
RESEARCH ARTICLE

Adaptive identity and session security architecture for mobile healthcare applications

Vamsi Krishna Minnakanti

Independent Researcher, USA

ORCID: 0009-0004-9758-5240

Corresponding Author: Vamsi Krishna Minnakanti, **E-mail:** vamsikrishna3106@gmail.com

ABSTRACT

Mobile healthcare applications give patients access to prescriptions, appointments, clinical records, insurance information, and direct communication with providers, and that access carries information sensitive enough to demand stronger protection than a typical consumer application requires. At the same time, authentication that is too rigid works against the population it is meant to protect: repeated sign-in prompts, brittle recovery flows, and fixed session expiration discourage adoption and can delay care for patients who are already unwell, unfamiliar with the technology, or working from an unreliable connection. This article proposes an adaptive identity and session security architecture for mobile healthcare that treats protection and access as a single design problem rather than a tradeoff to be split down the middle. The architecture separates authentication, authorization, patient identity, device trust, and clinical consent into distinct governed layers; anchors native application authorization in OAuth 2.0 with Proof Key for Code Exchange; uses device-level biometrics to unlock, rather than replace, backend identity validation; stores credentials and tokens in hardware-backed encrypted storage; evaluates session risk on a continuous basis instead of only at login; and applies session policy that scales with the sensitivity of the operation a patient is attempting. This article draws on standards guidance from the Internet Engineering Task Force, the National Institute of Standards and Technology, OWASP, and the FIDO Alliance, as well as recent empirical and review literature on mobile healthcare security, biometric authentication, and zero trust architecture. Unlike prior frameworks that treat authentication, authorization, and session governance as separate controls, this article proposes an integrated six-component architecture where each component, identity separation, OAuth 2.0/PKCE authorization, biometric reauthentication, hardware-backed storage, continuous risk assessment, and operation-sensitive session governance, functions as a coordinated input into a single, continuously evaluated access decision. The contribution is the integration model itself: a governance structure designed to address the protection-access tension architecturally rather than through policy tradeoff. The architecture is proposed as a conceptual governance model; experimental validation of its performance claims remains as future work.

KEYWORDS

Mobile Health Security, OAuth 2.0, Biometric Authentication, Adaptive Session Management, Continuous Risk Assessment

ARTICLE INFORMATION

ACCEPTED: 10 September 2026

PUBLISHED: 10 October 2026

DOI: 10.32996/jcsts.2026.8.10.5

1. Introduction

Mobile healthcare applications have moved from a convenience layer sitting alongside traditional care delivery to a primary channel through which patients manage prescriptions, schedule appointments, review clinical results, and communicate with providers. That shift increases the risk in two ways. The information these applications handle, prescription histories, diagnoses, insurance identifiers, and appointment details tied to specific conditions, is more sensitive than what a typical retail or productivity application stores. A compromise of that information carries consequences a lost shopping cart does not, and the sector has borne this out: a review synthesizing 99 research papers on healthcare data breach mitigation found electronic health records to be the most frequently addressed breach domain in the literature, with attack surfaces spanning electronic health records, connected devices, and the mobile applications patients use directly (Nemec Zlatolas et al., 2024).

The second front is access. Patients who use mobile healthcare applications are not a uniform population of technically comfortable users logging in from a stable network. Many are managing an active health condition. Some are older adults with lower tolerance for authentication friction. A large share are relying on inconsistent mobile connectivity outside of controlled office or hospital networks. Pushing authentication too hard in this population, with repeated sign-in requests, complicated account recovery, and session policies that expire without regard to what the patient is actually doing, leads not only to frustration but also to a real risk that patients delay or abandon care tasks they need to complete (Rezaee et al., 2023). This tension is not hypothetical. A systematic review of mHealth application privacy and security found that patients themselves weigh confidentiality concerns against the practical burden of accessing the application and that overly burdensome security measures can erode the trust the application depends on just as much as a breach can (Alhammad et al., 2024).

This article proposes an adaptive identity and session security platform built around the idea that protection and access are not opposing goals to be balanced through compromise but a single design problem that a sufficiently adaptive architecture can resolve. Six components structure the proposal: an identity architecture that separates authentication, authorization, patient identity, device trust, and clinical consent into distinct governed concerns; OAuth 2.0 authorization built on the Proof Key for Code Exchange extension for native mobile clients (Hardt, 2012; Sakimura et al., 2015); biometric authentication used to unlock cryptographic material and reauthenticate, never as a replacement for backend identity validation; encrypted credential storage backed by hardware where the device supports it; continuous, multi-signal risk assessment that treats authentication as an ongoing evaluation rather than a completed event; and session governance that scales its restrictions to the sensitivity of the specific operation a patient is attempting rather than applying one policy to an entire session.

The argument threading through these components is that static authentication, a single check performed at login and left unexamined until the session ends or a fixed timer expires, cannot answer how risk actually behaves in a mobile healthcare context. Risk changes continuously: a device can become compromised mid-session, a network can shift from trusted to suspicious, and a user can attempt an operation far more sensitive than anything else in that session. An architecture that evaluates identity and session risk on a continuous basis and that scales its response according to what a patient is actually trying to do can hold both protection and access as first-order goals rather than trading one for the other.

