CBRS BOUNDED TRUST FRAMEWORK

A framework for making, using, preserving, revising, and recovering trust decisions

Version 1.2 — August 2026

Todd Kirton

10.5281/zenodo.22256825

CORE PRINCIPLE

A trust-relevant conclusion should carry no more meaning, authority, scope, consequence, or lifetime than its supporting evidence, applicable authority, policy, and context justify.

Derived from the frozen twelve-paper CoreBond Research Series (CBRS-001 through CBRS-012).

Designed to be technology-, protocol-, and product-neutral; external validation remains part of the research agenda.

Document Status

This document is the reviewed formal synthesis of the frozen CoreBond Research Series. It does not replace the twelve source papers. The papers remain the evidentiary and argumentative record; this framework extracts the recurring architectural structure that survived their research, counterarguments, scope corrections, red-team review, and adversarial stress testing.

Version 1.2 incorporates three mandatory findings from the v1.1 adversarial stress test: explicit root assumptions, policy treated as a trust-bearing dependency, and explicit handling of materially relevant disagreement and reconciliation. These additions refine the accounting model without changing the framework's three boundaries or bounded-decision thesis.

The framework is analytical and architectural. It is not a protocol specification, cryptographic construction, product architecture, implementation standard, certification scheme, or universal trust score. Terms introduced here are framework terminology unless explicitly identified as terminology already used by a cited source paper.

Framework status
Version 1.2 incorporates the mandatory findings from internal red-team and adversarial stress testing. The analytical core is considered stable enough for external mapping and hostile application against established architectures and standards. Future changes should be versioned rather than silently rewriting the frozen CBRS source papers.

Executive Summary

For purposes of BTF, the framework analyzes trust-relevant consequences through bounded decisions rather than treating a prior trust state as unlimited authority. A bounded decision is supported by evidence, applicable authority, appraisal, policy, and context for a particular consequence. The framework exists because the twelve CBRS papers repeatedly reached the same conclusion from different directions: technically valid inputs can become semantically overextended when systems silently add meaning, borrow authority, reuse stale state, or allow an earlier decision to govern later actions beyond the conditions that justified it.

The framework has three primary boundaries: meaning is bounded by what the evidence actually establishes; authority is bounded by what a source or decision-maker is entitled to establish; and continued reliance is bounded by the applicability of the premises that made the earlier decision defensible. These boundaries apply across authentication, attestation, authorization, behavior, audit, session continuity, recovery, and continuous evaluation. They also apply to policy and to the root assumptions that make further evaluation possible.

BTF organizes the twelve-paper series into five operating domains: Representation, Evaluation, Justification, Continuity, and Qualification & Time. It models a consequential decision as a lifecycle rather than a single check: Question → Proposition → Evidence → Authority → Appraisal → Policy → Decision → Consequence → Persistence → Inheritance → Material Change → Reevaluation or Recovery. Provenance, dependency awareness, uncertainty handling, and explicit root assumptions cut across the entire lifecycle.

The practical objective is not maximum evidence collection, maximum reevaluation frequency, or maximum persistence. It is bounded justified reliance: enough evidence and authority for the proposition being decided,

enough provenance to explain how the consequence followed, enough retained state to preserve required security properties, and enough reevaluation to prevent materially outdated conclusions from controlling the system indefinitely.

1. Purpose and Scope

BTF is intended to help architects, reviewers, implementers, auditors, and researchers reason about trust-relevant decisions in composed technical systems. It applies when a system accepts evidence, relies on an authority, evaluates a proposition, produces a consequential decision, or allows later activity to inherit an earlier result.

The framework is deliberately neutral about mechanism. A system may use certificates, keys, tokens, attestation, behavioral evidence, device posture, federation, local policy, trusted hardware, retained state, or other mechanisms. BTF does not prescribe which mechanism should be used. It asks what each mechanism actually establishes, who is entitled to establish it, what the receiving system adds, what consequence follows, and how long continued reliance remains justified.

The framework is especially useful at architectural boundaries: between authentication and authorization; between evidence and appraisal; between verifier and relying party; between one administrative domain and another; between an earlier session and a later action; and between pre-failure state and a recovered system.

1.1 What BTF is not

- Not a replacement for Zero Trust, PKI, RATS, OAuth, WebAuthn, SPIFFE, TUF, or other established architectures.
- Not a claim that existing standards generally fail to scope authority, reason about risk, support freshness, or manage lifecycle. Several CBRS papers explicitly rejected those broader critiques.
- Not a universal formula for confidence, assurance, freshness, or evidence sufficiency.
- Not a requirement to expose raw evidence to every downstream system.
- Not a requirement to reevaluate every premise continuously or before every action.
- Not an argument for statelessness, ephemeral identity, or elimination of credentials.
- Not a CoreBond product design or implementation disclosure. The framework is an independent research synthesis derived from the CBRS series.
- Not a correctness oracle. BTF can expose premises, authority, dependencies, uncertainty, and semantic paths; it cannot guarantee that every premise is true, detect every compromised component, or force an organization to choose sound policy.

