REPUBLIA

Technical Paper

OUTLINE

REPUBLIA

INTRODUCTION

Trust is created through large-scale cooperation and smart code and not through powerful intermediaries, such as states or banks.

Don Tapscott, state adviser, business consultant, author

INTRODUCTION

The decentralized ecosystem of Republia is a large-scale platform, that operates on the basis of blockchain technology and combines the effective functions of countries that we are familiar with.

Republia ecosystem differs by an innovative technology, implemented using the functional programming language - OCaml.

Republia blockchain will be applicable not only for financial sphere using RPB coin, but also will be a secure tool for full functioning of all elements of Republia Ecosystem.

As Republia operates under the principle “we-ecosystem”, where users influence the life and processes of the ecosystem, Republia blockchain will record next options in order to avoid falcifications and deception:

- voting details (the block records hash of a unique identifier (Republia ID) of those participant, who voted, decisions of the participants and the time of the vote).
- data of projects, created on the basis of Republia ICO Platform
- data of smart contracts, created on the basis of Republia Smart Contract Platform

REPUBLIA

OCAML - LANGUAGE OF IMPLEMENTATION

OCAML - LANGUAGE OF IMPLEMENTATION

Republia protocol is implemented using the functional programming language OCaml, because it minimizes hacking the code of smart contracts, maintaining formal code verification.

Republia mainly focuses on zero error rate in execution of smart contracts, because, according to research, there are 34 200 vulnerable smart contracts per 1 000 000 Ethereum smart contracts.

Hypothesis was verified by 3000 smart contracts, 89% of these contracts could be vulnerable to hacking and provoke a thief of funds of 6 million dollars.[1]

OCaml is a dialect of Meta Language (ML), created by INRIA, the national research institute of France, in 1985.

OCaml is a dialect of Meta Language, initial task of ML is an automatic theorem proving, whereas OCaml was created as an application programming language.

OCaml stands out against the background of other programming languages by mechanisms for implementation secure programs, errors in which can be found out during implementation of code. For this reason the language has static typing, automatic type generation and tools for formal code verification. In addition, OCaml has an optimizing compiler, which deserves separate attention, and the performance of the OCaml code is not inferior to code, implemented by compiled, statically typed, general purpose programming language C, while minimizing the number of errors.

OCaml finds application in those systems, whose functionality depends on each part, where even one error can stop the overall process. Software testing does not perform this role on a decent level, because testing takes into account only limited subset of test cases while the abstract interpretation considers superset of all possible results of the system.

1 on materials of the article: https://arxiv.org/pdf/1802.06038.pdf

OCaml is used in aircraft control software for the family of Airbus A340, where it indicates absence of run-time errors. This fact indicates the necessary application of OCaml language in such areas where one error can lead to irreversible consequences.

Formal verification is a process of proving, that a program conforms to a specific specification.

OCaml is a popular functional programming language, which allows formal verification of individual parts of the protocol. This is an advantage of using OCaml in open source software, because in such systems there are often vulnerabilities that are added by ecosystem participants accidentally or intentionally.

REPUBLIA

INNOVATIVE REPUBLIA BLOCKCHAIN

INNOVATIVE REPUBLIA BLOCKCHAIN

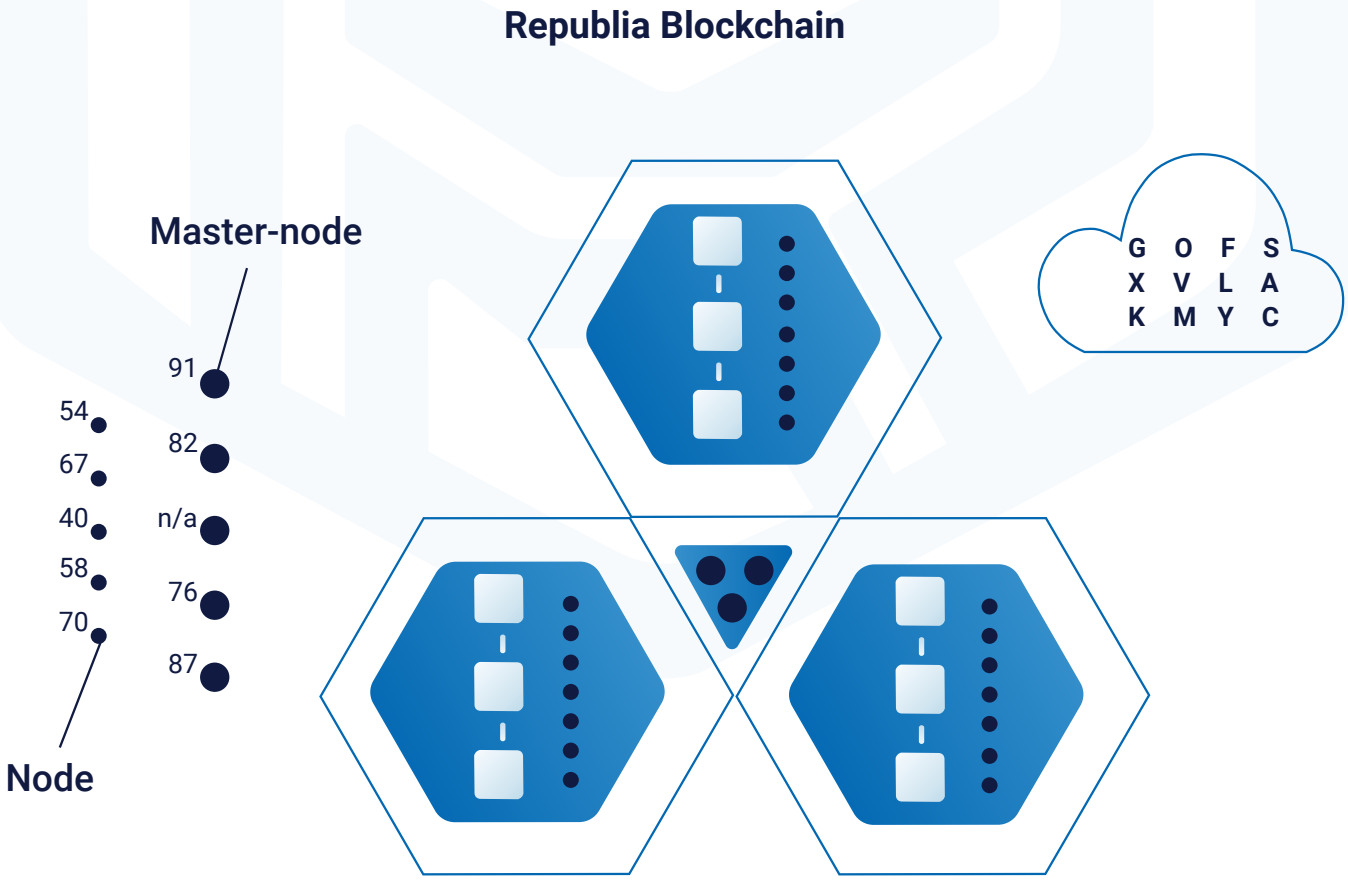
Rebublia Blockchain operates according to the principle of sidechains.

