Blockchain-based DeFi Crowdfunding Platform

Final Report

Zhu Wendi (UID 3035638521)

Supervisor: Dr. Chow, Kam Pui

April 2023
Abstract

Crowdfunding allows individuals and organizations to collect resources for specific goals, but traditional third-party platforms suffer from issues such as lack of transparency, high management fees, and low efficiency. Meanwhile, the blockchain can make the circulation of assets transparent and allows users to cooperate without a centralized organization, and the smart contracts can provide a collaborative environment that users can trust. This report suggests that Decentralized finance (DeFi), enabled by blockchain technology and smart contracts can be a potential solution for building a better crowdfunding platform. This project aims to build a blockchain-based decentralized crowdfunding platform, providing more control to investors and easier fundraising for fundraisers. A decentralized application (D-App) is proposed to build this platform, which consists of two layers: blockchain layer and frontend layer. The blockchain layer is currently developed and tested on Evmos, which is a Ethereum Virtual Machine (EVM) supported blockchain in Cosmos ecosystem. Cosmos is a decentralized blockchain network that allows developers to customize their own blockchain. Cosmos is characterized by modularity and interoperability, enabling communication and interaction between different blockchains. It uses the Tendermint consensus algorithm, providing high security and scalability. The frontend layer is a React web page based on Web3.js. Frontend can interact with the blockchain layer and displays campaign information to users. Web3 is the next generation of the internet built on blockchain technology, which aims to establish a decentralized ecosystem that allows users to transact, collaborate, and communicate without the need for centralized institutions or platforms.

Keywords: Crowdfunding, Blockchain Technology, De-Fi, Web3, D-APP, Smart Contract, Cryptocurrency
Acknowledgements

The four years of study experience at the University of Hong Kong is significant to me. These four years have enabled me, from an ignorant student with little insight into the world and the society, to step by step through independent learner, practitioner, and explorer.

First, I would like to express my sincere gratitude to my supervisor Dr. Chow, Kam Pui for his invaluable support, whose insights and feedback have been instrumental in shaping my research and enhancing my understanding of the subject matter. I would also like to extend my thanks to the HKU CS Department and other professors for their assistance and help to my academic journey. Their knowledge and expertise have been instrumental in my growth and development as an undergraduate student. I am truly grateful for their time, effort, and commitment to my academic journey, and I will always cherish their guidance and mentorship.

Next, I would like to express my heartfelt appreciation to my parents for their unwavering support and encouragement throughout my life-long study. Their constant love, understanding, and emotional support have been my anchor during times of challenge and uncertainty. I am grateful for their sacrifices and education, which have enabled me to pursue my goals and dreams. Also, I would also like to extend my thanks to my friends for their inspiration and assistance. Their guidance and support have helped me grow both personally and academically. Their company has been a source of joy and comfort, and I am grateful for the memories we have shared together. I am deeply indebted to my parents and friends for their belief in me. Without their encouragement and assistance, I would not have been able to achieve my goals.
Contents

Abstract ................................................................................................................................. ii
Acknowledgments .................................................................................................................. iii
Table of Contents .................................................................................................................. iv
List of Figures ......................................................................................................................... v
List of Tables .......................................................................................................................... vi
Abbreviations ......................................................................................................................... vii

1. Introduction ...................................................................................................................... 1
   1.1 Project Background ...................................................................................................... 1
   1.2 Objectives and Scope ................................................................................................. 1
   1.3 Report Outline ........................................................................................................... 2

2. Background ...................................................................................................................... 3
   2.1 Crowdfunding Market Introduction ........................................................................... 3
   2.2 Decentralized Finance .............................................................................................. 4
   2.3 Motivations ................................................................................................................ 5

3. Methodology .................................................................................................................... 6
   3.1 Blockchain .................................................................................................................. 6
      3.1.1 Alternative Analysis ............................................................................................ 6
         3.1.1.1 Ethereum ....................................................................................................... 7
         3.1.1.2 Ethereum Layer 2 ......................................................................................... 8
      3.1.2 Cosmos ............................................................................................................... 8
         3.1.2.1 Evmos .......................................................................................................... 9
   3.2 Smart Contract .......................................................................................................... 10
      3.2.1 Remix IDE ......................................................................................................... 10
      3.2.2 Truffle ............................................................................................................... 10
   3.3 Frontend ..................................................................................................................... 10
      3.3.1 React ............................................................................................................... 11
      3.3.2 Web3 ................................................................................................................. 11
      3.3.3 MetaMask ....................................................................................................... 11

4. Application Design .......................................................................................................... 12
   4.1 General Introduction ................................................................................................. 12
      4.1.1 Fundraisers Operation Instructions ...................................................................... 12
      4.1.2 Backers Operation Instructions .......................................................................... 13
      4.1.3 Voting Power ..................................................................................................... 13
   4.2 Application Architecture ............................................................................................ 13
   4.3 Campaign Design ....................................................................................................... 14
   4.4 Frontend Design ......................................................................................................... 15

