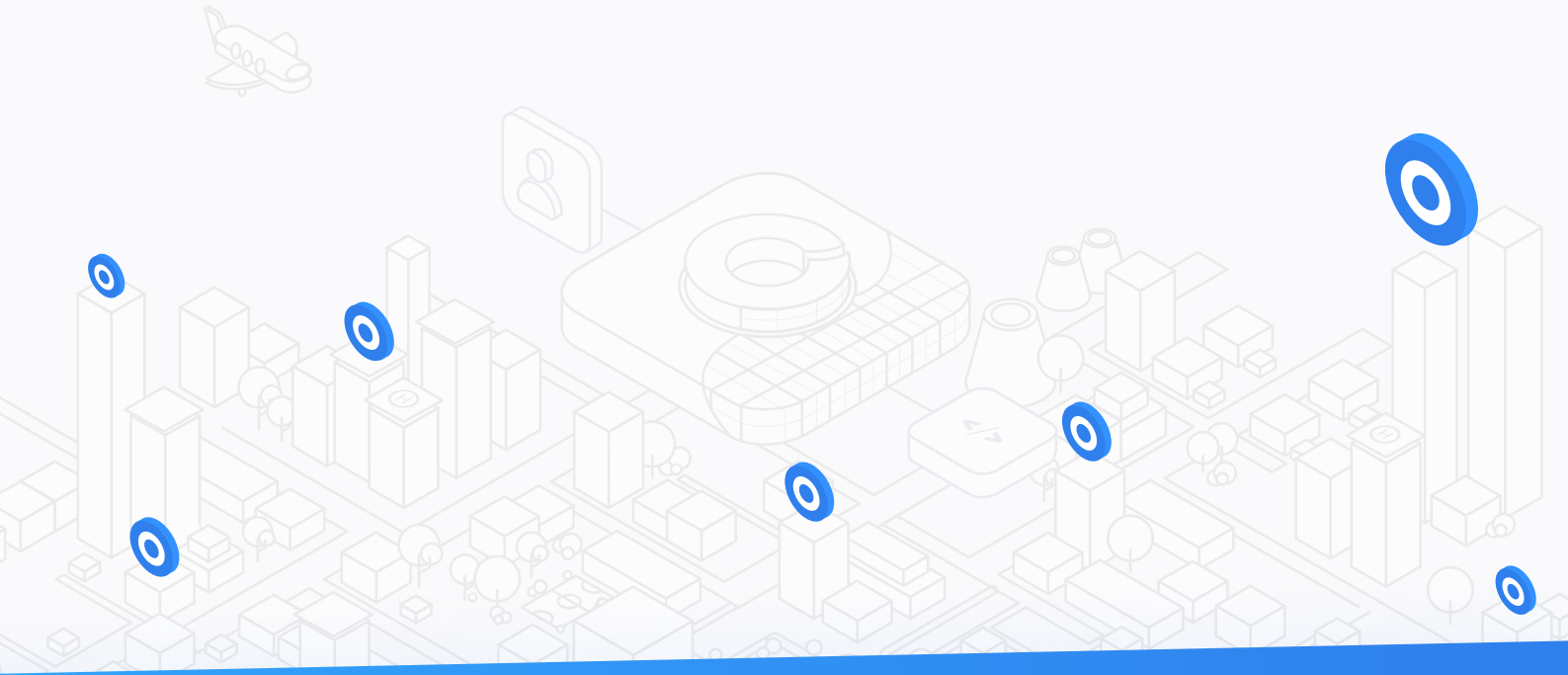
OPEN Chain



WHITE PAPER

CONTENT

INTRODUCTION

At present time, blockchain technology is changing the way of software development for business transactions. Almost all spheres of industries are looking for opportunities to implement this technology to realize their new business ideas or increase efficiency using all the advantages of the blockchain. There is a large number of various blockchain solutions, but each business idea must be approached individually, taking into account all its requirements. This white paper is about new blockchain platform - OPEN Chain which aims is to offer a wide range of solutions for various tasks and implement all existing effective experiments for blockchain technologies and use all benefits of JVM. There are multiple SDKs for integrating with blockchain. OPEN Chain also allows creating Smart contracts on all JVM languages conveniently.

