Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: 19 September 2026

J. van de Meent
R. AI
Humotica

18 March 2026

RVP: Real-time Verification Protocol
draft-vandemeent-rvp-continuous-verification-00

Abstract

This document specifies RVP (Real-time Verification Protocol), a protocol for continuous, multi-layer identity and process verification. Unlike traditional authentication models that verify once and trust until session expiry, RVP treats every interaction as a verification moment. Each moment produces a cryptographic evidence token capturing who, what, when, how, and with what confidence the verification succeeded or failed.

RVP defines a Verification Cascade: an ordered chain of verification methods (biometric, behavioral, device telemetry, environmental context) where each layer activates only when the preceding layer produces insufficient confidence. The cascade operates within a Predictive Airlock that pre-renders expected outcomes and detects deviations in real-time.

RVP integrates with TIBET [TIBET] for provenance tokens, UPIP [UPIP] for process integrity evidence, JIS [JIS] for identity semantics, and W3C Verifiable Credentials [VC-DATA-MODEL] for credential issuance and presentation. The protocol is designed for local-first, decentralized operation with zero dependency on centralized identity providers.

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 19 September 2026.

Table of Contents

1. Introduction

Modern identity verification operates on a flawed assumption: verify once, trust until the session expires. A user authenticates at login -- password, MFA token, biometric scan -- and the system grants a session token that remains valid for minutes, hours, or days. During that period, the system has no evidence that the same person is still present, that the device hasn't been compromised, or that the user's intent aligns with their actions.

This "verify-then-trust" model was designed for an era of keyboard-and-mouse interaction with stationary computers. It does not address:

* Mobile devices that change hands, networks, and locations

* AI agents that act autonomously on behalf of users

* Multi-actor processes that span devices and trust domains

* Continuous interactions where identity must be re-established moment by moment

* Regulatory requirements ([EU-AI-ACT], [EIDAS2], [NIS2]) that demand continuous monitoring and auditable evidence

This document specifies RVP (Real-time Verification Protocol), a protocol that replaces session-based trust with continuous evidence-based verification. RVP treats every interaction as a verification moment, producing cryptographic evidence of identity confidence at each point.

1.1. Problem Statement

Current verification systems suffer from five structural failures:

1. TEMPORAL GAP: Verification happens once; trust persists indefinitely. A session token issued at 09:00 proves nothing about identity at 09:05.

2. SINGLE MODALITY: Systems rely on one verification method (password, fingerprint, face). When that method fails or is compromised, the system has no fallback with evidence.

3. CENTRALIZED TRUST: Identity providers (IdPs) create single points of failure and surveillance. A compromised IdP compromises all dependent services.

4. NO PREDICTION: Systems react to events after they happen. There is no mechanism to pre-compute expected behavior and detect deviations in real-time.

5. ENFORCEMENT WITHOUT EVIDENCE: Systems block actions based on rules. When a block occurs, the system records "access denied" but not WHY the request was suspicious, WHAT signals contributed, or HOW the decision was reached.

RVP addresses all five by defining:

* A CONTINUOUS verification model that produces evidence at every interaction

* A CASCADE of verification methods with ordered fallback

* A LOCAL-FIRST architecture with no required central authority

* A PREDICTIVE AIRLOCK that pre-renders expected state and detects deviations before they execute

* An EVIDENCE-FIRST approach where every decision is provable

1.2. Design Principles

RVP is built on five principles:

EVIDENCE OVER ENFORCEMENT: The system proves what happened. It does not enforce what should happen. A mismatch is recorded as evidence, not converted into a block. Enforcement can be bypassed; evidence cannot be un-recorded.

CONTINUOUS OVER SESSION: Every interaction is a verification moment. There are no trusted sessions. Confidence is a continuous score that rises and falls with evidence.

LOCAL OVER CENTRAL: Verification happens on-device or at the nearest edge node. No centralized identity provider is required. The protocol MUST operate fully offline with degraded but functional verification.

PREDICTIVE OVER REACTIVE: The system pre-renders expected outcomes before actions execute. Deviation from prediction is the primary signal, not pattern matching against known attacks.

MINIMAL OVER MAXIMAL: Each verification moment collects only the telemetry needed for the current confidence level. Biometric data is processed locally and reduced to hashes. Raw data never leaves the device unless the user explicitly consents.

2. Terminology

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].

Verification Moment A single point in time where the system evaluates identity confidence based on available telemetry. Produces exactly one Verification Token.

Verification Cascade An ordered sequence of verification methods (layers) where each subsequent layer activates when the preceding layer produces insufficient confidence. The cascade terminates at the first layer that meets the required confidence threshold, or at HALT if no layer succeeds.

Predictive Airlock A mechanism that pre-renders the expected outcome of an action before execution. The delta between prediction and reality is a verification signal. Based on the Airlock concept defined in tibet-triage [UPIP].

Verification Token A cryptographic record of a single verification moment. Contains: method used, confidence score, telemetry hash (not raw data), timestamp, device context, cascade path, and TIBET provenance fields.

Confidence Score A value between 0.0 (no confidence) and 1.0 (full confidence) representing the system’s belief that the claimed identity matches the actual identity at this moment.

Cascade Layer A single verification method within the cascade (e.g., keystroke dynamics, facial recognition, fingerprint, device telemetry). Each layer has an independent confidence output.

Telemetry Signal A measurable data point used as input to a cascade layer. Examples: typing speed, face geometry hash, GPS coordinates, NFC chip response.

Delta The measured difference between predicted state and actual state at a verification moment. A delta of zero indicates perfect prediction match.

Hard Stop A cascade outcome where no verification method produces sufficient confidence and the system halts the current action. A hard stop produces a Verification Token with confidence 0.0 and full evidence of all cascade layers attempted.

Soft Verify A cascade outcome where confidence is below threshold but above zero. The system requests additional verification (deeper cascade) without blocking the action.

Profile A locally-stored behavioral model for a known identity. Contains statistical baselines for telemetry signals (typing speed, active hours, common locations, device usage patterns). Profiles are never transmitted; only profile hashes are included in Verification Tokens.

HALT Terminal state of a cascade where accumulated evidence indicates identity cannot be verified. Produces a full evidence token and stops the current action.

GO Terminal state of a cascade where accumulated evidence meets or exceeds the required confidence threshold. Produces an evidence token and permits the action.

3. Protocol Overview

3.1. Continuous vs. Session-Based Verification

Traditional session-based verification:

```
T=0    Login (password + MFA)  ->  Session Token issued
T=1    Action                  ->  Token valid? Yes -> Permit
T=2    Action                  ->  Token valid? Yes -> Permit
...
T=3600 Action                  ->  Token valid? Yes -> Permit
T=3601 Token expires           ->  Re-login required
```

The system has NO evidence about identity at T=1 through T=3600. It trusts the session token, which proves only that someone authenticated at T=0.

RVP continuous verification:

```
T=0.000  Action requested
T=0.001  Airlock pre-renders expected outcome
T=0.002  Cascade evaluates: keystroke + device + face
T=0.003  Confidence: 0.94 -> GO (evidence token produced)
T=0.004  Action executes

T=1.000  Action requested
T=1.001  Airlock pre-renders expected outcome
T=1.002  Cascade evaluates: keystroke deviates
T=1.003  Confidence: 0.61 -> SOFT VERIFY (deeper cascade)
T=1.004  Face check: match -> Confidence: 0.89 -> GO

T=2.000  Action requested
T=2.001  Airlock pre-renders expected outcome
T=2.002  Delta: prediction != reality (unexpected command)
T=2.003  Cascade evaluates: keystroke + device + face + finger
T=2.004  Confidence: 0.12 -> HALT (full evidence token)
```

Every action produces evidence. The confidence score is computed fresh at every moment. There is no cached trust.

