Independent Submission
Internet-Draft
Intended status: Informational
Expires: 27 November 2026

C. Hopley
AlgoVoi

26 May 2026

Categorical Compliance Screening Receipt Format for Agentic-Payment Flows

draft-hopley-x402-compliance-receipt-00

Abstract

This document specifies a categorical compliance screening receipt format for agentic-payment flows. The format is designed for use by AI agents and agentic-payment gateways that perform regulatory screening at admission time and must retain the screening decision under framework-bound retention obligations.

The receipt format uses a closed enumeration of categorical outcomes (ALLOW, REFER, DENY) rather than a continuous score or tier projection. The categorical outcome is load-bearing for downstream regulatory obligations: in jurisdictions with mandatory disclosure obligations triggered by specific screening outcomes, a REFER outcome can carry a mandatory reporting obligation that a DENY does not, and a score or tier projection collapses this distinction.

Receipts are canonicalised under JSON Canonicalization Scheme with an in-band canonicalisation rule pin. The pin enables long-term re-verification from retained bytes alone, without dependence on an out-of-band rule registry that the operator must continue to publish.

This document is Informational. It documents an admission-time compliance screening receipt format used in production and proposes it for wider adoption; it does not create mandatory implementation obligations on third-party systems.

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/.

Copyright Notice

Table of Contents

1. Introduction

1.1. Motivation

Agentic-payment flows are payment transactions initiated or routed by autonomous AI agents acting on behalf of a principal (a natural person or legal entity). Where these payments cross regulated payment rails, the payment gateway is obligated to perform compliance screening at admission time: sanctions screening, anti-money-laundering checks, and jurisdictional eligibility checks against the payer and the receiving counterparty.

The output of compliance screening is a categorical decision: either the payment is allowed to proceed, it is allowed but flagged for enhanced due diligence and a regulatory report, or it is refused. This document specifies a receipt format for recording that categorical decision in a manner suitable for retention under framework-bound retention obligations.

Publication as an Informational RFC provides the compliance receipt format with a stable, citeable specification that can serve as a common reference point across implementations. Standardising the categorical outcome enumeration (ALLOW / REFER / DENY), the canonicalisation discipline, and the audit-chain shape enables interoperability between independent agentic-payment gateways

performing screening under different regulatory frameworks. The format is intended to document a deployed practice rather than to establish a new standards-track mandate.

1.2. Regulatory frameworks (examples)

The receipt format is designed to satisfy retention and re-verifiability obligations imposed by a range of regulatory frameworks. Specific examples include UK Money Laundering Regulations 2017 and Proceeds of Crime Act 2002, EU Anti-Money Laundering Directives 5 and 6, EU Markets in Crypto-Assets Regulation Article 80, EU Anti-Money Laundering Regulation Article 56, and EU Digital Operational Resilience Act Article 14. In jurisdictions with mandatory disclosure obligations triggered by specific screening outcomes (for example, where a REFER outcome triggers a Suspicious Activity Report obligation that a DENY does not), the categorical enumeration preserves the operational distinction that scoring or tier projection would collapse. These frameworks are cited as examples of the design space; the receipt format itself is jurisdiction-neutral.

1.3. Scope

This document is Informational. It documents the compliance receipt format and proposes it for wider adoption; it does not define a protocol standard or create mandatory implementation obligations on third-party systems.

In scope:

* The categorical receipt format (Section 3)
* The canonicalisation rule applicable to the receipt (Section 4)
* Audit-chain composition properties (Section 5)
* Long-term auditability properties (Section 6)
* Composition with other x402 substrate specifications (Section 7)

Out of scope:

* The screening algorithm itself. The specific business logic that produces ALLOW / REFER / DENY (the sanctions feeds consulted, the anti-money-laundering rules applied, the jurisdictional risk-scoring methodology) is implementation-defined per screening provider.
* The retention storage backend. Object Lock COMPLIANCE retention on S3-compatible storage is recommended (informative, Section 5.4) but the specific backend is deployment-defined.

