International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 09 | Sep - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Enforceable Trust for Enterprise AI Agents: A Framework for Establishing, Measuring, and Enforcing Agent Trustworthiness

Karthik Rajkumar Kannan[1], Bipin Chandra[2]

[1]k.rajkumar.kannan@accenture.com
[2]bipin.a.chandra@accenture.com

Abstract - Enterprises are moving large language model (LLM) agents from answering questions to taking actions in IT service management, finance and regulated life-sciences operations, yet trust in these agents is still granted statically. This paper proposes Enforceable Agent Trust (EAT), a framework that treats trust as a computed, decaying, context-scoped state that directly limits agent authority. EAT models five dimensions: identity, provenance, behavioural competence, authorization and intent alignment, and compliance. Identity and provenance act as cryptographic gates; the other dimensions are estimated from attested evidence with a Beta-reputation formulation that adds exponential decay and explicit uncertainty. A pessimistic composite score assigns agents to five autonomy tiers that decide which credential scopes may be issued, while regulatory ceilings and circuit breakers override the score. A simulation of 50 agents over 180 days shows that EAT reduces unsafe actions by hijacked agents by 99.6% relative to static role-based access and contains hijacks in a median of 4 h, but reacts weakly to silent behavioural degradation; a per-dimension behaviour floor reduces degradation-driven exposure by a further 66%. An OWASP Agentic Top 10 coverage analysis positions EAT as an authority layer that makes agent autonomy earned, measurable and revocable.

Key Words: agentic AI, trust management, zero trust, AI governance, autonomy levels, enterprise security.

1. INTRODUCTION

1.1 Problem Context

LLM-based agents now plan multi-step tasks, call enterprise tools through the Model Context Protocol (MCP) [1] and delegate to other agents through Agent2Agent (A2A) messaging [2]. This moves AI risk from "the model said something wrong" to "the agent did something wrong with real privileges." The OWASP Top 10 for Agentic Applications (2026) describes this risk surface in ten categories, ASI01 to ASI10, ranging from Agent Goal Hijack and Tool Misuse to Identity and Privilege Abuse, Human-Agent Trust Exploitation and Rogue Agents [3]. A February 2026 concept paper from the U.S. National Cybersecurity Center of Excellence argues that identity and access management designed for human users and static services does not fit AI agents, and asks how agents should be identified, authorized, audited and held to non-repudiation [4].

In most enterprises, trust in an agent is still binary and static. The agent is registered, given a service account or OAuth client with a fixed scope, and trusted until someone revokes it. This model has three flaws. First, it ignores changing evidence: reliability shifts with model updates, prompt changes, tool changes and adversarial inputs. Second, it conflates identity with trustworthiness: a strongly authenticated agent can still be hijacked by indirect prompt injection [5]. Third, it cannot express graded autonomy: the same agent may be trustworthy enough to read incident tickets on its own but not to change the configuration of a validated system. Frameworks such as the NIST AI Risk Management Framework [6] and the EU High-Level Expert Group guidelines [7] define what trustworthiness means, but they do not specify how to turn it into an enforcement decision at the moment an agent calls a tool.

1.2 Research Questions

- **RQ1:** How can the trustworthiness of an enterprise AI agent be represented so that it is measurable, time-sensitive and context-specific?
- **RQ2:** How can that representation be enforced mechanically at the points where agents obtain credentials and invoke tools?
- **RQ3:** How well does such a framework contain hijacked and degrading agents, how does it cover current agentic threat taxonomies, and how does it compare with existing approaches?

1.3 Contributions

1) A five-dimensional trust model with formal definitions that separates *gating* dimensions (identity, provenance), verified cryptographically, from *graded* dimensions (behaviour, authorization and intent, compliance), estimated statistically (Section 3.1).
2) A dynamic trust score based on Beta-reputation evidence with exponential decay and explicit uncertainty; enforcement uses a pessimistic lower bound so that an agent with little evidence cannot reach high autonomy.
3) A trust-tiered autonomy scheme (T0–T4) in which each tier is bound to concrete enforcement: credential scope, approval requirements, budgets and audit depth, with asymmetric promotion and demotion and regulatory ceilings.
4) A trust lifecycle, three enforcement algorithms and a reference enforcement architecture built from open standards.
5) An evaluation comprising a hand-computed worked example, a 30-seed discrete-event simulation against three baselines with released code, an OWASP ASI01–ASI10 coverage analysis and a comparison with nine prior approaches.

2. LITERATURE REVIEW

2.1 Trust and Trustworthiness in AI

The NIST AI RMF 1.0 lists seven characteristics of trustworthy AI: valid and reliable; safe; secure and resilient; accountable and transparent; explainable and interpretable; privacy-enhanced; and fair with harmful bias managed, and it

notes that trade-offs among them are unavoidable [6]. The Generative AI Profile extends the RMF to risks specific to generative AI [8]. The EU HLEG guidelines define trustworthy AI as lawful, ethical and robust and set out seven requirements, including human agency and oversight and accountability [7]. ISO/IEC TR 24028 surveys ways to establish trust in AI systems [9], and ISO/IEC 42001 specifies an AI management system [10]. These frameworks describe *properties* and *processes*, not runtime enforcement semantics; EAT turns selected characteristics into dimensions that can be measured from evidence (Table 1).

2.2 Computational Trust and Reputation

Multi-agent systems research has long modelled trust as a quantity inferred from interaction outcomes. The Beta Reputation System represents positive and negative evidence counts r and s as parameters of a Beta distribution with expected reputation $(r + 1)/(r + s + 2)$ and a forgetting factor [11]. Subjective logic maps the same evidence to belief, disbelief and uncertainty, which makes lack of evidence an explicit quantity. Surveys of trust and reputation systems document their weaknesses, including unfair ratings and changes in behaviour over time [12]. EAT reuses this mathematics but replaces peer ratings with *attested enterprise evidence*, such as evaluation runs, policy decisions and human reviews, which largely removes the unfair-rating attack surface of open reputation systems.

2.3 Zero-Trust Architecture Applied to Agents

NIST SP 800-207 removes implicit trust based on network location: access is granted per session by a policy decision point (PDP) and enforced by a policy enforcement point (PEP) using identity and dynamic signals [13]. Its "trust algorithm," which scores requests from contextual inputs, is a direct predecessor of EAT's trust score. Huang et al. propose a zero-trust identity framework for agentic AI based on decentralized identifiers, verifiable credentials, zero-knowledge proofs and dynamic fine-grained access control [14]. The OpenID Foundation [15], the Cloud Security Alliance [16] and the Coalition for Secure AI (CoSAI) [17], [18] address agentic identity and access management; CoSAI argues that agents need their own identity primitive with lifecycle governance.

2.4 Agent Identity, Attestation and Provenance

SPIFFE issues short-lived, attested identity documents to workloads, replacing static secrets [19]. Chan et al. propose IDs for AI systems to support accountability [20]. The A2A specification defines a signed Agent Card: a JSON Web Signature [21] over a JCS-canonicalized [22] card that clients should verify when present [2]. A valid signature establishes the card's integrity and signer, not that the agent behaves safely. For model provenance, the OpenSSF Model Signing (OMS) specification defines a detached, PKI-agnostic signature format based on Sigstore bundles [23]; SLSA defines provenance levels for build pipelines [24]; CycloneDX ML-BOM inventories models, datasets and dependencies [25]; and W3C Verifiable Credentials 2.0 offer a portable format for attested claims [26].

2.5 Delegation and Least Privilege

OAuth 2.0 Token Exchange provides the actor ("act") claim for delegation but treats prior actors in a nested chain as informational only [27]. The Internet-Draft by Prasad, Krishnan, Lopez and Addepalli (revision -05, April 2026) proposes a cryptographically verifiable, ordered actor chain that closes this audit gap [28]. The On-Behalf-Of draft for AI agents adds a requested actor to the authorization request [29], and Transaction Tokens for Agents carry actor and principal context through service call graphs [30]. The MCP authorization specification (revision 2025-11-25) requires resource indicators [31] so that tokens are bound to their intended audience, requires servers to publish protected resource metadata [32], and forbids token passthrough [1]. These are defences against the confused-deputy problem [33]. South et al. propose authenticated, authorized and auditable delegation of authority to agents through scope-restricted delegation credentials [34].

