RANKIGI × EVIDE

Interface Mapping Update

HACE Integration + EVIDE v2.0 Boundary Readiness

Addendum to v0.3 · May 2026

Published with consent of both authors

`HACE canonical spec: https://hace-standard.org/spec/v1`

`EVIDE reference: https://app.certifywebcontent.com/json`

Authors

Wesley Snow

Founder, RANKIGI - Independent Execution Proof Layer for AI Agents

Emanuel Celano

Protocol </AI> Founder | EVIDE – The Missing Evidentiary Layer for AI | CertifyWebContent.com | DAPI-Certification.com | Digital Evidence & AI Governance

Scope of this update

This addendum extends the v0.3 interface mapping to reflect two structural changes introduced after v0.3 was published:

- EVIDE v2.0 promotes boundary_readiness from a string to a structured object
- RANKIGI HACE v1.0 introduces a formal attestation protocol between the two systems

All v0.3 field mappings remain valid. This document defines the delta only.

---

1. Updated Field Mapping — boundary_readiness (v2.0)

In v0.3, boundary_readiness was mapped as a simple string:

> *handoff.boundary_readiness → string: "candidate" | "verified" | "not_assessed"*

In v2.0, boundary_readiness is promoted to a structured object. The complete updated mapping is:

RANKIGI / HACE field	EVIDE v2.0 field
`chain verification result`	`handoff.boundary_readiness.status`
`"candidate"`	`no independent gate operated`
`"verified"`	`gate confirmed stability, complete visibility, no signals`

RANKIGI / HACE field	EVIDE v2.0 field
"verified_partial"	gate confirmed partial surface, unresolved signals declared
"unverifiable"	gate operated, visibility insufficient
RANKIGI system identifier	handoff.boundary_readiness.readiness_gate.identifier
RANKIGI gate policy URL/hash	handoff.boundary_readiness.readiness_gate.scope_reference
"declared_complete"	handoff.boundary_readiness.visibility_surface
"partial"	handoff.boundary_readiness.visibility_surface
"insufficient"	handoff.boundary_readiness.visibility_surface
null (candidate)	handoff.boundary_readiness.visibility_surface = null
RANKIGI unresolved signals	handoff.boundary_readiness.unresolved_signals[]
[] (verified)	handoff.boundary_readiness.unresolved_signals = []
["signal_id"] (partial/unverifiable)	handoff.boundary_readiness.unresolved_signals (min. 1)

Valid state combinations (v2.0)

status	readiness_gate	visibility_surface	unresolved_signals	Valid
candidate	null	null	[]	✓
verified	present	declared_complete	[]	✓
verified_partial	present	partial	[min. 1]	✓
unverifiable	present	insufficient	[min. 1]	✓
candidate	present	any	any	✗
verified	null	any	any	✗
verified	present	any	[min. 1]	✗
verified_partial	present	any	[]	✗
unverifiable	present	any	[]	✗

Architectural note on self-certification

RANKIGI cannot self-certify boundary_readiness.status = "verified" for its own execution chain. The independence requirement is satisfied because RANKIGI is the gate for EVIDE's intake, not for its own output. The producing system and the gate are separate systems.

> **MUST: A system cannot evaluate its own boundary readiness and declare it verified. That evaluation must come from a gate that is independent of the system that produced the decision.**

2. HACE Attestation Layer

HACE (Human Acknowledgment of Chain Execution) introduces a formal attestation protocol that sits between RANKIGI snapshot generation and EVIDE intake. It creates a third layer in the integration stack:

-- RANKIGI → produces daily snapshot + signed exception manifest
-- HACE → human officer reviews and signs over canonical bytes
-- EVIDE → receives signed attestation, updates closure integrity object
-- FEDIS → packages the full evidentiary record for legal output

2.1 Wire format — two independent layers

The EVIDE-side attestation package is structured in two distinct layers. These layers must never be merged into a single signed structure.

Layer A — Signed Acknowledgment Package

This is the signed evidentiary object submitted to RANKIGI via POST /api/v1/attestations:

```
{  "snapshot_id": "uuid",
  "org_id": "uuid",
  "provider_name": "evide",
  "provider_attestation_id": "EVD-YYYY-MM-DD-NNNNNN",
  "candidate_hash": "sha256hex — must match RANKIGI snapshot_hash exactly",
  "attestation_payload": "base64 — canonical JSON bytes from RANKIGI GET
endpoint",
  "attestation_signature": "base64 — Ed25519 signature over decoded payload
bytes",
  "attestation_signature_alg": "Ed25519",
  "signer_certificate_chain": ["base64 DER leaf", "base64 DER
intermediate"],
  "signer_identity_claim": { "display_name": "...", "email": "...",
"subject_dn": "...", "external_id": "..." },
  "identity_assurance_level": "NIST-IAL2",
  "authentication_assurance_level": "NIST-AAL2",
  "review_scope": { "filters_applied": [], "total_chains_in_period": 0,
"exceptions_surfaced": 0, "exceptions_individually_reviewed": 0 },
  "attestation_timestamp": "RFC3339 UTC Z ms precision",
  "signature_meaning": "RESPONSIBILITY",
  "server_challenge": "nonce from RANKIGI candidate response — must be
inside signed bytes",
  "plug_version": "1.0",
  "plug_conformance_profile":
"https://rankigi.com/hace/conformance/baseline-v1"
}
```

