IJMA WALLET

Architecture & Design Document

Version 0.3.1 · March 2026

IDENTITY · PAYMENTS · SAVINGS

Sovereign · Private · Halal

Built by Blockchainology | blockchainology.co.uk
ijmawallet.com | github.com/amisatoshi/ijmawallet

بسم الله الرحمن الرحيم

In the name of Allah, the Most Gracious, the Most Merciful

1. Overview

Ijma Wallet (Arabic: إجماع — consensus) is an open-source, non-custodial Bitcoin wallet built as a Progressive Web App (PWA). It unifies the full Bitcoin protocol stack — on-chain, Lightning Network, Cashu ecash, Fedimint community vaults, and Nostr decentralised identity — into a single interface designed for both first-time users and power users.

The name reflects the wallet's core design principle: achieving consensus between competing priorities — security and usability, sovereignty and convenience, technical depth and human accessibility.

1.1 Mission

Ijma is built for the digital Ummah and beyond — anyone who values financial sovereignty, privacy, and a halal approach to digital money. The mission of Blockchainology's Digital Hijrah framework is to help individuals and organisations transition away from extractive, interest-based, surveilled financial infrastructure toward open, censorship-resistant, permissionless alternatives.

1.2 Core Properties

Property	Description
Non-custodial	Your seed phrase and private keys never leave your device. No server ever sees them.
Open source	MIT licence. Every line of code is publicly auditable at github.com/amisatoshi/ijmawallet
No trackers	Zero analytics, telemetry, advertising, or user tracking of any kind
Installable PWA	Installs on iOS, Android, and desktop directly from the browser — no app store required
Shariah-compliant	No riba (interest), no gharar (deception), no maysir (gambling) by architectural design
Software product	Blockchainology is a software company. Ijma is not a financial product, money service business, or custodian

1.3 Product Context

Ijma Wallet sits within the broader Blockchainology product portfolio alongside My Legacy (Islamic estate planning PWA at mylegacywills.com) and the Digital Hijrah consulting and educational services. It is the flagship technical demonstration of Bitcoin sovereignty made accessible to a Muslim-majority audience globally.

2. Protocol Stack

Ijma implements a four-layer protocol architecture. The higher layers sit on top of and depend upon the lower layers. This mirrors the way the internet protocol stack works, but for Bitcoin-native sovereignty infrastructure.

Layer	Label	Technologies	Purpose
L4	Application	Ijma Wallet PWA (React 18)	User interface, UX flows, identity management
L3	Privacy & Identity	Cashu · Fedimint · Nostr	Private ecash, community custody, decentralised identity
L2	Lightning Network	LN routing · NWC · Boltz swaps	Instant payments, low-fee channels, atomic swaps
L1	Bitcoin	Bitcoin base layer · Electrum	Settlement, long-term savings, UTXO management

2.1 UX Framing: Payments vs Savings

Technical layer names are hidden from normie users. The UI presents two buckets that map onto human financial mental models:

Bucket	Composition
Payments (everyday spending)	Lightning Network balance + Cashu ecash proofs
Savings (long-term storage)	On-chain Bitcoin UTXOs + Fedimint community vault

Users can toggle Advanced View in Settings → Interface. In Advanced View, a fee estimate row appears on the Send confirm screen, and the home screen badge changes from "Simple" to "Pro". All tabs and features are visible in both modes. Full feature differentiation is planned for a future version.

3. Architecture

3.1 Application Type

Ijma is a Progressive Web App (PWA) built with React 18 and Vite 6. It runs entirely in the user's browser. There is no application server, no database, and no backend API owned by Blockchainology. All network requests go to open, public third-party APIs (block explorers, price feeds, Nostr relays) which the user can configure or replace.

3.2 Deployment Architecture

The app is deployed as a static site to Hostinger shared web hosting. The structure is:

```
public_html/
├── index.html              ← Landing page
├── terms.html              ← Terms of Service
├── privacy.html            ← Privacy Policy
```

```
├── disclaimer.html         ← Disclaimer
├── .htaccess               ← URL routing + security headers
└── app/                    ← PWA (contents of dist/ after npm run build)
    ├── index.html
    ├── manifest.webmanifest
    ├── sw.js               ← Workbox service worker
    └── assets/             ← Hashed JS/CSS bundles
```

GitHub Actions CI/CD pipeline automates the build and FTP deployment on every push to the main branch.

