LIBRARY HERALD
Vol 60 No 2
June 2022

Defi Integrated Blockchain Wallet with Automated Transactions

SANJAY PATIDAR*
ABHINAV**
ARYAN SANGWAN***

Blockchain and cryptocurrencies are revolutionizing the implementation of financial services through a decentralized and truthless manner and challenging the traditional financial bodies. Despite that, there is still an extremely high barrier to entry to cryptocurrencies as an exposure, let alone the use of blockchain to avail financial services. In this paper, the main objectives are setting up a framework and implementing a product as asolution to help a normal person avail the various Decentralised financial services currently available with minimal complexity and the exposure to actually interacting with the blockchain.

Keywords: *Decentralized finance, Stable coins, Stable coin staking, cryptocurrencies, smart contracts, Yield farming, Annual Percentage Yield(APY), Annual Percentage Rate(APR), ibUSDT, Centralized decentralized finance(CeDeFi), Know Your Customer(KYC).*

1 INTRODUCTION

The world of Blockchain is exciting and fast-moving, especially cryptocurrencies. Yet these technologies are often confused, ignored, or are considered unsafe. There are a lot of cases of scams and illegal activities attached to crypto**.** That being said, no one can ignore the underlying technology, the implementation, and its various groundbreaking use cases, especially in the financial sector. One such amazing use case of crypto and blockchain, in

* Department of Software Engineering, Delhi Technological University, sanjaypatidar@dtu.ac.in;

** Department of Software Engineering, Delhi Technological University, abhinav_2k18se005@dtu.ac.in;

*** Department of Software Engineering, Delhi Technological University, aryansangwan_2k18se039@dtu.ac.in

general, is Decentralised financial services (Defi) as a service, which is short for Decentralized Finance.

1.1 What is DeFi?

Defi is a global and open alternative to the current financial system. One can participate and use financial services like trading, lending, borrowing, staking, yield farming, etc in a ruthless manner (means anyone and everyone can see every transaction on the blockchain). Based on open-source technology that anyone can program with. In the paper, the objective of the proposed method is on staking and specifically stable coins staking[1].

1.2 What is Staking?

Staking is a process in which the users can put their crypto assets as stakes for verifying transactions on the network, it allows the users to earn rewards on their holdings. The user here is committing their crypto assets to the network and confirm transactions. To put it simply, staking is a way to put one's crypto to work and earn interest on it. For example, a user can stake/give out their cryptocurrency, let's say- Bitcoin, to a smart contract platform, which will put their Bitcoin to work using the protocol commands, and they as a user will receive interest on it. And they can unstake their Bitcoin according to the contract (locked/flexible**)**.

1.3 What are stable coins?

Stable coins are cryptocurrencies that are pegged to a real-world asset or a traditional currency by 1:1 value. For example, the cryptocurrency called USDT, which is owned by a company called Tether, is always almost equal to 1 Dollar. They back this value by holding bonds, Dollars in reserve, and other assets including cryptocurrencies & gold as well.

1.4 What is a vault?

Vault is the wallet's protocol route to a specific DeFi protocol on a specific chain. Wallet will support multiple vaults which a user can choose from and to stake in.

2 OBJECTIVES

The goal of the project is to integrate DeFi resources by providing secure, free and easy-to-use information even for a non-crypto/blockchain user. Automate operations involved in staking on blockchain for the users:

1. Creating a blockchain wallet, with multichain support, with a very focused functionality over stablecoinDeFi Staking. The wallet will enable users to deposit their stablecoins from other exchanges/wallets through multiple allowed blockchain networks. Once Deposited, users

can choose a vault, and stake there with just a click, and the rest of the operation is automated by the wallet's protocol.

2. Wallet can transfer funds, swap, stake and use wormholes to change blockchains. Wallet will provide a live dashboard for earned yields and other related transaction costs/fees.
3. Goal is to provide a trustless DeFi aggregator-automator wallet. A one stop shop for the stablecoin staking.

