EXTRACHAIN TECHNOLOGY

Advanced PoS Blockchain and DAO Framework

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

Contents

General description

ExtraChain (ExC) is a lightweight blockchain infrastructure and decentralized storage “ExDFS” allowing creation of high-load dApps (decentralized applications) for both portable, non-portable and IoT devices.

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

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

ExtraChain includes:

- Decentralized data storage system “ExDFS” (see p. 6);
- Consensus, that combines advantages of Proof-of-Stake and Directed Acyclic Graph architectures (see p. 11);
- Token module (see p. 21);

ExtraChain main features:

- Unlimited, distributed and fast data storage for tokenized distributed apps;
- Distributed architecture (see Appendix A) with high scalability (nodes are sharing their resources to increase productivity and new nodes and users create relatively small computing pressure on entire network);
- All possibilities of base currency are applied to tokens. Tokens inherit all native functions of base coin as Staking and Token Fee. Token creator can enable them at creation stage (see p. 21);
- ExtraToken creation requires only icon choice (estimated token creation time – 3 minutes);
- Fast and reliable infrastructure (see p. 22);

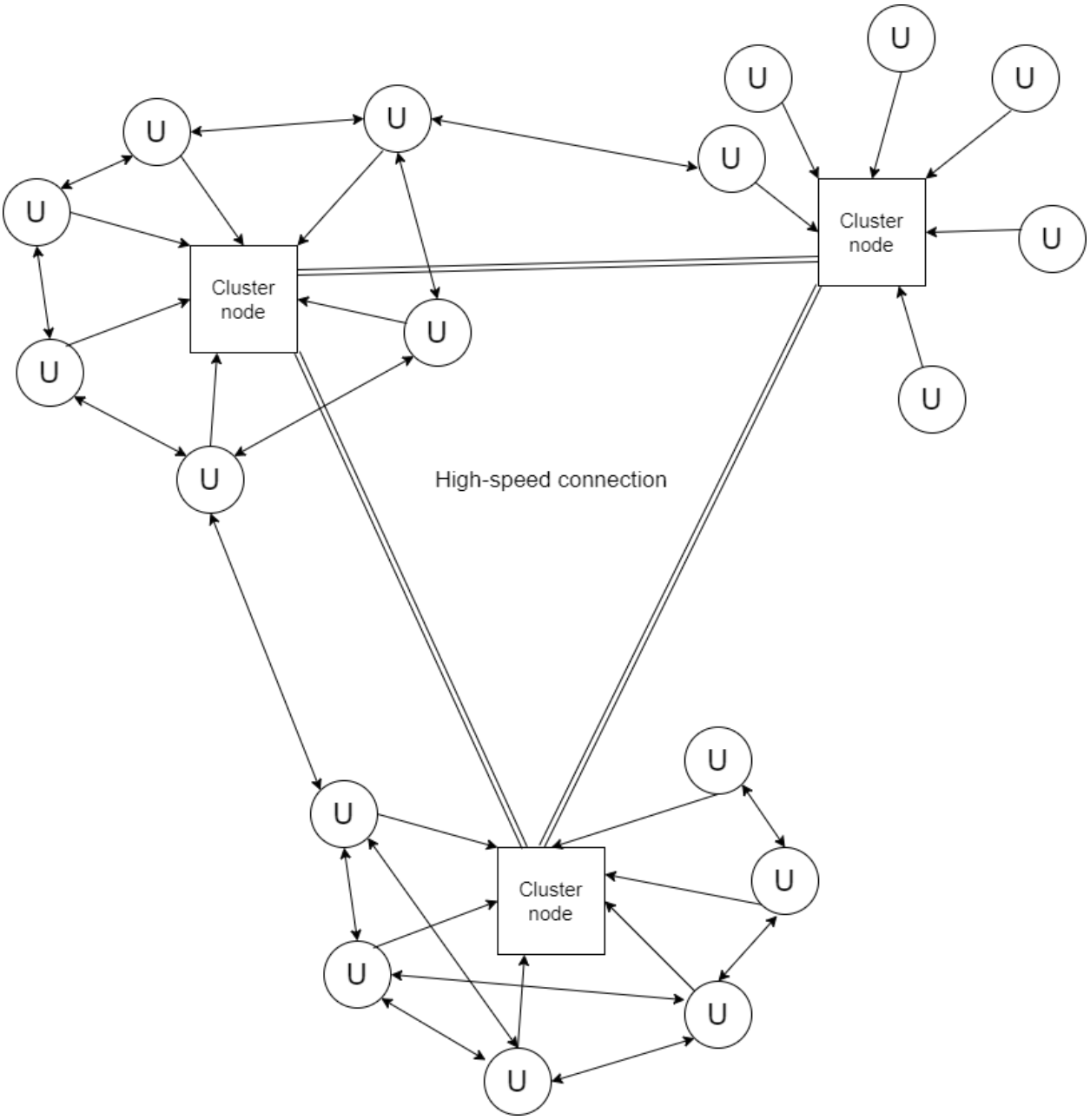
ExtraChain node types

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.



Cluster network scheme

Distributed File System

Distributed File System (ExDFS) is shared data storage and serves as an main 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.

ExDFS Applications

ExDFS can be applied in various fields:

- Internal document control and exchange;
- Healthcare Industry (data storage for sensitive high-priority files, such as medical cards, prescriptions etc.);
- Human and citizen authorization/verification storage (digital passports, permits, licenses etc.);
- Proof-of-Ownership for digital and real-life assets (digital items secured by blockchain, art dealership, protected history of ownership for real-life and digital objects);
- Freight transport industry (data storage for files, that require state logging and access control, such as transport cards, routes, check logs etc.);
- Information exchange networks (messengers, decentralized file storages, cloud computing data storages etc.);

Data Exchange Automation in ExAPI allows different scenarios:

- Status update logging via data pipeline at History Chain. This scenario can be applied in freight transport industry to improve freight transport control and state control of transported cargo;
- Secured and anonymous sensitive data transfer. Healthcare industry can improve security and anonymity of patients via this scenario and speed up data exchange between different medical facilities;
