# Authority Has a Boundary

CoreBond Research Series — CBRS-007
August 2026
Todd Kirton
DOI: 10.5281/zenodo.22241956

## Abstract

Technical trust systems depend on authorities, but mature architectures rarely treat authority as unlimited. Public-key infrastructure, remote attestation, workload identity, federation, and secure software-update frameworks divide roles, constrain policies, bind authority to domains, and preserve relying-party decision authority. This paper examines what happens after those carefully scoped outputs cross architectural boundaries.

The central distinction is between upstream authority scope and downstream policy consequence. A verifier, issuer, certificate authority, or other recognized source may establish a bounded proposition; a receiving system may then legitimately combine that result with local policy and other inputs to reach a stronger decision. The stronger conclusion, however, belongs to the downstream decision process unless the upstream authority was itself entitled to establish it. Using RATS, X.509, SPIFFE, OpenID Federation, TUF, and the DigiNotar compromise, the paper argues for explicit architectural accounting at these handoffs while rejecting the broader claim that modern trust architectures generally fail to scope authority.

**Keywords:** authority; trust scope; relying party; federation; remote attestation; PKI; policy; trust boundaries

## 1. The Signature Is Valid. Now What?

A certificate validates under the rules the relying system has chosen to accept. Its signatures verify, its certification path terminates at an accepted trust anchor, and the applicable policy and constraints permit the path. That result is meaningful. It tells the system that a defined validation process succeeded. (Cooper et al., 2008; Housley, 2024)

Then another system consumes the result.

That second system may not care about certification-path mechanics. It may care whether a workload should receive a secret, whether a device should enter a network, whether an identity assertion should authorize a transaction, or whether signed software should be installed. The validated result becomes an input to a different decision.

At that boundary, the question changes.

The first system established something. The second system wants to conclude something. Those propositions may be related, but they are not automatically identical.

For this paper, authority means a recognized entitlement, within some scope, to assert, appraise, delegate, set policy, or decide. The definition is deliberately operational. A certificate authority, attestation verifier, identity provider, policy owner, and relying party can all exercise authority, but they do not exercise the same authority.

This paper began with a broader suspicion: perhaps technical trust systems granted authority too freely. The research did not support that argument. The architectures examined here repeatedly divide roles, constrain policies, bind authority to domains, restrict delegation, and leave relying parties responsible for their own

decisions. (Birkholz et al., 2023; Cooper et al., 2008; OpenID Foundation, 2026; SPIFFE, n.d.-a; The Update Framework, n.d.-a)

The harder problem was not that authority lacked boundaries. It was what happened when a carefully bounded authoritative output crossed into another system.

The governing question became narrower: What does an authoritative output actually entitle another system to conclude when that output crosses an architectural boundary?

## 2. Authority Is Already Scoped

Any responsible examination of authority has to begin by giving existing engineering credit for what it already knows. Authority is not generally modeled as an unlimited property that an entity simply possesses.

The IETF Remote ATtestation procedureS architecture separates the Attester that produces Evidence, the Verifier that appraises Evidence, and the Relying Party that consumes Attestation Results and applies policy to make an application-specific decision. These are not interchangeable jobs. The architecture separates evidence production, appraisal, and consequence. (Birkholz et al., 2023)

X.509 certification-path validation is also more constrained than the cartoon version of public-key infrastructure suggests. RFC 5280 supports certificate policies, policy mappings, name constraints, path constraints, and other validation rules. RFC 9618 updates the certificate-policy validation algorithm without changing the basic point: acceptance can be policy-scoped rather than an unlimited declaration that a CA is authoritative for everything. (Cooper et al., 2008; Housley, 2024)

SPIFFE makes another boundary explicit. A trust domain represents an administrative or security boundary, and the bundle associated with that domain contains material used to authenticate identities within it. Federation allows separately administered trust domains to recognize one another while preserving the binding between a domain and its authority material. (SPIFFE, n.d.-a, n.d.-b)