The primary contribution of this article is the integration architecture: a governance model that coordinates six distinct identity and session concerns into a single continuously evaluated access decision, rather than treating them as independent controls applied in sequence. This distinguishes the proposal from existing standards guidance, which addresses each control individually, and from prior review literature, which documents adoption gaps without proposing a unified resolution.

The remainder of this article proceeds as follows. Section 2 describes the identity architecture that separates authentication from authorization, patient identity, device trust, and consent. Section 3 details OAuth 2.0 and PKCE as the authorization foundation for native mobile clients. Section 4 addresses biometric authentication as a reauthentication layer. Section 5 covers encrypted storage and credential protection. Section 6 introduces continuous risk assessment as an ongoing evaluation rather than a one-time gate. Section 7 describes session governance tied to operation sensitivity. Section 8 addresses fault tolerance and workflow continuity under real-world network conditions. Section 9 discusses privacy and regulatory alignment. Section 10 proposes an evaluation framework. Section 11 considers broader implications beyond healthcare, and Section 12 concludes.

2. Identity Architecture for Mobile Healthcare

A recurring weakness in mobile application security design is treating authentication as though it answers every question a healthcare workflow needs answered. It does not. Successful authentication confirms that a user presented valid credentials, nothing more. It says nothing on its own about whether that user should be permitted to view a specific patient's clinical record, modify prescription delivery instructions, or access a function reserved for a caregiver acting on a dependent's behalf. The architecture proposed here treats authentication, authorization, patient identity, device trust, and clinical consent as five distinct concerns, each governed separately, each contributing to an access decision rather than any one of them making that decision alone.

Authentication answers the question of who is presenting credentials. Authorization answers what the authenticated identity is permitted to do. Patient identity resolves which specific patient record or set of records a request concerns, which matters directly in family or caregiver account structures where one authenticated user may have legitimate access to more than one patient's information. Device trust captures whether the device making the request meets the security posture the operation requires, a concern that has grown more central as zero trust models move away from treating network location as a proxy for trustworthiness (Gambo & Almulhem, 2026). Clinical consent tracks whether the patient has authorized a specific use of their information, separate from whether the requesting party is otherwise permitted to access it.

Backend policy combines these five concerns using OAuth scopes and role-based rules that evaluate, for each request, the user's role, the specific operation requested, the patient relationship involved, whether consent has been captured for that use, the

device's trust posture, any applicable geographic or legal restriction, and the current session risk level. Multi-factor structures that combine possession, knowledge, and inherence-based signals have been shown in reviews of authentication within connected healthcare environments to reduce single-point failure risk far more effectively than any one factor applied alone, which is part of why this architecture treats identity as a composite of several governed signals rather than a single credential check (Suleski et al., 2023). No single signal is sufficient on its own. A correctly authenticated user on a fully trusted device can still be denied a specific operation if consent has not been captured or if the patient relationship does not support it, and a request that would otherwise be permitted can be restricted when session risk has risen. This layered structure is what allows the remaining sections of this architecture, biometric reauthentication, encrypted storage, continuous risk assessment, and operation-sensitive session governance, to function as coordinated inputs into a single access decision rather than as isolated controls bolted onto a conventional login flow.

3. OAuth 2.0 and Secure Native Application Authorization

Native mobile applications require an authorization approach suited to a client that cannot reliably keep a secret. A mobile application's binary can be decompiled, and any credential embedded in it has to be treated as already exposed. The OAuth 2.0 Authorization Framework anticipates exactly this problem, defining a category of public clients that cannot hold a confidential client secret (Hardt, 2012). The Proof Key for Code Exchange extension closes a specific gap in that model for native applications: an intercepted authorization code cannot be redeemed by an attacker, because redemption now requires a code verifier that only the legitimate client generated (Sakimura et al., 2015). RFC 9700 formalizes the current recommended pattern for native applications specifically, directing implementations toward the operating system's external user agent, rather than an embedded web view, for the authorization step, since an external browser benefits from the platform's own security posture, cookie isolation, and the user's ability to recognize a familiar authentication surface (Lodderstedt et al., 2025).

Beyond the authorization code exchange itself, several operational choices determine whether this foundation holds up in practice. Access tokens should be short-lived, limiting the exposure window if one is captured. Refresh tokens should be rotated on each use and immediately revocable so that a compromised refresh token has a narrow and closing window of usefulness rather than a standing one. The backend must validate every token presented to it on four dimensions before honoring a request: the intended audience, the issuer, the expiration, and the requested scope, alongside signature verification, since accepting a token without checking all four leaves gaps that a token valid for one purpose could be replayed against another.

Deep link and custom URL scheme handling deserves particular attention in mobile OAuth flows, because it is a point where a malicious application can attempt to intercept a redirect intended for the legitimate client. Android's verified app links and iOS's universal links both provide platform-level mechanisms to bind a redirect URI to a specific, cryptographically verified application, and an implementation that skips this binding in favor of a generic custom scheme reopens exactly the interception risk PKCE was designed to close on the authorization code itself. Known divergence between OAuth specification requirements and real-world implementations, particularly around redirect handling and authorization code interception, is a documented risk in the security literature, which is why redirect handling belongs in the architecture's threat model rather than being treated as a solved implementation detail (Hardt, 2012).

