OpenAIIS

Artificial Intelligence Integrity Sentinel

A Decentralized Identity and Accountability Protocol
for Software and AI Agents

Authored by

Tarun Sethi

Founder, OpenAIIS Protocol
Munich, Germany

Version 2.2 · March 2026

openaiis.org · openaiis.com

ABSTRACT

The world is deploying millions of AI agents at unprecedented speed. These agents make decisions, execute transactions, interact with humans, and operate autonomously across every sector of society. Yet no universal mechanism exists to verify who created them, who is responsible for them, or whether they carry any recognised attestation. OpenAIIS proposes a decentralized blockchain protocol where every piece of software and every AI agent carries a permanent cryptographic identity — filed by its creator, verifiable by anyone, controlled by nobody.

OpenAIIS is not a replacement for existing identity, provenance, or compliance systems. It is the missing universal layer that can carry them: a permanent responsibility ledger that binds any artifact to a scoped accountable cryptographic identity, plus portable proofs that any institution can verify. The chain proves key-level responsibility, timestamps, transfer history, and declared safety posture. The ecosystem provides the attestation layer that determines what VERIFIED means in any given context.

SECTION 1

The Problem

Software has no birth certificate. An AI agent has no passport. And when harm occurs nobody is accountable.

For decades software authorship has relied on centralized institutions — private platforms, corporate registries, government bodies — to establish who created what and when. These institutions are fragile, controllable, and increasingly inadequate for the pace and scale of modern software development.

Today the problem has reached a critical threshold. The emergence of autonomous AI agents — software systems capable of acting, deciding, and transacting independently — has created an accountability vacuum that society is entirely unprepared for. These agents operate across borders, at machine speed, and at a scale no human institution can monitor with existing tools.

Yet there exists no universal way to verify provenance and responsibility. No mechanism to establish which cryptographic identity created this agent, who bears accountability for its actions in a given deployment context, and whether it carries any form of recognised attestation. Legitimacy is enforced by institutions using verification rules; the chain provides the provenance record those institutions rely on.

> *"Deploying AI agents without accountability is handing a loaded weapon with*
> *no safety mechanism and no license to anyone who asks."*

The world needs a birth certificate for software. A passport for AI agents. And a legally meaningful accountability bridge that connects every autonomous agent to a responsible cryptographic identity — and through attestation, to a responsible human or organization who can be held answerable for its actions in the specific context of deployment.

SECTION 2

Why Existing Solutions Fail

The problem is not new. Various mechanisms exist today that attempt to address software ownership and identity. Each fails in a fundamental way.

Copyright Registration

Copyright protects the expression of code but relies on centralized government institutions. It is slow, expensive, inaccessible to most developers, and cannot verify an AI agent's provenance in real time.

Version Control Platforms

Centralized version control platforms provide timestamped history but are owned by private corporations. They can be acquired, shut down, or pressured by governments. A developer's record exists only as long as that corporation chooses to preserve it.

Patents

Patents are prohibitively expensive, geographically limited, take years to process, and are entirely irrelevant to AI agent accountability.

Existing Blockchain Attempts

Existing chains were designed for financial transactions and token economies, not software provenance and agent identity. They lack filing mechanisms, scoped responsibility models, safety posture infrastructure, and adapter interfaces for institutional ecosystems.

The Common Thread

Every existing solution shares one structural flaw: a single controlling party over the record of history. The world needs a solution where no centralized party controls history and where verification is plural and enforceable at the edge — by platforms and institutions.

SECTION 2A

Positioning — A Universal Responsibility Passport

OpenAIIS is **not** a replacement for existing identity, provenance, or compliance systems. It is the missing universal layer that can **carry** them: a permanent responsibility ledger that binds any artifact — software, agents, media, research — to a **scoped accountable cryptographic identity**, plus portable proofs that any institution can verify.

Non-Competition: The Passport Standard

Think of OpenAIIS as a **passport standard**. A passport does not replace a country's internal records; it provides a globally verifiable credential. In the same way, OpenAIIS does not compete with signing frameworks, transparency logs, media provenance standards, academic registries, or enterprise compliance programs. It provides a neutral, decentralized substrate where those signals can be referenced, combined, transferred, revoked, and enforced.

OpenAIIS does not ask ecosystems to abandon their tools. It asks them to speak a shared language across four dimensions:

Dimension	The Question It Answers
Artifact identity	What is it? (canonical hash of the artifact)
Scoped responsibility	Who accepts what, where, and for how long?
Attestation status	Is the responsible party VERIFIED under the platform's policy?
Declared safety posture	What controls were committed, and how recently?

Adapters, Not Replacements

To remain universal, OpenAIIS defines small interfaces — **adapters** — that let ecosystems plug in their existing systems without surrendering control. Adapters translate local evidence into verifiable on-chain signals and a portable **Proof Bundle** that a platform can validate offline.

Adapter	What It Ingests	What OpenAIIS Standardises
Artifact adapter	Container digest, build output hash, canonical archive, provenance manifest hash	A deterministic artifact_id and bundle manifest

Attestation adapter	Transparency-log entries, supply-chain receipts, device/capture credentials, audit and insurance proofs	Bounded attestation claims: hash pointers, expiry, revocation
Policy adapter (edge)	Platform and institution rules: who is allowed, under what scope	VERIFIED thresholds + SAFETY_POSTURE freshness + scope matching checks

Three Ecosystem Examples

1 — Software & Agents (Supply Chain and Deployment)

A CI pipeline produces a release artifact (e.g. container digest) and files a registration plus ASSUME_RESPONSIBILITY under a scope such as 'production-deploy,' with explicit capability limits. The operator publishes a SAFETY_POSTURE (kill switch, incident contact, audit/SBOM hashes). A cloud provider admits only Proof Bundles that show a VERIFIED responsible key, fresh posture, and a scope that matches the provider's context. If the project is acquired, TRANSFER_RESPONSIBILITY moves the passport cleanly — accountability follows the current responsible key, not history.

2 — Media (Provenance for Images, Video, Audio)

A creator or generation pipeline produces a media artifact with a provenance credential (e.g. a signed manifest). An artifact adapter binds the manifest hash to an artifact_id. Platforms can require VERIFIED distribution keys for sensitive categories and treat missing or revoked passports as quarantine signals. OpenAIIS does not judge truth; it makes provenance and distribution responsibility verifiable: who published, under what scope, and what controls were declared.

3 — Academia (Papers, Datasets, Models)

A researcher registers a bundle containing the paper PDF hash, code snapshot hash, and dataset and model digests. A university portal sets a submission scope (archiving and access only; claims remain the author's responsibility) and requires VERIFIED status plus a current posture filing for official publication. A commit-reveal mechanism can establish priority without early disclosure, while the final Proof Bundle provides an auditable chain-of-custody for reproducibility and attribution.

Borders set policy; the passport stays universal.
OpenAIIS provides the global, tamper-proof responsibility record and portable proofs. Each ecosystem decides what it will accept — which attestations count, what posture is required, and what scopes are permitted. The protocol is the passport. The ecosystem is the border.

SECTION 3

The Solution — OpenAIIS

OpenAIIS proposes a decentralized blockchain protocol purpose built for one mission: to provide every piece of software and every AI agent with a permanent, verifiable, tamper-proof cryptographic identity — controlled by nobody and accessible to everyone.

The Core Principle

A developer or organization creates software or deploys an AI agent. They file a cryptographic fingerprint of that work on the OpenAIIS blockchain. That fingerprint is permanently timestamped, signed with their private cryptographic key, and recorded across a decentralized network of nodes worldwide. The filing is immutable. The timestamp is undeniable. The key-level accountability is permanent. No central authority approves or rejects filings. The protocol itself is the provenance record.

What Gets Filed

Not the code itself. Never the code itself. Only its cryptographic fingerprint — a SHA3-256 hash of a canonical representation of the artifact (a build output, container digest, canonical archive, or provenance manifest hash). Any modification however small produces a completely different hash. The original code remains entirely private. A structured behavioral fingerprint schema accompanies the hash. In Phase 2, the OpenAIIS Local Tool generates this fingerprint automatically and binds it cryptographically.

What the Filing Proves

A successful filing establishes three irrefutable facts: this specific artifact existed; it existed at this specific moment in time; it was filed by the holder of this specific cryptographic identity. These facts together constitute the provenance record that software has never had before.

The Identity Pass

Every registered agent receives an identity pass — a verifiable registration record on the chain. Any system, platform, organization, or individual can instantly query the OpenAIIS network: Is this agent registered? Which key is responsible? What is its attestation status? What safety posture has been declared? What deployment scope applies? The answer comes from the network. Not from a company. From mathematics. Platforms may configure their admission policies to require VERIFIED status.

Responsibility is Scoped

Responsibility on OpenAIIS is never a blanket assignment. Every deployment occurs within a defined **ResponsibilityScope** — a structured declaration specifying the deployment context, the accepting party, and the precise boundaries of what each party accepts responsibility for.

This prevents infrastructure operators from being unfairly assigned liability for content or behaviour they did not originate.

