Internet-Draft
Intended status: Experimental
Expires: October 29, 2026

Y. Wang
April 29, 2026

HJS: Accountability Receipts for AI Agents
A Minimal JEP Profile for Exportable AI Receipts
draft-wang-hjs-accountability-05

Abstract

This document defines HJS, a minimal accountability receipt infrastructure for AI agents. HJS is a profile of the Judgment Event Protocol (JEP) and does not define an independent event signing protocol. HJS uses JEP events to bind signed event claims to AI-agent behavior records, receipt manifests, optional privacy-preserving human participant references, and optional deployment-specific evidence references.

The HJS core is intentionally small. It defines behavior-record digest binding, receipt manifests, receipt bundles, and validation of cryptographic and structural consistency. All other capabilities, including human privacy modes, participant-supplied references, post-event review references, explanation-material references, risk descriptors, model evidence, tool-call evidence, policy-check evidence, multi-party export, and cryptographic capability profiles, are optional extensions or deployment profiles.

HJS is infrastructure. It does not assign legal liability, prove subjective intent, define governance rules, enforce monitoring, define fairness, define appeal rights, define explanation rights, determine authorization validity, or establish regulatory compliance.

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 October 29, 2026.

Copyright Notice

Table of Contents

Author’s Address

1. Introduction

1.1. Motivation

AI agents increasingly perform decision-related operations across platforms, organizations, and jurisdictions. Such operations may include planning, delegation, tool calls, content generation, verification, escalation, or termination of a workflow. Operators, affected parties, auditors, and downstream reviewers often need evidence that a particular AI-agent behavior record was created, bound to an accountable issuer, and not altered after signing.

HJS addresses this need by defining a minimal accountability receipt infrastructure over JEP. HJS records observable AI-agent behavior evidence, including inputs, outputs, tool calls, model descriptors, policy checks, and optional human participant references. It binds this evidence to JEP events by digest, while leaving signature syntax, event hashes, references, replay protection, validation modes, extension criticality, and cryptographic capability profiles to JEP.

HJS is not intended to solve the algorithmic black-box problem in a general philosophical or legal sense. HJS records evidence about observable behavior. It can support audit, incident review, explanation-material reference, and receipt export workflows, but it does not by itself prove internal motivation, subjective intent, causal completeness, legal responsibility, fairness, or regulatory sufficiency.

1.2. Relationship to JEP

HJS is a profile of the Judgment Event Protocol (JEP) [I-D.wang-jep-judgment-event-protocol]. JEP defines the minimal verifiable event protocol: the J/D/T/V verbs, event object, algorithm-tagged digest strings, event hash semantics, detached JWS signing over JCS-canonicalized payloads, key resolution, validation modes, and the ext/ext_crit extension framework.

HJS does not redefine those JEP mechanisms. An HJS event is a JEP event with HJS-specific extension content and HJS-specific semantics for the object referenced by the JEP what field.

```
JEP:  minimal verifiable event protocol
HJS:  minimal AI-agent receipt profile over JEP
```

This separation is a design requirement. HJS implementations MUST NOT use an HJS-specific signature syntax that bypasses JEP. HJS implementations MUST NOT reinterpret JEP event hashes, detached JWS signatures, ext/ext_crit processing, JEP validation modes, or JEP cryptographic profile semantics.

1.3. Scope

HJS defines:

* An AI-agent accountability receipt profile over JEP-Core-1.
* HJS behavior records for observable AI-agent behavior evidence.
* HJS receipt manifests and receipt bundles for portable validation packages.
* HJS-specific optional extension identifiers for receipt, privacy, identity rotation, external references, explanation-material references, multi-party export, risk, model evidence, tool-call

evidence, and policy-check evidence.

* Validation rules for binding JEP events to HJS behavior records and receipt manifests.

* Privacy guidance for separating machine behavior evidence from human participant identity.

HJS explicitly does not define:

* Legal liability, culpability, fault, negligence, good faith, or responsibility assignment.

* Authorization validity, delegation validity, permission-chain correctness, lawful basis, consent validity, or data-subject rights fulfillment.

* Monitoring mandates, enforcement rules, sanctions, escalation duties, employment consequences, or penalty procedures.

* Jurisdictional policy, political constraints, regulatory compliance determinations, procedural fairness, appeal rights, explanation rights, access rights, disclosure duties, or evidentiary effect.

* A global identity or trust framework.

* A new cryptographic signature container independent of JEP.

1.4. Minimal Core and Optional Extensions

HJS is infrastructure for creating, binding, exporting, and validating AI-agent accountability receipts. The HJS core is intentionally minimal: it defines a JEP-based receipt profile, behavior-record digest binding, receipt manifests, receipt bundles, and validation of cryptographic and structural consistency.

All other capabilities, including human privacy modes, participant-supplied references, post-event review references, explanation-material references, risk descriptors, model provenance, tool-call evidence, policy-check evidence, multi-party export, and cryptographic capability profiles, are optional extensions or deployment profiles.

