Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 26 November 2026

L. Melegassi
Catellix
J. A. S. Barbosa
RedTeamLeaders

25 May 2026

MVPS Trust Profile: Lightweight Authentication via HMAC-SHA256, Operator Epoch Anchors, and Independent Witness Cosignatures for Multi-Vantage Path Snapshots

draft-melegassi-santos-ippm-mvps-cwt-00

Abstract

The Multi-Vantage Path Snapshot (MVPS) bundle format [I-D.melegassi-ippm-mvps-bundle] specifies a deterministic wire envelope but defers cryptographic binding of vantage reports. This document specifies the MVPS Trust Profile, named Coherent-Witness Trust (CWT), which closes that gap with minimal cryptographic overhead: HMAC-SHA256 per-snapshot authentication under an epoch-bound vantage session key, anchored every epoch by an Ed25519-signed Operator Epoch Manifest, and cosigned by independent witnesses on a bundle Merkle checkpoint.

The measured per-snapshot crypto cost is 2.1 us -- strictly less than the 4.2 us required to parse the JSON snapshot it protects. At N = 1000 vantages and 1 Hz tick, the total verification load is 0.21 % of one CPU core.

The profile inherits the full MVPS v4.0 theorem catalogue and proves two theorems not available in simpler per-snapshot signature designs: T-COAL-1 (multi-operator coalition resistance via witness independence) and T-SPLIT-1 (collector split-view resistance via cosignature quorum). Numerical receipts are in evidence/mvps_cwt_overhead_receipt.json and scripts/validate_cwt_theorems.py (12/12 PASS).

The design draws on two production precedents: the HMAC-personalized authentication of the Tesla vehicle-command protocol [TESLA-VEHICLE-CMD] and the witness-cosignature model of the Sigsum / tlog-witness ecosystem [C2SP-TLOG-WITNESS]. Neither codebase is reused; constructs are independently composed for the multi-vantage measurement setting.

Status of This Memo

progress."

This Internet-Draft will expire on 26 November 2026.

Copyright Notice

Table of Contents

1. Introduction

The MVPS bundle format [I-D.melegassi-ippm-mvps-bundle] is consumed by several detection profiles ([I-D.melegassi-coherence-bfd], [I-D.melegassi-ippm-mvps-coherence-leadtime], [I-D.melegassi-mvps-ddos-resilience], [I-D.melegassi-mvps-ai-coherence]). Section 10.2 of the base bundle acknowledges bundle poisoning and defers cryptographic binding. Section 10.3 acknowledges replay without mandating a mechanism.

Without authenticated vantage reporting, any detection profile that consumes MVPS bundles operates on unverified input. A compromised or spoofed vantage can inject arbitrary values into C_1, C_2, C_3 and D^2 without detection. This document closes that gap.

Three concrete problems motivate the design:

(a) Per-snapshot public-key verification (e.g., Ed25519) costs ~79 us, scaling to ~7.9 % of one CPU core at N = 1000 vantages / 1 Hz tick -- a non-trivial tax on the broker.

(b) Per-operator Sybil quotas protect only against a SINGLE-OPERATOR adversary. Multi-operator collusion (each operator within individual quota, coalition mass >= N/2) is not formally defeated without an independent third party.

(c) A compromised collector can present different bundle contents to different consumers (split-view) without breaking per-snapshot signatures.

This document specifies the MVPS Coherent-Witness Trust (CWT) profile. It addresses (a), (b), and (c) simultaneously with three minimal cryptographic layers at matched cadences:

LAYER 1: HMAC-SHA256 per snapshot (hot path; 2.1 us measured; Section 6).
LAYER 2: Ed25519-signed Operator Epoch Manifest (anchor; once per hour; Section 7).
LAYER 3: Ed25519 witness cosignatures on bundle Merkle checkpoint (split-view / coalition defense; Section 8).

CWT inherits the full MVPS v4.0 theorem catalogue and proves two additional theorems T-COAL-1 and T-SPLIT-1 (Appendix C). Proofs are in [MVPS-CWT-PROOF]; numerical receipt is in evidence/mvps_cwt_overhead_receipt.json; validator exit code is scripts/validate_cwt_theorems.py (12/12 PASS).

1.1. Design Goals

CWT is designed to be simultaneously:

- NEAR-ZERO OVERHEAD. The hot-path cost (HMAC-SHA256 per snapshot) is 2.1 us -- below the 4.2 us cost of parsing the snapshot it authenticates. Crypto overhead is invisible in any realistic broker budget (Section 14.1, measured receipt evidence/mvps_cwt_overhead_receipt.json).

- EMBEDDED-READY. HMAC-SHA256 is universally available as hardware primitive on ARM Cortex-M, IoT SoCs, and LEO satellite payloads. No per-snapshot Ed25519 key pair is required at the vantage; only a 32-octet session key.

- FORENSICALLY TRANSPARENT. Append-only Merkle checkpoints cosigned by independent witnesses give public auditors a tamper-evident log of every bundle without requiring access to per-snapshot MAC keys.

- COALITION AND SPLIT-VIEW RESISTANT. The witness layer closes two attack classes not addressed by simpler designs: multi-operator Sybil coalitions (T-COAL-1) and collector split-view (T-SPLIT-1). These are strengthening, not relaxation, relative to signature-per-snapshot approaches.

1.2. Design Rationale vs. Per-Snapshot Ed25519

A straightforward trust design would place one Ed25519 signature per snapshot (sign at vantage, verify at consumer). CWT does not use that approach for the following reasons, each quantified:

(a) COST. Ed25519 verify costs ~79 us on a modern core. At N = 1000 vantages, 1 Hz: 79 ms/s = 7.9 % of one core for authentication alone, before any coherence algebra. HMAC-SHA256 at the same workload: 2.1 ms/s = 0.21 %.

