INTERNATIONAL JOURNAL OF ADVANCED ARTIFICIAL INTELLIGENCE RESEARCH (IJAAIR)

eISSN: 3087-4270

Volume. 02, Issue11, pp. 23-31, November 2025

An Explainable, Context-Aware Zero-Trust Identity Architecture for Continuous Authentication in Hybrid Device Ecosystems

Dr. Alejandro Moreno
University of Barcelona

Article received: 01/11/2025, Article Accepted: 15/11/2025, Article Published: 29/11/2025

ABSTRACT

Background: The contemporary landscape of user authentication is evolving rapidly as mobile devices, cloud services, and agentic artificial intelligence converge. Traditional reliance on single-factor credentials and static, perimeter-based security models has proven inadequate for resisting sophisticated attacks and for preserving privacy and usability in ubiquitous computing environments (Jakobsson, 2009; Abowd et al., 2000). Contemporary work emphasizes context-aware authentication, continuous and implicit methods, and zero-trust principles, yet there remains a gap in integrating explainability, device integrity mechanisms, and enterprise device management constructs into a unified identity architecture that supports both human and machine (agentic) actors (Hayashi et al., 2013; Badal Bhushan, 2025).

Methods: This article presents a theoretically grounded design for an explainable zero-trust identity architecture that fuses context-aware continuous authentication techniques, device attestation and integrity (including operating-system level protections such as system integrity mechanisms and disk encryption), enterprise device provisioning and management, and privacy-aware explainable decisioning for authentication and access decisions. The methodology is a conceptual synthesis: we systematically analyze the reference corpus provided, extract design primitives and threat models, and then elaborate an architectural blueprint that maps primitives to operational components, authentication flows, and explanation-generation modules. The work adopts rigorous evaluative criteria (security, privacy, usability, scalability, and explainability) and applies them descriptively to anticipated deployments.

Results: The architecture integrates eight functional components—Context Sensing, Behavioural Profiling, Device Integrity Attestation, FIDO-style Public Key Authentication, Continuous Risk Engine, Explanation Generator, Enterprise Management Bridge, and Audit and Recovery Services—and specifies interfaces, data flows, and trust anchors. The design articulates how device features such as FileVault encryption (Apple, 2023a), System Integrity Protection (Apple, 2023b), and backup/restore considerations (Apple, 2023c) affect attestation and key-protection strategies. It further explains how message interception risks (Shah, Jeong & Doss, 2021) and second-factor device-mirroring threats motivate minimizing SMS usage and favoring device-bound cryptographic authenticator approaches (Shah & Kanhere, 2018).

Conclusion: By systematically combining context awareness, continuous implicit authentication, device attestation, enterprise management, and explainability, the proposed zero-trust identity architecture addresses many contemporary deficiencies in authentication ecosystems. The paper articulates implementation guidance, nuance on privacy trade-offs, counter-arguments, and a research agenda for empirical evaluation and standardization. The architecture aims to be extensible to both human users and machine agents, promoting resilient, transparent, and privacy-respecting authentication in hybrid modern IT environments (Hayashi et al., 2013; Badal Bhushan, 2025).

KEYWORDS

Zero-trust, continuous authentication, context-aware security, device attestation, explainability, enterprise device management, implicit authentication

INTRODUCTION

Security, identity, and privacy concerns have become central in an era where personal devices act as nodes in a larger socio-technical fabric that blends human behaviour, automated agents, cloud services, and managed enterprise resources. Historically, authentication research concentrated on knowledge factors (passwords) and, later, token and biometric factors (Bonneau et al., 2012). However, passwords have demonstrated persistent weaknesses in usability, memorability, and resistance to phishing and replay attacks (Jakobsson, 2009). The limitations of static credentials spurred research into second factors, hardware tokens, and public-key infrastructures, and progressively toward contextual and continuous authentication mechanisms which exploit device sensors, behavioural patterns, and environmental cues (Abowd et al., 2000; Hayashi et al., 2013).

