Continuous Assurance Fabric Reference Architecture (CAF-RA)

A Control-Plane Architecture for Governed, Scalable Continuous Assurance for Agentic Operations at Mission Tempo

Adam Boas

Agentic Warfare Architect, Department of War

February 2026

Contents

Executive Summary

Mission demands are pulling the Department of War (DoW) toward mission-tempo operations. The environment changes continuously, while governance remains uneven, episodic, and often retrospective. Dependencies mutate, identities drift, and adversaries adapt on timescales that outpace periodic review cycles. Programs still express authority as point-in-time approvals, and teams still assemble assurance as snapshots; that gap is the operational risk.

Automation already drives consequential change in limited, uneven pockets of the enterprise, and agentic systems are poised to expand the set of non-person principals that can propose—and in some cases execute—bounded actions. In this paper, *agentic operations* means non-person principals that can propose, adjudicate, and request bounded changes via governed tool calls, with all side effects mediated by enforcement points and recorded as decision/action evidence. In that world, point-in-time authorization becomes a category error. Cyber resilience is not a milestone; it is a continuously maintained operating state.

CAF-RA defines the assurance control plane that makes that state governable at mission tempo. CAF does not assume today's enterprise is continuously governed; it defines the control-plane mechanics required to govern continuous, agentic operations without relying on periodic re-approval.

CAF is a single governed loop—EMIT → ENRICH → SCORE → SURFACE → ACT—where each verb is a responsibility boundary, a contract surface, and an audit trail requirement. The loop converts raw telemetry into decision-grade assurance and binds decisions to enforceable authority at the change boundaries where reality becomes irreversible. CAF is designed for end-to-end inspection:

authority → evidence → posture/confidence → decision record → enforcement → verification

CAF-RA enforces three non-negotiables. Authority is explicit and executable through signed manifests and policy bundles. Evidence is portable, labeled, and replayable through sealed envelopes and tamper-evident anchoring. Consequential action is mediated and attributable at real change boundaries through signed decision records, not inferred from advisory dashboards.

This paper keeps the baseline simple and pushes sophistication into conformance profiles. It defines testable technical positions, a governance overlay, degraded-mode behavior, and an adoption path that starts in shadow mode and graduates to bounded automation. The objective is practical: sustain continuously governed cyber resilience at mission tempo without sacrificing control, attribution, or rollback discipline [1, 2, 3].

1 Continuous Assurance Fabric Reference Architecture (CAF-RA)

A reference architecture for governed, scalable continuous assurance—designed for mission tempo, federated enclaves, and contested/degraded operations.

CAF is written for an agentic enterprise. As autonomy shifts from human operators to non-person principals, governance cannot remain a periodic documentation workflow or a monitoring program. CAF treats assurance as an executable control plane that constrains autonomous change at runtime boundaries, preserves replayable proof of what was known and done, and keeps human authority relevant by making it enforceable at machine tempo.

Normative language: "MUST / SHOULD / MAY" indicate requirements, recommendations, and options.

CAF keeps baseline complexity to a **single governed loop**; additional capabilities are implemented as **profiles** and **patterns**, not as baseline ontology.

1.1 CAF-RA Overview

CAF is a control plane for assurance. It exists because the unit of operational change is now smaller than the unit of human governance, and because an increasing share of that change is initiated by non-person principals operating at machine tempo.

When code ships hourly, posture shifts continuously, and agents can propose and execute bounded actions, assurance cannot remain a point-in-time document workflow without becoming either irrelevant (rubber-stamp) or destructive (paralyzing friction). In a continuous enterprise, authorization is not an event; it is a continuously maintained state produced by governed evidence, explicit authority, and mediated enforcement.

CAF solves that by standardizing a single operational loop:

EMIT → ENRICH → SCORE → SURFACE → ACT

This loop is not a metaphor. It is the architecture. Each verb names a responsibility boundary, a contract surface, and an audit trail requirement. If CAF is implemented as ten services or fifty does not matter; what matters is that the five responsibilities remain distinct and governed, and that they produce evidence and replay sufficient to sustain continuous authorization state and after-action reconstruction. [1]

CAF is deliberately aligned to the same "control shift" posture found in *Stand Up Delegated Autonomy Directorate (DAD) as a Joint Control Plane Authority* and ACP-RA: this is not a tooling cycle; it is a shift in how authority is executed when non-person entities act at machine tempo. CAF provides the assurance control plane that keeps autonomous execution governable, with acquisition leverage and conformance as the mechanism that prevents fragmentation. [2, 1]

1.2 Operational Problem Statement

The operating environment is continuous: dependencies mutate, identity posture changes, configuration drifts, network policy evolves, detection logic changes, and adversaries move faster than governance cycles. The problem is not that we lack evidence. The problem is that evidence is rarely **decision-grade**: it is not attributable to strong identity, not schema-consistent, not labeled for cross-enclave operation, not confidence-scored, and not bound to enforceable authority.

CAF makes assurance decision-grade by enforcing three non-negotiables that remain valid even as autonomy increases. Authority is explicit and enforceable: human risk authority is represented as a signed, versioned manifest and signed policy bundles that constrain what actions are allowed, in what environment, at what consequence tier, and under what uncertainty tolerance. This is the

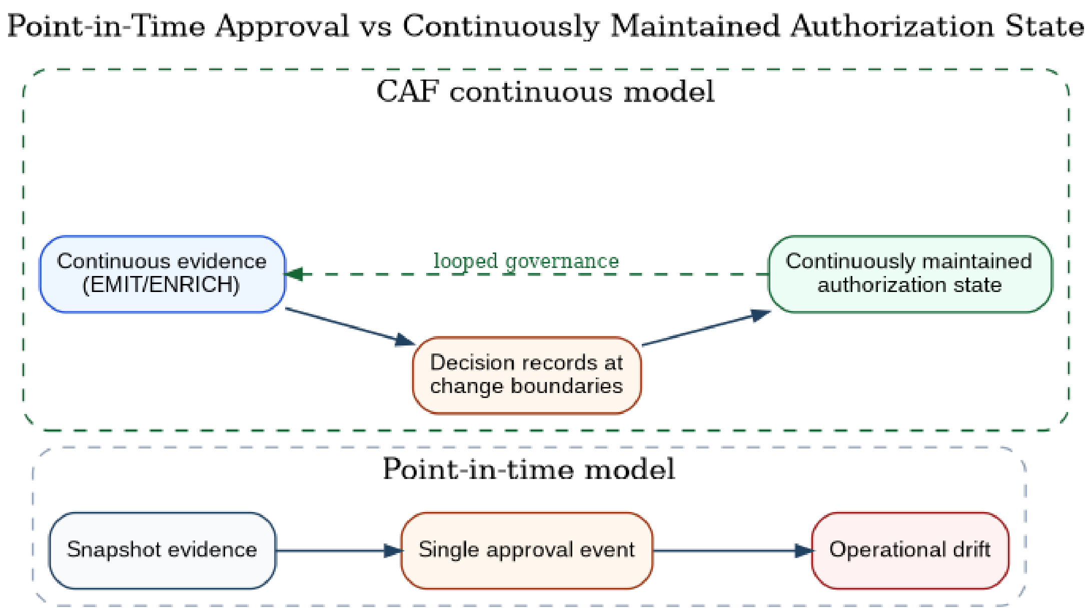

Point-in-Time Approval vs Continuously Maintained Authorization State
CAF continuous model
Continuous evidence (EMIT/ENRICH)
looped governance
Continuously maintained authorization state
Decision records at change boundaries
Point-in-time model
Snapshot evidence
Single approval event
Operational drift

Figure 1: Point-in-time approval drifts; CAF maintains authorization state continuously at governed change boundaries

assurance analogue to ACP-RA's trust-scope posture: intent is not authority; manifests and policy are. [1]

Evidence is portable and replayable: evidence is emitted in sealed envelopes, admitted through an ingestion gateway, anchored in a tamper-evident mechanism, and queryable by stable IDs. If you cannot reconstruct what was known at the time and bind it to who asserted it, you cannot credibly claim you governed the decision—especially when agents and automation act faster than human recollection. [1]

Enforcement is mediated and attributable: CAF does not "recommend" safety. It makes safety enforceable at the boundaries where change becomes real—pipeline gates, admission gates, segmentation gates, data-access gates, and bounded response automation—under signed decision records that bind policy hash, manifest hash, and evidence references. Centralize decisions; distribute enforcement. [1]

Everything else—risk graphs, ensemble scoring, digital twins, formal policy coherence checks, autonomous validation agents—can and should exist, but they belong in profiles and patterns. They must not bloat the core mental model, and they must never become accidental authority sources.

1.3 Scope and Non-Goals

Before debating design details, we bound the problem. CAF-RA focuses on the control-plane mechanisms that make assurance governable at scale and at tempo. It guides and constrains downstream architectures; it does not prescribe a single implementation. [1]

CAF-RA is **in scope** for enterprise cloud and platform services, software factories and delivery pipelines, mission runtimes (containers/VMs/managed services), tactical/edge deployments (including disconnected operations), multi-enclave operation (including cross-domain posture exchange), and integration with operations centers, change control, and audit/assurance functions.

CAF-RA is **out of scope** for selecting vendors, mandating a single scoring algorithm, or defining mission tactics. Where CAF integrates into safety-critical or mission-critical effects chains, additional safety doctrine applies; CAF's role remains the same: make authority explicit, mediate actions, and preserve evidence and replay.

1.4 Strategic Drivers

CAF is shaped by constraints, not trends.

A posture of acceleration. When the Department adopts a posture of acceleration, the architecture must assume high change rates rather than "best-effort modernization." A control plane that cannot keep pace will be bypassed.

Agentic execution changes the identity of "who changes the enterprise." Non-person principals can generate, configure, and remediate continuously, sometimes faster than humans can observe, much less convene. Governance that assumes humans initiate consequential change will either be routed around or will collapse operations into serialized approvals. CAF exists to keep autonomous action governable by pre-binding authority in enforceable artifacts and mediating side effects at real boundaries.