* Identity proofing. The payer's identity is referenced through a content-addressed reference (Section 3.1); how the identity is established and proofed is the subject of separate work.

1.4. Relationship to IETF work

This document does not conflict with any current IETF chartered work. The x402 payment protocol operates as an application-layer convention built on HTTP and does not modify or supersede any IETF protocol specification. The compliance receipt format described here is specific to admission-time screening in agentic systems and is not within the scope of any active IETF working group known to the authors at the time of submission.

1.5. Relationship to draft-vauban-x402-stark-receipts

At the time of writing, draft-vauban-x402-stark-receipts is an individual Internet-Draft on the IETF datatracker, with stream unassigned. It covers cryptographic settlement-time payment-condition proofs using STARK receipts; this document covers admission-time compliance screening receipts. The two documents were developed in coordination and are designed to compose, but each is an independent submission. Both documents pin the same urn:x402:canonicalisation:jcs-rfc8785-v1 canonicalisation discipline.

The two receipt families compose via a composite trust-query algorithm specified in the x402 protocol substrate: a downstream verifier consuming a composite hash sees one emitter row per provider, where each emitter row references the JCS-canonical bytes of its underlying receipt.

2. Conventions and Definitions

2.1. Notation

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.

2.2. Definitions

screening provider: an entity that performs compliance screening on an agentic-payment transaction at admission time and emits a categorical receipt as the screening decision record.

payer_ref: a content-addressed reference to the payer's identity. Typically of the form "sha256:<lowercase-hex>" where the hash is taken over the JCS-canonical bytes of an identity object held by the screening provider. The receipt does not carry cleartext identity content -- only the content-addressed reference -- so the screening provider's retention layer is GDPR Article 5(1)(c) minimal by construction.

screen_result: the categorical screening outcome. One of ALLOW, REFER, or DENY. The closed set is load-bearing (Section 3.2).

screen_timestamp_ms: an integer giving the number of milliseconds since the Unix epoch (1970-01-01T00:00:00Z) at which the screening decision was recorded. See Section 3.3.

screen_provider_did: a Decentralised Identifier (DID) identifying the screening provider. See Section 3.4.

jurisdiction_flags: an ordered list of regulatory jurisdictions under which the screening was performed. See Section 3.5.

canon_version: an in-band string identifying the canonicalisation rule under which the receipt was canonicalised. See Section 3.6.

audit chain: a monotonic hash-linked sequence of receipts retained by the screening provider; see Section 5.

3. Receipt Format Specification

A compliance screening receipt is a JSON [RFC8259] object containing exactly the following required fields. Implementations MUST reject receipts that omit any required field or include fields not listed below.

```
{
  "payer_ref":           "<string>",
  "screen_result":       "<enum: ALLOW | REFER | DENY>",
  "screen_timestamp_ms": <integer>,
  "screen_provider_did": "<DID URI>",
  "jurisdiction_flags":  ["<jurisdiction-code>", ...],
  "canon_version":       "jcs-rfc8785-v1"
}
```

The JSON Schema (draft-07) for this format is published at https://json.schemastore.org/algovoi-compliance-receipt-v1.json (https://json.schemastore.org/algovoi-compliance-receipt-v1.json).

3.1. payer_ref

The payer_ref field MUST be a non-empty string identifying the payer by content-addressed reference. The recommended form is "sha256:<lowercase-hex>" where the hex is the lowercase hexadecimal representation of the SHA-256 digest of the JCS-canonical bytes of the underlying identity object.

The receipt MUST NOT carry cleartext identity content. This ensures that the screening provider's retention layer is GDPR Article 5(1)(c) minimal by construction; identity content can be re-presented to the receipt holder out-of-band when necessary.

3.2. screen_result Closed Enumeration

The screen_result field MUST be exactly one of the following three string values:

Value	Meaning
ALLOW	The transaction is cleared to proceed under the regulatory gate at the screening provider's jurisdictions.
REFER	The transaction is flagged for enhanced due diligence. Depending on the screening provider's jurisdictions (jurisdiction_flags) this MAY trigger a mandatory Suspicious Activity Report or equivalent regulatory filing obligation.
DENY	The transaction is refused under a sanctions or anti-money-laundering rule.