(b) EMBEDDED DEPLOYMENT. A 32-octet HMAC session key fits in any SoC key store. Ed25519 requires a 64-octet private key and a constant-time scalar-multiplication library, not universally available on constrained hardware.

(c) COALITION GAP. Per-snapshot signatures do not prevent a group of operators from collectively owning >= N/2 admitted vantages while each stays within individual quota. CWT closes this with witness cosignatures (T-COAL-1).

(d) SPLIT-VIEW GAP. A collector that signs bundles can present different "current" bundles to different consumers without violating any per-snapshot signature. CWT closes this by binding the collector to a single Merkle root per (bundle_id, bundle_seq) via witness cosignatures (T-SPLIT-1).

Implementations that require individual-snapshot non-repudiation (e.g., for legal evidence chains) MAY combine CWT Layer 1 with an additional per-snapshot Ed25519 signature in an extension field; this is out of scope for this document.

1.3. Relationship to MVPS v4.0 Mathematics

The MVPS v4.0 mathematical catalogue [MVPS-V4] is inherited verbatim under H-TRUST-CWT + H-ADM + H-WIT + f < N/2 + MVPS-A4. See Appendix C, Theorem T-COMP-2-CWT.

CWT additionally proves two theorems beyond the v4.0 catalogue:

T-COAL-1. Multi-operator coalition with combined Sybil mass f' >= N/2 is detected with overwhelming probability when at least floor(w/2)+1 of w independent witnesses are honest (Appendix C.6).

T-SPLIT-1. Collector split-view of a bundle checkpoint is detected by any two auditors querying any honest witness within one checkpoint window (Appendix C.6).

1.4. Production Precedents (Tesla vehicle-command and Sigsum)

Two production systems inform CWT's wire format choices without reusing their code:

- TESLA VEHICLE-COMMAND PROTOCOL <https://github.com/teslamotors/vehicle-command>. Authenticates safety-critical commands (vehicle unlock, drive enable) using HMAC-SHA256 with personalization fields (counter, epoch, expires_at). Demonstrates that HMAC-SHA256 with anti-replay personalization is acceptable for safety-critical authorization at scale on embedded hardware (BLE SoCs, Secure Elements). CWT's Section 6.1 personalization fields follow this pattern, adapted for measurement bundles.

- SIGSUM / TLOG-WITNESS [C2SP-TLOG-WITNESS]. Transparency log ecosystem with independent witnesses cosigning checkpoints to prevent split-view attacks. Built-in policy "sigsum-generic-2025-1" requires two logs and three witnesses. CWT's Layer 3 (Section 8 and Section 12) directly adopts the tlog-witness API surface (POST /add-checkpoint) and the cosignature format of [C2SP-TLOG-COSIGNATURE].

CWT is NOT a Sigsum log; the bundle Merkle tree is a small, per-coordination-window structure, not a long-running append-only log. However, the cosignature semantics and witness trust model are deliberately compatible to maximize reuse of existing tooling.

2. Terminology and Conventions

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.

Admitted set:
: Same as [I-D.melegassi-santos-ippm-mvps-trust]. Denoted A.

Coalition (adversary):
: A set of operators C with combined |coalition_vantages| >= N/2 in A, each operator within its individual quota.

Epoch:
: A bounded time interval, typically one hour, during which a

single operator session key K_v_epoch is used by vantage v. Identified by uint32 epoch_id, monotonically increasing per operator.

K_v_epoch:
: The HMAC-SHA256 key used by vantage v during epoch_id. Derived via HKDF [RFC5869] from an operator master key and the tuple (vantage_id, epoch_id).

Operator Epoch Manifest (OEM):
: An Ed25519-signed JSON document, published by an operator at the start of each epoch, binding K_v_epoch commitments to vantage identities.

Checkpoint:
: A Merkle tree root over the SHA-256 digests of snapshots contained in a bundle, plus metadata (bundle_id, bundle_seq, issued_at).

Witness:
: An independent service that verifies checkpoint consistency and cosigns checkpoints it accepts. Witness identity and cosignature format follow [C2SP-TLOG-COSIGNATURE].

Quorum:
: The minimum number of independent witness cosignatures REQUIRED for a checkpoint to be accepted by a consumer. Default 2.

Split-view:
: A failure mode in which a malicious collector presents different "current" bundle contents to different consumers.

3. Trust Boundary and Preconditions

3.1. H-TRUST-CWT: Authenticated Vantage Reporting

HYPOTHESIS H-TRUST-CWT. Every snapshot consumed by a CWT-conformant detector carries a valid HMAC-SHA256-Personalized tag (Section 6) computed under K_v_epoch, where the (vantage_id, K_v_epoch commitment, epoch_id) tuple is bound by a current Operator Epoch Manifest (Section 7).

Enforcement:

- Each snapshot carries a CWT trust wrapper (Section 6.3).
- The consumer verifies the HMAC tag using K_v_epoch obtained from the current OEM (Section 7) before parsing snapshot fields beyond the trust wrapper.
- Snapshots failing verification MUST be discarded and MUST NOT contribute to N or to C_1, C_2, C_3.

3.2. H-ADM: Admitted Vantage Set

Identical to Section 3.2 of [I-D.melegassi-santos-ippm-mvps-trust]: the consumer maintains an explicit admitted set A; only vantages in A count toward N.

Admission is established by the Operator Epoch Manifest plus the campaign Admission Policy Document; vantage_id strings alone do NOT prove membership in A.

3.3. H-WIT: Independent Witness Network

HYPOTHESIS H-WIT. The consumer accepts bundle checkpoints only when they carry cosignatures from at least q independent witnesses, where q is the campaign-configured quorum (default q = 2), drawn from a published witness anchor set W.

"Independent" in this document means:

- distinct organizational control (different legal entities or distinct administrative domains),
- distinct jurisdiction or distinct upstream AS path, where possible,
- no shared key material (each witness uses its own Ed25519 key pair).