Agentic execution will not arrive as a single "agent." It will arrive as populations of specialized non-person principals: validators, triage agents, detection-tuning agents, remediation planners, and bounded effectors. In that environment, resilience cannot depend on the correctness of any one producer or one model. CAF therefore treats multiplicity as a security primitive: posture and action eligibility should be derived from independent principals, and adjudicated for agreement, disagreement, and outliers. Consensus does not grant authority—manifests and policy bundles remain the only authority sources—but consensus raises confidence and increases adversary cost by forcing corruption across multiple independent paths to move governance outcomes.

Continuous monitoring alone cannot produce assurance in a machine-tempo enterprise. Monitoring observes after-the-fact; assurance governs before-the-fact by constraining what actions are eligible under what authority and what confidence. CAF therefore treats monitoring as evidence input to a governed loop, not as a substitute for decision records, enforcement points, and replayable proof.

The core problem shifts from "can we collect evidence?" to "can we decide with evidence at tempo?" Evidence is abundant. Decision-grade evidence is not. CAF exists to transform raw telemetry into attributable, replayable, confidence-scored artifacts that can safely gate change and response.

Contested and degraded environments are the baseline, not the exception. CAF assumes partial connectivity, delayed evidence arrival, time skew, sensor failure, and deception attempts. The control plane must degrade safely without collapsing into either paralysis or optimism.

Interoperability and federation are unavoidable. The Department is federated. CAF must be protocol-neutral and vendor-neutral while enforcing a consistent policy surface across enclaves and programs. That consistency is achieved through contracts and conformance, not through central procurement of a single toolchain. [2]

1.5 Architectural Principles

CAF-RA adopts the same architecture posture used in ACP-RA and related papers: explicit authority artifacts, mediated side effects through gateways, governance-as-code with gates, evidence and replay, and degraded-mode survivability. [1]

1.5.1 P1 — Producers and Enforcers Are Principals (Including Agents), Not "Tools"

Evidence producers and enforcement points are actors with identity, attributes, credential lifecycle, and accountability. This includes autonomous validation agents and other agentic producers whose outputs may be policy-relevant. Anonymous producers create anonymous evidence; anonymous evidence creates unauditable governance; and unauditable governance cannot safely authorize machine-tempo action.

1.5.2 P2 — Centralize Policy Decisions; Distribute Enforcement

CAF uses a policy decision function and multiple enforcement points at the boundaries where change becomes real. Central decisioning without enforcement is theater; enforcement without centralized decisioning is fragmentation. [1]

1.5.3 P3 — Default Deny for Consequential Change; Allow by Explicit Tolerance

Change is permitted because authority artifacts and policies allow it under defined conditions. When authority is ambiguous or confidence is low, CAF fails safe: block, contain, or escalate.

1.5.4 P4 — The "Change Boundary" Is Explicit

Systems may compute posture, propose remediation, or recommend exceptions. They do not execute side effects directly. All consequential effects are mediated through gatekeepers that enforce policy and record evidence.

1.5.5 P5 — Evidence and Replay Are First-Class

CAF must support deterministic reconstruction: what was known, what was decided, what was done, and under what authority. Without replay, continuous authorization collapses into narrative. [1]

1.5.6 P6 — Rollback and Containment Are Routine Capabilities

Speed wins only when reversal is fast and scoped. CAF treats rollback, quarantine, and compensating controls as normal operations, not heroic improvisations.

1.5.7 P7 — Context Is Governed Data Movement

Context influences decisions; it must not grant authority. Context must be provenance-preserving, minimized, labeled, freshness-aware, and replayable. This is the assurance analogue to “context engineering is governed data movement.” [1]

1.5.8 P8 — Evaluation and Validation Are Gates

Tests, drift checks, detection-fidelity checks, and adversarial validation are not reports. They are release gates and runtime gates. [1]

1.5.9 P9 — Confidence Is Explicit; Uncertainty Drives Safe Behavior

CAF treats confidence as policy-relevant input. Evidence completeness, provenance, freshness, and consistency drive whether CAF allows action or escalates.

1.5.10 P10 — Interfaces Are Government-Owned; Components Are Replaceable

CAF is an architecture, not a product. Replaceability is enforced through stable schemas, API contracts, and conformance tests—consistent with the broader “code-as-policy / governance-as-code” posture. [3]

1.6 Reference Architecture Structure

CAF-RA is organized as: strategic purpose, principles, technical positions, patterns, and vocabulary. The loop provides the narrative spine; technical positions define the hard edges that make the loop governable; patterns show how to implement without prescribing a vendor stack. This is deliberately congruent with ACP-RA’s structure and tone. [1]

1.7 Conceptual View: CAF as a Governed Loop

CAF is a control system with two planes: the runtime plane, where the enterprise actually changes (factories, registries, clusters, identity, networks, data stores, mission services), and the assurance control plane, where authority, evidence, posture, decisions, and enforcement are mediated and proven.

Here’s the minimal mental model; everything in CAF-RA is a refinement of these boxes, not an expansion into a new ontology:

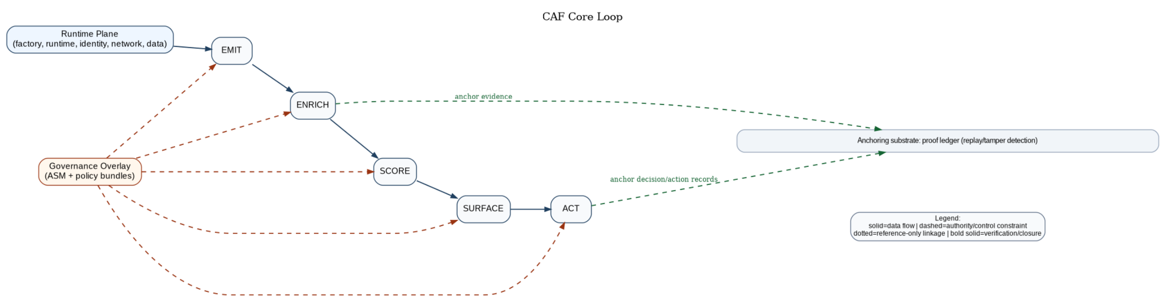

CAF Core Loop
Runtime Plane
(factory, runtime, identity, network, data)
EMIT
ENRICH
SCORE
SURFACE
ACT
Governance Overlay
(ASM + policy bundles)
anchor evidence
Anchoring substrate: proof ledger (replay/tamper detection)
anchor decision/action records
Legend:
solid=data flow | dashed=authority/control constraint
dotted=reference-only linkage | bold solid=verification/closure

Figure 2: CAF core loop with complete governance touchpoints and proof-ledger anchoring substrate

1.8 Core Vocabulary

CAF keeps vocabulary intentionally small. If the first ten pages introduce thirty nouns, the architecture will feel like bureaucracy.

Assurance Scope Manifest (ASM). A signed, versioned artifact that represents delegated authority for assurance actions within a scope. It binds four parameters that must be explicit and testable: **Authority** × **Consequence** × **Environment** × **Uncertainty tolerance**. This structure mirrors the trust-scope decomposition used in the related papers, but applied to assurance. [4]

Policy bundle. A signed, versioned set of executable rules consumed by CAF decisions and enforcement points. Policy bundles encode ABAC constraints, gate conditions, exception constraints, degraded-mode rules, and action eligibility.

Evidence envelope. A sealed wrapper around an evidence event that binds producer identity, schema reference, time and clock confidence, handling labels, integrity hashes, anti-replay fields, and pointers to raw artifacts. For agentic producers, the envelope also binds model/toolchain identifiers and explicit references to the governed context and tool outputs used to form the claim, so that after-action reconstruction remains possible.

Decision record. A signed record binding policy hash + ASM hash + evidence references to a decision (permit, deny, contain, require approval, advisory), with confidence, expiry, and pointers sufficient to replay the decisioning function that produced the verdict.

Enforcement point (EP). A mediated boundary where decisions become real effects: merge gates, promotion gates, admission gates, segmentation gates, data-access gates, and response automation gates.

Proof ledger. A tamper-evident anchor for evidence envelopes, decision records, and enforcement actions. Its purpose is replay and tamper detection, not novelty.

Assurance work unit. A bounded thread of assurance supervision. Humans supervise *work units* (changes, incidents, releases, environments) rather than individual low-level events. This mirrors ACP-RA's use of work units as the supervised unit of long-running execution, translated to assurance. [1]

