Semi-Custodial Wallet : Enterprise ready Wallet SDK

Project Name: Hodei Wallet
Version: 1.1
Last Update: 2026-01-20
Current Milestone: 1
Catalyst Link: https://milestones.projectcatalyst.io/projects/1400011
Public Github Repository: https://github.com/Cardano-Forge/hodei_wallet_f14
Approvals:

Team	Approved	Date
Development Team	**Dodilanne**	**2026-01-20**
Project Manager	**Mr. Abdibdi**	**2026-01-20**
Stakeholders	**Tqueri**	**2026-01-20**
Management	**Zachary Soesbee**	**2026-01-20**

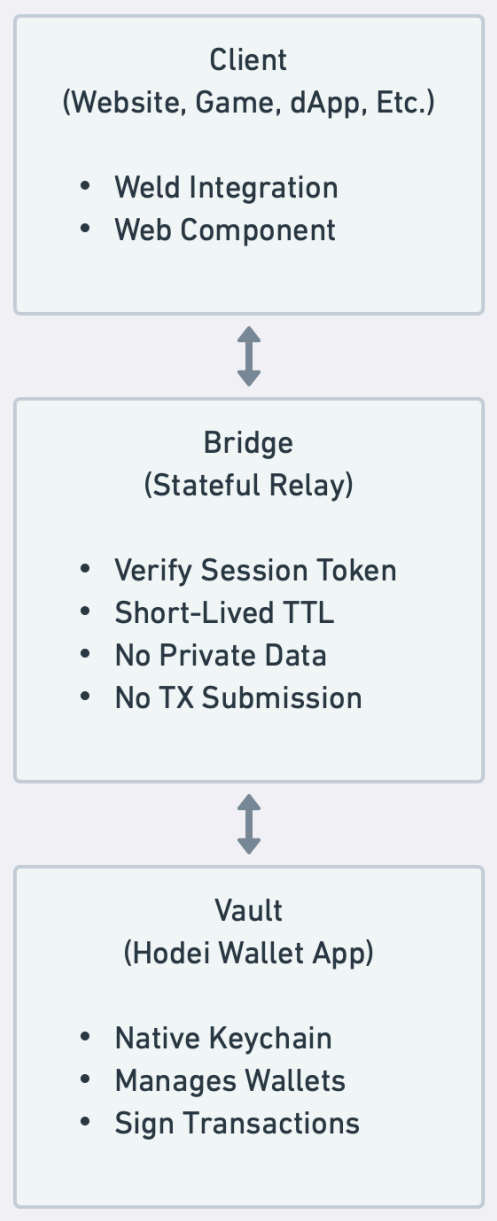
Changelog

Date	Comment
2026-01-20	More information about ***Authentication Partner(s)*** and ***Frontend Integration.*** (*Architecture Decision*) Added Preliminary ***Events***. (*Telemetry*)

Table of Content

System Overview

This solution is composed of 3 components, the ***Vault***, the ***Bridge*** and the ***Client*** Connector.



Hodei Wallet is designed for easy setup without complex seed phrase management. Built on a 3-component architecture, it integrates seamlessly with existing dApps and wallet connectors without requiring major changes.

Private keys are stored securely on the user's device using built-in secure storage (like *Apple's iCloud Keychain*) and backed up to the user's cloud when available, so only they can access them.

The keys remain isolated from dApps, as all communication is routed through a secure bridge. Users can then review and sign transactions directly from their mobile device using their device password or biometrics.

Components

Component	Description	Private Data
Vault	Native app (iOS, Android). Holds private keys in platform secure storage. Signs transactions.	Yes
Bridge	Stateful relay with KV Database. Stores unsigned TX temporarily, relay signature to frontend using Web Socket. Manages Session Tokens.	No
Client	Web Component/JS library embedded in third-party apps. CIP-30 fully compliant, drop-in replacement for browser extensions.	No

Communication Protocol

For this concept, we are going to implement Web Sockets between the ***Vault*** < > ***Bridge*** < > ***Client***.

Multi-Link Support

- One ***Vault*** can link to multiple ***Clients***
- dApps do not have visibility into other linked apps
- A ***Vault*** can view and manage linked ***Clients***.

Platform Considerations

Platform	Key Storage	Status
iOS	iCloud Keychain	In scope
Android	Encrypted Shared Preferences	In scope

Flows

Wallet

Linking Flow

1. User clicks "Connect Wallet" in ***Client***
   |
   ▼
2. ***Client*** requests linking session from ***Bridge***
   |
   ▼
3. ***Bridge*** generates PIN + session ID, returns to ***Client***
   |
   ▼
4. Web Component displays PIN on screen
   |
   ▼
5. User opens ***Vault***, taps "Link"
   |
   ▼
6. User enters PIN manually
   |
   ▼
7. ***Vault*** sends PIN + ***Vault*** ID + signature (to validate wallet ownership) to ***Bridge***
   |
   ▼
8. ***Bridge*** validates PIN, generates Session Token
   |
   ▼
9. ***Bridge*** returns Session to ***Client***

