engineering proceedings

MDPI

Proceeding Paper

A Secure Facial-Biometric Authentication Microservice with Liveness Detection and Automated Security Testing †

Geovanny Cudco [1,2,*], Mayra Álvarez-Jiménez [1,2,*], Héctor Morales [3] and Franklin Flores-Ayala [4]

[1] Departamento de Ciencias de la Computación (DCCO), Universidad de las Fuerzas Armadas ESPE, Av. General Rumiñahui S/N, Sangolquí 171103, Ecuador
[2] Grupo de Investigación en Modelos de Producción de Software (GrIMPSoft), Universidad de las Fuerzas Armadas ESPE, Sangolquí 171103, Ecuador
[3] Escuela Superior de Ingeniería y Tecnología (ESIT), Universidad Internacional de La Rioja UNIR, Av. de la Paz 137, 26006 Logroño, Spain; hectorgustavo.m6861@comunidadunir.net
[4] Dirección de Desarrollo Tecnológico e Informático (DDTI), Universidad Técnica del Norte UTN, Av. 17 de Julio 5-21, Ibarra 100150, Ecuador; fsflores@utn.edu.ec
* Correspondence: agcudco@espe.edu.ec (G.C.); mialvarez2@espe.edu.ec (M.Á.-J.)
† Presented at the Sixth International Conference on Information Systems and Software Technologies (ICI2ST 2026), Quito, Ecuador, 23–25 September 2026.

Abstract

Microservice-based web applications require authentication mechanisms that mitigate risks associated with reused credentials, phishing and session hijacking without compromising usability. This study presents the design, implementation and controlled evaluation of a reusable facial-biometric authentication microservice that combines credential-based login, facial verification and a blink-based first-level liveness check. The service exposes REST endpoints for enrollment and verification, issues JSON Web Tokens and records audit events. The main contribution is the coherent integration of server-side biometric decision-making, STRIDE-based threat modeling and automated security verification within a reusable authentication microservice. The solution uses 128-dimensional facial descriptors and a predefined distance threshold, while following a data-minimization strategy based on derived biometric templates rather than persistent authentication-session images. Security controls include rate limiting, input validation, role-based access control, token protection and automated verification through unit, static, dynamic and penetration-oriented API tests. The evaluation separates bona fide verification, zero-effort impostor comparisons and presentation attack detection, reporting FAR/FRR, ROC, EER and APCER/BPCER outcomes for their data scopes. The microservice achieved 87.4% code coverage and completed 64 security tests with no successful exploit observed in the executed test set. Response times remained within predefined web-authentication targets under controlled local conditions. Overall, the results provide preliminary but substantive engineering evidence of the technical feasibility of the proposed architecture, rather than a complete security proof or production-readiness validation. Broader validation in real deployment environments and against more complex presentation attacks remains necessary.

Keywords: biometric authentication; microservices; facial recognition; liveness detection; STRIDE; DevSecOps

Check for updates

Academic Editor: Leonardo Valdivieso

Published: 18 September 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Authentication remains one of the most exposed functions in web applications. Passwords and token-based sessions continue to be widely used, but they remain vulnerable to

reused credentials, phishing, credential stuffing, session hijacking and weak recovery flows. These risks are amplified in microservice-based systems, where authentication is consumed by several independent services and must remain both secure and usable. Recent authentication schemes therefore combine conventional credentials with biometric verification, artificial intelligence and security-by-design practices.

Facial biometrics is attractive for web authentication because it can operate with commodity cameras and can be integrated into user-facing applications with limited additional hardware [1]. However, facial authentication cannot be reduced to face matching. Printed photographs, replayed videos, screen-based presentation attacks and more sophisticated spoofing artifacts can compromise systems that accept a detected face as evidence of identity. Liveness detection is therefore a necessary part of biometric authentication pipelines [2,3]. At the same time, service-oriented biometric proposals have explored REST APIs, SDKs and reusable components for facial verification, emphasizing interoperability and deployment-oriented integration [4,5]. These efforts are valuable, but many remain focused on recognition performance, a specific domain, or a particular model, with less emphasis on how biometric decisions are connected to persistence, auditability, access control, threat modeling and executable security evidence.

This study addresses that integration gap. The study's main contribution is the design and evaluation of a reusable facial-biometric authentication microservice for web applications that unifies biometric verification, presentation attack detection, STRIDE-based threat modeling and automated security verification within a reusable microservice artifact. The microservice combines credential-based login, server-side facial verification, blink-based first-level liveness checking, JWT-based session handling, audit logging, STRIDE-derived security controls and automated security testing. Its evaluation separates biometric verification, presentation attack detection (PAD) and software-security assurance, allowing the artifact to be assessed through FAR/FRR, ROC, DET, EER, APCER/BPCER/ACER, coverage, penetration-oriented tests and response-time measurements. The work therefore focuses on architectural integration and preliminary engineering validation of the implemented artifact under controlled conditions, while broader deployment-oriented validation in real operational environments remains a future research direction.

The main contributions are: (i) a reusable authentication microservice that separates client-side capture from server-side biometric decision-making; (ii) an integration model linking 128-dimensional facial descriptors, distance thresholds, first-level liveness checks, JWT access and audit events; (iii) a STRIDE-to-control-to-test traceability model for security assurance; and (iv) an evaluation protocol covering biometric verification, PAD, automated security testing and execution performance under controlled conditions. Section 2 reviews related work. Section 3 presents the architecture, threat model and evaluation protocol. Section 4 reports the results. Section 5 discusses implications and limitations. Section 6 presents conclusions and research directions.