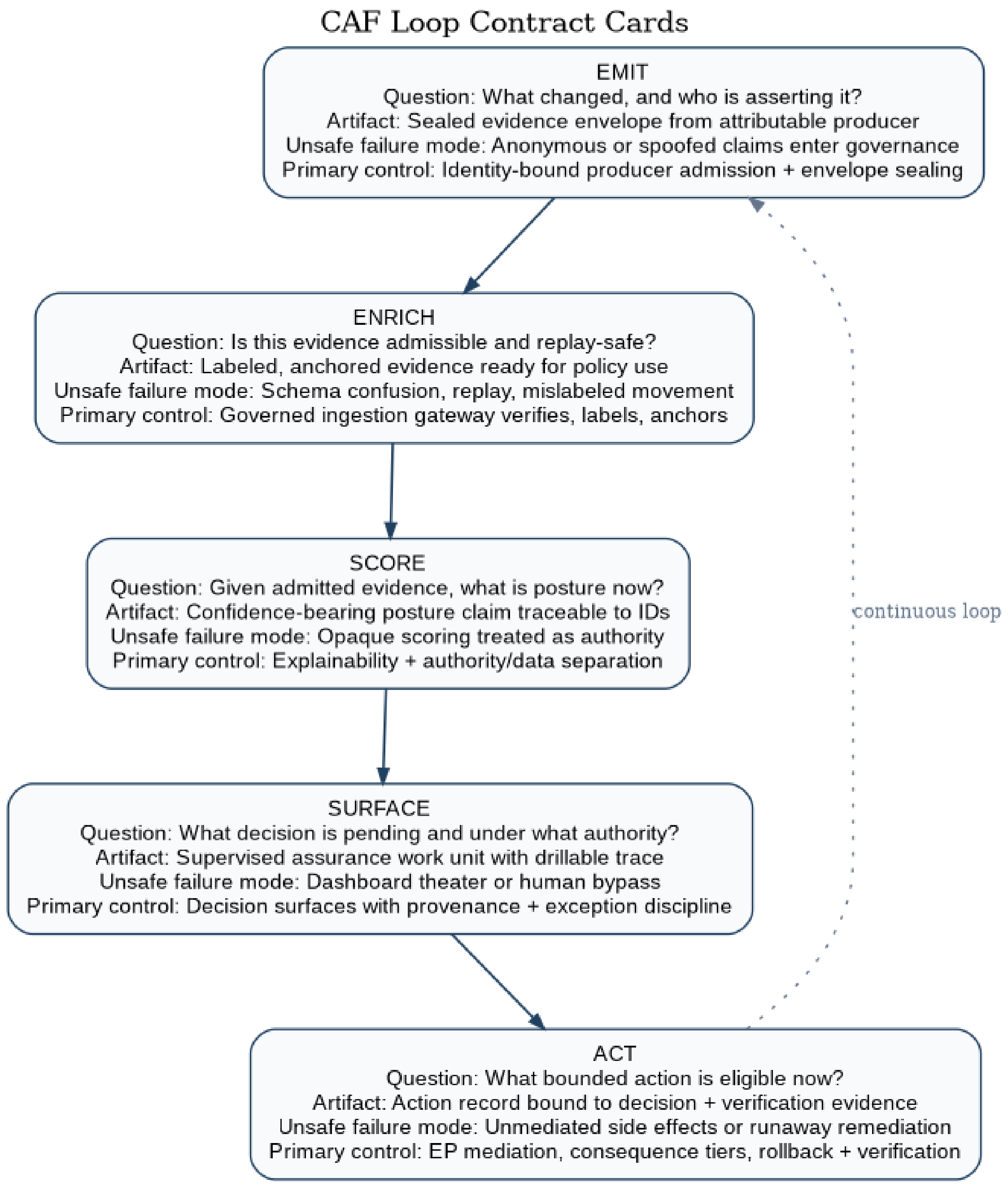

CAF Loop Contract Cards
EMIT
Question: What changed, and who is asserting it?
Artifact: Sealed evidence envelope from attributable producer
Unsafe failure mode: Anonymous or spoofed claims enter governance
Primary control: Identity-bound producer admission + envelope sealing
ENRICH
Question: Is this evidence admissible and replay-safe?
Artifact: Labeled, anchored evidence ready for policy use
Unsafe failure mode: Schema confusion, replay, mislabeled movement
Primary control: Governed ingestion gateway verifies, labels, anchors
SCORE
Question: Given admitted evidence, what is posture now?
Artifact: Confidence-bearing posture claim traceable to IDs
Unsafe failure mode: Opaque scoring treated as authority
Primary control: Explainability + authority/data separation
SURFACE
Question: What decision is pending and under what authority?
Artifact: Supervised assurance work unit with drillable trace
Unsafe failure mode: Dashboard theater or human bypass
Primary control: Decision surfaces with provenance + exception discipline
ACT
Question: What bounded action is eligible now?
Artifact: Action record bound to decision + verification evidence
Unsafe failure mode: Unmediated side effects or runaway remediation
Primary control: EP mediation, consequence tiers, rollback + verification
continuous loop

Figure 3: CAF loop contract cards: each stage specifies question, artifact, unsafe failure mode, and primary control

2 The CAF Loop

CAF is written as five verbs because the verbs are the architecture.

Figure 3 defines the contract language used in the next five sections: each stage is written as a governed answer to a specific question, produces an explicit artifact, names its unsafe failure mode, and identifies the control that prevents that failure from becoming mission impact.

2.1 EMIT

Make Reality Legible.

CAF begins by turning operational reality into evidence. This sounds obvious until you try to do it across enclaves, vendors, and contested environments. The failure mode is not "no logs." The failure mode is that evidence is not attributable, not labeled, not schema-consistent, and not confidence-scored, so it cannot safely gate anything.

CAF therefore treats evidence producers as first-class principals. Evidence comes from pipelines, runtime sensors, identity systems, network controls, data-access controls, and validation harnesses. But "comes from" is not enough: each producer must be onboarded as a non-person entity with identity and lifecycle controls, because evidence without provenance becomes narrative.

In an agentic enterprise, some evidence is generated by reasoning systems rather than sensors: autonomous validators, triage agents, detection-tuning agents, and response planning agents. CAF treats these as producers, not oracles. Their outputs become decision-grade only when the producer is an attributable principal and the output is sealed into an envelope that binds inputs, confidence, and provenance so the Department can later reconstruct what was asserted and why.

The EMIT contract is explicit: it answers "what changed, and who is asserting it?"; it emits a sealed evidence envelope from an attributable producer; its unsafe failure mode is anonymous or spoofed claims entering governance; and its primary control is identity-bound producer admission plus envelope sealing.

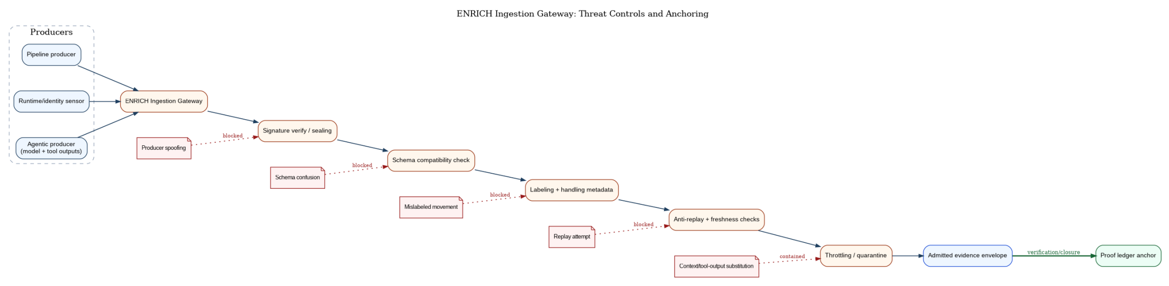
Technical position (EMIT-1): Producers MUST be attributable principals. Every producer that can influence posture or gating MUST have identity, credentials, and attributes sufficient for ABAC and audit. "The scanner said. . . " is not an assurance claim unless "scanner" is a principal whose evidence can be verified, rotated, and revoked. This aligns directly to the control-plane posture in ACP-RA: identity for non-person entities is the foundation of governed automation. [1]

Technical position (EMIT-2): Emissions MUST be sealable into an evidence envelope. CAF does not mandate what every producer emits internally. It mandates that what enters the fabric can be sealed into a standard envelope and verified independently. Producers that cannot be enveloped cannot be treated as decision-grade.

Technical position (EMIT-3): Continuous validation is an evidence producer. CAF integrates autonomous assessment concepts by treating validation outcomes as first-class evidence emissions. Scenario-as-code suites, adversarial emulation (bounded), drift probes, and detection-fidelity checks are not separate programs. They are producers that emit evidence with the same integrity and labeling rules as any other producer.

This is the point where CAF gains power without gaining complexity: validation is not a new plane; it is a privileged emitter whose outputs are policy-relevant.

Technical position (EMIT-4): Agentic assertions MUST be provenance-bound. When a producer's output is derived from model reasoning, the evidence envelope MUST include a model identifier/version, policy-governed context references, and a rationale pointer sufficient for



ENRICH Ingestion Gateway: Threat Controls and Anchoring
Producers
Pipeline producer
Runtime/identity sensor
Agentic producer (model + tool outputs)
ENRICH Ingestion Gateway
Producer spoofing
blocked
Signature verify / sealing
Schema confusion
blocked
Schema compatibility check
Mislabeled movement
blocked
Labeling + handling metadata
Replay attempt
blocked
Anti-replay + freshness checks
Context/tool-output substitution
contained
Throttling / quarantine
Admitted evidence envelope
verification/closure
Proof ledger anchor

Figure 4: ENRICH gateway threat-controls: producer admission, anti-replay/schema/label controls, quarantine, and proof-ledger anchoring

after-action reconstruction. Agent outputs without provenance are advisory at most and MUST NOT directly gate consequential change.

2.2 ENRICH

Turn Emissions into Decision-Grade Evidence.

ENRICH is the most important word in the loop, because it is where "telemetry" becomes "proof." CAF's ENRICH function is a gateway that normalizes, labels, seals, and anchors evidence. The gateway is a security boundary: it is where the Department prevents replay, schema confusion, mislabeled data movement, and producer spoofing from becoming governance failure.

Agentic systems expand the same failure modes ENRICH exists to prevent. Context can be spoofed, tool outputs can be substituted, and ungoverned retrieval can smuggle mislabeled data across boundaries. ENRICH therefore treats "context" and "agent outputs" as data that must be admitted, labeled, and anchored under the same discipline as any other evidence, or it is not decision-grade.

The ENRICH contract is explicit: it answers "is this evidence admissible, transportable, and replay-safe?"; it emits labeled and anchored evidence ready for policy use; its unsafe failure mode is schema confusion, replay, or mislabeled movement; and its primary control is a governed ingestion gateway that verifies, labels, and anchors before admission.

Technical position (ENRICH-1): Evidence ingestion MUST be a gateway, not an open endpoint. CAF requires an ingestion gateway that verifies signatures (or applies sealing signatures where appropriate), enforces schema compatibility, applies labeling rules, checks anti-replay fields, and enforces throttling and quarantine for misbehaving producers. If ingestion is not a gate, then adversaries—and accidental misconfiguration—can inject false posture.

Technical position (ENRICH-2): Evidence MUST be labeled at ingestion. Cross-enclave operation cannot be bolted on later. Evidence must carry handling metadata from the moment it is admitted into CAF. Labeling is not a UI attribute; it is part of the security and governance model.

