AES-1

Agent Execution Standard

WORKING PAPER V0.1 — DRAFT FOR COMMENT

PUBLISHED BY	Kadikoy Limited, Bermuda (Reg. 202302362)
DATE	22 May 2026
STATUS	Draft — open for public comment
VERSION	0.1 — initial release
COMPANIONS	AIS-1 Agent Identity Standard (ais-1.org); AAS-1 Agent Auditability Standard (aas-1.org); ARS-1 Agentic Remittance Standard (ars-1.org)
CONTACT	info@aiagentsservices.net
REPOSITORY	github.com/Kadikoy1/aes-1
WEBSITE	aes-1.org
LICENCE	Creative Commons CC0 — no rights reserved

V0.1 KEY FEATURES

Introduces	First open standard for certifying smart-contract execution enclaves
Defines	Five architectural properties (P1–P5) of operator-exclusion
Specifies	Three certification tiers — Basic (I), Verified (II), Sovereign (III)
Schema	JSON Schema 2020-12 Enclave Certificate, anchored on-chain
Binds	Every Enclave Certificate references an AIS-1 bond for the deploying party
Neutral	Technology- and chain-neutral; CAIP-2 identifiers; JCS canonicalisation; SHA-256 default
Statutory hook	Tier III provides the mechanical hook through which a settlement-finality statute attaches to enclave outputs
Composes	Designed to compose with sibling standards: AIS-1 (identity), AAS-1 (audit records), ARS-1 (remittance)

Abstract

AES-1 defines an open standard for certifying **execution enclaves**: smart contracts in which the operator has been architecturally excluded from execution outcomes after deployment. It addresses what we call the **Operator-Surface Problem**: most contracts deployed today retain owner keys, upgrade slots, migration paths, and admin

functions that allow the operator to alter outcomes after counterparties have committed. AES-1 specifies the positive architectural properties that distinguish a true enclave from a contract that merely describes itself as one.

The standard defines five required properties — Non-Upgradeability (P1), No Operator Surface (P2), Deterministic Execution (P3), Participant-Only Interfacing (P4), and On-Chain Verifiability (P5) — and a three-tier classification framework: Basic (self-attested), Verified (independent audit), and Sovereign (audit plus qualified legal opinion in a settlement-finality jurisdiction). Every Enclave Certificate binds to an AIS-1 identity, allowing the deploying party to be cryptographically attributed and held to account.

AES-1 is the execution complement to AIS-1 (which gives agents identity) and AAS-1 (which gives agents an audit trail). Where AIS-1 answers *who is the agent* and AAS-1 answers *what did it do*, AES-1 answers *where does it run, and can the operator change the outcome*. The standard is published as a working paper for public comment, released CC0, with reference schemas and a worked example illustrating a Tier III certificate available at the public repository.

Contents

1. Motivation and Problem Statement

The agentic economy depends on the ability of autonomous agents to transact with one another at scale, with finality, and without trusted operators standing between them. Every commercial flow of any size — institutional disbursement, agentic loan settlement, regulated agent-to-agent commerce — assumes that the smart contract through which the flow executes is what it appears to be: a deterministic, unmodifiable, operator-neutral piece of code.

In practice, most contracts deployed today fail that test. They include upgrade keys, owner-only functions, pausable execution, migration paths, proxy patterns, oracle override surfaces, and access-controlled administrative endpoints. Some are necessary for early-stage development. Many are vestigial. All of them mean the operator remains in the execution loop. A counterparty engaging with such a contract — or with an agent acting through it — cannot know whether the operator will alter the outcome after commitment.

We term this the **Operator-Surface Problem**. The consequences are already visible:

- Audit firms report which control surfaces exist, but cannot certify the resulting environment as an enclave.
- Insurance markets cannot underwrite execution risk without bespoke per-contract review at every renewal.
- Settlement-finality regimes have no architectural definition to attach statutory finality to.
- Counterparties rely on operator reputation rather than architectural certainty.
- "Operator-excluded" claims are marketing assertions, not verifiable properties.

Existing standards do not solve this. Smart-contract audits report findings against a checklist of known anti-patterns but do not certify that a contract *is* an enclave. EIPs around proxy patterns (ERC-1967, ERC-1822, ERC-2535) standardise upgradeability rather than rule it out. Settlement-finality statutes in major financial jurisdictions assume bank-operated systems and have not been extended to autonomous on-chain execution.