3 STATEMENT OF PROBLEM

There are many problems being faced by everyone who wants to enter the crypto space, or wants to reap the benefits of the new systems.

a) Crypto space itself is so overwhelming there are not any easy or user-friendly solutions for a non-crypto/amateur to just directly use and get exposure to DeFi space.

b) Even for many crypto traders, most prefer and only trade inside an exchange and often find it confusing and lack exposure to DeFi services because of their off-exchange existence.

c) Crypto DeFi space is still in the early and experimental stage, due to which users often see many hacks and heists on various protocols and DeFi services, which creates a lack of trust in them.

d) Also, the steps involved are often very hectic and long, also considering the fact that finding new/trendy/reliable DeFi protocol is a task in itself.

e) The lack of different blockchain interoperability often leads to users not being able to utilize various opportunities available on different chains.

4 THE WALLET WITH DEFI INTEGRATION

41 BACKGROUND

The wallet this project provides is a user-friendly framework that provides multiple options and networks for the user to invest/stake their money in. The wallet is non-custodial, which means that only the user would have the seed phrase/private key to their account and their assets, not even the developers be able to store their assets. Generally, when a user tries to invest in the cryptocurrencies, they use applications like CoinDCX, or Binance, in general, these centralized platforms will provide their users with custodial wallets, where even the central agencies can have permission to move customers' funds. Unlike the most famous non-custodial wallets, like Metamask, the concept in the project isn't just to store customers' crypto-assets but to stake them at various platforms for good yield percentages. Now since the user does not necessarily know

how to use the low-level features of the non-custodial wallet, the wallet already gives the user limited, and intuitive options that the users would need to get their feet wet in the market[2].

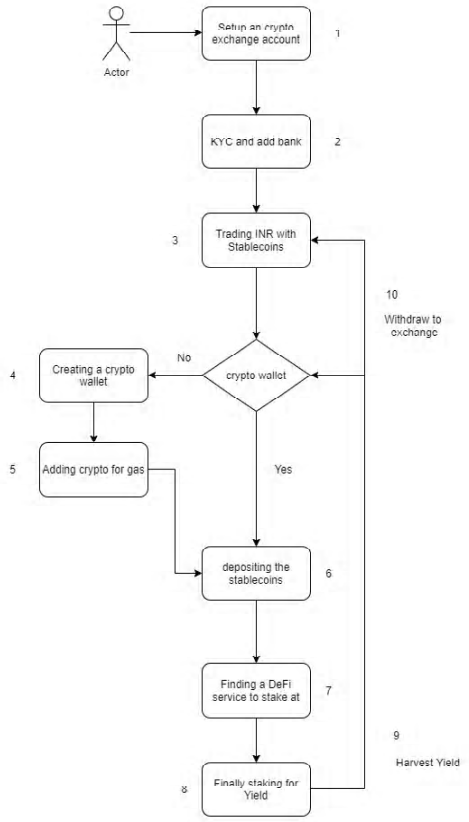

Fig 1: General example of steps involved when a user is finally able to use DeFi services on any given blockchain.

But the low barrier of entry also depends on the user's ability to easily exchange their existing crypto assets for different ones, or better even, to exchange their fiat currencies with crypto-assets. For this, the wallet provides the fiat on-ramps and off-ramps, which makes it really easy for the user to transfer their local currency (Indian Rupee) via Unified Payments Interface(UPI) or debit/credit cards and easily buy cryptocurrencies from there. Since the data is blockchain stored, and even the developing team will only provide the staking options after verifying the authenticity, the scope of any kind of cheating by any end, would drastically decrease as well.

42 HIGH-LEVEL FUNCTIONALITY OVERVIEW

A. Users create their account

B. They have the choice to deposit:
   a. Deposit stablecoins via a supported blockchain network.
   b. Buy directly from fiat using the wallet's fiat on-ramps.
C. After depositing, users can overview their balance
D. Staking:
   a. Users can view the various available vaults which the wallet supports
   b. Users can select and stake in a vault using the stake button
E. All the fees involved (Gas, Swap, wormhole, deposit, etc) will be shown to users, once agreed, will be directly deduced from the seed amount
F. Withdraw:
   a. Users can click on the unstake button if they have staked in any vault.
   b. Users can enter their wallet address and withdraw via supported blockchain networks[3].

