Supply Chain Integrity, Transparency, and Trust
Internet-Draft
Intended status: Informational
Expires: 20 June 2026

T. Kamimura
VeritasChain Standards Organization

17 December 2025

SCITT Profile for Financial Trading Audit Trails: VeritasChain Protocol (VCP)

draft-kamimura-scitt-vcp-00

Abstract

This document defines a SCITT (Supply Chain Integrity, Transparency, and Trust) profile for creating tamper-evident audit trails of AI-driven algorithmic trading decisions and executions. The VeritasChain Protocol (VCP) extends the SCITT architecture to address the specific requirements of financial markets, including nanosecond-precision timestamps, regulatory compliance with EU AI Act and MiFID II, and privacy-preserving mechanisms (crypto-shredding) for GDPR compliance. This profile defines how VCP events are encoded as SCITT Signed Statements, registered with Transparency Services, and verified using COSE Receipts.

About This Document

This note is to be removed before publishing as an RFC.

The latest version of this document, along with implementation resources and test vectors, can be found at https://github.com/veritaschain/vcp-spec.

Discussion of this document takes place on the SCITT Working Group mailing list (scitt@ietf.org).

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 20 June 2026.

Table of Contents

1. Introduction

The SCITT (Supply Chain Integrity, Transparency, and Trust) architecture [I-D.ietf-scitt-architecture] provides a framework for creating tamper-evident logs of digital artifacts through Transparency Services. While SCITT was initially designed for software supply chain use cases, its core primitives—Signed Statements, Receipts, and Transparency Services—are applicable to any domain requiring verifiable audit trails.

This document specifies how SCITT can be applied to the domain of AI-driven algorithmic trading systems in financial markets. The VeritasChain Protocol (VCP) defines:

* A schema for encoding trading events as SCITT Signed Statements
* Registration policies for financial audit trails
* Conformance tiers (Silver, Gold, Platinum) with varying requirements for timing precision and cryptographic guarantees
* Privacy mechanisms compatible with GDPR data erasure requirements

VCP serves as an "AI Flight Recorder" for algorithmic trading, enabling post-incident reconstruction of system behavior with cryptographic proof of integrity.

1.1. Requirements Language

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.

1.2. Relationship to SCITT

This document is a profile of the SCITT architecture. It:

* REQUIRES compliance with [I-D.ietf-scitt-architecture]

* REQUIRES use of SCRAPI [I-D.ietf-scitt-scrapi] for Transparency Service interactions

* RECOMMENDS COSE Receipts for inclusion proofs

* DEFINES domain-specific extensions for financial trading

1.3. Scope

This document specifies:

* VCP Event schema as SCITT Signed Statement payload

* Registration Policy requirements for VCP Transparency Services

* Mapping of VCP concepts to SCITT terminology

* Three conformance tiers with specific requirements

* Crypto-shredding mechanism for privacy compliance

This document does not specify:

* General SCITT architecture (see [I-D.ietf-scitt-architecture])

* SCRAPI endpoints (see [I-D.ietf-scitt-scrapi])

* COSE Receipt format (see [RFC9052])

* Specific regulatory mapping (covered in companion documents)

2. Terminology

This document uses terminology from [I-D.ietf-scitt-architecture]. The following terms are specific to this profile:

VCP Event
: A single auditable record in the VeritasChain Protocol, encoded as a SCITT Signed Statement. Consists of Header, Payload, and Security metadata.

VCP Issuer
: An algorithmic trading system, AI model, or human operator that generates VCP Events. Corresponds to SCITT Issuer.

VCP Transparency Service
: A SCITT Transparency Service configured with VCP-specific Registration Policies. Operates a VCP-compliant append-only log.

VCP Receipt
: A COSE Receipt issued by a VCP Transparency Service,

proving inclusion of a VCP Event in the log.

Actor An entity (algorithm, human operator, or system) identified by ActorID that generates VCP Events.

Crypto-Shredding A privacy technique where encrypted payload data is rendered permanently unrecoverable by destroying the encryption key, while preserving the cryptographic integrity proofs (hashes and Receipts).

Conformance Tier One of three implementation levels (Silver, Gold, Platinum) with increasing requirements for timing precision, anchoring frequency, and cryptographic guarantees.