Enforcement:

- Bundles whose checkpoint lacks q valid cosignatures from W MUST be rejected.
- Witness key rotation follows Section 12.1.

3.4. Impact on Downstream Theorems

Table 1 maps trust failures specific to CWT to MVPS preconditions and downstream effects.

Failure mode	Precondition lost	Downstream effect
Missing/invalid HMAC tag	H-TRUST-CWT	Snapshot rejected; N not incremented
Stale OEM (expired epoch)	H-TRUST-CWT (K_v binding lost)	All snapshots in stale window rejected
Coalition f' >= N/2 across operators	H-ADM (single-operator quota satisfied)	Theorem 9 would void except witnesses detect (T-COAL-1)
Collector split-view	H-WIT	Two views of same checkpoint detected by quorum disagreement (T-SPLIT-1)
Counter regress	anti-replay	Snapshot rejected
Expired snapshot (expires_at)	freshness	Snapshot rejected
Witness unavailability	H-WIT (quorum lost below q)	Consumer falls back to safe mode or rejects

4. Threat Model

4.1. Adversary Capabilities

Adversary A is modeled as in Section 4.1 of [I-D.melegassi-santos-ippm-mvps-trust] (compromise, impersonation, Sybil, collector compromise, MITM, replay, parser DoS, gauge gap),

extended with:

(i) Multi-operator coalition: A controls k <= number_of_operators operator accounts, each within individual quota, with combined admitted vantages >= N/2.

(j) Split-view collector: A controls the collector, observes bundle queries from different consumers, and tailors differing bundle contents per consumer.

A does NOT have:

(k) Ability to forge HMAC-SHA256 tags without K_v_epoch (HMAC unforgeable under PRF assumption [RFC2104]).

(l) Ability to forge Ed25519 signatures on OEMs without the operator private key.

(m) Ability to compromise floor(w/2)+1 of w independent witnesses (witness independence is operational hypothesis H-WIT).

4.2. Attack Classes

The following attack classes MUST be addressed by CWT implementations.

T-POISON (vantage lying):
Defense: HMAC tag verification + multi-vantage corroboration + C_1 speed-of-light bound.

T-COLL (collector tampering):
Defense: per-snapshot HMAC verified at consumer; checkpoint Merkle root cosigned by witnesses (Section 8).

T-MITM (unsigned transport):
Defense: HMAC over canonical snapshot bytes; TLS 1.3 [RFC8446] or DTLS 1.3 [RFC9147] on transport RECOMMENDED.

T-SYBIL (fake vantage majority within one operator):
Defense: operator quota (Section 5.2) and Admission Policy.

T-REPLAY (stale bundle):
Defense: per-snapshot counter + expires_at (Section 9).

T-PARSER (DoS):
Defense: Section 10 mandatory parser limits.

T-GAUGE (same fingerprint, different physical path):
Defense: Section 11 (multi-signal consumption).

T-COAL (multi-operator coalition): NEW.
Defense: witness independence (H-WIT) and quorum cosignature (Section 12). Even when sum of individual operator quotas reaches N/2, an adversary cannot influence the consumer's accepted bundle stream without colluding with floor(w/2)+1 witnesses (Theorem T-COAL-1, Appendix C).

T-SPLIT (collector split-view): NEW.
Defense: witness cosignature commits the witness to a single checkpoint per (bundle_id, bundle_seq). Two auditors querying any honest witness within one checkpoint window will detect disagreement (Theorem T-SPLIT-1, Appendix C).

4.3. Operational Impact Matrix

When H-TRUST-CWT, H-ADM, or H-WIT fail, the downstream profiles [I-D.melegassi-coherence-bfd], [I-D.melegassi-ippm-mvps-coherence-leadtime], [I-D.melegassi-mvps-ddos-resilience], and [I-D.melegassi-mvps-ai-coherence] lose integrity guarantees. Two CWT-specific failure modes are notable:

- Without H-WIT, AI-Coherence Part B is vulnerable to a single compromised operator + collector pair manufacturing "consensus" across all customer queries.
- Without H-WIT, Coherence-BFD ALARM can be selectively masked to specific customers via split-view.

Implementers SHOULD treat CWT as a hard dependency for production deployment, analogous to RPKI for origin validation.

5. Three-Tier Key Hierarchy

CWT defines exactly three layers of key material. Each layer uses a distinct primitive matched to its cadence.

5.1. Layer 1: Vantage HMAC Session Key

Each vantage v holds a per-epoch HMAC-SHA256 session key K_v_epoch, valid for the duration of one epoch. K_v_epoch is derived (Section 5.4) at the start of each epoch from:

- an operator master key K_op (held by the operator only),
- the vantage identifier vantage_id,
- the epoch_id (uint32, monotonically increasing per operator).

K_v_epoch is REQUIRED to be 32 octets. It is NOT distributed to consumers; only its commitment H(K_v_epoch) appears in the Operator Epoch Manifest (Section 7).

Consumers MUST obtain K_v_epoch via one of the following channels:

(a) An out-of-band per-campaign secret distribution (preferred for closed-operator deployments).

(b) A delayed-disclosure mechanism similar to TESLA [TESLA]: the operator publishes K_v_epoch in the NEXT epoch’s OEM along with the new H(K_v_(epoch+1)). Consumers verify past-epoch snapshots upon disclosure. This trades real-time verification for public auditability.

This document defines mode (a) as MANDATORY-TO-IMPLEMENT; mode (b) is OPTIONAL.

5.2. Layer 2: Operator Epoch Manifest (Ed25519)

Each operator holds a long-lived Ed25519 key pair (op_sk, op_pk). Once per epoch (typically once per hour), the operator publishes an Operator Epoch Manifest (OEM) signed under op_sk:

```
OEM = (epoch_id, operator_id, issued_at, expires_at,
       vantage_commitments[], policy_hash, op_signature)
```

where vantage_commitments[i] = (vantage_id_i, H(K_v_epoch_i), admission_tier_i).

