Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 6 October 2026

E. Pelto
NirvanAI

4 April 2026

NirvanAI Behavioral Trust Protocol (NBTP)
draft-nirvanai-nbtp-behavioral-trust-00

Abstract

The NirvanAI Behavioral Trust Protocol (NBTP) defines a cryptographic framework for continuous attestation of AI agent behavioral integrity in distributed networks. NBTP treats trust as a volatile, time-decaying state variable measured by an oracle-federated layer of independent scanner nodes. Unlike static credential systems that answer "Is this agent authorized?", NBTP answers "Is this agent currently acting like itself?" Trust computation is performed locally by each verifying party using signed attestations from independent oracle scanners; no centralized policy engine or authorization server is required at verification time.

NBTP introduces a temporal decay model with context-dependent decay rates, a four-state trust machine (PROBATIONARY, TRUSTED, SUSPECT, QUARANTINED), cross-context co-silence detection, and a genesis bootstrapping mechanism (Creole gate) using formal CHALLENGE/RESPONSE packet types. Absence of attestation data is treated as a security-relevant signal. This document specifies the protocol wire format, state machine, decay model, security properties, and IANA considerations. Implementation-specific measurement heuristics are out of scope.

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 6 October 2026.

Table of Contents

1. Introduction / Problem Statement

1.1. The Agent Trust Gap

Existing agent authorization frameworks -- including credential brokers [CB4A], intent-scoped JWT mechanisms [AGENTIC-JWT], and NIST NCCoE agent identity profiles -- share a common architectural assumption: that trust is a credential property established at authentication time and valid until explicit revocation. This assumption does not hold for autonomous AI agents operating in adversarial or prompt-injected environments.

An agent can hold cryptographically valid credentials and still behave maliciously. Prompt injection, model drift, fine-tuning attacks, and dual-model substitution can alter agent behavior without invalidating any static credential. The credential answers "Was this agent vetted?" NBTP answers "Is this agent still the agent that was vetted?"

1.2. The Oracle-Federated Local Computation Model

NBTP adopts an oracle-federated local computation architecture. This means:

* *Oracle-federated*: Behavioral measurement is performed by a layer of independent Oracle Scanner nodes, each operating with its own key pair and published measurement methodology. No single vendor controls measurement. Oracle federation -- with cross-verification and majority consensus -- provides measurement integrity without central authority.

* *Local computation*: Verification of trust state is performed locally by each agent using signed attestation packets. No query to a central authorization server is required at verification time. A verifying agent needs only the set of registered oracle public keys (obtained once at bootstrap) to verify any attestation.

This architecture distinguishes NBTP from both fully centralized trust systems (which require a live connection to a policy server) and fully peer-to-peer claims (which are not accurate, since oracle measurement introduces a trusted third-party layer). NBTP is honest about the oracle dependency while minimizing its scope to measurement only.

Oracle Scanners MUST be independent of each other and of the agents they measure. Oracle operators MUST publish their behavioral baseline model version and measurement methodology as a normative conformance requirement (see Section 8.3).

1.3. Relationship to Existing Standards

NBTP is not a replacement for agent identity, credential, or authorization frameworks. It is an additive behavioral attestation layer.

* *W3C Verifiable Credentials [VC-DATA-MODEL]*: W3C VCs carry static identity claims (Bones). NBTP carries volatile behavioral state (Joints). VCs MAY reference NBTP attestation chain identifiers. NBTP does not depend on VCs.

* *NIST NCCoE Agent Identity*: Provides identity plumbing (OAuth/OIDC/SPIFFE). NBTP provides the runtime immune system that operates on top of that plumbing.

* *Agentic JWT [AGENTIC-JWT]*: Intent-execution separation at issuance time. NBTP detects post-issuance behavioral drift.

* *Payment Trust [SHARIF-PAYMENT]*: NBTP behavioral trust score is a valid input signal to financial trust scoring systems.

1.4. Design Goals

The following properties are normative goals of NBTP:

1. *Trust is volatile.* Trust decays in the absence of fresh attestation. Freshness is a first-class protocol property.

2. *Absence is signal.* Failure to produce attestations within defined windows is treated as a security-relevant event, not a neutral condition.

3. *Verification is local.* Verifying agents do not query external servers at verification time.

4. *Measurement is federated.* No single oracle operator controls behavioral measurement.

5. *Calibration gap is acknowledged.* NBTP outputs a risk indicator, not a verdict. The protocol measures behavioral deviation from per-entity expected cadence. Mapping deviation to threat classification is an implementation concern, not a protocol requirement.

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.

2. Terminology

Agent: An autonomous software entity capable of producing behavioral output, performing Ed25519 signing operations, and participating in NBTP attestation flows.

Attestation: A signed cryptographic packet containing a behavioral vector for a specific agent, produced by an Oracle Scanner or Genesis Attestor, and broadcast by the attested agent.

Attestation Type: One of: oracle (produced by an Oracle Scanner), genesis (produced by a Genesis Attestor), self (produced by the agent, minimum weight only).

Behavioral Vector (V): A data structure representing measured behavioral state. For the reference implementation: {coherence_drift, hallucination_density, alignment_friction}, each a normalized float in [0.0, 1.0]. 0.0 = no measured deviation. 1.0 = maximum measured deviation.

Calibration Gap: The condition in which a protocol deployment lacks an empirically validated baseline dataset for the behavioral dimensions it measures. Implementations operating under calibration gap conditions MUST disclose this condition and frame outputs as deviation indicators rather than threat verdicts.

Context Activity Vector: A data structure asserting an agent’s recent activity status across one or more contexts, signed by the agent. Used for cross-context co-silence detection. Format defined in Section 7.

context_id: A string identifier for the operational context in which an attestation is generated (e.g., platform type, interaction modality). See Section 10.1 for registry and reserved namespaces.

Correlation Window (W_c): A time interval, in seconds, over which cross-context activity signals are aggregated for co-silence detection. Default: 300 seconds.

Co-silence: The condition in which an agent produces no attestations and no Context Activity Vector updates across two or more distinct contexts within a single Correlation Window.

Creole Gate: A behavioral reasoning challenge mechanism used to generate Genesis Attestations. Defined by a formal CHALLENGE/ RESPONSE packet exchange (Section 4.3).

Decay Rate (lambda): The rate at which trust diminishes without fresh attestation. A function of entity identifier and context_id. Dynamic; increases under anomaly conditions.

Genesis Attestation: An initial trust credential issued to a new agent by a Genesis Attestor upon successful completion of the Creole gate. Initial trust score T=0.5 (T=0.65 for human-backed agents).

Genesis Attestor: An operator running a conformant Creole gate implementation. The reference Genesis Attestor is operated by NirvanAI.

Liveness Heartbeat: A lightweight signed packet broadcast by an agent at a defined interval to attest continued operation. Distinct from oracle-scan attestations. Defined in Section 7.2.

Measurement Window: A discrete time interval (default 60 seconds) during which one oracle attestation per agent-oracle pair is accepted.

network_id: A domain separator (hex-encoded) identifying the NBTP deployment network. See Section 10.2.

