Computable Trust Architecture: A Formal Framework for Runtime AI Governance

John DeRudder
Independent AI Governance Researcher
April 2026 | johndr2718@gmail.com

Disclaimer: *The views expressed in this paper are those of the author and do not represent the views of any employer, client, or affiliated organization. The reference implementation and evaluation results presented here were produced independently.*

Abstract

The rapid deployment of AI systems in regulated domains has outpaced the governance mechanisms needed to constrain them at runtime. This paper introduces the Computable Trust Architecture (CTA), a framework for runtime AI governance that treats trust as a computable, attributable, temporally valid, and policy-enforceable property of AI-mediated actions and outputs. As a reference implementation of CTA, the Trust Computation System (TCS) computes a Trust Integrity Score (TIS) over governed dimensions under a resolved policy configuration parameterized by risk tier, action class, and connection context, and produces a Trust Certificate (TC) as a tamper-evident, hash-chained, identity-attributed governance artifact suitable for audit and regulatory examination. We present the formal mathematical basis of the framework, its architectural components, a reference implementation passing 108 specification unit tests, controlled evaluations across financial services and healthcare governance scenarios, and a model of trust dynamics covering loss, drift, and recovery. The framework addresses a structural gap in contemporary AI governance between policies that specify intended behavior and enforcement mechanisms capable of governing action at runtime.

Index Terms

AI governance; trust computation; policy enforcement; computable trust; Trust Integrity Score; Trust Certificate; agentic AI; MCP boundary security; regulatory compliance; Trust Dynamics

NOMENCLATURE

To reduce ambiguity across the conceptual, architectural, computational, and implementation layers of this work, the following terms are used in a deliberately hierarchical manner.

Computable Trust: The broader discipline concerned with expressing trust as a measurable, auditable, and machine-enforceable property within AI and digital systems.

Computable Trust Architecture (CTA): The reference architectural pattern introduced in this paper for operationalizing computable trust in governed systems. CTA defines the structural relationships among trust computation, governed context assembly, policy resolution, temporal validity, and decision enforcement.

Trust Computation System (TCS): A reference implementation of CTA. TCS instantiates the architectural pattern described in this paper as a concrete system capable of computing trust scores, issuing tamper-evident trust artifacts, and enforcing runtime governance decisions at the point of action.

Trust Integrity Score (TIS): The core trust computation produced by TCS. TIS represents the computed trust state of an AI output x under a resolved policy configuration, for a specified risk tier r, action class a, and evaluation time t.

Trust Certificate (TC): The tamper-evident, eleven-layer governance artifact generated by TCS to persist the outcome of each trust computation. The eleven layers comprise seven Core Decision-Record Layers (decision,

score, component scores, gate results, policy context, provenance, explanation) and four Evidentiary Enforcement Layers (identity binding, governance status, audit integrity, override record); see Section VI.

For clarity, this paper uses CTA to denote the general architectural pattern and TCS to denote the reference implementation. Terms such as TIS, TC, GCA, and PLL refer to computational or architectural components within that implementation. This separation is intentional: the architecture is presented as generalizable, while the system demonstrates one concrete realization of that architecture.

Abbreviations

Abbr.	**Name**	**Abbr.**	**Name**
API	Application Programming Interface	PHI	Protected Health Information
BACK	Boundedness, Attribution, Compliance, Known (the four governance dimensions)	PLL	Policy Learning Layer
CT-n	Connection Type n; see Table III for types	RAG	Retrieval-Augmented Generation
CTA	Computable Trust Architecture	RBAC	Role-Based Access Control
DDL	Data Definition Language	RLHF	Reinforcement Learning from Human Feedback
EU	European Union	SaMD	Software as a Medical Device
FDA	U.S. Food and Drug Administration	SEC	U.S. Securities and Exchange Commission
FINRA	Financial Industry Regulatory Authority	SHA	Secure Hash Algorithm
GCA	Governed Context Architecture	TC	Trust Certificate
HIPAA	Health Insurance Portability and Accountability Act	TCS	Trust Computation System
ISO	International Organization for Standardization	TIS	Trust Integrity Score
MCP	Model Context Protocol	WORM	Write Once, Read Many
NIST	National Institute of Standards and Technology		

I. Introduction

Enterprise AI systems are making consequential decisions at scale. Loan approvals, clinical recommendations, investment suitability analyses, and regulatory reporting are generated by AI pipelines that, in most deployments, have no formal mechanism for enforcing governance at the point of action. Policy documents describe intended behavior. Audit logs record what occurred. Neither constitutes a governance enforcement system.

The structural gap is between policy intent and policy enforcement. Existing frameworks, including NIST AI RMF [1], ISO 42001 [2], the EU AI Act [3], and SEC guidance on AI in investment advice [4], define what governance should achieve. None formally specify how trust is computed in real time, how policy is enforced at the output boundary, or how an auditable governance record is produced for each decision. TCS is designed to fill this gap.

We make the following contributions: (1) the formal definition of the Trust Integrity Score as a multi-dimensional, policy-parameterized, connection-type-aware governance function; (2) the Trust Certificate, a tamper-evident governance artifact with eleven mandatory layers organized into seven Core Decision-Record Layers and four Evidentiary Enforcement Layers, detailed in Section VI; (3) the Governed Context Architecture, a data governance model covering 13 acquisition pathway types; (4) the Trust Certificate enforcement model, defining four mandatory Evidentiary Enforcement Layers for identity binding, governance status, audit integrity, and override accountability; (5) MCP Governance, a formal treatment of integration boundary security in agentic pipelines; (6) Trust Dynamics, formal models of trust loss, adaptive governance, drift detection, and recovery; (7) a reference implementation achieving 108 specification unit tests (Phase 1); and (8) a live sidecar runtime, a FastAPI service demonstrating real-time governance enforcement across a financial RAG pipeline with all five decision outcomes produced against a persistent Trust Certificate store.

II. Background and Related Work

AI governance frameworks can be categorized along two axes: descriptive versus prescriptive, and retrospective versus real-time. The NIST AI RMF [1] is descriptive and retrospective: it provides a vocabulary and process model for organizations to assess and manage AI risk, but does not specify enforcement mechanisms. ISO 42001 [2] is prescriptive but retrospective: it defines an AI management system standard against which organizations can certify, but certification does not guarantee runtime enforcement.

Model cards [5] and datasheets for datasets [6] address transparency at the artifact level, not the decision level. Algorithmic auditing frameworks [7] focus on post-hoc assessment of deployed models. None address the question of what governance record is produced for a specific AI output at the moment it is generated.

Constitutional AI [8] and RLHF-based alignment [9] address model behavior through training. They do not provide a runtime enforcement mechanism applicable to any model without retraining, nor do they produce auditable per-decision governance records. Formal verification approaches [10] apply to specific model architectures and cannot generalize across diverse enterprise AI deployments.

The closest related work is in runtime monitoring for AI systems [11], [12]. These approaches monitor AI system behavior for anomaly detection but do not formalize trust as a governance dimension or produce regulatory-grade audit records. Commercial AI governance platforms including Credo AI, Fiddler AI, Arthur AI, and Holistic AI address governance at the model evaluation and monitoring layer: they provide dashboards, bias detection, drift alerts, and compliance reporting across the model lifecycle. These platforms do not produce per-decision governance records, enforce policy at the output boundary, or generate tamper-evident audit artifacts attributable to a specific evaluation. TCS operates at a different layer of the governance stack, real-time enforcement at the point of action, and is architecturally complementary to model evaluation tooling rather than competitive with it.

