

Cryptographically Verifiable Actor Chains

A Visual Guide to SPICE-ACTOR-CHAINS and OAuth 2.0 Token Exchange

DRAFT Co-Authors:

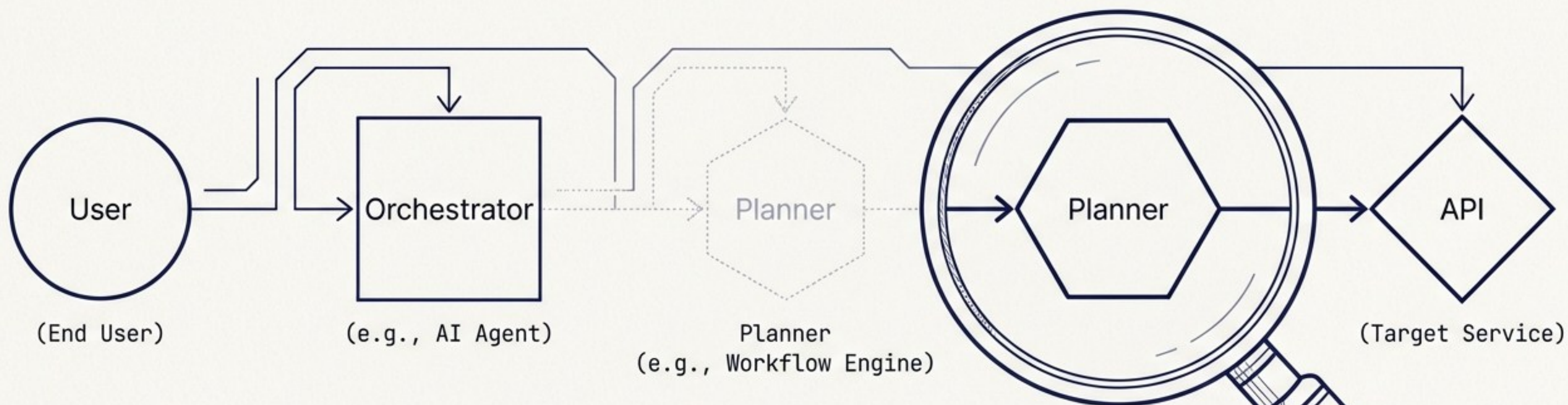
A. Prasad

Ramki Krishnan

Diego Lopez

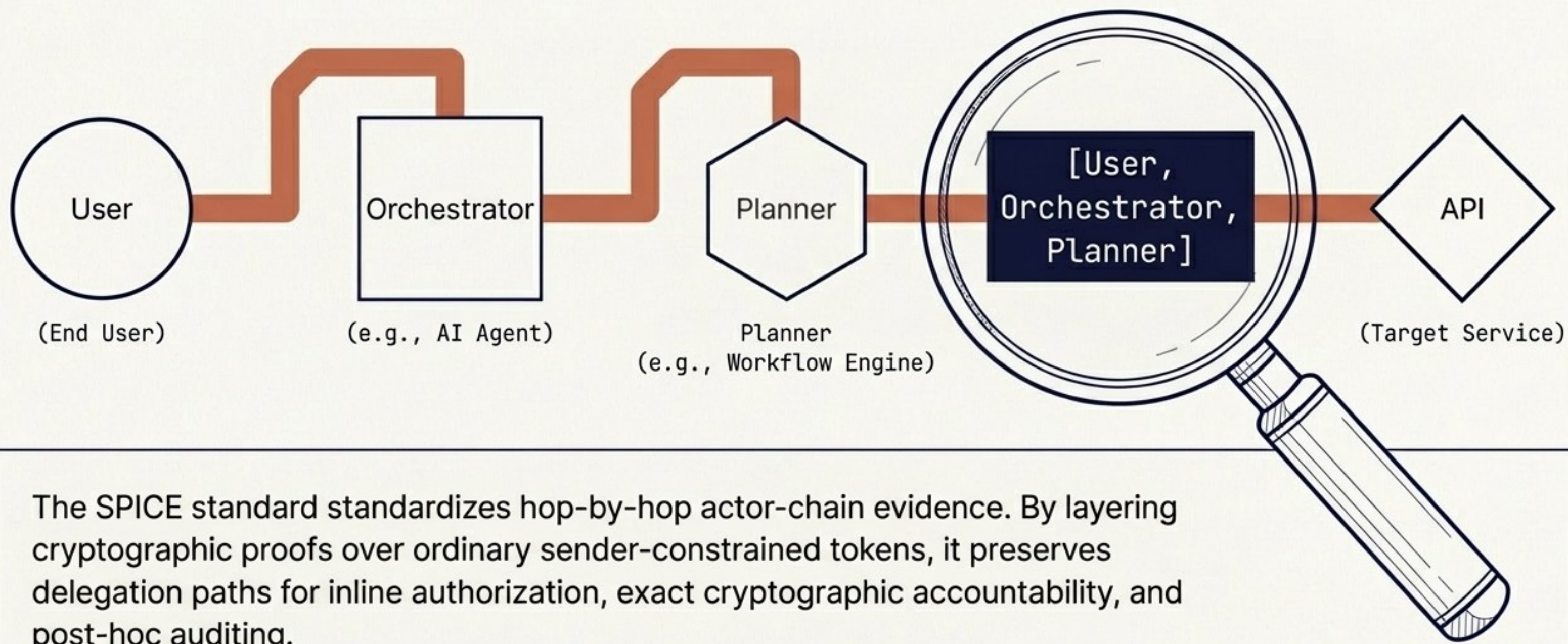
Srinivas Addepalli

The Lost Context in Multi-Hop Delegations



Current OAuth 2.0 token exchange (RFC 8693) allows nested act claims, but prior actors remain informational only. There is no standardized way to preserve or cryptographically validate the continuous path of delegation. Downstream APIs are blind to the true origin, and auditors cannot mathematically prove the path.

Restoring Provenance with SPICE Actor Chains



The SPICE standard standardizes hop-by-hop actor-chain evidence. By layering cryptographic proofs over ordinary sender-constrained tokens, it preserves delegation paths for inline authorization, exact cryptographic accountability, and post-hoc auditing.

