Nick Clark · Published July 2, 2026 · PDF

How to Move to Post-Quantum Identity Without PKI

If you own an authentication path built on long-lived keypairs and certificate authorities, you have a migration problem: Shor-vulnerable key exchange, revocation that fails offline, and trust anchors you cannot reach on a jammed or spaceborne link. This guide walks through an alternative architecture that derives identity from locally retained unpredictability and a verifiable history of behavior instead of a persistent private key. It describes an architecture disclosed in United States Patent Application 19/388,580, the Keyless Identity inventive step, not a shipping library you can install.

What You Are Building

You are building an authentication path that does not rest on a persistent public-private keypair or an external certificate authority. The searcher's problem is concrete: you have systems that authenticate with PKI, you know the key-exchange primitives underneath are exposed to quantum attack, and the usual "move to post-quantum" advice is to swap in a lattice-based signature or KEM and keep the same certificate machinery. That still leaves you with long-lived key material, a trust hierarchy, and revocation that assumes you can reach a registry.

The architecture in this guide takes a different route. Identity is expressed as a *trust slope*: the cumulatively validated sequence of dynamic hashes an agent or device produces as it evolves, where each step is a verifiable descendant of the previous one. A receiver authenticates a peer by checking that the presented step is an on-slope successor of a step it already trusts, using only locally retained state. There is no keypair to steal, no certificate to expire, and no authority to call. This is the design a developer would reach for when the deployment is decentralized, memory-constrained, intermittently connected, or otherwise hostile to standing credentials.

Why the Obvious Approaches Fall Short

PKI is a well-understood, widely deployed system, and for the connected web it works. The structural point is narrower: several of its assumptions are exactly the ones a quantum-era or disconnected deployment cannot hold.

- **Long-lived key material.** A private key is a standing secret. As long as it exists it can be exfiltrated, and if the underlying hardness assumption (integer factorization, discrete log) falls to Shor's algorithm, historical traffic and any retained keys are exposed. Swapping the primitive for a post-quantum signature removes the Shor exposure but keeps the standing secret and its whole lifecycle.
- **A trust anchor you have to reach.** Certificate validation and revocation checking assume connectivity to a hierarchy or a registry. On a delay-tolerant, mesh, opportunistic, or spaceborne link, that round trip may be minutes, hours, or simply unavailable. Revocation that cannot propagate is revocation that does not work.
- **Correlatable, persistent identifiers.** A certificate is stable by design, which makes it a durable handle for correlation across contexts.

The filed specification frames this directly: conventional identity and authentication rely on persistent keypairs and signature-based validation, which exposes users to key compromise, metadata correlation, revocation failure, and quantum attack, and PKI in particular assumes centralized trust anchors and persistent key material that are ill-

suited to decentralized or memory-constrained environments. The gap is not that PKI is badly built; it is that its core dependencies do not survive the environment you are targeting.

The Architecture

Every mechanism below traces to United States Patent Application 19/388,580. The unifying idea: replace the standing secret with locally retained unpredictability, and replace the certificate check with a continuity check against a history you already hold.

Identity as a trust slope. A device or agent expresses identity as a sequence of Dynamic Device Hashes (DDH) or Dynamic Agent Hashes (DAH). Each step is computed from the immediately prior step plus a source of non-exported unpredictability and a volatile, non-repeating salt, under an update rule with a domain-separating tag. The specification gives the form directly, for example `DAH_t = H(DAH_{t-1} || Ext(X_t) || salt_t || tag)` where `X_t` derives from a local state vector and `Ext` is a strong extractor, or `DAH_t = H(DAH_{t-1} || KDF(HWID, salt_t) || tag)` using a hardware anchor with a volatile salt. Both may be combined in one step. Because each step binds to the prior step and to unpredictability the device never exports, an attacker who lacks that local state or salt cannot feasibly synthesize valid successors.

Two sources of unpredictability. The spec is deliberately source-agnostic. A constrained device can use a static hardware anchor (TPM, TEE, or SoC identifier) combined with a per-epoch volatile salt. A richer platform can collect locally observable signals into a *local state vector* (monotonic counters, high-resolution timing deltas, scheduler jitter, I/O inter-arrival micro-jitter, and similar), normalize and project them, and run them through an extractor to yield a bounded token. Stability-tuned projections and error-tolerant sketches are used so that small measurement fluctuations produce the same token, while a genuine role or context change intentionally flips a controlled subset of bits and forces the identity to move.

Stateless symmetric messaging. There is no key exchange. A sender derives a symmetric key by applying a key-derivation function to the *recipient's current* DDH or DAH plus a domain-separating context, encrypts the payload with authenticated encryption, and embeds its own current DAH inside the ciphertext. The message carries the sender's current DAH in the transport header and the encrypted payload; it does not carry the symmetric key. This yields two-stage validation at the receiver: a fast header continuity screen rejects malformed traffic before decryption, and after the receiver derives the same key from its own current identity and decrypts, it checks the embedded sender DAH against the sender's slope to defeat post-decryption substitution.

