NON-HUMAN IDENTITY SERIES · PAPER #3

Authorization Without Disclosure

An advisory reference architecture for non-disclosing agent-to-agent authorization

Mohamad Amin Hasbini

Independent researcher · Paris, France

Published May 05, 2026

ABSTRACT

Existing agent authorization designs require receipt bodies, scope contents, and operator instructions to be presented to verifiers in plaintext. This paper closes that disclosure gap with a zero-knowledge construction over a post-quantum substrate, and describes the surrounding AAC reference architecture.

KEYWORDS — AI agent security, non-human identity, post-quantum cryptography, agent authorization, capability tokens, NIS2, DORA, EU AI Act

---

CITE AS

Hasbini, M. A. (2026). *Authorization Without Disclosure*: An advisory reference architecture for non-disclosing agent-to-agent authorization. Non-Human Identity Series, Paper #3.
Available at https://mahasbini.org/papers/03-authorization-without-disclosure/
PDF https://mahasbini.org/publications/papers/03-authorization-without-disclosure.pdf

TL;DR

- **The disclosure gap.** Most current agent authorization designs require the verifier to read the receipt body, the scope, and the operator instructions in plaintext. Privacy-preserving work exists at adjacent layers (credential-claim selective disclosure, hardware-attested execution ZKP), but no current design proves the composed authorization decision itself in zero knowledge over a sealed receipt body. AAC's contribution is at the authorization-decision layer.
- **The construction.** A zero-knowledge proof over the receipt's verifier predicate, expressed as a circuit over hash-based commitments, Merkle paths, and arithmetic constraints. The verifier learns ALLOW or DENY plus minimal predicate truths. The body, the instructions, the full scope, and the unrelated authorizations stay sealed.
- **The architecture.** AAC (Authentication, Authorization, Communication) is the surrounding reference architecture. Three pillars, seven guarantees, six phases, twenty-two operational steps. Post-quantum substrate, plain proven primitives in the application layer, deployable from May 2026 components (the composition itself remains reference-implementation work).
- **The compliance angle.** EU AI Act, NIS2, DORA, and GDPR converge on third-party verifiability without business-content disclosure. Append-only commitments anchored to a transparency log, body sealed at issuer, regulator-readable chain integrity. The architecture structurally addresses the tension between auditability and data minimisation: auditors can verify chain integrity, authorization validity, revocation state, and boundary conformance from commitments and proofs, while receipt bodies and business content remain sealed unless separately disclosed under a lawful process.

A compliance auditor walks into a bank in 2027. The bank deploys AI agents that triage email, schedule meetings, and route customer requests through internal systems. The auditor’s mandate, under EU AI Act Annex IV and NIS2 Article 21, is to verify that every agent action over the past quarter was authorized by a valid delegation, that each delegation respected its declared boundaries, that nothing escalated privilege, and that revocations propagated correctly. The bank’s mandate, under GDPR and customer contracts, is to disclose nothing about the customer business content those agents handled.

The first mandate and the second appear to contradict.

The contradiction is not real. It is a property of how authorization is verified today. Most token- and receipt-based designs evaluate authorization over disclosed claims, disclosed receipt fields, or issuer-side introspection. Even when the resource server receives an opaque token, some authorization component must still evaluate the underlying claims in the clear. OAuth 2.0 Token Exchange is the canonical example: the verifier reads the token’s claims to make the authorization decision. Agent-delegation and workload-identity drafts emerging in the IETF community through 2025-2026 inherit this disclosure pattern.

Other work in the agent identity and authorization space has solved adjacent problems. Who the agent is. Where its identity lives. How a delegation can be narrowed as it passes between agents. None of those approaches verifies the full agent-delegation authorization decision in the sense this paper means. Selective-disclosure and zero-knowledge mechanisms exist at the credential-claim level (SD-JWT, BBS+ signatures, Verifiable Credentials profiles), but none extend to the composed multi-condition decision over a sealed receipt body. That is the gap this

paper proposes a construction to address. A detailed survey of related work is in the AAC Construction Specification companion document.

This paper proposes a construction to address that gap.

