

# Multichain Unity Toolkit

For webGL builds



# It's time to Move your browser based games to Web3

transition to Web3 as simple as it's a piece of cake

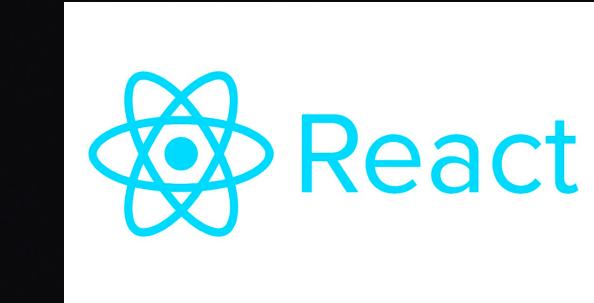
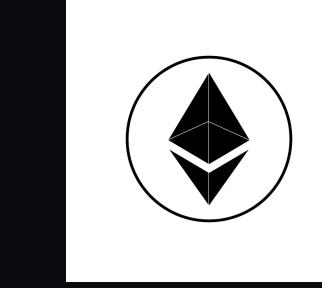
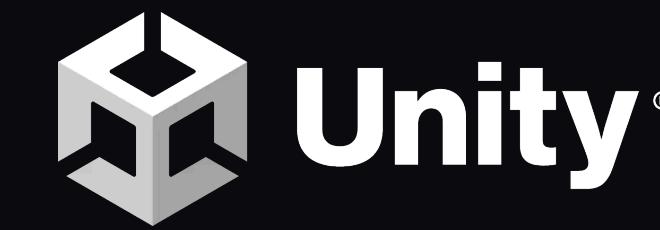


## Limited Flexibility

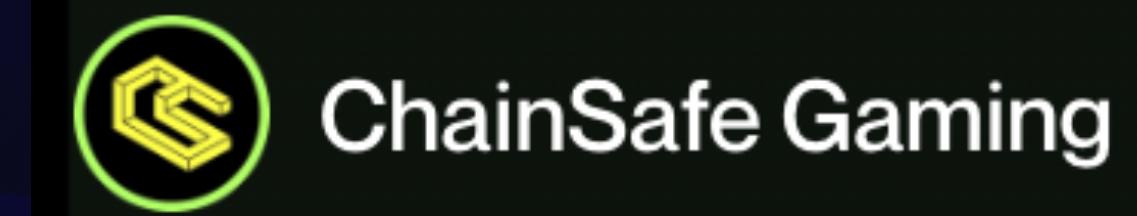
- Not truly multichain
- Heavy dependencies

**Advanced APIs, True Multichain, Ready to use!**

Ease of Development ↓



...