3.3 Source Code Structure

```
src/
├── lib/
│   ├── security.js         AES-256-GCM vault · WebAuthn · PBKDF2
│   ├── bitcoin.js          BIP39/32/84/86 · Electrum API
│   ├── nostr.js            NIP-06/04/57/05 · relay pool
│   ├── cashu.js            Mint/melt/send/receive · multi-mint
│   ├── swaps.js            Atomic swaps · Boltz · cross-mint
│   ├── hardware.js         WebUSB/HID · PSBT · air-gap
│   ├── identity.js         Passport · WoT · VCs · selective disclosure
│   └── providers.js        Block explorer/price/node registry
├── context/
│   ├── WalletContext.jsx   Global session state
│   └── NodeConfigContext.jsx  Network provider config
├── components/
│   ├── SplashScreen.jsx    Diagonal panel slide startup — three full-screen wipe
animations with onboarding image support and Islamic geometry fallback
│   ├── Onboarding.jsx      7-step wallet creation
│   ├── LockScreen.jsx      PIN keypad + biometric
│   ├── MainWallet.jsx      8-tab navigation shell
│   ├── icons.jsx           50+ flat SVG icon system
│   ├── normie.jsx          Scratch reveal · QR scanner · UX helpers
│   └── shared.jsx          Design tokens · Card · Badge · Toggle
└── screens/
    ├── HomeScreen.jsx      Portfolio · contacts · activity feed
    ├── IdentityScreen.jsx  Passport · WoT · Credentials · Present
    ├── SwapScreen.jsx      7-route atomic swap UI
    ├── HardwareScreen.jsx  Device browser · PSBT signing
    ├── AdvancedSettingsScreen.jsx  Node / provider config
    └── AllScreens.jsx      Send · Receive · Ecash · Settings
```

4. Security Model

4.1 Encryption and Key Storage

Data	Storage	Encryption	Notes
Seed phrase (24 words)	IndexedDB only	AES-256-GCM	Never transmitted. Never plaintext on disk.

Data	Storage	Encryption	Notes
Private keys	Memory (session only)	N/A	Derived at unlock. Cleared on lock or timeout.
PIN	localStorage	SHA-256 + 32-byte salt	Hash stored; PIN itself never stored.
Cashu proofs	IndexedDB	AES-256-GCM	Encrypted before write. Session-only access.
Nostr private key	IndexedDB	AES-256-GCM	Never published. Used for event signing only.
NWC connection string	Session only	N/A	Contains secret. Never persisted to storage.
LND/CLN macaroons	Session only	N/A	Credentials are session-only — not persisted.
Wallet settings	localStorage	Plaintext	Non-sensitive preferences only (explorer choice etc).

4.2 Cryptographic Parameters

Parameter	Value
Symmetric encryption	AES-256-GCM (NIST-approved)
Key derivation	PBKDF2-HMAC-SHA-256 · 600,000 iterations (OWASP 2023 minimum)
Salt	32 bytes (256-bit) random per operation via crypto.getRandomValues()
IV/Nonce	12 bytes (96-bit) random per operation
Key derivation (BIP32)	HMAC-SHA-512
Key derivation (Nostr)	BIP32 at m/44'/1237'/0'/0/0 (NIP-06)
Primitive source	Web Crypto API (OS-level, hardware-accelerated where available)
External crypto libs	@noble/hashes · @scure/bip32 · @scure/bip39 (all Cure53 audited)

4.3 Authentication Tiers

Transaction security is tiered based on amount and risk:

Tier	Threshold / Trigger
Tier 1 — Biometric	Payments under £10 equivalent (low-risk Lightning/Cashu)

Tier	Threshold / Trigger
Tier 2 — PIN required	Payments £10–£100 equivalent
Tier 3 — PIN + MFA confirmation	Payments £100–£1,000 equivalent
Tier 4 — Hardware wallet required	Payments over £1,000 equivalent (on-chain)

4.4 Session Management

- Auto-lock after 5 minutes of inactivity
- All private key material cleared from memory on lock
- Session state destroyed on browser tab close
- Panic mode: wallet wipe requires typing "DELETE" + PIN confirmation
- Encrypted backup export available — file cannot be decrypted without PIN