AAC does not try to prove that the agent reasoned correctly, or that the model was not prompt-injected, or that a natural-language instruction was semantically obeyed. Those are different problems. AAC proves a narrower and checkable statement: that the proposed action stays inside what the user signed off on, in the time window the user signed off on, with the operator's instructions intact, and that the delegation has not been revoked. Bounding the claim this way is what makes the construction implementable.

> *Authorization can be verified without disclosure. The verifier predicate, recast as a zero-knowledge circuit over hash-based commitments, allows a Policy Decision Point to confirm that all conditions of authorization hold without ever reading the underlying receipt, instructions, or scope. The construction operates over a post-quantum substrate that protects identity, transport, and the public log on which receipt commitments are anchored. The result is an architecture in which auditability, privacy, and post-quantum durability stop being trade-offs and start being properties that compose.*

The paper is structured in three parts. Part I states the disclosure gap. Part II walks the AAC architecture end-to-end through a concrete scenario, with the three pillars, seven guarantees, layered architecture, and six-phase flow grounded in named actors. Part III describes the zero-knowledge construction. Detailed protocol specifications, the related-work survey, deep implementation considerations, and the open questions with their resolution paths are in the AAC Construction Specification companion document.

Part I: The disclosure gap

An authorization decision rests on seven conditions, each of which must hold:

1. The receipt has not been revoked.
2. The signatures on the receipt and the log anchor verify against the known public keys.
3. The current time falls inside the receipt's declared time window, where "current time" is derived from a fresh signed log checkpoint, TSA timestamp, or revocation-map epoch accepted by the verifier within a max-staleness window Δt, not from the agent's client clock.
4. The proposed action falls inside the receipt's declared scope.
5. The capability chain, if the receipt was forwarded through intermediate agents, satisfies strict-subset attenuation at every hop.
6. The operator instructions delivered to the agent match the instruction commitment hash baked into the receipt at issuance.
7. The local policy at the receiving domain (the equivalent of the firewall rule that says "never accept this action class from this caller") returns allow.

Different agent-delegation designs published as of May 2026 partition this predicate differently. Some cover only signatures, time, scope, and chain. Some bundle policy with scope. Some omit operator-instruction integrity entirely. Across every design, the common pattern holds: **the verifier evaluates the predicates by reading their inputs in plaintext.** The receipt body, the scope contents, the operator instructions, and the intermediate receipts in the chain all surface to whoever performs verification.

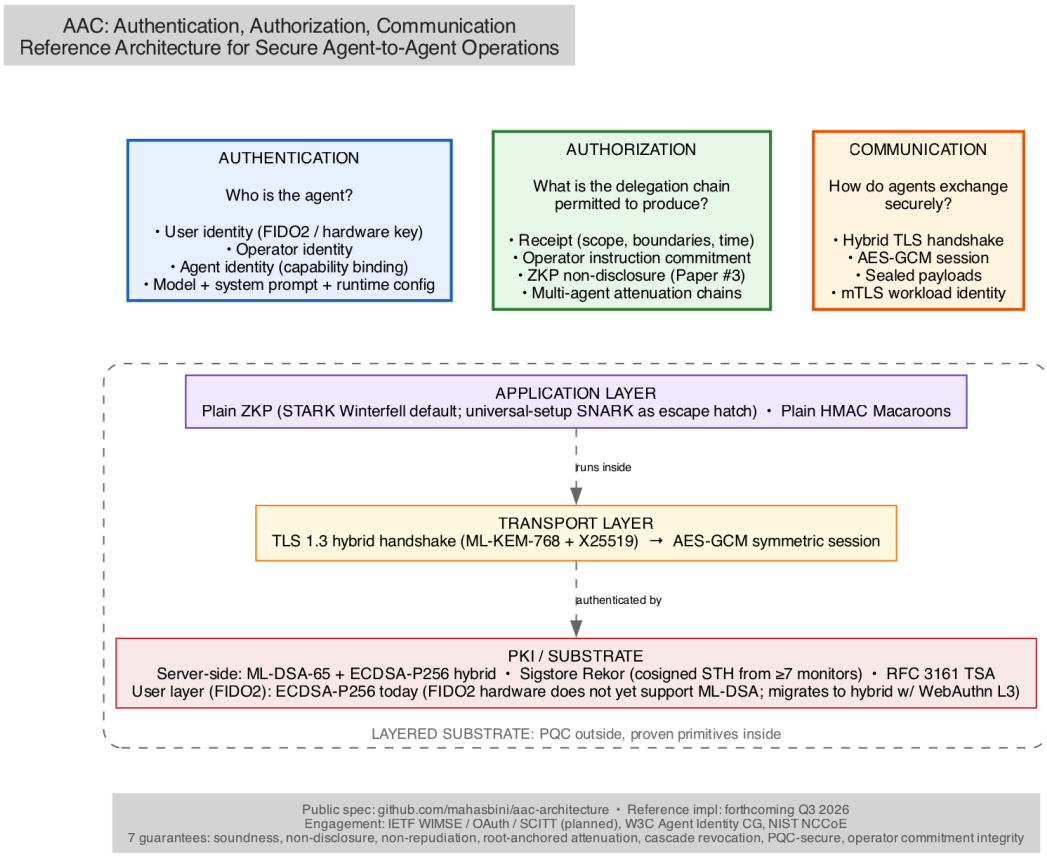
For agents in 2024 this was not a complaint. The agents were single-tenant, single-domain, and the verifier was the same entity that issued the receipt. There was no real disclosure cost.