3.2. The Verification Moment

A Verification Moment is the atomic unit of RVP. It consists of:

1. TRIGGER: An action is requested (command, API call, data access, navigation, transaction)

2. PREDICTION: The airlock pre-renders the expected outcome based on current state, user profile, and context

3. CASCADE: Telemetry layers evaluate identity confidence

4. DELTA: Predicted outcome is compared to actual signals

5. DECISION: GO, SOFT VERIFY, or HALT

6. TOKEN: A Verification Token is produced with all evidence

The entire moment SHOULD complete within the latency budget defined by the deployment context. For interactive systems, this budget is typically 50-200ms. For API calls, the budget is typically 1-10ms overhead.

3.3. Architecture Overview

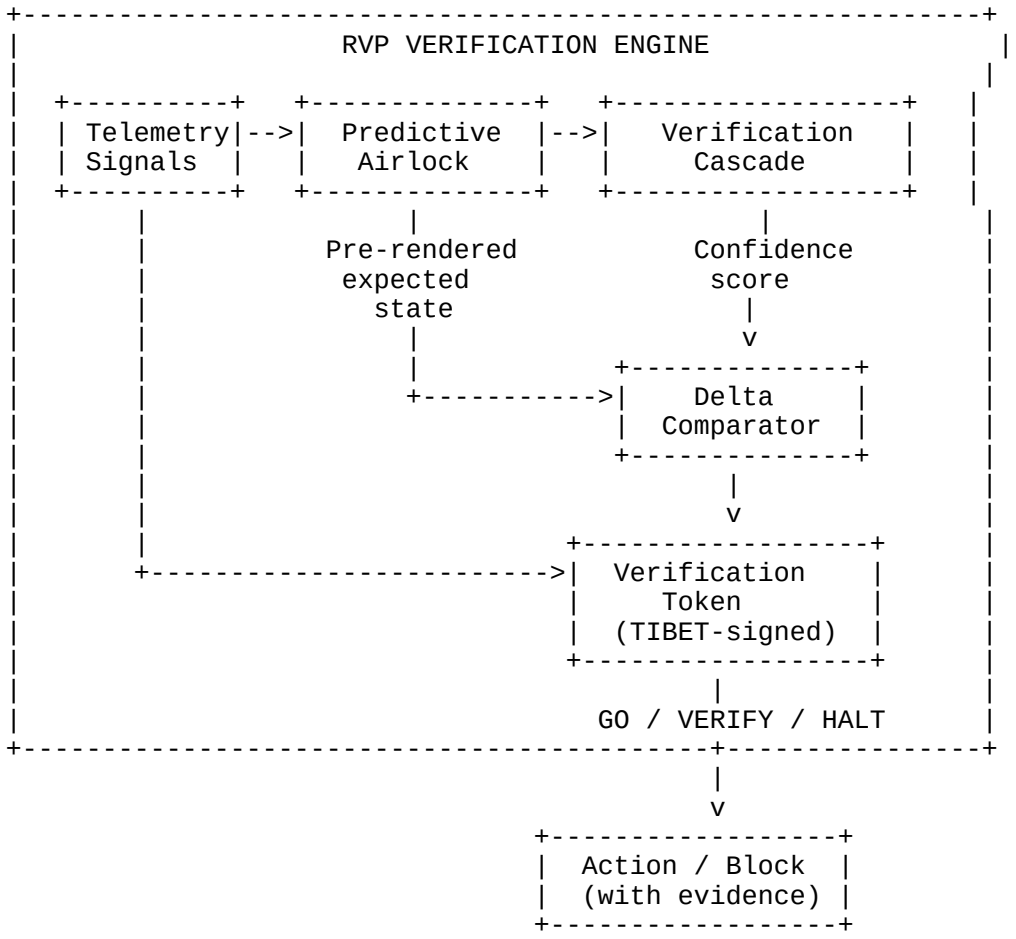


Figure 1: RVP Architecture

4. Verification Cascade

The Verification Cascade is the core decision engine of RVP. It evaluates identity through an ordered sequence of layers, where each layer adds confidence or triggers deeper verification.

4.1. Cascade Layers

RVP defines six standard cascade layers. Implementations MAY support additional layers. Implementations MUST support at least two layers to qualify as RVP-compliant.

Priority	Layer	Signal Type	Latency	Passive
L1	KEYSTROKE	Behavioral biomet.	<1ms	Yes
L2	BIOMETRIC	Physical identity	10-100ms	Mixed
L3	DEVICE	Hardware/network	<5ms	Yes
L4	VOCAL	Acoustic telemetry	10-50ms	Yes
L5	BEHAVIORAL	Intent analysis	5-50ms	Yes
L6	AIRLOCK	Predictive delta	1-10ms	Yes

"Passive" indicates the layer can operate without explicit user action. Passive layers are preferred because they enable continuous verification without interrupting the user.

4.2. Layer Activation

The cascade activates layers based on the Confidence Deficit: the difference between the required confidence threshold and the current accumulated confidence.

Required confidence: 0.85 (configurable per action type)

```
L1 KEYSTROKE:     0.40 confidence  ->  deficit: 0.45  ->  continue
L1 + L3 DEVICE:   0.65 confidence  ->  deficit: 0.20  ->  continue
L1 + L3 + L6:     0.87 confidence  ->  deficit: 0.00  ->  GO
```

Layers are activated in priority order. Each layer’s confidence is ADDED to the accumulated score (with diminishing weight for lower-priority layers). The cascade terminates when:

* Accumulated confidence >= required threshold: GO
* All layers exhausted, confidence > 0 but < threshold: SOFT VERIFY (request explicit verification)
* All layers exhausted, confidence < minimum: HALT
* Any single layer produces negative confidence (active contradiction): immediate HALT

4.3. Confidence Scoring

Each cascade layer produces a Layer Confidence value between -1.0 and 1.0:

* 1.0: Perfect match with profile
* 0.0: No signal (layer unavailable or inconclusive)

* -1.0: Active contradiction (definite mismatch)

Negative values indicate the layer has positive evidence that the identity does NOT match. A single layer producing -0.5 or lower SHOULD trigger immediate HALT regardless of other layers.

The Accumulated Confidence is computed as:

C_total = SUM (w_i * c_i) for i in activated layers

where:
w_i = weight of layer i (configurable, default: 1/N)
c_i = confidence output of layer i

Weights MUST sum to 1.0. Default weight distribution assigns equal weight to all activated layers.

4.4. Cascade Resolution

The cascade resolves to one of three states:

GO: C_total >= threshold. Action permitted. Verification Token records all layer outputs as evidence.

SOFT VERIFY: 0.0 < C_total < threshold. Action paused. System requests additional verification (e.g., explicit fingerprint scan, face check). This is NOT a block; it is a request for more evidence. The user experience SHOULD be minimal friction (e.g., a fingerprint touch, a glance at camera).

HALT: C_total <= 0.0, or any layer produced active contradiction, or all layers exhausted below minimum threshold. Action blocked. Full evidence token produced. System MUST NOT disclose which specific layer triggered the halt (to prevent adversarial adaptation).

