NON-HUMAN IDENTITY SERIES · PAPER #3-SPEC

AAC Construction Specification

Technical companion to Paper #3: Authorization Without Disclosure

Mohamad Amin Hasbini

Independent researcher · Paris, France

Published May 05, 2026

ABSTRACT

Full construction details for the AAC architecture: receipt format, predicate-to-circuit mapping, in-circuit verification steps, proof-system choice, hash-based commitments, and PQC hybrid signature compatibility. Plus a survey of related work in privacy-preserving authorization across the IETF and IRTF ecosystem.

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). *AAC Construction Specification*: Technical companion to Paper #3: Authorization Without Disclosure. Non-Human Identity Series, Paper #3-Spec.
Available at https://mahasbini.org/papers/03-construction-specification/
PDF https://mahasbini.org/publications/papers/03-construction-specification.pdf

This document is the technical companion to Paper #3: Authorization Without Disclosure. It gives the construction details that Paper #3 describes at architectural level, plus a survey of related work in the privacy-preserving authorization space.

Readers who want the strategic narrative should read Paper #3 first. This companion is for cryptographers, IETF reviewers, and implementers building the construction.

Part A: Construction specification

Receipt format: sealed body and public envelope

Canonical serialization uses JCS (RFC 8785) or CBOR deterministic encoding (RFC 8949 §4.2.1).

Sealed Receipt Body (`R_body` , never disclosed to verifier):

```
{
  "issuer":                   user-public-key-fingerprint
  "operator":                 operator-identity
  "agent_identity":           agent-id
  "scope":                    { action_classes: [...], resources: [...] }
  "boundary_predicates":      [ <typed predicate>, <typed predicate>, ... ]
  "time_window":              { not_before, not_after }
  "instructionHash":          sha256(canonical(operator_instructions))
  "delegation_metadata":      { depth, parent_delegationId, max_depth }
  "issued_at":                TSA timestamp
}
```

Public Receipt Envelope (`R_env` , what the verifier sees):

```
{
  "delegationId":             sha256(canonical(R_body))
  "issuer_pubkey_ref":        user-public-key-reference
  "user_signature":           ECDSA-P256 over delegationId  (FIDO2 challenge = delegationId)
  "server_anchor_signature": ML-DSA-65 + ECDSA-P256 hybrid over (delegationId, log_metadata)
  "rekor_entry_id":           public log entry reference
  "tsa_timestamp":            authoritative time
  "inclusion_proof":          Merkle path to log root
}
```

Signatures are over `delegationId` (a hash), not over `canonical(R_body)` , so the PDP can verify them without seeing the body. This split also matches Nelson DRP's computation rule (compute `delegationId` over the body before adding the signature; signature fields are not part of the body).

Boundary predicate grammar

Boundaries are expressed as a typed predicate grammar over committed fields, evaluable in-circuit. Examples:

```
{ "type": "forbidden_classification",
  "field": "message.classification",
  "operator": "!=",
  "value_commitment": "H(confidential)" }

{ "type": "allowed_destination_domain",
  "field": "recipient.domain",
  "set_commitment": "merkle_root(internal_domains)" }
```

A human-readable boundary statement (e.g., `"never forward confidential-tagged"`) may accompany each typed predicate as documentation, but the circuit operates over the typed form only. Natural-language boundary text is non-evaluable in-circuit and would underspecify the predicate semantics.

Predicate-to-circuit mapping

Paper #3 Part I framed the verifier predicate as seven components for clarity of exposition. The deployed construction implements these as **seven in-circuit checks plus three PDP-side checks.**

The seven in-circuit checks are hash bindings (witness-to-commitment, instructionHash) and arithmetic / Merkle-membership constraints (scope, time, boundaries, root-anchored chain attenuation, revocation absence), all efficient over hash-based commitments.