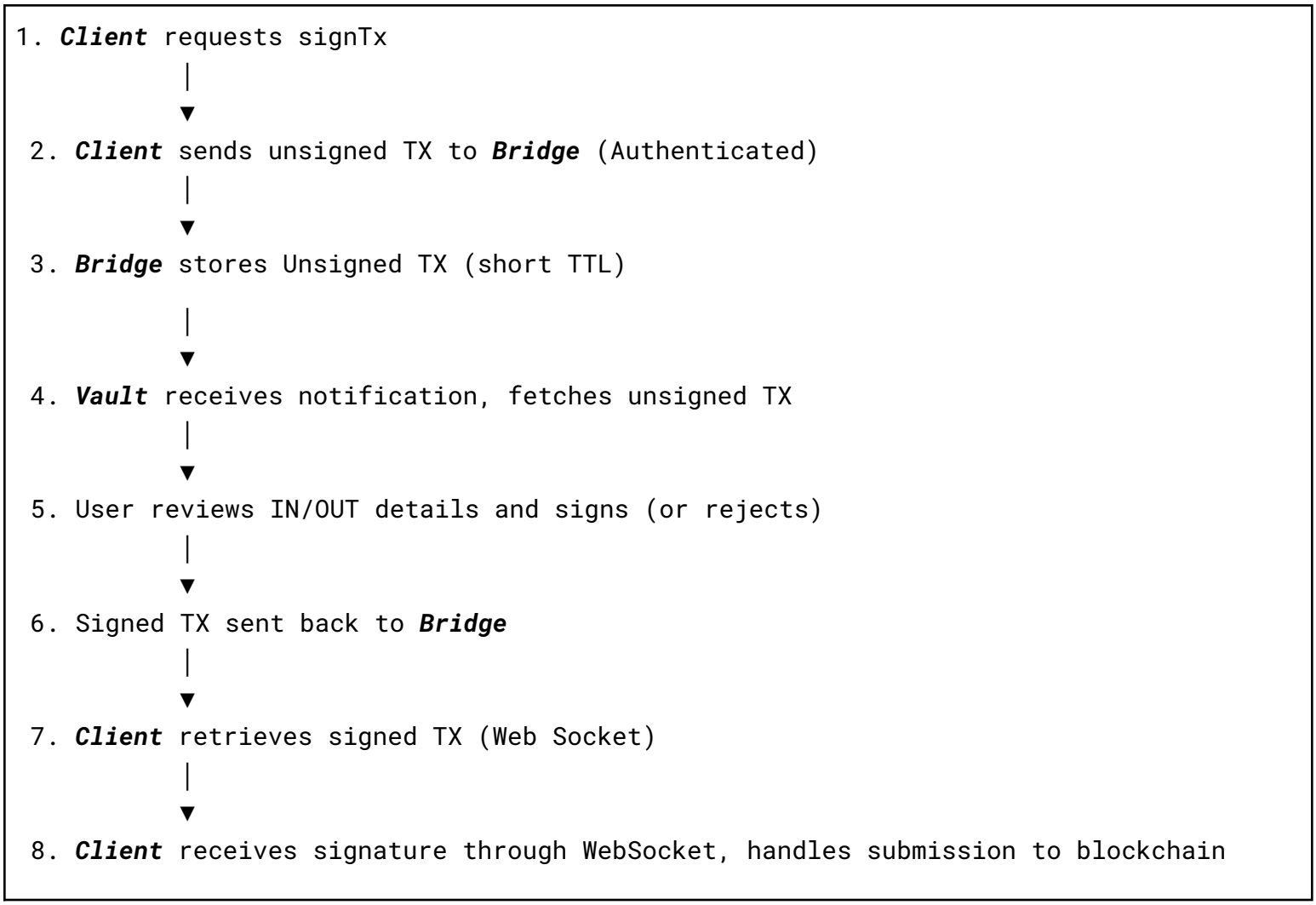
Unlinking Flow

1. User initiates unlink (from ***Vault or Client***)
   |
   ▼
2. Authenticated Request sent to ***Bridge***
   |
   ▼
3. ***Bridge*** removes device-app link
   |
   ▼
4. **Bridge** notifies and closes active ***Clients/Vault***
   |
   ▼
5. Both sides cleanup

Linking methods

- **Manual PIN entry**: Fallback for same-device scenarios (e.g., mobile browser + mobile app)

Transaction Signing Flow



Note: The ***Bridge*** does NOT submit transactions. Submission is always the ***Client***'s responsibility.

Data Classification

Sensitivity Levels

Level	Definition	Examples
Critical	Compromise leads to total loss of funds	Private keys, seed phrase
High	Compromise enables unauthorized actions	Session Token (can request signatures)
Medium	Exposure reveals transaction intent, no direct fund loss	Unsigned/signed TX
Low	Minimal impact, no financial risk	Device IDs, app metadata

Seed Phrase Management

Default behavior: Seed phrase is stored in platform keychain (Synced to cloud provider keychain when available). This approach targets non-crypto-enthusiast and non-tech-savvy users who benefit from automatic, secure backup without manual steps (When Available).

Export option: Users can export their seed phrase manually (e.g., write on paper). The wallet must display a confirmation alert warning that revealing the seed phrase is unsafe and should only be done for personal backup purposes.

Bridge Database

Storage: The recommended database is a KV Store.

Encryption at rest: Optional. Transaction data originates from and is submitted to a public blockchain, so encryption at rest provides limited additional security.

TTL and cleanup: Unsigned transactions should have a short TTL to minimize data retention.

Signatures and Retries: No retry is available for signed transactions as the client disconnected and the dApp will not necessarily keep/reconnect the hooks to process the signature.

Telemetry

Approach: Privacy-first. No PII stored.

Goals: Prevent abuse, improve our software and related services and to make design decisions for future releases.