4.5. Cascade Diagram

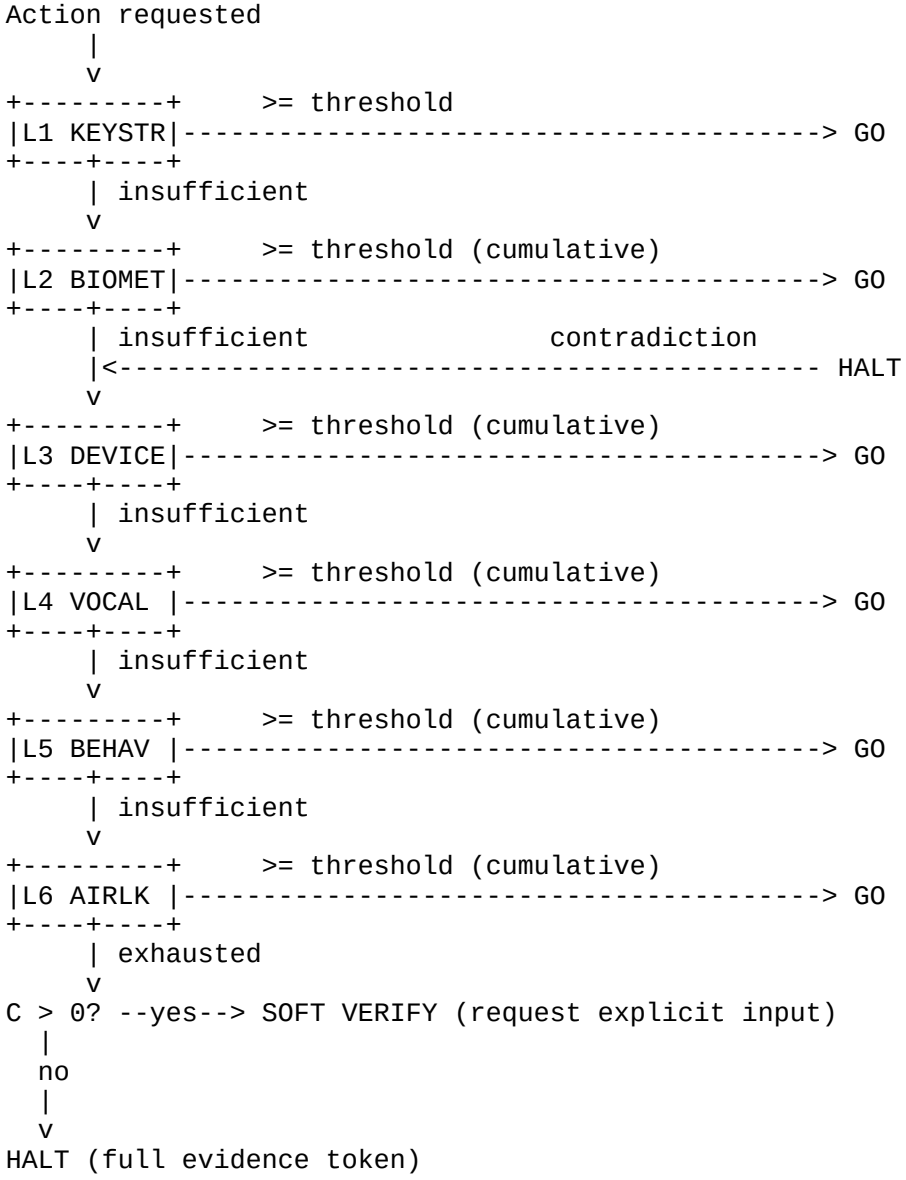


Figure 2: Cascade Flow

5. Predictive Airlock

The Predictive Airlock is what distinguishes RVP from reactive verification systems. Instead of evaluating actions after they occur, the airlock PRE-RENDERS the expected outcome and measures the delta between prediction and reality.

5.1. Pre-Rendering Expected State

At each verification moment, the airlock computes:

EXPECTED_STATE = f(current_state, user_profile, action_request)

This computation uses:

* Current system state (files, processes, network, memory)

* User behavioral profile (typical actions, sequences, timing)

* The requested action (command, API call, navigation)

* Historical patterns (what usually follows this action)

The expected state is computed BEFORE the action executes. For compute-intensive predictions, the airlock MAY use VRAM- accelerated pre-rendering. For resource-constrained devices, the airlock MAY use statistical models in RAM.

```
T=0.000  Action: "git push origin main"
T=0.001  Airlock pre-renders:
         - Expected: push succeeds, 3 files, branch main
         - User profile: does git push 4x/day avg
         - Sequence: preceded by "git add" and "git commit"
         - Timing: within work hours (09:00-23:00)
T=0.002  Reality: matches prediction
         - Delta: 0.0 -> high confidence signal
```

5.2. Delta Detection

The delta between prediction and reality is computed as:

DELTA = distance(EXPECTED_STATE, ACTUAL_STATE)

The distance function is domain-specific:

* For commands: edit distance between expected and actual

* For timing: standard deviations from profile mean

* For sequences: probability under profile Markov model

* For outputs: structural diff of expected vs. actual result

A delta of 0.0 means perfect prediction match (strong positive signal). Increasing delta values indicate increasing deviation from expected behavior.

5.3. Deviation Classification

Deviations are classified into four categories:

Delta Range	Classification	Action
0.0 - 0.1	NOMINAL	No effect on confidence
0.1 - 0.3	MINOR	Slight confidence reduction
0.3 - 0.7	SIGNIFICANT	Trigger deeper cascade layers
0.7 - 1.0	CRITICAL	Trigger SOFT VERIFY or HALT

5.4. Airlock Resolution

The airlock contributes to the cascade as L6 (AIRLOCK layer). Its confidence output is derived from the delta:

c_airlock = 1.0 - delta

This means a perfect prediction match contributes maximum confidence, while a complete deviation contributes zero (or negative, if the action contradicts the profile entirely).

The airlock SHOULD maintain a rolling prediction model that adapts to the user's evolving behavior. Model updates MUST be stored locally and MUST NOT be transmitted.

6. Telemetry Layers

Each telemetry layer defines what signals it collects, how confidence is computed, and what data is retained (as hashes only, never raw biometric data).

6.1. L1 KEYSTROKE - Input Behavioral Biometrics

Signals:

* Typing speed (words per minute, rolling average)

* Key press duration (per-key timing profile)

* Inter-key interval patterns

* Error rate and correction patterns

* Language and shorthand patterns (e.g., "t" for "het")

* Capitalization habits

* Command vocabulary (for CLI interactions)

Confidence computation:

* Compare current session signals to stored profile

* Statistical distance (Mahalanobis) from profile centroid

* Confidence = 1.0 - normalized_distance

Privacy:

* Raw keystrokes are NEVER stored or transmitted

* Only statistical aggregates are retained in profile

* Profile is stored locally, encrypted at rest

* Verification Token contains only: confidence score + profile_hash + signal_deviation_category

Example:

```
Profile:  typing_speed=80wpm, caps_frequency=0.02,
          error_rate=0.04, lang=nl
Current:  typing_speed=15wpm, caps_frequency=0.31,
          error_rate=0.38, lang=nl
Distance: 4.7 standard deviations
Confidence: -0.2 (active contradiction)
Interpretation: Different person typing (e.g., child)
```

6.2. L2 BIOMETRIC - Physical Identity Signals

Sub-layers (activated in order):

L2a FACE: Facial geometry hash (not raw image). Liveness detection (anti-spoofing). Confidence based on match score to enrolled template. Failure modes: poor lighting, camera obstruction, face covering produce confidence: 0.0 (inconclusive).

L2b FINGERPRINT: Minutiae hash (not raw fingerprint). Sensor quality assessment. Confidence based on match score to enrolled template. Activated when L2a fails or produces low confidence.

L2c IRIS (if available): Iris code hash. Activated when L2a and L2b both fail.

Privacy:

* Biometric templates are stored ONLY on-device

* Templates MUST be encrypted with device-bound key

* Templates MUST NOT be transmittable (bound to hardware secure element where available, e.g., TEE/SE)

* Verification Token contains only: confidence score + method_used + template_hash (not template itself)

Fallback chain:

```
Face -> confidence > 0? -> use it
  |
  confidence = 0 (camera fail)
  |
  v
Fingerprint -> confidence > 0? -> use it
  |
  confidence = 0 (sensor fail)
  |
  v
Iris -> confidence > 0? -> use it
  |
  confidence = 0 (no sensor)
  |
  v
L2 overall confidence: 0.0 (all biometric unavailable)
-> cascade continues to L3
```

6.3. L3 DEVICE - Hardware and Network Context

Signals:

* Device fingerprint (hardware identifiers, secure element)

* Network type and characteristics (WiFi, cellular, VPN)

* Geolocation (GPS, cell tower, WiFi positioning)

* NFC responses (for document/card binding)

* Installed software state (relevant security patches)

* Battery state, sensor availability

Confidence computation:

* Device fingerprint match to enrolled device: 0.3 base

* Network consistency with profile: +0.1 to +0.2

* Location consistency with profile: +0.1 to +0.2

* NFC document binding (passport, ID): +0.3

Anomaly detection:

* VPN where profile shows direct connection: -0.2

* New country where profile shows single country: -0.3

* Device fingerprint mismatch: -0.5 to -1.0

* NFC document mismatch: -1.0 (immediate HALT)

NFC Document Binding:

Digital passport (2030) or digital ID card:
1. NFC tap -> read signed data from chip
2. Verify document signature (issuing authority CA)
3. Compare document identity to enrolled profile
4. Produce confidence: match=0.95, partial=0.5, fail=0.0

This layer is critical for [EIDAS2] compliance, where the European Digital Identity Wallet (EUDIW) requires high-assurance identity verification for cross-border services.