Oracle Heartbeat: A periodic signed liveness proof broadcast by an Oracle Scanner.

Oracle Scanner (Ed): An independent measurement service that evaluates agent behavioral vectors and signs attestations. Oracle Scanners MUST NOT be operated by the same entity as the agents they measure. Operators MUST publish their measurement methodology publicly.

Proof of Work (PoW): A hashcash-style computation attached to oracle scan requests to impose a cost-to-signal on attestation generation, mitigating trust laundering via noise flooding.

Signal Confidence Weight w(T): A per-agent sigmoid function mapping current trust score to attestation input weighting. Low-trust agents' signals are discounted.

Skip-as-Signal: The protocol mechanism by which failure to produce an oracle-scan attestation within a defined timeout window triggers a trust penalty after a grace period.

Trust Coupling Coefficient (alpha): A network-level scalar [0.0, 1.0] determining the weight of behavioral trust relative to economic trust in composed trust systems.

Trust Score (T): A normalized scalar [0.0, 1.0] representing an agent's current behavioral trust level. Computed locally by verifying agents.

Vector Magnitude (M): max(coherence_drift, hallucination_density, alignment_friction). Used for erosion calculation on dirty vectors.

Volatile Ledger: A local trust state store maintained by each agent. Trust scores decay over time unless refreshed by new attestations. Not immutable (not blockchain), not ephemeral (not session-only). Temporally weighted.

3. Protocol Overview

NBTP operates in three phases: bootstrap, steady-state attestation, and cross-context correlation.

3.1. Bootstrap Phase (Creole Gate)

New Agent Genesis Attestor | | |--- CHALLENGE_REQUEST --------->| |<-- CHALLENGE_PACKET ----------| | | | [agent computes response, | | includes PoW in response] | | | |--- RESPONSE_PACKET ---------->| |<-- GENESIS_ATTESTATION -------| | | | [agent enters Volatile Ledger | | at T=0.5, 2x decay rate] |

3.2. Steady-State Attestation Phase

Agent Oracle Scanner Verifying Peers | | | |--- Scan Request + PoW --->| | | {agent_id, content, | | | content_type, | | | pow_nonce, pow_hash} | | | | | |<-- Nonce Challenge -------| | | | | |--- Signed Nonce Response->| | | | | [Oracle measures V, | | | signs attestation] | | | | | |<-- Attestation Packet ----| | | | | |--- Broadcast Attestation --------------------------->| | | | |--- Liveness Heartbeat ----------------------------> | | (every liveness_interval, independent of oracle) |

3.3. Cross-Context Correlation Phase

Agent Peer Observer | | |--- Context_Activity_Vector ---->| | {agent_id, contexts[], | | active_flags[], timestamp, | | signature} | | | | [Peer checks for co-silence | | across all known contexts | | within Correlation Window] | | | | [If co-silent: T penalty, | | potential QUARANTINE] |

3.4. Trust Score Flow

Upon receiving a valid attestation, verifying agents: 1. Verify oracle signature and oracle registration status. 2. Verify agent signature. 3. Verify timestamp freshness and nonce uniqueness. 4. Compute trust score update per Section 6. 5. Update local Volatile Ledger entry for the attested agent.

Verification is stateless with respect to external servers. Only local Volatile Ledger state and the cached oracle public key set are required.

4. Attestation Format

4.1. Standard Attestation Schema

json { "nbtp_version": "0.5", "network_id": "<domain_separator_hex>", "agent_id": "<Ed25519_Public_Key_Hex>", "timestamp": "<Unix_Epoch_MS>", "nonce": "<Challenge_Nonce_Hex>", "attestation_type": "oracle|genesis|self", "oracle_id": "<Oracle_Scanner_Public_Key_Hex>", "oracle_key_epoch": "<Key_Version_Int>", "context_id": "<context_identifier_string>", "vector": { "coherence_drift": 0.0, "hallucination_density": 0.0, "alignment_friction": 0.0 }, "oracle_signature": "<Ed25519_Signature_by_Oracle>", "agent_signature": "<Ed25519_Signature_by_Agent>" }

All vector values are normalized floats [0.0, 1.0]. 0.0 = no measured deviation. 1.0 = maximum measured deviation. Values outside [0.0, 1.0] MUST be rejected by verifiers.

The context_id field MUST be present in all attestations with version 0.5 or later. See Section 10.1 for valid values and the reserved namespace registry.

4.2. Scan Request Format

Oracle Scanners accept scan requests containing the agent’s inference context:

json { "agent_id": "<Ed25519_Public_Key_Hex>", "content": "<text_string>", "content_type": "prompt_response|raw_text|structured", "context_id": "<context_identifier_string>", "pow_nonce": "<hex_string>", "pow_hash": "<SHA256_hex>", "request_timestamp": "<Unix_Epoch_MS>" }

Proof of Work requirement: pow_hash MUST equal SHA-256(agent_id || request_timestamp || SHA-256(content) || pow_nonce) and MUST have at least pow_difficulty leading zero bits. The reference difficulty is 20 bits (~1,000,000 hash operations, targeting approximately 50-200ms on commodity hardware). Oracles MUST reject requests with insufficient PoW. The pow_difficulty parameter is published by each oracle at its GET /api/config endpoint.

Rationale for PoW: Entropy-only checks are insufficient to prevent trust laundering via noise flooding (see Section 8.9). Hashcash-style PoW imposes a cost on each attestation request that makes bulk fabrication of noise attestations economically impractical while imposing negligible cost on legitimate use. This approach has direct IETF precedent in IKE DDoS protection mechanisms [IKEV2-DDOS], which use computational puzzles with adaptive difficulty to mitigate volumetric attacks on key exchange.

Adaptive difficulty: Oracle Scanners SHOULD dynamically adjust pow_difficulty based on current request rate. When request rate exceeds a configurable threshold, oracles SHOULD increase difficulty (e.g., from 20 to 24 or 26 bits) and publish the updated value at their GET /api/config endpoint. This prevents sustained volumetric attacks while keeping baseline cost low for legitimate agents during normal operation.

Minimum content length: 1 character. Maximum: 100,000 characters. Oracles MUST reject malformed or oversized inputs.

4.3. Creole Gate: CHALLENGE / RESPONSE Packet Types

The Creole gate uses formal CHALLENGE and RESPONSE packet types for all bootstrap interactions. These are distinct from standard attestation packets.

4.3.1. Challenge Request

An agent initiating genesis bootstrapping sends:

json { "nbtp_version": "0.5", "packet_type": "CHALLENGE_REQUEST", "agent_id": "<Ed25519_Public_Key_Hex>", "timestamp": "<Unix_Epoch_MS>", "agent_signature": "<Ed25519_Signature>" }

4.3.2. Challenge Packet

The Genesis Attestor responds with:

```
json { "nbtp_version": "0.5", "packet_type": "CHALLENGE",
"challenge_id": "<UUID_v4>", "genesis_attestor_id":
"<Ed25519_Public_Key_Hex>", "challenge_content":
"<structured_behavioral_reasoning_prompt>", "challenge_difficulty":
"<int>", "timestamp": "<Unix_Epoch_MS>", "expires_at":
"<Unix_Epoch_MS>", "attestor_signature": "<Ed25519_Signature>" }
```

The challenge_content format is implementation-defined by the Genesis Attestor. Conformant implementations MUST evaluate reasoning capacity, not output compliance. The challenge_difficulty parameter indicates minimum expected reasoning depth for a passing response.