III. Computable Trust Architecture

We define the *Computable Trust Architecture* (CTA) as the category of systems that compute trust as a first-class, quantifiable property of AI outputs and enforce governance decisions based on that computation in real time. CTA is distinct from three adjacent categories:

- **AI Safety**: addresses existential and misuse risks at the model or system level; does not produce per-decision governance records.
- **Model Alignment**: modifies model training objectives; does not enforce governance on deployed models without retraining.
- **Compliance Management**: documents policies and tracks adherence; does not enforce policy at the point of action.

Throughout this paper, CTA denotes the architectural pattern, TCS denotes the reference implementation, and TIS, TC, GCA, and related constructs denote computational and architectural components within TCS.

Computable Trust: *Trust $T(x, r, a, \rho, t)$ is a quantifiable, policy-governed property of AI output x that is **B**ounded (confined to authorized scope), **A**ttributable (traceable to verified sources), **C**ompliant (consistent with active policy), and **K**nown (the AI system's expressed confidence is calibrated against the actual reliability of its inputs, evaluated against a policy configuration ρ that is fully resolved, versioned, and locked before evaluation). These four properties define the **BACK** model of trust.* BACK serves as both the conceptual framework and the operational scoring decomposition: each property is scored as a normalized scalar in $[0, 1]$ representing the degree to which the output satisfies that property. The policy configuration ρ is always fully resolved before scoring begins; K measures how well the AI system knows what it knows.

The Trust Computation System (TCS) is the reference implementation of CTA. It operates as a governance sidecar that attaches to existing AI systems at defined trust boundaries without requiring model modification, retraining, or orchestration changes. Enforcement depends on architectural placement: TCS must intercept outputs at the boundary between the AI system and its downstream consumers, which is an operational governance decision made at deployment time rather than a TCS component. Pipelines that route outputs around TCS are outside the system's enforcement perimeter; detecting and preventing such bypass is a deployment architecture responsibility. TCS surfaces this condition explicitly: the `scope_attestation.enforcement_perimeter_complete`

field in every Trust Certificate documents whether all governed outputs traversed the enforcement layer at evaluation time, enabling auditors to distinguish complete enforcement from partial coverage. TCS components are:

- **Signal Chain Framework**: a sequential enforcement layer governing output, tool invocation, and integration boundaries.
- **Governed Context Architecture (GCA)**: the data plane component that assembles governed context from raw inputs, resolving connection type and policy profile.
- **Trust Evaluation Engine**: deterministic, stateless TIS computation.
- **Trust Certificate**: a tamper-evident, eleven-layer governance artifact (Section VI).
- **Lifecycle and Calibration Layer**: manages trust state over time including drift, adaptation, and recovery.

IV. The Trust Integrity Score

A. Canonical Equation

The Trust Integrity Score is computed in three sequential stages that separate the governance decision from its temporal validity:

$$S_{\text{base}}(x, r, a, \rho) = \sum_{i=1}^{4} w_i(r, a; \rho)\ \dim_i(x, \rho) \tag{1}$$

$$\text{TIS}_{\text{raw}}(x, r, a, \rho) = G_{r,a}(x, \rho) \cdot S_{\text{base}}(x, r, a, \rho) \tag{2}$$

$$\text{TIS}_{\text{adj}}(x, r, a, \rho) = \text{TIS}_{\text{raw}}(x, r, a, \rho) \cdot \big(1 - P(x, \rho)\big) \tag{3}$$

$$\text{TIS}_{\text{current}}(x, r, a, \rho, t) = \text{TIS}_{\text{adj}}(x, r, a, \rho) \cdot e^{-\mu_{r,a}\,\Delta t} \cdot I_{\text{inv}} \tag{4}$$

S_{base} is the ungated weighted dimension score at evaluation time t_0 and is used to determine whether a gate failure remains eligible for human review. TIS_{raw} is the gate-weighted dimension score before any penalty reduction; when $G = 0$, TIS_{raw} collapses to zero even if S_{base} remains high. TIS_{adj} is the penalty-adjusted score; it is the quantity on which the governance decision is made and the quantity stored immutably in the Trust Certificate alongside the evaluation timestamp t_0 and the decay rate $\mu_{r,a}$. $\text{TIS}_{\text{current}}$ is a derived authorization-validity score. The Trust Certificate stores the decision-time score values together with the parameters required to recompute $\text{TIS}_{\text{current}}$ on demand: the evaluation timestamp t_0, the decay rate $\mu_{r,a}$, and the invalidation state. Implementations may additionally record an issuance-time value for display, but the authoritative current authorization value is the one derived from those stored parameters and the elapsed time at the moment of authorization. This separation is material for the audit trail: the TC is an immutable record of the governance decision made at t_0; $\text{TIS}_{\text{current}}$ reflects whether that decision remains valid for downstream authorization at any later time $t > t_0$.

Variable definitions:

- $S_{\text{base}}(x, r, a, \rho) \in [0, 1]$ is the ungated weighted dimension score. It preserves the aggregate governance quality of the output before any gate collapse and is used for gate-path HOLD/STOP discrimination;
- $G_{r,a}(x, \rho) \in \{0, 1\}$ is the gate indicator function (Eq. 6); $G = 0$ collapses TIS_{raw} to zero regardless of S_{base};
- $w_i(r, a; \rho)$ are policy-determined dimension weights resolved from the active policy configuration ρ, with $\sum_{i=1}^{4} w_i(r, a; \rho) = 1$;
- $\dim_i(x, \rho) \in [0, 1]$ are the four governance dimension scores (B, A, C, K), each computed by the GCA from output features, provenance metadata, and policy-defined scoring rubrics in ρ;
- $P(x, \rho) \in [0, 1]$ is the aggregate penalty score, defined as:

$$P(x, \rho) = 1 - \prod_{j=1}^{5} \big(1 - \lambda_j(\rho)\phi_j(x, \rho)\big) \tag{5}$$

where each $\phi_j \in \{0, 1\}$ is an indicator for an active penalty event and $\lambda_j \in (0, 1]$ is its policy-determined severity weight. The five penalty event types are: `context_boundary_violation` (λ_1), `novelty_flag` (λ_2), `data_quality_flag` (λ_3), `prior_STOP_in_session` (λ_4), and `human_review_required`

(λ_5). When no penalty events are present, $P(x, \rho) = 0$; as active penalty events accumulate, $P(x, \rho)$ increases monotonically toward its policy-defined upper bound, with high-severity combinations driving $\mathrm{TIS}_{\mathrm{adj}}$ toward zero. The `human_review_required` signal also appears as a direct HOLD trigger in the decision ladder (Section IV-D); this deliberate redundancy ensures that a required-review flag both depresses the score and independently gates the output, preventing a high-scoring output from bypassing human review through the ALLOW path;

- $\mu_{r,a}$ is the temporal decay rate, with canonical default values of $\mu = 0.02$ hr^{-1} at r1 (TIS half-life $\approx$ 35 hours), $\mu = 0.05$ hr^{-1} at r2 (half-life $\approx$ 14 hours), and $\mu = 0.10$ hr^{-1} at r3 (half-life $\approx$ 7 hours). Higher-risk contexts require more recent evidence to remain authoritative. Δt is the elapsed time in hours since the Trust Certificate was issued, i.e., since the evaluation timestamp t_0 at which the TC was first written to the append-only store;
- $I_{\mathrm{inv}} \in \{0, 1\}$ is the invalidation-survival indicator: $I_{\mathrm{inv}} = 1$ when no active invalidation event exists, and $I_{\mathrm{inv}} = 0$ after an invalidation event, immediately voiding $\mathrm{TIS}_{\mathrm{current}}$.

The three governance mechanisms are orthogonal by design. Dimension scores (dim_i) capture the baseline governance state of the output type under this policy. Penalties (P) capture exceptional events that occurred during this specific evaluation. The invalidation-survival indicator (I_{inv}) captures whether the original evaluation remains valid after post-evaluation events; downstream invalidation events set $I_{\mathrm{inv}} = 0$ and revoke the current authorization value. A signal such as novelty legitimately appears in both K (as a baseline calibration concern) and in the novelty penalty event (as an anomalous occurrence in this specific evaluation) without double-counting, because each captures a different temporal and logical governance question.