The operator's op_pk is distributed via the campaign Admission Policy Document (signed by the campaign root, out of band).

OEM verification cost is paid once per epoch per operator (~150 us); it is amortized across all snapshots in the epoch.

5.3. Layer 3: Witness Network (Ed25519 Cosignatures)

The witness layer is composed of w >= 3 independent witnesses, each holding an Ed25519 key pair. Witness public keys are published in the campaign Admission Policy Document and SHOULD ALSO be published by each witness operator at a well-known URI.

When a collector emits a bundle, it computes the bundle checkpoint (Section 8) and submits the checkpoint to each witness via the POST /add-checkpoint endpoint specified by [C2SP-TLOG-WITNESS]. Each accepting witness returns a cosignature in the format of [C2SP-TLOG-COSIGNATURE].

A bundle is "witness-anchored" once it carries at least q cosignatures from distinct witnesses in W (Section 12.2).

5.4. Key Derivation via HKDF

K_v_epoch is derived as:

```
K_v_epoch =
   HKDF-SHA256(
      salt   = "MVPS-CWT-v1|operator:" || operator_id,
      ikm    = K_op,
      info   = "vantage:" || vantage_id || "|epoch:" || epoch_id,
      length = 32)
```

K_op is itself a 32-octet operator master secret, generated with a CSPRNG and stored with operational-secret protection.

Implementations MUST NOT reuse K_op across operators or campaigns.

5.5. Rotation and Compromise

K_op rotation:
SHOULD occur at least every 90 days. Triggers a campaign-wide manifest refresh.

K_v_epoch rotation:
Implicit: a new K_v_epoch is derived at every epoch_id increment. Default cadence: hourly.

Witness key rotation:
SHOULD occur at least every 180 days. Witness operators MUST publish revocation entries on compromise.

Operator Ed25519 key rotation:
SHOULD occur at least every 180 days.

Compromise of K_op exposes all in-window K_v_epoch derivatives; immediate K_op rotation (and OEM re-issuance with a fresh policy_hash) restores soundness for future epochs.

6. Snapshot Authentication (HMAC-Personalized)

6.1. Personalized Authentication Fields

Each authenticated snapshot carries the following personalization tuple in its trust wrapper:

```
{
  "epoch_id":   <uint32>,
  "counter":    <uint64, strictly increasing per (vantage_id,
                                                  epoch_id)>,
  "expires_at": <uint32 RFC 3339 seconds-since-epoch>,
  "vantage_id": "<string>",
  "operator_id":"<string>"
}
```

The expires_at field MUST be no later than the OEM’s expires_at for the matching epoch_id.

6.2. HMAC Computation

Let canonical_payload be the canonical UTF-8 byte sequence of the base bundle snapshot fields, sorted lexicographically per [I-D.melegassi-ippm-mvps-bundle] Section 5.1:

- destination, end-timestamp, hops, path_fingerprint, start-timestamp, vantage-id.

The authenticated input M is:

```
M = "MVPS-CWT-v1|"
 || canonical_payload
 || "|epoch:"   || ascii(epoch_id)
 || "|counter:" || ascii(counter)
 || "|expires:" || ascii(expires_at)
 || "|vantage:" || vantage_id
 || "|operator:"|| operator_id
```

The tag is:

```
tag = HMAC-SHA256(K_v_epoch, M)
```

tag is 32 octets, base64url-encoded for wire format.

6.3. Wire Format

The CWT trust wrapper is carried alongside the base snapshot:

```
{
  "snapshot": { ... base MVPS snapshot fields ... },
  "mvps_cwt": {
    "version":    1,
    "operator_id":"<string>",
    "vantage_id": "<string>",
    "epoch_id":   <uint32>,
    "counter":    <uint64>,
    "expires_at": <uint32>,
    "tag":        "<base64url, 32 octets>"
  }
}
```

The wrapper overhead is ~48 octets per snapshot (32-octet tag plus small uint fields), versus ~128 octets for a per-snapshot Ed25519 design (64-octet signature plus key identifier).

6.4. Verification Procedure

For each snapshot, the consumer:

1. Enforces Section 10 limits on raw byte size before JSON parse.

2. Extracts (operator_id, vantage_id, epoch_id, counter, expires_at, tag).

3. Loads OEM for (operator_id, epoch_id); verifies its Ed25519 signature, expires_at validity, and bundle admission policy.

4. Loads K_v_epoch via the channel of Section 5.1 (out of band) and checks H(K_v_epoch) against the OEM commitment.

5. Reconstructs M (Section 6.2) and computes expected_tag = HMAC-SHA256(K_v_epoch, M).

6. Performs constant-time comparison expected_tag == tag.

7. Checks counter strictly greater than last accepted counter for (vantage_id, epoch_id); checks now < expires_at.

8. Only then merges snapshot into bundle processing.

Failure at any step: discard snapshot, log trust failure, do not adjust N.

7. Operator Epoch Manifest

7.1. Manifest Content

```
{
  "oem_version":  1,
  "operator_id":  "<string>",
  "epoch_id":     <uint32>,
  "issued_at":    "<RFC3339Z>",
  "expires_at":   "<RFC3339Z>",
  "vantage_commitments": [
    {
      "vantage_id":      "<string>",
      "k_v_commitment":  "<hex, SHA-256 of K_v_epoch>",
      "admission_tier":  "atlas-probe | operator-noc | ...",
      "key_disclosure":  "out-of-band | delayed"
    }
  ],
  "policy_hash":    "<hex, SHA-256 of Admission Policy Document>",
  "ed25519_signature": "<base64url, 64 octets>"
}
```

The signature is computed over the canonical JSON of the OEM with the ed25519_signature field removed.

7.2. Distribution and Caching

OEMs SHOULD be served over HTTPS from a well-known operator URI. Consumers cache OEMs for the lifetime of expires_at; cache size is bounded by the number of operators and the overlap window between epochs (typically <= 2 OEMs per operator at any time).

