FidoMeta

World's First Virtual Financial Institution into Metaverse

Project Documentation

TABLE OF CONTENTS

INTRODUCTION

The Metaverse is a shared virtual environment that leverages cutting-edge technologies like VR, AR, and blockchain to create engaging experiences for its denizens. It's a virtual world, accessed via the Internet, where people can interact with one another and with the environment in a variety of ways, often through their own virtual avatars. Playing games, attending events like virtual concerts, creating and trading digital land, property and other digital assets –are just some of the activities that are available to denizens of the Metaverse.

In the Metaverse, we have a technology that can grow organically, as developers gradually build more and more tools and products around it. At the same time, much of its growth will likely be driven by corporate interest, as companies – especially the big ones – are always on the lookout for promising new markets. In this respect, the Metaverse's journey will likely resemble that of the Internet, which also rose from humble beginnings to become the connective tissue of our modern world.

So if we take the development of the Internet as a benchmark, we'll see that even the most transformative technologies need a period of maturation to truly unlock their potential. Twenty years ago my every visit to the online world was heralded by the screeching sound of my modem. Today, I can have an Internet connection practically everywhere.

FIDO Metaverse focuses on creating businesses in the world of Virtual Reality. All products and projects have been designed and developed to serve the real-world community through the advancement of technology and digital presence.

Fido Central bank is an apex body that controls, operates, regulates, and directs the entire banking and monetary structure in the Fido Metaverse. Fido Central Bank makes regulations for the operation of other crypto projects that want to run their banks in the FIDO Metaverse. All the other banks on the land of FIDO Metaverse have to work under the instruction and regulations developed by the FIDO Central Bank.

1. PURPOSE OF FIDOMETA

The idea of banking institutions in the metaverse space is rare and limited. FIDOMETA is one of the world's first crypto banks on metaverse. Unlike traditional banking businesses, FIDOMETA takes a further step in the banking field and aims towards establishing a unique crypto addition.

In simple words, FIDOMETA is the world's first crypto project that aims to provide banking services in the Metaverse space. Unlike traditional Banking, a FIDOMETA user will be able to experience Banking in the virtual space with better and more advantages like being able to perform banking-related tasks from the comfort of their own home.

FIDOMETA is a decentralized financial institution that removes the middlemen from the position of central authority. Decentralized finance helps financial institutions use distributed ledger technology to improve cash flow, market share, and credibility by offering financial instruments and services without relying on other banks, brokerages, and exchanges as intermediaries.

Every physical institution has certain limitations like opening and closing timing, governmental interference, internal interference, etc. Environmental disasters affect some institutions like banks, stores, offices, etc. the most due to their being offline and physical structure. Every institution had to take the help of technology and go online due to the recent global pandemic. But sometimes, visiting such physical institutions becomes necessary to perform certain activities like visiting banks for some crucial work. This process is not only time taking and tiresome, but also may become hazardous. There is a high possibility of getting infected by invisible viruses like Covid-19.

To avoid this dangerous situation, FIDOMETA has been developing a metaverse where users can perform various activities like meeting with banking personnel to perform their important work without wasting their precious time and being safe at the same time. Hence all these activities would be happening virtually, and the chances of getting infected or wasting time become zero.

2. PRODUCT AND FEATURES

Most of us would be knowing that the support functions required for any traditional business are unlimited. Starting from banking services, payment solutions, hiring talents, training, and many more.

1. FIDOMETA Governance Token (FMC):

FIDOMETA token will act as the core utility token across a portfolio of FIDOMETA products, including staking, reward sharing through providing liquidity to or securing the protocol, as well as governance procedures. For FIDOMETA token Lending particularly, there will be a liquidity mining program rewarding users with FIDOMETA TOKEN for bootstrapping liquidity on the protocol by depositing and borrowing assets.
As the FIDOMETA token continues to innovate in the Defi space and build more FIDOMETA products, FIDOMETA token holders will be able to propose and decide upon key protocol parameters for all FIDOMETA products and how the products interoperate. FIDOMETA TOKEN tokens will continue to be the primary tool to align incentives of community builders and supporters, creating a strong community of Defi enthusiasts who will collectively help propel the FIDOMETA token ecosystem forward.

2. FIDOMETA Virtual Central Bank (FVCB):

Fido Central bank is an apex body that controls, operates, regulates, and directs the entire banking and monetary structure in the Fido Metaverse. Fido Central Bank makes regulations for the operation of other crypto projects that want to run their banks in the FIDO Metaverse. All the other banks on the land of FIDO Metaverse have to work under the instruction and regulations developed by the FIDO Central Bank. Fido Virtual Central Bank controls the entire banking system in the FIDO Metaverse. If any crypto project wants to use its land for creating a commercial banking institution, it will have to take permission from the FIDO Central Bank.
FCB doesn't interfere with the operations of any bank working on the land of FIDO Metaverse, it provides them with the right direction to handle their operations in a certain and safe kind of way. It facilitates other crypto projects and assists them in performing banking operations fluently.

- **Functions of FIDOMETA Central Bank:**
  1. **Bankers' bank and supervisor –**

     Fido Central Bank acts as the supreme authority that assists with new crypto projects on its metaverse, facilitates them, makes rules & regulations, and keep control over financial activities being performed on the lands of FIDO Metaverse.

  2. **Banker to the Fido Banking Ecosystem –**

     FIDO ecosystem consists of a network of interlinked crypto projects that dynamically interact with each other through competition and cooperation to grow sales and survive. This FIDO Ecosystem includes suppliers, distributors, customers, competitors, and other participants.

  3. **FIDO Token Reserve –**

     On every deduction of 6% as a transaction fee, 5% transaction will be redistributed to all the token holders, and the rest 1% will be transferred to the FIDO Central Bank. In this way, Fido Central bank will always have liquidity as a reserve.

4. **Lender of Last resort –**

   When commercial banks on the FIDO Metaverse fail to meet their financial requirements from all the other resources, they can approach FIDO Central Bank for crypto loans. They are open to pitch their future plan and business model to the FIDO Central Bank. FCB has the power to either accept their request or reject to provide them funds.

3. FIDOMETA Wallet

The FIDOMETA will have a wallet application of its own named Fido Wallet App. With the help of the Fido Wallet app users will be able to make purchases and transactions in the Metaverse and will also be able to trade the FIDOMETA coin with the help of the Wallet. Users and token holders buy and sell products of the Fido Metaverse in a convenient and hassle-free manner. Another added advantage is that the users of the wallet can make transactions with near-zero network transaction fees. The wallet will also be used in the Fido Market where users can easily gain discounted prices for their market purchases. The Fido wallet will be expanded to provide a payment gateway for all businesses running in the Fido metaverse

4. FIDO Markets

FIDOMETA aims to create a one-stop eCommerce shopping experience that will help users shop at pocket-friendly prices. In the FIDOMETA Market, FIDOMETA token holders will be allowed to shop with associated brands through Fido coins. Every time a user token holder makes a purchase through the FIDOMETA token, they will be granted a discount as a result of the usage of the coin.

5. FIDO Virtual University

In the Fido Metaverse, a virtual university will also be set up where users will be allowed to take up academic and professional certification courses. In the case of academic courses like particular degrees, students will also be allowed to directly attempt exams without having to spend hours on classes or tutoring sessions. The Fido Virtual University will have courses and MOOCs in various fields like STEM science, liberal arts, Business and finance, etc.

6. FIDO Advisory Platform

In the Fido, Metaverse users will also be able to consult various professionals on matters like financial modeling, human resources, etc. Any business or individual in need of consultancy services can hire professionals from the Fido advisory platform on a contractual or freelance basis.

3. TEAM

1. MR. Balamohan Krishnan

Mr. Balamohan Krishnan is Ex-Vice President at Citi Bank and he is a business coach, Trader, Trading trainer, Trading software architect, Fund Manager, Inspiring Speaker and Mentor. Possess 15+ years' experience in the forex industry in top Investment banks, alongside the top trading floors in New York and London.

2. DR. K. Mohideen

Dr. K. Mohideen is the founder and managing director of Zontia Group. He started his early career serving the RCU division of a reputed bank. He is a first-generation entrepreneur, in his family, and his diligent attitude brought about many successful events gradually.

3. Mr. Inayat Hussain

Mr. Inayat Hussain is Co-Founder at TheCryptoLaunchpad, Ex-Developer at IBM and Swiggy, Smart Entrepreneur deeply enthusiastic about merging different approaches and techniques into Blockchain with metaverse, NFT's and DeFi

4. SMART CONTRACT

LIST OF AVAILABLE FUNCTIONS IN THE ENTIRE SMART CONTRACT

1. Interface IBEP20
2. Contract Context
3. Library SafeMath
4. Contract Ownable extends Context
5. Contract BEP865 Token extends Context, IBEP20, Ownable

1. INTERFACE IBEP20

Interface IBEP20 is a standard ERC20 interface defined in the EIP where it includes functions like

1. **totalsupply ()** - Returns the total number of tokens in existence.
2. **balanceOf (account)** – Returns the number of tokens owned by
3. **transfer (recipient, amount)** – moves the number of tokens from the callers account to the recipient and return Boolean and emit a Transfer event.
4. **allowance(owner, spender)** – Returns the remaining number of tokens that spender will be allowed to spend on behalf of the owner through
5. **. approve(spender, amount) –** sets amount as the allowance of spender over the caller's tokens and returns Boolean and emits Approval event.
6. **transferFrom(sender, recipient, amount) –** moves amount tokens from sender to the recipient using the allowance mechanism. amount is deducted from the caller's allowance and returns Boolean and emits a Transfer event.
7. **Transfer(from, to, value) –** Emitted when the amount of tokens are moved from one to another account
8. **Approval (owner, spender, value) –** Emitted when the allowance of a spender for an owner is set by a call to 'approve'. 'value' is the new allowance.

2. Context

The **Context** contract provides the information about the current execution context where the sender of the transaction and its data. While these are generally available via msg.sender and msg.data, they should not be accessed in such a direct manner, since when dealing with GSN meta transactions the account sending and paying for execution may not be the actual sender (as far as an application * is concerned).
This contract is only required for intermediate, library-like contracts.
1. **_msgSender() –** returns the msg.sender

2. **_msgData() –** returns the msg.data

3. SafeMath

The **SafeMath** library provides solidity's arithmetic operations where Arithmetic operations in Solidity wrap on overflow. This can easily result in bugs because programmers usually assume that an overflow raises an error, which is the standard behavior in high-level programming languages. `SafeMath` restores this intuition by reverting the transaction when an operation overflows. Using this library instead of the unchecked operations eliminates an entire class of bugs, so it's recommended to use it always.
We have arithmetic functions like
1. **add() -** addition
2. **sub() -** subtraction
3. **mul() –** multiplication
4. **div() –** division

5. **mod() –** modulus

4. Ownable extends Context

The Ownable contract extends context where the Contract module which provides a basic access control mechanism, where there is an account (an owner) that can be granted exclusive access to specific functions. By default, the owner account will be the one that deploys the contract. This can later be changed with {transferOwnership} This module is used through inheritance. It will make available the modifier `onlyOwner`, which can be applied to your functions to restrict their use to the owner.
List of methods
1. constructor() - Initializes the contract setting the deployer as the initial owner.

2. owner() – returns the address of the current owner.

3. onlyOwner() – throws an error if called by any account other than the owner

4. renounceOwnership() – leaves the contract without owner and not possible to run onlyowner functions anymore. This function can only be run by the owner itself.

5. transferOwnership() – transfer ownership of this smart contract to a new account where this function can only be run by the owner itself.

ALL THESE FUNCTIONS DESCRIBE THE OWNER'S AUTHORITY

5. Fidometa extends Context, IBEP20, Ownable

Name: Fido Meta

Symbol: FMC

Decimal: 9

Supply: 15,000,000,000

Tax Fee: There are 4 types of tax deduction per transaction: (6%)

1. **_community_charge :** by default, it is **3%,** It can be configured later by the owner. As solidity does not support floating-point number So this fee would be **3 * 10 * decimals**

2. **_ecoSysFee:** by default, ecosystem fee is **1.5 percent (1.5 * 10 *decimals)**

3. **_surcharge1:** This is an optional fee placeholder that can be used to deduct any kind of service charge for fido virtual central bank. By default, it is **0.5 percent** and goes to burn wallet. Admin can set a wallet for surcharge1.

4. **_surcharge2:** This is an optional fee placeholder that can be used to deduct any kind of service charge for fido virtual central bank. By default, it is **0 percent.** Admin can set a wallet for surcharge2.

5. **_surcharge3:** This is an optional fee placeholder that can be used to deduct any kind of service charge for fido virtual central bank. By default, it is **0 percent.** Admin can set a wallet for surcharge3.

Token Vesting Period:

FMC Owner will use **transferWithLock(recipient, tAmount, initialLock)** function to put an investor vesting period with cliff time of initialLock in days. Afer cliff timing we allows only 20% of the token to be released every month by the investor. Investor can unlock their token using **unlock(acccount)** function after serving cliff time and check the status using **locks** state variable.

State Variables:

1. **_rOwned** - User's balance represented in r-space

2. **_tOwned** - User's balance represented in t-space (only used by non-stakers)

3. **_allowance** – Current allowance of an account

4. **_isExcludedFromCommunity_charge** – mapping of an address to bool to display if an

5. account is excluded from deduction of the community charges

6. **_isExcludedFromReward** - mapping of an address to bool to display if an account is excluded from getting rewards in community distribution.

7. **_excluded** – Array that maintains the excluded account from getting a reward

8. **_isExcludedFromEcoSysFee** – mapping that keeps track of the address that are excluded from ecosystem charge.

9. **_isExcludedFromSurcharge1** - mapping that keeps track of the address that are excluded from surcharge1

10. **_isExcludedFromSurcharge2** - mapping that keeps track of the address that are excluded from surcharge2

11. **_isExcludedFromSurcharge3** - mapping that keeps track of the address that are excluded from surcharge3

12. **_tTotal** – its total supply

13. **_rtotal** – its total reward has been distributed so far

14. **tCommunityChargeTotal** – its total reflections so far

15. **ecoSysWallet, _surcharge_1_Wallet, _surcharge_2_Wallet;, _surcharge_3_**Wallet – corresponding wallets for ecosystem,surcharge1,surcharge2, surcharge3.

16. **_community_charge, _ecoSysFee, _surcharge1, _surcharge2, _surcharge3**: these are the fees that comes under total tax per transaction

17. **_previousCommunityCharge, _previousEcoSysFee, _previousSurcharge1, _previousSurcharge2, _previousSurcharge3** – these are the variable that store the old charges while admin try to change it.

18. **_ maxTxAmount** – it fix the max token a user can transfer at a time.

19. **Burn** and **Mint** event – it is fired when burn or mint action is fired.

20. **locks** – it keeps track of the vesting account

21. **LockDetails** – struct with all required variables

22. **TValues** and **MValues** – structs that is used to calculate token distribution

Functions:

1. **constructor()** - Initializes the contract setting the deployer as the initial owner
2. **totalSupply()** - gives total Supply
3. **balanceOf(account)** - gives a balance of an account
4. **transfer(recipient, amount)** - transfer Fund
5. **allowance(ownerAccount,spender):** Returns the remaining number of tokens that `spender` will be allowed to spend on behalf of `owner` through {transferFrom}. This is zero by default.
6. **Approve(address,amount)** - Sets `amount` as the allowance of `spender` over the caller's tokens.
7. **increaseAllowance(spender, addedvalue)** - increases allowance
8. **decreaseAllowance(spender,Subtractedvalue)-** decreases allowance
9. **_approve(owner,spender,amount)** - sets amount as the allowance of spender over the caller’s tokens and returns Boolean and emits Approval event.
10. **Burn(value)** – burn a specific amount of token by the owner only
11. **Mint (value)** – mint specific amount of token by owner only not greater than the supply of token
12. **setCharges(community_charge,ecoSysFee,surcharge1,surcharge2,surcharge3** – set percentage of all the charges in percentage * 10 ** decimals form
13. **setServiceWallet(ecoSysWallet, surcharge_1_wallet, surcharge_2_wallet, surcharge_3_wallet):**
    update the service wallet and exclude them from all charges and reward
14. **tokenFromReflection(rAmount)** – calculates the token for given rAmount from reflection
15. **excludeFromCharges(address, communityCharge, surcharge1, surcharge2, surcharge3)** – excluded the wallet from all the charges. True bool values show the wallet is excluded while false means the wallet is included in the reward
16. **excludeFromReward(account)** – owner can exclude an account from getting rewards. once the wallet is excluded cannot be included back

17. **setMaxTxPercent(maxTxPercent)** - it set the maximum amount of token an address can tranfer at once in the form of maxTxPercent* 10 ** decimals

18. **_reflectFee(rFee, tFee)** – used to reflect fee to the holders

19. **_getValues, _getTValues, _getRValues** - these are the native functions that calculates the tax and community distributions

20. **_getCurrentSupply()**- it shows the current supply of the token

21. **takeCharges(tEcoSys, tSurcharge1, tSurcharge2, tSurcharge3)** – this function is responsible for taking corresponding charge from the transaction if percentage is greater than 0.

22. **transferFrom(sender, recipient, amount)** - Transfers the tokens from a sender account to the receiver account,

23. **_transfer(from,to account)** – It transfers the tokens from an sender account to the receiver account,

24. **_tokenTransfer(sender, recipient, amount, takeCommunityCharge, takeEcosysFee, takeSurcharge1, takeSurcharge2, takeSurcharge3)** - it is the main function to transfer that transfer fund after taking decision how much tax has to be deducted

25. **transferWithLock(recipient, tAmount, initialLock)** - Transfer with a lock to put an account into vesting period. Initial lock is the cliff in days

26. **unlock(target_) –** unlock token during vesting period by an owner or user

27. **_transferFromExcluded, _transferToExcluded, _transferStandard, _transferBothExcluded** – these are the functions which are called by _tokentransfer after taking a decision how the transaction is happening.

28. **_doTransfer(mvalues, sender, recipient)** - final transfer of token after fee deduction and token reflection to the community.

6. Setup Instructions

After deployment we need to perform following setup.

1. Exclude the owner wallet from reward using excludeFromReward(account)

2. Set **0x000000000000000000000000000000000000dead** to **surcharge wallet-1** and set **ecosystem** wallet using setServiceWallet(ecoSysWallet, surcharge_1_wallet, surcharge_2_wallet, surcharge_3_wallet)
   Use **0x0000000000000000000000000000000000000000** for surcharge-2 and surcharge-3 wallet.
