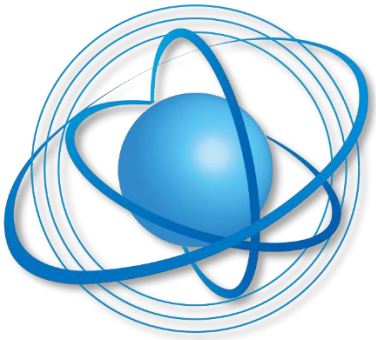


BNChain White Paper

A simple, stable, and scalable blockchain network

V1.0

Preface

At present, the distributed bookkeeping and token incentive timestamp system represented by bitcoin is generally considered to be the pillar of future finance. An emerging technology needs to be able to continuously upgrade and optimize technology, improve its function and performance, so as to be recognized by the market and widely applied, and further lead to the change of an era. The main goal of BNChain is to achieve decentralized governance, so the holders can formulate relevant rules and have enough development funds to mobilize the strength of the global community to promote the development of BNChain.

The core of BNChain will be as stable as bitcoin, with flexible and efficient scalability. Developers can build a strong DAPP and Multi Chain ecology on BNChain to jointly maintain the development of BNChain system. Multiple basic chains can be extended on the BNChain blockchain. Each basic chain can build a variety of application ecology, and realize the cross chain exchange function between multiple chains.

BNChain will provide technical services for each basic chain, and BNC will be the necessary fuel for all transactions.

BitNasdaq Chain White Paper

1. Public chain technology content of Bitnasdaq chain1.1 Features of BNChain modularization 4
1.2 BNChain service module 6
1.3 BNChain operation process 8
1.4 EVM WASM JSVM three types of virtual machine support 9
1.5 SPOS 10
1.6 random number support 11
1.7 super node 15
1.8 Privacy Support 16
1.9 decentralized exchange, C2C support 17
1.10 bit cross-link credits support (BTC Relay) 18
1.11 support MVCCKVDB of storage 19
1.12 support assets on a regular basis thaw function 21
1.13 supports multiple signature feature 22
1.14 supports oracles function 23

1. Public chain technology content of Bitnasdaq chain1.1 Features of BNChain modularization

The BNChain platform is a pluggable and easily upgradeable blockchain architecture that supports consensus, databases, and actuators.

BNChain creatively supports a hierarchical structure, focusing on the public chain of blockchain asset safe storage, transaction clearing, value exchange, cross-chain transfer and supply chain financial ecology. .

To put it simply, two directions can be used to summarize the characteristics of the BNChain platform architecture, that is, from the perspective of vertical expansion, BNChain has the characteristics of consensus, database, and executor (contract). From the perspective of horizontal expansion, the public chain is another advantage.

The modular design of BNChain divides and designs a series of functional modules based on the analysis of the underlying architecture of the blockchain and the different functions and requirements of application development. Through the selection and combination of modules, different products can be formed to meet different market demands.

Developing a software is actually an iterative and evolving process. Therefore, BNChain adopts the development mode of "from chaos to order", which is convenient for developers to adjust and expand at any time. In addition, some developers may find during the development process that some special business logic needs to be customized to match this business logic.

From the perspective of iteration and reconfiguration, as well as the scalability of the system, BNChain takes into account the underlying architecture of the blockchain, the functions and requirements of different application development, and makes the system modular design. Including the queuing mode of MemPool, encryption and signature mode, consensus mode, RPC function, command line command, the internal logic of wallet, the storage mode of database, etc., all modules of blockchain core can be customized.

The modular design is similar to building a robot with building blocks. All parts of the body, such as hands and feet, are placed according to categories, and can be assembled according to their own ideas. BNChain provides a variety of modules such as consensus, encryption and storage. Under the framework provided by BNChain, developers can freely

combine applications and develop easily. Therefore, they can build a public chain only with basic programming ability, and do not need to spend a lot of money to develop the bottom layer of blockchain.