7.3. Epoch Transitions

The operator MUST publish OEM_(epoch+1) before the current OEM's expires_at, with overlap >= 5 minutes (default), to avoid verification gaps. Snapshots submitted during overlap MAY carry either epoch_id; consumers MUST accept either, provided the OEM for that epoch is valid.

8. Bundle Checkpoint and Witness Cosignatures

8.1. Merkle Aggregation

For each bundle b containing snapshots s_1, ..., s_m, the collector computes:

```
leaf_i        = SHA-256(canonical(s_i with mvps_cwt wrapper))
checkpoint_root = MerkleTreeRoot(leaf_1, ..., leaf_m)
```

The Merkle tree construction follows [RFC9162] Section 2.1 (RFC 6962-bis), with leaf domain separator 0x00 and node domain separator 0x01.

8.2. Checkpoint Format

A checkpoint is a textual record in the format of [C2SP-TLOG-CHECKPOINT]:

```
origin:        "mvps-cwt/v1/<campaign_id>"
tree_size:     m
root_hash:     <base64 checkpoint_root>
timestamp:     <RFC3339Z issued_at>
extra:         "bundle_id=<uuid>;bundle_seq=<uint64>"
```

followed by one or more signature lines:

```
"- <key_id> <base64 signature>"
```

The collector signs the checkpoint with its own Ed25519 key. Witnesses cosign the checkpoint per Section 8.3.

8.3. Witness Submission Protocol

The collector POSTs the checkpoint (and any required consistency proof) to each witness's /add-checkpoint endpoint per [C2SP-TLOG-WITNESS] Section 3.

On accept, the witness returns a cosignature line in the [C2SP-TLOG-COSIGNATURE] format. The collector appends the cosignature lines to the checkpoint and includes the final checkpoint in the bundle envelope as field "cwt_checkpoint".

8.4. Quorum Policy

A consumer MUST accept a bundle only if its cwt_checkpoint carries at least q valid cosignatures from witnesses listed in the campaign Admission Policy Document, where q is the quorum parameter (default q = 2).

Cosignatures are valid when:

- the signing key matches a witness key_id in W,
- the witness has not been revoked at the checkpoint timestamp,

- the cosignature verifies under [C2SP-TLOG-COSIGNATURE] semantics over the checkpoint bytes.

9. Anti-Replay (Embedded Counter)

9.1. Counter Semantics

counter is a uint64 strictly increasing per (vantage_id, epoch_id). counter resets to 0 at every epoch_id increment.

This eliminates the need for the bundle-level (bundle_id, bundle_seq) cache REQUIRED by [I-D.melegassi-santos-ippm-mvps-trust] Section 8.2; CWT consumers maintain only the per-(vantage_id, epoch_id) last-accepted counter, which is bounded by N vantages.

9.2. Freshness via expires_at

The per-snapshot expires_at field defines the consumer-side freshness window. Consumers MUST reject snapshots where now >= expires_at. No separate freshness_ttl parameter is needed.

9.3. Minimal Replay Cache

Per-(vantage_id, epoch_id) state at the consumer:

```
last_counter[vantage_id][epoch_id] : uint64
```

Size: O(N) per active epoch. Two epochs maximum during overlap window. Garbage-collected on epoch expires_at.

10. Parser Safety Limits

The mandatory limits of [I-D.melegassi-santos-ippm-mvps-trust] Section 9 apply unchanged:

max_bundle_bytes:	1 048 576
max_snapshots:	256
max_hops_per_snapshot:	128
max_json_depth:	32
max_string_length:	4096
max_vantage_id_length:	128
max_extensions_per_hop:	8

In addition, CWT-specific limits:

max_oems_cached:	64
max_witness_cosigs:	16
max_epoch_overlap:	7200 s

11. Path Fingerprint: Security Semantics

The gauge-gap semantics of [I-D.melegassi-santos-ippm-mvps-trust] Section 10 apply unchanged. CWT consumers MUST combine path_fingerprint with per-snapshot RTT (C_1), cross-vantage distribution divergence (C_2), and edge-set Jaccard (C_3) before declaring coherence.

Detectors using ONLY the path-fingerprint are NOT CWT-conformant.

12. Witness Network Architecture

12.1. Witness Identity and Key Publication

Each witness publishes:

- an Ed25519 public key in [C2SP-TLOG-COSIGNATURE] format,
- a "witness identity descriptor" specifying organizational identity, jurisdiction, contact, key validity window,
- a revocation feed (signed entries listing revoked keys).

The campaign Admission Policy Document lists the witness public keys to be accepted, with assigned key_id labels.

12.2. Cross-Organization Requirement

The witness anchor set W MUST satisfy:

```
|W|     >= 3,
q       >= 2     (default; campaigns MAY require higher),
diversity(W) such that no single organizational entity controls
                 more than floor(|W|/2) witnesses.
```

For |W| = 3, no entity controls more than 1. For |W| = 5, no entity controls more than 2. These are the operational instantiations of H-WIT (Section 3.3).

12.3. Split-View Detection

A consumer suspecting split-view MAY query any honest witness (out of band) for its history of cosigned checkpoints for a given campaign + bundle_id. Any two consumers comparing their accepted bundles for the same (bundle_id, bundle_seq) MUST receive identical root_hash values when any single honest witness is queried by both.

This is the operational instantiation of Theorem T-SPLIT-1 (Appendix C).

12.4. Witness Liveness

A witness MAY fail or be temporarily unreachable. If fewer than q cosignatures are obtained within a configured timeout (default 5 s after collector submission), the collector:

- MAY publish a partial-quorum bundle marked "degraded": true,
- MUST NOT silently omit the degraded flag.

Consumers in production mode MUST reject degraded bundles unless campaign policy permits degraded ingestion explicitly (e.g., for continuity during scheduled witness maintenance).