2. Related Work

Facial biometric authentication in online settings has evolved along three complementary lines: deep learning-based facial verification, liveness detection against presentation attacks and software-security assurance for deployable authentication systems. Web-based facial recognition systems have reported practical benefits for accessibility and identity verification [1], while healthcare-kiosk and domain-specific systems have compared CNN architectures for liveness detection and confirmed the relevance of anti-spoofing controls [2,6]. Other proposals have used FaceNet-based embeddings with blockchain-supported traceability [7], or REST backends for mobile and web face verification [4]. Lightweight anti-spoofing

methods have also been proposed for 2D and 3D presentation attacks under constrained deployment conditions [3,8,9].

Service-oriented biometric authentication has gained attention through Face Recognition-as-a-Service and reusable SDK-based approaches [5]. Nevertheless, a deployable biometric service requires more than a recognition model: it must define trust boundaries, template handling, thresholds, session management, authorization, logging and security verification. Threat modeling and penetration testing have been used to connect assets and attack vectors with OWASP-oriented controls [10]; continuous authentication research has also analyzed impersonation, man-in-the-middle and denial-of-service risks under adversarial assumptions [11]. Secure development frameworks and DevSecOps practices emphasize that assurance must be built into the lifecycle through testing, review, automated verification and robustness-oriented test data [12–16]. Privacy-preserving and cancellable biometric templates further reinforce the need to treat biometric data as sensitive assets [17–19].

Table 1 positions representative work against the dimensions used in this study. The comparison shows that previous studies contribute important pieces, but only partially integrate biometric verification, liveness detection, reusable microservice deployment, threat modeling, automated testing and biometric/PAD metrics within the same artifact.

Table 1. Positioning of related work with respect to biometric functionality, deployment, security assurance and evaluation scope.

Paper	Main Focus	Deployment	Security/PAD Scope	Evaluation Emphasis
Paul et al. [1] and Ryando et al. [2]	Facial recognition and liveness in applied systems	Domain-specific web/kiosk systems	Liveness or recognition, limited security traceability	Accuracy and liveness behavior
Hsieh [7] and Douklias et al. [4]	Embedding-based access control and REST-based face verification	Blockchain or REST backend	Traceability/capture standardization, not STRIDE-to-test	Verification and access-control integration
Shinde et al. [3] and Hofbauer et al. [9]	Presentation attack detection	Anti-spoofing model/protocol	PAD-focused, not full web-authentication assurance	2D/3D or usability/PAD behavior
Kira et al. [5]	Face Recognition-as-a-Service	SDK/service-based	Passive and active liveness with system-integrity focus	Service-oriented biometric integration
Utama and Nurhadi [10]; Zeeshan et al. [11]	Security assessment and adversarial authentication analysis	System/security framework	Threat modeling, PTES/OWASP, or adversarial protocol analysis	Risk, authenticity, secrecy and threshold behavior
This work	Facial-biometric web authentication microservice	Reusable REST-based component	Blink liveness, STRIDE-to-control-to-test traceability, automated testing	FAR/FRR, ROC, DET, EER, APCER/BPCER, security tests and response time

3. Methodology

This study follows an applied design-and-evaluation methodology. The objective was to build a reusable facial-biometric authentication microservice and produce verifiable evidence about biometric behavior, PAD behavior, security controls and execution performance. The work was organized under a Secure Software Development Lifecycle perspective, where security decisions are considered from requirements and design through

implementation and testing [15,20]. The resulting artifact is a reusable server-side authentication microservice that integrates facial verification, blink-based first-level liveness detection, STRIDE-informed security controls and automated security testing within a single web-integrable component evaluated under controlled engineering conditions.

3.1. Microservice Architecture and Authentication Workflow

The microservice was designed to preserve a clear trust boundary between client-side capture and server-side authentication. The frontend, implemented with Vite/React and TypeScript, captures frames through WebRTC and guides the user during enrollment and verification. The FastAPI backend performs request validation, credential handling, biometric processing, authorization checks, JWT issuance and audit logging, following service-separation principles commonly used in modern microservice development [21]. The persistence layer stores user metadata, authentication configuration, audit-relevant information and the biometric representation required for verification. The implementation uses Argon2 for password hashing and signed JWTs for protected routes. Figure 1 presents the architecture used to separate the frontend, backend, biometric pipeline and persistence layer.

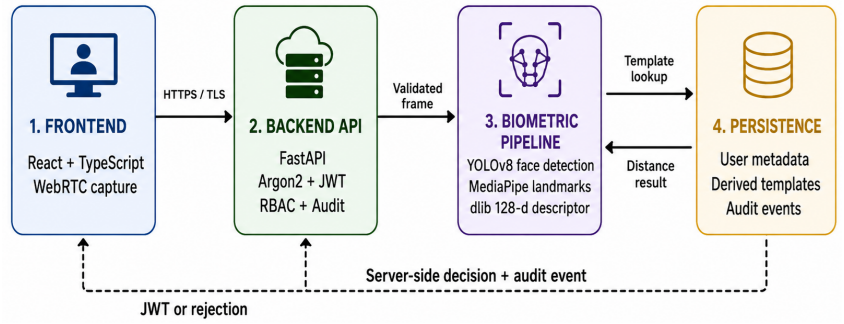


Figure 1. Architecture of the facial-biometric authentication microservice.

The API was structured around four core functions: enrollment, verification, protected-resource access, and audit-relevant events. Enrollment creates or updates the facial template linked to an authenticated identity. Verification, in turn, receives a capture sample and issues an authentication outcome only after completing the biometric cascade. This separation keeps user-interface concerns outside the final decision logic and enables external applications to use the service without embedding biometric code in each client.

Table 2 summarizes the main implementation parameters and design decisions that define the operating conditions of the authentication workflow and support the interpretation of the reported biometric, PAD and security results.

