Internet-Draft
draft-wang-hjs-accountability-02
Intended status: Standards Track
Expires: September 2026

Y. Wang
HJS Foundation Ltd.

March 2026

HJS: An Accountability Layer for AI Agents
draft-wang-hjs-accountability-02

Abstract

This document defines the Human Judgment Structure (HJS) v2.0, a modular accountability l ayer for AI agents built on the Judgment Event Protocol (JEP). HJS v2.0 separates core ac countability semantics from optional governance features, enabling a narrow-waist infrast ructure that can be deployed incrementally while supporting full regulatory compliance wh en needed. This specification provides a complete replacement for HJS v1.0 that preserves all original functionality while introducing modular architecture.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BC P 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 Intern et-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 2026.

Copyright Notice

Table of Contents

1. Core Accountability Model (HJS-Core) 4
   1.1. Core Principles 4
   1.2. Core Field Definition 5
   1.3. Core Semantics 5
   1.4. Core Verification 6
   1.5. Core Guarantees 6
2. Optional Extension Modules 7
   2.1. Extension Framework (HJS-Ext) 7
   2.2. Privacy Module (HJS-Privacy) 8
   2.3. Compliance Module (HJS-Compliance) 10
   2.4. Time Module (HJS-Time) 12
   2.5. Authorization Module (HJS-Auth) 13
   2.6. State Machine Module (HJS-State) 14
   2.7. Dispute Resolution Module (HJS-Dispute) 16
3. Complete Coverage Mapping 18
4. Implementation Guide 19
   4.1. Minimal Implementation 19
   4.2. Adding Privacy Extension 19
   4.3. Error Code Reference 20
5. IANA Considerations 21
   5.1. HJS Extensions Registry 21
   5.2. HJS State Registry 22

6. Security Considerations 23
7. Privacy Considerations 24
8. References 25
8.1. Normative References 25
8.2. Informative References 25
Author's Address 26

1. Core Accountability Model (HJS-Core)

1.1. Core Principles

HJS-Core is the minimal, non-negotiable accountability layer that all HJS-compliant systems MUST implement. It consists of:

1. JEP Event Format (as defined in draft-wang-jep-judgment-event-protocol-01)
2. One additional field: based_on for responsibility chaining

No other fields or semantics are required for core compliance.

1.2. Core Field Definition

```
{
  "jep": "1",
  "verb": "J",
  "who": "did:example:alice",
  "when": 1742345678,
  "what": "122059e8878a...",
  "nonce": "f47ac10b-58cc-4372-a567-0e02b2c3d479",
  "based_on": null,
  "sig": "eyJhbGciOiJFZERTQSJ9..."
}
```

1.3. Core Semantics

Verb	Core Accountability Meaning
J	Initial responsibility claim. based_on MUST be null.
D	Responsibility transfer. based_on MUST reference the delegating event.
T	End of responsibility chain. based_on MUST reference the active event.
V	Attest to chain validity. based_on MUST reference the verified event.

1.4. Core Verification

```
function verify_core(event, public_key):
    # Step 1: JEP signature verification
    if not verify_jep_signature(event, public_key):
        return "INVALID"

    # Step 2: Chain integrity (if based_on present)
    if event.based_on and not event.based_on.exists():
        return "INVALID"

    # Step 3: Chain head check
    if event.verb == "J" and event.based_on is not None:
        return "INVALID"

    return "VALID"
```

1.5. Core Guarantees

- Non-repudiation: Events are signed by the actor
- Integrity: Any modification breaks the signature
- Chain integrity: based_on ensures causal ordering
- Replay protection: nonce prevents replay

2. Optional Extension Modules

Extensions are independent, optional modules that add functionality to HJS-Core. Each extension defines its own fields, verification rules, and IANA registration.

2.1. Extension Framework (HJS-Ext)

All extensions MUST be placed in the extensions object at the top level of the JWS Payload.

```
{
  "jep": "1",
  "verb": "J",
  "who": "did:example:alice",
  "when": 1742345678,
  "what": "122059e8878a...",
  "nonce": "uuid",
  "based_on": null,
  "sig": "...",
  "extensions": {
    "https://hjs.org/privacy": { ... },
    "https://hjs.org/compliance/gdpr": { ... }
  }
}
```

IANA Registration Requirements:

- Extension Identifier: URI (HTTPS recommended)
- Semantics Description: URL to specification
- Compatibility Level: "full" (ignorable), "wire" (may affect interoperability), "none" (required)
- Version: Semantic version

2.2. Privacy Module (HJS-Privacy)

Purpose: GDPR-compliant data minimization and right to erasure.