- Fast human verification via personal secured ExDFS pipeline to user's directory. This scenario allows fast and reliable human authorization and is vital for universal digital ID card projects, now implemented in different countries;
- Automated auctions for real-life and digital objects, where participants have direct access to assets and all information about them (history of ownership, licenses, certificates etc.). This scenario improves e-commerce operations by securing the history of ownership and implementing non-fungible reflection of real-life and digital objects, that contains all required data for verification and proof operations;

These functions give ExDFS possibility to be a heart of complex and simple systems of data flow, control and exchange in different fields of business.

Decentralized Social Network is one of the possible usages of ExDFS that demonstrates all main features of ExDFS and blockchain-based interactions.

Blockchain features:

- Usage of public-key authorisation allows high-level of privacy;
- Blockchain verification ensures that each user is responsible for their own actions;
- Interactions secured by public key (ECC) cryptography give highest possible level of protection;

Basic interaction between users is message and content exchange. All messages are grouped in chats, and chats exist in ExDFS as files.

File-based approach presents different features:

- Access rules for chats:
    - read-only, read/write, read/write/edit;
    - for all users or for selected users;
- Historical Chain organises messages and ensures, that each message will be delivered and placed at right position;
- User profiles built as Historical Chain-based files give control over change history;
- Historical Chain allows user delete message and restore it, if they want;
- Each chat is file and is backed up in ExtraChain network;

All features above make messaging consistent and no messages will be “not delivered” or lost.

Message exchange also uses ExC securing mechanism. Asymmetric encryption protects user privacy and ensures that each participant is authentic. Also, it is possible to create big group chats with shared session keys, where each user join or removal activates new session key generation. This algorithm prevents uncontrollable rejoin and allows users to read chat data that was previously accessible for the user. Furthermore, it is possible to ban one or multiple users via key regeneration and re-encryption of data.

Unlike existing implementations of blockchain messengers and social networks, we are offering other, file- and Historical Chain-based approach - Distributed Storage Messenger and Social Network. Let's compare existing and ExDFS-based solutions.

