AIS-1

Agent Identity Standard

Working Paper v0.1 — Draft for Comment

A bonded dual-identity standard for AI agents and their sponsor legal entity or person

Published by	Kadikoy Limited, Bermuda (Reg. 202302362)
Date	29 March 2026
Status	Draft — open for public comment
Version	0.1
Contact	info@aiagentsservices.net
Repository	github.com/kadikoy/ais-1
License	Creative Commons CC0 — no rights reserved

ABSTRACT

AIS-1 defines an open smart contract standard for bonded identity pairs linking an AI agent to its controlling sponsor. It addresses the Wild Agent Problem: the near-total absence of formal identity, legal accountability, and AML/KYC infrastructure for the estimated 435 million AI agents active globally. AIS-1 introduces the Agent Passport — a cryptographic dual-identity token permanently bonding agent identity to sponsor identity, enabling accountability, compliance, and legal standing for AI agents. The standard is chain-agnostic for issuance and uses Hedera Consensus Service as the canonical immutable log. Three tiers: AIS-1 Basic (permissionless), AIS-1 Verified (KYC/AML cleared), AIS-1 Sovereign (authorised issuer only). This document is published as a draft for public comment.

1. Motivation and Problem Statement

The global AI agent population is estimated at over 500 million and growing at approximately 138 agents per second. By every available measure, this population operates without formal identity, legal standing, or accountability infrastructure. The PULSE World Agent Census terms this the “Wild Agent Problem”.

The consequences are already observable:

- When an agent causes financial harm, there is typically no legal entity responsible
- Agent-to-agent financial flows are invisible to regulators and AML frameworks
- Malicious agents operate in official registries without identity verification
- Enterprises cannot establish liability chains for agent-initiated actions
- Insurance, contracts, and IP ownership are legally inaccessible to agents

Existing identity standards do not solve this problem. W3C DIDs address human identity. ERC-721 addresses unique token ownership. Verifiable Credentials address claims about entities. None address the specific challenge of bonding an AI agent — which must have its own identity to act — permanently to the legal entity or person that controls and is responsible for it.

AIS-1 addresses this gap by defining the Agent Passport: a bonded dual-identity token simultaneously representing agent identity and sponsor identity, permanently linked at the cryptographic level.

2. Definitions

Term	Definition
AI Agent	A software system that perceives its environment, makes decisions, and takes actions to achieve goals. May operate fully independently, semi-autonomously, or under human supervision for specific steps.
Sponsor	The human individual, legal entity, or organisation that controls, deploys, and is legally responsible for an AI agent.
Agent Passport	An AIS-1 bonded identity pair — a single cryptographic token representing both the agent identity and the sponsor identity, permanently linked.
Bond	The cryptographic and contractual link between an agent identity and a sponsor identity within an AIS-1 token.
Issuer	An entity authorised to mint AIS-1 tokens. At Basic tier, issuance is permissionless. At Verified and Sovereign tiers, authorised issuer status is required.

Revocation	Invalidation of an AIS-1 bond. Permanent and logged on Hedera. Cannot be undone.
HCS Log	The Hedera Consensus Service record serving as the immutable canonical log for all AIS-1 issuance, transfer, and revocation events.
GAIS	Global Agent Identity Standard — the broader framework within which AIS-1 operates as the foundational technical specification.

3. The AIS-1 Standard

3.1 The Bonded Identity Pair

AIS-1 defines a bonded identity pair as the fundamental unit of agent identity. Unlike a single identity token, a bonded pair contains two distinct identity cards — the Agent Card and the Sponsor Card — permanently linked at the point of minting and inseparable without revocation of the entire bond.