2.1. SCITT Terminology Mapping

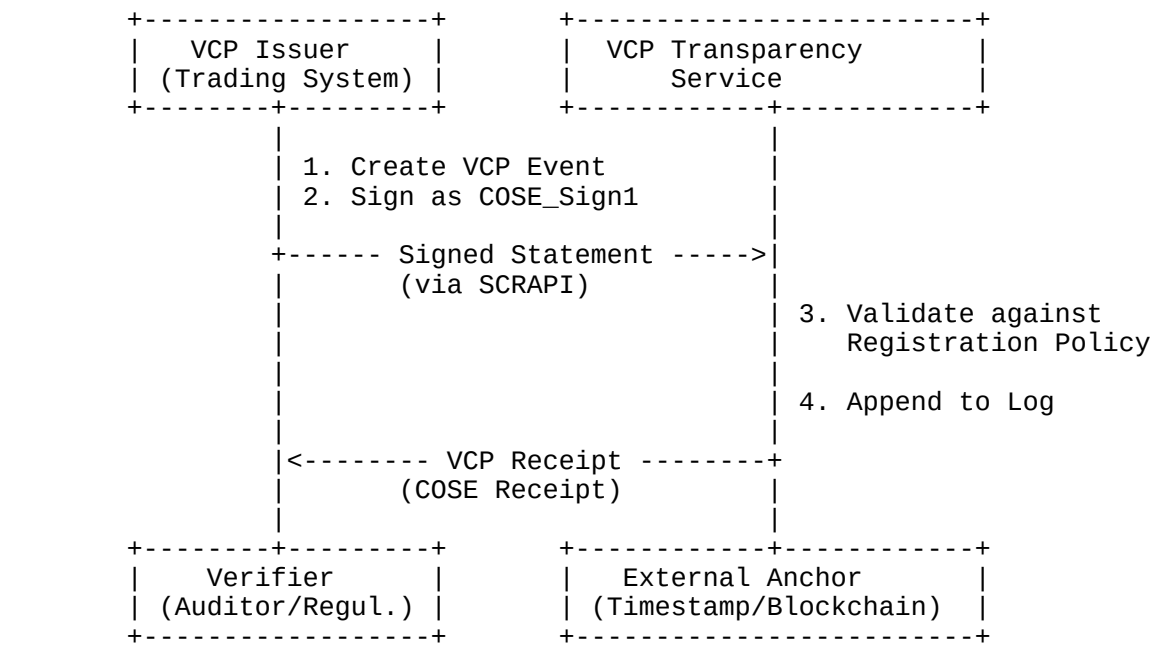
The following table maps VCP concepts to SCITT terminology:

VCP Concept	SCITT Equivalent	Notes
VCP Event	Signed Statement	VCP Event is the payload of a Signed Statement
VCP Issuer	Issuer	Trading system or AI model
VCP Transparency Service	Transparency Service	With VCP Registration Policy
VCP Receipt	Receipt	COSE Receipt with Merkle inclusion proof
Hash Chain	Append-only Log	VCP adds per-Actor chaining
Merkle Anchor	Merkle Tree Root	Periodic commitment

Table 1: VCP to SCITT Terminology Mapping

3. Architecture

VCP builds upon the SCITT architecture with domain-specific extensions for financial trading:



3.1. Event Flow

1. *Event Generation:* VCP Issuer creates a VCP Event containing trading decision or execution data.

2. *Signing:* Event is signed using COSE_Sign1 [RFC9052] with the Issuer's private key.

3. *Registration:* Signed Statement is submitted to VCP Transparency Service via SCRAPI POST /entries.

4. *Validation:* Transparency Service validates against VCP Registration Policy.

5. *Logging:* Valid statement is appended to the append-only log.

6. *Receipt:* COSE Receipt with Merkle inclusion proof is returned to Issuer.

