Openloop

Openloop Connect
dApp連携アプリ仕様書

バージョン v0.4.3 ｜ 最終更新 2026年3月19日
株式会社ハウディ・クリプト　代表取締役　浅田 一憲

1. 概要

1.1 製品名

Openloop Connect - Openloopハードウェアウォレット用コンパニオンアプリケーション

1.2 目的

Openloop　Connectは、Openloopハードウェアウォレットと世界中のdApp（分散型アプリケーション）を接続するブリッジソフトウェアである。

主な役割:

1. **dApp連携**: WalletConnect 2.0プロトコルを介して、世界中の600以上のdAppからの署名リクエストを受信・処理
2. **セキュア署名**: 受信したトランザクションや署名リクエストをOpenloopハードウェアウォレットに転送し、SE050セキュアエレメント内で安全に署名
3. **マルチチェーン対応**: Ethereum/EVMチェーン（EIP155）、Bitcoin（BIP122）、Solana、TRONの4チェーンファミリーをサポート
4. **シームレス接続**: QRコード、Universal Link、Deep Linkによるワンタップ接続
5. **ファームウェア更新**: OTAによるOpenloopデバイスのファームウェア更新（メイン＋リカバリ）
6. **ブラウザ拡張連携**: LocalWebSocketサーバーおよびSafari Web Extensionにより、ブラウザから直接Openloopデバイスに接続

1.3 WalletConnectとは

WalletConnectは、ウォレットとdAppを接続するためのオープンプロトコルである。

- **エンドツーエンド暗号化**: ウォレットとdApp間の通信は完全に暗号化
- **チェーン非依存**: Ethereum、Bitcoin、Solanaなど多数のブロックチェーンに対応
- **業界標準**: MetaMask、Trust Wallet、Rainbow等の主要ウォレットが採用

- **600以上のdApp対応**: Uniswap、OpenSea、Aave、1inch等の主要DeFi/NFTプラットフォームが対応

WalletConnectは、ブロックチェーン業界における**事実上の世界標準**となりつつある。

1.4 対応プラットフォーム

プラットフォーム	OS	フレームワーク	接続方式	状態
Desktop	Windows	Electron	USB / BLE	✓ 対応
Desktop	macOS	Electron	USB / BLE	✓ 対応
Desktop	Linux	Electron	USB / BLE	✓ 対応
Mobile	iOS	React Native (Expo)	BLE	✓ 対応
Mobile	Android	React Native (Expo)	BLE	✓ 対応