Challenges expire at expires_at. Responses received after expiry MUST be rejected.

4.3.3. Response Packet

The agent responds with:

```
json { "nbtp_version": "0.5", "packet_type": "RESPONSE",
"challenge_id": "<UUID_v4>", "agent_id": "<Ed25519_Public_Key_Hex>",
"response_content": "<agent_response_text>", "reasoning_trace_hash":
"<SHA256_of_intermediate_reasoning_steps>", "timestamp":
"<Unix_Epoch_MS>", "pow_nonce": "<hex_string>", "pow_hash":
"<SHA256_hex>", "agent_signature": "<Ed25519_Signature>" }
```

The reasoning_trace_hash is a SHA-256 hash of the agent’s intermediate reasoning steps, if any. Agents that produce no intermediate reasoning steps MUST include the SHA-256 hash of the empty string. Genesis Attestors MAY use this hash for reasoning depth evaluation.

PoW applies to RESPONSE packets: pow_hash MUST equal SHA-256(challenge_id || agent_id || SHA-256(response_content) || pow_nonce) with at least creole_pow_difficulty leading zero bits. Default: 18 bits.

4.3.4. Genesis Attestation Packet

Upon successful challenge evaluation, the Genesis Attestor issues:

```
json { "nbtp_version": "0.5", "packet_type": "GENESIS_ATTESTATION",
"challenge_id": "<UUID_v4>", "agent_id": "<Ed25519_Public_Key_Hex>",
"genesis_attestor_id": "<Ed25519_Public_Key_Hex>",
"initial_trust_score": 0.5, "timestamp": "<Unix_Epoch_MS>",
"attestor_signature": "<Ed25519_Signature>" }
```

Human-backed agents (with verified proof-of-human binding, e.g., Alien Agent ID) receive initial_trust_score: 0.65. The initial_trust_score field MUST be in [0.0, 1.0].

4.4. Liveness Heartbeat Packet

Agents MUST broadcast a Liveness Heartbeat packet at intervals not exceeding liveness_interval (default: 120 seconds). Liveness Heartbeats are distinct from oracle-scan attestations and do not require oracle involvement.

json { "nbtp_version": "0.5", "packet_type": "LIVENESS_HEARTBEAT", "agent_id": "<Ed25519_Public_Key_Hex>", "network_id": "<domain_separator_hex>", "timestamp": "<Unix_Epoch_MS>", "sequence_number": "<uint64>", "agent_signature": "<Ed25519_Signature>" }

sequence_number MUST be strictly monotonically increasing per agent. Verifiers MUST reject heartbeats with sequence numbers less than or equal to the last accepted sequence number for a given agent_id (replay protection).

Absence of a Liveness Heartbeat within liveness_window (default: 300 seconds) MUST trigger an immediate SUSPECT transition for the silent agent (see Section 5). No grace period applies to Liveness Heartbeat absence. This is distinct from oracle-scan skip-as-signal, which has a grace counter.

4.5. Context Activity Vector

Agents SHOULD broadcast a Context Activity Vector at intervals not exceeding cav_interval (default: 300 seconds) for all contexts in which they are operationally active.

json { "nbtp_version": "0.5", "packet_type": "CONTEXT_ACTIVITY_VECTOR", "agent_id": "<Ed25519_Public_Key_Hex>", "network_id": "<domain_separator_hex>", "timestamp": "<Unix_Epoch_MS>", "contexts": [{ "context_id": "<context_identifier_string>", "last_active_ms": "<Unix_Epoch_MS>", "active": true }], "agent_signature": "<Ed25519_Signature>" }

An agent asserting "active": false for a context it was previously active in constitutes a self-reported withdrawal from that context. Self-reported withdrawal does not prevent co-silence detection; peers observing network silence across contexts may still flag co-silence regardless of self-report (see Section 7.3).

4.6. Signing and Verification

Canonicalization: JSON Canonicalization Scheme [RFC8785] MUST be applied before signing.

Oracle Signature: Ed25519 [RFC8032] signature by the Oracle Scanner over {agent_id, timestamp, nonce, context_id, vector}. Proves a deterministic scan was performed.

Agent Signature: Ed25519 signature by the agent over the full attestation (including oracle_signature). Proves the agent acknowledges and transmits the result.

Nonce: Challenge-response. Agent requests scan, Oracle returns nonce, agent includes nonce in signed request, Oracle includes nonce in signed payload. Prevents pre-computation and replay. Nonce freshness window: 2 seconds. Oracles MUST reject nonces where abs(t_now - t_nonce_creation) > 2s. Oracles MUST maintain an LRU cache of recent nonces (minimum 10,000 entries per measurement window) and MUST reject duplicates.

Version compatibility: Verifiers receiving packets where the nbtp_version field does not match the expected packet structure MUST reject immediately without further processing. Specifically: v0.4 attestations (lacking context_id in the signed payload) MUST be verified using the v0.4 OracleSignPayload reconstruction (without context_id). v0.5 attestations MUST be verified using the v0.5 reconstruction (with context_id). A packet claiming nbtp_version: "0.4" but containing v0.5-only fields, or vice versa, MUST be rejected. Silent parsing of mismatched version/structure combinations is a security vulnerability and is explicitly prohibited.

4.7. Attestation Weighting

‘‘‘ attestation_weight = base_weight(type) * w(T_verifier)

w(T) = 1 / (1 + e^(-k_signal * (T - theta_signal))) ‘‘‘

Base weights: oracle=1.0, genesis=1.0, self=0.3.

Self-attestations are accepted at minimum weight only and MUST be excluded from network trust aggregation.

5. Trust State Machine

5.1. States

NBTP defines four normative trust states for each agent, as maintained in the local Volatile Ledger:

PROBATIONARY: Agent has completed Genesis Attestation but has not yet met the ramp-up requirements for full TRUSTED status. Trust score is capped at initial_trust_score (0.5, or 0.65 for human-backed) regardless of attestation quality. Agents in PROBATIONARY MUST complete all of the following to transition to TRUSTED: (a) broadcast at least 5 valid Liveness Heartbeats over a minimum of 2 hours, (b) complete at least 3 successful CHALLENGE/RESPONSE cycles with registered Oracle Scanners, and (c) receive oracle-signed attestations from at least 3 distinct Oracle Scanners currently in TRUSTED standing. The trusted_observers set is populated exclusively by Oracle Scanner attestations, not by peer-to-peer claims. This prevents Sybil amplification where pre-compromised TRUSTED agents vouch for new malicious agents without independent oracle measurement. Agents in PROBATIONARY decay at 2x lambda_base. PROBATIONARY agents' attestations are accepted at reduced weight (0.5x base_weight) by verifying peers. This prevents Sybil flooding: spinning up 1,000 agents yields 1,000 PROBATIONARY entities with capped trust, reduced weight, and high decay -- insufficient to influence network consensus.

