# EXTRACHAIN TECHNOLOGY

## Advanced PoS Blockchain and DAO Framework

Overview of technical details, possibilities and functions of ExtraChain Blockchain technology

# Contents

Terms and Theory ..... 2
Directed acyclic graph ..... 2
Blockchain ..... 2
Merkle tree ..... 2
Actor ..... 2
Proof-of-Existence ..... 3
Genesis Block ..... 3
Fast-Blockchain mode ..... 3
Full-Blockchain mode ..... 3
General description ..... 4
ExtraChain (ExC) ..... 4
ExConsensus ..... 6
Transaction Verification Algorithm (TVA) ..... 6
Block Prove Algorithm (BPA) ..... 6
Block Merge Algorithm (BMA) ..... 7
State Snapshot Algorithm (SSA) aka “Genesis Algorithm” ..... 9
Additional features ..... 10
DAG features ..... 10
PoS features ..... 10
Token Module ..... 11
ExtraChain indicators ..... 12
Distributed File System ..... 13
Distributed File System (ExDFS) ..... 13

# Terms and Theory

Directed acyclic graph – finite directed graph with no directed cycles. DAG is a directed graph that has a topological ordering, a sequence of the vertices such that every edge is directed from earlier to later in the sequence. DAGs allow for multiple chains of blocks to co-exist and interconnect while never forming an edge with a parent node. Nodes can exist in parallel, as long as information is directed in the same way. There are no blocks in DAG-based chains, each node is transaction.

Blockchain – organized directed database, bound by cryptography and Merkle tree data structure. By design, a blockchain is resistant to modification of the data. It is "an open, distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way". For use as a distributed ledger, a blockchain is typically managed by a peer-to-peer network collectively adhering to a protocol for inter-node communication and validating new blocks. Once recorded, the data in any given block cannot be altered retroactively without alteration of all subsequent blocks, which requires consensus of the network majority. Although blockchain records are not unalterable, blockchains may be considered secure by design and exemplify a distributed computing system with high Byzantine fault tolerance. Decentralized consensus has therefore been claimed with a blockchain.

Merkle tree – a tree in which every leaf node is labelled with the cryptographic hash of a data block, and every non-leaf node is labelled with the cryptographic hash of the labels of its child nodes. Hash trees allow efficient and secure verification of the contents of large data structures.

Actor – an acting unit of network, that:

1. Has “public-private” key pair;
2. Is verified by other network actors;
3. Can send and receive transactions;
4. Can load data into network;

Actor entities examples: user, smart contract, bot etc.

Proof-of-Existence – a concept aimed to verify existence of real world object or digital unit via digital algorithms or systems (like blockchain). This concept is used to:

- secure the ownership of real-life and digital entities;
- protect property rights;
- prove the existence of selected entities;
- allow fast and decentralized verification of all named above

Genesis Block – a blockchain unit that holds state of blockchain. Bitcoin- and Ethereum-like blockchains have Genesis Blocks as initial blocks of chain, holding initial addresses and their balances.

Fast-Blockchain mode – ExC operation mode, when device downloads only part of blockchain from current block to last Genesis Block. This mode is made for devices with hard limit of memory and users, who want to start network operation faster.

Full-Blockchain mode – ExC operation mode, when device downloads full blockchain. This mode requires more memory, but grants possibility to make faster checks during all prove algorithms.

# General description

ExtraChain (ExC) is a lightweight blockchain system providing fast and easy blockchain interactions for portable and non-portable devices.

ExtraChain was built with main idea in architecture: connect as many platforms as possible and implement full decentralization of decisionmaking in network.

