Security Architecture: SIM-Based Verification

Deep dive into GSMA TS43 protocols and the Fortress Architecture.

TECHNICAL SPECIFICATION & THREAT MODELING

A carrier-authoritative approach to identity

This system utilizes the GSMA TS43 specification to verify phone numbers directly via the device's SIM card, bypassing SMS intermediaries entirely.

Carrier-Authoritative

Verification comes directly from the mobile carrier core network.

Cryptographically Secured

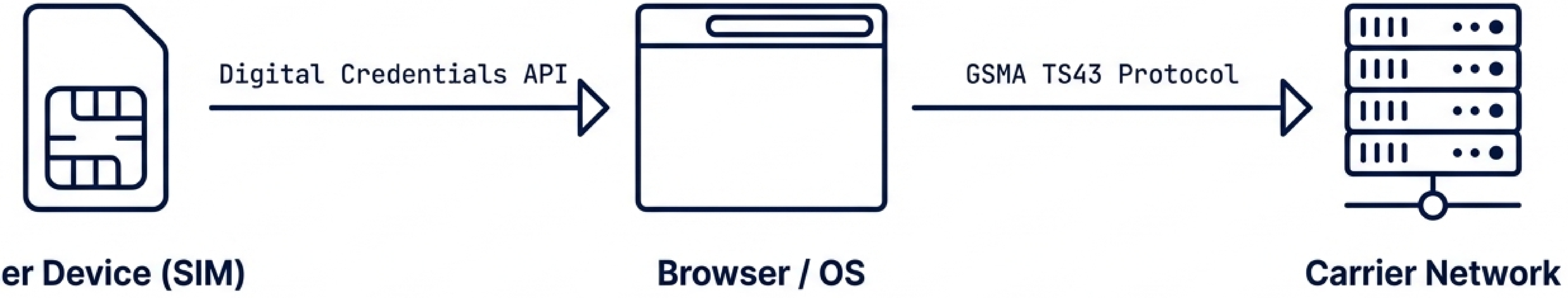
Built on industry-standard encryption and digital signatures. No shared secrets.

Network Agnostic

Functions seamlessly over WiFi, cellular data, or ethernet.

`Implementation: Android Digital Credentials API (Chrome 128+ / Google Play Services 24.0+)`

The Protocol Landscape: TS43 & Digital Credentials



The Standard (TS43)

GSMA specification for Service Entitlement Configuration. The global standard for device-to-network identity proofing.