HJS extensions provide technical mechanisms only. They do not define governance rules, fairness, consent, appeal rights, explanation rights, liability, legal compliance, access rights, disclosure duties, remedies, sanctions, or evidentiary effect.

The absence of an optional extension MUST NOT be interpreted by the protocol as agreement, waiver, admission, lack of objection, lack of harm, absence of relevant context, absence of external rights, or absence of an available external process.

1.5. 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.

1.6. Terminology

HJS Event:
: A JEP event that carries HJS-specific receipt or behavior semantics.

HJS Behavior Record:
: A JSON object describing observable AI-agent behavior evidence. A behavior record is normally referenced by the JEP what field through an algorithm-tagged digest string.

HJS Receipt Manifest:
: A JSON object that packages references to JEP events, behavior records, and external evidence needed for cross-platform validation.

HJS Receipt Bundle:
: A collection containing one or more JEP events, associated behavior records, optional receipt manifests, optional external evidence, and optional privacy-preserving views.

Human Participant:
: A natural person or human-controlled role referenced by an HJS behavior record. Human participants are not necessarily the JEP signer.

External Reference:
: A digest-bound or otherwise integrity-protected pointer to externally supplied material. HJS does not define the meaning or effect of the referenced material.

AI Agent:
: A software agent or AI-enabled service whose observable behavior is recorded by HJS.

Issuer:
: The entity identified by the JEP who field and bound to the signing key under the applicable JEP trust profile.

Deployment Profile:
: An external profile that defines local policies, evidence meanings, privacy handling, export rules, or governance processes outside the HJS core.

2. HJS Model

2.1. Design Goals

HJS has five design goals:

Machine Evidence Integrity:
: HJS binds AI-agent behavior evidence to signed JEP events. Signed records are tamper-evident, not physically immutable.

Minimal Core:
: HJS-Core-1 defines only receipt infrastructure and digest binding. Capabilities that could imply governance, fairness, access, explanation, appeal, risk, or disclosure outcomes remain optional and external.

Human Participant Privacy:
: HJS supports pseudonymous, digest-only, rotating, opaque, or withheld references for human participants. These mechanisms reduce unnecessary exposure of personal data but do not guarantee universal anonymity or unlinkability.

Technical Neutrality:
: HJS records observable event claims and evidence references. It does not judge legality, intent, fault, liability, authorization validity, consent validity, fairness, appeal validity, explanation sufficiency, or compliance.

Deployment Adaptability:
HJS supports optional evidence profiles for privacy handling, external references, multi-party export, model descriptors, tool calls, policy checks, risk classification, and data lifecycle management.

2.2. Layering

HJS relies on JEP for:

* JEP verbs J, D, T, and V.
* The JEP event object and top-level field definitions.
* Algorithm-tagged digest strings.
* Event hash and ref semantics.
* Detached JWS signatures over JCS-canonicalized unsigned events.
* Key resolution through trust profiles.
* Acceptance validation and archival validation.
* The ext and ext_crit extension container rules.
* JEP cryptographic profile extension semantics.

HJS defines the content and interpretation of HJS behavior records and HJS receipt manifests. HJS-specific content MUST be carried as external content referenced by JEP digest strings, as JEP extension content under ext, or both.

2.3. Infrastructure Positioning

HJS is infrastructure for creating, binding, exporting, and validating AI-agent accountability receipts. It is not an accountability tribunal, governance framework, compliance authority, monitoring mandate, fairness engine, appeal system, explanation-rights framework, liability determination system, or human-conduct interpretation system.

HJS can support fairness-oriented or rights-respecting workflows by allowing deployments to attach, export, and verify relevant evidence references. HJS does not define fairness, determine whether a process is fair, prescribe remedies, assign sanctions, define appeal rights, or determine whether any explanation is sufficient.

2.4. Machine Behavior and Human Participant Identity

HJS distinguishes machine behavior evidence from human participant identity.

Machine behavior evidence includes observable records such as model invocations, prompts or prompt digests, outputs or output digests, tool-call descriptors, policy-check results, decision labels, and workflow transitions. Such evidence SHOULD be represented in HJS behavior records and bound to JEP events by digest.

Human participant identity includes natural-person identifiers, account identifiers, subject identifiers, or role identifiers. Such identity SHOULD be represented through privacy-preserving references when possible. The JEP who field identifies the issuer or signer of the JEP event; HJS implementations MUST NOT overload the JEP who field to mean "human subject" unless the human participant is also the JEP event issuer under the applicable trust profile.

2.5. HJS Events, Behavior Records, and Receipts

An HJS event is a JEP event that references an HJS behavior record, receipt manifest, or validation report. The usual pattern is:

* Create an HJS behavior record describing observable AI-agent behavior.
* Compute an algorithm-tagged digest string over that behavior record.
* Place that digest in the JEP what field.
* Add the HJS receipt extension to the JEP ext object.
* Sign the JEP event according to JEP.
* Optionally package the JEP event, behavior record, and evidence descriptors in an HJS receipt manifest or receipt bundle.

