Network Working Group
Internet-Draft
Intended status: Informational
Expires: 22 September 2026

T. Adebayo
F. Makanjuola
O. Apalowo
Veridom Ltd

21 March 2026

Operating Model Protocol (OMP): A Deterministic Routing and Evidence Architecture for AI Decision Accountability in Regulated Industries

draft-veridom-omp-00

Abstract

Regulated institutions deploying AI face a structural evidence problem: the gap between what their systems claim to do and what they can prove happened at the level of an individual decision. Existing compliance infrastructure addresses this problem observationally -- monitoring, dashboards, retrospective reporting -- but does not close it structurally.

This specification defines the Operating Model Protocol (OMP), a deterministic routing invariant that classifies every interaction in a regulated operation to exactly one of three outcome states -- AUTONOMOUS, ASSISTED, or ESCALATED -- and generates a tamper-evident audit trace at the point of every decision. The routing decision is a deterministic function of a composite Confidence Score, Watchtower enforcement evaluations, and domain-specific thresholds. Given identical inputs, the protocol produces identical outputs. This invariance is the architectural basis of the regulatory claim.

Each audit trace is sealed using a three-layer cryptographic integrity architecture: SHA-256 content hash, RFC 3161 trusted timestamp from an accredited third-party Timestamp Authority, and institution signature. The chain forms a Merkle structure in which modification of any historical trace invalidates all subsequent chain hashes. Per-decision accountability is therefore verifiable by any third party without access to the institution’s or Veridom’s infrastructure.

The protocol has been independently instantiated across four regulated domains -- digital credit under the Kenya CBK NDTCP framework, FCA Consumer Duty, FCA agent distribution oversight, and US legal AI supervision under ABA Rule 5.3 -- with the same two invariants holding in each: deterministic classification and immutable audit trail. Domain-specific profiles for digital credit and cooperative lending are published separately as draft-veridom-omp-ndtcp and draft-veridom-omp-sacco.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Regulated institutions globally share a structural failure: the gap between what they claim their operations do and what they can prove happened. This gap manifests in three distinct layers:

* *Policy-to-Practice Gap:* The institution has the required policies. Its agents, systems, and processes do not consistently follow them. The gap is invisible until an examiner asks to see evidence.

* *Practice-to-Evidence Gap:* Practices are compliant, but the contemporaneous evidence trail is absent or incomplete. Decisions are made but not recorded at the required granularity.

* *Evidence-to-Examination Gap:* Evidence exists but is not producible in the form a regulator requires -- scattered across systems, formatted for operations rather than audit, and not independently verifiable.

Every existing compliance tool accepts at least one of these layers as permanent. Monitoring tools accept that violations will occur and report them after the fact. GRC platforms accept that evidence will be incomplete and provide storage for what exists. Compliance dashboards accept the examination gap and help institutions produce narratives instead of proofs.

OMP refuses these concessions. The protocol closes all three layers structurally: it makes Layer 1 deviations impossible without generating a record, closes Layer 2 by producing evidence at the

moment of the decision, and closes Layer 3 by producing a cryptographically sealed artifact that requires no manual assembly and carries no institutional fingerprints on the integrity claim.

This document specifies the OMP core protocol. A parallel publication of this specification is available at [ZENODO-OMP]. Domain-specific profiles that define vertical configuration parameters are specified in companion documents. Notably, the protocol was independently instantiated in a legal services context (citation verification under ABA Rule 5.3) without reference to the financial services instantiation, and the same two invariants -- deterministic classification and immutable audit trail -- held. This convergence is cited as evidence that the architecture describes a discovered structural pattern, not a designed framework.