1.2 BNChain service module

Client

The client provides users with management and query functions for accounts, blocks, nodes and wallets, such as creating new accounts, sending transactions, generating random seeds, obtaining block information, and obtaining wallet status. All transactions pass through the client, signed and encrypted, and then sent to the blockchain.

RPC module

The RPC interface is provided to the client, and the client operates the blockchain through the RPC interface, such as creating accounts, querying accounts, sending transactions, querying transactions, querying block information, etc.

Mempool module

Transaction buffer pool, mempool stores transactions from the RPC interface and transactions from P2P. The realization of Mempool is

mainly to solve the problem that the processing speed of the consensus module is slower than that of the RPC module.

Consensus module

Pluggable consensus module design. There are two consensus algorithms for public chains, one is a pure POS algorithm that supports tens of thousands of people to mine together for consensus . The other one is a strong consensus Byzantine consensus algorithm, which introduces the concept of DPOS voting rights: each node can have different voting rights.

Actuator module

The executor is the logic processing center of the blockchain. The executor reads the status through a read-only database and executes it virtually. The execution result only affects the memory and does not save it. There are many types of transactions, and different transactions correspond to different actuators.

P2P module

The P2P module connects each node and broadcasts transactions and block related information throughout the network.

Blockchain module

The Blockchain module is mainly responsible for receiving blocks from the consensus module and storing them on the local hard disk.

Cryptographic signature module

Responsible for the signature and encryption of the transaction, the signature ensures that the transaction can be traced back, and the encryption ensures the security of the transaction information.

1.3 BNChain operation process

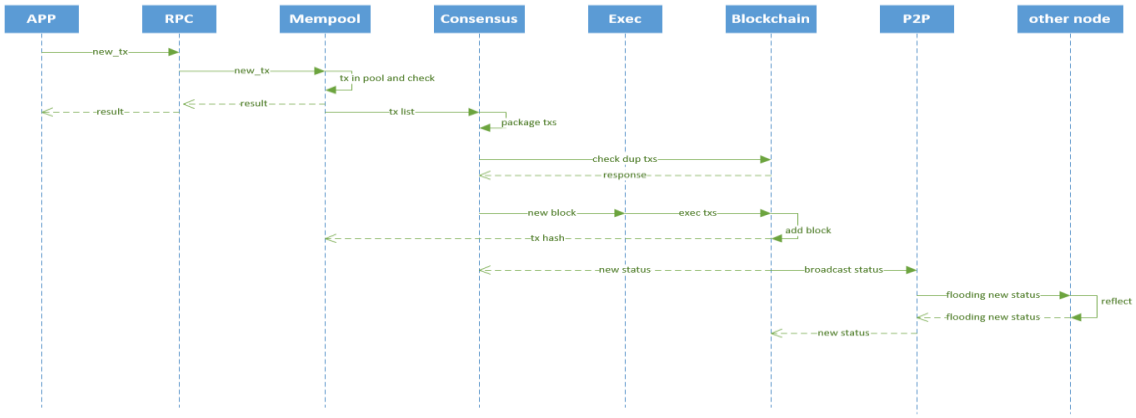
1. The client receives the transaction, signs and encrypts it, and sends it to the MemPool module cache of the node through the RPC module. Transactions received by different nodes are broadcast in the network through P2P module to ensure the consistency of messages in MemPool of all nodes.

2. The consensus module judges conditions such as time or number of transactions, and pulls the transaction list from mempool. After the consensus module excludes duplicate transactions, it packs the transaction list into the block, and then starts to do consensus.

3. After the consensus is completed, the consensus module sends the block to the

executor module for pre-execution. At this time, the local database is not written. Different transaction types enter different executors, such as coins transactions enter the coins executor. After the pre-execution is completed, the consensus module then sends the block to the blockchain module.

4. The Blockchain module broadcasts the block to other nodes through the P2P network, and then all nodes store the block in the local database.



1.4 EVM WASM JSVM three types of virtual machine support

EVM is an Ethereum smart contract virtual machine that uses Solidity to write smart contracts and is compatible with contracts on Ethereum . EVM deployments way through BNChain interface provides a contract to deploy smart BNChain 's EVM virtual machine. You can also call EVM contracts through the interface to execute smart contracts.

