Ijma Wallet

High-Level Design Document [v0.1]

Executive Summary

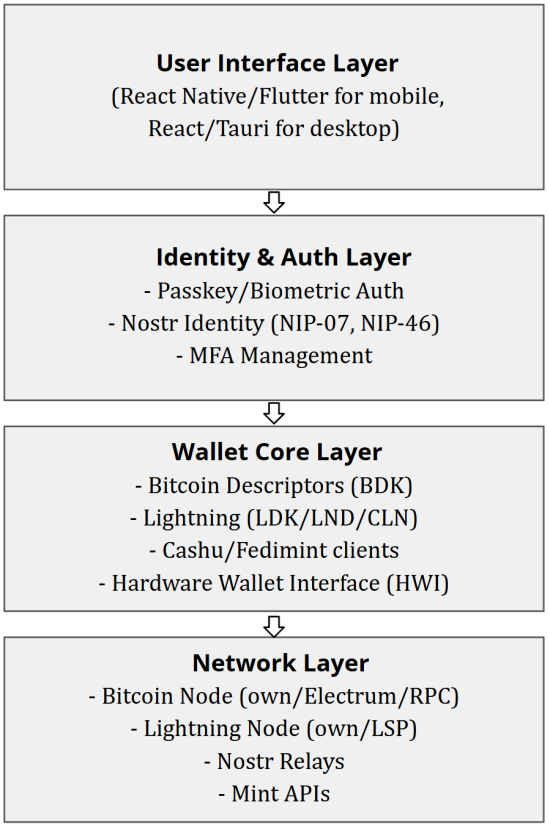
Ijma (Arabic: إجماع, meaning "consensus") is a self-custody Bitcoin/Lightning wallet that doubles as a decentralized identity solution through deep Nostr integration. It bridges traditional Bitcoin custody with emerging ecash systems (Cashu/Fedimint) while maintaining enterprise-grade security and consumer-friendly UX.

Core Philosophy: Consensus between security and usability, sovereignty and convenience, power user features and beginner-friendly design.

---

1. Architecture Overview

1.1 Technical Stack Layers



1.2 Key Technologies

- **Bitcoin**: BDK (Bitcoin Dev Kit) for on-chain wallet
- **Lightning**: LDK (Lightning Dev Kit) for embedded node or remote connection
- **Nostr**: NDK (Nostr Dev Kit) for identity and social features
- **Ecash**: Cashu-ts and Fedimint client SDKs
- **Hardware**: Bitcoin Hardware Wallet Interface (HWI)
- **Storage**: Encrypted local storage + optional cloud backup (encrypted)

---

2. Core Features

2.1 Bitcoin & Lightning Wallet

On-Chain Features

- Single-sig and multi-sig wallets (2-of-3, 3-of-5, custom)
- SegWit native, Taproot support
- UTXO management and coin control
- Custom fee selection (mempool.space integration)
- Batch transactions
- Replace-by-fee (RBF) and Child-pays-for-parent (CPFP)
- Hardware wallet signing support
- Watch-only wallets
- Descriptor-based wallet architecture

Lightning Features

- Non-custodial Lightning channels
- Lightning Service Provider (LSP) integration for liquidity
- Zero-conf channels option
- Submarine swaps (on-chain ↔ Lightning)
- Multi-path payments (MPP)
- LNURL-pay, LNURL-withdraw, LNURL-auth
- Bolt12 offers
- Channel management UI
- Force-close and recovery tools
- Lightning Address (username@ijma.app)

2.2 Nostr Identity Integration

Core Identity Features

- Nostr keypair as primary identity
- NIP-07 browser extension compatibility
- NIP-46 remote signing (Nostr Web Connect)

- Profile management (NIP-01, NIP-05)
- Contact lists and follows
- Direct messaging (NIP-04 encrypted, NIP-17 gift-wrapped)

Advanced Nostr Features

- **Nostr Web Connect (NWC)**: Control wallet from web apps
- **Zaps**: Lightning tips integrated with Nostr (NIP-57)
- **Social Recovery**: Contacts as recovery guardians
- **Verifiable Credentials**: NIP-based credential system
- **Web of Trust**: Reputation propagation through your network
- **Social Graph**: Visualize trust relationships
- **Decentralized Identity (DID)**: NIP-based DID implementation

