Nick Clark · Published April 25, 2026 · PDF

How to Keep a Login Session Tied to a Live Human, Not a Stolen Token

If your session is a bearer token, whoever holds the token is the user, even if the real person walked away ten minutes ago. This guide describes an architectural approach that binds a session to the continuity of a live person's biological and behavioral signals instead of to a stored secret, so the session suspends itself when that continuity breaks. It is an architecture disclosed in United States Patent Application 19/647,395 (not a shipping library), and it is grounded in the Biological Identity inventive step.

What You Are Building

You are building a session-identity layer that keeps asking one question for the entire life of a session: is the live person here right now still the same person who started this? Not "did someone present a valid credential once at login," but "is the human behind this session continuously the authorized human."

The people who need this are the ones for whom a hijacked session is expensive: privileged consoles, operator stations for physical machinery, high-value account access, anything where the gap between "authenticated at login" and "still the right person now" is a real attack window. If your entire notion of "who is logged in" collapses to possession of a cookie, an access token, or a refresh token, then a stolen token, a replayed sample, or a walk-up on an unlocked terminal is indistinguishable

from the legitimate user. The goal here is an architecture where that indistinguishability does not hold: the session is bound to the ongoing presence of a live human, and it degrades itself the moment that binding cannot be reconfirmed.

Why the Obvious Approaches Fall Short

The standard approaches are real and useful, and this guide does not caricature them. It identifies where each one structurally stops.

Bearer tokens (session cookies, OAuth access tokens, JWTs) authenticate possession. That is by design and it is efficient. But possession is transferable: once the token leaves the intended holder, the server has no built-in way to notice. Short expiry and refresh rotation shrink the window; they do not change the model.

Continuous authentication and behavioral biometrics get closer. They keep scoring the user after login against an enrolled profile or template of that user, and raise or lower a trust score as behavior stays consistent. The structural point, described in the filed disclosure, is that these systems locate identity in the enrolled profile: each observation is evaluated as a match against a stored reference. A stored reference is a fixed artifact. It can be stolen, and a fabricated sample that matches it will keep matching it, because the matcher has no notion of whether this observation is a plausible *next* observation in a live sequence, only whether it resembles the stored one.

Classic biometrics (fingerprint, iris, face) share the same shape at a coarser grain: enroll a template, later compare a fresh sample against it, return match or non-match. The disclosure notes three structural problems with locating identity in a static template: biological signals are not time invariant and drift with age, injury, illness, and fatigue; a stolen template can be replayed indefinitely because it is a fixed artifact; and a binary match discards the information carried in how a person's signals evolve over time.

The common gap: identity is treated as a thing you *hold* or *are at one instant*, checked at moments. A session tied to a moment inherits every weakness of that moment for the rest of its life.

The Architecture

The disclosed approach inverts the model. Identity is not a credential, a template, or a snapshot. It is *behavioral continuity over time*: the property of a signal stream that shows coherent, policy-verifiable continuity across a sequence of observations, where each new observation is validated as a plausible continuation of the prior sequence rather than matched against a stored reference. There is no enrolled profile. Identity lives in the continuity of the chain itself.

Concretely, the filed spec describes a sequential pipeline (FIG. 9A) with these stages:

- **Signal acquisition.** Raw biological and behavioral signals are captured. The disclosure defines three tiers: contact-based (fingerprint, palm, iris, requiring deliberate interaction, highest quality), semi-contact (wearables capturing pulse, electrodermal activity, gait, respiration, near-continuous coverage), and non-contact (voice, gait, keystroke and pointer dynamics, remote physiological observation, lowest friction, lowest per-sample quality). The tiers are fusible, and the tier informs how much the resulting signal is trusted downstream.
- **Feature extraction and noise-tolerant normalization.** Signals become *continuity-suitable* features: representations that preserve temporal dynamics (rate of change, short-term variability, cross-signal coupling, periodicity) rather than just instantaneous values, because what is being validated is a trajectory, not a snapshot. An adaptive normalization scheme maintains a running per-individual model of each feature's expected range and noise, so gradual physiological change is absorbed without re-enrollment.