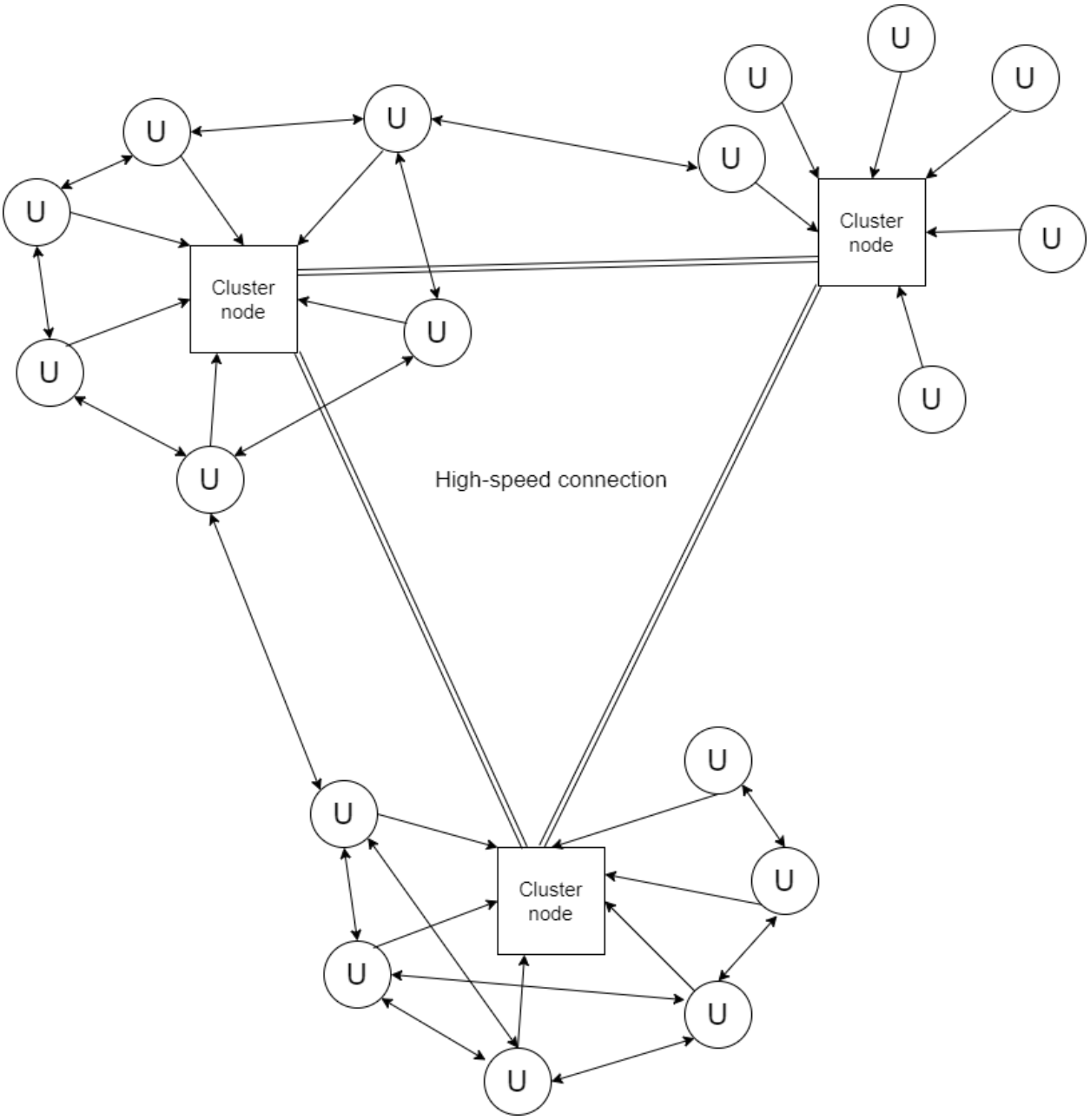
ExC nodes are sharing resources between each other to empower low performance devices, such as smartphones.

ExtraChain nodes form data-exchange clusters to speed up local data transactions and requests and create faster connections between network parts.

ExtraChain node types:

- User node – ordinary user. This node can be any device (mobile device, PC, low-performance server). It stores only required part of chain and uses data from other nodes;
- Cluster node – high-performance node with high-speed connection to other clusters. It stores all data of blockchain and provides it for all nodes;

As it shown at cluster scheme, decisionmaking is protected by different connections between clusters. This structure type allows fast data transfer and reliable data exchange for all types of devices.




ExtraChain Cluster structure

# ExConsensus

ExC consensus is based on block data checks and aimed to form consistent and stable block chain between two genesis blocks.

Each produced transaction transfers to network neighbors (“verifiers”) and goes through verification process made by them.

## Transaction Verification Algorithm (TVA):

1. Proof-of-Existence of sender and receiver
   1.1. Check if address exists in local Actor DB (should be done for sender and receiver);
   1.2. Check digital signature of transaction (done for sender) via EdDSA algorithm;
   1.3. Check sender status (user or smart contract);

   If at least one step fails, transaction will be denied.
2. Proof-of-Existence of entities:
   2.1. Sender and receiver balances (in blockchain);
   2.2. Edited directories and files (in ExDFS);
3. Proof-of-Allowance:
   3.1. Sender and receiver balances are >= 0 after transaction (in blockchain);
   3.2. Edited or created file is in ownership of sender or sender is marked as editor for the file (in ExDFS);

If all proofs are true, transaction is verified and put into block.