2.3 Ecash Integration

Cashu (Chaumian Ecash)

- Multi-mint support
- Automatic mint discovery via Nostr
- Token management
- Send/receive via Nostr DMs or NFC
- Offline transaction capability
- Mint trust indicators
- Proof backup and recovery

Fedimint (Federated Ecash)

- Federation joining and management
- Guardian reputation (via web of trust)
- Lightning gateway selection
- Fedimint module support
- Consensus monitoring

2.4 Hardware Wallet Integration

Potential Supported Devices

- **Ledger** (Nano S, Nano X, Nano S Plus, Stax)
- **Trezor** (One, Model T, Safe 3)
- **Jade** (DIY and retail versions)
- **Coldcard** (Mk4, Q)
- **Passport** (Batch 2+)

Features

- Multi-device coordination for multi-sig
- Firmware verification

- Secure channel communication
- PSBT (Partially Signed Bitcoin Transaction) support
- Address verification on device
- Anti-tamper checking

2.5 Node Integration

Bitcoin Node

- Connect to own Bitcoin Core node (RPC)
- Electrum server support
- Public Electrum fallback
- Block explorer integration
- Node health monitoring
- Tor support for privacy

Lightning Node

- Connect to own LND/CLN/Eclair node
- Embedded LDK node option
- LSP marketplace for channel liquidity
- Node management tools
- Channel backup (Static Channel Backups)
- Watchtower support

---

3. Security Architecture

3.1 Authentication & Access Control

Multi-Factor Authentication

1. **Primary Factor**: Passkey (FIDO2/WebAuthn) or biometric
2. **Secondary Factors**:
    - TOTP (Time-based OTP)
    - Hardware security keys (YubiKey)
    - Email/SMS confirmation (optional, for low-risk operations)
    - Nostr-based social recovery

Biometric Authentication

- Face ID / Touch ID on iOS
- Fingerprint / Face unlock on Android
- Windows Hello / Mac Touch ID on desktop
- Biometric data never leaves device

3.2 Key Management

Key Generation & Storage

- BIP39 seed phrase (12/24 words)
- Optional passphrase (25th word)
- Encrypted storage using platform keychain
- Multiple encryption layers:
    - Device-level encryption
    - App-level encryption (AES-256-GCM)
    - Optional user password

Key Hierarchy

BIP39 Seed
├── BIP84 (Native SegWit): m/84'/0'/0'
├── BIP86 (Taproot): m/86'/0'/0'
├── Lightning: Custom derivation
├── Nostr Identity: NIP-06 derivation
└── Encryption Keys: Custom derivation

Multi-Sig Configuration

- **2-of-3 Standard**:
    - Key 1: Mobile device
    - Key 2: Hardware wallet
    - Key 3: Backup device or cloud-encrypted key
- **3-of-5 Advanced**:
    - Multiple hardware wallets
    - Multiple trusted devices
    - Social recovery keys (Nostr-based)

3.3 Recovery Mechanisms

Primary Recovery

1. **Seed Phrase Recovery**: Standard BIP39 restoration
2. **Passphrase Recovery**: Optional 25th word
3. **Multi-Sig Quorum**: Recover with M-of-N keys

Social Recovery (Nostr-based)

- Shard secret across trusted Nostr contacts
- Shamir's Secret Sharing (SSS)
- Threshold recovery (e.g., 3 of 5 friends)
- Encrypted shards with NIP-04/NIP-17
- Recovery initiation via Nostr events
- Guardian notification system

Advanced Recovery

- **Time-locked Recovery**: Emergency access after delay
- **Dead Man's Switch**: Automatic inheritance transfer
- **Legal Recovery**: Integration with estate planning
- **Multi-Device Sync**: Encrypted backup across devices

3.4 Transaction Security

Signing Process

1. Transaction construction
2. PSBT creation
3. Hardware wallet verification (if applicable)
4. Device biometric confirmation
5. Optional MFA for large amounts
6. Broadcast with privacy (Tor option)

