Nick Clark · Published April 25, 2026 · PDF

Keyless Workload Identity for Serverless Functions: Authenticating Ephemeral FaaS Without a Persistent Keypair

Serverless functions spin up, do their work, and vanish in milliseconds, leaving platform teams to authenticate workloads that never live long enough to hold a stable keypair or a registered identifier. This application addresses that gap. It is built on the Keyless Identity, disclosed in United States Patent Application 19/388,580, which derives identity from locally retained, time-sensitive state and validates it by dynamic slope evaluation rather than from a long-lived secret.

What This Application Specifies

This application specifies how to give a short-lived serverless workload a verifiable identity without ever issuing it a persistent keypair, a registered client secret, or a stable identifier. The target is the function-as-a-service execution model: a function instance is created on demand, runs for a few milliseconds to a few seconds, and is then discarded. There is frequently no durable storage between invocations, no stable network identity, and no opportunity to enroll the instance with a certificate authority before it must act.

The underlying mechanism comes from the Keyless Identity, disclosed in United States Patent Application 19/388,580. In that disclosure, a device or agent expresses identity as a trust slope, meaning the cumulatively validated sequence of dynamic hashes formed by successive, verifiable identity mutations, rather than as a static credential. Each step is computed from the immediately prior step combined with a source of non-exported unpredictability and a volatile, non-repeating salt. A receiver that holds any previously trusted step can evaluate whether a presented successor is a valid descendant under policy-bounded continuity checks, and it can do so locally, without contacting a certificate authority, a registry, or any other central trust anchor.

Applied to a function instance, the identity is computed directly from inputs the instance already has at hand. The disclosure describes deriving the dynamic hash either from a static hardware anchor combined with a volatile salt, or from a local state vector of device-observable signals, such as monotonic counters, high-resolution timing deltas, scheduler jitter, and I/O inter-arrival micro-jitter, that is passed through a strong extractor to produce a bounded pseudorandom token. Either source, or a hybrid of both, yields a successor identity bound to time, context, and prior state. The function never needs to carry a secret across its short life.

Why It Matters

The dominant patterns for authenticating serverless workloads were not designed for workloads this short-lived. Issuing each instance a certificate or a keypair forces an enrollment round trip into a cold start, and it leaves key material sitting in a process whose entire lifetime may be shorter than a normal certificate provisioning step. Sharing one long-lived secret across every instance of a function collapses the blast radius of a single leak onto the whole fleet. Brokering tokens through a central identity service on every invocation puts that service on the critical path of every call and reintroduces exactly the centralized trust anchor and persistent registry that ephemeral compute was supposed to avoid.

The Keyless Identity disclosure was written for precisely this class of environment. It names stateless execution fabrics, ephemeral edge workers, serverless functions, and relay nodes operating without durable storage as target deployments, and it states that because no private key material or session state must be preserved across invocations, stateless operation remains fully interoperable with memory-aware peers. The security argument that matters here is structural: a dynamic hash is ephemeral, computed per step, and never reused as a standing credential, so observation or disclosure of any single value does not enable impersonation, because acceptance requires monotonic progression from a prior trusted state under the published update rule. There is no standing key to steal from a function instance because the model has no place for one.

How It Composes With the Domain

A practical deployment maps the disclosed components onto the serverless control and data planes without inventing new platform primitives.

Anchoring an instance. When a function instance is admitted to an execution context, the disclosure provides for binding identity to the host it runs on. The host maintains its own dynamic device hash, and an agent-side mutation is computed as a successor of the prior agent hash combined with a host mutation token derived from the host device hash and a mutation class. The host emits a signed entanglement trace recording the prior hash, the host device hash, the mutation token, the resulting successor, and the mutation class. For a serverless instance this is a direct fit: the function step is cryptographically tied to the specific worker, microVM, or sandbox that executed it, and a verifier rejects any successor whose entanglement trace does not open to a host device hash that is valid under policy. Off-substrate evolution, where a claimed successor cannot be tied to a real executing host, fails closed.

