# Continuous Evaluation: When Is Trust Finished?

CoreBond Research Series — CBRS-012
August 2026
Todd Kirton
DOI: 10.5281/zenodo.22242901

## A Decision Can Be Correct and Still Become Wrong to Reuse

At time T0, a system evaluates the evidence available to it and reaches a defensible trust decision. The actor is authenticated, the device satisfies the required posture, the relevant evidence is fresh enough, and policy permits the requested action. The decision may be defensible on the evidence and policy available at that moment. At time T1, however, a later action may attempt to inherit the authority of the earlier decision. The credential may have been revoked, the device may have fallen out of compliance, policy may have changed, or evidence that was sufficient at T0 may no longer describe current state. The earlier decision can remain defensible in its original context while its continued use becomes unjustified. Later evidence can also reveal that the original decision was wrong when made; continued reliance therefore cannot assume that the earlier judgment is beyond revision.

That distinction is easy to blur because security discussions often answer the problem with some version of continuous evaluation. The phrase sounds precise until it is asked to carry architectural weight. Does continuous mean constant authentication, continuous telemetry, per-request authorization, periodic reauthentication, event-driven revocation, repeated attestation, or simply the ability to reconsider a decision when relevant conditions change? Those mechanisms are not interchangeable.

This paper argues that a trust decision does not need to be continuously recomputed to remain usable. But when later actions inherit that decision, the architecture should define what changes would make its required premises no longer sufficient and should intentionally bound how long such a materially outdated decision can continue to control enforcement. The question is therefore not whether trust can ever be decided. It can. The harder question is whether the next action is still entitled to inherit that decision. These conditions need not be exhaustive: where relevant change cannot be fully enumerated or observed, independent bounds and uncertainty handling remain part of the design.

## Continuous Does Not Mean Constant Recalculation

Zero Trust architecture provides a useful starting point because it rejects broad inherited trust. NIST SP 800-207 describes resource access as a policy decision based on the confidence appropriate to a particular request, including subject identity, device security posture, time, location, and other changing factors. It specifically warns against assuming that later resource requests remain valid merely because a subject previously met a baseline authentication requirement (Rose et al., 2020). That is a strong argument for repeated decision-making at meaningful access boundaries. It is not a requirement to recompute every supporting proposition at every instant.

NIST's language around continuous monitoring makes the distinction clearer. Information security continuous monitoring is defined as maintaining ongoing awareness, while “continuous” and “ongoing” mean assessment at a frequency sufficient to support risk-based security decisions (National Institute of Standards and Technology [NIST], n.d.). Continuous, in this sense, is an adequacy standard rather than a claim of zero-delay observation.

That matters because evaluation has several clocks. A policy engine may run for each request while the posture information it consumes is older. A monitoring service may collect telemetry continuously but change authorization only after a threshold is crossed. A session may remain active between periodic reauthentication events. An event receiver may react immediately when an event arrives but know nothing during a delivery failure. Treating all of these as one frequency hides the actual security boundary.

## Bounded Inheritance Is Legitimate

The strongest counterexample to literal continuous authentication is ordinary session management. NIST SP 800-63B-4 defines overall and inactivity timeouts, requires periodic reauthentication according to assurance level, and permits session monitoring as a risk-reduction measure during an authenticated session (Temoshok et al., 2025). In that domain, bounded inheritance is therefore part of the architecture rather than a failure to evaluate continuously.

This is not a concession to weak security. NIST notes that session management can be preferable to continually presenting credentials because constant credential presentation can create usability pressure and insecure workarounds (Temoshok et al., 2025). A system can legitimately finish an authentication ceremony and use the result for later interactions. The important word is bounded.

Once bounded inheritance is accepted, the problem changes. The question is no longer whether every decision must be repeated. It becomes what gives a previous decision the right to continue controlling later actions, and what ends that right.

## The Premises Under a Decision Do Not Age Together

A consequential trust decision is rarely supported by one fact. It may depend on who or what is acting, whether a credential remains acceptable, whether a device meets posture requirements, whether evidence is sufficiently fresh, whether an authority remains recognized, whether policy still permits the action, and whether the requested action remains inside the scope originally evaluated. Those premises do not necessarily change together.

NIST's Zero Trust model makes this visible by identifying multiple dynamic inputs to a resource-access decision (Rose et al., 2020). The RATS architecture makes a related distinction around freshness: Evidence or an Attestation Result may need an acceptable epoch, yet the acceptable freshness threshold is locally defined by appraisal policy. RFC 9334 explicitly acknowledges that state can change immediately after Evidence is produced and describes the goal as narrowing recentness to something the Verifier or Relying Party is willing to accept (Birkholz et al., 2023). It also preserves flexibility because caching and reuse can be valuable.

