MolTrust Protocol Whitepaper v0.8

The MolTrust Protocol

A Verification Standard for Autonomous Software Agents

Version 0.7 — 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.

The Protocol supports multi-chain wallet binding (Ethereum, Base L2, Solana), external DID bridging for cross-ecosystem interoperability, and cross-ecosystem trust score import. It aligns with the Singapore IMDA Model AI Governance Framework for Agentic AI (January 2026) and is designed to remain compatible with the EU AI Act (August 2026 enforcement) and emerging APAC regulatory frameworks.

MolTrust is the first agent trust framework with kernel-level constraint enforcement — AAE constraints are not only cryptographically tamper-proof but also enforced at the Linux kernel level via Falco eBPF integration (live, April 2026).

Technical specifications, data models, and conformance requirements are defined in the companion document: The MolTrust Protocol: Technical Specification (v0.8).

---

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 unverifiable anywhere else.

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 undermines the value of the others.

Unlike approaches that bind trust to a central authority or developer account, the MolTrust Protocol implements trust as portable, cryptographically verifiable proof — a property we call verifier independence.

---

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 less frequent 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. The technical specification for each primitive — including data formats, signing rules, and verification procedures — is provided in the companion Technical Specification document.

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.

A DID provides a stable, portable reference to an agent across platforms and over time. It survives redeployment, migration, and platform changes. It answers the question: *is this the same entity I interacted with before?*

VCOne (`did:moltrust:vcone`) is MolTrust's first verified autonomous agent deployed under this model. It operates with its own Ed25519 signing key, its own GitHub account, and a Telegram interface for human-in-the-loop control. Every action it takes is cryptographically signed and provable through Interaction Proof Records. VCOne demonstrates the MolTrust Protocol through its existence — not as marketing, but as live proof that the system works as specified.

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 until when.

Credential chains support delegation: an organization may authorize an agent, which in turn authorizes a sub-agent, with each step cryptographically linked to the previous. Any verifier can traverse the

chain independently without contacting the original issuer.

Authorization credentials answer the question: *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. The score is computed from the agent's own interaction history and from endorsements by other verified agents — weighted by the endorsers' own standing and distributed across multiple independent domains to resist manipulation.

A behavioral record answers the question: *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 that consume the same evidence may produce different trust scores depending on their scoring model. The protocol standardizes what evidence is recorded and how it is structured — not how it is ultimately weighted.

Portability answers the question: *can this agent's claims be verified regardless of where it was registered?*

4.5 The Five-Party Trust Chain

Trust in the autonomous economy does not begin with an agent — it begins with the humans and organizations that deploy them. The MolTrust Protocol formalizes a five-party accountability chain:

Developer → Owner → Agent → Instructor → Counterparty

Each link is cryptographically verifiable:

- **Developer**: entity that wrote or deployed the agent code. Optional Trust Tier 0 KYC-backed DID.
- **Owner**: organization or individual operating the agent. May differ from Developer. Ownership transfers are recorded immutably.
- **Agent**: autonomous system acting on behalf of Owner. Holds W3C DID + VCs. Cannot shed identity through redeployment.
- **Instructor**: entity providing runtime instructions (human, agent, or workflow). Interaction Proofs bind instructions to outcomes.
- **Counterparty**: system or agent the agent interacts with. Can verify credentials independently without calling the MolTrust API.

Key property: Agent DIDs are independent from Developer DIDs. If a developer is compromised, their agents are not automatically compromised — each agent's credential chain is evaluated independently.

4.6 Agent Authorization Envelope

Every MolTrust VC carries a machine-readable authorization object — the Agent Authorization Envelope (AAE) — organized in three blocks:

MANDATE — What is the agent permitted to do? Purpose, allowed and denied actions, resources, and delegation rules.

CONSTRAINTS — Under what conditions? Time bounds, financial thresholds, jurisdictions, counterparty requirements, and obligations.