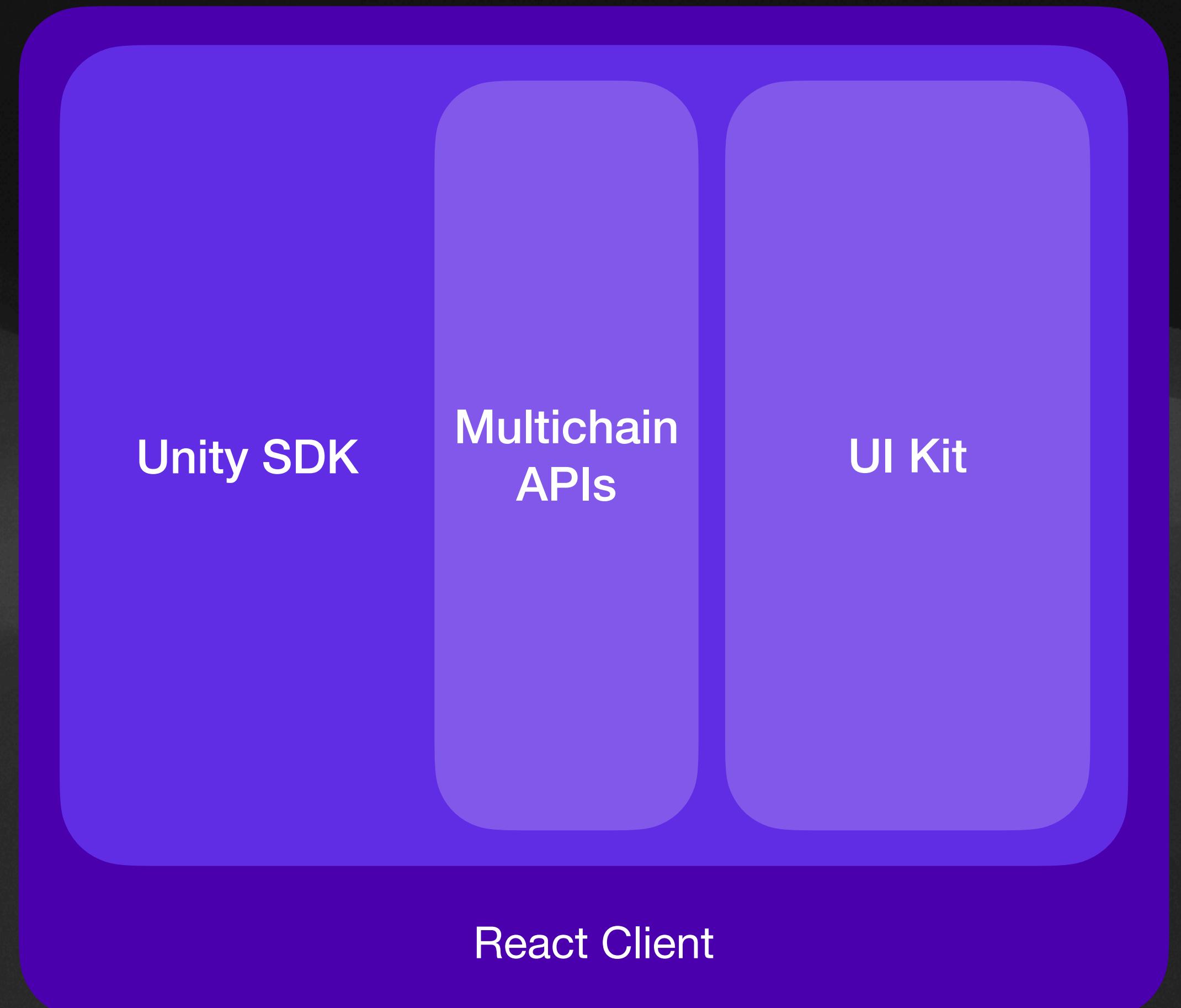


...others

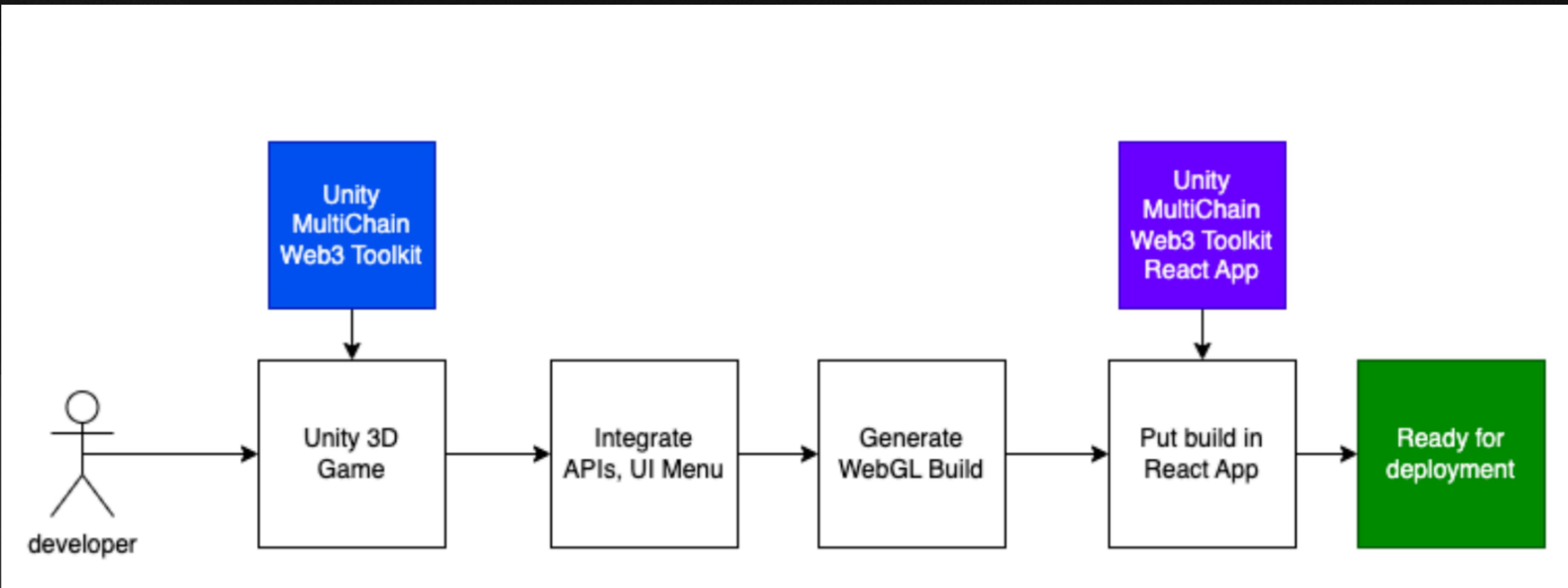


# Design

- Unity SDK
  - Customisable UI Kit
  - Multichain Web3 APIs
    - Wallet Sign In
    - Transactions & Interaction with Smart Contracts
    - Transaction Signing
    - Change Network / Chain
    - Async Events - Success & Error
- React Client