3. HJS-Core-1 Profile

3.1. Required JEP Profile

An implementation conforming to HJS-Core-1 MUST support JEP-Core-1 as defined by [I-D.wang-jep-judgment-event-protocol].

HJS-Core-1 producers MUST generate HJS events as JEP events. HJS-Core-1 verifiers MUST perform the appropriate JEP validation mode before performing HJS-specific validation.

HJS-Core-1 does not define a separate cryptographic algorithm policy. Cryptographic algorithm support, signature capability, canonicalization profile, and hash family are inherited from JEP and any applicable JEP cryptographic profile extension.

3.2. Use of JEP Verbs

HJS uses the JEP verbs as follows:

J (Judge):
: Records a decision-related AI-agent behavior event or root receipt event.

D (Delegate):
: Records a delegation-related behavior event, such as delegation of a task, tool authority, review responsibility, or workflow segment. HJS does not determine whether the delegation is legally or operationally valid.

T (Terminate):
: Records a termination-related behavior event, such as closure of an AI-agent task, workflow, audit boundary, or receipt chain. HJS does not enforce lifecycle closure.

V (Verify):
: Records a verification action or verification result concerning a referenced HJS event, behavior record, receipt manifest, receipt bundle, or optional external reference.

3.3. JEP who, what, ref, ext, and ext_crit in HJS

The JEP who field identifies the issuer of the HJS event. It is resolved to a signing key according to the applicable JEP trust profile. It SHOULD identify an AI agent, AI runtime, operator, accountability service, organizational issuer, or other signing

authority. It SHOULD NOT contain plaintext personally identifiable information unless required by the deployment profile and permitted by applicable policy.

The JEP what field in an HJS event normally contains the algorithm-tagged digest string of one of the following:

* an HJS behavior record;

* an HJS receipt manifest;

* an HJS validation report;

* another deployment-defined evidence object.

The JEP ref field SHOULD reference the event hash of a related JEP event when an HJS event participates in a chain. For an HJS V event, ref SHOULD reference the target JEP event being verified.

HJS-specific extension content MUST be placed under the JEP ext object using HJS extension identifiers. If an HJS extension is necessary for interpreting or validating the HJS event, the extension identifier MUST be listed in ext_crit.

3.4. HJS Receipt Extension

The HJS Receipt Extension identifies a JEP event as carrying HJS receipt semantics.

Identifier: https://hjs.org/receipt

The extension value is a JSON object. HJS-Core-1 producers SHOULD include this extension in HJS events. When the extension is required for validation, producers MUST list it in ext_crit.

Fields:

profile:
: The HJS conformance profile. For this document, the value is "HJS-Core-1".

record_type:
: The type of object referenced by the JEP what field. Values include "hjs-behavior-record", "hjs-receipt-manifest", and "hjs-validation-report".

record_digest:
: The algorithm-tagged digest string of the referenced record. This value SHOULD equal the JEP what field when the JEP what field is used for the same object.

media_type:
: The media type of the referenced object. The value "application/json" is used for JSON behavior records and receipt manifests.

Example:

```
"ext": {
  "https://hjs.org/receipt": {
    "profile": "HJS-Core-1",
    "record_type": "hjs-behavior-record",
    "record_digest": "sha256:3a6eb0790f39ac87c94f3856b2dd2c5d110e6811602261a9a923d3
bb23adc8b7",
    "media_type": "application/json"
  }
},
"ext_crit": ["https://hjs.org/receipt"]
```

4. HJS Behavior Records

4.1. General Requirements

An HJS behavior record is a JSON object describing observable AI-agent behavior. When a behavior record is referenced by a JEP event, the digest in the JEP what field MUST be computed over the behavior record using the digest rules in Section 4.5.

HJS behavior records SHOULD avoid plaintext personally identifiable information. Human participant references SHOULD use pseudonymous, salted digest, opaque, rotating, or withheld identifiers unless a deployment profile requires direct identifiers.

A behavior record MUST NOT include the event hash of the JEP event that signs it unless a profile explicitly defines a non-circular construction. This avoids circular dependencies between the record digest and the event hash.

4.2. Required Fields

HJS-Core-1 behavior records MUST contain the following fields:

hjs_record:
: The HJS behavior record format version. This document defines value "1".

record_type:
: The string "behavior".

agent:
: A JSON object describing the AI agent or AI runtime. It MUST contain an id field. It MAY contain role, deployment_id, or other profile-defined fields.

action:
: A JSON object describing the observed action. It MUST contain a type field. Examples include "decision", "tool_call", "delegation", "verification", "termination", "escalation", and "content_generation".

created_at:
: A Unix timestamp in seconds indicating when the behavior record was created.

evidence:
: A JSON object containing evidence descriptors. See Section 4.3.

