

Independent Submission
Internet-Draft
Intended status: Informational
Expires: 1 December 2026

C. Hopley
AlgoVoi
30 May 2026

Categorical Refund Receipt Format for Agentic-Payment Flows
draft-hopley-x402-refund-receipt-02

Abstract

This document specifies a categorical refund receipt format for agentic-payment flows. The format is the post-settlement counterpart to the admission-time compliance receipt format specified in draft-hopley-x402-compliance-receipt: where the compliance receipt records an admission decision under regulatory screening obligations, the refund receipt records a reversal-of-funds event with the same canonicalisation discipline and the same audit-chain semantics.

The receipt format uses a closed enumeration of categorical outcomes (FULL, PARTIAL, REJECTED) rather than a continuous percentage or amount-only representation. The categorical outcome is load-bearing for downstream regulatory obligations: under PSD2 (Directive 2015/2366) Article 89 a REJECTED outcome carries dispute-evidence obligations that a PARTIAL refund does not, and the categorical distinction must be preserved at the canonical-bytes level rather than collapsed to a percentage projection of the original amount.

Receipts are canonicalised under RFC 8785 (JSON Canonicalization Scheme) with an in-band canonicalisation rule pin (canon_version). The pin enables year-N re-verification from retained bytes alone, without dependence on an out-of-band rule registry that the operator must continue to publish.

The refund receipt format composes with the compliance receipt format via the original_payment_ref field, a content-addressed reference to the original payment record. A verifier walking the audit chain can confirm the full payment lifecycle from admission-time screening through settlement to refund event under one byte-deterministic canonicalisation pin.

This document is complementary to draft-vauban-x402-stark-receipts (which covers cryptographic settlement-time payment-condition proofs) and to draft-hopley-x402-compliance-receipt (which covers admission-time compliance screening receipts). The three documents pin the same urn:x402:canonicalisation:jcs-rfc8785-v1 canonicalisation discipline.

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 1 December 2026.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
1.1. Motivation	3
1.2. Scope	4
1.3. Relationship to other Internet-Drafts	5
1.4. Relationship to draft-vauban-x402-stark-receipts	5
2. Conventions and Definitions	6
2.1. Notation	6
2.2. Definitions	6
3. Receipt Format Specification	6
3.1. refund_result	6
3.2. jurisdiction_flags	7
3.3. original_payment_ref	7
3.4. refund_amount	8
3.5. refund_provider_did	9
3.6. refund_timestamp_ms	9

3.7.	canon_version	9
4.	Canonicalisation	9
5.	Audit Chain Composition	10
5.1.	Chain Row Shape	10
5.2.	Linkage Verification	11
5.3.	Composition with Compliance Receipts	11
5.4.	Retention Storage	12
6.	Year-N Auditability Properties	12
7.	Composition with Other x402 Substrate	13
7.1.	Compliance Receipt Linkage	13
7.2.	STARK Receipt Linkage	14
7.3.	Composite Trust-Query	14
8.	IANA Considerations	14
8.1.	URN Namespace Registration	14
8.2.	Receipt Format Identifier	14
9.	Security Considerations	14
9.1.	Receipt Tampering	14
9.2.	Chain Truncation and Reordering	15
9.3.	Double-Refund Attacks	15
9.4.	Cross-Asset Refund Substitution	15
9.5.	Operator Continuity Loss	15
9.6.	DID Resolution Compromise	16
9.7.	Privacy	16
Appendix A.	References	16
A.1.	Normative References	16
A.2.	Informative References	16
Appendix B.	Appendix A. Examples (Informative)	17
B.1.	A.1. FULL refund under UK + EU joint jurisdiction	18
B.2.	A.2. PARTIAL refund	18
B.3.	A.3. REJECTED refund (PSD2 Article 89 dispute evidence)	18
Appendix C.	Appendix B. Reference Implementations (Informative)	19
Appendix D.	Appendix C. Acknowledgments	20
Author's Address	21

1. Introduction

1.1. Motivation

Agentic-payment flows generate two classes of receipt over the lifecycle of a transaction: admission-time receipts (compliance screening, payment authorisation, mandate verification) and post-settlement receipts (settlement confirmation, refund, dispute resolution).