# Development Flow



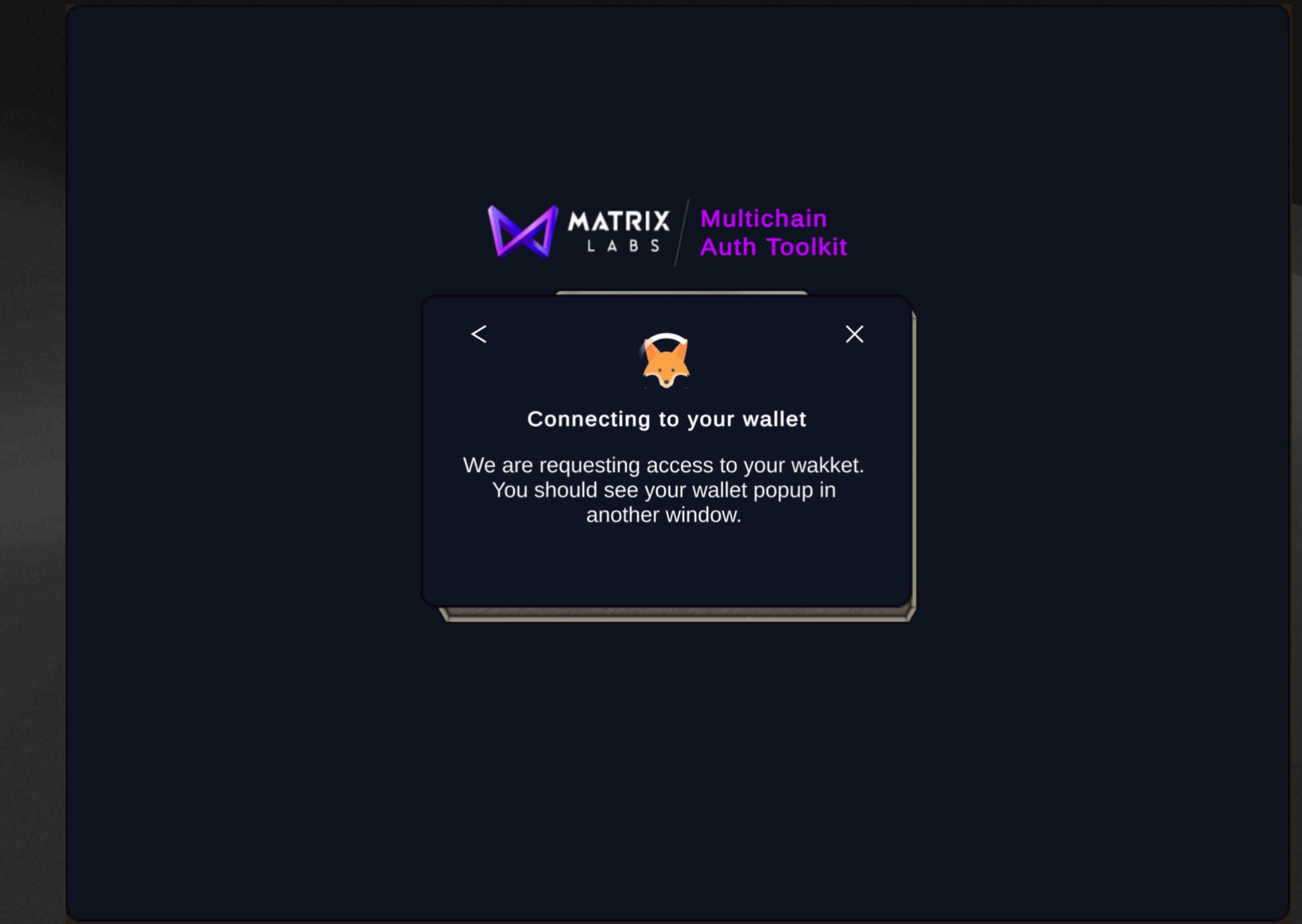