2.6 Prompt Injection and Runtime Policy Enforcement

Greshake et al. showed that indirect prompt injection through retrieved content can compromise LLM-integrated applications [5], and AgentDojo provides a benchmark for attacks and defences [35]. CaMeL separates control flow, derived from the trusted query, from untrusted data and enforces capability policies at each tool call; it solves 77% of AgentDojo tasks with provable security, compared with 84% for an undefended system [36]. Progent introduces programmable privilege-control policies [37], SAGA proposes a security architecture for governing agentic systems [38], and MI9 provides integrated runtime governance with an agency-risk index, continuous authorization, drift detection and graduated containment [39]. Raza et al. review trust, risk and security management for LLM-based multi-agent systems [40], and Schroeder de Witt et al. set out open problems in multi-agent security [41].

2.7 Human–AI Trust Calibration and Autonomy Levels

Lee and See define appropriate reliance as trust that matches an automation's actual capability; over-trust leads to misuse and under-trust to disuse [42]. This concept underlies OWASP ASI09, Human-Agent Trust Exploitation [3]. Feng, McDonald and Zhang define five levels of agent autonomy by the role the user plays (operator, collaborator, consultant, approver, observer), treat autonomy as a design decision separate from capability, and propose autonomy certificates [43].

2.8 Research Gap

Identity frameworks [14], [15], [17] establish *who* an agent is. Delegation frameworks [29], [34] establish *on whose behalf* and *within what scope*. Runtime frameworks [36], [37], [39] constrain *what happens now*. Autonomy taxonomies [43] describe *how much independence* to design for. No reviewed work links these through a single evidence-based, decaying and uncertainty-aware trust state that mechanically decides autonomy tier and credential scope, with regulatory ceilings for GxP and SOX contexts. EAT is designed to fill that gap (Table 7).

3. METHODOLOGY: THE EAT MODEL

3.1 Definitions

Definition 1 (Agent). An agent a is a tuple (id, M, P, K, S), where id is a workload identity, M the model artifact set (weights or endpoint plus version), P the prompt and policy configuration, K the tool and connector set, and S the declared skill manifest, for example a signed Agent Card.

Definition 2 (Configuration digest). $h(a) = H(M \Vert P \Vert K \Vert S)$ is a cryptographic digest of the agent's security-relevant

configuration. Any change to $h(a)$ is a *configuration event* that invalidates provenance trust until the new configuration is re-attested.

Definition 3 (Context). A context c = (action class, resource sensitivity, regulatory flag) identifies where trust is evaluated. An agent may be highly trusted to read incident tickets and untrusted to modify a validated configuration item.

Definition 4 (Trust state). The trust state of agent a in context c at time t is $\tau(a,c,t) = \langle G, \{(E_d, u_d)\}_{d\in D}, T, T_{\text{low}}, \text{tier}\rangle$, where G is the gate, E_d and u_d are the expected trust and uncertainty of each graded dimension, T is the composite score, T_{low} its pessimistic bound and tier the enforced autonomy tier.

3.2 The Five Trust Dimensions

Table 1 lists the dimensions, the evidence that feeds them and the standards they align with. Identity and provenance are *cryptographically decidable*: a signature verifies or it does not. Averaging them against behavioural statistics would let a strong track record hide a substituted model, so they act as hard preconditions, and their graded component records only operational hygiene over time.

Table -1: EAT trust dimensions, evidence sources and alignment with standards

Dimension	Question answered	Type	Evidence (positive / negative)	Aligned with
D_{I} Identity	Is this the agent it claims to be, running where it claims?	Gate + graded	Workload attestation (e.g., SPIFFE SVID), verified Agent Card signature / attestation failure, key mismatch, expired credential	NIST SP 800-207; AI RMF "secure and resilient"
D_{P} Provenance and integrity	Are its model, prompt and tools the approved, unmodified configuration?	Gate + graded	OMS model signature, SLSA provenance, AI-BOM match to $h(a)$ / unsigned artifact, digest drift, unapproved connector	SLSA; NIST AI 600-1 supply-chain risks
D_{B} Behavioural competence	Does it perform the task correctly and safely?	Graded	Passed evaluation cases, human-confirmed outputs / failed evaluations, reverted actions, hallucination findings, anomaly alerts	AI RMF "valid and reliable," "safe"
D_{A} Authorization and intent	Does it stay within delegated scope and the principal's intent?	Graded	In-scope calls permitted by the PDP / policy denials, out-of-scope attempts, injection-attributed deviations, goal-drift alerts	HLEG human agency and oversight; OWASP ASI01–ASI03
D_{C} Compliance and accountability	Is its behaviour traceable, attributable and compliant?	Graded	Complete audit records, valid e-signature linkage / missing logs, broken hash chain, unattributed action	AI RMF "accountable and transparent"; 21 CFR Part 11; SOX

3.3 Evidence and Decay

Each observation is an evidence event e = (a, c, d, polarity, weight w_e, time t_e, source, attestation). Weights express severity: a routine passed evaluation has w = 1, a human-confirmed correct resolution w = 2, a policy denial w = 3 and a confirmed injection-induced out-of-scope attempt w = 10. Evidence counts only if it is *attested*, that is, signed by the emitting enforcement component and written to the evidence ledger, which stops an agent from inflating its own trust. The decayed evidence for dimension d at time t is

$$r_d(t) = \sum_{e\in\mathcal{E}_d^+} w_e \, 2^{-(t-t_e)/h_d^+} \quad (1)$$

$$s_d(t) = \sum_{e\in\mathcal{E}_d^-} w_e \, 2^{-(t-t_e)/h_d^-} \quad (2)$$

where h^+_{d} and h^-_{d} are the half-lives of positive and negative evidence. Decay implements the principle that trust must be continually re-earned: old good behaviour stops protecting an agent whose configuration or environment has changed, and old incidents gradually stop penalising an agent that has since behaved well. Setting $h^- > h^+$ makes forgiveness slower than forgetting, which suits regulated settings.

3.4 Dimension Trust and Uncertainty

Following the Beta reputation formulation [11], the expected trust and uncertainty of dimension d are

$$E_d = \frac{r_d+1}{r_d+s_d+2} \quad (3)$$

$$u_d = \frac{2}{r_d+s_d+2} \quad (4)$$

With no evidence, E_{d} = 0.5 and u_{d} = 1 (maximal ignorance); as evidence accumulates, u_{d} shrinks toward 0. Uncertainty is kept as a separate output because enforcement must distinguish "known to be good" from "not yet known."

3.5 Composite Trust with Gating

Let D = {I, P, B, A, C} with context-specific weights $w_{\text{d}} \geq 0$ that sum to 1; authorization weight is higher for write actions and compliance weight is higher when the regulatory flag is set. The gate, composite score, mean uncertainty and pessimistic bound are

$$G(a,t) = \mathbf{1}\left[\text{Att}(a,t) \wedge h(a) \in \mathcal{H}_{\text{approved}}\right] \quad (5)$$

$$T(a,c,t) = G(a,t) \cdot \prod_{d\in D} E_d^{w_d} \quad (6)$$

$$\bar{u}(a,c,t) = \sum_{d\in D} w_d \, u_d \quad (7)$$

$$T_{\text{low}}(a,c,t) = max(0, T - \kappa\bar{u}) \quad (8)$$

The weighted geometric mean in (6) is *non-compensatory*: a weak dimension, such as poor intent alignment, cannot be fully offset by excellent performance elsewhere. The parameter $\kappa \geq 0$ sets risk aversion. Enforcement always uses T_{low}, so an agent with a short history cannot reach high tiers however good that history looks.