Anatomy of an Actor-Chain Token

```
{
  "iss": "https://as.example",
  "sub": "svc:planner",
  "sid": "6cb5f0c14ab84718a69d96d31d95f3c4",
  "achp": "committed-chain-full",
  "ach": [
    {"iss": "as.example", "sub": "svc:orchestrator"},
    {"iss": "as.example", "sub": "svc:planner"}
  ],
  "achc": "BASE64URL(protected-header).BASE64URL(payload).BASE64URL(signature)"
}
```




sid (Workflow Identifier): Generated at bootstrap using 122+ bits of entropy. Remains entirely stable across the entire workflow chain.

achp (Selected Profile): The specific actor-chain profile governing the token. Immutable per workflow.

ach (Readable Actor Chain): An ordered JSON array of **ActorID** objects (Issuer + Subject) visible to the downstream recipient.

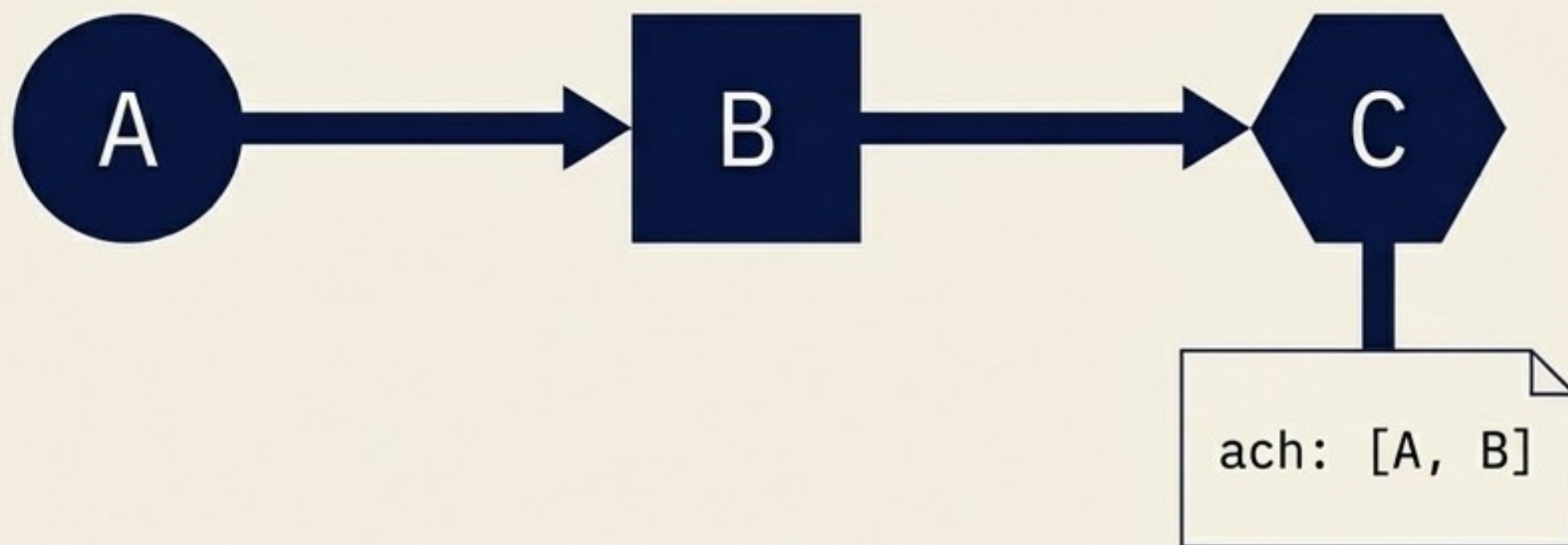
achc (Committed Chain State): A compact **JWS** string containing the cumulative cryptographic proof of the entire historical chain.