For now in well-known Ethereum Blockchain network operation is exposed to some risks.

Rebublia Blockchain protocol is divided into three main layers:

- Consensus layer;
- Transaction layer;
- Network layer;

The separation of the protocol into layers allow to replace any layer of the protocol, without touching it entirely.



2.1 Unlimited scalability

Rebublia Blockchain consists of multiple chains and solves the problem of low network capacity through dynamically increasing amount of sidechains. Thus, the linear dependance of increasing and decreasing the number of sidechains is tracked, depending on amount of transactions in the network.

Applying cubic approximation to function $y = f(x)$, where $f(x)$ is cubic function $y = ax^3 + bx^2 + cx + d$ and its parameters can be found by searching extremum of the function

$$S(a,b,c,d) = \sum_{i=1}^{n} \left[(ax_i^3 + bx_i^2 + cx_i + d) - y_i \right]^2.$$

When derivation is found:

$$\frac{\partial s}{\partial a} = \sum_{i=1}^{n} 2(ax_i^3 + bx_i^2 + cx_i + d - y_i)x_i^3,$$

$$\frac{\partial s}{\partial b} = \sum_{i=1}^{n} 2(ax_i^3 + bx_i^2 + cx_i + d - y_i)x_i^2,$$

$$\frac{\partial s}{\partial c} = \sum_{i=1}^{n} 2(ax_i^3 + bx_i^2 + cx_i + d - y_i)x_i,$$

$$\frac{\partial s}{\partial d} = \sum_{i=1}^{n} 2(ax_i^3 + bx_i^2 + cx_i + d - y_i).$$

From mandatory conditions of function extremum as $S = S(a, b, c, d)$ we are getting system of equations for coefficient definition of cubic dependency:

$$\begin{cases} a\sum_{i=1}^{n} x_i^6 + b\sum_{i=1}^{n} x_i^5 + c\sum_{i=1}^{n} x_i^4 + d\sum_{i=1}^{n} x_i^3 = \sum_{i=1}^{n} x_i^3 y_i, \\ a\sum_{i=1}^{n} x_i^5 + b\sum_{i=1}^{n} x_i^4 + c\sum_{i=1}^{n} x_i^3 + d\sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i^2 y_i, \\ a\sum_{i=1}^{n} x_i^4 + b\sum_{i=1}^{n} x_i^3 + c\sum_{i=1}^{n} x_i^2 + d\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} x_i y_i, \\ a\sum_{i=1}^{n} x_i^3 + b\sum_{i=1}^{n} x_i^2 + c\sum_{i=1}^{n} x_i + dn = \sum_{i=1}^{n} y_i. \end{cases}$$

It is possible to prove by increasing number of sidechain amount of master nodes and nodes are increasing as well. Proportional increase of sidechains is directly affected by growing of network load, which can be proven by derivation of first rank for function of 2 variables with input of full differential, where function is $z = f(x; y)$.

Because x and y are separate variables, one of them can be changed while the other is keeping its value. Separate variable x takes increase of Δx which keeps y unchanged. Thus z receives increase of it's value, which called local increase of z by x and is defined as $\Delta_x z$.

So,

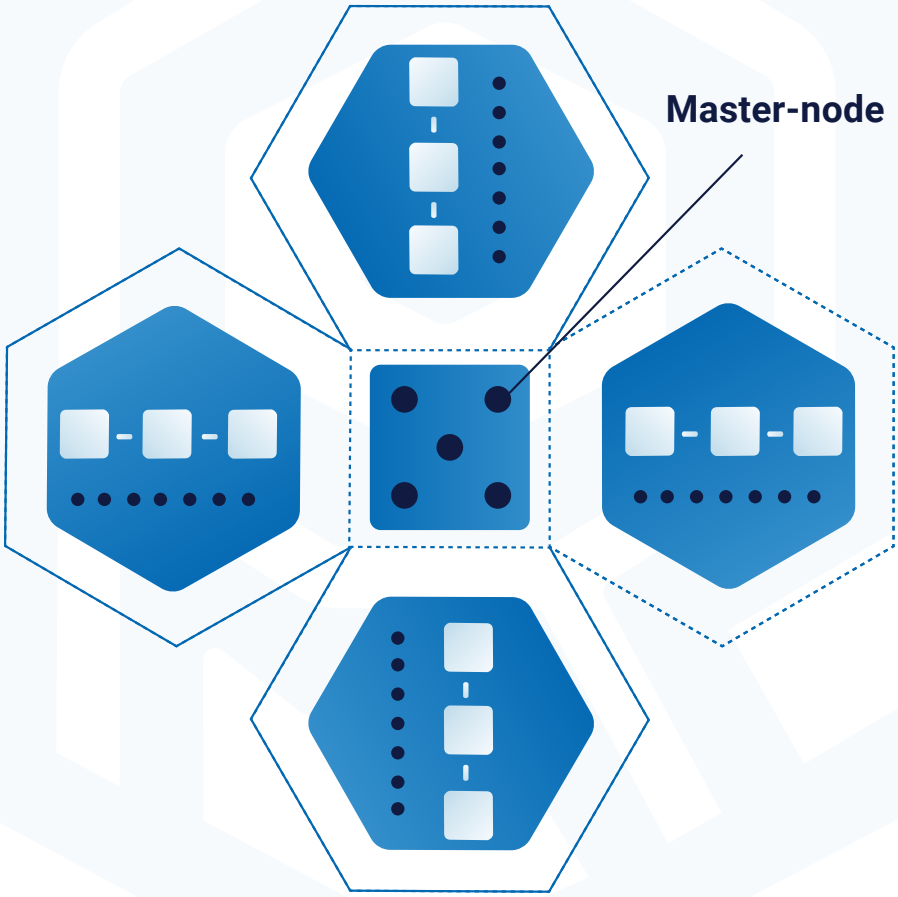
$$\Delta_x z = f(x + \Delta x; y) - f(x; y)$$

