JCF: Biometric-Aware Cold Crypto Wallets on the Javacard Platform (jNet)

V1.0

Mikhail Friedland
CEO, jNet Secure Inc.
Javacard Forum Member

Types of CryptoWallets:

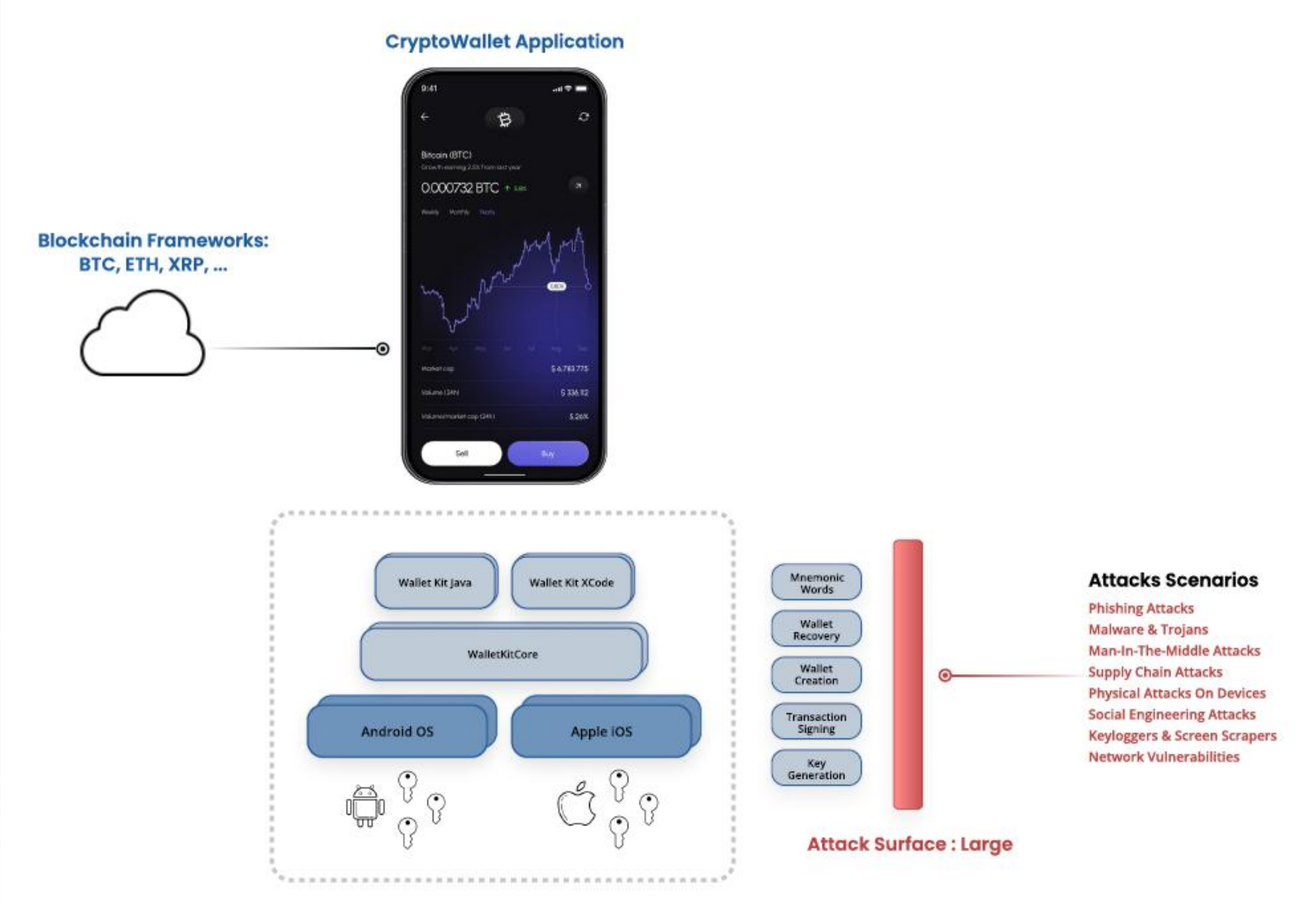
- **Warm Wallets:**

  - Hybrid approach that combines features of hot and cold wallets.
  - Typically connected to the internet intermittently for transaction processing.
  - Suitable for medium-term holdings or occasional transactions.
  - Keys typically reside within the smartphone or desktop file system
  - Offer more convenience than cold wallets but increased exposure to online threats.