4. Biometric Authentication as Reauthentication, Not Replacement

Biometric authentication has become the default reauthentication mechanism on modern mobile devices, and for a healthcare application seeking to reduce sign-in friction without weakening security, it is a natural fit, provided its role in the architecture is defined precisely. Biometric authentication should supplement backend identity validation, not substitute for it. The application should invoke the operating system's biometric APIs rather than attempting to capture, process, or store biometric data itself, both because the platform vendor's implementation has been hardened against a much broader range of attacks than an application team can replicate and because processing biometric data directly would introduce a category of sensitive data handling the architecture does not need to take on.

The practical effect of a successful biometric check should be to unlock a cryptographic key or session artifact already provisioned during a prior, fully validated authentication event, rather than to independently assert a user's identity to the backend. This design choice keeps the backend's trust anchored in the original OAuth authorization rather than in a biometric signal the backend has no way to verify was genuinely produced by the sensor. A systematic review of biometric authentication approaches across mobile environments underscores that biometric methods vary widely in reliability, spoofing resistance, and appropriateness across device categories and use contexts, reinforcing that biometrics function best as a convenience and reauthentication layer rather than as the sole determinant of access (Abu Al-Haija & Al-Salameen, 2024). Comparable findings from a review of biometric authentication specifically within mobile banking, another domain where transaction sensitivity and user convenience must both be protected, similarly conclude that biometric methods perform best when layered alongside, rather than in place of, an independently verifiable backend credential (Naji Ali & Salim Mahmood Al-Dabbagh, 2026).

Several edge cases require deliberate handling rather than allowing the platform to handle them by default. Take biometric enrollment changes: a new fingerprint or face added to the device should trigger a re-evaluation of trust for that device rather than silently continuing to honor the previous binding, since an enrollment change is exactly the signal that would indicate a device has changed hands. Device credential fallback, typically a PIN or pattern, needs its own policy rather than being treated as equivalent in strength to biometric verification. Sensor lockout after repeated failed attempts should route to full reauthentication rather than an indefinite retry loop. Rooted, jailbroken, or otherwise compromised devices undermine the operating system guarantees on which the entire biometric unlock model depends, so the architecture must detect and respond to that compromise rather than trust a biometric result from an environment it cannot verify. The risk engine response to a detected device integrity signal is described in Section 6 and Table 1. Shared device scenarios, common in some clinical and caregiver contexts, require that biometric enrollment be scoped tightly enough that one enrolled biometric cannot silently authorize access on behalf of a different patient. High-risk healthcare operations, changing a delivery address for controlled medications, and modifying consent on file should require full reauthentication regardless of a recent successful biometric check, a principle that connects directly to the operation-sensitive session governance described in Section 7. Accessibility matters here too. Biometric performance and usability vary meaningfully across age groups, and a design that assumes uniform biometric reliability risks excluding exactly the older adult population that mobile healthcare applications most need to serve well (Wang & Zhou, 2023).

5. Encrypted Storage and Credential Protection

Tokens, identifiers, and any other sensitive material the application retains locally should be stored using hardware-backed keystores where the device provides one, rather than in application-level storage protected only by the operating system's general sandboxing. This distinction matters more than it might appear. A large-scale empirical analysis of roughly 490,000 Android applications found that 56.3% of applications self-reporting as processing sensitive user data did not use the device's trusted hardware capabilities at all, relying instead on weaker software-based protection; the same study measured a real performance cost for the strongest hardware tier, with secure element encryption of 1 MiB taking approximately 15 seconds against about 0.4 seconds for trusted execution environment-based storage (Blessing et al., 2026). That adoption gap, and the performance tradeoff behind it, is precisely what a mobile healthcare application handling protected health information has to weigh deliberately rather than default around.

The application should never store passwords, clinical data, or long-lived credentials in plaintext, and the protection model needs to extend beyond the storage mechanism itself to the surrounding handling of that data. Key rotation should occur on a defined schedule and following any signal that a key may have been exposed. Encryption at rest needs to cover cached clinical data as well as tokens, since a device that is lost or stolen exposes whatever the application has written to disk, not only its credential store. Memory handling matters too for data held only transiently, since memory can be inspected by sufficiently privileged processes on a compromised device. The operating system's own backup mechanism is convenient for restoring a user's data to a new device, but it should explicitly exclude sensitive tokens and cached clinical data, since a cloud backup extends the attack surface well beyond the device itself. Application logs should redact identifiers, tokens, and clinical values rather than relying on log access controls alone, and clipboard access to sensitive fields should be restricted, since the clipboard is a shared system resource any other application can read. Cached data should expire on a schedule tied to its sensitivity rather than persisting indefinitely, and logout ought to trigger deliberate cleanup of cached tokens and clinical data rather than merely clearing an authentication flag while leaving the underlying data sitting in place. The OWASP Mobile Application Security Verification Standard provides a structured baseline against which each of these controls can be assessed and verified during development rather than discovered missing during an audit (OWASP Foundation, 2024).