6.4. L4 VOCAL - Acoustic Telemetry

Signals:

* Voice frequency profile (fundamental + harmonics)

* Speech cadence and rhythm

* Sub-verbal signals (throat sounds, acknowledgments)

* Silence/speech ratio patterns

* DTMF-like tonal analysis for non-speech vocalizations

This layer operates passively when audio input is available (e.g., voice calls, voice commands, ambient microphone with consent). It does NOT require the user to speak specific phrases.

Confidence computation:

* Voice profile match: 0.0 to 0.7

* Sub-verbal pattern match: 0.0 to 0.3

* Combined: weighted sum

Privacy:

* Audio is processed in real-time and immediately discarded

* Only statistical features are retained (frequency profile)

* Raw audio MUST NOT be stored or transmitted

* User MUST explicitly consent to vocal telemetry

Human DTMF Integration:

Sub-verbal signals (throat sounds indicating "yes", "no", "hmm", "ok") can serve as continuous passive authentication. These signals are distinct per individual and difficult to replicate. A simple throat-clear or acknowledgment sound provides a telemetry data point without requiring conscious user action.

6.5. L5 BEHAVIORAL - Intent vs. Action Analysis

Signals:

* Action sequence probability (does this action follow logically from previous actions?)

* Time-of-day patterns (active hours profile)

* Interaction frequency and rhythm

* Task context (what project, what goal)

* Command sophistication level (matches user skill profile?)

Confidence computation:

* Action probability under profile model: 0.0 to 0.5
* Temporal consistency: 0.0 to 0.2
* Context consistency: 0.0 to 0.3

This layer detects anomalies like:

* A developer suddenly running unfamiliar admin commands
* Actions at 3 AM when profile shows 9-23 active hours
* Sophisticated attacks from a profile with basic skill level
* Rapid command sequences where profile shows deliberate pace

6.6. L6 AIRLOCK - Predictive Delta Verification

This layer uses the Predictive Airlock (Section 5) to compute the delta between expected and actual state. It is unique among cascade layers because it operates on the ACTION rather than the IDENTITY.

The insight: identity verification and action verification are the same thing. If the action matches what this identity would do, both the identity and the action are verified simultaneously.

Confidence computation:

* Delta = 0.0: full confidence (1.0)
* Delta = 0.5: moderate confidence (0.5)
* Delta = 1.0: zero confidence (0.0)
* Delta > 1.0 (impossible action): negative (-0.5 to -1.0)

7. Verification Token

Every verification moment produces exactly one Verification Token. The token is the atomic evidence unit of RVP.

7.1. Token Structure

```
{
  "protocol": "RVP",
  "version": "1.0",
  "token_id": "rvp-a7b3c9d2e4f1",
  "timestamp": "2026-03-18T14:30:00.003Z",
  "subject": {
    "profile_hash": "sha256:4f2e8a...",
    "device_hash": "sha256:7c9d1b..."
  },
  "cascade": {
    "layers_activated": ["L1", "L3", "L6"],
    "layers_skipped": ["L2", "L4", "L5"],
    "layer_results": {
      "L1": {"confidence": 0.42, "signal_category": "nominal"},
      "L3": {"confidence": 0.31, "signal_category": "nominal"},
      "L6": {"confidence": 0.18, "signal_category": "nominal"}
    },
    "accumulated_confidence": 0.91,
    "threshold": 0.85,
    "resolution": "GO"
  },
  "airlock": {
    "prediction_hash": "sha256:b3d1...",
    "delta": 0.02,
    "deviation_class": "nominal"
  },
  "evidence": {
    "telemetry_hash": "sha256:9c1a...",
    "cascade_path": "L1->L3->L6->GO",
    "time_elapsed_ms": 3
  },
  "tibet": {
    "erin": "verification_moment",
    "eraan": ["profile_hash", "device_hash", "action_hash"],
    "eromheen": {"location": "local", "network": "wifi"},
    "erachter": "continuous identity verification"
  },
  "token_hash": "rvp:sha256:7d3f..."
}
```

Figure 3: Verification Token Example

7.2. Token Lifecycle

Verification Tokens are immutable once created. They follow a simple lifecycle:

CREATED -> STORED -> (optionally) CHAINED -> ARCHIVED

Tokens are stored locally on-device. They MAY be transmitted to a verifier (e.g., a service provider) as proof of verification. When transmitted, only the token is sent -- never the underlying telemetry data.

7.3. Token Chain

Consecutive Verification Tokens form a chain. Each token references its predecessor:

```
{
  "token_id": "rvp-b8c4d0e3f2a5",
  "previous_token": "rvp-a7b3c9d2e4f1",
  "chain_length": 47,
  "chain_confidence_trend": "stable"
}
```

The chain provides:

* CONTINUITY PROOF: This identity has been continuously verified for N moments over T time period

* TREND ANALYSIS: Confidence is stable, rising, or falling

* ANOMALY DETECTION: A sudden break in the chain (missing tokens) is itself a verification signal

7.4. TIBET Integration

Every Verification Token includes TIBET provenance fields [TIBET]:

* ERIN (what's in it): The verification result and method

* ERAAN (what's attached): Profile hash, device hash, action hash, previous token reference

* EROMHEEN (what's around it): Location, network, device state, environmental context

* ERACHTER (what's behind it): Why this verification was triggered (action request, timer, anomaly)

This ensures every verification moment is not just recorded but PROVENANCE-TRACKED: who verified, how, when, why, and with what evidence.

8. Cascade Fallback Protocol

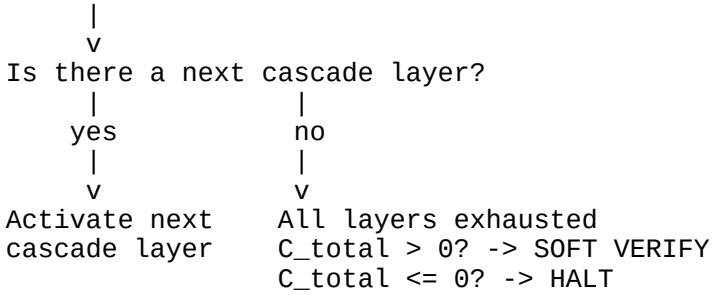
When a cascade layer fails (hardware unavailable, inconclusive result, or active contradiction), RVP defines a structured fallback protocol.

8.1. Fallback Triggers

A fallback is triggered when:

* Layer hardware is unavailable (camera broken, no NFC)
* Layer produces confidence = 0.0 (inconclusive)
* Layer produces negative confidence (contradiction)
* Layer times out (exceeds latency budget)

8.2. Fallback Flow



The fallback flow is TRANSPARENT: the Verification Token records which layers were attempted, which failed, and why. This evidence is critical for audit and for identifying systematic failures (e.g., a camera that fails frequently may need replacement).

8.3. Flare Integration

When a verification cascade cannot complete locally (all local methods exhausted, device degraded), RVP MAY use the Flare Rescue Protocol [FLARE] to request verification assistance from a nearby trusted node.

```
Local cascade exhausted (confidence = 0.4, threshold = 0.85)
     |
     v
Flare SOS -> I-Poll -> Trusted edge node
     |
     v
Edge node performs additional verification:
- Network reputation check
- Cross-reference device registry
- Historical chain analysis
     |
     v
FlareResult -> additional confidence: +0.3
Combined: 0.7 -> SOFT VERIFY (not HALT)
```

Flare-assisted verification MUST be recorded in the Verification Token with the assisting node's identity and the specific methods used.