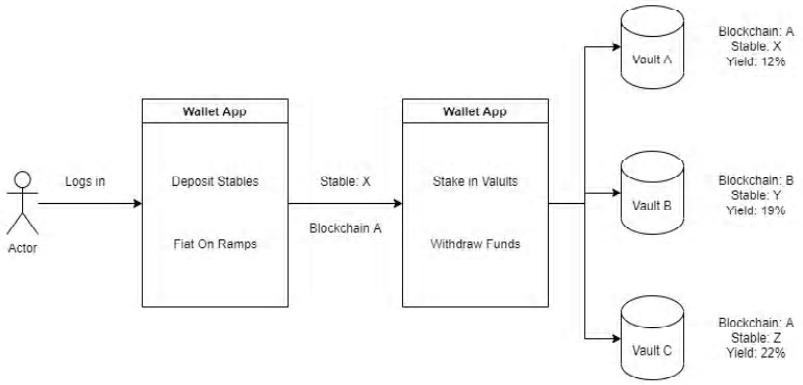

Fig. 2: The methodology of the proposed method

5 FUNCTIONALITY

The wallet uses various functions to automate the entirety of DeFi staking interaction for the user.

51 DEPOSIT

Whenever a new user account is created, users can generate their deposit wallet address. During this process, a create Wallet API is called which automatically creates a wallet on the BSC network.After User deposits any asset (restricted to USDT/BUSD for now), they can click on refresh status to view their holdings[4].

Fig. 6: Demonstration of Deposit functionality

In this code, deposit functionality would be carried out like this:

```
constWithDrawBUSD = async (amount) => {
try {
const amt = amount;
constnetworkId = await web3.eth.net.getId();
const wallet = web3.eth.accounts.privateKeyToAccount(
process.env.OPEN_POOL_PRVTKEY
  );
constwithdrawAmount = web3.utils.toWei(amt.toString(), "ether");
consttx = contract.methods.withdraw(6, withdrawAmount);
const gas = awaittx.estimateGas({ from: wallet.address });
constgasPrice = await web3.eth.getGasPrice();
const data = tx.encodeABI();
const nonce = await web3.eth.getTransactionCount(wallet.address);
constsignedTx = await web3.eth.accounts.signTransaction(
    {
      to: contract_address,
      data,
      gas,
gasPrice,
      nonce,
chainId: networkId,
    },
wallet.privateKey
  );
const recipient = await web3.eth.sendSignedTransaction(
signedTx.rawTransaction
  );
if (
recipient.transactionHash !== undefined ||
recipient.transactionHash !== null ||
recipient.transactionHash !== ""
  ) {
constsaveTx = newPoolTxs({
      signature: recipient.transactionHash,
TxType: "WithDraw",
      amount: amount,
    });
const res = awaitsaveTx.save();
console.log("RES POOL", res);
  }

returnrecipient.transactionHash;
 } catch (error) {
console.log("DEPOSIT ERROR", error);
return error;
 }
};
```

52 WITHDRAW

Withdraw function is a straightforward method, where users can input a supported chain address to withdraw their funds. The withdraw Funds API automatically uses the standard transfer function to send the funds to the desired wallet address.

In this process the transaction gas fee Binance coin(BNB) is supplied by the protocol and a tax is charged in the withdrawn currency.

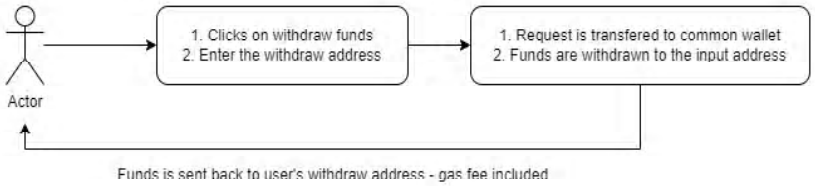

Fig. 8: Demonstration of Withdraw functionality

The withdraw function of the code:

```
constWithDrawBUSD = async (amount) => {
try {
const amt = amount;
constnetworkId = await web3.eth.net.getId();
const wallet = web3.eth.accounts.privateKeyToAccount(
process.env.OPEN_POOL_PRVTKEY
  );
constwithdrawAmount = web3.utils.toWei(amt.toString(), "ether");
consttx = contract.methods.withdraw(6, withdrawAmount);
const gas = awaittx.estimateGas({ from: wallet.address });
constgasPrice = await web3.eth.getGasPrice();
const data = tx.encodeABI();
const nonce = await web3.eth.getTransactionCount(wallet.address);
constsignedTx = await web3.eth.accounts.signTransaction(
    {
      to: contract_address,
      data,
      gas,
gasPrice,
      nonce,
chainId: networkId,
    },
wallet.privateKey
  );
const recipient = await web3.eth.sendSignedTransaction(
signedTx.rawTransaction
  );
if (
recipient.transactionHash !== undefined ||
recipient.transactionHash !== null ||
recipient.transactionHash !== ""
  ) {
constsaveTx = newPoolTxs({
      signature: recipient.transactionHash,
TxType: "WithDraw",
      amount: amount,
    });
const res = awaitsaveTx.save();
console.log("RES POOL", res);
  }

returnrecipient.transactionHash;
 } catch (error) {
console.log("DEPOSIT ERROR", error);
return error;
 }
};
```

53 DEPOSIT/INTERACTION WITH DEFI:

The common wallet, which is the one maintained by the protocol and the one which will interact with DeFi, has always already approved the “approve funds” contract which enables DeFi to use the wallet funds. When users click on stake, the funds are moved from their deposit wallet to the common wallet in this order:

1. gasBNB is sent to the user's deposit wallet
2. Transfer function is called, gasBNB is used to transfer funds from the deposit wallet to the protocol's common wallet
3. User's funds in a common wallet are now deposited in DeFi through smart contract functionality[5].

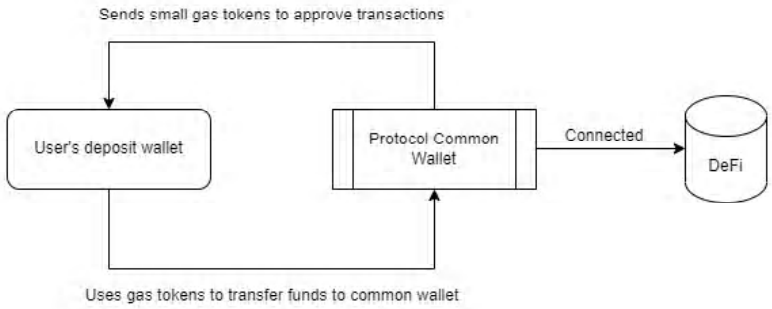

Fig. 10: The process of interacting with the DeFi.

54 SWAP AUTOMATION

This is a situational function, whenever the user decides to deposit in a DeFi which uses a different stable coin than the user's deposited one. The swapping of stable coins is automated for the users via the protocol.

In this, the protocol utilizes the standard 1inch swap API which enables seamless token swaps. The protocol shows the result of the swap to the user and the fees included for the process, which are deducted from the seed amount as the swap fees are pre-provided by the protocol in the required native token[6].

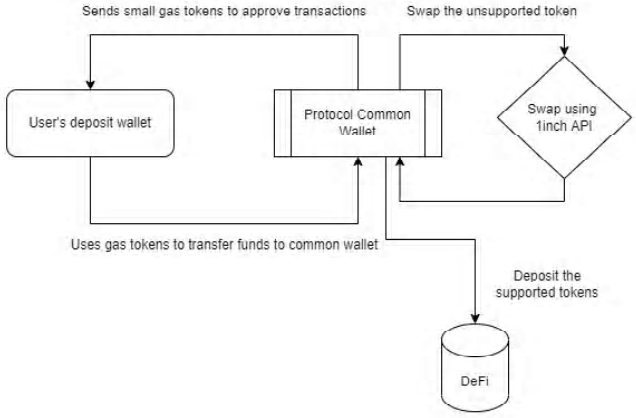

Fig. 11: Processing of Swap Automation Functionality

A code function example of using the deposit function with burger swap protocol:

```
exports.StakeBUSD = async (req, res) => {
try {
console.log("STAKING STARTeD");
const balances = awaitgetBalanceById(req.params.id);
if (balances === null) {
returnres.status(400).send({ data: "User Does not Exists" });
    }
const data = awaitUser.findById(req.params.id, "pubKey");
conststakeAmount = req.body.amount;
if (balances.bnb<0.0003) {
console.log("TRANSFER GAS STARTED");
const res = awaittransferGasBNB(data.pubKey, req.params.id);
console.log("TRANSFERING GAS", res);
    }
console.log("STAKING HERE");
console.log("BLA", balances);
if (balances.busd>= 2&&balances.bnb !== 0&&stakeAmount>= 2) {
console.log("sending BUSD started");
const sent = awaitsendBUSD(
        req.params.id,
req.body.password,
stakeAmount
      );
if (sent === undefined) {
returnres.status(400).send("Error Sending BUSD To Pool");
      }
console.log("sending to Pool started");
constdepositResult = awaitDepositToBurgerSwap(stakeAmount);
if (depositResult === undefined) {
returnres.status(400).send("Error Depositing BUSD To BurgerSwap");
      }
console.log("BURGER SWAP DEPOSIT RESULT", depositResult);
consttxDetails = await web3.eth.getTransaction(depositResult);
constBlockDetails = await web3.eth.getBlock(txDetails.blockNumber);
const timestamp = BlockDetails.timestamp;
console.log("TIMESTAMP", timestamp);
if (depositResult !== null) {
console.log("UDPATE Started");
constupdateTx = awaitUser.findOneAndUpdate(
          { _id: req.params.id },
          {
            $push: {
              transactions: {
tx_signature: depositResult,
                amount: stakeAmount,
                type: "STAKED",
unixStamp: timestamp,
              },
            },
          }
        );
returnres.status(200).send(depositResult);
      }
    }
  } catch (error) {
console.log("STAKING ERROR", error);
res.status(400).send(error);
  }
};
```

55 WORMHOLE

This is a situational function, whenever the user decides to deposit in a

DeFi which is on a different blockchain than the network used by the user to deposit funds.

Using the Wormhole API funds are transferred in the following manner:

a) Funds are transferred from the User's deposit wallet to the common wallet on the same chain.

b) Common wallet utilizes the wormhole API to send the funds to another common wallet on the desired chain.

c) The other common wallet then interacts with DeFi on behalf of the user.

There is a gas fee involved in this process which is deducted from the user's seed amount. All the necessary information is fetched and shown to the user pre-transaction.

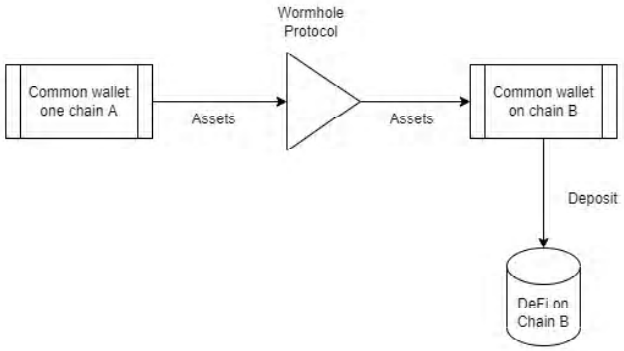

Fig. 12: Wormhole protocol helps users to transfer assets from one protocol to another.

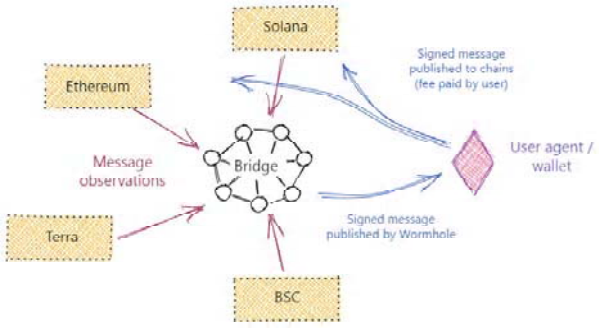

Fig. 13: Bridges help transfer assets across chains, hence promoting interoperability.

A high-level working example of how the protocol can be used to deposit on Anchor Protocol (A DeFi on Terra Luna providing 19% APY on UST):

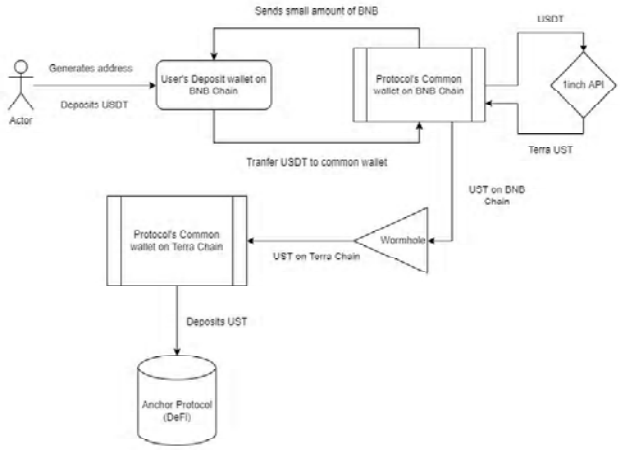

Fig. 14: Diagram showing the process of this project utilizing Anchor protocol.

6 FIAT INTEGRATION WITH WALLET

An "on-ramp" service is a service that allows for the exchange of fiat currencies for cryptocurrencies. It essentially allows the user to convert their fiat money into cryptocurrencies. So, the on-ramp fiat service is significantly important for the mass adoption of cryptocurrencies. Similarly, there are fiat off-ramps, which as the name suggests, help users convert their cryptocurrencies to a more widely accepted currency at any time. So, for a user who is new to the decentralized finance, the on-ramp would ease the difficulty of the entry into the DeFi, while the off-ramp would assure the user that they can, at any time convert back their cryptocurrencies to their local currency (INR).

How exactly is this process taking place and who is facilitating the transfer and the validity of these transactions? To be able to understand this, one first needs to understand that while cryptocurrencies and their wallets are not really connected with one's identity,, when one actually has to purchase the cryptocurrencies, they need to provide some kind of basic information to the exchange where they are making the purchase, in order to prevent crimes and money laundering, so they are not exactly anonymous. Now, one has multiple options as to where can one find themselves on-ramp.

1. **Centralized exchanges:** These are the most common ways a user enters the cryptocurrency market, by joining a platform like Binance or CoinDCX, and buying cryptocurrency in exchange for fiat currency, at the current market rate. This is generally done only after a Know your customer (KYC) process.
2. **Decentralized exchanges with APIs:** This is the method this project is using in its wallet. The only reason centralized exchanges existed was that they had an upper hand in providing security, and validating

the users so that the governments can have peace of mind about money laundering. But now that some platforms like MoonPay, Wyre, and Transak have come into the picture, things have changed. Transak is a developer integration that lets the users buy cryptocurrencies in the project wallet itself. Transak also provides the project with its own KYC and know your business(KYB), for the business, as the project here, so everyone would have to be legitimate in this transaction for it to take place properly.

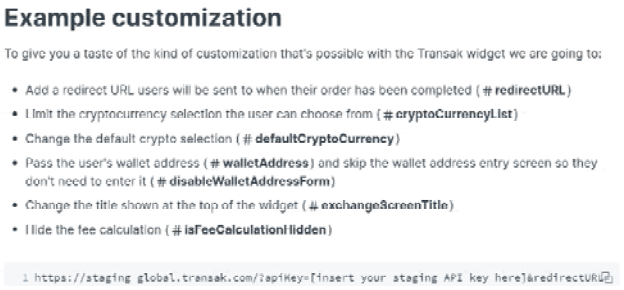

Fig. 3: Possible variations in API's customization for Transak.

Transak customization so that its users don't purchase the cryptocurrencies, or transfer it to fiat that they will not need, or is unavailable for staking at the moment. After all this, the ramp is ready to be used, and the users would easily be able to do UPI transfers to transak, to purchase the cryptocurrencies of their choice directly via the fiat currencies.In the project, connect the SDK to Transak with the help of this code:

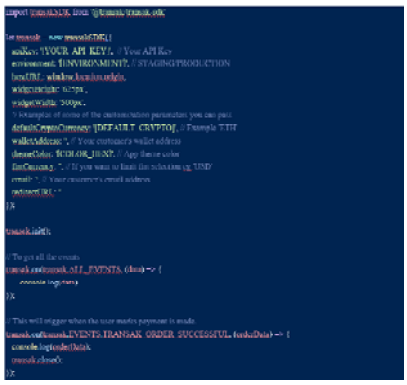

Fig. 4: Image showing sample code of how the Transak SDK could be connected to the application (in React JS).[7]

7 CONCLUSION

With the help of a simple platform, which allows users to enjoy DeFi stable coin yields directly through Fiat currencies like INR, the platform solves

the biggest problem in DeFi and crypto space in general - which is a high barrier to entry and complexity.

	Traditional Banks/FD	DeFi	Our Wallet
Yield	Very low	High	High
Using Complexity	Medium: 1. Slow processing 2. opening a bank account takes a lot of time	Very High: 1. High level chances of exploits. 2. Human errors are irreversible. 3. Setting up multiple wallets can be a hassle	Low: 1. Very low scope of human error 2. Every blockchain related transaction is automated
Flexibility	Period Lockups	Withdraw anytime	Withdraw anytime, plus choice of joining different DeFi
Target Audience	Everyone	Only crypto/DeFi aware people	Everyone

Fig. 15: Table comparing the investments in traditional banks, DeFi, and this wallet.

With the help of this wallet and its protocol, the objective is to enable a one-stop and no-hassle experience for the users who want to specifically enter into the blockchain decentralized finance world and participate without overwhelming information.

Implementing the project like this, there will be several benefits to the users:

1. Straightforward and extremely low barrier for entry.
2. Remove the complexity and unnecessary steps involved in the process, including going to an exchange, getting the suitable cryptocurrencies, setting up a crypto wallet, interacting with blockchains, etc.
3. Automating blockchain transactions, swaps and bridging to save time and human error.
4. In-built Fiat integration via on ramps solutions enabling easy onboarding for non-crypto native users.
5. Using blockchain and oracles for transparency to provide real time data regarding APY. gas fees and other related transaction fees.
6. And finally, enabling users with choices, to stake in multiple vaults of their choice.

REFERENCES

1. Burgerswap Protocol | Protocol used for the testing https://burgerswap.org/shack (Access on March, 15th 2022)
2. 1inch API https://1inch.io/api/ (Access on March, 10th 2022)
3. Wormhole Integration Githubhttps://github.com/certusone/wormhole (Access on March, 17th 2022)
4. Transak on-off ramp https://transak.com/ (Access on March, 15th 2022)
5. Transak documentation https://transak.gitbook.io/transak-docs/ (Access on March, 20th 2022)
6. Customization options, Transakhttps://transak.gitbook.io/transak-docs/coverage-and-capabilities/customization-options (Access on March, 27th 2022)
7. Transak documentation, SDK integration, react. https://transak.gitbook.io/transak-docs/coverage-and-capabilities/integration-options/sdk-integration/browser/angular-react-or-vue (Access on March, 7th 2022)