Internet-Draft
draft-wang-jep-judgment-event-protocol-04
Intended status: Experimental
Expires: September 30, 2026

Y. Wang

March 2026

Judgment Event Protocol (JEP)
A Minimal Verifiable Log Format for Agent Decisions
draft-wang-jep-04

Abstract

This document defines the Judgment Event Protocol (JEP) — a minimal, verifiable log format for decision-related operations in agent systems. The protocol specifies four immutable event verbs (J, D, T, V), a signed JSON event structure with anti-replay protection, and standardized signature verification rules. This protocol enables interoperability across platforms and autonomous agents.

Optional standard extensions are provided for privacy protection, accountability distribution, data lifecycle management, and data sovereignty adaptation.

Table of Contents

1. Introduction

1.1. Motivation

As autonomous AI agents operate across platforms and domains, there is a growing need for a minimal, standardized, and secure format to record and verify decision-making actions. The Judgment Event Protocol (JEP) addresses this need by providing a lightweight, verifiable, and interoperable event logging capability.

The protocol is designed to adapt to diverse cultural, legal, and political environments. Optional extensions address privacy protection, shared accountability, data lifecycle management, and data sovereignty requirements, enhancing the protocol’s deployability across jurisdictions.

1.2. Scope

JEP defines:

- Four immutable core decision event verbs (J, D, T, V)
- A minimal signed event format
- Anti-replay protection using nonce and timestamp
- Standardized signature and verification procedures
- A modular optional extension framework

JEP explicitly does NOT define:

- Accountability systems or legal liability determination rules
- Authorization delegation validity or permission chain constraints
- State machine or lifecycle enforcement logic
- Governance rules or business compliance determination

1.3. Requirements Language

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.

1.4. Terminology

- Event: A single observable record of a decision-related action
- Actor: The entity that generates and signs an event
- Nonce: A unique identifier used to prevent replay attacks
- Verifier: An entity that validates event signatures and legitimacy

2. Protocol Model

2.1. Event Verbs

JEP defines four immutable core verbs:

- J (Judge): Initiate a new decision
- D (Delegate): Transfer decision authority to another actor
- T (Terminate): Close the decision lifecycle
- V (Verify): Verify the authenticity of an existing event

Verify events (V) SHOULD use the ref field to reference the target event, avoiding circular dependencies and semantic ambiguity.

2.2. Core Event Format

A valid JEP event MUST contain the following top-level fields:

- jep: Protocol version (fixed as "1")
- verb: Verb enumeration (J, D, T, V)
- who: Actor identifier (URI/DID/public key hash)
- when: Unix timestamp (seconds since epoch)
- what: Cryptographic multihash of decision content. Implementations SHOULD support common hash functions such as SHA-256 and SM3, and encode the hash with its algorithm identifier (e.g., "sha256:", "sm3:") as per RFC 9122.
- nonce: UUIDv4 globally unique identifier
- aud: Intended recipient (domain/identifier, RECOMMENDED)
- sig: Digital signature over canonicalized JSON
- ref: Reference to related event hash/ID, used for chain linking (OPTIONAL). This field MAY be used with any verb to build event chains, and SHOULD be null for root events.

Example Judgment Event:

```
{
  "jep": "1",
  "verb": "J",
  "who": "did:example:agent-789",
  "when": 1742345678,
  "what": "sha256:e8878aa9a38f4d123456789abcdef01234",
  "nonce": "f47ac10b-58cc-4372-a567-0e02b2c3d479",
  "aud": "https://platform.example.com",
  "ref": null,
  "sig": "eyJhbGciOiJFZERTQSJ9..."
}
```

Example Verify Event with Chain Reference:

```
{
  "jep": "1",
  "verb": "V",
  "who": "did:example:verifier-123",
  "when": 1742345680,
  "what": null,
  "nonce": "a1b2c3d4-5678-4abc-8ef0-123456789abc",
  "aud": "https://platform.example.com",
  "ref": "sha256:e8878aa9a38f4d123456789abcdef01234",
  "sig": "eyJhbGciOiJFZERTQSJ9..."
}
```

2.3. Anti-Replay Mechanism

1. Event initiators MUST generate a new UUIDv4 nonce for each event
2. Nonces MUST be generated from a cryptographically secure random source
3. Receivers MUST cache nonces and reject duplicates
4. Unless otherwise configured, a clock skew tolerance of ±5 minutes (300 seconds) is RECOMMENDED
5. Use of the aud field to bind events to specific recipients is RECOMMENDED to reduce attack surface

2.4. Signature and Verification Rules

The protocol relies on:

- JSON Canonicalization Scheme (JCS | RFC 8785)
- JSON Web Signature (JWS | RFC 7515)

Verification Steps:

1. Parse and validate JSON structure
2. Verify JWS signature validity using the actor’s public key
3. Verify nonce uniqueness (no replay)
4. Verify timestamp falls within the allowed window
5. Verify recipient matches if aud field is present
6. Verify ref format compliance if present
7. Return "valid/invalid" result

2.5. Deployment Adaptation Standard Extensions

All extensions MAY be registered with IANA. Extensions are designed to be modular and non-intrusive.

