HRD white paper 1.0

2019.12

Global IoT business tree-like distributed application architecture system

Catalogue

Summary

Blockchain technology, as a decentralized value transmission system, was first proposed by anonymous Satoshi Nakamoto and applied to Bitcoin. In the Bitcoin system, in order to complete relatively complex types of transactions, Nakamoto creatively proposed scripts Mechanism. But when developers want to implement more functions through Bitcoin script, they are often subject to many restrictions. To this end, Ethereum proposed by Vitalik Buterin makes Turing-complete smart contracts and EVM to make blockchain-based The application development of technology has become possible and has been praised by the industry as "Blockchain 2.0" after Bitcoin. However, both Bitcoin and Ethereum are facing the expansibility and rapid expansion brought by the rapid growth of users and transactions The problem of transaction delay. The root cause lies in the structure of the single chain in the current blockchain system, which makes many outstanding projects lack sufficient flexibility in the face of these problems. The development of blockchain in the field of the Internet of Things is a natural application. It's also difficult.

In order to solve these problems and better integrate the blockchain

with the Internet of Things technology, after continuous exploration and demonstration, the HRD tree-type blockchain was proposed by South Korea's Shin Ming Energy Technology Group , and HRD tokens issued by Hong Kong Fushi Group. Shin Ming Energy Technology Group is located in Seoul No. 528 Teheran Road, Jiangnan District, Special City, the main promotion business: new material environmental protection lubricants; the company's core products are mainly used: automotive lubricants; industrial, agricultural, heavy equipment, marine, aircraft lubricants; satellite Special lubricants for robots, aerospace, rockets, etc. The company's subordinates include: Manufacturing Division, Distribution Division, Education Division Technology Research Institute Overseas Division (China Branch, Indonesia Branch, Malaysia Branch, Philippines Branch, Vietnam Branch) . HRD has a tree structure of "main chain + multi-application branch chain". Through the infinite expansion of branch chains, it can achieve transaction expansion and high concurrency problems that cannot be solved by the single chain structure. At the same time, as the infrastructure of the Internet of Things, HRD will establish multiple Physical device trust and data communication in heterogeneous environments, creating a stable and reliable technical foundation for the more complex business models of the future Internet of Things.