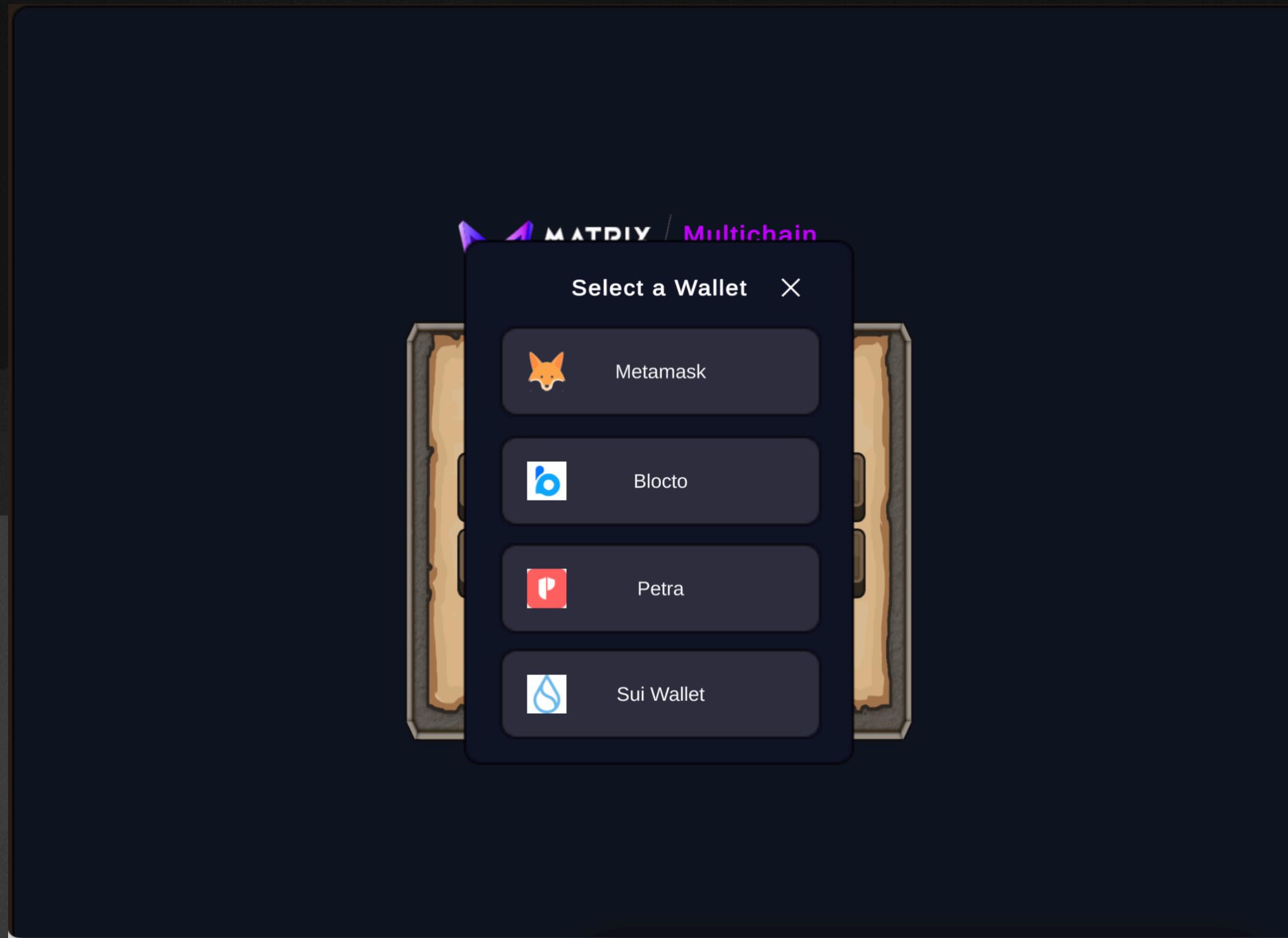
# APIs

