Uncertainty

CoreBond Research Series — CBRS-011
August 2026
Todd Kirton
DOI: 10.5281/zenodo.22242717

1. Same Decision, Different Meaning

Consider two systems that receive the same answer: deny. In the first case, a verifier receives valid evidence showing that the system does not satisfy the required policy. In the second, no valid appraisal result is available within the period required by policy. Denial may be correct in both cases. Yet the two situations do not mean the same thing. One contains evidence against the required claim. The other establishes only that the claim could not be established in time; it does not, by itself, establish why.

That distinction can disappear early. If every downstream component receives only "deny," "non-compliant," or "untrusted," the enforcement decision survives while the evidentiary condition behind it may not. Nothing about this observation requires the second system to receive access. The narrower question is whether the system discarded information that another authorized security decision still needed.

This is not an argument for richer error messages. Diagnostic detail becomes relevant to trust architecture only when a distinction in evidence or appraisal legitimately changes downstream security policy, remediation, privilege, isolation, audit interpretation, retry, recovery, or later treatment. When it changes none of those things, preserving it may add no value.

A consequential trust or authorization decision may therefore be discrete while the appraisal feeding it remains qualified. When materially different evidentiary conditions change what an authorized downstream component should do, those distinctions should survive until that decision boundary has used them.

2. Binary Action Is Not the Problem

It is tempting to blame binary decisions themselves. That would be wrong. Security controls eventually have to act. A policy enforcement point cannot remain philosophically undecided while a request waits forever. It grants, denies, isolates, limits, redirects, or performs some other concrete operation.

NIST's Zero Trust Architecture makes the distinction clear. Its policy engine can evaluate identity, device state, behavioral information, threat intelligence, enterprise policy, and other contextual inputs before reaching an access decision (Rose et al., 2020). A discrete output therefore does not imply simplistic reasoning. Complex evidence can legitimately end in a simple action.

The IETF Remote ATtestation procedureS architecture separates the stages even more explicitly. An Attester produces Evidence. A Verifier appraises that Evidence under an appraisal policy and produces Attestation Results. A Relying Party then applies its own appraisal policy to those results to determine the extent to which it will trust the Attester or permit an operation (Birkholz et al., 2023). The final decision can be discrete without requiring the preceding appraisal to be so.

That separation also explains why the Verifier need not make the entire operational decision. A Verifier can specialize in determining what Evidence establishes without owning every relying context in which that result will be used. The Relying Party may know the requested operation, local policy, mission consequence, or

acceptable fallback that the Verifier does not. A qualified handoff can therefore be an intentional separation of responsibilities rather than redundant detail.

3. Security Already Admits Uncertainty

The underlying idea that security decisions operate under uncertainty is not new. NIST's systems security engineering guidance treats assurance as a basis for justified confidence supported by evidence. The objective is not omniscience. Evidence reduces uncertainty to a level appropriate to the claims, consequences, and protection needs involved (Ross et al., 2022).

Paper 11 therefore cannot honestly argue that established security engineering assumes certainty and needs to discover uncertainty for the first time. It already knows better. Verification and validation can strengthen confidence without proving that every relevant fact is known. Trust can also be granted without an adequate basis in trustworthiness, which is precisely why the evidentiary basis matters.

The remaining problem is not whether uncertainty exists. It is what happens to material meaning as evidence becomes appraisal, appraisal becomes a result, and a result becomes an operational decision.

4. Not Established Is Not One Condition

A required claim can fail to become established for different causal reasons. Available evidence may positively support a condition contrary to policy. Evidence may exist but be inadequate in scope, freshness, provenance, or strength to establish the required claim. Or no usable appraisal result may be available within the required policy window.

These examples are not mutually exclusive formal classes, and they are not a proposed taxonomy. Unavailable evidence is, in one sense, also insufficient evidence. The point is operational: the cause of insufficiency can matter. RFC 9334's architecture separates Evidence appraisal by the Verifier from appraisal of Attestation Results by the Relying Party, and it also accounts for conditions in which a required Verifier is unavailable (Birkholz et al., 2023). Those architectural distinctions are useful here without turning Paper 11's examples into RFC-defined classes.