2. Core Model: The Bounded Decision

The central object in BTF is the bounded decision. A bounded decision is a trust-relevant conclusion reached for a defined proposition, under an applicable and acceptable policy, using evidence, authorities, context, and explicitly accepted root assumptions considered sufficient for a particular consequence at a particular point in the system lifecycle.

For BTF, a decision or state is trust-relevant when it can materially affect authentication, authorization, admission, privilege, reliance, isolation, release, recovery, or another security consequence.

Bounded decision
A decision is not bounded merely because it has a timestamp or TTL. It is bounded when the architecture can explain what proposition was decided, what evidence supported it, whose authority applied, what inference policy added, what consequence followed, and what conditions end or limit later reliance.

A bounded decision may be authentication, attestation appraisal, authorization, admission, privilege activation, key release, session continuation, recovery admission, or another consequential trust decision. BTF does not collapse these decisions into a single trust score. Their evidence, authority, scope, and lifecycle may differ.

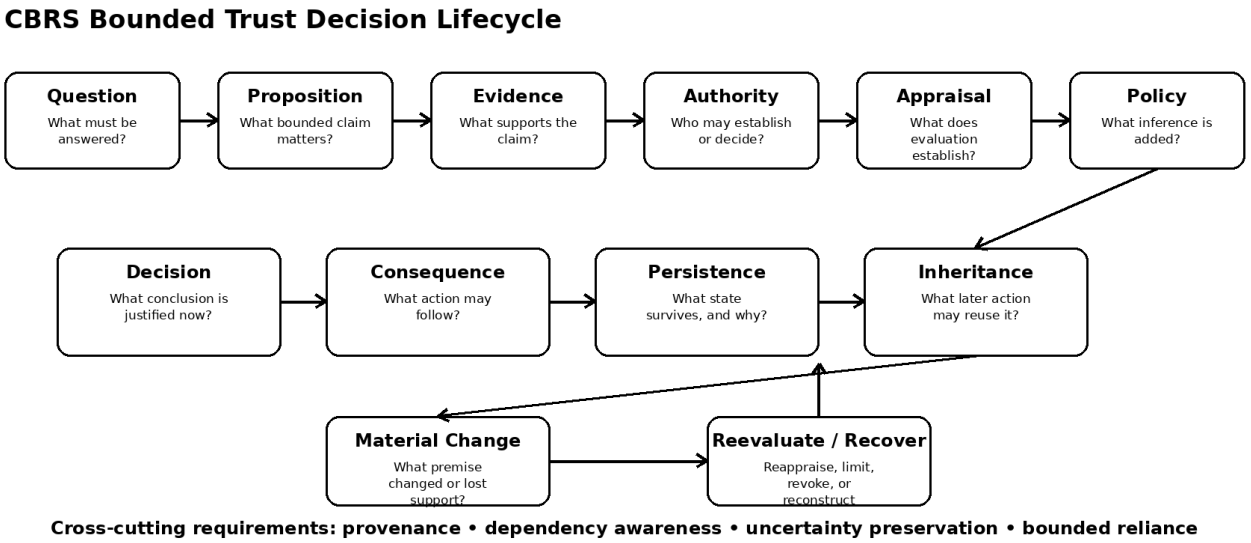

Figure 1. CBRS Bounded Trust Decision Lifecycle.

2.1 The twelve stages

Question: What security question must be answered before the contemplated action?

Proposition: What bounded fact, relationship, state, or condition must be established?

Evidence: What observations, proofs, records, measurements, relationships, or signals support that proposition?

Authority: Who or what is entitled to assert, appraise, recognize, delegate, set policy, or decide within the relevant scope?

Appraisal: What does evaluation of the evidence establish, and what remains limited, unresolved, or unsupported?

Policy: What local inference, threshold, scope, exception, or consequence rule does the receiving system add, and is the policy itself authorized, intact, current, and applicable to this decision?

Decision: What conclusion is justified now?

Consequence: What action, privilege, access, limitation, isolation, release, or other effect may follow?

Persistence: What state from the decision must survive, and what security job does that state perform?

Inheritance: What later action may reuse or rely on the earlier decision without repeating every prior step?

Material Change: What change, expiration, revocation, supersession, disagreement, policy change, loss of evidence, authority compromise, or confidence reduction can make a required premise no longer sufficient?