WASM (not currently open) , is written in C ++ smart contract, and compatible with the contract on the EOS . WASM deployment is via BNChain interface provides a contract to deploy smart BNChain 's WASM virtual machine. You can also call WASM contracts through the interface to execute smart contracts.

The deployment method of JSVM is as follows: Use Javascript to write smart contracts. There are many Javascript developers, which lowers the barrier of blockchain development. A JS programmer can develop DAPP alone , which quickly and improves development efficiency, which can also BNChain will contract to deploy intelligent interface provides a BNChain 's JSVM virtual machine. The JSVM contract can also be called through the interface to execute the smart contract.

1.5 SPOS

SPOS , or safe POS, realizes the mining logic of POS through ticket. In BNChain public chain such as BN coin , the user can use their BN balance to purchase the tickets. And a ticket has a mining right with a unique Ticket ID. A block can only be excavated by one vote, and the actual mining probability is equally divided (if there are n tickets in the whole network, the probability of one ticket digging to the mine is 1 / N).

The ticket mining process is as follows:

Wallet: Regularly check the BN balance in the account to purchase tickets. When the ticket purchase conditions are met, a ticket transaction is constructed and sent to the blockchain.

Consensus: It will always try to use locally held tickets to package blocks. Once the package is successful, it means that the corresponding Ticket holder has successfully mined and received the corresponding block reward.

Smart contract: The smart contract will write the ticket information corresponding to the address into the blockchain database. Each ticket corresponds to a unique Ticket ID, and a piece of data is recorded in the database.

1.6 random number support

In order to reflect fairness on the blockchain (for application scenarios such as games), a random number that cannot be predicted is required.

The current blockchain generally has the following implementation schemes:

1. Call an external centralized random number generator to obtain random numbers in the contract;

2. Use certain values in the block hash as random numbers.

But these two schemes have very obvious disadvantages, the reasons are:

1. The execution results of smart contracts between multiple nodes in the blockchain require strong consistency. If the contract reads data from the outside, it is very likely to obtain different results (for example, network reasons cause some nodes to read normally, and some Return an error) , which will lead to errors.
2. The hash of the block can be controlled, causing the random number to be controlled. For example, EOS does not provide a good random number algorithm, so many DApp developers will encapsulate what they consider to be a perfect random number algorithm to cause random numbers to be predicted.

For example, the following two examples:

Eosbet’ s first random number attack: This game uses a random number factor named ref_block_num in EOS when the lottery is drawn, but the value in the old block is still read in the contract when the game is drawn, causing the random number to be predicted, and then be attacked.

Eosbet's second random number attack: After modifying the previous question, the

developer introduced a new parameter: user balance as a random number factor. However, the attacker took advantage of this to simulate the exact same DApp code, and then kept modifying the balance to try the lottery logic until the lottery result was collided, and then attacked again.

Many other games on EOS have been attacked by similar means, causing a lot of losses.

Thus, BNChain realized the optimization on random numbers.

First, users use BN in their wallet account to buy tickets. . The wallet generates a randnum at the same time. After hashing, the private key of the wallet mining address, the index corresponding to the ticket (multiple tickets can be bought at a time) and other elements are hashed twice to obtain a public hash parameter (pubhash): the public hash parameter is as follows: the public hash parameter is as follows: the public hash parameter (pubhash) is used to generate a randnum

pubHash =

hash(hash(privateKey:index:hash(randNum)))

Then, the newly purchased ticket contains the pubhash and randnum and is stored in the blockchain. This ticket has a 12-hour maturity period, and it takes 12 hours to participate in mining.

Then the consensus algorithm finds a mature ticket from the blockchain and starts packaging. Since the consensus packaging block operation is only performed locally on the node, it can read the locally stored private key and calculate a private hash (privHash) and put this parameter into the mining transaction:

privHash =

hash(privateKey:index:hash(randNum))

Finally, the smart contract receives the mining transaction and compares the values of hash (privHash) and pubHash. The two are consistent with the successful mining transaction, and the corresponding node receives the mining reward. Otherwise, the execution of the mining transaction will fail.