The Interface

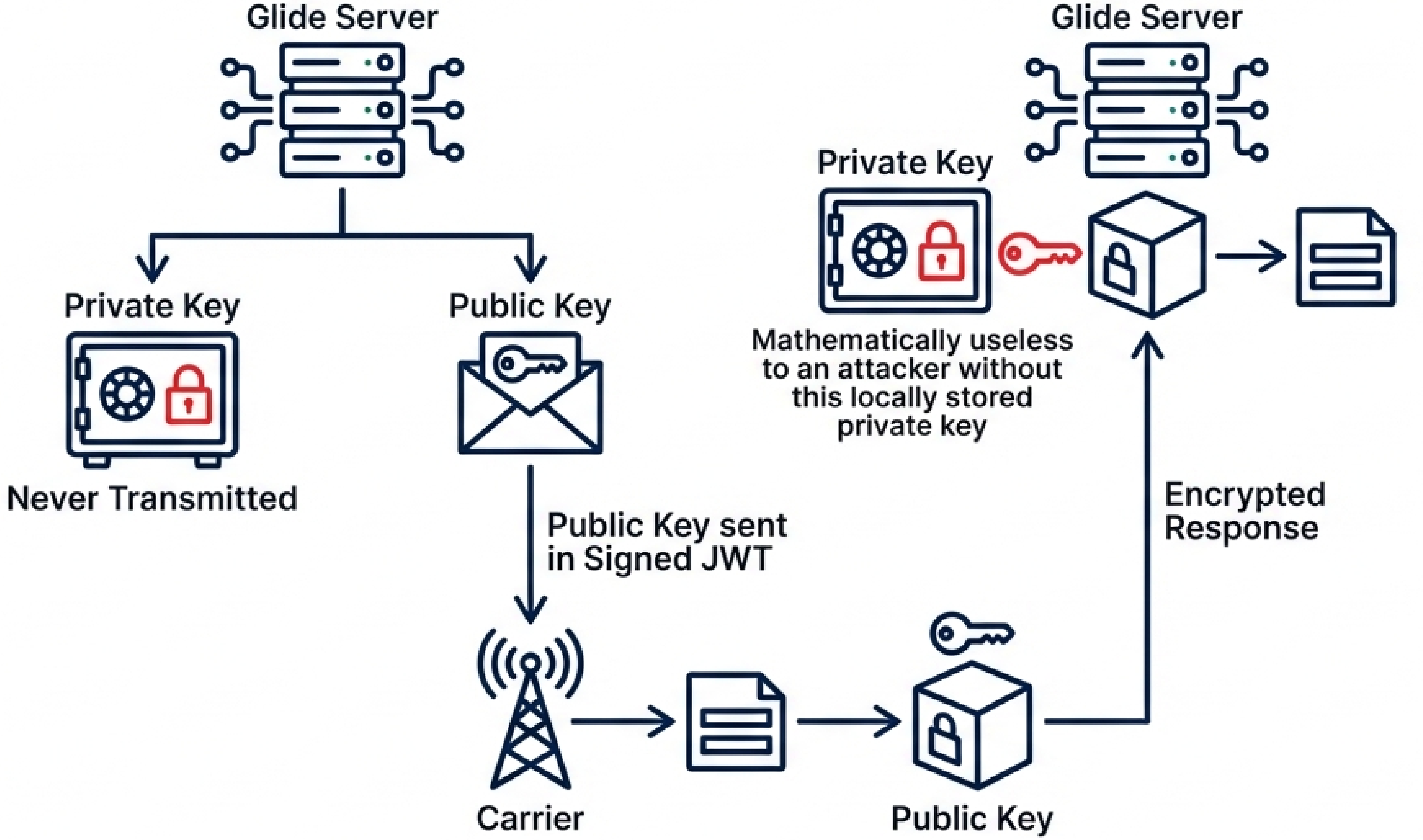
The Digital Credentials API acts as the secure pipe through which the browser requests cryptographic proofs from the SIM card.

Key Takeaway:

Replaces insecure OTPs with hardware-backed cryptographic proofs.