Technical position (ENRICH-3): Evidence MUST be anchored for replay. CAF requires that evidence envelopes are anchored in a tamper-evident mechanism (proof ledger) and retained with queryable indices. You cannot do continuous authorization or credible after-action reconstruction if evidence can be silently altered or is not replayable at the same semantic version. This is the same "evidence and replay" posture that ACP-RA treats as baseline for governed autonomy. [1]

Technical position (ENRICH-4): Schema discipline MUST exist, but it is not the mental model. CAF requires schema versioning rules and compatibility policies because replay depends on stable semantics. But schema registries and mapping layers belong under ENRICH as implementation detail, not as top-level architecture nouns.

ENRICH is also where CAF performs controlled context augmentation. It adds producer metadata (provenance score, freshness, attestation state), but it does not allow arbitrary context to become authority. Context remains data. Authority remains manifests and policy.

2.3 SCORE

Compute Posture with Confidence.

SCORE converts decision-grade evidence into a posture statement that can be used to gate change and bound response. The key idea is not "a score." The key idea is **decision-grade posture with explicit confidence**, so that uncertainty drives safe behavior rather than hidden risk.

CAF assumes that posture is multi-dimensional. It includes supply chain integrity, configuration drift, identity posture, network boundary fidelity, detection readiness, and validation outcomes. Some of these can be measured directly; others are inferred. CAF does not prohibit inference. It prohibits opaque inference.

SCORE therefore must support adjudication across multiple independent producers: sensors collect, assessors augment, the ensemble judges, and CAF records the basis for the verdict.

This can be implemented as "swarm voting" in operational terms: a governed set of attributable principals emit posture-relevant claims with explicit rationales; SCORE aggregates them into a confidence-bearing posture statement; and the system preserves both the consensus and the dissent so the Department can later reconstruct why the decisioning function trusted one path over another. For quorum purposes, "independent" means producers drawn from disjoint classes (e.g., distinct telemetry stacks + distinct validation methods + distinct agentic assessors); correlated producers SHOULD be treated as one vote to avoid false confidence from shared failure modes.

Swarm adjudication does not relax CAF's authority/data separation. Votes and ensembles inform the decisioning function; they never become authority.

Signed policy bundles MUST define quorum thresholds, producer weighting, and outlier rejection criteria, and decision records MUST capture that adjudication rule so teams can replay it. Policy owners set these adjudication parameters explicitly, and reviewers test them before deployment. The decision record remains the accountable artifact: it binds manifest + policy hash to the evidence set, including the vote set and dissent references, and records why CAF permitted, denied, contained, or escalated.

Technical position (SCORE-4): For moderate and higher consequence tiers, CAF SHOULD require corroboration across independent producers (including agentic assessors) and MUST preserve the adjudication set—consensus and dissent—as replayable evidence references bound to the decision record. The SCORE contract is explicit: it answers "given admitted evidence, what is posture now and how certain are we?"; it emits confidence-bearing posture claims traceable to evidence IDs; its unsafe failure mode is opaque scoring that is treated as authority; and its primary control is explainability with authority/data separation enforced in policy.

Technical position (SCORE-1): Posture outputs MUST include confidence. Every posture claim used for gating or enforcement must include confidence fields driven by evidence completeness, provenance, freshness, and consistency. When confidence drops, CAF must tighten authority and/or escalate. This is the assurance analogue to uncertainty tolerance in trust scopes. [4]

Technical position (SCORE-2): Posture MUST be explainable back to evidence IDs. If an AO, commander, or platform lead cannot drill from "red posture" to "these evidence envelopes caused this policy decision," the control plane becomes distrusted and will be bypassed. Explainability is not a dashboard feature; it is an audit requirement.

Technical position (SCORE-3): SCORE MUST not be an authority source. A score can influence decisions; it cannot grant privileges. The only authority sources are authenticated principals and signed manifests and policies. This separation prevents "model says allow" from becoming accidental authority, which is the same separation ACP-RA draws between authority sources and data sources. [1]

CAF may implement SCORE using graphs, attack-path reasoning, ensemble scoring, Bayesian inference, or simple rules. The architecture does not care, as long as the outputs remain replayable, confidence-scored, and explainable. Advanced techniques belong to high-assurance profiles; they should not complicate the core loop.

2.4 SURFACE

Make Decisions Inspectable and Supervisable.

SURFACE exists because human authority remains the ultimate authority, even when execution is machine-speed. CAF therefore treats human supervision as a first-class function: the system must show its work, make decision boundaries explicit, and provide fast, governed paths for exceptions and overrides without destroying auditability.

SURFACE is also where CAF becomes operational rather than conceptual. A control plane that is not operationally usable becomes shadow IT. SURFACE is how CAF earns legitimacy.

The SURFACE contract is explicit: it answers "what decision is pending, under what authority, and with what evidence?"; it emits a supervised assurance work unit with drillable traceability; its unsafe failure mode is dashboard theater or human bypass; and its primary control is decision surfaces that preserve provenance, exception discipline, and review accountability.

Technical position (SURFACE-1): CAF MUST provide a drillable decision surface. The decision surface must show posture, confidence, and change over time, and allow drill-down to the evidence and policies used. "Green/red" without provenance becomes theater.

Technical position (SURFACE-2): CAF MUST surface "assurance work units," not just events. Humans supervise bounded threads: a release, an incident, a platform baseline, a mission environment. Surfacing work units prevents supervision from devolving into log watching, and mirrors ACP-RA's work-unit model for scalable supervision. [1]

Technical position (SURFACE-3): Cross-enclave posture summaries MUST be first-

class artifacts. Federation cannot require raw evidence movement. CAF therefore treats posture summaries and approved evidence packages as portable artifacts with defined schemas and minimization rules. SURFACE includes "what can be shared" views by design, rather than by ad hoc report generation.

2.5 ACT

Enforce Bounded Change and Containment.

ACT is where CAF becomes real. If CAF cannot mediate and enforce at the boundaries where change becomes irreversible, it is not a control plane. It is advisory.

ACT is also where CAF must be most disciplined. Unbounded automation is not acceleration; it is fragile risk transfer. CAF uses consequence tiers and bounded orchestration so that action remains attributable, reversible, and contained.

In agentic operations, autonomous agents may propose and execute bounded actions. CAF's rule is simple: agents do not execute side effects directly. All consequential effects are mediated through enforcement points that verify decision records, enforce consequence-tier constraints, apply rate limits, and preserve rollback and verification discipline. This makes agentic speed compatible with governable authority.

Automated remediation in CAF is not "auto-fix." It is governed change executed at machine tempo under explicit scope. Drift and issues are detected through EMIT/ENRICH, adjudicated through SCORE, and surfaced as supervised work units with traceable evidence.

Remediation agents then propose bounded plans that are attributable, consequence-tiered, and verification-first. The plan is itself evidence: it must reference the triggering envelopes, declare intended effects and blast radius, and specify how independent verification will confirm success or force rollback.

When policy and confidence allow, CAF authorizes remediation only through enforcement points. Agents do not execute side effects directly; they submit plans and requests that enforcement points validate against signed decision records, rate limits, rollback requirements, and consequence-tier constraints.

After execution, ACT emits action evidence and requires independent verification signals to close the loop. If verification fails or confidence collapses, CAF tightens authority automatically—contain, reduce privileges, and escalate—rather than allowing "optimistic remediation" to become drift by another name.

Agents propose. Enforcement points execute. Verification closes the loop.

The ACT contract is explicit: it answers "what bounded action is eligible now?"; it emits an action record cryptographically bound to a decision record and verification evidence; its unsafe failure mode is unmediated side effects or runaway remediation; and its primary control is enforcement-point mediation with consequence-tier constraints, rollback, and independent verification.

Figures 5 through 7 should be read as one continuous doctrine, not three separate illustrations. Figure 5 shows how consequence tiers mechanize risk tolerance into enforceable constraints. Figure