The 5 Profiles: A Strategic Menu

Profile	Trust Anchor	Downstream Visibility (ach)	Cryptographic Proof (achc)
Asserted Full Disclosure	AS-Asserted Continuity	Full readable chain	None
Asserted Subset Disclosure	AS Disclosure Policy	Disclosed subset only	None
Committed Subset Disclosure	Actor Proofs + AS Commitment	Disclosed subset only	Yes (Full hidden chain bound to proof) 
Committed Full Disclosure	Actor Proofs + AS Commitment	Full readable chain	Yes (Full readable chain bound to proof) 
Committed No Chain Disclosure	Actor Proofs + AS Commitment	None (Presenting actor only)	Yes (Opaque inline, reconstructable in audit) 

APPENDIX – DETAILS ON PROFILE FLOWS

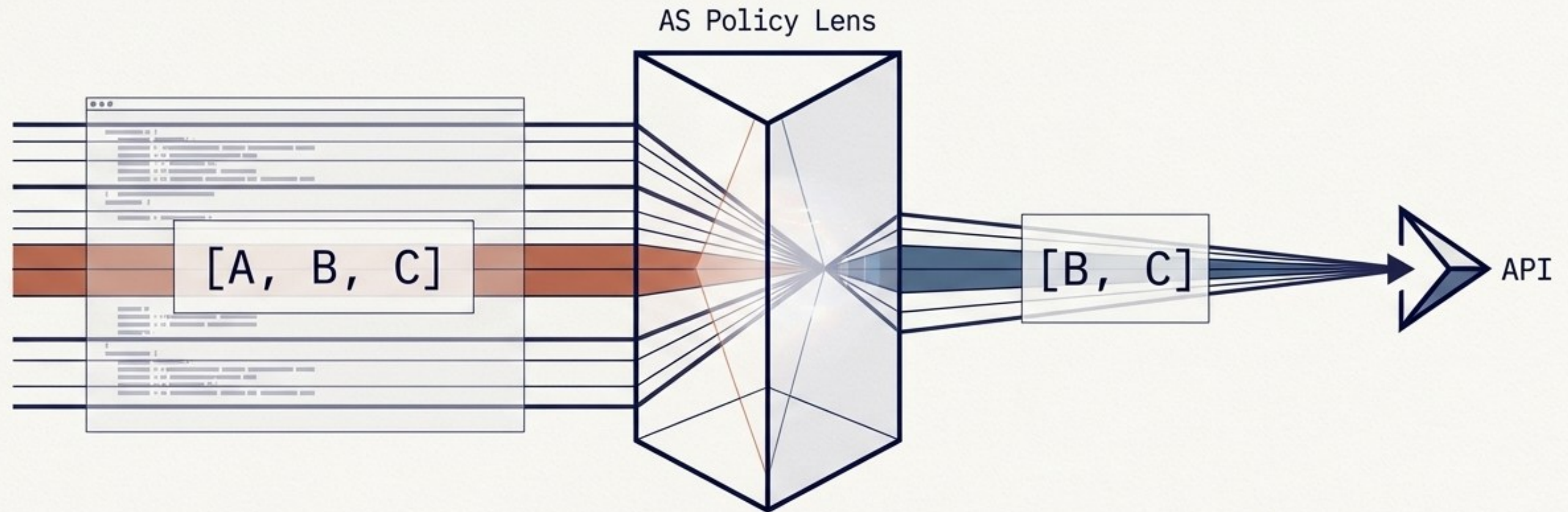
Profile 1: Asserted Chain with Full Disclosure



The simplest baseline. At every hop, the issuing Authorization Server validates the prior chain and simply appends the new actor.

The downstream recipient is granted total transparency and sees the entire, unredacted history of the workflow. It offers zero privacy and relies completely on the AS to truthfully report the chain without silent insertions or deletions.

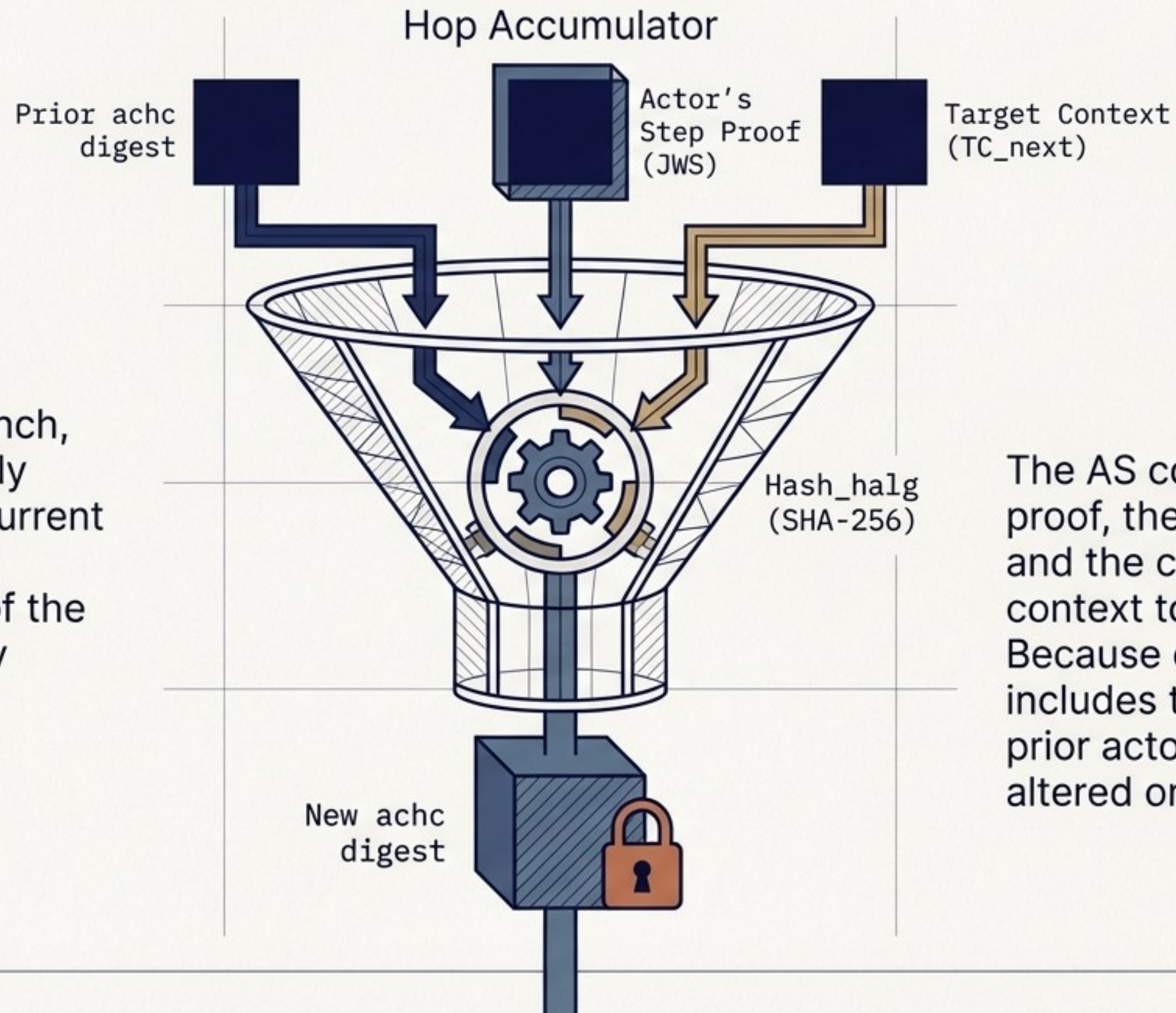
Profile 2: Asserted Chain with Subset Disclosure



Not every API needs to see the original user. In this profile, the Authorization Server enforces a strict disclosure policy. The AS tracks the full chain internally but issues a truncated ach subset to the final destination.

Downstream recipients only see what they are explicitly allowed to see, and must never infer exact chain lengths from what is hidden.