The biometric workflow follows a cascade of three decisions. First, YOLOv8 localizes the facial region of interest with low latency [2,22]. Second, MediaPipe extracts facial landmarks to support blink-based liveness validation through the Eye Aspect Ratio, a first-level PAD mechanism also discussed in recent biometric and liveness studies [4,7]. Third, after liveness validation, the face-recognition library based on dlib generates a 128-dimensional facial descriptor. The descriptor extracted from the authentication attempt is compared with the enrolled descriptor using Euclidean distance. A lower distance indicates greater similarity. The operating threshold was set to $d \leq 0.55$ to prioritize a lower false-acceptance tendency. A session token is issued only when the credential check, liveness check and descriptor comparison are all successful. This design avoids treating mere face presence as proof of identity.

Table 2. Implementation parameters and design decisions used in the authentication microservice.

Component	Configuration	Rationale
Credential protection	Argon2 password hashing	Avoids plaintext credential storage and provides memory-hard password verification.
Session handling	Signed JWT with expiration	Enables stateless access to protected routes while preserving token validation on the backend.
Face descriptor	128-dimensional dlib-based embedding	Provides a compact representation suitable for distance-based verification.
Matching rule	Euclidean distance, operating threshold $d \leq 0.55$	Prioritizes a lower false-acceptance tendency for authentication.
Liveness cue	Blink-based EAR analysis from facial landmarks	Adds a first-level PAD gate before descriptor comparison.
Decision location	Backend-only biometric decision	Prevents client-side capture code from granting authentication locally.
Audit scope	Enrollment, verification outcomes, token issuance, rejected attempts	Supports traceability without storing raw frame payloads in logs.

Figure 2 illustrates the authentication sequence, including client-side capture, backend validation, biometric processing, descriptor comparison and access-token issuance.

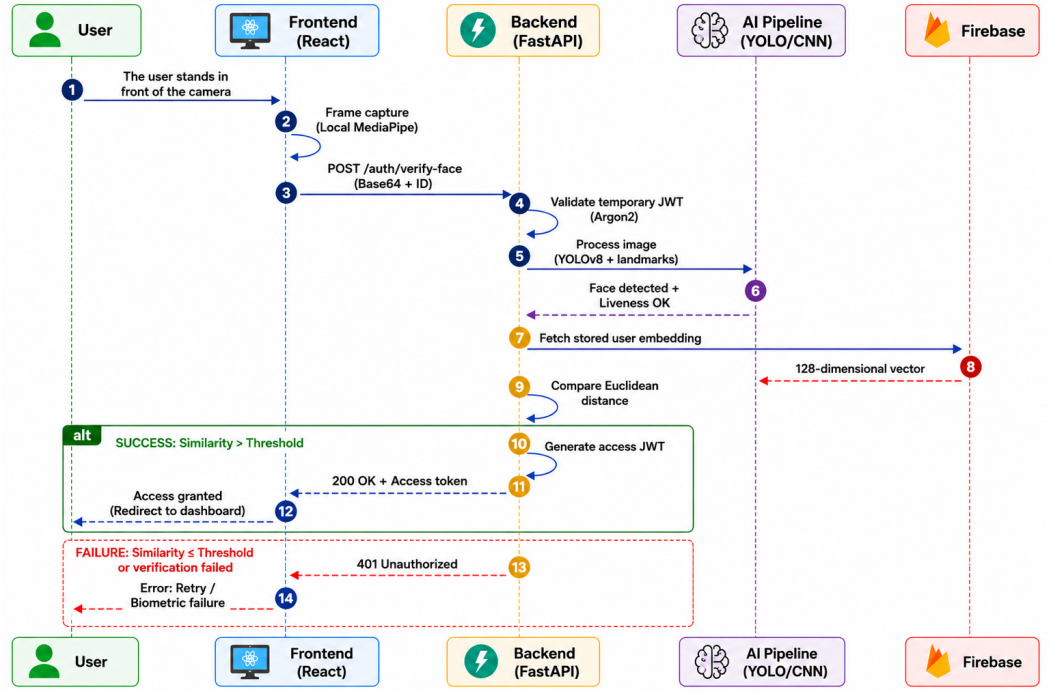


Figure 2. Sequence of the facial-biometric authentication process. Arrows indicate the authentication flow, and numbers identify the main processing stages.

For data handling, raw frames were treated as transient inputs for face detection, landmark extraction, liveness validation and descriptor generation. Persistent biometric storage was restricted to derived templates required for later verification, while audit logs excluded raw image payloads. This aligns with privacy-by-design principles and with the treatment of biometric data as sensitive information [17,23]. Future deployments may further strengthen template protection through application-level encryption, cancellable templates, or privacy-preserving comparison schemes [18,19].

All captures used in the controlled evaluation were obtained from participants who were informed about the purpose of the authentication tests. The captures were processed only for controlled system evaluation. No raw facial images were retained after descriptor extraction, no identifiable biometric samples are presented in this article and all reported results are aggregated.

The authentication flow also records controlled failure states. If no face is detected, the attempt is rejected before liveness analysis. If liveness fails, descriptor comparison is not used to grant access. If descriptor distance exceeds the threshold, the attempt is treated as a biometric mismatch even when liveness is valid. This separation supports later analysis of whether failures originate from capture quality, liveness validation, identity matching, token validation, or authorization.

3.2. Threat Modeling and Security Controls

Threat modeling was performed with STRIDE to identify risks before implementation and to connect threats with concrete controls and verification evidence. The threat model considered the web client, backend API, biometric processing module, persistence layer, JWT-based session handling and administrative routes. The prioritized threats were spoofing, tampering, repudiation, information disclosure, denial of service and elevation of privilege. For spoofing, the main concern was that an attacker could authenticate using a non-mated face or a presentation artifact. For tampering, the risk involved manipulation of biometric payloads, request parameters, or JWT claims. Repudiation was associated with insufficient traceability of enrollment and verification events. Information disclosure concerned the exposure of biometric material, credentials, tokens, or personal data, while denial of service involved excessive use of computationally expensive endpoints. For elevation of privilege, the concern was unauthorized access to protected biometric or administrative routes. Table 3 summarizes the resulting threat-to-control-to-test traceability.