6. Continuous Risk Assessment

Treating authentication as complete for the duration of a session, once granted at login and not re-evaluated until expiration, leaves a mobile healthcare application blind to exactly the kind of risk that develops after a session has already started. The architecture proposed here instead evaluates risk continuously, drawing on signals including a new or previously unseen device, an unusual location or network, repeated authentication failures within the session, a change in device integrity indicators, signs consistent with token replay, an attempt at an unusually high-risk operation, an extended period of inactivity followed by resumed activity, a sudden change to profile or contact information, and navigation behavior that deviates from the patterns a legitimate user of this application typically exhibits. At the risk engine level, token replay detection operates as a secondary signal alongside the refresh token rotation and revocation logic described in Section 3. When rotation detects reuse of an invalidated token, Section 3's revocation logic triggers first, revoking the entire token family. The risk engine's session termination response applies to replay signals detected through behavioral or anomaly indicators that reach the risk layer independently.

Signal combination follows a weighted scoring model in which each signal category is assigned a severity weight based on its predictive value for session compromise. Table 1 maps each signal category to its illustrative severity tier. Each signal contributes

an additive score normalized to 0–1 per category, with device integrity changes and token replay indicators assigned the highest illustrative weights (0.4 and 0.35) due to their severity. Location/network anomalies and abnormal navigation behavior receive intermediate weights (0.2 each), while lower-tier signals become more meaningful when combined with higher-severity indicators; no single signal is treated as conclusive evidence of compromise. The aggregate score is the weighted sum across all active signals at a given evaluation interval, capped at 1.0. The three response thresholds; low, medium, and high, are calibrated per deployment based on the sensitivity tier of the operation in progress rather than set as universal constants, consistent with the session governance model in Section 7. Specific weight values and threshold calibration should be determined empirically for each deployment context; the values cited here are illustrative anchors for the model's structure, not empirically validated constants.

No individual signal is treated as automatically disqualifying, since any one of them can have an innocent explanation: a patient traveling, a shared family network, or a genuinely forgotten password entered twice before succeeding. The risk engine instead combines these signals into an evaluation that can produce one of several graduated responses: allowing the session to continue unchanged, requesting a biometric confirmation as a lightweight check, requiring full reauthentication when the combined signal strength warrants it, restricting a specific sensitive action while leaving the rest of the session intact, or terminating the session outright when the evidence of compromise is strong enough to outweigh the cost of interrupting a legitimate user.

This graduated model reflects an approach increasingly explored in recent research on risk-based authentication. One hybrid framework combining ensemble machine learning with explainable AI techniques for risk-based authentication reported 97.77% accuracy, 99.41% precision, 98.04% recall, and an equal error rate of 0.0303 on a two million-record dataset, with performance holding consistent on datasets scaled up to 30 million records, indicating that graduated, signal-driven authentication decisions of the kind described here are achievable at production scale rather than only in principle (Sasikumar & Nagarajan, 2026). It also aligns with the zero trust principle that no signal, including a prior successful login, should be treated as a permanent grant of trust; trust in this model is continuously re-earned rather than issued once and assumed to persist (Gambo & Almulhem, 2026; National Institute of Standards and Technology, 2020). Applying this continuous evaluation specifically within healthcare IoT and mobile contexts has been identified in recent systematic review work as an area where enabling technologies exist but adoption still lags the threat landscape those technologies are meant to address (Ikegwu et al., 2025). Emerging work integrating cognitive threat detection with post-quantum cryptographic protection for healthcare-connected devices reports F1 scores between 0.972 and 1.000 across six evaluation datasets, with handshake latencies of 3.1 to 4.4 seconds judged acceptable for non-emergency healthcare IoT traffic, suggesting that continuous verification and forward-looking cryptographic resilience can be pursued together without an unworkable latency cost in regulated healthcare environments (Hussain et al., 2026).

Table 1. Risk Signal Categories and Adaptive Session Responses

Risk Signal	Example Indicator	Adaptive Response
New or Unrecognized Device	First-time device fingerprint not previously associated with account	Request biometric confirmation before granting standard access
Unusual Location or Network	Access from a geography or IP range inconsistent with recent session history	Elevate risk score; may trigger step-up authentication
Repeated Authentication Failures	Multiple failed login or biometric attempts within a short window	Require full reauthentication; temporary lockout after threshold
Device Integrity Change	Rooted, jailbroken, or otherwise compromised device state detected	Elevate to high-severity risk tier immediately; restrict sensitive and critical-tier actions pending investigation; escalate for human review; require reauthentication from a verified device if compromise is confirmed through additional signal correlation
Token Replay Indicators	Attempted use of a token from a source inconsistent with the original session, or reuse of a refresh token after rotation has already issued a successor	Terminate the session immediately
High-Risk Operation Attempt	Request to modify credentials, consent, or controlled medication delivery	Require operation-specific reauthentication regardless of session age

Risk Signal	Example Indicator	Adaptive Response
Extended Inactivity Followed by Resumption	Session resumes after an idle period exceeding the policy threshold	Request biometric confirmation or full reauthentication depending on tier
Abnormal Navigation Behavior	Access pattern inconsistent with typical user behavior for this account	Elevate risk score; increase monitoring sensitivity

Note: Responses in this table correspond to medium-threshold and high-threshold triggers as described in the scoring model above. Low-threshold evaluation leaves the session unchanged and is not listed separately.