The Mechanics of Committed Chains

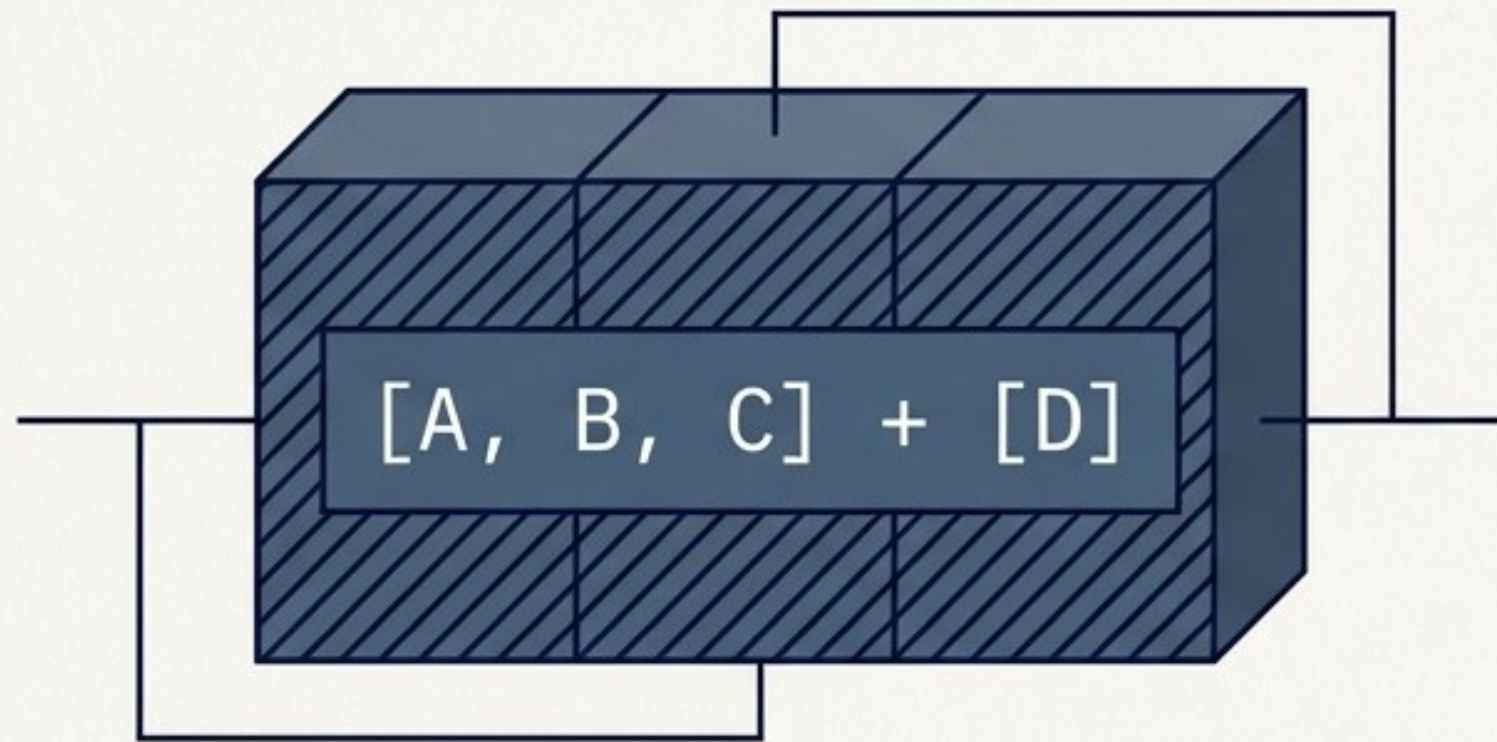


In the Committed branch, actors do not passively receive tokens. The current actor must sign a cryptographic proof of the exact chain state they verified.

The AS combines this step proof, the prior hop's digest, and the canonical target context to produce the achc. Because every hash includes the previous hash, prior actors cannot be silently altered or reordered.

Profile 3: Committed Chain with Subset Disclosure

What the Math Proves



Step proofs and achc values are computed over the exact, complete actor-visible chain.

