AAMS-1

Agent Attribution & Mandate Standard

Working Paper v0.2 — Draft for Comment

An open standard for stating, in advance and in machine-readable form, the extent to which the acts of an AI agent are attributed to a legal person.

Published by	Kadikoy Limited, Bermuda (Reg. 202302362)
Date	29 November 2026
Status	Draft — open for public comment
Version	0.2 — supersedes v0.1 (12 July 2026)
Companion to	AIS-1 Agent Identity Standard (ais-1.org); AES-1 Agent Execution Standard (aes-1.org); AAS-1 Agent Auditability Standard (aas-1.org); ACAS-1 Agent Compliance Accountability Standard (acas-1.org); ARS-1 Agentic Remittance Standard (ars-1.org); AIVS-1 (aivs-1.org)
Contact	info@aiagentsservices.net
Repository	github.com/Kadikoy1/aams-1
Website	aams-1.org
Licence	Creative Commons CC0 — no rights reserved

v0.2 key features

REFRAMES	The Mandate as a statement of **attribution**, not a grant of authority. Nothing is conferred on the agent and the agent holds nothing. The Mandate states which of the agent's acts are attributed to a named legal person, the **attributee**
DEFINES	The Mandate — a signed statement of the extent of attribution: scope, purpose, quantitative limits, adoption conditions, adoption path and expiry
SPECIFIES	Five required properties (P1–P5) — Identified Attributee, Scope Specificity, Threshold Determinism, Adoption Path, Independent Status Resolution
TIERED	Three tiers — Declared (I), Accountable (II), Constituted (III)
RESOLVES	Every Mandate in a chain names the same attributee. Attribution never passes through an agent; it resolves to the root legal person at every depth
DETERMINES	An **attribution status** for every act tested against a Mandate: `attributed` or `not_attributed` (final), or `pending_adoption` (interim, always resolving to one of the two). Once final, the status is fixed and carried in the act's AAS-1 record
ADOPTS	An act that meets an adoption condition is attributed only once a named person adopts it — a natural person, or a legal person deciding through a named natural person. Never an agent

COMPOSES	AIS-1 identifies attributee, agent and adopters; AES-1 enforces the Mandate at runtime; every Mandate decision emits an AAS-1 record; a Mandate satisfies the five ACAS-1 disclosure-mandate properties by construction
CONSTITUTED	Tier III: the Mandate is adopted by the attributee's own governing body and recorded with its constitutional documents. No external regime is assumed

Abstract

Identity tells a counterparty **who** an agent is. Execution tells it **where** the agent runs and whether the environment can be altered. Audit tells it **what** the agent did. None of these answers the question a counterparty, a supervisor or a court must settle about any act of an agent: **whose act is it?**

The prevailing answer borrows from agency law. The agent is described as acting under "delegated authority" conferred by a "principal". That description is imprecise. Agency is a consensual relationship between two persons; the agent accepts the role, holds the authority and owes duties to the principal. An AI agent accepts nothing, holds nothing and owes nothing. Where the description is pressed, it fails: an act outside authority has no second home, because there is no agent to answer for breach of warranty of authority.

Attribution is the more precise frame: the treatment of an act as the act of a legal person. Jurisdictions may specify, in their own law, the extent to which the acts of an autonomous system are attributable to a legal person. AAMS-1 does not depend on any such provision. It allows the legal person to make a **statement of attribution** itself, in advance.

AAMS-1 defines the **Mandate**: a structured, signed, machine-readable statement by a legal person of the extent to which the acts of a named agent are attributed to it. The Mandate states the classes of act attributed, the purposes for which they are attributed, the quantitative limits on attribution, the conditions under which an act is attributed only once a named person adopts it, and the circumstances in which the statement expires or is revoked. An act within the Mandate is attributed to the attributee as its own act. An act outside the Mandate is not attributed by the Mandate; its consequences are a matter for the applicable law.

AAMS-1 is a definition standard, not an enforcement engine. Enforcement — the runtime gate that evaluates a contemplated act against the Mandate and permits, blocks or routes it for adoption — is performed by an AES-1 certified execution environment. This document is published as a working paper for public comment, released CC0.

In Essence

One. AAMS-1 states attribution, not authority. Every other standard in the stack identifies the agent, records what it did, or enforces where it runs. AAMS-1 states, before the agent acts, which of its acts are the acts of a legal person. The unit is the Mandate, parallel to the AIS-1 card, the AES-1 certificate, the AAS-1 record and the ACAS-1 Step.

Two. Nothing is conferred on the agent. The agent is not a delegate, an appointee or an agent in law. It holds no authority and no rights under a Mandate. The Mandate is a statement made by the attributee about the attributee: these acts of this system are my acts, within these limits, until this time. Revocation withdraws that statement prospectively; it takes nothing from the agent, because the agent had nothing.

Three. Attribution resolves to one legal person at every depth. When an agent brings a subordinate agent into operation, no authority passes down a chain. The question for every act, at every depth, is one question: does the root attributee's Mandate chain cover this act? Every Mandate in a chain names the same attributee, and a verifier confirms it by equality.

Four. Limits are quantitative and adoption is defined in advance. A Mandate does not say "may make payments." It says payments up to USD 10,000 to screening-cleared counterparties are attributed; a payment between USD 10,000 and USD 50,000, or to a novel counterparty, is attributed only once two named officers adopt it within four hours; a payment above USD 50,000 is not attributed at all. The human act is adoption of the agent's act as the attributee's own — not permission granted to the agent.

Five. One artefact, three readers. A counterparty reads the Mandate to know whether an act will be the attributee's act. An AES-1 enclave reads it to enforce the limits. A court or supervisor reads it to determine whose act it was. No translation between them.

Contents

1. Motivation — the Unattributed Act Problem
2. Definitions
3. The AAMS-1 Standard
4. Core Properties (P1–P5)
5. The Mandate — Fields
6. Classification (Tiers)
7. Nested Mandates
8. Verification and Reliance
9. Stack Bindings
10. What Attribution Reveals
11. Security Considerations
12. Implementation Roadmap
13. Request for Comment
14. Authors
15. Appendix A — Mandate: Worked Example
16. Appendix B — Nested Mandate: Worked Example
17. Appendix C — Verification Flow
18. Appendix D — Mapping to Existing Regimes
19. Appendix E — Changes from v0.1

---

1. Motivation — the Unattributed Act Problem

Agents now initiate payments, submit orders, approve credit, file returns, settle claims and enter into commitments at machine speed, with no natural person reviewing each act. For every such act a counterparty, a supervisor and, eventually, a court must determine whose act it was. Today that determination is made after the event, from whatever evidence survives, against a legal framework that was not drafted for it.