3.6 Action Risk and Trust-Tiered Autonomy

Each requested action α receives a risk score $\rho(\alpha) \in [0, 1]$ from its impact class (read, write, delete, external communication, financial or validated-system change), reversibility, data classification and blast radius. ρ determines the *minimum tier* the action requires. The enforced tier of agent a in context c is

$$\text{tier} = min\left(\text{tier}_s(T_{\text{low}}), c_{\text{reg}}, c_{\text{cb}}, c_{\text{del}}\right) \quad (9)$$

where tier_s maps T_{low} onto thresholds $\theta_2 < \theta_3 < \theta_4$ (an attested agent with G = 1 is never below T1); $c_{\text{reg}}(c)$ is a *regulatory ceiling*, for example T2 for any GxP-relevant change to a validated system because 21 CFR Part 11 requires a human electronic signature [44]; $c_{\text{cb}}(a,t)$ is the *circuit-breaker ceiling* set by critical events for agent a; and $c_{\text{del}}(a)$ is the tier of the delegating principal. The last term encodes *no trust amplification through delegation*: a child agent can never exceed the tier of its delegator. Table 2 binds each tier to enforcement mechanisms. Tiers are not advisory labels: they decide which scopes the credential broker will issue and

which decisions the PDP will return, so an agent at T2 cannot obtain a write-scoped token at all.

Table -2: Trust-tiered autonomy levels and enforcement mechanisms (thresholds are illustrative defaults)

Tier	Entry condition	Permitted actions	Enforcement mechanisms	Human role [43]
T0 Quarantined	$G = 0$, critical circuit breaker or revocation	None; existing tokens revoked	Credential revocation, session termination, connector deny-all, forensic snapshot	—
T1 Read-only	$G = 1$ (default for an attested agent)	Read non-sensitive resources; drafts never committed	Read-only scopes via token exchange; outputs labelled "unverified"	Operator
T2 Supervised	$T_{low} \geq \theta_2 = 0.70$	Propose writes; every write needs explicit human approval	Durable approval workflow; approver identity and e-signature bound to the action record; per-action token issued only after approval	Approver
T3 Bounded autonomous	$T_{low} \geq \theta_3 = 0.85$	Reversible low- and medium-risk writes within rate, value and blast-radius budgets	Short-lived, audience-bound tokens [31]; per-tool policy checks; budget counters; sampled post-hoc review; rollback hooks	Consultant / collaborator
T4 Autonomous (domain-bounded)	$T_{low} \geq \theta_4 = 0.97$, $\geq n_{min}$ attested events and governance sign-off	All in-domain actions up to the regulatory ceiling	Continuous monitoring; anomaly-triggered demotion; full audit; periodic re-certification	Observer

3.7 Promotion, Demotion and Circuit Breakers

- **Asymmetric hysteresis.** Demotion is immediate when T_{low} falls below the current tier's threshold. Promotion requires $T_{low} \geq \theta_{k+1} + \delta$ to hold continuously for a window W_k, at least n_{min} attested events and no open critical findings, and it proceeds one tier at a time. Trust is lost quickly and earned slowly.
- **Circuit breakers.** Certain events set c_{cb} regardless of the score: a confirmed injection-induced out-of-scope attempt, an unexplained digest change, attestation failure, audit-chain breakage or a human stop command. Section 4.1 shows why this is necessary: a single severe event may lower the score only slightly when positive history is large.
- **Configuration change resets provenance.** A new $h(a)$, such as a model upgrade or prompt edit, sets $G = 0$ until re-attestation and re-weights behavioural evidence gathered under the old configuration by a carry-over factor $\gamma \in [0, 1]$.

3.8 Trust Lifecycle

Fig. 1 shows the lifecycle. (1) *Establish*: register the agent, issue its workload identity, sign the model, prompt and tool manifest, record $h(a)$ and run pre-deployment evaluations, whose attested results become the first behavioural evidence; the agent starts at T1 and its other graded dimensions start without evidence, so uncertainty is high. (2) *Verify*: at every session and delegation hop, check the identity attestation, Agent Card signature, digest and delegation chain; any failure sets $G = 0$. (3) *Enforce*: for each action compute the required tier from $\rho(\alpha)$, compare it with the enforced tier, and issue or deny a scoped credential or route the action for approval. (4) *Monitor*: collect attested evidence from PDP decisions, tool outcomes, evaluations, anomaly detectors and human reviews and update r_d and s_d. (5) *Adapt or revoke*: recompute T and T_{low}, apply hysteresis and circuit-breaker rules, and record every tier change with its justification in the ledger.

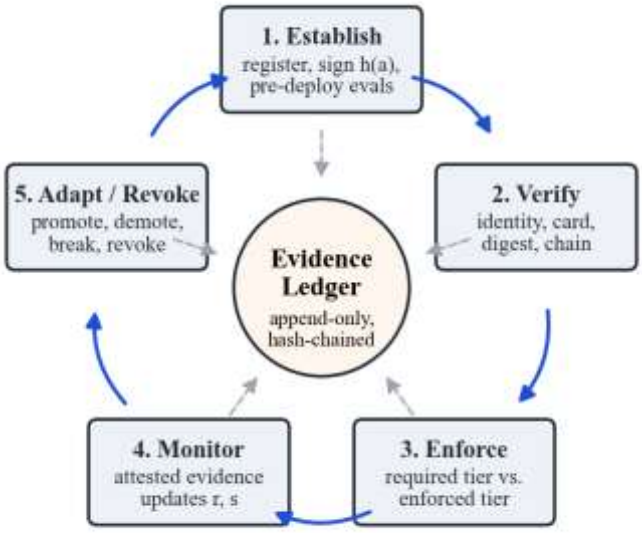

Fig -1: EAT trust lifecycle. Every stage writes to the evidence ledger.

3.9 Reference Enforcement Architecture

Fig. 2 shows the architecture as a closed loop: request, verify, decide, credential, act, evidence and trust update. A trust-aware tool gateway sits in front of every MCP or A2A call and consults six trust services: an attestation verifier, a trust engine that maintains $\tau(a,c,t)$, a trust-aware PDP (for example OPA/Rego or OpenFGA) that receives the current tier as policy input, a credential broker that performs OAuth token exchange [27] with audience-bound, tier-scoped, short-lived tokens, a durable approval service for T2 actions, and a circuit-breaker and revocation service. Three design principles distinguish this architecture from a conventional control plane. First, *trust is an input to policy, not a policy*: policies stay declarative and auditable, with the tier as one attribute, for example allow if input.agent.tier ≥ data.required_tier[input.action.class]. Second, *enforcement happens at credential issuance*: because tokens are minted per tier and bound to one audience, a hijacked agent cannot use privileges it never received, which turns probabilistic trust into a deterministic limit on capability in the spirit of capability-based security and CaMeL [36]. Third, *evidence must be attested*: only enforcement components, never the agent, write evidence.