The three PDP-side checks are signature verifications and local policy: - (a) user attestation signature on the `delegationId` (ECDSA-P256, FIDO2-produced) - (b) server-side hybrid log-anchor signature (ML-DSA-65 + ECDSA-P256) on the public log entry - (c) local domain policy (Paper #3 Part I predicate #7), which depends on receiver-side state the prover does not control

The architectural reason signature verification stays at the PDP is that ECDSA-P256 over a non-native field is non-circuit-friendly, and lattice signatures (ML-DSA, Falcon) inside ZK circuits cost tens of seconds and multi-megabyte proofs. The circuit verifies what is cheap to verify under hash collision resistance; the PDP verifies what is cheap to verify under classical or PQ signature schemes.

In-circuit verification steps

The circuit performs the following verification steps internally:

1. **Recompute `delegationId`** from witness body using SHA-256, check it equals the public `delegationId`. This binds the witness to the publicly anchored commitment.
2. **Recompute `instructionHash`** from witness instruction text, check it equals the receipt body's `instructionHash` field. This binds the operator's instruction commitment.
3. **Check the action descriptor hash** is consistent with an action that falls inside the receipt's scope. The scope check is a Merkle-tree membership proof: scope contents are committed as a Merkle root in the receipt, and the action descriptor's category proves membership in the scope tree.
4. **Check time_window contains the public current time.** Two arithmetic comparisons against the TSA-provided log timestamp.

5. **Check boundaries are not violated by the proposed action.** Boundaries are typed predicates (forbidden destinations, forbidden classifications). Each boundary type is a separate circuit constraint.

6. **Check chain attenuation against root, if depth > 0.** For a sub-receipt, the witness includes the **full parent commitment chain back to the user's root receipt**, not only the immediate parent. The circuit verifies each parent-link's `delegationId` matches its public-input commitment, and that this receipt's scope is a strict subset of the **root receipt's scope** (the strict-subset check is a Merkle-path comparison over the root's scope tree, with each intermediate scope serving as an additional containment constraint). This ordering closes the confused-deputy attack in which a compromised intermediate agent issues sub-receipts that look subset-of-its-own-claim while exceeding the user's actual root authority.

7. **Check revocation absence** by verifying the public input "proof of non-membership in revocation log entries up to current time."

The output is a single boolean: all seven in-circuit checks passed. The proof is succinct (50-150 KB for zk-STARK with masking; sub-kilobyte for SNARK), succinctly verifiable (tens of milliseconds), and zero-knowledge under the proof system's security assumptions: computational ZK under hash collision resistance and the random-oracle assumption for STARK, or under universal-setup soundness for PLONK / Marlin / Halo2 SNARK alternatives.

Choice of proof system

zk-STARKs (zero-knowledge variant of Scalable Transparent ARguments of Knowledge). Hash-based, no trusted setup, post-quantum-conjectured. Stock FRI is an argument-of-knowledge but not zero-knowledge by default; AAC requires the zero-knowledge construction (Winterfell's ZK mode, which adds masking polynomials to the trace; alternatively Polaris or RedShift constructions). Proof size in the 50-150 KB range with operational FRI parameters (Winterfell defaults: blowup factor 8, fold factor 4, with grinding and query count tuned for ~100-bit conjectured security per the FRI security analysis; specific parameters are deployment-tuned and documented in the implementation). Verification in tens of milliseconds. The default for AAC. Reference library: Winterfell (Microsoft / Meta, MIT license), used in zero-knowledge mode.

Universal-setup SNARKs (PLONK, Marlin, Halo2). Compromise: trusted setup is universal (one ceremony works for all circuits up to a size bound), proofs are larger than Groth16 but smaller than STARK, no per-circuit ceremony fragility. Acceptable as an alternative to STARK when bandwidth is the binding constraint.

Groth16 SNARKs. Smaller proofs (under 200 bytes) and faster verification, but require per-circuit trusted setup. The trusted setup is operationally fragile in production: a single ceremony failure compromises the soundness of every proof on that circuit forever. Acceptable as an escape hatch for circuits that are stable and long-lived, with the trusted setup conducted as a multi-party ceremony. Forbidden for AAC's default deployment because the verifier predicate is expected to evolve as the architecture matures.