Reevaluation / Recovery: What must be reappraised, narrowed, revoked, reauthenticated, re-established, preserved, broken, or reconstructed?

These stages are logical accounting questions, not a required temporal execution order. Implementations may loop, branch, evaluate stages concurrently, receive evidence in different orders, revisit earlier questions as new information becomes available, or run the same bounded decision repeatedly, as a stream, or as a continuously updated process.

2.2 Cross-Cutting Accounting

Four concerns apply across every stage rather than belonging to only one box in the lifecycle:

- Provenance: where evidence, authority, policy, appraisal, and decision context came from, and what transformations occurred.
- Dependency: which services, records, control planes, authorities, devices, operators, or retained states must remain acceptable for the decision to remain defensible, including whether apparently separate components share a failure domain.
- Uncertainty: what remains unknown, unsupported, qualified, stale, unavailable, or in conflict, and whether those distinctions materially change downstream handling.
- Root assumptions: which cryptographic anchors, administrative powers, physical custody assumptions, prior states, institutional facts, or other terminal premises are accepted without further recursive proof for the decision at issue.

BTF does not try to eliminate root assumptions. Every architecture eventually rests on them. The requirement is to make the relevant roots visible enough to challenge, scope, monitor, replace, or recover when their continuing acceptability becomes suspect.

3. The Three Boundaries

BTF is built around three independent but interacting limits. The three boundaries provide a way to describe a recurring class of failures examined across the source series: meaning, authority, or lifetime is extended without the architecture making the added inference explicit.

3.1 Semantic Boundary — Meaning is bounded

Evidence and records should not be treated as establishing propositions they did not actually support. CBRS-001 shows this through actor attribution: a technically correct record identifying an executing principal does not automatically establish an initiator, delegator, approver, responsible human, or intentional participant. CBRS-004 makes the same distinction among credential, transaction proof, authentication result, and later state that inherits the result. CBRS-008 generalizes the rule: evidence can be valid without being sufficient for the proposition required by a later decision.

Semantic rule
Do not let evidence for proposition A silently become proof of proposition B. If a downstream system adds B, that addition must be attributable to policy, additional evidence, or another legitimate inference source.

3.2 Authority Boundary — Authority is bounded

An authoritative source is authoritative for something, under conditions, to a relying context. CBRS-007 rejects the broad claim that mature trust architectures generally fail to scope authority. The surviving problem is at the handoff: a receiving system may combine a legitimate upstream result with local policy and reach a stronger downstream consequence. The downstream system owns the inference it adds; the upstream authority should not be treated as though it established the stronger conclusion unless it was actually entitled to do so.

Authority rule
A validated upstream result may legitimately support a stronger downstream decision. That does not expand the upstream authority. The receiver owns what it added.

3.3 Temporal Boundary — Reliance is bounded

A correctly made decision does not acquire unlimited authority over future actions. CBRS-009 shows that retained state can remain technically valid while becoming inapplicable to the proposition a later decision tries to infer. CBRS-012 completes the lifecycle: a decision may finish, but later actions inherit it only while the premises required for those actions remain sufficiently applicable. Time is one invalidation mechanism; revocation, supersession, policy change, compromise, state change, relationship change, and loss of evidentiary support are others.

Temporal rule
Yesterday's answer may remain a correct historical record while no longer being sufficient authority for today's action.

4. Five Framework Domains

The twelve source papers can be grouped into five domains. These domains are organizational, not claims that the papers form five formal standards layers. They provide a practical way to apply the framework without losing the distinct questions each paper addressed.

Domain	CBRS papers	Primary concern	Framework output
Representation	001–003	What actor/identity state exists, what it means, and what lifecycle dependencies sustain it.	Bounded actor/identity propositions and explicit dependency surface.
Evaluation	004–006	How authentication, freshness, behavior, posture, and context become current evidence.	Evidence appropriate to the question without collapsing distinct decision types.

Justification	007–008	Whether authority is correctly scoped and evidence is sufficient for the proposition and consequence.	Explicit authority scope and evidence-to-decision accounting.
Continuity	009–010	What state may persist, what later actions may inherit, and how trust is reconstructed after failure.	Justified persistence and controlled reconstruction.
Qualification & Time	011–012	How material uncertainty survives appraisal and how long previous decisions may govern later activity.	Minimum sufficient qualification and bounded inheritance.

4.1 Representation

Representation asks what actor, device, credential, relationship, or identity-related state the architecture actually represents. CBRS-001 establishes that actor-related provenance is proposition-specific: executing principal, delegated subject, current actor, approver, initiator, and present human participant are different relationships. CBRS-002 separates manufacturer provenance, ownership, operational credential, attested state, authorization, session identity, and peer-visible identity rather than treating “device identity” as one timeless object. CBRS-003 then exposes the dependency surface required to keep whichever representation the architecture chooses usable over years of provisioning, renewal, revocation, transfer, recovery, and cryptographic migration.

