MetaMaskのBitcoinウォレット考察

MetaMaskなどはウェブの拡張アプリとしてウォレットを持っている。
最初にニモニックを打ち込ませるし、パスワードを別途つければ以降はそちらでアクセスできるところからすると何らかのストレージを持っているはずである。
しかし普通にストレージを持つのは怖いので何か保護する仕組みを使っていると思う。

Gemini

MetaMaskのつくり Geminiの想像だとは思うが

- npmリポジトリ
- ブラウザ拡張
  - データの保存先は拡張アプリ専用の同期しないストレージ(`storage.local` 系)
  - ニモニックなどはそのまま保存せず、ユーザがつけるパスワードをもとにしたデータで暗号化した保管庫(Vault)に保存しているので、Vault自体が流出してもデコードは難しい。
  - パスワードは保存せず、パスワードを使ったデータもメモリ上にしか残さないようにしている。なのでログアウトしたら元データは残らない。
  - 画面はHTML+CSSでアプリはJavaScript/TypeScript。
- スマホアプリ
  - React Nativeを使っている
  - 保存先はAndroid/iOS次第だがどちらにせよ保護される場所に置く
  - 言語が同じなので npm を使うことで裏側のコードはまあまあ共有できるのだろう.
- Snaps
  - MetaMaskのプラグイン機能
  - MetaMask本体とはサンドボックスとして切り離されているので、MetaMaskが提供する情報だけが漏洩源になる(Snaps自身の情報もそうだけど)。
  - 拡張公開鍵のデータなどはもらえるようなのでウォレットとして生きていける。そうするとどのBitcoin Snapsも同じHDウォレットのパスにするかどうかという問題があったようだが、今のところは同じパスを参照するらしい。

Advanced Permissions(EIP-7715)

EVM系ということでCitreaのことも調べたが、MetaMask で当然扱えるようだ。

そしてAdvanced Permissions(EIP-7715)というのにも対応したそうなので AI支払いとかに使えるとかなんとか。

https://metamask.io/ja/news/introducing-advanced-permissions

https://x.com/yilmaz_btc/status/2041490992239366369

技術的な調査

ウェブアプリがクライアント側にデータを保存する方法

- Web Storage
  - key-valueペアで保存する。テキストのみ保存可能で容量は5MB程度。
  - 2タイプあるようだ
    - Local Storage
      - ブラウザを閉じてもデータが残る。設定情報など。
      - 拡張機能のアンインストール時に削除。
    - Session Storage
      - ブラウザを閉じるとデータが消える。一時的な入力内容の保持など。
      - ブラウザやプロファイルを閉じるときに削除。
- IndexedDB
  - ブラウザ内で動くデータベース。保存容量はPCの空き容量に依存。バイナリデータも扱うことができる。
  - 拡張機能のアンインストール時、またはブラウザで履歴やデータの削除などでも可能。
- Cookie
  - 容量は4KB程度。
  - これはデータ保存というよりは通信セッションのための一時的な保存という感じか
- Extension Storage(拡張機能専用ストレージ)
  - Web Storageとは別の機構で、拡張機能専用。つまり前述の3つはそれ以外からもアクセスできるということになる。
  - `storage.local` や `storage.sync` などというらしい(Chromeなら `chrome.storage.local` 、Firefoxなら `browser.storage.local` など)
    - `storage.local`
      - ローカルに保存する。上限は5MB程度らしいが増やすことができるそうだ。
      - 拡張アプリのアンインストールで削除。ユーザの履歴削除でも消えるかもしれない。
      - ウォレットアプリのデータを保存するならここか
    - `storage.sync`

`storage.local` のサーバ同期あり版。容量は100KB程度。
ChromeならGoogleアカウントと同期するので拡張アプリのアンインストールでしか削除されない。

MetaMaskの保存方法 Geminiの推測？

秘密鍵やシードの保存はパスワードを元にした鍵で暗号化する。Vaultという名前をつけている。
アプリを開いてパスワードを入力した後はメモリ上に復元したデータを持っている。時間経過やブラウザのクローズで破棄される。

`KeyringController` というのは JavaScript のライブラリか？

パスワードは保存しないので、Vaultのデータを抜き取られたとしてもデコードは難しい

モバイルのブラウザから動く？

ブラウザ拡張はモバイルChromeでは対応していないらしい。

WASM

WebAssemblyの略。ブラウザ上でJavaScript以外の言語を(高速に)動かすための仕組み。
JavaScript以外でプログラムを作り、ブラウザで動くバイナリデータを wasmファイルに変換して動かす。JavaScriptでは動作が重たい部分に使ったりなど。
WASMのファイルを読み込んで関数呼び出しのように使うことができる。スレッドとしてはJavaScriptと同じだがファイルを読んだりはできずメモリ経由でしかデータをやり取りできない。

bdk_wasm

BDKのwasmは公式ではなくコミュニティ管理になっている。fork元は MetaMaskでそちらは既に Deprecated になっている。
bitcoindevkit/bdk-wasm: The Bitcoin Dev Kit for Browsers and Node
MetaMask本体ではなくSnapsプラグイン用らしい。