Context-aware and continuous authentication approaches aim to reduce user friction while enhancing security by inferring identity from rich contextual inputs — e.g., location traces, Wi-Fi signatures, activity patterns — and by refreshing authentication state across sessions (Ashibani, Kauling & Mahmoud, 2019; Niinuma, Park & Jain, 2010). The literature demonstrates multiple paradigms: implicit authentication based on kinematic and behavioural signals (Jakobsson, 2009), geo-trace and n-gram modelling of mobility (Buthpitiya et al., 2011), and multimodal FIDO authenticators that incorporate context to adapt authentication strength (Kim et al., 2018). Yet the operationalization of these techniques in enterprise and consumer contexts must contend with device integrity, deployment management, explainability of automated decisions, and resilience to modern attack modalities such as message mirroring and SIM-swap-style interception (Shah, Jeong & Doss, 2021).

Zero-trust security paradigms emphasize that implicit trust in network perimeters is harmful; instead, every access decision should be authenticated, authorized, and continuously validated irrespective of network location (Stafford, 20XX). Within identity systems, zero-trust is realized by combining strong cryptographic binding of credentials to devices, continuous risk scoring, and micro-segmentation of access privileges. While zero-trust constructs offer conceptual clarity, practical deployment requires detailed specification for attestation of device integrity, secure key protection (e.g., full-disk encryption), enterprise provisioning workflows, and the capacity to explain automated denials or step-ups to users and auditors (Apple, 2023a; Apple, 2023d; Apple, 2023e).

This article responds to a confluence of gaps identified in the literature and practice. First, while context-aware methods are promising, few architectures unify them with device attestation mechanisms and enterprise provisioning systems in a manner that remains explainable and auditable (Hayashi et al., 2013; Benzekki, El Fergougui & ElAlaoui, 2018). Second, many continuous authentication systems emphasize novelty rather than the interplay between device OS security features (e.g., system integrity protection), cryptographic key management (e.g., disk encryption), and backup/restore semantics — all of which materially influence threat models and recovery strategies (Apple, 2023a; Apple, 2023b; Apple, 2023c). Third, there is an emerging need to design identity solutions that can accommodate not only human users but also agentic entities (AI agents) that require persistent, explainable, and auditable identity and privilege management (Badal Bhushan, 2025). Finally, there exists a demand for architectures that minimize privacy invasiveness while providing justifiable, comprehensible explanations for automated authentication decisions — a requirement that intersects ethical considerations for data minimization (Annabelle, 2017).

Accordingly, the central aim of this paper is to present a detailed, publication-ready architectural proposal: an explainable zero-trust identity architecture tailored for hybrid ecosystems, integrating context-aware continuous authentication, device integrity attestation, enterprise provisioning and management, and an explanation generation layer that supports transparency, accountability, and compliance. The architecture is developed through a rigorous synthesis of the provided references and an extensive theoretical elaboration of each component, its rationale, data flows, and implications. The proposed design is not presented as an empirical implementation but as a conceptually rigorous and actionable blueprint intended to guide future system design, empirical validation, standardization efforts, and policymaking. (Abowd et al., 2000; Hayashi et al., 2013; Shah & Kanhere, 2018; Badal Bhushan, 2025)

METHODOLOGY

This work follows a conceptual, analytic methodology appropriate to theoretical systems design. The methodological steps are described in detail to justify design decisions and to enable reproducibility of the reasoning process, even though no experimental system was implemented as part of this study.

1. Corpus Curation and Thematic Extraction: The starting point is the provided reference list. Each reference was read and its salient contributions, assumptions, threat models, and proposed techniques were extracted. For example, works on implicit and context-aware authentication (Jakobsson, 2009; Hayashi

et al., 2013; Alk. Kim et al., 2018) provided behavioural primitives and sensor-based features; device management and OS integrity documentation (Apple, 2023a; Apple, 2023b; Apple, 2023d; Apple, 2023e) provided specifics on how modern operating systems present attestation and key-protection facilities; applied threat analyses (Shah, Jeong & Doss, 2021) informed constraints on second factor design.

2. Design Primitive Identification: From the extracted themes, design primitives were enumerated. A design primitive is a reusable capability or constraint, such as: (a) local device attestation; (b) privacy-preserving context sensing; (c) behavioural profile lifecycle; (d) cryptographic key binding and escrow; (e) enterprise provisioning and revocation operations; (f) explainable risk scoring. Each primitive was defined in precise terms and tied to the supporting references. For instance, device attestation primitives leverage OS mechanisms like system integrity protection and disk encryption as cited (Apple, 2023a; Apple, 2023b).