Field	Contents	Purpose
scope_id	Unique identifier for this deployment context	Links the filing to a specific operational context
platform_key	Public key of the accepting platform or institution	Identifies who takes on scoped responsibility
accepts_responsibility_for	Explicit list of what the platform accepts	E.g. distribution, storage, processing, hosting
excludes_responsibility_for	Explicit list of what remains with the filer	E.g. content, claims, behavioral outputs

Scope Examples

YouTube Upload Pipeline Scope

YouTube accepts responsibility for processing, storage, and distribution. It explicitly excludes responsibility for the content, behavioral outputs, or claims of the agent itself. The original filer retains full responsibility for what the agent does.

University Submission Scope

The university accepts responsibility for archiving and access. It excludes responsibility for research claims, experimental methodology, or behavioral characteristics. The submitting researcher retains responsibility for the artifact's content.

Law Enforcement Internal Processing Scope

The agency accepts responsibility for secure handling and chain of custody within its jurisdictional authority. It excludes responsibility for the artifact's provenance prior to receipt — which the OpenAIIS filing provides.

The Three Pillars of a Complete Filing

Pillar	What It Proves	How It Works
Cryptographic Hash	Immutable proof of what it is	SHA3-256 fingerprint. Any modification produces a different hash.
Identity Signature	Immutable proof of which key filed it	Ed25519 private key signature. UNVERIFIED (pseudonymous) or VERIFIED (attested).
Behavioral Fingerprint	Structured capability surface claims	Phase 1: developer-declared. Phase 2: ZK proof of analyzer-consistency.

SECTION 4

The Behavioral Verification Layer

The Behavioral Fingerprint is the foundation of OpenAIIS's most significant architectural innovation — the Behavioral Verification Layer. This layer transforms OpenAIIS from a simple registry into an intelligent accountability filter, with one uncompromising principle: the code never leaves the developer's machine. Ever.

The OpenAIIS Local Tool (Phase 2)

The Behavioral Fingerprint is derived mathematically and automatically from the actual code by the OpenAIIS Local Tool — a fully open source utility running entirely on the developer's machine. It performs three operations locally before any information touches the network: static behavioral analysis producing the fingerprint schema; cryptographic hashing of the codebase; and generation of a Zero Knowledge Proof. The code itself never moves.

What the Zero Knowledge Proof Actually Proves

> **Precise claim:** The ZK proof proves consistency — this specific Behavioral Fingerprint is the output of analyzer version X applied to the artifact with hash H. This does not prove semantic safety. It does not prove intent. It proves that the fingerprint was genuinely derived from the code matching the submitted hash, by the declared version of the analysis tool. Analyzer-consistency is the guarantee.

In Phase 1, filings are accepted on the basis of: valid SHA3-256 hash, valid Ed25519 signature, and bounded behavioral fingerprint schema. The ZK proof field carries a placeholder — STARKs are Phase 2. VERIFIED tier filings in Phase 2 require a valid ZK proof as a condition of VERIFIED status.

Dynamic Proof of Work Escalation (Proposed Extension)

The Behavioral Fingerprint creates the possibility of similarity-aware dynamic proof of work escalation. A legitimate developer registering genuinely novel software faces minimal resistance. A bad actor registering thousands of behaviorally identical agents faces exponentially increasing computational cost. This mechanism is a proposed extension — until similarity scoring is fully deterministic, computationally cheap to verify, and formally specified, it operates as a mempool and policy layer, not a consensus rule.

SECTION 5

Technical Architecture

5.1 The Decentralized Node Network

The OpenAIIS network is composed of nodes — computers running the open source node software. Each node holds a complete copy of the entire chain. There is no master server, central database, or single point of failure. In Phase 1, a filing must satisfy: valid artifact hash, authentic identity signature, and bounded fingerprint schema. Phase 2 adds ZK proof verification as a condition of VERIFIED status.

5.2 The Chain Structure

Each block contains validated filings, a timestamp, the previous block's hash, and a proof of work solution. The Genesis Block is the permanent anchor. To alter any historical record an attacker must recalculate every subsequent block across the entire network simultaneously — computationally impossible at scale. Serialization uses canonical JSON with sorted keys; protobuf will replace this in a future release.

5.3 The Proof of Work Mechanism

To add a new block the submitting node must find a value that, combined with the block's contents, produces a hash meeting a specific difficulty target. Finding this requires real computational effort. Verifying it requires almost none. Base difficulty targets 10-minute block times, adjusting every 2016 blocks.

5.4 Identity on OpenAIIS

Every participant is represented by an Ed25519 key pair. The base ledger is key-based: it records which cryptographic identity filed which artifact, when, and under what declared scope and posture. VERIFIED status is achieved through protocol-recognised attestation mechanisms. The Proof Bundle packages all verifiable signals for offline validation.