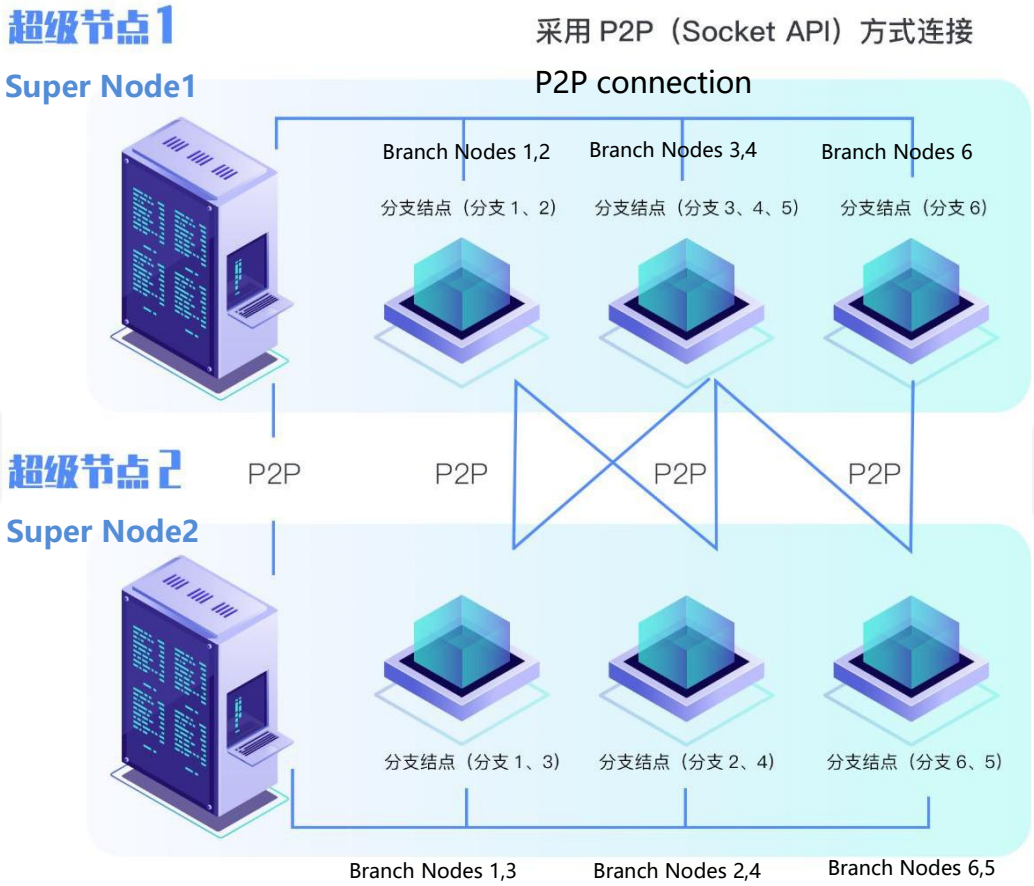
System specification

Basic introduction

HRD is a block system built on a P2P network. It is similar to the current popular P2P digital currency system.It maintains a transparent ledger in a decentralized manner to achieve autonomous and secure management and efficient flow of user digital assets. The HRD system is targeted at the IoT (Internet of Things, (Internet of things) data business requirements design, using block technology to provide a decentralized security management platform for IoT data services, to achieve the high concurrency and low latency performance requirements required by IoT systems.

HRD organizes user transactions through security consensus and forms data blocks in chronological order. Unlike single-chain systems such as Bitcoin, HRD uses a tree structure to store and arrange blocks, which can be forked to form multiples according to the type of business and data load. Branches. Blocks between branches are independent of each other, and new blocks are only related to their own branch data. In the case of multiple branches, according to

business data traffic, they can be distributed to multiple branch blocks, resulting in scalability High concurrency is the basic performance required for IoT systems.The multi-branch structure of HRD is composed of a single secure main chain and numerous application branches.The security main chain is used to support the entire network consensus mechanism, and the application branch chain is used for actual business. In the application branch chain, a minimum of 2 seconds of low-latency transaction confirmation can be provided.The user can specify the urgency of the transaction and pay the corresponding transaction fee to achieve low-latency business.



Brief description of consensus mechanism

As we all know, in the various discussions of the "impossible triangle", the result of decentralization often means inefficient TPS, and the massive data of the Internet of Things has become a huge stone that cannot be removed in the consensus building. In the field of IoT, what is the right consensus? Let's start with the evolution of consensus algorithms.

Proof of X is a type of consensus that is currently widely used in the public chain field.PoW is the first to be applied, but there are waste of resources, concentrated computing power, lack of finality, and low performance.

PoS is currently a strong competitor, which can avoid waste of resources, weaken the demand of the central mining pool, and reduce the possibility of attacks by 51% .However, it also has difficulties in determining the number of bookkeeping nodes, the problem of unexpected centralization, and Nothing at Stake. .

In order to solve the above drawbacks, many hybrid consensuses have also been born, hoping to combine the advantages of the two and avoid certain disadvantages, including PoW + PoS, DPoS + BFT, etc. Therefore, the hybrid consensus mechanism may be the later

development of the public chain A way out.

PoW consensus algorithm

PoW (Proof of Work) is the proof of workload.The digital currency is allocated according to the workload of the miner. The higher the performance of the miner, the more the number, the larger the workload, and the more digital currency will be obtained.

BTC is the most typical prototype of the PoW solution. It involves solving a mathematical problem through the mining process. The miner completes PoW through this technical means and obtains the right to account. Because it requires computing resources, successful miners get BTC is used as a reward. In order to control the monetary base, mining is set to a more complex mode. Because the probability of each miner solving a problem depends on his computing power, the difficulty of mining is determined by the sum of all computing power in the system .

For cryptocurrencies with the PoW mechanism, miners confirm and fix transfers by competing to solve mathematical problems. The first miner who solves the problem is rewarded. The complexity of the problem is deliberately manufactured to control the monetary base.

This process is considered by some to be a genius move, which solves

the problem of General Byzantine well. But others have criticized it for being inefficient because it lost resources for free. At the same time, the single PoW mechanism is also facing a 51% hashrate attack, etc. Security issues.

With the development of BTC and the development of the blockchain industry, the shortcomings of the PoW mechanism have also been exposed. Currency holders cannot participate in any decision, and the discourse right is concentrated in the hands of miners, which runs counter to the concept of decentralization and centralized decision-making power. In the hands of a few miners.