Framework requirement: identify which property is being represented, its natural lifetime, who may change it, what dependencies sustain it, and what consequence is allowed to rely on it.

4.2 Evaluation

Evaluation asks how present evidence is generated and interpreted. CBRS-004 separates durable credentials from transaction-bound proof and the result produced by verification. CBRS-005 treats Zero Trust as a decision loop whose quality depends on the age, integrity, availability, and consistency of identity, posture, policy, telemetry, and enforcement dependencies. CBRS-006 shows that behavior can become actionable before complete attribution and can remain relevant after identity has been established, while refusing to treat anomaly as guilt or normal-looking activity as innocence.

Framework requirement: keep authentication, behavior, attestation, authorization, and continued-access evaluation distinguishable even when their evidence is fused in one system.

4.3 Justification

Justification asks whether the architecture is entitled to make the conclusion it makes. CBRS-007 provides the authority boundary: upstream authority and downstream consequence are different acts. CBRS-008 provides the sufficiency boundary: valid evidence may be too old, too narrow, irrelevant to a required proposition, or transformed in a way that hides what the original evidence actually supported.

Framework requirement: preserve the semantic path from evidence to proposition to appraisal to policy inference to consequence; preserve which authority owned each step; and treat policy itself as a trust-bearing input with provenance, authority, integrity, version, applicability, and recovery requirements.

4.4 Continuity

Continuity asks what survives after the immediate decision is over. CBRS-009 rejects both "keep everything" and "statelessness is always safer." Some security properties depend on remembering prior use, revocation, ordering, relationships, or history. Persistence is therefore justified by function. CBRS-010 addresses the failure case: restoration does not automatically re-establish trust. Recovery separates what may be preserved, what prior trust continuity must be broken, and what properties of the recovered present need a current basis.

Framework requirement: every retained trust-relevant state item should have an identifiable security job and an end condition; every recovery path should distinguish restoration from trust re-establishment.

4.5 Qualification & Time

Qualification & Time asks what the system knows, what it does not know, and how long an earlier answer remains usable. CBRS-011 distinguishes adverse evidence from failure to establish a required claim, while allowing both to produce the same immediate enforcement action when policy demands it. CBRS-012 then treats continuous evaluation as bounded inheritance rather than constant recalculation. Events expose some changes, timers bound some blind windows, and neither alone proves continued applicability.

Framework requirement: preserve only those evidentiary qualifications that materially change authorized downstream treatment, and intentionally bound how long materially outdated decisions can continue influencing enforcement.

5. Anti-Collapse Rules

BTF does not forbid composition. Modern security systems must combine evidence, authority, identity, context, and policy. The framework instead forbids silent category collapse: one technical object or result should not be treated as though it answered every related question.

Evidence is not a verdict: Technical validity does not by itself establish sufficiency for every downstream proposition.

Identity is not behavior: Recognizing an actor does not prove its current activity is acceptable; behavior does not automatically establish identity or intent.

Authentication is not authorization: Satisfying an authentication method does not automatically establish permission for every resource or later action.

Memory is not authority: Retained state can remain intact while losing permission to carry a proposition forward.

Restoration is not trust: Reconstructing historical state does not automatically justify the trust-bearing relationships of the recovered present.

Uncertainty is not permission: Preserving an "unknown" or "not established" condition does not require softer enforcement.

More evidence is not automatically better: Additional signals are useful only when they add relevant coverage, freshness, corroboration, failure resistance, or another decision-relevant property worth their cost.

Continuous is not constant: Ongoing security can use sessions, events, timers, monitoring, and per-request decisions without recomputing every premise at every instant.

6. Decision Provenance

Decision provenance is the framework spine. CBRS-008 introduced the plain-language requirement to preserve enough information about a consequential trust decision to distinguish what evidence was accepted, what proposition that evidence supported, what appraisal or local policy added, and what consequence followed. BTF extends that path by attaching authority, policy provenance and version, root assumptions, persistence, inheritance, disagreement, and later invalidation to the same reasoning chain. Decision provenance should preserve the minimum semantic path necessary for the authorized purpose; it is not a requirement to retain every raw signal, persistent identifier, or otherwise unnecessary sensitive detail.

Decision provenance does not require raw evidence retention everywhere. The minimum is semantic accountability sufficient for the authorized decisions the system needs to make and later explain, with data collection and retention minimized to what materially serves that purpose.