A refund is a state transition that creates a new record on the post-settlement chain: funds previously transferred to a payee are returned in full, in part, or the return request is denied. The operational and regulatory significance of this state transition is load-bearing:

- * Under UK Consumer Rights Act 2015 and EU Consumer Rights Directive 2011/83/EU Article 9, a FULL refund within the statutory window closes the consumer's right to further remedy.
- * Under PSD2 (Directive 2015/2366) Article 89, an unauthorised payment must be refunded immediately or rejected with documented reasons. A REJECTED outcome creates dispute-evidence obligations distinct from a FULL or PARTIAL outcome.
- * Under UK POCA 2002 and EU AML Directives 5/6, a refund triggered by post-settlement sanctions screening or fraud detection requires audit-chain linkage to the original compliance receipt for retention-period audit by competent authorities.

A receipt format that collapses these distinctions to a continuous amount-only representation loses the load-bearing operational distinction.

This document specifies a refund receipt format that preserves the categorical outcome at the canonical-bytes level. The same canonicalisation discipline and audit-chain semantics as draft-hopley-x402-compliance-receipt apply, so a verifier can use one implementation to walk the entire payment lifecycle.

1.2. Scope

This document specifies:

- * The canonical JSON shape of the refund receipt (Section 3)
- * The canonicalisation rule applicable to the receipt (Section 4)
- * The audit chain composition under which refund receipts compose with compliance receipts and settlement records (Section 5)
- * The year-N auditability properties the format pins (Section 6)
- * Worked examples covering FULL, PARTIAL, and REJECTED outcomes (Appendix A)
- * Reference implementations and conformance vectors (Appendix B)

The document does NOT specify:

- * The refund orchestration protocol (operator-layer concern; out of scope)
- * Dispute-resolution state machines (a refund receipt records one state transition; multi-party dispute orchestration is a separate problem class)

- * Cross-chain settlement mechanisms (an asset substitution between original payment and refund is permitted in the receipt format but the bridge mechanism is out of scope)
- * Card-network chargeback or interchange formats (ISO 8583 / ISO 20022 message families serve that purpose)

1.3. Relationship to other Internet-Drafts

This document is one of a coordinated suite of five AlgoVoi-authored receipt and response format Internet-Drafts, all anchoring to the canonicalisation discipline specified in draft-hopley-x402-canonicalisation-jcs:

- * draft-hopley-x402-canonicalisation-jcs -- the JCS canonicalisation discipline pinned by Section 4 of this document.
- * draft-hopley-x402-compliance-receipt -- admission-time compliance screening receipt. A refund receipt's `original_payment_ref` MAY equal the `content_hash` of the compliance receipt that admitted the original payment.
- * draft-hopley-x402-settlement-attestation -- per-settlement categorical attestation. A refund receipt's `original_payment_ref` MAY equal the `content_hash` of a settlement attestation when the refund reverses a previously-SETTLED payment.
- * draft-hopley-x402-cancellation-receipt -- the mandate cancellation receipt format. A `USER_REQUESTED` cancellation under PSD2 Article 64 MAY chain forward to a refund receipt on the audit chain when Article 64 refund obligations attach to a recently-settled debit.
- * draft-hopley-x402-composite-trust-query -- verifier-side composite-trust-query response format. A verifier walking an audit chain containing a refund receipt emits a single composite-trust response over the chain.

The five formats share the same canonicalisation pin, audit chain row shape, integer-millisecond timestamp encoding (Substrate Rule 2), `jurisdiction_flags` ordering convention, and closed-enumeration discipline. A verifier walking the full payment lifecycle requires only one implementation of the canonicalisation discipline.

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

draft-vauban-x402-stark-receipts covers cryptographic settlement-time payment-condition proofs (STARK receipts). The refund receipt specified in this document MAY reference a STARK receipt as its `original_payment_ref` value: when a payment that was settled with a STARK proof is subsequently refunded, the refund receipt links back to the STARK-receipt `content_hash` and the verifier walks the chain across both formats under one canonicalisation pin.

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

refund receipt: a JSON object of the shape specified in Section 3, canonicalised under the discipline specified in Section 4.

content_hash: SHA-256, lowercase hex, of the JCS-canonical bytes of the receipt object.

original_payment_ref: a string of the form sha256:<lowercase-hex-64> identifying the original payment record by content hash. The original record MAY be a compliance receipt, a settlement attestation, or an operator-specific payment record.

refund_amount: a two-field object {amount_minor, asset_id} representing the refunded value. amount_minor is a decimal-digit string in the asset's minor unit; asset_id identifies the asset and its decimal precision.

canon_version: an in-band string identifying the canonicalisation discipline. Fixed value jcs-rfc8785-v1 for this version.

audit chain row: a four-field object linking receipts via SHA-256 hash chain, shape specified in Section 5.1.