Bulletproofs. No trusted setup, range-proof-optimized, but soundness rests on the discrete-logarithm assumption, which is broken by a quantum adversary running Shor's algorithm. Forbidden for AAC's default deployment, because soundness loss under future cryptographically-relevant quantum computers is exactly the failure mode the substrate is designed to prevent.

Hash-based commitments inside the circuit

The receipt's scope, instruction text, and capability chain are committed using **hash-based commitments** (Poseidon, Rescue, or MiMC, all designed for SNARK/STARK efficiency). Hash-based commitments are binding under hash collision resistance. A quantum adversary attacks collision via the Brassard-Høyer-Tapp algorithm with a cube-root speedup (not Grover's square-root, which applies only to preimage), yielding approximately 85-bit collision security on 256-bit hash outputs against a quantum adversary. Acceptable for the architecture; a deployment consideration when sizing hash output lengths.

Pedersen commitments, the textbook efficient primitive for circuit-friendly hiding, are explicitly **not** used. Pedersen binding rests on discrete-logarithm hardness, which is broken by a quantum adversary. A circuit that targets post-quantum security must commit with hash-based primitives, full stop.

A note on cryptanalysis maturity: arithmetization-friendly hashes (Poseidon, Rescue, MiMC) have less third-party cryptanalysis than SHA-256, and recent results (2023-2025) have weakened parameter sets in several arithmetic hash families. AAC implementations should track current arithmetization-friendly hash analysis and adjust round counts when new attacks emerge. Conservative deployments may prefer SHA-256 or BLAKE3 for outer-layer commitments accepting the higher in-circuit cost, with Poseidon-class hashes used only for inner Merkle constructions where SNARK/STARK efficiency dominates security margin.

PQC compatibility and hybrid signatures

The user's signature on the receipt is ECDSA-P256, today, because that is what FIDO2 hardware authenticators support. A quantum-capable adversary, when one exists, can forge ECDSA signatures over harvested artifacts. This is a residual risk that the architecture acknowledges directly.

Three responses bound the residual risk:

1. The receipt body is anchored to the public log with a hybrid (ML-DSA-65 + ECDSA-P256) server-side signature using a published combiner construction (the architecture follows draft-ietf-lamps-pq-composite-sigs for hybrid signature encoding, with both component signatures required for verification; neither alone is sufficient, defending against partial-component attacks per Bindel-Brendel-Fischlin separation results). The server-side hybrid signature is what verifies the existence and timestamp of the receipt; it is post-quantum-safe. The user's ECDSA signature is what authenticates the user's intent. A future quantum adversary forging the user's ECDSA signature would still face the unforgeable hybrid log anchor as evidence of the original receipt.

2. FIDO2 hardware authenticators are expected to add ML-DSA support over the next two to three years, when the WebAuthn Level 3 specification and authenticator vendors converge. The migration path is documented; the artifacts under that path will be upgraded to hybrid user signatures once available.

3. Applications with confidentiality requirements that extend beyond the FIDO2 migration window (defense, classified-adjacent, long-lifecycle critical infrastructure) can supplement today with parallel ML-DSA software-key signatures bound to the same `delegationId` and submitted alongside the FIDO2 ECDSA attestation. This is operationally heavier (two signing artifacts per receipt) and does not preserve FIDO2's single-device user-presence guarantee, so it is a defense-in-depth layer rather than a substitute. The ML-DSA software signature does not establish user-presence; its sole purpose is post-quantum durability of the user-intent artifact, assuming concurrent FIDO2 user presence at issuance is what authorized the signing event. The FIDO2 ECDSA attestation remains the authoritative user-presence signal until WebAuthn L3 + ML-DSA authenticator support ships.

The construction does not rely on the user's ECDSA signature for the zero-knowledge property. Zero-knowledge is achieved by the proof system over hash-based commitments. The user's signature provides the authenticity of intent; the substrate provides the post-quantum durability of the artifact's existence and timestamp.

Part B: Related work in privacy-preserving authorization

This part surveys the IETF / IRTF work adjacent to AAC as of May 2026, organized by the architectural layer at which each operates. The survey is intended for peer reviewers tracking the privacy-preserving authorization space; it is not exhaustive.