HJS-Core-1 behavior records MAY contain the following fields:

* model: Model or deployment descriptor.
* policy_checks: Array of policy-check descriptors.
* human_participants: Array of human participant references.
* risk: Risk descriptor.
* context: Deployment-defined context descriptor.
* redaction: Redaction and minimization descriptor.
* external_refs: Deployment-defined references to optional external materials.

4.3. Evidence Descriptors

Evidence descriptors identify content or metadata that supports audit

or review workflows. HJS evidence descriptors SHOULD be digest-addressed. A descriptor MAY include a URI, but validation MUST NOT depend solely on dereferencing the URI.

A descriptor SHOULD contain:

kind:
: The kind of evidence, such as "input", "output", "prompt", "completion", "tool_request", "tool_response", "policy_result", or "external_context".

digest:
: An algorithm-tagged digest string for the evidence object.

media_type:
: The media type of the evidence object, if known.

uri:
: Optional storage or retrieval location.

redaction:
: Optional redaction status, such as "none", "partial", "digest-only", or "withheld".

4.4. Human Participant References

Human participant references SHOULD be represented as structured objects rather than as plaintext identifiers in the JEP who field.

A human participant reference MAY contain:

role:
: The participant role, such as "user", "reviewer", "operator", "subject", or "approver".

reference_type:
: The identifier type, such as "opaque", "did", "public_key_hash", "salted_digest", or "withheld".

reference:
: The participant reference value. If reference_type is "salted_digest", the reference SHOULD be an algorithm-tagged digest string.

privacy_mode:
: A value such as "plaintext", "pseudonymous", "rotating", "digest-only", or "withheld".

salt_holder:
: Optional identifier of a salt holder or escrow authority. This field does not by itself prove that the salt holder is trustworthy.

HJS does not guarantee that pseudonymous or rotating identifiers are unlinkable against all observers. Timing, network metadata, device identifiers, content similarity, or deployment context can create correlation risks.

4.5. Behavior Record Digest

When an HJS behavior record is a JSON object, its digest is computed over the UTF-8 octets of the JCS-canonicalized behavior record unless a deployment profile specifies another canonicalization profile.

The textual digest representation MUST use the algorithm-tagged digest string format defined by JEP. HJS-Core-1 implementations MUST support sha256 behavior record digests.

5. HJS Receipt Manifests and Bundles

5.1. Receipt Manifest Structure

An HJS receipt manifest is an optional JSON object that packages the components needed to validate an HJS receipt bundle. A manifest is useful when an HJS receipt spans multiple JEP events, behavior records, external evidence objects, optional external references, or storage locations.

An HJS-Core-1 receipt manifest SHOULD contain:

hjs_receipt:
: Receipt manifest format version. This document defines value "1".

profile:
: HJS profile identifier, such as "HJS-Core-1".

root_event:
: Event hash of the root JEP event in the receipt bundle.

events:
: Array of event descriptors, each containing an event_hash and optionally a URI or inline event.

records:
: Array of behavior record descriptors, each containing a digest, media_type, and optionally a URI or inline object.

evidence:
: Array of external evidence descriptors.

external_refs:
: Array of optional participant-supplied, review, correction, contestation, or explanation-material reference descriptors.

created_at:
: Unix timestamp for manifest creation.

The manifest itself MAY be referenced by a JEP event through the JEP what field. If so, its digest is computed according to the same behavior record digest rules for JSON objects.

5.2. Receipt Bundle

An HJS receipt bundle is a packaging concept. It MAY contain JEP events, HJS behavior records, HJS receipt manifests, external evidence, optional validation reports, and optional external references. HJS does not define a mandatory archive format in this version. Deployment profiles MAY define a ZIP, CBOR, JSON, or other packaging format, provided that JEP event signatures and digest values remain verifiable without trusting the packaging layer.

5.3. Optional Multi-Party Exportable Receipt Bundles

HJS supports optional exportable receipt bundles. A deployment profile MAY allow multiple parties to export the same receipt bundle, different views of the same receipt bundle, or separately signed receipt fragments. Such exports are technical evidence packages. HJS does not determine which party is entitled to export, receive, rely on, disclose, or withhold a receipt bundle.

When multi-party export is used, each exported bundle SHOULD preserve the verifiability of included JEP event signatures and digest bindings. Redaction, minimization, and privacy controls MAY be applied, provided that the remaining digest and signature relationships are not misrepresented.

HJS does not require all parties to receive identical information. HJS also does not define procedural fairness, discovery rights, access rights, evidentiary privilege, confidentiality duties, legal admissibility, or entitlement to export.

6. HJS Optional Extensions

6.1. Extension Principles

HJS extensions use the JEP ext/ext_crit framework. Unless an extension is listed in ext_crit, a verifier that does not understand the extension MAY ignore it according to JEP rules. If an HJS extension is listed in ext_crit, the verifier MUST understand and process that extension or reject the event.

HJS extensions provide technical mechanisms for referencing, binding, exporting, minimizing, or validating external materials. They do not define the meaning, sufficiency, truth, fairness, legality, moral character, governance consequence, or evidentiary effect of those materials.