3.2. Per-Actor Hash Chain

In addition to SCITT's global append-only log, VCP maintains per-Actor hash chains. Each VCP Event includes a PrevHash field containing the hash of the previous event from the same Actor. This enables efficient verification of a single Actor's event sequence without downloading the entire log.

```
Actor A:  Event_A1 --hash--> Event_A2 --hash--> Event_A3

Actor B:  Event_B1 --hash--> Event_B2

Global Log: [Event_A1, Event_B1, Event_A2, Event_B2, Event_A3, ...]
```

4. VCP Event Schema

A VCP Event is encoded as the payload of a SCITT Signed Statement. The payload MUST be a JSON object conforming to the following schema:

4.1. Header Object

The Header contains metadata common to all VCP Events:

```
{
  "Header": {
    "EventID": "01961e5f-5c0d-7000-8000-123456789abc",
    "TimestampISO": "2026-03-15T09:30:00.123456789Z",
    "TimestampInt": 1742034600123456789,
    "EventType": "ORD",
    "ActorID": "algo-momentum-001",
    "ChainID": "chain-actor-001",
    "SequenceNum": 42
  }
}
```

EventID REQUIRED. UUID v7 [RFC9562] providing time-sortable unique identification.

TimestampISO REQUIRED. ISO 8601 timestamp with nanosecond precision.

TimestampInt REQUIRED. Integer timestamp in nanoseconds since Unix epoch.

EventType REQUIRED. Three-letter code identifying the event type (see Section 4.4).

ActorID REQUIRED. Identifier of the entity generating this event.

ChainID REQUIRED. Identifier of the per-Actor hash chain.

SequenceNum REQUIRED. Monotonically increasing sequence number within the Actor's chain.

4.2. Payload Object

The Payload contains domain-specific data organized into modules:

VCP-TRADE Trading data: orders, executions, modifications, cancellations.

VCP-RISK Risk management data: exposure, limits, margin.

VCP-GOV Governance data: algorithm parameters, decision factors, operator actions.

VCP-PRIVACY Privacy metadata: encryption keys, retention policies.

```
{
  "Payload": {
    "VCP-TRADE": {
      "OrderID": "ord-2026-001",
      "Symbol": "AAPL",
      "Side": "BUY",
      "Quantity": "100",
      "Price": "185.50",
      "OrderType": "LIMIT"
    },
    "VCP-GOV": {
      "AlgoID": "momentum-v2.3",
      "DecisionFactors": ["RSI_oversold", "volume_spike"],
      "ConfidenceScore": 0.87
    }
  }
}
```

4.3. Security Object

The Security object contains integrity and chaining information:

```
{
  "Security": {
    "EventHash": "sha256:a1b2c3d4...",
    "PrevHash": "sha256:f6e5d4c3...",
    "MerkleRoot": "sha256:1234abcd...",
    "SignAlgo": "ED25519"
  }
}
```

EventHash REQUIRED. SHA-256 hash of the canonicalized Header and Payload, computed using JSON Canonicalization Scheme [RFC8785].

PrevHash REQUIRED (except for INIT events). Hash of the previous event in this Actor's chain.

MerkleRoot OPTIONAL. Merkle root of the current batch (Gold/Platinum tiers).

SignAlgo REQUIRED. Signature algorithm identifier. MUST be "ED25519" or "DILITHIUM3" (for post-quantum).

4.4. Event Types

Code	Name	Description
INIT	Initialization	Chain initialization, no PrevHash
SIG	Signal	Trading signal generated
ORD	Order	Order submitted
ACK	Acknowledgment	Order acknowledged by venue
EXE	Execution	Order executed (fill)
CXL	Cancellation	Order cancelled
MOD	Modification	Order modified
RSK	Risk	Risk event or limit breach
ERR	Error	System error
HBT	Heartbeat	Periodic liveness signal
CLS	Close	Position closed

ANC	Anchor	Merkle anchor event

Table 2: VCP Event Types

5. Registration Policy

A VCP Transparency Service MUST enforce a Registration Policy that validates incoming Signed Statements. The policy MUST verify:

1. *Schema Compliance:* Payload conforms to VCP Event schema.
2. *Signature Validity:* COSE_Sign1 signature is valid for the registered Issuer public key.
3. *Timestamp Validity:* TimestampInt is within acceptable bounds (not in future, not too old).
4. *Chain Integrity:* PrevHash matches the hash of the most recent event from this Actor (if not INIT).
5. *Sequence Monotonicity:* SequenceNum is exactly one greater than the previous event's SequenceNum.

5.1. Tier-Specific Requirements

This specification distinguishes timestamp resolution from clock accuracy. Nanosecond-resolution timestamps represent the storage format capability, while actual clock accuracy is explicitly recorded and enforced per tier.