The Update Framework makes compartmentalization a security control. Root, Targets, Snapshot, Timestamp, and delegated roles have different responsibilities. Keys and thresholds are assigned by role, and target authority can be delegated over defined paths. Compromise of one role is therefore not intended to grant every other role's powers. (The Update Framework, n.d.-a, n.d.-b)

OpenID Federation likewise builds trust through signed Entity Statements, Trust Chains, metadata policy, and locally resolved metadata. Authority is not simply poured unchanged from a root into an endpoint. Statements are validated and policy is applied as the federation relationship is resolved. (OpenID Foundation, 2026)

The architectures examined here did not forget that authority needs scope. They contain substantial engineering devoted to defining it.

That finding killed the easy version of Paper 7.

The useful question is not whether authority has boundaries. It is whether those boundaries remain visible after systems are composed.

## 3. A Chain Is Not One Decision

Technical diagrams encourage a convenient shortcut. We draw a chain: root, intermediate, endpoint; issuer, verifier, relying party. The arrows make the system look like one continuous act of trust.

It usually is not.

In RATS, an Attester produces Evidence. A Verifier appraises that Evidence and produces an Attestation Result. A Relying Party then uses the result under its own policy to make an application-specific decision. (Birkholz et al., 2023)

Federation follows a similar pattern. An identity provider or credential service provider may authenticate a subscriber and issue an assertion. A separately administered relying party consumes that assertion and decides what service, access, or transaction consequence follows. (National Institute of Standards and Technology, 2025)

OpenID Federation makes composition explicit. Signed statements can form a Trust Chain, but metadata policy is applied while the subject's metadata is resolved. The result used by the receiving entity is therefore the product of validation plus policy processing, not merely an untouched statement inherited from the Trust Anchor. (OpenID Foundation, 2026)

The word trust can hide several different acts: establishing provenance, recognizing a source for a purpose, appraising an assertion or evidence under policy, and deciding what consequence follows.

Calling the whole structure a trust chain is useful shorthand. Treating the chain as one decision is not.

## 4. What the Authority Actually Established

Suppose a signed assertion is authentic. That establishes something valuable about provenance and integrity. It does not automatically establish that the signer is recognized for every possible use a downstream system might make of the assertion.

This distinction is not new. X.509 policy mechanisms constrain the circumstances under which a certification path is acceptable. SPIFFE binds trust material to a particular trust domain. TUF binds keys to roles and delegated target scopes. In each case, the key matters, but so does the scope attached to it. (Cooper et al., 2008; SPIFFE, n.d.-a; The Update Framework, n.d.-a)

A 2026 W3C working draft on recognized entities makes the distinction particularly explicit for verifiable credentials: it defines a way to describe entities as recognized to perform specific actions, such as issuing or verifying a credential. Because that document remains a First Public Working Draft, it is better treated here as corroboration of a distinction already visible in mature architectures rather than as the foundation of the argument. (World Wide Web Consortium, 2026)

A useful model follows: technical authority is relational.

An entity is authoritative for something, under some conditions, to some relying context.

A manufacturer may be authoritative about the provenance of a device model without being authoritative about the identity of the human operating it. An identity provider may be authoritative for an authentication assertion without deciding whether a particular financial transaction should proceed. An attestation verifier may be trusted to appraise Evidence without owning the application policy that consumes the Attestation Result.

The boundary is not a weakness. It is what makes authority manageable.

Problems become possible when that boundary disappears from the handoff.

## 5. When the Meaning Changes at the Boundary

Remote attestation provides a clean example because the architecture already separates the roles.

Assume an Attester produces Evidence describing measurements and claims about a platform. A Verifier evaluates that Evidence under an appraisal policy and returns an Attestation Result stating, in effect, that

the evaluated state satisfies the verifier's defined appraisal criteria. That is the Verifier's result. (Birkholz et al., 2023)

Now assume a Relying Party is deciding whether a workload should receive a production secret.