EXISTING CONCEPTS

There are many ways to use blockchain technology for your project. You can build public or private blockchain with your individual requirements, hard or soft fork existing blockchains with your updates, use Smart contracts to create decentralized applications (DApps) or own cryptocurrency.

BLOCKCHAINS

There are many ways to use blockchain technology for your project. You can build public or private blockchain with your individual requirements, hard or soft fork existing blockchains with your updates, use Smart contracts to create decentralized applications (DApps) or own cryptocurrency.

FORKS

Forking every Blockchain is different, based on the design architecture and the chain use case. This occurs when developers want to change the rules which the software uses to perform transactions, validating or even consensus. Changes and updates (typically in consensus algorithms) in the way which makes previous versions of the software incompatible are called “hard forks”. “Soft forks” are software updates that still work with older versions.

SMART CONTRACTS

There are multiple blockchains which offer to implement your business idea through Smart contracts. They also provide APIs and various tools for conveniently Smart contracts creation and management. With these type of contracts, you can build DApps or own cryptocurrency. Smart contracts make it possible to exclude a third party from the agreement of other parties. Contract transactions are credible, trackable and irreversible.

CONSENSUS

Consensus rules are a specific set of rules which nodes on the network will ensure the block follows when validating that block and the transactions within it. The key requirement to achieve a consensus is an unanimous acceptance between nodes on the network for a single data value, even in the event of some of the nodes failing or being unreliable in any way.

PROOF OF WORK (POW)

Any member of the system, with some probability, can become a block builder at any point of time and thus participate in the security maintenance of the blockchain and be rewarded for this work. This probability is proportional to the computational power of the participant.

The possible reward is the stimulus for investing in hardware and electricity (to prove workability). At the same time, these expenses make impractical to act maliciously for an unscrupulous member. Basically, the members are voting for the updating of the blockchain with the next block with their computation power, i. e. "Proof-of-work is essentially one-CPU-one-vote".

PROOF OF STAKE (POS)

Due to the amount of computational power required, PoW is costly and energy expensive. Proof of stake is an alternative approach where the probability to be rewarded for the blockchain maintenance is proportional to the participant's stake. A maliciously acting participant, in order to get a reward, will need to possess over 50% of the total stake which means that such participant will act against core self-interest.

Basically, the members are voting for the updating of the blockchain with the next block with their current stake, i.e. "Proof-of-stake is essentially one-stake-unite-one-vote".

DELEGATED PROOF OF STACK (DPOS)

Delegated Proof of Stake (DPoS) is the fastest, most efficient, most decentralized, and most flexible consensus model available. DPoS leverages the power of stakeholder approval voting to resolve consensus issues in a fair and democratic way. All network parameters, from fee schedules to block intervals and transaction sizes, can be tuned via elected delegates. Deterministic selection of block producers allows transactions to be confirmed in an average of just 1 second. Perhaps most importantly, the consensus protocol is designed to protect all participants against unwanted regulatory interference.

BYZANTINE FAULT TOLERANCE (BFT)

When it comes to the distributed systems, Byzantine Fault Tolerance (BFT) refers to the ability of a distributed computer network to function as desired, and correctly reach an accurate consensus, even when some of the nodes in the system are failing or propagating incorrect information to other peers, either accidentally or maliciously.

OPEN Chain

The intent of the OPEN chain is to create a convenient and flexible technology for developing blockchain solutions and decentralized applications.

OPEN chain strives to divide its logic for separate modules. Each module of the OPEN chain can be substituted with another implementation satisfying individual requirements. That approach provides wide opportunities for using the OPEN chain in a variety of projects.

You can use any JVM language to extend or modify OPEN chain. JVM languages have a wide community.