Mechanism 1: Per-Session Ephemeral Encryption

Unique keys for every single request



Technical Detail

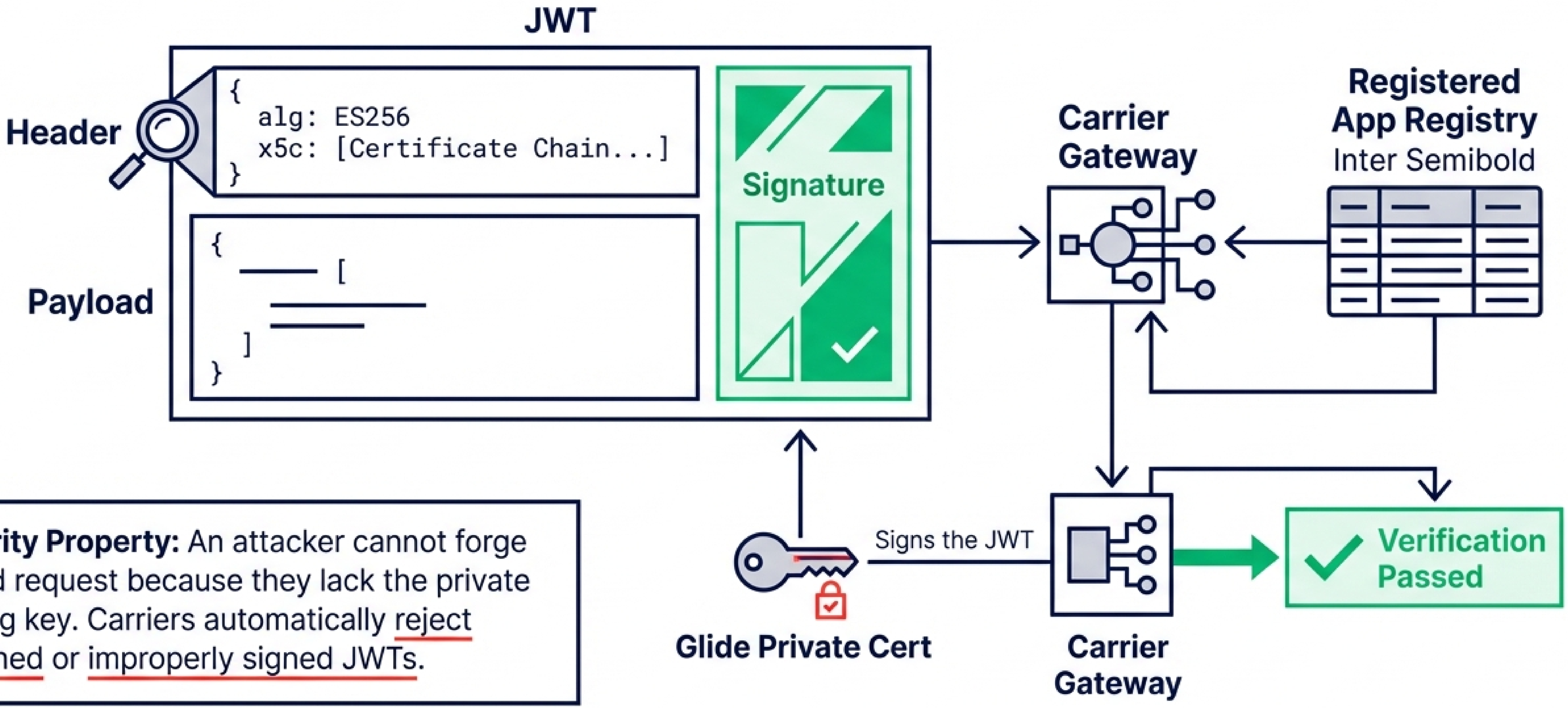
Algorithm: `ECDH-ES`

Curve: `P-256`

Type: `Ephemeral Static`

Mechanism 2: Identity & Trust via Signed JWTs

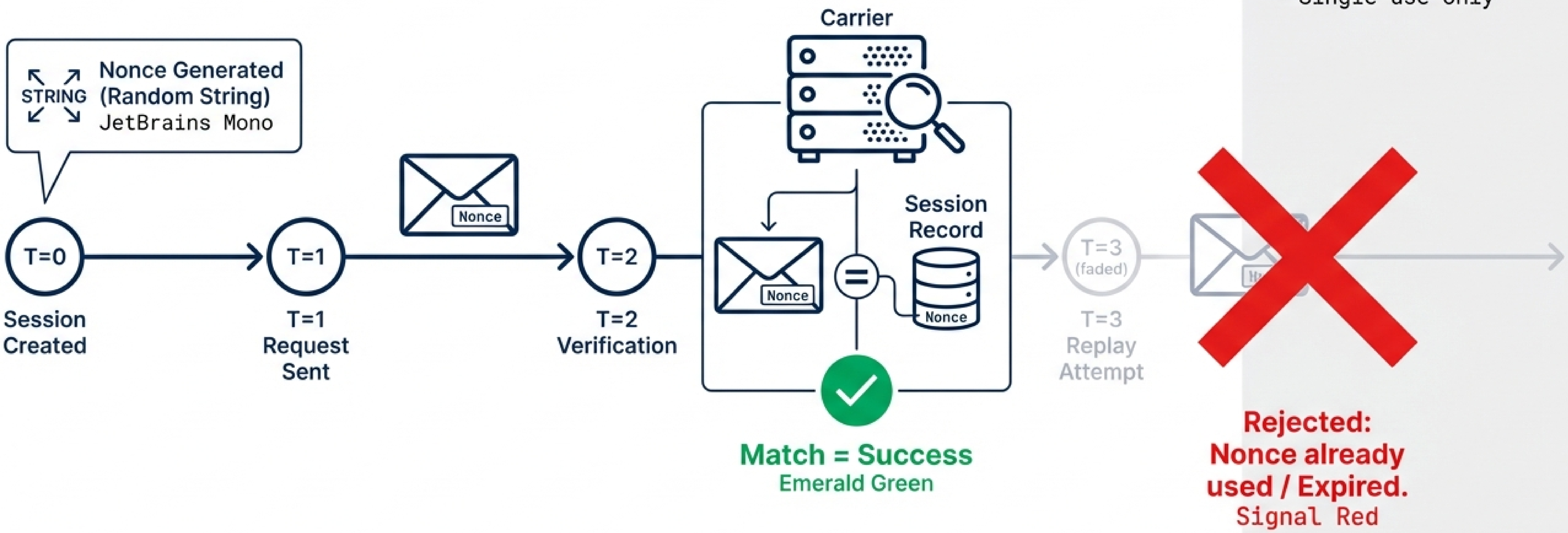
Cryptographic proof of origin.