6.2. Human Participant Privacy Extension

Identifier: https://hjs.org/human

Purpose: Carries privacy-preserving references to human participants when such references need to be associated with a JEP event itself rather than only with the HJS behavior record.

Example value:

```
{
  "participants": [
    {
      "role": "user",
      "reference_type": "salted_digest",
      "reference": "sha256:8b39f3c7d5e9a1f2aabbccddeeff00112233445566778899aabbccdd
eeff00",
      "privacy_mode": "digest-only",
      "salt_holder": "did:example:hjs-escrow"
    }
  ]
}
```

6.3. Identity Rotation Extension

Identifier: https://hjs.org/identity-rotation

Purpose: Describes rotation of pseudonymous or opaque identifiers used for human participant references. Fields MAY include rotation_epoch, previous_reference_digest, next_reference_digest, and rotation_policy. Identity rotation can reduce linkability but does not erase previously signed events.

6.4. Participant-Supplied and External Process References

Identifier: https://hjs.org/external-refs

This extension provides a mechanism for attaching integrity-protected references to externally supplied materials, including participant statements, correction records, dispute records, post-event review requests, review materials, explanation requests, explanation materials, or other deployment-defined records.

These references are technical pointers only. HJS does not define the content, meaning, sufficiency, truth, legal effect, moral character,

governance consequence, or evidentiary effect of the referenced materials.

The rel values used by this extension are deployment-defined unless an external profile gives them additional meaning. HJS does not assign normative meaning to rel values.

The absence of a participant-supplied reference MUST NOT be interpreted by the protocol as agreement, waiver, admission, lack of objection, absence of relevant context, or absence of external rights.

Example value:

```
{
  "profile": "https://example.org/external-process-profile-v1",
  "refs": [
    {
      "rel": "participant-statement",
      "ref": "sha256:aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aa"
    },
    {
      "rel": "post-event-review-material",
      "ref": "sha256:bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
bb"
    }
  ]
}
```

6.5. Explanation Material References

Identifier: https://hjs.org/explanation-ref

This extension provides a mechanism for referencing explanation requests and explanation materials associated with an HJS behavior record, receipt manifest, receipt bundle, or JEP event.

HJS does not determine whether an explanation is legally required, who is entitled to receive it, whether the explanation is sufficient, clear, meaningful, timely, complete, or compliant with any legal, organizational, human-rights, or governance framework.

An explanation reference proves only the existence of a signed or digest-bound reference to external material. The interpretation and legal effect of that material are outside HJS.

Example value:

```
{
  "profile": "https://example.org/explanation-process-v1",
  "request_ref": "sha256:cccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
cccccc",
  "response_ref": "sha256:ddddddddddddddddddddddddddddddddddddddddddddddddddddddddd
ddddddd",
  "related_behavior_record": "sha256:5555555555555555555555555555555555555555555555
555555555555555555",
  "redaction_profile": "https://example.org/redaction-profile-v1"
}
```

6.6. Multi-Party Export Extension

Identifier: https://hjs.org/multiparty-export

Purpose: Describes optional export views or separately signed receipt fragments for different parties. This extension supports deployments that provide different privacy-preserving views of the same underlying receipt bundle.

Fields MAY include export_view, view_digest, redaction_profile, recipient_ref, fragment_ref, and export_profile. HJS does not define access rights, disclosure duties, legal admissibility, or entitlement to export.

6.7. Risk Extension

Identifier: https://hjs.org/risk

Purpose: Carries deployment-defined risk classification metadata. Fields MAY include level, taxonomy, and rationale_digest. HJS does not define universal risk semantics. A verifier MUST NOT assume that risk levels from different taxonomies are equivalent.

6.8. Model Evidence Extension

Identifier: https://hjs.org/model

Purpose: Carries model or deployment descriptors when those descriptors need to be attached to the JEP event. Fields MAY include provider, model_id, deployment_id, version_identifier, model_card_digest, and configuration_digest. HJS does not require disclosure of proprietary model weights or secrets.

6.9. Tool Call Evidence Extension

Identifier: https://hjs.org/tool-call

Purpose: Carries digest-addressed descriptors for tool calls made by an AI agent. Fields MAY include tool_name, tool_provider, request_digest, response_digest, authorization_context_digest, and result_status. HJS does not determine whether a tool call was authorized; it records evidence relevant to that question.

6.10. Policy Check Evidence Extension

Identifier: https://hjs.org/policy-check

Purpose: Carries policy-check evidence associated with an AI-agent behavior event. Fields MAY include policy_id, policy_version, result, evaluator, rationale_digest, and evidence_digest. HJS does not determine whether a policy is legally sufficient.

6.11. Use of JEP Cryptographic Profile Extension

HJS implementations MAY use the JEP Cryptographic Profile Extension to declare issuer-level signature capability, canonicalization profile, and hash family. HJS does not define a separate cryptographic capability model.

