

TECHNICAL BRIEF

Execution-Time Authority, Governance and Evidence

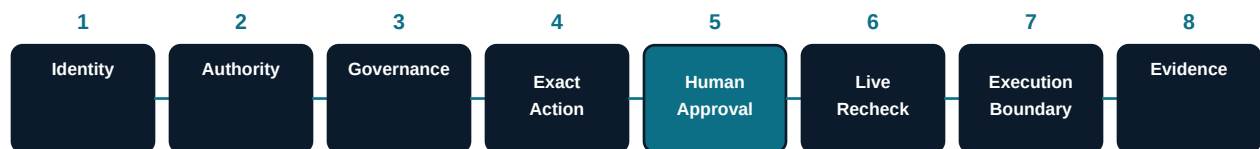
Technical security description of the current Trace system, with validation results and stated limits.

Every Decision Begins With Authority.

Version 5 · October 2026 · Trace Continuity Labs · tracecontinuity.com
Patent Pending — U.S. Provisional Patent Application No. 64/137,172. Not a certification.

1 · Positioning

Trace Continuity is an execution-time authority, governance and evidence layer for AI agents and enterprise systems. It determines whether an actor is authorized to perform a specific action at the moment of execution, enforces that decision at the execution boundary, and produces verifiable evidence of what was authorized and what occurred.



Human approval applies when the action requires it. Authority is re-checked live at the boundary.
Evidence is recorded at every step and signed; tenant-wide checkpoints can be retained outside Trace.

2 · Authority

- Every governed request is evaluated at execution time through one decision point. Credentials are pinned to a single authority.
- The full authority lineage is walked at runtime and fails closed on a missing, suspended, revoked, expired, cross-tenant, cyclic or over-deep link.
- Delegation containment (tenant, scope, execution rights, lifetime) is enforced in three independent layers: the configuration gate, a database rule and the runtime chain walk.
- Suspension is reversible only through governed controls; permanent revocation is enforced by the database and cannot be reversed.
- Person offboarding: an authenticated administrative endpoint suspends a person within a tenant. Person status is checked inside the same decision point, so dependent authority, delegated authority, keys and agents are refused immediately; pending approvals and unconsumed external-action permissions are refused.
- New API keys must carry an expiration within a configurable maximum; expiry is enforced at execution. Extending a key's lifetime is treated as weakening protection and needs a second administrator. Legacy non-expiring keys continue to work and are flagged for review.
- Registered agents carry owner, purpose and bound keys and can be suspended.

3 · Governance

- Allow / Deny / Transform on every governed decision. Material conditions recorded at execution time are a separate decision with their own hash-chained history.
- Matter boundaries, ethical walls and legal holds are enforced on the same path. Reported derivation lineage lets upstream status govern derived artifacts; where lineage is required, an incomplete walk is a refusal.

- Governance-over-governance: configuration changes that reduce protection require a second administrator's approval of the exact change, enforced inside the single configuration path.
- A tenant-scoped, read-only stale-access view lists non-expiring, expired and unused keys, keys without a registered agent, agents and authorities whose holder is suspended, and orphaned delegation. It changes nothing by itself.

4 • Human approval and exact-action authorization

- An external action is reduced to a canonical form (including normalized decimals) and fingerprinted. Authorization, approval and redemption all bind that fingerprint.
- Approvers see the exact governed action. The requester cannot approve, and neither can an authority that depends on the requester.
- Permissions are single-use and time-limited. Redemption by the registered enforcement point re-runs authority and governance live and consumes the permission in one atomic database operation.
- The enforcement point reports the outcome. Authorized, executed, failed and could-not-establish remain distinct states.

Execution boundary

External-action enforcement is non-bypassable when the protected credential or action is placed behind the Trace enforcement point. If the customer leaves the real credential reachable elsewhere, Trace cannot stop its direct use.

5 • Evidence

- Per-action evidence is hash-chained and append-only against application code and ordinary database roles.
- Evidence packages are signed with Ed25519. Private keys never leave server custody; the offline verifier uses only the package and a trusted copy of Trace's published public keys. Verdicts are VERIFIED, FAILED or INCOMPLETE; anything that cannot be established is never VERIFIED. Entries that would need redaction make a package INCOMPLETE rather than VERIFIED.
- Tenant-wide signed checkpoints commit to the tenant's entire evidence history through chained segment digests and a cumulative root. Evidence ordering and time are assigned by the database, not by the writer, and coverage advances only past transactions that have finished, so slow transactions cannot land in an already-sealed period.
- Customers export checkpoint anchors and retain them outside Trace. Deleting an action, rewriting or backdating history, or deleting or reordering checkpoints is detectable against a retained anchor.
- Before signing any evidence package or extending a checkpoint chain, Trace re-verifies committed history against its checkpoints, confirms checkpoint signatures against its own key custody, and refuses — recording the refusal and reporting FAILED VERIFICATION — if anything no longer matches. A tenant whose history was sealed cannot silently restart at checkpoint 1.

6 • Signing keys

- Keys move active → retired → compromised. Public keys are never deleted or altered, so old evidence stays verifiable.
- A compromised key can never sign again or return to active. Declaring compromise requires a replacement key, which signs the compromise statement and a per-tenant anchor pinning the last trusted checkpoint. The trusted boundary can only be narrowed.
- Verification never trusts a timestamp a package or checkpoint claims for itself.

7 · Non-bypassability

There is one server-side path to governed data and governed actions. A build-blocking scanner parses the source tree and fails the build if code reaches governed storage, configuration or external actions around that path, and is itself tested against adversarial fixtures.

8 · Validation

October 10, 2026: the database-owner validation run passed 567 tests across 35 files with zero failed and zero skipped, on an isolated PostgreSQL cluster built from all 55 migrations. The ordinary run passed 427 tests with 140 special-environment skips. The build-blocking bypass scanner reported no bypass, and the production build succeeded. These results validate application and database rules; they are not a certification, a formal penetration test or customer acceptance testing.

9 · Stated limits

- Signed checkpoints do not prevent a database owner from changing data. They make later alteration or deletion of committed history detectable against checkpoints retained outside Trace. Activity after the most recent retained checkpoint is covered only once a later checkpoint is retained.
- If a database owner erased every Trace-held record of earlier checkpoints, only checkpoints retained outside Trace would reveal it. Retaining anchors is part of the control.
- External-action enforcement is non-bypassable when the protected credential or action is placed behind the Trace enforcement point. A credential left outside it can be used without Trace; Trace can only show that no Trace authorization exists for that use.
- Signing keys are held in platform secret storage today, not an HSM or KMS. There is no independent timestamping or transparency log.
- Reviewers must obtain Trace's public keys through a channel they trust; an old copy of the key list will not show a later key compromise.
- Trace evaluates the state reported to it. It does not discover changes in outside systems that nobody reported.
- No SOC 2, HIPAA or other certification, and no third-party penetration-test certification, is claimed.

10 · Deployment responsibilities

Area	Belongs to
HSM / KMS-backed signing keys	Deployment architecture
Dedicated tenancy; staging/production separation	Deployment architecture
SSO and SCIM / identity-provider offboarding	Customer IdP, calling Trace's person-suspension endpoint
Rate limiting, WAF, network edge	Deployment / network
Monitoring and SIEM forwarding	Customer security operations
Backups and point-in-time recovery procedures	Deployment operations
Write-once retention of checkpoint anchors	Customer storage or escrow
Placing external credentials behind the enforcement point	Customer integration