Delayed and bounded validation. Continuity does not require live connectivity. Each step is committed into a compact append-only chain with periodic anchors, so a sender can attach a bounded set of per-step proofs (the extractor tokens and/or keyed derivations and salts for the missing steps) that let a disconnected receiver deterministically replay the slope forward from its last trusted anchor to the presented identity. If the receiver's stored state predates the available anchor, it requests a bounded checkpoint. The spec calls out delay-tolerant, mesh, opportunistic, and spaceborne links as target environments for exactly this path.

Why this is post-quantum aligned. The security of the scheme does not reduce to factoring or discrete log. Per the spec, it reduces to the min-entropy of the per-step unpredictability and the preimage resistance of the hashes and extractors. Writing `λ` for the effective min-entropy after extraction, an offline next-step forgery succeeds with probability about `2^{-λ}`, and under Grover-style quantum search the generic attack gets only a quadratic speedup to about `2^{-λ/2}`. The spec's stated parameter guidance is 256 to 512-bit extractor outputs and hash digests. Note the honest boundary here: these are the analytic properties the disclosure asserts from its construction, not measured results from a deployed system.

Supporting mechanisms. The disclosure also describes: substrate entanglement, where an agent's mutation is bound to the executing host's device identity by a host mutation token and a signed entanglement trace, so a step cannot be forged off-substrate; quorum-based recovery, where a device that loses its state reseeds and rejoins the trust graph by aggregating attestations from previously trusted peers rather than restoring a secret; entropy-anchor rotation with forward links, so an identity epoch can be refreshed without breaking auditability; and a strictly isolated legacy bridge that lets you interoperate with existing PKI counterparties by deriving a session-scoped fallback identifier that is never hashed into the trust slope.

How to Approach the Build

You are implementing this yourself. The steps below are the order the disclosure suggests; the interface sketches are illustrative and faithful to the spec, not a package to import.

1. **Choose your unpredictability source.** Decide per device class: hardware anchor plus volatile salt for constrained endpoints, local state vector plus extractor for richer platforms, or the hybrid that hashes both into each step. This choice sets everything downstream, and the validation logic stays uniform across the three.
2. **Fix the update rule and its domain separation.** Pin the hash, the extractor or KDF, the salt discipline (non-repeating per device per epoch), and the domain-separating tag. Illustrative shape, spec-faithful:

   ```
   next(prev_hash, unpredictability, salt, tag):
       return H(prev_hash || unpredictability || salt || tag)
   ```

   The volatile salt is what preserves freshness even when a hardware anchor is constant, so treat single-use salting as non-negotiable.

3. **Design the local state vector and extractor** if you use local state. Build the feature map with normalization, clipping, signed random projections under a public seed, and locality-sensitive binarization so ordinary jitter yields a stable token while a real role change moves it. Calibrate the acceptance radius to observed intra-role variation.
4. **Establish the slope root and mutation classes.** Compute `DAH_0` / `DDH_0` from the anchor or extractor output, optionally binding a semantic context vector and memory-state indicator. Record a mutation class (role update, delegation, policy commit, migration) on each step so provenance is auditable.
5. **Implement receiver-side continuity validation.** Store the last trusted successor per sender and context. On receipt, reconstruct the expected successor neighborhood and check the presented value is an on-slope successor within policy-bounded continuity parameters. Enforce monotonic progression and non-reuse within a replay horizon; reject regressions and repeats as replay.
6. **Wire the two-stage message path.** Derive the symmetric key from the recipient's current identity, encrypt, embed the sender DAH, and put the current DAH in the header. On the receive side: header continuity screen, then derive-and-decrypt, then embedded-DAH check. Define the fallback for recipient identity drift (bounded rekey handshake or checkpoint request under a fixed retry window to avoid oracle leakage).
7. **Add the append-only lineage, anchors, and checkpoints.** Fold each entry into a cumulative chain hash and emit a periodic anchor every J entries so you can prove long histories compactly. Implement bounded-proof-window replay for delayed and sparse receivers.
8. **Add the operational paths you need:** entanglement traces if agents migrate across hosts, quorum recovery for state loss, anchor rotation for freshness, and the isolated legacy adapter if you must talk to PKI systems. Keep the isolation boundary fail-closed: no PKI artifact may ever enter a DAH/DDH update.

What This Does Not Give You

Be clear-eyed about scope.

- **This is an architecture, not a drop-in library.** There is no SDK to install and nothing here "just works." You implement the hash choice, extractor, salt discipline, policy bounds, storage, and transport yourself, and the security depends on getting those right, particularly the min-entropy of your per-step inputs.
- **It is a patent disclosure, not a benchmarked product.** The quantitative properties (the `2^{-λ}` and `2^{-λ/2}` forgery bounds, the parameter ranges) are what the disclosure asserts analytically from its construction. Treat them as design targets to validate, not as measured performance of a shipping system.
- **It does not replace PKI where PKI is the requirement.** If a counterparty mandates certificate-based authentication, you interoperate through the isolated legacy bridge; you do not eliminate PKI from that exchange. The trust-slope model governs your side.
- **Correct behavior rests on policy and local state you must protect.** Continuity bounds, replay horizons, checkpoint cadence, and the confidentiality of the non-exported unpredictability are all your responsibility. Weak entropy or loose bounds undermine the whole model.