In particular, post-quantum or composite signature capability is a property of the issuer, signing identity, trust profile, or deployment profile. It is not an HJS event type and MUST NOT multiply the JEP verb set.

7. Validation

7.1. JEP Validation

Before HJS-specific validation, a verifier MUST perform the applicable JEP validation mode:

* JEP acceptance validation for newly received events.
* JEP archival validation for previously recorded events.

HJS validation MUST NOT treat an event as valid if the underlying JEP event fails the required JEP validation mode.

7.2. HJS Receipt Validation

To validate an HJS receipt, a verifier MUST:

1. Validate the relevant JEP event or event chain according to JEP.
2. Confirm that the event contains the HJS Receipt Extension or is otherwise identified by a deployment profile as an HJS event.
3. Confirm that the extension profile is supported.
4. Obtain the referenced behavior record, receipt manifest, validation report, or other referenced object.
5. Recompute the digest of the referenced object.
6. Confirm that the recomputed digest matches the JEP what field or the record_digest field in the HJS Receipt Extension, as applicable.
7. Process all HJS extensions listed in ext_crit.
8. Apply local evidence, trust, privacy, and policy rules.

Successful HJS receipt validation proves that the signed JEP event and referenced HJS object are cryptographically bound under the applicable trust profile. It does not prove that the referenced content is true, complete, legally sufficient, fair, or causally determinative.

7.3. Behavior Record Validation

A verifier validating an HJS behavior record SHOULD verify:

* Required fields are present.
* The record digest matches the JEP what field or receipt extension.
* Evidence descriptors use valid digest formats.
* Human participant references comply with the deployment privacy profile.
* Model, tool, policy, and external-reference descriptors are internally consistent where required by the deployment profile.

HJS behavior record validation is structural and evidentiary. It is not a legal, forensic, scientific, or human-rights determination that the recorded behavior is complete or that the underlying AI system behaved correctly.

7.4. Optional Extension Validation

Optional extension validation is deployment-specific. If an optional extension is present but not listed in ext_crit, a verifier MAY ignore it according to JEP rules. If it is listed in ext_crit, the verifier MUST either process it according to the applicable extension and deployment profile or reject the event.

Processing an optional extension does not give HJS authority to determine the legal, moral, governance, or evidentiary effect of that extension.

7.5. Exported Bundle Validation

A verifier validating an exported HJS receipt bundle SHOULD verify that each included JEP event signature, event hash, behavior record digest, manifest digest, and external reference digest remains valid after any redaction or view generation.

A redacted or partial export MUST NOT misrepresent omitted material as nonexistent, unchanged, irrelevant, agreed, waived, or verified. Deployment profiles MAY define redaction notices, withheld markers, or view descriptors to avoid such misrepresentation.

8. Security and Privacy Considerations

8.1. Signature and Chain Integrity

HJS inherits signature and chain integrity from JEP. HJS signatures are JEP signatures. HJS does not define an independent signature string such as "Ed25519:<signature>". Producers MUST use the JEP signature container and signing input rules.

A valid HJS receipt shows that a valid JEP event was signed by a key resolved under an applicable trust profile and that the event is bound to the referenced HJS object by digest. This is a tamper-evidence property, not a guarantee of physical immutability.

8.2. HJS Does Not Prove Truth, Intent, or Liability

HJS receipt validation proves cryptographic binding and structural consistency. It does not prove:

* that the behavior record is complete;
* that the AI agent’s internal reasoning or subjective motivation was fully captured;
* that the output was correct;
* that a delegation or tool call was authorized;
* that a human participant consented;
* that a legal duty was satisfied;
* that any party is liable or not liable;
* that a process was fair; or
* that an explanation, review, or appeal material is sufficient.

8.3. Limits of Behavioral Determination under Partial Observation

HJS is designed for audit evidence under partial observation. In many real systems, logs, traces, explanations, measurements, and receipts are projections of a larger causal history. Multiple real histories may be consistent with the same observed log while differing on a target fact of interest. This structural limitation is discussed in [TARGET-DETERMINABILITY].

Therefore, HJS supports accountability workflows but does not replace domain-specific evidence policies, incident response procedures, scientific validation, forensic methods, human-rights due diligence, or legal determinations.

8.4. Human Privacy and Linkability

HJS supports pseudonymous, rotating, digest-only, opaque, and withheld human participant references. These mechanisms can reduce unnecessary exposure of personal data. They do not guarantee

unlinkability against all observers.

Linkability risks can arise from timing, network metadata, repeated content, device identifiers, account context, tool-call patterns, storage locations, or salt reuse. Salted digest schemes are vulnerable to dictionary attacks when the underlying identifier has low entropy or the salt is disclosed, reused, or poorly protected.

HJS implementations SHOULD minimize plaintext personal data and SHOULD use digest-only, pseudonymous, opaque, rotating, or withheld references where appropriate.