3. Threat Model Assembly: A threat model was constructed by synthesizing attacker capabilities described in the literature: message mirroring and SMS interception (Shah, Jeong & Doss, 2021), device cloning and SIM attacks, malware that disables or spoofs sensor feeds, and adversaries capable of physical device access. For each threat, the system's capability to detect, mitigate, or tolerate the threat was explicated. The threat model guided component interfaces—e.g., requiring remote attestation of anti-rollback and integrity counters to defend against cloned images.

4. Functional Decomposition and Component Design: Using the primitives and the threat model, the architecture was decomposed into functional components. Each component—Context Sensing, Behavioural Profiler, Device Integrity Attestor, FIDO-style Authenticator, Continuous Risk Engine, Explanation Generator, Enterprise Management Bridge, Audit and Recovery Services—was specified with its responsibilities, inputs, outputs, required trust assumptions, and acceptable privacy leakage bounds.

5. Interface and Protocol Specification (Descriptive): The information flows between components were specified textually, including how keys are provisioned and salted, how attestation tokens are generated and validated, and how continuous signals feed the risk engine. The protocols were described without low-level code, but with enough detail to inform an implementer of essential message semantics: e.g., attestation tokens contain device model, firmware hash, signer certificate path, and nonce.

6. Evaluation Criteria and Qualitative Analysis: Given the lack of empirical data, the architecture was evaluated qualitatively against security, privacy, usability, scalability, and explainability criteria. For each criterion, expected strengths and weaknesses were discussed, with citations justifying why particular design choices improve or compromise specific attributes (Jakobsson, 2009; Hayashi et al., 2013; Kim et al., 2018).

7. Ethical and Legal Considerations: The design was analyzed in light of ethics and data minimization principles (Annabelle, 2017) and with an eye toward compliance with likely regulatory constraints for device telemetry and personal behavioural profiling. Trade-offs and mitigation approaches, such as local differential privacy or on-device aggregation, were described.

8. Research Agenda Formulation: Finally, gaps for empirical validation were identified—e.g., quantitative performance of behavioural models under adversarial noise (Anderson & McGrew, 2017) and user acceptance of explanation formats—outlining concrete experiments and metrics.

This methodological approach ensures that the design is tightly coupled to the literature while remaining explicit about the assumptions and the intended scope.

RESULTS

The core result is a richly detailed architectural blueprint: an explainable zero-trust identity architecture designed for hybrid human/agent environments. The description that follows enumerates the architecture’s components, their interactions, and the expected operational properties. Each subsection explains both functional behaviour and rationale, and references are provided to ground major claims.

Architecture Overview

The architecture comprises eight integrated functional components:

1. Context Sensing Layer — Collects device, environmental, and behavioural signals.

2. Behavioural Profiling Module — Builds and updates user or agent profiles and computes similarity scores to observed behaviours.

3. Device Integrity Attestation Service — Validates device state using hardware- and OS-provided attestation primitives.

4. Cryptographic Authenticator — Employs public-key, device-bound credentials (FIDO-style) and manages key lifecycle with secure enclaves and disk encryption awareness.

5. Continuous Risk Engine — Aggregates signals and computes a dynamic risk score that determines authentication posture (allow, step-up, deny).

6. Explanation Generation Module — Produces human-readable and auditable explanations for access decisions.

7. Enterprise Management Bridge — Integrates with enterprise device enrollment, provisioning, and lifecycle-management tools.

8. Audit and Recovery Services — Supports forensic trail generation, key recovery, and account recovery in the event of device loss.

The components cooperate within a zero-trust policy loop: every access request is locally authenticated; the device provides a fresh attestation token; contextual signals update the risk engine; the decision is made; and an explanation is generated and returned to the user and to audit logs.

Context Sensing Layer

The Context Sensing Layer is responsible for acquiring a curated set of signals that meaningfully contribute to continuous authentication while minimizing privacy exposure. Signals include: short-term GPS or coarse location, Wi-Fi environment signatures (SSID/BSSID patterns), recent device interactions (active applications, input cadence), physical sensors (accelerometer patterns for gait), network characteristics (cell tower patterns), and temporal rhythms (login times). The selection of signals is guided by prior work showing that multi-modal context improves authentication accuracy and robustness (Hayashi et al., 2013; Buthpitiya et al., 2011). The system employs a privacy-first collection strategy: whenever possible, raw signals are processed locally into features and only aggregated feature vectors or risk deltas leave the device, thereby limiting telemetry exposure (Annabelle, 2017).

