Team Name: LIC 2.0

Project:Crypto Wallet Browser Extension

Team Members:

- Natik Prajapati
- Nirav Changela
- Keval Dhabalia
- Vraj Patel

Cross-Chain Crypto Wallet Browser Extension

We presents a cross-chain crypto wallet. This is a browser extension. It seamlessly integrates with React. This wallet supports multiple blockchains. It provides an easy-to-use API. The goal is to address fragmentation issues. Current wallets are difficult to integrate.

The Need for a Unified Wallet Solution

Current wallets cause fragmentation. They're difficult to integrate into React apps. Developers face inconsistent APIs. Authentication is also a challenge.

Fragmented Ecosystem

Existing wallets offer inconsistent APIs.

Integration Hurdles

React app integration is complex.

Authentication Issues

Authentication processes are difficult to manage.

Problem Statement: Fragmented Wallet Landscape

Modern blockchain applications require seamless wallet connectivity. Developers struggle with fragmented APIs. They also face authentication complexities.

Inconsistent APIs

Fragmented APIs across different wallets.

Authentication

Authentication complexities in React applications.

Compatibility

Cross-platform compatibility issues.

Solution: Standardized interface for smooth React integration.

Project Goals and Key Features

This project aims to provide a standardized wallet interface. It should connect easily to any React app. It will support multiple blockchains. A developer-friendly API is crucial. A user-friendly UI is essential.

Standardized Interface

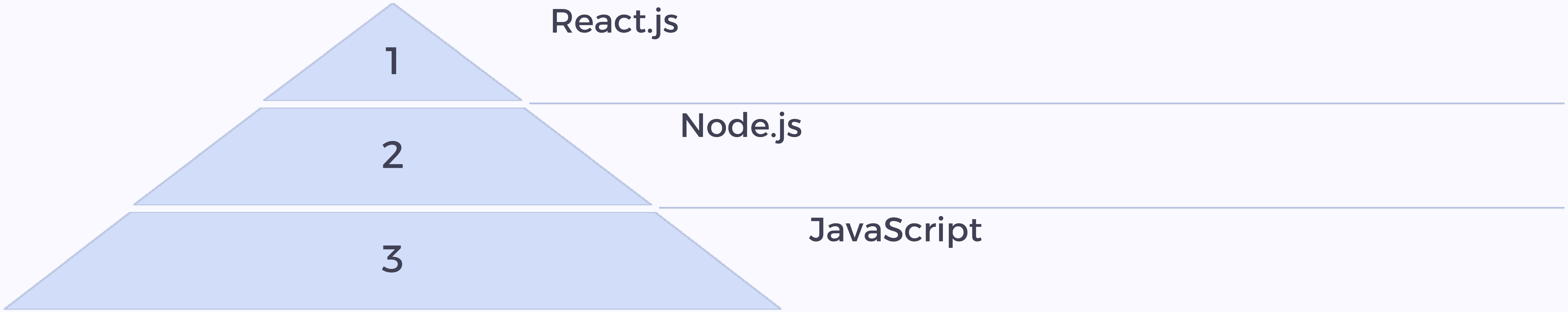
Multi-Blockchain Support

Developer-Friendly API

User-Friendly UI

Technology Stack Overview

The browser extension uses JavaScript (Manifest V3). Blockchain networks include Ethereum and Solana. The backend is Node.js with Express.js. The frontend is React.js.



Development Progress Highlights

UI setup for the browser extension is complete. React application injection is done. Basic blockchain interaction for Ethereum and Solana is implemented. A transaction approval mechanism exists.

1

UI Setup

Popup & settings page designed

2

React Injection

Wallet API injects into React apps.

3

Blockchain Support

Basic interaction implemented.