8.5. Participant-Supplied References and Non-Inference

HJS MAY provide mechanisms for attaching, referencing, and verifying externally supplied materials. These mechanisms preserve a technical relationship between a receipt and external material; they do not define the meaning or effect of that material.

HJS does not determine intent, negligence, fault, good faith, coercion, reasonableness, consent, harm, liability, appeal validity, explanation sufficiency, or compliance. Such determinations, if any, are outside the protocol and belong to external legal, organizational, human-rights, or governance processes.

The absence of a participant-supplied reference MUST NOT be interpreted by the protocol as agreement, waiver, admission, lack of objection, lack of harm, absence of relevant context, absence of external rights, or absence of an available external process.

8.6. Governance Neutrality and Human Rights-Supporting Use

HJS is a technical evidence and receipt profile. It is designed to support rights-respecting deployments by minimizing unnecessary exposure of human participant identity and by preserving verifiable records of AI-agent behavior. HJS does not impose monitoring duties, prescribe governance outcomes, determine consent validity, assign liability, define sanctions, or establish legal compliance.

Deployments that use HJS remain responsible for their own human-rights, privacy, labor, safety, and regulatory obligations. HJS records can support such processes, but they do not replace human-rights due diligence, access to remedy, organizational governance, or legal review.

8.7. Regulatory Support, Not Regulatory Determination

HJS can support regulatory, audit, data-minimization, incident review, explanation-material reference, and transparency workflows by providing portable, verifiable evidence. HJS does not determine legal compliance, lawful basis, data-subject rights fulfillment, consent validity, audit sufficiency, or liability.

Terms such as "risk", "policy", "review", "approval", "explanation", "appeal", or "verification" in HJS records are technical descriptors unless an external legal or organizational framework gives them additional meaning.

8.8. Model, Tool, and Policy Evidence Limitations

Model identifiers, tool-call descriptors, policy-check records, and external references can be incomplete, redacted, or deployment-specific. Verifiers SHOULD evaluate whether referenced evidence is sufficient for their audit question. A digest of a model descriptor does not prove that the model behaved as described. A policy-check result does not prove that the policy is adequate. A tool-call record

does not prove that the tool result is true or safe.

9. IANA Considerations

9.1. HJS Extension Identifier Registrations

This document does not request creation of a separate HJS Extensions Registry. HJS-specific extensions are intended to be registered in the JEP Extensions Registry defined by [I-D.wang-jep-judgment-event-protocol], if that registry is created.

Initial HJS extension identifiers requested for registration are:

* https://hjs.org/receipt
* https://hjs.org/human
* https://hjs.org/identity-rotation
* https://hjs.org/external-refs
* https://hjs.org/explanation-ref
* https://hjs.org/multiparty-export
* https://hjs.org/risk
* https://hjs.org/model
* https://hjs.org/tool-call
* https://hjs.org/policy-check

Extension identifiers are stable identifiers and are not required to be dereferenceable.

9.2. No Separate HJS Risk Level Registry

This document does not request creation of an HJS Risk Level Registry. Risk levels are values inside the HJS Risk Extension and are meaningful only relative to the declared risk taxonomy. Deployments that require stable risk taxonomies MAY define them in separate specifications.

10. References

10.1. Normative References

[I-D.wang-jep-judgment-event-protocol]
Wang, Y., "Judgment Event Protocol (JEP): A Minimal Verifiable Log Format for Agent Decisions", Work in Progress, Internet-Draft, draft-wang-jep-judgment-event-protocol-05, April 2026.

[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>.

[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>.

10.2. Informative References

[TARGET-DETERMINABILITY]
Wang, Y., "Target Determinability under Partial Causal

Observation: A Faithful Reduction Framework", Zenodo, Version v1,
DOI 10.5281/zenodo.19678205, April 2026,
<https://doi.org/10.5281/zenodo.19678205>.

Appendix A. Non-Normative Examples

A.1. HJS Behavior Record Example

The following example is illustrative. Digest values are example values and are not test vectors.


```
{
  "hjs_record": "1",
  "record_type": "behavior",
  "agent": {
    "id": "did:example:agent-789",
    "role": "planner",
    "deployment_id": "agent-runtime-42"
  },
  "created_at": 1743398400,
  "action": {
    "type": "tool_call",
    "name": "calendar.create_event"
  },
  "model": {
    "provider": "example-ai",
    "model_id": "example-model-family",
    "deployment_id": "example-deployment"
  },
  "evidence": {
    "inputs": [
      {
        "kind": "prompt",
        "digest": "sha256:1111111111111111111111111111111111111111111111111111111111111111
1111",
        "media_type": "text/plain",
        "redaction": "digest-only"
      }
    ],
    "outputs": [
      {
        "kind": "tool_request",
        "digest": "sha256:2222222222222222222222222222222222222222222222222222222222222222
2222",
        "media_type": "application/json",
        "redaction": "partial"
      }
    ]
  },
  "policy_checks": [
    {
      "policy_id": "example-policy-1",
      "policy_version": "2026-04",
      "result": "pass",
      "rationale_digest": "sha256:3333333333333333333333333333333333333333333333333333333333
333333333333"
    }
  ],
  "human_participants": [
    {
      "role": "user",
      "reference_type": "salted_digest",
      "reference": "sha256:4444444444444444444444444444444444444444444444444444444444444444
44444",
      "privacy_mode": "digest-only",
      "salt_holder": "did:example:hjs-escrow"
```

```
      }
    ],
    "risk": {
      "level": "medium",
      "taxonomy": "hjs-risk-v1"
    }
  }
```