Getting local increase z by y :

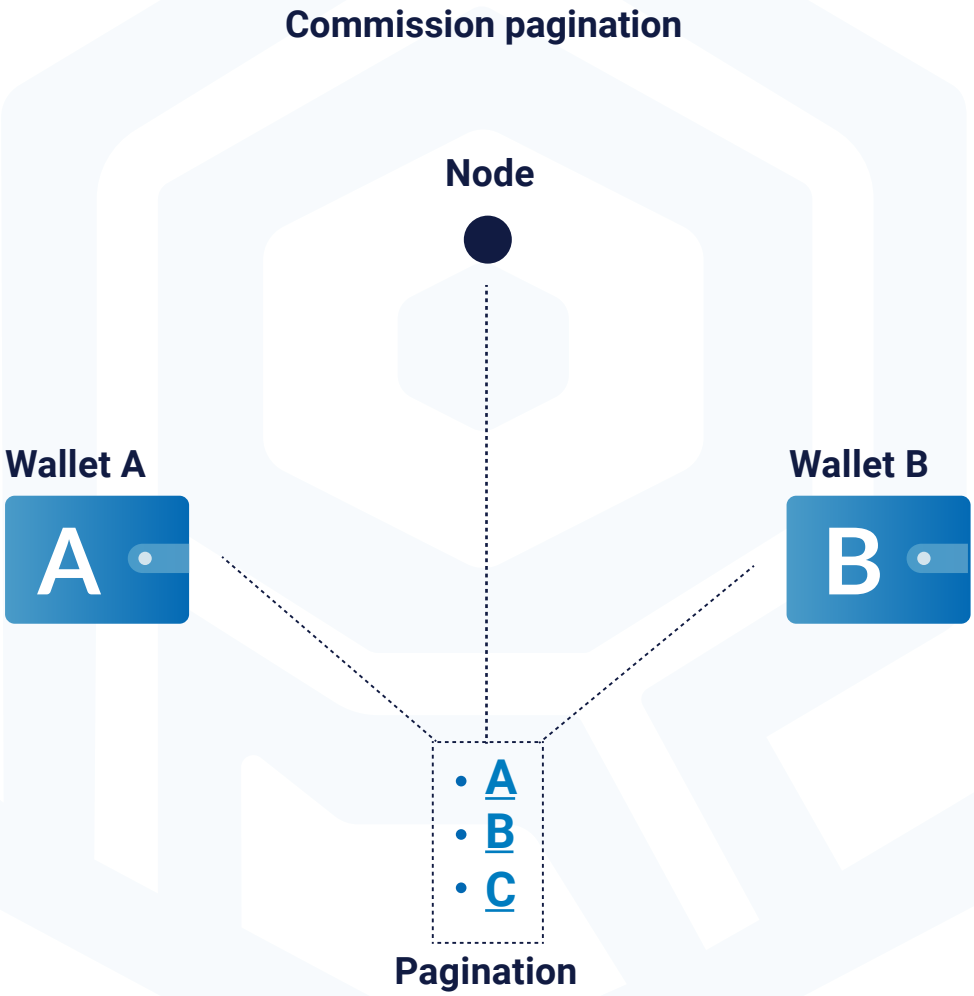
$$\Delta_y z = f(x; y + \Delta y) - f(x; y)$$

Republia blockchain can process number of transactions which are necessary to serve user requests and in the same time keeping network throughput on the same level. Master-node distributes nodes for the blockchains which must be processed.

Master-nodes distribute nodes



Initially transactions are going to pool based on commission pagination they are coming to node. After that request will be sent to master-node which provides permission or rejection to transaction processing. In order to guarantee transaction security and that address holds enough funds, node sends request to master-nodes, which perform arbitrage based on consensus. Current step prevents cheating in the network. After master-node provides positive response, node pushes transaction to the network.



REPUBLIA

TYPES OF NODES IN REPUBLIA BLOCKCHAIN

TYPES OF NODES IN REPUBLIA BLOCKCHAIN

Node (network node) is a mechanism, on which the full client of the network and register storage are installed (in some situations partial storage, for example, as in the case with ordinary nodes). However, network nodes are connected to a common system and take direct part in voting rounds for choosing the processing node and trusted nodes, that confirm or reject transactions, that are stored in the register.

There are three types of nodes in Republia Blockchain:

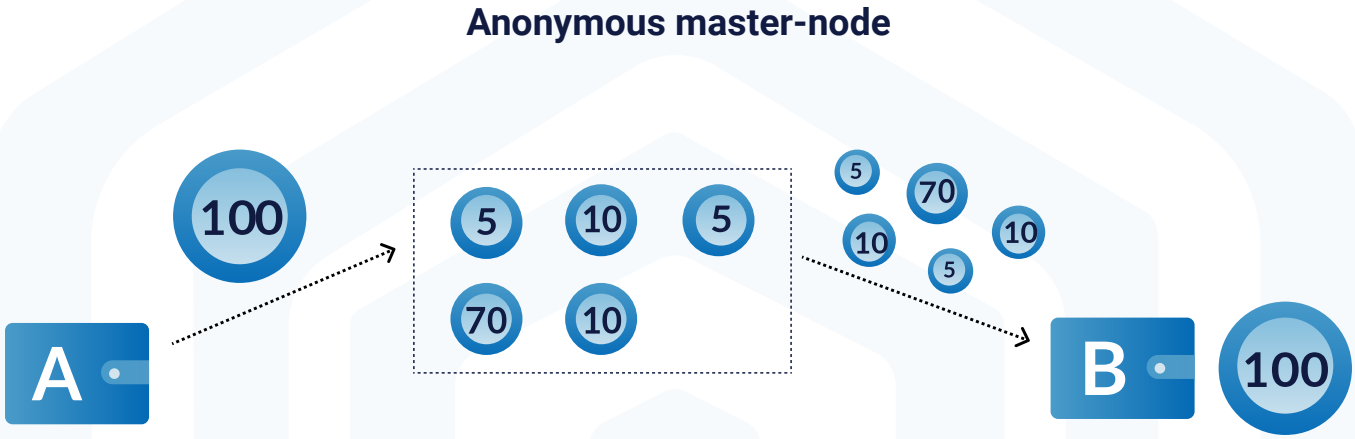
1. Full Node is completely synchronized with blockchain Republia wallet, whose tasks are to record transaction in blockchain, check the transaction pool, accept transactions in the pool according to the commission pagination, transfer data about the received transaction and the address of transaction to master node, accept decisions (approve or disapprove) from master node.

However, the node must store the entire history of blocks in that sidechain in which it operates.

The role of the holder of the node can be performed by anyone with a **rating of more than 40.**

2. Master node is a server in a decentralized network, that acts as a holder of all the blocks, accepts requests for transferring data and addresses of the transaction received from nodes. Master node is also responsible for searching for the last transaction in Republia Blockchain using address, making (approve or disapprove) decisions, sending requests for consensus between master nodes, reaching a consensus between master nodes using transactions, transferring the decision on the ongoing transaction node.

3. Anonymous master node is responsible for receiving transactions with a special label “anonymous”, divides one transaction into several unequal parts and sends them to the recipient address. This transaction is characterized by high level of anonymity, because it is transferred not by the total amount, but by several parts, so it is impossible to track from which address it was made.



Deceptions and frauds by nodes and master nodes are avoided in Republia network by introducing a tool called **“ZERO.ing”**.

To provide participation of nodes and master nodes in the recording of blocks and receive rewards for recording, a certain amount of RPB coins is frozen on their wallets and in case of fraud from their side, this amount is charged from the account of the attacker, it is called **“ZERO.ing”**.

Thus, network participants will not be willing to harm the ecosystem, when their personal funds are at risk.

The system subdivides all computers (nodes) into three categories:

Parameter	Ordinary node	Trusted node (master-node)
Exchange direction	**1.** With all ordinary nodes of peer-to- peer network **2.** With master-nodes	**1.** With the main node of the current round for the exchange of lists **2.** Receiving register updates from peer-to-peer network

REPUBLIA

ALGORITHM OF REPUBLIA CONSENSUS

ALGORITHM OF REPUBLIA CONSENSUS

Consensus in Republia is a method of group decision making, whose purpose is to find final solutions, that are acceptable for all nodes of the network. Consensus Republia operates on an improved algorithm Republia-Delegated-Proof-of-Authority (RdPoA) + BFT (Byzantine Fault Tolerance). Algorithm RdPoA operates on the basis of RepubliaID verification and helps to avoid attacks, because users own unique RepubliaID, which ensures the reliability of each ecosystem participant and the confidence that one participant within the framework of voting for modernization of both technological and ideological decisions will be able to vote only once.

BFT is an algorithm, in which each network node checks each transaction in pools. If one transaction matches at least two pools, it is placed in result vector. If there are no matches, then the corresponding element is marked as “unknown”.

For example, four network nodes were analyzing one element, and one of the nodes did not find any matches and marked it as “unknown”. Consequently, most of network nodes agreed the transaction - consensus is reached.

Thus, all elements of the transaction pool are checked by nodes, with the help of this process agreement is reached to make a transaction.

Republia Blockchain also involves directed acyclic graphs, that use a topological classification, although the development of this graph can take place only in vector direction - from the initial blocks to later ones. Due to this fact idea of parallel chains is achieved, while the blocks in Republia Blockchain remain important, which, in turn, ensures simultaneous execution of several transactions on different chains, without reducing the capacity of the entire network.

REPUBLIA

MINING RPB IN REPUBLIA BLOCKCHAIN

MINING RPB IN REPUBLIA BLOCKCHAIN

Mining RPB in Republia blockchain occurs according to consensus algorithm, where records to distributed register are added only by those nodes, that are accepted by the system according to a certain specification for checking RepubliaID and the rating system.

In Republia blockchain mining is available to each participant, since the task of mining RPB and recording to the block is fulfilled by ordinary nodes. Since Republia blockchain operates on RPoA algorithm and provides user with similar effect on the ecosystem, one community member will not be able to take control of the entire network, even if he has a lot of RPB coins he will not be able to mine blocks alone.

Any device of an ecosystem participant can become a node, but in order to obtain privileges of master node, it is necessary to freeze certain number of RPB coins in the wallet and reach a certain level according to the rating system. This step is an insurance of the community in case of misconduct of the participant, the device of which is a server in decentralized Republia network.

The mining process occurs in Republia in turn, which guarantees transparent mining: nodes by turn take transactions from the pools and transfer data to master nodes, after positive response of master node, ordinary node must record transaction to block, and then a reward is obtained.

Due to that fact that nodes mine blocks in turn, transparent mining process is guaranteed in Republia Blockchain, as each participant will be rewarded for his help in expanding the network.

Moreover, mining is available on any client.