In conclusion, the implementation of SPOS consensus combines random numbers. In general, the consensus information of other nodes cannot be predicted, so the consensus random number of SPOS can not be obtained.

Moreover, the system sets the privhash not to be disclosed in advance. Even if a malicious miner exposes himself in advance, its corresponding ticket will be voided, and the principal will be frozen for a long time (more than 2 days).

In addition, the system set tickets need to be mature for 12 hours before they can participate in mining. When these conditions are combined, the random number of

the system can hardly be manipulated. In this way, when developers need to ensure fair random number in DAPP, they can directly use the safe random number provided by the system.

1.7 super node

In order to improve the performance of the blockchain, many developers and project parties in the market expect to adopt the consensus of DPOS (Proof of Share Authorization Mechanism), that is to select a number of super nodes with computing power and broadband support on the chain. These super nodes must package the transaction information into the block, and broadcast the block information to other nodes, store the transaction information on the block, and play a common governance role The function of the community.

To measure the success of a public chain, one of the key indicators is the number of nodes in the chain. The super node mechanism can help the public chain quickly establish the ecosystem on the chain. Depending on the operation and maintenance of each super node, the public chain ecology will become more prosperous and a more stable, powerful and decentralized blockchain system can be realized.

At the same time, the public chain operators can set up foundations to promote the

initiative and enthusiasm of super nodes through various token incentive mechanisms and operation means of the foundation, and promote the healthy and sustainable development of the public chain by means of token repurchase and transaction fees.

1.8 Privacy Support

The non-tamperable and distributed characteristics of blockchain technology can indeed prevent users' privacy from being controlled by centralized institutions, which can lead to problems such as being trafficked and hacked. However, the open and transparent ledger allows massive user data to be exposed on the chain. The privacy problem still has not been fundamentally resolved. For example, the original shopping on Taobao is now decentralized. Instead of trading through Taobao, both parties directly mail. Although Taobao does not have the data of these two transactions, their transaction data is recorded on the blockchain network and anyone can view it.

Based on the hybrid model of account and utxo, BNChain implements the blockchain privacy trading system. While using utxo system, BNChain retains the account system, adds ring signature and one-time address, so that the account can freely flow between privacy and openness, and has untraceability and non connectivity.

1.9 decentralized exchange, C2C support

The traditional centralized transaction mode relies on the platform to make credit endorsement to ensure the authenticity and reliability of the transaction, but it also exposes the risk of personal privacy and asset theft. Individuals can't master their own information, but in BNChain network, personal transaction information is stored in all nodes in a decentralized manner, and anyone can review it publicly, forming a multi centralized data storage mode. Skipping the centralized platform and directly conducting transactions between individuals, the transaction efficiency is high. In the BNChain system, each node has a high degree of autonomy. Any node may become a stage center, but it has no mandatory central control function. Nodes and nodes will form a non-linear causal relationship through the network to achieve decentralized, open, flat and equal system.

Compared with the centralized transaction, there are many obstacles to be overcome due to the regulatory requirements of regulatory agencies. Users who trade in this way must abide by the rules of centralized trading service providers and pay the corresponding fees. BN's DEX (decentralized exchange) decentralized trading rules can solve this problem and realize convenient and secure transactions. There are two ways for BNChain to realize the decentralized transaction of DEX: BTC relay and hash locking.

1.10 bit cross-link credits support (BTC Relay)

The use of BTC relay refers to placing BTC light wallets on BNChain, so as to realize the decentralized transaction of DEX.

Simplified payment verification refers to simple payment verification. Bencong Zhong briefly mentioned this concept in his paper. He pointed out that payment can be verified without running full nodes, and users only need to save all block headers. Although the user cannot verify the transaction by himself, if he can find a matching transaction from somewhere in the blockchain, he can know that the network has approved the transaction and obtained the number of confirmed nodes on the network.