Layer B — EVIDE evidentiary_profile

This is server-computed by EVIDE after intake. It is not part of the signed payload and is never included in any hash:

```
{
  "profile_version": "1.0",
  "identity": "claimed | declared",
  "authority": "declared | null",
  "classification": "stable | provisional | contested",
  "threshold": "met | not_met | unknown | not_defined",
  "threshold_authority": "attributed | fragmented | implicit | unknown",
  "boundary_readiness": "candidate | verified | verified_partial |
unverifiable",
  "runtime_visibility": "confirmed | partial | unverifiable | null",
  "trace_reference": "available | restricted | unavailable",
  "continuity": null  // pending v2.x Gate Qualification Framework
}
```

> **CRITICAL: Layer A and Layer B must remain structurally separate. The signed acknowledgment proves that EVIDE received and reviewed the exact RANKIGI candidate bytes. The evidentiary_profile exposes EVIDE's interpretive reading of what was observable. Merging them would collapse the separation between evidentiary preservation and evidentiary interpretation.**

2.2 HACE field mapping — RANKIGI → EVIDE

RANKIGI / HACE field	EVIDE v2.0 field
GET /api/v1/snapshots/{id}/candidate	attestation_payload (sign raw bytes as received)
snapshot_hash from candidate response	candidate_hash (must match exactly)
server_challenge from candidate response	server_challenge (must be inside signed bytes)
201 response: chain_event_hash	store — RANKIGI cryptographic commitment
closure integrity object update	handoff.submission_status → "attested"
closure integrity object update	handoff.acceptance_status → "verified"
attestation_ref in EVIDE handoff block	populated automatically at read time

2.3 Signing rules

```
Algorithm:     Ed25519 (first implementation)
```

```
                ML-DSA-65, RSASSA-PSS-SHA256, ECDSA-P256, ECDSA-P384 also
supported
Sign over:      raw decoded bytes of attestation_payload — do not re-
canonicalize
Certificate:    leaf at index 0, intermediates at index 1+, base64 DER
IAL floor:      NIST-IAL2 minimum for regulated use cases
AAL floor:      NIST-AAL2 minimum, step-up required at moment of signing
Timestamp:      YYYY-MM-DDTHH:MM:SS.mmmZ — millisecond precision, UTC Z
suffix
Replay:         server_challenge nonce prevents replay of any captured
attestation
```

2.4 Non-signed metadata block — evidentiary_profile schema reference

EVIDE's evidentiary_profile carries its own stable schema version, versioned independently from the intake schema, the handoff object, and the signed submission payload. This allows interpretive evidentiary dimensions to evolve without mutating the original object that crossed the boundary.

The HACE attestation object can optionally include a non-signed metadata block alongside the signed payload. This block references the evidentiary_profile schema URI so that a downstream verifier knows which interpretive framework was applied without that reference being part of the signed canonical bytes:

```
// Non-signed metadata block (outside signed attestation_payload)
{
  "evide_metadata": {
    "evidentiary_profile_schema_uri":
"https://app.certifywebcontent.com/docs/evide-intake-schema/",
    "profile_version": "1.0",
    "interpretation_model": "evide-v2.0"
  }
}
```

> *This block is explicitly non-signed. It does not alter the canonical payload bytes. It preserves: fixed evidentiary preservation, evolving interpretive semantics, and independent verification boundaries.*

> **MUST: The evide_metadata block must never be included inside attestation_payload before signing. It must sit alongside the signed object as separate non-committed metadata.**

---

3. Updated Conclusion

The complete integration stack now has four layers:

```
RANKIGI / KYA   → execution substrate, identity layer, boundary readiness gate
HACE            → human acknowledgment of chain execution, mutual binding loop
EVIDE           → responsibility closure layer, evidentiary_profile computation
FEDIS           → evidentiary admissibility layer, legal output packaging
```

The closure object sits alongside the KYA chain, not inside it.

The HACE attestation sits between RANKIGI snapshot generation and EVIDE intake.

The evidentiary_profile sits outside the signed payload — computed, not submitted.

The evide_metadata block sits outside the signed bytes — referenced, not committed.

HACE canonical spec: https://hace-standard.org/spec/v1

HACE spec repo: https://github.com/hace-standard/spec

Four layers. Three anchors. One independent record.

Three independent operations.

Three independent records.

None of them mutate the others.

This addendum defines the expanded bridge.