Context sensing must also contend with adversarial manipulation. Attackers may attempt to spoof location or Wi-Fi signals (e.g., evil twin access points). Therefore, the sensing layer incorporates signal provenance metadata and cross-checks—e.g., GPS corroborated with Wi-Fi and cellular signatures—to calculate signal trust scores. The provenance metadata becomes an input to the Continuous Risk Engine, enabling contextual weighting and detection of likely spoofed inputs (Abowd et al., 2000; Benzekki et al., 2018).

Behavioural Profiling Module

The Behavioural Profiling Module constructs a behavioural fingerprint from temporally segmented features (e.g., keystroke dynamics, touch patterns, app usage sequences). Such fingerprints are represented as probabilistic models—e.g., n-gram models for geo-trace and activity sequences as demonstrated by Buthpitiya et al. (2011) and as behavioural token distributions for keystroke or touch dynamics (Gupta et al., 2012). The module maintains models locally, updates them with user consent and with privacy preserving defaults, and supports configurable decay to forget old behaviours. The behavioral models prioritize interpretability by producing human-comprehensible metadata: the top contributing features for a high or low similarity score are tracked so they can be surfaced in explanations (Niinuma, Park & Jain, 2010; Dandapat et al., 2015).

A critical design point is the continuous and adaptive nature of the model: rather than producing binary accept/deny outputs, the module emits similarity scores with confidence intervals and change-points that are interpretable by downstream decision logic. This allows for adaptive thresholds and step-up authentications. Prior research on implicit Authentication (Jakobsson, 2009) and continuous models in smart homes (Ashibani et al., 2019) supports the viability of this approach.

Device Integrity Attestation Service

Device attestation is the backbone of zero-trust for device-bound credentials. The Attestation Service verifies that the device is running an expected, untampered software stack, that anti-rollback protections are in place, and that cryptographic keys are bound to secure hardware where possible. Contemporary operating systems provide primitives that the attestation service can leverage. For instance, FileVault provides encryption of disk volumes and indicates whether the disk is encrypted and whether keys are properly protected (Apple, 2023a). System Integrity Protection (SIP) enforces kernel and system file protection that mitigates certain classes of instrumentation or hooking that attackers use to hide malware (Apple, 2023b).

Attestation tokens are generated by a local attestation agent and signed by a device-resident key whose private material is never exported (residing in a secure enclave or TPM). The token contains a freshness nonce provided by the verifying service to avoid replay and includes a compact description of measured boot components (firmware hash, kernel extension status), current OS version, disk encryption status, and a timestamp. The validator verifies the token signature, cross-checks the included measurements against a policy database, and computes a device integrity score. The literature on device attestation highlights the utility of these measurements for establishing device trust (Apple, 2023a; Apple, 2023b).

The service must treat backup and restore semantics carefully. Disk encryption and key management tie into recovery mechanisms; for example, Time Machine backups (Apple, 2023c) may contain key material or privileged data. The architecture prescribes that backups be evaluated in attestation (e.g., a device restored from an untrusted backup may necessitate additional verification) and that keys be re-provisioned and re-sealed post-restore. The enterprise provisioning bridge can enforce policies requiring re-attestation after restore operations (Apple, 2023c; Apple, 2023d).

Cryptographic Authenticator

The Cryptographic Authenticator embodies the recommended primary authentication modality: device-bound public-key credentials following FIDO principles (Kim et al., 2018). The private key is generated on device and, where hardware support exists, is sealed in a secure enclave or TPM that prevents key exfiltration. Authentication asserts possession of the private key plus optional user presence or user verification (biometric). Importantly, the architecture recommends minimizing reliance on SMS-based second factors because of vulnerabilities to message mirroring and interception illustrated by Shah, Jeong & Doss (2021). Instead, push-based cryptographic attestations or physical security keys should be favored as out-of-band attestations.

For agentic entities (software agents or AI actors), the authenticator issues machine-oriented keys that carry metadata about the agent's identity, allowed actions, and lifespan. Machine keys may be stored in dedicated hardware security modules where available, or in enterprise key stores with hardware root-of-trust. The design emphasizes lifecycle management: key rotation policies, revocation lists, and emergency key invalidation routines. Enterprise enrollments performed via Apple Business Manager or Apple Configurator provide a controlled path for key provisioning and device enrollment (Apple, 2023d; Apple, 2023e).