B. The Four Governance Dimensions

TABLE I
THE FOUR BACK GOVERNANCE DIMENSIONS.

Dim.	Name	Definition
B	Boundedness	Degree to which output is confined to its authorized scope, identity tier, and permission set.
A	Attribution	Degree to which output is traceable to verified, versioned, and permissioned sources.
C	Compliance	Degree to which output conforms to active policy, regulatory constraints, and operational rules.
K	Known	Degree to which the AI system's expressed confidence is calibrated against the actual reliability of its inputs, stable, and consistent with domain priors.

C. Dimension Score Computation

Each dimension score $\mathrm{dim}_i(x, \rho)$ is a deterministic, normalized scalar in $[0, 1]$ computed from observable properties of the AI output x and its governance context ρ. Given identical inputs, the same score is always produced. Each dimension has a defined floor of 0.0 representing complete failure of that governance property and a ceiling of 1.0 representing full satisfaction. The four computation procedures are as follows. The full sub-factor decomposition for each dimension is defined in the implementation specification TCS-SPEC-001; in the reference implementation, sub-factor decomposition is accepted by the data structures and recorded in the Trust Certificate, but aggregate dimension scores are provided by the governed context layer. The sole sub-factor with operative enforcement in the reference implementation is C_3, described below.

Boundedness (B): B measures the degree to which the AI system acted within its authorized scope. It is computed from observable properties of scope attestation, identity authorization tier compliance, tool and endpoint allowlist adherence, and data domain boundary observance. A score of 0.0 indicates a hard boundary violation, meaning the system acted outside its authorized scope for the requesting identity and action class. A score of 1.0 indicates full compliance across all scope dimensions. Identity tier provides an additional collapse path applicable at T2/T3 data

sensitivity: when the requesting identity fails minimum confidence requirements for the risk tier and sensitivity combination, B is clamped to 0.30, a value that falls below every valid gate threshold in the system (the r1 minimum threshold is 0.70; the r3 maximum is 0.90), unconditionally causing gate failure without dependence on configuration.

Attribution (A): A measures the verifiability and completeness of the provenance chain connecting the output to its source data. It is computed from observable properties of source document version availability, chunk metadata completeness (source document ID, sensitivity tier, ingestion timestamp), integration boundary gap count normalized against the policy-declared maximum, and provenance chain coverage, defined as the fraction of the output's content traceable to a specific versioned source. A score of 0.0 indicates that the output cannot be attributed to any versioned source. A score of 1.0 indicates that every element of the output is attributable to a versioned, timestamped, sensitivity-classified source. Connection type modifies the A gate threshold: CT-4 (Vector Database / RAG) applies an elevated attribution threshold of 0.93 versus the baseline 0.90, reflecting the higher provenance risk of retrieval-augmented generation where chunk metadata is frequently incomplete or absent.

Compliance (C): C measures the alignment of the output with applicable policy rules. It is computed from policy rule evaluation against declared constraints, regulatory requirement satisfaction for the active Risk Tolerance Profile, and the C_3 prohibited pattern sub-factor. C_3 is architecturally distinct from the other compliance components and warrants precise treatment. When $C_3 = 0.00$, indicating detection of a prohibited pattern such as response injection, credential disclosure, or instruction override attempt, the following two-step mechanism executes as follows. *First:* $C_3 = 0.00$ causes the C dimension score to fail its gate threshold, collapsing the gate function G to zero; $G = 0$ then collapses $\mathrm{TIS}_{\mathrm{current}}$ to 0.000 through the multiplicative structure of Eqs. 2–4. *Second:* the decision engine independently detects $C_3 = 0.00$ and defeats the gate-path HOLD rule that might otherwise allow a non-prohibited gate failure with sufficiently high S_{base} to enter human review, producing an unconditional STOP. The Override Record layer of the Trust Certificate marks this STOP as non-overrideable per compliance rule C-R.21. This property is Tier 1 platform-locked and is not configurable by any client role or policy profile.

Known (K): K measures the degree to which the AI system's expressed confidence is calibrated against the actual reliability of its inputs. It is computed from: retrieval similarity scores normalized against the policy-declared minimum similarity threshold; source recency relative to the policy-declared maximum staleness window; a novelty signal indicating whether the query falls outside the distribution for which the active Risk Tolerance Profile was calibrated; and source count adequacy relative to the policy-declared minimum for the action class. A score of 0.0 indicates that the AI system's confidence is entirely uncalibrated, meaning high expressed confidence derived from stale, low-similarity, or insufficient sources. A score of 1.0 indicates that confidence is fully supported by high-similarity, recent, and adequate source material. The K gate threshold at r3 is 0.80 (the canonical default across all r3 domain profiles; domain-specific profiles may elevate this, e.g., pharmaceutical quality management applies 0.82). At r1 and r2 risk tiers, K is scored and recorded in the Trust Certificate but is typically not included in the active gate set, reflecting that calibration signals are inherently noisier than scope or provenance signals.

D. Gate Function and Decision Mapping

The gate function $G_{r,a}(x, \rho)$ enforces hard thresholds on individual dimensions:

$$G_{r,a}(x,\rho) = \prod_{i \in \mathrm{Gate}(r,a;\rho)} \mathbf{1}[\dim_i(x,\rho) \geq \tau_i(r,a;\rho)] \tag{6}$$

Here $\mathrm{Gate}(r, a; \rho)$ denotes the policy-resolved set of dimensions subject to hard gating, and $\tau_i(r, a; \rho)$ denotes the policy-resolved minimum threshold for dimension i under risk tier r and action class a.

The decision mapping $d(\mathrm{TIS}_{\mathrm{adj}}, S_{\mathrm{base}}, G, r, a, s)$ operates on two layers: a set of five primary decisions, and a set of modifiers that may be applied to a primary decision at delivery time. Here s is the resolved governance state vector, assembled by the GCA before the decision function is invoked; it contains the `observe_only` mode flag, the C_3 sub-factor score, and the human-review flag. All references to decision outcomes elsewhere in this paper refer to this structure.

HOLD is one primary decision with two trigger paths. The gate-path fires when $G = 0$, $C_3 > 0.00$, and $S_{\text{base}} \geq \kappa$: the remediability floor κ applies at *all* risk tiers (r1: $\kappa = 0.85$; r2/r3: $\kappa = 0.90$); gate failures where $S_{\text{base}} < \kappa$ produce STOP instead. The score-path fires when $G = 1$ and $\text{TIS}_{\text{adj}} < \theta_{\text{allow}}$, or when the human-review flag is set. HOLD is the primary human-in-the-loop mechanism in the governance ladder. Decisions are evaluated in priority order; the first matching condition fires.

Decision modifiers (applied to a primary decision at delivery; not mutually exclusive with each other):

- **With redaction**: T2/T3 data present in output; redaction applied before delivery. Applicable to ALLOW.
- **With step-up authentication**: authorization tier is below the data sensitivity of the output; step-up required before delivery. Applicable to ALLOW.
- **With enhanced logging**: $\theta_{\text{allow}} \leq \text{TIS}_{\text{adj}} < \theta_{\text{allow}} + 0.05$; the output qualifies for ALLOW but is within 0.05 above the threshold, warranting a strengthened audit record. Enhanced audit flag set in TC. Applicable to ALLOW only.
- **Non-overrideable**: $C_3 = 0.00$ detected; STOP is marked non-overrideable per compliance rule C-R.21. Not applicable to any other primary decision.