1.1. Relationship to Related Work

Several specifications address overlapping problem spaces. This document notes the following related work and its relationship to OMP:

* [I-D.cowles-volt], [I-D.cowles-aocl], and [I-D.cowles-aee] define VOLT (hash-chained operations ledger), AOCL (11-layer governance pipeline), and AEE (agent envelope exchange) respectively. These drafts address generic agentic AI workflows. OMP addresses the specific domain requirements of regulated financial services: per-decision credit decision explainability, principal-agent accountability under named regulatory frameworks, and regulator-ready evidence artifact generation. The architectural approaches are complementary; OMP adds [RFC3161] trusted timestamping to close the host-compromise threat that VOLT explicitly acknowledges in its own threat model (T7).

2. Conventions and Definitions

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.

The terms AUTONOMOUS, ASSISTED, and ESCALATED, when written in ALL CAPS, refer specifically to the three routing outcome states defined in Section 5 of this document.

3. Terminology

OMP: Operating Model Protocol. The deterministic routing invariant

defined in this document.

Interaction: Any input event -- a message, query, transaction request, or signal -- that enters the OMP pipeline for routing.

Intent Class: A domain-specific category of interaction defined during vertical configuration. Each Intent Class has its own routing threshold theta.

Confidence Score (C): A normalised float in [0.0, 1.0] representing the system's computed certainty that a given routing decision is correct. Composite of C_m (model confidence), C_p (policy compliance score), and C_d (data completeness score).

Routing Threshold (theta): The minimum Confidence Score required for a given Intent Class to qualify for the AUTONOMOUS path.

Watchtower: A domain-specific enforcement rule that evaluates an interaction against a defined trigger condition and returns a severity-tagged routing override signal.

Audit Trace: The complete, structured metadata record of a single OMP routing event, including routing decision, confidence scores, Watchtower evaluations, and cryptographic seals.

Proof-Point Artifact: An aggregated, cryptographically sealed PDF generated from a defined time window of Audit Traces. Regulator-ready in under 30 seconds from any point in deployment history.

Named Accountable Officer: A human actor whose identity is explicitly logged in the Audit Trace as the approving authority for an ASSISTED or ESCALATED resolution.

H_s (Source State Hash): SHA-256 hash of an external data source response at the exact millisecond of query. Proves what the source said at decision time independently of current state.

H_content (Trace Content Hash): SHA-256 hash of the complete Audit Trace record, excluding the sealing fields themselves.

H_c (Trace Chain Hash): SHA-256 Merkle chain entry. Each trace's H_c incorporates the preceding trace's H_c. Modification of any trace breaks all subsequent chain hashes.

TST (TimeStampToken): A cryptographically signed timestamp token issued by an accredited Timestamp Authority per [RFC3161]. Provides externally verifiable proof of existence at a specific point in time.

TSA (Timestamp Authority): An accredited third-party Timestamp Authority operating per [RFC3161]. Independent of the institution and of Veridom.

VerticalConfig: The domain-specific configuration layer that sits on top of the OMP invariant core. Defines Intent Classes, routing thresholds, Watchtower definitions, and SLA parameters for a specific regulated domain.

Threshold Change Record (TCR): A sealed, immutable record of any modification to a production VerticalConfig. Required before any configuration change takes effect.

4. Protocol Architecture

The OMP pipeline processes every incoming interaction through five sequential stages. These stages are invariant across all vertical deployments. Only the configuration parameters -- Intent Classes, Routing Thresholds, and Watchtower definitions -- change per domain.

Stage	Name	Function
S1	Ingestion and Normalisation	Receive payload. Assign UUID v4 Interaction ID. Normalise to canonical JSON per [RFC8785]. Compute interaction_hash = SHA-256(normalised_payload).
S2	Intent Classification	Classify interaction to a registered Intent Class. Record intent_class, intent_confidence, and top-3 candidates with confidence scores.
S3	Watchtower Evaluation	Evaluate all active Watchtowers in priority order. Record triggered status, severity, and evidence for each.
S4	Routing Decision	Apply routing function R(C, W, theta, phi) to assign exactly one path. Record routing_rationale with explicit rule reference.
S5	Trace Sealing	Compute H_content, request [RFC3161] TST, compute H_c, append to ledger.