13. Interoperability with Coherence-BFD

Coherence-BFD [I-D.melegassi-coherence-bfd] uses AuthHMAC-SHA256 on the UDP control plane. That HMAC key MUST be cryptographically independent from K_v_epoch of this document. Implementations MUST derive the Coherence-BFD HMAC key under a distinct HKDF label (e.g., "coherence-bfd-auth-v1") even if the underlying K_op is

shared.

Consumers using both protocols MUST verify CWT trust on bundle ingest independently of Coherence-BFD session state.

14. Operational Considerations

14.1. Cost Analysis (Measured Receipt)

The following per-operation latencies were measured on an end-to-end run of scripts/bench_cwt_overhead.py and stored at evidence/mvps_cwt_overhead_receipt.json (machine-readable, reproducible). The corresponding figure is docs/figures/bench_cwt_overhead.png.

Median per-op latency (canonical snapshot = 497 bytes):

```
   json_decode (baseline parse)  :  4.20 us
   sha256 (1 leaf)               :  0.70 us
   hmac_sha256 (CWT hot path)    :  2.10 us    <--- CWT per-snap
   hkdf_sha256 (per-epoch derive):  4.10 us
   ed25519_sign  (Trust signer)  : 28.70 us
   ed25519_verify (Trust verify) : 78.80 us    <--- Trust per-snap
```

KEY CLAIM (verified): the CWT hot-path HMAC tag costs LESS than parsing the JSON snapshot it protects (2.10 us vs 4.20 us). Crypto overhead is therefore strictly below the parse-itself overhead a consumer ALREADY pays.

At N = 1000 vantages with 1 Hz tick:

```
   CWT load   : 0.21 % of one CPU core
   Trust load : 7.88 % of one CPU core
   Baseline   : 0.42 % of one CPU core (json_decode alone)
```

CWT speedup over Trust per-snapshot: 37.5x.

Per-epoch costs (amortized hourly per operator) and per-checkpoint costs (amortized per witness submission) are dominated by Ed25519 verify at ~29 us sign / ~79 us verify; at the default epoch = 1 hour, q = 2 witnesses, the per-epoch overhead is well under 1 ms per operator per epoch and is therefore measurement-noise relative to baseline broker operations.

The receipt also records platform metadata (CPU, OS, Python version) and the iteration counts used (200 000 fast ops, 30 000 slow ops with warmup), to permit independent reproduction.

14.2. Embedded and LEO Vantages

On ARM Cortex-M class devices, HMAC-SHA256 is typically available as a hardware primitive or as a well-optimized software routine. Ed25519 implementations exist but are heavier in code size and energy.

CWT vantages need to:

- hold K_v_epoch (32 octets) for the current epoch,
- compute HMAC-SHA256 over canonical_payload + personalization,
- derive K_v_(epoch+1) at epoch boundaries.

Memory footprint at vantage: < 1 KiB persistent state.

14.3. Production Precedents

The HMAC-SHA256-Personalized pattern of Section 6 is derived from the Tesla vehicle-command protocol, which uses the equivalent structure (epoch + counter + expires_at + tag) to authenticate safety-critical commands at scale on embedded automotive hardware.

The witness cosignature pattern of Section 8 is derived from the Sigsum / tlog-witness ecosystem, whose policy "sigsum-generic-2025-1" (two logs, three witnesses) is in production use for transparency-log signatures.

Neither precedent is reused as code; both are referenced to demonstrate that the underlying primitives have been deployed in security-critical contexts.

14.4. Archival Mode

For archival research, consumers MAY accept expired epochs provided:

- the OEM’s ed25519_signature still verifies,
- the bundle checkpoint carries the original witness cosignatures,
- the consumer explicitly declares "archival" mode out of band.

Archival mode MUST NOT feed live detectors.

15. Security Considerations

This entire document is security-focused. Additional notes specific to CWT:

- K_v_epoch unforgeability reduces to HMAC-SHA256 PRF assumption [RFC2104].
- K_op compromise exposes all in-epoch K_v_epoch. Mitigation: keep K_op in HSM or sealed storage; rotate every 90 days.
- Operator Ed25519 compromise allows manifest forgery for that operator. Mitigation: witness cosignatures bind consumers to a checkpoint independent of the operator’s manifest, limiting the attack to past-epoch reattribution.
- Witness compromise reduces the effective quorum. Campaigns SHOULD set quorum q with margin (e.g., q = 2 of 5 rather than q = 2 of 3).
- Post-quantum migration: HMAC-SHA256 is widely believed to retain unforgeability against quantum adversaries with halved security margin (Grover); Ed25519 OEM and witness signatures are quantum-vulnerable in the long term. A future PQ-MVPS-CWT revision MAY substitute post-quantum signature schemes at Layers 2 and 3 without changing Layer 1.

16. Privacy Considerations

OEMs and witness cosignatures may contain operator identifiers and

ASN scope. Publications MUST respect the redaction guidance of [I-D.melegassi-ippm-mvps-bundle] Section 9. Admission tiers SHOULD use coarse labels in public bundles.

17. IANA Considerations

This document requests registration of one MVPS Bundle Capability Flag in the registry defined by [I-D.melegassi-ippm-mvps-bundle] Section 11.3:

Flag name: cwt-profile-v1
Semantics: Bundle includes mvps_cwt per snapshot, an Operator Epoch Manifest reference, and a witness-cosigned cwt_checkpoint. Consumers MUST verify per Section 6, Section 7, and Section 8 before coherence computation.

If the base registry is not yet established, implementations SHOULD use the string "cwt-profile-v1" in a bundle-level "capability_flags" array until IANA assignment.

18. Acknowledgements

The authors thank Joas Antonio dos Santos Barbosa for the security review of the MVPS bundle format that motivated this trust profile and for the outcome-driven red-team perspective that guided the choice of minimal cryptographic primitives.

