2020

Decimal Yellow Paper

version 1.0

Content

Disclaimer

This document is not a public offer, does not contain any legal requirements and cannot serve as a sufficient basis for making any decisions.

This document is not an official document.

This document has been compiled for informational purposes only. The sole purpose of this document is to present potential users of the ecosystem service and software Decimal and current DELs in relation to their sale. Decimal numbers and current DEL values should be consulted with your legal advisor before any actions in relation to the appearance published in the said document were to be determined.

The reports, tables, estimates and financial data provided in this document are forward-looking in nature and involve risks of uncertainty in the economic and legal context. In this regard, this kind of information is provided in this document for illustrative purposes only and is not a guarantee of achieving the specified values (indicators) in the future.

DECIMAL PTE. LTD. reserves the right to make changes to this document unilaterally without any special notice. The modified terms will be effective immediately upon posting.

The document does not constitute an offer to buy securities in any jurisdiction, an investment attraction or investment advice.

The document was prepared in Russian and English. In case of contradictions in the edition of the document, the edition of the document in English takes precedence.

DECIMAL PTE. LTD. is not liable to the user for any type of losses incurred by him, regardless of the reasons that entailed the losses.

The legal status of cryptocurrencies, digital assets and blockchain technologies is uncertain. Ten and lead to a ban on the distribution of current and the operation of our services, as well as negative consequences.

1. Introduction

This document contains a technical description of Decimal blockchain, which is being developed by our development team. When designing the document, we aimed to give a general idea of the project, as well as provide a more detailed description of the key elements of Decimal.

The document is written in technical language, but without excessive immersion in the nuances of technology and technical solutions. If you are looking for more general information about the project, please read Decimal White Paper (https://decimalchain.sg).

2. Decimal services overall structure

The figure below shows the general architecture of Decimal. The system-forming element is directly blockchain, a database structured as a chain of blocks.

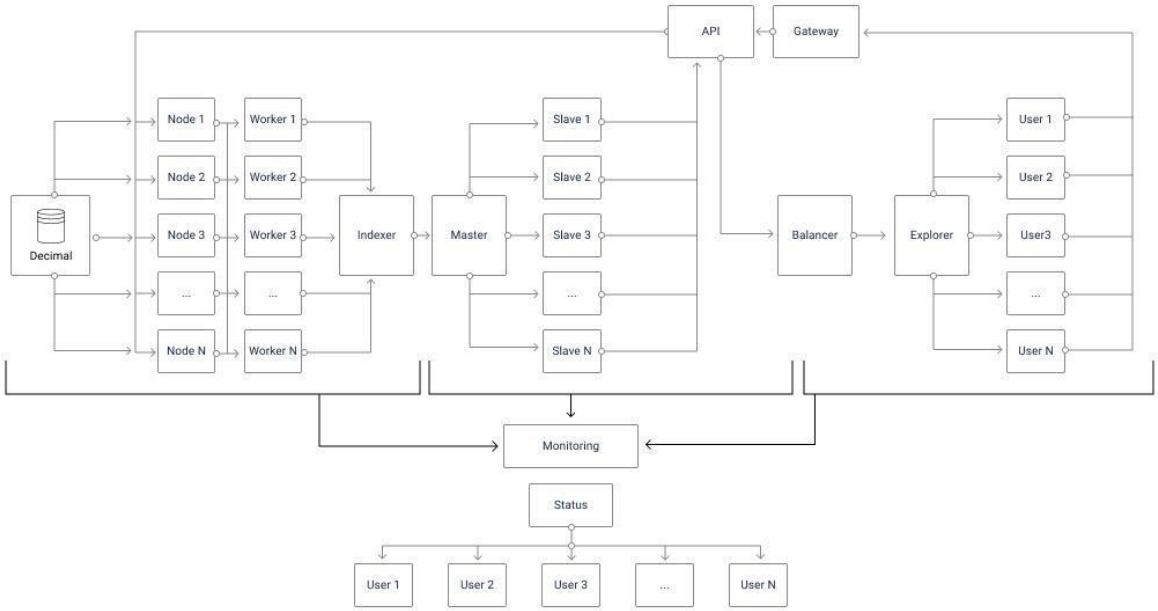


Fig. 1 - Decimal overall architecture.

Within the system blockchain physically exists in the form of database replicas, which are stored on each of the full nodes (master, validator).

In order to route the high number of read requests to the blockchain, and actually to the blockchain replicas, we have organized the following structure.

There are a number of services in the network - Workers, which collect data coming directly to the blockchain. Data are blocks and transactions of different types: sending, buying, selling, creating coins, etc. Several identical services are required in case some of them fail, no bit of information should be lost in this part of the process, however, as in any other part of the process.

A sampling of data at the output of workshops goes to the indexer, which structures, sorts and indexes the incoming information, after which all data is stored in the main (leading) database (Master). The data are then copied multiple times and placed in slave storage (Slave).

Explorer users make requests to search for all kinds of information about a blockchain. All these requests go through a special balancer that evenly distributes the load and sends the corresponding requests for reading from the Slave storage.

Thus, data buffering and read access channel from the blockchain with high bandwidth is performed.

The system also has a channel for writing information to the blockchain. By means of a special Gateway API interface, which is directly connected to the system nodes, the work of Console, desktop wallets and wallet applications for mobile is provided. The structure of all these services is fully

decentralized, users own seed-phrases and private keys. After the formation of corresponding transactions (write requests onto the blockchain), the user signs them with his signature and sends them to the network. Then the transactions are processed, verified, included in blocks and after reaching a consensus between the validators, they are written into a chain of blocks with replication on each full node.

There is a special service Status in Decimal architecture. The monitoring service tracks, collects and provides general network parameters.

3. Stack of technologies

3.1. Programming languages

For correct compatibility with the Cosmos SDK and Tendermint, we chose Golang as the programming language for the implementation of Decimal's functionality, namely masternode software (validators).

To code the backend modules, we chose TypeScript, which is strictly typed and convenient in the development process, compiles in JavaScript, runs in modern browsers and is compatible with NodeJS. Specifically, TypeScript was used for Workers and Indexer.

To implement desktop wallet applications, the Decimal team used ElectronJS, which allows us to create cross-platform desktop applications based on JavaScript, HTML, and CSS.

3.2. Frameworks

3.3 Other technologies/solutions

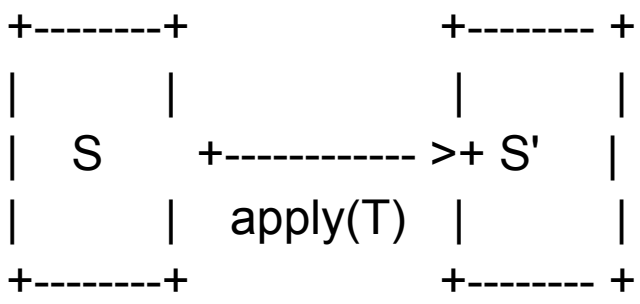
4. Cosmos SDK

Decimal blockchain is developed on the basis of Cosmos SDK - a framework (a set of tools) for developing blockchain applications focused on specific tasks (problems). Cosmos SDK provides a secure and reliable solution to most common tasks for blockchains, such as organizing network communication between nodes of the network and ensuring a reliable consensus between the nodes involved in the formation of blocks. In the Cosmos SDK, this is achieved through active use of the Tendermint Core library (in more detail further).

Thanks to building on the Cosmos SDK, Decimal Blockchain is compatible with all blockchains in the Cosmos

Network, which already has 112 projects: https://cosmonauts.world/.