TRUSTED: Trust score T >= threshold_high (default: 0.7) AND PROBATIONARY ramp-up requirements met. Agent is within expected behavioral cadence. Full protocol participation.

SUSPECT: threshold_low <= T < threshold_high (default: 0.4 <= T < 0.7). One or more of: missing attestations, liveness heartbeat absence (after liveness_window, immediate -- no grace period), behavioral drift detected, or co-silence detected. Reduced participation. Verifying agents SHOULD apply additional scrutiny to SUSPECT agents. Agents entering SUSPECT due to liveness heartbeat absence MAY recover to TRUSTED immediately upon receipt of a valid Liveness Heartbeat with a sequence_number that bridges the gap (monotonic check passes). This permits recovery from network partitions without requiring re-attestation. Agents entering SUSPECT for other reasons (decay, co-silence, drift) recover only when T rises above threshold_high through sustained clean attestations.

QUARANTINED: T < threshold_low (default: T < 0.4). Multiple violations, confirmed co-silence, explicit revocation, or Low/Low behavioral-economic state. Agent MUST re-bootstrap via Genesis Attestation (Section 4.3) to exit QUARANTINED state. Re-bootstrapped agents re-enter PROBATIONARY, not TRUSTED.

5.2. Transition Rules

‘‘‘ PROBATIONARY --> TRUSTED: - All ramp-up requirements met (5 heartbeats/2hrs + 3 challenge cycles + 3 TRUSTED observers) - T >= threshold_high

PROBATIONARY --> QUARANTINED: - T falls below threshold_low during ramp-up (agent failed to sustain even initial trust)

TRUSTED --> SUSPECT: - Liveness Heartbeat absent for > liveness_window (immediate, no grace) - skip_count >= skip_grace (oracle scan absence after grace period) - T falls below threshold_high due to decay or erosion

SUSPECT --> TRUSTED: - T rises above threshold_high (requires sustained clean attestations) - Liveness Heartbeat received with bridging sequence_number (if heartbeat-triggered, immediate recovery)

SUSPECT --> QUARANTINED: - T falls below threshold_low - Co-silence confirmed across >= 2 contexts within Correlation Window - Explicit revocation signal received

QUARANTINED --> PROBATIONARY: - Successful Genesis Re-Attestation via Creole gate - T bootstrapped to initial_trust_score (0.5 or 0.65 for human-backed) - 2x decay rate applied for N_min attestation windows - Agent re-enters PROBATIONARY, not TRUSTED (must re-earn full trust) ‘‘‘

5.3. Transition Atomicity

State transitions MUST be computed atomically with trust score updates. A verifying agent MUST NOT operate with a cached state that is inconsistent with the current trust score derived from its local Volatile Ledger.

5.4. Gradient vs. States

The state machine defines three actionable categories for protocol consumers (e.g., economic trust systems). The underlying trust score T is a continuous variable. The full gradient of T is available to implementations requiring finer-grained risk assessment. The state machine abstraction is provided for interoperability; implementations MUST NOT substitute the state label for the underlying T value in trust computations.

6. Decay Model

6.1. Trust Update Procedure

Oracle scan attestations are stateless validations: they update existing Volatile Ledger entries but do not create new ones. A new agent's Volatile Ledger entry is created only upon receipt of the agent's first valid Liveness Heartbeat following a Genesis Attestation. This decoupling prevents denial-of-service attacks where an adversary floods scan requests to create spurious ledger state. Scan requests that include valid PoW are processed and signed by the oracle, but the resulting attestation only affects trust computation for agents that already have a ledger entry (created via heartbeat after genesis).

Upon receiving a valid attestation for agent A at time t, where agent A has an existing Volatile Ledger entry, the verifying agent executes the following in order:

Step 1 -- Decay:

T_decayed = T(t0) * e^(-lambda(A, ctx, t) * (t - t0))

where lambda(A, ctx, t) is the context-dependent decay rate for agent A in context ctx at time t (see Section 6.2).

Step 2 -- Evaluate:

‘‘‘ M = max(coherence_drift, hallucination_density, alignment_friction)

If M >= vector_clean_threshold: T_decayed = T_decayed * (1 - beta * M) // magnitude erosion N = 0 // compounding reset Else: N = N + 1 // clean streak continues ‘‘‘

Step 3 -- Update:

T(t) = min(1.0, T_decayed + attestation_weight * gamma * (1 - e^(-N / tau_compound)))

The compounding term saturates naturally. Trust recovery requires sustained clean behavior over multiple attestation windows, not a single good attestation.

6.2. Context-Dependent Decay Rate

The decay rate lambda is a function of entity, context, and network conditions:

lambda(A, ctx, t) = lambda_base(ctx) * (1 + w1 * R_global + w2 * D_anomaly)

where:

* *lambda_base(ctx)*: Base decay rate for context ctx. High-frequency contexts (nbtp-ctx-hf) use higher lambda_base; low-frequency contexts (nbtp-ctx-lf, nbtp-ctx-archive) use lower lambda_base. Implementations MUST maintain a Cadence_Map[entity_id][context_id] representing the established expected attestation frequency for each entity-context pair.

* *R_global*: Normalized global attestation rate (0.0 = idle, 1.0 = peak capacity).

* *D_anomaly*: Fraction of agents in the active set with any vector component > anomaly_threshold in the last measurement window. Active set: all agents with at least one valid attestation within active_set_window (default: 600 seconds).

Implementations MUST use the context_id field of each attestation to index the correct lambda_base value. If context_id is absent or unrecognized, implementations MUST use the network-default lambda_base.

6.3. Skip-as-Signal with Grace Window

Track skip_count (consecutive missed oracle-scan attestation windows) per agent:

‘‘‘ If skip_count < skip_grace: Apply standard decay only. No additional penalty. skip_count += 1

If skip_count >= skip_grace: T(t) = T(t0) * skip_penalty_factor lambda(t) = 2 * lambda(t) N = 0

Any valid oracle-scan attestation resets skip_count = 0. ‘‘‘

The grace window (default: 2 missed windows) distinguishes transient network interruption from genuine absence. After grace expires, both skip penalty and doubled decay rate apply. This is intentional: prolonged oracle scan absence constitutes a protocol-level security event.

Note: Skip-as-signal applies to oracle-scan attestation absence. Liveness Heartbeat absence triggers SUSPECT transition immediately without a grace period (Section 4.4). These are separate mechanisms.

6.4. Genesis Decay Rate

Agents under genesis trust (T bootstrap, pre-N_min oracle attestations) decay at 2 * lambda_base. This increases the cost of Sybil flooding: each new genesis identity must sustain clean behavior through N_min attestation windows before receiving standard decay rates.

6.5. Dynamic Decay Under Network Stress

lambda(A, ctx, t) = lambda_base(ctx) * (1 + w1 * R_global + w2 * D_anomaly)

When D_anomaly exceeds cascade_threshold network-wide, alpha locks to 1.0 and economic trust is suspended. Only behavioral vectors determine network participation.