- Custom namespaces, clean commented coding practices
- Editable invokable functions
- Easy to add new web3 APIs

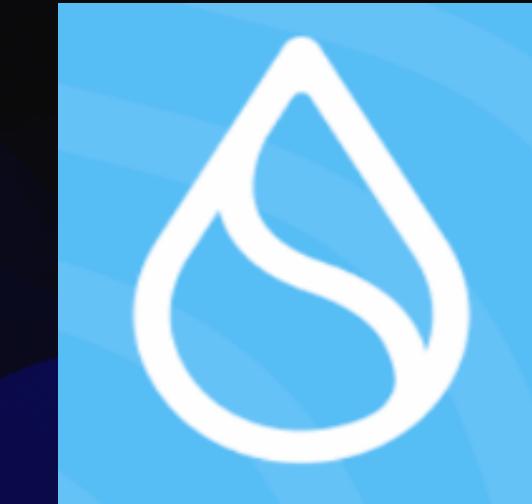
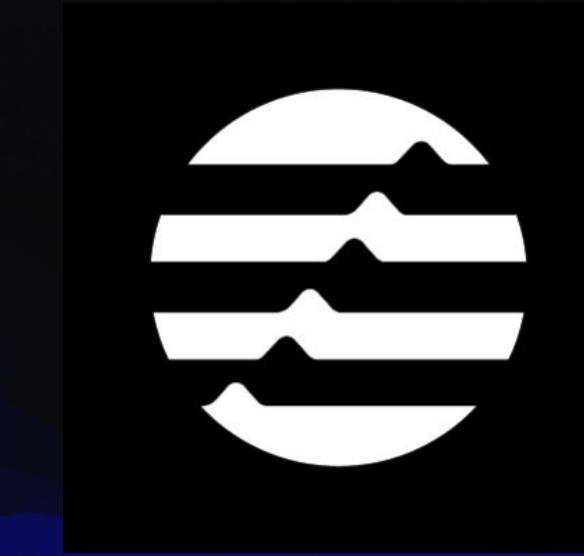
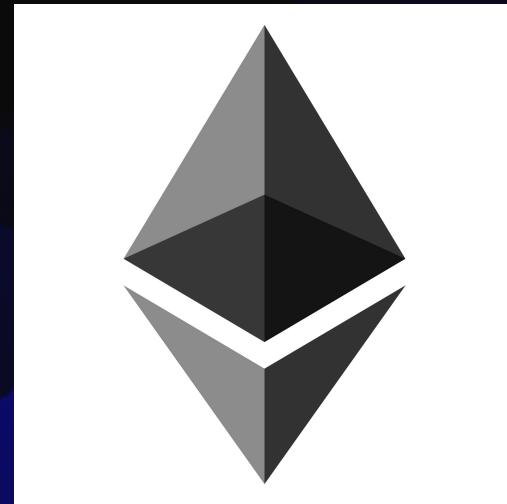
```
EventHandler.cs
1 // Example function to sign a message
2 public void HandlePerformSignMessage()
3 {
4     SignMessage("Matrix Labs", "Nounce");
5 }
6
7 /**
8 * This method signs a message. It accepts a string message and a string nounce.
9 * The active blockchain would be used to sign this message.
10 */
11 public void SignMessage(string message, string nounce)
12 {
13     StartCoroutine(InvokeJSMETHOD("Web3SignMessage", new JSData()
14     {
15         signMessageRequest = new Web3SignMessageRequest()
16         {
17             message = message,
18             nonce = nounce
19         }
20     }, 10));
21 }
22
23 /**
24 * This method receives the updated wallet balance and updates the UI.
25 */
26 public void UpdateWalletBalance(string jsonString)
27 {
28     Web3WalletBalResponse res = JsonUtility.FromJson<Web3WalletBalResponse>(jsonString);
29     Debug.Log("bal->" + res.bal.ToString());
30     pendingInvocations.Remove(res.invocationId);
31     TextMeshProUGUI textmeshPro = walletBalanceText.transform.GetComponent<TextMeshProUGUI>();
32     textmeshPro.SetText(res.bal.ToString());
33     StartCoroutine(HandleMessageDisplay("success", res.invocationId, "Wallet balance updated"));
34 }
35
36 /**
37 * This method requests wallet balance from the active blockchain and network.
38 */
39 public void FetchLatestBalance()
40 {
41     StartCoroutine(InvokeJSMETHOD("FetchWalletBalance", new JSData()
42     {
43         pendingInvocations.Add(res.invocationId);
44     })
45 }
```

```
Bridge.jslib
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using System.Runtime.InteropServices;
5 using UnityEngine.UI;
6 using TMPro;
7 using System;
8
9 namespace MatrixLabs
10 {
11     [Serializable]
12     public enum MatrixState
13     {
14         NOT_CONNECTED,
15         WALLET_CONNECTING,
16         IDLE,
17         ERROR,
18         WALLET_BALANCE_FETCHING,
19         TRANSACTION_PENDING,
20         TRANSACTION_COMPLETED,
21     }
22     [Serializable]
23     public enum Blockchain
24     {
25         None,
26         ETHEREUM,
27         FLOW,
28         APTOS,
29         SUI
30     }
31     [Serializable]
32     public enum BlockchainNetwork
33     {
34         None,
35         ETHEREUM_TESTNET,
36         ETHEREUM_MAINNET,
37         POLYGON_MAINNET,
38         FLOW_DEVNET,
39         FLOW_MAINNET,
40         APTOS_DEVNET,
41         APTOS_MAINNET,
42         BSC_MAINNET,
43         SUI_DEVNET
44     }
45 }
```