5. Feature Set

5.1 Wallet Foundation

Feature	Status	Version	Notes
BIP39 24-word seed generation	✅ Live	v0.1.0	OS entropy via crypto.getRandomValues()
BIP84 Native SegWit (P2WPKH)	✅ Live	v0.1.0	Derivation path m/84'/0'/0'
BIP86 Taproot (P2TR)	✅ Live	v0.1.0	Derivation path m/86'/0'/0'
Wallet restore from seed	✅ Live	v0.1.0	Import existing 12/24-word mnemonic
AES-256-GCM vault	✅ Live	v0.1.0	PBKDF2 600k iterations
PIN + WebAuthn biometric	✅ Live	v0.1.0	FIDO2 · device secure enclave
Seed phrase backup	✅ Live	v0.3.0	24-word tap-to-reveal grid. Words held in React state only, never persisted. PIN re-entry not required (uses active session).
Encrypted backup export	✅ Live	v0.3.0	JSON file · safe to store in cloud
Permanent wallet delete	✅ Live	v0.3.0	Requires PIN + "DELETE" confirmation
Hide balance toggle	✅ Live	v0.3.0	Eye icon · persists to next session
BIP39 word autocomplete	✅ Live	v0.3.0	Suggestions appear after 2 letters using the full 2048-word wordlist. Works on both restore entry and verify quiz screens.
Colour-coded address display	✅ Live	v0.3.0	Boilerplate prefix in grey, key checksum characters in orange, BOLT11 amounts in purple. Applied on Receive screen and Send confirm.

Feature	Status	Version	Notes
QR code generation (Receive)	✅ Live	v0.3.0	Canvas-based. Loads qrcodejs from CDN on first use. No npm dependency. Regenerates on address type change.

5.2 Bitcoin & Lightning

Feature	Status	Version	Notes
On-chain balance via Electrum	✅ Live	v0.1.0	Configurable: Mempool.space, Blockstream, self-hosted
Live fee rates (3 tiers)	✅ Live	v0.1.0	Fast/medium/slow from Mempool.space
Lightning via NWC	✅ Live	v0.2.0	Nostr Wallet Connect — works with Alby, Mutiny, etc.
Lightning via LND REST	✅ Live	v0.2.0	Own node — Umbrel, RaspiBlitz, Start9
Lightning via CLN REST	✅ Live	v0.2.0	Core Lightning + CLNRest plugin
Embedded Lightning node	🔜 v0.4.0	—	Breez SDK — non-custodial embedded LN
BTC price feed	✅ Live	v0.3.0	CoinGecko / Kraken / Bisq / self-hosted / none
QR code scanner (camera)	✅ Live	v0.3.0	BarcodeDetector API — Android Chrome

5.3 Cashu Ecash

Feature	Status	Version	Notes
Mint invoice (Lightning → ecash)	✅ Live	v0.1.0	Chaumian blind signatures — private by default
Melt token (ecash → Lightning)	✅ Live	v0.1.0	Redeem at any mint via Lightning invoice
Send ecash token	✅ Live	v0.1.0	Bearer token string — share any channel
Receive ecash token	✅ Live	v0.1.0	Swap for fresh proofs (prevents double-spend)
Multi-mint support	✅ Live	v0.1.0	Multiple mints with independent balances
Cross-mint atomic swap	✅ Live	v0.2.0	Via Lightning as bridge — trustless
Peer-to-peer offline transfer	🔜 v0.4.0	—	Via BLE/mesh — research phase

5.4 Atomic Swaps

Seven swap routes are implemented via src/lib/swaps.js:

Route	Mechanism
On-chain → Lightning	Submarine swap via Boltz HTLC (trustless)
Lightning → On-chain	Reverse submarine swap via Boltz HTLC (trustless)
Lightning → Cashu	Pay Lightning invoice at Cashu mint
Cashu → Lightning	Melt Cashu proofs via Lightning invoice
Lightning → Fedimint	Deposit via Fedimint Lightning gateway
Fedimint → Lightning	Withdraw via Fedimint Lightning gateway
Cashu → Cashu (cross-mint)	Melt at source mint, mint at destination via LN bridge