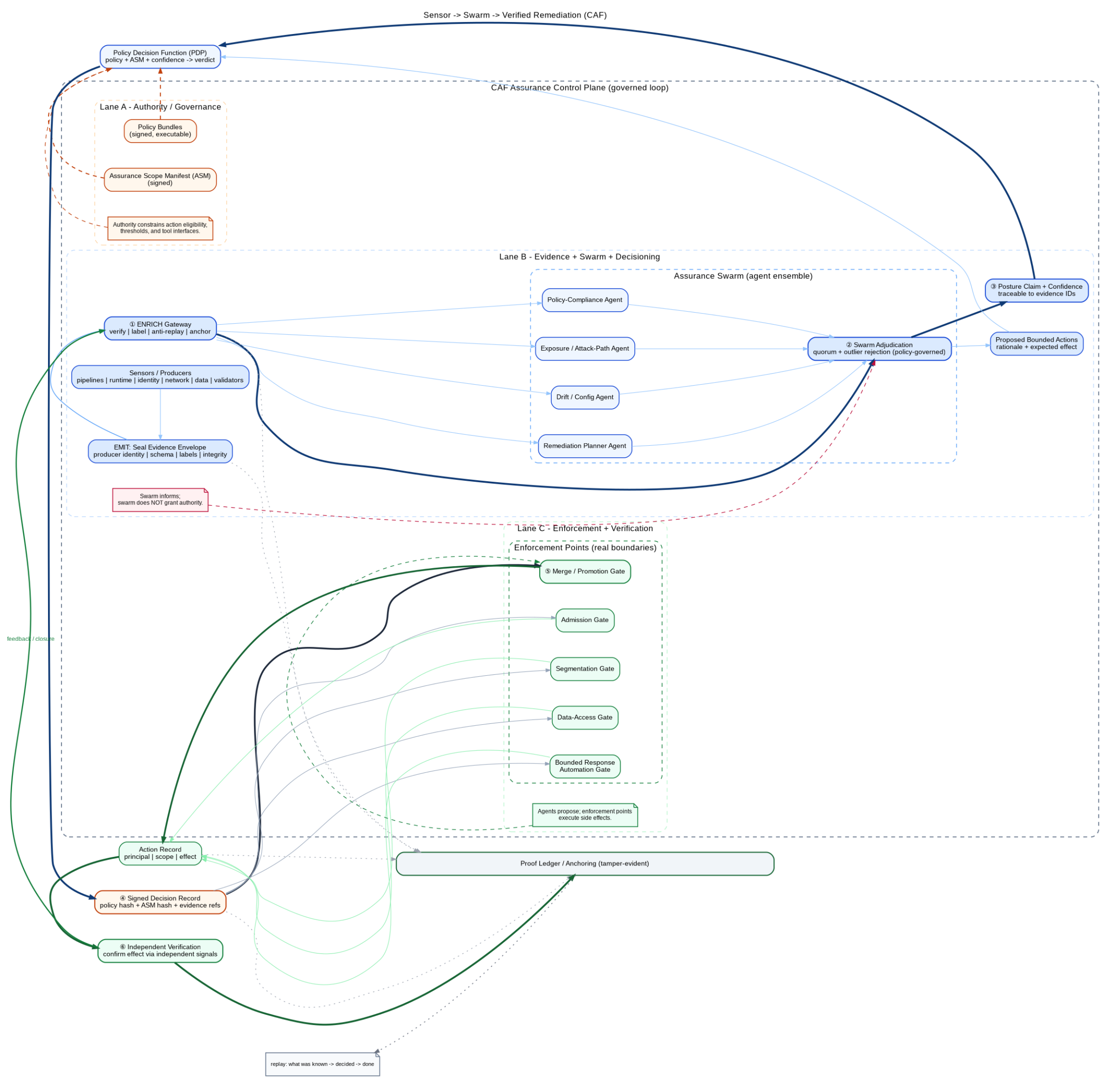

Sensor -> Swarm -> Verified Remediation (CAF)
Policy Decision Function (PDP)
policy + ASM + confidence -> verdict
CAF Assurance Control Plane (governed loop)
Lane A - Authority / Governance
Policy Bundles
(signed, executable)
Assurance Scope Manifest (ASM)
(signed)
Authority constrains action eligibility, thresholds, and tool interfaces.
Lane B - Evidence + Swarm + Decisioning
Assurance Swarm (agent ensemble)
Policy-Compliance Agent
Exposure / Attack-Path Agent
Drift / Config Agent
Remediation Planner Agent
① ENRICH Gateway
verify | label | anti-replay | anchor
Sensors / Producers
pipelines | runtime | identity | network | data | validators
EMIT: Seal Evidence Envelope
producer identity | schema | labels | integrity
② Swarm Adjudication
quorum + outlier rejection (policy-governed)
③ Posture Claim + Confidence
traceable to evidence IDs
Proposed Bounded Actions
rationale + expected effect
Swarm informs;
swarm does NOT grant authority.
Lane C - Enforcement + Verification
Enforcement Points (real boundaries)
⑤ Merge / Promotion Gate
Admission Gate
Segmentation Gate
Data-Access Gate
Bounded Response
Automation Gate
Agents propose; enforcement points execute side effects.
feedback / closure
Action Record
principal | scope | effect
④ Signed Decision Record
policy hash + ASM hash + evidence refs
⑥ Independent Verification
confirm effect via independent signals
Proof Ledger / Anchoring (tamper-evident)
replay: what was known -> decided -> done

Figure 5: Sensor-to-swarm-to-verified-remediation loop. Sensors emit sealed evidence; an assurance swarm adjudicates with quorum/outlier rejection to produce posture + confidence and bounded recommendations; policy + manifests issue a signed decision record; enforcement points execute bounded action; independent verification closes the loop and anchors proof for replay.

6 shows the sensor-to-swarm-to-verified-remediation loop: sensors and assessors produce evidence, swarm adjudication produces posture and bounded recommendations, policy and manifests issue the decision record, enforcement points execute bounded action, and independent verification closes the loop. Figure 7 shows the replay chain that proves, after the fact, what was known, what was decided, and what was done under which authority.

Technical position (ACT-1): No consequential action without a decision record. Any enforcement beyond advisory MUST be preceded by a signed decision record binding evidence references, policy hash, and ASM hash. This single position prevents silent operator drift and silent automation drift.

Technical position (ACT-2): Consequence tiers MUST constrain authority and automa-

tion. CAF must represent consequence explicitly and enforce it mechanically. Low consequence allows automation; high consequence requires stronger evidence thresholds, explicit approvals, dual control, tighter rate limits, and tighter rollback requirements. This is how CAF turns “risk tolerance” from a meeting artifact into an enforceable runtime constraint. Figure 5 illustrates the minimal tier ladder and the way higher consequence mechanically tightens evidence, approval, rollback, and action eligibility.

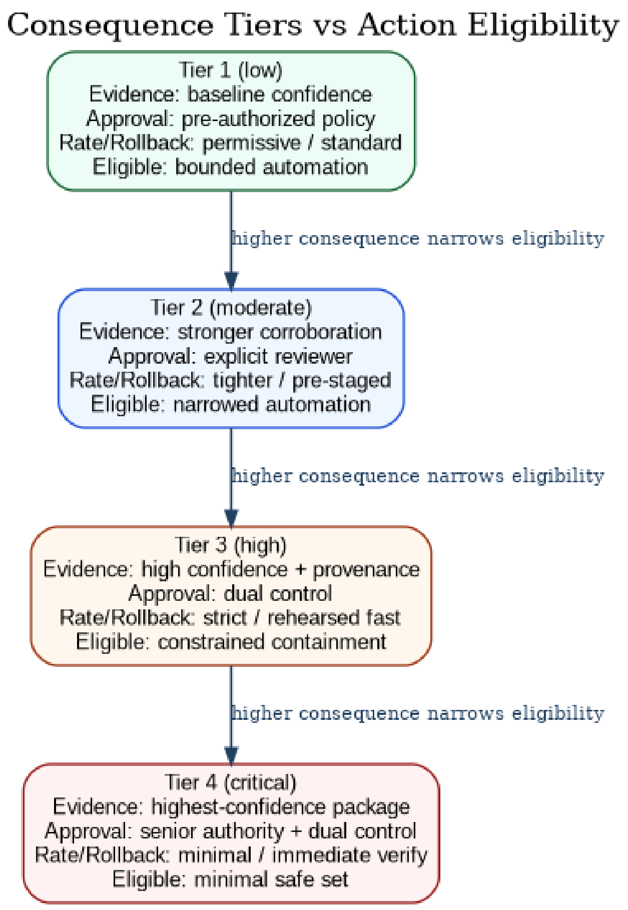

Consequence Tiers vs Action Eligibility
Tier 1 (low)
Evidence: baseline confidence
Approval: pre-authorized policy
Rate/Rollback: permissive / standard
Eligible: bounded automation
higher consequence narrows eligibility
Tier 2 (moderate)
Evidence: stronger corroboration
Approval: explicit reviewer
Rate/Rollback: tighter / pre-staged
Eligible: narrowed automation
higher consequence narrows eligibility
Tier 3 (high)
Evidence: high confidence + provenance
Approval: dual control
Rate/Rollback: strict / rehearsed fast
Eligible: constrained containment
higher consequence narrows eligibility
Tier 4 (critical)
Evidence: highest-confidence package
Approval: senior authority + dual control
Rate/Rollback: minimal / immediate verify
Eligible: minimal safe set

Figure 6: Consequence tiers tighten evidence/approval constraints and narrow action eligibility

Technical position (ACT-3): Enforcement points MUST be at real boundaries. CAF enforcement points must exist where change becomes real: merge/promotion in the factory, admission at deployment, segmentation at network boundaries, token issuance in identity, policy at data access, and bounded response automation in incident tooling. If CAF lacks enforcement at those boundaries, delivery will route around it.

Technical position (ACT-4): Every action MUST emit its own evidence. ACT closes the loop by emitting action evidence: what was done, where, by whom (principal), under which decision record, and with what result. Without this, CAF cannot prove it acted within authority.

Technical position (ACT-5): Verification MUST close the loop. CAF must verify that an action had the intended effect using independent signals. Quarantine without verification is a story.

Rollback without verification is hope. CAF is built to replace hope with evidence.

Technical position (ACT-6): Agentic orchestrators MUST be mediated. Any agent or automation that can trigger side effects MUST do so through an enforcement point that enforces policy, rate limits, and rollback discipline, and that emits action evidence cryptographically bound to the decision record.

Technical position (ACT-7): Remediation plans MUST be treated as change proposals with bounded scope and verification requirements, and MUST execute only through CAF enforcement points under signed decision records.

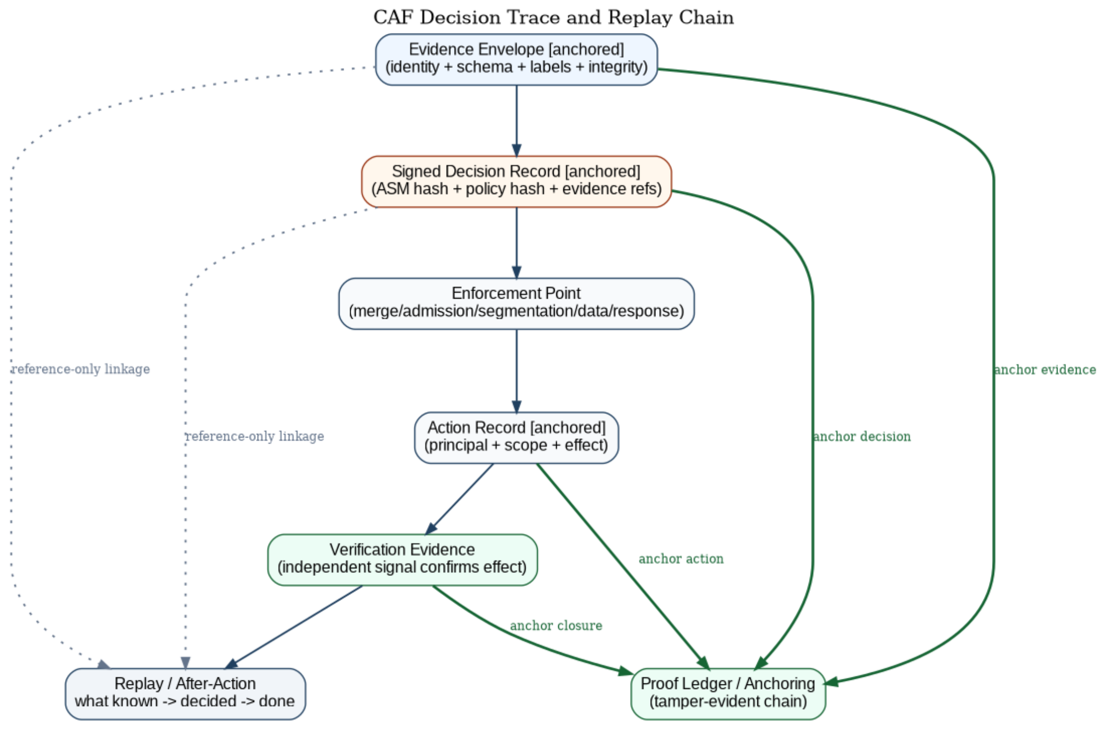

CAF Decision Trace and Replay Chain
Evidence Envelope [anchored] (identity + schema + labels + integrity)
Signed Decision Record [anchored] (ASM hash + policy hash + evidence refs)
Enforcement Point (merge/admission/segmentation/data/response)
Action Record [anchored] (principal + scope + effect)
Verification Evidence (independent signal confirms effect)
Replay / After-Action what known -> decided -> done
Proof Ledger / Anchoring (tamper-evident chain)
reference-only linkage
reference-only linkage
anchor evidence
anchor decision
anchor action
anchor closure