# (Plug n' Play) UI Kit!

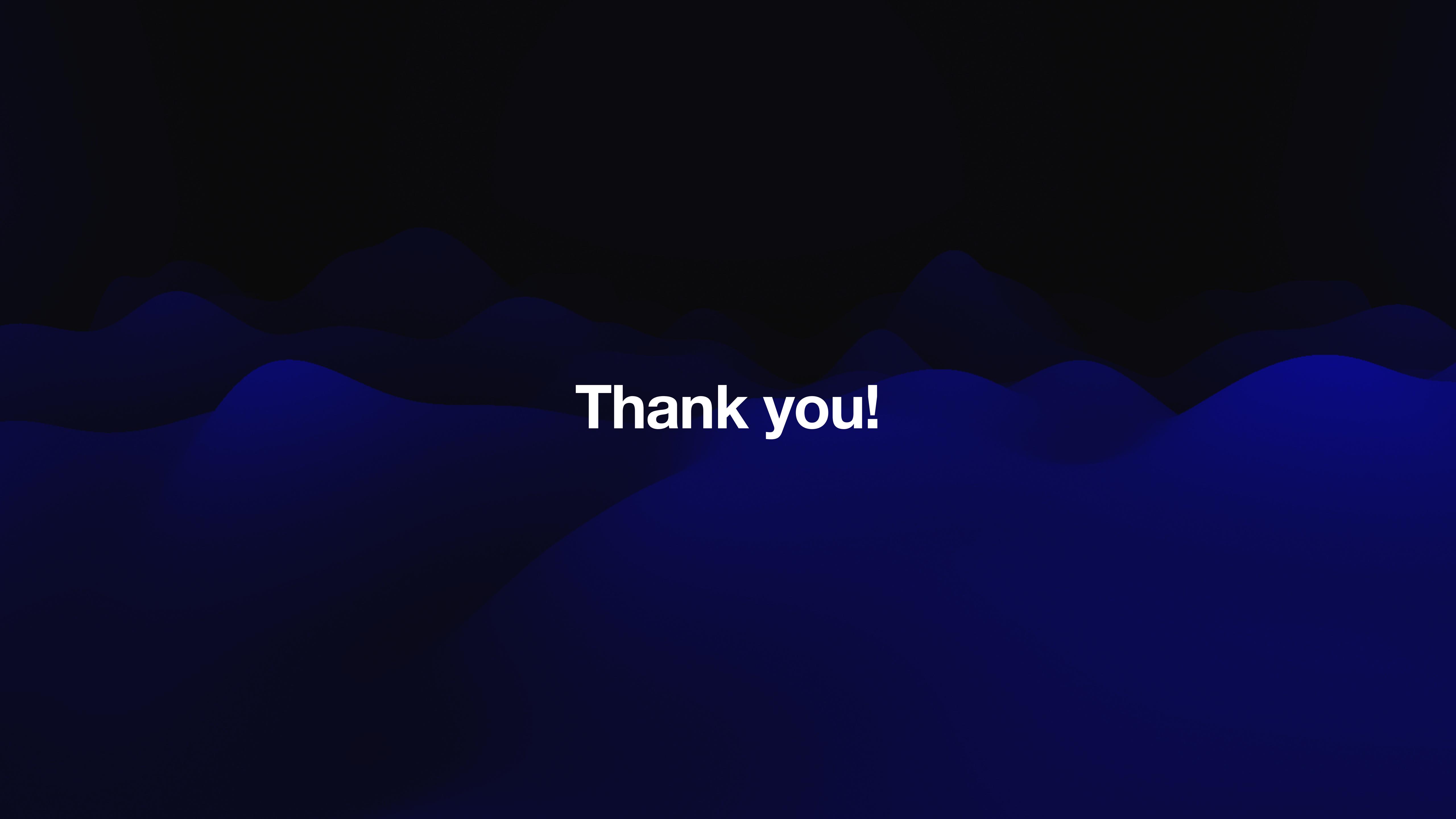


# Designed for multichain Web3 experience!



4 Wallets supported & Easy to integrate new wallets!



The background features three horizontal layers of wavy, organic shapes in a dark navy blue color. The top layer is the darkest, the middle layer is a medium shade, and the bottom layer is the lightest. These waves create a sense of depth and motion.

Thank you!