Security Property: An attacker cannot forge a valid request because they lack the private signing key. Carriers automatically reject unsigned or improperly signed JWTs.

Mechanism 3: Nonce-Based Replay Prevention

Ensuring freshness and preventing reuse



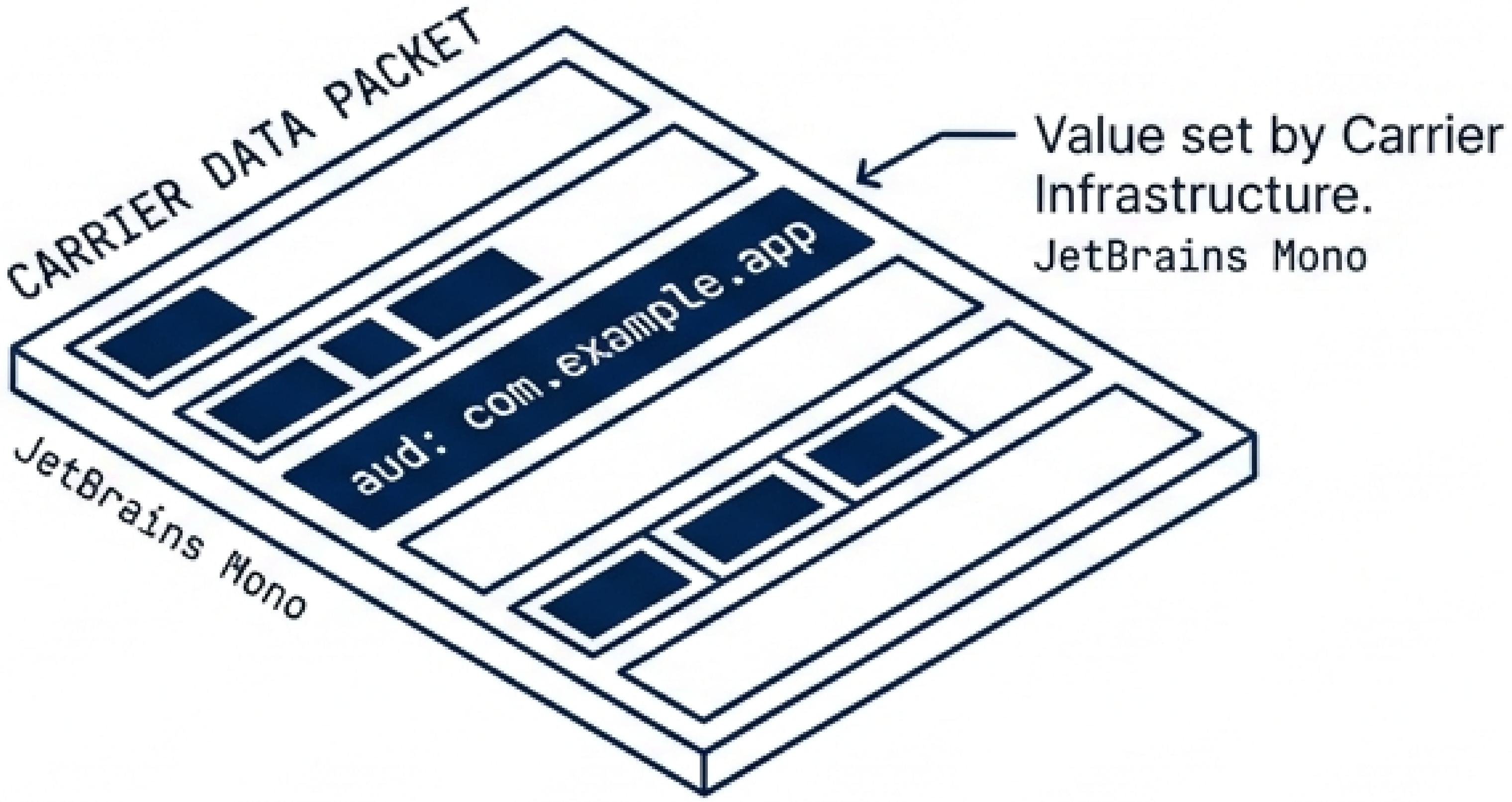
Lifecycle Rules:

- Unique per session
- Embedded in JWT
- Short expiration window
- Single-use only

Mechanisms 4 & 5: Context & Client Binding

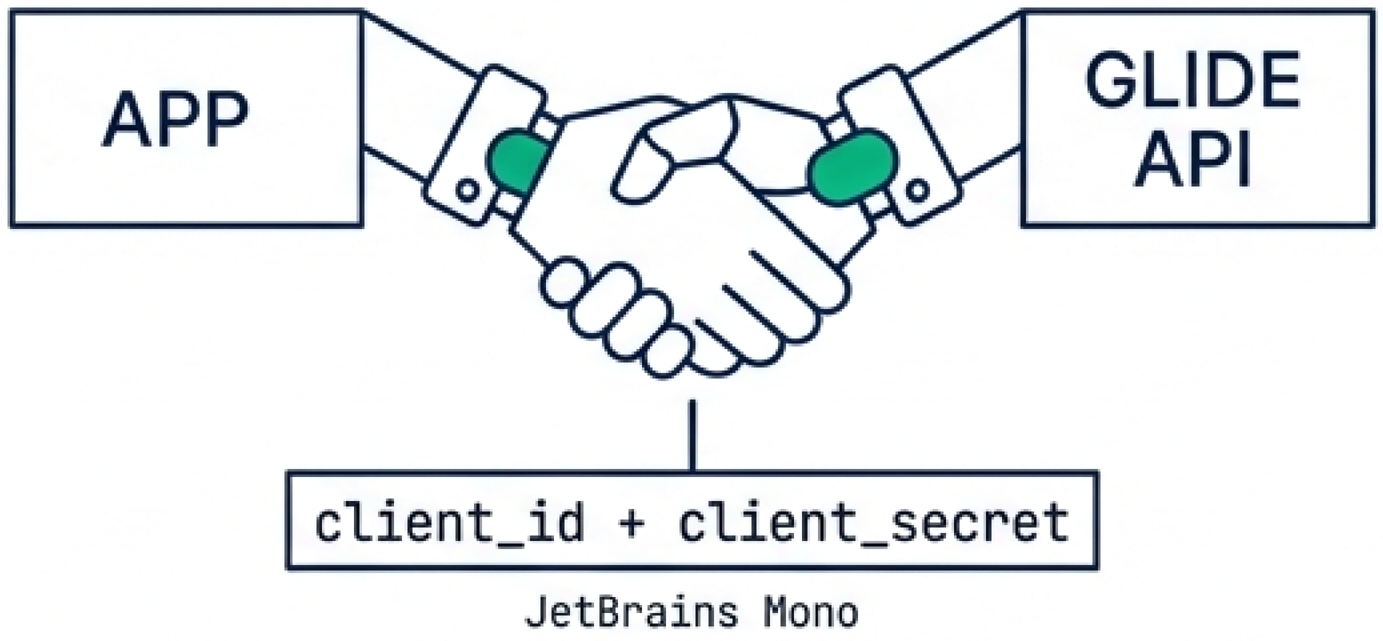
Secure app context and credential isolation.