The authors acknowledge the Tesla vehicle-command protocol maintainers and the Sigsum / tlog-witness community whose production designs are the conceptual ancestors of Sections 6 and 8 respectively. No code from either project is incorporated in this specification.

The authors thank Benoit Donnet (ULiege) for earlier correspondence on key-management granularity that informed the three-tier model.

19. References

19.1. Normative References

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5869] Krawczyk, H. and P. Eronen, "HMAC-based Extract-and-Expand Key Derivation Function (HKDF)", RFC 5869, May 2010.

[RFC6234] Eastlake 3rd, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, May 2011.

[RFC8032] Josefsson, S. and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)", RFC 8032, January 2017.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, May 2017.

[RFC9162] Laurie, B., Messeri, E., and R. Stradling, "Certificate

Transparency Version 2.0", RFC 9162, December 2021.

[I-D.melegassi-ippm-mvps-bundle]
Melegassi, L., "Multi-Vantage Path Snapshot (MVPS): A Canonical Bundle Format for Coordinated Traceroute Measurements", draft-melegassi-ippm-mvps-bundle-00, May 2026.

19.2. Informative References

[I-D.melegassi-santos-ippm-mvps-trust]
Melegassi, L. and Santos, J. A., "MVPS Trust Profile: Authentication, Parser Safety, and Threat Model for Multi-Vantage Path Snapshots", draft-melegassi-santos-ippm-mvps-trust-00, May 2026.

[I-D.melegassi-coherence-bfd]
Melegassi, L., "Coherence-BFD: Sub-Second Coherence Detection Using Bidirectional Forwarding Detection Patterns", draft-melegassi-coherence-bfd-00, May 2026.

[I-D.melegassi-ippm-mvps-coherence-leadtime]
Melegassi, L., "Empirical Lead-Time and Zero-Day Predictability Profile for Multi-Vantage Path Coherence", draft-melegassi-ippm-mvps-coherence-leadtime-00, May 2026.

[I-D.melegassi-mvps-ddos-resilience]
Melegassi, L., "Volume-Independent DDoS Detection via Coherence-BFD: The MVPS DDoS Resilience Profile", draft-melegassi-mvps-ddos-resilience-00, May 2026.

[I-D.melegassi-mvps-ai-coherence]
Melegassi, L., "MVPS AI-Coherence Extension: Semantic, Byzantine, and Infrastructure-Cognitive Monitoring", draft-melegassi-mvps-ai-coherence-00, May 2026.

[I-D.melegassi-iab-mvps-architecture]
Melegassi, L., "MVPS Architecture: Specification Conformance for the Multi-Vantage Path-Coherence Drafts", draft-melegassi-iab-mvps-architecture-00, May 2026.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, August 2018.

[RFC9147] Tschofenig, H. and T. Fossati, "The Datagram Transport Layer Security (DTLS) Protocol Version 1.3", RFC 9147, February 2022.

[C2SP-TLOG-WITNESS]
Community Cryptography Specification Project, "tlog-witness: A Witness Protocol for Transparency Logs", <https://c2sp.org/tlog-witness>.

[C2SP-TLOG-CHECKPOINT]
Community Cryptography Specification Project, "tlog-checkpoint: Checkpoint format for transparency logs", <https://c2sp.org/tlog-checkpoint>.

[C2SP-TLOG-COSIGNATURE]
Community Cryptography Specification Project, "tlog-cosignature: Cosignature format for transparency logs", <https://c2sp.org/tlog-cosignature>.

[TESLA-VEHICLE-CMD]
Tesla, Inc., "vehicle-command Protocol Specification", <https://github.com/teslamotors/vehicle-command/blob/main/pkg/protocol/protocol.md>, 2024.

[TESLA] Perrig, A., Canetti, R., Tygar, J.D., and D. Song, "The TESLA Broadcast Authentication Protocol", CryptoBytes Vol. 5, No. 2, 2002.

[MVPS-V4] Melegassi, L., "MVPS Mathematical Existence Proof -- Version 4.0", Catellix technical note, May 2026.

[MVPS-CWT-PROOF]
Melegassi, L. and Santos, J. A., "MVPS Coherent-Witness Trust -- Mathematical Existence Proof", Catellix technical note, May 2026.

Appendix A. JSON CWT Extension (Informative)

Minimal CWT-wrapped snapshot (illustrative keys only):

```
{
  "snapshot": {
    "destination": { "address": "203.0.113.1" },
    "end-timestamp": "2026-05-25T12:00:01Z",
    "hops": [
      { "index": 1, "address": "198.51.100.1", "rtt_ms": 1.2 }
    ],
    "path_fingerprint":
      "abc123...",
    "start-timestamp": "2026-05-25T12:00:00Z",
    "vantage-id": "v-probe-sp4-001"
  },
  "mvps_cwt": {
    "version":    1,
    "operator_id":"redteamleaders-br",
    "vantage_id": "v-probe-sp4-001",
    "epoch_id":   42,
    "counter":    17,
    "expires_at": 1779274800,
    "tag":        "base64url-32-octets..."
  }
}
```

Appendix B. Worked Verification Example

1. Campaign publishes Admission Policy Document listing operator Ed25519 keys and witness Ed25519 keys (out of band).

2. Operator generates K_op, derives per-vantage K_v_epoch for epoch 42, publishes Operator Epoch Manifest signed under op_sk.

3. Vantage receives K_v_epoch via out-of-band channel (Section 5.1(a)). Computes HMAC-SHA256 tag for each snapshot with monotonic counter.

4. Collector aggregates snapshots into a bundle, computes Merkle checkpoint, submits to each of 3 witnesses via POST /add-checkpoint.

5. Witnesses verify checkpoint consistency (this campaign's append-only history), return cosignatures.

6. Collector publishes bundle with cwt_checkpoint containing the

collector signature + 2 of 3 witness cosignatures (q = 2).