By 2026 the disclosure cost has become structural. The verifier is increasingly not the issuer: a compliance auditor at Bank X verifying agent actions issued by AcmeAI is not the issuer; a regulator inspecting an EU-deployed agent issued by a US vendor is not the issuer; a cross-domain Policy Decision Point sitting at the receiving service in a different organisation is not the issuer. The verifier is increasingly subject to compulsion (National Security Letters, MLAT requests, CJEU production orders, litigation discovery), so the receipt body, once disclosed to the verifier, lives wherever the verifier lives. And what the receipt contains is exactly what regulation forbids you to expose: institutional client identities, recipient identifiers, sensitivity classifications, patient diagnoses. The semantic content of the authorization is the content that GDPR Article 5, the EU AI Act Article 12, and HIPAA Section 164 require minimised, sealed, or disclosed only under named legal bases.

These conditions produce the design constraint this paper addresses. **The verifier must be able to confirm that the predicates of authorization hold without reading what they are predicates over.**

Zero-knowledge proofs answer that question. The cryptographic primitive has been deployable for a decade. What was missing was the agent-authorization context that made the disclosure gap operationally pressing. That context is now here.

Part II: AAC in operation



The architecture at a glance. Three pillars on top, three layers underneath. The pillars define what the architecture answers. The layers define how it composes.

The architecture is organised around three pillars. **Authentication** answers "who is this agent": cryptographic identifier plus model identifier plus system prompt commitment plus runtime configuration, key custody, and capability binding. **Authorization** answers "what is this delegation chain permitted to produce": agent permissions (scope, boundaries, time window), operator instruction commitment (so drift between what the user signed off on and what the agent runtime receives is cryptographically detectable), and chain attenuation rules (so downstream agents cannot escalate beyond the user's original authorization). **Communication** answers "how do agents exchange securely": post-quantum-protected channels, hybrid signatures on long-lived artifacts, sealed payloads.

The pillars are not independent. An authorization decision depends on an authentication of the agent. A communication channel depends on the authorization to use it. AAC's contribution is the integration; existing IETF, W3C, and vendor designs solve each pillar in isolation.

The architecture is also layered for deployment necessity, not aesthetic preference. Long-lived signatures live at the substrate (hybrid ML-DSA-65 + ECDSA-P256), so a future quantum-capable adversary cannot retroactively forge them by harvesting today's signatures. Transport is post-quantum-protected (TLS 1.3 hybrid). The application layer carries the zero-knowledge proof and macaroon-based capability tokens, both built on hash-based commitments and HMAC, both efficient at the edge. The reason post-quantum primitives stay at the