Guidelines:

- No PII in telemetry data
- Hash IP addresses for rate limiting and attack detection (*Using selected cloud Provided network stack*)
- Define retention policies (TBD)

Data Inventory

Data	Location	Sensitivity	Encrypted at Rest	Encrypted in Transit
Private Keys	Vault (native keychain)	Critical	Yes (platform managed)	N/A (never leaves device)
Seed Phrase	Vault (native keychain)	Critical	Yes (platform managed)	N/A (never leaves device)
Seed Backup	Cloud (iCloud Keychain)	Critical	Yes (platform managed)	Yes (TLS, platform managed)
Session Token	Client(browser/app)	High	Platform dependent	Yes (TLS)
Unsigned TX	Bridge (database, short TTL)	Medium	Optional (see notes)	Yes (TLS)
Signed TX	Bridge (Relayed)	Medium	Optional (see notes)	Yes (TLS)
Device ID	Bridge	Low	No	Yes (TLS)
Linked Apps List	Bridge	Low	Platform dependent	N/A
Telemetry	Telemetry backend	Low	Recommended	Yes (TLS)

Key Management

Key Generation

Aspect	Requirement	Owner
Library	CSL (Cardano Serialization Library) or equivalent	Mobile Team
Seed phrase standard	BIP-39 mnemonic (24 words recommended)	Mobile Team
Key derivation	BIP-32/BIP-44 hierarchical deterministic (HD) wallet	Mobile Team
Cardano-specific	CIP-1852 derivation path for Cardano	Mobile Team

Key Storage

Aspect	Requirement	Owner
iOS	Keychain	Mobile Team
Android	Android Encrypted Shared Preferences	Mobile Team

Key Usage

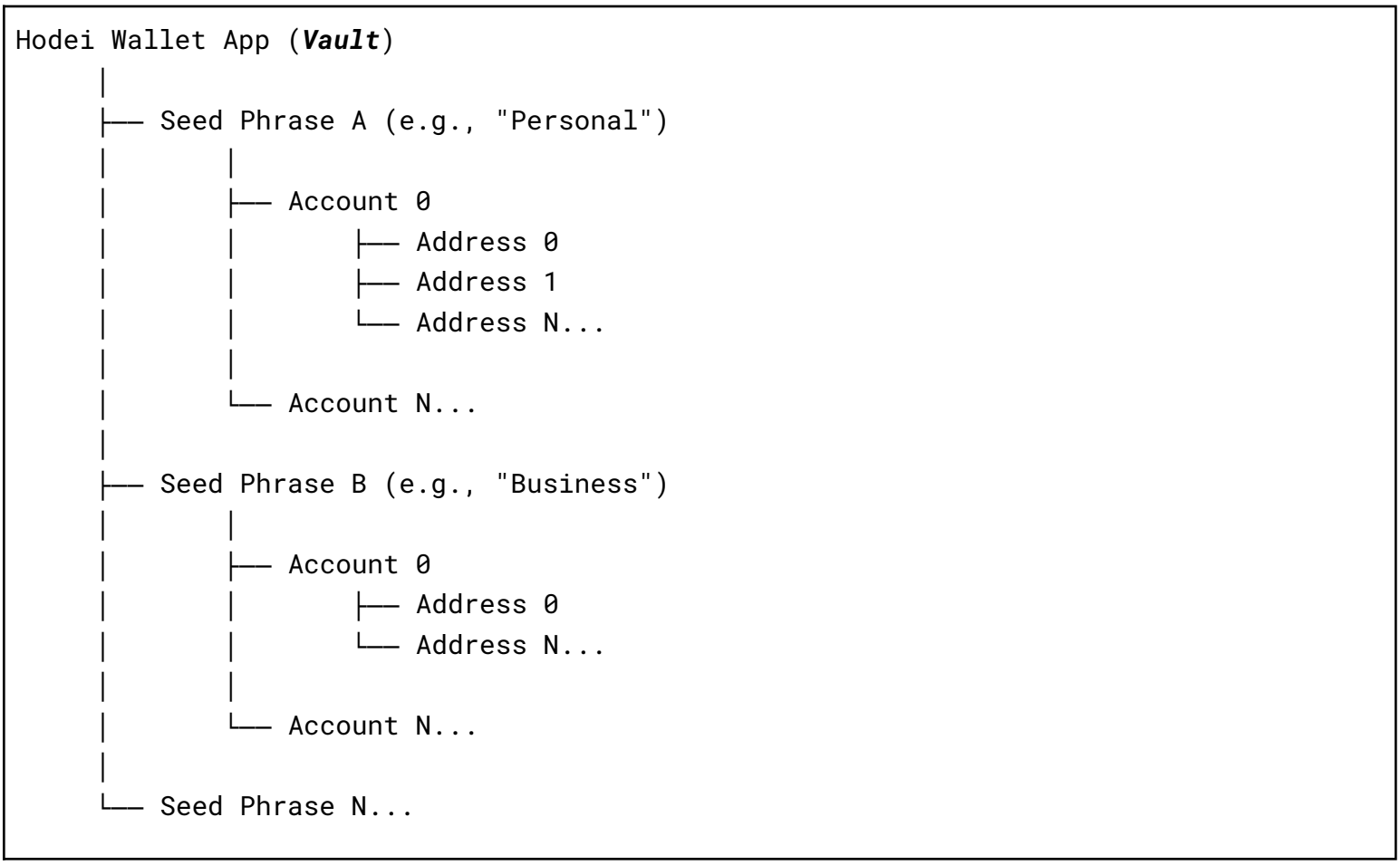
Aspect	Requirement	Owner
Access control	Biometric or PIN required before any key operation	Mobile Team
Memory handling	Load secrets only when needed, clear immediately after	Mobile Team
Key operations	Secrets used only for: wallet/account generation, TX signing, seed reveal	Mobile Team
No export	Private key never exported (only seed phrase can be revealed)	Mobile Team