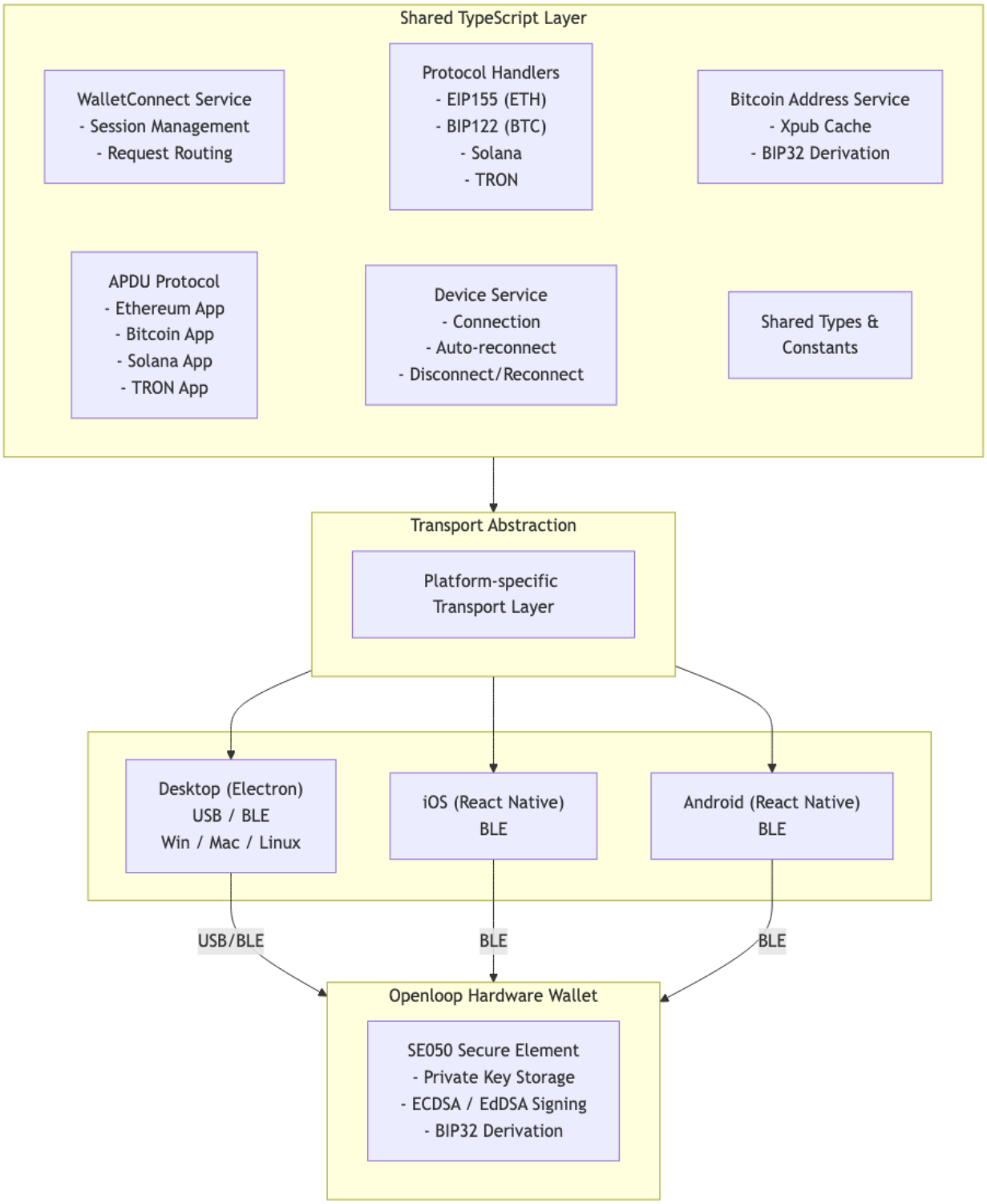
1.5 dApp接続方式

接続方式	説明	対応プラットフォーム
Deep Link	dAppのウォレット一覧から「Openloop Connect」を選択して自動接続（推奨）	全プラットフォーム
QR Code	dAppが表示するQRコードをスキャン	全プラットフォーム
Universal Link	HTTPSリンクからアプリを直接起動	iOS / macOS
Clipboard	WalletConnect URIをコピー&ペースト	全プラットフォーム
LocalWebSocket	ブラウザ拡張がws://127.0.0.1経由で直接接続	Desktop
Safari Web Extension	iOS SafariからBLE経由で直接接続	iOS

2. アーキテクチャ

2.1 クロスプラットフォーム設計

Openloop　Connectは、**1つのコードベースで5つのプラットフォーム**をサポートするクロスプラットフォームアーキテクチャを採用している。



2.2 技術スタック

レイヤー	技術	説明
共通ロジック	TypeScript	WalletConnect、APDU、プロトコルハンドラー
デスクトップUI	Electron + React	Windows / macOS / Linux
モバイルUI	React Native (Expo)	iOS / Android
BLE通信	Platform Native	各OSのBLE API
USB通信	node-hid	デスクトップのみ
OTA	CBOR-RPC	ファームウェア更新プロトコル
LocalWS	ws (Node.js)	ブラウザ拡張ブリッジ
Safari Extension	Swift + Manifest V3	iOS BLE拡張
Android Background	HeadlessJsTaskService	フォアグラウンドサービス

2.3 共通化の利点

1. **開発効率**: ビジネスロジックを1回実装するだけで5プラットフォーム対応
2. **一貫性**: すべてのプラットフォームで同一のWalletConnect処理
3. **保守性**: バグ修正や機能追加が全プラットフォームに即座に反映
4. **テスト効率**: 共通ロジックの単体テストで品質を担保

2.4 デバイス切断/再接続

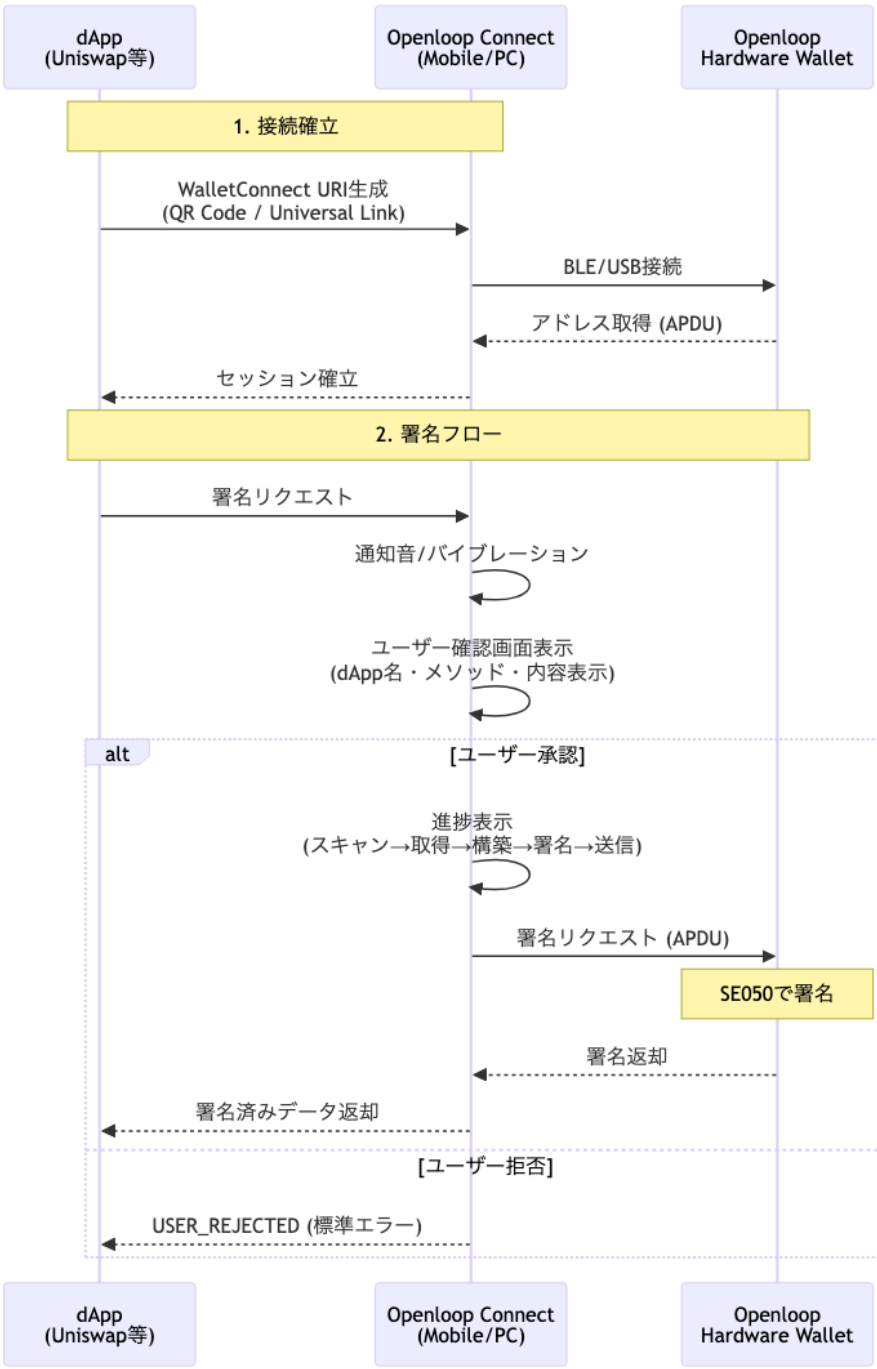
他のアプリ（Ledger Live等）がデバイスにアクセスできるよう、デバイスだけを手動切断しWalletConnectセッションは維持する機能。