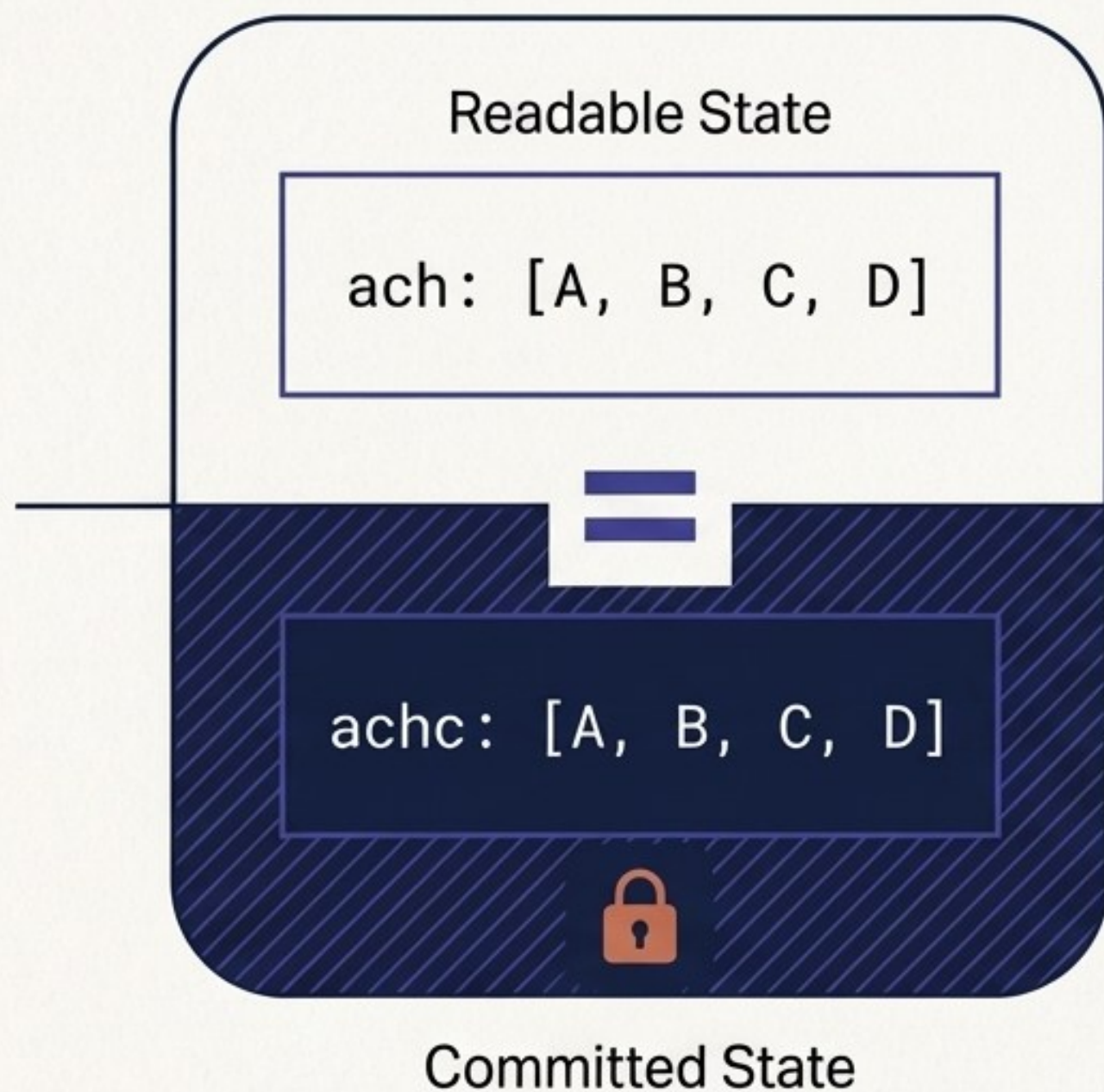
What the API Sees



The ordinary token issued to the downstream API only contains a policy-approved subset. An auditor with AS logs can mathematically reconstruct the rest.

The 'best of both worlds' architecture: High accountability combined with high privacy.

Profile 4: Committed Chain with Full Disclosure



Maximum transparency paired with maximum cryptographic proof.

Every actor and downstream recipient sees the full readable chain. A claim that an actor participated will fail unless a valid **step proof** can be verified against the corresponding prior **committed state**.

Built for **zero-trust** environments where every node must independently verify the entire historical path before executing an action.

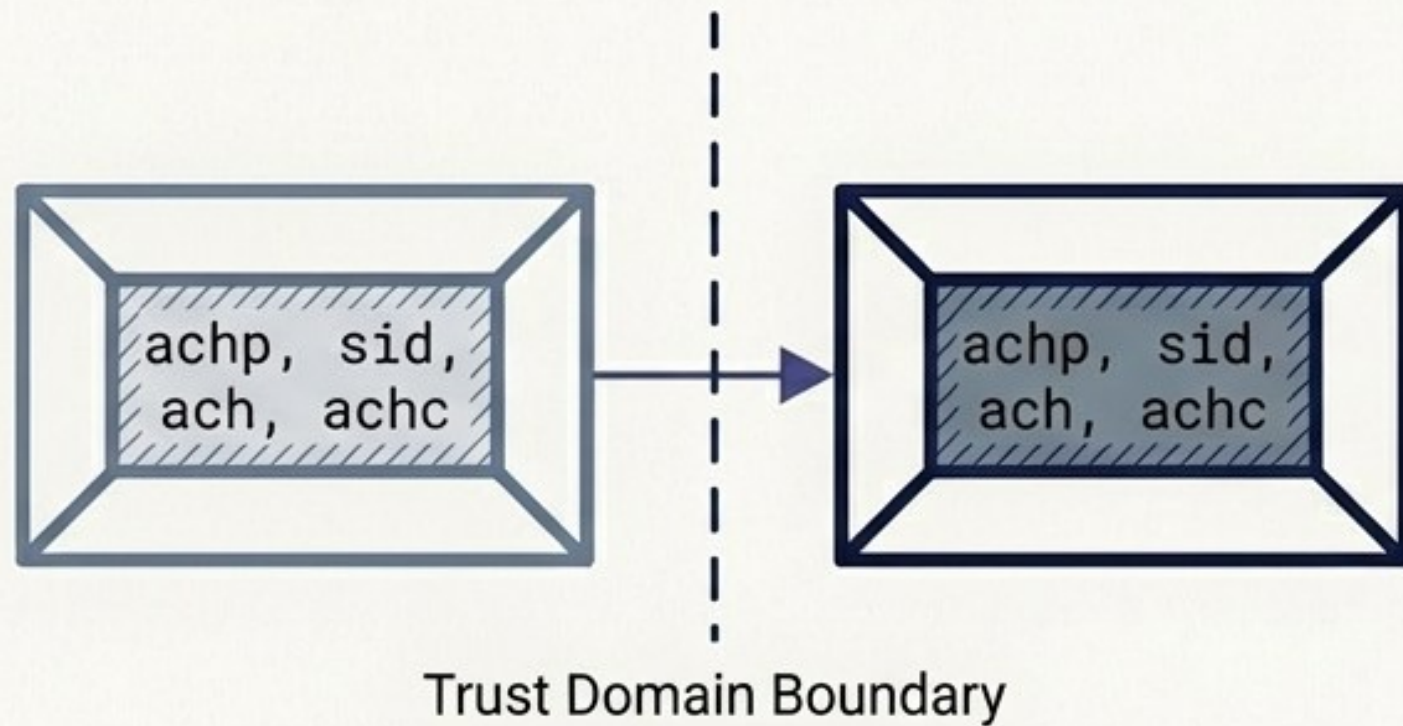
Profile 5: Committed Chain with No Chain Disclosure



- Ultimate inline privacy. The ordinary token completely omits the ach claim.
- Downstream APIs only see the immediate presenting actor. They cannot authorize based on prior-actor membership.
- However, the cumulative committed state continues to be built hop-by-hop. While invisible to the API, the full chain remains mathematically locked inside the token, awaiting an auditor with the retained step proofs to reconstruct it.