Function-to-function calls. When one function calls another, the disclosed two-stage message construction applies directly. The caller derives a symmetric key from the callee's current dynamic identity using a key-derivation function over that identity and

a domain-separating context, encrypts the payload, and embeds a copy of its own current sender hash inside the ciphertext. It places its current hash in the transport header. The callee first screens the header hash for on-slope continuity before doing any expensive work, then derives its own key, decrypts, and validates the embedded sender hash against the sender's slope. Malformed or off-slope traffic is discarded before decryption, and substitution after decryption is caught by the embedded hash. The message itself never carries the symmetric key.

Surviving the absence of state. Serverless instances rarely retain memory between invocations, which is exactly the sparse-state condition the disclosure handles. It provides an append-only lineage of identity steps committed into a compact chain with periodic anchors, so that a verifier holding only a recent anchor can request a bounded proof window and deterministically replay the intervening steps. Where a sender and recipient drift out of sync, the disclosed fallback is a short challenge-response rekey scoped to the current epoch, or a checkpoint request that yields a bounded proof window sufficient to advance to the current identity. A platform can keep these sparse anchors in its existing control plane while the functions themselves stay stateless.

Cold-start triage. The disclosed predictive verification forecasts a near-future successor and an acceptance envelope from observed cadence and role-transition history. A gateway can use this to triage incoming calls under load: claims far outside the envelope are rejected or quarantined immediately, while near-boundary claims are held until a checkpoint or short proof arrives. This composes with per-sender rate limits keyed to header continuity, giving an admission-control surface that discards bad traffic early without a central lookup.

What This Enables

A platform built this way can authenticate a function instance the moment it begins executing, with no enrollment round trip and no secret to provision, because identity is derived from inputs the instance already holds. Each invocation that mutates identity

records a host-entangled, signed step, so a fleet operator can reconstruct after the fact which worker ran which step, producing verifiable execution provenance across a migrating or fan-out workload without a shared ledger. Compromise of any single instance does not yield a reusable credential, because the model has no standing key and acceptance demands forward progression along the slope. The same trust layer spans heterogeneous substrates, so a call that crosses from one tenant, region, or administrative domain into another is validated by memory-resolved behavior rather than by a registry shared across that boundary, with a scope tag carrying the cross-domain policy context. And because security reduces to the unpredictability of per-step inputs and the preimage resistance of the underlying hashes rather than to hardness assumptions targeted by Shor-type attacks, the disclosure characterizes the model as post-quantum aligned.

Boundary Conditions

This is an enabling application of the disclosed mechanism, not a turnkey product, and it inherits the disclosure's honest limits. The security of a stateless instance rests on the quality of its non-exported unpredictability: the disclosure parameterizes forgery resistance by the min-entropy of the per-step contribution after extraction, so a platform whose instances expose too little usable local state, or whose hardware anchor and salt are weak, gets a correspondingly weaker guarantee. The local-state path depends on stability-tuned projections and error-tolerant sketches so that benign measurement fluctuation does not trip spurious rejections while genuine role or context changes still force rekeying; tuning that envelope is deployment work, and a poorly calibrated acceptance radius will either admit drift or generate false rejections. Verifiers still need access to some prior trusted anchor or a bounded proof to validate a successor, so a platform must retain sparse anchors somewhere even though the functions stay stateless. Host entanglement assumes the executing worker can produce a valid, signed trace that opens to its device hash; a platform that cannot expose a trustworthy device anchor on its workers loses that containment property.

Interoperability with existing certificate-based services runs through the disclosure's segregated legacy adapter, which constructs a session-scoped fallback identifier that is explicitly walled off from slope formation, so bridging to a legacy verifier is possible but deliberately isolated rather than seamless. None of these are platform throughput or latency figures, and this application does not assert any.

Disclosure Scope

The identity, authentication, and provenance mechanisms described here are disclosed in United States Patent Application 19/388,580, which specifies memory-native identity expressed as a verifiable trust slope, keypair-free successor derivation from local unpredictability and a volatile salt, two-stage message authentication, host-entangled mutation, append-only lineage with bounded proofs, predictive verification, and segregated legacy interoperability. The serverless and function-as-a-service framing in this article, including references to function instances, microVMs, sandboxes, cold starts, gateways, multi-tenant boundaries, and platform control planes, is external domain context provided to illustrate one faithful enabling implementation; it is not part of the disclosure and does not define or limit it. Any specific platform behaviors, service-level expectations, or regulatory and compliance obligations that a given serverless deployment must satisfy are determined by that deployment's own environment and governing requirements, not by the cited application, and nothing here should be read as a representation about a particular commercial platform.