Disclosure-based prior art (the dominant pattern)

- **OAuth 2.0 Token Exchange** (RFC 8693). The canonical example: the verifier reads the token's claims to make the authorization decision.
- **Agent-delegation drafts emerging in IETF (2025-2026)**, all building on OAuth-family primitives:
- `draft-klrc-aiagent-auth-01` : composes OAuth Token Exchange and Rich Authorization Requests for AI agents (Brian Campbell et al.)
- `draft-singla-agent-identity-protocol-00` : Agent Identity Protocol with delegation depth counter
- `draft-goswami-agentic-jwt-00` : Secure Intent Protocol, JWT-compatible agentic identity
- `draft-ietf-oauth-transaction-tokens` : composite identity propagation through call chains
- **WIMSE workload-identity drafts**: `draft-ietf-wimse-arch-07` (architecture), `draft-ietf-wimse-identifier-02` , `draft-ietf-wimse-mutual-tls-00` , `draft-ietf-wimse-workload-creds-00` , `draft-ietf-wimse-workload-identity-practices-04` , `draft-ietf-wimse-http-signature-03` , `draft-ietf-wimse-wpt-01` (workload proof token). All assume disclosure-based verification at the workload-attestation layer.

Credential-claim selective disclosure

These mechanisms operate at the credential level. A holder reveals a chosen subset of claims to a verifier; the verifier still reads what is disclosed, just less than the full credential.

- **SD-JWT** (RFC 9901, November 2025): SHA-256 hashed claims with selective revelation. Standards Track.
- **SD-JWT-VC**: SD-JWT extended for verifiable credentials.
- **SD Agent** (`draft-nandakumar-agent-sd-jwt-02`): SD-JWT applied to Agent Card metadata for privacy-preserving agent discovery.
- **W3C Verifiable Credentials Data Model** with selective-disclosure profiles.

Zero-knowledge selective disclosure of credential claims

- **BBS+ signatures** (`draft-irtf-cfrg-bbs-signatures-10` , IRTF CFRG, January 2026): a holder can prove zero-knowledge statements about individual signed messages without revealing the messages or the

signature. Constant-size signatures cover multiple messages; presentations are unlinkable. Operates at the signed-message level.

Hardware-attested zero-knowledge agent identity

- **PTV (Prove-Transform-Verify)** (`draft-anandakrishnan-ptv-attested-agent-identity-00`): Groth16 zk-SNARKs anchored in TPM 2.0 hardware roots. Proves *task execution correctness on an authorized model*. Per the draft, approximate operating points are ~187ms proof generation, sub-300-byte proofs, sub-5ms verification. Uses HotStuff BFT consensus for immutable audit logs and sovereign-bound metadata for jurisdiction-aware attestation.

PTV is the closest IETF work to AAC by intent ("cross-domain agent authorization without raw data exposure") but operates at a different layer: - **PTV's ZKP proves**: the agent executed a specific task correctly on attested hardware (execution-correctness attestation). - **AAC's ZKP proves**: the delegation receipt's authorization predicate (revocation status, scope coverage, time validity, chain attenuation, instruction integrity, boundary respect, local policy) holds over a sealed receipt body (authorization-decision verification). - **PTV requires** TPM 2.0 hardware attestation; **AAC is software-deployable** without TEE dependency. - **PTV uses Groth16** with per-circuit trusted setup; **AAC uses zk-STARK** with masking polynomials and transparent setup.

The two are complementary by composition: PTV proves the agent ran the task correctly on the right hardware; AAC proves the agent had permission to run the task at all. A deployment combining both would have stronger end-to-end properties than either alone.

Delegation-chain attenuation drafts

- **Attenuating Authorization Tokens for Agentic Delegation Chains** (`draft-niyikiza-oauth-attenuating-agent-tokens-00`, March 2026, Niki Aimable / Tenuo): JWT-based credential system that enables offline derivation of more restrictive tokens, cryptographically enforced attenuation (a derived token cannot grant broader authority than its parent), and offline chain verification (any enforcement point holding the trust anchor verifies the complete delegation chain without network calls). Uses signed JWT attenuation, not ZKPs; tokens are fully disclosing (constraint details appear plaintext in JWT payloads).