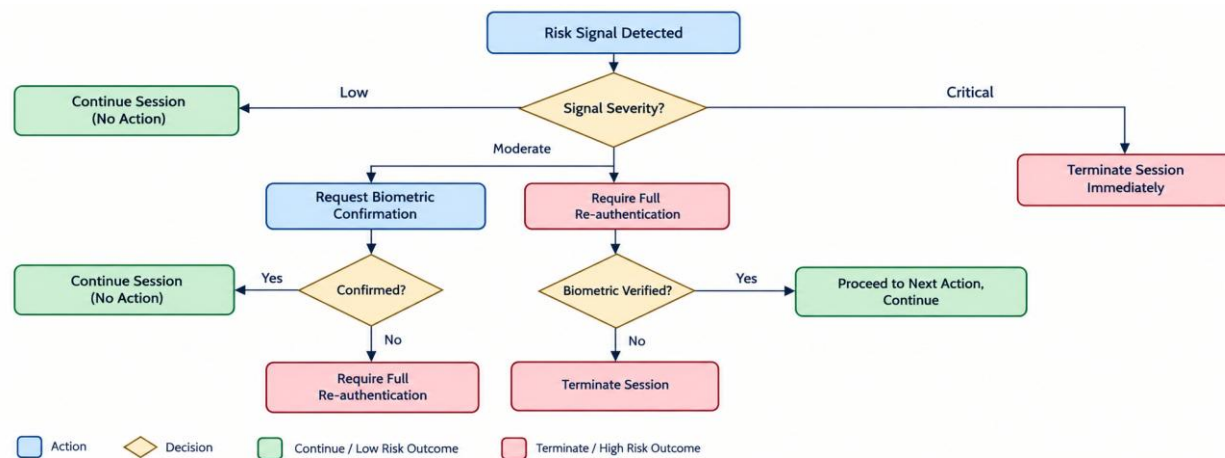


Figure 1. Adaptive Authentication and Session Decision Logic

Figure 1 illustrates the graduated response logic applied when the continuous risk engine detects an elevated signal, routing sessions toward continuation, step-up authentication, restriction, or termination based on signal severity.

7. Intelligent, Operation-Sensitive Session Governance

A session policy applied uniformly across every function in a mobile healthcare application asks the same question in every context: has enough time passed, or has too much time passed, since the user last authenticated? That is the wrong question to ask uniformly, because the operations available inside a healthcare application vary enormously in sensitivity. Viewing general pharmacy hours or browsing educational content about a medication carries little risk if a session persists a while longer than strictly necessary. Changing account credentials, viewing a protected clinical record, and modifying prescription delivery instructions carry meaningfully more risk, and a session policy that treats both categories identically ends up either too permissive for the sensitive operations or too restrictive for the routine ones.

The architecture instead applies session policy as a function of operation sensitivity rather than as a single session-wide setting. Idle timeout duration, the period of inactivity before a session is suspended, can be longer for low-sensitivity browsing and shorter for sessions where a sensitive operation has recently occurred. Absolute session lifetime, a hard ceiling independent of activity, provides a backstop regardless of how actively a session is used. Sliding expiration extends the session on continued legitimate activity rather than expiring on a fixed clock that ignores whether the user is actively engaged. Operation-specific reauthentication requires a fresh authentication event, not merely a valid existing session, before permitting specific high-sensitivity actions, consistent with the high-risk operation handling introduced for biometrics in Section 4. Background session restriction limits what a session that is backgrounded on the device, rather than actively in the foreground, is permitted to do without a fresh check. Concurrent session control governs how many active sessions a single account may hold at once and what happens when a new session begins while another remains active. Remote revocation allows a session to be terminated from outside the device itself, essential when a patient reports a lost device or suspects compromise. Device-bound sessions tie a session's validity to the specific device that established it, closing the gap where a stolen or replayed token could otherwise be used from an entirely different device.

Applying these controls as a tiered structure rather than a single policy connects session governance directly to the risk signals evaluated continuously in Section 6: a session flagged as elevated risk can have its policy tightened dynamically, shortening idle timeout or requiring operation-specific reauthentication earlier than the default tier would otherwise call for, without requiring the entire session to end. This kind of layered, context-aware session control reflects patterns increasingly described in recent security

literature on session management for high-volume applications, where static, uniform session policy is treated as a known limitation rather than an acceptable baseline (Habib et al., 2023).