Carrier-Issued Audience Claim



Based on certificate validation, not developer input. Spoofing is impossible.

OAuth2 Client Binding



Sessions are strictly bound to the developer who created them. Credentials from App A cannot complete a session for App B.

Threat Analysis: Network Layer Attacks

The Attack: Credential Interception

0110010
0111015
0310010
0213010
Sniffer
0138010
0111010
0110010
0111010
0110010
0113016

Attacker captures data in transit.

The Defense: Per-Session Encryption

0110010
0112010
0130010
0111016
ECDH-ES
Encrypted
Senxer: Randee Noise
RSR01-&X%-R".SRKS1/E5
NSR5Mo?1 Random Noise

Without the private key (stored only on server), data is random noise.

The Attack: Man-in-the-Middle (MitM)

Attacker modifies traffic.

The Defense: Signed JWTs

JWT
Payload
ES256
Signature
REJECTED

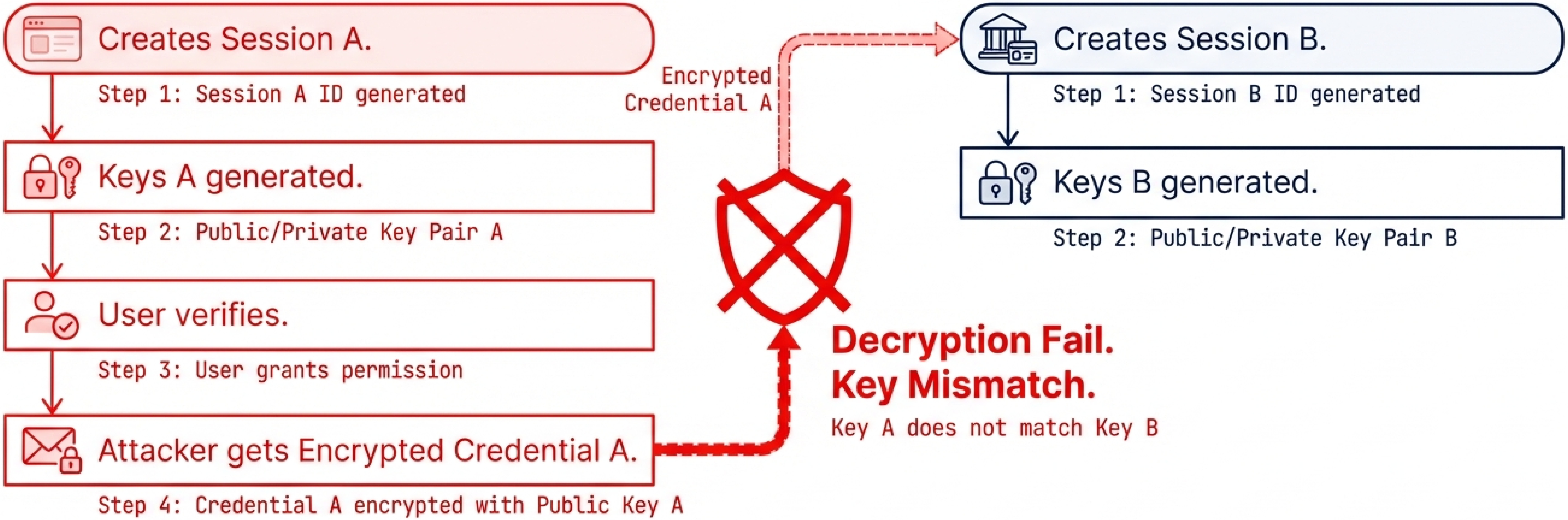
Any modification breaks the ES256 signature. Request rejected.