7. Consumer verifies:
   (a) OEM signature under operator's op_pk,
   (b) HMAC tag of each snapshot under K_v_epoch,
   (c) counter monotonicity per (vantage_id, epoch_id),
   (d) freshness via expires_at,
   (e) checkpoint cosignatures meet q witnesses from W,
   (f) Section 10 parser limits before any of the above.

If step 7 succeeds, snapshots feed C_1, C_2, C_3, D^2.

Appendix C. Mathematical Core (Normative)

This appendix states the trust-layer theorems of CWT. Full proofs appear in [MVPS-CWT-PROOF]. Profile-independent necessity theorems (T-VOID-*, T-GG-*, T-PARSE-1) are cited from [MVPS-CWT-PROOF] without re-statement.

C.1. Definitions (summary)

Verify-CWT(S) is the acceptance predicate of Section 6.4 plus the bundle-checkpoint quorum check of Section 8.4. B_trusted is the bundle content after all snapshots and the bundle checkpoint pass Verify-CWT. Admitted set A, Byzantine count f, gauge-gap witness (P1, P2), witness anchor set W with quorum q are as in [MVPS-CWT-PROOF] Section 1.

C.2. Authentication theorems

THEOREM T-AUTH-CWT-1 (Origin authentication soundness).
Under HMAC-SHA256 PRF security [RFC2104] and Ed25519 EUF-CMA [RFC8032] on the OEM, Verify-CWT(S) = ACCEPT implies the snapshot was produced by an entity holding K_v_epoch for the claimed (vantage_id, epoch_id), and that K_v_epoch was bound to vantage_id by a current OEM signed by the operator, except with negligible probability.

THEOREM T-AUTH-CWT-2 (Honest completeness).
A correctly tagged honest snapshot under a valid OEM, with monotonic counter and live expires_at, is accepted by Verify-CWT.

C.3. Impossibility without authentication (NECESSITY)

THEOREM T-VOID-1 (see [MVPS-CWT-PROOF] Section 3).
THEOREM T-VOID-2 (see [MVPS-CWT-PROOF] Section 3).
THEOREM T-VOID-3 (see [MVPS-CWT-PROOF] Section 3).
THEOREM T-VOID-4 (see [MVPS-CWT-PROOF] Section 3).

These NECESSITY theorems are profile-independent: they show what fails in any unauthenticated MVPS deployment.

C.4. Gauge-gap theorems

THEOREM T-GG-1 (see [MVPS-CWT-PROOF] Section 2).
THEOREM T-GG-2 (see [MVPS-CWT-PROOF] Section 2).
THEOREM T-GG-3 (see [MVPS-CWT-PROOF] Section 2).

C.5. Composition with MVPS v4.0 (SUFFICIENCY)

THEOREM T-COMP-1-CWT.
Under H-TRUST-CWT, H-ADM, H-WIT, and f < N/2, [MVPS-V4] Theorem 9 (geometric-median max-bias on C_2) applies to B_trusted.

THEOREM T-COMP-2-CWT.
Under H-TRUST-CWT, H-ADM, H-WIT, f < N/2, and MVPS-A4, the full v4.0 catalogue and the Architecture Invariance Theorem of [I-D.melegassi-iab-mvps-architecture] apply unchanged to B_trusted.

C.6. Coalition and Split-View theorems (CWT-specific)

THEOREM T-COAL-1 (Coalition resistance via witness independence).
Let an adversary control a coalition of operators with combined admitted vantage mass f' >= N/2. Suppose:
(i) the witness anchor set W contains w >= 3 independent witnesses,
(ii) no organizational entity controls more than floor(w/2) witnesses,
(iii) quorum q >= 2 is enforced.
Then any bundle whose contents reflect the coalition's poison is detected by an external auditor querying any honest witness within one checkpoint window. Equivalently: T-COMP-1-CWT holds with effective f_effective = (f' - witness_correction) < N/2 from the auditor's perspective.

THEOREM T-SPLIT-1 (Split-view resistance via cosignature quorum).
Let a malicious collector hold a valid collector key. Under q >= 2 and witness independence, any two consumers receiving "the same" bundle (matching bundle_id, bundle_seq) will obtain identical root_hash values when at least one honest witness participates in the cosignature, except with negligible probability. Equivalently: collector split-view is detectable with probability at least 1 - (1 - 1/w)^q for q honest cosignatures of w witnesses.

C.7. Parser and replay

THEOREM T-PARSE-1 (see [MVPS-CWT-PROOF] Section 6).

THEOREM T-REPLAY-1-CWT.
Strictly monotonic per-(vantage_id, epoch_id) counter combined with per-snapshot expires_at field rejects byte-identical replay and rejects beyond-window stale replay, using only O(N) consumer state per active epoch.

C.8. Designs and conjectures (not theorems)

Design D-CWT-ROT: key rotation cadence (Section 5.5).

Design D-CWT-DLY: delayed-key disclosure (TESLA-style; Section 5.1(b)) trades real-time verification for public auditability.

Conjecture CONJ-CWT-A4: witness diversity restores approximate conditional independence (MVPS-A4) of vantage observations. EMPIRICAL.

C.9. Numerical receipt

Constructive witnesses for T-AUTH-CWT-1, T-AUTH-CWT-2, T-COAL-1,

T-SPLIT-1, T-COMP-1-CWT, T-REPLAY-1-CWT, T-VOID-*, T-GG-*,
and T-PARSE-1 are validated by scripts/validate_cwt_theorems.py
(12/12 PASS). Implementations SHOULD run this script in CI
alongside scripts/validate_v4_against_all_attacks.py.

Authors' Addresses

Leonardo Melegassi
Catellix
Andradina, SP
Brazil

Email: melegassi@catellix.com
URI: https://catellix.com/

Joas Antonio dos Santos Barbosa
RedTeamLeaders
Brazil

Email: joasantonio108@gmail.com