Table 1

The closed three-value enumeration is load-bearing. Under UK Proceeds of Crime Act 2002 Section 330, a REFER outcome triggers a mandatory Suspicious Activity Report obligation to the UK National Crime Agency for institutions in the regulated sector. A DENY outcome does not trigger the same obligation: it refuses the transaction but does not by itself require a SAR.

A receipt format that compresses screen_result to a continuous score or to a coarser tier set (e.g. "high / medium / low") loses this load-bearing distinction. A long-term auditor (typically operating at the year-five mark or beyond) reading a retained receipt under

such a projection cannot determine, from the retained bytes alone, whether the historical decision triggered a mandatory regulatory report. This document REQUIRES the closed three-value enumeration for that reason.

Future extensions to the receipt format MAY add additional fields alongside the categorical screen_result (e.g. a continuous score for internal monitoring use), but MUST NOT remove or alter the closed enumeration on screen_result itself.

3.3. screen_timestamp_ms

The screen_timestamp_ms field MUST be a non-negative integer giving the number of milliseconds elapsed since 1970-01-01T00:00:00Z (Unix epoch).

The field MUST NOT be a floating-point number, MUST NOT be a JSON-formatted string, and MUST NOT be in RFC 3339 ISO-8601 format. Implementations that receive a non-integer value MUST reject the receipt at the validation layer before canonicalisation; silent type-cast from float to integer is forbidden.

The integer-only restriction is required because RFC 8785 canonicalises floating-point numbers and integers differently. A receipt whose timestamp was emitted as an integer and later re-canonicalised after a type-cast from float would produce different canonical bytes and a different content_hash -- which would fail the long-term auditability property (Section 6).

3.4. screen_provider_did

The screen_provider_did field MUST be a Decentralised Identifier (DID) URI conforming to the DID syntax: "did:" followed by one or more alphanumeric characters identifying the DID method, followed by a method-specific identifier.

Implementations MAY use any DID method, including but not limited to did:web, did:key, did:ion, did:peer. The did:web method is RECOMMENDED where the screening provider operates a stable HTTPS-resolvable endpoint, because did:web allows verifiers to retrieve the screening provider’s public key directly from a well-known URI without consulting any external registry.

3.5. jurisdiction_flags

The jurisdiction_flags field MUST be a non-empty JSON array of strings. Each string identifies a regulatory jurisdiction under which the screening was performed.

Recommended values follow ISO 3166-1 alpha-2 country codes (e.g. "UK", "DE", "JP"). Aggregate jurisdiction codes such as "EU" MAY also be used.

The array element ORDER is SIGNIFICANT and load-bearing. RFC 8785 Section 3.2.3 specifies that JSON array element order is preserved during canonicalisation. The array ["UK", "EU"] and the array ["EU", "UK"] therefore canonicalise to different bytes and produce different content_hash values.

Producer-side ordering MUST match the order in which the regulatory rules were applied. Verifiers SHOULD validate ordering consistency across a screening provider's retained chain.

3.6. canon_version

The canon_version field MUST be the string "jcs-rfc8785-v1" for receipts emitted under the canonicalisation discipline specified by this version of this document.

Future revisions of the canonicalisation discipline MAY introduce successor pin values ("jcs-rfc8785-v2", etc.). Implementations re-canonicalising a retained receipt MUST consult the receipt's canon_version field and apply the corresponding canonicalisation rule.

The in-band pin enables long-term re-verification: the rule active at emission is recorded in the retained bytes themselves and does not require an out-of-band rule registry that the screening provider must continue to publish.