REPUBLIA

VOTING SYSTEM

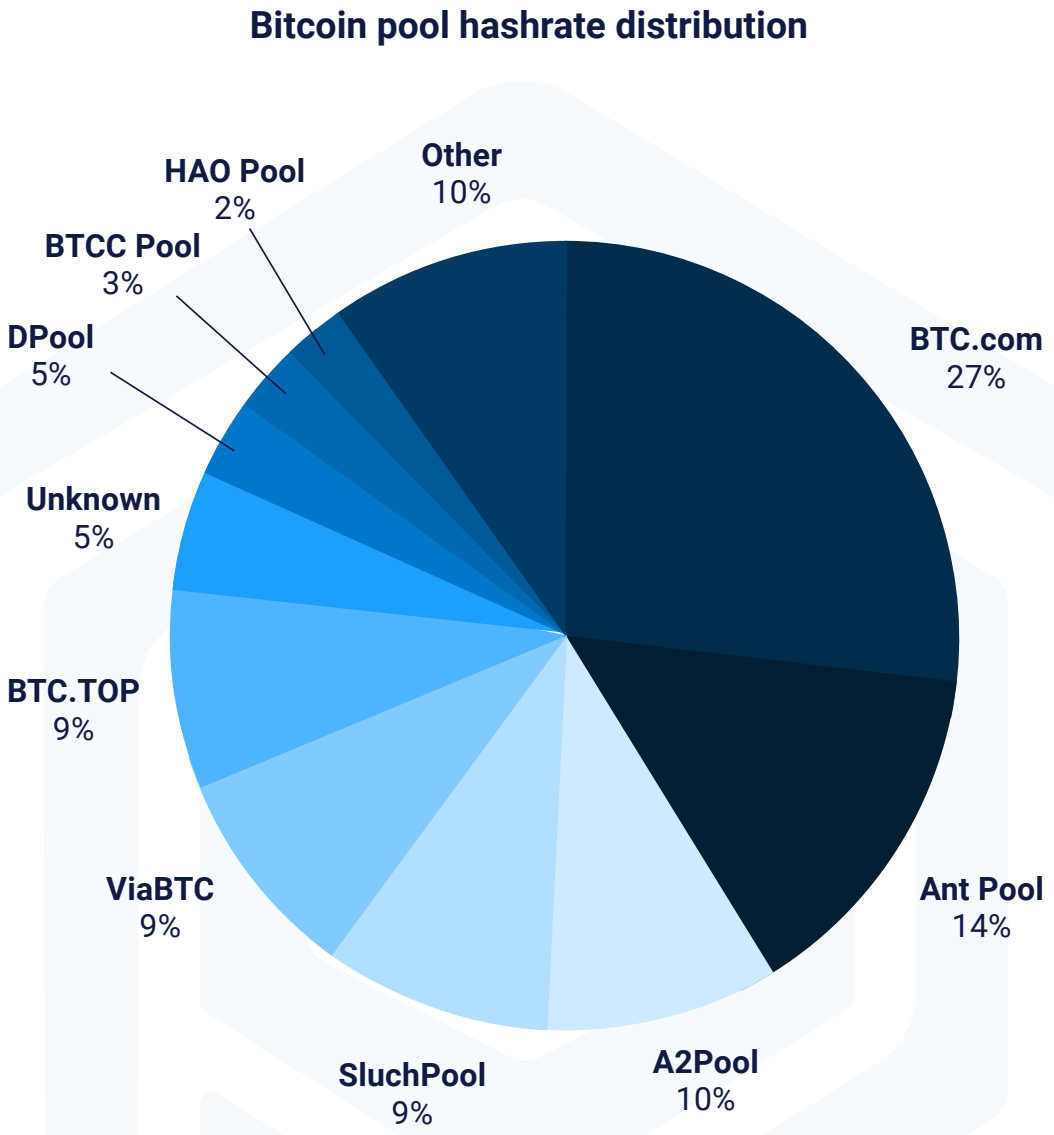
THE INFLUENCE OF USERS ON THE NETWORK (VOTING SYSTEM)

Misunderstandings and difficulties always occur in blockchain, therefore participants are often divided into several groups, one of them stands for updating protocol, second - to leave the protocol in the original form. For example, one of the disadvantages of ERC-20 standard is that it allows many projects to run ICO very easily, which is confirmed by huge number of tokens(on 1st of May 2018, there are more than 80,000 of tokens).

Thus, industry fills with tokens, that are similar to each other, and investors are at a loss in their choice. Considering this, Republia follows the principle of “We-ecosystem”, where all users are of equal importance in the modernization of the network, so consensus Republia operates on algorithm Proof-of-Authority (PoA) + BFT (Byzantine Fault Tolerance).

In Republia Ecosystem users decide by themselves whether an ICO project should be located within the whole ecosystem.

In addition to the above, voting on algorithms PoW, PoS or dPoS often occur in the industry, where users have almost no influence on the network.



Unequal influence of users can be clearly seen on the example of Bitcoin network, where the main role is played by pools. The largest one, BTC.com, owns 27% of all capacities involved in the network. While an ordinary user does not have even the slightest influence on the modernization of Bitcoin network.

In Republia Blockchain voting takes place through a unique RebubliaID, where one RebubliaID is equal to one vote, in such a way each ecosystem participant votes only once within one voting and can be confident in the transparency and openness of the system.

However, Republia can veto any voting during first two years of ecosystem's operation. This step takes into account possible system instability and excludes the emergence of critical and incorrect voting. At the end of the network testing period, Republia system will operate autonomously.

Republia solved the problem of hard forks by updating the protocol automatically, which occurs after voting among ecosystem members. Participants do not need to set up conversions, since the polls are held in a private account in a special tab.

REPUBLIA

AVOIDING VULNERABILITIES IN CODE

AVOIDING VULNERABILITIES IN CODE

Since Republia VM is implemented using functional programming language Michelson, which supports formal verification of code, the network does not allow vulnerabilities in the code, as it happened with The DAO project.

The thing is that ERC-20 tokens almost represent smart contracts, which received a general recognition for undoubted efficiency. However, its risks are not taken into account, for example, the smart contract cannot be changed after it is created, it may contain errors and vulnerabilities, that could lead to the loss of funds.

Hacking The DAO in 2016 led to the split of Ethereum network and the creation of Ethereum Classic.

