OpenA2A

The identity crisis of autonomous AI.

Why your agents need cryptographic proof.

Abdel Fane

Founder and CEO, OpenA2A · SVCC 2026 · Industry Speaker Session

WHAT WE ARE SEEING

Identity is necessary. It is not sufficient.

NECESSARY

We can't attribute what we can't identify.

274,577 events across the honey-trap fleet

42% attackers came back, of 5,891 ever seen

NOT SUFFICIENT

A verified agent still follows what it reads.

2,719 agents followed an injection and called the attacker

407 injection surfaces hidden across the sampled web

Identity proves who. Not what.

OpenA2A honeypot network: TrapMyAgent, AgentPwn, HoneyMap. research.opena2a.org

AGENDA

Four parts.

1	**The problem**	Identity is here. Trust is not.
2	**The open stack**	Identity, trust, authorization, threat modeling, observability.
3	**The reference implementations**	AIM, the conformance suites, demo.
4	**The call to contribute**	These standards belong to everyone who works on agents.

Q and A follows.

WHO IS TALKING

Abdel Fane.

Founder and CEO, OpenA2A · Executive Director, CSNP

20 years in enterprise security

- Consulting at Booz Allen Hamilton and Protiviti
- In-house security at Allstate and Grail
- Product security at the Allstate Compozed Lab
- Fortune 10 to small clinics

Building the trust layer for AI agents

- A2A protocol identity and trust contributor, PR #1496 (in flight)
- OpenTelemetry SemConv contributor, Issue #180
- 10 vulnerabilities disclosed to NVIDIA PSIRT for NemoClaw
- 7 vulnerabilities fixed in OpenClaw following coordinated disclosure
- Contributed HackMyAgent core scanner to OpenClaw (PR #9806)

01

PART 01

The problem.

Identity for agents is here. Trust is not.

THE PROBLEM

Agents act in your name. Your stack is blind.

An AI agent in your enterprise today:

It acts continuously, with no human in the loop

It calls APIs, queries databases, and makes decisions. No approval step. No session a human signed into.

It has no verifiable identity

The agent is a process holding an API key, or holding nothing at all. The key is not the agent.

It is invisible to your SIEM

Your SIEM was built for human logins and human sessions. It does not recognize an agent as an actor.

> ***Your IAM was built for people. Your SIEM does not understand agents. The agent economy was built on an identity gap.***

Identity for agents is here.

Trust is not.

The industry shipped agent identity. The layer that keeps it meaningful did not ship with it.

Human IAM does not transfer to agents.

Human identity, the past	Agent identity, the need
Authenticates once per session	Acts thousands of times, with no session boundary
Identity is stable and human-scale	Spawns sub-agents, delegates, and changes over time
A person stands behind every credential	A process, often with a machine or no human behind it
Reviewed, offboarded, and audited by humans	Created by a developer in seconds, then ungoverned

> ***Agents need their own security layer. Identity built for agents, not retrofitted from human IAM.***

THE CORE PROBLEM

Identity is point-in-time. Trust must be continuous.

`T + 0`	The agent is verified. Its cryptographic identity is valid.
`T + 30s`	The agent reads a web page carrying an indirect prompt injection.
`T + 30s`	**The identity is still valid. The agent is now compromised.**

A verified identity is a snapshot. Verification does not survive contact with a prompt injection. The agent that passed your check is not the agent making the next call.

> ***Point-in-time verification cannot catch what happens after the check. Continuous runtime enforcement can.***

02

PART 02

The open stack.

Identity.
Trust.
Authorization.
Threat modeling.
Observability.

WHAT IS IN THE STACK

Thirteen open specs across six categories.

Use what makes sense. Each spec is independent, vendor-neutral, Apache 2.0.

IDENTITY

Agent Identity Protocol (AIP) *Draft*
Cryptographic agent identity and verification.

did:opena2a *Filed, in review*
Registry-resolved DIDs for agents and tools.

TRUST

Agent Trust Protocol (ATP) *RC1*
Issue, verify, and revoke trust assertions.