Fast and efficient consensus protocol requires significantly less energy than PoW. It provides faster processing of transactions than PoW and PoS. Also participants don't need costly specialized equipment. A regular computer is powerful enough.

OPEN chain allows you to create cryptocurrency or distributed ledger, or any blockchain based solution depending on your needs.

ACCOUNTS

Accounts are crucial in OPEN Chain. There are three types of accounts: Wallet Account, Node Account and Smart Contract Account. Wallet Accounts will be referred to simply as wallets, Node Account will be referred to as nodes and Smart Contract Accounts will be referred to as contracts. This division is justified in that all of these entities are the so-called state objects. Wallets state is a balance, Nodes have a role and votes, as for Contracts it is a balance and a contract code. The state of all accounts is the state of the OPEN network which is updated with every block. encrypted, and it is encrypted with the password you enter when you create the wallet. The keystore file is stored on a user side.

WALLET

Wallets are essential for users to interact with the OPEN Chain via transactions. Wallets represent identities of external agents (e.g., human personas). Wallets use public key cryptography to sign transactions so that the OPEN Chain can securely validate the identity of a transaction sender.

KEYSTORE FILE

Every Wallet is defined by the pair of keys: a private key and public key. Wallets are indexed by their address which is derived from the hashed public key by taking the first 20 bytes. Every private key/address pair is encoded in a keystore file. The keystore file is JSON text file which you can open and view in any text editor. The critical component of the keystore file, your Wallet's private key, is always encrypted, and it is encrypted with the password you enter when you create the wallet. The keystore file is stored on a user side.

NODE

A Node represents an OPEN chain client running instance. A person/organization can control a number of nodes and the node can be controlled by a number of persons/organizations. Node indexed by their address which is derived from the hashed public key by taking the first 20 bytes.

OPEN chain defines three roles of nodes: Peer, Delegate and Active Delegate. A role is determined by a node's degree of participation in the consensus reaching.

A Peer is a node with minimal influence on reaching consensus. Peer provides transactions creation only.

A Delegate is a Peer bonded with specific Wallet. Each Peer can be bonded with only one Wallet. But a wallet can be bonded with a lot of Peers. Wallets can vote for delegates or recall their vote. Each Delegate has a rating which is calculated as a sum of weights of votes for it.

Active delegates are the top N Delegates by rating. N is the number of active delegates. The number of delegates in OPEN chain is equal to 21. This was done to maximize efficiency in the network, allowing the average block time to hover between 4 and 6 seconds.

Active Delegates validate transactions, run smart contracts as well as produce, validate and commit blocks. They are paid for block producing. Active Delegates are the most responsible for achieving consensus.

Initial Active Delegates list is defined in the first genesis block which is hardcoded.

SMART CONTRACT

A contract is a collection of code (its functions) and data (its state) which resides at a specific address on the OPEN chain.

Contracts are stored on the blockchain in a Java bytecode. But the state of smart contract is stored apart in the binary representation. It allows skipping contract execution on nodes which are not Active Delegates. It aims to improve efficiency.

Code of smart contract is loaded and executed in Java Virtual Machine (JVM). Using JVM has allowed writing contracts in any JVM languages: Java, Clojure, Groovy, Kotlin, Scala etc. The number of classes and methods allowed to be used in smart contracts is bounded. It is allowed to use primitives, arrays, ArrayList, HashSet, HashMap and StrictMath.

The process of executing contract code is the part of the block validation process, so if a transaction is added into block B the code execution spawned by that transaction will be executed by all Active Delegates which validate block B. The rest of nodes will apply the only state of contract.

TRANSACTIONS

A transaction is the signed data package which stores a message to be sent.

Transactions contain:

- **Header**
  - Timestamp
  - Fee
  - Sender address
- **Payload** - depends on transaction type
- **Footer**
  - Hash
  - Signature
  - Public key

There are four transaction types: Delegate, Vote, Transfer and Reward. They differ by a payload.