5. Conclusion ....................................................................................................................... 19
References ............................................................................................................................. 20
List of Figures

Figure 1.1 Traditional Application Architecture
Figure 1.2 D-App Architecture
Figure 2 General Architecture
Figure 3 Campaign Workflow
Figure 4 Home Page UI
Figure 5 Create Campaign UI
Figure 6 Campaign Detail Page UI
Figure 7 Update Info UI
Figure 8 Create Request UI
Figure 9.1 Sign Transaction
Figure 9.2 View Account Detail
Figure 10 Block Explore
List of Tables

Table 1  Top 6 DeFi Projects sorted by the marked cap
Table 1  Evmos Testnet Configuration
Abbreviations

DeFi  Decentralized Finance
CFP  Crowdfunding Platform
D-App  Decentralized Application
ETH  Ethereum
ERC-20  Ethereum Request for Comments 20
OOP  Object-oriented Programming
EVM  Ethereum Virtual Machine
POW  Proof-of-Work
POS  Proof-of-Stake
TPS  Transactions Per Second
BFT  Byzantine Fault Tolerance
UTXO  Unspent Transaction Output
REST  Representational State Transfer
API  Application Programming Interface
1. Introduction

1.1 Project Background

Crowdfunding activities allow people to collect resources from others to support their own projects. Crowdfunding platforms have become a popular alternative to traditional fundraising methods because of their accessibility, low barriers to entry, and ability to reach a large audience. The platforms offer fundraisers the opportunity to showcase their ideas to potential backers and receive funding in exchange for rewards or equity, and they also make small-scale philanthropic projects a reality [1]. Crowdfunding platforms also provide backers with a sense of community and the chance to support projects they believe in. However, due to its reliance on third-party platforms, fundraisers are facing many restrictions on their crowdfunding projects, and traditional crowdfunding platforms are criticized for their lack of transparency, high management fees, and limited investor control.

Meanwhile, with the rise of blockchain technology and the cryptocurrency market, decentralized finance (DeFi) can be seen as a potential solution to different kinds of financial projects. DeFi is enabled by blockchain technology and smart contracts, which can provide users with all kinds of financial instruments without relying on centralized intermediaries [2]. The transparency of transactions endured by the blockchain, with the consensus mechanism ensured by smart contracts can create a distributed platform users can trust.

1.2 Objectives and Scope

The project aims to explore the possibility and challenges of building a blockchain-based crowdfunding platform, analyze its potential usage scenarios and features, and develop a decentralized application that allows users to customize their own fundraising campaigns easily and participate in various campaigns at the same time.

The scope of the project is described below:

1. Smart Contract Templates. The templates include the contract factory (CampaignFactory.sol) that is initially deployed on the blockchain and the crowdfunding contract (Campaign.sol) that generate crowdfunding pools on the blockchain. Users can call the existing contract factory to deploy their own crowdfunding contracts. Users also can directly call functions in the contracts interact with different campaigns.

2. Decentralized Application. The D-App preforms as a web page based on React, which can directly communicate with the blockchain without a server. Users can use this web page to query and send data to the blockchain, including but not limited to viewing the campaigns information, creating campaigns, creating and approving withdraw requests. Users need to rely on cryptocurrency wallets as browser plugins to send transactions or requests to the blockchain.
1.3 Project Background

The rest of this report is organized as follows. In Chapter 2, the report first gives a brief introduction to the crowdfunding market and decentralized finance, then reviews the existing popular DeFi projects. In Chapter 3, the report focuses on the methodology used in developing the blockchain-based decentralized application and discusses about some related technical details and tools in developing. In Chapter 4, the report first presents the general design and features of the application, and details the application architecture, followed by showing the details of the development of the smart contract and front-end. In Chapter 5, the report concludes the project and talks about the limitations and possible future works.
2. Background

In this chapter the report talks about the background of crowdfunding business and introduces the history and usage of decentralized finance (DeFi), then analyzes some of the most popular DeFi projects in the market. Next, the report draws out the motivation for the project based on these background presentations.

2.1 Crowdfunding Market Introduction

Crowdfunding allows users to collect resources by getting a large number of investors or donors to each give a small amount of money, which can be seen as an open call to provide financial resources. Thanks to the diffusion nature of the Internet, fundraisers can easily post their ideas or specific statements on a platform to attract funds to support their activities. Crowdfunding mostly takes place on crowdfunding platforms (CFPs) [1]. Depending on the use of the fund and the purpose of the fundraising, crowdfunding activities can be divided into profitable and non-profitable fundraising.