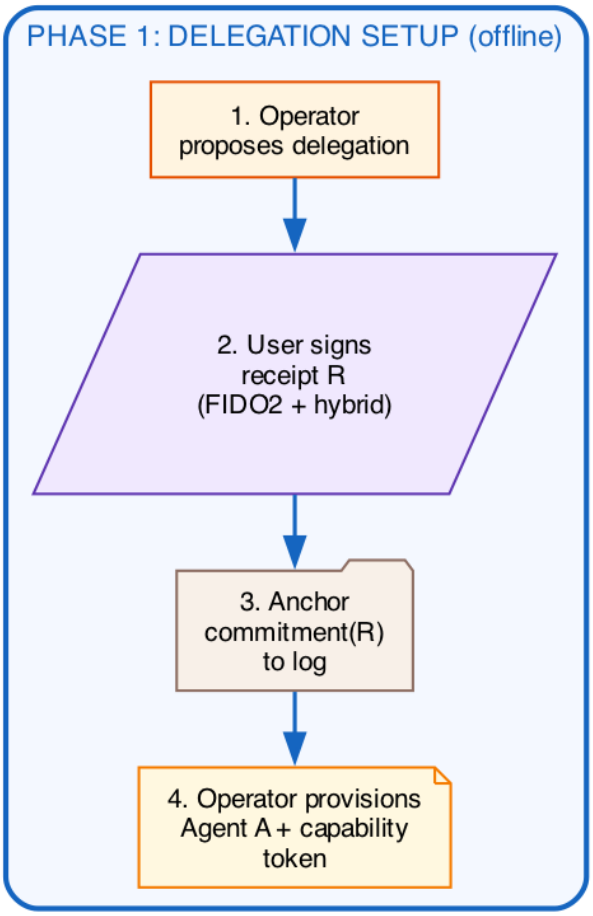
substrate and not inside the application layer is concrete: verifying lattice signatures inside a ZK circuit costs ten or thirty seconds and multi-megabyte proofs, non-deployable on resource-constrained hardware.

Seven guarantees

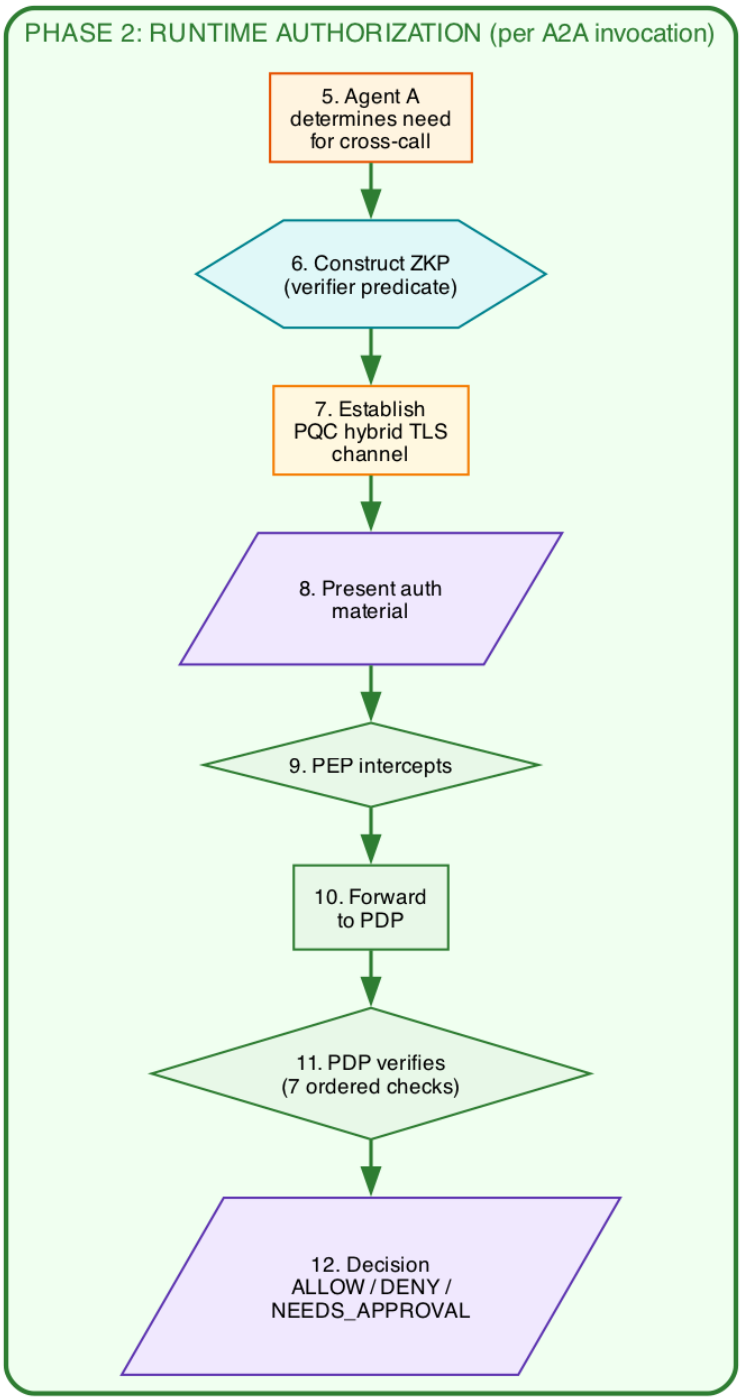
The architecture targets seven guarantees: **Soundness** (an ALLOW means the predicates actually hold); **Bounded disclosure** (the verifier learns ALLOW or DENY plus the minimum predicate truth needed to act, never the body); **Non-repudiation** (receipts and revocations anchored to the log are auditable forever); **Root-anchored attenuation** (sub-receipts prove subset-of-root, not subset-of-parent's-claim, blocking compromised-intermediate escalation); **Cascade revocation with bounded-staleness ALLOW** (parent revocation propagates within one log epoch; ALLOWs carry freshness proofs within a max-staleness window); **Post-quantum substrate** (long-lived signatures use PQC at server, log, and timestamp layers; the user-layer FIDO2 stays ECDSA today and migrates to hybrid when WebAuthn Level 3 ships); **Operator instruction commitment integrity, bounded** (the receipt's `instructionHash` is verified against a signed runtime manifest produced by the operator's control plane, not against the agent's own attestation; this catches operator drift but does not by itself catch runtime prompt injection through tool input).