Disclosure Scope

The approach described here, memory-native identity and authentication expressed as a verifiable trust slope of dynamic hashes, with transiently derived symmetric keys, two-stage message validation, and delayed or bounded proof windows, and without persistent keypairs or an external certificate authority, is disclosed in United States Patent Application 19/388,580. This guide is educational. It explains the architecture so a developer can understand and build it, and it is not a warranty, a guarantee of security or performance, or an offer of software. Implementations, parameter choices, and their security are the responsibility of the implementer.

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)

- Authenticating Spaceborne and Interplanetary Links: Keyless Identity for Delay-Tolerant Networks With No Reachable Certificate Authority (/articles/keyless-identity/spaceborne-dtn-authentication)
- Authenticating Federated Learning Nodes Without Keypairs or a Certificate Authority (/articles/keyless-identity/federated-learning-node-authentication)

APPLICATIONS · SPECIFIC

- Okta Alternative for Keyless Identity: Federated IdP vs Credential-Free Continuity (/articles/keyless-identity/okta)
- Auth0 Alternative: Keyless Identity Beyond Stored Credentials (/articles/keyless-identity/auth0)
- YubiKey Alternative for Keyless Identity: Beyond the Stored Private Key (/articles/keyless-identity/yubico)
- CLEAR Alternative: Biometric Identity Without a Stored Template Database (/articles/keyless-identity/clear)
- Worldcoin Scans Irises to Prove Humanity. The Proof Depends on a Central Enrollment System. (/articles/keyless-identity/worldcoin)
- Jumio Alternative for Continuity-Based Identity: Keyless Identity Beyond Document Verification (/articles/keyless-identity/jumio)
- Microsoft Entra Alternative: Keyless Identity Beyond Stored Credentials (/articles/keyless-identity/microsoft-entra)
- Ping Identity vs Keyless Identity: A Post-Quantum Alternative to PKI-Bound Federation (/articles/keyless-identity/ping-identity)
- OneLogin Alternative: Keyless Identity Beyond the Stored SSO Credential (/articles/keyless-identity/onelogin)
- Duo Security Made MFA Ubiquitous. The Second Factor Is Still a Credential. (/articles/keyless-identity/duo-security)
- Thales HSM vs Keyless Identity: Protecting Keys or Eliminating Them? (/articles/keyless-identity/thales-hsm)
- Entrust Alternative: Keyless Identity Beyond Certificate-Based Trust (/articles/keyless-identity/entrust)
- DigiCert Alternative for Keyless Identity: Beyond Certificate-Chain Trust (/articles/keyless-identity/digicert)
- Let's Encrypt Alternative: Keyless Identity Beyond the Certificate Model (/articles/keyless-identity/lets-encrypt)
- Qorvo Secure Element Alternative: Continuity Identity Above the Root of Trust (/articles/keyless-identity/qorvo-secure-element)

- NXP EdgeLock Secure Element Alternative: Continuity Above Key Custody (/articles/keyless-identity/nxp-trust-anchor)
- Infineon OPTIGA Alternative: Keyless Continuity Identity Beyond Stored-Key Roots (/articles/keyless-identity/infineon-secure-microcontroller)
- Microchip Trust Platform Alternative: Keyless Identity Above the Secure Element (/articles/keyless-identity/microchip-trust-platform)
- Indicio SSI alternative: keyless identity beyond wallet-held keys (/articles/keyless-identity/indicio-ssi)
- Sovrin Foundation Alternative: Keyless Identity Beneath Self-Sovereign Identity (/articles/keyless-identity/sovrin-foundation)
- W3C DIDs vs keyless identity: who holds the controller's keys? (/articles/keyless-identity/w3c-dids)
- W3C Verifiable Credentials and Keyless Holder Binding (/articles/keyless-identity/w3c-verifiable-credentials)
- Keycard Alternative: Issuer-Free Agent Identity Beyond Token-Based IAM (/articles/keyless-identity/keycard)
- Aembit Alternative: Carried Continuity Beyond External Attestation (/articles/keyless-identity/aembit)
- Astrix Security Alternative: Keyless Identity Beneath the NHI Governance Overlay (/articles/keyless-identity/astrix-security)
- Oasis Security Alternative: Keyless Non-Human Identity Beyond External Lifecycle Anchoring (/articles/keyless-identity/oasis-security)
- Token Security Alternative: NHI Catalog vs. Keyless Cryptographic Continuity (/articles/keyless-identity/token-security)
- Entro Security Alternative: Secret Discovery vs. Keyless Secret Elimination (/articles/keyless-identity/entro-security)
- SPIFFE/SPIRE alternative: workload identity without a certificate authority or persistent keypair (/articles/keyless-identity/spiffe-spire)
- HashiCorp Vault vs a keyless trust slope: where does the identity of the caller come from? (/articles/keyless-identity/hashicorp-vault)

Keyless Identity overview → (/keyless-identity)