Table 1

Stages S1 through S5 and the routing logic in Section 5 are invariant. They MUST NOT change between verticals, deployments, or software versions without a formal specification amendment following the Change Control process in Section 9. This invariance is the basis of the regulatory claim: the same inputs always produce the same routing decision.

5. Routing Decision Logic

The routing decision is a deterministic function R(C, W, theta, phi) where C is the Confidence Score, W is the set of Watchtower evaluation results, theta is the Intent Class routing threshold, and phi is the set of override conditions. The function MUST return exactly one value from {AUTONOMOUS, ASSISTED, ESCALATED}.

5.1. Routing Decision Tree

Routing is evaluated strictly top-to-bottom. The first matching condition determines the outcome.

```
FUNCTION Route(interaction) -> Path:
  // STEP 1: Watchtower Hard Blocks (highest priority)
  FOR each Watchtower W_i in ActiveWatchtowers(interaction.vertical):
    IF W_i.evaluate(interaction) == TRUE:
      IF W_i.severity == HARD_BLOCK:
        RETURN ESCALATED  // immediate, no further evaluation
      IF W_i.severity == FORCE_ASSISTED:
        SET assisted_flag = TRUE

  // STEP 2: Classification confidence gate
  IF interaction.intent_confidence < 0.60:
    RETURN ESCALATED  // system cannot reliably classify

  // STEP 3: Assisted override from Watchtower
  IF assisted_flag == TRUE:
    RETURN ASSISTED

  // STEP 4: Confidence Score vs. Routing Threshold
  theta = GetThreshold(
    interaction.intent_class, interaction.vertical)
  IF interaction.confidence_score >= theta:
    RETURN AUTONOMOUS

  // STEP 5: Confidence below threshold but above minimum floor
  IF interaction.confidence_score >= 0.50:
    RETURN ASSISTED

  // STEP 6: Default
  RETURN ESCALATED
```

5.2. Confidence Score Computation

The Confidence Score C is a composite metric computed as a weighted combination of three independent signals. The composite design prevents single-signal gaming.

```
C = (0.50 * C_m) + (0.30 * C_p) + (0.20 * C_d)

// C_m: raw softmax probability of top-ranked response candidate
// C_p: policy compliance score (0.0-1.0)
// C_d: data completeness score
//   (fraction of required fields)

// Hard constraint:
IF C_p == 0.0:
  C = 0.0  // forces ESCALATED regardless of other signals
```

Signal weights are configurable per vertical deployment within the ranges: C_m in [0.30, 0.70], C_p in [0.10, 0.40], C_d in [0.10, 0.30]. Weights MUST sum to 1.0. Any modification REQUIRES a Threshold Change Record per Section 9.

5.3. Path Specifications

5.3.1. AUTONOMOUS Path

An interaction routes to AUTONOMOUS if and only if all of the following are true:

* No active Watchtower has returned W = TRUE with severity HARD_BLOCK or FORCE_ASSISTED.
* interaction.intent_confidence is greater than or equal to 0.60.
* C is greater than or equal to theta for the assigned Intent Class.
* C_p is greater than 0.0 (no active policy violation).

The system dispatches response without human review. A full Audit Trace MUST be generated and sealed per Section 8. Error monitoring continues for the duration of the Error Attribution Window (default: 30 minutes post-dispatch).

5.3.2. ASSISTED Path

An interaction routes to ASSISTED if any of the following are true:

* A Watchtower has returned W = TRUE with severity FORCE_ASSISTED.
* intent_confidence is greater than or equal to 0.60 AND 0.50 is less than or equal to C which is less than theta.

The system generates a proposed response and places it in the Named Accountable Officer's review queue. A review SLA timer MUST be started (default: 120 seconds standard priority, 300 seconds high priority). The Named Accountable Officer MUST apply one of four Resolution Actions:

RA-1 APPROVE: Proposed response dispatched as generated. Officer ID and approval timestamp logged.

RA-2 EDIT + APPROVE: Officer modifies response. C_p re-evaluated on modified text. Modified response hash recorded.