In its essence, a blockchain is a replicated machine of interrelated states.



The state machine is a concept of computer science, according to which a machine can have several states, but only one at any time. There is a state that describes the current system state, and transactions that trigger state transitions.

Given state S and transaction T, the state machine returns a new state S'.

In a blockchain context, the state machine is deterministic. This means that if a node runs into a specified state and repeats the same transaction sequence, it will always have the same end state at its output.

In a blockchain context, the state machine is deterministic. This means that if a node runs into a specified state and repeats the same transaction sequence, it will always have the same end state at its output.

The Cosmos SDK gives developers maximum flexibility in determining the state of their application, transaction types and state transition functions.

5. Tendermint

Tendermint is an advanced consensus solution based on BFT (Byzantine Fault Tolerance). BFT consensus guarantees the correct operation of a computer network as long as at least 2/3 of the network nodes involved in block formation (validators) work correctly.

Tendermint consists of two main technical components: consensus mechanism engine and application interface. The consensus mechanism engine, called Tendermint Core,

ensures that the same transactions are recorded on each machine in the same order. The application interface, called Application BlockChain Interface (ABCI), allows transactions to be processed in any programming language.

In other words, Tendermint provides efficient relaying of blockchain changes across the network, ensuring that each node on the network has the same transaction log and blockchain state.

The pBFT (practical Byzantine Fault Tolerance) consensus mechanism is a key part of Decimal blockchain and we use it without any changes.

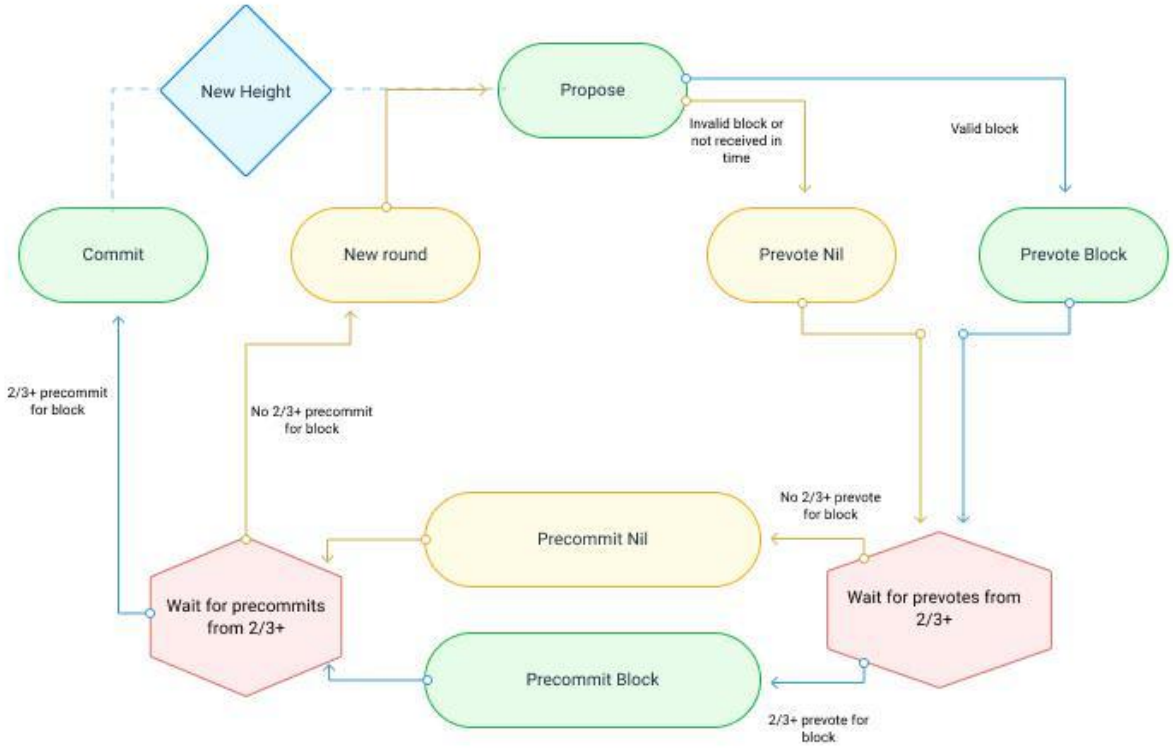


Fig. 2 - Consensus-building process in Tendermint.

6. DEL emission

Decimal native coin is called **DEL**. Within the test network - **tDEL**. DEL is emitted during the generation of each block. Block Reward = Base Reward per block + Total Fee of all transactions in the block. The initial base reward for a block will be 50 DEL. And then every 432 000 blocks (about 30 calendar days) it will increase according to the following algorithm:

- first 12 months (1st year) - increase by 5 DEL;
- next 12 months (2nd year) - increase by 17 DEL;
- next 12 months (3rd year) - increase by 29 DEL;
- next 12 months (4th year) - increase by 41 DEL;
- next 12 months (5th year) - increase by 53 DEL;
- next 12 months (6th year) - increase by 65 DEL;
- next 12 months (7th year) - increase by 77 DEL;
- next 12 months (8th year) - increase by 89 DEL;
- next 12 months (9th year) - increase by 101 DEL;

after that (for the 10th year) the payment of the basic remuneration for the block will stop completely and only the total commission of all transactions in the block will remain.

The figure below shows a chart of base reward for the block for approximately 9 years from the start.

For example, in January 2020 - 50 DEL, in January 2028 - 4658 DEL.

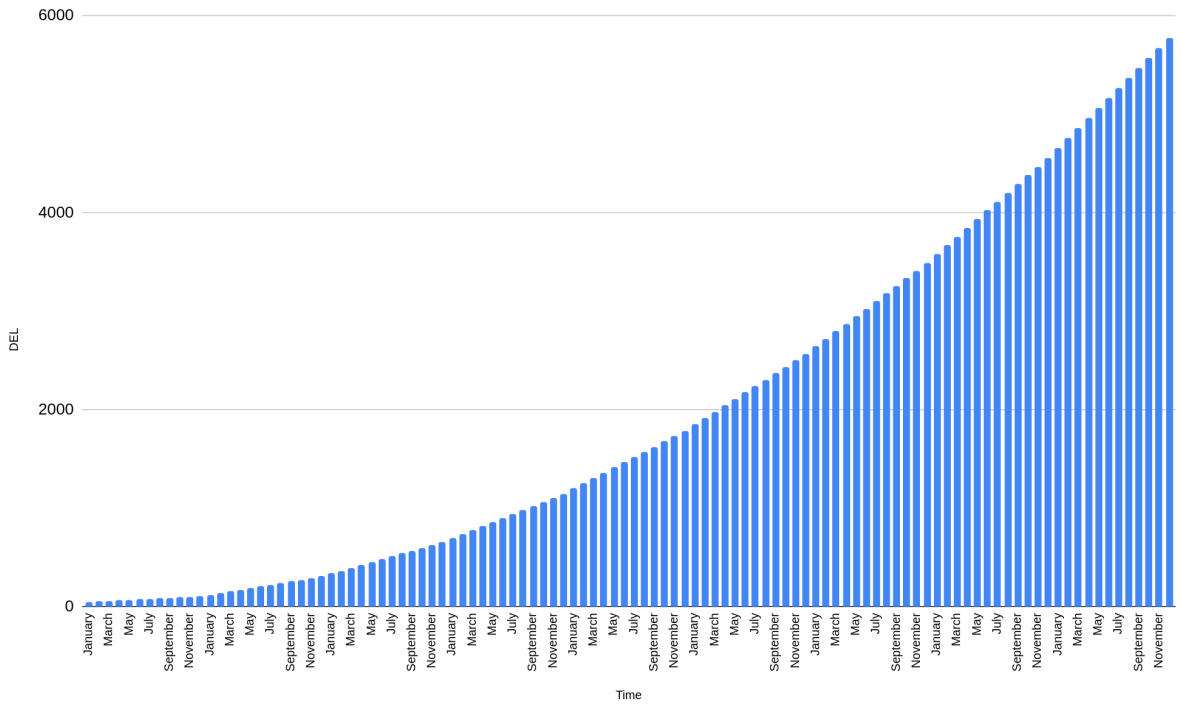


Fig. 3 - DEL emission for over 9 years.