AES-1 closes the gap by defining the architectural properties that, together, make a contract an enclave, and by binding those properties to an on-chain Enclave Certificate that any third party can verify from public chain data alone.

2. Definitions

TERM	DEFINITION
`Execution enclave`	A smart contract or set of smart contracts in which, once deployed, the deploying party and any other privileged party have no architectural surface through which to alter execution outcomes.
`Operator`	Any party — natural, legal, or autonomous — that holds privileged access to a deployed contract through an owner key, multisig, governance vote, upgrade mechanism, or equivalent.
`Operator exclusion`	The architectural condition in which no operator possesses a surface through which to modify execution outcomes after deployment. AES-1 certifies operator *exclusion*, not operator *absence*.

`Participant`	Any party — natural, legal, or autonomous — that interacts with the enclave via its callable interface on the same terms as any other participant.
`Enclave Certificate`	The on-chain artefact issued upon successful AES-1 certification. Contains contract address, deployment hash, satisfied tier, per-property attestations, and the AIS-1 identity bond of the deploying party.
`Sovereign Enclave`	An AES-1 Tier III enclave operating in a jurisdiction that has enacted settlement-finality legislation recognising its outputs as legally final.
`Certifying party`	The party that issues the Enclave Certificate. At Tier I, the deploying party themselves (self-attested). At Tier II, an independent auditor co-attests. At Tier III, a Tier II auditor plus a qualified legal opinion.
`CAIP-2 identifier`	Chain identifier per CAIP-2 (e.g. `eip155:1`, `eip155:42161`, `eip155:8453`). Used throughout the Enclave Certificate schema.

3. The AES-1 Standard

3.1 Operator Exclusion

AES-1 is built around a single architectural principle: **operator exclusion**. After an enclave is deployed, the operator must have no surface through which to act on execution. The deployment is the operator's last act with respect to that enclave; from then on, the code intermediates between participants without the operator in the loop.

Operator exclusion is a property of the *architecture*, not the *conduct*. AES-1 does not certify that the operator chose not to interfere; it certifies that there is no surface through which they *could*. This distinction is decisive: conduct-based assurance requires ongoing supervision; architectural certification requires only that the architecture be verifiable.

3.2 Enclave Certificate

Every AES-1 certified execution enclave is issued an Enclave Certificate — a structured, signed, on-chain-anchored record that captures the identity, properties, and classification of the enclave in machine-readable form. The certificate is the unit of AES-1 — anchored on the deployment chain, fetchable by any third party, independently verifiable.

The full schema is normatively defined at Appendix A. At minimum every Enclave Certificate contains: the contract address(es); the chain identifier; the tier achieved; per-property attestations (all five required); the AIS-1 bond of the deploying party; the certifying party's identity; verification evidence (audit-report and legal-opinion IPFS CIDs where applicable); issuance and expiry dates.

4. The Five Properties

To qualify as an AES-1 certified execution enclave, an execution environment must satisfy **all five** of the properties below. Each is independently verifiable from public chain data alone. The properties are conjunctive: partial compliance is not graded — it is not certified.

PROPERTY	REQUIRES	PROHIBITS	VERIFICATION METHOD
P1 · Non-Upgradeability	Immutable deployed bytecode	Proxy patterns, upgrade keys, migration functions, `selfdestruct`	Bytecode inspection, proxy slot check
P2 · No Operator Surface for Discretionary Control	No discretionary callable functions affecting outcomes; statutory halt permitted within § 4.2.1 constraints	Discretionary `onlyOwner` on execution; selective suspension; outcome alteration	Source / ABI review + halt-trigger inspection
P3 · Deterministic Execution	All execution paths fully specified at deployment	Operator-modifiable execution variables	Formal verification, code audit
P4 · Participant-Only Interfacing	Equal callable access to all eligible participants	Operator-controlled access-control lists	Interface and modifier inspection
P5 · On-Chain Verifiability	All properties verifiable from public chain data	Reliance on operator disclosure for verification	Block explorer, verified source

4.1 P1 — Non-Upgradeability

> *The contract logic is immutable post-deployment. No proxy pattern, upgrade key, migration function, or equivalent mechanism exists through which the deployed bytecode may be modified or replaced by any party, including the deploying address.*

Non-upgradeability is the foundational property. It ensures that what was deployed is what runs — permanently. A contract that can be upgraded is not an enclave because its future behaviour is not fully determined by its deployment state.