5.5 Ownership Transfer Protocol

The current owner signs a transfer transaction specifying the recipient's public key, the registration being transferred, and optionally a new ResponsibilityScope for the new deployment context. The complete chain of custody is always visible and verifiable.

5.6 The OpenAIIS Local Tool (Phase 2)

Fully open source software running on the developer's machine. Supports all major programming languages. The Local Tool carries only mathematical proofs across the bridge from private machine to public chain. Never code. Never private information.

SECTION 6

The Responsibility Layer

Every technical guarantee OpenAIIS provides serves one ultimate purpose: to make responsibility traceable, permanent, scoped, and legally meaningful. This is the Responsibility Layer — the bridge between the cryptographic world of the chain and the human world of accountability.

Two Tiers of Registration

Tier	What It Means	Who Requires It
UNVERIFIED	Key exists. Not attested. Filing is valid and permanently timestamped, traceable to a key but not to any attested real-world entity.	Open registrations, personal projects, early-stage development. Serious commercial and regulatory environments will not accept this tier.
VERIFIED	Meets attestation threshold policy. Real-world identity is cryptographically bound and sealed via threshold signatures. VERIFIED status is publicly visible; the identity behind it is sealed until a lawful process engages the keyholders.	Required by serious ecosystems: enterprise platforms, regulated industries, government procurement, academic publication, and any context where accountability to a real-world entity is a legal or institutional requirement.

Sealed Identity and Threshold Signatures

The sealed identity in VERIFIED registrations is encrypted using a threshold signature scheme requiring a defined minimum number of independent keyholders to act simultaneously. These keyholders are independent entities distributed globally. No single government, corporation, or individual controls enough keys to unilaterally reveal a sealed identity. OpenAIIS itself holds no keys.

Safety Posture & Safe Harbor

VERIFIED status establishes who is responsible. Safety Posture establishes how responsibly they operate. OpenAIIS supports a dedicated **SAFETY_POSTURE** filing type — a structured declaration that the responsible party can file and update to document their operational safety practices.

Field	Contents	Significance
controls	Enumerated safety measures in the agent	Documented operational controls at time of filing

kill_switch	How the agent can be deactivated	Critical for incident response
incident_contacts	Who to contact if the agent causes a problem	Enables fast escalation without revealing sealed identity
audit_pointers	References to audit trails or compliance records	Supports regulatory inspection
posture_timestamp	When this posture filing was made	Freshness is the key signal — stale posture is high risk

The Safe Harbor Framework

Tier	Conditions	Platform Treatment
SAFE HARBOR	VERIFIED + current SAFETY_POSTURE (within platform-defined freshness window) + prompt mitigation filings for any known issues	Reduced liability exposure. Streamlined admission. Regulatory audit trail available on demand.
CONDITIONAL	VERIFIED but posture missing, stale, or incomplete. Or: UNVERIFIED with current posture.	High friction admission. Additional review required. Possible sandboxing or enhanced monitoring.
QUARANTINE / REJECT	UNVERIFIED with no posture. Or: revoked registration. Or: posture inconsistent with observed behaviour.	Rejected or strict quarantine. No access to production resources until remediated.

Safe harbor is not a promise the protocol makes — it is a framework the protocol enables. Each platform defines its own freshness window, required controls, and escalation procedures. OpenAIIS supplies the verifiable signals. The ecosystem supplies the policy.

SECTION 7

The Node Incentive Model

Why Nodes Run

Nodes run on OpenAIIS for the same reason thousands of nodes run on other decentralized networks today with no direct financial reward — because the people and organizations running them believe this infrastructure must exist. A developer who has had their work stolen runs a node because they know what it means to have no proof of authorship. A university runs a node because understanding the provenance of AI systems is fundamental to responsible research. A civil liberties organization runs a node because decentralized accountability infrastructure is the antidote to centralized surveillance and control.

No Fees. No Tokens. No Foundation.

There are no filing fees on OpenAIIS. There are no tokens. There is no foundation with a treasury and a board and a mandate that can drift. The proof of work costs the filer computational energy — a commitment to the network, not a payment to anyone. Querying OpenAIIS costs nothing. Running a node costs nothing except hardware and electricity contributed freely.

Governance by Rough Consensus

Protocol decisions on OpenAIIS are made through rough consensus and running code — the same model that governs the internet's foundational protocols. No single entity can unilaterally change the protocol. Changes occur through client upgrades and broad adoption. On-chain signaling may exist as an information layer but does not bind nodes.