2.2.1. Three-Tier Privacy Architecture

Tier	Content	Visibility	Purpose
Tier 1	verb, who, nonce	Public	Basic accountability verification
Tier 2	based_on, time_anchor, audit extensions	Encrypted/ Limited	Audit and authorization
Tier 3	what (hash only)	Public hash; external encrypted	Content integrity; supports erasure

2.2.2. Extension Fields

```
{
  "extensions": {
    "https://hjs.org/privacy": {
```

```
      "tier": 2,
      "encrypted": true,
      "erase_mode": "ERASED",
      "erase_key_id": "key-123",
      "erase_proof": "base64...",
      "erase_timestamp": 1742345678
    }
  }
}
```

Field	Type	Required	Description
tier	integer	REQUIRED	1=public, 2=encrypted audit, 3=external
encrypted	boolean	REQUIRED if tier=2	Whether Tier 2 is encrypted
erase_mode	string	OPTIONAL	ARCHIVED or ERASED
erase_key_id	string	REQUIRED if erase_mode=ERASED	Key identifier for destruction
erase_proof	string	OPTIONAL	Proof of key destruction
erase_timestamp	integer	OPTIONAL	When erasure was performed

2.3. Compliance Module (HJS-Compliance)

Purpose: Regulatory compliance fields for GDPR, EU AI Act, jurisdiction marking.

2.3.1. GDPR Compliance Extension

```
{
  "extensions": {
    "https://hjs.org/compliance/gdpr": {
      "legal_basis": "consent",
      "data_subject": "hash:did:example:user123",
      "retention_period": "P5Y",
      "dpo_signature": "base64...",
      "purpose": "credit_scoring"
    }
  }
}
```

Field	Type	Description
legal_basis	string	consent, contract, legal_obligation, etc.
data_subject	string	Pseudonymized subject identifier
retention_period	string	ISO 8601 duration
dpo_signature	string	Optional DPO signature
purpose	string	Processing purpose

2.3.2. EU AI Act Transparency Extension

```
{
  "extensions": {
    "https://hjs.org/compliance/aiact": {
      "risk_level": "high",
      "human_oversight": true,
      "human_supervisor_id": "did:example:supervisor",
      "conformity_assessment": "CE-2026-123",
      "transparency_marking": true,
      "system_version": "v2.1.0"
    }
  }
}
```

Field	Type	Description
risk_level	string	minimal, limited, high, unacceptable
human_oversight	boolean	Whether human oversight was applied
human_supervisor_id	string	Identifier of supervising human
conformity_assessment	string	Conformity assessment identifier
transparency_marking	boolean	Whether output is marked as AI-generated
system_version	string	AI system version

2.3.3. Jurisdiction Marking Extension

```
{
  "extensions": {
    "https://hjs.org/compliance/jurisdiction": {
      "applicable_law": "GDPR",
      "data_residency": "EU",
      "processing_location": "FRA",
      "retention_days": 1825
    }
  }
}
```

2.4. Time Module (HJS-Time)

Purpose: High-precision timestamps and external anchoring.

```
{
  "extensions": {
    "https://hjs.org/time": {
      "high_res": 1742345678123456789,
      "anchor_type": "scitt",
      "anchor_ref": "https://scitt.example/entries/abc123",
      "anchor_proof": "base64...",
      "time_sources": ["nts", "hw_clock"]
    }
  }
}
```

Field	Type	Description
high_res	integer	Nanosecond-precision timestamp

anchor_type	string	scitt, tsa, blockchain, none
anchor_ref	string	Reference to external anchor
anchor_proof	string	Cryptographic proof of anchor
time_sources	array	Sources used for time consensus

2.5. Authorization Module (HJS-Auth)

Purpose: Delegation acceptance signatures and authorization validation.

```
{
  "verb": "D",
  "extensions": {
    "https://hjs.org/auth": {
      "acceptance_sig": "base64...",
      "acceptance_sig_alg": "Ed25519",
      "authority": "https://auth.example/policy",
      "aip_session": "session-123",
      "aip_capability": "delegate"
    }
  }
}
```

Field	Type	Required	Description
acceptance_sig	string	REQUIRED	Delegatee's signature
acceptance_sig_alg	string	REQUIRED	Signature algorithm
authority	string	OPTIONAL	Authority that issued delegation right
aip_session	string	OPTIONAL	AIP session identifier
aip_capability	string	OPTIONAL	AIP capability being delegated

2.6. State Machine Module (HJS-State)

Purpose: Lifecycle state tracking for accountability chains.