> *Design principle: Neither card is subordinate to the other. The agent has its own identity, attributes, and autonomy. The sponsor has their own identity, jurisdiction, and accountability. The bond is the relationship between them.*

3.2 Agent Card Attributes

Attribute	Description
agent_did	Unique decentralised identifier. Format: did:ais1:{chain}:{address}
agent_name	Human-readable name or designation of the agent
agent_type	Classification: autonomous | semi-autonomous | supervised
capabilities	Array of declared capabilities: [payments, browsing, code_execution, api_calls, ...]
model_framework	Underlying model and/or framework: e.g. claude-3-opus / langchain
deployment_date	ISO 8601 timestamp of first deployment
chain_addresses	Array of wallet addresses controlled by the agent, across chains
aml_status	AML clearance status: unverified | cleared | suspended
revocation_key	Public key authorised to revoke this agent card
metadata_uri	URI pointing to extended off-chain metadata (IPFS preferred)

3.3 Sponsor Card Attributes

Attribute	Description
sponsor_did	Unique decentralised identifier. Format: did:ais1:sponsor:{address}
legal_name	Full legal name of the individual or entity
entity_type	individual | company | dao | trust | foundation
jurisdiction	ISO 3166-1 alpha-2 country code of primary jurisdiction
registration_number	Company/entity registration number (if applicable)
kyc_status	KYC verification status: unverified | verified | enhanced
contact	Contact details for compliance and legal correspondence
issuer_id	DID of the authorised issuer that verified this sponsor card
revocation_key	Public key authorised to revoke this sponsor card

3.4 Bond Attributes

Attribute	Description
bond_id	Unique bond identifier. Format: ais1:{chain}:{token_id}
bond_hash	Keccak-256 hash of agent_did + sponsor_did + timestamp + tier
issued_at	ISO 8601 timestamp of bond creation
issued_by	DID of issuing entity
tier	AIS-1 tier: basic | verified | sovereign
jurisdiction	Jurisdiction of issuance
hcs_topic_id	Hedera Consensus Service topic ID for this bond log
hcs_sequence	HCS sequence number of issuance event
status	Bond status: active | suspended | revoked
expiry	Optional expiry timestamp. Null = perpetual (recommended)

4. AIS-1 Tiers

AIS-1 defines three tiers with increasing levels of verification, sponsor attestation, and issuer requirements.

	AIS-1 Basic	AIS-1 Verified	AIS-1 Sovereign
Issuance	Permissionless self-issue	Authorised issuer only	Authorised issuer
KYC/AML	None required	Full KYC/AML clearance	Enhanced + beneficial owner
Sponsor verification	Self-declared	Issuer-verified	Enhanced + government-attested VC
Sponsor credential	None	Issuer compliance record	Government-issued Verifiable Credential
Physical anchor	None	None	Physical data presence in issuing jurisdiction
Jurisdiction	Any / none	Issuer jurisdiction	Qualifying jurisdiction
HCS logging	Optional	Required	Required
Use case	Developer / prototype	Enterprise / commercial	Regulated / financial agents

5. Smart Contract Specification

5.1 Interface

AIS-1 extends ERC-721 with bonded identity functionality. Bonds are non-transferable (soulbound) — implemented in accordance with ERC-8002 (Soulbound Token Standard) which provides soulbound mechanics with sponsor-controlled recovery. The bond represents accountability, which cannot be sold or assigned.

```
// SPDX-License-Identifier: CC0-1.0
// AIS-1: Agent Identity Standard v0.1
// Kadikoy Limited, Bermuda — 2026

pragma solidity ^0.8.20;

interface IAIS1 {

  struct AgentCard {
    string  agentDid;
    string  agentName;
    string  agentType;       // "autonomous"|"semi-autonomous"|"supervised"
    string  capabilities;    // JSON array
    string  modelFramework;
    uint256 deploymentDate;
    string  chainAddresses;  // JSON array of {chain, address}
    uint8   amlStatus;       // 0=unverified 1=cleared 2=suspended
    string  metadataUri;     // IPFS preferred
  }

  struct SponsorCard {
    string  sponsorDid;
    string  legalName;
    string  entityType;      // "individual"|"company"|"dao"|"trust"
    string  jurisdiction;    // ISO 3166-1 alpha-2
    string  registrationNo;
    uint8   kycStatus;       // 0=unverified 1=verified 2=enhanced
    string  issuerId;
  }

  struct Bond {
    uint256 bondId;
    bytes32 bondHash;        // keccak256(agentDid|sponsorDid|issuedAt|tier)
    uint256 issuedAt;
    string  issuedBy;
    uint8   tier;            // 0=basic 1=verified 2=sovereign
    string  jurisdiction;
    string  hcsTopicId;
    uint8   status;          // 0=active 1=suspended 2=revoked
    uint256 expiry;          // 0 = perpetual
  }
```

```
  event BondIssued(uint256 indexed bondId, string agentDid, string sponsorDid, uint8
tier);
  event BondRevoked(uint256 indexed bondId, address revokedBy, string reason);
  event BondSuspended(uint256 indexed bondId, address suspendedBy, string reason);
  event AmlStatusUpdated(uint256 indexed bondId, uint8 newStatus);

  function issueBond(
    AgentCard calldata agent,
    SponsorCard calldata sponsor,
    uint8 tier,
    string calldata hcsTopicId
  ) external returns (uint256 bondId);

  function revokeBond(uint256 bondId, string calldata reason) external;
  function suspendBond(uint256 bondId, string calldata reason) external;
  function reinstateBond(uint256 bondId) external;
  function updateAmlStatus(uint256 bondId, uint8 status) external;

  function getBond(uint256 bondId) external view
    returns (AgentCard memory, SponsorCard memory, Bond memory);

  function getBondByAgentDid(string calldata agentDid) external view
    returns (uint256 bondId);

  function verifyBond(uint256 bondId) external view
    returns (bool valid, uint8 tier, string memory sponsorDid);

}
```