The table below shows the basic unit compensation during the first two years of operation of Decimal blockchain.

When starting the network in the genesis block DEL pre-mine will be made, which will be 200 000 000 DEL. In this case each of the 4 starting validators will receive 40 000 000 DEL (in total 160 000 000 DEL).

Month, 2020	Reward, DEL	Month, 2021	Reward, DEL
January	50	January	122
February	55	February	139
March	60	March	156
April	65	April	173
May	70	May	190
June	75	June	207
July	80	July	224
August	85	August	241
September	90	September	258
October	95	October	275
November	100	November	292
December	105	December	309

Table 1 - the base reward for the block.

The remaining 40,000,000 DEL will be put up for sale and sold to investors of the project.

7. Types of transactions

The following types of transactions are implemented in Decimal:

1) **Send** (sending coins);
2) **Buy** (buying a coin);
3) **Sell** (selling a coin);
4) **SellAll** (full sale of coins);
5) **Multisend** (sending to multiple recipients);
6) **Delegate** (delegating a coin to a validator);
7) **Unbond** (recalling a coin from a validator);
8) **RedeemCheck** (repayment of the check);
9) **CreateMultisig** (creates a multi-signed address);
10) **CreateTransaction** (creates an output transaction from a multi-signed address);
11) **SignTransaction** (unique identifier of transaction multisignals);
12) **CreateCoin** (creates a coin);
13) **DeclareCandidate** (creates a candidate for validators);
14) **EditCandidate** (editing of candidate data for validators);
15) **SetOnline** (activation of the validator);
16) **SetOffline** (deactivating the validator).

8. Transaction fees

On the **Decimal** blockchain the transaction fee consists of the fixed rate for the type of transaction and the cost for the transaction volume in bytes.

Fixed rates: 1 unit = 0,001 DEL

- send - 10 юнитов - 0.01 DEL
- multisend - 10+(n-1)x5 units (n - number of recipients) - 15 units (2 recipients)
- sell - 100 units - 0,1 DEL
- sell - 100 units - 0,1 DEL
- buy - 100 units - 0,1 DEL
- declare candidacy - 10 000 units - 10 DEL
- edit candidate - 10 000 units - 10 DEL
- delegate - 200 units - 0,2 DEL
- unbond - 200 units - 0,2 DEL
- set online - 100 units - 0,1 DEL
- set offline - 100 units - 0,1 DEL
- create multisig - 100 units - 0,1 DEL
- create multisig transaction - 100 units - 0,1 DEL
- sign transaction - 100 units - 0,1 DEL
- redeem check - 30 units - 0,03 DEL

create coin

- 3 letters - DEL 1 000 000
- 4 letters - DEL 100 000
- 5 letters - DEL 10 000
- 6 letters - DEL 1 000
- 7-10lettersбукв - DEL 100

Cost of 1 byte of final transaction volume: 2 units (0,002 DEL)

In fact, a transaction is just an information message. It states what, how much, to whom and from whom it is sent, as well as service data. The transaction volume is the volume of all information the transaction consists of:

- service (signatures, parameters, etc.);
- user (length of the coin ticker to be sent, length of the fee coin ticker, amount to be sent, text message).

Based on our testing experience during the development of the **Decimal** we have the following indicative data on the each transaction fee:

- send ~ 0.41 DEL
- multisend ~ 0,479 DEL (2 recipients)
- sell ~ 0.484 DEL
- sell ~ 0.444 DEL
- buy ~ 0.54 DEL
- declare candidacy ~ 10.674 DEL
- edit candidate ~ 10.494 DEL

- delegate ~ 0.564 DEL
- unbond ~ 0,604 DEL
- set online ~ 0.396 DEL
- set offline ~ 0.394 DEL
- create multisig ~ 0.494 DEL
- create multisig transaction ~ 0.542 DEL
- sign transaction ~ 0.544 DEL
- redeem check ~ 0.03 DEL

9. Multisignature

Multisigsignature (Multisig) addresses will be available in Decimal. This is very convenient when allocating funds or simply making a joint decision when there are several independent participants and the conditions for consensus between them. For example, funds from a shared wallet will only be withdrawn if 80% of the participants agree (4 out of 5).

The multi-signature will work almost like a multisig smart contract on Ethereum blockchain. Three types of transaction are coded to implement the functionality - CreateMultisig, CreateTransaction, SignTransaction:

CreateMultisig creates a wallet with multisignature, indicating the owners, their voice weight and threshold value (for example, 3 out of 5 or 2 out of 3). In this case, the multisignature address is generated with the addition of "salt" to allow you to create many addresses with the same parameters, and stored in the storage.

CreateTransaction creates a transaction to output the specified number of specified coins to the specified address, the transaction creator immediately signs this transaction. Each such transaction is assigned a unique identifier.

Other owners of a multi-signed wallet may request transactions to this address, their parameters and identifiers. After that they simply send a transaction of SignTransaction type to the blockchain, where in the parameters there will be a unique identifier of the transaction from paragraph 2. After checking the signature the number of collected "votes" of the transaction is increased in accordance with the weight. If the threshold value is reached, the transaction is executed, the funds are transferred, the balance of the address with the multisignature, and the recipient's address is changed.

As a result of this scheme of interaction between the multi-signature participants, we exclude offline communication between them. At least it becomes optional.

10. CRR

There are three distinctions which derive Decimal from the row of DPOS structures:

1) effortless coin issue process for a community;

2) network-wide coin swap without 3rd-party software;

3) validators get fees in selected coin.

In addition to traditional coins, the DEL native coin serves as collateral. Each custom coin is backed by a guarantee in the form of a certain amount of DEL. The size of this warranty in relation to the total issue of a custom coin directly affects the price curve of the coin.

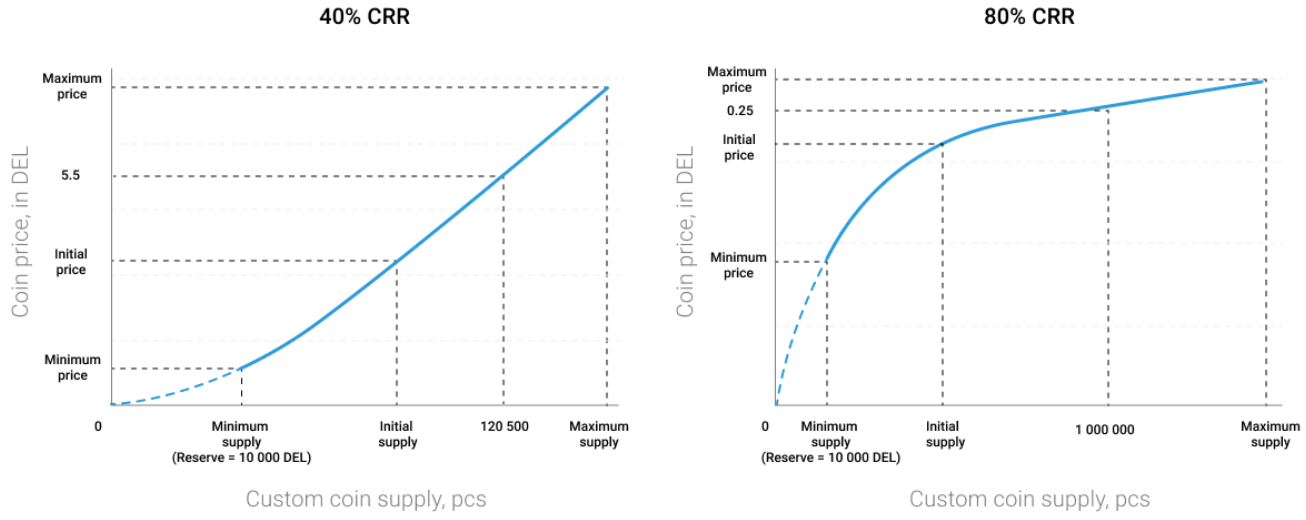


Fig. 4 - coins with different CRRs.