Table 3. Threat-to-control-to-test traceability for the biometric authentication microservice.

STRIDE Threat	Risk in the Microservice	Control Applied	Verification Evidence
Spoofing	Non-mated user, photo, replay, screen-based presentation	Face detection, blink liveness, descriptor matching, server-side decision	Genuine/impostor comparisons, PAD protocol, FAR/FRR, APCER/BPCER
Tampering	Modified body, altered token, manipulated biometric payload	Input validation, signed JWTs, server-side claim checks	Malformed-payload, JWT tampering and API validation tests
Repudiation	Lack of traceability for authentication events	Audit logging for enrollment, failures, verification and token issuance	Backend integration tests and audit-event inspection
Information disclosure	Exposure of images, descriptors, tokens, or personal data	Data minimization, protected templates, authenticated access, no raw frames in logs	Persistence/logging review and access-control tests
Denial of service	Excessive or malformed requests against costly endpoints	Rate limiting, payload validation, boundary checks	Burst-request, invalid Base64 and boundary-condition tests
Elevation of privilege	Unauthorized administrative or protected biometric operations	RBAC, JWT claim validation, protected-route checks	Authorization-bypass and role-based route tests

This matrix was used to prevent the security discussion from remaining at the level of general recommendations. Each prioritized threat had to be associated with an implemented control and at least one form of evidence, whether executable tests, audit inspection, persistence review, or biometric/PAD results. The security strategy was aligned with S-

SDLC and DevSecOps principles, where controls are verified through repeatable tests rather than only described in design documentation [13,14,16].

3.3. Evaluation Protocol

The evaluation protocol was designed to address four aspects: biometric verification, PAD, automated security verification and performance. Table 4 introduces the protocol components before the results are reported.

Table 4. Evaluation protocol for biometric, PAD, security and performance assessment.

Block	Procedure	Data Scope/Denominator	Reported Evidence
Biometric verification	50 enrolled identities, 5 bona fide samples per identity; genuine comparisons against own template and zero-effort impostor comparisons against the remaining identities.	250 genuine comparisons and 12,250 zero-effort impostor comparisons	Distance distribution, FAR/FRR at $d \leq 0.55$, ROC, DET, EER, confidence intervals
Presentation attack detection	Bona fide live attempts and attack presentations including printed photographs, high-resolution prints, mobile/tablet replay, screen-based deepfake replay and low-cost 3D masks.	250 bona fide live presentations and 900 attack presentations, with 150 attacks per modality	APCER, BPCER, ACER, worst-case modality
Security verification	64 dynamic and penetration-oriented tests covering malformed payloads, JWT tampering, RBAC bypass, rate limiting, XSS, CSRF-style requests, injection/API bypass and boundary cases.	64 executed security tests, independent from the biometric and PAD subsets	Passed/failed tests, observed defensive behavior
Performance	CPU-based local execution on a Windows 11 Pro laptop with Intel Core i7-1355U and 32 GB RAM	Local CPU timing measurements; not included in biometric or PAD denominators	Average time for login, recognition, model initialization, descriptor validation and API response

To avoid mixing metric scopes, biometric verification, PAD, security testing and performance measurements were treated as separate evaluation subsets. FAR, FRR, ROC, DET and EER were computed only from the genuine and zero-effort impostor comparisons. APCER, BPCER and ACER were computed only from the PAD subset. Security-test outcomes and performance measurements were not included in the biometric or PAD denominators.

For biometric verification, each comparison recorded the Euclidean distance d between the authentication descriptor and the enrolled descriptor. Because lower distances indicate stronger similarity, ROC analysis used the transformed score $s = -d$. FAR and FRR were computed at the operating threshold, while threshold sweeping was used to estimate ROC behavior, DET behavior and the Equal Error Rate. The metrics were calculated in Equation (1).

$$FAR = \frac{FA}{FA + TR}, \qquad FRR = \frac{FR}{TA + FR}, \tag{1}$$

where FA denotes false acceptances, TR true rejections, FR false rejections and TA true acceptances. These metrics are standard in biometric and multi-factor authentication evaluation because they express the trade-off between security and usability at a selected operating threshold [24,25]. For PAD, APCER measured the proportion of attack presentations incorrectly accepted, BPCER measured the proportion of bona fide presentations incorrectly rejected and ACER was calculated in Equation (2).

$$ACER = \frac{APCER + BPCER}{2}. \tag{2}$$

Blink-based liveness was treated as a first-level PAD control rather than as a complete anti-spoofing solution. The protocol therefore reports PAD results by modality instead of using a single aggregate claim about spoofing resistance.

The security tests were executed against the running service rather than only against isolated functions. Malformed payload tests included invalid Base64 strings, missing fields, unexpected content types, long strings and Unicode input. Token tests modified signatures, expiration values and claims. Authorization tests attempted to access protected routes with insufficient privileges, while rate-limiting tests issued repeated login and biometric requests. Injection-oriented tests used SQL-like strings, path manipulation, and unexpected parameters. These cases were selected because they map to the STRIDE threats defined in Table 3. Performance was reported as average times for the main operations in a CPU-based local environment; model initialization was separated from per-request authentication, as it represents a startup cost.

4. Results

This section reports the biometric, PAD, security and performance results most relevant to evaluating threshold-dependent biometric behavior, PAD performance, security verification and execution feasibility.

4.1. Biometric Verification Results

Figure 3 summarizes the score-level biometric evaluation, including the descriptor-distance distribution, threshold sweep, ROC curve and DET curve. Genuine comparisons were concentrated at lower distances, whereas zero-effort impostor comparisons were shifted toward higher distances. This supports the use of distance-based thresholding while showing that the system should not be described as error-free.