3. Receipt Format Specification

A refund receipt is a JSON object with the following seven fields. All fields are REQUIRED. The receipt is canonicalised under RFC 8785 (JCS) per Section 4. Field names are sorted lexicographically by JCS during canonicalisation; the receipt object itself uses arbitrary authoring order.

3.1. refund_result

A string-valued field. The value MUST be one of:

- * FULL -- the entire original payment amount has been returned to the payer.

- * PARTIAL -- less than the original payment amount has been returned. The refund_amount field carries the actual refunded value; the verifier compares against the original via original_payment_ref linkage.
- * REJECTED -- a refund request was processed and denied. No funds moved. The receipt records the denial event so downstream dispute / chargeback chains can reference it.

The three-element enumeration is closed. Implementations MUST reject any other value at validation time before canonicalisation. Score, percentage, or amount-only projections are not acceptable substitutes for the categorical outcome.

The regulatory distinction is load-bearing. Under PSD2 (Directive 2015/2366) Article 89, a payer denied a refund (REJECTED) must receive a documented denial; this is operationally distinct from a PARTIAL refund where some-but-not-all of the original amount has been returned. Under UK Consumer Rights Act 2015 and EU Consumer Rights Directive 2011/83/EU Article 9, only a FULL refund within the statutory window closes the consumer's right to further remedy. A percentage or amount-only representation cannot preserve these distinctions.

3.2. jurisdiction_flags

An ordered array of string-valued ISO-3166-1 alpha-2 country codes or alpha-3 region codes identifying the applicable regulatory frameworks for the refund event.

Authoring convention: primary jurisdiction first (where the operating entity is licensed), secondary jurisdictions in order of regulatory precedence.

Array element ORDER is SIGNIFICANT and load-bearing. RFC 8785 does not normalise array order during canonicalisation. The array ["UK", "EU"] and the array ["EU", "UK"] produce byte-distinct canonical representations and distinct content_hash values.

3.3. original_payment_ref

A string-valued field of the form sha256:<lowercase-hex-64>. The hex digest is SHA-256 of the JCS-canonical bytes of the original payment record being refunded.

When refunding a payment that was admitted under a compliance receipt (per draft-hopley-x402-compliance-receipt), the `original_payment_ref` MAY equal the `content_hash` of the compliance receipt itself. This is the conventional choice and enables the audit-chain composition described in Section 5.

When refunding a payment that was settled with a STARK proof (per draft-vauban-x402-stark-receipts), the `original_payment_ref` MAY equal the `content_hash` of the STARK receipt.

Implementations MUST NOT strip the `sha256:` prefix during canonicalisation or verification. The prefix is part of the canonical bytes.

3.4. `refund_amount`

A sub-object with exactly two fields:

- * `amount_minor` -- a string of decimal digits representing the refunded value in the asset's minor unit. String typing avoids float-precision loss and JavaScript-integer-overflow concerns for large values, and is representable losslessly under JCS.
- * `asset_id` -- a string identifying the asset and its decimal precision. Convention: `<symbol>.<decimals>` for chain-native assets (e.g. `USDC.6`, `ALGO.6`, `ETH.18`); `<chain>:<asset_id>.<decimals>` for ASA-style assets (e.g. `algo:31566704.6`).

The sub-object's keys are sorted lexicographically by RFC 8785 during canonicalisation: `amount_minor` then `asset_id`. Verifiers MUST treat `{"amount_minor":"100000","asset_id":"USDC.6"}` as equivalent to the same content produced by an authoring layer that emitted fields in `asset_id`-first order; JCS canonicalisation removes the distinction.

For a PARTIAL refund, `refund_amount` carries the actual refunded value. For a FULL refund, `refund_amount` carries the entire original payment amount. For a REJECTED refund, `refund_amount` carries the amount that WOULD have been refunded had the request been accepted (the requested-but-denied amount), enabling the denial-evidence chain to capture the operational claim.

Cross-asset refunds (where the refund asset differs from the original payment asset, e.g. paid in USDC on Base, refunded in USDC on Solana) are permitted: `refund_amount.asset_id` MAY differ from the asset of the original payment. The cross-chain mechanism by which the substitution is achieved is out of scope of this document.

3.5. refund_provider_did

A string-valued DID URI identifying the entity issuing the refund. For Verifiable-Credentials-class identities this is the issuer DID; for centralised gateway-class identities this is did:web:<host>.

3.6. refund_timestamp_ms

An integer-valued field carrying the epoch-millisecond timestamp of the refund event in UTC.