- **Stable sketching.** The normalized stream is reduced to a noise-tolerant, non-invertible *stable sketch* through dimensional reduction, projection, and quantization into bands. The spec is explicit that non-invertibility here is structural, not a bet on computational hardness: dimensional reduction discards information, the projection is many-to-one, and quantization discards within-band precision. Helper data lets a later capture reproduce the same band assignment within noise tolerance without revealing the underlying values.
- **Biological hash generation with domain separation.** From the stable sketch, a cryptographic hash is computed over a composite input: the band assignments, a **temporal binding value** encoding the capture time, a **domain separation tag** scoping the hash to one context, and a per-chain **salt** rotated on a policy schedule. Two consequences the spec calls out directly: the temporal binding makes hashes **non-replayable** (a hash minted at time T is not valid at T+delta because the temporal value differs), and the domain tag makes hashes **unlinkable across contexts** (the same person's hash in domain A cannot be correlated with their hash in domain B).
- **Trust-slope construction and continuity validation.** The hashes form a *trust-slope*: an ordered chain, each entry linked to its predecessor by continuity validation. This is the heart of it. A new capture is not matched against a template; its stable sketch is compared against the *recent* entries in the chain to produce a **graded continuity score** (not a binary match) reflecting how many band assignments are consistent with the recent trajectory, whether band transitions look like expected noise versus a genuine change, and whether any change is temporally plausible given the elapsed time and expected drift. The spec defines four outcomes: strong continuity (append with full confidence), acceptable continuity (append with reduced confidence), degraded continuity (append with a flag that triggers enhanced monitoring), and **continuity failure** (do not append; trigger recovery). Because each check is against the recent trajectory, gradual drift is accommodated automatically; abrupt substitution is what fails.

Three architectural properties make this a session story rather than a login story:

Anti-spoofing is inside continuity, not a bolt-on filter. A spoofed sample must not only pass instantaneous quality checks but also read as a plausible continuation of the target's trajectory, including temporal dynamics and cross-signal coupling that are not observable from any single captured sample. The disclosure names four integrated mechanisms: continuity-consistent challenge-response (the response dynamics must match what the trust-slope predicts, not merely prove a live human), sensor attestation, temporal consistency enforcement (blocks replay), and proximity constraints (blocks remote presentation).

Authorization is continuously re-evaluated, and capabilities auto-suspend. A resolved identity carries the trust-slope's cumulative confidence and the assurance level of its most recent event. Capability tokens are *bound to the trust-slope*: per the spec, if the trust-slope's confidence degrades through failed events, excessive sparsity, or detected anomalies, the bound capabilities are automatically suspended or revoked. Authorization is not granted once and assumed indefinitely.

Continuity break drives a proportional safety response, not a hard drop. In operational-handoff terms (Section 9.25), verification runs continuously through the session; if continuity breaks, indicating the operator changed, left, or became incapacitated, the system enters a governed degradation mode restricted to safety-minimal operations, and records the break in lineage. Resuming full capability requires re-establishing continuity with the authorized person or an explicit delegation to a newly verified one.

How to Approach the Build

You are implementing this yourself. The steps below are the order the architecture implies; the interface sketch is illustrative and faithful to the spec, not shipping code.

1. **Pick your acquisition tier(s) to your friction budget.** Most software sessions cannot demand a fingerprint mid-task, so lean on non-contact behavioral signals (keystroke and pointer dynamics, interaction rhythms) for continuous coverage, and reserve contact-based captures as high-assurance anchor points at login or on step-up. Let each captured signal carry its tier as a confidence weight downstream.
2. **Build continuity-suitable features, not snapshots.** Extract temporal dynamics, not just instantaneous values, and maintain a per-user running normalization model so slow drift does not read as a discontinuity.
3. **Make the stored artifact non-invertible by construction.** Store stable sketches (banded, reduced, projected) plus helper data and the hash chain, never raw signals. The spec's non-invertibility is structural, so lean on dimensional reduction, many-to-one projection, and quantization, not on secrecy of a template.
4. **Bind every hash in time and scope.** Fold a temporal binding value, a per-context domain separation tag, and a rotating per-chain salt into the hash. This is what gives you non-replayability and cross-context unlinkability for free.
5. **Validate by continuity, score in grades, act by policy.** For each new capture, compute a graded continuity score against the recent chain and branch into the four outcomes. Bind capabilities to the chain so that degraded or failed continuity suspends them automatically.

```
# Illustrative only. Faithful to the disclosed pipeline, not a library.
sketch      = stable_sketch(normalize(extract_features(capture)))
bio_hash    = hash(sketch.bands, temporal_binding(now), domain_tag, chain.sa
score       = continuity_score(sketch, chain.recent_window)  # graded, not m

if   score >= HIGH:      chain.append(bio_hash, confidence="strong")
elif score >= MIN:       chain.append(bio_hash, confidence="reduced")
elif known_degradation:  chain.append(bio_hash, flag="degraded"); raise_moni
else:                    suspend_bound_capabilities(chain); begin_recovery()
```

