Network Working Group
Internet-Draft
Intended status: Informational
Expires: 28 March 2026

L. J. Reilly
Independent

September 2025

Reilly Government Integrity Protocol (RGIP): DOI-Archived and Blockchain-Timestamped Framework for Permanent Public Records

draft-reilly-government-integrity-00

Abstract

The Reilly Government Integrity Protocol (RGIP) defines a standards-aligned method to create tamper-evident, citable public records by combining content hashing, public-blockchain anchoring, and DOI-based archival. RGIP introduces the Evidence Receipt (ER), a signed, canonicalized metadata object that binds a record's hash, a persistent identifier (DOI), and a blockchain transaction reference. The protocol specifies formats, processing steps, and verification procedures using existing IETF and related standards.

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 28 March 2026.

Table of Contents

1. Introduction
2. Terminology
3. Protocol Goals and Non-Goals
4. High-Level Overview
5. Evidence Receipt (ER) Data Model
6. Step-by-Step Implementation (Normative)
7. Verification Procedure (Normative)
8. Operational Considerations
9. Security Considerations
10. Privacy Considerations
11. IANA Considerations

12. References
    12.1. Normative References
    12.2. Informative References
Acknowledgments
Author's Address

1. Introduction

Public institutions benefit from records that are both durable and independently verifiable. RGIP defines a lightweight, technology-neutral framework to produce permanent public records by binding:

* a canonical representation of a record (for hashing),
* a persistent identifier for archival access (DOI),
* and a public, append-only timestamp (blockchain anchor).

The protocol specifies an Evidence Receipt (ER) that encapsulates the above with a digital signature. The ER enables third parties to validate continuity between the published artifact, a DOI landing page, and an immutable blockchain transaction reference.

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.

2. Terminology

Artifact: The public record content being made permanent (e.g., PDF, JSON, CSV, image, or bundle).

Canonicalization: The deterministic transformation of an Artifact's serializable representation into a unique byte stream for hashing. RGIP uses JSON Canonicalization Scheme (JCS) [RFC8785] for JSON; other formats use a defined byte representation (see Section 6).

DOI: Digital Object Identifier resolving to an archival landing page for the Artifact.

ER: Evidence Receipt defined by this document.

ni URI: A content-hash URI per [RFC6920].

Anchor: The inclusion of a content hash or commitment in a public blockchain transaction, identified by chain id and transaction reference.

3. Protocol Goals and Non-Goals

Goals

* Verifiable binding of Artifact → hash → DOI → blockchain anchor.

* Reproducible, implementation-independent verification.

* Minimal new syntax: reuse existing IETF standards where possible.

Non-Goals

* Mandating a specific blockchain or timestamping service.

* Defining new cryptographic algorithms.

4. High-Level Overview

Producers create an ER for each Artifact by: (1) canonicalizing and hashing the Artifact, (2) minting/assigning a DOI, (3) anchoring the hash on a public blockchain, and (4) signing the ER. Verifiers retrieve the Artifact via DOI, recompute the hash, check the anchor, and verify the signature over the ER contents.

5. Evidence Receipt (ER) Data Model

The ER is a UTF-8 JSON object, canonicalized per JCS [RFC8785] for signing and hashing. The following members are REQUIRED unless marked OPTIONAL.

```
{
  "er_version": "1",
  "subject": {
    "title":        string,
    "doi":          string,          // DOI URI
    "artifact_uri": string OPTIONAL  // direct URL if available
  },
  "content": {
    "media_type":   string,          // e.g., application/pdf
    "hash_alg":     "sha-256",       // per [RFC6234]
    "hash":         string,          // hex lowercase
    "ni":           string           // ni URI per [RFC6920]
  },
  "event": {
    "event_type":   string,          // e.g., "publish" | "amend"
    "event_time":   string           // RFC 3339 timestamp [RFC3339]
  },
  "anchor": {
    "chain":        string,          // e.g., "bitcoin" | "ethereum"
    "txid":         string,          // transaction identifier
    "merkle_root":  string OPTIONAL  // if applicable
  },
  "sign": {
    "alg":          string,          // COSE alg name/number
    "cose_sign1":   string           // base64url COSE_Sign1 [RFC9052]
  }
}
```

Field Notes

* URIs MUST conform to [RFC3986]. The "ni" value MUST be an ni URI [RFC6920] encoding the content hash.

* Timestamps MUST be RFC 3339 [RFC3339].

* Signatures MUST use COSE_Sign1 [RFC9052] with a RECOMMENDED key algorithm of Ed25519 [RFC8032]. Implementations MAY support additional COSE algorithms.

6. Step-by-Step Implementation (Normative)

This section is normative.

Step 1: Prepare the Artifact

1. Choose the Artifact format. For JSON, producers MUST canonically

serialize using JCS [RFC8785]. For formats with a defined binary representation (e.g., PDF), producers MUST use the exact byte stream intended for publication.

2. Compute the content hash using SHA-256 [RFC6234]. The hex lowercase digest is used in the ER and to derive an ni URI.

3. Construct an ni URI of the form "ni:///sha-256;<digest>" [RFC6920].

Step 2: Persistent Identifier (DOI)

4. Archive the Artifact in a DOI-issuing repository. The DOI landing page SHOULD include or link to the Artifact byte stream. Record the DOI URI for the ER.