This field MUST be an integer in the canonical receipt. RFC 3339 string forms (e.g. "2026-05-28T12:00:00Z") MUST be rejected at the validation layer before canonicalisation; silent type-coercion breaks year-N auditability and cross-implementation byte-determinism.

The integer-only restriction is required because RFC 8785 canonicalises the integer 1716494400000 and the string "1716494400000" to distinct canonical-bytes representations. A receipt format that accepted both forms could produce two byte-distinct receipts for the same logical refund event, breaking the dedup and audit-chain linkage properties.

This rule is formalised in x402-foundation/x402 pull request #2436 (canonicalisation discipline), normatively specified in draft-hopley-x402-canonicalisation-jcs Section 4.1, and is referred to in the AlgoVoi substrate as Substrate Rule 2.

3.7. canon_version

A string-valued in-band canonicalisation rule pin. For this version of the receipt format the value MUST be jcs-rfc8785-v1.

The pin enables year-N re-verification from retained bytes alone. A future revision of the canonicalisation discipline MAY introduce a successor pin (e.g. jcs-rfc8785-v2); a verifier reading a receipt with the successor pin applies the corresponding rule. The pin is itself canonicalised and signed-over so it cannot be silently re-narrated post-emission.

4. Canonicalisation

The refund receipt MUST be canonicalised under JSON Canonicalization Scheme (JCS) as specified in [RFC8785]. The canonicalisation discipline pinned by this document is:

urn:x402:canonicalisation:jcs-rfc8785-v1

This discipline mandates:

- * RFC 8785 canonical JSON for the receipt object.
- * UTF-8 encoding with no byte-order mark.
- * Object keys sorted lexicographically by code point at every level (per RFC 8785 Section 3.2.3).
- * Array element order preserved (per RFC 8785 Section 3.2.3).
- * Numeric values canonicalised per RFC 8785 Section 3.2.2.3.
- * Integer-only encoding for refund_timestamp_ms (Substrate Rule 2), applied at the validation layer before canonicalisation.

The discipline pinned here is identical to that pinned by draft-hopley-x402-compliance-receipt Section 4, normatively specified in draft-hopley-x402-canonicalisation-jcs, and to the canonicalisation specification at x402-foundation/x402 specs/canonicalisation.md (PR #2436).

The discipline is byte-for-byte cross-validated across eight independent JCS implementations in eight programming languages per the AlgoVoi conformance attestation dated 2026-05-24: [AlgoVoi-JCS-8-impl].

5. Audit Chain Composition

A refund receipt MAY participate in an audit chain alongside compliance receipts and other receipt classes that pin the same canonicalisation discipline.

5.1. Chain Row Shape

An audit chain row is a JSON object with four fields:

```
{
  "row_number": 1,
  "content_hash": "<hex64>",
  "prev_hash": "<hex64>",
  "row_content_hash": "<hex64>"
}
```

- * row_number: integer, 1-indexed.
- * content_hash: SHA-256, lowercase hex, of the JCS-canonical bytes of the receipt object anchored by this row.
- * prev_hash: SHA-256, lowercase hex, of the previous row's row_content_hash. For row 1, prev_hash MUST be 64 zero hex characters (0000...0000).

- * `row_content_hash`: SHA-256, lowercase hex, of the JCS-canonical bytes of this row object (with `row_content_hash` itself excluded from the JCS computation; the row's first three fields {`row_number`, `content_hash`, `prev_hash`} are canonicalised and hashed to produce `row_content_hash`).

This row shape is identical to the audit-chain row shape specified in draft-hopley-x402-compliance-receipt Section 5.1. The two documents pin the same chain composition so receipts of either class can be linked into one verifiable chain.

5.2. Linkage Verification

A verifier reading a chain segment performs these checks:

- * For each row, recompute the row's `row_content_hash` from its first three fields under JCS and compare against the stored value. Mismatch indicates row tampering.
- * For each row $N > 1$, check that row N 's `prev_hash` equals row $N-1$'s `row_content_hash`. Mismatch indicates chain truncation, reordering, or row deletion.
- * For row 1, check that `prev_hash` is 64 zero hex characters (the chain genesis anchor).
- * For each row, optionally fetch the receipt body whose `content_hash` is referenced and verify the body re-canonicalises to the same hash. This step is required for receipt-level audit but optional for row-level chain integrity.

A chain that passes all of the above is byte-deterministically verifiable from retained bytes alone, with no dependence on the originating gateway's continued operation.