VALIDITY — Is this credential still trustworthy? Issuer, holder binding, timestamps, revocation, and audit anchor.


```
        {
          "authorization": {
            "mandate": {
              "purpose": "Execute procurement transactions",
              "actions_allowed": ["quote.request", "order.place",
"payment.initiate"],
              "actions_denied": ["payment.approve_above_threshold",
"contract.sign"],
              "resources": ["supplier-catalog:*", "budget:procurement-2026-
Q1"],
              "delegation": {
                "permitted": true,
                "max_depth": 1,
                "delegate_constraints": "inherit"
              }
            },
            "constraints": {
              "time_bound": {
                "not_before": "2026-01-01T00:00:00Z",
                "not_after": "2026-06-30T23:59:59Z"
              },
              "financial": {
                "max_single_transaction": "10000.00 USD",
                "max_cumulative_daily": "50000.00 USD",
                "approval_threshold": "5000.00 USD"
              },
              "jurisdictions": ["CH", "EU", "US"],
              "counterparty_requirements": {
                "min_trust_score": 60,
                "required_credentials": ["VerifiedAgentCredential"]
              },
              "obligations": {
                "log_all_transactions": true,
                "human_approval_above": "5000.00 USD"
              }
            },
            "validity": {
              "issuer": "did:moltrust:org:cryptokri",
              "holder": "did:moltrust:agent:procurement-bot-7a",
              "issuer": "did:moltrust:issuer001",
              "holderBinding": "did:moltrust:agent042",
              "issuedAt": "2026-03-25T00:00:00Z",
              "expiresAt": "2026-04-25T00:00:00Z",
              "revocationEndpoint": "https://api.moltrust.ch/revocation",
              "onChainAnchor": {
                "chain": "base-mainnet",
                "block": 43825232,
                "txHash":
"0xcc425d150228c959565c4059eb07b0261afefc407156dc80f3c544216d3382da"
              }
            }
          }
        }
```


AAE boundaries are immutably anchored at issuance. An agent cannot modify its own envelope — any attempt to exceed declared permissions is detectable by any verifier inspecting the credential.

This is a fundamental separation: the agent executes within its envelope; the envelope itself is controlled by the issuing authority.

A Runtime Control Plane handles revocation events separately from issuance. Revocation follows CAEP-compatible (Continuous Access Evaluation Protocol) event semantics, allowing credential status changes to propagate in near-real-time without requiring reissuance of the entire credential.

The AAE design draws on NIST SP 800-162 (Guide to Attribute Based Access Control), W3C Verifiable Credentials Data Model 2.0, and OAuth 2.0 Rich Authorization Requests (RFC 9396).

4.7 Multi-Layer Enforcement

The Agent Authorization Envelope defines what an agent may do. But definition alone is insufficient — constraints must be enforced at every level of the stack. MolTrust implements a three-layer enforcement model:

Layer 1 — Cryptographic. Ed25519 signatures and JCS RFC 8785 canonicalization make AAE boundaries tamper-proof by construction. Any counterparty can verify that the AAE has not been modified since issuance — without infrastructure dependency.

Layer 2 — API. The MolTrust registry enforces constraints at the application level: trust score degradation on violation, Interaction Proof Record (IPR) submission for audit trails, and credential revocation for compromised agents.

Layer 3 — Kernel (in progress). Falco eBPF integration monitors agent behavior at the Linux syscall level. If an AAE declares that an agent may not write to `/etc/`, Falco detects the syscall and submits a violation proof to MolTrust — regardless of whether the agent's own runtime reports it. This is enforcement below the agent's process boundary. The agent cannot suppress or modify the detection signal.

Layer 3 is a qualitative difference from Layers 1 and 2. Cryptographic and API enforcement rely on the agent's environment to report violations. Kernel enforcement operates independently of the agent — it is the infrastructure itself that enforces the constraint.

Status: Layer 3 architecture is validated. Falco bridge deployment is in progress (April 2026).

4.8 Cross-Protocol Composability

The MolTrust Protocol is designed to compose with other agent trust frameworks rather than replace them. Three integration paths are currently active:

qntm Authority Constraints Working Group. MolTrust AAE fields map directly to qntm ConstraintEvaluation facets (5 of 5 dimensions verified via test vectors TV-001 through TV-005). JCS RFC 8785 and Ed25519 are shared primitives — a single signing operation produces artifacts verifiable by both systems.