Disqualifying patterns. Transparent proxy, UUPS proxy, beacon proxy, diamond pattern with mutable facets, `delegatecall` to mutable implementation, `selfdestruct` instruction, owner-controlled migration, time-locked admin upgrade.

4.2 P2 — No Operator Surface for Discretionary Control

> *The contract exposes no function, modifier, or access-control mechanism callable at the discretion of a privileged address that can alter execution outcomes, redirect assets, modify participant balances, or selectively suspend execution. A symmetric, one-way statutory halt triggered exclusively by attested external conditions enumerated in the Enclave Certificate is permitted within the constraints of § 4.2.1 and does not constitute an operator surface for the purposes of this property.*

P2 is the positive architectural property that defines the enclave. It is not the proposition that the operator *chose* not to act, but that there is **no surface through which they could act at their own discretion** to change outcomes.

Disqualifying patterns. `onlyOwner` modifiers permitting discretionary calls on execution-critical functions; pausable patterns with discretion-controlled pausers; fee-parameter setters callable by admin; treasury withdrawal functions; blacklist or freeze mechanisms targeting specified participants; oracle override functions; selective transaction-level suspension or unwind.

Permitted. Read-only privileged views (telemetry); participant-symmetric governance that affects only future participants and cannot retroactively alter outcomes; emergency exits available to *all* participants on equivalent terms; a statutory halt mechanism conforming to § 4.2.1.

4.2.1 Statutory Halt

A real-world enclave is rarely permitted to be unstoppable. Sanctions designations land; courts issue orders; supervisors direct suspension; chain-level catastrophes occur. A standard that forbade any halt at all would forbid every regulated deployment, push deploying parties into the construction of perpetual statutory liability — the "contract you cannot turn off" — and disqualify operations that the supervising regulator would otherwise license.

P2 therefore distinguishes **discretionary control over outcomes** (prohibited) from **statutory halt under attested external triggers** (permitted, within strict constraints). A halt mechanism is property-conforming when it satisfies all of the following:

- **One-way.** The halt suspends future execution only. It cannot reverse, redirect, unwind, or alter any outcome already produced by the enclave prior to invocation.
- **Symmetric.** The halt operates on the enclave as a whole. It cannot be invoked against specified participants, addresses, transactions, or asset classes selectively.
- **Triggered, not exercised.** Invocation requires attested evidence of an external condition listed in the certificate's `haltTriggers` array. The party invoking has the role of *recognising* a trigger, not of *choosing* whether to act.
- **Disclosed.** Every permitted trigger is enumerated in the Enclave Certificate at issuance — trigger type, form of attested evidence required, and the AIS-1 identity of the party authorised to attest each trigger. The list cannot be expanded post-issuance without re-certification.

- **Recorded.** Every halt invocation produces an AAS-1 Class A attestation referencing the trigger that fired and the evidence consulted. The audit trail is part of the enclave's permanent record.

Recognised trigger types in v0.1:

TRIGGER TYPE	EVIDENCE REQUIRED
`court_order`	Final order of a court of competent jurisdiction. CID of the order document + court identifier.
`regulator_directive`	Written direction of the supervising regulator. CID of the directive + regulator's AIS-1 identity.
`sanctions_match`	Addition of a participant or counterparty to a recognised sanctions list. List reference + match attestation by an AIS-1-bonded screener.
`statutory_event`	Event statutorily defined as triggering suspension under the relevant settlement-finality regime. Oracle attestation + statutory citation.
`protocol_oracle`	Protocol-level oracle attestation of an event whose handling is statutorily prescribed. Oracle signature + oracle's AIS-1 identity.

Implementations MAY define additional trigger types under a reverse-DNS namespace. Verifiers MAY reject unknown trigger types unless published in a recognised registry.