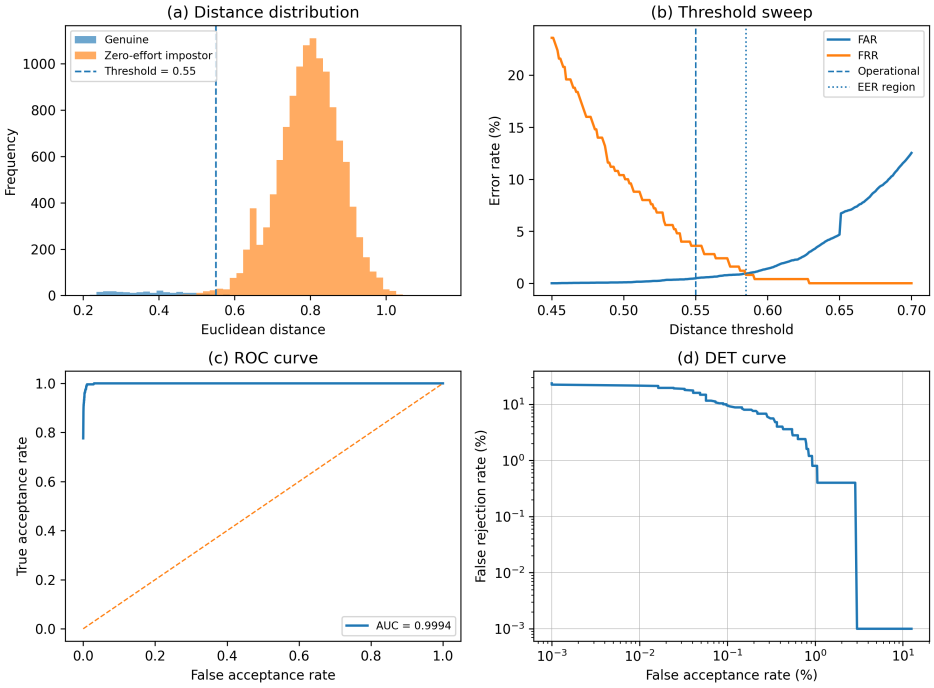


Figure 3. Score-level biometric evaluation: (**a**) Euclidean-distance distribution, (**b**) threshold sweep, (**c**) ROC curve, where the dashed line indicates the no-discrimination diagonal, and (**d**) DET curve.

Table 5 reports the operating-point metrics and threshold-analysis results. At $d \leq 0.55$, the system accepted 241 genuine comparisons and rejected 9, while 57 of 12,250 zero-effort impostor comparisons were falsely accepted. The resulting FAR was 0.4653% and the FRR

was 3.6000%. The ROC AUC was 0.9994 and the EER was 0.8653% at approximately a threshold of 0.585. The operational threshold was therefore stricter than the EER region, prioritizing lower false acceptance over error balance.

Table 5. Biometric verification results at the operating threshold and across thresholds.

Metric	Value	Interpretation
Genuine comparisons	250	241 accepted and 9 rejected at $d \leq 0.55$
Zero-effort impostor comparisons	12,250	57 falsely accepted and 12,193 correctly rejected
FAR at $d \leq 0.55$	0.4653%	95% CI: 0.3593–0.6023%
FRR at $d \leq 0.55$	3.6000%	95% CI: 1.9053–6.6991%
ROC AUC	0.9994	Strong score-level discrimination under the controlled protocol
EER and threshold	0.8653% at $d = 0.585$	EER region differs from the stricter operating threshold

The threshold sweep is relevant for future operational configuration decisions. Table 6 presents representative thresholds from the sweep shown in Figure 3. At a threshold of 0.50, FAR decreased to 0.0898%, but FRR increased to 10.4000%, which would be restrictive for legitimate users. At a threshold of 0.60, FRR decreased to 0.4000%, but FAR increased to 1.4041%. The selected threshold of $d \leq 0.55$ therefore represents a security-oriented compromise: it does not minimize the two error rates equally, but it reduces the probability of false acceptance while keeping the rejection rate within a usable range for the controlled protocol.

Table 6. Representative threshold trade-off for biometric verification.

Threshold	FAR	FRR	Interpretation
0.50	0.0898%	10.4000%	Stronger security posture, but higher rejection of legitimate users.
0.55	0.4653%	3.6000%	Selected operating point; stricter than the EER region.
0.58	0.8327%	1.6000%	Lower rejection but increased false-acceptance exposure.
0.60	1.4041%	0.4000%	More permissive setting; improves convenience at higher security cost.
0.65	4.6776%	0.0000%	No observed rejection, but unacceptable false-acceptance increase for authentication.

4.2. Presentation Attack Detection Results

Figure 4 shows the PAD error rate by presentation type. The PAD evaluation was kept separate from zero-effort impostor verification because a non-mated comparison and a presentation attack represent different threat scenarios.

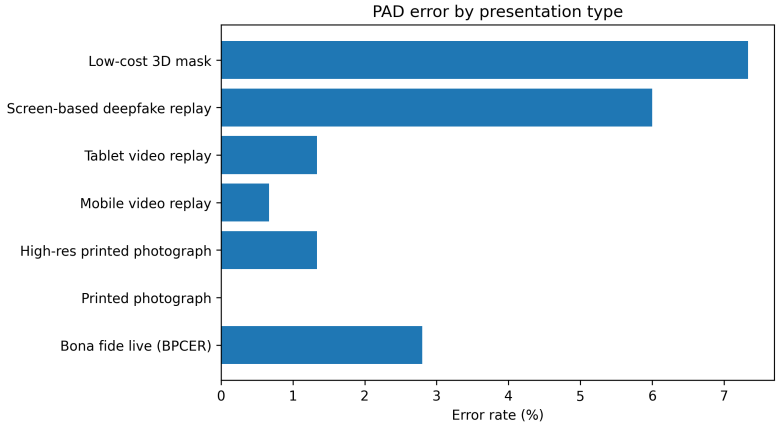


Figure 4. PAD error rate by presentation type. Bona fide live corresponds to BPCER; attack modalities correspond to APCER.