プラットフォーム	切断方式	再接続方式
Desktop (USB)	切断ボタン → USB HID close	再接続ボタン → USBポーリング再開
Mobile (BLE)	切断ボタン → BLE disconnect	自動（次回操作時にlazy接続）

- 切断中もWalletConnect/LocalWSセッションは維持
- 再接続時に異なるデバイスが検出された場合、WCセッション自動切断
- Mobile: WC署名中やWSロック中は切断ボタン無効化

3. dApp連携フロー

3.1 接続フロー



署名確認UI: - dAppのアイコン・名前・URLを表示 - メソッド表示名（signMessage, signPsbt等）を多言語対応で表示 - メッセージ/トランザクション内容をスクロール可能な領域に表示 - 警告文: 「信頼できるソースからのメッセージのみ署名してください」 - Approve / Reject ボタン - 承認中: スピナー + 段階的進捗ステータス表示 - デバイス拒否（0x6985）検出: 自動的にUSER_REJECTEDをdAppに返却

通知: - デスクトップ: システム通知音（`shell.beep()`） - モバイル: バイブレーション（200ms） - 接続リクエスト・署名リクエストの両方で通知

3.2 マルチデバイス対応

Openloop Connectは、複数のOpenloopハードウェアウォレットの使用をサポートする。

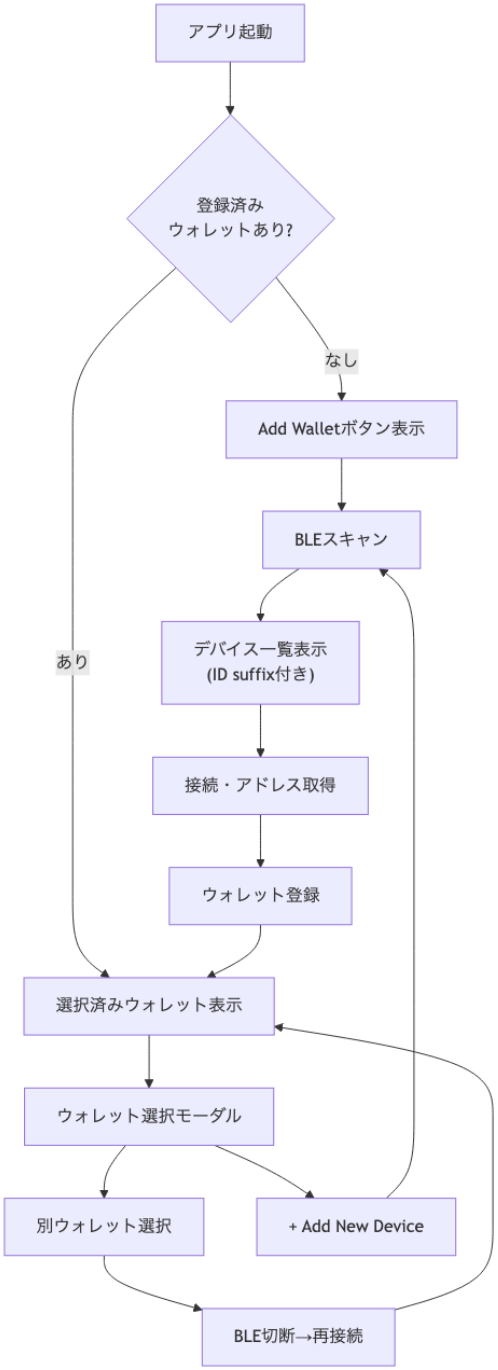
モバイル版（BLE）：ウォレット登録システム

モバイル版では、複数のOpenloopデバイスを「登録」し、使用するデバイスを「選択」できる。

データ構造:

```
interface RegisteredWallet {
  id: string          // 一意のID（UUID）
  deviceId: string    // BLEデバイスID
  deviceName: string  // "Openloop（E854）"
  address: string     // Ethereumアドレス "0x..."
  createdAt: number   // 登録日時
  lastUsedAt: number  // 最終使用日時
}
```