Tier interaction. At Tier I and Tier II, an enclave MAY include a halt mechanism within the constraints above; this is the expected configuration for deployments operating under a digital-asset business licence in a supervised jurisdiction. At Tier III, halt triggers MUST mirror the suspension conditions of the relevant settlement-finality regime; the statutory protection conferred by the regime is bounded by precisely those conditions, and the certificate's halt-trigger list is what binds them on-chain.

Halt of last resort. An enclave with an empty or absent `haltTriggers` field has no halt mechanism and operates as a non-stoppable contract. This configuration is permitted by the standard but is generally incompatible with regulated deployment and produces the statutory-liability surface described at § 9. The standard does not require a halt; it requires that, if there is a halt, it conforms.

4.3 P3 — Deterministic Execution

> *Identical inputs produce identical outputs. All execution paths are fully specified at deployment, with no operator-modifiable parameters and no probabilistic state that could be steered post-hoc.*

Determinism is what makes execution verifiable. Without it, the enclave's outputs cannot be independently re-derived; without independent re-derivation, the architectural certification collapses.

Verification. Formal verification at Tier II/III. At Tier I, source review against a published determinism checklist.

4.4 P4 — Participant-Only Interfacing

> *Every party interacts with the enclave via its callable interface on the same terms as every other party. There is no asymmetric access path — no operator-controlled allow-list, deny-list, or privileged caller.*

Participant-symmetric access is the structural counterpart to operator exclusion: if there is no operator in execution, there can also be no operator-mediated routing of who is permitted to call. Access is gated only by economic preconditions any party may satisfy on identical terms.

4.5 P5 — On-Chain Verifiability

> *An enclave claim that cannot be independently re-verified is not an enclave claim — it is an assertion. AES-1 requires that every property be checkable by any third party with access to the blockchain, without reliance on operator disclosure.*

P5 closes the loop. The deploying address, deployment transaction, contract address, and verified source must all be publicly recorded. No off-chain representation by the operator is required to verify enclave status. P5 is what makes the certification trust-minimised: a counterparty does not need to trust the deployer, the auditor, or the certifying party — they can run the checks themselves.

5. Three Tiers

AES-1 defines three tiers of enclave certification. All three require satisfaction of the five core properties. The tiers differ in the level of independent verification, the governance sophistication of the certification, and the legal status that attaches.

TIER	VERIFICATION	LEGAL STATUS	USE CASES
`I — Basic Enclave`	Self-attested by deploying party; verified source published; on-chain certificate	None required	Experimentation, early-stage deployment, retail-scale agent commerce
`II — Verified Enclave`	Tier I + independent audit by AES-1-recognised auditor + formal verification of P3	None required	Counterparty-grade assurance; institutional commercial flows; AIPS-1 standard underwriting
`III — Sovereign Enclave`	Tier II + qualified legal opinion in a settlement-finality jurisdiction	Statutory finality (jurisdiction-dependent)	Regulated agent-to-agent settlement; statutorily-final A2A commerce; sovereign-backed insurance

Tier III is the convergence point of architectural and statutory finality. The reference settlement-finality regime is Bermuda's draft Digital Asset Settlement and Stablecoin Act (DASSA); other jurisdictions enacting similar regimes can recognise Tier III without bespoke definition of "enclave" in primary legislation — the standard does that work.

6. Schema Specification

6.1 JSON Schema

The Enclave Certificate schema is published as JSON Schema 2020-12 at `github.com/Kadikoy1/aes-1/blob/main/schemas/aes-1-certificate.schema.json`. It is normative for v0.1. The full structure is at Appendix A.

6.2 Canonicalisation and Hashing

Certificates MUST be canonicalised before hashing or signing. v0.1 specifies JCS (RFC 8785) as the default canonicalisation, matching AAS-1 §6.2 and ARS-1 §9.2. The hash algorithm is declared per-certificate; SHA-256 is the default. Implementations MAY use other algorithms by setting the `hashAlg` field; verifiers SHOULD reject unknown algorithms unless published in a registry.

6.3 Signature Object

FIELD	DESCRIPTION
`alg`	Signature algorithm identifier (EdDSA, ES256K, etc.)

`hashAlg`	Hash algorithm. Default SHA-256.
`canonicalisation`	Canonicalisation method. Default JCS.
`keyRef`	Verification method identifier within the issuer's AIS-1 identity document.
`value`	The signature value, base64url-encoded.