The Relying Party may legitimately use the Attestation Result as one input. Its policy might require an acceptable platform appraisal, an approved workload identity, the correct deployment environment, and an authorization rule before releasing the secret.

If those conditions are met, the Relying Party can decide to release it.

But notice what happened.

The Verifier did not decide that the workload was entitled to the production secret. It appraised Evidence and produced an Attestation Result within its scope. The Relying Party added other inputs and local policy, then produced the authorization consequence.

The boundary can be stated plainly:

**Upstream authority scope:** what the authoritative source was entitled to establish.

**Downstream policy consequence:** what the receiving system decides to do after combining that result with its own rules and, potentially, other inputs.

Nothing about the second step is improper. It is the expected architecture.

The problem would appear if the later record were interpreted as though the Verifier itself had established the workload's entitlement to the secret. It did not. The authorization may be sound, but the additional conclusion belongs to the Relying Party's policy.

The same pattern can occur elsewhere. Certificate validation can feed identity or authorization. An identity assertion can feed governance. Signed federation metadata can be transformed by local policy. A software signature can feed deployment policy.

In each case, the receiver may legitimately derive a consequence stronger than the upstream assertion. The receiver simply owns the inference it adds.

This is more than the observation that context matters. Composed systems exchange validated outputs through interfaces, but the exact policy, scope, assumptions, and semantic limits behind those outputs do not always travel as cleanly. The receiving system therefore has to preserve enough meaning to distinguish what arrived from what it concluded.

## 6. When Valid Is No Longer Enough

The DigiNotar compromise shows a different authority-boundary failure.

In 2011, attackers compromised the Dutch certificate authority DigiNotar and issued fraudulent certificates for numerous domains. The ecosystem eventually withdrew trust from DigiNotar because the issuance authority could no longer be relied upon. (European Network and Information Security Agency, 2011)

The important distinction is this: a signature may remain mathematically verifiable even after the relying ecosystem no longer accepts the issuer for the purpose at issue. Mathematical verification and warranted reliance are related, but they are not identical.

The response to DigiNotar was therefore not a declaration that signature mathematics had stopped working. The ecosystem changed its decision about an authority whose operational basis for trust had failed.

TUF treats related failure assumptions as design inputs. It compartmentalizes signing authority, uses thresholds, separates roles, supports expiration, and limits delegated target responsibility. Its threat model also recognizes that properly signed content can still be malicious. (The Update Framework, n.d.-a, n.d.-b)

That is the more useful lesson. Cryptographic mechanisms can establish the properties they were designed to establish while the broader authority or process context changes around them.

The correct response is not to demand that cryptography prove everything. It is to remain precise about what still holds and what no longer does.

## 7. Some Authorities Do More Than Describe

Not every authority reports an independently existing fact. Some authorized acts create governed state.

An administrator can grant a role. A governance process can revoke a credential. A system owner can establish or remove a trust anchor. A policy authority can change the conditions that produce authorization.

Those actions matter because the authorized act itself changes the system. The question is not merely whether the authority accurately observed reality.

This does not erase the paper's boundary argument. It limits it. The administrator authorized to grant a database role does not thereby become authoritative about endpoint integrity. The authority is real because its power is recognized within a scope.

## 8. The Boundary Belongs to the Receiver

The research behind this paper began by looking suspiciously at authorities. That was too easy.

The better architectures already know that authority needs boundaries. They divide roles, constrain policies, delegate narrowly, separate appraisal from consequence, and leave relying parties responsible for their own decisions.

The harder accountability begins at the handoff.

When a receiving component adds local policy to an authoritative input, that component owns the additional conclusion produced by that policy. This is architectural accounting, not a moral judgment.