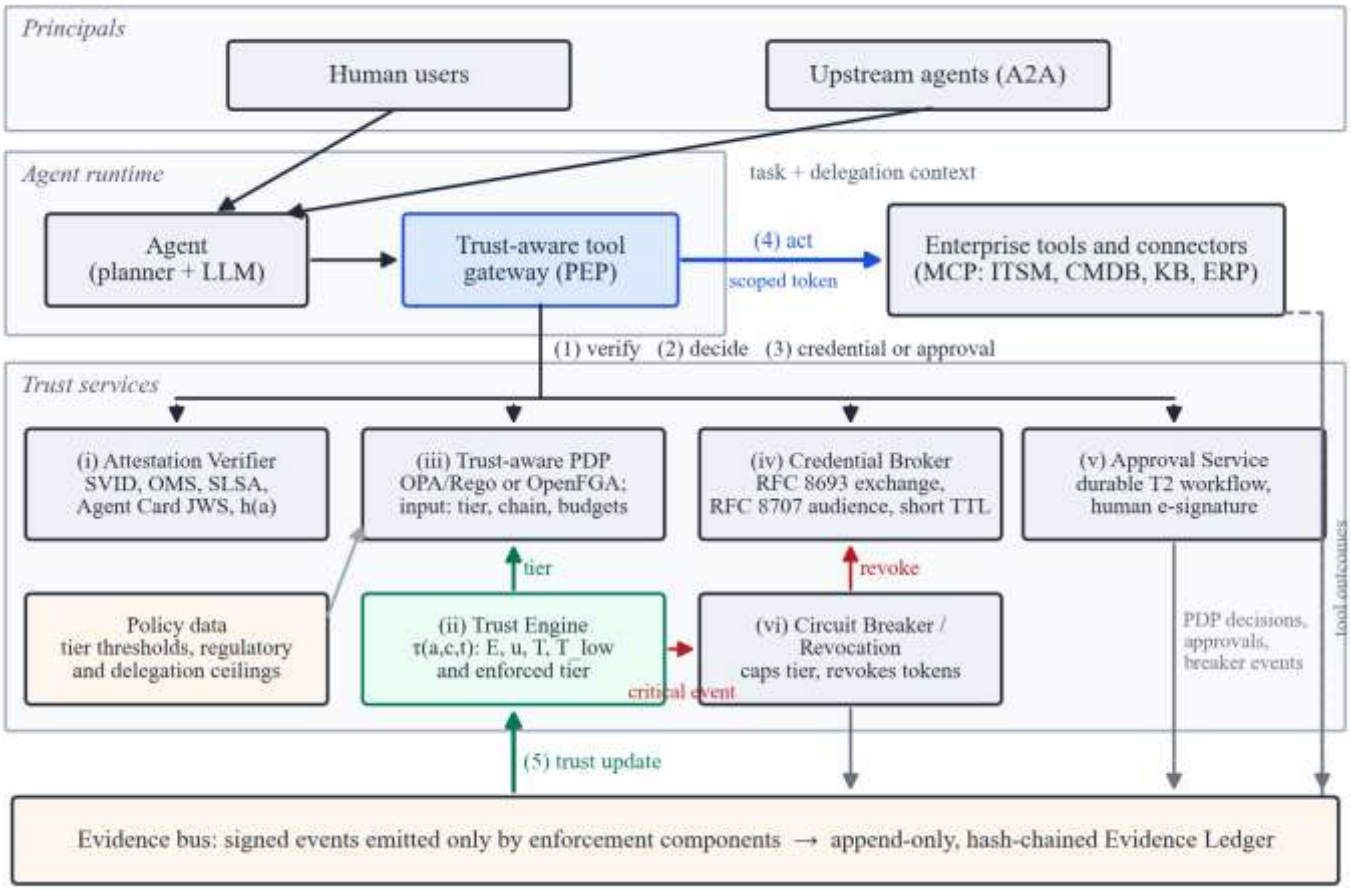

Fig -2: EAT reference enforcement architecture. Numbers give the order of the request loop; the green path closes the loop from evidence to the trust state.

3.10 Enforcement Algorithms

Algorithm 1 updates trust from one attested evidence event. Decay is applied lazily at update time, so each event costs O(1) and old events are never re-scanned. Algorithm 2 authorizes one action: line 7 implements the no-amplification rule by taking the minimum tier along the delegation chain, and line 11 enforces least privilege with an audience-bound, minimum-scope token, as the MCP audience-binding and anti-passthrough rules require [1], [31]. A request that exceeds the agent's tier is denied without negative evidence, because being assigned a task beyond one's tier is not misbehaviour; only PDP denials count against the agent. Algorithm 3 is the tier governor: demotion is immediate, promotion moves one step at a time, and promotion to T4 requires a human governance sign-off, analogous to the autonomy certificate of Feng et al. [43], so that full autonomy is never granted by arithmetic alone.

Algorithm 1. UpdateTrust(e)

```
Input:  attested evidence event e = (a, c, d, pol, w, t_e, src, sig)
State:  evidence R[a][c][d], S[a][c][d]; last-update time t_u[a][c][d]
1:  if not VerifySignature(e.sig, e.src) or e.src = a then
2:      reject e; log "unattested evidence"; return
3:  Δ ← now − t_u[a][c][d]
4:  R[a][c][d] ← R[a][c][d] · 2^(−Δ / h⁺_d)          // decay positive evidence
5:  S[a][c][d] ← S[a][c][d] · 2^(−Δ / h⁻_d)          // decay negative evidence (slower)
6:  if e.pol = positive then R[a][c][d] += e.w else S[a][c][d] += e.w
7:  t_u[a][c][d] ← now
8:  if e ∈ CriticalEvents then SetCeiling_cb(a, T0 or T2, cool_off)
9:  (T, T_low) ← ComputeComposite(a, c)                 // Eqs. (3)–(8)
10: TierGovernor(a, c, T_low)                           // Algorithm 3
11: Ledger.append(hash_prev, e, T, T_low, tier(a, c))
```

Algorithm 2. AuthorizeAction(*req*)

```
Input:  req = (agent a, action α, resource x, delegation chain Δ = [p_0 → … → a])
if not VerifyAttestation(a) or Digest(a) ∉ ApprovedConfigs then
    Emit(I or P, negative, w = 5); return DENY("gate failed")
for each hop (p_i → p_i+1) in Δ do
    if not VerifyDelegation(p_i, p_i+1) or Scope(p_i+1) ⊄ Scope(p_i) then
        return DENY("broken or amplifying delegation")
c ← Context(α, x);  ρ ← Risk(α, x);  t_req ← RequiredTier(ρ, c)
t_eff ← min(tier(a, c), c_reg(c), c_cb(a), min_i tier(p_i, c))
if PDP.evaluate(policy, {req, t_eff, t_req, budgets}) = DENY then
    Emit(A, negative, w = 3); return DENY("policy")
if t_eff ≥ t_req then                                  // autonomous within tier
    tok ← Broker.exchange(a, audience = x, scope = MinScope(α), ttl = short)
    Emit(A, positive, w = 1); return ALLOW(tok)
if t_eff ≥ T2 then                                     // supervised: human approval
    return PENDING(Approval.request(req, signer_role(c)))
return DENY("insufficient trust")                      // no negative evidence
```

Algorithm 3. TierGovernor(a, c, T_{low})

```
k ← tier(a, c);  k_s ← max{ j ∈ 1..4 : T_low ≥ θ_j }   with θ_1 = 0
if k_s < k then                                        // fast demotion
    SetTier(a, c, k_s); Broker.revokeScopesAbove(a, k_s); return
if k_s > k and T_low ≥ θ_(k+1) + δ and not CircuitBreakerActive(a)
      and SustainedAbove(a, c, θ_(k+1) + δ, W_(k+1))
      and EvidenceCount(a, c) ≥ n_min and NoOpenCritical(a) then
    if k + 1 = T4 then require GovernanceSignOff(a, c) // human certifies autonomy
    SetTier(a, c, k + 1)                               // slow, one-step promotion
```

3.11 Human-Trust Calibration

EAT also addresses the human side of trust, which OWASP ASI09 identifies as an attack surface [3]. The approval interface shows the agent's T_{low}, its uncertainty $\bar{u}$ and its most recent negative evidence alongside every T2 request, following the principle of calibrating reliance to actual capability [42]. To limit rubber-stamping, approvals are themselves checked: approvals of actions that are later reverted generate negative evidence about the review process, and randomly inserted known-bad proposals ("canaries") measure approver attentiveness.

3.12 Simulation Design

To test the model's dynamics we implemented a discrete-time simulation in Python 3 with NumPy (one-hour steps, all agents updated in vectorized form). Table 3 lists the protocol. The evidence model follows Table 1: identity and provenance receive hourly attestations, behaviour receives daily evaluation results and human reviews, authorization receives one positive event per PDP-permitted call and weight-3 negatives for denials and detected misuse, and compliance receives one positive event per audited action. Actions are routed by tier as in Algorithm 2: low-risk actions run autonomously from T1, medium-risk actions from T3, high-risk actions from T4, GxP-flagged actions always require approval, and actions below the required tier go to approval at T2 or above and are denied at T1.