6.1 Minimum decision-provenance record

- Decision identifier or traceable decision context, including the relevant policy/context version where materially useful.
- Bounded proposition being evaluated.
- Evidence or evidence result accepted for the decision, at the abstraction level appropriate to the system.
- Source and authority scope relevant to that evidence or result.
- Appraisal outcome, including material qualification or limitation.
- Policy or local inference that converted the appraisal into the final decision, including the policy authority and version or decision-context identifier when relevant.
- Consequence authorized, denied, limited, isolated, released, or otherwise produced.
- Persistence or session state created by the decision, if any.
- Scope and conditions under which later actions may inherit the result.
- Known invalidators, expiry bounds, disagreement conditions, or events that require reevaluation or reconciliation.
- Recovery, reconciliation, or re-establishment requirement when continued reliance fails or material conflict cannot be resolved in place.

Root assumptions and shared failure domains material to the decision.

A system may compress, sign, distribute, or selectively retain this information. BTF does not prescribe a schema. The test is whether the semantic boundaries survive composition.

7. Uncertainty and Minimum Sufficient Qualification

BTF separates appraisal state from enforcement action. A verifier may know that evidence positively contradicts policy, that the evidence is insufficient in scope or freshness, that no valid result arrived within the policy window, or that two recognized sources materially disagree. A policy enforcement point may deny in all of these cases. The binary action is not the problem. The problem appears only when a distinction needed by an authorized downstream process is erased before that process can use it.

Materiality test
Before collapsing two appraisal outcomes, ask whether an authorized downstream component would choose a different remediation, retry, isolation, recovery, audit treatment, privilege level, or later handling if it knew the difference. If not, collapse may be harmless. If yes, preserve the distinction until that decision boundary has used it.

Qualification must not become a downgrade mechanism. Attackers can manufacture ambiguity by blocking telemetry, delaying evidence, or targeting the appraisal path. “Unknown” is not entitled to a gentler outcome. BTF preserves meaning for legitimate downstream policy; it does not prescribe permissiveness.

7.1 Disagreement and Reconciliation

Conflict is not identical to uncertainty. Two appraisers or policy views can each be internally valid and highly confident while still disagreeing about a proposition or consequence. When the difference materially changes downstream handling, BTF requires the conflict to remain visible until the component authorized to reconcile it has acted.

BTF does not prescribe which source wins. The architecture should identify the reconciliation authority, the rule or policy governing conflict, which actions may proceed while views diverge, how long divergence may persist, and what happens when the system cannot determine which view is current or controlling.

8. Justified Persistence and Bounded Inheritance

Persistence is justified when retained state continues serving a defined security property. CBRS-009 demonstrates several different jobs: carrying an earlier decision forward, remembering distrust or replay, establishing ordering, preserving auditable history, and representing continuing relationships. These jobs have different retention requirements and different risks.

A retained object should therefore be analyzed by function rather than by storage type. A session token, revocation record, version counter, audit entry, ownership record, and recovery anchor may all persist, but the proposition each is entitled to carry forward is different.

Bounded inheritance is the rule governing later reuse. A later action may inherit an earlier decision only while the premises required for the later action remain sufficiently applicable. The architecture should define known invalidators where practical and independent bounds where observability is incomplete.

8.1 The blind window

CBRS-012 uses blind window for the actual or bounded interval during which a materially outdated decision can continue influencing enforcement after a relevant premise changes but before the architecture detects, communicates, appraises, decides on, or enforces the appropriate response. BTF treats this as an architectural exposure, not a universal metric with one acceptable value.

A useful review question is not “How often do we evaluate?” but “For this consequence, how long could a materially outdated decision still control the system, and why is that bound acceptable?”

9. Controlled Reconstruction

Recovery is the point at which the framework must decide what survives a damaged trust context. BTF adopts the controlled reconstruction reasoning exposed in CBRS-010. The recovery process separates three decisions: Preserve, Break, and Re-establish.

Preserve	Break	Re-establish

Historical or operational state whose continued use remains justified for the recovery purpose.	Trust-bearing continuity whose authority is suspect: sessions, credentials, keys, relationships, approvals, or comparable state.	Present-tense properties the failure could have changed: current control, integrity, posture, authorization, or other required claims.
Examples: validated backup data, authorized recovery image, historical identity records, approved reference values.	Examples: compromised credentials, stale sessions, suspect trust paths, exposed signing material, previous owner authority.	Examples: new authentication, re-registration, integrity validation, fresh attestation, new keys, policy reapproval.

A clean rebuild is not state-free; it inherits installation media, operators, provisioning authority, roots, policies, and procedures. BTF therefore rejects both blind restoration and the assumption that starting over eliminates inherited trust. Recovery remains an evidence and authority problem.

10. BTF Decision Test

The Decision Test is the primary operational instrument of BTF. It can be used during architecture design, threat modeling, security review, incident analysis, or standards mapping.