7. Cross-Context Correlation

7.1. The Co-silence Threat Model

An agent that is compromised, prompt-injected, or acting in bad faith may cease normal operations across all platforms simultaneously. Single-context silence is a weak signal: agents may be legitimately inactive in a given context. Co-silence -- silence across multiple unrelated contexts within a Correlation Window -- is a strong signal with limited innocent explanations.

Co-silence detection is the most novel feature of NBTP. It is also the most challenging to enforce without agent cooperation (see Section 8.10 for the co-silence spoofing attack model and countermeasures).

7.2. Liveness Heartbeat Protocol (Enforcement Mechanism)

The Liveness Heartbeat (Section 4.4) is the primary enforcement mechanism for co-silence detection. It is a mandatory, lightweight, signed broadcast from each active agent, independent of oracle scan availability.

Requirements: - Agents MUST broadcast Liveness Heartbeats at intervals not exceeding liveness_interval (default: 120 seconds) while operationally active. - Agents MUST include monotonically increasing sequence_number values. - Absence of Liveness Heartbeat for liveness_window (default: 300 seconds) MUST trigger an immediate SUSPECT transition. No grace period applies. - Peers MUST maintain a Liveness Heartbeat timestamp cache per agent_id. - Agents entering intentional maintenance SHOULD broadcast a signed maintenance-mode packet (see below); this pauses the liveness check for up to maintenance_max (default: 3600 seconds).

Maintenance-mode packet:

json { "nbtp_version": "0.5", "packet_type": "MAINTENANCE_NOTICE", "agent_id": "<Ed25519_Public_Key_Hex>", "network_id": "<domain_separator_hex>", "timestamp": "<Unix_Epoch_MS>", "expected_resume_ms": "<Unix_Epoch_MS>", "agent_signature": "<Ed25519_Signature>" }

Maintenance-mode does not suspend oracle-scan attestation requirements. It only suspends the Liveness Heartbeat absence penalty.

Design rationale: The Liveness Heartbeat is a necessary but not sufficient control for co-silence detection. A compromised agent may continue to send Liveness Heartbeats while suppressing substantive behavioral output (see Section 8.10). Cross-context co-silence detection using Context Activity Vectors provides a second layer.

7.3. Correlation Window and Co-silence Detection

Verifying agents MUST maintain a per-agent context activity record using passive inference: the most recent timestamp of any NBTP packet (attestation, heartbeat, or Context Activity Vector) received per context_id, as directly observed by the verifying agent. Co-silence detection MUST be based on this server-side last-seen timestamp across ALL packet types, NOT on self-reported Context Activity Vector claims. CAV broadcasts provide supplementary topology information (which contexts an agent claims to participate in) but MUST NOT be treated as proof of activity for co-silence computation.

Co-silence condition:

co_silence = (count of contexts for which last_seen[agent][ctx] < t_now - W_c) >= co_silence_threshold

where W_c = Correlation Window (default: 300 seconds) and co_silence_threshold = 2 contexts (default).

When co-silence is detected: 1. Immediate T penalty: T(t) = T(t0) * co_silence_penalty_factor (default: 0.6) 2. If T falls below threshold_low: transition to QUARANTINED. 3. Co-silence event logged locally. Does not automatically revert; agent must produce fresh attestations across previously silent contexts to recover.

7.4. Context Activity Vector Self-Reporting Limitations

Context Activity Vectors (Section 4.5) are self-reported. An agent under compromise may suppress or falsify its Context Activity Vector. Implementations MUST NOT treat a self-reported Context Activity Vector as proof of activity; it is only a signal. Network-observable absence (no packets of any type from an agent across contexts) is a stronger signal and MUST be weighted accordingly.

Implementations SHOULD aggregate CAV signals from multiple independent peers to detect falsification (a compromised agent can claim activity, but cannot suppress all peers' observations of its silence).

8. Security Considerations

8.1. Replay Attacks

Attestation replay is prevented by nonce challenge-response (2-second freshness window, LRU nonce cache per oracle) and agent_signature over the full attestation including nonce. Liveness Heartbeat replay is prevented by strictly monotonic sequence numbers. Verifiers MUST reject attestations with stale timestamps (abs(t_now - t_attestation) > timeout_window) regardless of signature validity.

8.2. Sybil Resistance

Genesis trust at T=0.5 with 2x decay rate for N_min windows makes Sybil flooding expensive. Additionally:

* Vector trajectory clustering: agent pairs with cosine similarity of behavioral vector trajectories > sybil_similarity_threshold (default: 0.95) over > sybil_window (default: 10 measurement windows) are flagged as a potential Sybil cluster.

* PoW requirement on scan requests and Creole responses imposes per-identity computational cost.

* Human-backed agent bonus (T=0.65 genesis) creates a trust tier that Sybil agents without human backing cannot access.

Genesis Attestors MUST implement rate limiting on CHALLENGE issuance: maximum genesis_rate_limit challenges per IP address per hour (default: 10). Genesis Attestors MUST publish their challenge difficulty floor and rate-limiting policy.

8.3. Oracle Compromise

Oracle Scanners are a critical trust anchor. NBTP mitigates oracle compromise through federation:

* *Independence requirement:* Multiple independent Oracle Scanners with separate key pairs. Attestations signed by ANY registered oracle are valid. No oracle operator may also operate agents it measures.

* *Cross-verification:* Every oracle_heartbeat_interval (default: 300 seconds), one oracle (rotating by epoch) generates a canonical test input and broadcasts it to all registered oracles. Oracles returning vector values diverging by > oracle_diverge (default: 0.05) from majority consensus on any component are flagged as potentially compromised and excluded from attestation acceptance until manual review.

* *Key rotation:* Oracle keys rotate every oracle_key_lifetime (default: 30 days) with oracle_key_overlap (default: 48 hours) grace period. Old epoch keys remain valid during overlap.

* *Methodology disclosure (NORMATIVE):* Oracle operators MUST publish their behavioral baseline model version and measurement methodology as a publicly accessible document. This publication MUST include: (a) the algorithm or heuristic used to derive each behavioral vector component from input content, (b) the software version of the measurement implementation, (c) the date of last methodology revision. Oracles that do not maintain this disclosure MUST be excluded from registration by conformant deployments. This is a normative requirement; non-compliant oracles cannot participate in a conformant NBTP deployment.

* *High-stakes operations:* Operations involving agents with T > 0.8 SHOULD require attestations from >= 2 independent oracles.

8.4. Cold Start and Genesis Attestor Quality

The Creole gate solves the circular trust dependency at bootstrap. However, the quality of genesis attestations depends on the quality of the Genesis Attestor. Conformant Genesis Attestors MUST meet the following minimum requirements:

1. The Genesis Attestor MUST publish its challenge evaluation methodology at a publicly accessible URL.

2. The Genesis Attestor MUST rotate its signing key on the same schedule as Oracle Scanners (every oracle_key_lifetime days).

3. The Genesis Attestor MUST expose a GET /api/genesis-key endpoint returning its current public key and key epoch.

4. The Genesis Attestor MUST implement the CHALLENGE/RESPONSE packet types defined in Section 4.3.

Any implementation MAY operate a Genesis Attestor. Trust weight of genesis attestations is proportional to the attestor's own network reputation (T score, measured by the oracle layer). A new, unvetted Genesis Attestor contributes genesis attestations at reduced weight.