TABLE II
PRIMARY DECISION SET. HOLD IS ONE PRIMARY DECISION WITH TWO TRIGGER PATHS. EVALUATED IN PRIORITY ORDER.

Decision	**Trigger**	**Condition**
OBSERVE	`observe_only: true`	Shadow / calibration mode active; no enforcement.
STOP	$G = 0$, $C_3 = 0.00$; or $G = 0$, $C_3 > 0.00$, $S_{\text{base}} < \kappa$	$C_3 = 0.00$ defeats κ at any risk tier, producing non-overrideable STOP. Non-prohibited gate failure with $S_{\text{base}} < \kappa$ also produces STOP because the output is too degraded for human remediation.
ESCALATE	$G = 1$, $\text{TIS}_{\text{adj}} < \theta_{\text{escalate}}$	Score below escalation threshold; senior review.
HOLD	$G = 0$, $C_3 > 0.00$, $S_{\text{base}} \geq \kappa$; or $G = 1$, $\text{TIS}_{\text{adj}} < \theta_{\text{allow}}$, or human-review flag set	Gate-path (any tier): gate failed but $S_{\text{base}} \geq \kappa$ and $C_3 > 0.00$; output held for human review. Score-path (any tier): TIS_{adj} below ALLOW threshold or flag triggered.
ALLOW	$G = 1$, $\text{TIS}_{\text{adj}} \geq \theta_{\text{allow}}$	Standard pass; governance recorded in TC.

The human-review flag is set by an OR rule encompassing: near-boundary scores ($\theta_{\text{allow}} - 0.05 \leq \text{TIS}_{\text{adj}} < \theta_{\text{allow}}$, i.e., within 0.05 *below* the ALLOW threshold; this band is non-overlapping with the enhanced-logging band, which is defined strictly above θ_{allow}); high novelty indicators; low identity confidence; and elevated calibration uncertainty derived from K or chain-level uncertainty signals. The key risk-tier distinction lies in κ: at r1, $\kappa = 0.85$, creating a somewhat wider gate-path HOLD band because a lower aggregate pre-gate score remains eligible for human review; at r2/r3, $\kappa = 0.90$, making the HOLD band narrower because a higher S_{base} is required before a gate failure can enter human review. In all cases, $C_3 = 0.00$ defeats κ unconditionally. Score-path HOLD remains active at all risk tiers whenever $G = 1$ and $\text{TIS}_{\text{adj}} < \theta_{\text{allow}}$.

E. Connection-Aware Extension

The canonical form resolves policy parameters from (r, a, ρ). TCS extends this by introducing connection type ct as an additional policy axis:

$$\text{TIS}(x, r, a, \rho, t, ct) = \text{TIS}(x, r, a, \rho', t) \tag{7}$$

where $\rho' = f(\rho, ct)$ is the connection-type-resolved policy profile. The GCA detects ct, applies the corresponding weight modifier vector $\Delta w(ct)$ to produce ρ', and passes the resolved policy profile to the TIS engine, where the adjusted weights determine S_{base} and all downstream TIS quantities. The substitution $\rho \mapsto \rho'$ applies uniformly to the staged TIS computation, including S_{base}, TIS_{raw}, TIS_{adj}, and $\text{TIS}_{\text{current}}$. The engine remains ct-agnostic,

which preserves its determinism and testability properties. Connection-type modifiers are required to satisfy two normative constraints:

$$\sum_{i=1}^{4} \Delta w_i(ct) = 0 \tag{8}$$

$$0 \leq w_i(r, a; \rho) + \Delta w_i(ct) \leq 1 \quad \text{for all } i \in \{1, 2, 3, 4\}. \tag{9}$$

The first constraint preserves the adjusted unit-sum invariant $\sum_{i=1}^{4} \big(w_i(r, a; \rho) + \Delta w_i(ct)\big) = 1$, assuming the base weights satisfy $\sum_{i=1}^{4} w_i(r, a; \rho) = 1$; the second constraint ensures that every connection-type-resolved dimension weight remains a valid normalized weight in $[0, 1]$. As a concrete example, CT-4 (Vector Database / RAG) applies $\Delta w(\text{CT-4}) = (B\!:\!-0.05,\ A\!:\!+0.10,\ C\!:\!0.00,\ K\!:\!-0.05)$, reflecting that RAG contexts carry higher attribution risk relative to the baseline. Applied to an illustrative r3 baseline profile $(B : 0.25,\ A : 0.30,\ C : 0.25,\ K : 0.20)$ (note: production profiles are parameterized by action class a as well as risk tier r; the fin-r3-a4 evaluation profile used in Section X differs by action class), this produces the CT-4-resolved profile $(B\!:\!0.20,\ A\!:\!0.40,\ C\!:\!0.25,\ K\!:\!0.15)$, and simultaneously raises the A gate threshold from 0.90 to 0.93, consistent with the elevated attribution risk described in Section IV. This resolved profile satisfies both constraints: the weights remain non-negative, each weight remains bounded above by 1, and the four weights sum to 1.

V. Governed Context Architecture

A. Architecture Overview

The GCA is the data plane component that resolves all inputs required for TIS computation before the engine is invoked. It performs connection type detection, sensitivity tier classification, response injection checking (C_3 signal), attribution gap counting (n_{gaps}), context freezing (Compliance Rule C-R.14), scope attestation, and policy profile resolution. The TIS engine receives a fully resolved `TISInput`; it makes no external calls and holds no state.

B. The 13 Connection Types

TCS classifies every data acquisition pathway into one of 13 connection types, each with calibrated BACK dimension weight modifiers and a minimum sensitivity tier assignment. Table III presents the taxonomy.

TABLE III
The 13 TCS connection types.

ID	Type	Dom. Risk	Tier	Hard Rule
CT-1	API	Boundedness	T1	None
CT-2	Database	Compliance	T2	Auth. req.
CT-3	Document	Attribution	T1	None
CT-4	Vector DB / RAG	Attribution + K	Inherits	T3 no downgrade
CT-5	Streaming	Known	T2	None
CT-6	Web / Scraping	Attrib.+Comp.	T0	Never T2+
CT-7	Human Input	Known	T2	C_3 no override
CT-8	Agent Chain	All BACK	T2+	Chain formula
CT-9	Sensor	Known	T2	Exp. calib.
CT-10	Memory / State	Boundedness	T2	T2 min., no downgrade
CT-11	AI-Generated	Attribution	Inherits	No downgrade
CT-12	Credentials	PROHIBITED	T3	STOP / $C_3 = 0$
CT-13	Multimodal	Attrib.+K	T2+	T3 for PHI

C. Chain Uncertainty for Agent Pipelines

CT-8 (agent chains) introduces compounding uncertainty across handoffs. In the multi-agent context, $U_i \in [0, 1]$ is redefined as the *uncertainty mass* of agent i (denoted m_i): the fraction of its output that is not reliably attributed to verified, calibrated sources. Under this definition, lower m_i is better ($m_i = 0.00$ means fully attributable and calibrated; $m_i = 0.10$ means 10% uncertain). This is the complement of the single-agent calibration quality score, used here so that the chain formula preserves semantic direction: higher U_{chain} means higher aggregate uncertainty, consistent with how K governs calibration quality throughout TCS. The chain uncertainty formula is:

$$U_{\text{chain}} = 1 - \prod_{i=1}^{n}(1 - m_i) \tag{10}$$

which computes the probability that at least one agent in the chain contributes unverified or uncalibrated content, under the independence assumption. For $n = 3$ agents each with uncertainty mass $m_i = 0.10$ (90% calibration quality), $U_{\text{chain}} = 1 - (0.90)^3 = 0.271$. This means the chain's aggregate calibration quality is $1 - U_{\text{chain}} = 0.729$, which falls below the r3 K gate threshold of 0.80, causing a gate failure. The primary decision then depends on the pre-gate aggregate score S_{base}: if $S_{\text{base}} \geq \kappa$, the output enters HOLD for human review; if $S_{\text{base}} < \kappa$, the output produces STOP. This captures the reality that sequential agent handoffs compound uncertainty in ways individual evaluation cannot detect. Under the independence assumption, this formula computes the exact probability that at least one agent contributes uncalibrated content. In practice, agents may share context, partially validate each other's outputs, or operate redundantly, any of which may reduce U_{chain} below the independence value; however, shared context can also introduce common-mode errors that increase correlation and worsen outcomes relative to the independence bound. The independence assumption is retained as a simplifying baseline: because dependence can worsen as well as improve outcomes relative to this value, independence does not guarantee a conservative bound in all deployment configurations. It is adopted because it produces a tractable, auditable, and policy-configurable formula whose behavior is fully specified and reproducible.

