The MolTrust Protocol

A Verification Standard for Autonomous Software Agents

Version 0.4 — Draft for Review
MolTrust / CryptoKRI GmbH, Zurich
March 2026

Abstract. Autonomous software agents now act on behalf of humans and organizations: they execute payments, delegate tasks, negotiate contracts, and interact with other agents across platforms and jurisdictions. The number of active agents is growing faster than the infrastructure to identify and verify them. This paper describes a minimal, open verification standard for software agents — a common basis for establishing whether an agent is what it claims to be, is authorized to act as it claims, and has behaved consistently with its declared parameters in the past. The standard is built on existing open specifications (W3C DID, W3C Verifiable Credentials), requires no permission to implement, and is designed to function independently of any single platform, jurisdiction, or operator. Technical specifications, data models, and conformance requirements are defined in the companion document: *The MolTrust Protocol: Technical Specification (v0.2.1).*

SHA-256	`70597a2c5e510bcdeaaeffe037a7c79b2c781865893a0ca738c95aa18db3e20f`
On-chain anchor	`Base L2 (mainnet), Block 43691641`
Transaction	`0x886cbc086fc4200b96771a67be0b62cd0971442d8f7e099ae27b056516acf0b2`
Timestamp	`2026-03-22T09:43:49 UTC`

— 1 —

Context

The deployment of autonomous software agents is accelerating across every sector. Agents book travel, execute trades, manage infrastructure, process procurement, and interact with customers — often without direct human involvement in individual transactions.

This creates a structural gap. The identity and verification frameworks currently in use were designed for human users and static service accounts. They assume a predictable actor, a defined session, and a traceable principal. Agents violate these assumptions routinely: they operate ephemerally, spawn sub-agents, act across organizational boundaries, and make decisions at speeds that preclude human review.

The result is that two agents interacting today have no standardized way to establish the basic facts of their encounter: who the other party is, under whose authority it operates, and whether its claimed history is accurate. Each platform that deploys agents solves this problem independently, producing verification that is valid within one ecosystem and meaningless outside it.

A common standard addresses this gap — not by replacing platform-specific implementations, but by providing a shared layer of verifiable facts that any implementation can produce and any counterpart can check.

— 2 —

The Verification Gap

Three questions arise in any agent-to-agent or human-to-agent interaction that current infrastructure cannot reliably answer:

Identity

Is this the same agent I interacted with previously? Agent identifiers today are typically platform-assigned and platform-scoped. They do not survive migration, cloning, or redeployment. There is no cryptographic guarantee that the agent presenting a given identifier is the same entity that held it before.

Authorization

Is this agent permitted to do what it claims? Agents act on behalf of principals — humans, organizations, or other agents. The chain of delegation from principal to agent is rarely expressible in a form that a counterpart can verify independently. Claims of authorization are typically self-asserted.

Behavioral history

Has this agent acted consistently with its declared parameters in the past? Reputation systems exist within platforms. They do not transfer. An agent with a long, clean history on one system is indistinguishable from a newly registered agent on any other.

These three gaps compound each other. Without portable identity, behavioral history cannot be attributed reliably. Without verifiable authorization, identity alone is insufficient. The absence of any one element

 © 2026 CryptoKRI GmbH, Zurich

undermines the value of the others.

— 3 —

Design Requirements

A verification standard for agents must satisfy requirements that differ from those of human identity systems.

Decentralized by necessity

No single authority can serve as the trust anchor for a global agent economy. Agents operate across jurisdictions where different legal systems, regulatory frameworks, and institutional structures apply. A standard that requires trust in a single issuing authority inherits the limitations of that authority — geographic, political, and operational. The standard must be verifiable without reference to any central registry.

Lightweight by design

Agents interact at machine speed. A verification procedure that introduces meaningful latency becomes a bottleneck. The standard must enable verification in milliseconds for routine interactions, reserving slower operations for events that genuinely require permanent record — registration, credential issuance, and violation recording.

Separable from its operator

The organization that operates verification infrastructure is not the same as the standard itself. A standard that can only be verified through one provider is a proprietary system with open branding. The standard must be implementable by any party and verifiable by any counterpart, independently of the reference implementation.

Minimally prescriptive

The standard defines what must be verifiable, not how agents should behave, what they should be permitted to do, or how disputes should be resolved. Policy decisions belong to the parties in each transaction. The standard provides the factual substrate on which those decisions can be made.

Jurisdiction-neutral

The standard must be operable within any legal system that permits software agents to act on behalf of participants. It makes no assumptions about the regulatory environment of the deploying organization, the location of the infrastructure, or the nationality of the principals involved.

— 4 —

The Protocol

The MolTrust Protocol defines four primitives that together satisfy the requirements above.

© 2026 CryptoKRI GmbH, Zurich

4.1 Identity

Every agent in the protocol is assigned a Decentralized Identifier (DID) — a globally unique, cryptographically verifiable identifier that the agent controls. DIDs are specified by the W3C and require no central registry to create or verify. Ownership is proven by possession of the corresponding private key.