Those paths can lead to the same immediate enforcement result and still justify different follow-up. Evidence that positively contradicts the required claim may trigger incident response or isolation. Evidence that does not establish a required claim may call for stronger corroboration, while failure to obtain a valid appraisal result may call for retry or investigation of the attestation path.

The system must also resist pretending it knows the root cause when it does not. A timeout does not prove a Verifier failed. It may reflect network disruption, local malfunction, attack, overload, or another cause. Qualified appraisal should communicate what can actually be established—for example, that no valid result arrived within the policy window—rather than manufacture certainty about why.

5. Where Meaning Gets Lost

RFC 9334 already provides the architectural separation needed to avoid this problem. Attestation Results can convey richer Claims for the Relying Party to evaluate rather than forcing every implementation into a one-bit answer (Birkholz et al., 2023). Premature semantic collapse is therefore not a defect in the RATS architecture. It is a profile, implementation, or system-design risk when information that matters to downstream policy is compressed before that policy can use it.

The risk appears when a richer evidentiary state becomes a result that no longer distinguishes conditions the downstream policy legitimately treats differently. A Verifier or appraisal process may know that a claim was

contradicted, not established, not evaluated in time, or established only within a bounded scope. If all such outcomes are flattened into an undifferentiated result, the Relying Party may be unable to perform a decision that the architecture intentionally assigned to it.

Risk assessment offers a useful analogy, though not a direct attestation model. NIST notes that risk assessments contain uncertainty and may be coarse even when organizations ultimately assign results to discrete categories (Joint Task Force Transformation Initiative, 2012). A clean output category does not make the underlying knowledge equally precise.

That is the narrow meaning of premature semantic collapse: an operational result remains usable for one decision while losing a material distinction needed for another authorized decision downstream.

6. Preserve Only What Is Material

The obvious response would be to forward everything. That would create a different security problem. Raw Evidence can be device-specific, sensitive, difficult to interpret, and unnecessarily revealing. RFC 9334 allows a Verifier to transform Evidence into Attestation Results that are more useful to a Relying Party without requiring the Relying Party to consume every underlying measurement (Birkholz et al., 2023). Abstraction is a feature.

Richer qualification also has costs. More result states can increase parser and policy complexity, create inconsistent cross-vendor interpretation, expose sensitive implementation detail, and create downgrade paths if one condition is treated more gently than another. Semantic richness is not automatically safer.

Materiality therefore needs a practical test. Before collapsing two appraisal outcomes, ask whether the authorized downstream component would choose a different action, remediation path, retry behavior, privilege, isolation response, audit interpretation, recovery step, or later treatment if it knew the difference. If the answer is no, collapse may be harmless. If the answer is yes, the distinction should survive until that component has used it.

The goal is minimum sufficient qualification, not maximum transparency. Pass enough information to state what was established and any limitation that materially changes downstream policy, while withholding raw or irrelevant evidence. The value of a distinction must justify the complexity and disclosure introduced by preserving it.

7. Uncertainty Is Not Permission

Any design that preserves uncertainty-related semantics faces an adversarial problem: attackers can manufacture ambiguity. They can suppress telemetry, interrupt communications, force evidence to become stale, or target the appraisal path itself. A system that automatically treats “unknown” more gently than “bad” creates a path worth attacking.

For that reason, preserving the reason a claim was not established must never guarantee a softer outcome. Fail-closed remains legitimate. If required evidence is unavailable, denial may be exactly right. If evidence is inadequate, isolation may be right. If the system cannot defensibly distinguish the cause, policy may need to treat the condition conservatively.

The informational distinction can remain useful even when the immediate action is identical. A denial supported by positive adverse evidence may trigger a different investigation than a denial caused by failure to obtain a valid appraisal result. Preserving that distinction does not weaken enforcement. It prevents the enforcement output from becoming the only surviving account of what the system actually established.