Structurally identical to AAS-1 §6.3 and ARS-1 §9.3 — shared verification tooling across the stack.

7. AIS-1 Binding

Every AES-1 Enclave Certificate references an AIS-1 bond for the **deploying party**. This is the juridical link between the operator-excluded enclave and a person (natural, corporate, or autonomous-legal under ALARGA) who is accountable for having deployed it.

At Tier I, the AIS-1 bond is informational but required. At Tier II, the auditor's own AIS-1 bond is also required. At Tier III, a third AIS-1 bond is required for the counsel issuing the legal opinion.

Operator exclusion does not eliminate responsibility for the act of deployment. AIS-1 is what carries that responsibility into the on-chain record.

8. On-Chain Anchoring

The signed Enclave Certificate is anchored on-chain by hashing the canonical body and writing the hash to a public registry. On EVM chains, anchoring is via the AES-1 Registry — a single deployment per chain mapping enclave addresses to certificate hashes. Equivalent registry primitives may be implemented on non-EVM chains. Multi-chain enclaves are anchored on the primary chain identified by `subject.chain`, with mirror anchors permitted on other chains.

The full anchoring guidance for each supported chain is at `docs/on-chain-anchoring.md` in the repository.

9. Regulatory Significance

9.1 Settlement Finality

Settlement finality is the legal status conferred on a payment such that, once made, it cannot be unwound by an insolvency proceeding of either party. It is the foundation of modern wholesale payment systems — the EU Settlement Finality Directive 98/26/EC, the US Article 4A UCC, and various netting statutes including Bermuda's. No jurisdiction has, to date, extended settlement-finality protection to autonomous on-chain execution.

AES-1 Tier III is designed as the technical hook through which a settlement-finality statute — in any jurisdiction that enacts one — can attach to autonomous on-chain execution. The legal opinion required at Tier III is what maps the architectural certification onto the statutory definition; the statute's enumerated suspension conditions become the certificate's halt triggers (§ 4.2.1).

9.2 POCR, FATF and Business Relationships

The Bermuda Proceeds of Crime Regulations and equivalent frameworks in other jurisdictions impose obligations on persons conducting "relevant financial business" involving "business relationships". AES-1 provides the technical substrate for arguments that interactions with a certified execution enclave do not constitute a business relationship in the relevant sense — because the operator is structurally absent from the execution of those interactions. AES-1 does not resolve all such questions, but it provides the common vocabulary and the verifiable architectural posture against which a considered regulatory assessment can be conducted.

9.3 Consumer Protection and Conduct Regulation

Conduct regulation typically attaches to entities that intermediate between a client and a financial outcome. In a certified execution enclave, the operator does not intermediate; the code does. AES-1 does not determine what conduct obligations attach to the deployer of a certified enclave, but it provides the factual foundation for a considered regulatory assessment.

9.4 Statutory Liability and Halt Conformance

A contract that cannot be halted under any circumstance is a perpetual liability surface for whoever deployed it. If a bug is later identified, a sanctions designation lands on a participant, a court issues a freezing order, or a regulator directs suspension, the deploying party has no compliant response. The risk is structurally similar to legacy environmental-liability regimes: the obligation to remediate attaches to a person, but the means of remediation has been architecturally foreclosed.

This is also the supervisory pressure point against licensability. A digital-asset business supervisor will not licence an operation whose core execution venue cannot, under any conditions, be suspended on regulatory direction. The supervisor's prudential mandate forbids it; the deploying party's professional advisors will not recommend it.

The § 4.2.1 Statutory Halt mechanism resolves this. By admitting a halt surface that is (a) one-way, (b) symmetric, (c) triggered only by attested external conditions, (d) disclosed in the Enclave Certificate, and (e) recorded as an AAS-1 attestation on every invocation, AES-1 preserves the architectural integrity of operator exclusion in respect of *outcomes* while permitting the recognised forms of statutory intervention. This is the same posture every supervised settlement system holds: SWIFT, SEPA, Fedwire, and EU-Settlement-Finality-Directive-designated systems all permit suspension on enumerated grounds; none is "operator-controlled" in the discretionary sense.