Figure 7: CAF decision trace from evidence to replay, including explicit proof-ledger anchoring on core artifacts

The operational effect of these three figures is that CAF becomes a control plane rather than a reporting plane. Consequence constrains what automation is eligible; swarm adjudication and bounded remediation preserve authority discipline at machine tempo; and replay ensures the Department can reconstruct governance under contested conditions. With that operational discipline established, the governance overlay exists to preserve trust categories: authority artifacts remain authoritative, data artifacts remain informative, and proof-ledger anchoring makes both replayable without collapsing them into the same trust class.

---

3 Cross-Cutting: The Governance Overlay

The loop is the spine. Governance is the overlay that prevents the loop from becoming an untrusted dashboard. Figure 8 is the doctrinal guardrail: authority artifacts and data artifacts cooperate, but never collapse into the same trust category.

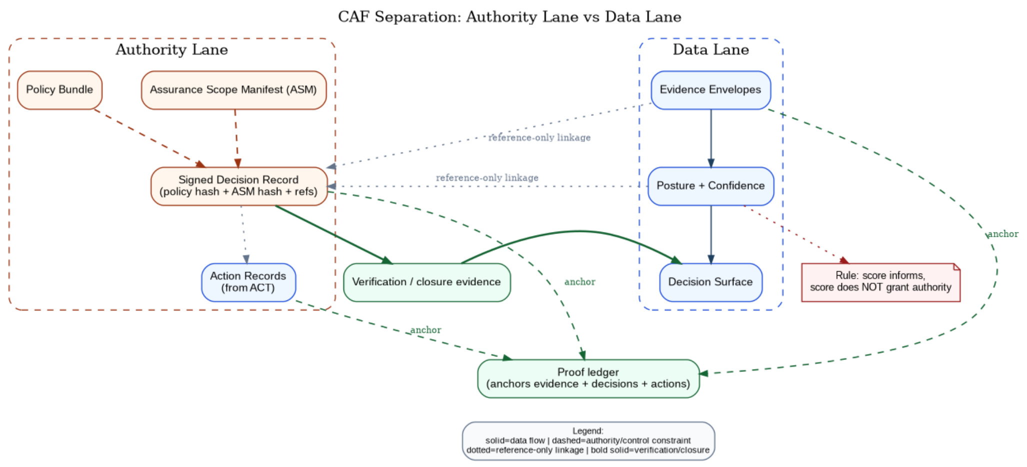

CAF Separation: Authority Lane vs Data Lane
Authority Lane
Policy Bundle
Assurance Scope Manifest (ASM)
Signed Decision Record (policy hash + ASM hash + refs)
Action Records (from ACT)
Verification / closure evidence
Data Lane
Evidence Envelopes
Posture + Confidence
Decision Surface
Rule: score informs, score does NOT grant authority
reference-only linkage
reference-only linkage
anchor
anchor
anchor
Proof ledger (anchors evidence + decisions + actions)
Legend:
solid=data flow | dashed=authority/control constraint
dotted=reference-only linkage | bold solid=verification/closure

Figure 8: Authority artifacts and data artifacts remain separate while sharing proof-ledger anchoring

3.1 Authority: Assurance Scope Manifests and Policy Bundles

CAF requires that authority is explicit and enforceable through signed artifacts. The manifest is not bureaucracy; it is how human authority remains relevant at tempo.

In an agentic enterprise, these authority artifacts also constrain autonomous action surfaces. A manifest and its policy bundles must be able to express which non-person principals are permitted to act, what classes of actions are eligible, what tool interfaces may be invoked, and what rate and consequence limits apply. This keeps "agent capability" from becoming accidental authority by ensuring that capability is always subordinate to signed, reviewable, enforceable scope.

CAF does not define tool-call governance; it consumes it. Tool invocation is an action surface, and the authority artifacts must constrain it explicitly: which tools may be called, under what parameters, at what budget/rate, and with what consequence-tier limits. ACP-RA defines the control-plane discipline for governed tool use by non-person principals; CAF treats those tool-call records and tool outputs as admissible evidence when they influence posture, gating, exceptions, or remediation eligibility. If a claim depends on a tool call, the system must bind the tool-call transcript (or a cryptographic reference to it), its outputs, and the governing policy context into evidence envelopes and replay them alongside the decision record. [1]

CAF's manifest structure deliberately mirrors the "authority × consequence × environment × uncertainty tolerance" decomposition used in the related writing, because it is the minimum sufficient structure to bound machine-speed decisioning. [4]

Policies become enforceable only as signed bundles. Drafts, discussions, or tickets may be recorded as evidence, but they are not authority sources. The code-as-policy posture exists because large organizations cannot scale governance through serialized document workflows; they scale through versioned artifacts, review, and automated verification gates. [3]

3.2 Overrides and Exceptions

CAF assumes reality will demand exceptions. CAF's requirement is not "no exceptions." CAF's requirement is that exceptions are scoped, time-bounded, compensating, and replayable.

Overrides are treated as privileged actions with mandatory evidence, because under pressure the Department cannot rely on memory and chat logs to reconstruct why a decision boundary was crossed. This is directly aligned to the control-plane posture described in ACP-RA: mediated action and evidence are non-negotiable under contested conditions. [1]

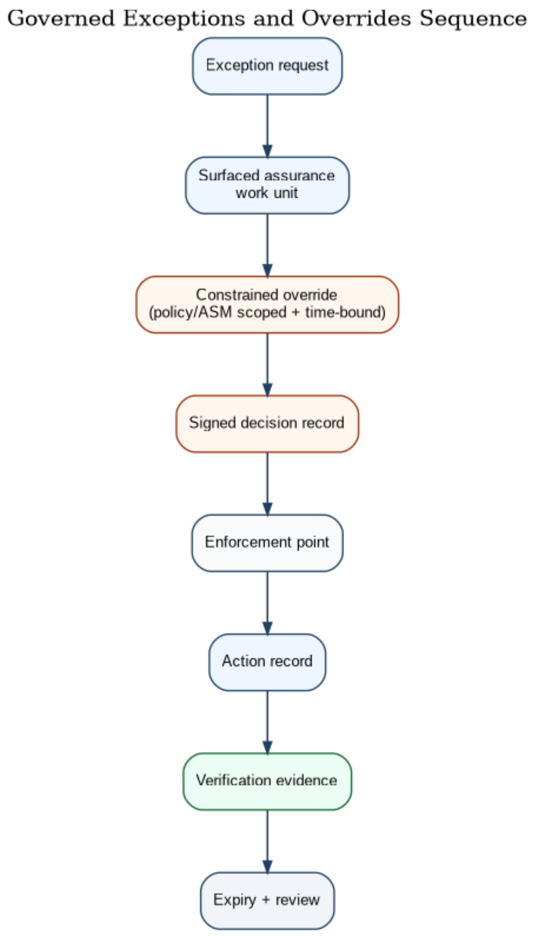

Governed Exceptions and Overrides Sequence
Exception request
Surfaced assurance work unit
Constrained override (policy/ASM scoped + time-bound)
Signed decision record
Enforcement point
Action record
Verification evidence
Expiry + review

Figure 9: Exceptions and overrides remain governed loop actions with signed records, enforcement, verification, and expiry review

---

4 Degraded-Mode Survivability

CAF is not allowed to assume perfect connectivity. Degraded mode is not a "failure case"; it is a mode of operation.

In degraded mode, CAF must tighten authority rather than relax it. That does not necessarily mean "stop all operations." It means that operations proceed only within what can be credibly governed locally: cached manifests and policy bundles, local evidence buffering, bounded local enforcement, and explicit reconciliation rules when connectivity returns.

CAF therefore supports "autonomy cells" not as a separate architecture, but as a deployment pattern: local enforcement points and local evidence buffering that can operate under denial while preserving replay and later reconciliation.