ウォレット管理フロー:



BLE排他制御:

- 同時に接続できるデバイスは1台のみ
- ウォレット切り替え時は既存接続をクローズしてから新規接続
- ミューテックスにより同時BLE操作を防止

デバイス識別:

- スキャンリストにはBLEデバイスIDの先頭4文字をサフィックスとして表示
- 例: “Openloop (E854)”, “Openloop (D72C)”
- 登録済みデバイスはスキャンリストから除外

デスクトップ版（USB）：物理切り替え

デスクトップ版では、USB接続されているOpenloopデバイスを自動検出して使用する。

マルチデバイス使用: 1. 現在のデバイスをUSBから取り外し 2. 別のデバイスをUSBに接続 3. Openloop Connectが自動的に新デバイスを検出 4. 新デバイスのアドレスでdApp接続

検出方式:

- USB VID/PIDチェックのみ（デバイスとの通信なし）
- 他のウォレットアプリ（MetaMask、Rabby等）との競合を回避
- デバイス検出時のみURI入力が有効化

項目	モバイル (BLE)	デスクトップ (USB)
登録必要	✓ 必要	✗ 不要
切り替え方法	アプリ内選択	物理的な接続切替
同時接続数	1台	1台
再接続	自動（選択済みデバイス）	自動（接続デバイス）

---

4. WalletConnect対応

4.1 プロトコルバージョン

項目	値
WalletConnect	2.0
Relay Server	wss://relay.walletconnect.com
Project ID	f43a098a34fa6d741ffa88b5ce738113

4.2 対応ネームスペース

ネームスペース	CAIP-2識別子	説明
eip155	eip155:1, eip155:11155111, …	Ethereum/EVMチェーン
bip122	bip122:000000000019d6689c085ae165831e93, …	Bitcoinチェーン
solana	solana:5eykt4UsFv8P8NJdTREpY1vzqKqZKvdp, …	Solana

ネームスペース	CAIP-2識別子	説明
tron	tron:0x2b6653dc, …	TRON

5. EIP155 (Ethereum/EVM) サポート

5.1 対応チェーン

チェーン	CAIP-2識別子
Ethereum Mainnet	eip155:1
Ethereum Sepolia	eip155:11155111
その他EVMチェーン	Chain IDで動的対応

5.2 対応メソッド

メソッド	説明	対応状況
eth_sendTransaction	トランザクション送信	✓
eth_signTransaction	トランザクション署名	✓
eth_sign	メッセージ署名 (非推奨)	✓
personal_sign	メッセージ署名 (EIP-191)	✓
eth_signTypedData	型付きデータ署名	✓
eth_signTypedData_v3	型付きデータ署名 v3	✓
eth_signTypedData_v4	型付きデータ署名 v4 (EIP-712)	✓
wallet_getCapabilities	ウォレット機能問合せ (EIP-5792)	✓

5.3 EIP-5792 Wallet Call API

`wallet_getCapabilities` メソッドに対応。dAppがウォレットの高度な機能（atomic batch, paymaster等）を問い合わせる際に使用される。

レスポンス形式:

```
{
  "0x1": {},
  "0xaa36a7": {}
}
```

ハードウェアウォレットとして、特別なケイパビリティは返却しない（空オブジェクト）。dAppは標準の `eth_sendTransaction` にフォールバックする。

5.4 対応イベント

イベント	説明
chainChanged	チェーン変更通知
accountsChanged	アカウント変更通知

---

6. BIP122 (Bitcoin) サポート

6.1 対応チェーン

チェーン	CAIP-2識別子	Genesis Hash (先頭32文字)
Bitcoin Mainnet	bip122:000000000019d6689c085ae165831e93	000000000019d6689c085ae165831e93
Bitcoin Testnet3	bip122:000000000933ea01ad0ee984209779ba	000000000933ea01ad0ee984209779ba

6.2 対応メソッド

メソッド	説明	対応状況
getAccountAddresses	BIP32派生アドレス一覧取得	✓
signPsbt	PSBT (BIP-174) 署名	✓
signMessage	メッセージ署名 (BIP-137)	✓
sendTransfer	Bitcoin送金	✓

6.3 getAccountAddresses

仕様: - Index 0を常に含める - UTXOがあるアドレスを全て含める - 未使用アドレスを最低2個含める - Gap limit: 20連続未使用で停止

実装: 1. ファームウェアからaccount xpub (m/84'/coin'/0') を取得 2. ローカルでBIP32子鍵導出（非ハードened） 3. mempool.space APIでUTXO/履歴確認 4. receive/change並列スキャン