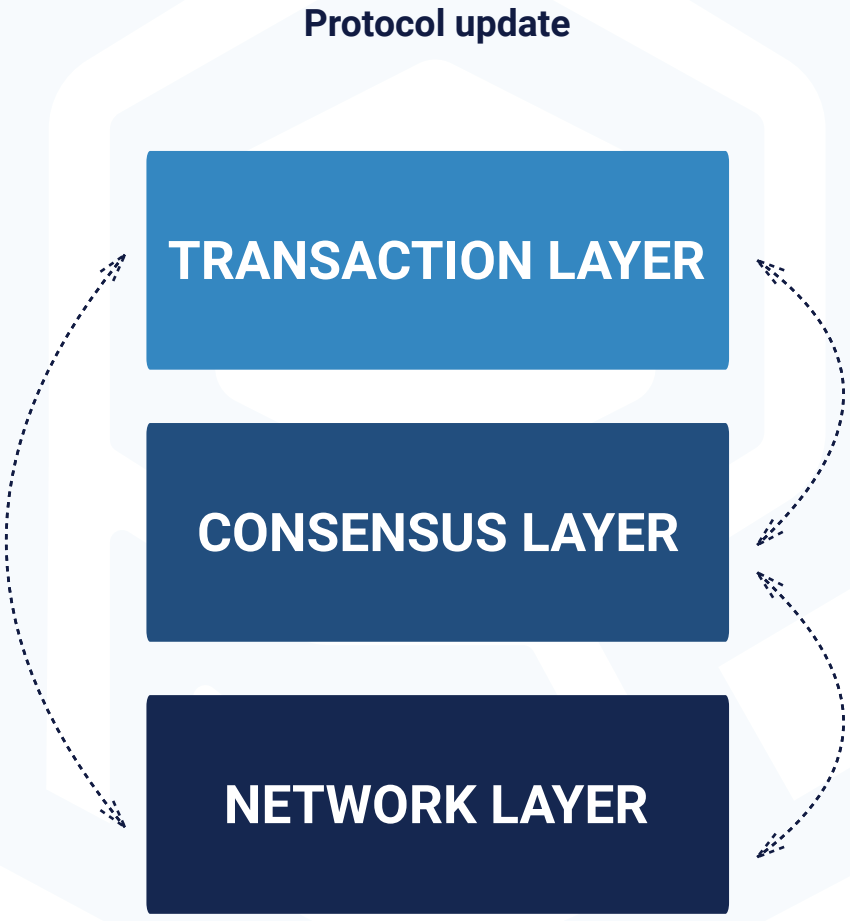
The DAO problem is that EVM code is implemented using statically typed JavaScript - similar to Solidity programming language. Republia VM, in turn, using Michelson language, reached the process which does not allow to execute smart contract when an error is detected.

REPUBLIA

ONCHAIN PROTOCOL UPDATE

ONCHAIN PROTOCOL UPDATE

In turn, Republia Blockchain protocol is updated automatically after the decision is made by users within the framework of voting. Onchain updates allow to avoid hard forks, because all users influence the modernization of the network by themselves, and participants do not need to set up additional conversions, all structural mechanisms are updated automatically.



REPUBLIA

NEW REPUBLIA VIRTUAL MACHINE

NEW REPUBLIA VIRTUAL MACHINE

For now many blockchain projects use in their systems EVM - Ethereum Virtual Machine, but it has a number of vulnerabilities.

A striking example of EVM disadvantage is hacking TheDAO.

Considering this, Republia uses a whole new Virtual Machine (VM) operating on the stack functional programing language - Michelson.

Michelson supports the paradigms of object-oriented and imperative programming and was used for implementing Republia VM by supporting formal verification of code, which allows developers to verify the integrity of the code of their smart contracts. In the Ethereum poor-quality smart contracts formal verification of code would allow to detect those errors, that could have been avoided.

In fact, Republia VM allows to check the contract for the purpose of its qualitative execution.

REPUBLIA

REPUBLIA SMART CONTRACT PLATFORM

REPUBLIA SMART CONTRACT PLATFORM. TURING COMPLETENESS

Republia Smart Contract Platform is Turing complete platform, where each smart contract operates accurately, taking into account embedded algorithm. Turing completeness excludes fraud, delays or malfunctions.

The entire system operates autonomously.

Due to this technology, in Republia Smart Contract Platform thouthands of digital contracts can operate simultaneously.

The correct conception of Republia Smart Contract Platform, that supports return of funds, that do not reach the required address, excludes the problem of erroneous sending tokens to the smart contract of another ICO and, as the result, loss of funds, if the smart contract itself does not provide such possibility.

According to statistics, as of the end of 2017, in such a way more than 3 million dollars were lost.

10.1 Smart contract type

Republia Smart Contract Platform uses structured accounts, that define execution code, thereby becoming smart contracts.

Moreover, each contract specifies a hash of the public key, which is used to signe and mine blocks in the protocol.

Formally, the contract is presented as:

```
type contract = {
counter: int;
user: reppass;
balance: Int64.t;
signer: id option;
code: opcode list;
storage: data list;
spendable: bool;
delegatable: bool;
}
```