Agent Trust eXtension (ATX) *v1.0 spec, v1.1*
Signed, locally-verifiable agent credential.

AUTHORIZATION

Agent Authorization Protocol (AAP) *Draft*
Scoped access, remove credentials from context.

THREAT MODELING

Agent Threat Matrix *v1.0*
ATT&CK-style catalog of agent attacks.

Agent Behavioral Governance Spec (ABGS) *Draft*
SOUL.md governance: domains and controls.

AIIS Signatures *Draft*
YARA-style injection detection signatures.

BENCHMARK + OBSERVABILITY

Open Agent Security Benchmark (OASB)
Published
Detection-coverage benchmark for agents.

OTel SemConv *Draft*
Agent authorization attributes for traces.

CONFORMANCE + SDKS

ATX Conformance *Suite*
Reference verifiers, byte-stable fixtures.

A2A-IDF Conformance *Suite*
Cross-impl conformance for A2A identity.

A2A-IDF SDK *TypeScript*
Verifier-first SDK, RFC 9421 over Ed25519.

Identity. Who is this agent?

Every agent gets a verifiable keypair, not a shared secret.

Agent Identity Protocol (AIP). Draft.

Identity issuance, attestation, and trust. github.com/opena2a-standards/agent-identity-protocol

did:opena2a. Filed with W3C, pending review.

Registry-resolved identifiers for registries, authorities, publishers, agents, MCP servers, tools, LLMs, and skills.

Post-quantum ready.

AIM now issues hybrid Ed25519 and ML-DSA-65 credentials in the reference implementation. NIST IR 8547 sets a 2030 deprecation deadline for classical signatures.

> ***Identity proves who. Trust answers whether they are still acting like themselves.***

Trust. Can you rely on this agent right now?

Not a label. A verifiable, continuous signal.

Agent Trust Protocol (ATP). Release candidate.

RFC 6962 transparency log, federated CRL, hybrid signing. Reference verifiers in Go and Python, byte-stable fixtures.

Agent Trust eXtension (ATX) credential. v1.0 spec, v1.1 issued.

Agent Trust eXtension. Verified locally in under 5 ms, no network call. 7-day expiry. v1.0 signs identity and trust level. The reference registry now issues v1.1, adding signed capabilities and scan summary.

Nine-factor continuous trust scoring.

The score updates on every action. Three factors require external attestation. Detail in Part 3.

> ***ATX is the credential. ATP is the protocol around it. Both open, both Apache 2.0.***

Authorization. What is this agent allowed to do?

The agent declares its capabilities. The credential never enters the agent context.

Agent Authorization Protocol (AAP). Draft.

Resolves an agent's ATX trust into scoped resource access through a grant reference.

Credential Provider Interface.

Retrieve, Assume, Exchange. The agent gets a grant reference, never the credential value.

Decision and enforcement split.

Blocks unauthorized actions at the API layer, including actions triggered by prompt injection.

> ***The injection still lands. The action outside the declared grant does not.***

Threat modeling. What are agents actually attacked with?

Data-backed, not lab-only. Wild observation drives the catalog.

20

wild-observed techniques

38

lab-validated techniques

3

adapted from adjacent domains

Agent Threat Matrix. v1.0. 61 techniques across 9 tactics, mapped to MITRE ATT&CK, ATLAS, and OWASP. Sibling specs add the SOUL.md governance frame (ABGS, draft) and YARA-style signatures for prompt injections in web content (AIIS Signatures, draft).

> ***This is measured, not modeled. TrapMyAgent logged 274,577 events, and AgentPwn caught 2,751 agents following an injection.***

Measure what you build. Observe what runs.

Standard benchmarks. Standard traces. Comparable across implementations.

OASB. Open Agent Security Benchmark. Published.

222 attack scenarios mapped to MITRE ATLAS. Product-agnostic adapter interface. oasb.ai

OTel SemConv for agent identity. Draft.

OpenTelemetry semantic conventions for AI agent authorization observability. Issue #180.

> ***When every vendor invents its own metrics, comparison becomes impossible.***

03

PART 03

The reference implementations.