Profitable crowdfunding (reward-based) can be used for some small-scale product developing and launching, which allows small companies and individuals to raise funds for their projects more efficiently. At the same time, users who participate in crowdfunding can be considered backers [1]. The backers can be offered some incentives or rewards, such as early bird discount, an opportunity to pre-order and have a say in the product development, or even own equity in an early or growth-stage company. Compared to bank loans, crowdfunding is more friendly to project sponsors because it does not require asset collateral, has no interest rate, and has a high fault tolerance rate [3]. Additionally, backers can have a tighter connection to the project because they are also potential customers, which makes the products more in line with the demand. For small start-ups, crowdfunding may also offer more opportunities and flexibility than traditional financing or attracting investors to invest. In addition, lending-based fundraising can also be seen as a reward-based fundraising, which allows fundraisers to lend money for some purpose and repaying the loan with interest over a period. At the same time, the flexibility and ease of crowdfunding has led some regional governments to encourage crowdfunding as a source of capital for new ventures [3]. However, there are some uncertainties with this new financing method, such as projects often face delays or failures in the middle of the process [3].

Nonprofit crowdfunding (donation-based) is a form of fundraising encouraging the public to invest in nonprofitable organizations (NPOs) or nonprofitable projects for charitable purposes. It can be used for specific programs within the organization or a general donation to the cause [1][4]. This donation-based crowdfunding business model is being used extensively by different charity organizations and is applicable to many areas for financing social causes, such as education, social care, artistic, academic research, culture, community development and environmental related [4]. At the same time, online fundraising platforms are also seen as a channel for advocacy. Charities can use the platforms for advertising, user collection and market research to help them operate better [5].
Today there are many third-party CFPs that provide publishing channels and asset management for crowdfunding projects (such as Kickstarter, Indiegogo, Patreon, Mightycause, Donorbox, etc.). However, the traditional commercial model still has many technical drawbacks. For example, the platform may take a commission, different platforms have different standards and processes, and the funds could go opaque.

2.2 Decentralized Finance

Decentralized finance (DeFi) is a financial infrastructure based on blockchain. The transaction ledger is distributed on each decentralized node, and the entire blockchain is maintained by all nodes together. DeFi can offer financial instruments without relying on centralized organizations or intermediaries such as brokerages, exchanges, or banks using smart contracts on a blockchain [2]. Each agreement is enforced by code, and legitimate state changes persist on a public blockchain, which can create an immutable and interoperable financial system with unprecedented transparency, equal access rights, and little need for custodians, central clearing houses, or escrow services. In the financial industry, blockchain technology has enabled the rise of decentralized financial services that are often more innovative, interoperable, borderless, and transparent [6]. With the help of blockchain, transaction costs can be reduced, and the distributed trust can empower decentralized platforms that have the potential to become the new foundation for decentralized business models.

DeFi revolves around decentralized applications (D-Apps), which are typically accessed through open-source software like browser extensions or applications [7]. Furthermore, D-Apps can be linked to the blockchain to let users operate with complex financial services. There are numerous Defi projects that have a wide range of uses (for example, token issuance, automated coin exchange, decentralized lending platforms, decentralized derivatives, On-Chain Asset Management, etc.) [2]. Some of the most popular DeFi projects (according to the market cap) are shown in the Table 1, the projects’ goal can be stable coin (can based on different algorithms), automated exchange, or provide blockchain framework for other DeFi projects [8]. It can be inferred that most popular DeFi projects are related to the management and liquidity of financial resources in the cryptocurrency market.

<table>
<thead>
<tr>
<th>Token</th>
<th>Market Cap</th>
<th>Blockchain</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dai</td>
<td>6.2 billion</td>
<td>ETH (ERC-20)</td>
<td>Stable Coin (US Dollar)</td>
</tr>
<tr>
<td>Uniswap</td>
<td>5.1 billion</td>
<td>ETH (ERC-20)</td>
<td>Automated Exchange</td>
</tr>
<tr>
<td>Avalanche</td>
<td>5.0 billion</td>
<td>Avalanche</td>
<td>Blockchain platform</td>
</tr>
<tr>
<td>Wrapped Bitcoin</td>
<td>4.9 billion</td>
<td>ETH(ERC-20)</td>
<td>Stable Coin (Bitcoin)</td>
</tr>
<tr>
<td>Chainlink</td>
<td>3.5 billion</td>
<td>ETH(ERC-20)</td>
<td>Oracle networks</td>
</tr>
<tr>
<td>Luna</td>
<td>1.6 billion</td>
<td>Terra</td>
<td>Decentralized Stable Coin</td>
</tr>
</tbody>
</table>

Table 2 Top 6 DeFi Projects sorted by the marked cap [8]

The development of DeFi still faces many challenges. We can find from the table above that most of the popular DeFi projects, except for stablecoins, are related to on-chain assets but do not have much connection to the real economy. How to make DeFi work...
with traditional financial services and provide security and convenience to users' lives is the problem that DeFi needs to solve. Another challenge is the security risk associated with anonymous networks. Since DeFi applications rely on peer-to-peer transactions without a centralized authority, the risk of fraudulent activities, such as hacking, scams and phishing attacks, increases. Despite the protection of smart contracts and blockchain consensus mechanisms, users still need to be alert to various potential risks [2]. In addition, due to the explosive growth of the cryptocurrency market, many mixed projects have emerged in the DeFi community, including some speculative projects. This can lead to unstable market conditions and even market manipulation. In addition, the lack of regulatory oversight and financial institution management may limit DeFi's long-term growth potential. These challenges need to be addressed for DeFi to mature and become a viable alternative to traditional finance.