4. Canonicalisation

Compliance receipts are canonicalised using JSON Canonicalization Scheme (JCS) as specified in [RFC8785]. The canonicalisation discipline applicable to this document is identified by the URN:

urn:x402:canonicalisation:jcs-rfc8785-v1

The discipline imposes the following rules on receipt content:

* RFC 8785 canonical JSON for the receipt object.
* Object keys are sorted lexicographically by Unicode code-point order (per RFC 8785 Section 3.2.3).
* Array element order is preserved (per RFC 8785 Section 3.2.3).
* Numeric values are canonicalised per RFC 8785 Section 3.2.2.3.
* The integer-only restriction on screen_timestamp_ms (Section 3.3) is applied at the validation layer before canonicalisation.

The discipline applies to:

* The receipt object itself, for computation of the content_hash (Section 5).
* The bundle of receipts emitted as a selective-disclosure audit bundle (out of scope for this document; see related work in Section 5.4).

5. Audit Chain Composition

A screening provider that emits compliance receipts under this format SHOULD compose those receipts into a monotonic hash-linked audit chain.

5.1. Chain Row Shape

Each row in the audit chain MUST carry the following three fields in addition to the receipt itself:

* chain_position: a non-negative integer that increases monotonically with each new row, starting at zero for the chain head.
* content_hash: the lowercase hexadecimal SHA-256 hash of the JCS-canonical bytes of the receipt object.
* prev_hash: the content_hash of the previous chain row, or null for the chain head (chain_position = 0).

5.2. Linkage Verification

A verifier walking the chain MUST confirm:

* chain_position increases by exactly one between successive rows.
* prev_hash on row N equals content_hash on row (N - 1).
* content_hash on each row recomputes to SHA-256(JCS(receipt_object)) from the retained bytes.

Any linkage break MUST cause the verifier to reject the chain.

5.3. Selective Disclosure

A screening provider MAY emit a selective-disclosure audit bundle containing a subset of the retained chain (typically filtered by a selection_criteria such as tenant_id, payer_ref, screen_result, or a time range). Such bundles MAY include "bridging_rows" that fill chain gaps between selected rows so that the verifier can confirm linkage continuity across the disclosed subset.

The specific bundle envelope format is out of scope for this document. A reference implementation of the selective-disclosure bundle format and the corresponding verifier is published at https://github.com/chopmob-cloud/algovoi-audit-verifier (https://github.com/chopmob-cloud/algovoi-audit-verifier) (MIT licensed).

5.4. Retention Storage

The retention storage backend for the chain is out of scope for this document. An S3-compatible Object Lock COMPLIANCE retention is RECOMMENDED where the screening provider is subject to retention obligations of seven years or more (e.g. UK MLR 2017 Regulation 40(3)).

6. Long-Term Auditability Properties

Regulatory frameworks including the examples listed in Section 1.2 impose retention periods of five years or more, with re-verification of retained records expected across the full horizon. The properties in this section are designed to satisfy re-verification from retained bytes alone across that horizon.

This receipt format is designed to satisfy the following auditability properties:

1. *Re-verification from retained bytes alone.* A long-term verifier reading a retained chain can recompute every content_hash from the row's payload and confirm linkage against prev_hash without consulting any external service. The in-band canon_version pin identifies which canonicalisation rule to apply.

2. *Operator-continuity independence.* Verification does NOT require the screening provider to continue to publish a canonicalisation rule registry, a sanctions feed snapshot, or any other out-of-band resource. The receipt is self-contained.

3. *Cross-implementation verifiability.* The JCS RFC 8785 canonicalisation discipline has been byte-for-byte cross-validated across multiple independent reference implementations including Python rfc8785, JavaScript canonicalize, Go gowebpki/jcs, Java cyberphone/json-canonicalization, and Rust serde_jcs. A verifier built with any of these implementations produces identical canonical bytes for any compliance receipt under this format.

4. *Tamper detection.* Per-row content_hash and chain prev_hash linkage detect single-row tampering and chain truncation / reordering respectively.

5. *Regulatory distinction preserved.* The closed enumeration on screen_result (Section 3.2) preserves the load-bearing distinction between REFER (mandatory SAR under UK POCA 2002 s.330) and DENY for the full retention period.

