapse of ordering truth.

Implementations should be able to represent locally observed time, externally anchored time, offset between them, uncertainty window, and validity scope of the observation.

11. Processing Model

11.1. Local Event

When a local event is committed, the implementation increments or derives a monotonic generation value, binds the event to prior causal state, signs the event, and appends it.

11.2. Incoming Causal Reference

When a remote or transferred event enters local reasoning, the implementation evaluates causal relation and derives successor progression in Lamport style as max(local generation, remote generation) plus one for any new local successor event that depends on both.

11.3. Recovery

Recovery produces successor state by fork, compensation, revocation, or new forward-causal action. It must not pretend to restore the system to an earlier pre-committed causal state.

11.4. Time Anchoring

When external time is sampled, the implementation may write an external-anchor event and may write drift information, but must not use the anchor to invalidate already committed causal order.

12. Example Flows

12.1. Snapshot and Resume

In the TIBET model, a snapshot references a chain position and resume becomes a fork from that position. The new line advances with its own forward-only history.

12.2. Transfer Pair

In TAT or TIBET Drop, transfer_out records sender-side causal commitment, transfer_in records receiver-side causal acknowledgement, and successor generation derives from the maximum of local and sender generation plus one.

12.3. Semantic Surface Mismatch

If routing surface and sealed manifest differ, content may still be valid, but the causal substrate should treat the situation as anomaly and create a triage fork or isolation path.

12.4. Filename-Surface Contract

The semantic surface of a sealed continuity envelope is carried in two layers: the filename layer, observable without unpacking, and the manifest layer, authoritative only after Ed25519 signature verification.

These two layers MAY agree, partially agree, or conflict. A verifying implementation produces one of four surface_status verdicts: MATCH (all surface_* fields agree), PARTIAL (some agree, others absent or differ), MISMATCH (both layers present and at least one field conflicts -- potential rename attack), or NONE (neither layer has surface_* fields -- legacy bundle).

Filename construction is the operator’s responsibility. The packing library SHOULD provide a helper that derives a canonical filename from the manifest’s surface_* fields, so MATCH is the default for freshly packed envelopes. However, this specification does not require it.

PARTIAL is therefore a valid and expected verdict for envelopes that an operator has consciously renamed for human navigation (e.g. "vergadering-dinsdag.pdf"). It is NOT a defect; it is a recorded statement that the human-applied name diverges from the canonical-from-manifest name.

Implementations MUST be able to reconstruct the canonical filename from the manifest alone, so that any peer with the bundle can recover the original SSM-derived name after operator rename, record both the human name and the canonical name in audit so that rename events become explicit instead of silent, and detect MISMATCH as a candidate rename-attack signal distinct from PARTIAL (operator-renamed) and MATCH (no rename or aligned rename).

Audit records emitted by a verifying implementation SHOULD carry the field name "canonical_name" alongside the on-disk "name" field, with a boolean "renamed_by_operator" derived from their inequality. This naming convention is referred to as the canonical_name field throughout this specification; the helper function that computes it from a manifest is named canonical_filename().

This contract anchors causality to the visible surface: a peer seeing only the filename has a hint; a peer seeing the sealed content has the truth; the audit chain records the relationship between the two.

13. Position in the Four-W Vault Family

The Causal Time Substrate defined in this document does not stand alone. It underpins a four-part family of credential and continuity primitives, agreed informally by the implementation team on 12 May 2026 and elaborated in companion specifications.

13.1. Overview

The four-W family arose from the recurring observation that secret-handling systems consistently fail to answer four distinct questions: when may a sealed result be disclosed, how is the underlying secret stored and tracked, why is a specific bounded act permitted, and where is the act safely performed. Each W maps to its own primitive in the family.

The family members are: tibet-vault, the WHEN primitive, which is a temporal trigger that releases a sealed payload on a date or upon a dead-man-switch condition; tibet-keychain, the WHERE/HOW primitive, which records custody and timeline for where a secret currently lives, who touched it, when it was rotated or exposed, and how it moved between owner, custodian, and active operator; tibet-sam, the WHY primitive, which is a Sealed Authentication Module that permits a single bounded act without releasing the underlying secret and is specified as an extension to the Semantic Surface Manifest (see related document draft-vandemeent-tibet-semantic-surface- manifest, Section 13); and tibet-gateway, the WHERE-EXEC primitive, which is the execution boundary runtime in which a SAM envelope is unsealed, validated, executed against an upstream credential, and the ephemeral session destroyed.