6. **Design escalation and delayed/sparse modes as first-class.** When ambient continuity confidence drops, escalate to a richer or higher-assurance capture before failing outright, and de-escalate when confidence recovers. Where connectivity is intermittent, the spec supports generating a hash locally with a proof-of-capture attestation and validating later within a bounded proof window; sparse intervals widen the continuity thresholds and lower the resulting confidence.
7. **Enforce consent-gated resolution structurally.** The mode the system may use (one-to-one verification, one-to-many identification, or privacy-preserving anomaly detection only) should be constrained by how the person engaged, enforced as a structural limit on which queries are even reachable, not as a policy flag that can be misconfigured or overridden.

What This Does Not Give You

This is an architecture, not a drop-in library. There is no package to install and no SDK behind this guide; you build the acquisition, extraction, sketching, hashing, and validation stages yourself, against sensors and threat model you choose. The approach is disclosed in a patent filing; it is not presented here as a benchmarked or production-proven system, and the filed spec states no accuracy, latency, or spoof-resistance numbers, so neither does this guide. Do not infer any.

It also will not fix problems outside its scope. It presumes you can acquire real, continuity-bearing signals; a context with no usable ongoing signal from the human has nothing to build continuity on. Abrupt legitimate change (injury, surgery, acute illness) can itself produce continuity failure and route into recovery, which you must design. Biological state inference described in the disclosure is explicitly non-diagnostic, deviation-from-own-baseline only, and must not be repurposed as medical or fitness assessment. And continuity binding sits alongside, not on top of, transport security, device attestation, and authorization policy; it is one substrate among several, meant to be composed, not a replacement for the rest of your stack.

Disclosure Scope

The continuity-based identity approach described in this guide, including trust-slope continuity validation, non-replayable temporally bound biological hashing, domain-separated unlinkability, and auto-suspension of trust-slope-bound capabilities on continuity failure, is disclosed in United States Patent Application 19/647,395. This guide is educational: it explains an architecture a developer can study and implement independently. It is not a warranty, a specification of any product, or an offer of software, and nothing here should be read as a performance guarantee or as a claim that a shipping implementation is provided.

Biological Identity (/biological-identity)

All 40 steps → (/inventive-steps)

Identity from behavioral continuity. No stored templates. No keys.

Chapter 9 (/patents/19-647395/chapters/biological-identity)

PRIMARY TECHNICAL DISCLOSURE

- Continuity-Based Biological Identity Using Trust-Slope Validation (/articles/continuity-based-biological-identity-using-trust-slope-validation)

SECONDARY TECHNICAL

- Biological Trust Slope Construction: Identity Through Behavioral Continuity (/articles/biological-identity/trust-slope-construction)
- Resolution Modes for Biological Identity: Verification, Identification, Hybrid Narrowing (/articles/biological-identity/resolution-modes)
- Biological Hash Generation With Domain Separation (/articles/biological-identity/biological-hashing)
- Biological State Inference From Continuity Baseline (/articles/biological-identity/state-inference)
- Cross-Modal Biological Hash Fusion (/articles/biological-identity/cross-modal-fusion)
- Biological Continuity as Handoff Verification (/articles/biological-identity/handoff-verification)