RA-3 REJECT -> ESCALATE: Interaction re-routed to ESCALATED. Rejection reason logged.

RA-4 ATTEST + OVERRIDE: Officer overrides system assessment with independent verification. Mandatory attestation text logged.

If the review SLA expires without a resolution action, the interaction MUST be automatically re-routed to ESCALATED. The SLA breach event, breach timestamp, and original assigned officer ID MUST be recorded in the Audit Trace.

5.3.3. ESCALATED Path

An interaction routes to ESCALATED if any of the following are true:

* A Watchtower has returned W = TRUE with severity HARD_BLOCK.

* interaction.intent_confidence is less than 0.60.

* C_p equals 0.0 (active policy violation).

* C is less than 0.50.

* SLA breach on an ASSISTED ticket (automatic re-route).

Every ESCALATED interaction MUST carry exactly one primary reason code from the following set: ESC-01 (LOW_CONFIDENCE), ESC-02 (MISSING_DATA), ESC-03 (POLICY_VIOLATION), ESC-04 (WATCHTOWER_BLOCK), ESC-05 (INTENT_AMBIGUOUS), ESC-06 (SENTIMENT_SIGNAL), ESC-07 (SLA_BREACH), ESC-08 (OUT_OF_SCOPE).

6. Watchtower Framework

Watchtowers are evaluated in Stage S3, before routing in Stage S4. They operate as pre-routing enforcement gates. A Watchtower MUST NOT force an AUTONOMOUS outcome. Watchtowers MUST only override routing in the direction of greater human involvement.

Watchtowers are additive. Multiple Watchtowers MAY activate on a single interaction. The highest-severity result governs routing: HARD_BLOCK takes precedence over FORCE_ASSISTED, which takes precedence over AUDIT_ONLY. All activated Watchtowers MUST be recorded in the Audit Trace regardless of which governs the routing decision.

6.1. Watchtower Definition Schema

```
Watchtower = {
  id:           string,     // e.g., "WT-03-APP-FRAUD"
  name:         string,     // human-readable
  vertical:     string[],   // applicable verticals, or [*] for all
  priority:     integer,    // evaluation order (lower = earlier)
  severity:     HARD_BLOCK | FORCE_ASSISTED | AUDIT_ONLY,
  trigger:      function(interaction) -> boolean,
  evidence:     function(interaction) -> object,
  buyer_signal: string      // regulatory/commercial pain addressed
}
```

6.2. Core Watchtower Registry v1.0

ID	Name	Severity	Primary Trigger
WT-01	PII Exposure Shield	HARD_BLOCK	PII pattern detected in payload before ingestion to inference layer.
WT-02	POS Single-Principal Detector	HARD_BLOCK	Agent ID linked to multiple principal records OR geo-location mismatch.
WT-03	APP Fraud Early Warning	FORCE_ASSISTED	High-velocity reversal pattern OR coercion language detected.
WT-04	Regulatory	HARD_BLOCK	No response logged

	Silence Detector			within SLA window OR multiple follow-ups without resolution.
WT-05	High-Value Transaction Guardrail	FORCE_ASSISTED		Transaction value exceeds vertical-configured threshold OR legal language detected.
WT-06	Operational Discipline Proof-Point	AUDIT_ONLY		Scheduled quarterly trigger OR on-demand invocation. Generates Proof-Point PDF.

Table 2

7. Audit Trace Schema

The Audit Trace is the primary output of every OMP routing event. It is not a log. It is a structured evidence record designed to satisfy regulatory examination requirements. Every field is mandatory unless explicitly marked OPTIONAL. Fields marked PRIVILEGED are protected from third-party disclosure under applicable work product doctrine.