5.5 Hardware Wallet Support

Device	Connection
Blockstream Jade	USB Serial · Bluetooth · Air-gap QR
Coinkite Coldcard	USB · NFC (Mk4+) · SD card PSBT · QR (Q model)
Foundation Passport	USB · QR air-gap · SD card
SeedSigner	QR air-gap only (stateless signing device)
Krux	QR air-gap only (open-source DIY)
Ledger Nano S+/X/Stax	WebHID
Trezor One/T/Safe 3	WebUSB

5.6 Nostr Identity & Social

Feature	Status	NIP	Notes
Key derivation from BIP39 seed	✅ Live	NIP-06	One seed generates both Bitcoin and Nostr keys
Profile publish/fetch	✅ Live	NIP-01	kind-0 metadata events
Follow list management	✅ Live	NIP-02	kind-3 events — social graph source for WoT
Relay list	✅ Live	NIP-65	kind-10002 preferred relays
NIP-05 identity verification	✅ Live	NIP-05	DNS-based — user@domain verification
Encrypted DMs	✅ Live	NIP-04	AES encryption using shared Diffie-Hellman secret
Zap requests (Lightning tips)	✅ Live	NIP-57	LNURL-based Lightning zaps over Nostr
Nostr Wallet Connect	✅ Live	NIP-47	Connect any NWC-compatible Lightning wallet

Feature	Status	NIP	Notes
Gift-wrapped DMs	🔜 v0.4.0	NIP-17	Improved DM privacy via sealed events

5.7 Decentralised Identity (DID)

Implemented in src/lib/identity.js and src/screens/IdentityScreen.jsx across four tabs:

5.7.1 — Identity Passport (Tab 1)

Fetches and displays a complete Nostr-based identity passport from relays in real time. Shows:

- Display name, NIP-05 verification status (with live DNS check)
- Profile picture, Lightning address, website
- npub and nprofile (shareable encoded links)
- Follow count, active relay list (read/write status)
- Shareable identity QR and one-tap copy of npub/nprofile

5.7.2 — Web of Trust (Tab 2)

Live social graph analysis fetched from Nostr relays. Trust score algorithm:

Component	Weight + Source
Direct trust	50% — target pubkey appears in user's follow list (kind-3)
Network trust	30% — fraction of user's follows who also follow target (second-degree)
Activity score	20% — credential endorsements (kind-30079 events) referencing target

Contacts are displayed sorted by trust score with colour coding: green (≥70), amber (40–69), red (<40). Filterable by tier and searchable by name or npub.

5.7.3 — Verifiable Credentials (Tab 3)

Full issuer/holder/verifier triangle implemented using custom Nostr event kinds (30078–30080) and SHA-256 Merkle trees for selective disclosure.

Implemented credential schemas:

- AgeOver18 — confirms holder is 18+ without revealing birth date
- AccreditedInvestor — jurisdiction-aware accreditation status
- IdentityVerification — name, nationality, document type (individual fields selectively disclosable)
- Membership — organisation, member since, membership level
- BitcoinHolder — cryptographic proof of ownership above a threshold without revealing balance

Key privacy design: claim values are never published to Nostr relays. Only a SHA-256 Merkle root is public. The issuer transmits the full private data (values + salts) to the holder via NIP-04 encrypted DM.

5.7.4 — Selective Disclosure / Present (Tab 4)

Holder selects which claims to reveal from a stored credential. For each revealed claim: value, salt, and Merkle proof are provided. For concealed claims: only the leaf hash is shared. The verifier independently recomputes all hashes and verifies all Merkle proofs against the published root without contacting any server.

Supports optional challenge nonce (anti-replay) and verifier pubkey binding. Presentation output is a signed Nostr event (kind-30080) exportable as JSON or QR.

5.8 Shariah Mode

Enabled by default. Persists globally via WalletContext (same mechanism as Advanced View). Controls the following features:

Feature	Behaviour when enabled
Send confirm screen	Displays "Shariah compliant · No riba · Peer-to-peer transfer" badge
Swap screen	Each route shows a gharar compliance note. Trustless HTLC routes flagged as minimal gharar. Trusted routes (Cashu mint, Fedimint) flagged with operator verification warning.
Zakat Calculator	Unlocked. Calculates 2.5% nisab using 85g gold standard against live BTC price. Shows result in both GBP and sats.
Sadaqah module	Unlocked. Displays five verified Bitcoin-accepting Islamic charities with category filter. Includes disclaimer that Blockchainology has no affiliate relationship with any listed organisation.