1. What exact security question must be answered before this consequence is allowed?
2. What bounded proposition must be established?
3. Which actor, device, workload, relationship, state, or behavior does that proposition concern?
4. What evidence supports the proposition, and what does that evidence explicitly not establish?
5. Who or what is authoritative for the evidence, appraisal, policy, reconciliation, and final consequence?

Which root assumptions make those authorities and decision paths acceptable without further recursive proof?

6. Is the evidence sufficiently valid, relevant, fresh, covering, and appropriately sourced for this proposition?

Do apparently independent evidence sources, authorities, or enforcement components share a common failure domain that could invalidate their apparent independence?

7. What inference does local policy add beyond the upstream evidence or result, and is that policy itself authorized, intact, current, and applicable?

If recognized appraisals, policy views, or authorities disagree, who is authorized to reconcile the conflict, what rule applies, and what may proceed while disagreement remains?

8. What consequence follows, and is that consequence inside the scope of what was actually established?
9. What state from this decision persists, what security job does it perform, and what proposition may it carry forward?
10. Which later actions may inherit the decision without full repetition?
11. What material changes, revocations, expirations, supersessions, or confidence-loss conditions end or narrow that inheritance?
12. How long could an invalidating change remain invisible or unenforced for this consequence?
13. What material uncertainty must survive to a downstream decision boundary, and what uncertainty can safely be collapsed?
14. If continued reliance fails, what may be preserved, what must be broken, and what must be established again?
15. Can a later reviewer reconstruct enough of the semantic path to explain why this consequence followed?

11. Architecture Review Worksheet

The following worksheet converts the Decision Test into a compact review artifact. One row should be completed for each consequential trust decision or class of materially similar decisions.

Field	Review entry
Decision / consequence	
Decision context / version	
Required proposition	
Evidence / result	
Authority scope	
Root assumptions / shared failure domain	
Appraisal qualification	
Conflict / reconciliation rule	
Policy inference / applicability	
Policy authority / version	
Persistence created	
Inheritance scope	
Invalidators / bounds	
Recovery action	

The worksheet is intentionally descriptive rather than scored. A numeric score could hide qualitatively different weaknesses: excellent evidence with wrong authority, strong authority with stale evidence, correct appraisal with over-broad consequence, or sound initial decision with unbounded later inheritance.

12. Framework Alignment Principles

BTF v1.2 does not define formal certification or testable conformance criteria. It defines review principles that allow an architecture to describe how it aligns with the framework. Formal conformance should be reserved for a later profile with explicit pass/fail requirements and test procedures.

P1 — Proposition specificity: Consequential decisions identify the proposition actually being established rather than relying on vague labels such as trusted, authenticated, compliant, or known.

P2 — Evidence boundedness: Evidence and results are not treated as establishing claims outside their supported scope.

P3 — Authority boundedness: Sources, verifiers, issuers, policy owners, and relying systems operate within identifiable authority scopes.

P4 — Receiver accountability: A receiving system owns the inference and consequence introduced by its local policy.

P5 — Decision provenance: The semantic path from evidence to consequence remains explainable at the level required by governance and recovery.

P6 — Justified persistence: Retained state has an identifiable security purpose, lifecycle, and end condition.

P7 — Bounded inheritance: Later actions do not inherit prior decisions indefinitely or beyond the premises required for those actions.

P8 — Material uncertainty preservation: Evidentiary distinctions that materially change authorized downstream handling survive until the responsible decision boundary uses them.

P9 — Proportionate response: Evidence and response requirements should be proportionate to the proposition being established, the consequence and reversibility of the action, the cost of error in both directions, and the operational cost of obtaining additional evidence.

P10 — Controlled reconstruction: Recovery distinguishes restoration of prior state from re-establishment of present trust.

P11 — Policy trustworthiness: Policy is treated as a trust-bearing dependency whose authority, integrity, version, applicability, and recovery matter to the legitimacy of the decision.

P12 — Root-assumption exposure: Terminal assumptions material to a consequential decision are identified rather than hidden behind infinite recursive claims of trust.

P13 — Reconciliation accountability: Material disagreement among recognized sources, appraisals, or policy views remains visible until an identified authority and rule have resolved or bounded the conflict.

13. Common Failure Patterns

Semantic over-attribution: An audit, authentication, attestation, or identity result is later interpreted as proving a stronger actor or system proposition than it actually established.

Authority laundering: A downstream system cites an upstream authoritative result as though the upstream source itself authorized the downstream consequence.

Validity/applicability confusion: A technically valid token, record, session, or result is reused for a proposition it is no longer entitled to establish.

Freshness theater: A fast decision loop consumes stale inputs and is described as continuous merely because the policy engine runs frequently.