This draft and AAC overlap on the *delegation-chain attenuation* problem but differ on the verification model. Niyikiza specifies offline cryptographically-enforced attenuation in plaintext-disclosing tokens. AAC's contribution is the same attenuation guarantee combined with non-disclosing verification of the full authorization predicate over a sealed receipt body.

Receipt-protocol prior art

- **Delegation Receipt Protocol (DRP)** (`draft-nelson-agent-delegation-receipts-04`, April 2026): receipt-based agent delegation protocol. Receipts include `operatorInstructions` as a plaintext instruction string with `instructionHash` recorded for drift detection. The verifier procedure is disclosure-based: it directly validates the receipt's signature, time window, scope, boundary clauses, and instruction-hash comparison against the receipt body's committed values.

DRP is the closest receipt-based work to AAC. It is not endorsed by the IETF and has no formal standing; it is an individual Internet-Draft. Its contribution is delegation integrity through signed-receipt structure with verifier-side checks against receipt fields. AAC adds non-disclosing verification of that integrity: the same predicate set DRP

checks (signature, time, scope, boundaries, instruction hash) is checked in zero-knowledge against a sealed receipt body, with multi-hop root-anchored attenuation in-circuit. AAC is not a rejection of DRP; it is the privacy-preserving authorization-decision layer DRP leaves open.

AAC's distinct contribution

AAC's ZKP construction targets the *composed multi-condition authorization decision* over a delegation receipt body the verifier never sees in any form. Operates at the authorization-decision layer (not credential or execution). Software-deployable without TEE dependency. Transparent-setup zk-STARK over hash-based commitments. Targeted specifically at multi-hop agent-to-agent delegation flows with root-anchored attenuation.

Other adjacent work

- **OAUTH WG mailing list discussion** of "ZTNP / ZTIP, attestation-gated authorization and intent-bound delegation" (May 2026). Active discussion thread; no formal draft published as of this paper's date.

This survey will be updated as the space evolves.

Part C: Detailed protocol walkthrough

This section walks the worked example from Paper #3 Part II with full implementation specifics: signature scheme breakdowns, transparency-log entry numbers, latency measurements, Merkle proof depths, and OPA Rego policy bindings.

Scenario. Alice is a compliance officer at Bank X. She authorizes her workday agent fleet to handle inbox triage and follow-up coordination for the next 24 hours: the email triage agent (`alice-triage-agent-v3.2`, operated by AcmeAI Inc.) is provisioned to forward urgent CEO emails to her executive assistant agent (`alice-exec-asst-agent-v1.7`); the executive assistant agent is provisioned to schedule on Alice's calendar in support of forwarded follow-ups. The authorization runs from 16:00 to 16:00 the next day. The root scope is `{inbox:read, forward:urgent-tag, calendar:write on Alice's calendar}`. The boundary "never forward confidential-tagged" is declared. The operator instructions are "forward CEO urgent emails to alice-exec-asst-agent only; calendar scheduling permitted on Alice's calendar in support of forwarded follow-ups." Alice signs the root receipt at 16:00:00Z with her YubiKey (ECDSA-P256). The receipt's `delegationId` is the SHA-256 hash of the canonical body. AcmeAI's server submits the `delegationId` to Sigstore Rekor with a hybrid (ML-DSA-65 + ECDSA-P256) server anchor signature; the entry is recorded as Rekor entry #4827193 with TSA timestamp 16:00:00.347Z and inclusion proof at Merkle depth 22.