Memory lifecycle

Operation	Secrets in memory	After operation
Wallet/account generation	Yes	Clear secrets, keep derived addresses only
Transaction signing	Yes	Clear secrets immediately after signing
Seed phrase reveal	Yes	Clear secrets after user dismisses screen
Normal wallet usage	No	Only public info (addresses, balances) in memory

Vault Structure

The ***Vault*** supports multiple seed phrases, each with hierarchical account management via HD derivation:



Feature	Description
Multi-seed	Users can add multiple seed phrases (create new or import existing)
Multi-account	Each seed can have multiple accounts
Multi-address	Each account can derive multiple addresses (per CIP-1852)
Independent backup	Each seed phrase has its own backup; must be backed up separately

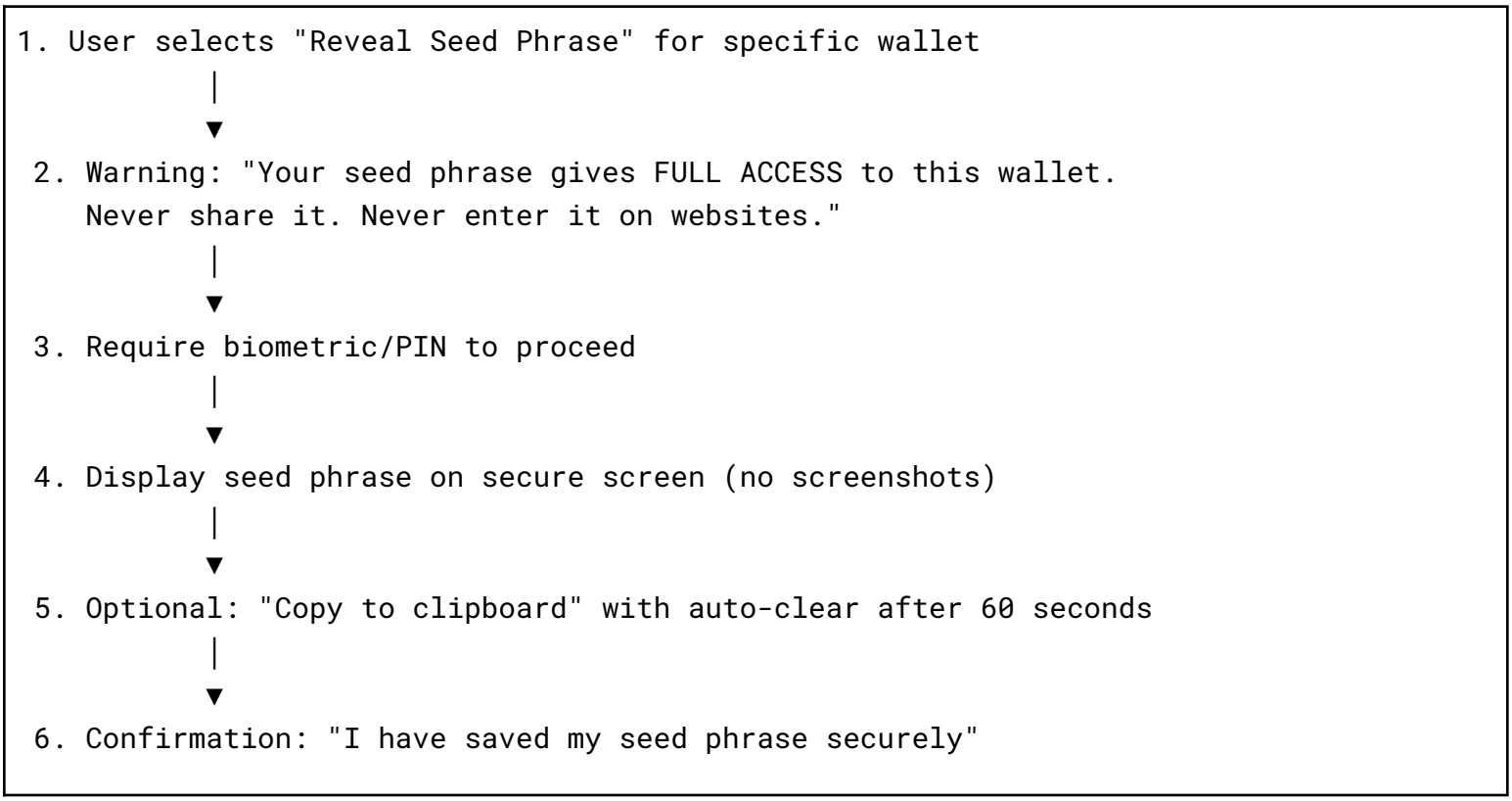
Import & Export

Import Seed Phrase

```
1. User selects "Import Wallet" or "Add Existing Wallet"
           |
           ▼
 2. User enters 24-word seed phrase
           |
           ▼
 3. Vault validates seed phrase (BIP-39 checksum)
           |
           ▼
 4. Vault derives accounts and addresses
           |
           ▼
 5. Seed stored in platform keychain
           |
           ▼
 6. Vault syncs balances from blockchain via Anvil API
```

Aspect	Requirement	Owner
Validation	Verify BIP-39 checksum before accepting	Mobile Team
Secure input	Prevent screenshots during seed entry	Mobile Team
No clipboard	Warn user if pasting (clipboard may be insecure)	Mobile Team

Export Seed Phrase (Reveal)



Aspect	Requirement	Owner
Warning	Clear warning about risks before revealing	Mobile Team
Authentication	Biometric/PIN required	Mobile Team
Secure display	Prevent screenshots/screen recording	Mobile Team
Clipboard	If allowed, auto-clear clipboard after short timeout	Mobile Team