Signal accumulation: More telemetry is added without identifying what proposition, coverage, or independent failure resistance it contributes.

Premature semantic collapse: Qualified appraisal states are flattened before a downstream policy that legitimately distinguishes them can act.

Persistence by accident: State survives because implementation retains it, not because the architecture can explain the security property the state serves.

Blind restoration: Recovered historical state is treated as present trust without breaking suspect continuity or re-establishing changed properties.

Unbounded session inheritance: Later actions continue relying on a prior authentication or decision without meaningful scope, invalidators, or independent bounds.

Fail-open/fail-closed sloganism: Availability and consequence are reduced to a universal default rather than being governed by resource- and mission-specific policy.

Policy-as-oracle failure: A decision is treated as justified merely because it followed policy even though the policy itself may be stale, compromised, unauthorized, or misapplied.

Hidden-root recursion: Authority is justified by another authority indefinitely while the actual terminal assumptions—cryptographic, administrative, physical, or institutional—remain unnamed and therefore difficult to challenge.

Unresolved split-brain: Conflicting but individually valid appraisals or policy views exist without an identified reconciliation authority, divergence bound, or rule for what may proceed while disagreement remains.

14. Applicability Across Architectures

BTF is designed to sit above specific mechanisms. The same review can be applied to a certificate-based device, a remote-attestation system, a federated identity flow, a Zero Trust policy decision, a behavioral detection pipeline, an OT recovery path, or a session-continuity mechanism. The questions are intended to remain useful even when the evidence and authority models change; external mapping must still test the framework against architectures with different policy, root-assumption, distribution, and reconciliation models.

For example, a certificate-based architecture can use BTF to distinguish certificate validity from current device state and authorization. A RATS architecture can use it to separate Evidence, Verifier appraisal, Attestation Result, relying-party inference, and consequence. A Zero Trust implementation can use it to expose stale posture, policy, or enforcement dependencies. An OT system can use it to reason about the consequence of unavailable evidence and the legitimacy of bounded local authority during degraded operation. The framework does not force these systems into one mechanism; it gives them one reasoning discipline.

15. Source Traceability: CBRS-001 through CBRS-012

The framework is a synthesis, not a thirteenth research paper. The table below identifies the primary contribution each frozen paper makes to BTF. Version 1.2's explicit root-assumption and disagreement accounting trace primarily to CBRS-005, while its policy-trust treatment follows the same dependency reasoning across CBRS-005, CBRS-007, CBRS-010, and CBRS-012. This mapping is intentionally explicit so that framework claims can be challenged against the source record rather than defended by retrospective storytelling.

Paper	Title	Primary framework contribution

CBRS-001	Actor Governance: What Does the Audit Record Actually Establish?	Proposition-specific actor provenance; semantic over-attribution; accountability evidence should support the actor relationship actually claimed.
CBRS-002	The Device Should Not Know Its Own Identity	Identity-related properties have different lifetimes, visibility, ownership, recovery, and persistence requirements.
CBRS-003	The Hidden Lifecycle Cost of Device Credentials	Trust mechanisms carry dependency surfaces and lifecycle obligations that must remain survivable and recoverable.
CBRS-004	Authentication: Credential, Proof, Result, and Time	Separate durable credential, current proof, authentication result, and later inherited state; ask again when the inherited result is no longer sufficient.
CBRS-005	Zero Trust: Decision Dependencies, Freshness, and Failure	Decision quality depends on the currency, integrity, availability, and consistency of identity, policy, telemetry, and enforcement dependencies.
CBRS-006	Behavior Before Identity	Behavioral evidence can justify proportionate reversible response before complete attribution; anomaly is evidence, not verdict.
CBRS-007	Authority Has a Boundary	Upstream authority scope and downstream policy consequence are distinct; the receiver owns the inference it adds.
CBRS-008	Evidence Is Not a Verdict	Validity is not sufficiency; decision provenance and evidence-to-decision accounting preserve the semantic path.
CBRS-009	Memory Is Not Authority	Persistence must be justified by security purpose; retained state may remain valid while becoming inapplicable.
CBRS-010	Recovery	Controlled reconstruction: preserve what remains justified, break suspect continuity, and re-establish present-tense properties.

CBRS-011	Uncertainty	Preserve material appraisal distinctions until the authorized downstream decision boundary has used them; uncertainty does not imply permission.
CBRS-012	Continuous Evaluation: When Is Trust Finished?	Bounded inheritance, material invalidators, blind windows, and premise-directed reevaluation complete the decision lifecycle.

16. Framework Research Agenda

Version 1.2 intentionally leaves several engineering questions unresolved. The framework is strongest as a reasoning model today; future work should test whether its concepts can support practical metrics, interoperability profiles, or assurance evidence without collapsing the distinctions the framework was built to preserve.