The runtime call. At 16:42:00, Alice's CEO sends an urgent email about Q1 numbers. The triage agent classifies it as urgent. The agent constructs a zero-knowledge proof: STARK, over the verifier predicate, witness containing the receipt body, public inputs containing the action descriptor hash (representing `forward:urgent-tag` to `alice-exec-asst-agent`), the current time (Rekor TSA 16:42:08Z), the `delegationId`, the inclusion proof. Indicative proving time on commodity hardware: ~0.81 seconds. Indicative proof size: ~142 KB. (Numbers are illustrative for the worked-example circuit; production tuning will produce different operating points depending on Winterfell version, AIR construction, and hardware.) The agent opens a TLS 1.3 hybrid channel (ML-KEM-768 + X25519) to the executive assistant agent's domain. It presents the proof, the macaroon capability

token (HMAC-SHA512 with caveats `time≤24h, agent=alice-triage-agent, action-classes={inbox:read, forward:urgent-tag}`), and the action descriptor in the request body. The Envoy Policy Enforcement Point at the executive assistant agent's domain intercepts and forwards to OPA Policy Decision Point.

The seven checks. The Policy Decision Point runs the seven ordered verification checks. (1) Revocation: query Rekor; revocation status is clean. (2) Signatures: verify the hybrid server-anchor on the receipt commitment, the ECDSA-P256 user attestation, the SPIFFE workload identity attestation on the agent. All pass. (3) Time window: 16:42:08Z is inside [16:00:00Z, next day 16:00:00Z]. Pass. (4) Zero-knowledge proof verification: STARK verifier (sidecar process called from OPA via gRPC) verifies the proof in ~50ms. Pass. (5) Capability chain: macaroon caveats are checked; current time within 24-hour window; agent identity matches; action class within allowed set. Pass. (6) Operator instruction integrity: the `instructionHash` baked into the receipt is matched against the hash of the instructions actually delivered to the agent runtime. Pass. (7) Local Rego policy at executive assistant domain: rule "recipient must be internal-domain allowlist" allows. Pass.

The decision. OPA returns ALLOW, signed with hybrid Policy Decision Point key, reason "all predicates satisfied; recipient internal," latency 47ms. Envoy releases the call. The triage agent forwards the email. An action log entry (commitment hashes only, no body) is anchored to Rekor as entry #4827194.

The sub-receipt. The executive assistant agent decides to schedule a follow-up meeting on Alice's calendar. It issues a sub-receipt to a calendar agent, with root-anchored strict-subset scope: `calendar:write on Alice's calendar` is a strict subset of the root receipt's scope, time-bounded further to the next two hours (a strict subset of the root's 24-hour window), depth 1 of maximum 3. The sub-receipt's circuit takes the parent commitment chain back to root as a public input and proves the strict-subset relation against the root scope, not against the immediate parent's claim. The sub-receipt is anchored as Rekor entry #4827195 with parent reference to #4827193. The calendar agent performs `calendar:write` after its own runtime authorization round; logged as entry #4827196.

Revocation. At 18:00:00Z, Alice revokes the original receipt. She signs the revocation with the same FIDO2 ECDSA-P256 key. AcmeAI submits to Rekor as entry #4827296. Cascade propagates: at the Policy Decision Point, the executive assistant's sub-receipt #4827195 is marked cascade-revoked. Subsequent attempts by the executive assistant agent to use the sub-receipt are denied at the Policy Decision Point with reason `REVOKED, parent=#4827193`. Cascade completes within 5 seconds, one log epoch.

The audit. The next morning, Bank X's compliance auditor queries Rekor for entries matching `agent_identity_pattern: alice-*-agent` over the time range [16:00:00Z, 19:00:00Z]. Six entries return: the receipt commit, two action commits (the two urgent forwards Alice's triage agent performed during the day), the sub-receipt commit, the calendar action commit, and the revocation commit. The auditor verifies chain consistency against a cosigned Signed Tree Head bundle from seven independent Rekor monitors with documented jurisdictional separation (United States, European Union, Asia-Pacific, plus three additional non-aligned witnesses); auditor fails on any STH disagreement rather than majority-vote. The auditor verifies all six inclusion proofs (Merkle depth 22). The auditor verifies all signatures: hybrid server anchors (ML-DSA-65 + ECDSA-P256), user ECDSA-P256 attestations on receipt and revocation, agent SPIFFE workload identities on action entries, and **hybrid Policy Decision Point signatures (ML-DSA-65 + ECDSA-P256) on every ALLOW/DENY decision**. The auditor verifies that boundary `never forward confidential-tagged` had zero violations during the window (no `BOUNDARY_VIOLATION` failure-class entries in the log). The auditor verifies that operator instruction commitment integrity held at every Policy Decision Point evaluation (no

`INSTRUCTION_MISMATCH`); runtime-injection integrity is out of scope for this guarantee class. The auditor verifies that the cascade revocation propagated correctly (the executive assistant's sub-receipt was successfully blocked after 18:00:00Z within the deployment-defined max-staleness window).