Threat Analysis: Identity & Spoofing

Scenario Header: The “Fake Integration” Attack

Path A (Red): “Attacker App”

Path B (Blue): “Legitimate Bank App”



Creates Session A.

Step 1: Session A ID generated

Keys A generated.

Step 2: Public/Private Key Pair A

User verifies.

Step 3: User grants permission

Attacker gets Encrypted Credential A.

Step 4: Credential A encrypted with Public Key A

Encrypted Credential A

Creates Session B.

Step 1: Session B ID generated

Keys B generated.

Step 2: Public/Private Key Pair B

**Decryption Fail.
Key Mismatch.**

Key A does not match Key B

OAuth2 & Audience Binding ensures that credentials from an attacker’s session cannot be decrypted by or used in a legitimate application’s session.

Threat Analysis: Replay & Hijacking

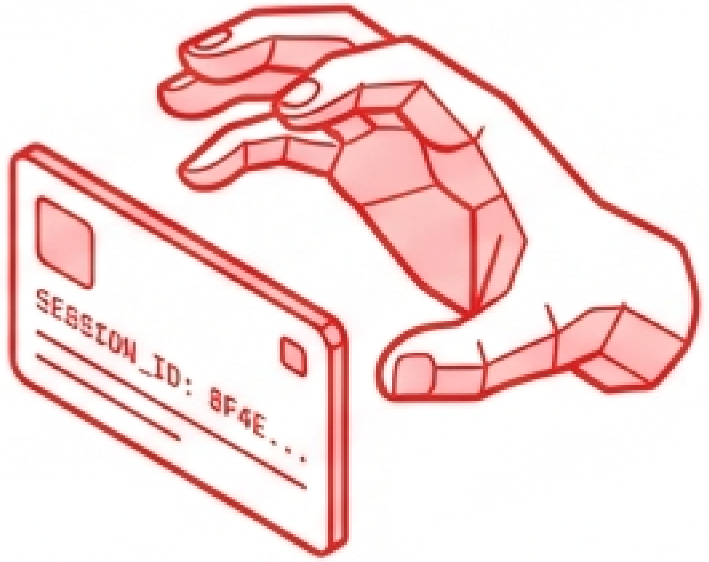
Replay Attack

Scenario: Attacker resubmits valid credential 10 minutes later.

Defense Mechanism: Nonce Validation.

Explanation: Unique nonce is checked against database. If used or expired, request fails. `JetBrains Mono`

Session Hijacking



Scenario: Attacker steals a session ID reference.