4

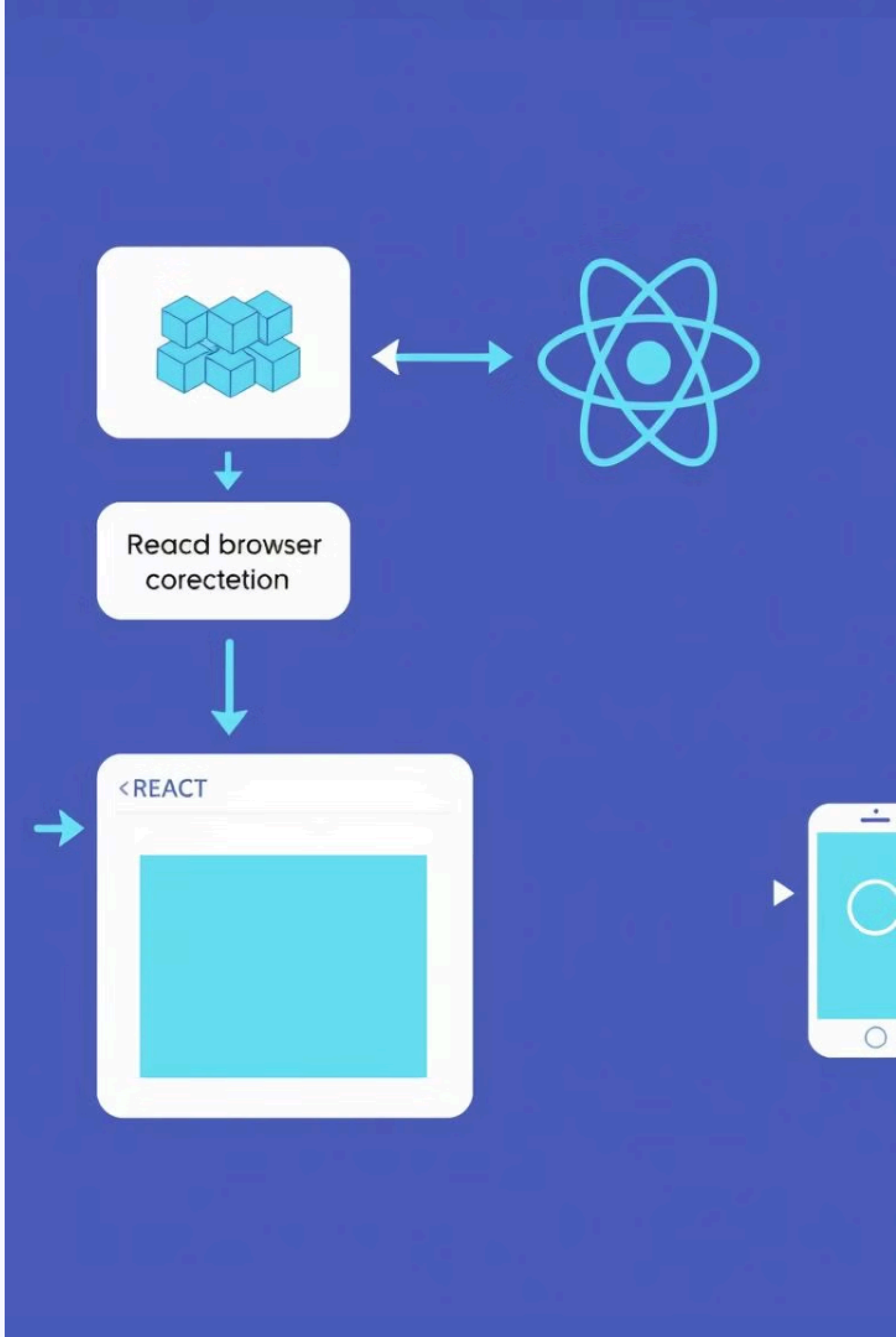
Transaction Approval

User confirmation is required.