Security Thresholds

- **< $100**: Biometric only
- **$100-$1000**: Biometric + PIN
- **$1000-$10,000**: Biometric + MFA
- **> $10,000**: Biometric + MFA + Hardware wallet
- *(User configurable)*

3.5 Privacy Features

- Tor integration for all network traffic
- Coin control and UTXO labeling
- Payjoin (BIP78) support
- CoinJoin integration (optional)
- Gap limit management
- Address reuse prevention
- Lightning private channels
- Nostr relay privacy tools

---

4. User Experience Design

4.1 Design Principles

1. **Progressive Disclosure**: Simple by default, powerful when needed
2. **Security Transparency**: Clear visual indicators of security state
3. **Trust Cues**: Show verification status at every step
4. **Guided Flows**: Onboarding that educates without overwhelming

5. **Adaptive Interface**: UI adjusts to user expertise level

4.2 UI Design System

Visual Language

- **Color Palette**:
    - Primary: Bitcoin Orange (#F7931A)
    - Secondary: Lightning Purple (#8B4CF7)
    - Nostr: Electric Blue (#00C3FF)
    - Success: Green (#00D98C)
    - Warning: Amber (#FFB800)
    - Error: Red (#FF3B30)
    - Background: Dark mode primary, light mode option
- **Typography**:
    - Headers: SF Pro Display (iOS), Roboto (Android)
    - Body: SF Pro Text / Roboto Regular
    - Monospace: JetBrains Mono (for addresses/seeds)
- **Components**:
    - Cards with soft shadows
    - Rounded corners (8-16px radius)
    - Micro-interactions and haptic feedback
    - Smooth transitions (200-300ms)
    - Glass-morphism for overlays

Layout Structure

[TO BE ADDED]

4.3 Key User Flows

First-Time Setup Flow

1. Welcome screen with value proposition
2. Security education (2-3 screens)
3. Choose setup method:
    - Create new wallet
    - Restore existing
    - Import from hardware wallet
4. Identity creation:
    - Generate Nostr keys
    - Set up username
    - Profile customization

5. Authentication setup:
    - Enable biometrics
    - Configure MFA
    - Set spending limits
6. Backup process:
    - Seed phrase display
    - Verification quiz
    - Optional social recovery setup
7. Success & wallet activation

Sending Bitcoin Flow

1. Home → Send button
2. Recipient input:
    - Bitcoin address
    - Lightning invoice
    - LNURL
    - Lightning Address
    - Nostr contact (via NIP-57 Zap)
    - QR scan
3. Amount entry:
    - BTC/Sats/Fiat toggle
    - Max button
    - Fee selection (slow/medium/fast/custom)
4. Review screen:
    - All details visible
    - Security indicators
    - Estimated confirmation time
5. Authentication:
    - Biometric
    - Hardware wallet (if configured)
    - MFA (if required by threshold)
6. Success animation
7. Transaction details with:
    - Confirmation status
    - Block explorer link
    - Share receipt option

Nostr Identity Management Flow

1. Profile tab
2. View/edit profile:
    - Avatar, banner
    - Name, about
    - NIP-05 verification
    - Lightning Address
3. Manage contacts:

    - Follow/unfollow
    - Trust score display
    - Contact categories
4. Web of Trust:
    - Visualize trust network
    - Reputation scores
    - Endorsements
5. Credentials:
    - View issued credentials
    - Verify others' credentials
    - Revoke credentials

4.4 Dashboard Design

Home Screen (Beginner Mode)
[TO BE ADDED]

Home Screen (Power User Mode)
[TO BE ADDED]

4.5 Learning & Inspiration

Best-in-class wallets to study:

- **Muun**: Excellent UX, simplified Lightning
- **Phoenix**: Non-custodial Lightning, great onboarding
- **BlueWallet**: Feature-rich, clean UI
- **Zeus**: Power user features, node management
- **Sparrow**: Desktop wallet, advanced features
- **Breez**: Lightning-first approach
- **Alby**: Nostr integration excellence
- **Damus**: Nostr UX reference
- **Primal**: Nostr wallet integration

UI/UX Patterns to Adopt:

- Smooth animations from Telegram
- Security indicators from banking apps
- Progressive disclosure from Stripe
- Micro-interactions from Apple apps
- Information architecture from Revolut

5. Advanced Features

5.1 Verifiable Credentials

Implementation

- Based on Nostr events (NIP-inspired)
- Cryptographic proof of claims
- Decentralized identifier (DID) integration
- Zero-knowledge proofs for privacy

Use Cases

- Age verification (without revealing birthdate)
- Accredited investor status
- KYC/AML compliance (privacy-preserving)
- Professional certifications
- Event tickets
- Membership badges

Credential Flow

1. Request credential from issuer
2. Issuer signs credential with their Nostr key
3. Store credential locally
4. Present credential when needed
5. Verifier checks signature against web of trust

5.2 Web of Trust & Social Score

Trust Algorithm

Trust Score = (Direct Trust × 0.5) +
(Network Trust × 0.3) +
(Activity Score × 0.2)

Direct Trust: Explicit endorsements from your contacts
Network Trust: Reputation through mutual connections
Activity Score: Positive interactions, zaps, credentials

Visual Representation

- Trust rings: concentric circles showing trust distance
- Color coding: Green (high trust) → Yellow → Red (low trust)
- Trust path visualization: Show how you're connected

- Reputation badges: Earned through verified actions

Applications

- Mint selection for Cashu
- Federation selection for Fedimint
- Nostr relay recommendations
- Lightning channel peer selection
- Transaction confirmation (social proof)

5.3 Advanced Security Features

Miniscript Support

- Custom spending conditions
- Time-locked transactions
- Complex multi-sig schemes
- Covenant-like behavior

Taproot Advanced Features

- Schnorr signatures
- MuSig2 for efficient multi-sig
- Taproot scripts for complex conditions
- Privacy improvements

Lightning Advanced

- Dual-funded channels
- Channel factories (when available)
- Splicing (add/remove liquidity)
- Trampoline payments
- Async payments (hold invoices)

5.4 Developer & Power User Tools

Built-in Developer Tools

- Console for debugging
- Raw transaction viewer
- PSBT inspector
- Descriptor analyzer
- Nostr event inspector
- Network traffic monitor

Advanced Settings

- Custom derivation paths
- RPC endpoint configuration

- Relay management
- Tor circuit isolation
- Custom fee algorithms
- Transaction broadcasting options

Plugin System

- Extensible architecture
- Community plugins marketplace
- Custom scripts for automation
- Integration with external tools

---

6. Technical Implementation

6.1 Platform Strategy

Mobile (Primary Platform)

- **iOS**: Swift/SwiftUI + Rust core
- **Android**: Kotlin + Rust core
- Shared Rust library for:
    - Bitcoin wallet logic
    - Lightning node
    - Cryptography
    - Nostr client

Desktop

- Electron or Tauri
- Shared web technologies
- Full feature parity with mobile

Web

- Progressive Web App (PWA)
- Limited features (security constraints)
- Focus on Nostr Web Connect

6.2 Key Dependencies

Bitcoin:
- BDK (Bitcoin Dev Kit)
- rust-bitcoin

Lightning:
- LDK (Lightning Dev Kit)

- LNURL library

Nostr:
- NDK (Nostr Dev Kit)
- nostr-tools

Ecash:
- cashu-ts
- fedimint-client

Hardware Wallets:
- HWI (Hardware Wallet Interface)

Security:
- Keychain (iOS/macOS)
- Keystore (Android)
- Secure Enclave integration

6.3 Performance Optimization

- Background sync with bloom filters
- Optimistic UI updates
- Lazy loading of transaction history
- Efficient UTXO indexing
- Channel state caching
- Nostr event pagination

6.4 Testing Strategy

Unit Tests

- Wallet logic
- Cryptographic operations
- Transaction construction
- PSBT handling

Integration Tests

- Node connectivity
- Hardware wallet interaction
- Multi-sig coordination
- Recovery scenarios

Security Audits

- Regular third-party audits
- Bug bounty program

- Responsible disclosure policy
- Open source security reviews

---

7. Roadmap

Phase 1: Foundation (TBC)

