From Agent-Visible to Agent-Actionable: KYA-OS, a Rail-Agnostic Trust Layer for E-commerce

Summary of Proposed Contribution

Being noticed by AI shopping protocols is only half of a retailer's problem. An offer an agent can find but cannot safely act on is a listing, not a sale. An offer becomes fully agent-actionable only when the agent's authority is verifiable, which makes the trust layer part of the publisher surface. Our proposed contribution is KYA-OS, the specification for exactly that layer: open-source identity, delegation, and accountability for agents acting on behalf of users, with a candidate profile proving an agent's authority to any merchant, on any rail. Authored by Dylan Hobbs at Vouched, an identity-verification company, it builds on open standards: W3C Decentralized Identifiers and Verifiable Credentials, JOSE/JWS, RFC 9421 HTTP Message Signatures, and the Model Context Protocol (MCP). We donated the specification to the Decentralized Identity Foundation in March 2026; DIF's Trusted AI Agents Working Group (TAAWG) now stewards it. Vouched was also ranked #1 by CB Insights in the KYA (know your agent space) based on its reference implementation of KYA-OS.

Relevance to the Workshop

A retailer's path to being noticed runs through structured data and, increasingly, commerce protocols: UCP and ACP standardize how carts and orders move between agent and merchant; MCP provides the generic tool channel. But visibility ends at the moment of action. Adjacent trust problems are progressing (Web Bot Auth at the HTTP layer, AP2 mandates in payment flows, card-network tokens on their rails), yet each answers part of the question. What remains open is what companies need before a visible offer safely converts: rail-agnostic proof of an agent's authority to act.

This touches each audience that the workshop names. Content creators gain an accountable counterparty where they now see anonymous bot traffic, and can serve, license, or gate accordingly. SEO practitioners inherit a new optimization surface: a verifiable offer converts where an unverifiable one stalls. Brands scope authority to their GS1-identified product lines; retailers verify at the edge. Agent providers, including in-browser agents, present one profile everywhere instead of onboarding per platform. Online data providers supply the vocabularies delegations scope to and the registries reputation and revocation ride on.

The demand is public. The IMF urges regulators to weigh Know-Your-Agent requirements with verifiable identities bound to legal entities, and lists revocation among core delegation capabilities; NIST has opened an industry-led agent-standards initiative with identity and authorization among the first work items. Companies operating at scale concede the same gap: no shared primitive proving delegated authority. Solving this in the open keeps agentic commerce from consolidating around a few closed platforms that own identity, consent, and reputation.

Main Points

1. An offer is only fully noticed when an agent can act on it; declaring accepted trust schemes belongs on the discovery surface, beside catalog and endpoints.
2. Agent authorization should be a rail-agnostic, cross-protocol grant: bounded by scope, spend, and time, chained to an accountable party (where reputation must attach, or bad actors launder history via fresh agents), independently verifiable by any merchant, with revocation and reputation that travel across platforms.
3. Bounded authority belongs in the signed credential; spend enforcement stays with the merchant or payment system, keeping the trust layer out of PCI scope.
4. Our most instructive failures were operational, not cryptographic: clock skew tripped replay windows more often than forged signatures, and whether a failed revocation check fails open or closed proved merchant policy, not protocol.

Use Cases and Evidence

Consider a shopping assistant authorized to buy a retailer's offer under a set budget. The agent presents a W3C DID and signs each request with a local Ed25519 key. Its delegation is a Verifiable Credential carrying scope, spend cap, time window, and audience, chained to a Responsible Party. The merchant edge verifies the signature, replay window, scopes, and revocation status; a deterministic Bayesian reputation score, bound to the Responsible Party rather than a disposable agent, gates sensitive actions. The sale completes because the retailer verified authority, not because both sides trusted one platform.

All of this runs today: a published MCP package, a public did:web registry, a merchant-edge verification service, a deployed reputation service, and a compliance matrix mapping every conformance requirement to an executable test.

Relationship to Web Standards and Technologies

The grant is JSON-LD and composes with vocabularies retailers already publish: a delegation can scope authority to a schema.org Offer or a GS1 Digital Link-identified product class, so one authorization reads across catalogs and stores. Rather than propose a new well-known document, KYA-OS rides surfaces agents already index: well-known metadata publishes a verifier's DID, capabilities, proof algorithms, clock-skew tolerance, and revocation list location, so an agent learns what a merchant accepts before presenting. We have implemented this pattern as the Entity Card profile, shipped in @kya-os/mcp: one typed, DID-anchored card projected onto surfaces agents already index (MCP server.json metadata, A2A AgentCards, NANDA AgentFacts) rather than a new well-known document. If the community converges on a publisher-side convention for agent-facing capabilities, this profile composes naturally: the surface declares accepted trust schemes; the profile is what the agent presents.

Questions for Workshop Discussion

- Should agent-facing trust signals live in a new well-known document, or be layered into discovery surfaces agents already index?

- Once agent authority is verifiable, where should liability sit: with the Responsible Party, the merchant, or the payment rail?
- Should agent reputation and revocation be governed by a single trust root, or federated across registries as DNS and CA trust are?
- What would meaningful representation for companies in governing an agent-trust standard concretely look like?

Suggested Outcome

We will contribute our conformance suite, test vectors, and reference implementation toward a shared, minimal on-behalf-of authorization profile publishable on the agent-facing surface, ideally via a W3C Community Group liaising with DIF's Trusted AI Agents Working Group. KYA-OS is raw material, not a finished answer; the profile should be shaped by retailers, agent providers, payments, and browser communities. We offer a talk with a live demonstration on the published stack: an agent discovers accepted trust schemes from well-known metadata, presents a delegation scoped to a GS1 Digital Link product class, and we revoke it mid-session so the next request fails per merchant policy. We would welcome a breakout treating trust, merchant liability, and portable reputation as one connected problem.

Supporting Resources:

1. KYA-OS Protocol Specification (donated to DIF TAAWG, under review for ratification): https://github.com/decentralized-identity/kya-os-mcp/blob/v1.9.0/SPEC.md
2. npm package: https://www.npmjs.com/package/@kya-os/mcp
3. DIF Trusted AI Agents Working Group: https://identity.foundation/working-groups/trusted-agents.html
4. Google, Announcing Agent Payments Protocol (AP2): https://cloud.google.com/blog/products/ai-machine-learning/announcing-agents-to-payments-ap2-protocol
5. Visa, Trusted Agent Protocol announcement: https://investor.visa.com/news/news-details/2025/Visa-Introduces-Trusted-Agent-Protocol-An-Ecosystem-Led-Framework-for-AI-Commerce/default.aspx
6. IMF Note 2026/004, How Agentic AI Will Reshape Payments (April 2026): https://www.imf.org/en/publications/imf-notes/issues/2026/04/22/how-agentic-ai-will-reshape-payments-575560
7. NIST AI Agent Standards Initiative (February 2026): https://www.nist.gov/news-events/news/2026/02/announcing-ai-agent-standards-initiative-interoperable-and-secure
8. Merchant-edge verification platform: https://kya.vouched.id
9. Conformance requirements (compliance matrix, each requirement mapped to an executable test):https://github.com/decentralized-identity/kya-os-mcp/blob/v1.9.0/CONFORMANCE.md