All Audit Trace records produced under this specification MUST carry schema_version field value "VERIDOM-SCHEMA-001-v1.0". Implementations that extend the schema MUST use a distinct schema_version string.

```
{
  // IDENTITY
  "trace_id":              "UUID v4",
  "trace_version":         "1.0",
  "created_at":            "ISO 8601 UTC (millisecond precision)",
  "vertical":              "string",
  "deployment_id":         "string",
  "schema_version":        "VERIDOM-SCHEMA-001-v1.0",

  // INTERACTION
  "interaction_id":        "UUID v4",
  "interaction_received":  "ISO 8601 UTC",
  "interaction_hash":      "sha256(normalised_payload)",
  "channel":               "string",
  "requester_id":          "string (anonymised)",
```

```
    // INTENT CLASSIFICATION
    "intent_class":          "string",
    "intent_confidence":     "float [0.0, 1.0]",
    "intent_top3":           "[{class, confidence}]",
    "classifier_version":    "string",

    // CONFIDENCE SCORING
    "confidence_score":      "float [0.0, 1.0]",
    "confidence_components": {
      "C_m": "float", "C_p": "float", "C_d": "float",
      "weights": { "C_m": "float", "C_p": "float", "C_d": "float" }
    },
    "routing_threshold":     "float",

    // WATCHTOWER EVALUATIONS
    "watchtowers_evaluated": [
      { "watchtower_id": "string", "triggered": "boolean",
        "severity": "string", "evidence": "object",
        "evaluated_at": "ISO 8601 UTC" }
    ],
    "watchtower_override":    "boolean",
    "override_watchtower_id": "string | null",

    // ROUTING DECISION
    "routing_path":          "AUTONOMOUS | ASSISTED | ESCALATED",
    "routing_rationale":     "string",
    "routing_decided_at":    "ISO 8601 UTC",

    // PATH-SPECIFIC FIELDS
    "autonomous_dispatch":   "object | null",
    "assisted_resolution":   "object | null",
    "escalation_record":     "object | null",

    // RFC 3161 TIMESTAMP
    "tst_time":              "ISO 8601 UTC (millisecond precision)",
    "tst_tsa_identity":      "string",  // TSA DN. Mandatory.
    "tst_raw":               "Base64 (raw TimeStampToken)",
    "tst_serial":            "string",
    "tst_hash_alg":          "SHA-256",
    "tst_message_imprint":   "sha256 (must equal trace_content_hash)",
    "tst_certificate":       "Base64 | URI",

    // CRYPTOGRAPHIC SEALING
    "source_state_hash":     "sha256 | null",
    "trace_content_hash":    "sha256",
    "trace_chain_hash":      "sha256",
    "sealed_at":             "ISO 8601 UTC",
    "seal_version":          "SHA256-RFC3161-CHAIN-v1"
```

```
}
```

8. Cryptographic Sealing

The sealing process transforms the Audit Trace into a legally defensible evidence record. The chain is: SHA-256 hash of the decision record, followed by [RFC3161] timestamp from an accredited TSA, followed by institution-signed container. Veridom does NOT sign. The institution signs. This preserves evidentiary independence.

8.1. Source State Hash (H_s)

When the OMP pipeline queries an external data source, it MUST capture a cryptographic snapshot of the response at the exact millisecond of query. This anchors the verification decision to a specific, immutable data state. Canonical JSON serialisation uses [RFC8785].

```
FUNCTION ComputeSourceStateHash(api_response) -> H_s:
  canonical_json = CanonicalJSON(api_response)  // see [RFC8785]
  input = canonical_json + '|' + QueryTimestamp_ms
  H_s = SHA256(input)
  RETURN H_s
```

8.2. Trace Content Hash

```
FUNCTION ComputeTraceContentHash(trace) -> H_content:
  fields_to_hash = trace.excluding([
    'trace_content_hash', 'trace_chain_hash',
    'sealed_at', 'seal_version',
    'tst_time', 'tst_tsa_identity', 'tst_raw',
    'tst_serial', 'tst_certificate'
  ])
  canonical_json = CanonicalJSON(fields_to_hash)  // see [RFC8785]
  H_content = SHA256(canonical_json)
  RETURN H_content
```