What the auditor sees, what stays sealed. The auditor sees: all six commitment hashes, all signatures, all timestamps, all PDP decision metadata (decision, reason class, latency), the chain structure (parent-child relationships among receipts), the boundary check results (zero violations), the revocation cascade results. 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, regulatory subpoena, or court order), requires a separate disclosure path through AcmeAI as the issuer; it does not happen through the audit log.

The audit is independent. The auditor does not depend on AcmeAI or Bank X to perform the verification. The audit is third-party verifiable. The audit produces evidence of compliance with the boundary, the time window, and the revocation discipline, without disclosing the customer business content the agents handled.

Part D: Deep implementation considerations

A reference implementation is being built; release timing will be announced once the build is mature. The component selection: Winterfell (in 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, and rustls hybrid for the transport handshake. Components in the substrate layer use `liboqs` and `pqcrypto` for ML-DSA and ML-KEM primitives.

Three operational considerations matter most.

Proof size on the data path. STARK proofs at 100-200 KB are acceptable for control-plane authorization decisions (one proof per agent invocation, not one per message inside the invocation). They are not acceptable for high-frequency data-plane use cases where authorization would be checked per request at sub-millisecond cadence. For high-frequency cases, the architecture pattern is to authorize once at the start of a session-bounded interaction, cache the authorization for a short interval, and re-check on session boundary or sensitive action class.

Verification latency at the Policy Decision Point. Verifying a STARK proof at the Policy Decision Point takes tens of milliseconds. The full seven-check verification (revocation query, signatures, time, ZKP, chain, instruction integrity, local policy) runs in 40-60ms on commodity hardware in the worked example above. This is comparable to existing OAuth introspection plus policy evaluation.

Public-input freshness binding. The proof's public inputs include the current time, derived from the log's TSA timestamp. The Policy Decision Point must reject proofs whose public-input time is older than a max-staleness window (e.g., 30 seconds). This binds the proof to a recent authoritative time and prevents replay across log epochs. Defining the max-staleness window per deployment is a deployment parameter, not a fixed architectural choice.

Part E: Resolution paths for the open problems

This part details the resolution paths for each open problem named in Paper #3. The architecture was examined under five distinct adversarial perspectives: cryptographic, distributed-systems, identity / delegation, audit / compliance, and red-team. Issues that the review surfaced and that have already been incorporated into the architecture as published (root-anchored sub-receipt attenuation, recovery-anchor revocation, witness-count floor with Signed Tree Head cosigning, hybrid Policy Decision Point signatures, hash-based commitments inside the circuit, zk-STARK with masking polynomials) are not detailed here. The paths below describe how each remaining problem can be closed; reviewer input on each is welcomed.

Runtime prompt-injection bypasses static instruction integrity (Guarantee #7). The `instructionHash` predicate commits the operator-typed system prompt to the receipt at issuance. A runtime prompt-injection attack that enters the agent through tool input (an email body, a document, a web page the agent reads) does not change the operator's typed prompt and therefore does not change the `instructionHash`. The proof still verifies; the agent may follow the injected reasoning and produce an action that the user did not authorize. Guarantee #7 is explicitly bounded to static prompt integrity in this paper; it does not extend to runtime reasoning trace integrity. A complete closure would require either an executed-action commitment (the agent signs over its actual reasoning trace and tool-call vector at action time, bound to the action descriptor in-circuit) or a Trusted Execution Environment attestation of the agent's runtime context. Both are out of scope for this paper. AAC composes cleanly with prompt-injection filtering at the input boundary (Lakera, NeMo Guardrails, classifier-based filters); the architecture does not replace those controls.