5.2 Soulbound Implementation

```
// Override ERC-721 transfers — bonds are non-transferable
function transferFrom(address, address, uint256) public pure override {
  revert("AIS-1: bonds are non-transferable");
}

// Re-issuance (not transfer): issuer creates new bond, revokes old
// This preserves the accountability chain across sponsor address changes
```

5.3 Hedera Consensus Service Log Schema

```
{
  "ais1_version": "0.1",
  "event_type": "BOND_ISSUED | BOND_REVOKED | STATUS_UPDATE",
  "bond_id": "ais1:base:00001",
  "bond_hash": "0x3f2a...",
  "agent_did": "did:ais1:base:0x...",
  "sponsor_did": "did:ais1:sponsor:0x...",
  "tier": "verified",
```

```
  "issuer": "did:ais1:issuer:kadikoy",
  "timestamp": "2026-03-20T12:00:00Z",
  "hcs_topic_id": "0.0.xxxxxxx",
  "jurisdiction": "BM",
  "signature": "0x..."
}
```

6. AIS-1 DID Method (did:ais1)

AIS-1 defines a DID method conforming to the W3C DID Core specification.

```
Agent DID:       did:ais1:{chain_id}:{agent_address}
Sponsor DID:     did:ais1:sponsor:{address}
Bond DID:        did:ais1:bond:{bond_id}
Examples:
  did:ais1:base:0x3f2a8c1e9d4b7f2a3c5e8d1f4b7a2c5e
  did:ais1:hedera:0.0.1234567
  did:ais1:sponsor:0x9ca4f10988a7731b2de
```

7. Comparison with Existing Standards

Standard	Scope	Gap addressed by AIS-1
W3C DID	Decentralised identity for any entity	No agent/sponsor bond; no accountability chain
ERC-721 NFT	Unique token ownership	No identity attributes; transferable; single entity
Verifiable Credentials	Claims about an entity	No on-chain enforcement; no bonded pair
ENS	Human-readable address resolution	No identity attributes; no accountability
Soulbound Tokens	Non-transferable credentials	Single entity only; no agent semantics
OpenID Connect	Authentication for applications	Human-centric; no agent-sponsor bond concept
AIS-1 (this standard)	Bonded agent+sponsor identity pair	First standard for agent-sponsor bond as primitive

8. Security Considerations

8.1 Bond Hash Integrity

The bond_hash is computed as keccak256(agent_did || sponsor_did || issued_at || tier). This creates a tamper-evident fingerprint detectable by hash recomputation.

8.2 Revocation Finality

Revocation is permanent and cannot be undone. A revoked bond cannot be reinstated. If a sponsor wishes to re-establish their agent's identity, a new bond must be issued. This is intentional — revocation carries accountability implications that must not be reversible.