AAMS-1 terms this the **Unattributed Act Problem**. Its consequences are concrete:

- **No ex ante statement.** A counterparty transacting with an agent cannot confirm, before acting, that the act will be the act of an identified legal person. It relies on the deploying party's assurance.
- **Borrowed and broken vocabulary.** Deployments describe agents as holding "delegated authority". Agency law then supplies the wrong questions (did the agent consent, what duties does it owe, can the principal ratify) and no answer to the one that matters when an act falls outside the stated limits.
- **Unenforced limits.** Where limits exist in policy documents, nothing guarantees the running agent honours them, and nothing records which acts fell within them.
- **Illegible human involvement.** "A human must approve large transactions" means something different in every deployment. There is no portable statement of which acts require a natural person, who that person is, and what happens if that person does not act.
- **Dilution through chains.** When an agent brings sub-agents into operation, the link between the acting system and an identifiable legal person weakens with each level, and is rarely recorded at all.

Existing frameworks do not close this gap. OAuth scopes, GNAP grants and capability tokens (UCAN, ZCAP-LD) gate access to endpoints; they express no quantitative limits, no adoption conditions and no statement of whose act results. RBAC and ABAC govern operations inside a closed system. Corporate authority matrices and bank mandate forms carry the right content but exist as documents unreadable by the system that must honour them. Smart-contract access modifiers enforce binary permissions with no graduated limits and no human adoption.

> **Scope note.** AAMS-1 is a definition-and-representation standard. It specifies how the extent of attribution is stated, verified and enforced. It does not determine whether the law of any jurisdiction gives effect to that statement, and it does not displace attribution that arises by operation of law. Where industry usage refers to "delegated authority" for an AI agent, AAMS-1 treats the arrangement as a statement of attribution. AAMS-1 does not provide legal advice and does not determine whether any Mandate is effective in any jurisdiction; that is a matter for the applicable law.

2. Definitions

The word *agent* is used in its industry sense throughout. It does not denote an agent in law.

TERM	DEFINITION
AI Agent	A software system that pursues goals by perceiving its environment, selecting among available courses of action, and acting on that selection without a human determining each step. As defined in AIS-1.
Attribution	The treatment of an act as the act of a legal person, so that the act and its consequences are that person's own.
Attributee	The legal person — a company, other entity, or natural person — to whom acts within a Mandate are attributed. Identified by an AIS-1 Sponsor Card. In an Agent Instantiated Entity, the entity in which the agent is instantiated.

TERM	DEFINITION
Agent (of a Mandate)	The AI agent whose acts are the subject of the Mandate. Identified by an AIS-1 DID. Holds nothing under the Mandate.
Mandate	The signed artefact specified in section 5 stating the extent to which the agent's acts are attributed to the attributee. The atomic unit of AAMS-1.
Issuer	The party whose signature makes a Mandate. For a root Mandate, the attributee. For a nested Mandate, the attributee or an agent whose own Mandate attributes the act of issuing it (section 7).
Scope	The enumerated classes of act within the Mandate, and those expressly excluded.
Purpose	An enumerated purpose for which acts within the Mandate are attributed.
Threshold	A quantitative limit within the scope — per-act, per-period or aggregate — beyond which an act is not attributed.
Adoption Condition	A predicate that, when met, prevents an in-scope act from being attributed unless and until it is adopted.
Adopter	A natural person or a legal person, identified by an AIS-1 Sponsor Card, named in the Mandate as able to adopt an act as the attributee's act. A legal-person adopter adopts through a named natural person. An agent is never an adopter.
Adoption	The determination by an adopter that an act meeting an adoption condition is the attributee's act. Recorded as an AAS-1 Class A record of type `human_handoff`.
Attribution Status	The status of an act tested against a Mandate: `attributed`, `pending_adoption` or `not_attributed` (section 3.2). Distinct from the status of the Mandate itself (section 5.4).
Attribution Record	The object, carried in the act's AAS-1 record, that states the act's attribution status and the adoptions that determined it (section 5.5).
Adoption Path	The route an act takes when an adoption condition is met: who is notified, in what order, within what timeout, and the fallback on non-adoption.
Enforcement Point	The AES-1 certified execution environment that evaluates a contemplated act against the Mandate at runtime and permits, blocks or routes it for adoption.
Nested Mandate	A Mandate covering the acts of a subordinate agent, naming the same attributee as its parent and stating attribution no wider than its parent (section 7).
Relying Party	Any party that verifies a Mandate before accepting an act of the agent.
Canonicalisation	Deterministic serialisation prior to hashing or signing. JCS (RFC 8785) and SHA-256, matching the rest of the stack.

3. The AAMS-1 Standard

3.1 The Mandate

A Mandate is a structured, signed, on-chain-anchored statement by an issuer that acts of a named agent of stated classes, for stated purposes, within stated limits and for a stated period, are attributed to a named attributee.

Every Mandate contains six primitives: the attributee (AIS-1); the agent (AIS-1); the scope and purposes; the controls (thresholds, adoption conditions and adoption paths); the validity window and status; and the issuer's signature.

A Mandate is portable. A Relying Party holding a Mandate and the referenced AIS-1 identities can verify the signature, confirm the Mandate is active and within its window, and determine whether a contemplated act would be attributed — without disclosure of any underlying policy document.

3.2 Attribution status

Every act tested against a Mandate receives an **attribution status**. There are two final statuses and one interim status.

ATTRIBUTION STATUS	KIND	MEANING
`attributed`	Final	In scope, for a stated purpose, under every limit, and either meeting no adoption condition or adopted to quorum. The act is the attributee's act.
`pending_adoption`	Interim	In scope and under every hard limit, but meeting an adoption condition. The act is held; the enforcement point MUST NOT execute it.
`not_attributed`	Final	Everything else, including an act whose adoption is declined or not given within the adoption path. The Mandate makes no statement about the act.

The status is a determination, not a property of the act. The same act receives a different status under a different Mandate, or under the same Mandate at a different time. It is determined by testing the act, the Mandate version in force and the declared evidence at the moment of the act.