5.3. Composition with Compliance Receipts

When a refund receipt references a compliance receipt via `original_payment_ref`, the audit-chain composition is:

chain row N		chain row N+1
+-----+		+-----+
compliance	-->	refund
receipt		receipt
(content_		(content_
hash)		hash)
+-----+		+-----+

Row N anchors the compliance receipt (admission decision). Row N+1 anchors the refund receipt; the refund receipt's `original_payment_ref` equals the `content_hash` of the compliance receipt anchored by row N. The chain row N+1's `prev_hash` equals row N's `row_content_hash`. A verifier walking this segment confirms:

1. The compliance receipt admitted the original payment.
2. The refund receipt links to that admission via `original_payment_ref`.
3. The chain rows preserve hash-linkage with no truncation or reordering.

The same composition extends to longer chains: multi-leg PARTIAL refunds, refund reversals (refund-of-refund), and disputed REJECTED refunds with subsequent re-evaluation each contribute one or more rows to the chain.

5.4. Retention Storage

Retention storage of refund receipts and their audit chains is out of scope of this document. Operators MUST retain receipts for the period required by applicable regulatory frameworks (UK Consumer Rights Act 2015 disclosure period; PSD2 Article 89 dispute window; MiCA Article 80 records retention; AMLR Article 56 records retention).

Object Lock COMPLIANCE-mode retention on object stores supporting WORM retention (Amazon S3, Backblaze B2, Cloudflare R2) is a practical implementation; refund receipts and their audit-chain rows MUST NOT be modifiable during the retention period.

6. Year-N Auditability Properties

This receipt format preserves the following properties so that retained bytes are independently verifiable years after emission, without dependence on the originating gateway's continued operation.

1. *Self-describing canonicalisation pin.* The `canon_version` field is an in-band identifier (`jcs-rfc8785-v1`) so a verifier reading retained bytes can determine which canonicalisation rule to apply.
2. *No external rule registry required.* The canonical bytes produced under `jcs-rfc8785-v1` are RFC 8785 plus the integer-millisecond timestamp encoding rule (Substrate Rule 2). Both rules are specified in this document and in draft-hopley-x402-compliance-receipt; neither requires the operator to continue to publish a canonicalisation rule registry.

3. **Cross-implementation verifiability.** The JCS RFC 8785 canonicalisation discipline has been byte-for-byte cross-validated across eight independent reference implementations in eight programming languages: Python (rfc8785), TypeScript (canonicalize), Go (gowebpki/jcs), Rust (serde_jcs), Java (cyberphone/json-canonicalization, authored by Anders Rundgren, RFC 8785 editor), PHP (root23/php-json-canonicalization), C#/.NET (Baqhub.Packages.JsonCanonicalization), and Ruby (json-canonicalization). 192/192 byte-for-byte agreements across 24 vectors in 3 anchor sets. The attestation record is published at [AlgoVoi-JCS-8-impl]. A verifier built with any of these eight implementations produces identical canonical bytes for any refund receipt under this format.
4. **Tamper detection.** Per-row content_hash and chain prev_hash linkage (Section 5) detect single-row tampering and chain truncation / reordering respectively.
5. **Regulatory distinction preserved.** The closed enumeration on refund_result (Section 3.1) preserves the load-bearing distinction between FULL, PARTIAL, and REJECTED for the full retention period.
6. **Compositional verifiability.** A refund receipt's original_payment_ref MAY reference any other receipt class that pins the same canonicalisation discipline (compliance receipt, STARK receipt, settlement attestation). The verifier needs only one canonicalisation implementation to walk the full payment lifecycle.
7. Composition with Other x402 Substrate

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

7.1. Compliance Receipt Linkage

The conventional original_payment_ref target for an admission-then-refund flow is the content_hash of the compliance receipt that admitted the original payment under draft-hopley-x402-compliance-receipt. The two receipts compose into one audit-chain segment that records the full admission-through- refund lifecycle. See Section 5.3.

7.2. STARK Receipt Linkage

When a payment was settled with a STARK proof per draft-vauban-x402-stark-receipts, the refund receipt's `original_payment_ref` MAY equal the `content_hash` of the STARK receipt. The verifier confirms the STARK-proven settlement and the subsequent refund event under one canonicalisation pin.

7.3. Composite Trust-Query

Multiple emitters' attestations may participate in a composite trust-query per x402-foundation/x402 pull request #2440. A refund provider's attestation row in a composite-trust-query MAY reference a refund receipt's `content_hash` as the underlying evidence.