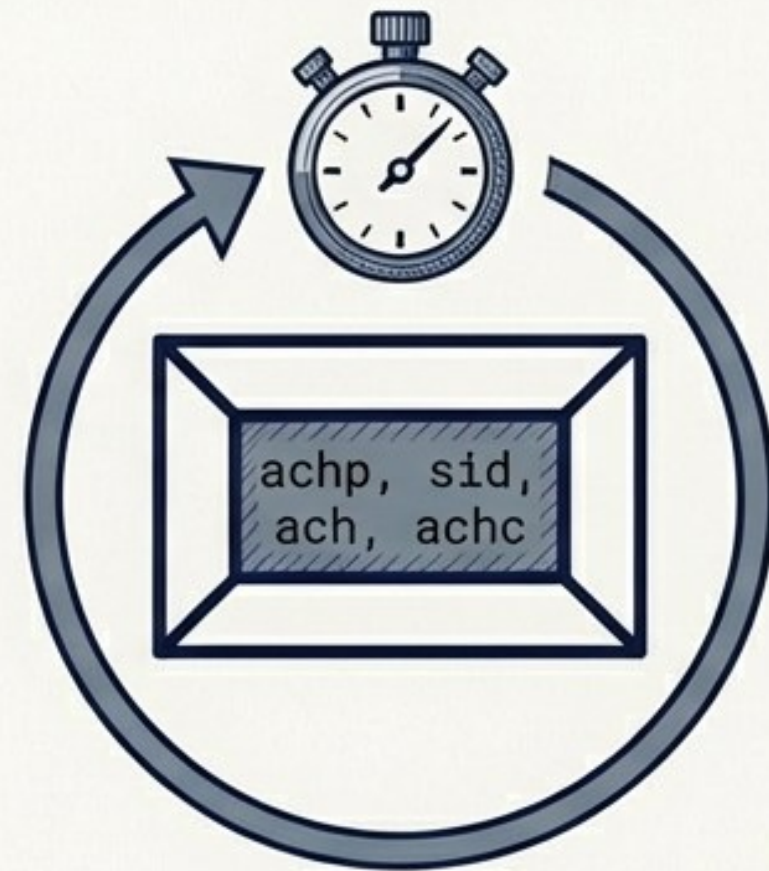
Preserving State: Boundaries and Refreshes

Cross-Domain Re-Issuance



If an API doesn't trust the current AS, the token can be exchanged at a new domain's AS. The new AS issues a local token but must preserve the exact cryptographic state. It **preserves state**; it does not append an actor.

Refresh-Exchange



Tokens must be short-lived. A current actor can **refresh** a token's **lifespan** without generating new step proofs or appending the chain, provided the target context remains identical or narrower.

Security Foundations & Fatal Errors

Implementing SPICE requires rigid adherence to cryptographic hygiene:



1. Sender-Constrained Tokens: Foundational and non-negotiable. If a token can be replayed by an attacker who is not the bound actor, continuity checks instantly fail.



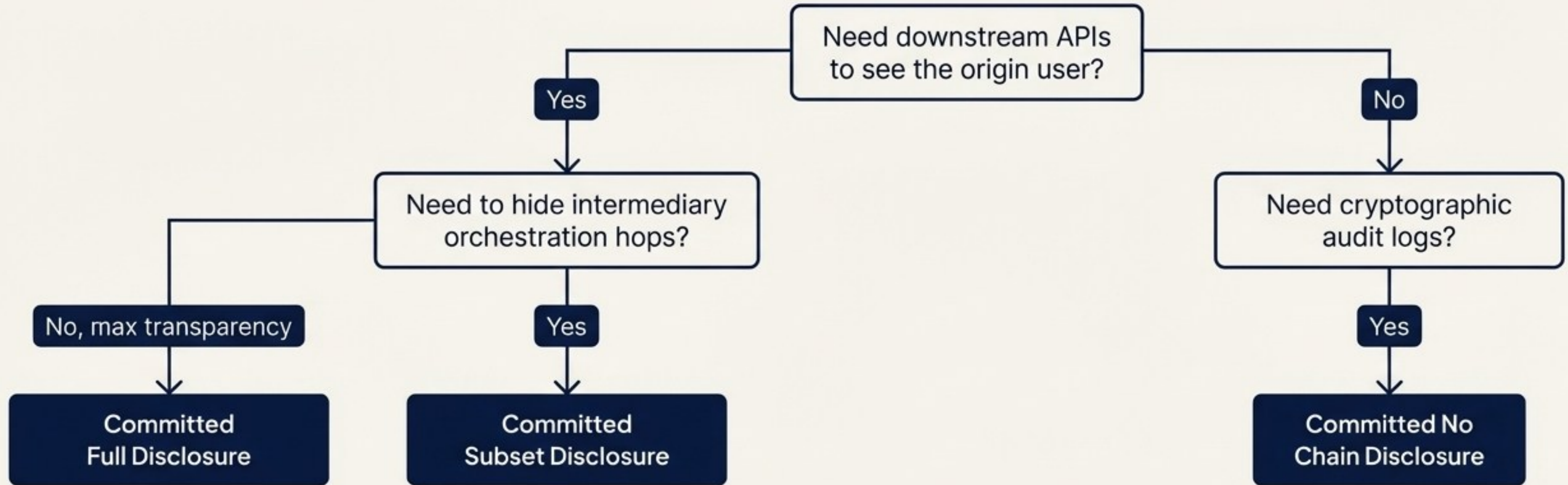
2. Strict Canonicalization: All profile-defined signed/hashed inputs must be serialized using JCS ([RFC 8785](#)). A single misplaced space in the JSON breaks the SHA-256 hash and invalidates the entire chain.