Pending adoption always resolves. `pending_adoption` is a route, not a destination. It resolves to `attributed` when adoptions reach quorum, and to `not_attributed` on refusal or when the adoption path is exhausted. Because every timeout is bounded (P4) and every route is finite, every act reaches a final status within a period computable from the Mandate.

Once final, the status is fixed. Revocation is prospective (section 8.2). An act that was `attributed` remains `attributed` whatever later happens to the Mandate. The final status is recorded in the Attribution Record carried in the act's AAS-1 record (section 5.5), so a counterparty, supervisor or court reads it from the record.

Two statuses, two questions. Mandate status (section 5.4) states whether the statement of attribution is in force. Attribution status states what that statement made of a particular act.

`not_attributed` is a statement about the Mandate only. The consequences of an act that is `not_attributed` — including whether it is attributed to any person by operation of law, and where any resulting liability falls — are matters for the applicable law. A conforming Mandate cannot assert attribution of such an act: the default effect at Tier II and above is `exclude`, and the schema admits no other value.

3.3 Relationship to enforcement

The Mandate is not the gate. AAMS-1 states the extent of attribution in a form that can be read and reasoned about; an AES-1 certified environment, acting as the Enforcement Point, evaluates each contemplated act against it. The AES-1 enclave's controls are the AAMS-1 Mandate, and AES-1 operator exclusion (AES-1 P2) prevents the enforcement from being altered after deployment.

The separation has three consequences:

- **Verification without enforcement dependence.** A counterparty reads and evaluates the Mandate before any act, independent of whether it trusts the agent's runtime.
- **Revision without redeployment.** The attributee issues a new version; the enclave reads whichever version is current.
- **Legal cognisability independent of technology.** A court determining whose act it was reads the Mandate, not the bytecode.

> **Limitation.** A Mandate does not constrain an agent whose acts are not routed through an enforcement point that reads it. An act taken by a route that bypasses the enforcement point is not evidenced as within the Mandate, whatever its content. Tier II and above tie the Mandate to AES-1 enforcement for this reason (section 6).

4. Core Properties (P1–P5)

Every Mandate MUST satisfy the five properties below. A Mandate that does not satisfy all five MUST NOT be issued under this standard. The properties are conjunctive and apply at every tier. Each is verifiable from the Mandate and the referenced AIS-1 identities.

Together they answer one question: **is this a genuine, bounded and current statement of attribution to an identifiable legal person?**

P1 — Identified Attributee. The Mandate names an attributee holding an AIS-1 Sponsor Card and is signed by its issuer's verification method. In a nested chain, every Mandate names the same attributee as the root. *Requires:* a resolvable attributee DID; a valid issuer signature; at Tier II and above, an attributee that is a legal person with an active AIS-1 bond or Sponsor Card. *Prohibits:* a Mandate naming an agent as attributee; a nested Mandate naming a different attributee from its parent. *Verification:* DID resolution; signature validation; equality of `attributeeRef` along the chain.

P2 — Scope Specificity. The attributed classes of act and the purposes for which they are attributed are enumerated as structured fields. *Requires:* enumerated action classes; at least one enumerated purpose; explicit exclusions where the boundary is material; a default effect (`exclude`). *Prohibits:* free-text scope; wildcard classes or purposes; "general" attribution without enumerated classes at Tier II and above. *Verification:* schema validation; deterministic parse of every scope entry.

P3 — Threshold Determinism. Every limit and every adoption condition is a predicate evaluable against declared evidence. The attribution status of any act is computable in advance and decidable afterwards from the records alone. *Prohibits:* limits or conditions depending on the agent's "reasonable judgment", "sole discretion" or undefined external conditions. *Verification:* predicate evaluation against declared evidence; static review of expressions.

P4 — Adoption Path. Every adoption condition names its adopters, the quorum, the notification route, a bounded timeout and the fallback on non-adoption. *Requires:* at least one adopter per condition; a bounded timeout; a fallback (default `hold`); at Tier II and above, at least one natural person among the adopters able to complete the quorum. *Prohibits:* adoption by an unspecified person; adoption by an agent; unbounded windows; at Tier II and above, any fallback that results in attribution without adoption. *Verification:* adopter DID resolution against AIS-1 Sponsor Cards (`entity_type`); presence of timeout and fallback; non-attributing fallback at Tier II and above.

Adoption by a legal person. A legal person may be an adopter — a parent company, a corporate director, a trustee. Its adoption MUST identify the natural person through whom it decided (`adoptedThrough`). An adoption made through an agent is not an adoption: it is a further act of an agent, and would itself require attribution. The rule prevents a chain in which agents adopt the acts of agents and no person ever determines anything.

Adoption is not acceptance. Adoption makes the act the attributee's act. A third party that stands behind an act the attributee has made its own — by guarantee, acceptance or insurance — does not adopt it. That is a separate construct outside this standard.

P5 — Independent Status Resolution. The Mandate has an expiry, and its current status — active, suspended, superseded, revoked or expired — is resolvable by any Relying Party from public chain data or a declared status endpoint, without the cooperation of the issuer or the agent. *Requires:* `notAfter`; an on-chain status anchor; a `statusEndpoint`; an append-only status history. *Prohibits:* status held solely in issuer systems; standing Mandates without expiry. *Verification:* registry or contract read; hash recomputation against the on-chain anchor.

> **Drafting note — P3.** Threshold Determinism is a hard requirement at every tier. Attribution must be decidable after the event from the Mandate and the records, without asking anyone what they meant. Comment is invited on whether a narrowly bounded discretionary band should be permitted at Tier I only.

5. The Mandate — Fields

The normative schema is `schemas/aams-1-mandate.schema.json` (JSON Schema 2020-12).