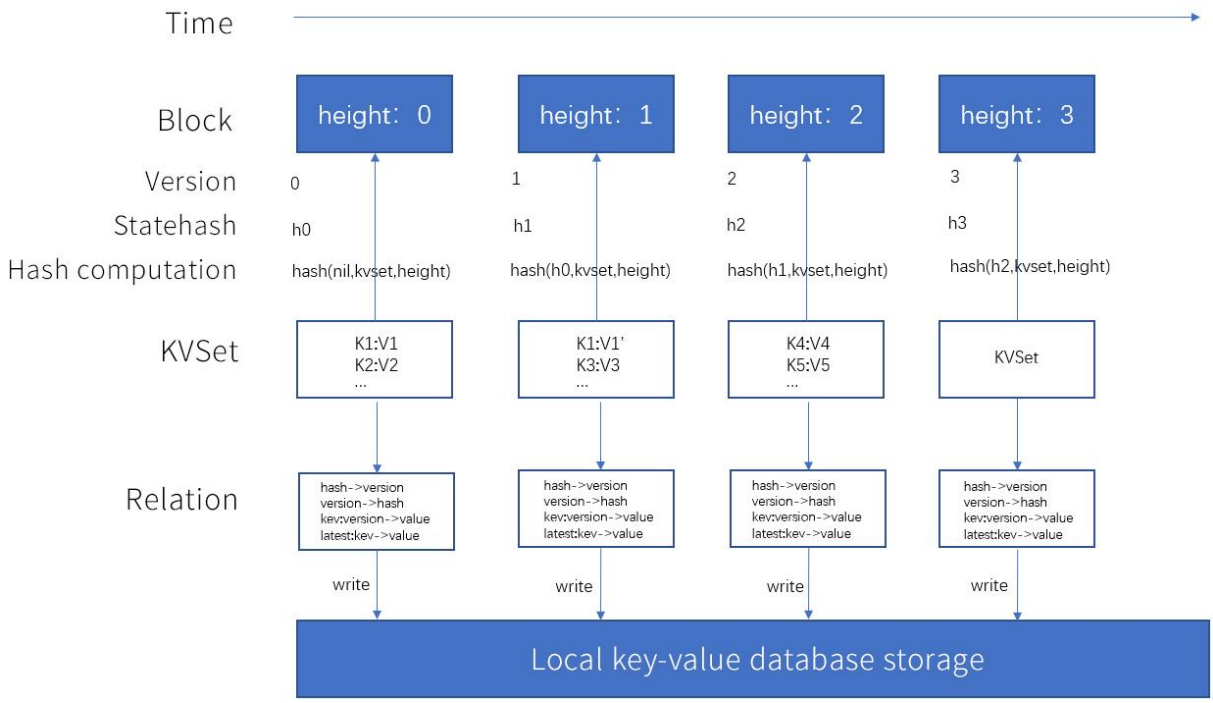
BTC relay refers to copying the bitcoin block header to BNC. Although the transaction cannot be verified in BNC, it can be known that the network has approved the transaction if a matching transaction can be found somewhere in bitcoin. In this way, any interested parties can be matched and the transaction is guaranteed to be completed within 6 hours. In the whole process, the information of both parties is anonymous and no third party guarantee is needed.

1.11 support MVCCKVDB of storage

BNChain implements mvcckvdb (multi version kV data storage). The traditional blockchain stores data in the form of Merkle tree or MPT tree. Every time the data is changed, the tree will be reconstructed, which is inefficient. For example, for a 20 layer Merkel tree, it needs 20 reads to query the data of a leaf node, which results in a data query efficiency of only 1 / 20 of the query efficiency of a common database. For a system that can complete 100000 reads per second, it can only read 5000 transactions per second, which greatly limits the reading performance of the system. When writing data, it is also necessary to load the data of multiple nodes on the tree branch, and finally write it to the disk after the update. This operation consumption is relatively large.