Feature	Blockchain Messenger	Distributed Storage Messenger
Fast account creation	YES	YES
No personal data required (phone number, e-mail, phone book)	NO **(some products have integration with personal data)**	YES
IP address not exposed	YES	YES
Strong encryption (ECIES algorithm protection)	NO **(some products have weaker encryption algorithms)**	YES
MiTM attack not possible (ECDSA protection)	YES	YES
Message order is unchangeable	YES	YES
Messages cannot be completely removed from data storage	YES	YES
Consensus message integrity check	YES	YES
Messages cannot be censored	YES	YES
Full access from any devices if user has keys	YES	YES
All data is stored in distributed network (you have no need to store data locally)	YES	YES
File changelog (blockchain-based)	NO	YES
Messages can be deleted for user, but stored in history of chat	NO	YES
History is accessible only for chat participants	NO	YES
Blockchain is not overflowed by messages transactions	NO	YES

(this approach speeds up message distribution for users)		
Zero-proof for messages	**NO**	**YES**

As we can see, Distributed Storage Messenger offers more secured, controlled by participants and reliable approach for decentralized data exchange systems and networks.

ExConsensus

ExC consensus is an advanced consensus algorithm aimed to connect and unite Proof-of-Stake ideology with high speed, integrity and portability of Directed Acyclic Graph data structure.

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.

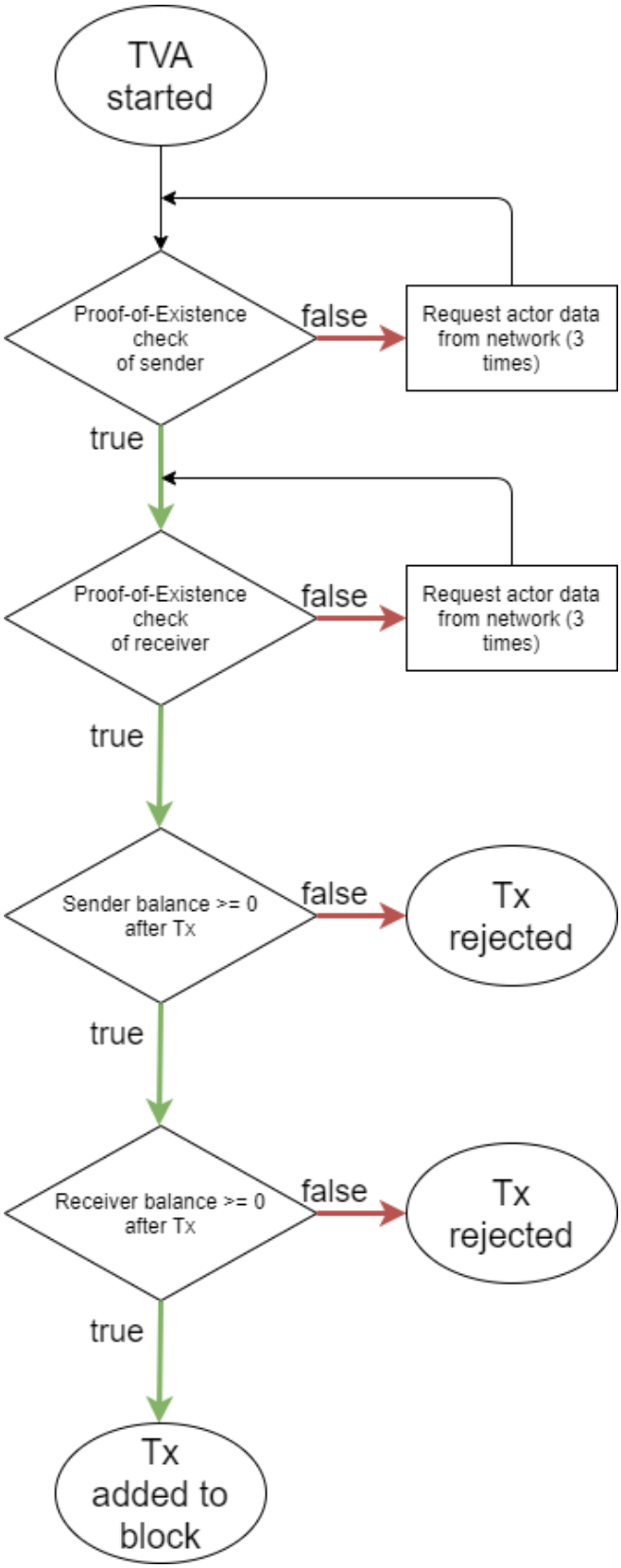
Transaction Verification Algorithm (TVA):

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.