7. Composition with Other x402 Substrate

This receipt format composes with other elements of the x402 substrate:

7.1. Composite Trust-Query

Multiple emitters' attestations -- including compliance receipts under this format, STARK payment-condition receipts under the companion draft-vauban-x402-stark-receipts work in progress, and service-reputation receipts from other providers -- compose into a single canonical composite_hash via the composite trust-query algorithm specified in the x402 protocol substrate:

composite_hash = SHA-256(JCS([rows sorted by source_id, sig excluded]))

A downstream verifier consuming a composite_hash sees one emitter row per provider, each row containing or referencing the JCS-canonical bytes of the provider's underlying receipt.

7.2. privacy_class Field

A receipt MAY carry an additional privacy_class field declaring the settlement-plane visibility of the receipt, as proposed in the x402 protocol substrate. The field is not part of the required-fields set of this document but is interoperable with it.

7.3. Compliance Category

This receipt format is one of the format options under the broader Compliance category proposed in the x402 protocol substrate, which provides the extension mechanism by which downstream registries may surface compliance attestations.

8. IANA Considerations

The IANA actions described in this document, if any, are subject only to allocation policies that do not require IETF Review or Standards Action.

8.1. URN Namespace

This document uses the URN identifier urn:x402:canonicalisation:jcs-rfc8785-v1 to name the JCS canonicalisation discipline applied to receipts under this document and companion documents. Formal registration of this URN namespace under [RFC8141] is undertaken by the document authors separately from RFC publication, by direct submission of the registration template; the registration is not requested as a publication action of this document. The identifier itself is used verbatim by implementations regardless of registration outcome.

8.2. Media Type

This document does not request the registration of a new media type. Receipts under this format use the application/json media type with the structural constraints specified in Section 3.

8.3. Receipt Format Identifier

The string algovoi-compliance-receipt-v1 is used by the JSON Schema identifier in Section 3 to name the receipt format. This is an informal in-document identifier and does not require a new IANA registry. Implementations adopting variants of this format are encouraged to use distinct, self-descriptive identifiers of their own.

9. Security Considerations

9.1. Receipt Tampering

The per-row content_hash provides single-row tamper detection. A verifier recomputing content_hash from the receipt’s JCS-canonical bytes detects any modification to the receipt content (including modification to a single field, byte, or whitespace).

9.2. Chain Truncation and Reordering

The prev_hash linkage detects chain truncation, row removal, and reordering. A verifier walking the chain confirms that prev_hash on each row equals content_hash on the previous row; any break in the walk indicates tampering, truncation, or reordering.

9.3. Bundle Forgery

A screening provider MAY sign a selective-disclosure audit bundle with an HMAC-SHA256 over the JCS-canonical bytes of the bundle (minus the signature field). The shared signing key is the proof-of-emission secret; verifiers possessing the key can confirm that the bundle was emitted by the holder of the key.

This document does not REQUIRE bundle signing -- the per-row content hashes and chain linkage already provide tamper detection -- but it RECOMMENDS bundle signing for selective-disclosure exchanges where the verifier must distinguish "bundle emitted by the screening provider" from "bundle reconstructed from elsewhere."

9.4. Operator Continuity Loss

The in-band canon_version pin (Section 3.6) ensures that re-verification does not require operator continuity. If the screening provider ceases operations, retained chains remain verifiable: the rule named by canon_version is publicly documented and stable.

9.5. DID Resolution Compromise

The screen_provider_did is a Decentralised Identifier. Verifiers SHOULD resolve it through multiple resolvers (where the DID method supports multi-resolver patterns) and confirm consistency before accepting public key material for signature verification on selective-disclosure bundles.

9.6. Privacy

The privacy analysis in this section follows the framework of [RFC6973]. The payer_ref content-addressed reference and the selective-disclosure mechanism (Section 5.3) are designed to support the data-minimisation principle described therein.