Each verifier put its ID and signature in the block.

Block is distributed by verifiers via network.

Network neighbors of verifiers check created blocks and approve their consistence. They become “approvers”.

## Block Prove Algorithm (BPA):

1. Check digital signature of block via EdDSA algorithm;
2. Check hash link to previous block;
   2.1. If hash link points to local last block – proceed;

2.2. If hash link points to other block – stop Block Prove and start Block Merge;
3. If block with same number is already in local blockchain, stop Block Prove and start Block Merge;
4. If received block is new, check each transaction in block via TVA;
5. If TVA check is successful – add approver signature and distribute approved block

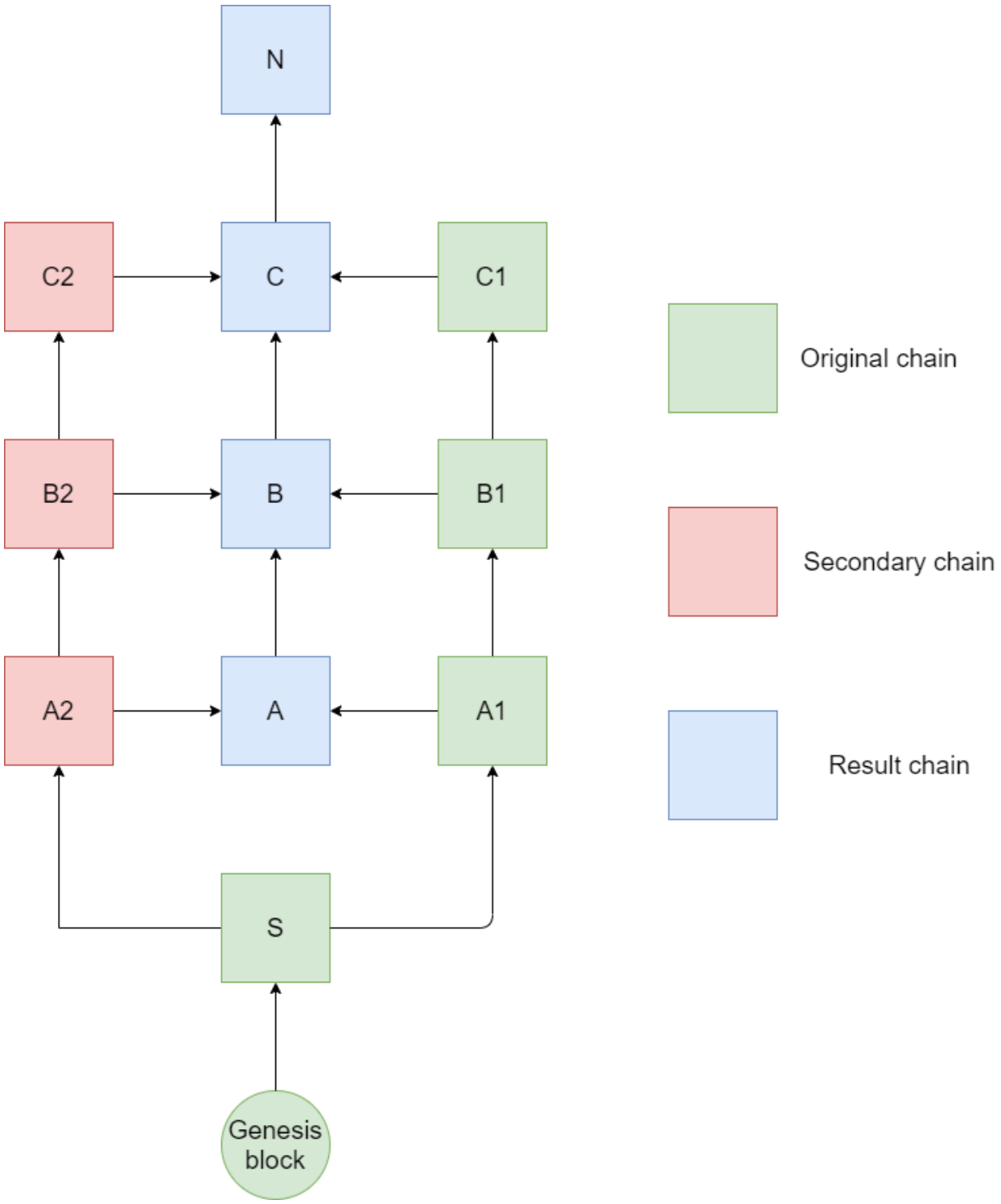
Block needs to be approved by at least 2 approvers to become “mature”.