Continuous Risk Engine

At the core of the architecture is the Continuous Risk Engine (CRE), which amalgamates signals from the Context Sensing Layer, Behavioural Profiler, and Device Attestation Service to produce a dynamic risk score. The CRE employs probabilistic fusion—explicitly modelling measurement uncertainty—and produces an interpretable scalar risk score augmented with contributions per signal. This modular design allows tuning and isolation: e.g., if GPS is unreliable in an environment, the CRE lowers its weight and increases reliance on Wi-Fi and behaviour signatures.

The CRE produces discrete policy action recommendations: silent allow (no user involvement), step-up authentication (require re-auth or biometric), constrained access (grant only low-risk operations), or deny (block access and require managed recovery). The CRE also recognizes continuous risk trends: it supports hysteresis to avoid flapping (rapid oscillation between allow/deny) and incorporates decay windows so that short, anomalous sensor glitches do not trigger unnecessary lockouts. The literature on adaptive context-aware authentication and continuous decisioning supports such risk-based adaptive policies (Hayashi et al., 2013; Kim et al., 2018).

Threat-aware weighting ensures that signals known to be spoofable (e.g., SSID lists) are given lower absolute weight unless corroborated. The CRE is also designed to accept policy overrides from enterprise administrators for high-value resources, enabling role-based or context-specific constraints (Benzekki et al., 2018).

Explanation Generation Module

A defining feature of this architecture is the Explanation Generation Module (EGM). The EGM produces human-readable explanations for decisions produced by the CRE. Explanations are structured to be actionable, concise, and privacy-preserving. For example, an explanation for a step-up decision might read: "Access to Financial Dashboard requires re-authentication because the device attestation indicates an unverified OS image and recent network environment differs from typical patterns." The EGM leverages the CRE's per-signal contributions and the behavioural profiler's top features to produce such statements.

Explanations serve several purposes: they improve user comprehension and adoption, support compliance and auditing, and enable administrators and security teams to perform triage. The EGM is mindful of adversarial concerns: explanations should avoid leaking sensitive detail that could help an adversary (e.g., exact matching thresholds or full telemetry logs). The module therefore follows redaction principles—explanations reveal causes at a coarse granularity and can be configured to reveal more or less detail depending on user role and regulatory constraints (Annabelle, 2017).

Enterprise Management Bridge

This component integrates enterprise enrollment, provisioning, policy distribution, and revocation. It interfaces with device enrollment services such as Apple Business Manager and Apple Configurator, and with corporate identity providers and policy engines (Apple, 2023d; Apple, 2023e). The Bridge enforces secure bootstrapping, provides authenticated channels for policy dissemination, and supports bulk operations such as decommissioning and emergency revocation.

The Bridge also manages exception workflows: it supports account recovery procedures—e.g., when a device is lost—by orchestrating re-provisioning, remote wipe, or supervised recovery. It must be designed to prevent an attacker from abusing recovery channels; therefore, policies such as multi-party approval, out-of-band verification, and temporal constraints are recommended. The research literature emphasizes the necessity of robust lifecycle management to avoid identity orphaning and to limit attack surfaces during enrollment (Shah & Kanhere, 2018).

Audit and Recovery Services

Comprehensive audit trails enable retrospective analysis and compliance. The Audit Service logs anonymized decision trees, risk factors, attestation tokens, and resolution outcomes. To preserve privacy, logs are pseudonymized and stored under strict access control. For incident response, the Recovery Service provides tools to revoke keys, re-provision devices, and restore user access. The architecture recommends cryptographic key escrow policies only in situations where legal or business requirements demand them and with transparent governance, as escrow increases exposure to insider risk and legal complexity (Annabelle, 2017).

Data Flows and Example Authentication Sequence

A representative authentication flow is as follows:

1. User attempts access to a resource. The request triggers local auth:

2. The Cryptographic Authenticator asserts possession of a device-bound key and optionally a biometric check; it then produces a signed assertion.

3. Simultaneously, the Attestation Agent creates a fresh attestation token, signed by the device key, including device measurements and a nonce from the verifier.