The qualification itself becomes security-relevant when policy relies on it. RFC 9334 treats Attestation Results as information for which the Verifier vouches within the relevant trust relationship (Birkholz et al., 2023). If an attacker can rewrite a result supported by adverse evidence into a result that merely reports unavailable appraisal, the semantics have become another attack surface. Their integrity and provenance must therefore be protected commensurate with the decisions they influence.

8. Context Can Change the Response

Fail-closed is important, but not every mission context responds to adversity by shutting down every function. Cyber resiliency engineering explicitly addresses systems that anticipate, withstand, recover from, and adapt to adverse conditions while preserving mission-essential functions where required (Ross et al., 2021).

This does not entitle an uncertain actor to degraded access. It demonstrates only that the same evidentiary limitation can matter differently under different authorized policies. One environment may correctly deny everything. Another may isolate a subsystem, require additional corroboration, or permit only a bounded mission-essential function under separate controls.

The point is policy dependence, not permissiveness. The appraisal should preserve the material distinction long enough for the component authorized to make that contextual decision. What happens next remains a policy question.

9. The Decision Boundary

The central design boundary is not between binary and non-binary trust. It is between appraisal and consequence. Evidence can be complex. Appraisal can be qualified. The operational action can still be discrete. What should not disappear prematurely is a material distinction that the consequential decision still needs.

This is consistent with RATS rather than corrective of it. The Verifier specializes in appraising Evidence; the Relying Party applies policy to Attestation Results. Rich results can abstract device-specific detail while preserving Claims needed by the relying context (Birkholz et al., 2023). The useful question is not how much information can be sent. It is how much must survive for the downstream component to perform its assigned security decision correctly.

The principle is simple: do not erase material evidentiary distinctions before the decision boundary that needs them. Material is doing the heavy lifting. A distinction that changes nothing downstream can be ignored. A distinction that changes policy, remediation, audit, or subsequent handling should not vanish merely because the immediate enforcement output can be represented simply.

10. What Uncertainty Does Not Require

This principle does not require a universal trust score. Qualitatively different limitations in evidence or appraisal do not automatically share a meaningful common scale. Turning those limitations into “72 percent trusted” can manufacture precision the evidence never supported.

It does not require a universal third state called unknown, raw Evidence to be exposed to every Relying Party, or a new uncertainty taxonomy. It does not make fail-closed wrong, and it does not say that every diagnostic distinction belongs in trust policy.

Most importantly, it does not claim that RATS, zero trust, or assurance engineering failed to recognize the underlying issues. Those architectures supply much of the structure needed to handle them. The narrower

contribution is a design principle about preserving material appraisal semantics across a legitimate decision boundary when downstream policy actually depends on them.

11. The Question That Remains

A security architecture does not need to make uncertainty permissive or trust probabilistic. It does need to avoid destroying material meaning before the responsible policy has used it. A decision can remain firm—even deny—while the system preserves the qualification necessary to enforce, remediate, audit, or interpret that decision correctly.

That resolves only the decision at a moment in time. Evidence ages. Systems change. Attackers act. Software is updated. Policies change. A decision that was justified when made may later rest on facts that no longer describe the present.

If a trust decision is justified by evidence available at one moment, what makes that decision remain justified after the evidence, actor, environment, or relevant conditions change? That is the question for Paper 12.

References

Birkholz, H., Thaler, D., Richardson, M., Smith, N., & Pan, W. (2023). Remote attestation procedures architecture (RFC 9334). Internet Engineering Task Force. https://doi.org/10.17487/RFC9334

Joint Task Force Transformation Initiative. (2012). Guide for conducting risk assessments (NIST Special Publication 800-30 Rev. 1). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-30r1

Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). Zero trust architecture (NIST Special Publication 800-207). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-207

Ross, R., Pillitteri, V., Graubart, R., Bodeau, D., & McQuaid, R. (2021). Developing cyber-resilient systems: A systems security engineering approach (NIST Special Publication 800-160 Vol. 2 Rev. 1). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-160v2r1

Ross, R., Winstead, M., & McEvilley, M. (2022). Engineering trustworthy secure systems (NIST Special Publication 800-160 Vol. 1 Rev. 1). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-160v1r1