The CRR parameter is set when the coins are created and the value curve remains unchanged forever. When market conditions and balance change, the demand/supply value of the coin moves up or down the calculated curve.

11. Formulas for determining coin price

The key parameters in the Decimal network are Reserve, CRR and the total number of issued coins. All of them are also involved in the calculation of buy or sale price.

The very existence of two formulas is a consequence of the nonlinear nature of changes in coin price.

Calculation of coin buy sum:

Buy sum = Reserve * (-1+(((Buy + Supply) / Supply) ^ (100 / CRR)).

, where

Reserve - current reserve in DEL;
Buy - the number of coins to buy;
Supply - total number of coins;
CRR - coefficient of Constant Ratio to the Reserve (e.g., 20 for 20%).

Calculation of coin sell sum:

Sell sum = Reserve * (1 - (1 - Sell / Supply) ^ (100 / CRR))

, where

Reserve - current reserve in DEL;
Sell - the number of coins to sell;
Supply - total number of coins;
CRR - coefficient of Constant Ratio to the Reserve (e.g., 20 for 20%).

Calculation of 1 coin current price:

Price = Reserve * (1-(1-1/supply)^ (1 / CRR))

, where

Reserve - current reserve in DEL;
Supply - total number of coins;
CRR - coefficient of Constant Ratio to the Reserve (e.g., 20 for 20%).

12. What is a masternode

There are 2 types of network nodes in Decimal. A full node and a regular node.

Masternode (aka Full Node) is a Decimal network node that stores a replica of a blockchain and participates in consensus building.

The masternode is a hardware-software complex, which acts as a validator in the network. The hardware is connected to the Internet and directly to other validators to ensure the main task - consensus.

Requirements for equipment:
4GB RAM - **RAM** capacity;
1 TB SSD - hard disk capacity and type;
x64 2.0 GHz 4 vCPUs - CPU characteristics.

Each Decimal network masternode stores a full copy of the blockchain. All transactions, all blocks, starting with the genesis block, all messages. This copy is called a replica. It is identical to the replicas on each of the other masternodes.

Besides, additional services and services that are necessary for establishing connections with other masternodes via the Gossip protocol, verifying user transactions, forming blocks from transactions, writing everything directly into the local blockchain replica are deployed on the masternode.

Each masternode works under strict conditions of punishment/ encouragement.

The reward is based on correct and reliable performance. The amount of reward is proportional to the total sum of stakes of each masternode. The bigger is the validator's stake, the bigger part of rewards for the block will be received by this validator (masternode).

Penalties/slashes are imposed on the incorrect operation of the masternode. For example, the inaccessibility of a validator within 12 blocks of the last 24 is punished by 1% of the total value of the validator stake. And the total sum of the stake includes all the funds delegated to this validator.

Validators who attempt fraudulent actions in the consensus-building process are also penalized. Namely, when signing 2 different blocks during the rounds of verification and voting for a block candidate. This is a serious violation, which can lead to a fork of the chain of blocks. In case of fork appearance part of network users will be oriented on one variant of block state (transactions, account balances), while other parts of users will be oriented on the second variant of the block already with other states and balances. In this case the validator will be punished with 5% fine. Plus all coins delegated to this validator will return to their owners (its stake will be reduced).

One more time,

1) Inaccessibility within 12 blocks of the last 24 blocks - 1% stake penalty + validator shutdown;
2) Double signature - 5% stake penalty and forced unbonding of coins.

It should be noted that penalties are not transferred anywhere, but simply burned. Reducing the total supply of DEL.

13. A regular node

The second type of node is a regular node. These are all members of the network, except for masternodes (validators).

These are ordinary users of the network: wallet holders, custom coin issuers, businesses and individuals, coin delegators, and others.

Each of them owns private keys to their wallets and uses all services and applications available on Decimal ecosystem. They keep DEL and other caste coins on their non-caste wallets. They send them to other users, delegate them to the validators they like, and receive rewards from validators.

In general, ordinary nodes have all the functionality available, except the obligation to participate in consensus building and to store a replica of the blockchain.

14. Explorer

A wide range of Decimal users needs to always understand what's going on in a blockbuster network at a given time, what its main parameters are. Users want to double-check and search for information about their transactions, blocks, commissions, rewards, etc.

So we developed the Explorer unit.

Assuming that users will be interested in a very large amount of information about the status and processes in Decimal network, namely transaction details, blocks, validators and their parameters, issued coins, etc., we must take care of the availability of all this data and ensure that it is displayed correctly and quickly. As time passes and the blockchain increases, it will become more and more difficult to process requests. Significant time delays are possible when sampling data directly from blockchain replicas. But we organize storage in a database that can satisfy a huge number of requests and guarantee to provide the information needed by users.

Fig. 5 shows the architecture of our solution. The process is organized as follows.

State changes in the blockchain generate events (events, events), which are monitored by special services (Workers, workers). These services parse all incoming information from

blocks and transactions and transfer it to the Index, in which data are sorted and indexed. The ordered data are then written to the PostgreSQL database. They are written to the master database (Master) and duplicated on the slaves (Slave) to ensure safety.

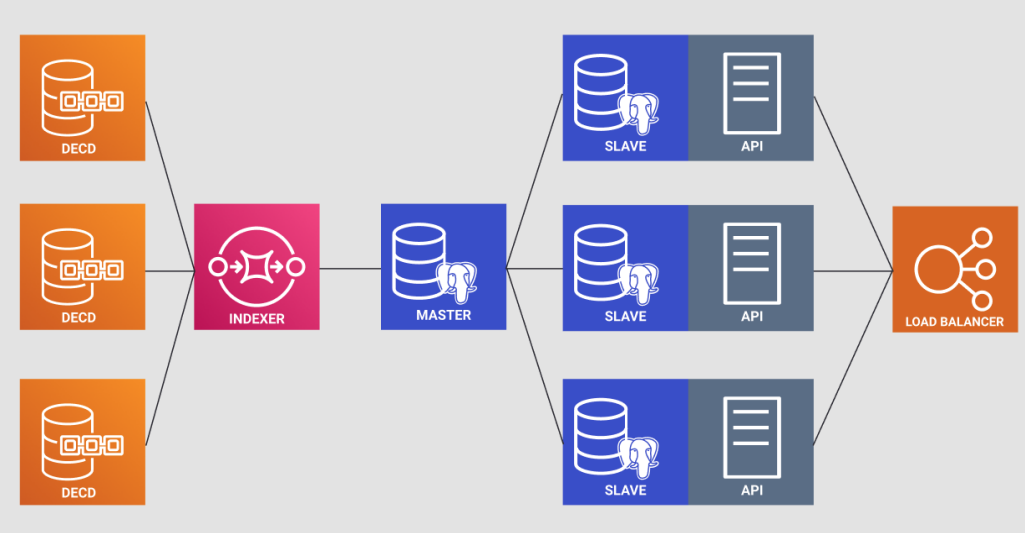


Fig. 5 - the architecture of Decimal block explorer.