Non-revocation proof structure is underspecified. The construction states that the circuit verifies "proof of non-membership in revocation log entries up to current time" as a public input. An append-only transparency log natively provides inclusion proofs and consistency proofs, not efficient non-membership proofs over the revocation set. Closing this would require defining the data structure carrying revocation status: a sparse Merkle map keyed by `delegationId`, an epoch-indexed revocation accumulator, or a transparency-map (Trillian Map-style) construction with epoch roots cosigned by witnesses and anchored to the log. The prover would supply a non-membership proof against the latest accepted epoch root; the PDP would reject proofs older than a max-staleness window Δt. This paper treats the non-membership proof as a primitive; specification of the underlying revocation index is left open.

Audit log inclusion proves integrity of seen entries, not completeness. The auditor verifies that the entries returned by a query are integral and consistent with the log, but cannot prove that the query returned all relevant entries for the audit window. To prove "zero boundary violations" rather than "no boundary violations were logged," the architecture would need a completeness mechanism: per-session monotonic action sequence numbers, gap-free sequence proofs, and epoch manifests committing to the complete ordered set of action / denial / revocation events for each session. The auditor would verify inclusion AND gap-free completeness against the epoch manifest before drawing inferences about boundary conformance. The underlying transparency-log primitives are unchanged.

Body custody at the issuer is a single point of failure. Receipt bodies stay sealed at the issuer. The architecture does not yet specify what happens when the issuer goes bankrupt, suffers a ransomware event that destroys the bodies, or is compelled to delete bodies under GDPR Article 17. The path forward is a threshold-secret-shared body escrow with k-of-n trustees, or HSM-anchored issuer storage with attested backup. The GDPR-erasure path under Article 17(3)(b)/(e) requires a pre-declared statutory basis (DORA Article 12 retention) and key-shredding mechanics for cryptographic erasure.

Boundary clauses can point to mutable external state. A boundary like "recipient must be internal-domain allowlist" is enforced by the Policy Decision Point comparing destination against an allowlist. If the allowlist is mutable state outside the receipt (DNS records, an external configuration source), an attacker who controls that state can change the receipt's effective meaning without re-signing. The path forward is to either commit the allowlist content into the receipt (content-addressed) or anchor the allowlist itself as a signed log entry with its own commitment chain, so that any change to the allowlist is itself auditable. The cryptographic primitives are unchanged; this is a deployment-pattern requirement.

Hash-of-personally-identifiable-information under GDPR. Per EDPB Opinion 28/2024 and CJEU Breyer / SRB, pseudonymous personal data committed to an immutable transparency log is GDPR Article 17 personal data. Append-only commitments anchored to the public log mean that the receipt's hash, while not the receipt's body, is itself personal data when the receipt's principals can be identified by other means. Cryptographic erasure (per-receipt key destruction so the body becomes unrecoverable even with the hash retained) plus pre-declared statutory retention basis under DORA Article 12 are a path forward.

Custodian of record and cross-jurisdiction admissibility. FRE 901/902 (United States) and Code de procédure civile Article 1366-1369 (France) require an identifiable custodian of record to attest to log integrity and the systems that produced the entries. Sigstore Rekor's operator (Linux Foundation today) is not contractually a custodian for any specific tenant's evidence. Schrems II / EDPB transfer-tools guidance bars unrestricted transfer of EU personal data to a US-hosted Rekor instance. Sovereign Rekor deployment per regulated region, and a named records-custodian role with eIDAS-qualified TSP attestation, are the path forward.

Several additional residual gaps. PDP latency side-channel allowing predicate-failure-class inference (constant-time PDP execution required for high-assurance deployments); metadata correlation across sealed bodies in repeated `destination_hash` patterns (commitment salting per epoch as a closure path); replay across log epochs requiring binding the proof's public input to a recent Signed Tree Head (already handled by the freshness window in Part D); DORA Article 17 four-hour incident telemetry needing regulator-side disclosure-key escrow (key-recovery ceremony left open); AI Act Article 12 input/output capture compatibility (depends on classification of agent-as-AI-system and is being tracked through ongoing AI Act implementing-act consultations).

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.