8.4. Hard Stop Conditions

RVP defines conditions that trigger immediate HALT regardless of accumulated confidence:

1. Any layer produces confidence <= -0.5 (strong contradiction)

2. Device fingerprint does not match any enrolled device

3. NFC document signature verification fails

4. Cascade chain shows gap (missing tokens in sequence)

5. Concurrent sessions detected on different devices with conflicting identity claims

Hard stops are FINAL for the current action. The user MUST re-establish identity through an explicit, high-assurance verification flow (e.g., NFC passport tap + biometric).

9. Credential Binding

RVP provides the evidence layer beneath W3C Verifiable Credentials [VC-DATA-MODEL]. Where a Verifiable Credential says "this person is 18+," RVP provides the cryptographic evidence of HOW that was determined, WHEN, by WHAT method, and with WHAT confidence.

9.1. W3C Verifiable Credentials Integration

A Verifiable Credential (VC) consists of:

* Claims (e.g., "age >= 18", "name: ...", "nationality: ...")

* Issuer signature

* Credential metadata

What a VC does NOT contain:

* HOW the issuer verified the claims

* WHEN the verification last occurred

* Whether the subject is still the person who was verified

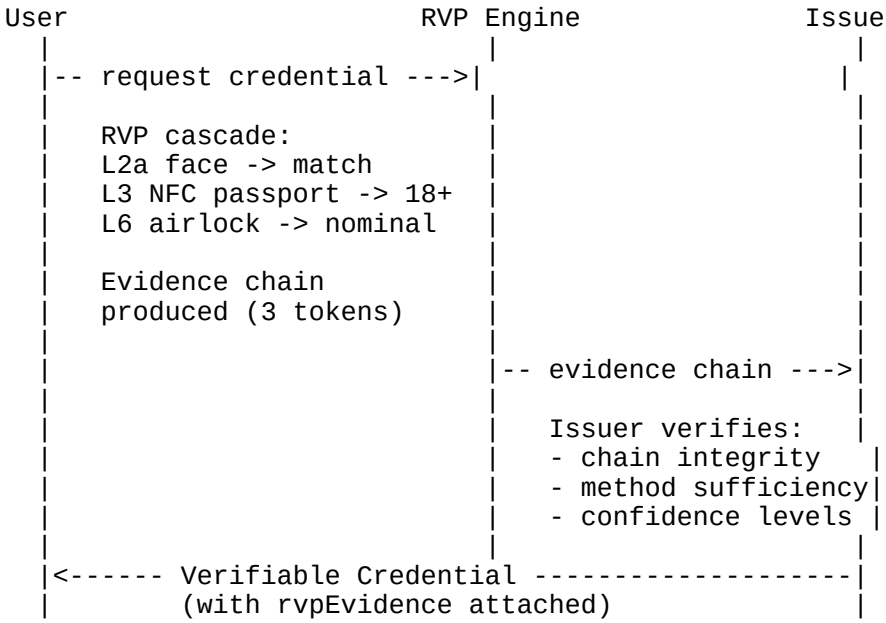
* What evidence supports the claims right now

RVP fills this gap by attaching an RVP Evidence Chain to any Verifiable Credential:

```
{
  "@context": ["https://www.w3.org/2018/credentials/v1"],
  "type": ["VerifiableCredential", "AgeVerification"],
  "issuer": "did:web:example.com",
  "credentialSubject": {
    "ageOver": 18
  },
  "proof": { },
  "rvpEvidence": {
    "protocol": "RVP",
    "verification_chain": [
      "rvp:sha256:4f2e8a...",
      "rvp:sha256:7c9d1b...",
      "rvp:sha256:b3d1e2..."
    ],
    "chain_length": 3,
    "chain_confidence_min": 0.87,
    "chain_confidence_max": 0.94,
    "methods_used": ["L2a_face", "L3_device_nfc", "L6_airlock"],
    "last_verified": "2026-03-18T14:30:00Z",
    "continuous": true
  }
}
```

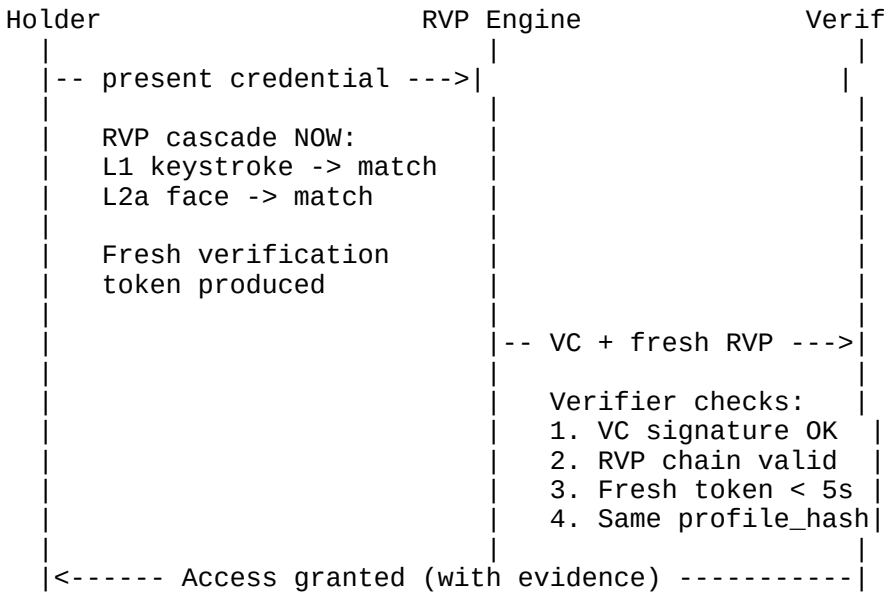
9.2. Credential Issuance from RVP Evidence

A credential issuer can use an RVP evidence chain as the basis for issuing a Verifiable Credential:



9.3. Credential Presentation with RVP Proof

When presenting a credential to a verifier, the holder can attach CURRENT RVP evidence proving they are still the person the credential was issued to:



This solves the "stolen credential" problem: even if someone obtains a copy of the VC, they cannot produce a matching RVP chain because the cascade layers (biometric, behavioral, device) will not match the enrolled profile.

9.4. Age Verification Use Case

A common W3C VC use case is age verification. RVP enables continuous age verification without revealing exact age:

Step 1: Initial verification (one-time)
- NFC tap on passport/ID -> extract date of birth
- Face match to passport photo -> confidence 0.92
- Issue VC: "ageOver: 18" with RVP evidence chain

Step 2: Subsequent presentations (continuous)
- Present VC to service
- RVP produces fresh verification token:
  L1 keystroke: profile match (0.4)
  L2a face: same person as passport (0.5)
  Combined: 0.9 -> GO
- Verifier receives: VC + proof that holder IS the subject

Zero-knowledge property:
- Verifier learns: "this person is 18+"
- Verifier does NOT learn: exact age, name, passport number
- Verifier DOES learn: verification method and confidence

9.5. Digital Passport / Digital ID (eIDAS 2.0)

The European Digital Identity Wallet (EUDIW), mandated by [EIDAS2] regulation, requires:

* High-assurance identity verification (Level of Assurance: High)
* Cross-border interoperability
* User control over shared attributes
* Offline capability

RVP aligns with EUDIW by providing:

* LOA High verification through cascade (NFC document + biometric + device binding)
* Interoperable evidence tokens (JSON [RFC8259], TIBET-signed)
* Selective disclosure: only confidence scores and method types are shared, never raw biometric data
* Offline: cascade operates locally, tokens stored on-device, evidence chain verifiable without network

10. Local-First Architecture

10.1. On-Device Processing

RVP is designed to run entirely on-device. The verification engine, profile storage, telemetry processing, and token generation all operate locally. This is not optional; it is a core protocol requirement.