6. UX Design

6.1 Design Philosophy

Ijma targets two distinct user archetypes simultaneously:

Mode	Target user and approach
Simple mode (default)	First-time Bitcoin users. Payments/Savings framing. Contacts not addresses. No technical jargon visible. Tap-and-go flows.
Pro mode	Shows fee estimates on Send confirm. Home badge changes to "Pro". All screens remain accessible in both modes. Full feature gating planned for a future version.

6.2 Design Tokens

Token	Hex	Role	Usage
Bitcoin Orange	F7931A	Primary accent	On-chain layer, savings, CTA buttons
Lightning Purple	8B4CF7	Secondary accent	Lightning layer, payments, Advanced View
Nostr Blue	0098D4	Identity	Nostr features, identity passport

Token	Hex	Role	Usage
Cashu Green	00A86B	Ecash / success	Ecash layer, success states, Shariah/Zakat features
Fedimint Indigo	4F46E5	Community vaults	Fedimint features, federation UI
Ijma Teal	4BAF92	Brand secondary	Sadaqah module, secondary brand elements
Warm White	FAF8F5	App background	Primary app background
White	FFFFFF	Card surface	Card backgrounds, elevated surfaces
Warm Grey 1	F3F0EB	Surface 2	Input backgrounds, secondary surfaces
Warm Grey 2	EDE9E2	Surface 3	Disabled states, deepest surface level
Warm Border	E5E0D8	Borders	All dividers, card outlines
Dark Text	1A1410	Primary text	All body text, headings
Mid Text	4A3F35	Secondary text	Subtitles, descriptions
Muted Text	9A8F83	Tertiary text	Placeholders, labels, metadata
Gold	C9A84C	Islamic / Luxury	Arabic typography, ornamental elements, landing page

6.3 Typography

Token	Value	Usage
Display font	Fraunces (variable, serif)	Arabic text, headings, balances, editorial elements
Body font	DM Sans	All UI labels, buttons, descriptions
Mono font	JetBrains Mono	Addresses, invoices, technical values, npub

Note (v0.3.1) The app previously used an obsidian/dark design (Obsidian `#0A0A0F` background, `#13131A` card surfaces, Cinzel/Cormorant Garamond typography). The design was migrated to a warm light palette in v0.3.0 to match the Arke wallet reference aesthetic.

6.4 Icon System

All UI icons are inline SVG — no network requests, works fully offline, scales perfectly at any pixel density. The icon system (src/components/icons.jsx) contains 50+ flat icons covering all wallet concepts, actions, social/identity, hardware, and status states. Emoji have been completely replaced throughout the UI.

6.5 Key UX Components

Component	Description and location
SplashScreen	4.5s animated startup. Three full-screen panels slide in diagonally using CSS clip-path wipe animations (cyberpunk anime style). Each panel shows a custom onboarding image

Component	Description and location
	(/public/images/onboarding-1/2/3.jpg). If images are absent, each panel falls back to an Islamic geometric SVG pattern in the corresponding accent colour (gold / purple / teal). Logo materialises over the final panel with three neon ring pings and a gold radial glow pulse. Typewriter tagline (IDENTITY · PAYMENTS · SAVINGS), Arabic blessing, and status line follow. Logo embedded as base64 WebP — no network request.
~~ScratchReveal~~	~~Canvas-based scratch-to-reveal for seed phrase backup. Gold layer must be actively scratched off each word — prevents accidental screenshot capture. Progress bar tracks all 24 words.~~
HintBubble	Dismissible first-run hints stored in localStorage. Four hints fire in sequence: Welcome, Payments vs Savings, Contact-based sending, Hide Balance. Human language, no jargon.
QrScanner	Camera-based QR scanner using BarcodeDetector API. Handles addresses, Lightning invoices, Nostr npubs. Graceful fallback for unsupported browsers.
BalanceSummary	Payments/Savings two-bucket display with per-layer breakdown. Animated hide/reveal with eye icon.
WalletBackupPanel	Encrypted backup export to file. Permanent delete with "DELETE" + PIN confirmation. Clear explanation of what the backup contains and how to restore.
ContactCard	"Contacts" framing throughout — never "Addresses". Each contact shows name, Lightning address, trust score badge, quick Zap and Send buttons.
SuccessScreen	Animated checkmark with amount, Arabic blessing, and 3.5s auto-dismiss.