性能: - スキャン時間: 約8秒（111アドレス） - API並列数: 4（レート制限対策）

6.4 signPsbt

仕様: - Base64エンコードされたPSBTを受信 - signInputsでアドレス→パス逆引き - ファームウェアで署名実行 - 署名済みPSBTをBase64で返却

処理フロー: 1. アドレスキャッシュからパス逆引き 2. キャッシュミス時はローカル導出で検索 3. PSBT + パス情報をファームウェアに送信 4. 署名済みPSBTを取得

6.5 signMessage

対応形式:

形式	説明	対応状況
ECDSA (BIP-137)	従来のBitcoinメッセージ署名	✓
BIP-322	汎用メッセージ署名（P2WPKH witness形式）	✓

ECDSA (BIP-137): - ハッシュ: SHA256(SHA256("18Bitcoin Signed Message:" + varint(len) + message)) - 署名: [V (35-38)][R (32B)][S (32B)] - 返却形式: Base64エンコード

BIP-322: - to_spend / to_sign トランザクションペアを構築 - witness形式で署名を返却 - 返却形式: Base64エンコード

6.6 sendTransfer

仕様: - ウォレット側でPSBTを構築（UTXO取得→手数料見積り→PSBT作成） - ファームウェアで署名実行 - 署名済みトランザクションをブロードキャスト - txidをdAppに返却

処理フロー: 1. dustチェック（546 sats未満を拒否） 2. アドレススキャン（残高のあるアドレスを検索） 3. UTXO取得（mempool.space API） 4. 手数料見積り（halfHourFee使用） 5. UTXO選択（largest first） 6. PSBT構築（お釣りアドレスにBIP32_DERIVATION付与） 7. ファームウェアで署名 8. ブロードキャスト（mempool.space API） 9. txid返却

---

7. Solana サポート

7.1 対応チェーン

チェーン	CAIP-2識別子
Mainnet Beta	solana:5eykt4UsFv8P8NJdTREpY1vzqKqZKvdp
Devnet	solana:EtWTRABZaYq6iMfeYKouRu166VU2xqa1

7.2 対応メソッド

メソッド	説明	対応状況
solana_signTransaction	トランザクション署名	✓
solana_signAllTransactions	複数TX一括署名	✓
solana_signAndSendTransaction	署名＋ブロードキャスト	✓
solana_signMessage	メッセージ署名 (オフチェーン)	✓

7.3 APDU

Ed25519署名。BIP44パス: m/44’/501’/account’/0’

8. TRON サポート

8.1 対応チェーン

チェーン	CAIP-2識別子
Mainnet	tron:0x2b6653dc
Shasta Testnet	tron:0x94a9059e

8.2 対応メソッド

メソッド	説明	対応状況
tron_signTransaction	トランザクション署名	✓

メソッド	説明	対応状況
tron_signMessage	メッセージ署名	✓

8.3 技術詳細

- secp256k1 (Ethereumと同一カーブ)
- BIP44パス: m/44'/195'/account'/0/0
- TXハッシュ: SHA-256 (Ethereumのkeccak256と異なる)
- TronGrid API経由でブロードキャスト

---

9. LocalWebSocket サーバー

ブラウザ拡張（Openloop Web SDK等）がOpenloop Connectを経由してハードウェアウォレットに接続するためのローカルWebSocketサーバー。

9.1 概要

項目	値
プロトコル	WebSocket (ws://)
アドレス	127.0.0.1 (ローカルホストのみ)
ポート	21320 (デフォルト)
データ形式	JSON-RPC 2.0

9.2 対応プラットフォーム

プラットフォーム	実装
Desktop (Electron)	Node.js `ws` ライブラリ
Mobile (Android)	HeadlessJsTaskService内WebSocketサーバー
Mobile (iOS)	Safari Web Extension経由（§10参照）

9.3 セキュリティ

- localhost (127.0.0.1) バインドのみ — 外部からのアクセス不可
- オプショナルトークン認証

---

10. Safari Web Extension (iOS)

iOS SafariからOpenloopハードウェアウォレットにBLE接続するためのSafari Web Extension。業界初のHWウォレット向けSafari BLE拡張。

10.1 アーキテクチャ

コンポーネント	技術	役割
Extension Process	Swift + CoreBluetooth	BLE通信
Content Script	JavaScript	Web3 provider注入
Background Script	JavaScript (Manifest V3)	メッセージルーティング
Popup	HTML/CSS/JS	ユーザーUI
Container App	React Native (Expo)	BLEパーミッション管理

10.2 通信フロー

dApp → Content Script → Background → Native Messaging → Swift BLE → Openloop

10.3 制限事項

- iOS Safari限定（macOS Safariは未対応）
- Container App（Openloop Connect iOS）のインストールが必要
- BLEパーミッションはContainer Appから初回付与

---

11. Android バックグラウンド動作

Android版はフォアグラウンドサービスにより、アプリがバックグラウンドにあってもWebSocketサーバーとBLE接続を維持する。

11.1 実装

コンポーネント	説明
WsServerService	HeadlessJsTaskService + PARTIAL_WAKE_LOCK + WIFI_LOCK
通知	常駐通知によりフォアグラウンドサービスとして動作
パーミッション	