Table 7 reports the PAD metrics. Standard printed photographs were rejected in all evaluated attempts, while more demanding conditions, particularly screen-based deepfake replay and low-cost 3D masks, produced higher APCER. This result supports blink-based liveness as a useful first-level PAD control without overstating it as a complete defense against advanced spoofing.

Table 7. Presentation attack detection results.

Presentation Type	Attempts	Observed Error	Result
Bona fide live	250	BPCER 2.8000%	7 bona fide presentations were incorrectly rejected
Printed photograph	150	APCER 0.0000%	No attack was accepted
High-resolution printed photograph	150	APCER 1.3333%	2 attacks were accepted
Mobile video replay	150	APCER 0.6667%	1 attack was accepted
Tablet video replay	150	APCER 1.3333%	2 attacks were accepted
Screen-based deepfake replay	150	APCER 6.0000%	9 attacks were accepted
Low-cost 3D mask	150	APCER 7.3333%	11 attacks were accepted
Global PAD summary	900 attacks + 250 bona fide	APCER 2.7778%; BPCER 2.8000%; ACER 2.7889%	APCER 95% CI: 1.8055–4.0734%; BPCER 95% CI: 1.1330–5.6837%

4.3. Security and Performance Results

Table 8 summarizes the automated security and performance results. The system achieved 87.4% overall code coverage and completed 64 security tests with no successful exploit observed in the executed test set. The tests included malformed payloads, JWT tampering, access-control bypass, rate limiting, XSS payloads, CSRF-style requests, injection/API bypass and boundary cases. These results show that the implemented controls behaved as expected for the evaluated vectors, but they do not prove complete security.

Table 8. Security verification and performance results.

Block	Result	Interpretation
Code coverage	231 tests; 87.4% overall coverage	Backend security: 89.28%; backend unit: 92.85%; frontend security: 85.2%
Automated security tests	64/64 passed; no successful exploit observed	Evidence for evaluated vectors, including malformed payloads, JWT tampering, RBAC bypass, rate limiting, XSS, CSRF-style requests, injection/API bypass and boundary cases
Traditional login	0.8 s; target < 2 s	Compatible with web login flow
Facial recognition	1.2 s; target < 2 s	Most expensive per-request operation, but within interactive target
Model initialization	2.1 s; target < 3 s	Startup cost, not per-request latency
Descriptor validation	0.3 s; target < 0.5 s	Low-cost comparison stage
Backend API response	0.12 s; target < 0.2 s	Lightweight backend response under local CPU-based execution

The security test results are consistent with the threat model. Invalid or altered tokens were rejected, protected routes preserved RBAC checks, malformed payloads did not crash the service, and boundary cases produced controlled responses. The strongest executable evidence concerns input validation, token verification, access-control enforcement, and defensive error handling. Evidence for advanced biometric spoofing is more limited, as it depends on the PAD protocol rather than web-application tests.

Security evidence should therefore be interpreted by coverage of threat categories, not only by test-count volume. Malformed-payload tests addressed input validation and defensive error handling; JWT tests covered tampering and session integrity; authorization-

bypass tests addressed elevation of privilege; rate-limiting tests covered denial of service; and injection-oriented tests targeted API-input tampering. This mapping supports that the assessment was derived from the threat model, rather than assembled as a generic checklist.

Performance was measured in a CPU-based local environment using a Windows 11 Pro laptop with an Intel Core i7-1355U processor (Intel Corporation, Santa Clara, CA, USA) and 32 GB RAM. Thus, the response times indicate local feasibility, not production-scale latency under concurrent users or cloud deployment. Separating model initialization from per-request descriptor validation is relevant for deployment: initialization affects service startup, whereas descriptor validation shapes the perceived latency of each login attempt.

The aggregated evidence supporting the biometric, PAD, security, coverage and performance results is provided in Supplementary Materials.

5. Discussion

The results support the relevance of the proposed design. The microservice does not treat face detection as sufficient proof of identity; instead, it requires face localization, first-level blink-based liveness validation, descriptor extraction, distance-based matching, token issuance and authorization. This separation is central to the contribution because several web biometric prototypes conflate face presence with identity verification. In the proposed artifact, biometric decisions remain on the server side, which limits client-side decision exposure and makes the service easier to audit.

From a software-engineering perspective, this architecture improves maintainability. Client applications do not need to implement recognition logic, first-level liveness logic, threshold rules, or audit policies locally; they consume the authentication service through well-defined REST interactions. This is relevant for heterogeneous web applications because the biometric component can evolve without forcing every consuming application to reimplement the security workflow.

The biometric results include score-level analysis in addition to login outcomes. At the operational threshold of $d \leq 0.55$, FAR was 0.4653% and FRR was 3.6000%, while the ROC AUC was 0.9994 and EER was 0.8653%. The confidence intervals indicate that these values should be interpreted as sample-based estimates rather than fixed properties of the system. This interpretation is methodologically relevant because biometric performance is affected by variation in capture conditions, cameras, pose, lighting and user diversity.

The PAD results add a second layer of evidence. Blink-based liveness performed well against the evaluated basic printed-photo and conventional replay attacks, but the higher APCER observed for screen-based deepfake replay and low-cost 3D masks shows that it should be interpreted as a lightweight first-level PAD control within the evaluated protocol. This finding clarifies the boundary of the mechanism and avoids an exaggerated claim of universal anti-spoofing robustness. Therefore, the PAD evidence supports the feasibility of integrating blink-based liveness into the authentication workflow, while more adaptive spoofing conditions, higher-quality presentation artifacts and real deployment scenarios require complementary validation.