The timeout values illustrated in Table 2 represent policy anchors rather than fixed constants. The rationale follows operation sensitivity directly: extended idle timeouts at the general tier reflect the low marginal risk of a persistent browsing session and the high usability cost of interrupting a patient navigating informational content. Short idle timeouts at the sensitive tier reflect that an unattended session with a visible clinical record or prescription detail represents a concrete, realized disclosure risk, not a theoretical one. Critical-tier operations carry no idle timeout because the reauthentication requirement is per-operation rather than time-based: the session does not expire the patient out; rather, the operation itself cannot proceed without a fresh authentication event, regardless of how recently the session was established. These tiers interact dynamically with the risk engine in Section 6: a session operating at the standard tier that accumulates a medium-threshold risk score is promoted to sensitive-tier policy for the remainder of that session, with its idle timeout shortened and its reauthentication window tightened, without requiring session termination.

Table 2. Session Policy Tiers by Operation Sensitivity

Sensitivity Tier	Example Operations	Idle Timeout	Absolute Session Lifetime	Reauthentication Requirement
General or Low	Browsing pharmacy hours, educational content, general navigation	Extended (e.g., 30+ minutes)	Standard daily limit	Standard session validity sufficient
Standard or Moderate	Viewing appointment schedule, refill status, non-clinical account details	Moderate (e.g., 10-15 minutes)	Standard daily limit	Recent session activity required
Sensitive or High	Viewing protected clinical records, prescription details	Short (e.g., 5 minutes)	Reduced ceiling	Recent authentication event required (e.g., within session tier window)
Critical	Changing credentials, modifying consent, and controlled medication delivery changes	N/A, per-operation reauthentication gate replaces idle timeout	N/A, always gated	Full reauthentication required regardless of session age

8. Fault Tolerance and Workflow Continuity

Security controls that assume reliable connectivity and orderly state transitions behave unpredictably under the conditions in which mobile healthcare applications actually operate: intermittent connectivity, tokens that expire mid-workflow, sign-out sequences interrupted before completion, and backend services that are themselves briefly unavailable. An architecture that does not plan for these conditions tends to fail in one of two ways: either it weakens security by falling back to a less protected state just to keep working, or it damages usability by forcing a patient to restart work they had already completed.

Token refresh requires particular care in this environment. A mobile client can issue multiple concurrent requests that each discover an expired access token and attempt to refresh it independently; without synchronization, this scenario produces a race that can invalidate a refresh token prematurely or leave the client holding an inconsistent token state. The refresh flow should be synchronized so that only one refresh attempt proceeds at a time, with concurrent requests waiting on its result rather than each initiating their own.

When a session does expire or a reauthentication step becomes necessary mid-flow, the application should preserve non-sensitive workflow context wherever that is appropriate so that a patient partway through scheduling a vaccine appointment or partway through a prescription refill is not forced to reconstruct the entire sequence of selections after reauthenticating. This distinction matters: preserving a patient's place in a workflow is a usability property, while preserving sensitive data itself across a security boundary would undermine the protections this architecture establishes. So, what persists across a reauthentication event should be limited to context that carries no sensitive value on its own, a selected appointment type rather than a clinical detail, or a step index rather than a record identifier. When a mid-workflow reauthentication event is triggered by a token expiry or refresh failure,

the session's risk score and tier classification from Section 6 and Section 7 should carry forward rather than reset to baseline. A patient who reauthenticates cleanly mid-workflow has not changed the session's prior risk context; resetting to a low-risk baseline on reauthentication would allow an elevated-risk session to shed its tier classification simply by timing out and signing back in, which is inconsistent with the continuous evaluation model this architecture depends on.

9. Privacy and Regulatory Alignment

An identity and session security architecture for mobile healthcare cannot be evaluated on technical soundness alone. It has to hold up against the privacy expectations and regulatory obligations that apply to health information specifically. The architecture should apply data minimization, collecting only the information a given function genuinely requires, purpose limitation that restricts collected data to the use for which it was gathered, explicit consent tracking, auditability of access decisions, and least privilege in how roles and scopes are defined. Analytics and diagnostic instrumentation deserve particular scrutiny here. It is easy for event tracking intended to improve the product to inadvertently capture protected health information as a side effect of logging what a user clicked or viewed.

United States guidance is direct about a limitation that is often assumed away: HIPAA's protections apply to covered entities and their business associates, not automatically to personal health information a patient chooses to store or access on their own personal device (U.S. Department of Health and Human Services, Office for Civil Rights, 2022). That distinction places real responsibility on the application's own security design, since regulatory coverage cannot substitute for architectural protection at the device and session level. A mixed-methods study surveying mobile health developers found persistent gaps between the privacy and security criteria developers consider important and what actually gets implemented, underscoring that regulatory awareness alone does not reliably translate into secure design without an architecture that operationalizes those expectations directly (Rezaee et al., 2023). Security design in this domain should therefore treat healthcare privacy requirements, applicable regulatory guidance, and established mobile security standards as inputs the architecture is built around from the outset, not criteria retrofitted after a design is otherwise complete.