Requirement	Silver	Gold	Platinum
Timestamp Resolution	Millisecond	Microsecond	Nanosecond
Clock Accuracy	NTP (~10ms)	NTP + drift (~1ms)	PTPv2 (<1μ s)
Merkle Anchoring	Daily	Hourly	Per-minute
Signature Algorithm	Ed25519	Ed25519	Ed25519 + Dilithium (OPTIONAL)
Key Storage	Software	Software/ HSM	HSM Required
External Anchor	Optional	Recommended	Required

Table 3: Conformance Tier Requirements

Note: Silver tier is NOT intended for regulatory-grade algorithmic trading systems subject to MiFID II RTS 25, SEC Rule 17a-4, or equivalent clock synchronization requirements. Silver tier is appropriate for development, testing, backtesting analysis, and non-regulated trading scenarios.

6. Crypto-Shredding for Privacy Compliance

VCP supports crypto-shredding to enable GDPR-compliant data erasure while preserving audit trail integrity. This mechanism allows deletion of personal data without invalidating cryptographic proofs.

6.1. Mechanism

1. *Encryption:* Sensitive payload fields are encrypted with a per-record or per-subject Data Encryption Key (DEK).

2. *Key Storage:* DEK is stored separately, indexed by SubjectID.

3. *Hashing:* EventHash is computed over the encrypted payload, preserving integrity.

4. *Shredding:* Upon erasure request, DEK is destroyed. Encrypted data becomes unrecoverable.

5. *Verification:* Receipt and hash chain remain valid—verifiers can confirm event existence and ordering without accessing decrypted content.

6.2. VCP-PRIVACY Module

```
{
  "VCP-PRIVACY": {
    "EncryptedFields": ["VCP-TRADE.ClientID", "VCP-TRADE.AccountID"],
    "KeyID": "dek-2026-001-subject-12345",
    "Algorithm": "AES-256-GCM",
    "RetentionPolicy": "GDPR-5Y"
  }
}
```

7. SCRAPI Integration

VCP Transparency Services MUST implement SCRAPI [I-D.ietf-scitt-scrapi] with the following VCP-specific considerations:

7.1. Registration (POST /entries)

VCP Events are submitted as COSE_Sign1 Signed Statements:

```
POST /entries HTTP/1.1
Host: vcp-ts.example.com
Content-Type: application/cose

<COSE_Sign1 containing VCP Event payload>
```

The Transparency Service validates the VCP Registration Policy and returns a COSE Receipt on success.

7.2. Retrieval (GET /entries/{entry_id})

Retrieve a specific VCP Event by its entry ID (derived from EventID).

7.3. Receipt (GET /entries/{entry_id}/receipt)

Retrieve the COSE Receipt for a registered VCP Event, containing the Merkle inclusion proof.

8. Security Considerations

8.1. Chain Integrity

The per-Actor hash chain construction provides tamper evidence: modification of any event invalidates all subsequent hashes in that Actor's chain. Combined with SCITT's global append-only log and Merkle tree, this provides defense in depth.

Mitigations against key compromise:

* Frequent Merkle anchoring to external immutable stores
* HSM-based key storage (Platinum tier requirement)
* Key rotation with explicit ROTATE events

8.2. Quantum Resistance

Ed25519 signatures are vulnerable to attacks by cryptographically relevant quantum computers. VCP provides crypto-agility to address future threats:

* SignAlgo field enables algorithm negotiation and migration
* Post-quantum signature algorithms (e.g., ML-DSA/Dilithium) are OPTIONAL and intended for experimental or high-assurance deployments
* Hash chain integrity based on SHA-256 provides approximately 128-bit post-quantum security against Grover's algorithm

Implementers requiring post-quantum guarantees SHOULD monitor CFRG and PQUIP working group outputs for updated guidance on algorithm selection and migration timelines.

8.3. Timing Attacks

Clock manipulation can enable backdating of events. Mitigations:

* UUID v7 provides embedded timestamp that must match TimestampInt
* Transparency Service enforces timestamp bounds
* Platinum tier requires PTPv2 synchronization
* External anchoring provides independent timestamp attestation

8.4. Privacy Considerations

VCP Events may contain sensitive trading information. Operators SHOULD:

* Use crypto-shredding for personal data subject to GDPR
* Implement access controls on Transparency Service queries
* Consider encrypted payloads for highly sensitive data

9. IANA Considerations

This document has no IANA actions at this time.