## Block Merge Algorithm (BMA):

If BMA is started, then one or more blocks in local blockchain can be corrupted or incomplete. Also, new block can have malicious data. Thus, approver must make deep scan of block data.

1. Compare received and old block data.
   1.1. If data hash is different – compare block data and detect new transactions.
   1.2. If there are no new transactions:
      1.2.1. And previous block hash is same – drop received block.
      1.2.2. And previous block hash is different – stop BMA and start BPA;
2. Check detected transactions with TVA. If transactions pass TVA, new block with added transactions is produced and distributed.

New block can trigger chain of BMA calls to make consistent chain for each network unit.




Block Merge scheme

## State Snapshot Algorithm (SSA) aka “Genesis Algorithm”:

State Snapshot Algorithm produces Genesis Blocks each 100 blocks. Genesis Blocks hold Actors’ balances data and serves as state backup and foothold for devices in “Fast Blockchain” mode.

Genesis block production algorithm:

1. Find last Genesis Block and load IDs and balances;
2. Collect all new balances and make all changes from last Genesis Block to current block;
3. Merge collected data with loaded data;
4. Distribute new Genesis Block;

If device receives new Genesis Block, then BPA starts, but only for previous and new Genesis Block.

# Additional features

ExConsensus also includes DAG and Proof-of-Stake features.

## DAG features:

- Possible existence of branched chains;
- Multiple genesis blocks;
- Each block and transaction can have more than one connection (token creation and transactions, genesis blocks);
- Non-limited block size;

These features allow faster search and smaller amount of blocks needed to interact with blockchain. ExC requires 100 blocks and genesis block (blockchain snapshot) to securely interact with main chain.

## PoS features:

- Cashback – up to 80% of transaction fee can be returned to tx sender, if other nodes prove that only 2 nodes from required 10 made similar block check and proved its coherence.
- Rewards – PoS rewards are paid from each proved transaction. Users can stake their coins and receive rewards based on formula:

$$R = 0.1 * Fee * \left(100 * \frac{CoinsStaked}{TotalSupply}\right) * StakingModifier$$

*StakingModifier* is a control coefficient and equals to 0.5

- Staking scheme applies to main coin and all tokens (if this feature is enabled by creator);

# Token Module

Tokens are created and used similar to other cryptocurrency token platforms (such as Graphene).

Token features include all ExC base coin features as blockchain snapshots, branched chains, fast search and PoS rewards. This features can be enabled by token owner.

Main difference is that fee from token transactions is charged in tokens. This improvement helps to distribute token faster and popularize it. User will have an opportunity to exchange all tokens at distributed exchange integrated in ExC platform.

# ExtraChain indicators

- Transaction bandwidth: 2321 – 3242 tx/sec
- Token creation time: 1 sec
- Block creation time: each 2 sec
- Fee: from 0,2% to 1% (see “Cashback”)
- Non-personalized transactions: yes
- Personalized transactions: yes

# Distributed File System

Distributed File System (ExDFS) is shared data storage and serves as an additional support module for ExtraChain DAO Framework.

Distributed File System can be used in various ways. It can:

- Hold user data of DAO-project
- Distribute files between users
- Authorize access to files
- Differentiate access rights for users

One of main ExDFS elements is Historical Chain. Historical chain is a data structure that holds all changes made in particular file, including creation and removal.

Historical chains are used to:

- Contain data changes
- Organize data changes
- Order changes in a non-time-based way
- Secure data changes (changes are non-removable)
- Proof of Existence (file can be retrieved on each state of change)

All these possibilities create stable and protected distributed file storage that can hold any amount of files without corrupted links, order and state errors.

ExDFS is protected by cryptographic methods of ExtraChain and allows to build secured and anonymized decentralized apps, without any need in localized data storage.

GitHub Repo: https://github.com/ExtraChain-Foundation