GetEvidex — ERF Conformance Whitepaper (v1.0)

Purpose: describe how ERF receipts are verified deterministically, what conformance means, and what it does NOT claim.

1. What ERF is

ERF (Evidex Receipt Format) is a portable receipt object used to represent a tamper■evident integrity seal over a content hash. It is designed to be verified without accounts and without user tracking.

2. Verification model

Input	Deterministic checks
Receipt structure	Required fields exist; types match; canonical constraints hold.
Signature	Signature verifies against the issuer public key (Ed25519/JWS or equivalent envelope).
Hash match (optional)	If a client provides SHA■256, verify receipt.hash == client_hash.
Status logic	Return Final / Provisional / Unknown using strict reason codes.

3. Conformance

A conformant implementation produces the same validation result on the same input across environments (browser/server) and passes the published test vectors.

Recommended checks: canonical JSON rules, strict field validation, deterministic reason codes, and no remote script dependencies in public surfaces.

4. Reality Audit (hard lock)

ERF verification provides integrity evidence only (tamper■evident). It does not prove identity, authorship, ownership, intent, delivery/condition, or legal admissibility.

This does not prove identity or legal admissibility.

5. Adoption & interoperability

To support ecosystem adoption, GetEvidex publishes: a discovery document (/.well-known/evidex), media types, registries (profiles, algorithms, reason codes), and a conformance runner.

References

- Open Receipt Standard page: /open-receipt-standard.html
- ERF Standard: /docs/ERF_STANDARD.md
- Test vectors: /docs/TEST_VECTORS.md
- Registries: /docs/REGISTRIES.md