10. Evaluation Framework

Assessing whether an adaptive identity and session security architecture is functioning as intended requires measurement across both security effectiveness and the usability outcomes the architecture is meant to protect. Ten candidate measurements are proposed. Unauthorized access prevention captures how effectively the layered access decision in Section 2 blocks requests that should not be permitted. Authentication completion rate measures how often users who begin an authentication flow successfully complete it, a direct signal of whether friction has become a barrier. Biometric success and fallback rate distinguish cases where biometric reauthentication succeeds on the first attempt from cases requiring fallback to a PIN or full reauthentication, indicating where the biometric layer described in Section 4 is underperforming for some portion of users. Session expiration frequency tracks how often sessions end before a user intended to finish, a usability signal tied directly to the tiered policy in Section 7. Token refresh failure rate surfaces problems in the synchronized refresh handling described in Section 8. Account recovery completion rate measures whether patients who lose access to their credentials can regain it without abandoning the process. Suspicious session detection rate evaluates how often the continuous risk assessment in Section 6 correctly identifies genuinely anomalous sessions. Security friction-driven abandonment estimates how often patients discontinue a task specifically because of an authentication or session obstacle, connecting security design directly to the access half of this article's central argument. Security incidents by device risk category break down incident rates by the device trust posture established in Section 2, testing whether device trust classification meaningfully predicts where problems occur. Accessibility of authentication workflows evaluates whether the biometric and fallback mechanisms in Section 4 perform consistently across the age and ability range of the patient population the application serves, a concern grounded directly in accessibility-focused biometric research (Wang & Zhou, 2023).

Together these measurements allow an organization to evaluate the architecture as a system rather than assessing security and usability as separate, potentially conflicting concerns, which is consistent with this article's central claim that adaptive design can hold both as first-order goals simultaneously.

Table 3. Evaluation Framework Candidate Metrics

Metric	Definition	Rationale
Unauthorized Access Prevention	Rate at which the layered access decision correctly blocks impermissible requests	Directly tests the identity architecture's core access-control function
Authentication Completion Rate	Proportion of initiated authentication flows that are completed successfully	Signals whether friction has become a barrier to legitimate use

Metric	Definition	Rationale
Biometric Success and Fallback Rate	Proportion of biometric reauthentication attempts succeeding on first try versus requiring fallback	Identifies where the biometric layer underperforms for portions of the user base
Session Expiration Frequency	Rate at which sessions end before the user intended to finish	Usability signal tied to the tiered session policy
Token Refresh Failure Rate	Rate of failed token refresh attempts	Rate of failed token refresh attempts, surfacing problems in synchronized refresh handling
Account Recovery Completion Rate	Proportion of recovery attempts that succeed without abandonment	Tests whether patients can regain access without giving up
Suspicious Session Detection Rate	Rate at which the continuous risk engine correctly identifies anomalous sessions	Tests the accuracy of continuous risk assessment
Security Friction-Driven Abandonment	Rate at which patients discontinue a task specifically due to an authentication or session obstacle	Connects security design directly to the access/usability goal
Security Incidents by Device Risk Category	Incident rate broken down by device trust classification	Tests whether device trust classification predicts risk
Accessibility of Authentication Workflows	Consistency of biometric and fallback mechanism performance across age and ability ranges	Ensures the architecture serves the full patient population, including older adults

11. Broader Implications

While this architecture is developed specifically for mobile healthcare, its core structure, layered identity governance, standards-based OAuth authorization, biometric reauthentication bound to backend validation, continuous risk assessment, and operation-sensitive session policy extends naturally to other regulated, high-stakes mobile domains. Financial services applications face a closely related tension between fraud prevention and account access, government service applications must serve populations with widely varying technical comfort while protecting sensitive personal data, and other regulated enterprise applications generally share the same underlying problem of needing security that scales with what a user is attempting rather than applying uniformly across an entire session. The specific risk signals, session tiers, and regulatory anchors would differ by domain, but the architectural principle that protection and access should be resolved together through continuous, adaptive evaluation rather than traded off against each other through a single static policy transfers directly.

12. Conclusion

Mobile healthcare security cannot be reduced to a choice between strong protection and practical patient access, because a static session policy, evaluated once at login and left unexamined until a fixed timer expires, cannot respond to how risk actually changes across a session or account for how differently sensitive the operations within a single application really are. This article has proposed an adaptive identity and session security architecture that integrates OAuth 2.0 authorization built on PKCE, biometric reauthentication bound rather than substituting for backend identity validation, hardware-backed encrypted storage, continuous multi-signal risk assessment, and session governance that scales with operation sensitivity. Grounded in established standards from the Internet Engineering Task Force, the National Institute of Standards and Technology, OWASP, and the FIDO Alliance, and supported by recent empirical and review literature spanning mobile healthcare privacy, biometric authentication, and zero trust security, the architecture is offered as a structured approach to a design problem that existing frameworks address only partially. Experimental validation, measuring unauthorized access prevention, authentication completion rate, and security friction-driven abandonment against the tiers and thresholds proposed here, remains as the necessary next step before the architecture's performance claims can be confirmed in deployment. The conceptual contribution is the integration model: a governance structure in which protection and access are resolved together through continuous, adaptive evaluation rather than traded off through a static policy applied once at login.