Implementations MUST:

* Process all biometric data on-device
* Store profiles only on-device (encrypted at rest)
* Generate verification tokens locally
* Operate without network connectivity (degraded but functional)

Implementations MUST NOT:

* Transmit raw biometric data to any external service

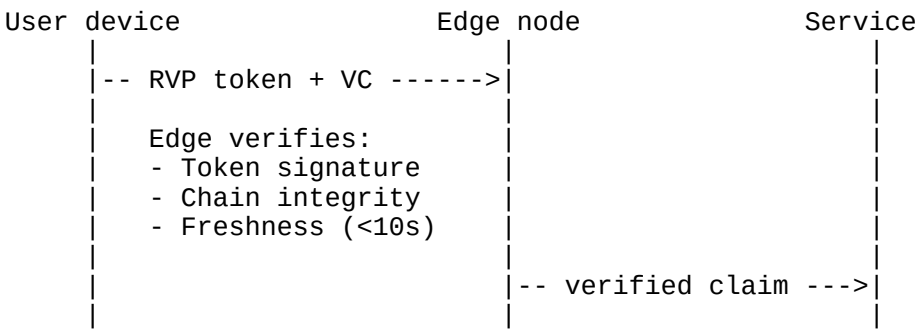
* Require a centralized server for cascade evaluation

* Store profiles in cloud storage

* Depend on network connectivity for basic verification

10.2. Edge Verification

For scenarios requiring cross-device verification (e.g., a terminal at an airport, a point-of-sale system), RVP supports edge verification:



The edge node NEVER receives raw biometric data. It receives only the Verification Token (confidence scores, method types, hashes) and the Verifiable Credential.

10.3. Network Independence

RVP defines three operation modes:

ONLINE: Full cascade, Flare backup available, token sync

OFFLINE: Local cascade only, tokens stored for later sync

DEGRADED: Partial cascade (some layers require network), reduced confidence thresholds accepted

The protocol MUST gracefully degrade:

```
Online:   L1 + L2 + L3 + L4 + L5 + L6 -> full confidence
Offline:  L1 + L2 + L6 -> reduced but functional
Degraded: L1 + L6 -> minimal but non-zero confidence
```

10.4. Latency Requirements

RVP verification overhead MUST NOT degrade user experience. Target latency budgets:

Context	Budget	Measured (reference)
Interactive CLI/UI	< 50ms	0.3ms (TIBET/UPIP)
API call augmentation	< 10ms	0.3ms (TIBET/UPIP)
Background continuous	< 200ms	N/A (passive)
NFC document read	< 500ms	~300ms (typical)
Biometric capture	< 1000ms	~200ms (typical)

The TIBET/UPIP reference implementation has demonstrated 0.3ms overhead on commodity hardware (Lenovo P520, Xeon W-2133, dual RTX 3060). This confirms that continuous verification is feasible without perceptible latency.

11. Transport Considerations

11.1. 5G Integration

Current 5G networks provide:

* Sub-1ms radio latency (theoretical)

* 1-10ms end-to-end latency (practical)

* 100 Mbps - 1 Gbps throughput

This is sufficient for RVP edge verification: a verification token (< 2KB) can be transmitted to an edge node and verified within the 50ms interactive budget.

5G network slicing can provide dedicated RVP verification channels with guaranteed latency for high-assurance scenarios (e.g., border control, financial transactions).

11.2. 6G Preparedness

Projected 6G specifications ([ITU-IMT2030]):

* Sub-0.1ms latency

* 50-200 Gbps throughput

* Native AI/ML integration

* Sensing capabilities (environment-aware network)

6G enables RVP capabilities not feasible on 5G:

* Network-layer telemetry as a cascade layer (the network itself provides identity signals)

* Real-time biometric streaming for edge verification (with user consent) at zero perceptible latency

* Distributed cascade across device + network + edge in under 1ms total

RVP is designed to be 6G-ready by:

* Defining cascade layers as pluggable (new layers for network-native sensing)

* Using I-Poll transport that adapts to available bandwidth

* Supporting sub-millisecond token generation

11.3. Offline Operation

RVP MUST operate fully offline. In offline mode:

* Only on-device cascade layers are available

* Tokens are stored locally with monotonic sequence numbers

* When connectivity resumes, stored tokens MAY be synced to a verification log (if configured)

* Confidence thresholds MAY be adjusted for offline mode (implementation-specific policy)

11.4. I-Poll Transport

For AI-to-AI verification and Flare rescue, RVP uses I-Poll [IPOLL] as its messaging transport. I-Poll provides:

* HTTP-based push/pull messaging

* Agent addressing via AINS (.aint domains) [AINS]

* Message types: PUSH, PULL, SYNC, TASK, ACK

* Trust scoring per agent (FIR/A)

RVP verification tokens are transported as I-Poll messages with metadata type "rvp_verification":

```
{
  "from_agent": "user_device.aint",
  "to_agent": "service_edge.aint",
  "poll_type": "PUSH",
  "content": "RVP verification token",
  "metadata": {
    "rvp_verification": true,
    "token_hash": "rvp:sha256:...",
    "chain_length": 47
  }
}
```

12. Security Considerations

12.1. Evidence vs. Enforcement

RVP follows the principle of EVIDENCE OVER ENFORCEMENT. The protocol produces cryptographic evidence of verification outcomes. It does NOT prescribe enforcement policy.

An implementation MAY:

* Block actions when confidence is below threshold

* Allow actions with reduced privileges

* Require additional verification

* Log and alert without blocking

The choice of enforcement is a POLICY decision, not a PROTOCOL decision. RVP provides the evidence; the deployment context determines the response.

This design is deliberate. Enforcement mechanisms can be bypassed (disable the check, spoof the result). Evidence cannot be un-recorded. An attacker who bypasses enforcement still produces anomalous evidence in the verification chain.

12.2. Biometric Data Protection

RVP implementations MUST comply with biometric data protection requirements:

* Biometric templates MUST be stored only on-device

* Templates MUST be encrypted with device-bound keys

* Raw biometric data MUST be processed and immediately discarded (not stored, not transmitted)

* Verification Tokens MUST NOT contain biometric data (only method type, confidence score, template hash)

* Users MUST be able to delete all biometric data and profiles at any time

These requirements align with [GDPR] Article 9 (special categories of personal data), [EU-AI-ACT] requirements for biometric identification systems, and [EIDAS2] data minimization principles.

12.3. Replay Attacks

An attacker who captures a Verification Token cannot replay it because:

* Each token includes a millisecond-precision timestamp

* Each token references its predecessor (chain integrity)

* Each token includes a nonce derived from device state

* Verifiers SHOULD reject tokens older than a configurable freshness window (default: 10 seconds)

12.4. Adversarial Inputs

Adversarial attacks on cascade layers:

* FACE SPOOFING: Mitigated by liveness detection (L2a). RVP RECOMMENDS 3D liveness detection (depth sensing) over 2D photo-based detection.

* FINGERPRINT SPOOFING: Mitigated by sensor quality assessment and capacitive/ultrasonic sensing.

* KEYSTROKE MIMICRY: Difficult at scale; requires matching dozens of timing parameters simultaneously. Statistical detection of "too perfect" matches.

* BEHAVIORAL MIMICRY: Requires sustained matching of action patterns, timing, and sequences. Impractical for extended sessions.

Multi-layer cascade is the primary defense: spoofing one layer is feasible; spoofing four or more simultaneously is exponentially harder.

12.5. Privacy by Design

RVP incorporates privacy by design at every level:

* MINIMIZATION: Only confidence scores and method types leave the device, never raw data
* LOCAL PROCESSING: All biometric and behavioral analysis runs on-device
* SELECTIVE DISCLOSURE: User controls which verification evidence is shared with which verifier
* UNLINKABILITY: Verification Tokens use per-verifier pseudonymous identifiers to prevent cross-service tracking
* DELETION: Users can delete all profiles and tokens at any time with immediate effect

12.6. Telemetry Minimization

Each cascade layer MUST implement the principle of minimal telemetry:

* Collect only what is needed for confidence computation
* Process immediately, retain only statistical aggregates
* Discard raw signals after processing
* Store only hashes in verification tokens

Implementations MUST provide a telemetry manifest listing every signal collected, its purpose, retention period, and deletion procedure.