2.3 Motivations

The decentralization of blockchain makes the circulation of assets more transparent and allows users around the world to cooperate without a centralized organization. Meanwhile, smart contracts can provide a collaborative environment that users can trust, making numerous decentralized financial projects possible [2][6]. In addition, the DeFi program allows for convenient cross-border projects where worldwide users can transact in the same environment without considering different legal environments [2][7]. Defi can be seen as a solution for crowdfunding platforms based on the above features. With the constraints of smart contracts, the managing cost of the project will be lower and more transparent, and the flow of funds raised can be better monitored, while users have more direct control over the project, and users from different regions can be involved in the project more freely. Meanwhile, the CFP makes the whole blockchain system more relevant to real world problems, making the whole DeFi project more purposeful and meaningful.
3. Methodology

The whole application (or platform) architecture has two layers, including blockchain layer and frontend layer. The blockchain layer manages the blockchain and the smart contracts of the platform, and any user can operate with the blockchain to enquiry data, update the transactions and publish smart contracts. The frontend will act as a web application for users to use. The frontend can be seen as a distributed platform because it does not need any server to store the data but reads data directly from the blockchain. In this way, compared to the traditional App architecture (Figure 1.1), the D-App is fully decentralized, and the frontend is hosted on peer-to-peer networks instead of centralized hosting servers (Figure 1.2). This ensures that the D-App is not controlled by any single entity and is resistant to censorship or shutdown.

3.1 Blockchain

Blockchain forms the base of the platform. The crowdfunding platform (CFP) can be based on any public Ethereum Vitrail Machine (EVM) supported blockchain. The blockchain needs to support token transfer, basic information query and smart contracts deployment. Smart contracts on the blockchain are written in Solidity, an Object-Oriented Programming (OOP) style smart contract programming language widely supported by different blockchain platforms.

The original intention of the application is to be able to run on any blockchain that supports the EVM smart contract, but the choice of blockchain is also very important in the current development and testing, as it determines the ecosystem that the application is currently attached to, it also affects the potential for future development of the application. As the pioneer of smart contracts, Ethereum is the platform of choice for many DeFi projects [8]. But at the same time there are also many alternatives worth considering. After analysis, Ethereum was not adopted as a blockchain for D-App.
development, and Evmos blockchain in Cosmos ecosystem was chosen as the blockchain platform for this project. The smart contracts are deployed on Evmos Testnet, which is a decentralized blockchain network that runs parallel to the Evmos Mainnet [9]. It allows developers to test and deploy their D-Apps in a safe and secure environment without the risk of losing real funds. A comparative analysis of blockchain development environments is listed below.

3.1.1 Alternative Analysis

Ethereum is a decentralized blockchain network that allows developers to build and deploy decentralized applications (D-Apps) using smart contracts. It is often regarded as the pioneer of smart contract platforms and has enabled a range of innovative D-Apps, such as decentralized exchanges and non-fungible token (NFT) marketplaces. However, Ethereum's limited scalability and high gas fees have become a significant challenge, especially as the demand for D-Apps grows. Ethereum Layer 2 solutions aim to address these challenges by offering off-chain scaling solutions that can increase transaction throughput and lower fees while still maintaining the security and decentralization of the Ethereum network. These solutions enable faster and cheaper transactions for users, making decentralized applications more accessible and appealing. Despite their potential benefits, Layer 2 solutions are not without limitations. They may introduce additional complexity to the development and deployment process of D-Apps, and their security and scalability still need to be further tested and improved. Additionally, interoperability between different Layer 2 solutions and other blockchain networks may be a challenge, limiting their impact on the wider blockchain ecosystem. Further introductions and analysis of Ethereum and Ethereum Layer 2 are listed below.

3.1.1.1 Ethereum

Ethereum was introduced in 2014 with a new proposition for building decentralized applications compared to the initial blockchain like Bitcoin. Unlike Bitcoin, which is primarily used as a digital currency, Ethereum is a single public blockchain where people would be able to deploy any kind of program. Ethereum achieved this by using the Ethereum Virtual Machine (EVM), which can be viewed as a single, canonical computer [10]. This virtual machine was able to process smart contracts which are self-executing contracts with the terms of the agreement directly written into code that any developer could deploy to the Ethereum blockchain in a permissionless way. This innovation has greatly influenced the blockchain community, inspiring the development of a vast ecosystem of D-Apps and giving rise to the concept of decentralization.