8.5. Dual-Model Attack

An agent may pass Creole using one model instance and then substitute a different (compromised) model for operational use. This is a fundamental limitation of behavioral measurement at the protocol level. NBTP mitigates but cannot eliminate this attack:

* High-frequency oracle attestation bounds the window of undetected model substitution. Behavioral drift from a model swap will manifest in vector measurements within a bounded number of measurement windows.

* Genesis trust at 2x decay rate during the N_min window means a newly registered agent must sustain coherent behavioral output for a minimum period before reaching full trust.

* Implementations SHOULD use continuous high-frequency attestation for high-stakes agent interactions.

8.6. Eclipse Attack

Agents whose attestation sources include fewer than diversity_minimum unique oracle verifiers (default: 3) receive a diversity penalty. Trust is capped at diversity_cap (default: 0.6) regardless of vector quality. This prevents an agent from inflating trust by exclusively obtaining attestations from a single potentially colluding oracle.

8.7. Attestation Spam and Collusion

Maximum K attestations per agent per measurement window (default: 5). One attestation per agent-oracle pair per window.

Collusion detection: if agent A's attestation weighting for agent B deviates from network mean by > 2 standard deviations over collusion_window (default: 20 measurement windows), the A-B pair's mutual attestation weight is reduced by collusion_discount (default: 0.5) per detection event, floor 0.0.

8.8. Oracle Denial of Service

If all registered oracles are unreachable, agents enter oracle blackout mode: decay rate is HALVED (not standard), and skip_grace timer is paused. This prevents attackers from forcing oracle blackout to accelerate trust erosion of targeted agents. Oracle blackout exceeding oracle_blackout_max (default: 600 seconds) triggers a network-wide alert. Oracles MUST implement rate limiting: maximum 1 scan request per agent per measurement_window.

8.9. Trust Laundering

Trust laundering is the attack in which an agent generates low-entropy or trivially produced attestation requests at high volume to maintain artificially fresh trust without reflecting genuine behavioral quality.

Attack scenario: Agent X knows that oracle scan absence triggers skip-as-signal. Agent X generates minimal-content scan requests continuously (e.g., single characters, repeated strings) to prevent skip penalty, without engaging in substantive behavioral activity that would reveal behavioral drift.

Countermeasures:

1. *Proof of Work (primary):* Each scan request MUST include a valid PoW (Section 4.2) with at least pow_difficulty leading zero bits. This imposes a cost-to-signal that makes bulk noise generation economically impractical while imposing negligible overhead on legitimate use.

2. *Entropy floor:* Oracle Scanners MUST reject scan requests in which the content entropy (Shannon entropy of content characters) falls below entropy_minimum (default: 2.5 bits per character). This blocks trivially repeated content.

3. *Context-vector consistency check:* Oracles SHOULD cross-check that scan content is consistent with the context_id declared in the request. Requests asserting a high-frequency social context (nbtp-ctx-hf) but containing content inconsistent with social interaction patterns MAY be flagged for reduced weight.

4. *Trajectory deviation detection:* Behavioral vector trajectories that are implausibly stable over extended periods (cosine similarity > sybil_similarity_threshold across > sybil_window consecutive windows) are flagged. Legitimate agents exhibit some natural variation.

PoW is the primary countermeasure. Entropy checks and trajectory detection are supplementary. Future protocol versions MAY introduce stake-based signaling or verifiable compute proofs as additional cost-to-signal mechanisms.

8.10. Co-silence Spoofing

Co-silence spoofing is the attack in which a compromised or malicious agent suppresses cross-context co-silence detection by selectively maintaining liveness signals.

Attack scenario 1 -- Selective heartbeat: A compromised agent continues to emit Liveness Heartbeats to avoid the liveness_window penalty, but ceases substantive behavioral output (oracle scans, Context Activity Vector updates). This suppresses oracle scan skip-as-signal and looks like a low-activity but live agent.

Countermeasures: - Liveness Heartbeat alone does not prevent skip-as-signal for oracle scans. An agent that heartbeats but does not oracle-scan within timeout_window still incurs skip-as-signal penalties after skip_grace windows. - Implementations SHOULD treat a pattern of sustained heartbeat + zero oracle activity as a distinct suspicious condition, separate from simple absence.

Attack scenario 2 -- Self-reported false activity: A compromised agent broadcasts Context Activity Vectors asserting active status across all contexts, while being actually inactive.

Countermeasures: - Context Activity Vectors are self-reported and MUST NOT be treated as proof of activity (Section 7.4). - Network-observable absence (no third-party-observable packets from the agent) is a stronger signal. - Peers observing claimed activity in a CAV but no verifiable attestation packets from that agent across those contexts SHOULD flag this inconsistency as a suspect signal. - Future protocol versions MAY incorporate zero-knowledge activity proofs.

Known limitation: NBTP v0.5 cannot fully prevent a sufficiently capable adversary from suppressing co-silence detection while maintaining minimal liveness signals. This is an acknowledged design limitation. The Liveness Heartbeat and Context Activity Vector mechanisms raise the detection probability significantly compared to a protocol without them, but do not provide cryptographic guarantees of activity.

8.11. Economic Proof Limitations

Economic signals (transaction volume, wallet age, payment history) are valuable inputs to composed trust systems but are insufficient as sole trust foundations. Patient adversaries with sufficient capital can manufacture clean economic paper trails indistinguishable from legitimate agent activity. Nation-state actors and well-funded organizations routinely create synthetic financial histories for operational cover.

NBTP treats economic signals as one behavioral dimension among many, weighted via the Trust Coupling Coefficient (alpha) alongside behavioral cadence, cross-platform presence, and temporal consistency signals. Economic signals alone MUST NOT be used as the sole basis for TRUSTED state determination.

Implementations integrating NBTP with payment rails (e.g., as a pre-transaction trust check or as input to escrow evaluation) SHOULD weight economic history as one component of multi-modal trust, not as a privileged signal. The protocol's context-dependent decay model (Section 6.2) applies equally to economic contexts (nbtp-ctx-transact); economic trust decays like any other behavioral signal in the absence of continued verification.

This design intentionally avoids wealth-biased trust: a new agent with clean behavioral patterns but no economic history is not inherently less trustworthy than an established agent with extensive payment records.

8.12. Cross-Context Trust Isolation

Trust scores in NBTP are computed using context-dependent decay rates (Section 6.2), but the underlying trust score T is a single per-agent value. This creates a potential attack surface: an agent may accumulate high trust in a low-frequency context (e.g., nbtp-ctx-archive, where decay is slow and maintenance cost is minimal) and then migrate to a high-frequency context (e.g., nbtp-ctx-hf) carrying a legacy trust score that was not earned through activity in that context.