A.2. JEP Event Carrying an HJS Receipt Extension

The following example shows a JEP event that references the HJS behavior record by digest. The sig value is illustrative and truncated.

```
  {
    "jep": "1",
    "verb": "J",
    "who": "did:example:agent-789",
    "when": 1743398400,
    "what": "sha256:5555555555555555555555555555555555555555555555555555555555555555",
    "nonce": "84d8c175-7b03-4b8d-9d27-1234abcd5678",
    "aud": "https://platform.example.com",
    "ref": null,
    "ext": {
      "https://hjs.org/receipt": {
        "profile": "HJS-Core-1",
        "record_type": "hjs-behavior-record",
        "record_digest": "sha256:5555555555555555555555555555555555555555555555555555555555555555
555555555",
        "media_type": "application/json"
      },
      "https://hjs.org/risk": {
        "level": "medium",
        "taxonomy": "hjs-risk-v1"
      }
    },
    "ext_crit": ["https://hjs.org/receipt"],
    "sig": "eyJhbGciOiJFZDI1NTE5Iiwia2lkIjoiZGlkOmV4YW1wbGU6YWdlbnQtNzg5I2tleS0xIn0..exa
mple-detached-signature"
  }
```

A.3. HJS Receipt Manifest Example

```
  {
    "hjs_receipt": "1",
    "profile": "HJS-Core-1",
    "created_at": 1743398402,
    "root_event": "sha256:6666666666666666666666666666666666666666666666666666666666666666
66",
    "events": [
      {
        "event_hash": "sha256:6666666666666666666666666666666666666666666666666666666666666666
666666",
        "uri": "https://evidence.example/events/6666"
      }
    ],
    "records": [
      {
        "digest": "sha256:5555555555555555555555555555555555555555555555555555555555555555
55",
        "media_type": "application/json",
        "uri": "https://evidence.example/records/5555"
      }
    ],
    "evidence": [],
    "external_refs": []
```

```
   }
```

A.4. External References and Explanation References Example

```
   {
     "https://hjs.org/external-refs": {
       "profile": "https://example.org/external-process-profile-v1",
       "refs": [
         {
           "rel": "participant-statement",
           "ref": "sha256:aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
a"
         },
         {
           "rel": "correction-record",
           "ref": "sha256:bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
b"
         }
       ]
     },
     "https://hjs.org/explanation-ref": {
       "profile": "https://example.org/explanation-process-v1",
       "request_ref": "sha256:ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
ccccc",
       "response_ref": "sha256:dddddddddddddddddddddddddddddddddddddddddddddddddddddddddd
dddddd",
       "related_behavior_record": "sha256:55555555555555555555555555555555555555555555555
55555555555555555"
     }
   }
```

A.5. Multi-Party Export View Example

```
   {
     "https://hjs.org/multiparty-export": {
       "export_profile": "https://example.org/export-profile-v1",
       "export_view": "participant-redacted-view",
       "view_digest": "sha256:eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee
eeeee",
       "redaction_profile": "https://example.org/redaction-profile-v1",
       "recipient_ref": "sha256:fffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffff"
     }
   }
```

Appendix B. Changes from draft-wang-hjs-accountability-04

This revision makes the following non-normative structural changes:

* Repositions HJS as a minimal profile of JEP v0.5 rather than a separate event signing protocol.

* Removes independent HJS definitions of signature syntax, canonicalization, event hashes, nonce replay handling, and verification windows. These are inherited from JEP.

* Clarifies that the JEP who field identifies the event issuer, not necessarily a human participant or decision subject.

* Replaces the prior HJS event model with HJS behavior records, HJS receipt manifests, and HJS receipt bundles.

* Adds the Minimal Core and Optional Extensions principle.

* Adds participant-supplied and external process references as optional technical pointers.

* Adds optional explanation-material references without defining explanation rights or sufficiency.

* Adds optional multi-party exportable receipt bundle support without defining access rights, disclosure duties, or evidentiary effect.

* Replaces regulatory compliance claims with regulatory support and technical neutrality language.

* Adds limits of behavioral determination under partial observation as an informative security consideration.

* Removes the separate HJS Risk Level Registry request. Risk levels are now values within the HJS Risk Extension and are interpreted according to a declared taxonomy.

* Aligns HJS-specific extension handling with the JEP ext/ext_crit framework.

Author’s Address

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