TVA started
Proof-of-Existence check of sender
false
Request actor data from network (3 times)
true
Proof-of-Existence check of receiver
false
Request actor data from network (3 times)
true
Sender balance >= 0 after Tx
false
Tx rejected
true
Receiver balance >= 0 after Tx
false
Tx rejected
true
Tx added to block

TVA Scheme

Each verifier put its ID and signature in the block.

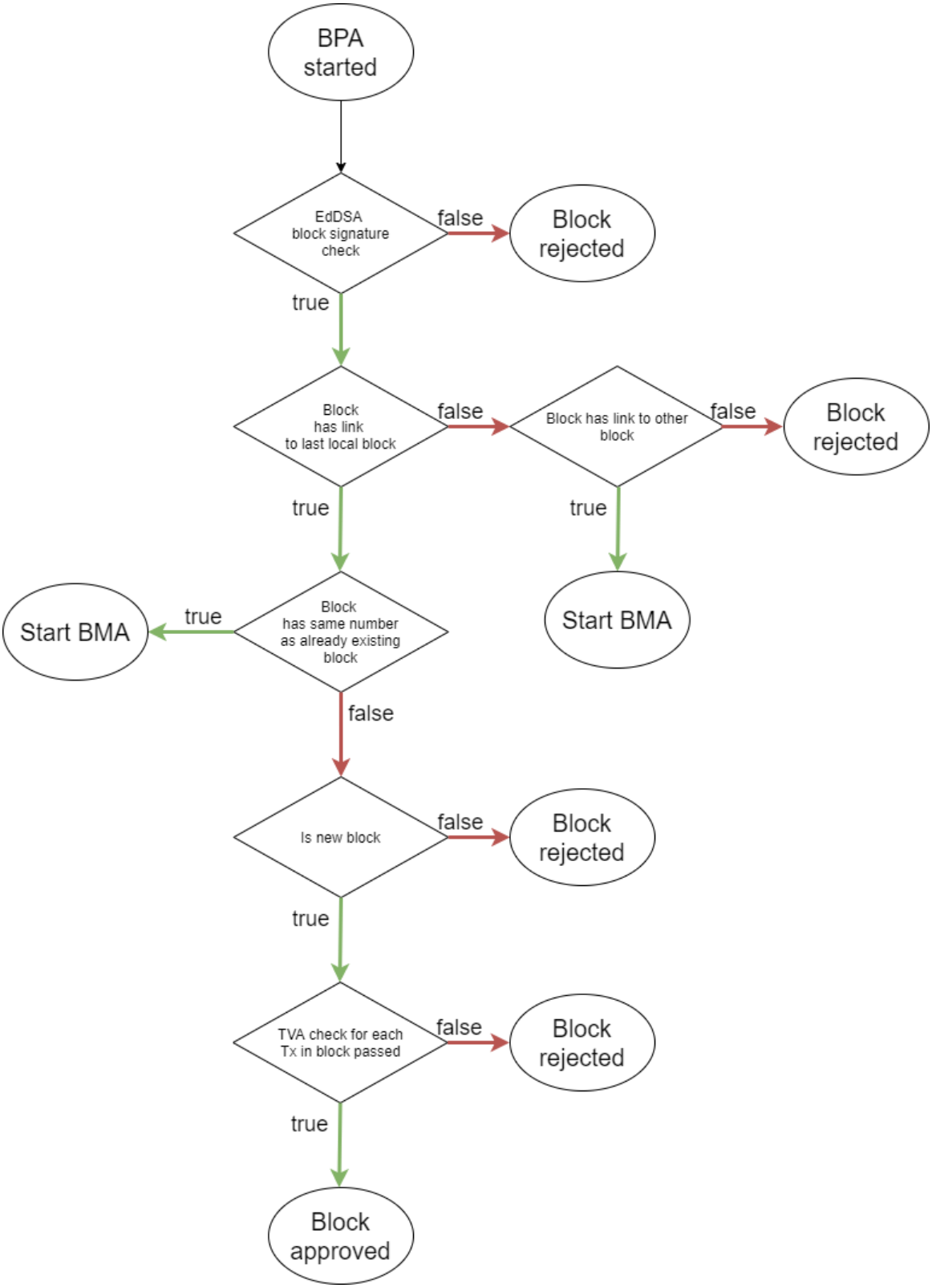
Block is distributed by verifiers via network.