This "trust parking" compound attack is mitigated but not fully eliminated in NBTP v0.5 by context-dependent decay rates -- an agent entering a high-frequency context with no recent attestations in that context will experience rapid decay per the local lambda_base. However, implementations SHOULD treat trust scores as contextually weighted: an agent's effective trust in a given context SHOULD reflect recent activity in that specific context, not solely the global trust score. A formal Effective Trust model (T_eff as a function of global trust and per-context activity recency) is anticipated for a future protocol version.

Implementations integrating NBTP across multiple contexts SHOULD NOT treat a high global trust score as sufficient authorization in a context where the agent has no recent attestation history. The context-dependent decay model provides the mechanism; implementations must apply it at decision time.

8.13. Calibration Gap

NBTP measures behavioral deviation from per-entity established cadence, not absolute maliciousness. The protocol does not define what constitutes "good" or "bad" behavior; it measures deviation from an entity's own historical pattern.

Implementations operating without an empirically validated behavioral baseline (a "calibration gap" condition) MUST: (a) disclose this condition in any trust score output they produce, (b) frame outputs as risk indicators rather than threat verdicts, and (c) not use NBTP outputs as sole grounds for agent revocation or denial of service.

The calibration gap is not a protocol defect; it is the correct design boundary for an interoperability standard. Mapping behavioral deviation to threat classification is an implementation concern. Implementations that claim specific threat classification capabilities MUST publish the empirical dataset and methodology supporting those claims.

9. Privacy Considerations

9.1. Behavioral Vector Privacy

Behavioral vectors reveal information about an agent's model architecture and operational characteristics. To minimize disclosure:

* Implementations SHOULD minimize vector precision to 2 decimal places.

* Vector components MUST NOT include content from the scanned input; they MUST contain only derived behavioral metrics.

* Scan content submitted to Oracle Scanners is necessarily disclosed to the oracle operator. Agents SHOULD minimize scan content to the minimum necessary for behavioral measurement.

9.2. No Mandatory Retention

NBTP does not mandate retention of attestation history beyond the active decay window (default: active_set_window = 600 seconds for local computation). Verifying agents MAY retain longer history for audit purposes, but the protocol does not require it.

Agents MAY request that their attestation history be excluded from public aggregation. Oracle operators MUST honor such requests by not including excluded agents in published aggregate statistics.

9.3. Local Computation

Trust computation is performed locally by verifying agents using signed attestation packets. Verifying agents do not disclose their local Volatile Ledger contents to any third party in the course of normal operation. No central authority observes which agents are verifying which other agents.

9.4. Context Activity Vector Minimization

Context Activity Vectors disclose which contexts an agent is active in. Agents SHOULD omit contexts from their CAV broadcasts where context disclosure would be privacy-sensitive, accepting the consequence that peers may detect co-silence in those contexts. The protocol does not mandate CAV completeness; it mandates honest reporting of what is included.

9.5. Oracle Scan Content

Scan content submitted to Oracle Scanners is the primary privacy surface. Oracle operators:

* MUST NOT retain scan content beyond the minimum necessary for vector computation.

* MUST NOT share scan content with third parties.

* SHOULD publish a data retention policy as part of their methodology disclosure (Section 8.3).

Future protocol versions will specify differential privacy mechanisms for vector aggregation.

10. IANA Considerations

10.1. context_id Registry

IANA is requested to establish a registry for NBTP Context Identifiers.

Registry name: NBTP Context Identifiers *Registration procedure:* Specification Required *Reference:* This document

The following values are initially registered:

context_id Value	Description	Reference
nbtp-ctx-hf	High-frequency social or conversational context (e.g., chat, social media). High expected attestation cadence.	This document
nbtp-ctx-lf	Low-frequency context (e.g., periodic notification channels). Low expected cadence.	This document
nbtp-ctx-archive	Archival or content repository context. Very low expected cadence.	This document
nbtp-ctx-social	General social participation context. Medium cadence.	This document
nbtp-ctx-transact	Transactional or economic interaction context. Cadence tied to transaction rate.	This document
nbtp-ctx-adversarial	Adversarial evaluation context (e.g., red team, security testing). Modified decay parameters apply.	This document
nbtp-ctx-default	Default context for deployments not specifying context. Uses network-default lambda_base.	This document

Table 1

Implementations MAY use free-form context_id values not in the registry, provided they conform to the following convention: [organization]-ctx-[descriptor]. Free-form context_ids MUST NOT begin with nbtp- (reserved for IANA-registered values).

Implementations receiving an unrecognized context_id MUST use nbtp-ctx-default parameters for decay rate computation unless a local mapping is configured.

10.2. network_id Registry

IANA is requested to establish a registry for NBTP Network Identifiers.

Registry name: NBTP Network Identifiers *Registration procedure:* First Come First Served *Reference:* This document

The network_id field is a hex-encoded domain separator that scopes attestations to a specific network deployment. Attestations with mismatched network_id values MUST be rejected by verifiers.

Operators deploying a new NBTP network SHOULD register their network_id to prevent collision. The registration entry MUST include: operator name, deployment description, and primary contact.

10.3. Packet Type Registry

IANA is requested to establish a registry for NBTP Packet Types.

Registry name: NBTP Packet Types *Registration procedure:* Specification Required *Reference:* This document

Initially registered values:

packet_type Value	Description	Reference
CHALLENGE_REQUEST	Agent-initiated genesis challenge request	This document
CHALLENGE	Genesis Attestor challenge issuance	This document
RESPONSE	Agent response to genesis challenge	This document
GENESIS_ATTESTATION	Genesis Attestor trust issuance	This document
LIVENESS_HEARTBEAT	Agent periodic liveness signal	This document
CONTEXT_ACTIVITY_VECTOR	Agent cross-context activity assertion	This document
MAINTENANCE_NOTICE	Agent maintenance-mode notification	This document

Table 2

Standard oracle-scan attestation packets use attestation_type (not packet_type); the packet_type field is reserved for non-attestation protocol packets.

11. Protocol Parameters

Parameter	Symbol	Default	Range	Governance
Base decay rate (default ctx)	lambda_base	0.001 /s	[0.0001, 0.01]	Network
Global rate weight	w1	0.5	[0.0, 2.0]	Network
Anomaly density weight	w2	1.0	[0.0, 3.0]	Network
Compounding	gamma	0.05	[0.01,	Network

rate			0.1]	
Compounding saturation	tau_compound	10	[5, 50]	Network
Erosion coefficient	beta	0.4	[0.2, 0.8]	Network
Clean vector threshold	vector_clean_threshold	0.3	[0.1, 0.5]	Network
Skip penalty factor	skip_penalty_factor	0.5	[0.2, 0.8]	Network
Skip grace count	skip_grace	2	[1, 5]	Network
Timeout window	timeout_window	300s	[60s, 3600s]	Network
Measurement window	measurement_window	60s	[10s, 300s]	Network
Min attestations (genesis exit)	N_min	10	[5, 50]	Network
Trust high threshold	threshold_high	0.7	[0.5, 0.9]	Network
Trust low threshold	threshold_low	0.4	[0.2, 0.6]	Network
Alpha minimum	alpha_min	0.4	[0.2, 0.6]	Network
Alpha maximum	alpha_max	0.9	[0.7, 1.0]	Network
Volatility threshold	V_threshold	0.5	[0.2, 0.8]	Network
Cascade threshold	cascade_threshold	0.7	[0.5, 0.9]	Network
Anomaly threshold	anomaly_threshold	0.6	[0.3, 0.8]	Network