2.6.1. Simplified State Machine

State	Description
ACTIVE	Judgment created, responsibility held
DELEGATING	Delegation initiated, awaiting acceptance
DELEGATED	Responsibility successfully transferred
TERMINATED	Lifecycle ended (archived or erased)

2.6.2. Full State Machine (Optional)

State	Description
DISPUTED_FROZEN	Conflict detected, chain frozen pending resolution
RESOLVED_CONTINUE	Dispute resolved, chain continues
RESOLVED_INVALID	Dispute resolved, chain invalidated

2.6.3. Extension Fields

```
{
  "extensions": {
    "https://hjs.org/state": {
      "status": "ACTIVE",
      "state_machine": "simplified",
      "timeout_at": 1742345678,
      "entered_at": 1742345600,
      "transition": "JUDGE"
    }
  }
}
```

Field	Type	Description
status	string	Current state
state_machine	string	simplified or full
timeout_at	integer	Time when delegation times out
entered_at	integer	When this state was entered
transition	string	Event that caused transition

2.7. Dispute Resolution Module (HJS-Dispute)

Purpose: Multi-party dispute detection and federal governance.

```
{
  "extensions": {
    "https://hjs.org/dispute": {
      "detected_at": 1742345678,
      "detection_mode": "dual",
      "resolution_mode": "federal",
      "resolution_proof": "base64...",
      "resolution_outcome": "CONTINUE",
      "governance_model": {
        "type": "weighted_threshold",
        "participants": [
          {"id": "did:example:org-a", "weight": 2, "signature": "base64..."},
          {"id": "did:example:org-b", "weight": 1, "signature": "base64..."}
        ],
        "threshold": 2
      }
    }
  }
}
```

Field	Type	Description

detected_at	integer	When dispute was detected
detection_mode	string	dual, platform, open
resolution_mode	string	federal, manual, automated
resolution_proof	string	Proof of resolution
resolution_outcome	string	CONTINUE, INVALID
governance_model	object	Weighted participant configuration

3. Complete Coverage Mapping

This table maps all original HJS v1.0 sections to their v2.0 equivalents:

Original HJS Section	v2.0 Location
Section 1.1 Introduction	Section 1.1 Core Principles
Section 1.2.1 Four Primitives	Section 1.3 Core Semantics
Section 1.2.2 Core Receipt Fields	Section 1.2 Core Field Definition + Section 2.4 HJS-Time
Section 1.2.3 Three-Tier Privacy	Section 2.2.1 Three-Tier Privacy Architecture
Section 1.2.4 Signature and Chain Verification	Section 1.4 Core Verification
Section 1.3 Cryptographic Erasure	Section 2.2.2 Privacy Module Fields
Section 1.4 Security Considerations	Section 1.5 Core Guarantees + Section 6
Section 2.1 Simplified State Machine	Section 2.6.1 Simplified State Machine
Section 2.2 Extension Field Format	Section 2.1 Extension Framework
Appendix A Full State Machine	Section 2.6.2 Full State Machine
Appendix A.3 Federal Governance	Section 2.7 Dispute Resolution Module
Appendix B Advanced Verification	Section 2.4 Time Module + Section 2.5 Authorization Module
Appendix C Compliance Mapping	Section 2.3 Compliance Module
Appendix D Concrete Extensions	Distributed across Sections 2.2-2.7
Appendix F Examples	Section 4 Implementation Guide

4. Implementation Guide

4.1. Minimal Implementation

```
class HJSReceipt:
    def __init__(self, verb, who, when, what, nonce, based_on):
        self.verb = verb
```

```
        self.who = who
        self.when = when
        self.what = what
        self.nonce = nonce
        self.based_on = based_on

    def verify(self, public_key):
        # JEP signature verification
        if not self.verify_signature(public_key):
            return "INVALID"

        # Chain integrity
        if self.based_on and not self.parent_exists():
            return "INVALID"

        # Chain head check
        if self.verb == "J" and self.based_on is not None:
            return "INVALID"

        return "VALID"
```