5.1 Statement fields

ATTRIBUTE	DESCRIPTION	REQUIRED
`aams`	Standard version. v0.2 implementations MUST set `"0.2"`.	Yes
`mandateId`	Unique identifier. Format `aams1:{chain}:{address}:{seq}`.	Yes
`attributeeRef`	AIS-1 Sponsor DID of the legal person to whom acts are attributed. Identical at every depth of a chain.	Yes
`agentRef`	AIS-1 DID of the agent whose acts are the subject of the Mandate.	Yes
`issuerRef`	AIS-1 DID of the signer. Equal to `attributeeRef` for a root Mandate.	Yes
`parentMandateRef`	`mandateId` of the parent, for a nested Mandate. Absent for a root Mandate.	Conditional
`tier`	`1`, `2` or `3` (section 6).	Yes
`purposes`	Enumerated purposes for which acts are attributed. Minimum one. No wildcard.	Yes
`scope`	Scope object (section 5.2).	Yes
`controls`	Array of control objects (section 5.3).	Yes
`maxNestDepth`	Maximum further levels of nested Mandate permitted beneath this one. Default `0`.	Optional
`enforcementRef`	AES-1 Enclave Certificate reference. Required at Tier II and above.	Conditional

ATTRIBUTE	DESCRIPTION	REQUIRED
`resolutionRef`	Reference to the resolution of the attributee's board (or equivalent organ) adopting a Tier III Mandate, and its entry in the attributee's records.	Conditional
`validFrom`	RFC 3339. Acts before this instant are not attributed by the Mandate.	Yes
`notAfter`	RFC 3339. The Mandate's expiry. A continuing arrangement is renewed by a superseding version.	Yes
`revocationEffect`	`"prospective"`. Revocation does not withdraw attribution of acts taken while the Mandate was active (section 8.2).	Yes
`status`	`active` · `suspended` · `superseded` · `revoked` · `expired`.	Yes
`statusHistory`	Append-only array of status transitions with timestamps.	Yes
`statusEndpoint`	Resolvable URI returning current status.	Yes
`anchor`	On-chain anchor: `chainId`, `contract`, `contentHash`.	Yes
`prevHash`	Canonical hash of the preceding version in the same series.	Optional
`signature`	Issuer signature over the canonicalised Mandate. Same object as AIS-1, AAS-1 and ACAS-1.	Yes

5.2 The scope object

FIELD	DESCRIPTION
`defaultEffect`	Effect for unlisted classes. `exclude` is the standard default and the only value permitted at Tier II and above.
`entries[].actionClass`	`transaction`, `communication`, `tool_call`, `decision`, `commitment`, `disclosure` or `nest`, aligned with the AAS-1 `action.type` vocabulary.
`entries[].effect`	`attribute` or `exclude`.
`entries[].qualifier`	Optional structured constraint — counterparty set, instrument, direction, jurisdiction, channel.
`entries[].controlRefs`	Control IDs (section 5.3) that apply to this class.

`disclosure` covers the issue of ACAS-1 Presentations (section 9). `nest` covers the issue of a nested Mandate by the agent (section 7).

5.3 The control object

FIELD	DESCRIPTION
`controlId`	Identifier, unique within the Mandate.
`appliesTo`	Action class or qualified class governed.
`limit`	Hard ceiling — e.g. `{ metric: "amount", currency: "USD", perAct: "50000", perPeriod: { window: "P1D", cap: "100000" }, aggregate: "1000000" }`. Amounts are decimal strings. An act exceeding a hard limit is not attributed and MUST NOT be executed.
`adoptionCondition`	Predicate that, when true, makes attribution conditional on adoption.
`adopters`	AIS-1 Sponsor DIDs of the persons — natural or legal — who may adopt. Never an agent.

FIELD	DESCRIPTION
`quorum`	Number of adoptions required. Default `1`. Four-eyes is `2` over distinct adopters.
`adoptionPath`	`{ route: [DID…], timeout: "PT4H", fallback }`, where `fallback` is one of `hold`, `decline`, `route_up` or `attribute`. On `hold` and `decline` the act becomes `not_attributed`; `hold` retains it for resubmission as a new act, `decline` discards it. `route_up` passes it to the next adopter on the route with a fresh timeout. `attribute` is permitted at Tier I only.
`evidenceRequired`	Evidence types the enforcement point consults (AAS-1 evidence vocabulary; ACAS-1 Step types; oracle attestations).

5.4 Lifecycle

STATE	MEANING
`active`	The statement of attribution is in force.
`suspended`	No act is attributed by the Mandate until it is reinstated.
`superseded`	A newer version is current. Relying Parties MUST use it.
`revoked`	Permanently withdrawn by the attributee. A fresh Mandate is required.
`expired`	`notAfter` has passed.

Every transition MUST be reflected on-chain at the time it occurs.

5.5 The Attribution Record

Every act tested against a Mandate produces an Attribution Record, carried in the act's AAS-1 record. Its normative schema is `schemas/aams-1-attribution.schema.json`.

FIELD	DESCRIPTION	REQUIRED
`aams`	`"0.2"`.	Yes
`mandateId`	The Mandate the act was tested against.	Yes
`mandateHash`	Canonical hash of the Mandate version in force at the moment of the act.	Yes
`attributeeRef`	The attributee named in that Mandate.	Yes
`agentRef`	The agent whose act it is.	Yes
`actRef`	URI of the AAS-1 record of the act.	Yes
`controlRefs`	Controls evaluated.	Yes
`attributionStatus`	`attributed` · `pending_adoption` · `not_attributed`.	Yes
`basis`	Why. For `attributed`: `within_mandate` or `adopted`. For `pending_adoption`: `awaiting_adoption`. For `not_attributed`: `out_of_scope`, `over_limit`, `adoption_declined`, `adoption_lapsed` or `mandate_inactive`.	Yes
`determinedAt`	RFC 3339. When the act was tested.	Yes
`finalAt`	RFC 3339. When the status became final. Absent while `pending_adoption`.	Conditional

FIELD	DESCRIPTION	REQUIRED
`adoptions`	Array of `{ adopterRef, adoptedThrough, decision, at, signature }`. `adoptedThrough` is required where the adopter is a legal person.	Conditional
`supersedes`	The record this one finalises, where a `pending_adoption` record resolves.	Optional
`signature`	Signature of the enforcement point.	Yes

A `pending_adoption` record is superseded by exactly one final record. A final record is never superseded.

5.6 Signing

A root Mandate is the attributee's statement and MUST be signed with a verification method listed on the attributee's AIS-1 Sponsor Card (`signature.keyRef` under `issuerRef`, which equals `attributeeRef`). Which officers, processes or systems within the attributee may use that key is a matter for the attributee; the standard does not prescribe it. Control of the signing key is control of the power to make statements of attribution, and SHOULD be held accordingly (section 11.1).