- [ ] Basic Bitcoin wallet (BDK integration)
- [ ] Seed phrase generation and recovery
- [ ] Basic Lightning (LDK embedded node)
- [ ] Nostr identity (key generation, profile)
- [ ] Biometric authentication
- [ ] Initial UI/UX implementation

Phase 2: Enhancement (TBC)

- [ ] Hardware wallet support (Ledger, Trezor, Jade)
- [ ] Multi-sig wallets (2-of-3)
- [ ] Advanced Lightning (LSP integration, channels)
- [ ] Nostr Web Connect (NWC)
- [ ] LNURL support
- [ ] MFA and advanced security

Phase 3: Integration (TBC)

- [ ] Cashu integration (multi-mint)
- [ ] Fedimint integration
- [ ] Own node connectivity (Bitcoin + Lightning)
- [ ] Verifiable credentials (basic)
- [ ] Web of Trust v1
- [ ] Social recovery

Phase 4: Advanced Features (TBC)

- [ ] Advanced multi-sig (3-of-5+)
- [ ] Taproot full support
- [ ] Plugin system
- [ ] Advanced privacy features
- [ ] Comprehensive credential system
- [ ] Full web of trust implementation

Phase 5: Ecosystem (TBC)

- [ ] Desktop applications

- [ ] Web platform
- [ ] API for third-party integrations
- [ ] Merchant tools
- [ ] Developer SDK
- [ ] Community features

---

8. Business & Compliance

- TO BE ADDED

---

9. Success Metrics

User Metrics

- Active wallets
- Transaction volume
- Lightning channel count
- Nostr identity adoption
- Average balance
- User retention (28-day, 90-day)

Technical Metrics

- Transaction success rate
- Lightning payment success rate
- Node uptime (for own node users)
- Hardware wallet connection success
- Average sync time
- Crash rate

Security Metrics

- Security incidents (target: 0)
- Successful recovery attempts
- Hardware wallet usage percentage
- Multi-sig adoption rate
- MFA enablement rate

Community Metrics

- GitHub stars
- Community contributions

- Plugin ecosystem size
- Educational content views
- Social mentions

---

10. Conclusion

Ijma Wallet represents a comprehensive vision for the next generation of Bitcoin wallets - one that embraces the full spectrum of Bitcoin innovation from base layer to Lightning to ecash, while integrating seamlessly with decentralized identity through Nostr.

Key Differentiators:

1. **Identity + Money**: First wallet to truly unify identity and payments
2. **Security without compromise**: Enterprise security meets consumer UX
3. **Ecosystem integration**: Works with everything in the Bitcoin/Nostr universe
4. **Progressive complexity**: Grows with user sophistication
5. **Community-driven**: Open source, transparent, user-owned

Product Vision: “Sovereign. Private. Halal.”

A non-custodial Bitcoin wallet that integrates Lightning, Nostr, Cashu, and Fedimint—built for users who value privacy, autonomy, and ethical finance. It’s not just a wallet; it’s a sanctuary for digital dignity.

Privacy-First Architecture

- No custody, no surveillance: Users hold their own keys. No tracking, no profiling.

- Cashu & Fedimint: Enable anonymous, Chaumian e-cash and community custody.

- Nostr integration: Decentralized identity and messaging—no centralized servers.

- Open-source transparency: Every line of code is auditable and forkable.

Shariah Compliance Foundations

- No Riba (interest): The wallet does not facilitate lending or borrowing with interest.

- No Gharar (excessive uncertainty): Clear fee structures, transparent routing, and no hidden risks.

- No Maysir (gambling): No speculative trading or high-risk financial instruments.

- Recognition of Maal (valuable property): Bitcoin is treated as a legitimate store of value and medium of exchange, aligning with scholarly views that affirm its halal status.

- Zakat & Sadaqah modules (future roadmap): Optional tools to calculate and donate directly from wallet.

The name "Ijma" (consensus) reflects the wallet's core mission: achieving consensus between competing priorities - security vs. usability, privacy vs. convenience, simplicity vs. power, individual sovereignty vs. social connection.

This wallet doesn't just store bitcoin - it becomes your sovereign digital identity and your gateway to the decentralized future.