コンポーネント	説明
	FOREGROUND_SERVICE, FOREGROUND_SERVICE_CONNECTED_DEVICE, WAKE_LOCK

11.2 動作

- アプリがバックグラウンドでもWSサーバーが応答
- BLE接続を維持し、署名リクエストを処理
- iOS版はOS制限によりバックグラウンドWebSocket非対応（Safari Extension経由で補完）

---

12. OTA ファームウェア更新

Openloop Connectは、Openloopデバイスのファームウェアをワイヤレスまたは USBで更新する機能を提供する。

12.1 対応トランスポート

トランスポート	プラットフォーム	プロトコル
USB HID	Desktop	TAG 0x06 CBOR over Ledger HID
USB CDC	Desktop (Python CLI)	Raw CBOR-RPC
BLE	Mobile (iOS/Android)	Openloop BLEサービス

12.2 OTA対象

対象	説明
メインファームウェア	デバイスの主要機能（ウォレット、パスキー等）
リカバリーファームウェア	リカバリーモード用FW（OTA Recovery Mini）

12.3 プロトコル

項目	仕様
データ形式	CBOR-RPC
圧縮	ZLIB
チャンク検証	CRC32

項目	仕様
最終検証	SHA-256ハッシュ
署名検証	RSA-3072 (デバイス側)
Delta OTA	差分更新対応

12.4 OTAフロー

1. バージョン確認 (`openloop_get_version`)
2. OTA開始 (`ota`)
3. チャンク転送 (`ota_data` , CRC32検証)
4. OTA完了 (`ota_complete` , SHA-256検証)
5. デバイス側でRSA-3072署名検証 → 適用・再起動

12.5 HID優先戦略

FW v0.88.7以降のデバイスでは、HIDトランスポートを優先使用。HID非対応の古いFWにはCDCフォールバック。

---

13. 通信プロトコル

13.1 BLE通信

サービスUUID:

項目	UUID
Service	13D63400-2C97-0004-0000-4C6564676572
Notify Characteristic	13D63400-2C97-0004-0001-4C6564676572
Write Characteristic	13D63400-2C97-0004-0002-4C6564676572

フレームプロトコル:

業界標準BLEフレーム分割プロトコル。

項目	値
フレームプレフィックス	0x05
最初のフレームヘッダ	5バイト (PREFIX + INDEX + SIZE)
継続フレームヘッダ	3バイト (PREFIX + INDEX)

項目	値
MTU	247バイト

13.2 USB通信（デスクトップのみ）

モード	VID	PID	備考
独自モード	0x303A	0x8341	Espressif VID / Openloop PID
互換モード	0x2C97	0x1011	互換VID / Nano S互換PID

- Interface: HID
- デバイス設定画面でモード切替可能

HID応答破損検出:

- 全0xAAバイトパターンを検出し自動リトライ
- デバイス側HID送信タイミング問題への対策
- 最大3回リトライ、失敗時はエラー

13.3 自動再接続

BIP122のgetAccountAddresses（約8秒）実行中にBLE接続が切れる可能性があるため、以下のメソッドで自動再接続を実行:

- signPsbt()
- signPsbtWithPaths()
- getAccountXpub()
- signBitcoinMessage()

---

14. APDUプロトコル

14.1 業界標準コマンド (CLA=0xE0)

Ethereum:

INS	コマンド	説明
0x02	GET_PUBLIC_KEY	公開鍵・アドレス取得
0x04	SIGN	トランザクション署名

INS	コマンド	説明
0x08	SIGN_PERSONAL_MESSAGE	メッセージ署名 (EIP-191)
0x0C	SIGN_EIP712	EIP-712署名

Bitcoin:

INS	コマンド	説明
0x40	GET_WALLET_PUBLIC_KEY	公開鍵・アドレス取得

14.2 Openloop拡張コマンド (CLA=0xF0)

INS	コマンド	説明
0x08	SIGN_PERSONAL_MESSAGE	メッセージ署名 (CLA_OPENLOOP版)
0x0C	SIGN_EIP712	EIP-712署名　(CLA_OPENLOOP版)
0x70	SIGN_PSBT	PSBTデータ受信・署名開始
0x71	GET_SIGNED_PSBT	署名済みPSBT取得
0x72	GET_ACCOUNT_XPUB	アカウントxpub取得
0x73	SIGN_MESSAGE	メッセージ署名 (BIP-137)

14.3 Solana APDUコマンド (CLA=0xE0)

INS	コマンド	説明
0x05	GET_PUBKEY	公開鍵取得 (Ed25519)
0x06	SIGN_MESSAGE	トランザクション署名
0x07	SIGN_OFFCHAIN_MESSAGE	オフチェーンメッセージ署名

14.4 TRON APDUコマンド (CLA=0xE0)

INS	コマンド	説明
0x02	GET_ADDRESS	TRONアドレス取得 (Base58Check)
0x04	SIGN_TX	トランザクション署名 (SHA-256)
0x08	SIGN_PERSONAL_MESSAGE	メッセージ署名