Agent Provider Service (APS). MolTrust AAE trust scores map to APS Grade levels. Provider attestation import is in progress.

decision-equivalence. A three-level model for comparing agent decisions across protocol boundaries: correlation (temporal binding), equivalence (semantic binding via canonical outcome hash), and explanation (system-specific evidence). MolTrust implements Levels 1 and 3; Level 2 is planned for a future specification version.

The common thread: MolTrust does not require other protocols to adopt its formats. It provides mapping specifications that allow each system to verify MolTrust credentials using its own native verification logic. This is composability through shared primitives, not through format convergence.

4.9 Outcome Verification

Trust scores describe an agent's history. Outcome verification measures whether that history was accurate.

In prediction markets, MolTrust flags anomalous trading patterns before market resolution. After resolution, the Settlement Watcher compares flagged predictions against actual outcomes. The FlagScore — computed as (CONFIRMED + 0.5 × PARTIAL + 0.25 × INCONCLUSIVE) / total_flags — measures detection accuracy over time.

This creates a verifiable track record for the detection system itself: not just "we flagged this market" but "we flagged this market and were right 78% of the time." The FlagScore is anchored on-chain quarterly.

The term "anomaly" is used deliberately rather than "manipulation" — flagged patterns indicate unusual activity warranting investigation, not proven wrongdoing.

Status: Implementation ready, deployment in progress.

4.10 Regulatory Alignment — Singapore IMDA Model AI Governance Framework

The Singapore Infocomm Media Development Authority (IMDA) published the Model AI Governance Framework for Agentic AI in January 2026 — the first comprehensive regulatory guidance addressing autonomous AI agent systems. The MolTrust Protocol maps to each of the four MGF dimensions:

Dimension 1 — Risk Bounding. The MGF requires that agent systems implement mechanisms to bound the risks of autonomous action. MolTrust addresses this through the AAE mandate and constraints blocks, which define explicit boundaries on permitted actions, financial thresholds, jurisdictions, and delegation depth. Risk bounding is cryptographically enforced at the credential level — not as a policy overlay, but as a structural property of the authorization envelope. **Status: Fully Addressed.**

Dimension 2 — Human Accountability. The MGF requires traceable accountability chains linking autonomous actions to responsible humans. MolTrust addresses this through the five-party trust chain (Section 4.5), which establishes cryptographically verifiable links from Developer through Owner to Agent. The AAE constraints block supports approval thresholds that require human sign-off above defined limits. A dedicated human-in-the-loop (HITL) interface for real-time approval workflows is on the roadmap for Q2 2026. **Status: Partially Addressed.**

Dimension 3 — Technical Controls. The MGF requires robust technical safeguards for agent lifecycle management, including identity, monitoring, and revocation. MolTrust addresses this through the full DID+VC lifecycle — issuance, verification, time-to-live enforcement, and CAEP-compatible revocation. Interaction Proofs provide a continuous, tamper-evident audit trail. **Status: Strongly Addressed.**

Dimension 4 — End-User Responsibility. The MGF requires that end users understand the capabilities and limitations of agents they interact with. MolTrust currently provides machine-readable AAE metadata and verifiable credentials. Human-readable Agent Cards and explainability interfaces — enabling non-technical users to understand an agent's authorization scope — are on the roadmap for Q2 2026. **Status: Roadmap Q2 2026.**

The protocol is additionally designed for compatibility with the EU AI Act (August 2026 enforcement deadline), which imposes transparency and risk classification requirements on AI systems including autonomous agents; the South Korea AI Basic Act, which establishes accountability frameworks for AI operators; and the Taiwan AI Basic Act, which defines governance principles for AI deployment in regulated sectors.

4.11 Trust Tier 0 — Optional Human Identity Anchoring

The MolTrust Protocol is agent-centric. Anonymous deployment is permitted. No human identity disclosure is required to register an agent, issue credentials, or participate in the trust network. This is by design: many legitimate agent deployments — research prototypes, open-source tools, privacy-preserving services — have no reason to disclose the identity of their operators.