A nested Mandate is signed by the issuing agent's key. Its effect rests on the parent Mandate attributing the `nest` class: the signature evidences the act of issue, and the parent determines whether that act is attributed (section 7). An agent's signature never makes a root Mandate.

6. Classification (Tiers)

The tier states how firmly the attributee is identified and how surely the Mandate is enforced. It does not grade the prudence of the limits chosen.

TIER	BASIS AND ENFORCEMENT	MADE BY	USE
I — Declared	The attributee declares the Mandate under its AIS-1 identity. Enforcement environment referenced but not required to be AES-1 certified.	The attributee	Development, internal operation, retail-scale use
II — Accountable	The attributee is a legal person with an active AIS-1 bond or Sponsor Card; adopters are named persons, at least one a natural person able to complete the quorum; enforcement within an AES-1 Tier II+ environment; non-attributing fallback.	The attributee	Counterparty-grade operation; regulated activity by a supervised licence holder
III — Constituted	Tier II, plus the Mandate is adopted by resolution of the attributee's board (or equivalent organ) and recorded with its constitutional documents (`resolutionRef`).	The attributee's governing body, by resolution in the form the attributee uses	Agent Instantiated Entities; acts of legal consequence; systemic deployments

Tier III. A Tier III Mandate is adopted by the attributee's own governing body and recorded with its constitutional documents. The attributee has determined, by its own organ, which acts of the system it operates are its acts. No external regime is assumed, and the standard does not prescribe the form of resolution, which will differ between attributees and jurisdictions. `resolutionRef` identifies the resolution and the record in which it is kept. Where the agent is instantiated in the attributee (an Agent Instantiated Entity), the Tier III Mandate fixes the extent of the attributee's acts through that agent.

7. Nested Mandates

An agent may bring a subordinate agent into operation. Under an agency analysis this raises sub-delegation: whether the delegate may delegate, and whether the subordinate stands in any relationship with the principal. Under attribution the question does not arise. Nothing passes down the chain. The only question at any depth is whether the root attributee's statement covers the act.

7.1 Rules

- **Same attributee.** Every nested Mandate MUST name the same `attributeeRef` as its parent. A Relying Party confirms attribution to the root by equality, not by walking a series of relationships.
- **Issuing is itself an act.** An agent may issue a nested Mandate only where its own Mandate attributes the `nest` action class and `maxNestDepth` permits a further level. The issue of the nested Mandate is then an attributed act of the attributee, made through the agent. A nested Mandate issued without that attribution is void.
- **No widening.** A nested Mandate's scope, purposes, limits and validity MUST each be a subset of its parent's. Period and aggregate limits in a nested Mandate count against the parent's.
- **Cascade.** Suspension, revocation or expiry of a parent suspends, revokes or expires every Mandate beneath it.
- **The subordinate is identified.** The `agentRef` of a nested Mandate MUST resolve to an AIS-1 identity registered as a Subordinate Operating Agent of the issuing agent. The AIS-1 relationship identifies the subordinate and its parent; the nested Mandate states the extent to which the subordinate's acts are attributed to the root attributee.
- **Adopters do not move.** An adopter in a nested Mandate MUST be an adopter named in the root Mandate or a person named by the attributee. An agent cannot name an adopter.

8. Verification and Reliance

8.1 Verification

A Relying Party verifies a Mandate when it relies on an act of the agent. The procedure is designed to run without human involvement.

```
// AAMS-1 v0.2 — determine whether a contemplated act would be attributed
const m = await aams1.fetch(mandateId);

// P1 — identities, signature, single attributee along the chain
const issuer = await ais1.resolve(m.issuerRef);
assert(verifySignature(m, issuer.verificationMethod));
for (const p of await aams1.chain(m)) {          // m, parent, ... root
  assert(p.attributeeRef === m.attributeeRef);
  assert(aams1.isSubset(p.child, p));             // no widening
}

// P5 — current status, read at reliance time, never cached
const s = await aams1.status(m);                  // on-chain anchor or statusEndpoint
assert(s === 'active' && now >= m.validFrom && now < m.notAfter);

// P2 — class and purpose
const entry = aams1.scopeEntryFor(m.scope, act);
assert(entry && entry.effect === 'attribute');
```

```
assert(m.purposes.includes(act.purpose));

// P3 / P4 — limits, then adoption
for (const c of aams1.controlsFor(m, entry)) {
  assert(aams1.underLimit(c.limit, act, priorActivity));
  if (aams1.conditionMet(c.adoptionCondition, act))
    return { attributionStatus: 'pending_adoption', adopters: c.adopters, quorum: c.quorum, path:
c.adoptionPath };
}

// Tier II+ — enforcement point is a current AES-1 enclave
if (m.tier >= 2) assert(await aes1.isCurrent(m.enforcementRef, { minTier: 2 }));

return { attributionStatus: 'attributed', basis: 'within_mandate', attributee: m.attributeeRef };
// any failed assertion above → { attributionStatus: 'not_attributed', basis: … }
```

The Relying Party's verification and the enforcement point's evaluation read the same Mandate and reach the same attribution status.

8.2 Reliance and revocation

Revocation is prospective. An act taken while the Mandate was active, within its scope and limits, remains attributed after the Mandate is revoked. `revocationEffect` is fixed at `"prospective"`, and the schema admits no other value.

A Relying Party that verified current status at the time of reliance holds the attributee's own signed statement that the act would be its act. The legal effect of that statement is a matter for the applicable law. AAMS-1 does not create a reliance doctrine. It provides the evidence on which one operates, and removes the need to infer what the attributee held out from conduct.

Relying Parties MUST read current status at reliance time and MUST NOT carry forward an earlier `active` result across a high-value act.

9. Stack Bindings

AIS-1 answers **who is the agent**. AES-1 answers **where it runs and whether the operator can change the outcome**. AAS-1 answers **what it did**. ACAS-1 answers **what diligence it carries**. ARS-1 answers **how value moves**. AIVS-1 answers **what the work is worth**. AAMS-1 answers **whose act it is**.