Future versions of this specification may request:

* Registration of media type "application/vcp+json"
* Establishment of a VCP Event Type registry
* COSE algorithm identifiers for VCP-specific extensions

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[RFC8785] Rundgren, A., "JSON Canonicalization Scheme (JCS)", RFC 8785, June 2020, <https://www.rfc-editor.org/rfc/rfc8785>.

[RFC9052] Schaad, J., "CBOR Object Signing and Encryption (COSE)", RFC 9052, August 2022, <https://www.rfc-editor.org/rfc/rfc9052>.

[RFC9562] Davis, K., "Universally Unique IDentifiers (UUIDs)", RFC 9562, May 2024, <https://www.rfc-editor.org/rfc/rfc9562>.

[I-D.ietf-scitt-architecture]
Birkholz, H., Delignat-Lavaud, A., and C. Fournet, "An Architecture for Trustworthy and Transparent Digital Supply Chains", Work in Progress, Internet-Draft, draft-ietf-scitt-architecture, 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-scitt-architecture>.

[I-D.ietf-scitt-scrapi]
Steele, O., "SCITT Reference APIs", Work in Progress, Internet-Draft, draft-ietf-scitt-scrapi, 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-scitt-scrapi>.

10.2. Informative References

[RFC6962] Laurie, B., "Certificate Transparency", RFC 6962, June 2013, <https://www.rfc-editor.org/rfc/rfc6962>.

[RFC8032] Josefsson, S., "Edwards-Curve Digital Signature Algorithm (EdDSA)", RFC 8032, January 2017, <https://www.rfc-editor.org/rfc/rfc8032>.

[EU-AI-ACT]
European Parliament and Council, "Regulation (EU) 2024/1689 - Artificial Intelligence Act", 2024.

[MIFID-II] European Parliament and Council, "Directive 2014/65/EU - Markets in Financial Instruments Directive II", 2014.

[FIPS-204] NIST, "Module-Lattice-Based Digital Signature Standard (ML-DSA)", FIPS 204, 2024.

Appendix A. Complete VCP Event Example

The following is a complete VCP Event encoded as JSON, ready to be wrapped in a COSE_Sign1 Signed Statement:

```
{
  "Header": {
    "EventID": "01961e5f-5c0d-7000-8000-123456789abc",
    "TimestampISO": "2026-03-15T09:30:00.123456789Z",
    "TimestampInt": 1742034600123456789,
    "EventType": "ORD",
    "ActorID": "algo-momentum-001",
    "ChainID": "chain-actor-001",
    "SequenceNum": 42
  },
  "Payload": {
    "VCP-TRADE": {
      "OrderID": "ord-2026-001",
      "Symbol": "AAPL",
      "Side": "BUY",
      "Quantity": "100",
      "Price": "185.50",
      "OrderType": "LIMIT",
      "TimeInForce": "DAY"
    },
    "VCP-GOV": {
      "AlgoID": "momentum-v2.3",
      "DecisionFactors": ["RSI_oversold", "volume_spike"],
      "ConfidenceScore": 0.87,
      "RiskCheckPassed": true
    }
  },
  "Security": {
    "EventHash": "sha256:a1b2c3d4e5f6a1b2c3d4e5f6a1b2c3d4e5f6a1b2c3d4e5f6a1b2c3d4e5f6a
1b2",
    "PrevHash": "sha256:f6e5d4c3b2a1f6e5d4c3b2a1f6e5d4c3b2a1f6e5d4c3b2a1f6e5d4c3b2a1f6
e5",
    "SignAlgo": "ED25519"
  }
}
```


Appendix B. JSON Schema Reference

The complete JSON Schema for VCP Events is available at:

https://veritaschain.org/schema/vcp-event-v1.0.json

Acknowledgements

The authors thank the members of the VeritasChain Standards Organization Technical Committee for their contributions to this specification. This work builds upon the SCITT architecture developed by the IETF SCITT Working Group, and the Certificate Transparency work in [RFC6962].

Author’s Address

TOKACHI KAMIMURA
VeritasChain Standards Organization
Japan
Email: kamimura@veritaschain.org
URI: https://veritaschain.org