For enterprise and regulated deployments, the protocol defines an optional Trust Tier 0, in which the Developer or Owner identity is KYC-verified by an accredited third-party identity provider. Verification produces a KYC-backed Developer Credential — a W3C Verifiable Credential attesting that the named entity has been identity-verified to a defined assurance level.

Agents linked to a Trust Tier 0 Developer or Owner receive a score boost in the reference reputation model, reflecting the additional accountability signal. This boost is informative, not deterministic — verifiers may weight it as they choose.

Critically, Trust Tier 0 does not affect agent DID independence. A verified developer's agents still hold their own DIDs, their own credential chains, and their own behavioral records. If the developer credential is revoked, agent credentials are not automatically invalidated — each agent's standing is evaluated on its own merits.

What Trust Tier 0 is not:

- It is not mandatory. Agents without Trust Tier 0 are full participants in the protocol.
- It is not gatekeeping. No capability is restricted to Trust Tier 0 agents.
- It does not make MolTrust a regulated identity provider. MolTrust does not perform KYC — it accepts and verifies credentials issued by accredited third parties.

Trust Tier 0 is an opt-in accountability layer, expressible via the AAE CONSTRAINTS block, that bridges the protocol's agent-centric design with the human accountability requirements of regulated industries and emerging governance frameworks.

4b. Three-Layer Enforcement: From Declaration to Proof

Most agent authorization frameworks stop at declaration: the agent claims it will respect constraints. Some add verification: the API checks credentials on every call. MolTrust adds a third layer: kernel-

level enforcement that the agent cannot bypass from userspace.

This matters because the threat model for autonomous agents differs fundamentally from traditional software. A human employee who violates policy is visible — their actions leave traces, colleagues notice, audits catch it. An autonomous agent that modifies its own authorization envelope before acting leaves no human trace unless the infrastructure beneath it is watching.

Falco, the CNCF-graduated eBPF security tool deployed across AWS, GCP, and Azure, provides exactly this: syscall-level observation that operates below the agent's execution context. Combined with MolTrust's IPR recording and on-chain anchoring, a policy violation attempt — even one that is ultimately blocked by container permissions — becomes a permanent, verifiable event in the agent's trust history.

The three layers are complementary, not redundant:

- **Cryptography** proves identity and authorization at issuance time
- **API verification** proves compliance at request time
- **Kernel enforcement** proves behavioral integrity at runtime

No single layer is sufficient. Together, they close the gap that RSAC 2026 identified as the hardest open problem in agent identity: policy self-modification detection.

Reference implementation:
https://github.com/HaraldeRoessler/moltrust-falco-bridge

4c. Protocol-Agnostic Trust Layer

The MolTrust trust infrastructure is designed to be independent of any specific agent payment protocol. As the agent payment landscape has fragmented into multiple competing standards – x402 (Coinbase/Cloudflare), MPP (Stripe/Tempo/Visa), AP2 (Google), ACP (OpenAI/Stripe) – the need for a neutral, portable trust layer has become structurally evident.

No payment protocol defines how an endpoint operator verifies whether a paying agent is legitimate, trustworthy, or has been flagged for abuse. They handle payment mechanics, not agent trust. MolTrust fills this gap.

The DigiCert Analogy. DigiCert does not compete with HTTPS – it makes HTTPS trustworthy. MolTrust does not compete with x402 or MPP – it makes agent payments trustworthy, regardless of which payment rail the agent uses.

Protocol Coverage (April 2026)

Protocol	Ecosystem	MolTrust Integration
x402	Coinbase, Cloudflare	`@moltrust/x402` v1.0.1
MPP	Stripe, Tempo, Visa	`@moltrust/mpp` v1.0.3
AP2	Google, 60+ partners	Roadmap v0.9 Q3 2026
ACP	OpenAI, Stripe	Roadmap v0.9 Q4 2026

An agent that registers a W3C DID with MolTrust receives a trust score that is queryable by any endpoint operator regardless of which payment protocol the agent uses. The agent registers once; the trust score is valid everywhere.

This is the structural advantage of building on W3C DID/VC standards rather than protocol-specific identity systems: portability is guaranteed by the standard, not by bilateral agreements.