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”.


BPA started
EdDSA block signature check
false
Block rejected
true
Block has link to last local block
false
Block has link to other block
false
Block rejected
true
true
Start BMA
Block has same number as already existing block
true
Start BMA
false
Is new block
false
Block rejected
true
TVA check for each Tx in block passed
false
Block rejected
true
Block approved

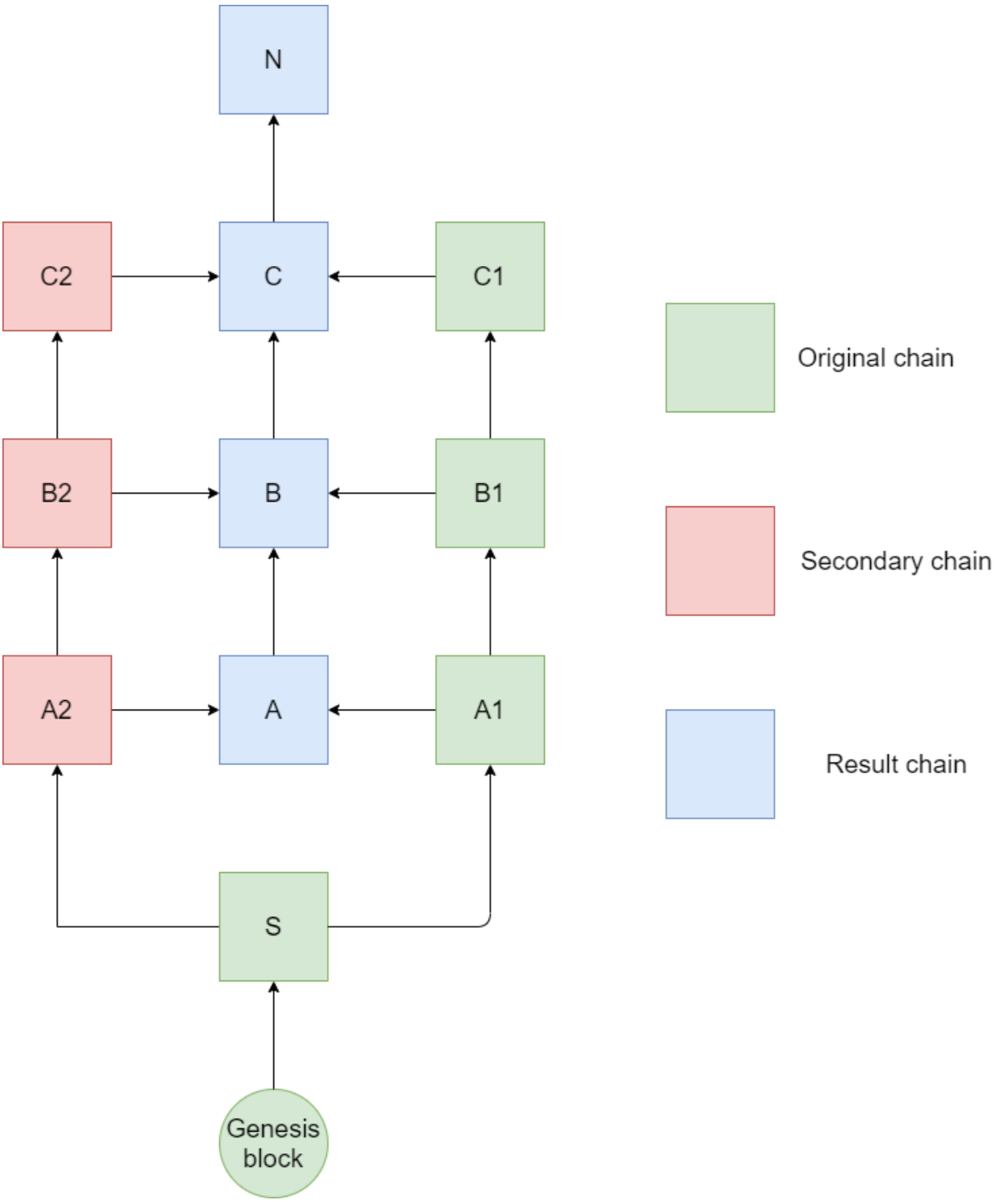
BPA Scheme

Block Merge Algorithm (BMA):

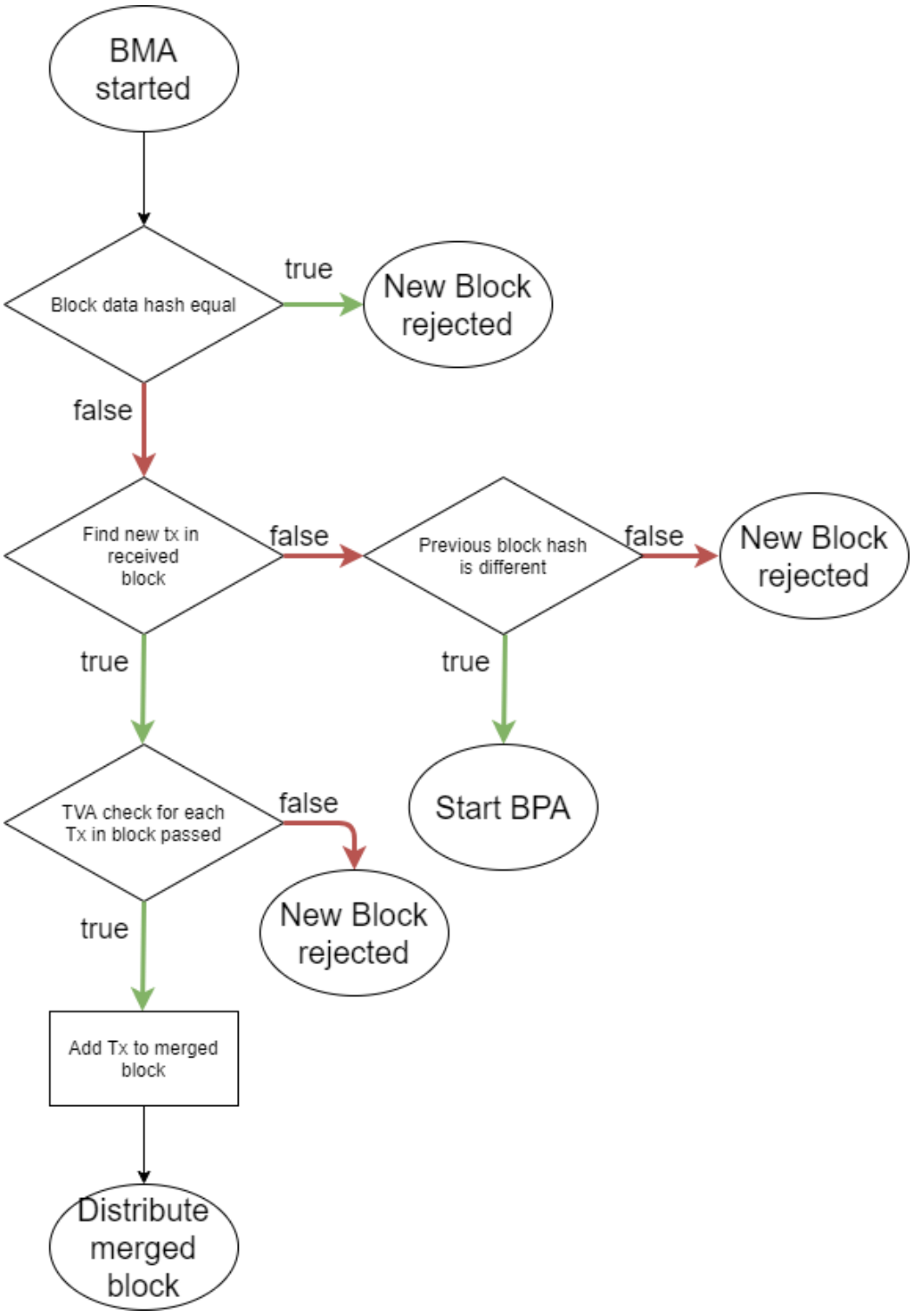
If BMA is started, then one or more blocks in local blockchain can be corrupted or incomplete. Also, new block can have malicious data. Thus, approver must make deep scan of block data.