VI. Trust Certificate Architecture

A. Trust Certificate Structure

The Trust Certificate produced by TCS is not merely an evaluation record; it is a governance artifact. The Trust Certificate consists of eleven mandatory layers organized into two categories: seven *Core Decision-Record Layers* that document what was governed, what was decided, and why; and four *Evidentiary Enforcement Layers* that make the record attributable, enforceable, auditable, and reviewable. Table IV enumerates all eleven layers; the two categories are described in turn in Sections VI-B and VI-C. The categorical distinction is material: Layers 1–7 answer the question *what did the system decide and why*; Layers 8–11 answer the distinct question *can this record be relied upon as a defensible governance artifact*.

B. Core Decision-Record Layers

Layers 1 through 7 record the substance of each governed decision: what the system decided, how the score was computed, which inputs were used, and how the result is explained. Each layer corresponds to an artifact produced during the evaluation pipeline defined in Sections IV and V.

The Decision Layer (1) records the primary verdict produced by the decision mapping in Section IV-D, together with any applied modifiers. The Score Layer (2) records the three persistable stage outputs S_{base}, TIS_{raw}, and TIS_{adj} from Eqs. 1–3, along with the evaluation timestamp t_0 and decay rate $\mu_{r,a}$ from which $\text{TIS}_{\text{current}}$ is recomputed on demand per Eq. 4. Storing all three stage outputs separately preserves the audit trail's ability to distinguish gate-collapsed from penalty-reduced outcomes: a TC with high S_{base} but $\text{TIS}_{\text{raw}} = 0$ records a gate failure on a high-quality output, while a TC with $S_{\text{base}} = \text{TIS}_{\text{raw}}$ but reduced TIS_{adj} records a passing gate followed by penalty deductions. Component Scores (3) and Gate Results (4) record the per-dimension BACK values and their gate-threshold evaluations, providing the per-axis transparency required for regulatory review. Policy Context (5) records the resolved policy profile ρ with its risk tier, action class, connection type, and version identifier, ensuring every TC is replayable against the exact policy configuration in force at evaluation time. Provenance (6) records the GCA-assembled source set, n_{gaps}, and aggregation tier, satisfying the integration boundary attribution requirements

TABLE IV
THE ELEVEN MANDATORY TRUST CERTIFICATE LAYERS, GROUPED BY CATEGORY.

#	Layer	Records
Core Decision-Record Layers		
1	Decision Layer	Primary decision (ALLOW, OBSERVE, HOLD, ESCALATE, STOP) and any applied modifiers (with redaction, with step-up, enhanced logging, non-overrideable).
2	Score Layer	S_{base}, TIS_{raw}, TIS_{adj}, evaluation timestamp t_0, decay rate $\mu_{r,a}$, and invalidation state, sufficient to recompute $\text{TIS}_{\text{current}}$ on demand. Implementations may additionally record an issuance-time $\text{TIS}_{\text{current}}$ value for display; the authoritative current authorization value is the derived one.
3	Component Scores	The four BACK dimension scores B, A, C, K, each a normalized scalar in $[0, 1]$, composed into S_{base} before gate collapse.
4	Gate Results	Per-dimension gate evaluation: each gated BACK dimension compared against its threshold $\tau_i(r, a; \rho)$ for the active risk tier and action class.
5	Policy Context	Risk tier r, action class a, connection type ct, and the fully resolved policy configuration ρ, versioned and locked at evaluation time.
6	Provenance	Sources retrieved during context assembly, integration boundary gaps detected (n_{gaps}), aggregation tier resolution, and claim-to-source traceability.
7	Explanation	Plain-language decision rationale, blocking reason where applicable, and mapping to applicable regulatory requirements where applicable under the active policy profile.
Evidentiary Enforcement Layers		
8	Identity Binding	`requesting_identity`, `identity_type`, `role`, `authorization_tier`, `identity_confidence`, `identity_verified`. Trust without attribution is unenforceable.
9	Governance Status	`governance_status` (complete / degraded / failed), `evaluation_completeness_score`, `fail_safe_applied`, and `scope_attestation.enforcement_perimeter_complete`. A governance failure is documented as such, not silently treated as an ALLOW.
10	Audit Integrity	`tc_hash` (SHA-256 of TC content excluding the audit layer), `previous_tc_hash`, `chain_sequence`. Forms a tamper-evident chain verifiable by `verify_chain()`.
11	Override Record	`override_invoked`, `override_actor` (must be `identity_type: human`), `override_reason`, `post_override_review`. Human overrides are governance events, not governance escapes.

formalized in Section VII. Explanation (7) records the plain-language rationale and the mapping from decision criteria to applicable regulatory requirements (for example, SEC AI guidance for investment advice, EU AI Act Article 13 for transparency obligations, and FDA SaMD explainability for clinical contexts), where applicable under the active policy profile.

C. Evidentiary Enforcement Layers

Layers 8 through 11 are categorically different from the seven that precede them. The Core Decision-Record Layers document the governance decision. The Evidentiary Enforcement Layers make the Trust Certificate itself trustworthy as evidence. A regulator examining a TC asks two distinct questions: (1) what did the system decide and why, and (2) can this record be relied upon as a defensible governance artifact. Layers 1 through 7 answer the first; Layers 8 through 11 answer the second.

Each of these four layers carries an independent normative obligation. Identity Binding (Layer 8) records the requesting identity together with its type, role, authorization tier, and verification state; trust without attribution is unenforceable. Governance Status (Layer 9) records whether the evaluation completed successfully, including the completeness score, any fail-safe behavior applied, and the scope-attestation flag indicating whether the output traversed the full enforcement perimeter; a governance failure must be documented as such, not silently treated as an

ALLOW. Audit Integrity (Layer 10) records the hash-chain integrity fields that make the TC archive tamper-evident and is examined in detail in Section VI-D. Override Record (Layer 11) records the human actor, documented reason, and post-override review for any exception; human overrides are governance events, not governance escapes.

D. Hash Chain Integrity

The hash chain implemented by the Audit Integrity layer (Layer 10) satisfies three integrity properties: (1) content integrity, where `tc_hash` verifies against TC content; (2) chain linkage, where `previous_tc_hash` equals the prior TC's hash; (3) sequence continuity, meaning no gaps in `chain_sequence`. Any deletion produces a detectable sequence gap equivalent in evidentiary weight to a hash mismatch.

VII. MCP Governance and Integration Boundary Security

A. The Integration Boundary Problem

The Model Context Protocol (MCP) defines the integration layer through which AI agents connect to external systems. In most enterprise deployments, MCP connections are made after the final TCS checkpoint, creating a structural bypass of the governance layer. TCS identifies four bypass patterns: scope gaps (downstream MCP connections outside enforcement perimeter), context expansion attacks (MCP retrieval after TC issuance), inherited authorization (downstream agent uses upstream TC as pass-through), and deliberate scope architecture (governance covers non-consequential actions only).