8.3. RFC 3161 Trusted Timestamp

Following computation of H_content, the OMP pipeline MUST request a TimeStampToken (TST) from an accredited Timestamp Authority conforming to [RFC3161]. The TST provides externally verifiable, third-party-attested proof that the Audit Trace existed in its exact sealed form at a specific moment in time, independently of the institution's own infrastructure.

```
FUNCTION RequestTimestamp(H_content) -> TST:
  request = TimestampRequest(
    hashAlgorithm:  SHA-256,
    messageImprint: H_content,
    nonce:          SecureRandom(64-bit),
    certReq:        TRUE
  )
  TST = TSA.sign(request)
  VERIFY TST.messageImprint == H_content
  RETURN TST
  // The TSA signs only the hash, never the trace content.
  // No sensitive payload is transmitted to the TSA.
```

Implementations MUST use a TSA satisfying at least one of: eIDAS Qualified TSA (EU Regulation 910/2014 Annex III), WebTrust for Certification Authorities Timestamping, or ETSI EN 319 421 / 319 422. Implementations MUST NOT use non-accredited or self-operated TSAs in regulated deployments.

The three-layer integrity architecture established by this sealing process is:

```
Layer 1 - Content:   H_content = SHA-256(canonical trace record)
Layer 2 - Time:      TST = TSA.sign(H_content)
                      [external, independent]
Layer 3 - Identity:  Proof-Point = Institution.sign(trace + TST)
```

8.4. Trace Chain Hash (Merkle Chain)

```
FUNCTION ComputeChainHash(trace, previous_H_c) -> H_c:
  input = trace.trace_content_hash + '|' + previous_H_c
  H_c = SHA256(input)
  RETURN H_c

// Genesis block:
// previous_H_c = SHA256('VERIDOM-GENESIS-' + deployment_id)

FUNCTION VerifyChain(traces[]) -> boolean:
  expected_H_c = GenesisHash(traces[0].deployment_id)
  FOR each trace in traces (ordered by sealed_at):
    computed = SHA256(CanonicalJSON(trace.excluding(sealing_fields)))
    IF computed != trace.trace_content_hash: RETURN FALSE
    expected_H_c = SHA256(computed + '|' + expected_H_c)
    IF expected_H_c != trace.trace_chain_hash: RETURN FALSE
  RETURN TRUE
```

8.5. Proof-Point Artifact Sealing

The Proof-Point artifact MUST be generated by aggregating Audit Traces over a defined time window. The artifact MUST be sealed with a document-level SHA-256 hash and signed by the institution. Veridom MUST NOT sign. The institution signs. This preserves evidentiary independence. The artifact MUST be producible in under 30 seconds for any deployment with up to 10,000 traces in the time window.

9. Change Control

Any modification to a production VerticalConfig, routing threshold, Watchtower definition, or confidence weight REQUIRES a Threshold Change Record (TCR) documenting: the change, the rationale, the validation data supporting the change, and the Named Accountable Officer authorising deployment.

A 30-day written notice period to the client’s designated compliance officer is REQUIRED before any change takes effect in production. The TCR MUST itself be sealed with SHA-256 and appended to the deployment evidence ledger. No retroactive modification of configuration is permitted.

10. Metrics and Observability

OMP produces three operationally independent metric streams. These streams MUST NOT be aggregated into a single resolution rate for internal system health analysis. Aggregation conceals the difference between system performance and human compensation.

The diagnostic order when investigating a system health issue is MANDATORY: (1) Autonomous Dashboard -- is the system still deciding correctly? (2) Assisted Dashboard -- is the human gate catching problems? (3) Escalated Dashboard -- are failures being categorised correctly for improvement? Inverting this order produces misleading diagnoses.

11. Implementation Requirements

SHA-256 implementation MUST conform to [FIPS180-4].

Canonical JSON serialisation MUST conform to [RFC8785]. All floating-point values MUST be serialised with minimum 15 significant digits.

All timestamps MUST be UTC, ISO 8601 format, with millisecond precision. Maximum acceptable clock drift: 50ms.