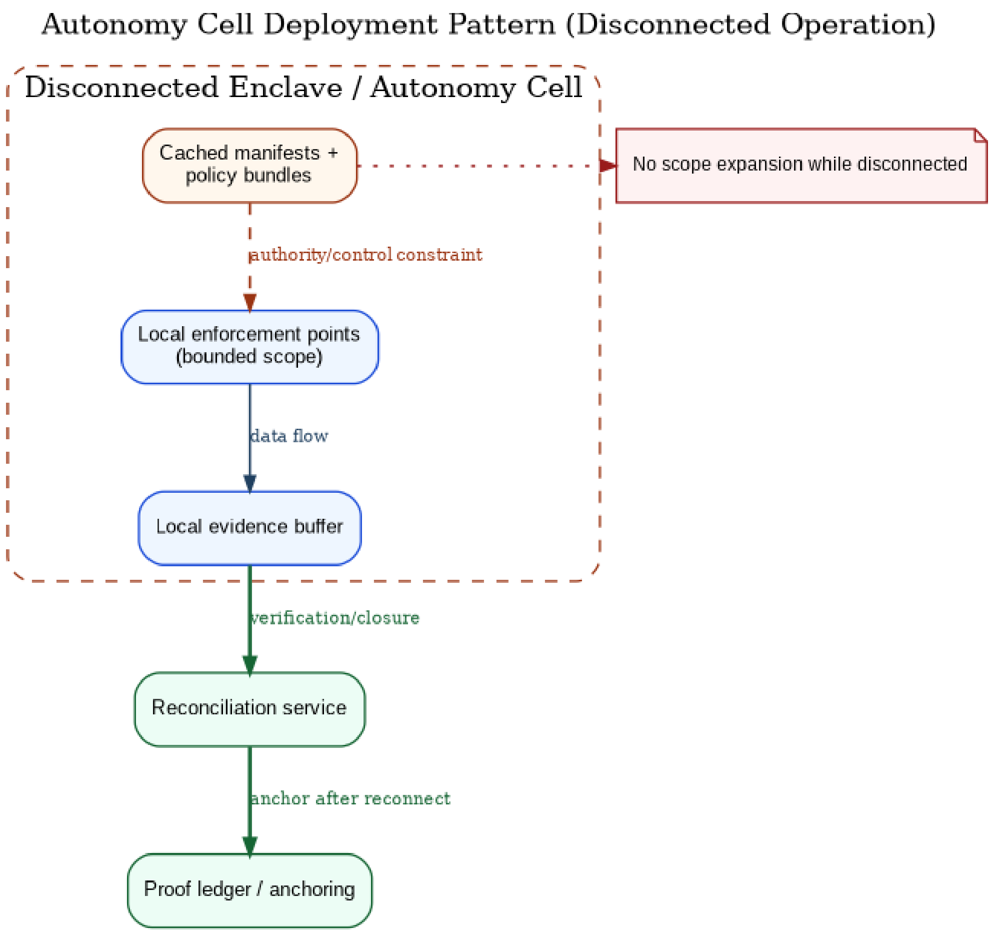

Autonomy Cell Deployment Pattern (Disconnected Operation)
Disconnected Enclave / Autonomy Cell
Cached manifests + policy bundles
No scope expansion while disconnected
authority/control constraint
Local enforcement points (bounded scope)
data flow
Local evidence buffer
verification/closure
Reconciliation service
anchor after reconnect
Proof ledger / anchoring

Figure 10: Autonomy cell deployment pattern: cached authority artifacts, bounded local enforcement, local buffering, and reconnect reconciliation

Figure 11 then makes the expected state transitions explicit so degraded behavior is governable in advance, not improvised mid-incident.

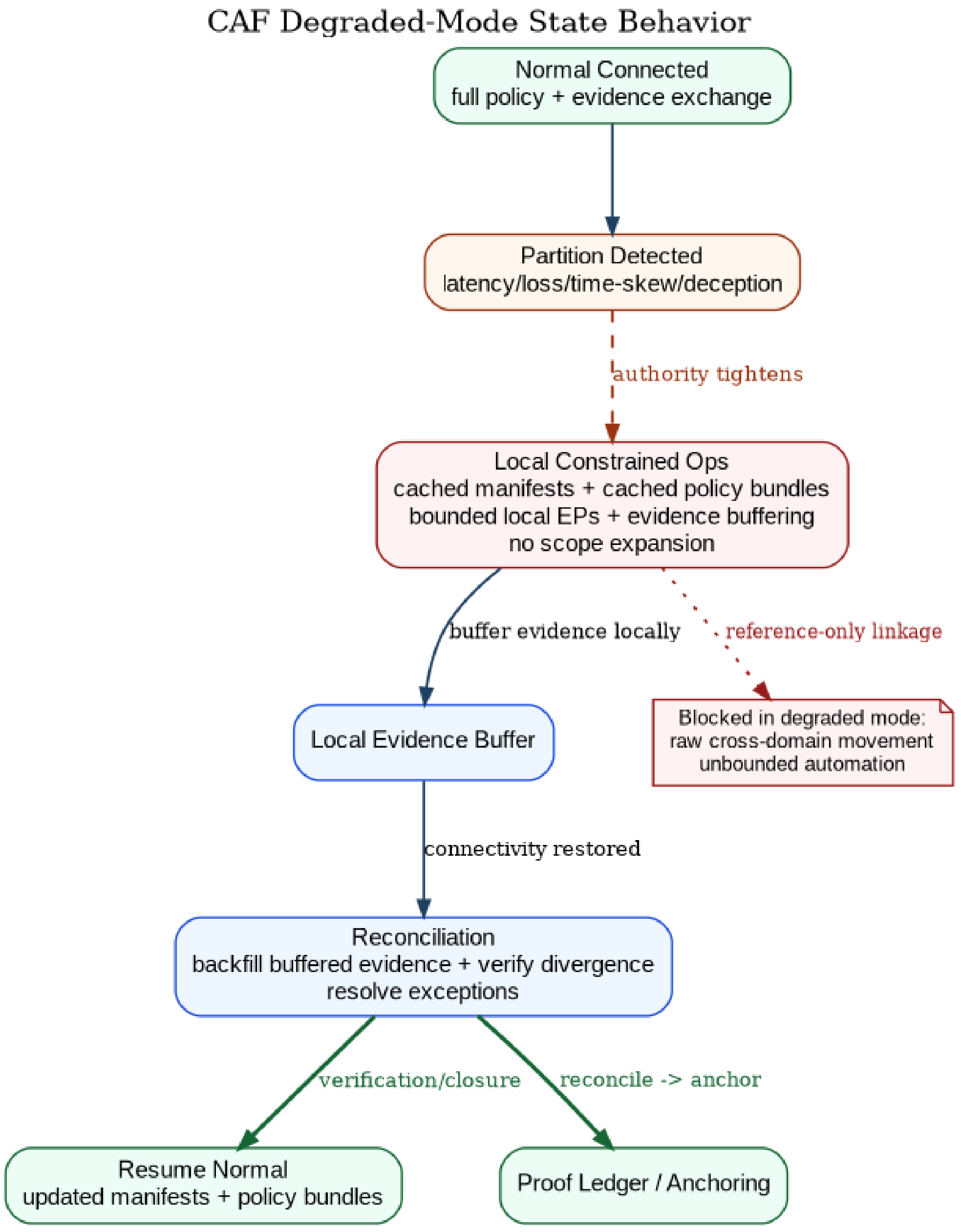


Figure 11: Degraded-mode state transitions with explicit authority-tightening rule, local evidence buffering, and reconciliation anchoring

5 Federation and Cross-Domain Posture Exchange

CAF is federated by default. Authority composes by intersection across manifests and policy bundles, not by union. This prevents accidental over-permission as systems connect. ACP-RA makes the same point for delegated autonomy; CAF adopts it for assurance authority. [1]

Cross-domain behavior is handled by treating posture summaries and evidence packages as explicit artifacts. The default is to exchange posture summaries with strict minimization; raw evidence movement is exception-based and governed. Figure 12 shows the summary-first posture that keeps federation useful without normalizing unnecessary raw evidence movement.

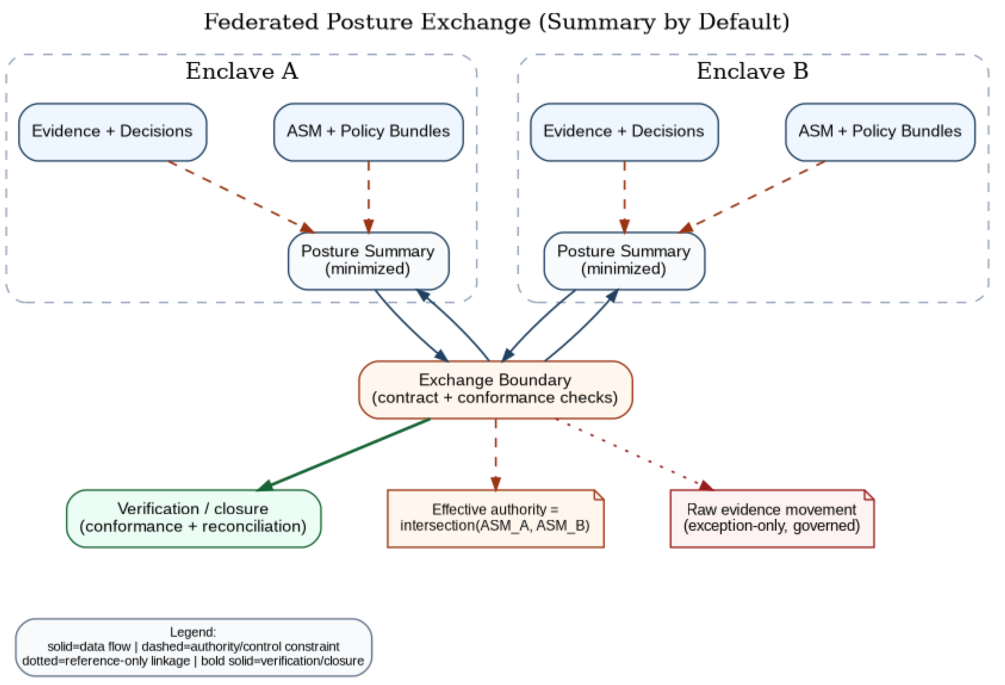

Federated Posture Exchange (Summary by Default)
Enclave A
Enclave B
Evidence + Decisions
ASM + Policy Bundles
Evidence + Decisions
ASM + Policy Bundles
Posture Summary (minimized)
Posture Summary (minimized)
Exchange Boundary (contract + conformance checks)
Verification / closure (conformance + reconciliation)
Effective authority = intersection(ASM_A, ASM_B)
Raw evidence movement (exception-only, governed)
Legend:
solid=data flow | dashed=authority/control constraint
dotted=reference-only linkage | bold solid=verification/closure

Figure 12: Federated posture exchange with summary-by-default discipline and intersection authority rule

---

6 Conformance Profiles

Profiles are how CAF preserves baseline simplicity while enabling high-assurance depth. The core loop does not change. The profile determines what sophistication is required.