2.5.1. Digest-Only Anonymity Extension (Privacy)

Identifier: https://jep.org/priv/digest-only

Purpose: Pseudonymization of actor identities during normal operation, with the ability to trace back to the original identity via a trusted salt holder during disputes or audits.

Core Fields: Salted identity hash, trusted salt holder identifier.

Characteristics: Identity cannot be reversed without the salt; signatures ensure digest authenticity.

2.5.2. Multi-Signature Extension (Accountability Distribution)

Identifier: https://jep.org/multisig

Purpose: Record collaborative decisions among multiple parties, enabling distributed accountability.

Core Fields: Signature threshold, participant list, aggregate signature.

Characteristics: Events are considered valid only when the threshold number of valid signatures is met.

2.5.3. Time-to-Live Extension (Data Timeliness)

Identifier: https://jep.org/ttl

Purpose: Set expiration time for plaintext data, with automatic anonymization or deletion upon expiry, retaining only the evidence hash.

Core Fields: Expiration timestamp, expiry handling policy.

2.5.4. Sovereign Storage Adaptation (Data Sovereignty)

Identifier: https://jep.org/storage

Purpose: Decouple the protocol from specific storage vendors, allowing users to independently choose storage locations to ensure data sovereignty.

Core Fields: Storage adapter type, storage address, integrity hash.

2.5.5. Subject Reference Extension (Transparent Traceability)

Identifier: https://jep.org/subject

Purpose: Explicitly identify the subject of a decision to enhance traceability transparency.

Privacy Protection Methods:
- The id field MAY contain: plaintext DID / public key hash, salted identity digest, or be omitted
- In privacy mode, id MUST use an irreversible hash and MUST NOT contain plaintext PII
- MAY be combined with TTL extension for automatic expiry or anonymization

Fields:
- id: URI, DID, salted hash digest, or empty

3. Security and Privacy Considerations

3.1. Core Security Guarantees

- Replay Protection: Nonce uniqueness is enforced; duplicate events are rejected
- Tamper Resistance: Full event cryptographic signature; any modification invalidates the signature
- Confidentiality: The protocol does not provide encryption; sensitive data MUST be transmitted via TLS or JWE
- Algorithm Compatibility: Ed25519 is RECOMMENDED for general use. Implementations MAY also support P-256, SM2, and post-quantum cryptography to meet regional compliance requirements or specific security policies.
- Randomness: Nonces MUST be generated from cryptographically secure random sources

3.2. Extension Security Specifications

- Digest-Only Anonymity: Privacy depends on salt confidentiality; loss of salt permanently prevents traceability
- Multi-Signature Security: Implementations MUST protect against malicious key attacks
- TTL Timeliness: TTL declares expiration policy but does not enforce physical deletion; deletion operations SHOULD retain verifiable logs
- Sovereign Storage: Security of external storage is outside JEP’s core scope
- Subject Reference: This is purely informational and does not imply consent or authorization from the referenced subject

3.3. Privacy and Data Minimization

- Digest-only anonymity mode SHOULD be used for routine logging
- TTL extension SHOULD be used to comply with "right to be forgotten" and data retention regulations
- Subject reference extension SHOULD be enabled only as needed to avoid unnecessary identity exposure

4. IANA Considerations

4.1. JEP Verbs Registry

IANA is requested to create a "JEP Verbs Registry". Registration policy: Expert Review.

Initial Registrations:

Verb	Description
J	Initiate a decision

D	Transfer decision authority
T	Terminate decision lifecycle
V	Verify an existing event

4.2. JEP Extensions Registry

IANA is requested to create a "JEP Extensions Registry". Registration policy: Specification Required.

Initial Registrations:

- https://jep.org/priv/digest-only
- https://jep.org/multisig
- https://jep.org/ttl
- https://jep.org/storage
- https://jep.org/subject

5. 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>.

[RFC7515] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515, May 2015, <https://www.rfc-editor.org/info/rfc7515>.

[RFC7516] Jones, M. and J. Hildebrand, "JSON Web Encryption (JWE)", RFC 7516, DOI 10.17487/RFC7516, May 2015, <https://www.rfc-editor.org/info/rfc7516>.

[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>.

[RFC9122] Peon, R. and D. Thaler, "Multihash", RFC 9122, DOI 10.17487/RFC9122, August 2021, <https://www.rfc-editor.org/info/rfc9122>.

[RFC9562] Leach, P., Mealling, M., and R. Salz, "Universally Unique IDentifier (UUID)", RFC 9562, DOI 10.17487/RFC9562, May 2024, <https://www.rfc-editor.org/info/rfc9562>.

Author’s Address

Yuqiang Wang
Email: signal@humanjudgment.org
GitHub: https://github.com/hjs-spec