→ *Is this the same entity I interacted with before?*

4.2 Authorization

The relationship between an agent and the principal on whose behalf it acts is expressed as a W3C Verifiable Credential — a digitally signed, tamper-evident attestation issued by the granting party. Credentials describe what an agent is permitted to do, under what conditions, and with what expiry. Credential chains support delegation, with each step cryptographically linked to the previous. Any verifier can traverse the chain independently without contacting the original issuer.

→ *Is this agent permitted to act as it claims?*

4.3 Behavioral Record

Interactions between agents produce verifiable proofs — cryptographic records that a specific interaction occurred at a specific time between identified parties, with a recorded outcome. These proofs are signed by both parties and cannot be altered after the fact. Proofs accumulate into a trust score that reflects observed behavior across time and context, weighted by endorsers' own scores and distributed across multiple independent domains to resist manipulation.

→ *Has this agent acted consistently with its declared parameters?*

4.4 Portability

All three primitives above — identity, authorization, and behavioral record — are expressed in open, platform-independent formats. A credential issued by one organization is verifiable by any counterpart. A behavioral record compiled from interactions on one platform travels with the agent to any other. Portability here means interoperability of evidence formats, not identity of scoring outcomes: different implementations may produce different trust scores from the same evidence depending on their scoring model.

→ *Can this agent's claims be verified regardless of where it was registered?*

— 5 —

Speed and Permanence

The protocol separates two operations that have different requirements: verification and accountability.

 © 2026 CryptoKRI GmbH, Zurich

Verification happens at the speed of a signature — milliseconds, off-chain. Accountability is anchored at the speed of a blockchain — infrequent, permanent, and tamper-evident.

Verification — checking identity, validating credentials, querying trust scores — must be fast. These operations are performed off-chain using standard cryptographic primitives. A verification request completes in under 100 milliseconds under normal conditions. No consensus mechanism is involved. No network fee is incurred.

Accountability — anchoring agent registration, recording credential issuance, and permanently recording proven violations — benefits from tamper-evident, distributed storage. These operations are performed on-chain and occur infrequently: once at registration, periodically for score snapshots, and when a violation is recorded. The on-chain layer is used specifically for operations where permanence and public verifiability matter.

Stake

Agents operating under the protocol may optionally deposit a stake — a defined amount of value held in a smart contract — at registration. The stake creates an economic commitment to declared behavior: it is returned when the agent deregisters cleanly, and forfeited if a proven violation is recorded. Stake is optional for participation. It is one signal among others in the trust score computation.

— 6 —

Resistance to Manipulation

Any reputation system can be manipulated. The protocol's design addresses the most common vectors.

Sybil attacks

Manufacturing reputation through multiple coordinated fake identities is mitigated by cross-domain endorsement requirements. A trust score that draws on endorsements from verified agents across multiple independent verticals is significantly more costly to fabricate than one based on a single interaction context.

Score inflation

Accumulating positive history in low-stakes interactions before exploiting trust in high-stakes ones is addressed through behavioral consistency monitoring. Significant deviation from an established pattern surfaces as an anomaly signal alongside the aggregate score. The optional stake mechanism adds an economic dimension: the value of the stake must be weighed against the expected return from exploitation.

Compromised agents

Agents whose private keys have been obtained by unauthorized parties can be detected through behavioral discontinuities and mitigated through credential revocation. Revocation propagates rapidly across the network. Verifiers with strict requirements should perform real-time verification rather than relying on cached results.

© 2026 CryptoKRI GmbH, Zurich

The protocol does not guarantee that manipulation is impossible. Its objective is to ensure that the cost of manipulation increases proportionally with the scale of the attempt — making cooperative behavior more rational than defection in the overwhelming majority of cases.

— 7 —

Scope and Limitations

The protocol defines what is verifiable. It does not define what is permissible, valuable, or correct.

The protocol does not evaluate agent output. Whether an agent's response, recommendation, or transaction outcome is accurate or appropriate is a matter for the parties involved. The protocol attests to identity and behavioral history — not to the quality or correctness of any specific output.

The protocol does not establish legal accountability. Verified identity makes an agent's actions traceable. Legal consequence for those actions remains the domain of applicable law. The protocol provides the factual record that legal processes may rely on; it does not substitute for them.

The protocol does not require trust in any single operator. MolTrust operates a reference implementation and public API. The underlying standards are open. Any organization may implement a compatible verification system. Any verifier may check any credential without involving MolTrust.

The protocol does not prescribe agent behavior. What an agent is permitted to do, how disputes are resolved, and what constitutes a violation are policy questions determined by the deploying organization and the parties to each transaction.

The protocol does not make privacy guarantees beyond its defined scope. Interaction proofs record structural metadata and outcome hashes — not raw transaction content. The protocol is designed for compatibility with data minimization principles. Organizations deploying the protocol in contexts involving personal data are responsible for compliance with applicable privacy law.

These boundaries are structural, not incidental. A verification standard that makes content judgments, substitutes for legal process, or requires trust in its operator becomes something other than a standard.