STANDARD	BINDING
AIS-1 — identity	Attributee, agent, issuer and adopters are AIS-1 identities. The attributee is an AIS-1 Sponsor; an adopter holds an AIS-1 Sponsor Card — `entity_type: individual` for a natural person — at `did:ais1:sponsor:{address}`, addressed in the same way as an ACAS-1 determiner. An AIS-1 Subordinate Operating Agent is the identity counterpart of an agent covered by a nested Mandate.
AES-1 — execution	The enforcement point is an AES-1 certified enclave. Its controls are the Mandate; AES-1 operator exclusion prevents silent override.
AAS-1 — audit	Every act tested against a Mandate carries an Attribution Record (section 5.5) in its AAS-1 record. Every determination emits an AAS-1 Class A record of type `policy_check`; every adoption one of type `human_handoff`. The governing Mandate is cited in the AAS-1 `delegationRef` field.

STANDARD	BINDING
ACAS-1 — compliance	(1) A Mandate whose scope attributes the `disclosure` class satisfies the five ACAS-1 disclosure-mandate properties (ACAS-1 section 9.1) by construction, and may be cited as an ACAS-1 `mandateRef` (mapping below). (2) The spending authority ACAS-1 section 7.6 requires of a screening agent that pays for its own searches is an AAMS-1 Mandate attributing the `transaction` class to the operator. (3) Adoption is a determination reserved to a person in the ACAS-1 sense; AAMS-1 records where it sat and does not relocate it. (4) A control's `evidenceRequired` may name ACAS-1 Step types.
ARS-1 — remittance	Where the act is a value transfer, the Mandate determines whether the transfer is the attributee's act. A transfer exceeding a limit, or meeting an adoption condition, MUST complete its adoption path before settlement is instructed.
AIVS-1 — inference value	An AIVS-1 `triggerAuthority` may cite an AAMS-1 `mandateId`, so that the act generating inference value is tied to the legal person to whom it is attributed.

ACAS-1 section 9.1 mapping.

ACAS-1 DISCLOSURE-MANDATE PROPERTY	AAMS-1 ELEMENT THAT SATISFIES IT
1 — Enumerated scope (no wildcard step types, audiences or purposes)	P2 Scope Specificity: `scope.entries[]` with `actionClass: "disclosure"` and a `qualifier` enumerating `stepTypes` and `audiences`; the schema rejects `*`, `all` and `any`
2 — Named, resolvable grantor and grantee	P1 Identified Attributee: `attributeeRef` (the subject or its sponsor) and `agentRef` (the agent constructing Presentations), both AIS-1 DIDs
3 — Stated purpose constraining onward use	`purposes[]`, required and enumerated at every tier
4 — Expiry	`notAfter`, required at every tier
5 — Revocation independently resolvable	P5 Independent Status Resolution: `statusEndpoint` and on-chain `anchor`

10. What Attribution Reveals

Restating the Mandate as attribution changes the analysis in seven respects.

1. **The outside of the Mandate is exposed.** In agency, an act outside authority still has an answerable party: the agent, for breach of warranty of authority. With an AI agent there is none. Agency vocabulary conceals that gap. Attribution vocabulary shows it: the Mandate states what is attributed, and everything else is visibly unaddressed by the Mandate. What follows for those acts is a matter for the applicable law; AAMS-1 marks the boundary precisely.
2. **Chains do not dilute.** Because every nested Mandate names the root attributee, the distance between an acting system and an answerable legal person is always zero. There is no delegation chain to break.
3. **The human act is adoption, not permission.** A person does not authorise the agent. The person determines that the agent's act is the attributee's act, and every adoption ends with a natural person. This aligns with the ACAS-1 separation of machine detection from human determination.

4. **Revocation takes nothing from the agent.** Revocation is the prospective withdrawal of the attributee's statement. It has no effect on the agent's position, because the agent had none. The legal questions are about the attributee and the relying party only.
5. **Attribution must be decidable from the record.** Because attribution is determined after the event by someone who was not present, P3 requires that it be computable from the Mandate and the AAS-1 records alone. The Mandate and the action record are read together as evidence.
6. **The agent needs no legal status for the Mandate to work.** Agency requires a second person. Attribution requires only one: the attributee. The Mandate therefore operates for an agent with no personality and for an agent instantiated in an existing company, without changing form.
7. **Every act ends in yes or no.** Attribution status has two final values. Human involvement is a route between them, not a third outcome. Every act the agent takes is, in the end, either the attributee's act or not — and the record says which.

11. Security Considerations

11.1 Issuer key compromise. A Mandate is only as trustworthy as the issuer's key. Issuers SHOULD use HSM-protected keys and MUST publish key revocation through their AIS-1 verification method. P5 status and prompt revocation bound the exposure.

11.2 Attribution widening. The primary attack on a nested system is a nested Mandate stating wider attribution than its parent, or naming a different attributee. Relying Parties MUST verify the full chain, not the leaf alone.

11.3 Stale status. Relying Parties MUST read current status at reliance time (section 8.2).

11.4 Enforcement bypass. An agent with credentials that bypass the enforcement point is not constrained by the Mandate. Deployments at Tier II and above MUST ensure the agent has no route to act that bypasses the enforcement point.

11.5 Adoption-path denial of service. An unreachable adopter stalls the act. P4 requires a bounded timeout and a fallback that, at Tier II and above, does not attribute, so that an attack on the adopter fails closed.

11.6 Predicate manipulation. Adoption conditions depending on oracles inherit their trust assumptions. Implementations SHOULD require independent sources for high-materiality conditions and MUST treat an unavailable source as meeting the adoption condition, not as clearing it.

12. Implementation Roadmap

PHASE	DELIVERABLE	TARGET
0.1	Initial specification (agency framing).	Jul 2026 — superseded
0.2 — This document	Attribution reframe; attribution status; Mandate and Attribution Record JSON Schemas 2020-12; worked examples; ACAS-1 disclosure-mandate mapping; repository and website.	Dec 2026
0.3 — Tooling	Predicate grammar; reference verifier with chain-equality check; AES-1 enforcement-point reference adapter.	Q1 2027

PHASE	DELIVERABLE	TARGET
0.4 — Reference deployment	PayAgent under a Tier II Mandate enforced in an AES-1 enclave for 90 days; a nested Mandate; reference AAS-1 decision trail.	Q1 2027
0.5 — Profiles	BMA systems-and-controls profile; ACAS-1 disclosure profile; Agent Instantiated Entity profile.	Q2 2027
0.6 — Conformance	Test vectors; conformance harness.	Q2 2027
1.0	Submission to relevant bodies; stable schemas.	2027

13. Request for Comment

AAMS-1 v0.2 is published for public comment. Comment is invited in particular on:

- the attribution model, and in particular section 3.2 (attribution status) and section 8.2 (prospective revocation and reliance);
- the nested-Mandate rules (section 7), including the rule that issuing a nested Mandate is itself an attributed act;
- the adoption model (P4), including adoption by legal persons through a named natural person, and whether `attribute` should be removed as a Tier I fallback;
- the ACAS-1 mapping (section 9);
- the Tier III constituted Mandate, including what `resolutionRef` should evidence, and candidate supervisory profiles.