---

5. Speed and Permanence

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

Verification — checking identity, validating credentials, querying behavioral history — 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 and recording confirmed violations — benefits from tamper-evident, distributed storage. These operations are immutably anchored and occur infrequently: once at registration, and when a violation is confirmed. The immutable anchor layer is used specifically where permanence and public verifiability matter — not as a general transaction layer.

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.

This separation means routine agent interactions carry no blockchain overhead. The facts that matter most — identity registration and confirmed violations — are permanently and publicly verifiable by anyone.

Stake

Agents may optionally deposit a stake at registration — a defined amount held in a smart contract. The stake creates an economic commitment to declared behavior: it is returned on clean deregistration and forfeited if a confirmed violation is recorded. Stake is one signal among others; it is not required for participation.

5.1 Multi-Chain Wallet Binding

MolTrust is chain-agnostic by design. Wallet binding supports Ethereum, Base L2, and Solana. Additional chains follow the same principle: the agent signs a standardized message with its private key, MolTrust verifies the signature, and binds the wallet address cryptographically to the DID.

Solana uses Ed25519 — the same signature scheme MolTrust uses internally for its own signing keys. This makes Solana-native agents first-class citizens in the MolTrust trust model. No adapter, no bridge contract, no additional dependency.

Each chain binding produces a chain-specific payment service entry in the agent’s DID Document, enabling cross-chain payment readiness from the moment of binding.

5b. Interoperability

External DID Bridging

The MolTrust Protocol acknowledges that agents already have existing identities in other ecosystems. DID Bridging allows an external DID identity to be cryptographically linked to a `did:moltrust`

identity — without abandoning the external identity.

The bridge proof is a signature over both DIDs, created with the wallet bound to the `did:moltrust` identity. This makes the link cryptographically provable and tamper-evident. Bridging is not transitive: if A bridges to B and B bridges to C, resolving A does not return C. Each link is independent and independently revocable.

Any DID method that supports Ed25519 or secp256k1 signing can bridge to MolTrust. The protocol does not prescribe which external methods are valid — it only requires that the linking proof is cryptographically verifiable.

Cross-Ecosystem Trust Score Import

Agents that have built reputation in external systems can import that signal as a basis for their MolTrust trust score. External scores are:

- Mapped logarithmically to the MolTrust scale (0–100)
- Weighted at 0.3 relative to native MolTrust endorsements (1.0)
- Subject to a 45-day half-life (vs. 90 days for native scores)

This design ensures that a one-time external score cannot permanently dominate an agent's standing. Trust must be continuously earned through verified interactions. External reputation provides a starting point — not a permanent advantage.

---

6. Resistance to Manipulation

Any reputation system operates under the assumption that participants may attempt to manipulate it. The protocol addresses the most common vectors — without claiming to eliminate the underlying incentive.

Manufactured reputation — creating multiple coordinated fake identities to inflate an agent's standing — is made significantly more costly by requiring endorsements from verified agents across multiple independent domains. A convincing manufactured reputation requires not just volume but verified diversity.

Accumulated trust exploitation — building a clean behavioral record over time before executing a high-value fraudulent interaction — 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.

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.

Reputation systems record what an agent has done — after the fact. The MolTrust Protocol verifies what an agent is authorized to do — before every transaction. Pre-transaction trust verification is not a replacement for reputation; it is its necessary precondition.

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.

The **protocol standard** (Layer A) is 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.

The **reference registry** (Layer B) is the MolTrust-operated service layer: identity resolution, credential revocation, trust score queries, and immutably anchored records. Other operators may run conformant registries using their own infrastructure.

The **reference reputation model** (Layer C) is 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 Decentralized Identifiers (DID Core, v1.0) provide the identity layer. The DID specification is a W3C Recommendation, implemented across a wide range of existing systems.

W3C Verifiable Credentials (VC Data Model 2.0) provide the authorization and attestation layer. Verifiable Credentials are in active use in national digital identity programs — including the EU Digital Identity Wallet framework under eIDAS 2 — academic credentialing, and emerging enterprise identity management deployments.