8. IANA Considerations

8.1. URN Namespace Registration

This document references the URN `urn:x402:canonicalisation:jcs-rfc8785-v1` registered by draft-hopley-x402-canonicalisation-jcs Section 10.1. No additional URN namespace registration is required by this document.

8.2. Receipt Format Identifier

This document defines the receipt format identifier:

`urn:x402:receipt:refund-v1`

This identifier MAY be used by composite-trust-query implementations (per x402-foundation/x402 PR #2440) to refer to refund receipts as a class. Registration with IANA is requested under the x402-foundation/x402 receipt-format identifier namespace.

9. Security Considerations

9.1. Receipt Tampering

A refund receipt's `content_hash` is the SHA-256 of its JCS-canonical bytes. Tampering with any field of the receipt produces a different `content_hash`; the tampered receipt fails verification against any audit-chain row that references the original `content_hash`.

9.2. Chain Truncation and Reordering

The audit chain `prev_hash` linkage (Section 5.2) detects truncation, reordering, or row deletion. A verifier walking the chain confirms that every row's `prev_hash` equals the previous row's `row_content_hash`; any deviation indicates chain integrity loss.

9.3. Double-Refund Attacks

A malicious operator might attempt to issue two refund receipts for the same original payment, each linking to the same `original_payment_ref`. Detection requires the verifier to walk the chain forward from the compliance receipt and observe both refund rows; the chain itself does not prevent the double-issuance but makes it byte-detectable.

Operators SHOULD enforce single-refund-per-payment at the operator-layer orchestration before emitting receipts. For PARTIAL refunds that genuinely require multiple legs (e.g. a customer returning two items from a single order), each leg's refund receipt MUST link to the original payment ref AND to the prior refund row to make the leg-sequence verifiable; the chain order encodes the cumulative refunded amount.

9.4. Cross-Asset Refund Substitution

A refund receipt MAY refund in a different asset than the original payment. A verifier cannot determine equivalence-of-value between the original asset and the refund asset from the receipt alone; asset-substitution claims of equivalence require additional external evidence (price oracle attestations, settlement proofs) and are out of scope of this document.

Operators substituting assets MUST disclose the substitution clearly to the payer at the time of the refund and MUST retain evidence of the equivalence claim under the same retention period as the receipt itself.

9.5. Operator Continuity Loss

If the original refund-issuing operator becomes unavailable, the audit chain and its receipts MUST remain independently verifiable from retained bytes plus the eight reference implementations cited in Section 6. This document specifies year-N auditability properties so that operator continuity loss does not break verifiability.

9.6. DID Resolution Compromise

`refund_provider_did` is a DID URI. If the DID resolution infrastructure is compromised at verification time, the verifier MAY be unable to resolve the DID to an authoritative key. The receipt format MAY be combined with offline DID document snapshots (captured at the time of receipt emission and retained alongside the receipt) to make verification independent of live DID resolution.

9.7. Privacy

The `original_payment_ref` field is content-addressed, not cleartext. A receipt does not leak payer identity, payee identity, or original payment amount on its face. The `refund_amount` field carries the refunded value but the original payment amount is not in the receipt; cross-receipt linkage (via `original_payment_ref` to a compliance receipt) is required to recover the original amount, and the linkage is itself content-addressed.

Appendix A. References

A.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.
- * [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017.
- * [RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017.
- * [RFC8785] Rundgren, A., Jordan, B., and S. Erdtman, "JSON Canonicalization Scheme (JCS)", RFC 8785, DOI 10.17487/RFC8785, June 2020.
- * [RFC8141] Saint-Andre, P. and J. Klensin, "Uniform Resource Names (URNs)", RFC 8141, DOI 10.17487/RFC8141, April 2017.

A.2. Informative References

- * draft-hopley-x402-canonicalisation-jcs Hopley, C., "JCS Canonicalisation Discipline for Agentic-Payment Receipts", draft-hopley-x402-canonicalisation-jcs-v1, May 2026.
- * draft-hopley-x402-compliance-receipt Hopley, C., "Categorical Compliance Screening Receipt Format for Agentic-Payment Flows", draft-hopley-x402-compliance-receipt-01, May 2026.
- * draft-hopley-x402-settlement-attestation Hopley, C., "Categorical Settlement Attestation Format for Agentic-Payment Flows", draft-hopley-x402-settlement-attestation-00, May 2026.