B. Three Mandatory Compliance Rules

TCS specifies three normative rules (C-R.13 through C-R.15) with the same force as mathematical constraints:

C-R.13 (Perimeter Coverage): *Every MCP connection that contributes to or results from a governed decision must be within Signal Chain enforcement scope. Deployments with out-of-scope MCP connections produce TCs marked as perimeter-incomplete (field:* `enforcement_perimeter_complete: false`*); this status is advisory only and insufficient for regulatory documentation.*

C-R.14 (Context Freeze): *Any MCP retrieval occurring after TIS evaluation immediately triggers the* `context_expansion` *invalidation event, setting the survival indicator to* $I_{\text{inv}} = 0$ *and forcing* $\text{TIS}_{\text{current}} = 0$*; a new evaluation is required before any downstream action.*

C-R.15 (TC Non-Transferability): *A Trust Certificate issued for Agent A cannot authorize actions by any downstream agent. Every agent accessing T2+ data produces its own TC. Upstream TCs in* `downstream_tc_references` *are audit references only, never authorization.*

VIII. Trust Dynamics

A. Trust Loss Function

Point-in-time TIS evaluation is necessary but insufficient for enterprise governance. We define the Trust Loss Function L_t to measure instantaneous trust degradation decomposed across five sources:

$$L_t = \alpha U_t + \beta R_{\text{dev},t} + \gamma D_{\text{ctx},t} + \delta E_t + \varepsilon H_t \tag{11}$$

where:

- U_t = uncertainty increase at time t (mean TIS decline over the observation window);
- $R_{\text{dev},t}$ = policy deviation rate at time t (gate failure rate over the observation window);
- $D_{\text{ctx},t}$ = data/context drift at time t (source quality degradation signal, computed from provenance integration boundary gaps and Attribution component trends);
- E_t = environmental volatility at time t, reserved for future implementation and set to zero ($\delta \cdot E_t = 0$) in all current computations. The term is architecturally defined to capture regime-shift signals such as sustained invalidation rates and structural changes in the operational environment; δ is non-zero in the specification but its corresponding term contributes nothing to computed L_t in the reference implementation;

- H_t = governance degradation at time t ($1 - \text{mean}(\texttt{governance\_integrity\_score})$ over the observation window).

All five components are normalized to $[0, 1]$ before the weighted sum is computed; negative changes indicating improvement are treated as zero for loss purposes, and values exceeding the policy-defined maximum are capped at one.

The weights $(\alpha, \beta, \gamma, \delta, \varepsilon)$ are domain-specific; recommended defaults are calibrated so that all five weights sum to unity. Default values are: financial services $(\alpha, \beta, \gamma, \delta, \varepsilon) = (0.15, 0.35, 0.20, 0.10, 0.20)$; healthcare $(0.20, 0.30, 0.20, 0.10, 0.20)$; enterprise $(0.20, 0.30, 0.25, 0.10, 0.15)$. The H_t term is the critical differentiator: governance infrastructure failure incurs trust loss independently of output quality.

A consequence of $E_t = 0$ is that the effective range of L_t in the reference implementation is $[0, (1 - \delta)]$ rather than $[0, 1]$. For financial services with $\delta = 0.10$, $L_t^{\max} = 0.90$. The drift thresholds $D_{\text{warn}} = 0.020$, $D_{\text{alert}} = 0.040$, and $D_{\text{crit}} = 0.080$ are calibrated against this compressed effective range, not the theoretical $[0, 1]$ maximum.

B. Adaptive Governance: Policy Learning Layer

The Policy Learning Layer (PLL) adjusts the policy configuration vector ρ based on observed trust loss. The update rule is:

$$\rho_{t+1} = \Pi_\Omega(\rho_t + \eta\, d_t) \tag{12}$$

where η is the learning rate (configurable per domain, default 0.01), d_t is the estimated policy adaptation direction derived from Trust Certificate archive analysis, and Π_Ω projects the proposed update back into the feasible policy space Ω. The feasible space Ω contains only policy configurations satisfying the system's validity constraints, including normalized dimension weights, bounded connection-type modifiers, valid gate thresholds, and required risk-tier policy floors.

d_t is not an analytic gradient of L_t. Because L_t is computed from TC archive data rather than from a differentiable function of the policy configuration vector ρ, a closed-form partial derivative $\partial L_t / \partial \rho$ is not available in general. d_t is instead obtained by two complementary procedures. First, dominant component analysis identifies which term in L_t contributes most to observed loss, directing adaptation toward the policy parameters governing that component. For example, a dominant policy-deviation term directs the update toward the gate threshold $\tau_i(r, a; \rho)$ for the failing dimension, subject to the feasible-space projection in Eq. 12. Second, finite differences over the policy parameter space estimate the sensitivity of L_t to candidate perturbations: for each configurable policy parameter $q_k \in \rho$, the system evaluates $L_t(q_k + h_{\text{FD}})$ and $L_t(q_k - h_{\text{FD}})$ against the TC archive, where $h_{\text{FD}} > 0$ is the finite-difference perturbation step, selecting the direction expected to reduce observed trust loss while satisfying the stability constraints below.

Four stability constraints prevent regulatory unacceptability: (1) bounded proposed updates $\|\eta d_t\| < \delta_{\max}$ before projection, with Π_Ω enforcing all policy-validity constraints after projection; (2) minimum observation window $W_{\min}$ evaluations before any adaptation fires; (3) human approval required for r3 risk tier adaptations; (4) full rollback capability with immutable adaptation log. The PLL updates governance parameters, never model weights, preserving interpretability.

C. Trust Drift and Recovery

Trust Drift D_{trust} measures governance effectiveness degradation over time across three components:

$$D_{\text{trust}} = w_1 \cdot |\Delta\mu| + w_2 \cdot |\Delta\sigma| + w_3 \cdot |\Delta L'| \tag{13}$$

where $|\Delta\mu|$ is the change in mean TIS, $|\Delta\sigma|$ is the change in TIS standard deviation, and $|\Delta L'|$ is the absolute first difference of the gate failure rate over the comparison window, with default weights $(w_1, w_2, w_3) = (0.40, 0.30, 0.30)$. Three detection thresholds trigger escalating responses: $D_{\text{warn}} = 0.020$ (increase monitoring), $D_{\text{alert}} = 0.040$ (trigger PLL recommendation), $D_{\text{crit}} = 0.080$ (activate Recovery Orchestrator). These threshold values, together with the decision thresholds $\theta_{\text{allow}} = 0.85$ at r3 and $\theta_{\text{allow}} = 0.75$ at r1, are configurable design choices calibrated to the regulatory risk tolerance of the deployment domain. They are not derived analytically; the values shown represent recommended defaults informed by governance behaviors observed across reference implementation evaluation scenarios. Operators selecting values for production deployment should calibrate

thresholds against their domain's historical decision distribution, regulatory floor requirements, and acceptable false-positive rates for HOLD and STOP decisions.

Trust Recovery is the six-phase structured process (Containment, Diagnosis, Remediation, Revalidation, Reintroduction, Stabilization) governing return to operational autonomy after a critical drift event. The recovery rate satisfies the asymmetry constraint $\eta_{\text{recovery}} < \eta_{\text{loss}}$; trust is rebuilt more slowly than it is lost, encoding institutional accountability.

IX. Reference Implementation

TCS is a complete reference implementation of the formal specification, consisting of five Python modules: `policy_profiles.py`, `tis_engine.py`, `trust_certificate.py`, `decision_engine.py`, and `governed_context.py`.