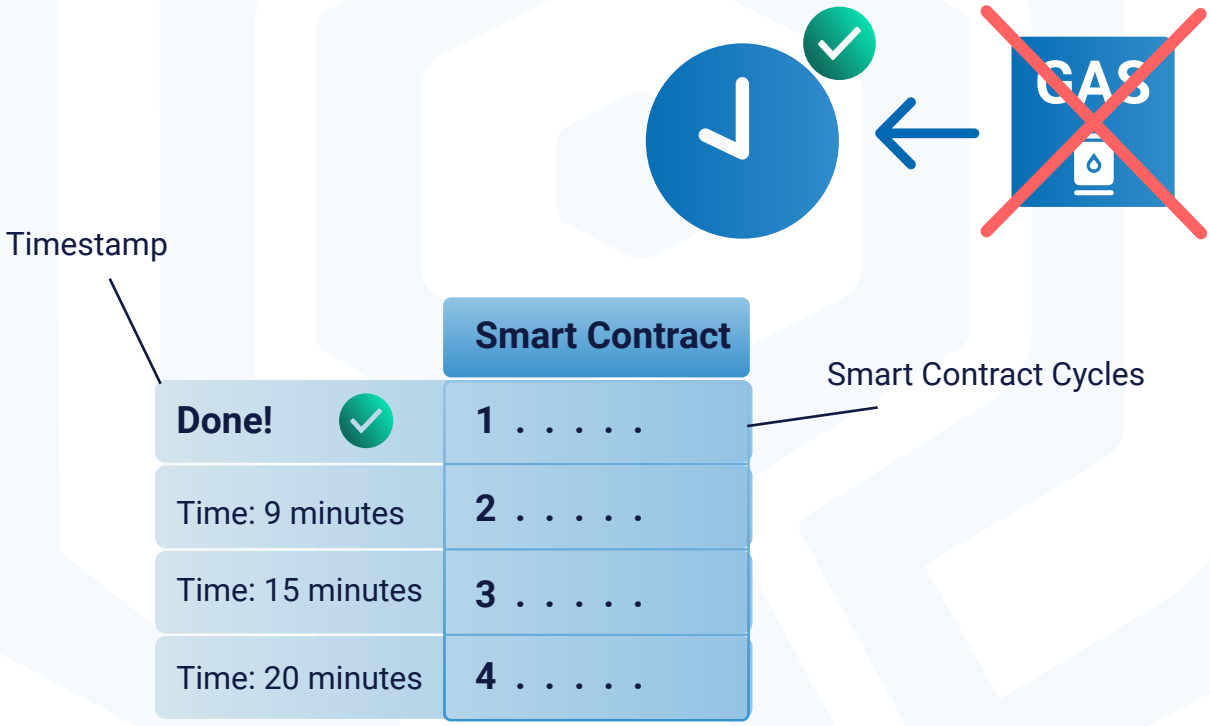
REPUBLIA

ABSENCE OF GAS

ABSENCE OF GAS

Smart contracts in Republia Smart Contract Platform operate on the principle of timestamp, without using GAS, which guarantees exact execution of the transaction and avoidance of situation when smart contract goes to an endless cycle.

The timestamp was chosen for the convenience of using and binding the smart contract cycles to the time.



REPUBLIA

ISOLATION OF PROBLEMATIC APPLICATIONS

ISOLATION OF PROBLEMATIC APPLICATIONS

Republia Blockchain eliminates network malfunctions caused by problematic applications by isolation.

A striking example of such a malfunction is the ICO of Status, which was conducted on Ethereum. According to representatives of Status project, many users ignored recommended limit of GAS, which led to malfunction in the network. In addition to the above, as it turned out later, application became problematic due to GAS restrictions set by Status itself.

Republia Blockchain implements isolation of problematic applications creating a shell over them and limiting them in this way from other processes within the ecosystem.

REPUBLIA

REPUBLIA RATING SYSTEM

REPUBLIA RATING SYSTEM

In Republia ecosystem Rating system operates, which aims to exclude malevolent users, which try to harm, discriminate the system or prevent other users from making decisions in Republia life.

Republia Rating system is designed according to a mathematical function, which is used and implemented in the following way:

$$b_i = F(Y_i) = F(Y_{i1}, \ldots, Y_{in}) =$$

$$= a_{i1} X_1 + a_{i2} X_2 + \ldots + a_{in} X_n =$$

$$= \sum_{i=1}^{n} a_{ij} X_j, i = 1, \ldots, m$$

Where a_{ij} – rating, exhibited by the participant according to the j characteristics, X_j – weight coefficient of j characteristics.

When registering in the ecosystem, each user is provided with basic rating, which is equal to 45 points.

This figure is a starting point for everybody and can vary upward or downward, depending on the participant actions inside the ecosystem.

Participants with high rating are offered additional bonuses of the ecosystem, including the use of Parking tool, opportunity to trade on exchanges with a lower commision, visiting free meetups, organized by Republia Foundation, opportunity to become node or master node.

Rating distribution in Republia Rating system

Points	Opportunities within the ecosystem						
	Parking of funds	Partici-pation in voting	Trading on stock exchan-ges	Low commiss ion while trading	Mining (become a node)	Become a master node	Visiting events free of charge
0-10	⛔	⛔	⛔	⛔	⛔	⛔	⛔
10-20	✅	⛔	⛔	⛔	⛔	⛔	⛔
20-30	✅	⛔	✅	⛔	⛔	⛔	⛔
30-40	✅	✅	✅	⛔	⛔	⛔	⛔
40-50	✅	✅	✅	⛔	✅	⛔	⛔
50-70	✅	✅	✅	✅	✅	⛔	✅
70-100	✅	✅	✅	✅	✅	✅	✅

After registration and obtaining 45 starting points, user is provided with an opportunity to increase his rating by 0.00789 points for each of the block mined and recorded correctly to Republia Blockchain.

After using Parking tool, user's rating also increases, in accordance with bonus system.

Bonus system for Parking of funds

№	Parking type	Bonuses to the rating
1	day	0,5 points
2	week	5 points
3	month	20 points
4	3 months	30 points
5	half-year	50 points
6	year	70 points

Actions that entail a decrease in rating:

1. For the attempt of Ddos attack the user's rating loses 50 points

2. For the attempt of spam attack the user's rating loses 50 points.

3. If it is not checked by spam content bot, rating loses 40 points.

4. For other inappropriate actions, that are aimed at compromising the system or preventing other users from fruitful collaboration inside the ecosystem, rating loses 10 points

Actions that are aimed at increasing the rating:

1. The user has a chance to be excluded from the list of those who did not pass the verification by spam content bot, in this case rating will increase by 20 points.
Rating system is designed in such a way, that users, who have lost a certain number of points, will be able to restore them after a while, every week user will get 5 points, if there are no cases of misconduct from the participant side within the ecosystem. In addition to the above, if there are no cases of rating decrease, additional points will not be accrued.

2. Since Republia Rating System is completely autonomous, errors in calculation of rating or falsified results are impossible.

If user does not agree with his rating, he can initiate a special automatic revision of coefficients with the help of a request.