6.1 CAF-Core (Fieldable Baseline)

CAF-Core requires the loop and the three non-negotiables: explicit authority artifacts, decision-grade evidence envelopes anchored for replay, and mediated enforcement at real boundaries. It is the minimum profile that prevents CAF from collapsing into advisory dashboards.

6.2 CAF-Standard (Enterprise-Scale Continuous Authorization)

CAF-Standard adds disciplined inheritance and validation-as-gate. It expects continuous validation results to be produced as evidence, bounded remediation actions to exist at moderate consequence tiers, and degraded-mode behavior to be rehearsed, not assumed.

6.3 CAF-High Assurance (Contested, Multi-Enclave, High Consequence)

CAF-High Assurance assumes contested conditions, partial truth, and active deception. It therefore reduces reliance on any single sensor, any single model, or any single enclave narrative by requiring redundancy and adjudication. Consensus/outlier detection is not a dashboard feature; it is a control-plane mechanism: posture claims and high-consequence action eligibility SHOULD require quorum across independent producer classes (e.g., multiple sensor stacks plus independent agentic assessors), with explicit handling of disagreement and outliers. When the swarm cannot converge, CAF treats that as uncertainty and tightens authority—contain, narrow eligibility, or require explicit human review—rather than averaging disagreement into false confidence. The proof-ledger anchors not only the resulting posture claim, but also the underlying vote set and dissent references, so after-action review can reconstruct which paths agreed, which paths resisted, and why the decisioning function selected the governing verdict.

---

7 Patterns

Patterns are not mandatory components; they are reusable moves that preserve conformance while allowing implementation flexibility. Their purpose is to capture the quickest ways to get operational leverage from the loop without expanding the core ontology or turning CAF into a product catalog.

Evidence-first releases treat promotion as a proof act rather than a calendar event. Promotion becomes contingent on decision-grade evidence and explicit authority, so progressive delivery is not only a resilience mechanism but also an assurance mechanism: it produces bounded proof of what changed, what was observed, what was decided, and what was rolled back if needed.

Validation-as-gate treats scenario suites as governed artifacts and treats their outcomes as first-class evidence. In this pattern, automated validation—including bounded adversarial emulation and detection-fidelity checks—is not an external reporting activity. It is a producer whose outputs are enveloped, anchored, and made policy-relevant so that promotion and runtime escalation can be gated by verifiable results.

Control inheritance as proof treats inheritance as a lineage graph bound to evidence and policy versions rather than a spreadsheet claim. Inheritance remains auditable because the inherited control is tied to specific evidence envelopes, specific policy bundles, and specific decision records, so an inherited claim can be replayed and validated rather than asserted.

Bulkhead gateways treat ingestion, export, and orchestration boundaries as bulkheads that prevent cascades. This pattern focuses on failure containment: schema confusion, replay attempts, poisoned producers, and runaway remediation are contained at controlled boundaries where throttling, quarantine, and rollback are enforceable and attributable.

Assurance work units make supervision scalable by organizing human oversight around bounded threads—releases, incidents, environments, and baselines—rather than event floods. This mirrors the scalable supervision posture in ACP-RA and keeps human authority operationally relevant even when machine-tempo actors are generating and acting on evidence continuously. [1]

---

8 Acquisition and Governance Alignment

CAF cannot succeed as a "platform feature." It succeeds as a conformance surface.

The core insight from *Stand Up Delegated Autonomy Directorate (DAD) as a Joint Control Plane Authority* applies: governed control planes scale when they have (1) technical enforcement, (2) operational doctrine, and (3) acquisition leverage. CAF's governance should be designed for that dual reality: enforcement plumbing and doctrine are both required, and procurement must be able to say "no conformance, no scale." [2]

CAF also aligns with the code-as-policy posture: the artifacts that grant authority (manifests, policy bundles, conformance tests, scenario suites) must be versioned, reviewable, and continuously verified. Otherwise, CAF will drift into informal exceptions and untraceable variance. [3]

---

9 Adoption Path (How CAF Becomes Real)

CAF is adopted in phases because trust is earned through safe enforcement, not asserted through architecture.

In the beginning, CAF runs in shadow mode: it emits and enriches, computes posture and confidence, and surfaces decisions without enforcement. This phase is about proving that evidence can be made decision-grade.

Next, CAF enforces low-consequence gates (promotion and admission constraints) where rollback is cheap and lessons are fast. Then it graduates into bounded containment and remediation, including agentic remediation executed only through CAF enforcement points under signed decision records, where blast radius can be constrained. Finally, it scales across enclaves through posture summaries and conformance governance, with explicit phase-exit criteria as shown in Figure 13.

CAF is not deployed when software ships. CAF is deployed when governance behavior changes.

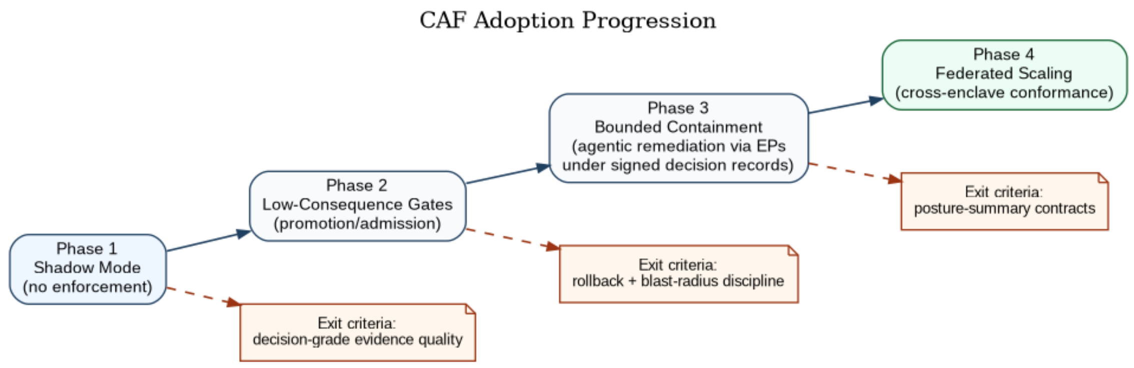


Figure 13: CAF adoption progression and phase-exit criteria, including bounded agentic containment in Phase 3

10 Minimal Schema Appendix (Illustrative)

CAF's schemas are contracts. The exact fields evolve, but the invariants do not: attributable identity, handling labels, integrity anchoring, anti-replay, and replayable references. The purpose of the schema is not to standardize every internal producer format; it is to standardize what enters the fabric and what can be audited, gated, and replayed across enclaves.

An evidence envelope binds a producer principal, schema reference, time and clock confidence, handling labels, subject identifiers, integrity anchors, anti-replay fields, and a signature. When the producer is agentic, the envelope also binds model/toolchain identifiers and explicit references to the governed context and tool outputs used to produce the claim, so that later reconstruction is possible even when the reasoning system is probabilistic.

A decision record binds policy hash, ASM hash, and evidence references to a verdict, with explicit confidence, expiry, and rationale pointers. The record exists to prove that a consequential action was eligible under a specific authority scope and policy version, based on specific evidence, at a specific time, and through a decisioning function that can be replayed. When adjudication is performed by multiple assessors, the decision record SHOULD include explicit references to the adjudication set (the set of posture-relevant assessment envelopes) and any dissent references that materially affected confidence. This allows CAF to replay not only the verdict, but the basis of consensus and disagreement that produced it.

An action record binds a decision record ID to the principal that executed the effect, the scope and boundary where it was enforced, the observed result, and the verification evidence that confirms the action achieved its intended effect. This closes the loop by turning enforcement into evidence rather than narrative.

11 Recommendations

Adopt CAF as a loop-first control-plane architecture and keep the core vocabulary constrained so it remains operable at mission tempo. Require signed ASMs and policy bundles before any consequential automation, and treat evidence envelopes, proof-ledger anchoring, and decision records as mandatory contracts for anything that can gate change. Establish enforcement points at pipeline, admission, segmentation, data-access, and bounded response boundaries so the control plane is real rather than advisory. Define confidence-driven escalation thresholds and rehearsed degraded-mode operating rules so uncertainty tightens authority instead of silently relaxing it.

12 Conclusion

CAF-RA is a practical control-plane architecture for continuous assurance at mission tempo. It does not add complexity for its own sake. It limits baseline complexity to one governed loop and uses profiles to scale depth only where mission consequence requires it.

The architecture is successful when it does three things consistently: it keeps authority explicit, it keeps evidence replayable, and it keeps enforcement mediated at real boundaries. When those conditions hold, continuous authorization becomes operationally credible in federated and contested environments [1, 2].

CAF's central claim is doctrinal as much as technical: in a continuous, agentic enterprise, authorization is not a point-in-time approval attached to a deployment. It is a continuously maintained state produced by explicit authority artifacts, replayable evidence, confidence-aware posture, and mediated enforcement at the boundaries where change becomes real. CAF makes autonomous speed compatible with governable control by ensuring that capability never becomes authority, that every consequential action is attributable to a decision record, and that every decision remains replayable under contested and degraded operations. The loop-contract framing in Figure 3 is therefore not presentation style; it is the operational doctrine that keeps assurance governable when machine-tempo actors are part of the force.

References

[1] Boas, A. (2026). *Agent Control Plane Reference Architecture (ACP-RA).* Available at: adamboas.com/writing/acp-ra/.

[2] Boas, A. (2026). *Stand Up Delegated Autonomy Directorate (DAD) as a Joint Control Plane Authority.* Available at: adamboas.com/writing/2026-02-12-stand-up-dad-joint-control-plane-authority/.

[3] Boas, A. (2026). *From PDFs to Pull Requests.* Available at: adamboas.com/writing/code-as-policy/.

[4] Boas, A. (2026). *From AI Force Multiplication to Force Creation.* Available at: adamboas.com/writing/agentic-force-creation/.