Step 3: Public Anchor

5. Anchor the SHA-256 digest or a commitment thereof in a public blockchain transaction. The exact anchoring mechanism is deployment-specific and out of scope, but the ER MUST record: chain name, transaction identifier (txid), and (if available) the anchor' s Merkle root.

6. The anchor timestamp SHOULD be independently observable through widely available blockchain explorers.

Step 4: Build the ER

7. Populate the ER fields per Section 5 with the computed hash, ni URI, DOI, event metadata, and anchor data.

Step 5: Sign the ER

8. Canonicalize the ER using JCS [RFC8785].

9. Create a COSE_Sign1 object over the JCS-canonicalized ER using Ed25519 [RFC8032] (RECOMMENDED) or another registered COSE algorithm per [RFC9052].

10. Embed the base64url encoding of COSE_Sign1 in "sign.cose_sign1". Set "sign.alg" accordingly.

Step 6: Publish

11. Publish the Artifact and the ER. The DOI landing page SHOULD link both. Producers SHOULD provide a stable "artifact_uri" if permissible.

Step 7: Recordkeeping

12. Retain signing keys securely. Maintain a registry of issued ERs with their DOIs and txids for audit and discovery.

7. Verification Procedure (Normative)

Verifiers MUST perform the following steps:

1. Resolve the DOI to obtain the Artifact (or a link to it).

2. Compute SHA-256 [RFC6234] over the obtained byte stream. Confirm that the digest matches ER.content.hash and ER.content.ni.

3. Validate the anchor by confirming that the recorded digest (or commitment) is provably included in the referenced blockchain

transaction (chain, txid, and optionally Merkle root).

4. Re-canonicalize the ER with JCS [RFC8785] and verify the COSE signature [RFC9052] using the publisher's public key (whose discovery/PKI is deployment-specific and out of scope).

5. Confirm timestamps are valid RFC 3339 and consistent with anchor observation and DOI publication time (tolerance policy is out of scope and deployment-specific).

8. Operational Considerations

* Key Management: Publishers SHOULD rotate keys periodically and publish key metadata (e.g., via their DOI landing page or another verifiable channel).

* Updates: Amendments SHOULD create a new Artifact, new ER, and a new anchor; DOIs MAY use versioning features of the repository.

* Multiple Formats: If the same content is distributed in multiple formats, each format SHOULD receive its own ER to avoid ambiguity.

* Transparency: Operators MAY maintain public catalogs of ERs to facilitate discovery and oversight.

* Interoperability: Profiles MAY restrict "chain" values, mandate specific COSE algorithms, or define PKI discovery methods to support procurement or sectoral policies.

9. Security Considerations

RGIP's assurances rely on collision-resistant hashing [RFC6234], correct canonicalization [RFC8785], and sound signature validation [RFC9052][RFC8032]. Implementations MUST protect private keys and ensure the canonical bytes hashed by producers are identical to those verified by consumers. Anchoring on a public blockchain provides an append-only timestamp; it does not, by itself, attest to semantics or legality of the Artifact. Implementers SHOULD consider replay, substitution, and downgrade risks (e.g., presenting a different file than the one hashed). Where possible, verifiers SHOULD acquire the Artifact from the DOI host over authenticated TLS and compare size and media type metadata for consistency.

10. Privacy Considerations

ERs SHOULD avoid embedding personal data. When personal data is unavoidable, publishers SHOULD minimize fields and consider pseudonymization. Anchors are public; publishers MUST NOT include sensitive data in anchor commitments. DOI landing pages SHOULD offer appropriate access controls if required by law or policy.

11. IANA Considerations

This document has no IANA actions.

12. References

12.1. Normative References

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

[RFC3339] Klyne, G., and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, July 2002.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", RFC 3986, January 2005.

[RFC6234] Eastlake 3rd, D., and T. Hansen, "US Secure Hash Algorithms (SHA and HMAC-SHA)", RFC 6234, May 2011.

[RFC6920] Farrell, S., Kutscher, D., Dannewitz, C., Ohlman, B., Keranen, A., and P. Hallam-Baker, "Naming Things with Hashes (ni)", RFC 6920, April 2013.

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

[RFC8785] Johansson, A., "The JSON Canonicalization Scheme (JCS)", RFC 8785, June 2020.

[RFC9052] Schaad, J., "CBOR Object Signing and Encryption (COSE): Structures and Process", RFC 9052, August 2022.

[RFC8032] Josefsson, S., and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)", RFC 8032, January 2017.

12.2. Informative References

[RFC3161] Adams, C., Cain, P., Pinkas, D., and R. Zuccherato, "Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)", RFC 3161, August 2001.

[RFC6962] Laurie, B., Langley, A., and E. Kasper, "Certificate Transparency", RFC 6962, June 2013.

[ZRGIP] Reilly, L. J., "The Reilly Government Integrity Protocol (RGIP): Blockchain-Timestamped and DOI-Archived Framework for Permanent Public Records", DOI:10.5281/zenodo.17114518, September 2025. (See DOI landing page for latest version and associated materials.)

Acknowledgments

The author thanks the IETF community for review and discussion.

Author’s Address

Lawrence J. Reilly
Independent

Email: lreilly250@gmail.com

(This document is work in progress.)