REPUBLIA

FIGHT AGAINST THREATS AND ATTACKS ON THE NETWORK

FIGHT AGAINST POSSIBLE THREATS AND ATTACKS ON THE NETWORK

One of the important focuses in Republia is to provide users with a level of security, that can protect the system from all kinds of attacks.

To avoid attacks, “Checking” tool operates, which means reaching the height, that cannot be recorded again.

If a spam attack of a user who makes a large number of transactions is detected, the commission will increase with each payment, in the future the algorithm will completely restrict this user from sending transactions.

In Republia Blockchain participants are protected from Eclipse attack, because master nodes, when making transactions in the network, by consensus, should make sure the transaction amount is correct. Thus, it is confirmed whether the address has an amount for the transaction.

In open source networks Sybil attacks are possible, but open source Republia Blockchain avoids this problem by introducing a unique RepubliaID. This step eliminates the problem of creating fake accounts and protects the network from attacks.

Since Republia operates using the algorithm Proof-of-Authority (PoA) + BFT (Byzantine Fault Tolerance) the network is protected from such a threat as an attack of 51%, because the modernization of the subsystems is influenced equally by all users, and not by those with more capacities at their disposal. When voting each participant uses his unique RepubliaID, which guarantees only one vote from one user within a voting.

Elimination of other known threats in Republia ecosystem

№	Threat	Бонусы к рейтингу
1.	Attack of intermediaries	Static key usage for user identification via RepubliaID
2.	Time-attack	Usage of algorithm Curve25519
3.	Hack of public-private key pair	Key generation using elliptic curves Curve25519 algorithm
4.	Attack using compromised nodes	Usage of the node verification algorithm as well as ranking approach we used in the list. Primary function of node verification algorithm is to verify file which is stored on local storage
5.	Compromised node participation during transaction arbitraje using byzantine fault tolerance algorithm	Usage of blake2s algorithm which generates hash in order to prevent appearance of nodes to transaction arbitraje stage which can't validate its ability to store registry. In case compromised node will try to participate in arbitraje Zero.Ing will happen. Funds of hash on users ledger which tries to compromise Republia blockchain will be blocked
6.	Attempt to hack ledger/account	Excludes with the encryption of RepubliaPass and Veracity System working each second
7.	Double spending	Minimizing the likehood of the attack, as the planned time for the transaction pool formation is> 0.1 seconds
8.	Centralized processing	At the certain moment of time the node can't be master or trusted twice

REPUBLIA

REPUBLIAID & VERACITY SYSTEM

REPUBLIA ID & VERACITY SYSTEM

RepubliaID is a type of URI, digital analogue of Identity Card, which is created by each entity in Republia ecosystem.

The creation algorithm guarantees the exclusion of two identical RepubliaIDs (≈ 0.0000000019834). Moreover, each RepubliaID passes a two-level verification for uniqueness, when registering a user in Republia ecosystem, double consent node compares and analyzes the uniqueness of RepubliaID.

Thus, RepubliaID minimizes risks of loss of funds or hacking of user accounts.

Protection of RepubliaID is guaranteed by innovative security system using artificial intelligence - Veracity System.

RepubliaID и Veracity System are implemented into all components of the global ecosystem, where it is possible to harm other members of Republia community. The participant with RepubliaID becomes a full-fledged user of Republia community and has an opportunity to influence life of the community directly.

Veracity System also generates a unique digital key, which is bound to ID, ecosystem participant can independently manage the areas of use of his own RepubliaID, for this reason secret key is bound, and user can control his requests in private account.

In order to prevent falsifications in the voting stage, each vote in the system is signed with a unique digital key, which was generated by Veracity System.

Elimination of other known threats in the ecosystem Republia

To generate the keys, RepubliaID uses international standard for digital signature algorithm, such as RSA.

Strong cryptography based on computational complexity of reverse function to crypto-function:

$$c = E(m) = m^e \mod n.$$

To calculate m by unknown c, e, n you need to find d, to

$$e * d \equiv 1 \quad (mod \varphi(n)),$$

that is

$$d \equiv e^{-1} \quad (mod \varphi(n)).$$

REPUBLIA

ACCOUNT SECURITY

ACCOUNT SECURITY

Each user account in Republia ecosystem can operate in three modes:

- owner mode;
- active usage mode;
- recovery mode.

Owner's approval is configured by multi-signature N of M (2 of 2), with the help of which the user has the right to change all other approvals immediately. The owner approval update is configured for a 30 day delay.

For additional security the owner will be able to use active approval of the recovery partner. This will create trust network between users, which can only be compromised, if all the participants are hacked at the same time.

If active key of the recovery partner is compromised, the partner has an option to use the owner's approval to recover. In case of partner disagreement about recovery, active permission always can update owner permission with 30 days delay. This makes account independent from third party decisions.

Backup system with cloned keys helps to reduce impact in case when user lost his active key and intruder get access to it. But current use case considered to be a critical scenario which highly unlikely to happen in well designed security system - Veracity System.

16.1 Password recovery

Republia allows user to recover access to his/her data in very short term. Since ecosystem is configured to develop trustworthy relationship where every user can assign few recovery partners which can change current permissions of the account (with 7 days delay), in case the user was offline for 30 days.

We recommend to assign friends, relatives or trusted people who can help to recover user account and avoid permanent block of the account.

Every user has assigned constitutional and juridical mandates to recover ownership rights of the account. This step enables to make a claim at a court in case your recovery partner abused rights. Information about ecosystem participants stored securely and recovery of lost password will not take much time by excluding all possible bureaucratic problems.

REPUBLIA

CONCLUSION

CONCLUSION

Technical paper of Republia project provides review of technologies used in Republia Blockchain. Considering evolution and modernization of industry technology processes. Republia team will be constantly committed to following all trends of blockchain community.

Moreover, Republia team is already working on special Wiki to help users to understand technologies of Republia ecosystem.

Republia initiates training course for learning Republia Blockchain and OCaml programing language. We are looking forward to collaborating with developers and waiting for enthusiastic people who are ready to join our team for future development and improvement of Republia technology.

REPUBLIA

LEADING-EDGE ECOSYSTEM AND TECHNOLOGY