7. Advanced Settings & Node Configuration

All external service dependencies are configurable by the user via the Advanced Settings screen (src/screens/AdvancedSettingsScreen.jsx). Configuration is managed by NodeConfigContext and persisted to localStorage (sensitive credentials are session-only and never persisted).

7.1 Configurable Services

Category	Default	Alternatives	Notes
Block explorer	Mempool.space	Blockstream, Bitaps, self-hosted	Live connection test shows block height + latency
BTC price feed	CoinGecko	Kraken, Bisq, self-hosted, None	None option removes all fiat conversion for maximum privacy

Category	Default	Alternatives	Notes
Fiat exchange rates	ECB (European Central Bank)	Frankfurter, Open Exchange Rates	Daily update, 17 currencies including NGN, IDR, MYR, BDT, PKR, SAR
Bitcoin node	Public Electrum fallback	Bitcoin Core RPC, Electrs, Fulcrum, BTCPay, Umbrel, Start9, RaspiBlitz, myNode	Own node means no third party sees address queries
Lightning backend	NWC (Nostr Wallet Connect)	LND REST, CLN REST, LNbits, Breez SDK (coming)	NWC works with Alby, Mutiny, Coinos, self-hosted
Tor routing	Disabled	Enabled (requires Orbot/Tor Browser)	Routes all requests via Tor; onion addresses used where available

7.2 Privacy Profile Summary

The Advanced Settings screen includes a live privacy profile table showing the privacy tier (HIGH / MEDIUM / LOW) of each currently selected service. This gives users an at-a-glance understanding of their current data exposure without requiring technical knowledge.

8. Web Presence

8.1 Landing Page (ijmawallet.com)

The landing page uses the same warm light design system as the app — #FAF8F5 background, white card surfaces, warm borders, Fraunces display serif, DM Sans body font, JetBrains Mono for technical elements. Gold (#C9A84C) is used for the hero Arabic إجماع title, ornamental dividers, and accent elements. The footer uses a dark ground (#1A1410) for contrast. The Islamic hexagonal geometric SVG pattern appears subtly in the hero section at 6% opacity.

8.2 Brand Identity

The Ijma Wallet logo features Kufi-style إجماع calligraphy with neon gold glow, the double gold ring, the

Islamic geometric rosette watermark, and three-colour tagline (Identity · Payments · Savings), giving it a built-in cyberpunk energy, and the tagline IDENTITY · PAYMENTS · SAVINGS in three colours (white · cyan · purple).

Brand mark: 1024×1024 PNG with transparent background. Optimised 115KB WebP version embedded in the SplashScreen component as base64.

8.3 Legal Pages

Page	Path and key contents
Terms of Service	/terms — 14 sections. Software-not-financial-product framing. Self-custody responsibilities. MIT licence. Risk warnings (7 specific risks). UK law (England and Wales).

Page	Path and key contents
Privacy Policy	/privacy — GDPR compliant. Data table confirming seed/keys/PIN never transmitted. No cookies, no analytics, no tracking. Server access logs 30-day retention only.
Disclaimer	/disclaimer — Software status table (audit pending). Shariah clarification (not a fatwa). Full list of applicable/non-applicable UK regulations (PSR 2017, EMR 2011, FCA, HMRC cryptoasset registration).

All three pages share the landing page's visual design system. Footer links to all three pages are on the landing page. Each legal page links to the others.