Funding: Funding information was not supplied in the manuscript.

Conflicts of Interest: A conflict-of-interest declaration was not supplied in the manuscript.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers.

References

- Abu Al-Haija, Q., & Al-Salameen, S. O. (2024). Biometric authentication system on mobile environment: A review. *Computer Systems Science and Engineering*, 48(4), 897–914. <https://doi.org/10.32604/csse.2024.050846>
- Alhammad, N., Alajlani, M., Abd-alrazaq, A., Epiphaniou, G., & Arvanitis, T. (2024). Patients' perspectives on the data confidentiality, privacy, and security of mHealth apps: Systematic review. *Journal of Medical Internet Research*, 26, Article e50715. <https://doi.org/10.2196/50715>
- Blessing, J., Anderson, R., & Beresford, A. R. (2026). KeyDroid: A large-scale analysis of secure key storage in Android apps. *Proceedings on Privacy Enhancing Technologies*, 2026(3), 215–235. <https://doi.org/10.56553/popets-2026-0079>
- FIDO Alliance. (2024). FIDO2: WebAuthn and CTAP specifications. <https://fidoalliance.org/specifications/>
- Gambo, M. L., & Almulhem, A. (2026). Zero trust architecture: A systematic literature review. *Journal of Network and Systems Management*, 34(1), Article 25. <https://doi.org/10.1007/s10922-025-09998-x>
- Habib, M. I., Al Maruf, A., & Nabil, M. J. A. (2023). An exploration into web session security: A systematic literature review (arXiv:2310.10687). arXiv. <https://arxiv.org/abs/2310.10687>
- Hardt, D. (Ed.). (2012). RFC 6749: The OAuth 2.0 authorization framework. Internet Engineering Task Force. <https://doi.org/10.17487/RFC6749>
- Hussain, H., Mishra, S., & Alshenaifi, R. (2026). Post-quantum cognitive zero trust architecture for healthcare IoT devices. *PLOS ONE*, 21(5), Article e0348600. <https://doi.org/10.1371/journal.pone.0348600>
- Ikegwu, A. C., Alo, U. R., & Nweke, H. F. (2025). Cyber threats in mobile healthcare applications: Systematic review of enabling technologies, threat models, detection approaches, and future directions. *Discover Computing*, 28(1), Article 152. <https://doi.org/10.1007/s10791-025-09686-z>
- Lodderstedt, T., Bradley, J., Labunets, A., & Fett, D. (2025). RFC 9700: Best current practice for OAuth 2.0 for native apps. Internet Engineering Task Force. <https://doi.org/10.17487/RFC9700>
- Naji Ali, H., & Salim Mahmood Al-Dabbagh, S. (2026). A systematic literature review on biometric authentication in mobile banking. *F1000Research*, 15, Article 5. <https://doi.org/10.12688/f1000research.173855.2>
- National Institute of Standards and Technology. (2020). Zero trust architecture (NIST Special Publication 800-207) (S. Rose, O. Borchert, S. Mitchell, & S. Connelly, Authors). U.S. Department of Commerce. <https://doi.org/10.6028/NIST.SP.800-207>
- Nemec Zlatolas, L., Welzer, T., & Lhotska, L. (2024). Data breaches in healthcare: Security mechanisms for attack mitigation. *Cluster Computing*, 27(7), 8639–8654. <https://doi.org/10.1007/s10586-024-04507-2>
- OWASP Foundation. (2024). OWASP Mobile Application Security Verification Standard (MASVS). <https://mas.owasp.org/MASVS/>
- Rezaee, R., Khashayar, M., Saeedinezhad, S., Nasiri, M., & Zare, S. (2023). Critical criteria and countermeasures for mobile health developers to ensure mobile health privacy and security: Mixed methods study. *JMIR mHealth and uHealth*, 11, Article e39055. <https://doi.org/10.2196/39055>
- Sakimura, N. (Ed.), Bradley, J., & Agarwal, N. (2015). RFC 7636: Proof key for code exchange by OAuth public clients. Internet Engineering Task Force. <https://doi.org/10.17487/RFC7636>
- Sasikumar, K., & Nagarajan, S. (2026). A hybrid machine learning and explainable AI framework for optimizing risk-based authentication. *PLOS ONE*, 21(5), Article e0349095. <https://doi.org/10.1371/journal.pone.0349095>
- Suleski, T., Ahmed, M., Yang, W., & Wang, E. (2023). A review of multi-factor authentication in the Internet of Healthcare Things. *Digital Health*, 9, 1–20. <https://doi.org/10.1177/20552076231177144>
- U.S. Department of Health and Human Services, Office for Civil Rights. (2022). Protecting the privacy and security of your health information when using your personal cell phone or tablet. <https://www.hhs.gov/hipaa/for-professionals/privacy/guidance/cell-phone-hipaa/index.html>
- Wang, Y., & Zhou, X. (2023). Biometrics-based mobile user authentication for the elderly: Accessibility, performance, and method design. *International Journal of Human–Computer Interaction*, 40(9). <https://doi.org/10.1080/10447318.2022.2154903>