4. The Context Sensing Layer processes recent signals into feature vectors and computes per-feature provenance metadata.

5. The Behavioural Profiling Module computes similarity to baseline and emits a score with confidence.

6. The CRE receives assertions, attestation tokens, and context features; it computes an overall risk score and chooses an action (e.g., allow with low privilege).

7. The EGM generates an explanation based on the top risk contributors and returns a concise human-readable explanation and the action code.

8. The Audit Service logs the inputs, the decision, and the explanation.

Each message in this flow is designed to be minimal and privacy-conscious: raw sensor data remains local; only derived features and signed attestations leave the device. The architecture's emphasis on freshness tokens and signed attestation guards against replay and many kernel-level attacks (Apple, 2023b).

Qualitative Evaluation

The architecture's strengths and trade-offs across five criteria are summarized here.

- Security: By binding credentials to device hardware, leveraging attestation, and using continuous risk scoring, the design reduces the attack surface from credential theft and spoofing. It addresses SMS-based vulnerabilities by preferring cryptographic attestation (Shah, Jeong & Doss, 2021). However, hardware enclaves and attestation mechanisms themselves are trust anchors; their compromise would be catastrophic, highlighting the need for defense in depth and multi-path recovery mechanisms.

- Privacy: Local feature extraction and minimal telemetry reduce exposed personal data. Yet behavioural profiling inherently involves sensitive signals; the architecture mitigates risk via local models, configurable retention, and explainable redaction policies (Annabelle, 2017).

- Usability: Continuous and implicit authentication reduces visible friction for users in normal conditions while enabling step-up only when necessary. Explainability supports user comprehension. The trade-off is that users may experience occasional step-ups due to sensor noise or mobility, which must be tuned carefully to avoid frustration (Hayashi et al., 2013).

- Scalability: The architecture is designed to offload heavy modeling to devices and to use compact attestation formats; enterprise components are horizontally scalable. However, per-device model support increases management complexity at scale.

- Explainability and Compliance: The EGM's structured explanations assist compliance and incident triage. Generating explanations that satisfy legal scrutiny (e.g., GDPR's right to explanation) is feasible because decisions are traceable and the top contributing signals are recorded (Annabelle, 2017).

These qualitative assessments reflect the literature consensus that multi-modal, context-aware architectures can balance security and usability but require careful governance and engineering attention to privacy and device lifecycle (Hayashi et al., 2013; Kim et al., 2018).

DISCUSSION

This architecture aims to reconcile competing objectives: strong, continuous assurance; minimal user friction; enterprise manageability; and respect for privacy and explainability. The discussion below examines design rationales, potential limitations, counter-arguments, and avenues for future work.

Rationales and Theoretical Implications

1. Why Device-Bound Keys and Attestation? Device-bound keys transform the authentication problem from knowledge possession into hardware possession plus local user verification. This reduces credential replay attacks and phishing susceptibility because private keys never leave devices (Kim et al., 2018). Attestation establishes the operational integrity of the device environment, which is essential to trust the local sensors and models. The theory of trusted computing and measured boot justifies this approach: a system that can prove its state cryptographically enables remote policy enforcement with lower reliance on behavioral inference alone (Apple, 2023b).

2. Why Continuous, Context-Aware Models? Static authentication provides a snapshot guarantee. Continuous models provide temporal depth and adaptivity: they can detect session hijacking and anomalies in long sessions or during agentic processes where credentials might be delegated. Theoretical work on context awareness suggests that fusing multiple weak signals often yields stronger identification than any single signal (Abowd et al., 2000).

3. Why Explainability? Explanations address two domains: human factors (user trust and comprehension) and institutional accountability (audits). Cognitive psychology indicates that users respond better and adhere more to systems they can understand; transparency reduces confusion from unexpected denials (Annabelle, 2017). Also, regulatory regimes increasingly expect traceable automated decisions.

4. Why minimize SMS? SMS-based second factors have well-documented weaknesses (Shah, Jeong & Doss, 2021). Message mirroring and SIM swap attacks reduce their security value. The architecture replaces SMS with cryptographic, device-bound attestations that are resistant to network interception.

Limitations and Counter-Arguments

1. Hardware Trust Anchor Dependency: The architecture relies on hardware roots-of-trust. Critics may argue that these anchors create single points of failure and raise supply chain trust issues. While this is true, the architecture recommends layered defenses: remote key revocation, multiple attestation paths, and enterprise oversight to mitigate catastrophic compromise.