RATS makes the separation concrete: the Verifier appraises Evidence; the Relying Party uses the Attestation Result for its application-specific decision. OpenID Federation similarly resolves signed statements through policy before the receiving entity uses the resulting metadata. (Birkholz et al., 2023; OpenID Foundation, 2026)

A certificate-validation result may become an input to authorization. An identity assertion may become an input to governance. An attestation result may become an input to admission. A signed artifact may become an input to deployment.

Those downstream decisions can be legitimate.

They simply should not borrow more authority from the upstream source than the source actually possessed.

This paper does not establish that systems commonly fail at this boundary. The evidence collected here does not support a prevalence claim. It establishes an architectural risk: as systems are composed, validated outputs can be transformed into stronger consequences, and the semantics of the handoff matter.

Security does not require preventing that transformation. It requires knowing where it occurs and who owns what was added.

**Authority has a boundary.**

The receiver that crosses it owns the inference on the other side.

## 9. Questions Worth Carrying Forward

Paper 7 does not establish that cross-system authority mistakes are universal, common, or the dominant source of trust failure.

It establishes something narrower.

Mature systems already contain sophisticated mechanisms for scoping authority. Their outputs can then be consumed by other systems that add local policy and derive new consequences. Those transformations may be necessary and legitimate, but the stronger downstream conclusion belongs to the downstream decision process.

That leaves several questions worth carrying forward.

- If an assertion is valid, what exactly was its issuer entitled to establish?
- When a downstream system derives a stronger conclusion, whose authority supports that stronger conclusion?
- How should authority scope survive federation, delegation, and policy transformation?
- What must remain visible when an authoritative output crosses an architectural boundary?
- Once the authority is legitimate and correctly scoped, what evidence is sufficient for the consequence that follows?

The last question belongs to the next paper.

Authority determines who is entitled to assert, appraise, recognize, delegate, set policy, or decide. It does not by itself determine whether the evidence is strong enough to justify what comes next.

That is where the inquiry goes next.

**Citation style:** author-date references; standards and guidance cited by named authors or issuing organization.

## References

Birkholz, H., Thaler, D., Richardson, M., Smith, N., & Pan, W. (2023). Remote ATtestation procedureS (RATS) architecture (RFC 9334). Internet Engineering Task Force. https://www.rfc-editor.org/rfc/rfc9334.html

Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., & Polk, W. (2008). Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile (RFC 5280). Internet Engineering Task Force. https://www.rfc-editor.org/rfc/rfc5280.html

European Network and Information Security Agency. (2011). Operation Black Tulip: Certificate authorities lose authority. https://www.enisa.europa.eu/sites/default/files/all_files/Operation_Black_Tulip_v2.pdf

Housley, R. (2024). Updates to X.509 policy validation (RFC 9618). Internet Engineering Task Force. https://www.rfc-editor.org/rfc/rfc9618.html

National Institute of Standards and Technology. (2025). Digital identity guidelines: Federation and assertions (NIST SP 800-63C-4). U.S. Department of Commerce. https://doi.org/10.6028/NIST.SP.800-63C-4

OpenID Foundation. (2026, February 17). OpenID Federation 1.0. https://openid.net/specs/openid-federation-1_0.html

SPIFFE. (n.d.-a). SPIFFE trust domain and bundle. Retrieved August 12, 2026, from https://spiffe.io/docs/latest/spiffe-specs/spiffe_trust_domain_and_bundle/

SPIFFE. (n.d.-b). SPIFFE federation. Retrieved August 12, 2026, from https://spiffe.io/docs/latest/spiffe-specs/spiffe_federation/

The Update Framework. (n.d.-a). Roles and metadata. Retrieved August 12, 2026, from https://theupdateframework.io/docs/metadata/

The Update Framework. (n.d.-b). Security. Retrieved August 12, 2026, from https://theupdateframework.io/docs/security/

World Wide Web Consortium. (2026, May 12). Recognized entities v1.0 (First Public Working Draft). https://www.w3.org/TR/vc-recognized-entities-1.0/