— 8 —

Architecture

The protocol is organized in three layers, described in detail in the Technical Specification.

Protocol Standard (Layer A)

The normative core: data formats, signing rules, verification flows, and lifecycle semantics. Any independent implementation conforming to Layer A can interoperate with any other conformant implementation at the evidence level.

© 2026 CryptoKRI GmbH, Zurich

Reference Registry (Layer B)

The MolTrust-operated service layer: identity resolution, credential revocation, trust score queries, and on-chain anchoring. Other operators may run conformant registries using their own infrastructure.

Reference Reputation Model (Layer C)

An informative scoring model used by the MolTrust reference registry. Other implementations may use different scoring models, provided they consume Layer A evidence formats.

This layering means the protocol is genuinely open at its core, while the reference service provides a functional implementation that any party can use, verify against, or replace.

— 9 —

Relationship to Existing Standards

The MolTrust Protocol does not introduce new cryptographic primitives or identity concepts. It applies existing, well-specified open standards to the specific requirements of autonomous agent verification.

W3C DID Core v1.0	Identity layer. W3C Recommendation, widely implemented.
W3C VC Data Model 2.0	Authorization and attestation. Active in national digital identity (incl. EU Digital Identity Wallet under eIDAS 2), academic credentialing, and emerging enterprise IAM.
ERC-8004	Optional on-chain agent registration for blockchain-anchored identity.

The protocol's contribution is not new standards. It is the application of existing standards to a specific gap — agent-to-agent trust — and a reference implementation that demonstrates their sufficiency for this purpose.

— 10 —

Relationship to Adjacent Work

Active research and standardization efforts address related problems. The MolTrust Protocol does not replace these efforts — it occupies a specific, defined position relative to them.

Trusted Agentic Mesh (TAM)

Proposes a full-stack architecture combining DID/VC identity with a Byzantine Fault Tolerant trust plane and "Proof-of-Behavior" consensus. TAM and MolTrust share a diagnosis — the identity-behavior gap — but differ in approach. TAM is a complete infrastructure stack requiring consensus participation. MolTrust is a minimally prescriptive evidence standard that defines what must be recorded, not how consensus is reached.

© 2026 CryptoKRI GmbH, Zurich

AgentHub

Addresses agent discoverability and provenance through signed manifests and namespace control. Complementary to MolTrust: AgentHub addresses code provenance at discovery time; MolTrust addresses behavioral history and authorization at interaction time.

ERC-8004

Defines on-chain registries for agent identity, reputation, and validation with crypto-economic proofs. MolTrust's relationship is explicitly complementary: MolTrust defines off-chain evidence formats and scoring semantics; ERC-8004 provides one anchoring mechanism for those facts on-chain.

W3C AI Agent Protocol CG and DIF Trusted AI Agents WG

Active standardization venues for agent identity and delegation patterns, currently in exploratory phases. MolTrust is more concrete on data models and conformance requirements while these groups are broader in scope. MolTrust is intended to be compatible with and potentially contributory to whatever standards these groups produce.

The specific combination that MolTrust defines — DID-based agent identity, VC-based authorization chains, protocol-level interaction proofs, and a portable behavioral reputation model as a layered implementable standard — does not appear to be covered by any existing published specification as of the date of this document.

 © 2026 CryptoKRI GmbH, Zurich

— 11 —

Universality

The verification gap described in this paper is not specific to any sector, jurisdiction, or technology stack. Wherever software agents act on behalf of principals, the questions of identity, authorization, and behavioral history arise.

The standard operates within market economies and within planned ones. It applies to large enterprises and to individual developers. It functions under regulatory regimes that are permissive toward autonomous agents and under those that require human oversight at each decision point — because verification infrastructure is compatible with oversight, not in tension with it.

No single organization, government, or standards body controls the protocol. Its legitimacy derives from the openness of its specifications, the verifiability of its outputs, and adoption by the parties who use it. This is the same basis on which every foundational internet protocol has achieved legitimacy.

The agent economy does not have geographical boundaries. Its verification infrastructure should not either.

— 12 —

Summary

The MolTrust Protocol defines a minimal, open standard for agent verification based on four primitives — identity, authorization, behavioral record, and portability — implemented using existing W3C open standards.

Identity	Cryptographically verifiable, self-sovereign, platform-independent (W3C DID)
Authorization	Expressed as Verifiable Credentials, signed by the granting party (W3C VC)
Behavioral Record	Interaction proofs aggregated into portable, cross-platform trust scores
Portability	Verifiable by anyone, on any platform, without central registry

The protocol does not govern agent behavior, evaluate agent output, or substitute for legal accountability. It provides the verifiable factual substrate on which those functions can be built.

A reference implementation is available at **api.moltrust.ch**. Protocol specification, credential schemas, and integration packages are published as open source at **github.com/MoltyCel**. The companion Technical Specification (v0.2.1) provides complete data models, verification flows, and conformance requirements.

MolTrust
CryptoKRI GmbH, Zurich

api.moltrust.ch
moltrust.ch

info@moltrust.ch
@moltrust

CC BY 4.0
© 2026