- **Delegate Transaction Payload**
  - Delegate address
  - Amount
- **Vote Transaction Payload**
  - Delegate address
  - Vote type
- **Transfer Transaction Payload**
  - Wallet address
  - Amount
- **Reward Transaction Payload**
  - Wallet address
  - Reward

Each payload is handled on its own way and pursuits specific aims. Delegate payload serves to bond sender address from header and delegate address from payload. In other words - delegate announcement. A sender should pay a fixed amount for this as a guarantee. Using vote payload users can vote for delegate or withdraw their vote. Transfer payload is the most spread payload. It allows transferring coins from one wallet to another wallet or smart contract. Reward payload provides the opportunity for active delegate to be rewarded for block producing. Only one Reward transaction signed by block producer is allowed per block.

RECEIPTS

As a result of transaction executing a receipt is created. Receipt refers to a transaction and contains transaction status and all changes which it produces. Receipts are stored in an appropriate block. Receipt contains:

- Transaction hash
- Transaction status
- Set of transaction results

Transaction result contains:

- Sender address
- Receiver address
- Amount
- Data

Transaction result is similar to a transaction except for signature. Simple transfer transaction has two results. The first is the coin transfer between sender and receiver. The second one is fee transfer between sender and block producer.

STATES

Transactions execution provoke state transitions. State transitions are based on receipts. For each account, only one final state can be generated per block. States are stored in an appropriate block. So, the last state of account is the most actual and confirmed.

Wallet state contains:

- Balance state
- Delegate address for whom he voted for

Node state contains:

- Node rating

Wallet state contains:

- Smart contract balance
- Smart contract code

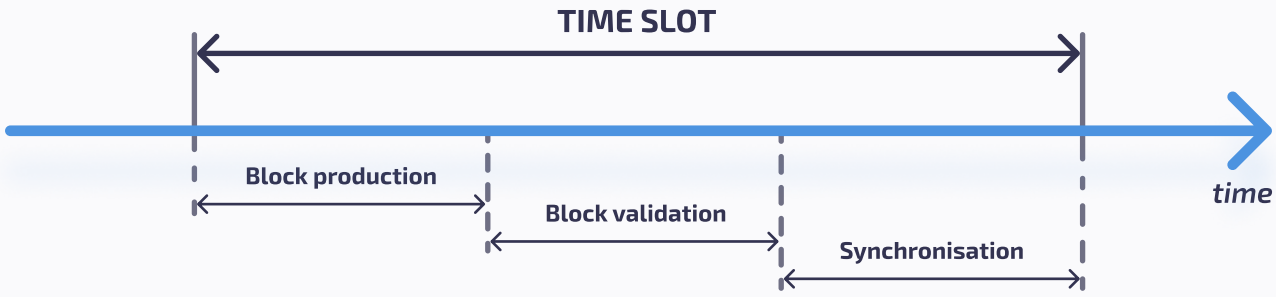
CONSENSUS ALGORITHM

OPEN Chain was implemented by cherry-picking the best aspects of DPoS and BFT. OPEN Chain Consensus took the ability from DPoS for stakeholders to have an influence in the system proportional to their stake in the system. BFT allows avoiding chain forks and provides transaction finality without the need for confirmations.

Every stakeholder can take a part in Delegate election. Every participant has one vote. They can vote for a Delegate or recall their vote. Participants form with their votes the list of Active Delegates. By voting results, top N of Delegates become Active Delegates. N is a number of active delegates. It is determined in accordance with the requirements of security and efficiency. At the current moment in OPEN chain N is set to 21.