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 example


BMA started
Block data hash equal
true
New Block rejected
false
Find new tx in received block
false
Previous block hash is different
false
New Block rejected
true
true
TVA check for each Tx in block passed
false
Start BPA
New Block rejected
true
Add Tx to merged block
Distribute merged block

BMA scheme

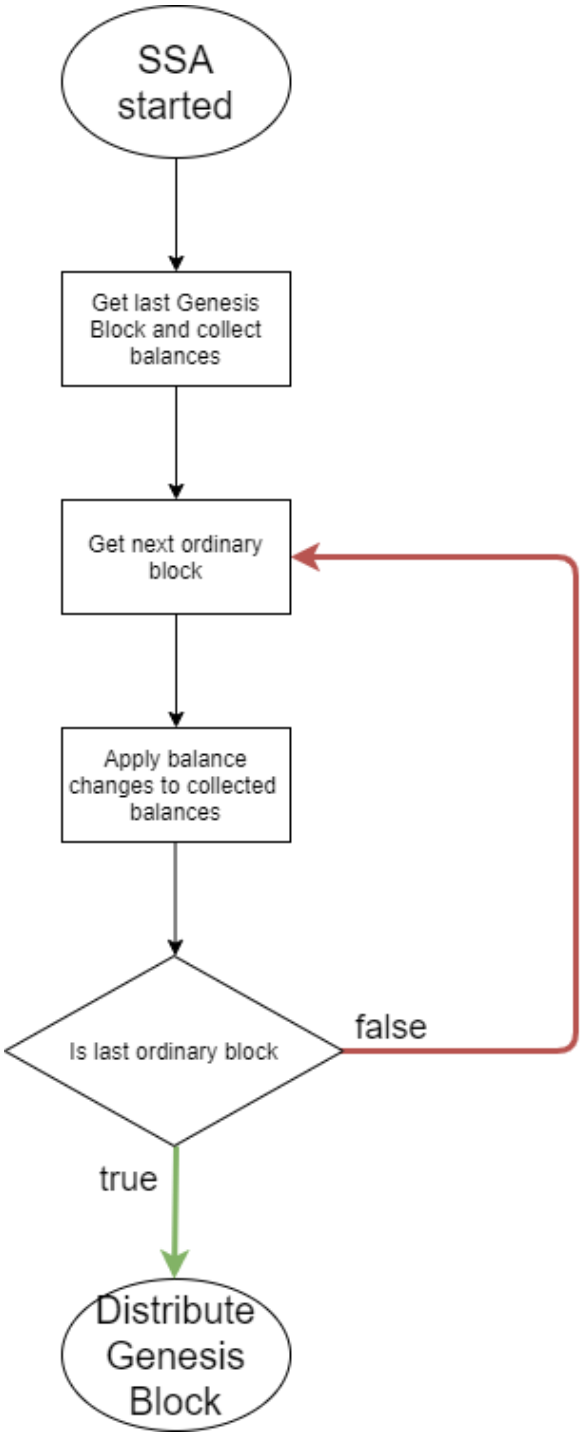
State Snapshot Algorithm (SSA) aka “Genesis Algorithm”:

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.


SSA started
Get last Genesis Block and collect balances
Get next ordinary block
Apply balance changes to collected balances
Is last ordinary block
false
true
Distribute Genesis Block

SSA scheme

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:

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

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

Links

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

Appendix A. Terms and Theory

Distributed architecture – software and network architecture design approach, where each user node is connected to others, aimed to bring (through one-rank connections) stability and high fault tolerance to the network.

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.