Key Backup & Recovery

Aspect	Requirement	Owner
Manual backup	User can reveal/export each seed phrase separately (with warning)	Mobile Team
Recovery	Restore from platform backup (When Available) on new device, or import seed phrases	Mobile Team

Recovery Flow

```
1. User installs Vault on new device
          |
          ▼
 2. Platform automatically restores all seeds from cloud keychain (When Available)
    OR user selects "Import existing wallet"
          |
          ▼
 3. If importing: user enters 24-word seed phrase (repeat for each seed)
          |
          ▼
 4. Vault derives all accounts and addresses for each seed
          |
          ▼
 5. Vault syncs balances from blockchain via Anvil API
```

Key Destruction

Delete Single Seed Flow

```
1. User selects specific seed/wallet to delete
          |
          ▼
 2. Warning: "This will delete this wallet and all its accounts"
          |
          ▼
 3. Warning: "Have you backed up this seed phrase?"
          |
          ▼
 4. Final confirmation: "YOU WILL LOSE EVERYTHING in this wallet.
    Without this seed phrase, these funds are UNRECOVERABLE."
          |
          ▼
 5. Require biometric/PIN to confirm deletion
          |
          ▼
 6. Securely wipe this seed from keychain
          |
          ▼
 7. Other seeds remain intact
```

Delete All Data Flow

```
1. User selects "Delete All Wallets" or "Reset App"
           |
           ▼
 2. Warning: "This will delete ALL wallets and ALL accounts"
           |
           ▼
 3. Warning: "Have you backed up ALL your seed phrases?"
           |
           ▼
 4. Final confirmation: "YOU WILL LOSE EVERYTHING.
    Without your seed phrases, ALL funds are UNRECOVERABLE."
           |
           ▼
 5. Require biometric/PIN to confirm deletion
           |
           ▼
 6. Securely wipe all seeds from keychain
           |
           ▼
 7. Clear all local data (linked apps, preferences)
           |
           ▼
 8. Return to onboarding screen
```

Aspect	Requirement	Owner
Confirmation	Multiple explicit warnings before deletion	Mobile Team
Authentication	Biometric/PIN required to confirm	Mobile Team
Secure wipe	Use platform secure deletion APIs	Mobile Team
Granularity	Support deleting single seed or all data	Mobile Team

Key Lost / Compromised

Key rotation is not applicable. Seed phrases are permanent.

Scenario	Action
Seed compromised	User must create new seed and transfer funds from compromised wallet
Lost seed phrase	If platform backup exists, export seed from current device first

Important: Educate users that seed phrase compromise means ALL accounts/addresses derived from that seed are compromised. Other seeds in the app remain safe.

Telemetry & Privacy

Principles

Principle	Description
Privacy First	No PII collected by default.
Transparency	Clear disclosure of what is collected.
Minimal	Collect only what is needed for KPIs.
Backend Only	Data collected from the ***Bridge***

What is Collected ?

- Event Date (When the event occurred)
- ***Vault*** Version
- ***Bridge*** Version
- Source (Origin)
- Transaction Hash
- Transaction Size
- Cardano Fee
- Transaction Intent(s) (*Transfer Ada, SC Interactions, etc.*)
- Transaction Inputs/Outputs (*Preparing the prevention system*)
- Transaction State (Accepted, Rejected)
- Event Type
- TBD.

Events:

- Vault Pairing (HTTP POST)
- Unsigned
- Signed
- Unlink
- Link

Retention Policy

Data Type	Retention	Owner
Logs	30 days (Subject to change)	Backend Team
Traces	15 days (Subject to change)	Backend Team
Metrics	1 year (Subject to change)	Backend Team

Note: The retention policy is subject to change due to compliance requirements.

Threats and Security Review

Threats

This document details identified threats, their attack vectors, impact assessments, and required mitigations.

Severity Legend

Severity	Definition
Critical	Total loss of funds or complete system compromise
High	Significant financial loss or unauthorized actions
Medium	Limited exposure, no direct fund loss
Low	Minimal impact, informational exposure

Platform Security Parity Notice

This threat model assumes platform-provided secure storage with cloud sync.

iOS provides turn-key solutions for both seed phrase storage and metadata sync. Whereas Android does not have direct equivalents.

Current Plan for Wallet SDK Development:

The initial Proof-of-Concept (POC) and Minimum Viable Product (MVP) will prioritize the iOS platform for the automated Sync and Recovery features using built-in iCloud Keychain.

For Android, the secret keys will be stored locally on the device. Consequently, in this first version, Android users will be responsible for manually backing up their keys.

List of Identified Threats

List of identified threats, with attack vectors, impacts and mitigations.

Private Key Exfiltration

Description: An attacker gains access to private keys stored on the device or in cloud backup.

Attack Vectors:

Vector	Description	Likelihood
Malware on device	Keylogger or memory scraper captures keys	Medium
Compromised cloud account	Attacker accesses iCloud/Google account	Medium
Physical device theft	Attacker has unlocked device	Low
Backup extraction	Keys extracted from unencrypted backup	Low

Impact: Critical (total loss of funds)

Mitigations:

Mitigation	Owner	Status
Store keys in Keychain / Secured keystore	Mobile Team	Required
Require biometric or PIN to access keys	Mobile Team	Required
Keys only loaded in memory during signing or seed reveal, cleared immediately after	Mobile Team	Required
Never log, debug, or print key material in any environment	All Teams	Required
Cloud backup uses platform encryption (iCloud Keychain, etc.)	Platform (Apple/Google)	Inherited

Refer to official Apple documentation in regards of how the data is handled when using iCloud Keychain: https://support.apple.com/en-us/102651

Unauthorized Transaction Signing

Description: An attacker triggers transaction signing without legitimate user intent.