System Architecture Diagram

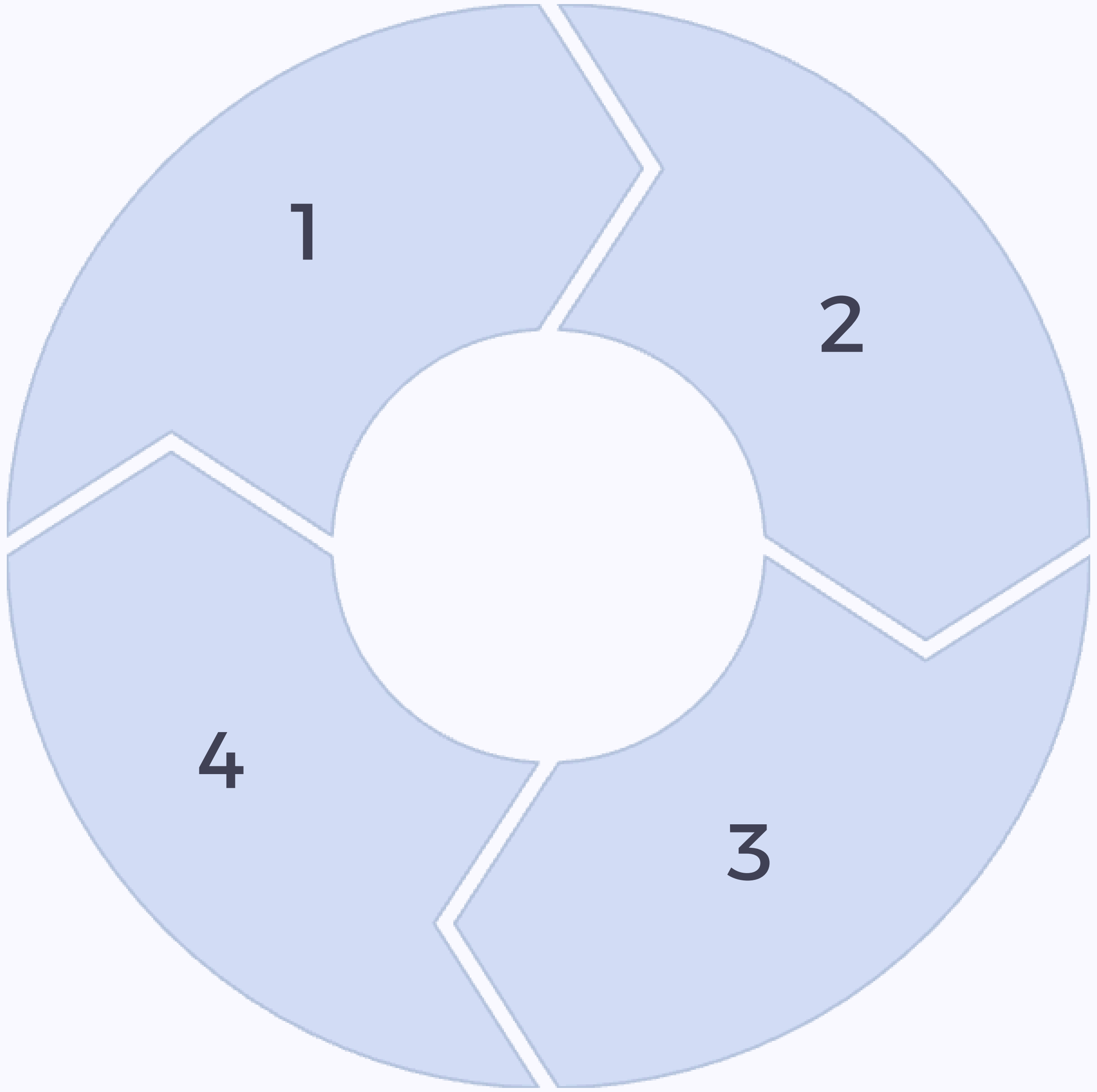
The browser extension interacts with React App. The extension injects the API. This enables blockchain transactions. The UI handles transaction approval.

1. Extension Injection
2. Transaction Request
3. Blockchain Interaction
4. Transaction Approval



Key Challenges and Solutions

Cross-browser support is a challenge. React app compatibility requires work. Seamless React integration is difficult. Transaction processing speed needs optimization.