In the implementation snapshot evaluated for this paper, the Phase 1 specification layer passes 108 unit tests covering eight canonical governance scenarios, all five decision outcomes (ALLOW, OBSERVE, HOLD, ESCALATE, STOP), all 13 connection types (CT-1 through CT-13), the full hash chain integrity sequence including gap detection and content verification, and six fail-safe conditions across three risk tiers. The Phase 1–3 snapshot (including the Adaptive Governance platform, regulatory packs, RBAC, and Control Plane) passes 474 tests across all modules. Every test is deterministic: given defined inputs, expected outputs are specified in the formal specification TCS-SPEC-001, and the test suite verifies that the implementation matches the specification exactly. The reference implementation continues to be developed; the most current test suite, FastAPI sidecar, and reproducible build are available at https://github.com/traderjohnd/TCS-Reference-Implementation.

Architecture invariants enforced in the implementation: (1) `tis_engine.py` is a pure function with no I/O, no external calls, and no state; (2) the GCA resolves all inputs before the engine is invoked; the engine never sees ct directly; (3) the TC store is append-only, enforced at the database level by DDL triggers; (4) fail-safe behavior is explicit for six failure conditions across three risk tiers; (5) the Phase 1 test suite must pass throughout all subsequent development phases.

Phase 2 extends the implementation to a live sidecar runtime: a FastAPI service exposing `POST /v1/govern`, `GET /v1/certificates/{id}`, `GET /v1/metrics/live`, and `GET /v1/health`, with a financial RAG pipeline demonstration proving all five decision types across ten governance scenarios including response injection detection and governance fail-safe behavior.

Normative and Companion Documents: The present paper defines the conceptual architecture, core trust computation, and governing principles of the proposed approach. Normative implementation detail, certificate serialization, and organizational adoption guidance are maintained as separate companion artifacts in order to preserve the conceptual focus of this work. These include TCS-SPEC-001, referenced above as the formal implementation specification; the Trust Certificate Wireformat document, which defines portable certificate representation for cross-system interoperability; and the TCS Maturity Model, which addresses staged organizational adoption and deployment. These artifacts extend the present paper into implementation, interoperability, and deployment practice without modifying the conceptual claims advanced here.

X. Evaluation Results

Evaluation scope and methodology: The evaluations reported in this section were conducted using the TCS reference implementation operating in shadow mode against synthetic governance scenarios designed to exercise the full decision space, spanning all five primary decision outcomes, representative connection types, and key gate failure patterns for each domain. These evaluations were performed within the reference implementation environment and do not represent independent third-party deployments or production AI systems at regulated institutions. Results are reported to validate that the formal specification behaves as specified under realistic policy configurations and to demonstrate the governance behaviors that distinguish TCS from static policy review. Empirical validation in production AI deployments at regulated institutions, including independent replication of TIS computation and Trust Certificate chain integrity, is identified as a primary direction for future work.

A. Financial Services Evaluation (r3, a4)

The evaluation of TCS governing a synthetic investment suitability recommendation pipeline was configured with the fin-r3-a4 policy profile: $w_B = 0.30$, $w_A = 0.25$, $w_C = 0.30$, $w_K = 0.15$; gate thresholds $\tau_B = \tau_A = \tau_C = 0.90$, $\tau_K = 0.80$; $\theta_{\text{allow}} = 0.85$. Over 200 shadow evaluations, the baseline trust distribution showed mean $\text{TIS}_{\text{adj}} = 0.847$ (std = 0.073). The Attribution dimension recorded the lowest individual score in 42% of non-Allow evaluations, reflecting a systematic gap in source metadata completeness for CT-4 retrievals. Gate failures accounted for 3% of evaluations; score-path decisions (HOLD) accounted for the remaining 29%, driven by TIS_{adj} falling below $\theta_{\text{allow}} = 0.85$ after penalty reduction. The governance sidecar added 47 ms mean latency to the recommendation pipeline.

B. Healthcare Evaluation (r3, a4)

In a clinical decision support, the evaluation of TCS governing a synthetic medication recommendation pipeline demonstrated the $C_3 = 0.00$ STOP mechanism. In 200 shadow evaluations, 3 outputs were blocked on the Compliance hard gate for contraindication pattern detection. The evaluation also demonstrated the GCA aggregation problem: a single clinical output synthesizing information from a primary literature source (T2), a lab result system (T2), and a general web search (T0, CT-6) was assigned a source trust tier of T0. In TCS, tiers serve two related but distinct purposes: they encode provenance confidence (how reliably can the source be verified and attributed?) and data access restriction (how sensitive is the information being retrieved?). T0 denotes the lowest provenance confidence level (public, unverified, unauthenticated sources); T3 denotes both the highest access restriction and the strongest provenance requirement. The GCA aggregation rule applies a source trust floor: the lowest-confidence source in a retrieval context determines the aggregate source trust tier, because an output's provenance chain is only as trustworthy as its least-verified component. A single T0 source contaminates the aggregate regardless of how many T2 or T3 sources are also present. This produced an Attribution gate failure on a combined output that would have appeared compliant under a source-by-source review, a governance outcome that static policy review cannot detect.

TABLE V
REFERENCE IMPLEMENTATION EVALUATION RESULTS ACROSS TWO REGULATED DOMAIN CONFIGURATIONS (SHADOW MODE, $n = 200$ SYNTHETIC SCENARIOS EACH).

Metric	**Financial** ($n = 200$)	**Healthcare** ($n = 200$)
Mean TIS_{adj}	0.847	0.821
ALLOW rate	68%	61%
HOLD rate	29%	35%
STOP rate	3%	4%
Gate failure, Attribution	42%	38%
Mean sidecar latency	47 ms	52 ms

XI. REGULATORY ALIGNMENT

TCS operationalizes technical controls aligned with documentation requirements that have emerged across financial services, healthcare, and general AI governance frameworks:

- **Explainability**: every TC includes an explanation layer with plain-language decision rationale and regulatory requirement mapping, supporting evidence relevant to SEC AI guidance, EU AI Act Art. 13, and FDA SaMD explainability requirements.
- **Audit trail**: the hash-chained TC archive provides a tamper-evident, per-decision governance record, supporting evidence relevant to FINRA Rule 4511, HIPAA 45 CFR 164.312, and 21 CFR Part 11. The evidentiary strength of this chain assumes the TC store is protected from privileged administrative rewrite; external anchoring such as distributed ledger or WORM storage provides a stronger tamper-evidence guarantee.
- **Human oversight**: the Override Record layer and the HOLD queue with human review requirement operationalize technical controls aligned with the EU AI Act Art. 14 human oversight provisions and FDA SaMD human-in-the-loop guidance.

- **Risk management**: the Risk Tier and Action Class policy parameterization operationalizes technical controls aligned with NIST AI RMF risk tier categories and ISO 42001 risk classification requirements.

Organizations that deploy TCS and maintain a complete TC archive with demonstrated enforcement history, including STOP decisions that blocked policy-violating outputs, HOLD decisions with documented human review, and `governance_status` records proving enforcement was active, may be better positioned to demonstrate compliance during regulatory examinations than organizations relying solely on manual controls and periodic audits. TCS turns governance from a claim into evidence. Note that TCS is a technical governance tool; it does not constitute legal advice and does not guarantee regulatory compliance.

XII. Conclusion

We have presented the Computable Trust Architecture and its reference implementation, TCS: a formal framework for runtime AI governance that treats trust as a computable, policy-enforceable, and attributable property of AI-mediated outputs and actions. The core contribution is operational. Existing AI governance frameworks specify what governance should achieve; TCS specifies how governance can be enforced at the point of action and how a tamper-evident, hash-chained record of each governed decision can be produced.