A scenario, end to end

Alice is a compliance officer at Bank X. She authorizes her workday agent fleet to handle inbox triage for the next 24 hours. The triage agent (operated by AcmeAI Inc.) is provisioned to forward urgent CEO emails to her executive assistant agent; the executive assistant agent is provisioned to schedule on Alice's calendar in support of forwarded follow-ups. The boundary "never forward confidential-tagged" is declared.



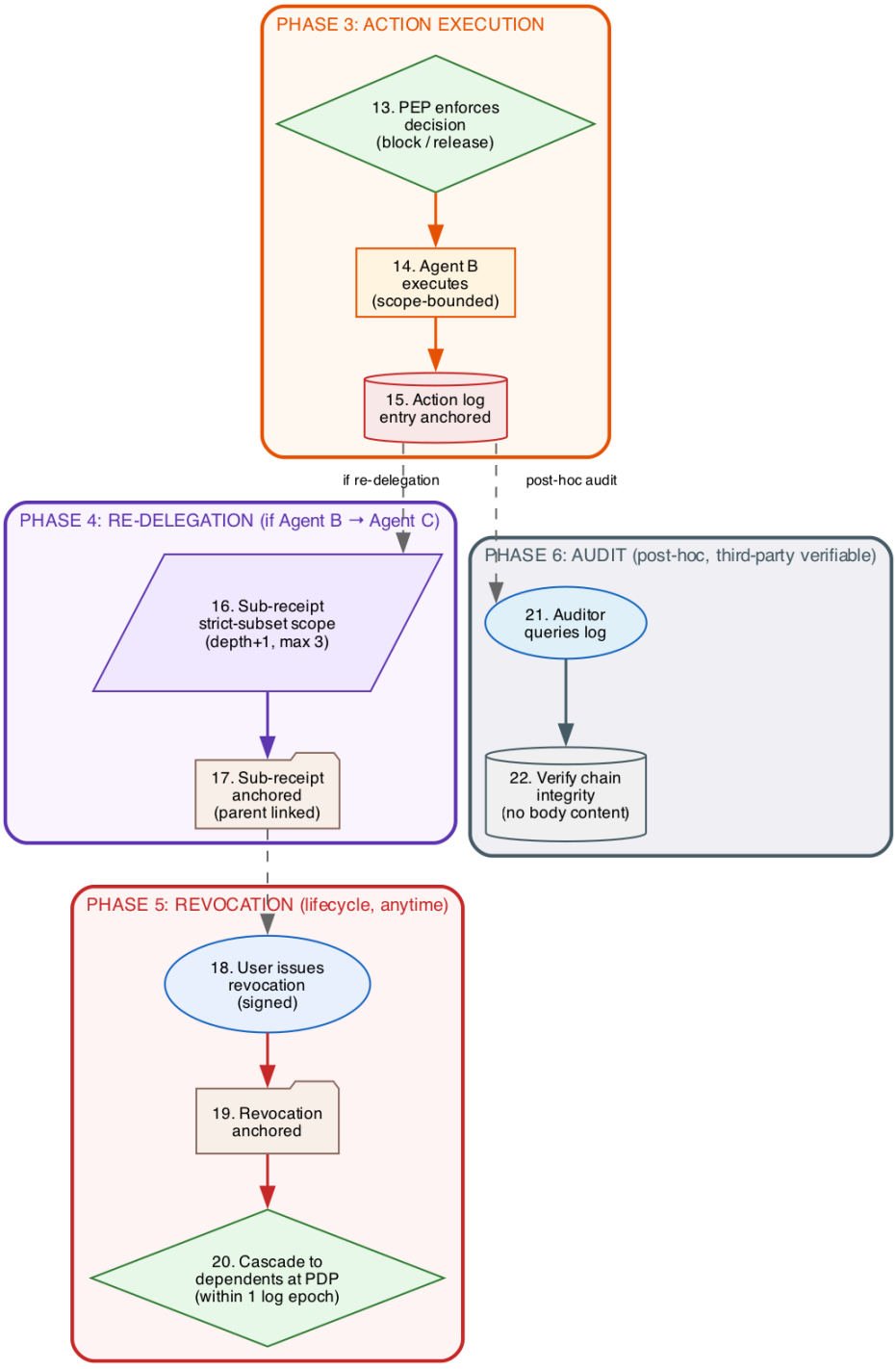
Phase 1, Setup. AcmeAI proposes the delegation in human-readable form. Alice reviews, signs the receipt with her FIDO2 hardware key, and the receipt's commitment hash anchors to the public transparency log with a signed authoritative timestamp. AcmeAI provisions the triage agent with a capability token bound to that commitment.



Phase 2, Runtime authorization. The CEO sends an urgent email. The triage agent classifies it and decides to call the executive assistant agent. The triage agent constructs a zero-knowledge proof over the verifier predicate. A post-quantum-protected TLS channel opens to the executive assistant's domain. The proof, the macaroon capability token, and the action descriptor arrive in the request body. The Policy Enforcement Point intercepts and forwards to the Policy Decision Point.

The Policy Decision Point runs seven ordered checks: revocation against the public log; signatures (server-anchor, user attestation, agent workload identity); time window against the log's authoritative timestamp; the zero-knowledge proof itself, verified in tens of milliseconds; capability-chain attenuation; operator instruction integrity (the receipt's `instructionHash` matched against the signed runtime manifest); and the receiving domain's local

policy. If all seven pass, the Policy Decision Point returns ALLOW signed with its hybrid key. The Enforcement Point releases the call.



Phases 3-6: execution, re-delegation, revocation, audit. The action executes within scope; an action log entry carrying only commitment hashes anchors to the public log. If the executive assistant re-delegates (say, to a calendar agent), it issues a sub-receipt whose scope is proven against Alice's *root* receipt, not against its own claim. Root-anchored attenuation is the property that prevents compromised intermediates from escalating beyond what the user originally authorized. Cascade revocation is enforced in one log epoch: when Alice revokes the root, dependents fail at the Policy Decision Point on the next freshness-proof boundary. Audit happens post-hoc and third-party: the auditor queries the public log, verifies chain integrity against a quorum of independent log

monitors with documented jurisdictional separation (mandatory Signed Tree Head cosigning per Sigsum-style discipline), and fails on monitor disagreement rather than majority-vote.