SECTION 8

The Roadmap

Phase	Name	Deliverables
1 · Now	Foundation	Chain + filings + adapters + scoped transfers + SAFETY_POSTURE schema + query/explorer
2 · Next	Local Tool	Fingerprint auto-generation, ZK proof of analyzer-consistency, Proof Bundle packaging
3	Node Network	Node packaging, documentation, global community onboarding
4	Responsibility Layer	Threshold unsealing, VERIFIED attestation, globally distributed keyholders
5	Global Standard	Ongoing forever — the permanent operational phase

SECTION 9

Adoption and Enforcement

A provenance infrastructure is only as valuable as the ecosystem that requires it. OpenAIIS is designed so that adoption is driven by independent incentives at every level of the AI agent economy — without requiring any central mandate to begin.

Concrete Platform Enforcement Rules

Rule	What It Means	Why It Matters
1. VERIFIED-only admission	No UNVERIFIED agents accepted for production. All admitted agents must carry VERIFIED status on their identity pass.	Every agent in production traces to an attested real-world entity. Platform liability is documented and bounded.
2. Scope must match context	The declared ResponsibilityScope of an incoming agent must be appropriate for the platform's operational context.	Prevents scope mismatch liability. An agent deployed outside its declared scope creates an immediately visible chain violation.
3. Posture required and fresh	A current SAFETY_POSTURE filing must exist for every admitted agent. Platforms define their own freshness window (e.g. 90 days).	Ensures safety commitments are maintained, not just declared at registration. A posture from 18 months ago is meaningless.
4. Revocation honored immediately	If a registration is revoked or SAFETY_POSTURE withdrawn, the platform pulls the agent from production immediately. No grace period.	Revocation without immediate enforcement is not revocation. The chain provides the signal; the platform's policy provides the enforcement.

University and Research Institutions

Research institutions can require that all submitted work carries an OpenAIIS provenance record as a condition of publication, within the university's declared ResponsibilityScope for the submission pipeline. The provenance record establishes that a specific entity filed a specific artifact at a specific moment and takes responsibility for what they released.

Law Enforcement and Regulatory Bodies

When VERIFIED registration exists and a registered agent causes documented harm, institutions can invoke the threshold unsealing process under their jurisdiction. The keyholders are independent and globally distributed — no single government can compel a unilateral reveal. Regulatory bodies can build audit infrastructure on the OpenAIIS query interface without requiring cooperation from any central party — because there is no central party to ask.

The Compounding Network Effect

Every platform that adopts an OpenAIIS admission policy creates an immediate incentive for every developer deploying to that platform to register their work. Every developer who registers discovers the permanent proof-of-authorship value. Every node operator who joins strengthens the infrastructure. The adoption loop is self-reinforcing.

SECTION 10

The Vision

This protocol was not born in a laboratory. It was not conceived in a boardroom. It was born from worry.

Worry about what is happening in the world right now. Misinformation spreading faster than truth. Social, cyber, and physical attacks executed by autonomous systems with no traceable hand behind them. AI technology deployed as a weapon by those who face no consequence and leave no fingerprint. A world that built the most powerful tools in human history and then handed them out with no accountability and no mechanism to trace harm back to its origin.

The Urgency

Every day without provenance infrastructure is a day the problem deepens. Without a permanent, tamper-proof, scoped record of who built what, who is responsible in which context, and what safety posture was declared — the innocent are blamed and the guilty disappear. The window to establish this infrastructure is not infinite.

The Horizon

A day is coming when AI systems will generate code themselves, deploy agents, and publish their work as autonomous actors. When that day comes OpenAIIS will be ready. The protocol makes no distinction between a human developer filing a registration and an AI system doing the same. The chain asks only that you take responsibility — scoped to your deployment context, with a declared safety posture — for what you release.

For Our Children

Everything written in this whitepaper serves one purpose that has nothing to do with technology. It serves the children who will inherit the world we are building right now. Who deserve to live without the fear that invisible autonomous systems can cause harm, destroy reputations, manipulate reality, and face no consequence.

The Invitation

OpenAIIS belongs to nobody. It is built for everyone. Run a node. File your work. Contribute to the protocol. Challenge it, improve it, make it stronger. This is not a product you are being asked to buy. It is infrastructure you are being asked to help build. For your children. For their children. For a future where both humans and AI coexist in the shared accountability that makes harmony possible.

OpenAIIS — Artificial Intelligence Integrity Sentinel
Authored by Tarun Sethi · Founder, OpenAIIS Protocol
Munich, Germany · Version 2.2 · March 2026
openaiis.org · openaiis.com

This protocol is open source and belongs to the world.