- **Cold Wallets:**

  - Fully offline wallets, designed for long-term storage of assets.
  - Immune to remote cyberattacks like hacking or phishing.
  - Require manual effort to access keys for transactions, sacrificing convenience.
  - Considered the most secure option for large holdings.

Warm Wallets : Large Attack Surface



CryptoWallet Application

Blockchain Frameworks: BTC, ETH, XRP, ...

Wallet Kit Java

Wallet Kit XCode

WalletKitCore

Android OS

Apple iOS

Mnemonic Words

Wallet Recovery

Wallet Creation

Transaction Signing

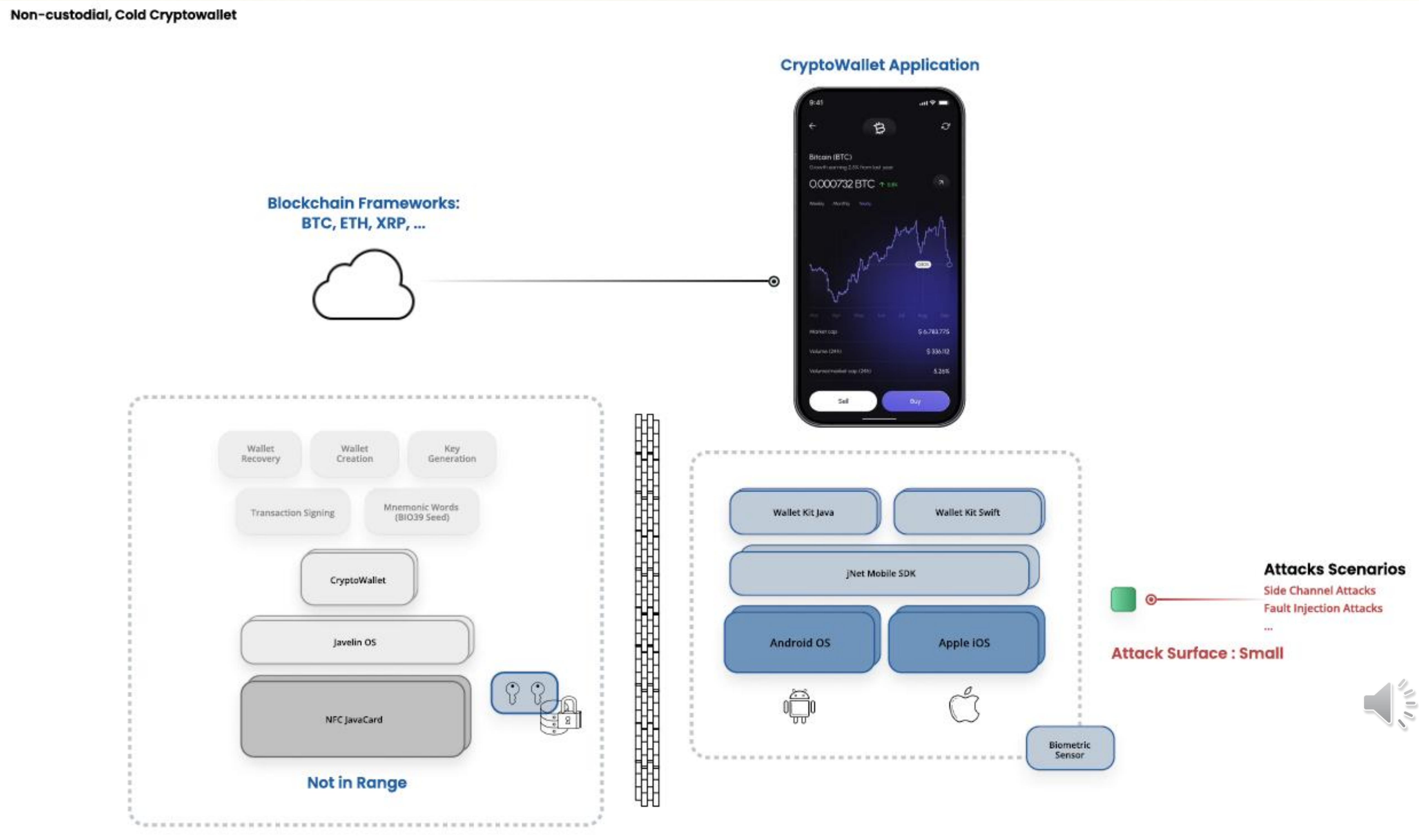
Key Generation

Attack Surface : Large

Attacks Scenarios

Phishing Attacks
Malware & Trojans
Man-In-The-Middle Attacks
Supply Chain Attacks
Physical Attacks On Devices
Social Engineering Attacks
Keyloggers & Screen Scrapers
Network Vulnerabilities

Cold Wallets : Small Attack Surface



Non-custodial, Cold Cryptowallet

CryptoWallet Application

Blockchain Frameworks:
BTC, ETH, XRP, ...

Wallet Recovery

Wallet Creation

Key Generation

Transaction Signing

Mnemonic Words (BIO39 Seed)