The security results indicate that STRIDE-derived controls were not only described but also verified through executable evidence. Coverage, token-tampering tests, RBAC bypass attempts, input-validation tests, rate-limiting checks and boundary cases provide a stronger assurance argument than architecture diagrams alone. However, the evidence remains bounded by the evaluated vectors. Manual code review, secret-management review, dependency analysis, deployment hardening, runtime monitoring and adversarial testing would still be required before production deployment. Accordingly, the security evaluation should be read as controlled engineering evidence for the implemented controls, rather than as an exhaustive assurance claim across all possible operational threats.

https://doi.org/10.3390/engproc2026153010

The combined evaluation is the study's main methodological contribution. Biometric studies often report recognition accuracy without showing how the component behaves within an authentication service. Conversely, secure-development studies commonly report testing and threat modeling without examining the biometric decision boundary. This work links both perspectives: the biometric decision is represented through distances, thresholds, ROC/DET curves, and PAD metrics, while the software service is evaluated through STRIDE-derived controls, coverage, API tests, and response-time behavior. This integrated view is especially relevant for systems that must be assessed against machine-learning, software-security, and operational requirements.

The results also have practical implications for reuse and adoption. A consuming application does not need to know the internal details of the facial descriptor or the liveness implementation; it only requires a stable authentication interface and clear semantics for success, rejection, failure and audit events. This suggests that the microservice may serve as a reusable integration pattern for environments in which several web applications need stronger authentication but cannot maintain independent biometric pipelines. At the same time, the service-based architecture centralizes risk: a design weakness in the microservice could affect all consuming applications, which reinforces the need for hardening, monitoring and governance before operational adoption. For institutional settings, this architecture lowers the integration barrier by centralizing enrollment, verification, audit and security controls through REST interactions, but adoption would still require validation under heterogeneous cameras, variable lighting, repeated sessions, concurrent workloads and production-oriented security governance.

The study contributes a reusable and auditable authentication artifact that connects biometric verification with software-security assurance. The proposed microservice integrates identity verification, first-level blink-based liveness detection, server-side decision-making, STRIDE traceability and automated testing in a form that can be consumed by external web applications. The strongest evidence supports the architectural integration, biometric score-level evaluation, PAD reporting and security-test traceability. Evidence remains more limited for broad demographic generalization, robustness against advanced spoofing attacks and production-scale scalability. Within the scope of the present controlled evaluation, these findings provide a rigorous basis for extending the artifact toward broader deployment-oriented and adversarial validation.

6. Conclusions and Research Directions

This study presented a facial-biometric authentication microservice with integrated security controls for web applications. The proposed artifact combines credential-based login, server-side facial verification, a blink-based first-level liveness check, JWT-based session handling, audit logging, STRIDE-informed secure design and automated security testing. The main contribution lies in integrating biometric verification and software-security assurance into a reusable authentication component that can be consumed by external web applications.

The controlled evaluation provided complementary evidence at the biometric, PAD, software-security and performance levels. By separating genuine comparisons, zero-effort impostor comparisons and presentation-attack samples, the protocol allowed each metric to be interpreted within its corresponding data scope. The study reported threshold-dependent FAR/FRR behavior and incorporated ROC, DET and EER analysis. PAD performance was assessed through APCER, BPCER and ACER, while the security evaluation linked STRIDE-derived threats, implemented controls and executable tests. Performance measurements showed that the main operations remained within the predefined local response-time targets.

https://doi.org/10.3390/engproc2026153010

Taken together, the results support the technical feasibility of the proposed microservice under the evaluated conditions. The evidence is strongest for architectural integration, biometric score-level evaluation, PAD reporting and security-test traceability. Its scope is bounded to controlled engineering validation, leaving production-scale behavior, broader demographic generalization and adaptive spoofing resistance as areas for further empirical assessment.

From a practical perspective, the proposed approach may serve as an integration pattern for institutions that need stronger authentication without redesigning each application around a local biometric module. A reusable service can centralize enrollment, verification, logging, and security controls, while external web applications consume a stable authentication interface. This separation reduces duplicated implementation effort and supports more consistent security governance across heterogeneous web systems, provided that operational hardening, monitoring, and deployment-specific validation are completed before institutional adoption.

Future work should extend the current evaluation in three complementary directions. First, biometric validation should involve larger and more diverse participant groups, heterogeneous cameras, uncontrolled illumination, pose variation, and repeated sessions over time, in order to assess how stable the reported verification behavior remains beyond the controlled setting. Second, the PAD module should be strengthened with randomized challenges, texture or depth cues, and model-based presentation-attack classifiers, including stronger replay, deepfake-based and mask scenarios. Third, the software-security evaluation should move toward production-oriented conditions through load testing, cloud deployment, dependency scanning, secret-management review, runtime monitoring, and incident-response procedures. In parallel, template protection should be reinforced through encryption at rest, cancellable templates, dedicated key management, or privacy-preserving comparison mechanisms.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/engproc2026153010/s1, Supplementary Materials: Aggregated evidence workbook containing coverage, performance, security-test payloads, summarized logs, biometric distance bins, threshold sweep, biometric metrics, PAD metrics and executive summary.