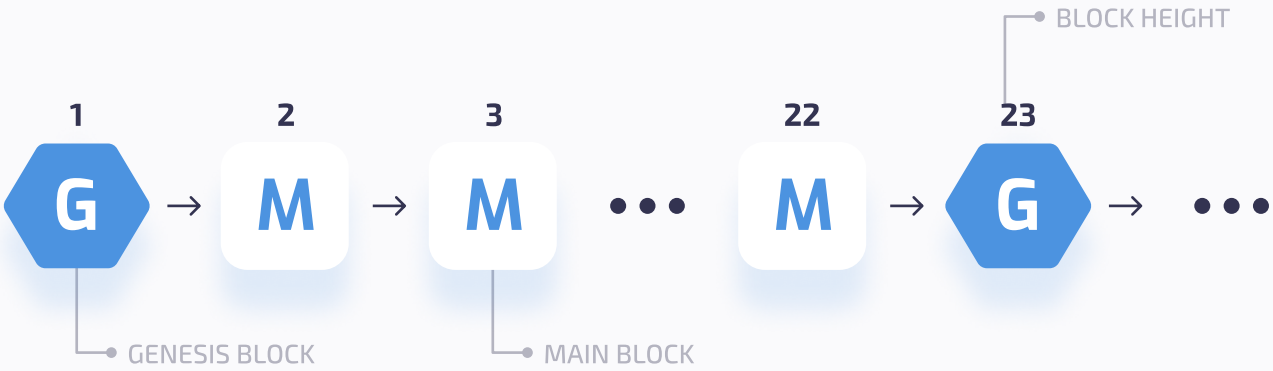
Active delegates are elected for one epoch. Epoch continues until N blocks are produced. That approach gives each active delegate opportunity to produce a block. Voters can immediately detect malicious actions and the malicious delegate can be voted out of the system.

Each Active Delegate should produce a block in the given Time Slot. The rest of the Active Delegates should validate and vote for or against this block. In case the number of votes for is more than $(\frac{2}{3}N+1)$ the block is added to the chain. After that, the network is given time to synchronize.



BLOCKCHAIN

Despite OPEN chain similarity with other blockchains, there are some essential differences. One of the main difference with regard to the blockchain architecture is that OPEN blocks contain a copy of transaction list, receipt list and the most recent state. Also the OPEN chain contains two types of blocks: Genesis and Main.



GENESIS BLOCK

The chain is labeled by Genesis blocks through all length. Main blocks are located between Genesis blocks. Each Genesis block announces the new Epoch and sets network parameters for this epoch. At current moment network parameters include only the active delegates list.
Genesis block doesn't contain transactions and is produced by each peer themselves. The identity of Genesis blocks on the same high among a whole network is a sign of consensus mechanism stability.

Genesis block contains:

- Timestamp
- High
- Hash
- Previous block hash
- Epoch index
- List of active delegates
- Signature
- Public key

MAIN BLOCK

Main blocks contain a copy of the transaction list, the receipt list, and the recent states. The approach is inefficient because it needs to store the entire receipts and states with each block but it allows to store all the aspects and nuances of the blockchain work. Storing will be improved in the nearest releases.

The main block contains:

- **Timestamp**
- **High**
- **Hash**
- **Previous block hash**
- **Transactions**
  - Transfer transactions
  - Reward transactions
  - Delegate transactions
  - Vote transactions
- **States**
  - Delegate states
  - Account states
- **Receipts**
- **Transactions Merkle Root**
- **States Merkle Root**
- **Receipts Merkle Root**
- **Signature**
- **Public key**

Block validation algorithm depends on the role of a node. The basic block validation process and each node participation are described in the table below.

Check	Peer	Delegate	Active Delegate
Check if the previous block referenced exists and is valid	✓	✓	✓
Check that the block is produced by the correct active delegate	✓	✓	✓
Check that the timestamp of the block is in the correct time slot	✓	✓	✓
Recalculate each transaction hash		✓	✓
Check each transaction sign		✓	✓
Smart Contract execution			✓
Check transactions merkle root	✓	✓	✓
Reproduce each receipt	✓	✓	✓
Check receipts merkle root	✓	✓	✓
Reproduce each state			✓
Check states merkle root	✓	✓	✓
Recalculate block hash		✓	✓
Check block signature	✓	✓	✓