The payer_ref content-addressed reference does not by itself leak identity content. A receipt holder presenting the receipt out-of-band can also present the underlying identity object; the receipt holder chooses when and to whom to disclose identity. This is the selective-disclosure pattern that justifies the content-addressed reference design.

Implementations MUST NOT log cleartext identity content alongside receipts in the chain. Implementations MUST NOT recover the underlying identity object from payer_ref alone -- the reference is a one-way hash.

9.7. Choice of Canonicalisation Scheme

JCS [RFC8785] is an Independent Submission Informational RFC. It is cited normatively here because the canon_version mechanism in Section 3.6 depends on the specific byte-level canonicalisation rules defined in JCS. JCS was selected over alternatives (such as RFC 8259 with locally specified ordering rules) because it provides a cross-implementation byte-deterministic canonicalisation discipline that has been validated across five independent implementations in four programming languages, making the long-term re-verification property of Section 6 reproducible by independent verifiers without relying on AlgoVoi-provided tooling.

10. References

10.1. Normative References

[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>.

[RFC8141] Saint-Andre, P. and J. Klensin, "Uniform Resource Names (URNs)", RFC 8141, DOI 10.17487/RFC8141, April 2017, <https://www.rfc-editor.org/info/rfc8141>.

[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>.

[RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <https://www.rfc-editor.org/info/rfc8259>.

[RFC8785] Rundgren, A., Jordan, B., and S. Erdtman, "JSON Canonicalization Scheme (JCS)", RFC 8785, DOI 10.17487/RFC8785, June 2020, <https://www.rfc-editor.org/info/rfc8785>.

10.2. Informative References

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <https://www.rfc-editor.org/info/rfc6973>.

Appendix A. Regulatory frameworks cited as examples

The regulatory frameworks referred to in Section 1.2 as examples are accessible at the following official sources. These are cited as illustrative examples of the framework-bound retention obligations the receipt format is designed to satisfy; they are not normative references.

* HM Treasury, "The Money Laundering, Terrorist Financing and Transfer of Funds (Information on the Payer) Regulations 2017", SI 2017/692, 2017, https://www.legislation.gov.uk/uksi/2017/692/contents (https://www.legislation.gov.uk/uksi/2017/692/contents).
* UK Parliament, "Proceeds of Crime Act 2002", Chapter 29, 2002, https://www.legislation.gov.uk/ukpga/2002/29/contents (https://www.legislation.gov.uk/ukpga/2002/29/contents).
* European Parliament and of the Council, "Directive (EU) 2018/843", OJ L 156, 19.6.2018, https://eur-lex.europa.eu/eli/dir/2018/843/oj (https://eur-lex.europa.eu/eli/dir/2018/843/oj).
* European Parliament and of the Council, "Directive (EU) 2018/1673", OJ L 284, 12.11.2018, https://eur-lex.europa.eu/eli/dir/2018/1673/oj (https://eur-lex.europa.eu/eli/dir/2018/1673/oj).
* European Parliament and of the Council, "Regulation (EU) 2023/1114 on Markets in Crypto-Assets", OJ L 150, 9.6.2023, Article 80, https://eur-lex.europa.eu/eli/reg/2023/1114/oj (https://eur-lex.europa.eu/eli/reg/2023/1114/oj).
* European Parliament and of the Council, "Regulation (EU) 2024/1624 on Anti-Money Laundering", OJ L 2024/1624, 19.6.2024, Article 56, https://eur-lex.europa.eu/eli/reg/2024/1624/oj (https://eur-lex.europa.eu/eli/reg/2024/1624/oj).
* European Parliament and of the Council, "Regulation (EU) 2022/2554 on Digital Operational Resilience", OJ L 333, 27.12.2022, Article 14, https://eur-lex.europa.eu/eli/reg/2022/2554/oj (https://eur-lex.europa.eu/eli/reg/2022/2554/oj).