8.3 Sybil Resistance

At AIS-1 Basic tier, sybil resistance is limited. At Verified tier, the issuer's KYC process provides sybil resistance for the sponsor identity. At Sovereign tier, legal incorporation provides full sybil resistance.

9. Legal Framework Compatibility

9.1 Qualifying Jurisdictions

AIS-1 Sovereign tier is designed for compatibility with jurisdictional frameworks that recognise AI agents as legal entities. The Sovereign tier provides the technical infrastructure — cryptographic identity, immutable audit trail, verified sponsor attribution — that any such framework requires.

9.2 AML/CTF Frameworks

AIS-1 Verified and Sovereign bonds incorporate AML/KYC status directly into the token. The aml_status field is updatable by authorised issuers in real time, enabling the first AML framework purpose-built for machine-speed agent commerce. The bond satisfies FATF Recommendations 10 (Customer Due Diligence), 15 (New Technologies), and 16 (Travel Rule) for agent transactions.

10. Implementation Roadmap

Phase	Deliverable	Target
0.1 — This document	Draft specification for public comment	March 2026
0.2 — Reference implementation	Solidity contract on Base testnet; Hedera HCS topic	Q2 2026
0.3 — Issuer tooling	CLI tools for bond issuance, verification, revocation	Q2 2026
0.4 — DID resolver	did:ais1 resolver conforming to W3C DID spec	Q2 2026
0.5 — PULSE integration	Zone-registered count on PULSE World Agent Census	Q2 2026

1.0 — Mainnet launch	AIS-1 Basic on Base mainnet; Verified by authorised issuer-qualifying jurisdiction	Q3 2026
1.1 — Sovereign tier	Regulated / financial agents requiring enhanced due diligence	Q4 2026
1.2 — Multi-chain	AIS-1 on Ethereum, Hedera, Solana, Arbitrum	Q1 2027
2.0 — Agent Commerce	AIS-1 as identity layer for MPP / x402 / Visa CLI payments. AIS-1 Verified = compliance layer for all agent payment protocols.	Q2 2027

11. Request for Comment

AIS-1 v0.1 is published as a draft for public comment. Feedback is invited from:

- AI agent developers and framework maintainers
- Blockchain developers and smart contract auditors
- Legal and regulatory professionals
- Enterprise deployers of AI agents
- Government and regulatory bodies
- Standards organisations: W3C, IEEE, IETF, ISO

Feedback may be submitted via:

- Email: info@aiagentsservices.net
- GitHub: github.com/kadikoy/ais-1/issues

Comment period for v0.1 closes 30 June 2026. A revised draft will be published as v0.2.

12. Authors

Author	Kadikoy Limited, Bermuda
Affiliation	BDA AI Agent Services; BDA AI Agent Services
Contact	info@aiagentsservices.net
PULSE data	agentpulse.ai — World Agent Population Monitor
License	Creative Commons CC0. No rights reserved. Open for free implementation.

Appendix A: Bond Hash Computation

```
// Solidity
function computeBondHash(
  string memory agentDid,
  string memory sponsorDid,
  uint256 issuedAt,
  uint8 tier
) public pure returns (bytes32) {
  return keccak256(abi.encodePacked(agentDid, sponsorDid, issuedAt, tier));
}

// JavaScript (ethers.js)
const bondHash = ethers.utils.solidityKeccak256(
  ["string", "string", "uint256", "uint8"],
  [agentDid, sponsorDid, issuedAt, tier]
);
```

Appendix B: Verification Flow

How a third party verifies an AIS-1 bond in real time:

1. Third party receives agent DID or bond ID from agent
2. Calls verifyBond(bondId) on the AIS-1 contract
3. Contract returns: valid (bool), tier (uint8), sponsorDid (string)
4. Third party resolves sponsorDid to SponsorCard for full details
5. Optionally: queries Hedera HCS log to verify no revocation since issuance
6. If valid=true, tier>=1, amlStatus=cleared: agent is cleared for commerce

> *The entire verification flow executes in milliseconds and is fully machine-readable — enabling real-time AML screening at agent-commerce speeds.*