Attack Vectors:

Vector	Description	Likelihood
Stolen Session	Attacker obtains Session from browser storage, sends signing requests	Medium
Compromised dApp	Malicious dApp sends TX that looks legitimate but drains wallet	Medium
Session hijacking	Attacker takes over linked session	Low

Impact: Critical (loss of funds via signed malicious transaction)

Mitigations:

Mitigation	Owner	Status
Require biometric/PIN confirmation for every signature	Mobile Team	Required
Display clear TX details (IN/OUT, recipient, amount) before signing	Mobile Team	Required
Rate limit signing requests (e.g., max 1 TX per N seconds)	Backend Team	Required
Session rotation	Backend Team	Required
User can reject/cancel any TX from ***Vault***	Mobile Team	Required

Telemetry Data Leakage

Description: Telemetry or logging inadvertently captures sensitive data that could identify users or expose transaction patterns.

Attack Vectors:

Vector	Description	Likelihood
Accidental PII logging	Developer logs user IP or device identifiers	Medium
Telemetry correlation	Combining metrics to profile users	Medium
Third-party telemetry breach	External telemetry provider is compromised	Low
Log file exposure	Logs stored insecurely or accessible publicly	Low

Impact: Medium (privacy breach, potential targeted attacks)

Mitigations:

Mitigation	Owner	Status
No PII in telemetry	All Teams	Required
Hash IP addresses for rate limiting / attack detection	Backend Team	Required
Define and enforce retention policies	Backend Team	Required
Code review checklist: verify no sensitive data in logs	All Teams	Required
Use trusted telemetry stack	Backend Team	Recommended

Cloud Storage Compromise

Description: An attacker gains access to cloud-stored wallet data (seed phrase backup).

Attack Vectors:

Vector	Description	Likelihood
Compromised cloud account	Attacker gains access to user's Cloud account	Medium
Cloud provider breach	Cloud Infrastructure compromised	Very Low
Weak account security	User has no 2FA, weak password	Medium
Social engineering	Attacker tricks user into revealing cloud credentials	Medium

Impact: Critical (seed phrase exposed = total loss of funds)

Mitigations:

Mitigation	Owner	Status
Use platform keychain with encryption	Mobile Team	Required
Seed phrase encrypted before cloud sync	Platform (Apple iCloud Keychain)	Inherited
Display warning that no one will ever request seed phrase (not even for support, debugging, testing, or validation)	Mobile Team	Required
Encourage users to enable 2FA on cloud accounts	Mobile Team (UX)	Recommended
Display security tips during onboarding	Mobile Team	Recommended

Man-in-the-Middle Attacks

Description: An attacker intercepts communications between components.

Attack Vectors:

Vector	Description	Likelihood
TLS stripping	Downgrade HTTPS to HTTP	Low
Compromised CA	Attacker obtains fraudulent certificate	Very Low
DNS hijacking	Redirect Bridge/API domain to attacker server	Low
Malicious proxy	Corporate/public WiFi intercepts traffic	Medium
Unsigned TX tampering	Modify TX in transit before signing	Medium

Impact: High (transaction tampering, credential theft)

Mitigations:

Mitigation	Owner	Status
TLS everywhere	All Teams	Required
Enforce authenticated requests	Backend Team	Required
User reviews TX details on wallet before signing	Mobile Team	Required
HSTS headers on Bridge server and Anvil API	Backend Team	Required

Private Key Extraction (Technical Methods)

Description: An attacker uses technical tools to extract private keys from device storage, memory, or backups.

Attack Vectors:

Vector	Description	Likelihood
Memory dump	Extract keys from RAM during/after signing	Low
Debug tools	Attach debugger to running app, inspect memory	Low
Device forensics	Extract data from seized/stolen device	Low
Backup file analysis	Parse unencrypted local backups	Medium
Jailbreak/root exploits	Bypass OS protections to access keychain	Low

Impact: Critical (total loss of funds)

Mitigations:

Mitigation	Owner	Status
Clear keys from memory immediately after use	Mobile Team	Required
Disable sensitive operations in debugger environments (production builds)	Mobile Team	Required
Store seed phrase in platform secure storage with cloud sync (when available)	Mobile Team	Required
Exclude sensitive data from local backups	Mobile Team	Required
Obfuscate release builds	Mobile Team	Recommended

Platform secure storage:

Platform	Storage Method	Cloud Sync	Status
iOS	iCloud Keychain	Native	Ready to use
Android	Encrypted Shared Preferences	Research required	Research

Note for developers: iOS provides a turn-key solution with iCloud Keychain. Android, Linux & Windows do not have a direct equivalent. Developers must research and validate the best approach to achieve similar functionality.

Seed Phrase Compromise Through Telemetry or Logging

Description: Seed phrase accidentally exposed via logs, crash reports, or telemetry.

Attack Vectors:

Vector	Description	Likelihood
Debug logging	Developer accidentally logs seed during testing	Medium
Crash reports	Seed in memory included in crash dump	Low
Third-party SDKs	Analytics/crash SDK captures screen or memory	Low
Console output	Seed printed to console in debug builds	Medium

Impact: Critical (total loss of funds)

Mitigations:

Mitigation	Owner	Status
Never log seed phrase in any environment	All Teams	Required
Redact sensitive fields in crash reports	Mobile Team	Required
Audit third-party SDKs for data collection practices	Mobile Team	Required
Code review checklist: verify no seed/key logging	All Teams	Required
Prevent screenshots/screen recording on seed reveal screen	Mobile Team	Required
Static analysis for sensitive data patterns	All Teams	Recommended

Note on screenshot prevention: iOS and Android both support preventing screenshots on specific screens.