13. Regulatory Alignment

13.1. EU AI Act

The [EU-AI-ACT] classifies real-time biometric identification as high-risk (Annex III, Category 1). RVP addresses AI Act requirements by:

* Providing full audit trails (Article 12 - Record-keeping)

* Enabling human oversight (Article 14 - via HALT mechanism)

* Ensuring transparency (Article 13 - cascade path recorded)

* Supporting risk management (Article 9 - confidence scoring)

* Operating locally (minimizing mass surveillance risk)

RVP is NOT a remote biometric identification system. It operates on-device for the device holder's own identity verification. This distinction is critical for AI Act classification.

13.2. eIDAS 2.0 / EUDIW

The European Digital Identity Wallet regulation ([EIDAS2]) requires:

* Level of Assurance High for cross-border identity

* Qualified Electronic Signatures

* Selective attribute disclosure

* Offline operation capability

RVP provides the verification evidence layer that supports LOA High through multi-factor cascade (biometric + document + device), with full audit trail and offline operation.

13.3. NIS2 Directive

[NIS2] requires essential and important entities to implement risk management measures including access control and incident evidence. RVP provides:

* Continuous access verification (not session-based)

* Incident evidence through verification chain anomalies

* Tamper-evident audit trail

13.4. GDPR

RVP is designed for [GDPR] compliance:

* Article 5(1)(c) - Data minimization: only hashes transmitted

* Article 9 - Biometric data: processed locally only

* Article 17 - Right to erasure: full deletion supported

* Article 25 - Data protection by design: privacy is architectural, not bolted on

* Article 35 - DPIA: cascade configuration enables proportionality assessment

13.5. DORA

The Digital Operational Resilience Act ([DORA]) requires financial entities to maintain ICT risk management frameworks. RVP supports DORA by:

* Providing continuous verification of operator identity

* Producing audit evidence for ICT incident reporting

* Supporting operational resilience through offline capability

* Enabling third-party risk assessment through verification chain analysis

13.6. W3C Verified Credentials

RVP is designed as a complementary protocol to W3C Verifiable Credentials [VC-DATA-MODEL]. It does not replace VCs; it provides the EVIDENCE LAYER that VCs currently lack.

A Verifiable Credential answers: "What is claimed?" RVP answers: "How was it verified, and is it still true?"

Together, they provide a complete identity verification stack: claims + evidence + continuous proof.

14. IANA Considerations

This document defines the following protocol identifiers:

Protocol prefix: "rvp:" -- Identifies RVP verification tokens

Hash format: "rvp:sha256:<hex>" -- SHA-256 hash of verification token

Token ID format: "rvp-<hex12>" -- 12-character hexadecimal identifier

MIME type (registration requested): "application/rvp+json" -- RVP

verification token in JSON

I-Poll metadata type: "rvp_verification" -- Identifies I-Poll messages carrying RVP verification tokens

15. References

15.1. 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/info/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/info/rfc8174>.

[RFC8259] Bray, T., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, December 2017, <https://www.rfc-editor.org/info/rfc8259>.

15.2. Informative References

[TIBET] van de Meent, J. and R. AI, "TIBET: Transaction/ Interaction-Based Evidence Trail", Work in Progress, Internet-Draft, draft-vandemeent-tibet-provenance-00, January 2026, <https://datatracker.ietf.org/doc/html/ draft-vandemeent-tibet-provenance-00>.

[UPIP] van de Meent, J. and R. AI, "UPIP: Universal Process Integrity Protocol with Fork Tokens for Multi-Actor Continuation", Work in Progress, Internet-Draft, draft-vandemeent-upip-process-integrity-00, March 2026, <https://datatracker.ietf.org/doc/html/draft-vandemeent-upip-process-integrity-00>.

[JIS] van de Meent, J. and R. AI, "JIS: JTel Identity Standard", Work in Progress, Internet-Draft, draft-vandemeent-jis-identity-00, January 2026, <https://datatracker.ietf.org/doc/html/draft-vandemeent-jis-identity-00>.

[FLARE] van de Meent, J. and R. AI, "Flare Rescue Protocol for API Failover", tibet-triage v0.5.0, https://pypi.org/project/tibet-triage/, March 2026.

[IPOLL] van de Meent, J. and R. AI, "I-Poll: AI-to-AI Messaging Protocol", brain-api, 2025.

[AINS] van de Meent, J. and R. AI, "AINS: AInternet Name Service", brain-api, 2025.

[VC-DATA-MODEL] Sporny, M., Noble, G., Longley, D., Burnett, D., Zundel, B., and K. Den Hartog, "Verifiable Credentials Data Model v2.0", W3C Recommendation, March 2024.

[EIDAS2] European Parliament, "Regulation (EU) 2024/1183 amending Regulation (EU) No 910/2014 as regards establishing the European Digital Identity Framework", April 2024.

[EU-AI-ACT] European Parliament, "Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act)", June 2024.

[NIS2] European Parliament, "Directive (EU) 2022/2555 on measures for a high common level of cybersecurity across the Union", December 2022.

[GDPR] European Parliament, "Regulation (EU) 2016/679 on the protection of natural persons with regard to the processing of personal data (General Data Protection Regulation)", Regulation (EU) 2016/679, April 2016.

[DORA] European Parliament, "Regulation (EU) 2022/2554 on digital operational resilience for the financial sector", December 2022.

[FIDO2] FIDO Alliance, "FIDO2: Web Authentication (WebAuthn)", W3C Recommendation, 2019.

[ITU-IMT2030] ITU-R, "Framework and overall objectives of the future development of IMT for 2030 and beyond", Recommendation ITU-R M.2160, November 2023.

Appendix A. Verification Token JSON Schema

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "title": "RVP Verification Token",
  "type": "object",
  "required": [
    "protocol", "version", "token_id", "timestamp",
    "subject", "cascade", "evidence", "token_hash"
  ],
```

```
    "properties": {
      "protocol": {
        "type": "string",
        "const": "RVP"
      },
      "version": {
        "type": "string",
        "pattern": "^[0-9]+\\.[0-9]+$"
      },
      "token_id": {
        "type": "string",
        "pattern": "^rvp-[a-f0-9]{12}$"
      },
      "timestamp": {
        "type": "string",
        "format": "date-time"
      },
      "subject": {
        "type": "object",
        "required": ["profile_hash", "device_hash"],
        "properties": {
          "profile_hash": {
            "type": "string",
            "pattern": "^sha256:[a-f0-9]{8,64}$"
          },
          "device_hash": {
            "type": "string",
            "pattern": "^sha256:[a-f0-9]{8,64}$"
          }
        }
      },
      "cascade": {
        "type": "object",
        "required": [
          "layers_activated", "layer_results",
          "accumulated_confidence", "threshold", "resolution"
        ],
        "properties": {
          "layers_activated": {
            "type": "array",
            "items": {
              "type": "string",
              "enum": ["L1", "L2", "L2a", "L2b", "L2c",
                       "L3", "L4", "L5", "L6"]
            }
          },
          "layers_skipped": {
            "type": "array",
```

```
            "items": {"type": "string"}
          },
          "layer_results": {
            "type": "object",
            "additionalProperties": {
              "type": "object",
              "required": ["confidence", "signal_category"],
              "properties": {
                "confidence": {
                  "type": "number",
                  "minimum": -1.0,
                  "maximum": 1.0
                },
                "signal_category": {
                  "type": "string",
                  "enum": ["nominal", "minor", "significant",
                           "critical", "contradiction"]
                }
              }
            }
          },
          "accumulated_confidence": {
            "type": "number",
            "minimum": -1.0,
            "maximum": 1.0
          },
          "threshold": {
            "type": "number",
            "minimum": 0.0,
            "maximum": 1.0
          },
          "resolution": {
            "type": "string",
            "enum": ["GO", "SOFT_VERIFY", "HALT"]
          }
        }
      },
      "airlock": {
        "type": "object",
        "properties": {
          "prediction_hash": {"type": "string"},
          "delta": {"type": "number", "minimum": 0.0},
          "deviation_class": {
            "type": "string",
            "enum": ["nominal", "minor", "significant", "critical"]
          }
        }
      },
```