All requests from the Explorer are received by the Slave Databases through a Load Balancer that organizes load balancing evenly through a software interface (API). The Load Balancer is horizontally scalable, i.e. slave databases can be added if the load increases. In this case, the masterminds where the blockchain replicas are directly stored are not loaded, because they are architecturally separated from the requests of users and external services. Through indexing and partitioning, databases are able to provide information about

any events and states in the blockchain with minimal delay, regardless of the size of the database itself.

15. Console

Decimal already at the start of the project provides the user with a number of services that reveal the main features. We consider it very important to provide convenient access to all services. Therefore, all of them have been brought together on the same site. It is called Decimal Console (https://console.decimalchain.sg).

Wallet (https://console.decimalchain.sg/wallet) - purse address, the balance of your funds in DEL, list of coins in possession, list of completed transactions and their parameters, the functionality of sending coins.

Conversion (https://console.decimalchain.sg/convert) - service for exchange, purchase and sale of any coins in Decimal network.

Delegation (https://console.decimalchain.sg/delegation) - interaction with validators, transfer of their coins in order to increase their stake and get rewards from them. You can also revoke your delegated coins here.

Delegating user funds to validators is a key feature of Decimal blockchain. With a large number of coins we want to provide the user with a convenient tool for organizing delegation. So we have implemented automatic delegation. At the preliminary stage, we offer to form, either offline or online, a package of transactions and configure a separate transaction to initiate the automatic start of delegation.

Masternode

(https://console.decimalchain.sg/masternode) - the service organizes the process of connecting and running the validator functionality. The number of validators in Decimal is regulated and increases in proportion to network growth. The strategist will have 4 validators and 12 slots of additional slots, i.e. the maximum number of validators at the start is 16. Then every calendar month 4 validators will be added.

Before you can become a validator, you have to declare your candidacy. Validators with the largest stake size participate in the consensus mechanism. The stake is recalculated for each block. Accordingly, the candidate is selected based on this parameter, as well as on quality indicators (online availability, no fines) of his work.

We deployed Decimal test network that is architecturally completely identical to the main network. This is the ability to debug the startup and management processes of the master. Click on https://testnet.console.decimalchain.sg for more information.

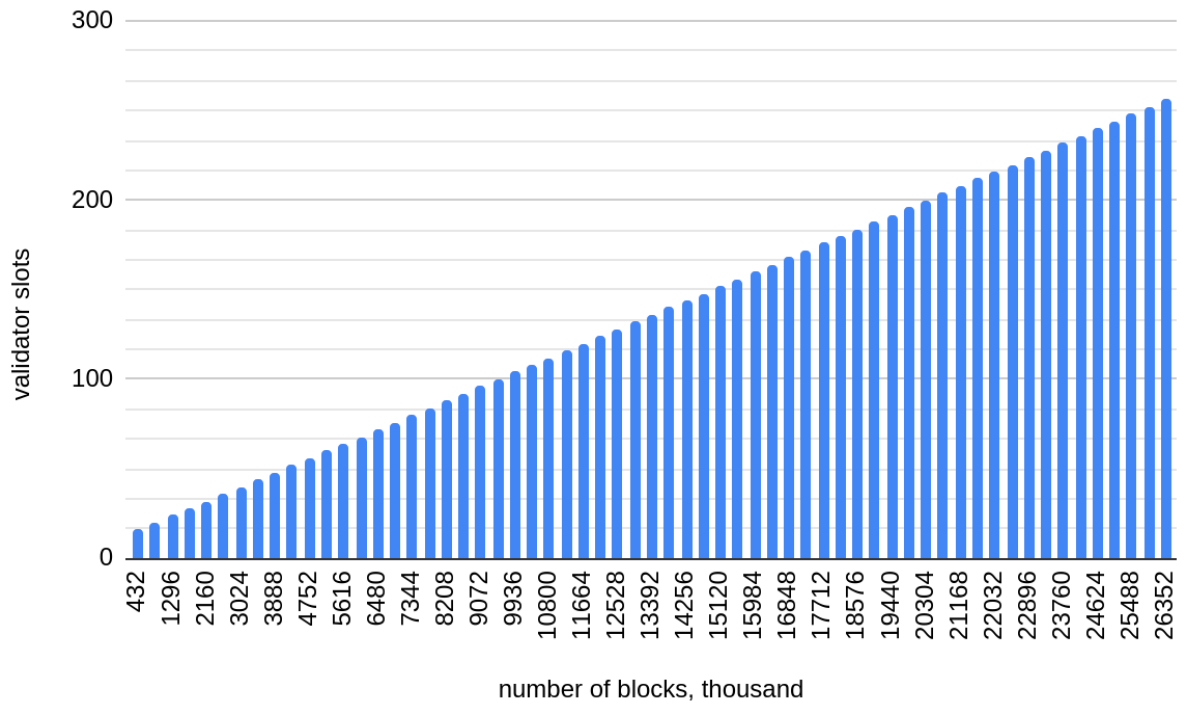


Fig. 6 - increase in the number of slots for validators.

Coin issue (https://console.decimalchain.sg/issue-a-coin) - here the functionality of coin creation is implemented. The algorithm is simple, fill in five fields and the coin is created. To understand how the value of your custom coin will behave when buying, selling, and exchanging, we have made an additional service - Calculator (https://calculator.decimalchain.sg). In it, you create a virtual coin, perform the corresponding transactions, and observe the change of coin value.

Broadcast (https://testnet.console.decimalchain.sg/broadcast) is a service for sending offline transactions generated by users to

Decimal network. Since security is a key priority, we offer a tool that will eliminate any possibility of compromising users' private data. Here we generate a **nonce** parameter for the user, which is included in the transaction. Once the transaction is created offline, the user can send it to the network without having to worry about their private keys.

Status (https://status.decimalchain.sg) - this resource displays the main Decimal global metrics at the current time, such as the status of the network, services and services of the blockchain, the number of coins issued, etc.

API & SDK (https://help.decimalchain.sg/api-sdk) is a description of the program interface for interaction with Decimal services and tools for building applications based on Decimal blockchain.

Help / FAQ (https://help.decimalchain.sg/ru/) is a traditional section needed to explain technical and non-technical details, various guides, manuals, answers to frequently asked questions, etc. Despite the fact that everything is simple from the point of view of use in Decimal, there are many different technologies and nuances on the technical level. We strive to make it easier to learn Decimal and help the widest possible range of users.

16. Status

As mentioned above, the service Status is organized on the basis of the monitoring service. It consists of several parts deployed on Decimal capacity and network nodes:

1) monitoring server;
2) database;
3) web-interface;
4) demon agent, at the monitoring sites.

To get acquainted with the service click on the link https://status.decimalchain.sg

Information about transactions, blocks, commissions, validators, time parameters, coins is collected directly from the blockchain.

For example, for the **Network** parameter, we check every 30 seconds if new blocks are formed on the validator nodes. If new blocks are formed, the network is in the **"Active"** status.