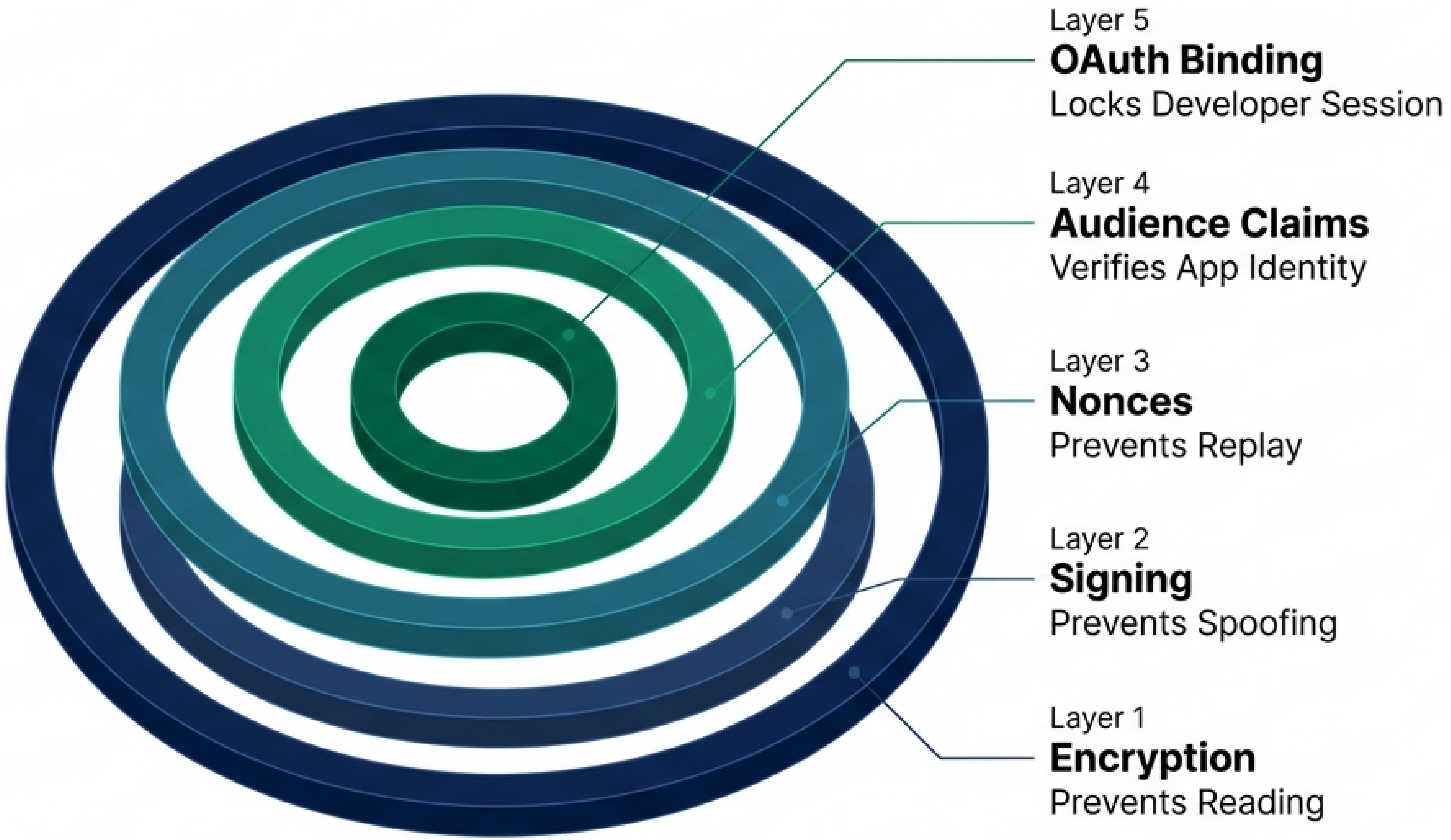
Defense Mechanism: Server-Side State.

Explanation: Session ID is just a pointer. Actual keys live server-side. Without the original `client_secret`, the ID is useless.

Cryptographic Standards & Specifications

Specification	Technical Implementation (JetBrains Mono)	Details
Key Agreement	ECDH-ES (Elliptic Curve Diffie-Hellman Ephemeral Static)	Curve P-256
Payload Encryption	`AES-128-GCM`	Authenticated Encryption
Authorization Signing	ES256	ECDSA using P-256 and SHA-256
Entity Authentication	`X.509 Certificate Chain`	Public Key Infrastructure
Protocol Version	`GSMA TS.43 v11.0`	Service Entitlement Configuration

Summary of Protections



The combination of these mechanisms makes it cryptographically infeasible to intercept, modify, replay, or forge credentials.

Technical References

- GSMA TS.43 v11.0: Service Entitlement Configuration
- W3C Specification: Digital Credentials API
- RFC 7518: JSON Web Algorithms (JWA)
- RFC 7516: JSON Web Encryption (JWE)
- RFC 6749: OAuth 2.0 Authorization Framework