CryptoWallet

Javelin OS

NFC JavaCard

Not in Range

Wallet Kit Java

Wallet Kit Swift

jNet Mobile SDK

Android OS

Apple iOS

Biometric Sensor

Attacks Scenarios

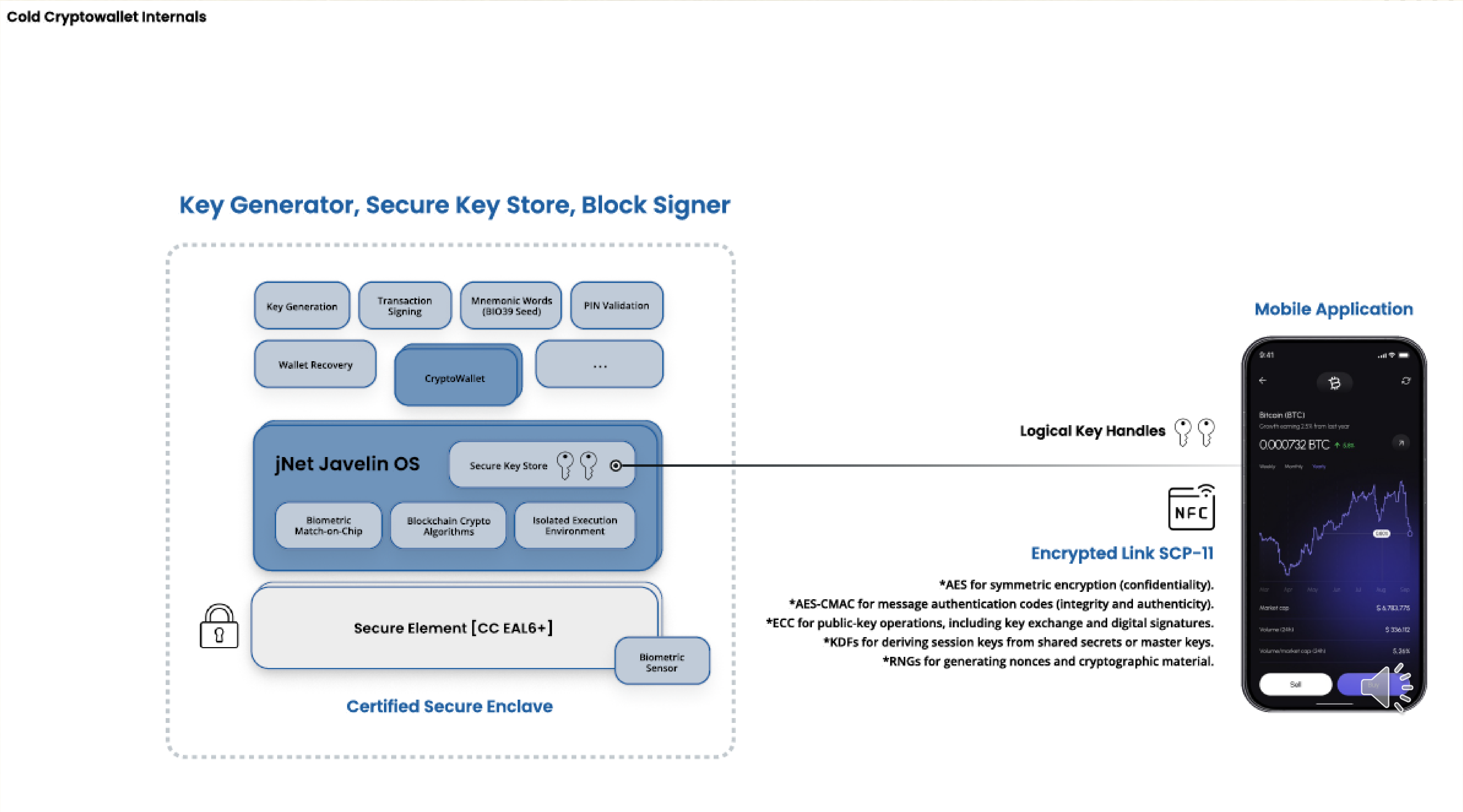
Side Channel Attacks

Fault Injection Attacks

...

Attack Surface : Small

Cold Cryptowallet : Why secure?



*AES for symmetric encryption (confidentiality).

*AES-CMAC for message authentication codes (integrity and authenticity).

*ECC for public-key operations, including key exchange and digital signatures.

*KDFs for deriving session keys from shared secrets or master keys.

*RNGs for generating nonces and cryptographic material.

JNET

Certifications for Trusted Security

- **Foundation of Trust:**
  - Trust is the cornerstone of any crypto wallet, ensuring users' confidence in the safety of their assets and private keys.
  - A trusted wallet must prevent unauthorized access, ensure transaction authenticity, and protect against compromise.

- **JavaCard Platform Certifications:**
  - **Common Criteria (CC) EAL5+ and EAL6+:** Validates the platform's tamper resistance and secure key management.
  - **FIPS 140-3:** Ensures compliance with cryptographic module security standards.
  - **EMVCo Certification:** Proven capability to host payment applets alongside cryptographic functions, expanding wallet versatility.

- **Support for Modern Cryptographic Algorithms:**
  - JavaCard supports advanced cryptography for blockchain operations, including:
    - **ECDSA (curves secp256k1, curve 25519, NIST curves)** for efficient and secure key generation and transaction signing.
    - **EdDSA (curve ED25519)**
    - **Secure RNG and HMAC**

- **Audited and Certified Security:**
  - JavaCard undergone rigorous independent evaluations, ensuring users and businesses can rely on its security and compliance.

- **Trust Through Proven Performance:**
  - With a 20+ year history in securing SIM cards, payment systems, ePassports and EMVCo solutions JavaCard is a time-tested platform for building trustworthy crypto wallets.

GlobalPlatform Runtime

- Benefits:
  - Secure executable content and apps lifecycle management.
  - Key separation and secure key usage.
  - Interoperability with international standards
  - In-field applet updates via Amendment H.
- Secure Channels:
  - SCP-02, SCP-03, SCP-11b, …
- CVM: Biometric and PIN authentication methods.

Javacard in EU Digital ID Wallet

- Open Standards & Interoperability
- Certification Advantages
- GlobalPlatform Integration
- Multi-applet Management
- Trusted by Governments & Industry Sectors
- Vendor Independence and Cost Efficiency

Biometric Authentication

- Enhanced Security
- Convenience for Users
- Multi-Factor Authentication
- Tamper-Resistant Processing
- Reduced Risk of Credential Theft
- Trusted Integration with GlobalPlatform CVM

Multi-Applet Environments

- Secure Coexistence of Applets
- Efficient Use of Resources
- Interoperability Across Use Cases
- Dynamic Applet Management
- Simplified User Experience
- Enhanced Security Across Applications
- Use Case Synergy