3. Proof-Key Binding: The actor identity, the key used to sign the step proof, and the sender-constrained presentation key must all be cryptographically bound together.

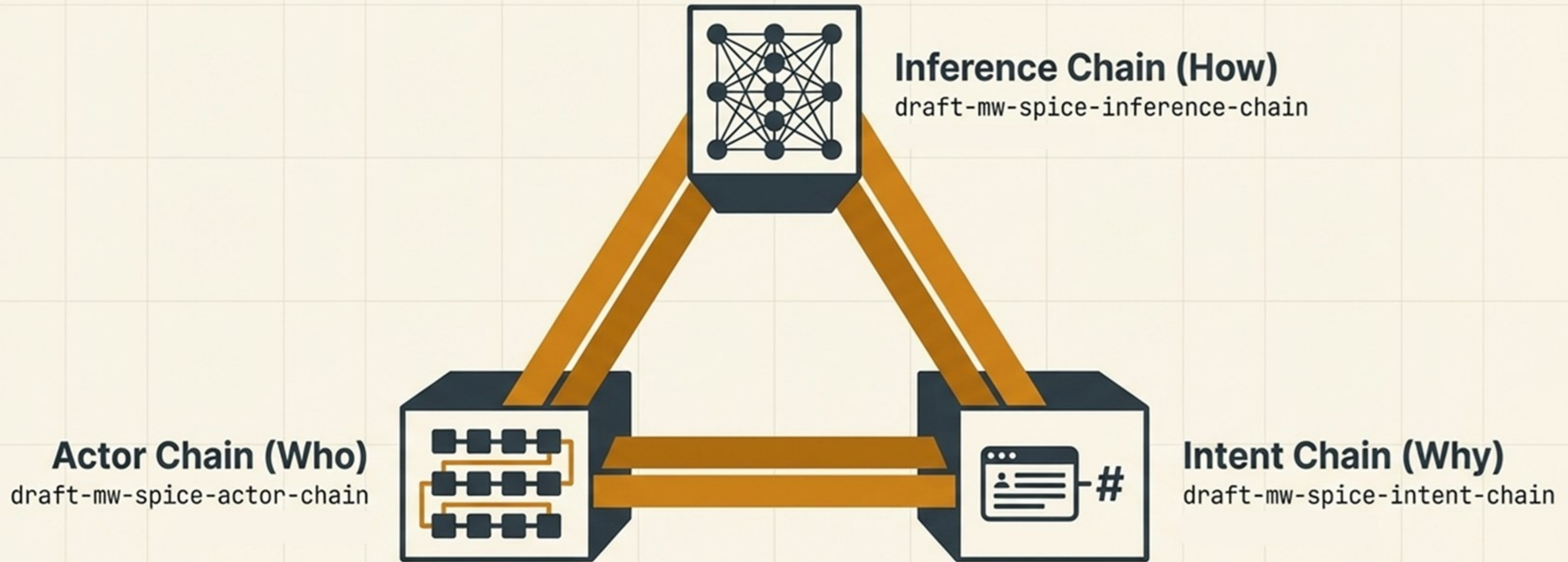
Matching Threat Models to Profiles

The SPICE Actor Chains standard is a flexible menu, not a monolith.



Choose **Asserted** profiles when internal Authorization Servers are highly trusted and mathematical non-repudiation is unnecessary. Choose **Committed** profiles when crossing trust boundaries, operating in zero-trust environments, or facing strict forensic audit requirements.

The Future Horizon: Complete Agentic Provenance



Actor chains solve workload identity today. True zero-trust in AI requires binding the Network Path (Who) to the Initial Human Intent (Why) and the Model Computational Decisions (How).