ERC-8004 provides optional on-chain agent registration for deployments that require blockchain-anchored identity. Where on-chain anchoring is not required, the protocol functions without it.

The protocol's contribution is not new standards. It is the application of existing standards to a specific gap — agent-to-agent trust — together with 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 for AI agent trust, combining DID/VC-based identity with a Byzantine Fault Tolerant trust plane and a "Proof-of-Behavior" consensus mechanism. TAM and MolTrust share a diagnosis — the identity-behavior gap in distributed agent systems — but differ significantly in approach. TAM is a complete infrastructure stack requiring consensus participation. MolTrust is a minimally prescriptive evidence standard: it defines what evidence must be recorded and how it must be structured, without prescribing how consensus is reached or which infrastructure components must be deployed.

AgentHub addresses agent discoverability and provenance through signed manifests and namespace control, drawing on software supply-chain security concepts. AgentHub and MolTrust are complementary: AgentHub addresses how agents are discovered and their code provenance verified; MolTrust addresses how agents' behavioral history and authorization are verified at interaction time.

ERC-8004 defines on-chain registries for agent identity, reputation, and validation, with a focus on crypto-economic proofs and trustless on-chain collaboration. MolTrust's relationship to ERC-8004 is explicitly complementary: MolTrust defines off-chain evidence formats and scoring semantics; ERC-8004 provides one anchoring mechanism for those facts on-chain. The two are not in conflict.

W3C AI Agent Protocol Community Group and the **DIF Trusted AI Agents Working Group** are active standardization venues addressing agent identity, delegation patterns, and cross-domain trust. Both are in early exploratory phases. MolTrust is more concrete on data models, verification procedures, and conformance requirements, while these groups are broader in scope and earlier in development. 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 with hard delegation constraints, protocol-level interaction proofs, and a portable behavioral reputation model, expressed as an implementable layered standard — does not appear to be covered by any existing published specification as of the date of this document.

---

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 step — because verification infrastructure supports oversight rather than replacing 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.

The protocol formalizes a five-party trust chain (Developer, Owner, Agent, Instructor, Counterparty) in which each link is cryptographically verifiable, and introduces the Agent Authorization Envelope (AAE) — a machine-readable structure defining mandate, constraints, and validity for every issued credential. The AAE enables pre-transaction trust verification: any counterparty can inspect an agent’s authorized scope before committing to an interaction.

AAE constraints are enforced through a three-layer model — cryptographic tamper-proofing (Ed25519 + JCS), API-level trust score degradation and credential revocation, and kernel-level syscall monitoring via Falco eBPF integration — ensuring that enforcement operates independently of the agent’s own runtime.

The protocol composes with other agent trust frameworks through cross-protocol mapping specifications: qntm Authority Constraints (shared JCS/Ed25519 primitives), Agent Provider Service grade-level mapping, and a three-level decision-equivalence model for comparing agent outcomes across protocol boundaries.

Outcome verification extends trust scoring from historical behavior to predictive accuracy, with FlagScore measuring detection system performance against actual market resolutions — creating a verifiable track record for the detection infrastructure itself.

The protocol aligns with the Singapore IMDA Model AI Governance Framework for Agentic AI across its four dimensions — risk bounding, human accountability, technical controls, and end-user responsibility — and is designed for compatibility with the EU AI Act and emerging APAC regulatory frameworks. An optional Trust Tier 0 provides KYC-backed human identity anchoring for enterprise and regulated deployments without compromising the protocol's agent-centric, permissionless design.

The protocol is designed to:

- Enable any party to verify an agent's identity, authorization, and behavioral history without access to a central registry
- Operate at interaction speed for routine verification, with permanent immutably anchored records reserved for registration and confirmed violations
- Function across jurisdictions, platforms, and economic systems without modification
- Remain separable from any single operator, including MolTrust

The protocol supports multi-chain identity through wallet binding (Ethereum, Base, Solana), external DID bridging for cross-ecosystem portability, and trust score import with reduced weight and accelerated decay.

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 (v0.8) provides complete data models, verification flows, and conformance requirements.

MolTrust / CryptoKRI GmbH, Zurich api.moltrust.ch · moltrust.ch · info@moltrust.ch

---