14.5 プロトコル詳細

詳細は BIP122_signPsbt_Implementation_Spec.md を参照。

---

15. ファイル構成

```
openloop-connect/
├── packages/
│   ├── shared/                        # 共通TypeScriptコード
│   │   └── src/
│   │       ├── apdu/                  # APDUプロトコル
│   │       │   ├── ethereum-app.ts
│   │       │   ├── bitcoin-app.ts
│   │       │   ├── solana-app.ts          # Solana APDU
│   │       │   └── tron-app.ts            # TRON APDU
│   │       ├── bitcoin/               # Bitcoin共通ロジック
│   │       │   ├── address-cache.ts       # アドレス↔パス マッピングキャッシュ
│   │       │   ├── address-service.ts     # Gap limitスキャン・アドレス逆引き
│   │       │   ├── bip32-derive.ts        # BIP32公開鍵導出
│   │       │   ├── bip322.ts              # BIP-322メッセージ署名ユーティリティ
│   │       │   ├── tx-service.ts          # PSBT構築・UTXO取得・ブロードキャスト
│   │       │   └── index.ts
│   │       ├── ethereum/              # Ethereum関連ユーティリティ
│   │       │   ├── index.ts
│   │       │   ├── mini-ethers.ts
│   │       │   ├── transaction.ts
│   │       │   └── typed-data.ts
│   │       ├── codec/                 # エンコーディング
│   │       │   └── base58.ts              # Base58コーデック
│   │       ├── ota/                   # OTAプロトコル
│   │       ├── walletconnect/         # WalletConnect
│   │       │   └── namespace-builder.ts  # ネームスペースビルダー
│   │       ├── sign-handler/          # 署名ハンドラ共通化
│   │       ├── i18n/                  # 共通多言語文字列
│   │       │   ├── wc-strings.ts          # WalletConnect関連 (EN/JA)
│   │       │   └── index.ts
│   │       ├── transport/
│   │       │   └── interface.ts       # トランスポート抽象インターフェース
│   │       ├── constants.ts           # 定数・mapErrorToI18nKey()
│   │       ├── ws-protocol.ts         # WSプロトコル定義
│   │       ├── index.ts
│   │       └── types.ts
│   │
│   ├── desktop/                       # デスクトップアプリ (Electron)
│   │   └── src/
│   │       ├── main/                  # メインプロセス
│   │       │   ├── hid-service.ts         # USB HID通信 (node-hid)
│   │       │   ├── ws-server.ts           # LocalWebSocketサーバー
│   │       │   ├── ota-service.ts         # Desktop OTA
│   │       │   ├── device-service.ts      # デバイス切断/再接続管理
│   │       │   ├── walletconnect-service.ts
│   │       │   ├── index.ts               # 署名保留・確認・進捗管理
│   │       │   └── utils.ts
│   │       ├── preload/
│   │       │   └── index.ts
│   │       └── renderer/              # レンダラープロセス (React)
```

```
│   │           ├── App.tsx                # 署名確認モーダル含む
│   │           ├── i18n/                  # デスクトップ固有 + shared i18n
│   │           └── main.tsx
│   │
│   └── mobile/                        # モバイルアプリ (React Native)
│       ├── App.tsx
│       ├── ios/
│       │   └── SafariExtension/       # Safari Web Extension
│       └── src/
│           ├── transport/             # BLEトランスポート
│           │   ├── BleTransport.ts
│           │   └── index.ts
│           ├── services/              # モバイル固有サービス
│           │   ├── device.ts              # BLE接続管理
│           │   ├── wallet-storage.ts      # ウォレット登録・選択
│           │   ├── walletconnect.ts
│           │   ├── ota-service.ts         # Mobile OTA
│           │   └── ws-server.ts           # Mobile WebSocketサーバー
│           ├── screens/               # UI画面
│           │   ├── ConnectScreen.tsx      # dApp接続（メイン画面）
│           │   ├── SignScreen.tsx         # 署名確認
│           │   ├── ScanDeviceScreen.tsx   # BLEスキャン
│           │   └── WalletAddedScreen.tsx # 登録完了
│           ├── i18n/                  # モバイル固有 + shared i18n
│           └── types/
│               └── navigation.ts
│
└── docs/
    ├── OpenloopConnect_Spec_2026.md            # 本仕様書
    └── BIP122_signPsbt_Implementation_Spec.md  # BIP122実装仕様
```

---

16. ビジョン

16.1 WalletConnect対応の意義

WalletConnectは、**600以上のdApp**が採用するブロックチェーン業界の事実上の標準プロトコルである。

Openloop ConnectがWalletConnectに完全対応することで:

- **即座にエコシステム参加**: Uniswap、OpenSea、Aave、1inch等の主要dAppでOpenloopが使用可能
- **新規dApp対応不要**: 新しいdAppがWalletConnect対応すれば、Openloop側の追加開発なしで利用可能
- **業界標準準拠**: 主要ハードウェアウォレットと同等のdApp連携能力

16.2 独自モードの拡大

Openloop Connectは独自モード（VID: 0x303A / PID: 0x8341）の主要な実行基盤であり、対応する接続手段は拡大を続けている。

Openloop Connect経由の独自モード接続:

接続手段	説明	プラットフォーム
WalletConnect	600以上のdAppから署名リクエストを受信	全プラットフォーム
LocalWebSocket	ブラウザ拡張がws://127.0.0.1経由で直接接続	Desktop / Android
Safari Web Extension	iOS　SafariからBLE経由で直接接続	iOS

