Internet-Draft
draft-wang-jep-judgment-event-protocol-01
Intended status: Experimental
Expires: 20 September 2026

Y. Wang
HJS Foundation Ltd.

20 March 2026

Judgment Event Protocol (JEP)

Status of This Memo

Copyright Notice

Abstract

This document specifies the Judgment Event Protocol (JEP), a minimal, verifiable format for logging judgment-related actions in distributed systems. It defines four immutable event verbs (J, D, T, V), a signed JSON event structure with replay protection, and standardized signature verification rules. The protocol is designed to be interoperable across platforms and autonomous agents, and may be used either natively or as a foundation for extended accountability, audit, and decision provenance frameworks.

Table of Contents

1. Introduction

1.1. Motivation

As distributed systems and autonomous agents operate across platforms and domains, a minimal, standard, secure format is needed to log and verify judgment actions. The Judgment Event Protocol (JEP) fills this gap with a compact, verifiable, and interoperable event record.

1.2. Scope

The Judgment Event Protocol (JEP) defines:
- A set of basic judgment event verbs
- A minimal signed event format
- Replay protection using nonce and timestamp
- Standardized signature and verification rules

JEP does NOT define:
- Accountability structure or liability rules
- Delegation validity or authorization chains
- State machines or lifecycle enforcement
- Multi-layer privacy architecture
- Cryptographic erasure
- Dispute detection or resolution
- Governance or compliance logic

JEP may be used natively for basic event recording, or incorporated into extended frameworks that add accountability, audit, privacy, or decision provenance capabilities.

1.3. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174].

1.4. Terminology

- Event: A single observable judgment-related action
- Actor: Entity that generates and signs the event
- Nonce: A unique value to prevent replay attacks
- Verifier: Entity that checks signature and validity

2. Protocol Model

2.1. Event Primitives

JEP defines four immutable event verbs:

- J (Judge): A judgment was created
- D (Delegate): A judgment authority was transferred
- T (Terminate): A judgment lifecycle was ended
- V (Verify): An existing event was verified

The V (Verify) verb SHOULD reference the target event via the optional 'ref' field to avoid circular dependencies and ambiguous verification semantics.

2.2. Core Event Format

A valid JEP event MUST contain these top-level fields:

- jep: Protocol version (MUST be "1")
- verb: One of [J, D, T, V]
- who: Actor identifier (URI, DID, or public key hash)
- when: Unix timestamp (seconds since epoch)
- what: Multihash [RFC9122] of judgment content (RECOMMENDED)
- nonce: UUIDv4 unique identifier [RFC9562]
- aud: Intended audience (domain/identifier, RECOMMENDED)
- sig: Digital signature over canonicalized JSON
- ref: For V (Verify) events: hash/ID of verified event (OPTIONAL)

Example JEP Event:

```
{
  "jep": "1",
  "verb": "J",
  "who": "did:example:agent-789",
  "when": 1742345678,
  "what": "122059e8878aa9a38f4d123456789abcdef01234",
  "nonce": "f47ac10b-58cc-4372-a567-0e02b2c3d479",
  "aud": "https://platform.example.com",
  "sig": "eyJhbGciOiJFZERTQSJ9..."
}
```

Example Verify Event:

```
{
  "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": "122059e8878aa9a38f4d123456789abcdef01234",
  "sig": "eyJhbGciOiJFZERTQSJ9..."
}
```

2.3. Replay Protection

To prevent replay attacks:

1. The issuer MUST generate a new UUIDv4 nonce for every event.
2. Nonce MUST be created with a cryptographically secure random generator to ensure unpredictability.
3. The receiver MUST cache and reject duplicate nonces.
4. A clock skew tolerance of +/-5 minutes (300 seconds) is RECOMMENDED if not explicitly configured.
5. The 'aud' field SHOULD bind events to specific recipients.

2.4. Signature and Verification

JEP uses:

- JSON Canonicalization Scheme (JCS) [RFC8785]
- JSON Web Signature (JWS) Compact [RFC7515]

Verification Steps:

1. Parse and validate JSON structure.
2. Verify JWS signature using the actor's public key.
3. Ensure nonce is unique and not previously seen.
4. Ensure timestamp is within acceptable time window.
5. Ensure audience matches if specified.
6. If 'ref' is present, validate its format as a reference.
7. Return VALID or INVALID.

3. Security Considerations

3.1. Replay Attacks

Without nonce caching and uniqueness enforcement, JEP events can be fraudulently replayed. All implementations MUST reject duplicate nonces.

3.2. Tampering

JEP events are cryptographically signed; any modification invalidates the signature.

3.3. Confidentiality

JEP does not provide encryption. Sensitive events SHOULD be transmitted over TLS or encrypted using JWE.

3.4. Algorithm Agility

Ed25519 is RECOMMENDED. ECDSA P-256, SM2, and post-quantum algorithms are permitted.

3.5. Randomness Requirements

Nonce generation MUST use cryptographically secure randomness to prevent prediction and collision attacks.

4. IANA Considerations

4.1. JEP Verbs Registry

IANA is requested to create a "Judgment Event Protocol (JEP) Verbs" registry. The registration policy for this registry is Expert Review. Initial entries are:

Verb	Description
J	A judgment was created
D	A judgment authority was transferred
T	A judgment lifecycle was ended
V	An existing event was verified

4.2. Extension Registry

IANA is requested to create a "Judgment Event Protocol (JEP) Extensions" registry for future compatible extensions. The registration policy for this registry is Specification Required.

5. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs", BCP 14.
[RFC7515] Jones, M., "JSON Web Signature (JWS)".
[RFC8174] Leiba, B., "Ambiguity in RFC 2119 Keywords", BCP 14.
[RFC8785] Rundgren, et al., "JSON Canonicalization Scheme".
[RFC9122] Maheshwari, H., "Multihash Format".
[RFC9562] Davis, D., et al., "Universally Unique IDentifiers (UUIDs)".

Author’s Address

Yuqiang Wang
HUMAN JUDGMENT SYSTEMS FOUNDATION LTD.
Email: signal@humanjudgment.org
GitHub: https://github.com/hjs-spec