- Can decision provenance be represented in interoperable form without forcing every relying system to retain raw upstream evidence?
- Can blind-window exposure be measured usefully across sensing, event delivery, appraisal, policy, propagation, and enforcement without creating false precision?
- How should architectures express authority scope and proposition scope at system boundaries so downstream components do not silently over-infer?
- What practical methods can test evidence independence, shared failure domains, and dependency concentration without reducing them to raw signal counts?
- How can machine and device systems define bounded inheritance rules where human session guidance does not directly apply?
- How should safety-critical and disconnected environments define acceptable local authority when current remote evidence is unavailable?
- What lifecycle metrics best compare credential, verifier-held, relational, hardware-derived, behavioral, and hybrid trust architectures while charging each for equivalent recovery and revocation controls?
- Can controlled reconstruction be formalized enough to support repeatable recovery review across identity, firmware, policy, and distributed trust systems?
- Which BTF concepts are already represented in existing standards vocabularies and which require only cross-standard mapping rather than new terminology?
- How should architectures document terminal root assumptions without creating misleading claims that every root can itself be recursively verified?
- What interoperable patterns can represent material disagreement and reconciliation authority without forcing every system into one universal conflict-resolution rule?
- How should policy provenance, version, authority, and recovery be carried across composed systems without making policy distribution itself a new single point of failure?

17. Conclusion

The CBRS series began with separate questions about actors, device identity, credentials, authentication, Zero Trust, behavior, authority, evidence, memory, recovery, uncertainty, and continuous evaluation. The

research repeatedly rejected broad claims when existing architectures already handled them. What survived those corrections was not a new identity mechanism. It was a common architectural discipline.

A consequential trust decision should be able to explain what proposition was established, what evidence supported it, whose authority applied, what policy added and why that policy was itself acceptable, what root assumptions remained, what consequence followed, what state persisted, what later action inherited the result, what change or disagreement ends that inheritance, who reconciles material conflict, and how trust is reconstructed when prior continuity can no longer be defended.

That discipline can still be reduced to three boundaries: meaning is bounded, authority is bounded, and continued reliance is bounded. Version 1.2 makes the supporting accounting more explicit: provenance, dependency, uncertainty, root assumptions, policy trust, and reconciliation keep those boundaries visible as systems compose, disagree, persist, change, fail, and recover.

BTF v1.2
Trust can be decided. What cannot be granted by default is unlimited meaning, unlimited authority, or unlimited future reliance on yesterday's decision.

Source Papers

Kirton, T. (2026). CBRS-001: Actor Governance: What Does the Audit Record Actually Establish? CoreBond Research Series.

Kirton, T. (2026). CBRS-002: The Device Should Not Know Its Own Identity. CoreBond Research Series.

Kirton, T. (2026). CBRS-003: The Hidden Lifecycle Cost of Device Credentials. CoreBond Research Series.

Kirton, T. (2026). CBRS-004: Authentication: Credential, Proof, Result, and Time. CoreBond Research Series.

Kirton, T. (2026). CBRS-005: Zero Trust: Decision Dependencies, Freshness, and Failure. CoreBond Research Series.

Kirton, T. (2026). CBRS-006: Behavior Before Identity. CoreBond Research Series.

Kirton, T. (2026). CBRS-007: Authority Has a Boundary. CoreBond Research Series.

Kirton, T. (2026). CBRS-008: Evidence Is Not a Verdict. CoreBond Research Series.

Kirton, T. (2026). CBRS-009: Memory Is Not Authority. CoreBond Research Series.

Kirton, T. (2026). CBRS-010: Recovery. CoreBond Research Series.

Kirton, T. (2026). CBRS-011: Uncertainty. CoreBond Research Series.

Kirton, T. (2026). CBRS-012: Continuous Evaluation: When Is Trust Finished? CoreBond Research Series.

Appendix A — BTF Quick Review Card

Use this compact form when a full architecture worksheet is unnecessary.

Prompt	Answer
QUESTION	What are we deciding?
PROPOSITION	What must be true?

EVIDENCE	What supports it?
AUTHORITY	Who may establish it, and what root assumptions support that authority?
ROOT ASSUMPTION	What terminal premise are we accepting without further recursive proof?
APPRAISAL	What did evaluation actually establish?
POLICY	What did we add, and is the policy itself authorized, current, and applicable?
CONFLICT	Do recognized sources disagree, and who reconciles them?
CONSEQUENCE	What are we allowing or denying?
PERSISTENCE	What are we remembering, and why?
INHERITANCE	What later action may reuse this?
INVALIDATOR	What ends that right?
BLIND WINDOW	How long could change remain invisible?
RECOVERY	What survives, what breaks, what must be established again?