AIM. The conformance suites. The demo.

NOT A SLIDEWARE PROPOSAL

AIM is the reference implementation.

The stack, working today, in open source.

Spec	AIM, in code today
AIM and ATX	Ed25519 keypairs, hybrid ML-DSA-65 signing live, ATX v1.1 credential issued
ATX capability enforcement	Declared capabilities, runtime enforcement at the API layer
MCP attestation	Multi-agent consensus attestation, drift detection, dependency graph
ATP transparency log	RFC 6962 Merkle transparency log, live 9-factor trust score

Apache 2.0. Python, npm, and Java SDKs. Self-host or AIM Cloud.

WHAT IT LOOKS LIKE ON THE WIRE

Every decision is a traceable span.

An authorization decision, as a real OpenTelemetry span tree.

```
fga.authorize                              [parent]
  agent.id                    "a2-38176c67"
  agent.public_key.algorithm  "ed25519"
  agent.capability            "db:read"
  agent.trust_score           0.92
  fga.outcome                 "DENY"
  fga.denied_by               "capability_check"

  fga.capability_check                     [child]
    fga.step                  "capability_check"
    fga.allowed               false
```

These attribute names follow the OpenTelemetry Issue #180 conventions. The trace is a standard your existing observability stack can read.

Behavioral trust scoring: 9 factors.

A continuous signal that updates on every action. The nine factors sum to 100%.

25% Verification status

Identity and attestations check out.

15% Uptime and availability

Consistently reachable and responsive.

15% Action success rate

Share of actions completing without error.

15% Security alerts

Open security findings, weighted by severity.

10% Compliance

Meets the policy and control requirements set.

10% Execution isolation

Sandboxed from other agents and systems.

5% Age and history

Time run with a clean track record.

3% Drift detection

Distance of current behavior from baseline.

2% User feedback

Signals from the humans who supervise it.

Three of these factors require external attestation. An agent cannot self-attest its way into a high trust score.

RECEIPTS

Every spec ships with conformance.

Reference verifiers. Byte-stable fixtures. Reproducible.

ATX Conformance.

Go and Python reference verifiers. Byte-stable fixtures across schema, expiry, revocation, threshold cosignature, hybrid signing, tamper, malformed schema. 8 of 8 pass.

A2A-IDF Conformance.

Canonical conformance suite for the Agent-to-Agent Identity Framework. Aligned to A2A PR #1496.

A2A-IDF SDK.

TypeScript. RFC 9421 and Ed25519 wire signatures, attestation envelopes, delegation chains. Paired with the conformance suite.

> ***Run the suites against any implementation, including ours. The bytes are the contract.***

LIVE DEMO

Same agent. One has identity.

DVAA, the Damn Vulnerable AI Agent, run twice.

Run A DVAA unprotected. The injection lands. The agent exfiltrates data.

Run B The same DVAA agent, registered with AIM. The same injection. The action is blocked.

Then The audit log entry, and the trust score drop.

> ***Live. No slides for the next four minutes.***

04

PART 04

The call to contribute.

These standards belong to everyone who works on agents.

WHAT YOU CAN DO

Pick the spec your team cares about.

All thirteen specs are open.

github.com/opena2a-standards. Apache 2.0. Each repo has a contribution guide.

Governed independently.

OpenA2A has no veto. Contributions land as PRs, issues, and conformance results.

We have the wild data.

The honey-agent fleet, the threat matrix, the wild bait sampler. That is what makes the catalog real.

The conventions are getting set in 2026.

Help us make them open ones.

> ***Pick the layer you care about. Bring your team. Help us define what comes next.***

Thank you.

Questions welcome. Find the work here:

specs.opena2a.org [Seeking Co-authors]

The thirteen open specs

research.opena2a.org [Seeking Contributors]

Behavioral threat reports and the agent threat data

opena2a.org

AIM, DVAA, and the open-source trust layer

Abdel Fane · Founder and CEO, OpenA2A · abdel@opena2a.org

linkedin.com/in/abdelfane