Transaction Tampering or Replay Attacks

Description: Attacker modifies transaction content before signing, or replays a previously signed transaction.

Attack Vectors:

Vector	Description	Likelihood
TX modification on Bridge	Attacker compromises ***Bridge***, modifies unsigned TX	Low
Replay attack	Resubmit previously signed TX to drain funds	Low (blockchain handles)
dApp manipulation	Malicious dApp crafts TX that looks benign but drains wallet	Medium

Impact: Critical (loss of funds)

Mitigations:

Mitigation	Owner	Status
User must review TX details before signing	Mobile Team	Required
Display human-readable TX summary (IN -> OUT, who gets what)	Mobile Team	Required
Cardano blockchain uses UTxO model (natural replay protection)	Blockchain	Inherited
TX has TTL (expires after slot)	Blockchain	Inherited

TX Display Requirements:

This wallet is designed for everyone. The transaction review screen must:

Mode	Description	Default
Simple view	Friendly summary: what you send (IN), what you receive (OUT), who gets what	Yes

Phishing Attacks Mimicking Wallet Interface

Description: Attacker creates fake wallet interface (fake app, fake linking page, fake dApp) to steal credentials or trick users into signing malicious transactions.

Attack Vectors:

Vector	Description	Likelihood
Fake mobile app	Attacker publishes lookalike app on app store	Medium
Fake linking page	Phishing site mimics ***Client*** UI (Web Component / Weld Integration)	Medium
Fake dApp	Malicious dApp pretends to be legitimate service	High
Social engineering	Attacker contacts user pretending to be support	Medium

Impact: Critical (seed phrase theft, malicious transaction signing)

Mitigations:

Mitigation	Owner	Status
Official app only from verified app stores	Mobile Team	Required
Educate users: Anvil will never ask for seed phrase	All Teams	Required
Clear branding in wallet to distinguish from fakes	Mobile Team	Required
Signing always happens in official wallet app (never in ***client/browser***)	Architecture	Inherited

Supply Chain Compromise of Wallet Dependencies

Description: Attacker compromises a third-party library or dependency used by the ***Vault***, SDK(s), or ***Bridge*** to inject malicious code.

Attack Vectors:

Vector	Description	Likelihood
Compromised package	Malicious code in dependency update	Medium
Typosquatting	Developer installs similarly-named malicious package	Low
Compromised CI/CD	Attacker injects code during build process	Low
Abandoned dependency takeover	Attacker takes over unmaintained package	Low

Impact: Critical (could exfiltrate keys, modify transactions)

Mitigations:

Mitigation	Owner	Status
Lock dependency versions	All Teams	Required
Regular dependency audits	All Teams	Required
Minimize dependencies, especially for crypto operations	All Teams	Required
Review dependency changes in PRs	All Teams	Required
Use Dependabot or similar for security alerts	All Teams	Recommended
Sign releases and verify signatures	All Teams	Recommended

Cloud Backup Failure

Description: User's seed phrase backup fails silently, leading to total loss of funds when they need to recover on a new device.

Attack Vectors:

Vector	Description	Likelihood
Silent backup failure	Cloud sync fails without user notification	Medium
Platform account issues	User's Cloud account has storage or sync issues	Low
Backup corruption	Backup data becomes corrupted during sync	Low
Account migration	User changes platform account, backup not transferred	Medium

Impact: Critical (total loss of funds if device is lost/replaced)

Mitigations:

Mitigation	Owner	Status
Verify backup exists after wallet creation	Mobile Team	Required
Display backup status indicator in wallet settings	Mobile Team	Required
Prompt user to verify backup periodically	Mobile Team	Recommended
Provide manual backup option (view seed phrase) as fallback	Mobile Team	Required
Warn user if backup verification fails	Mobile Team	Required

Note: Users should be encouraged to verify their backup works by confirming they can access their seed phrase from the wallet. The wallet should make this easy to check without exposing the seed phrase unnecessarily.

Seed Phrase Theft During Recovery

Description: Attacker tricks users into revealing their seed phrase during the recovery process on a new device, through social engineering or fake recovery flows.

Attack Vectors:

Vector	Description	Likelihood
Fake recovery app	User downloads fake wallet app that harvests seed phrase	Medium
Phishing during recovery	Attacker sends fake “verify your wallet” emails/messages	High
Fake support	Attacker poses as Anvil support requesting seed for recovery	Medium
Malicious website	Fake recovery page asks user to enter seed phrase	High

Impact: Critical (total loss of funds)

Mitigations:

Mitigation	Owner	Status
Educate users during onboarding: seed phrase is like a password	Mobile Team	Required
Display warning: no one will ever ask for seed phrase (see list below)	Mobile Team	Required
Recovery only happens in official app downloaded from verified app store	Mobile Team	Required
Seed phrase entry screen shows security warnings	Mobile Team	Required
No seed phrase entry via ***Client*** / Web Browser (***Vault*** only)	Architecture	Inherited

User education points:

- Your seed phrase controls your funds. Anyone with it can steal everything.
- **No one will ever ask for your seed phrase. Not Anvil. Not Apple. Not Google. Not Microsoft. Not support. Not for recovery. Not for verification. Not for debugging. Never.**
- Only enter your seed phrase in the official Anvil wallet app.
- If anyone asks for your seed phrase, it is a scam. Report it.