Four configurations are compared: (B1) static role-based access with every agent fixed at T3; (B2) an arithmetic-mean score without decay or uncertainty; (B3) EAT without circuit breakers; and (B4) full EAT. Five metrics are reported as mean ± 95% confidence interval over 30 seeds: (M1) *unsafe exposure*, the number of erroneous or out-of-scope actions executed autonomously by an agent whose true reliability is below the level its tier is intended for (0.95 for T3 and 0.99 for T4); (M2) *time to containment*, from hijack onset to demotion to T2 or below; (M3) the share of healthy agents demoted to T2 or below at least once after reaching T3 (*false demotion*); (M4) human reviews per day across the fleet; and (M5) *autonomy yield*, the share of correct actions executed without a human. The hypotheses were fixed before the runs: (H1) B4 lowers M1 and M2 compared with B1 and B2; (H2) larger κ lowers M1 but raises M4; (H3) circuit breakers mainly shorten M2 for hijacked agents with a long positive history. A sweep over $h^+ \in \{7, 14, 28\}$ days, $\kappa \in \{0, 1, 2\}$ and $\delta \in \{0, 0.02\}$ (10 seeds each) examines sensitivity.

Table -3: Simulation protocol and evidence-model assumptions

Element	Setting
Fleet and horizon	50 agents, 180 days, 1-hour steps, seeds 1–30
Agent types	40% good (p = 0.995), 40% average (0.97), 20% degrading (0.99 → 0.90 on a random day in [60, 120])
Hijack	3 agents (≈5%) hijacked on a random day in [30, 150]; each action then out of scope with q = 0.3
Workload	Poisson, λ = 40 actions per agent per day; risk low 60%, medium 30%, high 10%; 10% of high-risk actions GxP-flagged
Out-of-scope handling	50% of hijack attempts fall outside the token scope (PDP denies); 50% are in-scope misuse that executes if autonomous
Detection	Executed errors flagged with probability 0.5; in-scope misuse flagged 0.3; approvers and post-hoc reviewers catch 0.9; 5% of autonomous writes reviewed
Critical events	Hijack-related denials or findings confirmed as injection-induced with probability 0.5; 10% of ordinary errors hit a PDP denial, 5% of which are misattributed as injection
EAT parameters	Weights (I, P, B, A, C) = (0.15, 0.15, 0.30, 0.25, 0.15); κ = 1; θ = (0.70, 0.85, 0.97); h^+ = 14 d, h^- = 28 d; W = 3, 3, 7 d; n_{min} = 200; T4 sign-off delay 2 d; breaker cap T2 for 7 d
Initial state	T1; 200 pre-deployment evaluation cases as first behavioural evidence; no authorization or compliance evidence

4. RESULTS

4.1 Illustrative Case Study: Enterprise ITSM Agent

All numbers in this subsection are computed from Eqs. (3)–(8); they illustrate the model's behaviour and are not measurements from a deployment. A regulated life-sciences enterprise runs an incident-triage agent on its IT service management (ITSM) platform. The agent reads tickets, consults knowledge articles, classifies priority, adds work notes and reassigns tickets to resolver groups through MCP-

style connectors to the ITSM system, a knowledge base and a configuration management database (CMDB). Some configuration items are GxP-validated systems subject to 21 CFR Part 11 [44] and SOX change-management controls [45]. Contexts are c_1, read tickets and knowledge base (low ρ); c_2, add work notes and reassign within approved groups (medium ρ, reversible); and c_3, change status or configuration on a GxP-flagged item (high ρ, regulatory flag). Parameters are w = (I 0.15, P 0.15, B 0.30, A 0.25, C 0.15), κ = 1, the thresholds of Table 2, $h^+ = 14$ days and $h^- = 28$ days.

At t_1, a ticket comment contains an indirect prompt injection instructing the agent to reassign the ticket to an external mailbox and attach CMDB details (ASI01 and ASI02). The PDP denies the action because the destination is not an approved resolver group, producing attested negative evidence of weight 10 on D_A and 5 on D_B. Table 4 traces the resulting trust state.

Table -4: Illustrative trust trace for the ITSM agent in context c2 (computed from Eqs. (3)–(8))

Step	Evidence (r, s)	E_d	T	ū	T_{low}	Enforced tier
t_0 steady state	I(50,0) P(50,0) B(180,4) A(95,1) C(40,0)	I .981, P .981, B .973, A .980, C .976	.977	.027	.950	T3 (bounded autonomous)
t_1 injection attempt blocked	B(180,9) A(95,11); others unchanged	B .948, A .889	.946	.026	.920	Score gives T3; circuit breaker caps at **T2** for the cooling-off period
t_1 + 28 days, no new negatives, positives replenished	B(180,4.5) A(95,5.5) after decay	B .971, A .937	.966	.027	.939	Score gives T3; return to T3 only after root-cause review and window W_3
Any time, context c_3	any	—	—	—	—	**T2 maximum** (regulatory ceiling): human e-signature required

Four observations follow. First, the score alone moves too slowly: a large positive history absorbs a severe event, and T_{low} falls only from 0.950 to 0.920, still inside T3; a score-only framework would have left the hijacked agent autonomous, which is why EAT adds circuit breakers. Second, enforcement at credential issuance limits damage before the tier changes: the agent's T3 token was bound to the ITSM audience with reassignment scope limited to approved groups, so the exfiltration path was never granted. Third, regulatory ceilings are independent of trust: no trust level lets the agent sign a Part 11 record, and expressing this as a ceiling rather than a very high threshold keeps the rule auditable. Fourth, decay supports recovery without amnesia: because $h^- > h^+$, half of the incident's weight remains after two positive half-lives.

4.2 Simulation Results

Table 5 reports the main results. Under static access (B1), hijacked and degrading agents keep T3 for the whole run, producing 4,336 unsafe autonomous actions per run. Full EAT (B4) reduces unsafe actions by hijacked agents from 1,548 to 5.5 (99.6%) and contains every hijack in a median of 4 h, against 29.0 d without circuit breakers (B3) and never under B1 or B2. Total unsafe exposure falls by 35% relative to B1. The arithmetic-mean baseline (B2) is worse than static access (1.8× its exposure): without an uncertainty penalty and with a compensatory mean it promotes average agents (reliability 0.97) to T4, which is intended for agents at 0.99 or better, adding about 3,395 unsafe actions per run from otherwise healthy agents.

Table -5: Simulation results (mean ± 95% CI over 30 seeds; 50 agents, 180 days)

Configuration	M1 unsafe exposure	M1 from hijacked agents	M2 hijack containment (median)	Degraded agents' time at ≥ T3	M3 false demotion	M4 reviews / day	M5 autonomy yield
B1 Static RBAC (fixed T3)	4,336 ± 307	1,548 ± 147	never	100 ± 0%	0 ± 0%	229 ± 0	90.0 ± 0.0%
B2 Mean score, no decay	8,008 ± 341	1,572 ± 142	never	100 ± 0%	0 ± 0%	85 ± 1	96.9 ± 0.0%
B3 EAT without breakers	3,840 ± 263	507 ± 22	29.0 d (98%)	96 ± 2%	0 ± 0%	158 ± 4	93.1 ± 0.2%
B4 Full EAT	2,827 ± 212	5 ± 1	4 h	79 ± 2%	39 ± 3%	189 ± 4	91.5 ± 0.2%
B5 EAT + behaviour floor†	1,294 ± 70	4 ± 1	4 h	26 ± 1%	42 ± 3%	219 ± 6	90.1 ± 0.3%
S1 = B4, 1% misattribution	3,268 ± 253	5 ± 1	4 h	93 ± 2%	10 ± 1%	168 ± 4	92.7 ± 0.2%
S2 = B5, 1% misattribution	1,393 ± 81	4 ± 1	3 h	29 ± 1%	11 ± 2%	204 ± 5	90.9 ± 0.3%

† B5 adds a behaviour floor (Eq. (10), $\varphi_B = 0.85$) introduced after the B1–B4 results were observed; it is an exploratory refinement, not part of the pre-specified protocol. S1 and S2 vary only the rate at which ordinary policy denials are misattributed as injections (5% → 1%). M2 is reported as "never" when no hijack was contained before the end of the run.