Cross-Browser Support

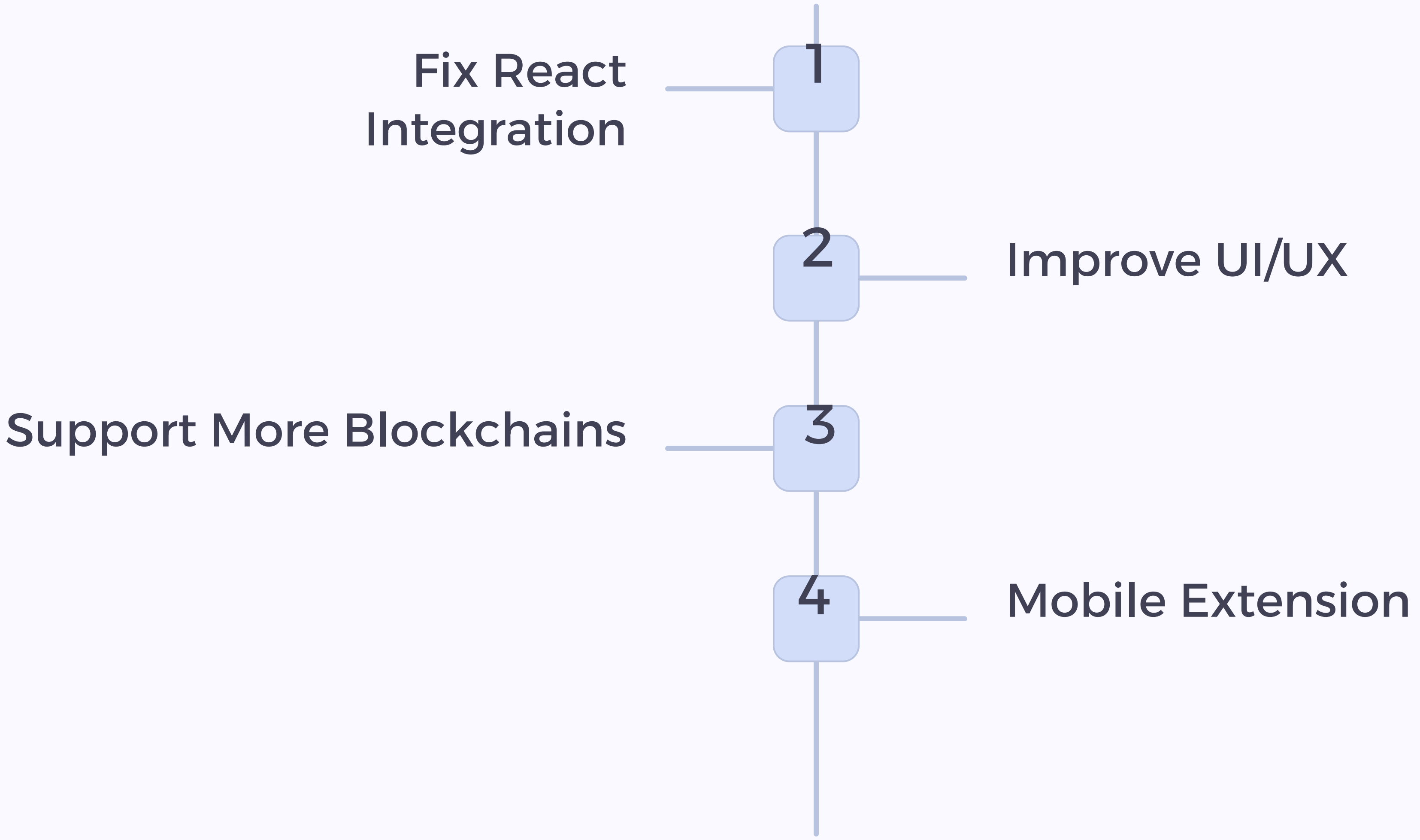
React Compatibility

Seamless Integration

Transaction Speed

Future Roadmap and Expansion

The short-term goal is to fix React app integration issues. Improving UI/UX is a priority. Long-term, support more blockchains. Mobile extension compatibility is planned.