- Relational Trust Trajectories: Trust as Temporal Relationship (/articles/biological-identity/relational-trust)
- Identity as Behavioral Continuity: Beyond Single-Point Capture (/articles/biological-identity/behavioral-continuity)
- Biological-Device-Agent Identity Layering (/articles/biological-identity/identity-layering)
- Biological Signal Acquisition Tiers (/articles/biological-identity/acquisition-tiers)
- Noise-Tolerant Feature Normalization for Biological Signals (/articles/biological-identity/feature-normalization)
- Stable Sketching and Helper Data for Biological Features (/articles/biological-identity/stable-sketching)
- Predictive Identity Trajectory: Forecasting Biological Identity Evolution (/articles/biological-identity/predictive-trajectory)
- Population-Scale Collision Resistance for Biological Hashes (/articles/biological-identity/collision-resistance)
- Adaptive Indexing of Biological Trust Slopes (/articles/biological-identity/adaptive-index-integration)
- Delayed and Sparse Validation for Disconnected Environments (/articles/biological-identity/delayed-validation)
- Policy-Governed Capability Binding for Biological Identity (/articles/biological-identity/capability-binding)
- Multi-Identity Delegation Without Biological Data Disclosure (/articles/biological-identity/multi-identity-delegation)
- External Credential Integration With Trust-Slope Integrity (/articles/biological-identity/credential-integration)
- Anti-Spoofing Through Continuity Validation (/articles/biological-identity/anti-spoofing)
- Identity Lifecycle Management and Phase-Based Reseeding (/articles/biological-identity/lifecycle-management)
- Quorum-Based Biological Identity Recovery (/articles/biological-identity/quorum-recovery)
- Privacy Governance and Revocation for Biological Identity (/articles/biological-identity/privacy-governance)
- Human-Agent Primitive Integration for Biological Identity (/articles/biological-identity/cognitive-integration)

APPLICATIONS · GENERAL

- Airport Security Without Biometric Databases (/articles/biological-identity/airport-security)

- Estate Verification That Survives the Decedent: Probate Identity Through Behavioral Continuity (/articles/biological-identity/estate-verification)
- Identity Continuity for Dementia Residents in Elder Care (/articles/biological-identity/elder-care-continuity)
- Child Development Tracking Without Re-Enrollment: Continuity-Based Pediatric Identity (/articles/biological-identity/child-development-tracking)
- Continuous Addiction Recovery Monitoring With Privacy-Governed Relapse Detection (/articles/biological-identity/addiction-recovery-monitoring)
- Continuous Operator Verification for Workplace Safety in Hazardous Industries (/articles/biological-identity/workplace-safety-monitoring)
- Athlete Identity and Readiness Monitoring Without Storing Biometric Templates (/articles/biological-identity/athletic-performance)
- Continuity-Based Identity Verification for Immigration and Asylum Processing (/articles/biological-identity/immigration-processing)
- Operator-to-Asset Binding for Fleets and Robotaxis: Who Is Driving Right Now (/articles/biological-identity/fleet-operator-binding)
- Continuous Clinician-Patient Binding for Audit-Grade Medical Decision Attribution (/articles/biological-identity/medical-clinician-binding)

APPLICATIONS · SPECIFIC

- TSA PreCheck vs Continuity-Based Biological Identity (/articles/biological-identity/tsa-precheck)
- Global Entry Alternative: Biological Continuity Beyond Credential Matching (/articles/biological-identity/global-entry)
- Apple Face ID vs Continuity-Based Biological Identity: Template Match or Trust Slope (/articles/biological-identity/apple-face-id)
- Samsung Knox vs Biological Identity: Container Security Meets Trust-Slope Continuity (/articles/biological-identity/samsung-knox)
- ID.me Alternative: Verifying Documents vs. Biological Continuity (/articles/biological-identity/id-me)
- Socure Alternative: Trajectory Validation Beyond Point-in-Time Risk Scoring (/articles/biological-identity/socure)
- Plaid Identity Alternative: Biological Continuity Beyond Account Verification (/articles/biological-identity/plaid-identity)
- Onfido Alternative for Continuity: Verify Documents, Then Validate Identity Drift (/articles/biological-identity/onfido)
- Veriff Alternative: Continuity-Based Identity Verification Beyond Per-Session Capture (/articles/biological-identity/veriff)

- Trulioo Alternative: Governed Biological Continuity Beyond Record Matching (/articles/biological-identity/trulioo)
- Seeing Machines DMS vs Continuity-Based Biological Identity: Detection or Identity Binding (/articles/biological-identity/seeing-machines-dms)
- Smart Eye Driver Monitoring vs Continuity-Based Biological Identity (/articles/biological-identity/smart-eye)

Biological Identity overview → (/biological-identity)