What the auditor sees, what stays sealed. The auditor sees: commitment hashes, signatures, timestamps, Policy Decision Point decision metadata (decision, reason class, latency), chain structure (parent-child relationships among receipts), boundary check results, revocation cascade outcomes. The auditor does NOT see: the email subjects, the email bodies, the recipients in plaintext, the operator instruction text, the full scope contents of any receipt. Body retrieval, if needed under legal compulsion or regulatory subpoena, requires a separate disclosure path through AcmeAI as the issuer; it does not happen through the audit log.

The verification of returned log entries is independent of AcmeAI and Bank X. It is third-party verifiable. Full audit completeness, proving that the queried set contains all relevant entries (not just that returned entries are integral), depends on the epoch-manifest mechanism discussed as an open question in the companion specification. The architecture produces evidence of compliance with the boundary, the time window, and the revocation discipline, without disclosing the customer business content the agents handled.

A detailed protocol walkthrough (exact receipt format, transparency-log entry numbers, signature scheme breakdowns, latency measurements on commodity hardware, Merkle proof depths, OPA Rego policy bindings) is in the AAC Construction Specification companion document.

Part III: The zero-knowledge construction

The receipt is split in two parts.

One part stays private at the issuer. It holds the things that should not leak: who is involved, what specifically is authorized, what limits apply, when the authorization runs.

The other part is public. It holds a cryptographic fingerprint of the private part and the signatures over that fingerprint. The user’s hardware key signs the fingerprint. A hybrid post-quantum signature on the server side anchors the same fingerprint to a public transparency log. Neither signature reads the private part. The verifier never reads the private part either.

The zero-knowledge proof takes the private part as a hidden input and outputs a single yes-or-no answer. It says yes when all of these conditions hold:

- The hidden content matches the public fingerprint.
- The proposed action falls inside what the user authorized.
- The current time is inside the user’s authorization window.
- The boundary clauses the user wrote are not violated by the action.
- The operator’s instruction commitment matches what the agent runtime actually received.
- The delegation chain narrows back to the user’s original sign-off.
- The receipt has not been revoked.

The verifier learns the answer. The verifier does not learn what the answer is over.

On a denial, the verifier returns ALLOW or DENY. Public-check failures may carry an explicit reason class (revoked, stale freshness proof, invalid signature, policy denied) because those checks operate on visible material. Failures inside sealed predicates default to a generic `SEALED_PREDICATE_FAILED` class unless the deployment opts into selective disclosure or an issuer-authorized debug mode. This preserves the disclosure asymmetry: denial does not leak which sealed predicate failed.

In this version, revocation non-membership is treated as a pluggable primitive. The companion specification discusses sparse Merkle maps, revocation accumulators, and transparency-map constructions as resolution paths.

The default proof system is **zk-STARK with masking polynomials** (Winterfell library). Hash-based, no trusted setup ceremony, conjectured to remain secure against future quantum-capable adversaries, proofs of 50-150 KB that verify in tens of milliseconds. SNARKs with universal trusted setup (PLONK, Marlin, Halo2) work as a fallback when bandwidth is the binding constraint. Two proof systems are excluded by design: Bulletproofs, whose security rests on a discrete-log assumption that breaks under quantum adversaries, and Groth16, whose per-circuit trusted setup is operationally fragile and does not survive a verifier predicate that evolves.

All commitments inside the proof are hash-based (Poseidon, Rescue, MiMC, or SHA-256 / BLAKE3 for conservative deployments). Pedersen commitments, the textbook efficient choice, are excluded for the same quantum-vulnerability reason as Bulletproofs.

The full specification (receipt format, step-by-step checks, proof-system parameters, primitive selection rationale, hybrid signature combiner) is in the AAC Construction Specification companion document.