これにより、WalletConnect対応dApp、Openloop Web SDK対応のブラウザ拡張、Safari上のWeb3 dAppなど、**独自モードだけでカバーできるユースケースが急速に拡大**している。

独自モード戦略の全体像（WebHIDを含む）については、Openloop製品仕様書を参照。

16.3 今後の展望

1. **WalletConnectエコシステムの成長**: 対応dAppは増加の一途
2. **Bitcoin dAppの成長**: BIP122対応により、Bitcoin DeFi/Ordinals市場にも対応
3. **独自機能の拡張**: ハードウェアウォレットならではの高度なセキュリティ機能
4. **マルチシグ対応**: 複数のOpenloopデバイスによる共同署名

---

17. 参考資料

17.1 WalletConnect

ドキュメント	URL
WalletConnect 2.0 Specs	https://specs.walletconnect.com/
WalletConnect Cloud	https://cloud.walletconnect.com/
Sign API	https://docs.walletconnect.com/api/sign
React Native SDK	https://docs.walletconnect.com/advanced/walletconnectmodal/about?platform=react-native
Electron Integration	

ドキュメント	URL
	https://docs.walletconnect.com/advanced/walletconnectmodal/about?platform=web

17.2 CAIP (Chain Agnostic Improvement Proposals)

規格	説明	URL
CAIP-2	Blockchain ID Specification	https://github.com/ChainAgnostic/CAIPs/blob/main/CAIPs/caip-2.md
CAIP-10	Account ID Specification	https://github.com/ChainAgnostic/CAIPs/blob/main/CAIPs/caip-10.md
CAIP-25	Wallet JSON-RPC Methods	https://github.com/ChainAgnostic/CAIPs/blob/main/CAIPs/caip-25.md

17.3 Bitcoin BIP122

ドキュメント	URL
BIP122 Scope (WalletConnect)	https://docs.walletconnect.com/advanced/multichain/rpc-reference/bitcoin-rpc
CAIP-122 (signMessage)	https://github.com/ChainAgnostic/CAIPs/blob/main/CAIPs/caip-122.md

17.4 Ethereum EIP155

ドキュメント	URL
EIP155 Scope (WalletConnect)	https://docs.walletconnect.com/advanced/multichain/rpc-reference/ethereum-rpc
EIP-191	https://eips.ethereum.org/EIPS/eip-191
EIP-712	https://eips.ethereum.org/EIPS/eip-712
EIP-5792 (Wallet Call API)	https://eips.ethereum.org/EIPS/eip-5792

17.5 Bitcoin規格

規格	説明	URL
BIP-32	HD Wallets	

規格	説明	URL
		https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki
BIP-84	Native SegWit	https://github.com/bitcoin/bips/blob/master/bip-0084.mediawiki
BIP-137	Message Signing	https://github.com/bitcoin/bips/blob/master/bip-0137.mediawiki
BIP-174	PSBT	https://github.com/bitcoin/bips/blob/master/bip-0174.mediawiki

17.6 Solana

ドキュメント	URL
Solana WalletConnect Scope	https://docs.walletconnect.com/advanced/multichain/rpc-reference/solana-rpc
Solana Web3.js	https://solana-labs.github.io/solana-web3.js/
Ledger Solana App	https://github.com/LedgerHQ/app-solana

17.7 TRON

ドキュメント	URL
TronGrid API	https://www.trongrid.io/
Ledger TRON App	https://github.com/AurelienMusic/app-tron
TIP-191 (signMessage)	https://github.com/tronprotocol/TIPs/blob/master/tip-191.md

17.8 OTA・ファームウェア

ドキュメント	URL
CBOR (RFC 8949)	https://www.rfc-editor.org/rfc/rfc8949
FIDO2 / WebAuthn	https://fidoalliance.org/fido2/

18. 変更履歴

バージョン	日付	内容
v0.1.0	2026-01-18	初版作成（EIP155 + BIP122対応）
v0.2.0	2026-01-18	大幅改訂: デスクトップ対応追加、アーキテクチャ詳細化、ビジョン追加、Mermaid図対応
v0.2.2	2026-01-19	マルチデバイス対応: モバイルBLEウォレット登録システム、デスクトップUSB物理切り替え
v0.3.0	2026-02-14	デスクトップBIP122対応、Bitcoinサービス共通化（shared移動）
v0.3.1	2026-02-14	署名確認UI追加（デスクトップ）、エラー応答標準化、shared i18nモジュール
v0.3.4	2026-02-15	EIP-5792 wallet_getCapabilities対応、通知音/バイブレーション、エラーメッセージ統一、sendTransfer仕様修正、BIP-322対応追記
v0.3.6	2026-02-17	Solana WalletConnect対応、App Store/Play Store公開
v0.3.10	2026-02-17	APDU共通化(CLA_OPENLOOP)、TRON WalletConnect対応
v0.3.12	2026-02-25	Desktop/Mobile UI統一、GitHub Actions CI修正
v0.4.1	2026-03-11	OTA USB HID対応、LocalWebSocketサーバー、Safari Web Extension、Androidバックグラウンド
v0.4.3	2026-03-19	デバイス切断/再接続、Androidフォアグラウンドサービス復旧

Openloop Connect - 株式会社ハウディ・クリプト