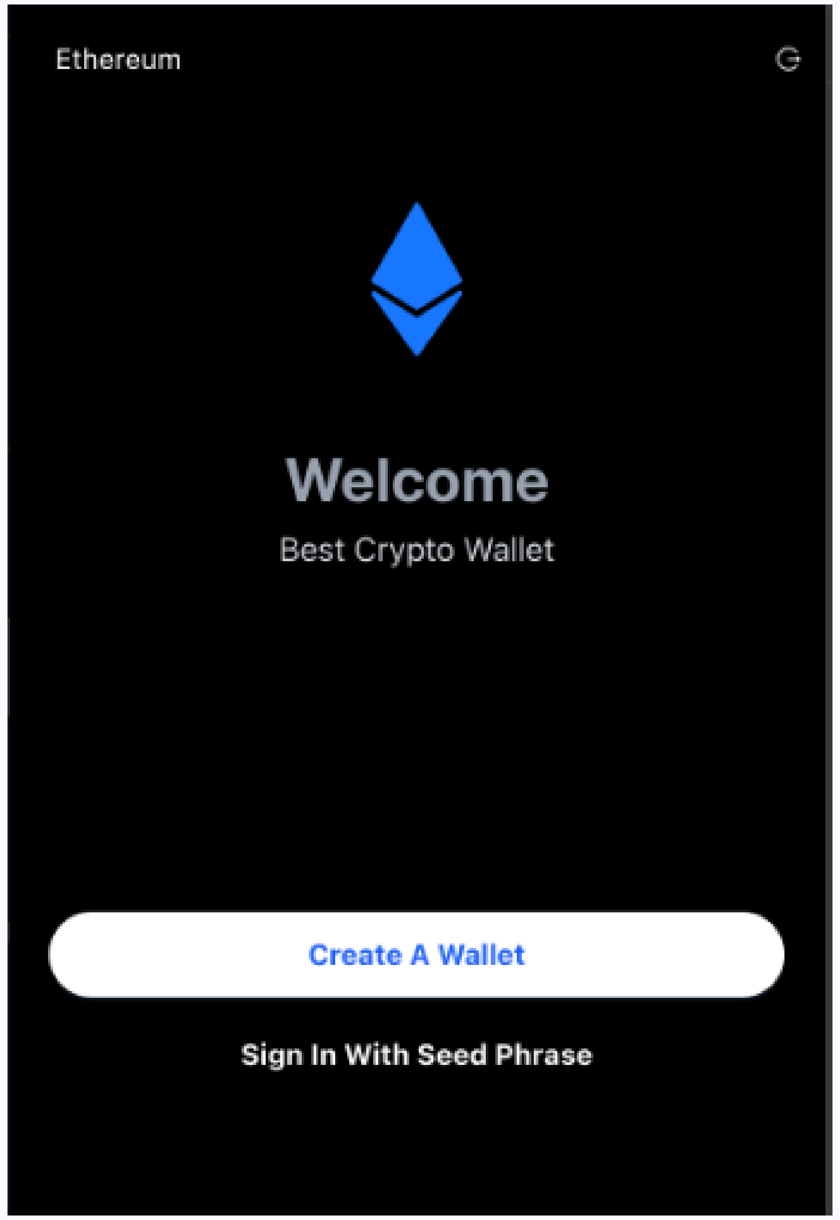
1 Fix React Integration

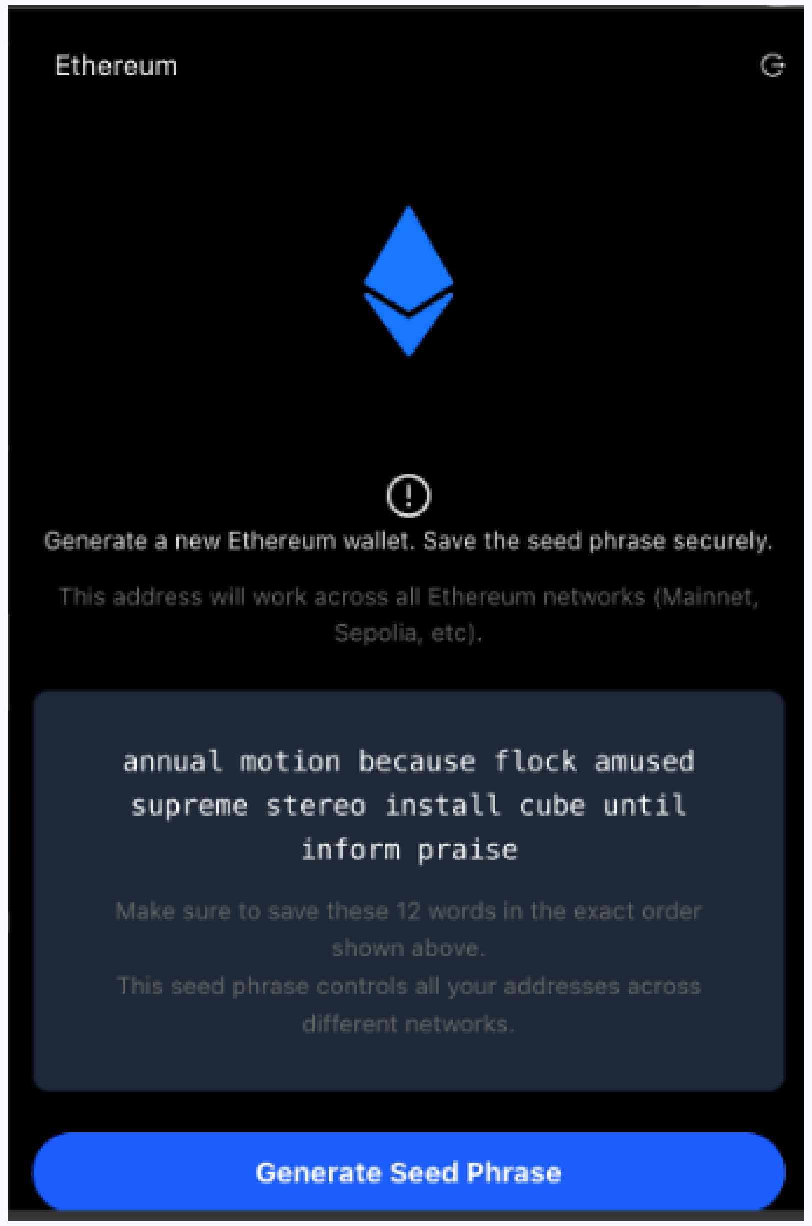
2 Improve UI/UX

3 Support More Blockchains

4 Mobile Extension

Implementation:



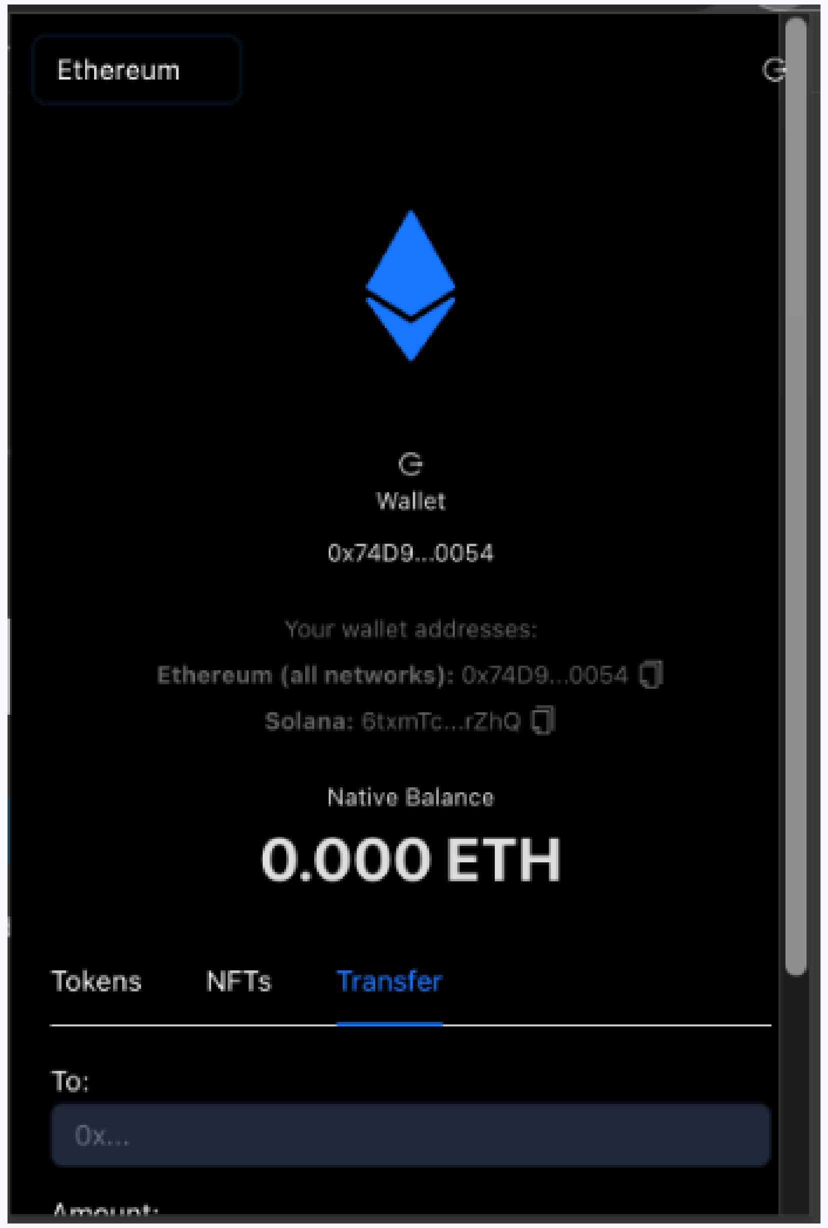


Generate a new Ethereum wallet. Save the seed phrase securely.

This address will work across all Ethereum networks (Mainnet, Sepolia, etc).

```
annual motion because flock amused
supreme stereo install cube until
inform praise
```

Make sure to save these 12 words in the exact order shown above.
This seed phrase controls all your addresses across different networks.



Wallet

0x74D9...0054

Your wallet addresses:

Ethereum (all networks): 0x74D9...0054

Solana: 6txmTc...rZhQ

Native Balance

0.000 ETH

To:

0x...

Amount: