computers

MDPI

Article

Building DeFi Applications Using Cross-Blockchain Interaction on the Wish Swap Platform

Rita Tsepeleva and Vladimir Korkhov *

Faculty of Applied Mathematics and Control Processes, Saint Petersburg State University,
7/9 Universitetskaya Emb., 199034 St. Petersburg, Russia; st062153@student.spbu.ru
* Correspondence: v.korkhov@spbu.ru

Abstract: Blockchain is a developing technology that can provide users with such advantages as decentralization, data security, and transparency of transactions. Blockchain has many applications, one of them is the decentralized finance (DeFi) industry. DeFi is a huge aggregator of various financial blockchain protocols. At the moment, the total value locked in these protocols reaches USD 82 billion. Every day more and more new users come to DeFi with their investments. The concept of decentralized finance involves the creation of a single ecosystem of many blockchains that interact with each other. The problem of combining and interacting blockchains becomes crucial to enable DeFi. In this paper, we look at the essence of the DeFi industry, the possibilities of overcoming the problem of cross-blockchain interaction, present our approach to solving this problem with the Wish Swap platform, which, in particular, provides improved fault-tolerance for cross-chain interaction by using multiple backend nodes and multisignatures. We analyze the results of the proposed solution and demonstrate how a prototype pre-sale application can be created based on the proposed concept.

Keywords: blockchain; distributed ledger technologies; solidity; smart contracts; decentralized finances; cross-chain

check for updates

Citation: Tsepeleva, R.; Korkhov, V. Building DeFi Applications Using Cross-Blockchain Interaction on the Wish Swap Platform. *Computers* **2022**, *11*, 99. https://doi.org/10.3390/computers11060099

Academic Editor: Osvaldo Gervasi

Received: 6 April 2022
Accepted: 5 June 2022
Published: 16 June 2022

1. Introduction

Blockchain is a developing and promising data storage technology. Unlike a conventional backend, this technology is able to provide its users with full transparency of transactions. This means that absolutely every user will be able to see all the actions that occur within a specific protocol. In addition, unlike a conventional backend, the blockchain stores data in a distributed manner. It helps not to lose data in case of failure of any node. In addition, in the blockchain, the data of each user is absolutely protected. No one can steal your funds if they do not know your private key. All of such advantages make blockchain a trusted technology.

Thanks to these advantages, blockchain is popular in the field of finance and in other spheres. In addition to the obvious advantages of blockchain in the form of reliability, transparency, etc., blockchain technology provides another very important thing. This is the ability to write smart contracts, i.e., executable code that helps users manage their finances independently, without resorting to the help of third parties (for example, the banking sector). However, there is an indisputable fact that smart contracts are not completely safe to use, because a mistake in the code can cost a lot of money. Despite this fact, the industry of decentralized finance is becoming more and more popular. Thanks to the advantages of blockchain, users are not afraid to invest their funds in different blockchain protocols. As a result, there is a huge set of protocols, each of which is a part of the one large ecosystem. How does one find a place for their own ideas in this variety of projects? How can one make their crypto-product as successful as possible? In this article, we propose that users become acquainted with such a DeFi protocol as a launchpad. In order to increase the reach of users of our project, we propose not to limit ourselves to one blockchain network and create a multichain platform using a cross-chain interaction protocol [1]. We will take a closer look at the decentralized finances concept and talk about the most important problem in this area, which, in a certain way, restricts the use of its tools. We will discuss possible solutions of the problem and will analyze them, as well as give our own version of the solution and compare the results.

The rest of this paper is organized as follows:

- Section 2 presents the background on decentralized finances and their problems;
- Section 3 gives an overview of related work in cross-chain interaction;
- Section 4 presents the proposed cross-chain token exchange architecture and implementation;
- Section 5 shows and analyzes a DeFi application example, namely, the Launchpad application, and presents the comparison of the proposed solution with other existing approaches, platforms, and applications;
- Section 6 concludes the paper.

The result of this work is the analysis of the decentralized finance market, the construction of its own solution to the problem of the interaction of individual blockchains, as well as the development of its own DeFi service based on it. In particular, we pay attention to the reliability and fault-tolerance of the cross-chain solution, the flexibility and efficiency of token exchange without unnecessary minting and burning procedures in interacting networks, and the process of building a sample presale application based on cross-chain interactions.

2. Background: Decentralized Finances and Their Cross-Chain Interoperability Challenge

Decentralized Finances (DeFi [2]) is an independent financial ecosystem that gives users a full control over their money without the involvement of governments and banks. Blockchain technologies have played a key role in the implementation of this industry. At the moment, the total value locked in these protocols reaches 82 billion dollars [3].

2.1. The Essence of DeFi

Interaction in decentralized ecosystems occurs without intermediaries according to the P2P scheme; that is, market participants independently cope with making transactions. In essence, DeFi:

1. Saves your time;
2. Saves your money;
3. Allows you to keep your privacy.

What is the reason for the active popularization of DeFi and why are projects increasingly integrating them into their systems? The monetary monopoly of the state is a problem that is quite difficult to solve. The control that we have over our own savings is only relative, and in recent years, this has been felt especially acutely. In many European countries, the population is gradually abandoning cash and switching to electronic payments. Banks, constant monitoring, and lack of anonymity are nuances that are absent in DeFi. App developers cannot influence the money of the participants, and the latter manage their own budget. The main advantages of decentralized finance are:

1. Simplicity and accessibility of technology for ordinary people;
2. Scalability and distribution of the registry; as a result, the system is more secure and resistant to technical errors;
3. Lower system maintenance costs and lower fees (or no fees);
4. Versability—DeFi can be used in almost any area of our life.

2.2. Application of DeFi Services

Developers have already offered quite a lot of options for using decentralized financing. First of all, of course, in the field of banking services.

Mortgages and insurance. The absence of intermediaries and transparency are the optimal conditions for issuing mortgage loans and insurance, including medical insurance. This category could include social benefits—such as pensions.

Lending. Classical lending is primarily a mass of restrictions. Salary requirements, sureties, and the many certifications that need to be obtained make these services inaccessible to a whole category of people. For example, person who spends his time making all of his documents properly might still be waiting for the result, which can be negative, too. DeFi should solve this problem. Moreover, without intermediaries, lending should be cheaper and more reliable. Instant transactions, transparency, and security—what is lacking in this area?

Trading. Cryptocurrency trading has remained popular for quite a long time, and the world knows dozens of trading platforms with different currencies in the listing and trading instruments. Most exchanges are centralized, i.e., they work according to the classical scheme. Decentralized exchanges [4], or DEX, operate without the participation of the administration, i.e., traders enter into a P2P relationship. The funds are not stored in the platform's wallets, which is also very important.

2.3. DeFi and Smart Contracts

Decentralized finance is inextricably linked to smart contracts [5]. Smart contracts are programs written in Turing-complete, modern, high-level programming languages (Solidity [6], for example). The main difference of such program code is that after its publication in the blockchain, the contract not only begins its autonomous work but also loses the ability to edit or change the code. It is also necessary to say that smart contracts are not available in all blockchain platforms. For example, the most popular platform for developing and deploying such contracts is Ethereum [7]. The Binance Smart Chain [8] and Matic [9] also provide such an opportunity. However, such a popular network as Bitcoin does not provide smart contracts. Smart contracts are used in DeFi because they help to fully automate the process, save time, and avoid paperwork.

Smart contracts in such services usually accumulate all the logic. Being the "brain" of the project, they provide users with complex transactions in the blockchain, while clicking only a couple of buttons on the site. Smart contracts consist of functions that represent a set of sequential actions. For example, the usual "transfer" function of a smart contract of any token-contract takes into account the recipient's address, the sender's address, and the amount of coins that the sender wants to give to the recipient. Thus, this function contains the following logic:

1. Lower the sender's balance by amount;
2. Increase the recipient's balance by amount.

Recording new user balance values in the blockchain is the result of the "transfer" function. This is an example of a very primitive function. In reality, functions contain a much larger volume of instructions that are executed sequentially, and each instruction changes the state of the blockchain network.

The development of smart contracts is a rather complex and time-consuming process. This differs from conventional programming because it is necessary to take into account the specifics of the blockchain:

1. The functions that a smart contract includes must be optimized because calling each function is a transaction, and a transaction is a waste of gas, and a waste of gas is money. If the code is not optimized, then users will not use your services due to huge commissions;
2. In the blockchain, it is necessary to monitor the "correctness" of user manipulations. After all, everyone knows that if you sent money to an unverified protocol in the blockchain or accidentally made a transaction to the wrong address, this means that it will never be returned again.

All these points and much more force developers to write and check their code very carefully, so as not to harm users and their assets.

2.4. One of the Major Challenges of DeFi

The concept of decentralized finance involves the creation of a single ecosystem in which there are many blockchains that interact with each other. However, each blockchain network was originally created and conceived as an independent, autonomous unit. In our opinion, it is the main challenge of DeFi. User's funds have been "locked up" for a very long time. Users do not have the ability to transfer their digital assets to another network, and this is a serious restriction on the mobility of their funds. In addition, blockchain developers, while choosing one of the platforms, today have to give up the advantages of using other platforms. They could use several blockchains in their project at once, along with their qualities. Instead of this, they should sacrifice such important development indicators as scalability, speed, low fees, and so on. That is why the issue of combining and interacting blockchain networks has been relevant in recent years.

3. Related Work

The problem of combining and interacting blockchains is of high interest today. The article [10] describes the basic principles of cross-chain transactions, classifies existing studies in three categories: Public Connectors, Blockchain of Blockchains, and Hybrid Connectors. Each category is further divided into sub-categories based on defined criteria. The authors demonstrate that cross-blockchain transactions are not feasible in practice without the participation of a trusted third party. They propose the Blockchain Interoperability Framework (BIF), a framework defining criteria to assess blockchain interoperability solutions. In particular, the authors discuss the cross-chain communication protocol (CCCP) that defines the process by which a pair of homogeneous blockchains interact to synchronize cross-chain transactions correctly.

W. Wang et al. [11] propose the cross-chain transaction processing based on version control. Compared to the existing cross-chain transaction approaches based on locking, the authors propose optimistic approaches in which the updated data can be used immediately, with a rolling back procedure that guarantees atomicity.

The article [12] discusses many different options and suggestions for solving this problem. The authors propose a solution to address cross-communication between blockchain-based systems without an intermediary, present the theoretical cross-communication model, and analyze it in comparison to the so-called mother blockchain model.

In [13], Li et al., propose a blockchain architecture aiming to meet industrial standards that often require interoperation. They use the notion of satellite chains that can privately run different consensus protocols in parallel in order to boost the scalability of the system. This solution also introduces a regulator that oversees the entire network and enforces specific policies by means of smart contracts. The prototype implementation is integrated with Hyperledger Fabric.

H. Wang et al., introduces a blockchain router [14], which enables blockchains to connect and interact. A specific economic model is introduced to enable different blockchains in the network to communicate with each other in a similar way as the Internet. One of blockchains in the system plays the role of a router which analyzes and transmits communication requests, dynamically maintaining a topology structure of the blockchain network according to the communication protocol.

Zamyatin et al. [15] provide a systematic exposition of cross-chain communication protocols and formalize the underlying research problem to show that cross-chain communication is impossible without a trusted third party. The authors develop a framework to design new and evaluate existing CCC protocols, focusing on the inherent trust assumptions. A trusted third party can be centralized or decentralized. An example of centralized trusted parties is proposed in the Hyperledger Cactus as trusted validators [16].

A decentralized trusted party can be another blockchain, which participants agree on the global ledger state via a consensus algorithm.

The Cosmos [17] project is a rapidly expanding ecosystem of independent interconnected blockchains built using special application components (with Cosmos SDK) and connected with the Inter-Blockchain Communication (IBC) protocol. The underlying technology behind Cosmos is the Tendermint Byzantine Fault Tolerance (BFT) consensus algorithm [18] designed to ensure finality, order consistency, and optional availability. Interoperability is achieved through the shared Cosmos Hub powered by the Tendermint consensus, which keeps track of the number of tokens in each connected chain and manages transfers between them. IBC consists of two distinct layers: the transport layer (TAO), which provides the necessary infrastructure to establish secure connections and authenticate data packets between chains, and the application layer, which defines exactly how these data packets should be packaged and interpreted by the sending and receiving chains. The IBC application layer can be used to build a wide range of cross-chain applications, including token transfers, interchain accounts (delegate calls between two chains), non-fungible token transfers, and oracle data feeds [19].

Polkadot [20] is a new-generation blockchain protocol that greatly simplifies cross-chain communication and interoperability by bringing multiple blockchains into one network. Similar to the Cosmos project, the Polkadot project aims to address cross-chain communication and transfer of values and data. The main blockchain in Polkadot is called a 'relay chain', and the connected chains are called 'parachains'. In this network, all the parachains have to adopt a pool consensus operated by the main relay chain. This allows each parachain and the relay chain to utilize the entire network's validators to secure the overall network. If a parachain is compatible with Polkadot, it can connect and use the security of Polkadot's consensus mechanism. Otherwise, the parachain can use a bridge to connect to the Polkadot network.

A blockchain bridge [21] is an interconnected link that provides communication and interaction between two blockchain systems. By connecting two blockchain networks, blockchain bridges help decentralized applications take advantage of both systems, not just their host platform. For example, an application hosted on Ethereum and linked to the EOS blockchain can use the functionality of Ethereum smart contracts, as well as the scalability of EOS. Thanks to blockchain bridges, any data, information and tokens can be transferred between two blockchain platforms. These bridges are regulated by the mint-and-burn protocol. The token transfer does not take place literally; rather, when a token is needed to transfer from one blockchain to another, it is burned on the first, and the equivalent token is minted on the other. An example of a blockchain bridge is the Panama Bridge [22].

Another reason for cross-chain operations is their potential energy efficiency. Different cryptocurrencies operating within different blockchains have different underlying consensus mechanisms. The most famous is the Proof of Work consensus used by Bitcoin and Ethereum blockchains. It involves huge amounts of calculations (thus processing power and energy) for the required mining process compared to some other consensus mechanisms such as Proof of Stake, Proof of Storage, and others that do not need mining. Cryptocurrencies that are based on the latter consensus types use far less energy; transferring the operations to such types of blockchains might bring significant energy efficiency to the whole DeFi infrastructure. To name a few energy-efficient blockchains, we will mention SolarCoin [23], which creates a Solarcoin for every Megawatt hour generated from solar technology, Cardano [24], EOSIO [25], and Tezos [26], which are all more energy efficient than Bitcoin and Ethereum, as they use the Proof of Stake consensus mechanism.

Furthermore, we consider the Polkadot ecosystem and bridges in more detail as they are the most popular cross-chain technologies nowadays. Based on the analysis of their advantages and disadvantages, we build our own solution.

3.1. Polkadot Ecosystem and Polkadot Projects

Polkadot [20] is a network protocol that allows arbitrary data, including tokens and other types of information, to be transferred across blockchains. Polkadot is a multi-chain application environment that enables cross-chain registries and cross-chain computation. The Polkadot blockchain network is secured by a GRANDPA consensus algorithm [27], tailored for the Polkadot (a flavor of a Proof of Stake). The most critical parts of the Polkadot network are the Relay Chain, Parachains, Parathreads, and Bridges (Figure 1).

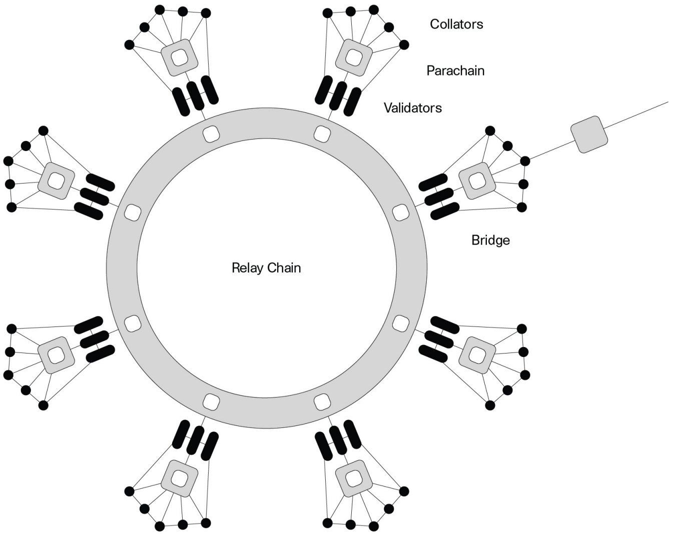

Figure 1. Polkadot ecosystem.

Relay chain is the backbone of Polkadot's network, and it is the main communication hub between parachains. Validators on this chain accept blocks from all parachains and thus provide security for the whole network.

Parachains are independent blockchains that run on top of the Relay Chain and provide chain-specific features to the Polkadot network. Each parachain serves a specialized purpose in the network—think of having a fine-tuned chain for smart contracts, another chain that provides a stable coin for payments between chains, or a parachain which brings a decentralized energy industry to the network. Each parachain is maintained by the collator, which is responsible for producing chain blocks. So, the activity of collators is similar to the work of miners [28] in blockchains. Parachains also benefit from a shared security model provided by the Relay Chain, so they are already secured against 51% of attacks or similar. However, there is only a limited amount of parachains in the network (and the number will be increasing in the future) so there is a system of public auctions where parachain candidates have to compete in order to obtain their own slot.

Parathreads are very similar to parachains from a technical point of view; however, they are very different from an economical standpoint. As we said in the previous paragraph, parachains have to compete in auctions in order to become part of the network. On the other hand, the parathread slot can be leased almost instantly and for only a short period of time. This provides a different way to run projects on the Polkadot — some of those projects can benefit from trying out the network before purchasing an expensive parachain slot, and others can run as a parathreads before they win an auction for a slot.

Bridges are a special kind of parachain. Bridges connect other already running blockchains into the ecosystem (such as a BTC or ETH) and allow for transfers of tokens between Polkadot and outside networks.

The peculiarity of the system is that transactions can be carried out simultaneously and distributed between blockchains. The main goal of the Polkadot ecosystem is to make sure that all participating blockchains remain secure and transactions are carried out in good faith.

The two issues most blockchain-based systems need to solve are scalability (the number of transactions per second the network can handle) and governance (how the community manages protocol upgrades and changes). Polkadot aims to solve both of these problems.

Many different projects are based on the Polkadot technology. Some of them coincide in their essence and purpose with our goals. For example, Polkaswap [29]. Polkaswap is a decentralized exchange for the Polkadot ecosystem. It provides a framework that allows users to connect multiple blockchains using bridges and become an exchange for connecting Polkadot participants and other blockchains for efficient asset trading. The principle of operation and implementation of this project is that Polkadot simplifies the process of combining many assets from as many chains as possible by providing a Host relay chain, a cross-chain message transfer protocol XCMP, and a SPREE module (Shared Protected Runtime Execution Enclaves). In addition to this project, there are several similar ones.

The disadvantages of projects based on Polkadot are as follows. Firstly, the analysis showed that all similar and interesting projects are stuck in the development stage and do not develop further. The second disadvantage is more significant and weighty. It is about security. In projects on Polkadot, the exchange process is implemented as follows:

- To make the transfer of tokens from one blockchain network to another, the process of freezing tokens takes place;
- The freezing of tokens implies the transfer by the user of his funds, which he wants to exchange, to the "storage";
- Next, the logic should be implemented: If the tokens come to the storage, then hold them, and an equivalent amount of them should be credited to the specified address in the target network.

The security issue is that the storage is just an address, not a smart contract. Accordingly, there are no guarantees and protection of users from fraud or technical failure. It can easily happen that the user will send the tokens that will remain in the storage, and the equivalent amount will not come to him.

3.2. Bridges

We briefly introduced blockchain bridges earlier in this section. An example of a blockchain bridge is the Panama Bridge [22], a new solution that allows users to transfer their cross-chain assets from centralized or decentralized wallets to the Binance Smart Chain (BSC). The Panama Bridge provides an API. This means that we can use the Panama Bridge on our platform to exchange the tokens. We can collect POST or GET requests through the form and send them to the bridge. The disadvantage of this approach is its limitation. The Panama Bridge, like the other bridges, connects only two blockchains (in this case, Ethereum and Binance Smart Chain). Thus, it is not possible to talk about a single ecosystem using only one bridge.

Also in this section, it should be mentioned about such technology as sidechains [30]. Sidechains are additional blockchains with an existing independent blockchain. In this paper, we consider blockchains independent of each other as parties to the exchange of information. However, the methods described in this article are also applicable for sidechains. The essence of this technology is simple: suppose that we have the main blockchain network. It starts a child chain (or several such chains) and establishes an exchange between them (for example, through a bridge). Thus, users have the opportunity to transfer assets or other information from the main network to the sidechain and back. Why is this necessary? Firstly, it reduces the load on the main network. Users can perform complex manipulations with their assets and at the same time not load the main blockchain. Secondly, there are blockchains without EVM (Ethereum Virtual Machine) [31] support (such blockchains do not have the ability to reproduce the work of smart contracts). For such blockchains, it is also possible to build a sidechain and enable developers of decentralized applications to use it as a platform for their placement.

4. Cross-Chain Token Exchange Implementation

To implement a mechanism for exchanging tokens between different blockchains, we need some kind of token example to conduct an experiment on it. We will use the WISH token [32] as an example. The WISH token is the main token of the MyWish [33] platform, which offers its services for deploying smart contracts in the blockchain without the need to write a smart contract code and other functionality, in particular the MyWish Crosschain Swap (or Wish Swap) platform service [34]. This is very convenient for users who are not technically educated, but would like to easily join the world of cryptocurrencies. The MyWish platform works with many different blockchains (for example, Tron [35], Ethereum [7], Binance [8], EOS [25], etc.), and users might be interested in learning how to transfer their tokens between these networks using the Wish Swap cross-chain token exchange.

The Wish token has the BEP2 format [36] and is placed in the Binance [8] blockchain, which limits its use. It is necessary to develop a mechanism for exchanging tokens (Figure 2):

- BEP2 Wish to Ethereum tokens ERC-20 and Binance Smart Chain tokens BEP-20;
- Ethereum Tokens ERC-20 and Binance Smart Chain Tokens BEP-20 to BEP2 Wish;
- Ethereum ERC-20 Tokens to Binance Smart Chain BEP-20 Tokens;
- Binance Smart Chain BEP-20 Tokens to Ethereum ERC-20 Tokens.

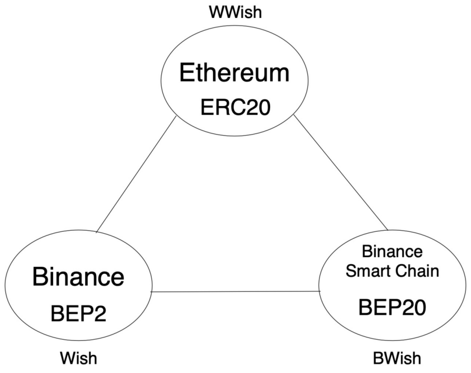

Figure 2. Token exchange.

4.1. Token Smart Contracts

In order to exchange the Wish token, we implemented the contracts of the token analogues in the Binance Smart Chain (BEP20) and Ethereum [7] (ERC-20) networks, with the names BWish and WWish, respectively. All project contracts were developed in the Solidity language. Solidity is a JavaScript-like, object-oriented language for developing smart contracts. It is cross-platform, so it was easy to generalize the task of writing tokens to two platforms. The token contracts, in addition to the standard functions and fields, had to contain the functions and events (events reflected in the transaction log) transferToEthereum/transferToBSC (transfer function to the Ethereum network, or Binance Smart Chain; depending on the network) and transferToBC (transfer of tokens to Binance Chain), as well as transferFromOtherBlockchain. These functions allow tokens to be locked in one network and to send a signal to the backend to unlock it in the other network. The transferFromOtherBlockchain function is only available to the contract owner (in our case, the backend) to prevent the uncontrolled unlocking of tokens. In addition to token contracts, a Python backend and a scanner were implemented.

4.2. Exchange Process with Binance Chain Network

In the Binance Chain network an address for exchange with the Ethereum and Binance Smart Chain blockchains was created. The address is a “swap contract”, and the scanner has to catch transfers to it. Transactions on the Binance network contain a Memo field, which may contain additional information. A user who wants to exchange Wish tokens from the Binance Chain network, via the frontend (Django + React), via binance.org, or via Binance Chain Wallet, sends their BEP2 Wish tokens to this address, filling in the memo field according to the rules. The memo field must contain the name of the network to which the user wants to transfer tokens, as well as their number. The scanner scans operations with the BEP2 Wish token. When a transaction is detected to the exchange address, it sends it to the backend via RabbitMQ. The backend accesses the BEP20 BWish token contract and issues tokens to the user's address minus the set commission (a commission is provided for the transfer in the target network tokens). Tokens in the Binance network cannot be destroyed, and they remain on the exchange address; only the backend has access to sending from the exchange address, so the number of tokens on the exchange address corresponds to the number of tokens in the Binance Smart Chain and Ethereum networks.

4.3. Exchange Process with Binance Smart Chain and Ethereum Networks

The exchange of tokens carried out from these networks is almost similar. The difference is twofold:

- Here, users interact not with the address, but with the smart contract;
- When tokens are transferred to the contract and successfully credited to the target account, they are locked.

4.4. Technical Details of the Exchange Process

The procedure for exchanging Wish tokens from one network to another will be as follows:

- The user specifies the parameters on the page:
    1. From which network the user wants to send tokens;
    2. To which network the user wants the tokens to be sent;
    3. The addresses of the sender (in the first network) and the address of the receiver (in the second network);
    4. The amount of tokens to swap.
- The user clicks the "Submit" button, then the frontend calculates the fee amount and forms the transaction to the swap-smart contract in the sender's network (or creates a transfer transaction on the BC swap-address). This transaction transfers tokens from the sender to our swap-smart contract and creates an event for the scanner;
- The scanner catches the event of the sender's network swap-smart contract and sends it to the backend;
- The backend receives a message from the scanner and creates a transfer-transaction (from the swap-smart contract or BC swap-address) on the receiver's network with current parameters.

4.5. Token Exchange Architecture

The final architecture of the token exchange platform is shown in Figure 3.

The Solidity Token smart contract template includes the following features:

- transferToBC and transferToBSC/transferToEthereum functions (depending on the network)—transfer of tokens in the Binance Chain and Binance Smart Chain/ Ethereum networks;
- transferFromOtherBlockchain—a function that is available only to the contract owner, in our case, the backend; this function is necessary to unlock tokens and send them to a specific address;
- Other ERC-20/BEP-20 standard functions.

The Django + React web application performs the following tasks:

- Collecting information from the user;
- Generating transactions;
- Calling contract functions (locks, transfers, unlocks);
- Calculating transaction fees.

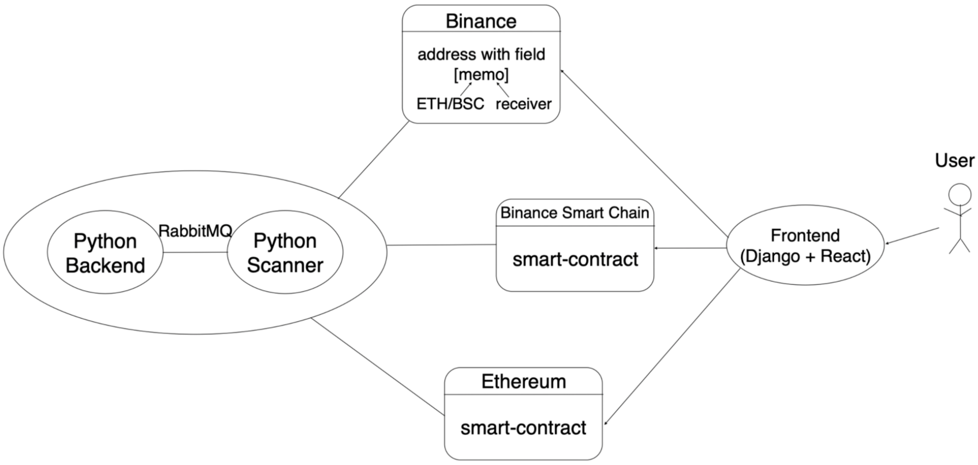

Figure 3. Token exchange architecture.

Ethereum and Binance Smart Chain Network Scanners and Binance Chain Network Scanner: It is important to note that the scanners for the Ethereum and Binance Smart Chain networks are identical, they are configured to check a specific token, token contract, and event. That is, in order for our application to be able to see the call of the functions of our contracts coming from the frontend, we just need to use the API provided by Etherscan [37] (The Ethereum Blockchain Explorer) or BscScan [38] (Binance Smart Chain Explorer) (block-observers). Moreover, the scanner for the Binance Chain network works in a unique way due to the high frequency of transactions in the Binance Chain. Instead of parsing all blocks in a row, a list of transactions involving the address for the last day is requested with some frequency. When you restart the first request, the data for the week are returned. Network registry scanners scan all transactions in an open registry to send confirmation messages to the backend via a queue (RabbitMQ).

5. De-Fi Application Example: The Launchpad

Nowadays one of the most popular types of DeFi applications is a presale app. This type also known as a launchpad platform. The essence of this project type is providing a user's platform on which people can place their own-developed tokens for its presale.

There are a number of reasons why such platforms are popular:

- For investor users, the benefit is that a huge number of different assets are collected in one place.
- For token creators, the benefit is the platform community. Token owners no longer have to look for their own audience.
- Token owners are also freed from the need to independently perform many complex transactions (token sale transfers, liquidity allocation, etc.).
- For future token holders the advantage is transparency and trust. Each investor can open the contract code and see the public price of the token, the percentage of tokens that are guaranteed to go to liquidity pool, the period of liquidity lock (the creator has no an ability withdraw liquidity and collapse the liquidity pool), etc.
- For the platform owner the benefit is an ability to collect fees from users for a different types of services

Such a platform can be built within a single blockchain. In this case, the number of supported assets and users will be limited to the selected network. More practical approach is to combine assets from a variety of blockchains within the platform. In this case, the more blockchains the platform owner will add, then it is more likely that the platform will be successful and bring higher earnings.

However, while creating such a platform, it is necessary to think through the architecture, scalability, security of an application, its performance, and other important characteristics.

5.1. The Main Idea and Components of the Presale Platform

First of all, it is very important to separate the users of the platform from the other owners of cryptocurrency wallets. It will help the owners of the platform to limit their community and users from malicious spam and other information. For example, preventing the publication of a "bad" token for sale, which may not be secured by anything or actually costs nothing.

In order to differentiate the platform users and the rest of the network participants, it is proposed to create the platform's own token and introduce a staking smart contract into the system. Staking is a common name for one kind of decentralized financial protocols. In general, its essence lies in the fact that users send a certain amount of assets to a smart contract in order to increase them (to obtain rewards for it). It is very similar to a bank deposit.

All such protocols differ from each other in the systems of accrual of rewards for deposits. There are many different staking protocols, ranging from trivial annual interest rates (APR, APY), and ending with a "share" system (that means that the smart contract counts each participant's share of the total deposit and awards him a reward accordingly the calculated share).

Thus, the created platform token will be invested by users in the staking protocol in order to be able to participate in all of the other events and to receive even more platform tokens. It confirms the intentions of users regarding the platform and encourages them to behave decently in the platform community.

A ranking system will be compiled on the platform. The more the user makes a deposit, the higher the rank he will acquire. A high rank means that a particular user has some privileges (guaranteed token allocation, a high pool weight, an ability to buy tokens earlier than the other participants, etc.). Such a system will encourage users to invest and take part in presales more and more. In addition, users will receive rewards for their deposits; thereby, in the future, their rank can grow thanks to the rewards. All of such logic details can increase platform's profit.

After the user has made at least a minimum deposit and has become a full-right member of the platform, he receives the rights to create presales and participate in available presales.

If the user is the owner of a new token and he wants to conduct an IDO (Initial DEX Offering) process, he can to fill the form and create a presale contract. The creator provides the following information:

- The token contract's address;
- Amount of tokens for the presale;
- The price per one token.

These are the most important parameters for the sale beginning. Other important parameters are a soft-cap and a hard-cap. These parameters means a soft-minimum and a hard-maximum of raised funds. So, a soft-cap border is a minimum goal of the presale, and a hard-cap is a maximum amount of raised funds (the presale creator can not earn more funds than the hard-cap value). For example, the creator decides to sell one million tokens. The price per one token is ETH 0.0001. So, the maximum of the raised funds will be 1,000,000 * 0.0001 ETH = ETH 100 (a hard-cap). A soft-cap value should be lower or equal to the hard-cap. This position is provided by the creator of the presale and reflects the success of the presale campaign. Usually, if no funds in the amount of a soft-cap or more were collected during the sale process, the presale ends in failure. Therefore, the soft cap is called the minimum boundary. In this case, it is very important to maintain the integrity and transparency of the contract logic: it is necessary to allow users to withdraw their investments in case of an unsuccessful outcome, and the token owner should be allowed to return all their tokens back. The presale may fail for other reasons, but more details on that will be covered later.

Other necessary parameters for conducting an IDO process are DEX parameters. The IDO process implies the initial placement of liquidity on the exchange. It means that the owner of a newly developed token decides to start selling his token. The owner should choose the decentralized exchange—a platform for token sales. There are a lot of such platforms in the crypto community. There are platforms that are very similar to each other, and this fact helps developers easily interact with all of them. However, there are platforms which are ideologically different.

5.2. The Principle of Operation of a Decentralized Exchange (DEX)

To begin with, we will consider only two assets: Token-1 and Token-2. Suppose that User-1 is a creator of Token-1. He wants to start sales of Token-1. Standard exchanges operate on the principle of pools and their reserves. The reserves (more precisely, the ratio of reserves) of current liquidity pool define the price between pool assets. So, to start sales of the Token-1, User-1 needs to create a pool on DEX with his token and with some different assets. For example, User-1 is a holder of Token-2. So, User-1 can create a pool with Token-1 and Token-2. Suppose that User-1 decides to provide x amount of Token-1 and y amount of Token-2 (Token-1's reserve of liquidity pool equals x, Token-2's reserve of liquidity pool equals y). Due to this fact, to buy one Token-1 from this pool, the buyer should pay y/x Token-2.

However, what happens with the liquidity pool in the exchange transaction? At the initial moment, we have x amount of Token-1 and y amount of Token-2. Imagine that we want to buy 1 Token-1. As we explored, we should to provide y/x amount of Token-2. How will the reserves change? The reserve of Token-1 will become $x - 1$ (because we have bought only one token), and the reserve of Token-2 will become $y + y/x$ (because we have provided such payment amount according to the price). It is not difficult to see that the ratio of reserves has changed. This means that the price has also changed. Now, Token-1 becomes more expensive than earlier, and the price per one Token-1 equals to $(x(y+1)+y)/x^2$. That is a main disadvantage of such a system. Everything rests on the will of sellers and buyers. It is assumed that from time to time, the demand will change and the price will fluctuate around the initial one, but no one is immune from a critical situation when one of the reserves becomes empty and the second one becomes worthless.

However, what is the advantage of using such platforms? Previously, so-called order books were used to exchange one crypto asset to another. The principle of the order book working is the creation of an order for selling or buying a certain amount of a certain asset. When someone finds your offer profitable, he agrees to it, and the transaction is completed. However, it should be borne in mind that it was necessary to wait for such a person for a very long time. Moreover, there might not be a person who would agree to the deal at all. In this case, your offer "burned out". Thus, the main advantage of decentralized exchanges is that they support fast exchanges. At any time, users can bring one asset and exchange it for another. In addition, if there is no pool with two assets that you are interested in on the exchange, but there are other pools with these assets, the exchange will still find a way to exchange the tokens you are interested in. For example, User-2 is a holder of the Token-3 and wants to buy Token-2. However, there is no Token-2–Token-3 pool on the DEX, but Token-1–Token-2 and Token-1–Token-3 pools exist. So, the DEX will change Token-3 to Token-1 (according the price in 1–3 pool) and will then change Token-1 to Token-2 (according the 1–2 pool price). In such a situation, Token-1 is a kind of connecting link. Such chains can actually be much longer, but they all work according to the principle described above.

These are not all the advantages of decentralized exchanges. After all, we managed to consider only the user's side, but we have not yet seen everything from the side of the liquidity provider. What is the benefit of a user who invests his funds in a liquidity pool? Decentralized exchanges are designed in such a way that they naturally charge a commission for each exchange transaction (otherwise, why invent such a platform?) and that fee is divided among all liquidity providers whose funds are currently in the pool. Moreover, the commission is divided proportionally to the ownership share of the pool. So, just by having any two crypto-assets, you can earn on them by investing in a liquidity pool.

All that we have described above are the rules and principles of the AMM (automatic market-making) [39] protocols group.

At this stage, we will finish a brief introduction to the work of decentralized exchanges, as we have already managed to assess how useful it is and why people use it. Let us go back to the question of what parameters are needed to conduct the IDO process on the launchpad.

DEX parameter fields include:

- The current DEX platform, which will be used in the liquidity allocation process;
- The listing price—from this price, token sales on the exchange will begin in the future;
- Liquidity percentage allocation—the percentage of the earned funds, which in the specified ratio will go to the liquidity pool. For example, presale's raised funds equal to ETH 0.1. Creator of this presale specified a percentage value equal to 30%, and the listing price equals to ETH 0.0008 per one token. It means that we should take 30% from ETH 0.1 (given that ETH 0.03 goes to the liquidity pool) and divide this number by the price (ETH 0.03/ETH 0.0008 = 37.5 tokens goes to the liquidity pool). This allows the token owner to automatically conduct the initial placement of his token on the exchange, specifying only a couple of parameters;
- Liquidity lock duration—the period of time for which the provided liquidity is "frozen". This procedure protects token buyers from falling prices (the token owner cannot take liquidity from the pool during this period of time and collapse the price);
- Liquidity allocation time—the timestamp after which the function of adding liquidity becomes available;

Of course, instead of studying the whole issue and making a huge number of transactions, it is much easier for the token owner to enter these five values, and the launchpad platform will do everything for him.

Other interesting parameters are the vesting parameters. The vesting process allows the creator of the presale to send tokens to buyers gradually, in equal parts of each period. Vesting parameters include:

- The percentage of purchased tokens that is available immediately after the completion of sales;
- The percentage of tokens that will be released gradually during the vesting process;
- Vesting period—the period of time, each time after which the percentage of tokens is released.

Let us look at this simple mechanism with an example: suppose that User-1 invested ETH 0.01 in Token-1 during the presale process (propose that the price equals ETH 0.0005 per one token). It means that User-1 is supposed to take 20 tokens. However, the creator sets vesting percents (suppose 20% as the first one and 15% as the second one). Thus, the User-1 has the right to take 20 * 20% = 4 tokens at once, and the remaining 16 tokens during the vesting process. Let the vesting period be 1 month. Then, User-1 will take his funds for 6 months (for 5 months User-1 will take 3 tokens per month, and for the sixth, he will receive the remaining one token from the entire amount allotted to him). Such a token release system will help the creator to smoothly distribute the asset in order to avoid sudden rash transactions of holders of this token.

5.3. The Crosschain Part of the Application

As mentioned earlier, it is necessary to provide for the possibility of interaction with various blockchains on the platform in order to expand the number of possible users and increase the number of investments in the project. Everyone knows the fact that each blockchain network is an autonomous, independent unit that does not know how to transmit any information to other blockchain networks. In order to make this possible on our platform, we will use the method summarized in the first sections of this paper and previously presented in the article [1].

In order to participate in presales and create your own, you need to make at least a minimum deposit of project tokens on the staking smart contract. This deposit confirms that you are a full member of the platform. In order to make it easier to keep track of platform users, it is proposed to publish staking contracts only in the most popular blockchain networks. These are an Ethereum blockchain [7] and the Binance Smart Chain [8]. Cross-chain methods can be used for this.

In order to transfer data on the size of user deposits from these networks to other networks to the platform contracts, we use a cross-chain backend and a signature system. The signature system implies that the backend role is introduced on the contract (the smart contract remembers an address of our backend), which will transmit valid information to the contract that it received from outside (in our case from another network). The backend encrypts this information, forms it into a form convenient for the contract to recognize and signs this message with its private key. In turn, when the contract receives a message, it can recognize whether the backend really owns the private key with which this message was signed. If the answer is true, the message is accepted. However, if the answer is false, the transaction fails.

What specific information will we use to compile signatures in our case? Firstly, the most necessary information is the deposit size of a particular user in our staking contract. Secondly, the user's wallet address which corresponds to this deposit is needed. In addition to the address and the amount of tokens deposited to the contract, we must require that the information is up-to-date. Therefore, the third parameter will be the time at which the information was read by the backend (see example in Figure 4).

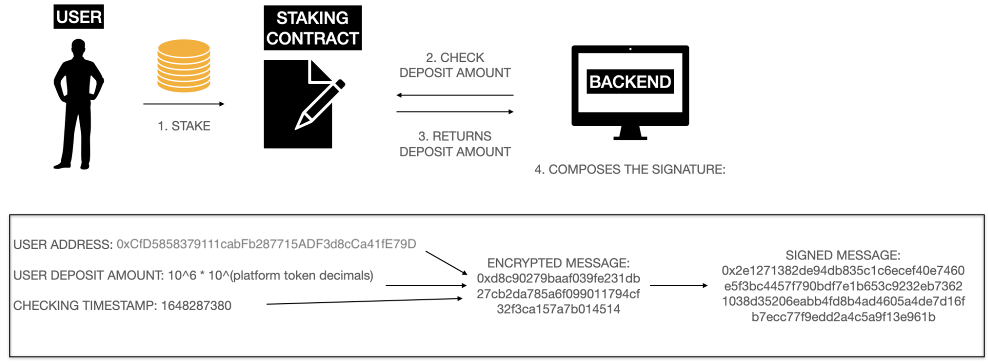
USER
1. STAKE
STAKING CONTRACT
2. CHECK DEPOSIT AMOUNT
3. RETURNS DEPOSIT AMOUNT
BACKEND
4. COMPOSES THE SIGNATURE:
USER ADDRESS: 0xCfD5858379111cabFb287715ADF3d8cCa41fE79D
USER DEPOSIT AMOUNT: 10^6 * 10^(platform token decimals)
CHECKING TIMESTAMP: 1648287380
ENCRYPTED MESSAGE: 0xd8c90279baaf039fe231db27cb2da785a6f099011794cf32f3ca157a7b014514
SIGNED MESSAGE: 0x2e1271382de94db835c1c6ecef40e7460e5f3bc4457f790bdf7e1b653c9232eb73621038d35206eabb4fd8b4ad4605a4de7d16fb7ecc77f9edd2a4c5a9f13e961b

Figure 4. Signature composing.

The figure clearly shows how the backend forms the signature. Firstly, the backend receives a request to create it from the frontend, which provides it with the user's address. For this user, the backend polls the staking contract and receives the amount of his deposit at the moment, and at this time, the backend stores a timestamp when he received these data. Next, the three parameters are converted into one hash, which then signs the backend with its private key.

Such a system was created to supply valid data received from outside (in our case: from another blockchain network) to smart contracts. It is assumed that the contract receives the same three parameters and the final signature for the transaction input. Next, the contract similarly hashes the parameters, receives a hash message, and then sends the message and signature to the recover function. The result of this function is the output of the public address of the account that signed this message. Since the contract remembers all public addresses that can supply valid information to the contract, it will not be difficult for the contract to compare the received address from the recover function with all available ones and decide whether this information really comes from a reliable source. These kinds of operations allow you to prevent the processing of incorrect data by the contract from outside.

To increase the reliability and fault-tolerance of the bridge, it is recommended to use at least three backend nodes at the same time. Using signatures of several backends at once (called "multisignature") will help to handle a situation when one node works incorrectly or breaks. This is one of the significant features that distinguish the presented solution from the others.

The diagram of the next level of bridge with multiple backends and multisignatures is shown in Figure 5. At point 1, swap-transaction arrives at the contract. At point 2, the swap-contract logs an event with swap-transaction details. At point 3, the backend (a set of validators) analyzes transaction parameters and verifies the validity of the transaction. If the transaction is valid the validators sign a message to conduct a receipt transaction in another blockchain. To successfully complete the exchange, 2/3 validators must be confirmed. Further at point 4, the relayers transfer the received signatures to a swap contract on another blockchain and initiate the issuance of tokens to the recipient (point 5).

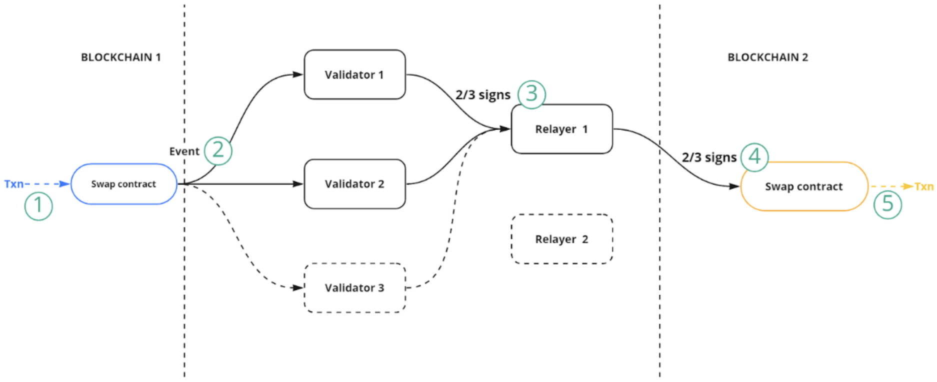

Figure 5. Architecture of the cross-chain bridge with multiple backends and multisignatures.

5.4. Project Architecture

The final architecture of the project is shown in Figure 6.

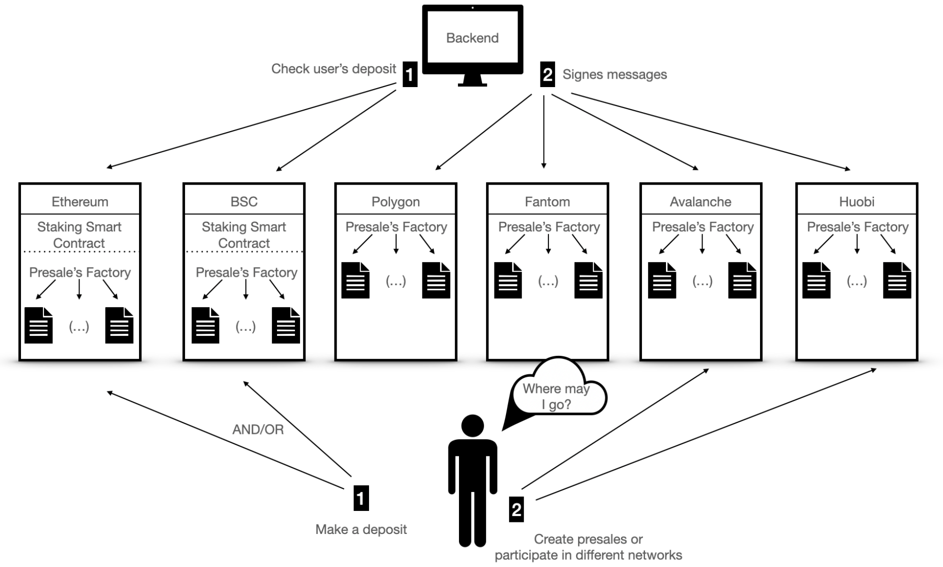

Figure 6. Project architecture.

Thus, in different networks, we have:

- Staking smart contracts which deployed only in Ethereum and Binance Smart Chain networks. Contracts accept deposits from users and send corresponding messages to the backend;
- Presale's factories—smart contracts which receive messages from the backend and deploy presale contracts with the parameters listed above;

- The backend that checks information in Ethereum and Binance Smart Chain networks and sends messages to other networks.

  The backend can send messages to the “main networks”; it extends its actions to absolutely all networks of the project. Similarly, the user can go to any network. The main thing is to make a minimal deposit in one (or both) of the main networks.

5.5. The Presale’s Lifecycle

In this section, we propose to go through the entire presale’s lifecycle in order to better understand each step of the user on our platform:

- The user should be a holder of the platform token (for example, the user can buy this token on a DEX) and approve a staking contract in the Ethereum/BSC blockchain for using his tokens;
- The user should make a minimum deposit to the staking contract on the main blockchains to have an ability to create presales and to participate in others;
- If the user wants to create his own presale, he should follow these steps:
  - Fill the form with all general necessary presale parameters:
    1. The presale token address;
    2. A hardcap and a softcap (max/min boundaries for the earning funds process);
    3. An open timestamp and a close timestamp (time boundaries for the earning funds process);
    4. A token price;
  - Fill the form with all DEX presale parameters:
    1. The DEX address;
    2. A listing price (the price per one token on current DEX);
    3. A liquidity allocation timestamp (the time after which adding liquidity function will be available; it is necessary to be lower than the close presale timestamp);
    4. A liquidity percentage allocation (a percentage of the raised funds that will be allocated to the current DEX in pair with presale token);
  - Approve the presale factory contract (allows smart contract to send the current amount of your tokens). The amount of approved tokens is calculated according to the following formula:

    $$\frac{hardcap}{price_t} + \frac{hardcap * \frac{listing\_percent}{100}}{price_l},$$

    where $price_t$ is a token price, and $price_l$ is a listing price;
  - Sign a transaction with these parameters for creating your own presale contract;
  - Waiting for investment time to pass;
  - If the investment process ends with “Success” (presale did not collect a soft cap payment amount): sign a transaction with a function that will add liquidity to the exchange in the previously specified proportion;
  - If the investment process ends with “Fail” (presale did not collect a soft cap payment amount): sign a transaction with function that will send you presale-tokens back (it is necessary to consider various cases of completion of the investment stage. In a bad case, the creator needs to return the tokens, and investors need to return their investments);
  - When the status of the investment collection is “Success” and liquidity is added to the DEX, the creator of the presale opens the opportunity to withdraw the earned funds, and investors open vesting (investors cannot take the purchased tokens immediately, but in parts, according to the parameters specified by the creator of the campaign);

  - Among other things, when adding liquidity, as mentioned earlier, liquidity is blocked on the presale contract for the period specified by the creator. When this deadline is reached, the owner can take this liquidity back;
- If the user wants to participate in presale campaign, he should follow these steps:
  - Choose a current presale-contract, explore the price per token and how much payment tokens have been already earned;
  - Decide on the investment amount and calculate the token amount value that you will receive at the end;
  - If such values suit you, you can sign an investment transaction (Before the invest transaction starts, you should send an approved transaction to allow the presale contract to obtain your payment tokens);
  - Thus, you can make several transactions with investments until the end of the presale campaign or until the hardcap is set;
  - After the investment process ends, the vesting opens: every investor can claim tokens only once in every vesting period and the amount of tokens that the user takes in every period equals to the total amount of his investment multiplied by the price and multiplied by the vesting percentage;
  - If the presale status is "Fail" after the presale campaign ending, every investor can withdraw his investments, and in this situation, vesting does not open.

User action strategies may differ from each other. Only the main stages of user activity are presented above.

5.6. Analysis of the Results Obtained

In this section we present the comparison of the resulting solutions with similar projects. Table 1 presents a high-level analysis of the cross-chain token exchanges Wish Swap, PolkaSwap, and Panama Bridge.

Table 1. Comparison of cross-chain token exchanges.

	Wish Swap	**PolkaSwap**	**Panama Bridge**
Security	yes	no	yes
Implementation	yes	in future	yes
Versatility	no	yes	no
Scalability	yes	yes	no
Fees	100 WWish or 5 Wish/BWish	0.3%	0.001 BNB

Figure 7 shows histograms that display quantitative estimates of the features of the compared services.

The proposed solution is not inferior to others in the number of connected networks. The solution is also optimized for gas, which allows users to use our service at the lowest cost. The number of different payment methods exceeds this indicator of competitors, and the platform's commission is not the lowest but also not the highest.

Table 2 presents high-level analysis of presale apps: Starter [40], PinkSale [41], and the solution presented in this paper. The choice to compare our solution with these existing solutions is due to the fact that Starter is one of the early prototypes of launchpad platforms, and PinkSale is one of the most popular presale platforms today. In addition, these platforms are trying to connect all popular blockchain networks.

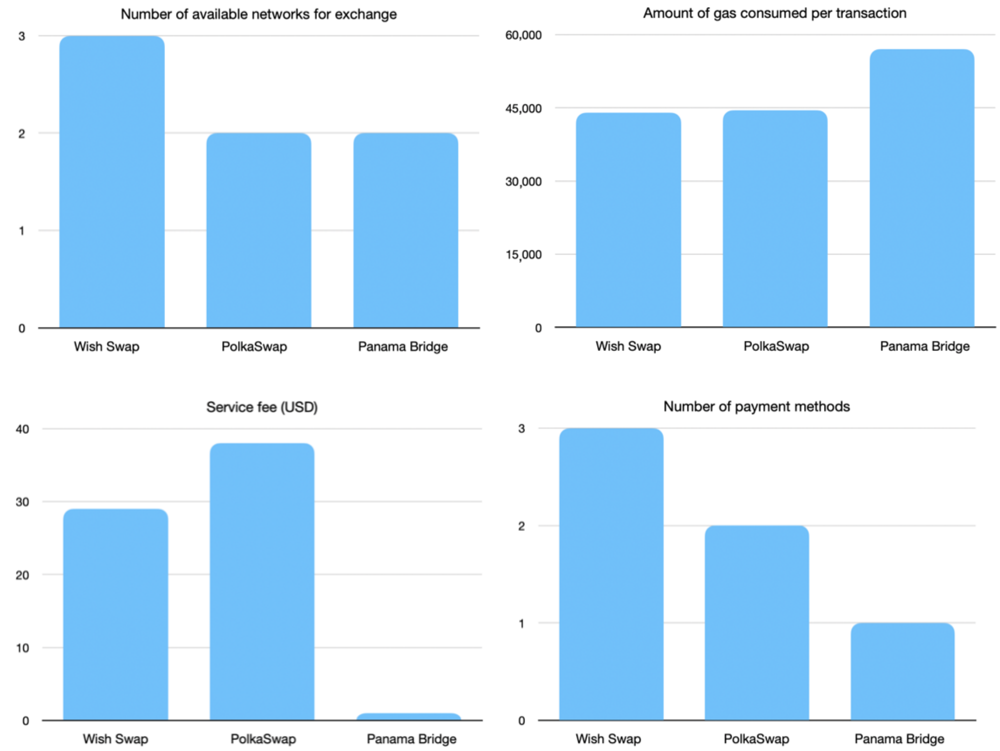

Figure 7. Comparison of Wish Swap with other projects.

Figure 8 shows histograms that display quantitative estimates of the compared presale apps.

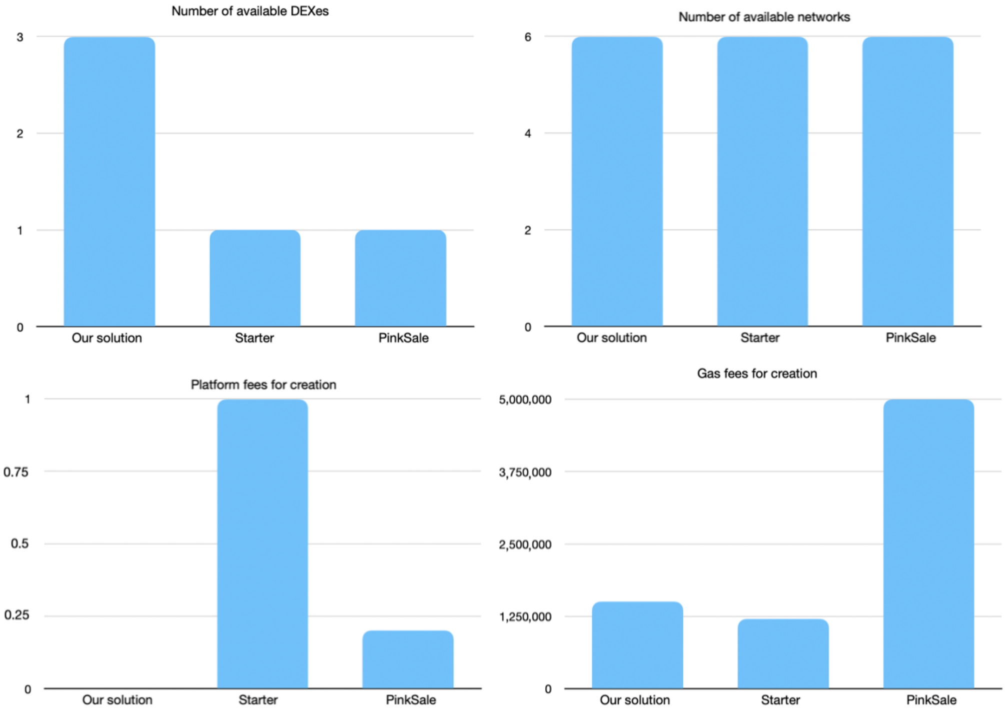

Figure 8. Comparison of proposed solution with other ones.

Table 2. Comparison of presale apps.

	Our Solution	**Starter**	**PinkSale**
Verified contracts (security)	yes	no	no
Cross-chain	yes	no	no
Different strategies	yes	yes	yes
Scalability	yes	yes	yes
Fees	10% from investment	not found	1 BNB/0.2 ETH

According to the histograms, it is clear that the proposed solution meets the security requirements and also has prospects for development when adding new blockchains, such as, for example, Tron, HuobiChain, PulseChain, Waves, and adding new DEXs in different networks.

The proposed solution has all the qualities of a modern reliable decentralized financial application. It is optimized for gas, has a large number of connected networks (which will only grow), and also does not take a commission from the creators of IDO.

6. Conclusions and Future Work

A few years ago, the transfer of tokens and any other information from one network to another was absolutely not possible. However, today, we have proved from our experience that, in this direction, it is possible and necessary to build useful and completely secure solutions to provide users with as much freedom as possible and to remove all possible boundaries in the use of blockchain technology.

In this paper, we present the solution that has improved fault-tolerance for cross-chain interaction between contracts of the platform by using multiple backend nodes while other solutions are normally supported by a single backend. The improvement and optimization of this architecture is our future work. Considering implications and limits, due to the usage of blockchain bridges, the users can allocate the assets only in several available blockchain networks which become available in all the connected networks. However, since the number of connected networks to allocate the assets is limited, it might not fully satisfy some of the users' expectations. However, the list of networks available for the placement of funds is planned to be actively expanded.

The service is planned to be developed further by connecting more new networks. In addition to adding new networks, the service improves the architecture of the blockchain bridges themselves. Currently, a bridge is normally understood as only one node of the backend, which is unreliable. The proposed solution will enable the usage of several backend nodes and multisignatures to increase the fault tolerance and quality of the system.

In addition to the cross-chain asset transfer protocol itself, we have built one of the most popular types of decentralized applications based on it. We have shown that such applications can be scalable, easy to use, and, most importantly, not limited to just one network. Such characteristics will help DApps owners receive many more benefits from their blockchain platforms.

Author Contributions: Conceptualization, R.T. and V.K.; methodology, R.T.; software, R.T.; validation, R.T. and V.K.; formal analysis, R.T.; investigation, R.T.; resources, R.T.; data curation, R.T.; writing—original draft preparation, R.T.; writing—review and editing, V.K.; visualization, R.T.; supervision, V.K.; project administration, V.K.; funding acquisition, V.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tsepeleva, R.; Korkhov, V. Implementation of the Cross-Blockchain Interacting Protocol. In Proceedings of the International Conference on Computational Science and Its Applications (ICCSA), Cagliari, Italy, 13–16 September 2021; Volume 12952, pp. 42–55. [CrossRef]
2. Zetzsche, D.A.; Arner, D.W.; Buckley, R.P. Blockchain disruption and decentralized finance: The rise of decentralized business models. *J. Financ. Regul.* **2020**, *6*, 172–203. [CrossRef]
3. DeFi Pulse—The Decentralized Finance Leaderboard [Electronic Resource]. Available online: https://www.defipulse.com/ (accessed on 5 April 2022).
4. Malamud, S.; Rostek, M. Decentralized exchange. *Am. Econ. Rev.* **2017**, *107*, 3320–3362. [CrossRef]
5. Buterin, V. A Next Generation Smart Contract and Decentralized Application Platform. Ethereum Whitepaper. 2014. Available online: https://ethereum.org/en/whitepaper/ (accessed on 14 June 2022)
6. Solidity Language Documentation. Available online: https://docs.soliditylang.org/en/v0.8.4/ (accessed on 5 April 2022).
7. Ethereum [Electronic Resource]. Available online: https://ethereum.org (accessed on 5 April 2022).
8. Binance [Electronic Resource]. Available online: https://docs.binance.org (accessed on 5 April 2022).
9. Matic [Electronic Resource]. Available online: https://matic.network/ (accessed on 5 April 2022).
10. Belchior, R.; Vasconcelos, A.; Guerreiro, S.; Correia, M. A Survey on Blockchain Interoperability: Past, Present, and Future Trends. *ACM Comput. Surv.* **2022**, *54*, 168. [CrossRef]
11. Wang, W.; Zhang, Z.; Wang, G.; Yuan, Y. Efficient Cross-Chain Transaction Processing on Blockchains. *Appl. Sci.* **2022**, *12*, 4434. [CrossRef]
12. Pillai, B.; Biswas, K.; Muthukkumarasamy, V. Cross-chain interoperability among blockchain-based systems using transactions. *Knowl. Eng. Rev.* **2020**, *35*, e23. [CrossRef]
13. Li, W.; Sforzin, A.; Fedorov, S.; Karame, G.O. Towards scalable and private industrial blockchains. In Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts, Abu Dhabi, United Arab Emirates, 2 April 2017; pp. 9–14. [CrossRef]
14. Wang, H.; Cen, Y.; Li, X. Blockchain router: A cross-chain communication protocol. In Proceedings of the 6th International Conference on Informatics, Environment, Energy and Applications, Jeju Island, Korea, 29–31 March 2017.
15. Zamyatin, A.; Al-Bassam, M.; Zindros, D.; Kokoris-Kogias, E.; Moreno-Sanchez, P.; Kiayias, A.; Knottenbelt, W.J. SoK: Communication across Distributed Ledgers. Technical Report. 2019. Available online: https://eprint.iacr.org/2019/1128.pdf (accessed on 5 April 2022).
16. Montgomery, H.; Borne-Pons, H.; Hamilton, J.; Bowman, M.; Somogyvari, P.; Fujimoto, S.; Takeuchi, T.; Kuhrt, T.; Belchior, R. Hyperledger Cactus Whitepaper. 2020. Available online: https://github.com/hyperledger/cactus/blob/main/whitepaper/whitepaper.md (accessed on 5 April 2022).
17. Cosmos. Available online: https://cosmos.network (accessed on 5 April 2022).
18. Kwon, J. Draft v.0.6 (Outdated). Tendermint: Consensus without Mining. 2014. Available online: https://tendermint.com/static/docs/tendermint.pdf (accessed on 5 April 2022).
19. IBC Protocol. Available online: https://ibcprotocol.org/ (accessed on 5 April 2022).
20. Polkadot [Electronic Resource]. Available online: https://wiki.polkadot.network/docs (accessed on 5 April 2022).
21. Stone, D. Trustless, privacy-preserving blockchain bridges. *arXiv* **2021**, arXiv:2102.04660.
22. Official Binance Panama Bridge Webpage. Available online: https://www.binance.org/en/bridge (accessed on 5 April 2022).
23. Solarcoin Is a Cryptocurrency That Incentivizes a Solar-Powered Planet. [Electronic Resource]. Available online: https://solarcoin.org (accessed on 16 May 2022).
24. Cardano [Electronic Resource]. Available online: https://cardano.org/ (accessed on 16 May 2022).
25. EOS Network [Electronic Resource]. Available online: https://eos.io/ (accessed on 5 April 2022).
26. Tezos [Electronic Resource]. Available online: https://tezos.com/ (accessed on 16 May 2022).
27. Stewart, A.; Kokoris-Kogia, E. GRANDPA: A Byzantie finality gadget. *arXiv* **2020**, arXiv:2007.01560.
28. Wang, W.; Hoang, D.T.; Hu, P.; Xiong, Z.; Niyato, D.; Wang, P.; Wen, Y.; Kim, D.I. A Survey on Consensus Mechanisms and Mining Strategy Management in Blockchain Networks. *IEEE Access* **2019**, *7*, 22328–22370. [CrossRef]
29. Official Polkaswap Webpage. Available online: https://polkaswap.io/ (accessed on 5 April 2022).
30. Back, A.; Corallo, M.; Dashjr, L.; Friedenbach, M.; Maxwell, G.; Miller, A.; Poelstra, A.; Timon, J.; Wuille, P. Enabling Blockchain Innovations with Pegged Sidechains. Technical Report. 2014. Available online: https://blockstream.com/sidechains.pdf (accessed on 14 June 2022).
31. Hildenbrandt, E.; Saxena, M.; Rodrigues, N.; Zhu, X.; Daian, P.; Guth, D. KEVM: A Complete Formal Semantics of the Ethereum Virtual Machine. In Proceedings of the 2018 IEEE 31st Computer Security Foundations Symposium (CSF), Oxford, UK, 9–12 July 2018.
32. Wish BEP2 Token [Electronic Resource]. Available online: https://explorer.binance.org/asset/WISH-2D5 (accessed on 5 April 2022).
33. MyWish Platform [Electronic Resource]. Available online: https://mywish.io/ (accessed on 5 April 2022).
34. MyWish Crosschain Swap (Wish Swap). Available online: https://bridge.mywish.io/ (accessed on 5 April 2022).
35. Tron Network [Electronic Resource]. Available online: https://tron.network/ (accessed on 5 April 2022).

36. The Definition of BEP2 Token Standard by Binance Academy [Electronic Resource]. Available online: https://academy.binance.com/en/glossary/bep-2 (accessed on 5 April 2022).
37. Etherscan (Ethereum Explorer) [Electronic Resource]. Available online: https://etherscan.io/ (accessed on 5 April 2022).
38. BscScan (Binance Smart Chain Explorer) [Electronic Resource]. Available online: https://bscscan.com/ (accessed on 5 April 2022).
39. Mohan, V. Automated Market Makers and Decentralized Exchanges: A DeFi Primer. *Financ. Innov.* ***2022***, 8, *20*. [CrossRef]
40. Starter Launchpad [Electronic Resource]. Available online: https://starter.xyz/#/home (accessed on 5 April 2022).
41. PinkSale Launchpad [Electronic Resource]. Available online: https://www.pinksale.finance/#/ (accessed on 5 April 2022).