We draw lessons from the mvcc concept of database design (multi version concurrency control). We design the data storage format of kvmvcc of BNChain, which is used to improve the inefficient problems in mavl or MPT structure, and better meet the requirements of maintaining high data read and write performance after the blockchain data grows to a certain scale.

The idea of KVMVCC data storage format is as follows:



- Hash calculation:

Statehash = hash (prevstatehash, kvset, height), which contains the state hash information of the previous block, the state data of this block, kvset information, and the height information of this block (i.e., version information).

The following correspondences will be stored in the database of each node:

hash->height(version)

height(version)->hash

key:height(version)->value

lastest:key->value

- Data query:

According to statehash, the corresponding height (version) can be found, and when the corresponding height can be found according to the height, the value corresponding to the specific key value can be found.

- Data verification:

For a KVSet with a specific height, the hash operation can be performed based on the hash values prevstatehash, KVSet, and height of the previous block. If the hash values match, the data has not been tampered with. Otherwise, the data has been changed or the data is wrong (the height is wrong, Or the KVSet data is incorrect).

- Maintenance of the latest version data :

In particular, when storing the key and value values of the latest block, the mapping of key:latest->value is related to the local key-value database while retaining (new key) or updating (key with historical version). storage. When you need to get the latest batch data, you can query the latest data in batches based on the latest prefix (which can be customized). Since the usual key-value database can well support prefix matching queries, the query efficiency will be higher, which is much higher than that of the Merkel tree storage structure.

1.12 support assets on a regular basis thaw function

Users can use these contracts to freeze a part of their assets and unfreeze them to the corresponding beneficiaries regularly according to the rules, which is suitable for application scenarios such as installment payments, employee incentives, and inheritance distribution.

The function provides the following 3 types of operations

1). Create a regular unfreezing contract: specify the asset type and total amount of assets to be paid when creating, and the form of regular unfreezing.

2). Beneficiary's withdrawal: the beneficiary withdraws the unfrozen assets.

3). The initiator terminates the contract: the initiator can terminate the performance of the contract.

Two forms of thawing are currently supported

1). Unfreeze a fixed amount: According to a specified time interval, unfreeze a fixed amount of assets.

2). unfreeze according to the fixed proportion of the remaining amount: according to the specified time interval, unfreeze according to the fixed proportion of the remaining amount. In this way, the more to the back, the less thawed.

1.13 supports multiple signature feature

BNChain system can support up to three 2 separate private control multi-signature account. Users can freely define the weight of each private key. Finally, the weight can be used to determine whether a transaction can be initiated to ensure the security of the account. At the same time, it can prevent the loss of a private key and the account amount can not be recovered.

For example, there is a wallet with 3 private keys, but the number of authorized votes for each private key is different. For example, Wang Yi, Zhang San and Li Si, Wang Yi' s private key can have two votes. Thus, when Wang Yi initiates a transaction, he only needs the private key signature of Zhang San or Li Si, and then the transaction can be successfully initiated. On the contrary, if only Zhang San and Li Si initiate the transaction, but Wang Yi's private key has two votes As soon as there is no signature, they have only two votes in total, so they can't start the transaction.

1.14 supports oracles function

The oracle realizes the link between the blockchain and the real world. The oracle is a trusted entity that introduces information about the state of the external world through signatures, allowing certain smart contracts to react to the uncertain external world . The oracle has the characteristics of non-tampering, stable service, and auditable .

The release data of the oracle contract is divided into three steps:

1. Release data release event (inform the whole network that the result of an event will be published in the future, and a unique event ID will be assigned. If the event does not occur, it can be revoked).

2. Pre-release results (the data provider pre-releases the time results, if the results are found to be problematic by the audit, they can be revoked).

3. Release results (after the pre-release results are audited, they will be finally released on the entire network, non-tamperable, auditable and traceable).

Other contracts (such as quiz contract) can be used in Step 1 above events the ID and specific events to carry out (quiz) activity. When the result of step 3 is announced, the contract will trigger the contract to complete the auction settlement according to the result corresponding to the event ID, so as to realize the objective, credible, auditable and traceable fair guessing without intervention.