Keyless Identity (/keyless-identity) All 40 steps → (/inventive-steps)

Identity from accumulated continuity. Post-quantum by construction.

U.S. 19/388,580 (/patents/19-388580)

PRIMARY TECHNICAL DISCLOSURE

- Stateless Device Pseudonymity and Secure Messaging in Cognition-Native Systems (/articles/stateless-device-pseudonymity-and-secure-messaging-in-cognition-native-systems)

SECONDARY TECHNICAL

- Trust Slope as Identity Primitive: Cumulative Hash Chains Replace Static Credentials (/articles/keyless-identity/trust-slope-identity)
- Dual-Source Identity Derivation: Hardware Anchors and Local State Vectors Combined Per Epoch (/articles/keyless-identity/dual-source-derivation)
- Stateless Symmetric Encryption: Session Keys Derived From Current Identity State (/articles/keyless-identity/stateless-encryption)
- Two-Stage Message Authentication: Transport Continuity Screening Before Semantic Validation (/articles/keyless-identity/two-stage-auth)
- Agent-Substrate Slope Entanglement: Binding Every Mutation Step to Its Execution Host (/articles/keyless-identity/slope-entanglement)
- Append-Only Mutation Lineage Log: Forward-Secure Identity Transition Chains (/articles/keyless-identity/mutation-lineage-log)
- Cumulative Slope Validation Across Substrates: Multi-Node Provenance Verification (/articles/keyless-identity/cross-substrate-validation)
- Quorum-Based Identity Recovery: Peer Attestation After Memory Loss (/articles/keyless-identity/quorum-recovery)
- Entropy Anchor Rotation: Proactive Identity Reseeding With Forward Links (/articles/keyless-identity/anchor-rotation)
- Biometric-Assisted Reseeding: Privacy-Preserving Fuzzy Extractors for Anchor Rotation (/articles/keyless-identity/biometric-reseeding)
- Delayed Slope Validation: Bounded Proof Windows for Disconnected Environments (/articles/keyless-identity/delayed-validation)
- Sparse Trust Slope Recovery: Compact Checkpoints for Resource-Constrained Devices (/articles/keyless-identity/sparse-checkpoints)
- Predictive Identity Validation: Drift Detection Before Full Discontinuity (/articles/keyless-identity/predictive-drift)
- Legacy PKI Fallback: Session-Scoped Adapters With Strict Isolation Boundaries (/articles/keyless-identity/pki-fallback)
- Post-Quantum Alignment: Hash-Based Security Without Vulnerable Hardness Assumptions (/articles/keyless-identity/post-quantum)
- Hardware-Anchor Embodiment of the Continuity-Identity Processor (/articles/keyless-identity/continuity-identity-processor-ic)

APPLICATIONS · GENERAL

- **Keyless Workload Identity for Serverless Functions: Authenticating Ephemeral FaaS Without a Persistent Keypair (/articles/keyless-identity/serverless-workload-identity)**
- Verifiable Agent Identity Without Credentials: Cryptographic Lineage for Distributed AI Agents (/articles/keyless-identity/trust-slope-entanglement)
- Post-Quantum Identity Migration Without a Public-Key Rebuild (/articles/keyless-identity/post-quantum-migration)
- IoT Device Authentication at Fleet Scale Without Keys or Certificates (/articles/keyless-identity/iot-authentication)
- Keyless Financial Identity Verification Without Credential Databases (/articles/keyless-identity/financial-identity-verification)
- Patient Matching Without a National Identifier: Keyless Identity for Cross-Institutional Patient Continuity (/articles/keyless-identity/healthcare-patient-continuity)
- Supply Chain Authentication Without PKI (/articles/keyless-identity/supply-chain-authentication)
- Keyless Smart Building Access Control: Credential-Free Entry Through Behavioral Continuity (/articles/keyless-identity/smart-building-access)
- Stopping Relay Attacks and Fob-Sharing: Binding Vehicle Access to the Driver, Not the Key (/articles/keyless-identity/vehicle-operator-identity)
- Refugee Identity Without Documents: A Keyless, Database-Free Approach (/articles/keyless-identity/refugee-identity)
- Licensing Keyless Identity at the Silicon Layer: Component-Level IP for Verifiable Device Provenance (/articles/keyless-identity/silicon-vendor-licensing)
- How Do Agents Prove Identity Without a Static Secret or an Issuer? The Market Is Converging on Keyless Continuity (/articles/keyless-identity/agent-identity-convergence)
- Drone Swarm Identity Under Jamming: Keyless Authentication When There Is No Certificate Authority to Reach (/articles/keyless-identity/drone-swarm-jammed-identity)
- Forced Off the Keys: Why the Post-Quantum Migration Points Past PKI to Keyless (/articles/keyless-identity/post-quantum-forcing-function)