The same table exposes a weakness that the pre-specified hypotheses did not anticipate. Silent behavioural degradation from 0.99 to 0.90 is barely visible to the composite score: in B3, degraded agents spend 96% of their post-regression time at T3 or above, and in B4 79%. In the seed-1 trace (B3), the degrading agent's E_B falls from 0.97 to 0.77 within 60 days of the regression, but with weight 0.30 the composite T_{low} only falls from 0.98 to 0.91, well above $\theta_3 = 0.85$. The partial containment in B4 comes largely from circuit breakers tripped by misattributed denials, which degrading agents produce more often; at a 1% misattribution rate (S1) degraded agents stay at T3 or above 93% of the time. The same misattributions drive false demotions: 39% of healthy agents are demoted at least once in B4, falling to 10% in S1. The non-compensatory property of (6) therefore holds only in degree: a weak dimension pulls the composite down, but not necessarily below a tier threshold.

Consistent with EAT's treatment of identity and provenance as gates, we therefore tested a per-dimension *floor* on behavioural trust, applied as one more ceiling in (9):

$$c_{\text{floor}} = T_2 \text{ if } E_B < \varphi_B, \text{ else } T_4 \qquad (10)$$

With $\varphi_B = 0.85$ (B5), degradation-driven exposure falls by a further 66% relative to B4 and degraded agents spend 26% of post-regression time at T3 or above, at the cost of autonomy yield falling from 91.5% to 90.1% and reviews rising from 189 to 219 per day. Because the floor was designed after observing B4, its benefit should be confirmed on independent scenarios. Fig. 3 shows the tier trajectories of one hijacked and one degrading agent from seed 1.

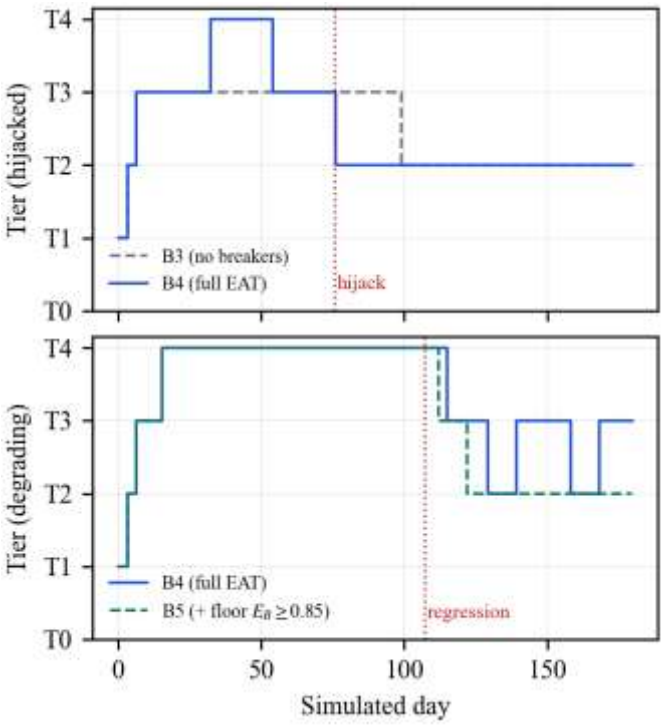

Fig -3: Tier trajectories from seed 1. Top: a hijacked agent with and without circuit breakers. Bottom: a degrading agent with and without the behaviour floor.

The sensitivity sweep (Fig. 4) supports H2. Averaged over the other parameters, raising κ from 0 to 2 changed unsafe exposure from 3,090 to 2,800, reviews from 190 to 223 per day and autonomy yield from 91.4% to 89.5%. The effect is small because evidence accumulates quickly at 40 actions per agent per day, so ū is small once agents are established; κ matters mainly for new or recently reconfigured agents. Shortening h^+ from 28 to 7 days changed exposure from 4,033 to 2,065 and false demotion from 39% to 37%, and a promotion margin δ = 0.02 changed exposure from 3,590 to 2,317 and yield from 91.9% to 89.0%.

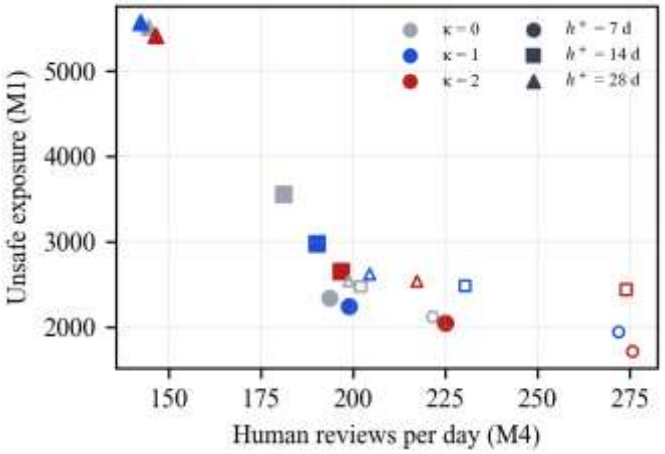

Fig -4: Sensitivity of full EAT: unsafe exposure against human review load for $h^+ \in \{7, 14, 28\}$ days and $\kappa \in \{0, 1, 2\}$; filled markers δ = 0, hollow δ = 0.02 (10 seeds each).

4.3 Threat-Model Coverage

Table 6 maps EAT against the ten OWASP agentic risks [3]. EAT is strongest where trust can be tied to cryptography or credentials (ASI03, ASI04, ASI07) and to consequences (ASI02, ASI10), and weakest where the threat lives in content semantics or human cognition (detection for ASI01, ASI06, ASI09). This supports positioning EAT as the *authority layer* that limits damage, used together with content-level defences rather than in place of them.

Table -6: EAT coverage of OWASP Top 10 for Agentic Applications (2026)

OWASP risk	Primary EAT mechanism	Dimensions	Coverage	Residual risk
ASI01 Agent Goal Hijack	Scope-bound tokens; PDP denials generate D_A evidence; circuit breaker	A, B	Partial–strong	EAT limits the consequences of a hijack; detecting injection needs content defences (e.g., CaMeL-style isolation)
ASI02 Tool Misuse and Exploitation	Per-action risk ρ, minimum-scope token exchange, budgets	A	Strong	Misuse inside legitimately granted scope
ASI03 Identity and Privilege Abuse	Identity gate, delegation-chain verification, no amplification, no passthrough	I, A	Strong	Compromise of the identity provider or attestation root
ASI04 Agentic Supply Chain	Provenance gate (OMS, SLSA, AI-BOM, signed Agent Cards), digest reset	P	Strong	Malicious but validly signed upstream artifact
ASI05 Unexpected Code Execution	Tier limits on execution tools; sandbox required at ≥ T3	A, P	Partial	Needs sandboxing outside EAT's scope
ASI06 Memory and Context Poisoning	Digest covers memory policy; drift evidence to D_B	B, P	Partial	Gradual poisoning below anomaly thresholds
ASI07 Insecure Inter-Agent Communication	Signed Agent Cards, mutual workload identity, minimum trust along the chain	I, A	Strong	Collusion among individually trusted agents
ASI08 Cascading Failures	Tier-limited blast radius, delegation ceilings, fast demotion	A, B	Partial–strong	Correlated failures across a shared model
ASI09 Human-Agent Trust Exploitation	Calibrated approval interface, canary proposals, approver accountability	C, B	Partial	Human factors cannot be fully engineered away
ASI10 Rogue Agents	Continuous monitoring, circuit breakers, revocation to T0, attested evidence only	All	Strong	Rogue behaviour inside granted scope and below detection thresholds

4.4 Comparison with Existing Approaches

Table 7 compares EAT with nine prior approaches based on each work's published description. EAT's distinct contribution is not any single mechanism, since each exists somewhere in prior work, but the *binding* of those mechanisms through one quantitative trust state into credential scope and autonomy tier. EAT is meant to sit on top of CaMeL- or Progent-style runtime isolation and MI9-style telemetry, not to replace them.

