EtherAuthority

Enterprise Grade Public Blockchain Technical Specification Document

[PRIVATE AND CONFIDENTIAL]

Prepared by: EtherAuthority Team
Prepared on: December 26th, 2022

Background

EtherAuthority is a leading blockchain company specialising in blockchain security services. More details can be found on the official website. The portfolio of security audit projects can be found on the official github.

This project is to set up an EVM based blockchain smart contract platform. It uses the DPoS consensus mechanism which makes it fast, high throughput and secure. This project includes a **Testnet** and **Mainnet** blockchain platform.

This document outlines the information which will help to run the blockchain platform. Also, **this document contains important and very confidential credentials**. So, please keep this document very private.

Infrastructure Components

(1) Blockchain Node: is the core component of the entire platform. This is a DPoS consensus mechanism. These will include mainnet as well as testnet node.

- We forked the code from Hecochain. This code can be found here.
- Node coinbase wallet will be set during the genesis deployment.

(2) Wallet: Users can hold, send, receive their coins and tokens. Users also will be able to interact with smart contracts and dapps.

- Metamask can be used.
- Other web3 and EVM compatible wallets which allow adding custom networks can be used too.
- Wallet credentials (will be provided):
    - Host URL:
    - Network / Chain ID:
    - Block explorer URL:

(3) Block explorer: Users can visualise the blockchain data, transactions, balances, etc. Users also can interact with smart contracts as well.

- The URL of the block explorer will be provided after infrastructure setup.
- It is built on an open source project, blockscout.
- Both Mainnet and Testnet explorers are included.

(4) Faucet: Users can request testnet coins so they can build dapps and smart contracts.

- The url of the faucet will be provided after infrastructure setup.
- Users can request 10 coins every day. This is to prevent abuse of the faucet and prevents bad actors from draining the entire faucet balance.
- The admin wallet from which testnet coins will be sent is:
    - 0xE916A6….923c9588 (this will be your wallet)
    - You need to keep funding this wallet. Do not put all the coins into this. Just keep small amounts like a million coins until it runs out. This is to prevent any rare event of server-hacking and being drained of entire coins.
- The peggy token smart contracts (These are demo ones. Real ones will be provided after infrastructure setup):
    - ETH = 0x5517A7c4a005d5cCCd8136C434537D369d6069F2
    - BNB = 0x37Ac44f381Ff3bC73947DEdCaC8F7A502beF74D3
    - USDT = 0x5d6B168E9e15262FcB1c433A5E11E89B768551e4

(4) Cross chain bridge: will allow users to get your blockchain assets with other blockchain assets. For example, users can swap Ethereum ETH to your blockchain pegged ETH. The pairs will be the one found in our demo system:

https://bridge.dithereum.org/

If you want to add other pairs, you can add. You need to do further development with additional cost.

(5) AMM: this will be called Automated Market Maker (DEX). This lets users swap the assets from one another, considering there is enough liquidity.

We will fork pancakeswap for this blockchain.

(6) **Developer documentation:** will be rebranded.

Payment milestones and Timeline

=> $3500 to get started to confirm the order.

=> $3500 when testnet is done

=> $3000 when the project is completed.

So, the entire project can be completed in a month's time with a total budget of $10k USD.

Ongoing costs

You will have ongoing costs of servers, domains, SSLs to run the blockchain platform. Initially it would be $300+ per month. But it will increase as your number of users increases and you have to increase the server configurations.

Server requirements

- 9 VPS servers having minimum 8 gb ram, and 4 CPU, and 160+ GB hard drive

Components which are not included in this project

- Custom mobile wallet and browser wallet extension is not included in this project. They can be developed separately if needed. This project will have Metamask and any other EVM compatible wallets, which can allow adding a custom network.
- Setting up a MyEtherWallet fork for this blockchain. This can be developed for an extra charge if really needed.
- Branding material such as logo, videos, banners, graphics, whitepaper, videos, etc. are not included in this project. You need to develop yourself.

Demo:

Please see the demo of the the platform.

=> Main website: https://dithereum.org (Not included in current project. You need to design it yourself based on your business need and branding)

=> Block Explorer: https://testnet.dthscan.io (by blockscout, which is open source project)

=> Developer doc: https://docs.dithereum.org/#/ (You can fork this. and if you want to redesign it, then you have to do it yourself as per your business branding.)

=> Faucet to get testnet coins: https://faucet.dithereum.org/

=> Cross chain bridge: https://bridge.dithereum.org/

=> DEX: https://dex.darthion.io/#/swap

=> Wallet to add in metamask: https://docs.dithereum.org/#/wallet

=> explanation video on how everything will work:

https://www.youtube.com/watch?v=Gzjt9cKuCso

FAQ:

(1) How to add or remove validators?

Validator can be added through the `validator` smart contract, which is the following: 000000000000000000000000000000000000f000
To interact with this smart contract, you need to use the remix interface to do that.

(2) How can a user become a validator?

To become a validator, the user needs to submit a proposal first and wait for other active validators to vote on it. After more than half of them pass, the user will be eligible to become a validator. Any address can stake to an address that qualifies to become a validator, and after the validator's staking volume ranks in the top 21, it will become an active validator in the next epoch.

(3) How many active validators can there be?

The current set of validators is responsible for packing out blocks, with a maximum of 21 active validators.

(4) How can any fraudulent validator be punished?

Whenever a validator is found not to pack a block as predefined, the `Punish` contract is automatically called at the end of this block and the validator is counted. When the count reaches 24, all income of the validator is punished. When the count reaches 48, the validator is removed from the list of active validators, and the validator is disqualified.

(5) How do users stake their coins to validators?

For any account, any number of coins can be staked to the validator, and the minimum staking amount for each validator is 32 Coins (This amount can be set).

To unstake, following things should be done:

- Send an unstaking transaction for a validator to the Validators contract.
- Wait for 86400 blocks before sending a transaction to Validators contract to withdraw all staking coins on this validator.