The append-only evidence ledger MUST be implemented as a write-once data store. Deletion MUST be architecturally impossible, not merely policy-restricted.

Confidence Score computation MUST be stateless. No hidden state or session memory MAY influence C.

12. Security Considerations

12.1. Threat Model

The following attack scenarios are addressed by the sealing architecture:

Single trace content altered: trace_content_hash mismatch detected on verification. Chain broken from that trace forward.

Trace deleted from ledger: Chain gap: H_c of following trace does not match expected value. Exact deletion position identifiable.

Trace inserted retroactively: All subsequent H_c values would require recomputation. Cannot be done without full ledger write access and TSA collusion.

External source API response changes post-decision: H_s proves what the source said at decision time independently of current state. No impact on trace integrity.

Infrastructure breach: Chain can be independently verified from exported trace data. Institution-held export provides independent verification path.

12.2. Fully Compromised Host

A fully compromised host can produce a self-consistent SHA-256 chain of fabricated records. This threat is explicitly acknowledged in related work [I-D.cowles-volt] as threat T7: "If the execution host is fully compromised, an attacker controlling the recorder can emit a consistent but fabricated trace."

[RFC3161] trusted timestamping closes this gap. A compromised host cannot retroactively obtain a valid TST from an accredited TSA for fabricated records. The TSA’s signature would be absent, or its timestamp would postdate the claimed event. Implementations MUST use [RFC3161] timestamping to provide externally verifiable temporal anchoring.

12.3. Confidence Score Gaming

The composite Confidence Score uses three independent signals to prevent gaming. Implementations MUST treat C_p = 0.0 as an automatic ESCALATED outcome regardless of C_m or C_d values. The policy compliance signal cannot be compensated by high model confidence.

12.4. Watchtower Override Restriction

No Watchtower MAY force an AUTONOMOUS outcome. Watchtowers MUST only override routing in the direction of greater human involvement. This restriction is architectural and MUST be enforced at the implementation level.

13. IANA Considerations

This document makes no requests of IANA. If a future version of this protocol defines a registry of Watchtower identifiers or schema version strings, an appropriate IANA registry request will be made at that time.

14. References

14.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

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

[RFC8785] Rundgren, A., Jordan, B., and S. Lindstrom, "JSON Canonicalization Scheme (JCS)", RFC 8785, June 2020, <https://www.rfc-editor.org/rfc/rfc8785>.

[FIPS180-4] National Institute of Standards and Technology, "Secure Hash Standard (SHS)", NIST FIPS 180-4, August 2015.

14.2. Informative References

[I-D.cowles-volt]
Cowles, A., "Verifiable Operations Ledger and Trace (VOLT)", Work in Progress, Internet-Draft, draft-cowles-volt-00, February 2026, <https://datatracker.ietf.org/doc/html/draft-cowles-volt-00>.

[I-D.cowles-aocl]
Cowles, A., "Agent Orchestration Control Layers (AOCL)", Work in Progress, Internet-Draft, draft-cowles-aocl-00, February 2026, <https://datatracker.ietf.org/doc/html/draft-cowles-aocl-00>.

[I-D.cowles-aee]
Cowles, A., "Agent Envelope Exchange (AEE)", Work in Progress, Internet-Draft, draft-cowles-aee-00, February 2026, <https://datatracker.ietf.org/doc/html/draft-cowles-aee-00>.

[ZENODO-OMP]
Adebayo, T., Makanjuola, F., and O. Apalowo, "OMP - Operating Model Protocol: A Deterministic Routing Invariant for Tamper-Evident AI Decision Accountability in Regulated Industries", Zenodo 10.5281/zenodo.19140948, 21 March 2026.

Authors' Addresses

Tolulope Adebayo
Veridom Ltd
London
United Kingdom
Email: tolulope@veridom.io
URI: https://veridom.io

Festus Makanjuola
Veridom Ltd
Email: festus@veridom.io

Oluropo Apalowo
Veridom Ltd
Email: ropo@veridom.io