9. Dependencies

9.1 Runtime Dependencies

Library	Version	Purpose	Audit status
react + react-dom	18.3.1	UI framework	Meta — production grade
nostr-tools	2.7.2	Nostr events, relays, encoding, NIPs	Widely used
@cashu/cashu-ts	2.2.0	Cashu ecash mint/melt/send/receive	Cashu project — active
@scure/bip32	1.6.2	HD key derivation (BIP32)	Cure53 audited Jan 2022
@scure/bip39	1.5.4	Mnemonic generation + validation	Cure53 audited Jan 2022
@noble/hashes	1.6.1	SHA-256, HMAC, PBKDF2	Cure53 audited Jan 2022
bitcoinjs-lib	6.1.7	Bitcoin address encoding, transactions	Industry standard
noble-secp256k1	1.2.14	secp256k1 elliptic curve operations	Cure53 audited Jan 2022
idb	8.0.1	IndexedDB promise wrapper	Jake Archibald — Google
@nostr-dev-kit/ndk	2.11.0	Higher-level Nostr relay management	NDK team

9.2 Build Tools

Tool	Version and role
Vite	6.2.0 — Build tool and dev server

Tool	Version and role
vite-plugin-pwa	0.21.1 — PWA manifest and Workbox service worker generation
workbox-window	7.3.0 — Service worker lifecycle management
ESLint	9.19.0 — Code linting
Vitest	3.0.9 — Unit testing

10. Roadmap

10.1 Version History

Version	Date	Theme	Key deliverables
v0.1.0	Mar 2026	Foundation	BIP39/84/86 keys · AES-256-GCM vault · Cashu · Nostr NIP-06/04/57 · PWA · CI/CD · 6-tab UI
v0.2.0	Mar 2026	Swaps + Hardware	Atomic swaps (7 routes) · Hardware wallet support (7 devices) · Landing page · Advanced settings · Node config
v0.3.0	Mar 2026	Identity + UX	Warm light design system (full palette and typography migration) · BIP39 autocomplete on restore and verify screens · Seed phrase backup (tap-to-reveal grid) · Colour-coded address display (on-chain, Lightning, Nostr) · QR code on Receive screen · Shariah Mode (global state, Zakat calculator, Sadaqah module, swap gharar warnings) · Advanced View toggle · Diagonal panel splash screen with Islamic geometry fallback · Landing page redesign · Advanced Settings wired · Multiple bug fixes

10.2 Upcoming

Version	Target	Priority items	Notes
v0.4.0	Q3 2026	Breez SDK embedded Lightning · Fedimint WASM client · Social recovery (Shamir SSS via Nostr DMs) · NIP-17 gift-wrapped DMs	Lightning becomes fully self-custodial with embedded node
v0.5.0	Q4 2026	i18n — Arabic (RTL), Urdu, Bangla, Bahasa Indonesia/Malaysia · BC-UR animated QR for hardware air-gap · qrcode.react for Receive screen · Unit test suite for all lib/ modules	Internationalisation for target markets
v1.0.0	2027	Multi-sig 2-of-3 coordination · Taproot advanced scripting · Professional independent security audit · Bug bounty programme launch · App Store submission (Capacitor wrapper)	First production-grade release. Audit required before.

10.3 Future Research (Not Scheduled)

Area	Summary
BLE Mesh Network (Bitchat protocol)	Research complete. iOS limitation: Web Bluetooth API not available in Safari — requires Capacitor native wrapper. Android PWA can access Web Bluetooth for point-to-point. Full mesh requires native app. Enables offline Cashu transfers and PSBT relay without internet.
LoRa gateway integration	ESP32 + LoRa (TTGO T-Beam ~£30) as community mesh relay. 1–10km range. Combines with BLE for phone connectivity. Community infrastructure model via Blockchainology/Fedimint guardian services.
Nostr-based geolocation chat	Geohash-based local channels (Bitchat v1.3+ pattern). Location-aware payments and community coordination without revealing exact GPS.
Taproot Assets on Lightning	Stablecoin-like assets on Lightning channels for regions with high BTC volatility sensitivity.

11. Halal Business Model

Ijma Wallet is free and open source (MIT licence). Blockchainology's sustainable revenue model without custodial services:

Revenue stream	Description
Software licensing	Premium features or annual subscription for professional/enterprise users. Pure trade in software goods — unambiguously halal.
Professional services	Consulting, Bitcoin node setup, Fedimint federation deployment, sovereign infrastructure migration. Time-for-money — no financial product.
Enterprise white-labelling	Licensed Ijma codebase for Islamic banks, halal fintech companies, Muslim-majority financial institutions wanting branded sovereign wallet.
Educational content & certification	Bitcoin education courses, Digital Hijrah certification, Islamic finance scholar programmes. Knowledge exchange — pure services.
Hardware referrals (disclosed)	Transparent affiliate referrals to Jade, Coldcard, Passport hardware wallets. Disclosed commissions on genuine recommendations — halal provided honest.
Fedimint guardian services	Running reliable Fedimint guardian nodes for community federations. Service fee for uptime/reliability — closer to infrastructure provision than custody.