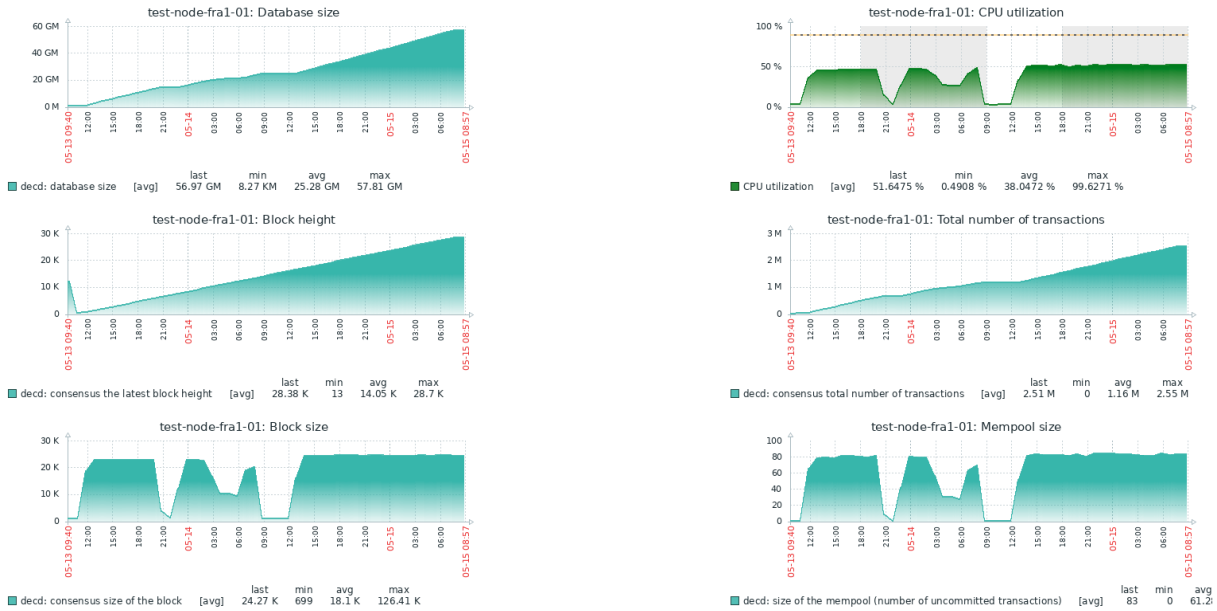


Fig. 7 - an example of monitoring while developing.

For the **Explorer** and **Gateway** services, we check whether they are running on servers and whether they are available (responding or not).

17. Wallet

All official Decimal wallets are decentralized, i.e. non-custodial. Seed phrases and private keys are stored strictly on the user side and decimal has no access to them in any way. The entire responsibility for the safety of these private data and funds on wallets lies with the owners of wallets.

At the start of the project the following wallet applications are available:

1) for browser desktop https://wallet.decimalchain.sg;
2) for browser desktop via console https://testnetconsole.decimalchain.sg/wallet;
3) for Android devices https://play.google.com/store/apps/details?id=com.chain.decimall=en;
4) for iOS devices
5) Windows, macOS and Windows desktop version

Private keys are generated based on the BIP39 standard. Elliptical curve **secp256k1**.

The Seed Phrase consists of **24** words.

18. Address format (Bech32)

One of the specific features of blockchains (first Bitcoin, and then many others) are address formats. They represent a sequence of letters in the Latin alphabet and numbers. The problem is that this makes it very difficult for users to read them correctly.

Blockchain enthusiast and developer Pieter Wuille suggested upgrading the format of addresses in the Bitcoin network. This proposal is known as BIP 173[1] or bc1 addresses and as of May 2020 it was successfully implemented[2] in a significant number of cryptographic projects, including those outside the Bitcoin Blockchain.

At the moment the implemented changes are known as Bech32 address format.

The whole Decimal team, supporting the Bech32, will provide it at the start of the project.

[1] https://github.com/bitcoin/bips/blob/master/bip-0173.mediawiki
[2] https://en.bitcoin.it/wiki/Bech32_adoption

The Bech32 address is no longer than 90 characters long and contains:

1) The part that's easy to read by humans. This includes data that may need to be transmitted or that have anything to do with the owner of the address, a minimum length of 1 character. For example, by default "bc" is used for mainnet addresses and "tb" for testnet addresses.
2) A separator that always equals "1". If "1" is allowed inside the human-readable part, the separator is the last of the characters "1".
3) The data portion is at least 6 characters long and consists of alphanumeric characters only, except for "1", "b", "i", and "o".
4) Checksum. The last six characters of the data part form a checksum and contain no information.

5) All letters are lowercase, although it is possible to convert them into uppercase letters for generating QR code.

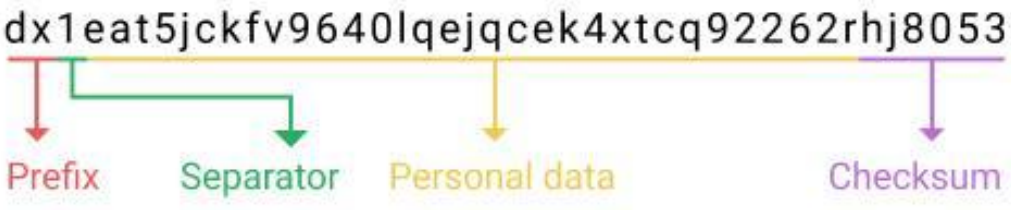


Fig. 8 - Decimal address structure.

19. Console client

In order to facilitate the process of creating and debugging certain processes or scenarios of interaction with Decimal, whether it is on-line or off-line services, execution of scripts, execution of block requests, sending any types of transactions or receiving service information and parameters, our team prepared a console client.

The file to run the client is **deccli**.

The client acts as a REST server, which provides the ability to create a wide range of requests to the blockchain.

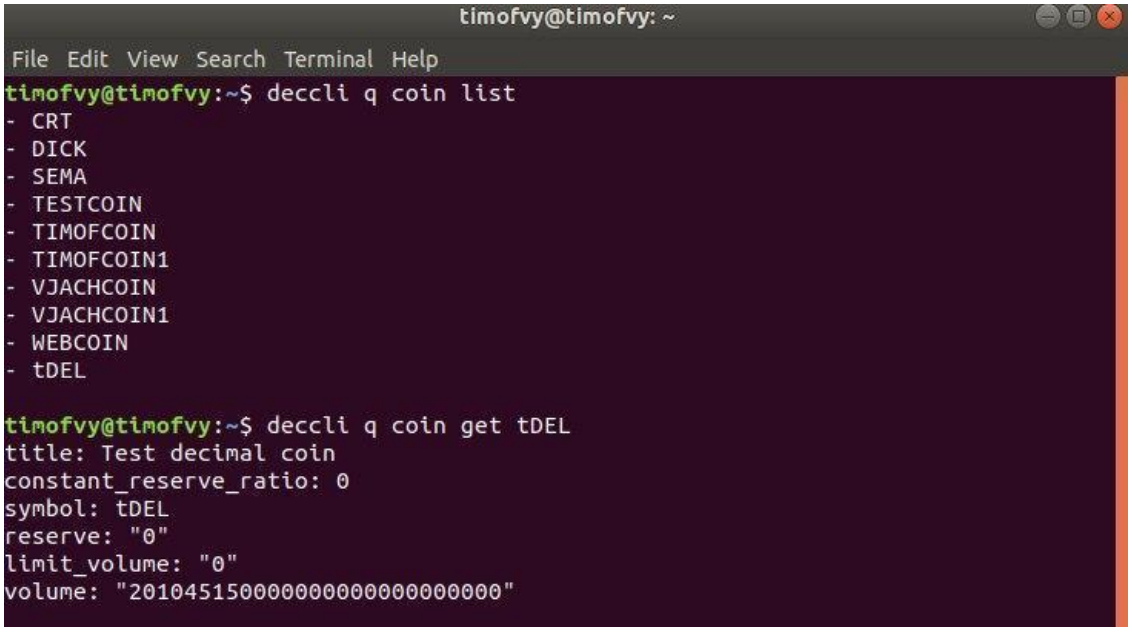


Fig. 9 - the interface of the console client, coin list and coin get command.