2. Privacy Concerns of Behavioural Profiling: Behavioural profiling can be perceived as intrusive. The architecture addresses this through local model stewardship, retention minimization, and explanation redaction. However, even with these mitigations, risk remains that sophisticated models reconstruct sensitive attributes. A countermeasure is to add differential privacy noise to shared features and to provide strict governance around which features can be externally transmitted.

3. Usability in Unpredictable Environments: In environments with high mobility or intermittent connectivity, the CRE may over-trigger step-ups. The architecture mitigates this with hysteresis and policy tuning but recognizes that imperfect sensors will produce false positives. A design choice is to permit bounded low-privilege access in high-uncertainty environments rather than blocking access outright.

4. Agentic Identity Complexity: Extending identity to AI agents introduces complex policy questions: How are agent motives and intents reconciled with authorization? The architecture suggests metadata tagging and explicit scopes for agents, but a richer policy language is required for expressive delegation semantics. This remains an open research area (Badal Bhushan, 2025).

Future Research and Empirical Validation

The proposed architecture opens multiple empirical research directions:

- Adversarial Robustness Studies: Evaluate the CRE and Behavioural Profiler under adversarial scenarios—spoofed signals, sensor manipulation, and poisoning of behavioral models. Methodologies from adversarial machine learning should be applied to assess resilience (Anderson & McGrew, 2017).

- User Study on Explanation Efficacy: Conduct user experiments to determine which explanation styles optimize comprehension, reduce frustration, and facilitate corrective action. Usability metrics and qualitative feedback should guide EGM design.

- Comparative Deployment Case Studies: Implement pilot deployments in varied contexts (enterprise-managed devices, consumer BYOD, industrial IoT) to measure false positive/negative rates, administrative overhead, and recovery workflows.

- Policy and Governance Models for Key Escrow: Investigate legal and operational frameworks for key escrow and recovery that balance business continuity with risk minimization. Comparative legal analysis across jurisdictions is warranted.

- Standardization of Attestation Formats: Work with standards bodies to formalize the attestation token formats and policy expression languages to achieve interoperability across platforms.

Operational Considerations

When adopting this architecture, practitioners must carefully design operational policies:

- Enrollment Policies: Enroll devices using strict identity proofing and multi-factor vetting. Enrollment flows should ensure administrative oversight and tamper-resistant provisioning (Apple, 2023d; Apple, 2023e).

- Recovery Policies: Define robust, multi-party recovery procedures to prevent social engineering exploitation of recovery channels. Escalation steps must include manual verification for high-value accounts.

- Retention and Deletion: For privacy, define minimal retention for behavioural and attestation metadata. Implement programmable deletion windows and support user requests for data removal in line with applicable regulations.

- Audit Logging: Ensure logs are protected and that the EGM maps to forensic artifacts suitable for legal proceedings when required. Logs should be tamper-evident and access controlled.

CONCLUSION

This paper has presented an explainable zero-trust identity architecture that integrates context-aware continuous authentication, device integrity attestation, enterprise provisioning, and explainability. Anchored in the provided literature, the design synthesizes principles from implicit authentication, measured device attestation, FIDO-style cryptographic binding, and privacy-conscious data practices to propose a comprehensive blueprint suitable for hybrid human and agentic ecosystems.

The architecture addresses modern attack vectors — such as SMS interception and device spoofing — and advances the state of the art by coupling continuous risk scoring with an explanation generator that supports comprehension, compliance, and incident triage. Recognizing the trade-offs involved, the design prescribes mitigations: local model stewardship, differential sharing, hardware enclaves, and robust lifecycle management.

Future work requires empirical validation through pilots, adversarial testing, and user studies on explanation effectiveness. Moreover, standardization efforts for attestation formats and policy languages are essential to achieving interoperability and broad adoption. The proposed architecture is intended to guide designers, researchers, and practitioners in building identity systems that are secure, usable, auditable, and respectful of individual privacy in an era of increasingly autonomous and interconnected devices.