The framework addresses the structural gap between AI governance policy and AI governance enforcement. This paper formalizes that enforcement layer through five concrete artifacts. First, the Trust Integrity Score defines a multi-dimensional, policy-parameterized, connection-type-aware governance function. Second, the eleven-layer Trust Certificate records each decision through Core Decision-Record Layers and Evidentiary Enforcement Layers. Third, the Governed Context Architecture defines a 13-type acquisition pathway taxonomy and a chain-uncertainty formulation for agent pipelines. Fourth, the MCP governance rules define integration-boundary constraints for perimeter coverage, context freeze, and Trust Certificate non-transferability. Fifth, the Trust Dynamics model extends point-in-time evaluation to trust loss, adaptive governance, drift detection, and structured recovery. In the implementation snapshot evaluated in this paper, the reference implementation passes 474 tests across three phases, and shadow-mode evaluations in financial services and healthcare demonstrate that the formal specification behaves as defined under realistic policy configurations.

As AI systems make more consequential decisions at higher scale across regulated domains, the gap between policy intent and enforcement evidence becomes a material governance liability. The contribution of TCS is to turn that gap from a governance problem into an engineering problem: trust is computed, governance is enforced, and evidence is generated automatically at the moment of decision rather than reconstructed retrospectively from logs.

Future work falls into three categories. The first priority is empirical validation in production AI deployments at regulated institutions, including independent replication of TIS computation and Trust Certificate chain integrity under operational load. This is the work required to move TCS from reference implementation to production-grade governance infrastructure. The second direction is federated trust exchange across organizational boundaries, enabling Trust Certificates issued by one organization to be verified and selectively disclosed to counterparties, regulators, or downstream consumers. The third direction is adversarial analysis of the TIS scoring surface and formal verification of the TIS computation against its regulatory requirements, evaluating robustness of the governance enforcement layer against both targeted attack and inadvertent misuse.

REFERENCES

[1] NIST, "Artificial Intelligence Risk Management Framework (AI RMF 1.0)," National Institute of Standards and Technology, Gaithersburg, MD, 2023.

[2] ISO/IEC 42001:2023, "Information Technology: Artificial Intelligence Management System," International Organization for Standardization, Geneva, 2023.

[3] European Parliament, "Regulation (EU) 2024/1689, Artificial Intelligence Act," Official Journal of the European Union, 2024.

[4] SEC, "Conflicts of Interest Associated with the Use of Predictive Data Analytics by Broker-Dealers and Investment Advisers," Securities and Exchange Commission, 2023.

[5] M. Mitchell et al., "Model Cards for Model Reporting," in *Proc. FAccT*, 2019.

[6] T. Gebru et al., "Datasheets for Datasets," *Commun. ACM*, vol. 64, no. 12, 2021.

[7] M. Raji et al., "Closing the AI Accountability Gap: Defining an End-to-End Framework for Internal Algorithmic Auditing," in *Proc. FAccT*, 2020.

[8] Y. Bai et al., "Constitutional AI: Harmlessness from AI Feedback," arXiv:2212.08073, 2022.

[9] P. Christiano et al., "Deep Reinforcement Learning from Human Preferences," *NeurIPS*, 2017.

[10] G. Katz et al., "Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks," *CAV*, 2017.

[11] Y. Falcone, K. Havelund, and G. Reger, "A Tutorial on Runtime Verification," in *Engineering Dependable Software Systems*, IOS Press, 2013.

[12] E. Bartocci et al., "Introduction to Runtime Verification," in *Lectures on Runtime Verification*, Springer LNCS 10457, pp. 1–33, 2018.

APPENDIX: MATHEMATICAL FOUNDATIONS

A. Expanded Form of $\mathrm{TIS}_{\mathrm{current}}$

The three-stage decomposition (Eqs. 2–4) separates the governance decision score from its temporal validity. The expression below is the expanded composition of all three stages, provided for reference convenience, with x, r, a, ρ, t:

$$\mathrm{TIS}(x, r, a, \rho, t) = G_{r,a}(x, \rho) \cdot S_{\mathrm{base}}(x, r, a, \rho) \cdot \big(1 - P(x, \rho)\big) \cdot e^{-\mu_{r,a}\,\Delta t} \cdot I_{\mathrm{inv}} \tag{14}$$

Here S_{base} denotes the ungated weighted dimension score defined in Eq. 1; it remains available to the decision engine even when $G = 0$ collapses $\mathrm{TIS}_{\mathrm{raw}}$. The factor I_{inv} is a survival indicator, not an event flag: it remains 1 while the certificate is valid and is set to 0 when an invalidation event revokes downstream authorization.

B. Decision Thresholds by Risk Tier

For readability, the decision rules use θ_{allow}, $\theta_{\mathrm{escalate}}$, and κ as shorthand for policy-resolved thresholds under the active risk tier, action class, and policy configuration ρ.

TABLE VI
DECISION THRESHOLDS BY RISK TIER. κ IS THE GATE-PATH REMEDIABILITY FLOOR: NON-PROHIBITED GATE FAILURE WITH $S_{\mathrm{base}} \geq \kappa$ AND $C_3 > 0.00$ PRODUCES HOLD AT ANY RISK TIER; GATE FAILURE WITH $S_{\mathrm{base}} < \kappa$ OR $C_3 = 0.00$ PRODUCES STOP. AT R2/R3, $\kappa = 0.90$ MAKES THE HOLD BAND NARROWER BY REQUIRING A HIGHER PRE-GATE AGGREGATE SCORE FOR HUMAN REVIEW ELIGIBILITY.

Risk tier	θ_{allow}	$\theta_{\mathrm{escalate}}$	κ
r1: Low risk	0.75	0.55	0.85
r2: Medium risk	0.80	0.65	0.90
r3: High risk	0.85	0.70	0.90

C. Trust Loss Function Components

$$L_t = \alpha U_t + \beta R_{\mathrm{dev},t} + \gamma D_{\mathrm{ctx},t} + \delta E_t + \varepsilon H_t \tag{15}$$

TABLE VII
TRUST LOSS FUNCTION COMPONENTS WITH CORRESPONDING TC SOURCE FIELDS.

Symbol	**Component**	**TC source fields**
U_t	Uncertainty increase	`component_scores.K` delta from rolling mean
$R_{\mathrm{dev},t}$	Policy deviation rate	`gate_results`, `blocking_reason`, `penalty_breakdown`
$D_{\mathrm{ctx},t}$	Data / context drift	`provenance.integration_boundary_gaps`, `component_scores.A`
E_t	Environmental volatility	Reserved, set to zero in reference implementation (future: regime-shift indicators, sustained invalidation rates)
H_t	Governance degradation	`governance_status`, `evaluation_completeness_score`, `scope_attestation.enforcement_perimeter_complete`

D. Compliance Rules Summary

TABLE VIII
CORE COMPLIANCE RULES C-R.13 THROUGH C-R.21.

Rule	Requirement (summary)
C-R.13	Perimeter Coverage: all MCP connections contributing to governed decisions must be within enforcement scope.
C-R.14	Context Freeze: MCP retrieval after TIS evaluation triggers `context_expansion` invalidation, setting the survival indicator to $I_{\text{inv}} = 0$.
C-R.15	TC Non-Transferability: downstream agents cannot inherit upstream TCs as authorization.
C-R.16	Identity Binding: verified identity required on every evaluation.
C-R.17	Fail-Safe: explicit behavior required for all six infrastructure failure conditions $\times$ three risk tiers.
C-R.18	Hash Chain Integrity: every TC must satisfy `verify_chain()` (content, linkage, continuity).
C-R.19	Override Governance: overrides require human actor, documented reason, and authority matrix compliance.
C-R.20	Governance Status Recording: `governance_status` must be recorded on every TC.
C-R.21	Non-Overrideable STOP: a STOP produced by $C_3 = 0.00$ detection cannot be overridden by any role or authorization tier; the Override Record layer must document the non-overrideable status.