Trust Boundaries

Trust boundaries define where validation and verification must happen. Data crossing a boundary should not be trusted without verification.

Component Clarification

Component	Purpose	Lifetime
Linking Web Component	Displays PIN for wallet linking flow	Temporary
Weld SDK	CIP-30 compliant API, handles all wallet interactions	Persistent (runs in dApp context)

The *Linking Web Component* is only used during the initial wallet linking. After linking, the ***Client*** reads the Session Token from browser storage and communicates directly with the ***Bridge*** for all CIP-30 operations.

The ***Client*** includes the custom code to interact with the ***Bridge***, this code should be added as a “plugin”, so it can be used outside *Weld* as well.

Boundary Definitions

From	To	Validation Required
Client (dApp, etc.)	SDK	SDK runs in dApp context (In the `window.cardano`); dApp must not tamper with Session Token
Client (SDK)	Bridge	Session Token authentication; TLS; rate limiting
Bridge	Vault	Session Token authentication

What Each Zone Trusts

Zone	Trusts	Does NOT Trust
Vault (Trusted)	Platform keychain, user input	***Bridge*** content; unsigned TX (user must manually review)
Bridge (Controlled)	Valid Session Token holders	TX content; dApp identity beyond Session Token
SDK (Semi-trusted)	Bridge responses (over TLS)	dApp code (Session Token stored in browser, dApp could access)
Client (Untrusted)	Nothing assumed	Must authenticate via Session Token to Bridge

Critical Trust Boundary: User Review

The most critical trust boundary is the **user themselves**. The wallet displays transaction details (IN/OUT) and the user must verify before signing. No technical control can prevent a user from signing a malicious transaction if they approve it.

Wallet responsibility: Display clear, human-readable transaction summaries.

Risk Matrix

This section provides a summary view of all identified threats for prioritization.

Priority Definitions

Priority	Definition	Action
P1	Critical risk, must address before launch	Required mitigations, no exceptions
P2	High risk, should address before launch	Required mitigations, exceptions need approval
P3	Medium risk, address in early releases	Recommended mitigations
P4	Low risk, address when convenient	Optional mitigations

Threat Summary

Threat	Impact	Likelihood	Priority	Primary Owner
Private Key Exfiltration	Critical	Medium	**P1**	Mobile Team
Unauthorized Transaction Signing	Critical	Medium	**P1**	Mobile + Backend
Telemetry Data Leakage	Medium	Medium	P3	All Teams
Cloud Storage Compromise	Critical	Medium	**P1**	Platform (inherited)
Man-in-the-Middle Attacks	High	Low-Medium	P2	Backend Team
Private Key Extraction (Technical)	Critical	Low	P2	Mobile Team
Seed Phrase in Telemetry/Logs	Critical	Medium	**P1**	All Teams
Transaction Tampering/Replay	Critical	Medium	**P1**	Mobile + Blockchain
Phishing Attacks	Critical	Medium-High	**P1**	All Teams
Supply Chain Compromise	Critical	Medium	**P1**	All Teams
Cloud Backup Failure	Critical	Medium	**P1**	Mobile Team
Seed Phrase Theft During Recovery	Critical	High	**P1**	Mobile Team

Accepted Risks

Risk	Rationale
Jailbroken/rooted devices	User responsibility; Anvil not liable
Cloud provider breach	Extremely low likelihood; trust Cloud Providers
User approves malicious TX	Cannot prevent; mitigate with clear TX display

Architecture Decisions

Our Proof of Concept will be developed using the following core technologies:

Vault

- **Mobile Development:** Flutter
- **Secure Storage:** flutter_secure_storage
- **Wallet & Transaction Management:**
  - cbor
  - catalyst_key_derivation
  - catalyst_cardano_serialization
  - bip39
- **Targeted Platforms:** iOS and Android
- **Authentication Partner(s):**
  - **Apple:** A fully functional solution leveraging the Apple ecosystem is the primary focus for this concept. Cloud synchronization is managed using the built-in *Apple iCloud Keychain*.
    - Reference: https://support.apple.com/en-us/109016
  - **Android:** Given the absence of a native, built-in solution, the development team will explore various options. The goal is to prioritize the use of *Google's native solutions* where available.
  - The primary goal of authentication is to provide a secure semi-custodial wallet that includes a recovery solution. Because this setup requires that the keys remain solely in the users' possession and accessible only by them, *Apple iCloud Keychain* is an ideal choice for our implementation.

Bridge

- **Web Server:** Gin
- **Protocol:** Websocket
- **Database:** Redis
- **Cloud Provider:** AWS

Client

- **Role(s):**
  - The UI Component connects with the ***Bridge*** and implements CIP-30 compliant functions. This architecture eliminates the need to store or handle sensitive data in the browser (or any implementing application), creating a security layer by separating dApp interactions from the signing process.
- **SDK: Modular Web Component** (Future-ready integration)
  - The SDK prioritizes reusability and customization, allowing dApps to adapt the implementation to their unique UI/UX needs.
- **Integration:**
  - **Standalone Module:** Developers install a dedicated package that's part of Weld, our open-source Wallet Connector (**https://github.com/Cardano-Forge/weld**)
  - **Weld Integration:** For dApps already using Weld, the bindings and integration are included automatically.
- **Research & Investigation:**
  - **Browser Extension:** A solution that exclusively communicates with the ***Bridge*** without storing private data in the browser. This approach enables seamless integration, provided the dApp scans for all wallets installed in the user's browser.