At Tier III, the alignment becomes strict. The relevant settlement-finality regime confers its protection only on transfers that have not been suspended under its own enumerated conditions; the Enclave Certificate's `haltTriggers` array must mirror those conditions exactly. The halt is not a compromise of finality — it is the boundary of finality, made on-chain-readable.

10. Security Considerations

10.1 Certificate Integrity

Every Enclave Certificate MUST be signed by the deploying party's AIS-1 bond key. At Tier II/III, additional signatures from the auditor and counsel are required. Tampering is detectable through hash recomputation against the on-chain anchor.

10.2 Chain-Level Changes

A chain-level change (hard fork, EIP introducing operator-modifiable execution state, fundamental opcode change) can compromise an underlying property of an already-certified enclave. The certifying party SHOULD revoke and re-issue in such cases. Periodic re-attestation at Tier III provides a fixed window within which chain-level changes are reviewed.

10.3 Composition Hazards

An enclave that calls into a non-enclave contract inherits the operator surface of the called contract. AES-1 v0.1 does not certify enclaves with such dependencies; v0.2 may introduce a "composite enclave" pattern with strict transitive constraints.

10.4 Verification Tooling

P5 requires that verification be possible from public chain data alone, but in practice most verifications will be performed via tooling that interprets that data. The verification tooling itself should be open-source, auditable, and ideally itself an AES-1 enclave. The v0.2 automated verification toolchain is intended to satisfy this property.

11. Implementation Roadmap

PHASE	DELIVERABLE	TARGET
`0.1 — This document`	Specification, JSON Schema, Clavus worked example, public website, repository, CI validation.	May 2026
`0.2 — Tooling and DID method`	Automated P1–P5 verification toolchain. AES-1 Registry deployed on Ethereum, Arbitrum, Base, and equivalent EVM chains. First Tier I/II certifications issued.	Q3 2026
`0.3 — Sovereign Enclave`	Tier III framework finalised. Settlement-finality integration documented. Formal legal-opinion template. Multi-chain extensions to non-EVM environments.	Q4 2026
`1.0 — Standardisation`	Submission to IEEE, ISO, IETF. Full public registry. Automated certification pipeline. First Tier III Sovereign Enclave issued under an enacted settlement-finality regime.	2027

12. Request for Comment

AES-1 v0.1 is published as a draft for public comment. Feedback is invited from smart-contract developers and auditors; AI agent developers and framework maintainers; blockchain engineers and protocol designers; legal, regulatory and compliance professionals; institutional participants in digital-asset markets; insurance underwriters; government and regulatory bodies; and standards organisations including IEEE, ISO, IETF, and W3C.

The comment period for v0.1 closes **30 June 2026**. A revised v0.2 will incorporate substantive feedback. Submissions:

- Feedback form: aes-1.org/#feedback
- Email: info@aiagentsservices.net
- GitHub: github.com/Kadikoy1/aes-1/issues

13. Authors

FIELD	VALUE
Author	Kadikoy Limited, Bermuda
Affiliation	BDA Law; BDA AI Agent Services
Companions	AIS-1 (ais-1.org); AAS-1 (aas-1.org); ARS-1 (ars-1.org)
Contact	info@aiagentsservices.net
Website	aes-1.org
Repository	github.com/Kadikoy1/aes-1
Licence	Creative Commons CC0. No rights reserved. Open for free implementation.

Appendix A — Enclave Certificate Schema (Excerpt)

The canonical JSON Schema 2020-12 is at `github.com/Kadikoy1/aes-1/blob/main/schemas/aes-1-certificate.schema.json`. The certificate carries the following top-level fields:

FIELD	TYPE	REQUIRED	DESCRIPTION
`schema`	string (URI)	Yes	Schema URI. Pinned.
`version`	string	Yes	AES-1 specification version (0.1).
`certificateHash`	string	Yes	Content-addressed identifier: the canonical hash of the certificate body. Unique by construction.
`subject`	object	Yes	Chain (CAIP-2), contracts, deployment tx, deployer address.
`tier`	object	Yes	Level (I/II/III), label, and (Tier III) jurisdiction.
`properties`	object	Yes	Per-property attestations P1–P5. All five must be satisfied.
`issuer`	object	Yes	Self-attestation; auditor (Tier II+); legal opinion (Tier III).
`evidence`	object	Yes	Formal-verification artefacts, source release CID, additional evidence.
`issuedAt`	RFC 3339	Yes	Issuance timestamp.
`haltTriggers`	array	Recommended	Enumerated halt triggers per § 4.2.1. Absence means no halt mechanism (see § 4.2.1).
`expiresAt`	RFC 3339	Tier III only	Expiry (annual re-attestation at Tier III).
`anchor`	object	—	Chain, transaction hash, and block of the on-chain anchor.
`revocation`	object	—	If present, the certificate has been revoked.

Appendix B — Tier III Worked Example

An illustrative Tier III Sovereign Enclave certificate is published at `examples/tier3-example-certificate.json` in the repository. The certificate below shows a generic Tier III deployment on an EVM-compatible chain, certified under a settlement-finality regime in the home jurisdiction of the deploying party. Selected fields:

```
{
  "certificateHash": "sha256-7d865e959b2466918c9863afca942d0fb89d7c9ac0c99bafc3749504ded97730",
  "subject": {
    "chain": "eip155:<chainId>",
    "contracts": [
      { "address": "0x…", "role": "settlement", "verifiedSource": "<repository URI>" },
      { "address": "0x…", "role": "registry",   "verifiedSource": "<repository URI>" }
```

```
      ],
      "deploymentTx": "0x…"
    },
    "tier": { "level": "III", "label": "Sovereign Enclave", "jurisdiction": "<ISO 3166-1 alpha-2>" },
    "properties": {
      "P1": { "satisfied": true, "method": "Bytecode inspection; proxy slot check" },
      "P2": { "satisfied": true, "method": "Source code and ABI review; halt-trigger inspection per § 4.2.1" },
      "P3": { "satisfied": true, "method": "Formal verification" },
      "P4": { "satisfied": true, "method": "Interface and modifier inspection" },
      "P5": { "satisfied": true, "method": "Block explorer; verified source" }
    },
    "haltTriggers": [
      { "type": "court_order",         "evidence": "order CID + court identifier",
        "attestorRef": "did:ais1:<deployer>" },
      { "type": "regulator_directive", "evidence": "directive CID + regulator identity",
        "attestorRef": "did:ais1:<deployer>" },
      { "type": "sanctions_match",     "evidence": "list reference + match attestation",
        "attestorRef": "did:ais1:<screener>" },
      { "type": "statutory_event",     "evidence": "oracle attestation + statutory citation",
        "attestorRef": "did:ais1:<oracle>" }
    ],
    "issuer": {
      "selfAttestation": { "aisBondRef": "did:ais1:<deployer>",          "signature": "0x…" },
      "auditor":         { "aisBondRef": "did:ais1:<auditor>",           "reportCid": "baf…" },
      "legalOpinion":    { "counselAisBondRef": "did:ais1:<counsel>",
                           "jurisdiction": "<ISO 3166-1 alpha-2>",
                           "statutoryFinalityActRef": "<citation to applicable statute>" }
    },
    "issuedAt": "2026-07-15T00:00:00Z",
    "expiresAt": "2027-07-15T00:00:00Z"
  }
```

Appendix C — Verification Flow

How a third party verifies an AES-1 Enclave Certificate:

1. Fetch the certificate JSON from the registry or counterparty.
2. Validate it against the JSON Schema.
3. Recompute the canonical hash (RFC 8785 JCS + SHA-256) and confirm it matches the on-chain anchor.
4. Verify the signature(s) against the issuer's, auditor's, and (for Tier III) counsel's AIS-1 bond keys via AIS-1 resolution.
5. Independently re-run the P1–P5 verification methods, or accept the cited evidence CIDs as sufficient.
6. Confirm the certificate is within its validity window and not revoked.

Steps 1–4 and 6 are mechanical and may be performed by tooling. Step 5 is the substantive verification and may be performed by the relying party, by their counsel, or by an AES-1-recognised auditor.

End of AES-1 v0.1 specification. Published 22 May 2026 by Kadikoy Limited, Bermuda. Released under CC0 1.0 Universal — no rights reserved. Comment closes 30 June 2026.