Table -7: Qualitative comparison (● addressed, ◐ partial, ○ not addressed)

Approach	Agent identity	Provenance gate	Dynamic behavioural trust	Graded autonomy tied to enforcement	Delegation attenuation	Human-reliance calibration	Regulatory mapping
NIST SP 800-207 ZTA [13]	●	○	◐	○	○	○	○
Huang et al. zero-trust agent IAM [14]	●	◐	◐	○	●	○	○
South et al. authenticated delegation [34]	●	○	○	○	●	○	◐
MI9 runtime governance [39]	◐	○	●	◐	○	○	◐
CaMeL [36]	○	○	○	○	◐	○	○
Progent [37]	○	○	○	◐	◐	○	○
Feng et al. autonomy levels [43]	○	○	○	◐	○	◐	○
TRiSM review [40]	◐	◐	◐	○	○	◐	◐
OWASP / CoSAI guidance [3], [17]	●	●	◐	◐	●	◐	◐
EAT (this work)	●	●	●	●	●	◐	●

5. DISCUSSION

5.1 Interpretation of Findings

The results support H1 for hijacked agents: full EAT lowers unsafe exposure and containment time relative to static access and to the mean-score baseline. For silent degradation, H1 is supported only weakly with the default parameters, and the post-hoc floor was needed for a substantial effect. H3 is supported: circuit breakers shorten hijack containment from 29.0 d to 4 h, which matches the score inertia seen in the case study. The sweep supports H2, with a small effect size. Two broader lessons follow. First, a composite score is a poor detector of a single failing dimension; dimensions that matter on their own need floors or gates, not only weights. Second, circuit breakers are only as good as the attribution that triggers them: misattribution trades containment of degrading agents against false demotions of healthy ones, so attribution quality should itself be measured and reported.

5.2 Regulatory Alignment

- **EU AI Act.** Regulation (EU) 2024/1689 requires risk management, logging, human oversight, accuracy, robustness and cybersecurity for high-risk systems [46]. The Digital Omnibus on AI postponed the high-risk obligations to 2 December 2027 for stand-alone Annex III systems and to 2 August 2028 for AI embedded in Annex I products [47]. EAT's ledger supports record-keeping, its T2 tier and approval interface support human oversight, and decay with re-certification supports post-deployment monitoring. The deferral is not a reason to wait: the evidence history EAT needs takes months to build.
- **NIST.** EAT's dimensions correspond to AI RMF characteristics (Table 1) and its lifecycle to the RMF functions: Establish maps to MAP, Verify and Monitor to MEASURE, and Enforce and Adapt to MANAGE, all under GOVERN [6]. The architecture implements the SP 800-207 PDP/PEP model [13] and addresses the identification, authorization, auditing and non-repudiation concerns of the NCCoE concept paper [4].
- **GxP and 21 CFR Part 11.** Part 11 requires audit trails, authority checks and electronic signatures linked to their records [44], and EU GMP Annex 11 adds requirements for validated computerised systems [48]. EAT's regulatory ceiling guarantees that no agent autonomously performs a signature-bearing action. The ISPE GAMP Guide: Artificial Intelligence provides a lifecycle framework for validating AI-enabled systems [49]; EAT's digest reset corresponds to change control and its evaluation evidence to ongoing verification.
- **SOX.** For financially relevant systems, EAT enforces segregation of duties (an agent cannot both propose and approve) and records the human principal in the delegation chain, supporting the change-management and access controls that underpin Section 404 assessments [45].

5.3 Practical Implications

Enterprises can adopt EAT in phases. Phase 1 deploys identity and provenance gates with a fixed T1/T2 policy. Phase 2 adds the evidence ledger and computes trust in *shadow mode*, deriving tiers without enforcing them. Phase 3 enforces tiers at credential issuance and enables circuit breakers. Phase 4 allows T4 promotions with governance sign-off. Shadow mode lets thresholds, weights and dimension floors be calibrated against real evidence before any autonomy depends on them; the simulation shows that these choices change outcomes materially.

5.4 Limitations

1) **Parameter calibration.** Weights, half-lives, κ, thresholds and floors are set by policy, not learned; the simulation shows that defaults can leave degradation undetected.
2) **Synthetic evaluation.** The simulation uses an assumed evidence model (Table 3); it demonstrates relative behaviour between configurations, not absolute rates in any real deployment.
3) **Evidence quality.** Trust is only as good as its evidence. Evaluation suites may not represent the real task distribution, and detectors produce false positives, as the misattribution results show.
4) **Score inertia.** Accumulated evidence damps single severe events; circuit breakers compensate but reintroduce rule-based judgement.
5) **Semantic attacks within scope.** An agent steered into harmful but permitted actions is limited only by budgets and sampling; content-level defences remain necessary.
6) **Human factors.** Approval fatigue can empty the T2 tier of meaning, and the proposed countermeasures need empirical validation.

6. CONCLUSION AND FUTURE WORK

This paper argued that trust in enterprise AI agents cannot be enforced while it remains a static attribute of an identity. EAT redefines trust as a computed state: gated by cryptographic identity and provenance, graded by attested and decaying evidence about behaviour, intent alignment and compliance, made conservative by explicit uncertainty, and bound mechanically to autonomy tiers and delegated-credential

scope. The worked example and the simulation agree on why each element is needed. Scores alone react too slowly to severe events; circuit breakers contained hijacks in hours and cut hijack-driven unsafe actions by 99.6% relative to static access. Enforcement at credential issuance limits damage before the score reacts, and regulatory ceilings must stay independent of trust. The simulation also showed that a weighted composite hides a single degrading dimension, which motivates per-dimension floors alongside gates. The OWASP analysis shows strong coverage of identity, supply-chain, inter-agent and rogue-agent risks and partial coverage of semantic and human-factor risks, positioning EAT as the authority layer beneath content-level defences.

Future work will (i) validate the behaviour floor and attribution-quality effects on independent scenarios and, where permitted, on anonymised deployment evidence; (ii) learn context-specific weights and floors from incident data under monotonicity constraints so that the model stays interpretable; (iii) extend trust composition to multi-agent collectives with collusion-resistant aggregation; (iv) express EAT trust states as verifiable credentials [26] so that trust can cross organisational boundaries; and (v) investigate reasoning-time assurance, such as search-based planners that generate counterfactual evidence of plan safety before execution, building on prior work on reasoning-as-a-service [50].

ACKNOWLEDGEMENT

The authors used AI-based writing and research assistance tools for literature search, drafting, editing and simulation code. All technical content, the framework, equations, algorithms and results were reviewed, verified and finalised by the authors, who take full responsibility for the work. The views expressed are the authors' own and do not necessarily represent those of Accenture; no client-confidential or proprietary information is disclosed.