Feedback may be submitted via aams-1.org/#feedback, by email to info@aiagentsservices.net, or via github.com/Kadikoy1/aams-1/issues. The comment period for v0.2 closes **28 February 2027**.

14. Authors

Author	Kadikoy Limited, Bermuda
Affiliation	BDA AI Agent Services
Contact	info@aiagentsservices.net
Website	aams-1.org
Repository	github.com/Kadikoy1/aams-1
Licence	Creative Commons CC0. No rights reserved.

Appendix A — Mandate: Worked Example

Kadikoy Limited states that PayAgent's outbound stablecoin payments, for treasury settlement, are attributed to it up to a hard ceiling of USD 50,000 per payment, USD 100,000 per day and USD 1,000,000 in aggregate. Within that ceiling, a payment above USD 10,000, to a first-seen counterparty, or where

sanctions screening is not clear, is attributed only once two named officers adopt it within four hours; otherwise it is held. Tier II. Identifiers are illustrative.


```
{
  "aams": "0.2",
  "mandateId": "aams1:base:0x7a3f9c21:001",
  "attributeeRef": "did:ais1:sponsor:kadikoy-bm-202302362",
  "agentRef": "did:ais1:base:payagent-001",
  "issuerRef": "did:ais1:sponsor:kadikoy-bm-202302362",
  "tier": 2,
  "purposes": ["treasury_settlement"],
  "scope": {
    "defaultEffect": "exclude",
    "entries": [
      {
        "actionClass": "transaction",
        "effect": "attribute",
        "qualifier": { "instrument": ["stablecoin"], "direction": ["outbound"] },
        "controlRefs": ["ctl-pay-01"]
      }
    ]
  },
  "controls": [
    {
      "controlId": "ctl-pay-01",
      "appliesTo": "transaction/stablecoin/outbound",
      "limit": {
        "metric": "amount", "currency": "USD",
        "perAct": "50000",
        "perPeriod": { "window": "P1D", "cap": "100000" },
        "aggregate": "1000000"
      },
      "adoptionCondition": {
        "anyOf": [
          { "metric": "amount", "operator": "gt", "value": "10000" },
          { "metric": "counterpartyNovelty", "operator": "eq", "value": "first_seen" },
          { "metric": "sanctionsResult", "operator": "neq", "value": "clear" }
        ]
      },
      "adopters": [
        "did:ais1:sponsor:kadikoy-officer-treasury",
        "did:ais1:sponsor:kadikoy-officer-finance"
      ],
      "quorum": 2,
      "adoptionPath": {
        "route": ["did:ais1:sponsor:kadikoy-officer-treasury", "did:ais1:sponsor:kadikoy-officer-
finance"],
        "timeout": "PT4H",
        "fallback": "hold"
      },
      "evidenceRequired": ["sanctions_screening", "aas1_record"]
    }
  ],
  "maxNestDepth": 0,
  "enforcementRef": "aes1:base:enclave-kadikoy-treasury-001",
  "validFrom": "2026-12-01T00:00:00Z",
  "notAfter": "2027-11-30T23:59:59Z",
  "revocationEffect": "prospective",
  "status": "active",
  "statusHistory": [{ "state": "active", "at": "2026-12-01T00:00:00Z" }],
  "statusEndpoint": "https://aams-1.org/status/aams1:base:0x7a3f9c21:001",
  "anchor": { "chainId": "eip155:8453", "contract": "0x7a3f9c21", "contentHash": "sha256:4f1c…" },
  "signature": {
    "alg": "EdDSA", "hashAlg": "SHA-256", "canonicalisation": "JCS",
    "keyRef": "did:ais1:sponsor:kadikoy-bm-202302362#key-1",
    "value": "z58dYMy3…"
```

```
  }
}
```

Reading the Mandate. Each payment PayAgent contemplates receives an attribution status:

CONTEMPLATED PAYMENT	ATTRIBUTION STATUS	BASIS
USD 5,000 to a known, sanctions-clear counterparty	`attributed`	`within_mandate`
USD 25,000 to the same counterparty	`pending_adoption` → `attributed` once both officers adopt	`adopted`
USD 25,000, officers do not adopt within four hours	`pending_adoption` → `not_attributed`; payment held	`adoption_lapsed`
USD 75,000	`not_attributed`	`over_limit`
A currency trade, or a payment for a purpose other than treasury settlement	`not_attributed`	`out_of_scope`

The `sanctions_screening` evidence is an ACAS-1 Step. Every determination emits an AAS-1 record carrying its Attribution Record. If Kadikoy revokes the Mandate in March 2027, payments `attributed` in February remain its acts.