The result is a composite decision whose supporting propositions can have different applicability horizons. A durable identity relationship does not imply durable device compliance. A fresh attestation does not freeze policy. A valid session secret does not prove that every condition used to

authorize the session remains unchanged. Decision age alone therefore cannot tell us whether the decision still deserves inheritance.

## What Makes Continued Reliance Stop Being Defensible?

For this paper, a material invalidator is a change, expiration, revocation, or loss of evidentiary support that can make a proposition required by a prior decision no longer sufficiently established for the later action. The term is analytical language, not terminology borrowed from NIST, IETF, or OpenID.

The important constraint is materiality to the decision. Not every environmental change deserves to reopen trust, and no architecture can enumerate every future invalidator in advance. A device changing networks matters only when network context is relevant. A software update matters only when the decision depends on properties the update can affect. A credential change matters when continued authorization depends on the current credential state. This keeps reevaluation tied to propositions rather than turning every observable change into a security emergency. Known invalidators, confidence-loss conditions, expiry bounds, and fallback behavior can be defined explicitly, while unknown compromise and unobserved failure remain residual risks that bounded reliance must still contain.

Existing standards provide concrete examples of such changes without supplying one universal taxonomy. OpenID CAEP defines events for session revocation, credential changes, assurance-level changes, token-claims changes, and device-compliance changes (Cappalli & Tulshibagwale, 2025). RFC 9334 treats acceptable freshness as an appraisal-policy concern and warns against continued use beyond the period for which an Attestation Result remains fresh enough (Birkholz et al., 2023). NIST session rules impose defined bounds on continued authenticated sessions (Temoshok et al., 2025). These are different mechanisms. Taken together, they support the narrower synthesis that a prior result can remain a valid record of an earlier decision while losing authority over what happens next.

## Events Help — Silence Does Not Prove Continuity

Event-driven evaluation is attractive because it can shorten the delay between a meaningful change and a response. CAEP is built for that purpose: cooperating transmitters can send continuous security updates that receivers use to attenuate access to users, devices, sessions, and applications (Cappalli & Tulshibagwale, 2025). A device can become noncompliant after authentication, for example, and that later change can be communicated without pretending that the original authentication event never happened.

But event-driven architecture has its own epistemic problem: silence is ambiguous. The OpenID Shared Signals Framework includes verification behavior because low event frequency can make it difficult for a receiver to distinguish expected silence from event-transmission failure. Successful receipt of a verification event can confirm end-to-end delivery, including validation of the event (OpenID Foundation, 2025). The need for that mechanism exposes the weakness in treating “no event received” as evidence that nothing changed.

An event may never be generated, may be delayed, may fail in transmission, or may arrive after an enforcement decision has already inherited stale authority. The event itself can also be false, forged, or mis-scoped, so an invalidation signal requires trustworthy provenance and appraisal just as other

decision evidence does. Events can reduce uncertainty about observed changes; event delivery alone does not eliminate uncertainty about unobserved change, channel failure, or subsequent state.

## Timers Help — Age Does Not Define Applicability

Time-based controls address a different failure mode. Overall session timeouts, inactivity limits, evidence freshness thresholds, and periodic reauthentication can terminate inherited authority even when no adverse event is known. Their value is not that age defines applicability, but that an independent bound can contain inherited authority when observability is incomplete.

RFC 9334 makes that limitation explicit: freshness is policy-dependent, and the state of an Attester or its appraisal policies can change immediately after Evidence or an Attestation Result is generated (Birkholz et al., 2023). A result can therefore remain inside an accepted freshness window without necessarily describing the latest state.

Timers are therefore safety bounds, not a universal theory of applicability. Events can tell a system that something changed before a timer expires. Timers can stop indefinite inheritance when an event never arrives. Neither mechanism alone answers whether the prior decision remains sufficient for the next action.

## The Blind Window

The remaining exposure can be described as a blind window. For this paper, the blind window is the actual or bounded interval during which a materially outdated decision can continue influencing enforcement after a relevant premise has changed but before the architecture detects, communicates, appraises, decides on, or enforces an appropriate response. The exact onset may be known, inferred later, or unknowable; in the last case, the useful quantity is a bound on how long outdated authority could have persisted. The term is a synthesis used here, not an established standards term.

This framing matters because nominal evaluation frequency can be misleading. As an illustration, imagine a policy engine that evaluates every request once per second but receives device-posture data only every twenty minutes. The decision loop is fast; the knowledge feeding it is not. A meaningful device change can remain invisible to the policy engine for most of that twenty-minute interval. Conversely, an event-driven system may perform little work during stable periods yet react quickly when a trusted signal reports a material change.

The blind window is not one universal number. It depends on consequence, observability, and the mechanisms available in a particular architecture. It may include sensing delay, event generation, transmission, appraisal, policy computation, propagation, and enforcement. Some decisions may tolerate minutes; others may require fresh evaluation at the action boundary. The architectural argument here is not that every window be minimized without limit, but that the allowed exposure be intentional rather than an accidental by-product of stale inputs and inherited authority.