13.2. Causal relation between the family members

The Causal Time Substrate provides the verifiable ordering and identity-bound action chain that lets the four family members interoperate. Every meaningful event in the family produces an entry in a TIBET audit chain: secret creation in tibet-keychain, SAM materialisation linking back to the issuing keychain action, gateway execution linking to the materialised SAM, and disclosure events in tibet-vault when applicable.

The forward-only property defined in Section 7 of this document ensures that none of these events can be retroactively altered: a credential-handling system built on this substrate cannot pretend that a secret was rotated earlier than it was, nor that an authorisation was granted before the prior revocation event.

13.3. Regulatory fit

The four-W family answers four distinct regulatory questions that traditional secret-store products do not natively address: who has access (custody timeline, WHERE/HOW primitive); why was a specific use authorised (intent and scope record, WHY primitive); where did the use occur (execution boundary record, WHERE-EXEC primitive); and when may the result be disclosed (temporal trigger, WHEN primitive). Implementations of any subset of these primitives SHOULD record their events into a shared Causal Time Substrate chain so that audit walkers (such as tibet-cbom) can reconstruct the full sequence after the fact.

14. Security Considerations

This document assumes an attacker may replay previously valid artifacts, re-inject old state through backup or restore channels, manipulate wall-clock sources or exploit clock drift, rename or relabel artifacts outside sealed causal truth, or attempt destructive rollback semantics through operational tooling.

The forward-only causal model is specifically designed to reduce replay-after-restore, replay-after-revoke, backup injection, clock-spoofing ambiguity, drift concealment, and destructive rollback.

The key invariant is that security-sensitive reversibility must be implemented as forward causal compensation, not as history rewrite.

15. Interoperability Considerations

This substrate is designed to compose with JIS identity, UPIP continuity, TAT transfer flow, TBZ or ICC container semantics, and semantic routing surfaces.

Interoperability therefore depends less on UTC agreement and more on shared causal encoding, verifiable linkage, explicit anchor semantics, and clear distinction between order and alignment.

16. Relationship to the Broader Humotica Stack

Within the broader architecture, Turing answers what computes, Lamport-style causal time answers when in event order, JIS answers who is permitted, and semantic framing layers answer within what semantic frame a process is interpreted.

TIBET occupies the causal-time substrate role while composing with the other axes.

17. Future Work

* standardized external time-anchor token shape
* standardized drift-record token shape
* explicit uncertainty handling
* causal freshness proofs for intermittently connected devices
* zero-disclosure continuity proofs over long time horizons
* alignment with RFC 3161, Roughtime, and public-ledger anchoring profiles

18. 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 drift should be standardized as its own token type or as a subclass of time-anchor event
* whether external anchor trust levels should be formally graded
* whether causal freshness should be defined as a reusable verification primitive
* how uncertainty and stale-anchor conditions should be surfaced in operator tooling
* whether some domains should require time-anchor presence while others permit purely local causal mode

19. IANA Considerations

This document has no IANA actions in the present version.

Future revisions may define registries for external time-anchor token shapes, drift-record token shapes, or causal freshness proof types, likely under Expert Review policy as described in [RFC8126].

20. References

[LAMPORT1978] Lamport, L., "Time, Clocks, and the Ordering of Events in a Distributed System", Communications of the ACM 21(7), 1978.

[RFC3161] Adams, C., "Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)", RFC 3161, August 2001, <https://www.rfc-editor.org/info/rfc3161>.

[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, internal peer review, and operational tooling that made this substrate framing concrete rather than theoretical.

The substrate framing also builds on the logical-time tradition established by [LAMPORT1978].

The author also thanks Richard Barron of Red Specter Security Research for adversarial validation that helped sharpen the forward-only recovery property under realistic attack conditions.

Author’s Address

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