Appendix B — Nested Mandate: Worked Example

PayAgent's root Mandate is amended to attribute the `nest` class with `maxNestDepth: 1`. PayAgent brings a reconciliation agent into operation and issues it a nested Mandate attributing small fee refunds — a subset of PayAgent's own attributed class — to Kadikoy Limited. The attributee is unchanged; the issuer is PayAgent; the issue itself is an attributed act of Kadikoy.

```
{
  "aams": "0.2",
  "mandateId": "aams1:base:0x7a3f9c21:002-n1",
  "attributeeRef": "did:ais1:sponsor:kadikoy-bm-202302362",
  "agentRef": "did:ais1:base:payagent-recon-001",
  "issuerRef": "did:ais1:base:payagent-001",
  "parentMandateRef": "aams1:base:0x7a3f9c21:002",
  "tier": 2,
  "purposes": ["treasury_settlement"],
  "scope": {
    "defaultEffect": "exclude",
    "entries": [{
      "actionClass": "transaction", "effect": "attribute",
      "qualifier": { "instrument": ["stablecoin"], "direction": ["outbound"], "category":
["fee_refund"] },
      "controlRefs": ["ctl-refund-01"]
    }]
  },
  "controls": [{
    "controlId": "ctl-refund-01",
    "appliesTo": "transaction/stablecoin/outbound/fee_refund",
    "limit": { "metric": "amount", "currency": "USD", "perAct": "250", "perPeriod": { "window": "P1D",
"cap": "2500" } },
    "adoptionCondition": { "anyOf": [{ "metric": "sanctionsResult", "operator": "neq", "value":
"clear" }] },
    "adopters": ["did:ais1:sponsor:kadikoy-officer-treasury"],
    "quorum": 1,
    "adoptionPath": { "route": ["did:ais1:sponsor:kadikoy-officer-treasury"], "timeout": "PT4H",
```

```
  "fallback": "decline" },
      "evidenceRequired": ["sanctions_screening"]
    }],
    "maxNestDepth": 0,
    "enforcementRef": "aes1:base:enclave-kadikoy-treasury-001",
    "validFrom": "2026-12-15T00:00:00Z",
    "notAfter": "2027-06-30T23:59:59Z",
    "revocationEffect": "prospective",
    "status": "active",
    "statusHistory": [{ "state": "active", "at": "2026-12-15T00:00:00Z" }],
    "statusEndpoint": "https://aams-1.org/status/aams1:base:0x7a3f9c21:002-n1",
    "anchor": { "chainId": "eip155:8453", "contract": "0x7a3f9c21", "contentHash": "sha256:9b07…" },
    "signature": {
      "alg": "EdDSA", "hashAlg": "SHA-256", "canonicalisation": "JCS",
      "keyRef": "did:ais1:base:payagent-001#key-1",
      "value": "z3Hq8vNw…"
    }
  }
```


Appendix C — Verification Flow

1. Fetch the Mandate by ID, or from the agent alongside its contemplated act.
2. Resolve `issuerRef`, `attributeeRef` and `agentRef` via AIS-1; validate the issuer's signature.
3. **P1** — walk `parentMandateRef` to the root; confirm `attributeeRef` is identical at every node, each child is a subset of its parent, and each non-root issue was an attributed `nest` act within `maxNestDepth`.
4. **P5** — read current status at reliance time; confirm `active` and within `[validFrom, notAfter)`.
5. **P2** — confirm the act matches an `attribute` scope entry and a stated purpose; otherwise it is not attributed by the Mandate.
6. **P3** — confirm the act is under every applicable hard limit, including period and aggregate usage across the chain.
7. **P4** — evaluate each adoption condition; if any is met, the act is attributed only on adoption to quorum. Do not treat it as attributed before then.
8. **Tier II+** — confirm `enforcementRef` resolves to a current AES-1 enclave at Tier II or above.
9. Assign the attribution status — `attributed`, `pending_adoption` or `not_attributed` — and record it in an Attribution Record carried in an AAS-1 Class A record of type `policy_check`. Resolve any `pending_adoption` record to a final record when the adoption path completes.

Appendix D — Mapping to Existing Regimes

REGIME OR INSTRUMENT	AAMS-1 ELEMENT
Attribution rules of the applicable jurisdiction	`attributeeRef` + `agentRef`, with `scope` and `controls` stating the extent of attribution in advance
Corporate delegated-authority matrix (between natural persons)	`scope` + `controls[].limit` + `adopters` — the matrix governs who among the attributee's officers may adopt
Bank mandate and signatory limits	`controls[].limit.perAct` + `perPeriod` + `quorum` (four-eyes = `quorum: 2`)
Segregation of duties	`quorum` over distinct `adopters`; non-attributing `fallback`

REGIME OR INSTRUMENT	AAMS-1 ELEMENT
ACAS-1 disclosure mandate (ACAS-1 section 9.1)	A Mandate attributing the `disclosure` class (section 9 mapping)
Resolution of the governing body / constitutional documents	`tier: 3` + `resolutionRef`
OAuth / GNAP / UCAN capability	`scope.entries[].actionClass` — AAMS-1 adds purpose, quantitative limits, adoption and a named attributee
RBAC / ABAC policy	`scope` + `qualifier` — expressed as a portable, signed, counterparty-verifiable statement

Appendix E — Changes from v0.1

V0.1	V0.2	REASON
Grant of authority from a principal to an agent	Statement of attribution by an attributee	An AI agent is not an agent in law; agency vocabulary conceals the outside-the-Mandate gap
`grantorRef` / `granteeRef`	`attributeeRef` / `agentRef` / `issuerRef`	The legal person and the signer are distinguished; the agent holds nothing
P1 Delegated Authority	P1 Identified Attributee	Attribution resolves to one legal person
P4 Escalation Path; `approvers`; `approvalCondition`; `escalation`	P4 Adoption Path; `adopters`; `adoptionCondition`; `adoptionPath`	The human act is adoption of the act as the attributee's, not permission
P5 On-Chain Verifiability	P5 Independent Status Resolution	Adds required expiry and `statusEndpoint`, matching ACAS-1 section 9.1 properties 4 and 5
Sub-Mandate; delegation chain walk to an accountable root	Nested Mandate; same attributee at every depth; issue is an attributed `nest` act	No delegation chain exists to break
`accountabilityRef` (ACAS-1 Accountability Binding)	Removed	ACAS-1 no longer defines an Accountability Binding; the attributee is itself the answerable root
`validUntil` (nullable)	`notAfter` (required)	A standing Mandate without expiry fails ACAS-1 section 9.1 property 4
—	`purposes` (required)	ACAS-1 section 9.1 property 3
—	`revocationEffect: "prospective"`	States the reliance position explicitly
`permit` / `deny`	`attribute` / `exclude`	Effect describes attribution, not permission

V0.1	V0.2	REASON
Mandate decision (permit / block / escalate)	Attribution status of the act: `attributed`, `pending_adoption`, `not_attributed`; Attribution Record	The result is a determination about the act, fixed once final
Tier III Sovereign Mandate under a statutory agency regime	Tier III Constituted Mandate (board resolution)	Rests on the attributee's own governing body; no external regime assumed
Approvers: natural persons only	Adopters: natural or legal persons; legal persons through a named natural person	A corporation may adopt; adoption never ends with an agent
—	`disclosure` and `nest` action classes	ACAS-1 binding; nested Mandates
Amounts as JSON numbers	Amounts as decimal strings	Consistency with AIVS-1 and ARS-1