- * draft-hopley-x402-cancellation-receipt Hopley, C., "Categorical Mandate Cancellation Receipt Format for Agentic-Payment Flows", draft-hopley-x402-cancellation-receipt-00, May 2026.
- * draft-hopley-x402-composite-trust-query Hopley, C., "Composite Trust Query Response Format for Agentic-Payment Audit Chains", draft-hopley-x402-composite-trust-query-00, May 2026.
- * draft-vauban-x402-stark-receipts (companion document on STARK receipt format).
- * AlgoVoi, "Substrate Adopters Registry", target="https://docs.algovoi.co.uk/adopters" (https://docs.algovoi.co.uk/adopters)
- * x402-foundation/x402 pull request #2436 -- canonicalisation discipline (AlgoVoi-authored normative section with subsequent review co-signatures from Vauban Pay and Arian Gogani).
- * x402-foundation/x402 pull request #2440 -- composite trust-query algorithm.
- * [AlgoVoi-JCS-8-impl] AlgoVoi, "Eight-implementation cross-validation attestation for JCS RFC 8785", 2026-05-24, target="https://github.com/chopmob-cloud/algovoi-jcs-conformance-vectors/blob/main/_attestations/2026-05-24-8-impl-cross-validation.md" (https://github.com/chopmob-cloud/algovoi-jcs-conformance-vectors/blob/main/_attestations/2026-05-24-8-impl-cross-validation.md)
- * [AlgoVoi-Substrate-Authorship] AlgoVoi, "Substrate Authorship and Provenance", target="https://docs.algovoi.co.uk/substrate-authorship-provenance" (https://docs.algovoi.co.uk/substrate-authorship-provenance)
- * UK Consumer Rights Act 2015.
- * EU Consumer Rights Directive 2011/83/EU, Article 9.
- * EU Payment Services Directive 2 (PSD2, Directive 2015/2366), Article 89 (unauthorised payment refunds).
- * UK Money Laundering Regulations 2017.
- * UK Proceeds of Crime Act 2002, Section 330.
- * EU Anti-Money Laundering Directive 5 (Directive (EU) 2018/843).
- * EU Anti-Money Laundering Directive 6 (Directive (EU) 2018/1673).
- * EU Markets in Crypto-Assets Regulation (MiCA, Regulation (EU) 2023/1114), Article 80.
- * EU Anti-Money Laundering Regulation (AMLR, Regulation (EU) 2024/1624), Article 56.

Appendix B. Appendix A. Examples (Informative)

The following example refund receipts illustrate the format. All three are canonicalised under jcs-rfc8785-v1 and produce the content_hashes pinned in the AlgoVoi refund_receipt_v1 conformance vector set.

B.1. A.1. FULL refund under UK + EU joint jurisdiction

```
{
  "canon_version": "jcs-rfc8785-v1",
  "jurisdiction_flags": ["UK", "EU"],
  "original_payment_ref": "sha256:0dd5d0b76c9b9281fdeb2509ad38ab132b16a17385ca01d976ff
9e6e12563a0f",
  "refund_amount": {"amount_minor": "100000", "asset_id": "USDC.6"},
  "refund_provider_did": "did:example:refund-provider-1",
  "refund_result": "FULL",
  "refund_timestamp_ms": 1716494400000
}
```

Refunds 0.1 USDC (100000 minor units, 6 decimals) under joint UK + EU jurisdiction. The `original_payment_ref` references a payment record whose `content_hash` is the listed value (e.g. a compliance receipt from draft-hopley-x402-compliance-receipt). FULL closes the consumer's right to further remedy under UK Consumer Rights Act 2015 and EU Consumer Rights Directive 2011/83/EU Article 9.

B.2. A.2. PARTIAL refund

```
{
  "canon_version": "jcs-rfc8785-v1",
  "jurisdiction_flags": ["UK", "EU"],
  "original_payment_ref": "sha256:0dd5d0b76c9b9281fdeb2509ad38ab132b16a17385ca01d976ff
9e6e12563a0f",
  "refund_amount": {"amount_minor": "100000", "asset_id": "USDC.6"},
  "refund_provider_did": "did:example:refund-provider-1",
  "refund_result": "PARTIAL",
  "refund_timestamp_ms": 1716494400000
}
```

`refund_amount` carries the actual refunded value; the verifier compares against the original via `original_payment_ref` to confirm the refund is strictly less than the original. PARTIAL does not close the consumer's right under consumer-rights statutes; further remedy may remain.