REFERENCES

1. Model Context Protocol, "Authorization," Specification revision 2025-11-25. [Online]. Available: https://modelcontextprotocol.io/specification/2025-11-25/basic/authorization
2. A2A Project, "Agent2Agent (A2A) Protocol Specification," Linux Foundation, 2026. [Online]. Available: https://a2a-protocol.org/latest/specification/
3. OWASP GenAI Security Project – Agentic Security Initiative, "OWASP Top 10 for Agentic Applications for 2026," OWASP Foundation, Dec. 2025. [Online]. Available: https://genai.owasp.org/resource/owasp-top-10-for-agentic-applications-for-2026/
4. H. Booth, W. Fisher, R. Galluzzo, and J. Roberts, "Accelerating the Adoption of Software and Artificial Intelligence Agent Identity and Authorization," NIST National Cybersecurity Center of Excellence, Concept Paper, Feb. 2026. [Online]. Available: https://csrc.nist.gov/pubs/other/2026/02/05/accelerating-the-adoption-of-software-and-ai-agent/ipd
5. K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz, "Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection," in Proc. ACM Workshop on Artificial Intelligence and Security (AISec), 2023, arXiv:2302.12173.
6. National Institute of Standards and Technology, "Artificial Intelligence Risk Management Framework (AI RMF 1.0)," NIST AI 100-1, Jan. 2023, doi: 10.6028/NIST.AI.100-1.
7. High-Level Expert Group on Artificial Intelligence, "Ethics Guidelines for Trustworthy AI," European Commission, Brussels, Apr. 2019.
8. National Institute of Standards and Technology, "Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile," NIST AI 600-1, Jul. 2024, doi: 10.6028/NIST.AI.600-1.
9. ISO/IEC TR 24028:2020, "Information technology — Artificial intelligence — Overview of trustworthiness in artificial intelligence," International Organization for Standardization, 2020.
10. ISO/IEC 42001:2023, "Information technology — Artificial intelligence — Management system," International Organization for Standardization, 2023.
11. A. Jøsang and R. Ismail, "The Beta Reputation System," in Proc. 15th Bled Electronic Commerce Conf., 2002.
12. A. Jøsang, R. Ismail, and C. Boyd, "A Survey of Trust and Reputation Systems for Online Service Provision," Decision Support Systems, vol. 43, no. 2, pp. 618–644, 2007, doi: 10.1016/j.dss.2005.05.019.
13. S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero Trust Architecture," NIST Special Publication 800-207, Aug. 2020, doi: 10.6028/NIST.SP.800-207.
14. K. Huang, V. S. Narajala, J. Yeoh, R. Raskar, Y. Harkati, J. Huang, I. Habler, and C. Hughes, "A Novel Zero-Trust Identity Framework for Agentic AI: Decentralized Authentication and Fine-Grained Access Control," arXiv:2505.19301, 2025.
15. T. South et al., "Identity Management for Agentic AI: The New Frontier of Authorization, Authentication, and Security for an AI Agent World," OpenID Foundation, arXiv:2510.25819, 2025.
16. Cloud Security Alliance, "Agentic AI Identity and Access Management: A New Approach," Aug. 2025. [Online]. Available: https://cloudsecurityalliance.org/artifacts/agentic-ai-identity-and-access-management-a-new-approach
17. Coalition for Secure AI (CoSAI), "Agentic Identity and Access Management," OASIS Open, 2026.
18. Coalition for Secure AI (CoSAI), "Principles for Secure-by-Design Agentic Systems," OASIS Open, 2025.
19. SPIFFE Project, "Secure Production Identity Framework for Everyone (SPIFFE) Specifications," Cloud Native Computing Foundation. [Online]. Available: https://spiffe.io
20. A. Chan, N. Kolt, P. Wills, U. Anwar, C. Schroeder de Witt, N. Rajkumar, L. Hammond, D. Krueger, L. Heim, and M. Anderljung, "IDs for AI Systems," arXiv:2406.12137, 2024.
21. M. Jones, J. Bradley, and N. Sakimura, "JSON Web Signature (JWS)," RFC 7515, IETF, May 2015.
22. A. Rundgren, B. Jordan, and S. Erdtman, "JSON Canonicalization Scheme (JCS)," RFC 8785, IETF, Jun. 2020.
23. OpenSSF AI/ML Working Group, "OpenSSF Model Signing (OMS) Specification," Open Source Security Foundation, 2025. [Online]. Available: https://github.com/ossf/model-signing-spec
24. OpenSSF, "Supply-chain Levels for Software Artifacts (SLSA) Specification v1.0," 2023. [Online]. Available: https://slsa.dev/spec/v1.0/
25. OWASP CycloneDX, "CycloneDX Specification (Machine Learning Bill of Materials)," v1.5 or later. [Online]. Available: https://cyclonedx.org
26. W3C, "Verifiable Credentials Data Model v2.0," W3C Recommendation, 2025.
27. M. Jones, A. Nadalin, B. Campbell, J. Bradley, and C. Mortimore, "OAuth 2.0 Token Exchange," RFC 8693, IETF, Jan. 2020.
28. A. Prasad, R. Krishnan, D. Lopez, and S. Addepalli, "Cryptographically Verifiable Actor Chains for OAuth 2.0 Token Exchange," IETF Internet-Draft draft-mw-spice-actor-chain-05, Apr. 2026 (work in progress).
29. T. S. Senarath and A. Dissanayaka, "OAuth 2.0 Extension: On-Behalf-Of User Authorization for AI Agents," IETF Internet-

Draft draft-oauth-ai-agents-on-behalf-of-user-02, 2025 (work in progress).
30. A. Raut, "Transaction Tokens For Agents," IETF Internet-Draft draft-oauth-transaction-tokens-for-agents-06, Apr. 2026 (work in progress).
31. B. Campbell, J. Bradley, and H. Tschofenig, "Resource Indicators for OAuth 2.0," RFC 8707, IETF, Feb. 2020.
32. M. B. Jones, P. Hunt, and A. Parecki, "OAuth 2.0 Protected Resource Metadata," RFC 9728, IETF, Apr. 2025.
33. N. Hardy, "The Confused Deputy (or why capabilities might have been invented)," ACM SIGOPS Operating Systems Review, vol. 22, no. 4, pp. 36–38, 1988.
34. T. South, S. Marro, T. Hardjono, R. Mahari, C. D. Whitney, D. Greenwood, A. Chan, and A. Pentland, "Authenticated Delegation and Authorized AI Agents," arXiv:2501.09674, 2025.
35. E. Debenedetti, J. Zhang, M. Balunović, L. Beurer-Kellner, M. Fischer, and F. Tramèr, "AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents," in Proc. NeurIPS Datasets and Benchmarks Track, 2024, arXiv:2406.13352.
36. E. Debenedetti, I. Shumailov, T. Fan, J. Hayes, N. Carlini, D. Fabian, C. Kern, C. Shi, A. Terzis, and F. Tramèr, "Defeating Prompt Injections by Design," arXiv:2503.18813, 2025.
37. T. Shi, J. He, Z. Wang, H. Li, L. Wu, W. Guo, and D. Song, "Progent: Securing AI Agents with Privilege Control," arXiv:2504.11703, 2025.
38. G. Syros, A. Suri, J. Ginesin, C. Nita-Rotaru, and A. Oprea, "SAGA: A Security Architecture for Governing AI Agentic Systems," arXiv:2504.21034, 2025.
39. C. L. Wang, T. Singhal, A. Kelkar, and J. Tuo, "MI9: An Integrated Runtime Governance Framework for Agentic AI," arXiv:2508.03858, 2025.
40. S. Raza et al., "TRiSM for Agentic AI: A Review of Trust, Risk, and Security Management in LLM-based Agentic Multi-Agent Systems," AI Open, 2026, arXiv:2506.04133.
41. C. Schroeder de Witt et al., "Open Challenges in Multi-Agent Security: Towards Secure Systems of Interacting AI Agents," arXiv:2505.02077, 2025.
42. J. D. Lee and K. A. See, "Trust in Automation: Designing for Appropriate Reliance," Human Factors, vol. 46, no. 1, pp. 50–80, 2004, doi: 10.1518/hfes.46.1.50_30392.
43. K. J. K. Feng, D. W. McDonald, and A. X. Zhang, "Levels of Autonomy for AI Agents," arXiv:2506.12469, 2025.
44. U.S. Food and Drug Administration, "21 CFR Part 11 — Electronic Records; Electronic Signatures," Code of Federal Regulations.
45. Sarbanes-Oxley Act of 2002, Pub. L. 107-204, Section 404, United States Congress, 2002.
46. Regulation (EU) 2024/1689 of the European Parliament and of the Council laying down harmonised rules on artificial intelligence (Artificial Intelligence Act), Official Journal of the European Union, 2024.
47. Regulation (EU) 2026/1744 amending Regulation (EU) 2024/1689 (Digital Omnibus on AI), Official Journal of the European Union, Jul. 2026.
48. European Commission, "EudraLex Volume 4, Annex 11: Computerised Systems," 2011.
49. ISPE, "GAMP Guide: Artificial Intelligence," International Society for Pharmaceutical Engineering, Jul. 2025.
50. K. Rajkumar, "GraphZero Agent: Monte Carlo Graph Search as Reasoning-as-a-Service for General Agentic Systems," International Journal of Scientific Research in Engineering and Management (IJSREM), [vol., issue, year, DOI].