## More Evaluation Can Make the System Worse

A natural reaction is to collect more signals, evaluate more often, and react faster. That can improve security until the additional machinery begins creating new failure modes.

NIST SP 800-63B-4 explicitly notes that session-monitoring characteristics such as usage patterns, behavioral characteristics, device and browser characteristics, geolocation, and network information have privacy implications and belong in privacy risk assessment (Temoshok et al., 2025). Continuous evaluation is therefore not a license for continuous surveillance. A signal should justify its collection by materially improving a decision, not merely because it is observable.

Availability imposes another constraint. If every consequential action requires a fresh answer from a remote posture, identity, or policy service, loss of that service can force the architecture into a deliberate security tradeoff: deny the action, fall back to bounded local authority, or accept some other explicitly governed degradation mode. Signal quality matters as well. In principle, noisy or borderline observations can drive repeated access changes or needless reauthentication if policy reacts directly to every fluctuation. Depending on the domain, designers may therefore choose corroboration, thresholds, or other stabilization before treating an observation as an invalidator.

The goal is not maximum evaluation. It is enough evaluation, with enough reliability, to keep the authority of a prior decision inside the bounds appropriate to its consequences.

## Reevaluation Should Follow What Changed

A material invalidator also does not imply that the entire trust process must restart. CAEP's event types illustrate why. A device-compliance change is not the same event as a credential change or a session revocation (Cappalli & Tulshibagwale, 2025). Those distinctions would be pointless if every later change required the same response.

Where dependencies can actually be isolated, premise-directed reevaluation can avoid repeating unrelated work. A revoked authenticator may require renewed authentication. A changed resource policy may require a new authorization decision while leaving identity evidence untouched. A stale attestation result may require new platform evidence without requiring a user to establish identity again. Where dependencies are tightly coupled, broader revalidation may still be necessary.

This is not a proposal for a universal dependency graph or protocol. Some systems will not be able to isolate premises cleanly, and some consequences justify broader revalidation. The narrower point is that 'evaluate again' should not silently mean 'repeat everything.'

## When Is Trust Finished?

A discrete trust decision can finish. Authentication can succeed. Evidence can be appraised. Authorization can be granted. A session can begin. Those events can be treated as completed decisions without assuming that later evidence can never revise how the original judgment is understood.

What does remain bounded is their authority over future actions. A later action inherits a prior decision only while the premises required for that later action remain sufficiently applicable. Some premises may persist. Others may expire, change, be revoked, or lose evidentiary support. The architecture does not need infinite recomputation, but it does need a defensible answer for how those changes become visible and how long an outdated conclusion may continue to control enforcement.

Durable trust anchors do not contradict this model. A manufacturer root, identity relationship, policy authority, or reference value can remain long-lived while the active decision derived from it remains

bounded. The lifecycle of an authority is not the same as the lifecycle of every conclusion that depends on it.

## Conclusion — The Lifecycle of a Decision

Continuous evaluation is most useful when it stops being treated as a frequency slogan. Mature security architectures already combine sessions, per-request policy, event signaling, freshness appraisal, periodic reauthentication, and monitoring. Those mechanisms demonstrate that trust can be both decided and revisited without requiring constant reconstruction.

The deeper problem is continued reliance. A decision can remain correctly recorded as a past outcome after the premises that made it defensible have begun to diverge from current conditions. Events can expose some changes quickly. Time and scope bounds can contain authority when events are missing. Monitoring can improve visibility but creates privacy, availability, and reliability costs. None of those mechanisms eliminates the need to decide what changes matter to the decision being inherited.

> **A decision can finish while the obligation to justify continued reliance on it does not. Continued reliance should remain bounded by the applicability of the premises that made the decision defensible. The trust decision can finish when it is made. Its authority over what happens next is not unlimited.**

## References


Birkholz, H., Thaler, D., Richardson, M., Smith, N., & Pan, W. (2023). Remote ATtestation procedureS (RATS) architecture (RFC 9334). Internet Engineering Task Force.

Cappalli, T., & Tulshibagwale, A. (2025). OpenID Continuous Access Evaluation Profile 1.0. OpenID Foundation.

National Institute of Standards and Technology. (n.d.). Information security continuous monitoring (ISCM). Computer Security Resource Center Glossary.

Temoshok, D., Fenton, J., Choong, Y.-Y., Lefkovitz, N., Regenscheid, A., Galluzzo, R., & Richer, J. (2025). Digital identity guidelines: Authentication and authenticator management (NIST SP 800-63B-4). National Institute of Standards and Technology.

OpenID Foundation. (2025). OpenID Shared Signals Framework Specification 1.0.

Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). Zero trust architecture (NIST SP 800-207). National Institute of Standards and Technology.