

Bank-Grade Mobile Wallet Architecture: The Secret to Wallet Core & Multi-Chain Mastery

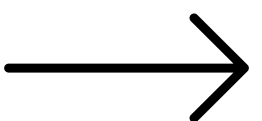
System Design

From Code to Business Value



Esmaeel Moradi

Senior Android Engineer



Bank-Grade Mobile Wallet Architecture: The Secret to Wallet Core and Multi- Chain Mastery

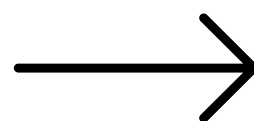
We know the heart of any mobile wallet is the **Wallet Core layer**.

A platform's success depends on its ability to handle multiple protocols with a **single cryptographic engine (Core)**, not building a separate app for every coin!



Esmaeetil Moradi

Senior Android Engineer



Defining Wallet Core and Security

What is Wallet Core? (The Trust Wallet Standard)

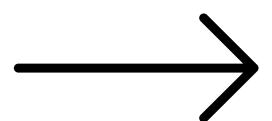
Wallet Core is a Cross-Platform library that implements low-level cryptographic and wallet functions.

1. **Security & Performance:** The code is mostly written in **C++** to ensure maximum security and speed for heavy cryptographic operations like Key Derivation.
2. **Secure Interface (JNI):** The Core connects to the mobile app layer (Java/Kotlin) via strict, secure **C interfaces (JNI)**, ensuring private keys are never exposed in the riskier Kotlin/Java environment.



Esmaeetil Moradi

Senior Android Engineer



Multi-Chain Protocol Challenges

Architectural Analysis:

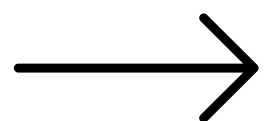
The Diverse Languages of the Blockchain

The Core's biggest challenge is managing protocols, as each has a different transaction structure and Fee system:



Esmaeetil Moradi

Senior Android Engineer

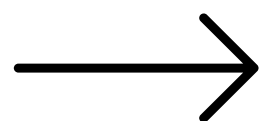


- **EVM Family (e.g., *ETH*, Polygon):**
 - **Model: Account-Based.**
 - **Fee Challenge:** Managing Gas and implementing algorithms for real-time price estimation.
- **UTXO Family (e.g., *BTC*, Zcash):**
 - **Model: UTXO** (Unspent Transaction Outputs).
 - **Fee Challenge:** Cost is based on transaction data size (Fee per Byte), requiring an Indexer to track unspent outputs.
- **Non-EVM / Modern (e.g., Cardano, Solana):**
 - **Model:** Highly unique structures.
 - **Challenge:** Requires the Core to implement alternative signing algorithms like **Ed25519**.



Esmaeetil Moradi

Senior Android Engineer



Handling Diversity: Separation of Concerns

For an architect, the solution is Separation of Concerns:

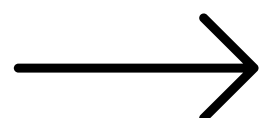
1. **Wallet Core (Native Layer): Function:** Transaction signing. (C++ / Rust)
2. **Protocol Adapter (Kotlin): Function:** Translation. Converts standard app requests into the specific protocol language (RPC). (Kotlin / Coroutines)
3. **Domain/UI Layer: Function:** Display. Requests data from the adapter, unaware of the underlying protocol.

WalletConnect: This system allows the mobile app to prepare the Core-required JSON structure for Multi-Chain signing (e.g., `trust_signTransaction`) and securely pass it to the Core.



Esmaeel Moradi

Senior Android Engineer



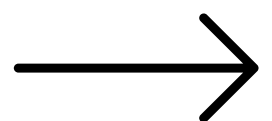
Final View: Future-Proof Architecture

This **separation of layers** ensures that our application is not only secure but is also scalable enough to adapt to any new blockchain innovation. This platform is the foundation for building future **Bank-Grade** financial services.



Esmaeetil Moradi

Senior Android Engineer



? Technical Question for Experts:

In building crypto financial infrastructure, do you prioritize implementing the critical **Core layer in Native (C++/Rust)** for maximum control over security and performance, or do you utilize higher-level libraries (like **Web3j**)?



Esmaeeil Moradi

Senior Android Engineer



System Design

From Code to Business Value

Architectural Insights

Let's Build a Scalable Future

by Esmaeel Moradi
Senior Android Engineer

Let's Connected