The companion individual Internet-Draft draft-vauban-x402-stark-receipts is a work in progress on the IETF datatracker with stream unassigned at the time of writing. The x402 protocol substrate referenced throughout this document is maintained in the working repository x402-foundation/x402 on GitHub; readers seeking current substrate specification text should consult the project’s specification directory rather than any individual revision of a working repository pull request.

Appendix B. Examples (Informative)

The following example receipts illustrate the format. All three are syntactically valid under the JSON Schema at https://json.schemastore.org/algovoi-compliance-receipt-v1.json (https://json.schemastore.org/algovoi-compliance-receipt-v1.json).

B.1. ALLOW under UK + EU joint jurisdiction

```
{
  "payer_ref": "sha256:0dd5d0b76c9b9281...0f",
  "screen_result": "ALLOW",
  "screen_timestamp_ms": 1716460800000,
  "screen_provider_did": "did:example:screening-provider-1",
  "jurisdiction_flags": ["UK", "EU"],
  "canon_version": "jcs-rfc8785-v1"
}
```

B.2. REFER under UK with mandatory SAR obligation

```
{
  "payer_ref": "sha256:4b781b0d3f8a82c5...72",
  "screen_result": "REFER",
  "screen_timestamp_ms": 1716460800050,
  "screen_provider_did": "did:example:screening-provider-1",
  "jurisdiction_flags": ["UK"],
  "canon_version": "jcs-rfc8785-v1"
}
```

Per Section 3.2: institutions in the regulated sector emitting a REFER under a UK jurisdiction MUST file a SAR with the UK National Crime Agency per UK POCA 2002 Section 330.

B.3. DENY under sanctions match across UK + EU + US

```
{
  "payer_ref": "sha256:10779e0aeb2e1d3a...38",
  "screen_result": "DENY",
  "screen_timestamp_ms": 1716460800100,
  "screen_provider_did": "did:example:screening-provider-1",
  "jurisdiction_flags": ["UK", "EU", "US"],
  "canon_version": "jcs-rfc8785-v1"
}
```

Appendix C. Reference Implementations (Informative)

The following open-source packages implement the format specified in this document. None of these implementations is privileged; the format is fully specified by Section 3.

* algovoi-substrate (Python) on PyPI: https://pypi.org/project/algovoi-substrate/ (https://pypi.org/project/algovoi-substrate/) Provides build_compliance_receipt(), action_ref(), and the JCS canonicalisation primitive. Apache 2.0 licensed.

* @algovoi/substrate (TypeScript) on npm: https://www.npmjs.com/package/@algovoi/substrate (https://www.npmjs.com/package/@algovoi/substrate) Byte-for-byte parity with the Python sibling. Apache 2.0 licensed.

* algovoi-audit-verifier (Python) on PyPI: https://pypi.org/project/algovoi-audit-verifier/ (https://pypi.org/project/algovoi-audit-verifier/) Implements the audit-chain verification logic (Section 5.2) including per-row content_hash recomputation and chain linkage walk. MIT licensed.

* @algovoi/audit-verifier (TypeScript) on npm: https://www.npmjs.com/package/@algovoi/audit-verifier (https://www.npmjs.com/package/@algovoi/audit-verifier) Byte-for-byte parity with the Python verifier. MIT licensed.

Cross-implementation conformance vectors are published at:

https://github.com/chopmob-cloud/algovoi-jcs-conformance-vectors (https://github.com/chopmob-cloud/algovoi-jcs-conformance-vectors)

The JSON Schema (draft-07) at https://json.schemastore.org/algovoi-compliance-receipt-v1.json (https://json.schemastore.org/algovoi-compliance-receipt-v1.json) is mirrored at the same repository.

Appendix D. Acknowledgments

This document builds on a canonicalisation discipline maintained in the x402 protocol substrate. The framework-bound-retention scoping clause in that discipline was contributed by FeedOracle. A receipt-format fixture compatible with this document is maintained within the x402 protocol substrate by the Vauban Pay team and references the AlgoVoi production schema as its source.

Author’s Address

Christopher Hopley
AlgoVoi
United Kingdom
Email: chopmob@gmail.com