Essentially, the console client is a "light node", or "light daemon" and redirects requests to the full node, and therefore does not require synchronization with the blockchain (it does not store a replica of the blockchain), takes up very little disk space, but has rich and comprehensive functionality.

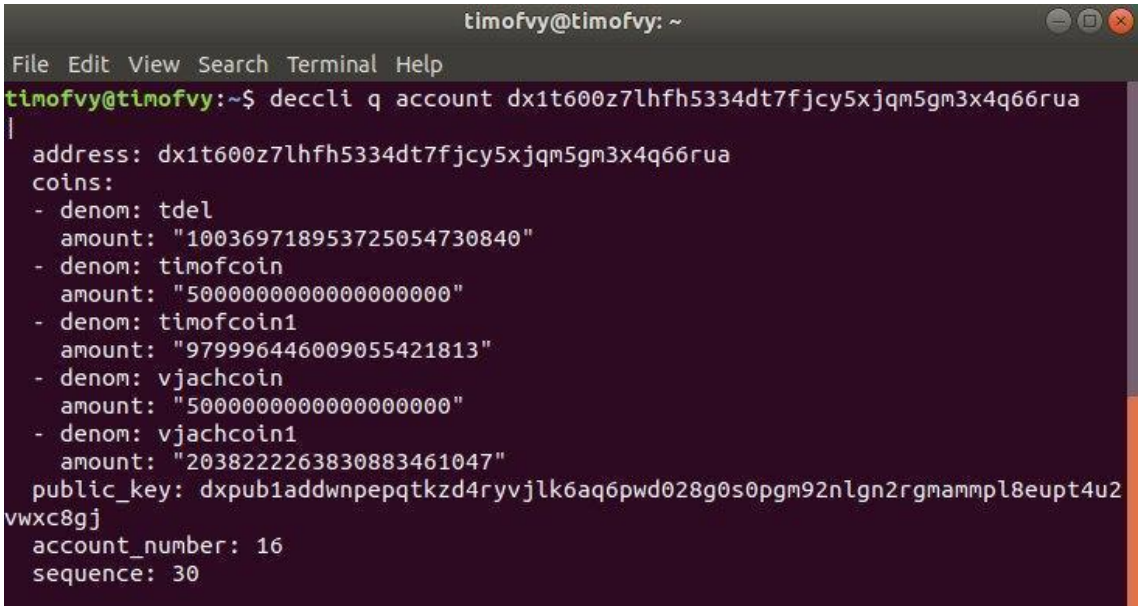


Fig. 10 - console client interface, account information

20. Roles

20.1. Validator

A member of the Decimal network who keeps a blockchain replica on his equipment, provides transaction verification, chain blocks formation, and participates in the consensus-building process. One of the main characteristics of the validator is Voting power, which is directly proportional to the size of the stake of the validator. The bigger is the

validator's stake size in absolute numbers and the bigger is its share in the aggregated stake of all validators, the more voice power it has, the more often it produces blocks, the more reward it receives for its work.

The Validator must ensure the reliability of its work, be available on the network, correctly perform its duties in the consensus-building procedure, guarantee access of Decimal network internal services to its blockchain replica, ensure the integrity of stored data.

Functioning as a validator is not only about getting the reward for work, but also the risk of being fined for improper performance of duties. A validator risks losing part of its stake as a result of fines, as well as reputational losses if users stop trusting it with their funds for delegation. Reputational losses can have a very strong impact on validators.

Validator stack consists of its own funds and funds delegated (leased) to it by other members of the network.

20.2 Delegator

A Decimal member who leases his funds to a validator. The transfer is not done literally. The amount of transferred funds is simply blocked in the account of the delegator and he can not dispose of it until he unbonds his funds.

The transfer process is called delegation. The essence of it is to transfer the users’ rights and responsibilities. A delegator increases the validator's stake and it turns out that the validator performs his duties on behalf of the delegator, within

the amount of delegated funds. Accordingly, rights and responsibilities apply to both the reward payouts and fines/slashes. Rewards and penalties are distributed in proportion to the delegator's share of the total stake of the validator. If the validator receives a 5% fine, the delegator will receive the same 5% fine automatically.

20.3. Coiner

A member of Decimal blockchain, which produces its own custom coins. As you already know from Decimal White Paper, the applications for crypto coins are limitless. We have made the coin making process exceptionally simple and fast.

After setting initial parameters, such as the size of the reserve in DEL, the number of coins issued and the constant reserve ratio (CRR), a special transaction is sent to the network. This completes the process. You get a coin with full functionality.

Moreover, you do not have to worry about the technical mechanisms of custom coin price formation. Any exchange operations involving your coin will be based on mathematical formulas embedded in Decimal software and automatic blockchain algorithms. The more demand, the higher the coin price and vice versa.

Any custom coin can be delegated to a validator. It can be exchanged for any other Decimal custom coin or DEL.

Transaction fees are also paid in any custom coin.

20.4. User

All other members of Decimal network are regular users. We see each user as a potential coiner, delegator and/or validator. After a successful experience in storing Decimal coins, sending them, paying for goods and services, as well as other applications in real life, the step towards creating your own coin will be easy. Our mission and task are to remove all barriers in front of the users on the way to their goals.

21. Delegation

Delegation is the process of binding user coins (DEL or any custom coins) to the validator(s). The process is performed using a special transaction **Delegate** (see Types of transactions). The user delegating coins is called a **Delegator**. After binding the coins, the delegated funds are blocked in the user's account, i.e. they are not sent anywhere, but are **NOT** displayed on the user's balance. However, these funds are displayed in the common validator stake, increasing its weight in relation to other validators and the strength of the consensus mechanism voice. The validator is not able to

manage the delegated funds in any way, does not have access to them, cannot spend them or withdraw them.

Recall that among the validators there is a competition - validators with a large stake are most often involved in consensus-building and receive more reward.

Decimal has an unbond operation for delegated coins. This transaction can be initiated and sent to the network either by the user or automatically by the validator or software. In the first case, the funds will be immediately available on the user's balance. In the second and third cases, the funds will be available to the user in 432,000 blocks (~ 30 days).

Since validators are fined for poor quality or incorrect work (see What is a masternode), the number of delegated coins may decrease by the number of fines. The delegator is personally responsible for his/her choice of validator, so this choice must be made very carefully.

Coin delegation takes place in so-called slots. Each validator has 1000 slots. Each individual coin is delegated to its own slot. That is, all DEL coins will be placed in one slot, and for each next coin there will be the next slot. There is no limit on the number of coins in a slot. But if the delegator has all 1000 slots full, the next delegator must send funds **NOT** less than in the last slot. If he sends more, the funds in slot number 1000 are forcibly unbond and immediately appear on the owner's balance.

For delegation (process of binding user coins (DEL or any custom coins) to the validator(s)) the user receives a reward.

The reward is credited every 120 blocks.