APPLICATIONS · SPECIFIC

- Okta Centralized Enterprise Identity. The Keys That Prove It Are Still Stored Somewhere. (/articles/keyless-identity/okta)
- Auth0 Made Developer Identity Easy. The Credential Model Underneath Did Not Change. (/articles/keyless-identity/auth0)
- YubiKey Made Hardware Authentication Practical. The Key Is Still the Vulnerability. (/articles/keyless-identity/yubico)

- CLEAR Made Airport Identity Fast. It Built a Biometric Database to Do It. (/articles/keyless-identity/clear)
- Worldcoin Scans Irises to Prove Humanity. The Proof Depends on a Central Enrollment System. (/articles/keyless-identity/worldcoin)
- Jumio Automated ID Verification. The Verification Still Depends on Documents. (/articles/keyless-identity/jumio)
- Microsoft Entra Unified Cloud Identity. Identity Still Depends on Stored Credentials. (/articles/keyless-identity/microsoft-entra)
- Ping Identity Built Enterprise Federation. The Federation Depends on Shared Secrets. (/articles/keyless-identity/ping-identity)
- OneLogin Simplified Enterprise SSO. The SSO Token Is Still a Credential. (/articles/keyless-identity/onelogin)
- Duo Security Made MFA Ubiquitous. The Second Factor Is Still a Credential. (/articles/keyless-identity/duo-security)
- Thales HSMs Protect Key Material. The Keys Still Exist. (/articles/keyless-identity/thales-hsm)
- Entrust Issues Digital Certificates. The Certificate Is a Stored Credential. (/articles/keyless-identity/entrust)
- DigiCert Secures the Web With TLS Certificates. The Certificate Model Has Structural Limits. (/articles/keyless-identity/digicert)
- Let's Encrypt Made TLS Free. The Certificate Model Is Still the Same. (/articles/keyless-identity/lets-encrypt)
- Qorvo Secure Elements Authenticate, but Don't Track Continuity (/articles/keyless-identity/qorvo-secure-element)
- NXP Trust Anchor Stores Keys, Not Trust Slope (/articles/keyless-identity/nxp-trust-anchor)
- Infineon Secure Microcontrollers Need Continuity Logic On-Die (/articles/keyless-identity/infineon-secure-microcontroller)
- Microchip Trust Platform Fits Hardware, Misses Behavioral Continuity (/articles/keyless-identity/microchip-trust-platform)
- Indicio SSI Network and Anonyome Labs (/articles/keyless-identity/indicio-ssi)
- Sovrin Foundation Self-Sovereign Identity (/articles/keyless-identity/sovrin-foundation)
- W3C Decentralized Identifiers (DIDs) (/articles/keyless-identity/w3c-dids)
- W3C Verifiable Credentials (/articles/keyless-identity/w3c-verifiable-credentials)
- Keycard: Token-Issuer IAM Reaches Toward Identity Continuity (/articles/keyless-identity/keycard)
- Aembit: External Attestation Is Half the Answer (/articles/keyless-identity/aembit)

- Astrix Security: Discovery and Governance Without a Substrate (/articles/keyless-identity/astrix-security)
- Oasis Security: Non-Human Identity Governance, Externally Anchored (/articles/keyless-identity/oasis-security)
- Token Security: NHI Catalog Without Cryptographic Continuity (/articles/keyless-identity/token-security)
- Entro Security: Secret Discovery vs. Secret Elimination (/articles/keyless-identity/entro-security)

Keyless Identity overview → (/keyless-identity)