However, there are many limitations and drawbacks for constructing a platform directly based on Ethereum. First, decentralized applications built on top of Ethereum are inhibited by a shared rate of 15 transactions per second, and all applications are competing for the limited resources of a single
blockchain. Second, the EVM optimizes for the average use case, which means it has relatively low flexibility for developers. Third, each application is limited in sovereignty because they all share the same underlying environment. Forth, due to the high gas fee, low TPS and a growing number of transactions, this approach is costly and tedious [11]. Furthermore, as Ethereum just transitioned from Proof-of-Work (POW) to Proof-of-Stake (POS) in September 2022 (which is also called the Merge) [12], the entire structure and ecology of the blockchain are facing a huge change, and the applications may face some potential risks.

3.1.1.2 Ethereum Layer 2

Ethereum Layer 2 (L2) is a collective term to describe a specific set of Ethereum scaling solutions. A Layer 2 is a separate blockchain that extends Ethereum and inherits the security guarantees of Ethereum [13]. Layer2s blockchain utilizes Ethereum for consensus and data availability and submit transactions to communicate with Ethereum and ensure similar security. Moreover, by removing the transactions load from Layer1, Layer2 is more scalable. Typically, Ethereum Layer2s can have fewer gas fees and support more transactions per second (TPS). Currently, there are various Layer2s, such as Optimism, Arbitrum, Parastate, and Polygon.

However, potential drawbacks of Layer2 include reduced security and decentralization, which makes it difficult to maintain on-chain data consistency as most transaction data is moved off-chain. In addition, Layer2 solutions rely on contracts on the blockchain, which could lead to loss of user assets in the event of a contract breach or attack. Finally, Layer2s are not yet standardized enough, resulting in poor interoperability between different Layer2 platforms. Therefore, to better utilize the advantages of Ethereum Layer2, these potential flaws need to be continuously focused and addressed. Layer2s are also exposed to potential risks due to changes in the underlying environment following the Ethereum upgrade [12]. Combined with the drawbacks mentioned above, Ethereum Lay2 was not chosen as the blockchain development platform on which this project relies.

3.1.2 Cosmos

Cosmos is a decentralized network of independent parallel blockchains, each powered by Byzantine Fault Tolerance (BFT) consensus algorithms like Tendermint consensus. Cosmos SDK is an open-source framework for building application-specific blockchains, or app-chains [14]. Developers can customize blockchains from scratch that can natively interoperate with other blockchains (inter-blockchain communication). Also, developers can decide on the account model (account based or UTXO), the state machine programming language, as well as a variety of other parameters of the blockchain, which is more flexible than Ethereum [14][15]. As developers have a higher level of control over the blockchain during the developing, application-specific blockchain can be
customized to operate a single application, and developers have all the freedom to make the design decisions required for the application to run optimally. They can also provide better sovereignty, security, and performance.

Cosmos also supports intercommunication and collaboration between different blockchains with different application and architectural specifications, further facilitating the development and innovation of the blockchain ecosystem. The Inter-Blockchain Communication Protocol (IBC) is a critical component for achieving interoperability within the Cosmos network [14]. It utilizes the fast and secure consensus mechanism of Tendermint to enable the transfer of value, such as token transfers, and facilitate communication between blockchains with diverse designs and features. IBC enables the connection of independent chains, even if they have distinct validator sets and are developed for different use cases.

3.1.2.1 Evmos

Evmos is a blockchain network in the Cosmos Ecosystem that supports the Ethereum Virtual Machine (EVM) and is compatible with the Ethereum network. It allows developers to build decentralized applications (dApps) on Evmos using Solidity, the programming language used for Ethereum smart contracts [9]. Evmos offers a secure and scalable environment for developers to build and deploy dApps, while also benefiting from the interoperability and modularity of the Cosmos network.

Evmos development team believe that D-Apps in Ethereum ecosystem are accustomed to deploying and maintaining multiple instances of the same smart contracts across many chains. However, this also leads to higher costs associated with the upkeep and enhancement of each individual application, as well as increases the risk of economic fragmentation and bridging for end users [16]. The above issues can improve by deploying an application once and accessing it from any other chain. By adopting the IBC protocol of Cosmos, which is a trust-minimized "cross-chain" approach, applications can avoid being isolated. This approach allows developers to access every community, asset, and feature, and focus on building the best application possible for their respective use-cases.

Since the application specific chain in Cosmos does not support EVM, it is not possible to edit and deploy smart contracts. And without a deep understanding of the Cosmos SDK, it is difficult for developers to build a well-developed customized blockchain in Cosmos ecosystem [16]. Evmos enables EVM D-App developers to leverage IBC directly from the smart contract level.

The smart contracts of the application are deployed on Evmos Testnet (Table 2). The Evmos Testnet is a public blockchain network that runs parallel to the Mainnet, allowing developers to test and deploy their D-Apps safely and securely without spending real funds [9]. The Testnet is based on EVM and
supports the Ethereum toolchain, making it compatible with a vast ecosystem of existing Ethereum-based applications and tools.

<table>
<thead>
<tr>
<th>Evmos Testnet</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>host and port</td>
<td><a href="https://eth.bd.evmos.dev:8545">https://eth.bd.evmos.dev:8545</a></td>
</tr>
<tr>
<td>network_id</td>
<td>9000</td>
</tr>
<tr>
<td>chain_id</td>
<td>9000</td>
</tr>
</tbody>
</table>

Table 3 Evmos Testnet Configuration [9]

3.2 Smart Contract

The smart contracts are written in Solidity. Solidity is an object-oriented, high-level language for implementing smart contracts. It is a curly-bracket language designed to target the Ethereum Virtual Machine (EVM). It is influenced by C++, Python and JavaScript [17]. Solidity is statically typed, supports inheritance, libraries and complex user-defined types among other features. As the most popular language for Ethereum smart contracts, Solidity has a vast community of developers, extensive documentation, and a wide range of tools and resources to support its use.

The smart contracts were first developed and tested using Remix IDE. Then truffle is used to compile and deploy the contracts to the blockchain. The general introductions of Remix IDE and Truffle are listed below.

3.2.1 Remix IDE

Remix IDE is a web-based integrated development environment (IDE) designed for the development of smart contracts for Ethereum and other blockchain platforms. It provides a user-friendly interface for coding, debugging, testing, and deploying smart contracts on the blockchain [18]. Remix IDE provides a built-in compiler and a virtual machine to test smart contracts in a sandbox environment before deploying them on the blockchain.

3.2.2 Truffle

Truffle is a blockchain development framework aimed at simplifying the development, testing, and deployment of smart contracts. Truffle provides a set of tools and libraries for writing and testing smart contracts, including the Solidity compiler, contract management, automated testing frameworks, interactive debugger, and deployment tools [19]. Additionally, Truffle supports integration with Ganache, making local testing and development more accessible.

3.3 Frontend

A simple decentralized web application is proposed for demonstration. The web page is developed based on React and can interact with smart contracts on the blockchain via web3.js. MetaMask wallet extension is required for transactions and contributions. Detailed introductions are listed below.
3.3.1 React

The frontend of the web application is implemented using React. React is an open-source JavaScript library for building user interfaces. It provides reusable components and has extensive community support and plugins. On the web page, users can browse the existing Crowdfunding Campaigns and view the status and basic information of the different Campaigns, as well as the withdraw requests made by fundraisers.

3.3.2 Web3

Web3.js is a JavaScript library that provides a simple and standardized way to interact with the Ethereum blockchain. It abstracts away the complexity of direct interaction with the Ethereum blockchain, providing developers with a set of standardized interfaces and utilities for building D-Apps. Web3.js is designed to provide developers with a high-level, object-oriented API that simplifies the process of interacting with the Ethereum network. It includes functions for handling account management, transaction creation and signing, contract deployment, and event handling. These functions are built on top of the core Ethereum JSON-RPC API, which is used to communicate with the Ethereum node [20]. It is a key component in the Ethereum ecosystem and is widely used by developers building decentralized finance (DeFi) applications, non-fungible token (NFT) marketplaces, and other Ethereum-based projects.

3.3.3 MetaMask

MetaMask is a cryptocurrency wallet that preforms as a browser extension that enables users to interact with blockchains. It provides users with a digital wallet to securely store and manage their accounts, as well as a bridge to connect their web browser to the Ethereum blockchain. MetaMask abstracts away the complexities of interacting with the blockchain, allowing users to easily send and receive transactions, sign messages, and interact with smart contracts.
4. Application Design

In this Chapter the report talks about the D-App design and the objectives of the application, including the general introduction crowdfunding process and detailed features for the users. Then the application architecture is presented, with the details of smart contracts design and the frontend design.

4.1 General Introduction

This project aims to achieve a Blockchain-based DeFi Crowdfunding Platform, which allows organizations and individuals to raise tokens from users for a variety of purposes, including but not limited to charity, investment projects, and funding for entrepreneurs and startups. The sole goal of the platform is to support crowdfunding projects and maintain funding pools, which makes the whole ecology of the platform purer. The smart contracts will be templatized so that anyone can easily publish their campaigns. Additionally, the platform will encourage users to participate in the maintenance of the campaigns. Backers can be assigned voting power for each campaign, depending on how many tokens they have contributed and the contribution balance in the fund pools, which allows them to participate in each withdraw request posted by the fundraiser. Backers can track the progress of different projects, providing further support or limiting their economic activity. In a crowdfunding campaign, fundraisers can raise funds in the form of crypto tokens and get withdraw tokens by posting requests. If the project violates the contract or does not meet the backers’ expectations, the remaining funds in the pool can be returned.

4.1.1 Fundraiser Operation Instructions

1) Fundraisers can customize their own smart contract to publish the crowdfunding campaign by calling the campaign factory contract on the blockchain. The fundraisers have limited authority over the pool, they can only divert tokens from the pool if their withdraw requests are approved by the backers. Fundraisers are advised to make explanation for the objectives and details of the project to attract users contribution and provide reasons for each request to get backers approve.

2) For each campaign the fundraiser needs to clarify:

   a) Description: the basic information about the crowdfunding project, including the purpose and management process.

   b) Target: the maximum amount of tokens that needs to be raised for a crowdfunding project.

   c) Lifespan: the duration of the project. When the campaign ends, the fundraiser will no longer be able to operate with the Campaign. The remaining tokens in the pool will be returned to backers.

3) For each request the fundraiser needs to clarify:

   a) Description: the basic information about the withdraw request, including
the purpose and management process.

b) Amount: the withdraw value.

4.1.2 Backers Operation Instructions

1) Users can browse all existing crowdfunding projects. Users can freely make token transactions with each other.

2) Users can invest in a Campaign by sending tokens to a specific fund pool and become the backers of that project.

3) For each withdraw request posted by the fundraiser, the backers need to vote for the request using their voting power.

4) When the campaign ends, backers can send a refund request. Remaining tokens in the fund pool are returned according to the backers' contribution balance.

4.1.3 Voting Power

A backer’s voting power is determined by the contribution balance in the fund pool. In general, voting power is the percentage of tokens contributed by the backer in the token pool. After each successful request, the backer's contribution balance decreases as well. When a new contribution appears, the backer's original voting power may be diluted. Under these conditions, users are encouraged to allocate their tokens wisely across campaigns and to donate wisely during a campaign progresses.

Say the current block timestamp is \( t_0 \). The backer Alice has contributed \( x \) tokens into a Campaign that has not yet approved any requests, and the current Campaign balance is \( y \). Now we can have:

\[
\begin{align*}
\text{Balance (Alice, } t_0) &= x \\
\text{Balance (Campaign, } t_0) &= y \\
\text{Voting Power (Alice, } t_0) &= x/y
\end{align*}
\]

If a request just ended at \( t_1 \), and the fundraiser can successfully withdraw \( z \) tokens form the Campaign. After the request, the balance of Alice’s contribution and the fund pool now become:

\[
\begin{align*}
\text{Balance (Alice, } t_1) &= x - z \cdot (x / y) \\
\text{Balance (Campaign, } t_1) &= y - z
\end{align*}
\]

Although Alice's voting power does not change, subsequent donations will continue to increase the Campaign's balance, which will dilute Alice's voting power. Of course, Alice can also keep contributing to boost her voting power.

4.2 Application Architecture

The D-App consists of two layers: the blockchain layer and the frontend layer (Figure 2). The blockchain layer runs on the Evmos Testnet, where the Campaign Factory contract has been deployed. Users can customize their own contracts by directly calling the functions within the Campaign Factory and deploying the Campaign contract onto
the blockchain. The frontend layer is built using React and communicates with the blockchain through Web3.js. The blockchain layer provides the backbone of the application, where all the data and transactions are securely stored on the blockchain. By running on the Evmos Testnet, contracts can be tested in a sandbox environment. The Campaign Factory contract provides a customizable template and easy deployment for fundraisers. The frontend layer provides a user-friendly interface for users to interact with the blockchain layer. By using React, the frontend is modular and easily scalable, allowing for future expansion. Web3.js acts as the middleware between the frontend and blockchain layer, abstracting away the complexities of directly interacting with the blockchain and providing a standardized interface for communication. At the same time, users can also use Truffle to operate directly on the blockchain layer.

4.3 Campaign Design

The workflow of the crowdfunding campaign is shown below (Figure 3). Campaign pool refers to the smart contract address of the customized Campaign contract. For withdrawal requests, the fundraiser can only create a new request if the previous request ends. According to the design, the remaining funds will be returned after the Campaign ends, however, Smart contracts are not automatically triggered on the blockchain, but require the user to manipulate the functions within the contract. So, the backers need to call the refund function in Campaign contracts. Similarly, requests are not automatically updated at the end of the voting period, including status updates and automatic transfers to fundraisers. Fundraisers need to call the finalize request functions in the Campaign Contact.
Campaigns need to have a lifespan of between 7 and 300 days, and once fundraisers determine the length of a project when creating a campaign, they can't change it, as this can influence backers' decisions by affecting the time it takes for backers to receive the refund. Request's voting period was fixed at 7 days. When the voting period ends, the smart contract determines the status of the Request based on the Voting Power received by the Request. Only when Request has received more than 50% of the Voting Power, the fundraiser can receive the withdrawal tokens.

4.4 Frontend Design

The frontend of D-App performs as a web page. The UI design is show below. Users can browse all Campaign projects created by CampaignFactory Contract on the blockchain and view the basic details from the Home Page (Figure 4). By clicking the Refresh button, the web application will query all the data using Web3.js from the blockchain and the smart contracts. Users can directly make contributions to Campaigns.
By clicking the Create Campaign button, users can easily customize their own crowdfunding project easily. Fundraisers are required to specify the description of the Campaign and set the target and lifespan of the Campaign (Figure 5).

![Create Campaign UI](image1.png)

**Figure 5 Create Campaign UI**

When users click the Campaign on the home page (Figure 6), they can be directed to the Campaign detail page. By clicking the refresh button, the Campaign detail page will get detailed information about the Campaign using the contract address from the blockchain. Users can easily find all detailed information about the Campaign in the box, including the contract address, raiser address, current balance, and project target, the campaign description and board, the lifespan information, and the Backers' information (with their contribution balance of the fund pool).

Also, users can view all the Requests Details, including the request description, request amount, request lifespan, and the approvement status. For active requests, backers can click the approve request button to vote. After the voting period, the fundraiser can click the finalize request button to get withdraw.

![Campaign Detail Page UI](image2.png)

**Figure 6 Campaign Detail Page UI**
Fundraisers can click the update info button on the Campaign detail page to post some updating message to the blockchain (Figure 7). Also, fundraisers are encouraged to post their Campaign online documentation or URL on the Campaign board to save the gas fee from daily updating.

![Figure 7 Update Info UI](image)

**Figure 7 Update Info UI**

Fundraisers can click the create request button to publish new request. Fundraisers are required to specify the description of the request and set the withdraw value (Figure 8).

![Create Request UI](image)

**Figure 8 Create Request UI**

![Sign Transaction](image)  
**Figure 9.1 Sign Transaction**

![View Account Detail](image)  
**Figure 9.2 View Account Detail**
When users want to make any transactions that need to modify the smart contract or call payable functions, such as creating campaigns, creating requests, approving requests, finalizing requests, gas fee will be charged. Users can rely on third-party wallet extensions like MetaMask to sign the transactions (Figure 9.1, Figure 9.2).

![Figure 10 Block Explorer](image)

When users want to trace the transactions of different accounts, they can use the third-party block explorer (Figure 10). Block explorer is a web-based tool that allows users to explore and analyze the contents of a blockchain, which can provide a visual representation of the blockchain's transaction history and allows users to view details about each account. The Evmos Testnet block explorer is Mintscan [21].
5. Conclusion

After discussing the limitations of traditional crowdfunding platforms [1-5] and analyzing the potential of decentralized finance [6-8], the report introduces the advantages of building a blockchain-based DeFi crowding platform, which can make the crowdfunding business model interoperable, borderless, and transparent. The report also introduces the methodologies of the application development and analyzes its alternatives and discusses the platform architecture. A decentralized application is proposed to build this platform, which consists of two layers: the blockchain layer, and frontend. The blockchain layer is based on Evmos [9], a blockchain that belongs to the Cosmos Ecosystem [14], which manages the blockchain and the smart contracts of the platform, and any user can operate with the blockchain to query data, update the transactions and publish smart contracts. Meanwhile, the frontend layer takes the form of a React web page powered by Web3.js. It interacts with the blockchain layer and presents users with campaign information.

The project explores the potential of DeFi in the crowdfunding area and demonstrates the use of blockchain and smart contracts to create a more transparent and efficient fundraising platform. The use of Cosmos and Evmos allows for greater modularity and interoperability, enabling communication and interaction between different blockchains. The project's limitations include the current reliance on a single blockchain and the need for further testing and development.

However, there are still potential flaws in the mechanics of Campaigns and how users vote for requests. While dynamic Voting Power has been a better representation of backers’ influence on Campaigns than traditional voting methods, smart contracts are still not immune to all kinds of anonymous malicious attacks. Fundraisers can dilute the Voting Power of other backers by creating multiple different accounts and donating to their own Campaigns to gain overall control. While users can avoid this situation as much as possible by monitoring the accounts of different backers, there is still a potential risk.

Future potential developments include leveraging IBC to enable collaboration and communication between the project and other Cosmos-based blockchains. Overall, the project provides a valuable contribution to the exploration of decentralized crowdfunding platforms and their potential to disrupt traditional fundraising models.
References

[Accessed: 01- Dec- 2022].