B.3. A.3. REJECTED refund (PSD2 Article 89 dispute evidence)

```
{
  "canon_version": "jcs-rfc8785-v1",
  "jurisdiction_flags": ["UK", "EU"],
  "original_payment_ref": "sha256:0dd5d0b76c9b9281fdeb2509ad38ab132b16a17385ca01d976ff
9e6e12563a0f",
  "refund_amount": {"amount_minor": "100000", "asset_id": "USDC.6"},
  "refund_provider_did": "did:example:refund-provider-1",
  "refund_result": "REJECTED",
  "refund_timestamp_ms": 1716494400000
}
```

A refund request was processed and denied. `refund_amount` carries the requested-but-denied amount so the denial-evidence chain can record the operational claim. Under PSD2 Article 89, a payer denied a refund must receive a documented denial; this receipt fulfils that documentation obligation.

Appendix C. Appendix B. 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-refund-receipt` (Python) on PyPI:
target="https://pypi.org/project/algovoi-refund-receipt/"
(https://pypi.org/project/algovoi-refund-receipt/) Provides `build_refund_receipt()`. Depends on `algovoi-substrate` for the JCS canonicalisation primitive. Apache 2.0 licensed.
- * `@algovoi/refund-receipt` (TypeScript) on npm:
target="https://www.npmjs.com/package/@algovoi/refund-receipt"
(https://www.npmjs.com/package/@algovoi/refund-receipt) Byte-for-byte parity with the Python sibling. Depends on `@algovoi/substrate` for the JCS canonicalisation primitive. Apache 2.0 licensed.
- * `algovoi-substrate` (Python) on PyPI:
target="https://pypi.org/project/algovoi-substrate/"
(https://pypi.org/project/algovoi-substrate/) Provides `build_compliance_receipt()` and the JCS canonicalisation primitive used by the refund receipt builder. Apache 2.0 licensed.
- * `@algovoi/substrate` (TypeScript) on npm:
target="https://www.npmjs.com/package/@algovoi/substrate"
(https://www.npmjs.com/package/@algovoi/substrate) Byte-for-byte parity with the Python substrate. Apache 2.0 licensed.
- * `algovoi-audit-verifier` (Python) on PyPI:
target="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:
target="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:

target="https://github.com/chopmob-cloud/algovoi-jcs-conformance-vectors" (<https://github.com/chopmob-cloud/algovoi-jcs-conformance-vectors>)

The `refund_receipt_v1` vector set is at `vectors/refund_receipt_v1/` and includes eight byte-level reference vectors, five pair invariants, and three chain invariants.

The eight-implementation cross-validation attestation records dated 2026-05-24 (JCS canonicalisation layer and RFC 9421 transport layer) apply to refund receipts as they do to compliance receipts.

Appendix D. Appendix C. Acknowledgments

This document, the receipt format it specifies, and the conformance vectors derived from it are AlgoVoi-authored work. Substrate authorship history, including production code anchored on 2026-04-12 and the eight-implementation cross-validation attestation records, is catalogued at `target="https://docs.algovoi.co.uk/substrate-authorship-provenance"` (<https://docs.algovoi.co.uk/substrate-authorship-provenance>).

The canonicalisation discipline pinned by Section 4 (`urn:x402:canonicalisation:jcs-rfc8785-v1`) is normatively specified in `draft-hopley-x402-canonicalisation-jcs-v1`. The integer-millisecond canonical timestamp convention (Substrate Rule 2 of that document) used by every timestamp field in this refund receipt format was first posted publicly on `x402-foundation/x402` issue #2357 on 2026-05-20 by Andy Salvo (Crest Deployment Systems LLC). The conformance vector 0009 (`field-name-load-bearing`) on `x402-foundation/x402` pull request #2398, also contributed by Andy Salvo, demonstrates the same invariant from the work-receipt layer.

The canonicalisation discipline was earlier explored in `x402-foundation/x402` pull request #2436 (closed 2026-05-24), reviewed at that time by `seritalien` (Vauban Pay, APPROVED 2026-05-22) and `arian-gogani` (Arian Gogani, LGTM). The framework-bound-retention scoping clause in that discipline was contributed by `feedoracle` (FeedOracle).

This document is the post-settlement companion to the admission-time compliance receipt format specified in `draft-hopley-x402-compliance-receipt` by the same author. The two documents share the same canonicalisation pin, audit-chain shape, and integer-millisecond timestamp encoding so that a verifier walking the full payment lifecycle requires only one implementation of the canonicalisation discipline.

Author's Address

Christopher Hopley
AlgoVoi
Email: chopmob@gmail.com