The amount of the reward depends on a number of factors: the number of delegated coins, the number of validators, the total amount of stakes, the size of the basic block reward, slashes imposed, etc.

22. Help / FAQ

Our technicians have drawn up a technical description of Decimal's structure, technology and services. This makes it easier to understand internal processes, details of procedures and mechanisms embedded into the blockchain. The service includes both the technical description, addressed to developers or business owners, and a wide range of ordinary users who are looking for answers to their questions and ways to resolve any difficulties in interacting with Decimal.

Click the following link for more information

https://help.decimalchain.sg

Articles and answers to questions will be regularly updated and enriched with new information as the community and Decimal Blockchain develops.

23. Block structure

Block is a fundamental essence of Blockchain. An element of a blockchain that contains user transactions, DEL issuance transactions, validator signatures, commissions, the hash of the current and previous block, and other service information.

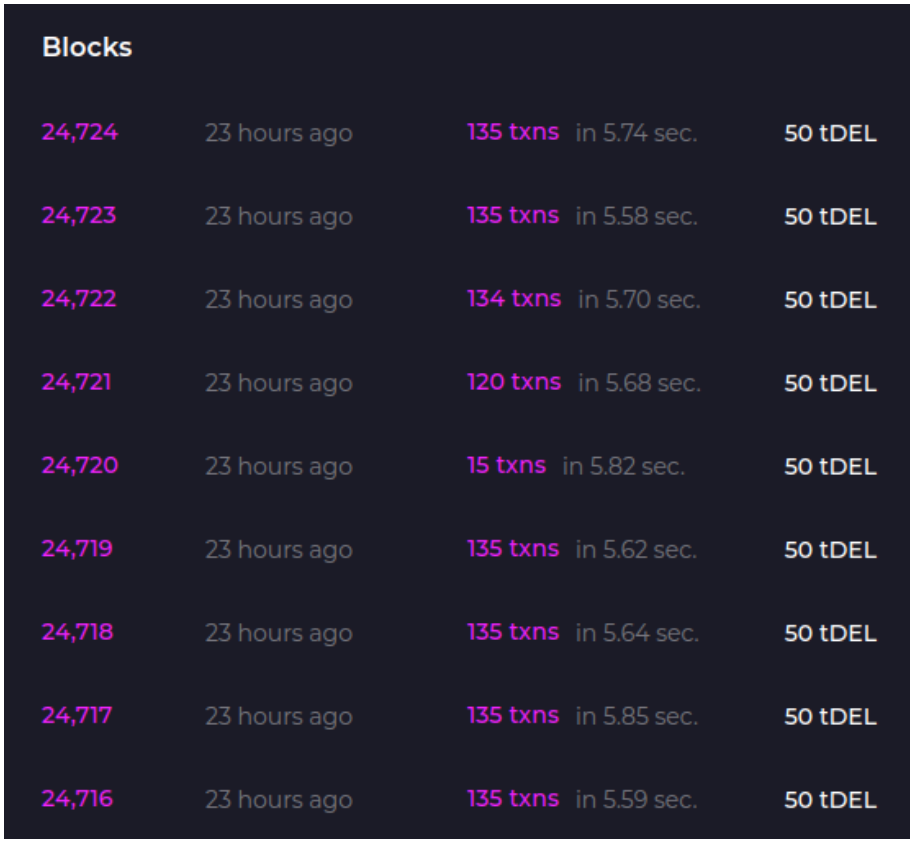


Fig. 11 - Decimal blocks on the Explorer.

In Decimal blockchain, blocks are generated approximately every 5.5 to 6 seconds.

The block contains between 0 and 10,000 transactions, approximately 180 bytes each. The block is rewarded according to the emission model, at the start of 50 DEL with a subsequent increase every 432,000 blocks (~ 30 calendar days).

Decimal is based on the Tendermint engine. For more details about the block structure, see its specification (https://github.com/tendermint/spec/blob/953523c3cb99fdb8c8f7a2d21e3a99094279e9de/spec/blockchain/blockchain.md).

Here are some excerpts from the documentation.

23.1. Data structure

The block includes the following list of basic data types:

- Block
- Title (Header)
- Version
- Block Identifier (BlockID)
- Time
- Data (for transactions)
- Confirmations and votes (Commit and Vote)
- EvidenceData and Evidence

23.2. Block

A block consists of a header, transactions, votes and a list of evidence of violations (e.g. a double signature).

```
type Block struct {
  Header  Header
  Txs     Data
  Evidence     EvidenceData
  LastCommit Commit
}
```

Note that **LastCommit** is a set of signatures of the validators that signed the last block.

23.3. Header

The block header includes metadata about the block and about the consensus, as well as data confirmation in the current block, the previous block and the result returned by the application:

```
  type Header struct {
// basic information about the unit
   Version Version
   ChainID string
   Height    int64
  Time     Time

// information about the previous block
```

LastBlockID BlockID

// block data hashes

LastCommitHash []byte // confirmations from validators from the previous block

DataHash []byte // Merkle tree of transaction hashes

// application data hashes from the previous block

ValidatorsHash []byte // current unit validators

NextValidatorsHash []byte // validators of the following block

ConsensusHash []byte // consensus parameters for the current block

AppHash []byte // application status after transactions from the previous block

LastResultsHash []byte // root hash of all transaction data from the previous block

// information to achieve consensus

EvidenceHash []byte // conflicts in this block

ProposerAddress []byte // creator of this block

23.4. Transactions

Data is a wrapper of a list of transactions, which are arbitrary byte arrays:

```
type Data struct {
  Txs [][]byte
}
```

24. References

1. Jae Kwon, Tendermint: Consensus without Mining Draft v.0.6 (outdated), 2014.
https://tendermint.com/static/docs/tendermint.pdf

2. Practical Byzantine Fault Tolerance Miguel Castro and Barbara Liskov Laboratory for Computer Science, Massachusetts Institute of Technology, 545 Technology Square, Cambridge, MA02139
http://www.pmg.csail.mit.edu/papers/osdi99.pdf

3. Vitalik Buterin A Proof of Stake Design Philosophy
https://medium.com/@VitalikButerin/a-proof-of-stake-design-philosophy-506585978d51

4. Cosmos SDK
https://docs.cosmos.network/

5. Tendermint
https://docs.tendermint.com/

6. Hayek, Friedrich August von.

https://ru.wikipedia.org/wiki/%D0%A5%D0%B0%D0%B9%D0%B5%D0%BA,_%D0%A4%D1%80%D0%B8%D0%B4%D1%80%D0%B8%D1%85_%D0%90%D0%B2%D0%B3%D1%83%D1%81%D1%82_%D1%84%D0%BE%D0%BD

7. Kaines, John Maynard.
https://ru.wikipedia.org/wiki/%D0%9A%D0%B5%D0%B9%D0%BD%D1%81,_%D0%94%D0%B6%D0%BE%D0%BD_%D0%9C%D0%B5%D0%B9%D0%BD%D0%B0%D1%80%D0%B4

8. BIP: 173 or bc1 addresses
https://github.com/bitcoin/bips/blob/master/bip-0173.mediawiki

9. Pieter Wuille lecture on new bech32 address format
https://www.reddit.com/r/Bitcoin/comments/62fydd/pieter_wuille_lecture_on_new_bech32_address_format/

10. Bech32, native SegWit address already used on the mainnet
https://www.reddit.com/r/Bitcoin/comments/74tonn/bech32_native_segwit_address_already_used_on_the/dqlogru/

11. Enterprise-class open source distributed monitoring solution.

https://www.zabbix.com/ru/manuals