4.2. Adding Privacy Extension

```
class HJSPrivacyReceipt(HJSReceipt):
    def __init__(self, **kwargs):
        super().__init__(**kwargs)
        self.privacy_tier = kwargs.get('privacy_tier', 1)
        self.encrypted = kwargs.get('encrypted', False)

    def verify_privacy(self):
        if self.privacy_tier == 2 and not self.encrypted:
            return "INVALID"
        return "VALID"
```

4.3. Error Code Reference

Code	Description
INVALID_SIGNATURE	Signature verification failed
BROKEN_CHAIN	Parent hash mismatch
UNAUTHORIZED_DELEGATION	Delegate not authorized
INVALID_TERMINATION	Invalid termination state
PAYLOAD_ERASED	Tier 3 content erased
CHAIN_TOO_DEEP	Delegation depth exceeds limit
EXPIRED_RECEIPT	Timestamp outside window
DISPUTED	Dual-mode conflict detected

5. IANA Considerations

5.1. HJS Extensions Registry

Extension Identifier	Compatibility	Reference
https://hjs.org/privacy	full	Section 2.2
https://hjs.org/compliance/gdpr	full	Section 2.3.1

https://hjs.org/compliance/aiact	full	Section 2.3.2
https://hjs.org/compliance/jurisdiction	full	Section 2.3.3
https://hjs.org/time	wire	Section 2.4
https://hjs.org/auth	wire	Section 2.5
https://hjs.org/state	full	Section 2.6
https://hjs.org/dispute	none	Section 2.7

5.2. HJS State Registry

State Value	Description
ACTIVE	Judgment created, responsibility held
DELEGATING	Delegation initiated, awaiting acceptance
DELEGATED	Responsibility successfully transferred
TERMINATED	Lifecycle ended
DISPUTED_FROZEN	Conflict detected, chain frozen
RESOLVED_CONTINUE	Dispute resolved, chain continues
RESOLVED_INVALID	Dispute resolved, chain invalidated

6. Security Considerations

HJS v2.0 inherits the security properties of JEP and adds accountability-specific guarantees:

- Token Forgery: Root signature prevents unauthorized modification of core fields.
- Chain Tampering: Each hop signature covers all previous hops; modification of any hop invalidates all subsequent signatures.
- Replay Attacks: nonce prevents replay; session binding (if implemented) adds additional protection.
- Key Management: Private keys SHOULD be stored in HSMs or secure enclaves. Key rotation is supported via the Key Evolution extension (Appendix D.6 of original HJS, now available as independent extension).
- Offline Verification: Core verification requires only the issuer’s public key and the current session identifier. No network calls are required.

7. Privacy Considerations

HJS v2.0 supports privacy regulations through the Privacy Module:

- Data Minimization: Three-tier architecture ensures only necessary data is exposed.
- Right to Erasure: Cryptographic erasure mechanism (erase_mode="ERASED") destroys encryption keys while preserving audit trail integrity.
- Pseudonymization: The who field SHOULD use opaque identifiers; data_subject fields use hashed identifiers.
- Retention: retention_period field supports explicit retention policy declaration.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC7515] Jones, M., "JSON Web Signature (JWS)", RFC 7515, May 2015.

[RFC7516] Jones, M. and J. Hildebrand, "JSON Web Encryption (JWE)", RFC 7516, May 2015.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, May 2017.

[RFC8785] Rundgren, A., et al., "JSON Canonicalization Scheme (JCS)", RFC 8785, June 2020.

[RFC9334] Birkholz, H., et al., "Remote ATtestation procedureS (RATS) Architecture", RFC 9334, January 2023.

[RFC9562] Davis, D., et al., "Universally Unique IDentifiers (UUIDs)", RFC 9562, May 2024.

[draft-wang-jep-judgment-event-protocol-01] Wang, Y., "Judgment Event Protocol (JEP)", Work in Progress, March 2026.

8.2. Informative References

[I-D.ietf-scitt-architecture] Birkholz, H., et al., "Supply Chain Integrity, Transparency, and Trust (SCITT) Architecture", Work in Progress.

[DID-CORE] Sporny, M., et al., "Decentralized Identifiers (DIDs) v1.0", W3C Recommendation, July 2022.

[GDPR-2016] European Parliament, "General Data Protection Regulation (GDPR)", 2016.

[EU-AI-ACT] European Commission, "Artificial Intelligence Act", 2024.

Author’s Address

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