Addressing CryptoWallet Challenges

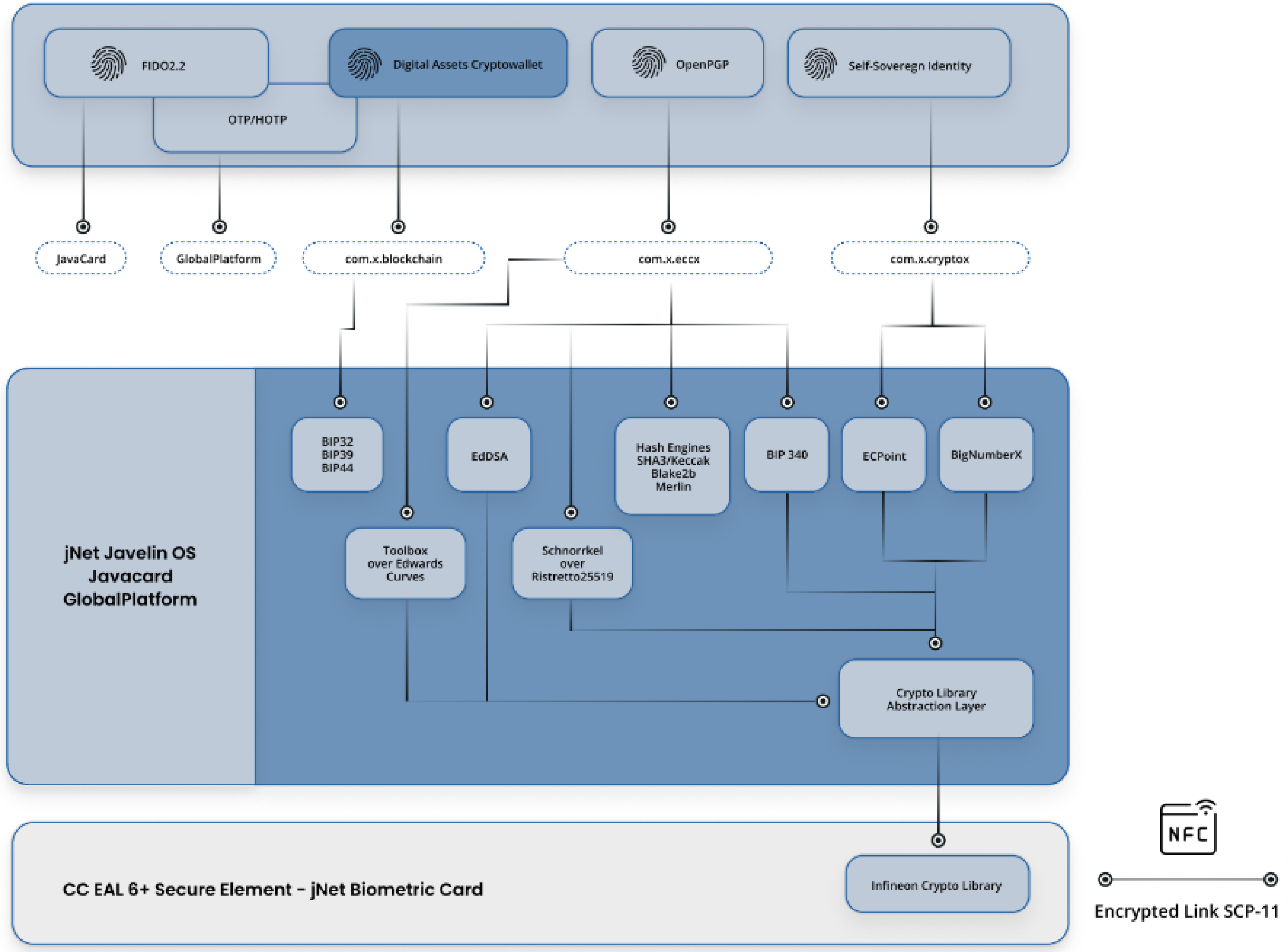
- Protecting Private Keys
- Countering Malware and Phishing
- Resisting Side-Channel and Physical Attacks
- Ensuring Software Integrity
- Supporting Interoperability Across Ecosystems
- Dynamic and Scalable Lifecycle Management
- Minimizing Human Errors

Key Takeaways

- Proven security across finance, government, and telecom.
- Meets global standards for wallet certification.
- Tamper-proof form factors with biometric smartcards
- Multi-factor authentication
- Support for crypto algorithms with extended Java packages
- Secure key storage and robust wallet recovery

Javacard Cold Cryptowallet : E2E

Cryptowallet and FIDO2.1 applets co-existing on a Javacard Runtime as part of jNet's Cold Wallet Solution