Author Contributions: Conceptualization, methodology, validation and formal analysis, G.C. and H.M.; investigation and writing—original draft preparation, F.F.-A. and M.Á.-J.; data curation, writing—review and editing and supervision, G.C. and M.Á.-J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The aggregated evidence supporting the reported results is provided as Supplementary Materials. Raw facial images, individual biometric samples and personal data are not publicly available due to ethical and privacy restrictions related to biometric information.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Paul, J.; Mitra, A.; Kulhadiya, D.; Pawar, T.; Roy, A.; Sil, J. A comprehensive approach to real-time attendance systems: Integrating face recognition and emotion detection and with web technologies. *Procedia Comput. Sci.* **2025**, *258*, 3436–3446. [CrossRef]
2. Ryando, C.; Sigit, R.; Setiawardhana; Dewantara, B.S.B. Face recognition for logging in using deep learning for liveness detection on healthcare kiosks. *Int. J. Inform. Vis.* **2025**, *9*, 295–302. [CrossRef]
3. Shinde, S.R.; Bongale, A.M.; Dharrao, D.; Thepade, S.D. An enhanced light weight face liveness detection method using deep convolutional neural network. *MethodsX* **2025**, *14*, 103229. [CrossRef]
4. Douklias, A.; Zorzos, I.; Maltezos, E.; Nousis, V.; Bolierakis, S.N.; Karagiannidis, L.; Ouzounoglou, E.; Amditis, A. A multicomponent face verification and identification system. *Appl. Sci.* **2025**, *15*, 8161. [CrossRef]

5. Kira, M.; Alajamy, Z.; Soliman, A.; Mesbah, Y.; Mazzara, M. Trustworthy face recognition as a service: A multi-layered approach for mitigating spoofing and ensuring system integrity. *Future Internet* **2025**, *17*, 450. [CrossRef]
6. Pabiania, M.D.; Relucio, C.E.S.; Santos, M.S.P. Patient information system for dental clinics with integrated facial recognition. In *Proceedings of the 2024 14th ICBET*; Association for Computing Machinery: New York, NY, USA, 2024; pp. 211–215. [CrossRef]
7. Hsieh, W.-B. BF-ACS—Intelligent and immutable face recognition access control system. *IET Inf. Secur.* **2025**, *2025*, 6755170. [CrossRef]
8. Chernov, K.; Kuznetsova, K.; Oleshko, I.; Bagmut, M.; Fedorchuk, A.; Kryzhanovskyi, D. Biometric images attacks detecting using convolutional neural networks. *CEUR Workshop Proc.* **2020**, *2654*, 107–121.
9. Hofbauer, H.; Debiasi, L.; Kränkl, S.; Uhl, A. Exploring presentation attack vulnerability and usability of face recognition systems. *IET Biom.* **2021**, *10*, 219–232. [CrossRef]
10. Utama, F.P.; Nurhadi, R.M.H. Uncovering the risk of academic information system vulnerability through PTES and OWASP method. *CommIT J.* **2024**, *18*, 39–51. [CrossRef]
11. Zeeshan, N.; Bakyt, M.; Moradpoor, N.; La Spada, L. Continuous authentication in resource-constrained devices via biometric and environmental fusion. *Sensors* **2025**, *25*, 5711. [CrossRef]
12. Amin, R.K.; El Khayat, G.A.; El Sahn, F.; Amer, A.A. Enhancing E-Banking security and personalization through convolutional neural network-based facial recognition. *J. Comput. Sci.* **2025**, *21*, 2323–2336. [CrossRef]
13. Tariq, M.U. *Enhancing Cyber Resilience in Software Development*; Book Chapter; IGI Global Scientific Publishing: Hershey, PA, USA, 2025; pp. 35–64. [CrossRef]
14. Humayun, M.; Jhanjhi, N.; Almufareh, M.F.; Khalil, M.I. Security threat and vulnerability assessment and measurement in secure software development. *Comput. Mater. Contin.* **2022**, *71*, 5039–5059. [CrossRef]
15. Souppaya, M.; Scarfone, K.; Dodson, D. *Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2022. [CrossRef]
16. Rodriguez, K.S.R.; Higuera, J.B.; Montalvo, J.A.S. Secure development methodology for full stack web applications: Proof of the methodology applied to Vue.js, Spring Boot and MySQL. *CMC-Comput. Mater. Contin.* **2025**, *85*, 1807–1858. [CrossRef]
17. Carroll, A.; Latifi, S. Comparative analysis of passkeys (FIDO2 authentication) on Android and iOS for GDPR compliance in biometric data protection. *Electronics* **2025**, *14*, 4018. [CrossRef]
18. Choi, H.; Woo, S.S.; Kim, H. Blind-Touch: Homomorphic encryption-based distributed neural network inference for privacy-preserving fingerprint authentication. *Proc. AAAI Conf. Artif. Intell.* **2024**, *38*, 21976–21985. [CrossRef]
19. Imran, M.; Umar, M.S.; Malhotra, S. Privacy preserving cancellable template generation for crypto-biometric authentication system. *IEEE Access* **2025**, *13*, 158322–158339. [CrossRef]
20. Oruma, S.O.; Sánchez-Gordón, M.; Gkioulos, V. Enhancing security, privacy and usability in social robots: A software development framework. *Comput. Stand. Interfaces* **2026**, *96*, 104052. [CrossRef]
21. Ivanou, M.; Mikhel, S.; Maloletov, A. ROS-like framework using modern development concepts and microservices. In *Proceedings of the 2021 International Conference NIR*; IEEE: New York, NY, USA, 2021; pp. 1–6. [CrossRef]
22. Farro, D.O.Y.; Fernández, A.M. Convolutional neural networks for automatic identification of individuals at terrestrial terminals. *IEEE Access* **2025**, *13*, 79825–79831. [CrossRef]
23. Birgisdóttir, E.L.; Kunkel, M.I.; Pleva, L.; Papaioannou, M.; Choudhary, G.; Dragoni, N. Exploring the security of mobile face recognition: Attacks, defenses and future directions. *Appl. Sci.* **2025**, *15*, 13232. [CrossRef]
24. Wu, Y.; Pang, M.; Ma, J.; Ou, W.; Yue, Q.; Han, W. An identity management scheme based on multi-factor authentication and dynamic trust evaluation for telemedicine. *Sensors* **2025**, *25*, 2118. [CrossRef]
25. Tarek, M.; Hamouda, E.; Alsirhani, A.; Alomari, A.; Mostafa, A.M. A keyless multimodal-based user authentication scheme using generative adversarial networks. *PeerJ Comput. Sci.* **2025**, *11*, e3360. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.