Explicitly avoided: anything resembling riba (staking yield, lending), custodial services, fractional reserve, speculative derivatives, undisclosed commissions.

Legal framing: Blockchainology is a software company (England and Wales). Ijma processes no transactions, holds no funds, and is not regulated under the Payment Services Regulations 2017, Electronic Money Regulations 2011, or as an FCA-authorised firm or HMRC cryptoasset exchange/custodian.

12. Deployment

12.1 Quick Start (Windows 11)

Prerequisites: Node.js 20 LTS (nodejs.org), Git for Windows (git-scm.com). No WSL required.

```
git clone https://github.com/amisatoshi/ijmawallet.git
cd ijmawallet
npm install
npm run dev      # → http://localhost:5173/app/
npm run build    # Produces dist/ folder for deployment
```

12.2 — Linux (Ubuntu / Debian)

Prerequisites: Node.js 20 LTS, Git.

```
# Install Node.js 20 via NodeSource
curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash -
sudo apt-get install -y nodejs git

# Clone and run
git clone https://github.com/amisatoshi/ijmawallet.git
cd ijmawallet
npm install
npm run dev        # → http://localhost:5173/app/
npm run build      # Produces dist/ folder
```

If you prefer Node version management:

```
# Install nvm
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.7/install.sh | bash
source ~/.bashrc
nvm install 20
nvm use 20
```

12.3 — macOS (Apple Silicon and Intel)

Prerequisites: Homebrew, Node.js 20 LTS, Git (included with Xcode Command Line Tools).

```
# Install Homebrew (if not already installed)
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

# Install Node.js
brew install node@20
echo 'export PATH="/opt/homebrew/opt/node@20/bin:$PATH"' >> ~/.zshrc
source ~/.zshrc

# Clone and run
```

```
git clone https://github.com/amisatoshi/ijmawallet.git
cd ijmawallet
npm install
npm run dev        # → http://localhost:5173/app/
npm run build      # Produces dist/ folder
```

If you prefer nvm on macOS:

```
brew install nvm
echo 'export NVM_DIR="$HOME/.nvm"' >> ~/.zshrc
echo '[ -s "$(brew --prefix nvm)/nvm.sh" ] && \. "$(brew --prefix nvm)/nvm.sh"' >>
~/.zshrc
source ~/.zshrc
nvm install 20
nvm use 20
```

Note: The `--minify-whitespace` esbuild flag must remain disabled in `vite.config.js` on all platforms. It corrupts template literals in the codebase. The build script is already configured correctly — do not add it.

12.4 Hosting

Ijma is a static PWA — shared hosting is entirely sufficient. No VPS, no Node.js process, no database required.

- Build: npm run build on local machine
- Upload contents of dist/ into public_html/app/ via File Manager or FileZilla FTP
- Upload landing page to public_html/index.html
- Upload legal pages to public_html/terms.html, privacy.html, disclaimer.html
- Create public_html/.htaccess with SPA routing rules and security headers
- Purge CDN cache in hPanel after every deployment
- Verify SSL certificate is active (free Let's Encrypt via hPanel → Security → SSL)

12.5 CI/CD Pipeline

GitHub Actions workflow (.github/workflows/ci.yml) automatically builds and deploys on every push to main. Required GitHub repository secrets:

```
HOST_FTP_HOST  — FTP hostname from hPanel
HOST_FTP_USER  — FTP username
HOST_FTP_PASS  — FTP password
```

13. Security Notice

> ⚠ *This document describes v0.3.0, a pre-release demonstration. Ijma Wallet has not yet undergone a professional independent security audit. The cryptographic primitives are sound and best-practice parameters are used throughout, but application-level vulnerabilities may exist. Do not store funds you cannot afford to lose. A professional audit by a qualified security firm is planned before v1.0.0. Responsible disclosure: ijma@blockchainology.co.uk*

Sovereign · Private · Halal