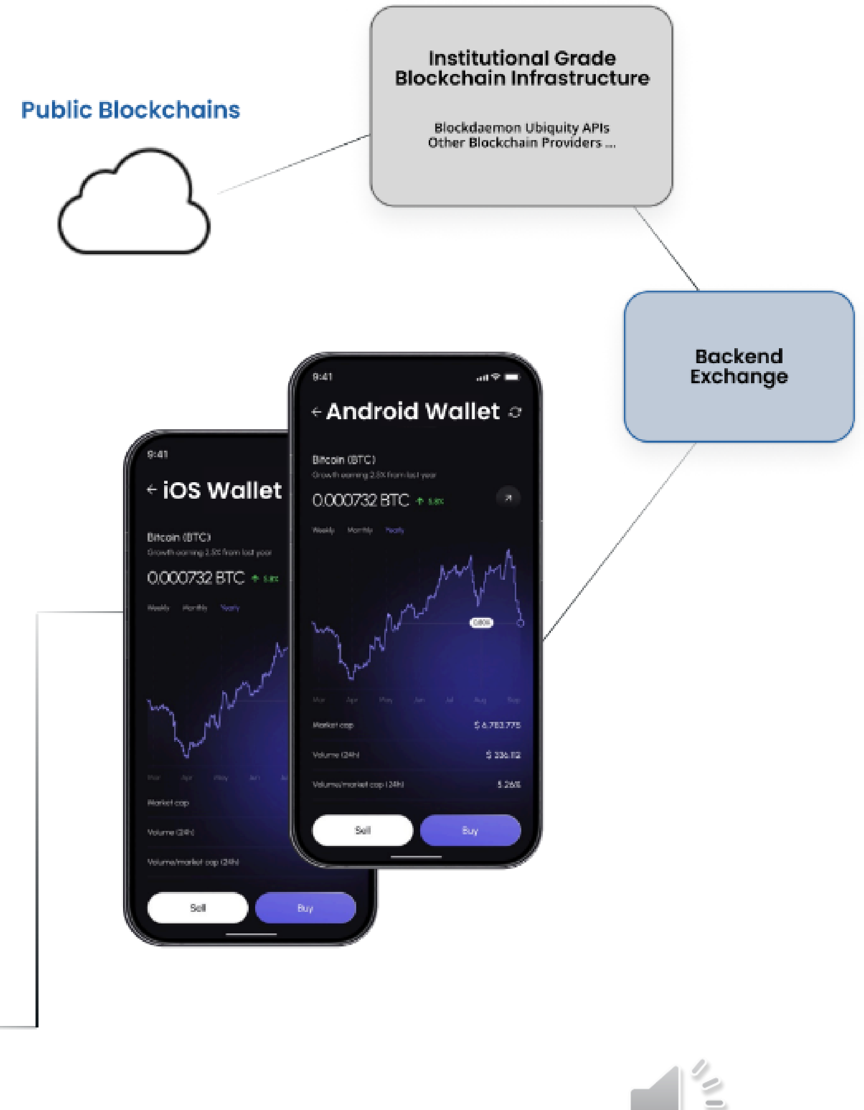


Javelin OS Java-programmable w/Biometric Sensor and crypto accelerated primitives to support Top100 coins (future proofing).

- ECDSA (curves secp256k1, curve25519, ristretto25519, NIST P-256, NIST P-384)
  Keypair generation
  Sign/Verify with ALG_ECDSA_SHA_256
  KeyAgreement with ALG_EC_SVDP_DH_PLAIN
- EdDSA (curve ED25519)
  Keypair generation (with optional Blake2b-512 hash during key derivation)
  Sign/Verify
- EC-Schnorr (curves secp256k1, curve25519, ristretto25519)
  Keypair generation
  Sign/Verify
- Sign with HMAC_SHA512
- Encrypt/Decrypt AES Cipher with ALG_AES_CBC_PKCS5
- Random Number Generator:
- ALG SECURE RANDOM
- ALG_PSEUDO_RANDOM
- Checksum computing with algorithm ALG_ISO3309_CRC16 and CRC32
  Messages digest hash functions:
  SHA-1, SHA-2x, SHA-3
  RIPEMD160
  Keccak
  Blake2b
- Big Integer Accelerator APIs mapped to Hardware Co-processor
  PBKDF2 function (as per BIP39 standard) using HMAC-SHA512 as digest function)

Mobile SDK
[supports all crypto enhancements]

Crypto Algorithms Provided by SDK



Conclusion and Q&A

- Thank you for attending!
- Start Questions …