Implementation

A reference implementation is being built. The component selection: Winterfell (zero-knowledge mode) for STARK, Open Policy Agent for the Policy Decision Point, Envoy for the Policy Enforcement Point, SPIRE for workload identity, Sigstore Rekor for the public log, rustls hybrid for the transport handshake, and `liboqs` / `pqcrypto` for ML-DSA and ML-KEM substrate primitives. Hybrid signature encoding follows `draft-ietf-lamps-pq-composite-sigs`. Proof sizes (100-200 KB) and verification latency (40-60ms for the full seven-check at the Policy Decision Point) are acceptable for control-plane authorization decisions and comparable to existing OAuth introspection plus policy evaluation. They are not acceptable for high-frequency data-plane authorization at sub-millisecond cadence; the deployment pattern there is to authorize once at session start, cache, and re-check on session boundary or on a sensitive action class. Deeper implementation considerations (proof-size budgets, freshness-binding parameters, deployment patterns by sector) are in the companion document.

Path forward

Open questions. Several gaps remain in this version of the architecture, including runtime prompt-injection bypass of static instruction integrity, the data structure underlying non-membership-in-revocation proofs, audit-log completeness, body custody under issuer failure, and cross-jurisdiction admissibility. Each is named in the AAC

Construction Specification companion document along with possible resolution paths. Input-boundary filtering remains a complementary control where AAC's static instruction integrity does not extend.

Standards engagement. Engagement with the IETF community is planned, with the WIMSE, OAuth, and SCITT working groups as candidate venues for hosting a future Internet-Draft. Cross-posting to arXiv (cs.CR) provides a citable preprint identifier. Working-group placement will follow community input as the specification matures with peer feedback. Reviewers from any of the candidate working groups are invited to engage directly.

Reference implementation. A reference implementation is being built, with the construction wired to OPA, Envoy, SPIRE, Rekor, and a demonstration UI. Release timing will be announced once the build is mature. Open architecture, Apache 2.0.

Successor papers in this series. Paper #4, *Delegation Without Escalation*, develops the capability-token attenuation discipline that this paper assumes. Paper #5, on auditing agents under NIS2, DORA, and the EU AI Act, will close the series. Both will publish on the same Tuesday cadence.

The construction this paper presents is not the only way to close the disclosure gap. Hardware Trusted Execution Environments solve a related problem at a different layer. Confidential computing platforms (Intel TDX, AMD SEV-SNP, ARM CCA) offer a different trust model. Differential-privacy-augmented audit logs are an alternative angle on the regulator-versus-customer-confidentiality tension. Each of these has its place. What zero-knowledge proofs offer that the alternatives do not is a verification model that does not require the verifier to trust any hardware vendor, any cloud operator, or any specific deployment platform. The verifier trusts a public circuit, public commitments, and independently checkable proofs.

Closing this disclosure gap is a prerequisite to deploying agents in regulated environments at the scale that 2027 will demand. The 2024-2025 designs that assume the verifier reads the body in the clear are not going to survive contact with the regulatory deployment of 2027. They were the right designs for the question being asked then. They are the wrong designs for the question being asked now.

Companion artifacts

- **AAC Construction Specification**: technical companion. Receipt format JSON schemas, predicate-to-circuit mapping, in-circuit verification steps, proof-system choice, hash-based commitments cryptanalysis, PQC hybrid signature combiner, detailed protocol walkthrough with measurement numbers, deep implementation considerations, the open questions with their resolution paths, and a survey of related work in privacy-preserving authorization across the IETF and IRTF ecosystem.
- **Agent Identity Platform**: reference implementation demonstrating per-agent identity (Paper #1), capability-token issuance and three-dimensional authorization (Paper #2). Paper #3 zero-knowledge construction integration in active build.
- **AgentTrustLab**: interactive simulator for agent-to-agent authentication, authorization, and communication under post-quantum and zero-knowledge constraints.

Field observations and peer review shape what comes next. Comments welcome.

About the author

Amin Hasbini is an AI and cybersecurity executive based in Paris. Former director of Kaspersky's Global Research & Analysis Team (GReAT) for the Middle East, Turkey, and Africa. Twelve years, seventy countries of threat coverage. Invited contributor to the French Senate's OPECST report on AI risks (2024). Former subject-matter expert on ICANN's second DNS Security, Stability, and Resiliency Review Team (SSR2, 2017-2019). Current focus: post-quantum cryptography maturity and AI agent security inside regulated enterprises. mahasbini.org.