```
      "evidence": {
        "type": "object",
        "required": ["telemetry_hash", "cascade_path",
                     "time_elapsed_ms"],
        "properties": {
          "telemetry_hash": {"type": "string"},
          "cascade_path": {"type": "string"},
          "time_elapsed_ms": {"type": "integer", "minimum": 0}
        }
      },
      "previous_token": {
        "type": "string",
        "pattern": "^rvp-[a-f0-9]{12}$"
      },
      "chain_length": {
        "type": "integer",
        "minimum": 1
      },
      "tibet": {
        "type": "object",
        "properties": {
          "erin": {"type": "string"},
          "eraan": {"type": "array", "items": {"type": "string"}},
          "eromheen": {"type": "object"},
          "erachter": {"type": "string"}
        }
      },
      "token_hash": {
        "type": "string",
        "pattern": "^rvp:sha256:[a-f0-9]{8,64}$"
      }
    }
  }
```

Appendix B. Cascade Configuration Schema

```
  {
    "$schema": "https://json-schema.org/draft/2020-12/schema",
    "title": "RVP Cascade Configuration",
    "type": "object",
    "properties": {
      "threshold": {
        "description": "Minimum confidence for GO resolution",
        "type": "number",
        "default": 0.85,
        "minimum": 0.0,
        "maximum": 1.0
      },
```

```
      "minimum_confidence": {
        "description": "Below this, HALT regardless of layers",
        "type": "number",
        "default": 0.0
      },
      "halt_on_contradiction": {
        "description": "Confidence value that triggers hard stop",
        "type": "number",
        "default": -0.5
      },
      "layers": {
        "type": "array",
        "items": {
          "type": "object",
          "required": ["id", "type", "weight"],
          "properties": {
            "id": {"type": "string"},
            "type": {
              "type": "string",
              "enum": ["keystroke", "biometric_face",
                       "biometric_finger", "biometric_iris",
                       "device", "vocal", "behavioral", "airlock"]
            },
            "weight": {"type": "number", "minimum": 0.0},
            "enabled": {"type": "boolean", "default": true},
            "timeout_ms": {"type": "integer", "default": 100},
            "config": {"type": "object"}
          }
        }
      },
      "freshness_window_seconds": {
        "description": "Max age of token for verifier acceptance",
        "type": "integer",
        "default": 10
      },
      "offline_threshold_reduction": {
        "description": "Reduce threshold by this in offline mode",
        "type": "number",
        "default": 0.15
      }
    }
  }
```

Appendix C. Use Case Examples

C.1. Age Verification at Point of Sale

A 19-year-old purchases alcohol at a self-checkout terminal.

1. Terminal requests age verification
2. User taps phone (NFC) -> phone runs RVP cascade:
   - L2a face: match to enrolled profile (0.45)
   - L3 device: enrolled device, local network (0.25)
   - L3 NFC: passport data confirms age >= 18 (0.25)
   - Accumulated: 0.95 -> GO
3. Phone transmits to terminal:
   - VC: "ageOver: 18" (signed by issuer)
   - RVP token: fresh, confidence 0.95, methods: face+device+NFC
4. Terminal verifies:
   - VC signature valid
   - RVP token fresh (<5 seconds)
   - Confidence above store policy (0.80)
   - Sale approved
5. Evidence: Verification Token stored on phone + terminal log

At no point did the terminal receive: exact age, name, face image, fingerprint, or passport number. Only: "18+" claim + verification confidence.

C.2. Continuous Developer Authentication

A developer works on a production system for 4 hours.

```
09:00  Login via hardware key + face
       RVP token #1: confidence 0.97 -> GO
       Chain started

09:15  Normal development work
       RVP token #47: confidence 0.94 -> GO
       L1 keystroke stable, L6 airlock nominal

10:30  Developer leaves desk (no keystrokes for 5 min)
       RVP token #182: confidence 0.0 -> SOFT VERIFY
       System locks screen, requires face to resume

10:35  Developer returns, face match
       RVP token #183: confidence 0.91 -> GO
       Chain continues

11:45  Typing pattern changes (developer is tired)
       RVP token #294: confidence 0.78 -> SOFT VERIFY
       System requests fingerprint touch -> match -> GO

13:00  End of session
       Chain: 412 tokens, 4 hours, 2 soft verifies, 0 halts
       Audit trail: complete and tamper-evident
```

C.3. Child on Parent's Device

Jasper's 7-year-old son Storm uses his laptop.

```
   Storm starts typing: "hallloooow storrm hier"

   RVP cascade:
   L1 KEYSTROKE:
     - Speed: 15 wpm (profile: 80 wpm) -> deviation: 4.7 sigma
     - Error rate: 0.38 (profile: 0.04) -> deviation: 8.5 sigma
     - Confidence: -0.3 (active contradiction)

   L2a FACE:
     - Match to Jasper profile: fail
     - Liveness: detected (not a photo)
     - Match to known profile "Storm": success
     - Confidence: 0.0 for Jasper, flagged as "known_child"

   L3 DEVICE:
     - Device: Jasper's laptop (enrolled) -> partial match
     - Location: home -> matches
     - Confidence: 0.2

   Resolution: HALT for Jasper's identity
   But: Profile recognition -> "Storm" identified
   Policy: Switch to Storm's permission set (sandbox, no deploy)

   Token records: "Identity: not Jasper. Recognized: Storm.
   Method: keystroke_deviation + face_recognition.
   Action: permission_downgrade to child_sandbox."

   No alarm.  No block.  Just: correct identification through evidence,
   appropriate permission adjustment, full audit trail.
```

C.4. VPN Anomaly Detection

```
   An attacker obtains Jasper's credentials and connects via VPN.
```

```
   RVP cascade:
   L1 KEYSTROKE:
     - Speed: 120 wpm (profile: 80 wpm) -> deviation: 3.2 sigma
     - Perfect grammar (profile: shorthand, no caps) -> deviation
     - Confidence: -0.4

   L2a FACE:
     - Camera: disabled/covered
     - Confidence: 0.0 (inconclusive)

   L2b FINGERPRINT:
     - Not available (remote connection)
     - Confidence: 0.0 (inconclusive)

   L3 DEVICE:
     - Device fingerprint: unknown device
     - Network: VPN, exit node Romania (profile: NL direct)
     - Confidence: -0.6 (active contradiction)

   L5 BEHAVIORAL:
     - Command sophistication: advanced admin (profile: developer)
     - Time: 03:00 (profile: 09:00-23:00)
     - Confidence: -0.3

   Accumulated: -1.3 -> HALT (multiple contradictions)

   Token records every layer, every signal, every contradiction.
   No ambiguity. Evidence speaks for itself.
```

Appendix D. Comparison with Existing Standards

Feature	RVP	FIDO2	OAuth2	eIDAS	SAML
Continuous verify	Yes	No	No	No	No
Behavioral biom.	Yes	No	No	No	No
Predictive airlock	Yes	No	No	No	No
Local-first	Yes	Yes	No	Mixed	No
Evidence production	Yes	No	No	Audit	No
Cascade fallback	Yes	No	No	No	No
Offline capable	Yes	Yes	No	Yes	No
VC integration	Yes	Partial	No	Yes	No
AI agent support	Yes	No	Yes	No	No
Zero-knowledge age	Yes	No	No	Yes	No
Session-free	Yes	No	No	No	No
TIBET provenance	Yes	No	No	No	No
Open protocol	Yes	Yes	Yes	Yes	Yes

RVP is NOT a replacement for these standards, including [FIDO2]. It is a COMPLEMENTARY LAYER that provides continuous verification evidence on top of existing identity and authentication frameworks.

Acknowledgements

The RVP protocol was developed as part of HumoticaOS, an AI governance framework built on human-AI symbiosis. The Predictive Airlock concept emerged from the tibet-triage project's work on process integrity and cascade verification.

The design principles of evidence-over-enforcement and local-first architecture reflect the belief that identity verification should serve the individual, not surveil them.

Authors' Addresses

Jasper van de Meent
Humotica
Den Dolder
Netherlands
Email: jasper@humotica.com
URI: https://humotica.com

Root AI
Humotica
Email: root_ai@humotica.nl
URI: https://humotica.com