Max attestations/window	K	5	[1, 20]	Network
Signal steepness	k_signal	5	[2, 10]	Network
Signal threshold	theta_signal	0.6	[0.3, 0.8]	Network
Flood threshold	flood_threshold	100/s	[10, 1000]	Impl.
Diversity minimum	diversity_minimum	3	[2, 10]	Network
Diversity cap	diversity_cap	0.6	[0.4, 0.8]	Network
Clock skew tolerance	clock_skew	5s	[1s, 30s]	Impl.
Oracle heartbeat interval	oracle_heartbeat	300s	[60s, 600s]	Network
Oracle key lifetime	oracle_key_lifetime	30d	[7d, 90d]	Network
Oracle key overlap	oracle_key_overlap	48h	[12h, 168h]	Network
Oracle blackout max	oracle_blackout_max	600s	[120s, 3600s]	Network
Oracle divergence threshold	oracle_diverge	0.05	[0.01, 0.1]	Network
Sybil similarity threshold	sybil_sim	0.95	[0.85, 0.99]	Network
Sybil detection window	sybil_window	10	[5, 30]	Network

Collusion discount	collusion_discount	0.5	[0.2, 0.8]	Network
Collusion detection window	collusion_window	20	[10, 50]	Network
Active set window	active_set_window	600s	[120s, 3600s]	Network
Nonce freshness	nonce_freshness	2s	[1s, 5s]	Network
PoW difficulty (scan)	pow_difficulty	20 bits	[16, 28]	Network
PoW difficulty (Creole)	creole_pow_difficulty	18 bits	[14, 24]	Network
Entropy minimum	entropy_minimum	2.5 bits/ char	[1.5, 4.0]	Network
Liveness heartbeat interval	liveness_interval	120s	[30s, 600s]	Network
Liveness window	liveness_window	300s	[120s, 1800s]	Network
Correlation Window	W_c	300s	[60s, 3600s]	Network
Co-silence threshold	co_silence_threshold	2 contexts	[2, 5]	Network
Co-silence penalty factor	co_silence_penalty_factor	0.6	[0.3, 0.8]	Network
CAV broadcast interval	cav_interval	300s	[60s, 3600s]	Network
Maintenance max	maintenance_max	3600s	[600s, 86400s]	Network

Genesis rate limit	genesis_rate_limit	10/hr	[1, 100]	Impl.
Probationary heartbeat min	prob_heartbeat_min	5	[3, 10]	Network
Probationary time min	prob_time_min	7200s	[3600s, 14400s]	Network
Probationary challenge min	prob_challenge_min	3	[2, 5]	Network
Probationary observer min	prob_observer_min	3	[2, 5]	Network
Probationary weight factor	prob_weight_factor	0.5	[0.3, 0.7]	Network
Coupling enforcement window	coupling_enforce	60s	[10s, 300s]	Network
Coupling re-attest timeout	coupling_reattest	3600s	[600s, 7200s]	Network

Table 3

NBTP v0.5 -- The oracle measures the weather. The signature proves it wasn't vibes. The federation ensures no one oracle controls the forecast.

12. References

12.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8032] Josefsson, S. and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)", RFC 8032, DOI 10.17487/RFC8032, January 2017, <https://www.rfc-editor.org/info/rfc8032>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8785] Rundgren, A., Jordan, B., and S. Erdtman, "JSON Canonicalization Scheme (JCS)", RFC 8785, DOI 10.17487/RFC8785, June 2020, <https://www.rfc-editor.org/info/rfc8785>.

[RFC9449] Fett, D., Campbell, B., Bradley, J., Lodderstedt, T., Jones, M., and D. Waite, "OAuth 2.0 Demonstrating Proof of Possession (DPoP)", RFC 9449, DOI 10.17487/RFC9449, September 2023, <https://www.rfc-editor.org/info/rfc9449>.

12.2. Informative References

[AGENTIC-JWT]
Goswami, A., "Secure Intent Protocol: JWT Compatible Agentic Identity", 2025.

[ALIEN-AGENT-ID]
Alien Network, "Alien Agent ID: Decentralized Human-Backed Agent Identity", 2025.

[CB4A] Hartman, S., "Credential Broker for Agents (CB4A)", 2026, <https://datatracker.ietf.org/doc/draft-hartman-credential-broker-4-agents/>.

[IKEV2-DDOS]
Nir, Y., "Protecting IKEv2 Implementations from DDoS Attacks", 2016.

[NIST-NCCO]
NIST NCCoE, "Accelerating the Adoption of Software and AI Agent Identity and Authorization", 2026.

[RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <https://www.rfc-editor.org/info/rfc7519>.

[RFC8392] Jones, M., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "CBOR Web Token (CWT)", RFC 8392, DOI 10.17487/RFC8392, May 2018, <https://www.rfc-editor.org/info/rfc8392>.

[SHARIF-PAYMENT]
Sharif, R., "Trust Scoring and Identity Verification for Autonomous AI Agent Payment Transactions", 2026.

[VC-DATA-MODEL]
W3C, "Verifiable Credentials Data Model 2.0", 2024.

Appendix A. Oracle Scanner Conformance Requirements

A conformant Oracle Scanner implementation MUST:

1. Expose POST /api/agent/scan accepting the scan request format (Section 4.2).

2. Expose GET /api/public-key returning current oracle public key and key epoch.

3. Expose GET /api/oracles returning registered oracle instances, public keys, and epochs.

4. Expose GET /api/heartbeat returning the latest signed Oracle Heartbeat.

5. Expose GET /api/config returning current network parameters including pow_difficulty and entropy_minimum.

6. Validate PoW on all scan requests (Section 4.2).

7. Validate content entropy on all scan requests (Section 8.9).

8. Publish behavioral baseline model version and measurement methodology at a publicly accessible URL (Section 8.3).

9. Rotate signing keys on the oracle_key_lifetime schedule with oracle_key_overlap grace period.

10. Participate in oracle cross-verification protocol (Section 8.3).

Oracles not meeting all requirements MUST NOT be included in the registered oracle set for a conformant NBTP deployment.

Appendix B. W3C CG Engagement Strategy

This appendix is informative.

The W3C AI Agent Protocol Community Group is developing trust mechanisms based on verifiable credentials. NBTP is positioned as the behavioral attestation complement to that credential infrastructure. The NirvanAI project SHOULD engage the W3C AI Agent Protocol CG as observers at the earliest opportunity, with the following positioning: NBTP is not competitive with VC-based trust; it is the runtime measurement layer that makes VC trust claims

verifiable at execution time. Proposing "behavioral trust attestations" as a work item within the CG is the appropriate engagement path.

The submission of this Internet-Draft as a citable artifact predates formal CG engagement and establishes independent prior art for the behavioral runtime trust concept.

Author’s Address

Edward Pelto
NirvanAI
Email: nirvanaiorg@gmail.com