REFERENCES

1. Annabelle, L. (2017). Ethics Defined. Retrieved from https://medium.com/the-ethical-world/ethics-defined-33a1a6cc3064

2. Apple. (2023a). How does FileVault encryption work on a Mac? Retrieved from https://support.apple.com/guide/mac-help/how-does-filevault-encryption-work-on-a-mac-flvlt001/mac

3. Apple. (2023b). About System Integrity Protection on your Mac. Retrieved from https://support.apple.com/en-us/HT204899

4. Apple. (2023c). Back up your Mac with Time Machine. Retrieved from https://support.apple.com/en-us/HT201250

5. Apple. (2023d). Intro to Apple Business Manager. Retrieved from https://support.apple.com/engb/guide/apple-business-manager/axm6a88f692e/1/web/1

6. Apple. (2023e). Intro to Apple Configurator. Retrieved from https://support.apple.com/engb/guide/apple-configurator-mac/cadf1802aed/mac

7. Shah, S. W. A., Jeong, J. J., & Doss, R. (2021). How Hackers Can Use Message Mirroring Apps to See All Your SMS texts—and Bypass 2FA Security. Retrieved from https://theconversation.com/how-hackerscan-use-message-mirroring-apps-to-see-all-your-sms-texts-and-bypass-2fa-security-165817

8. Shah, S. W., & Kanhere, S. S. (2018). Wi-sign: Device-free second factor user authentication. In Proceedings of the 15th EAI International Conference on Mobile Ubiquitous Systems, Comput., Netw. Services, New York, NY, USA, 2018, pp. 135–144.

9. Shah, S. W., & Kanhere, S. S. (2018). Wi-access: Second factor user authentication leveraging WiFi signals. In Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), March 2018, pp. 330–335.

10. Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M., & Steggles, P. (2000). Towards a better understanding of context and context-awareness. In Proceedings of the CHI Workshop What, Who, When, How Context-Awareness, 2000, pp. 304–307.

11. Hayashi, E., Das, S., Amini, S., Hong, J., & Oakley, I. (2013). CASA: Context-aware scalable authentication. In Proceedings of the 9th Symposium on Usable Privacy and Security (SOUPS), 2013, pp. 1–10.

12. Buthpitiya, S., Zhang, Y., Dey, A. K., & Griss, M. (2011). n-gram geo-trace modeling. In Proceedings

of the 9th International Conference on Pervasive Computing, 2011, pp. 97–114.

13. Badal Bhushan. (2025). An Explainable Zero Trust Identity Framework for LLMs, AI Agents, and Agentic AI Systems. International Journal of Computer Applications, 187(46), 42–52. DOI=10.5120/ijca2025925777

14. Jakobsson, M. (2009). Implicit authentication for mobile devices. In Proceedings of the 4th USENIX Workshop on Hot Topics in Security, 2009, pp. 25–27.

15. Benzekki, K., El Fergougui, A., & ElAlaoui, A. E. B. (2018). A context-aware authentication system for mobile cloud computing. Procedia Computer Science, 127, 379–387.

16. Kim, S. H., Choi, D., Kim, S. H., Cho, S., & Lim, K. S. (2018). Context-aware multimodal FIDO authenticator for sustainable IT services. Sustainability, 10(5), 1656.

17. Ashibani, Y., Kauling, D., & Mahmoud, Q. (2019). Design and implementation of a contextual-based continuous authentication framework for smart homes. Applied System Innovation, 2(1), 4.

18. Olejnik, K., Dacosta, I., Machado, J. S., Huguenin, K., Khan, M. E., & Hubaux, J.-P. (2017). SmarPer: Context-aware and automatic runtime permissions for mobile devices. In Proceedings of the IEEE Symposium on Security and Privacy (SP), May 2017, pp. 1058–1076.

19. Gupta, P., Wee, T. K., Ramasubbu, N., Lo, D., Gao, D., & Balan, R. K. (2012). HuMan: Creating memorable fingerprints of mobile users. In Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops, March 2012, pp. 479–482.

20. Dandapat, S. K., Pradhan, S., Mitra, B., Choudhury, R. R., & Ganguly, N. (2015). ActivPass: Your daily activity is your password. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI), 2015, pp. 2325–2334.

21. Niinuma, K., Park, U., & Jain, A. K. (2010). Soft biometric traits for continuous user authentication. IEEE Transactions on Information Forensics and Security, 5(4), 771–780.

22. Anderson, B., & McGrew, D. (2017). Machine learning for encrypted malware traffic classification: Accounting for noisy labels and non-stationarity. Proceedings of the 23rd ACM SIGKDD.