DPoS consensus algorithm

DPoS is a new type of consensus algorithm based on PoW and PoS that guarantees the security of digital currency networks. It can solve the problem of excessive energy consumption generated by PoW during the mining process, and can also avoid the distribution of PoS rights and interests. The problem of "trust balance" bias that may arise. Then, DPoS can logically become a representative consensus mechanism that stands out from the consensus mechanism 3.0. DPoS allows users to participate in mining widely, which means that every currency holding Anyone can vote, resulting in a certain number of representatives, or understand that a certain number of nodes or

mining pools, their rights are completely equal to each other. Holders can replace these representatives by voting at any time to maintain the chain "Long Purity" on the system.

EDPoS+CPoW

In order to prevent the collapse of the single consensus node and stop the operation of the entire ecosystem, and to prevent the overall blockage of the block network due to the collective strike of DPoS nodes, CPoW (sustainable proof of work mechanism) and the more "decentralized" EDPoS (can be The extended entrusted equity certification mechanism came into being. HRD's security consensus mechanism is EDPoS (expandable entrusted equity certification mechanism)+ CPoW (Sustainable Proof-of-Work Mechanism), the node income is the reward for block production plus the total transaction fee for intra-block transactions. Users can use Token to

vote for EDPoS nodes, and vote for EDPoS nodes to increase the block probability. When EDPoS nodes The new block is successfully generated, and the corresponding voting users also share the block reward according to the voting quota. The node needs to raise more than 2% of the total token supply to vote in order to become an EDPoS node.

CPoW is a supplement to the EDPoS consensus.Each round of EDPoS negotiation process has a certain probability of giving the first block right to the CPoW consensus. The fewer the tokens, the lower the security and reliability of the EDPoS consensus.In this case, the higher the probability of obtaining block weights through the CPoW consensus, the hybrid CPoW mechanism enhances the system security and reliability. In the tree structure of the HRD, Except for the secure main chain, the rest of the branch chains are

equal and independent. The EDPoS node group jointly builds a block sequence through security calculations, and at the same time generates a true random number beacon. The application of the branch chain block generation series allocates random numbers from the secure main chain. Beacon calculations are generated.

According to the Byzantine fault-tolerance principle, less than 1/3 of the malicious nodes will not be disturbed in the entire secure computing process; reasonable selection of negotiation algorithms and parameters can achieve non-51% attacks, and the secure computing process will not be controlled. In the HRD system The consensus mechanism can achieve high consistency, and systematic forks are very rare.In the case that the number of tokens held by malicious nodes is less than 50% of the total number of participating EDPoS, 3 confirmations can ensure that the historical data of the main chain cannot be rolled back.

Network description

HRD network consists of nodes running HRD software to form a P2P network. The overall network architecture of HRD can be divided into three layers: node network layer, terminal service layer, and IoT terminal layer.

The node network layer is composed of nodes running the core node program of HRD. Nodes synchronize check blocks and transaction data, and organize consensus block data.

The terminal service network forms a distributed terminal background, providing access services for IoT (Internet of Things) terminals.

In order to support the huge IoT (Internet of Things) business, the node network and terminal service network together form the HRD service platform.

The IoT terminal layer includes smart sensors, controllers, and mobile terminals.It embeds light client programs and saves private keys locally to complete transaction construction and verification.

System software composition

In order to better support multiple application scenarios in the complex environment of IoT, while ensuring the reliable operation of blockchain services and the needs of ordinary users, the design of the HRD system software includes five parts: the core wallet program, the light wallet background service system, Mobile light wallet program, embedded system light wallet SDK and online block browser.

Core wallet program

The core wallet program is used for backbone network nodes and ordinary users.It has certain requirements for the operating environment and hardware, and can fully use all functional modules of the block system.

Light wallet background service system

LWS is the abbreviation of light wallet service, which is a bridge between the HRD public blockchain backbone network and terminal data acquisition sensor equipment. Through it, the blocks and transaction data of the HRD core wallet are updated and cached in the LWS in a timely manner. Some high-speed memory databases and local databases.

Based on these data, it calculates the latest UTXO set of public key addresses corresponding to the keys held by different terminal devices, and publishes this information to the Amazon cloud facility through the mqtt connection to AWS 'IoT Core. Its message broker forwards to the corresponding terminal device that subscribes to this information. Correspondingly, the terminal device will package the data into the transaction after obtaining the data collected by the monitoring according to the UTXO list related to itself. mqtt is published to

Amazon IoT Core.

Through the latter's message broker, it is pushed to the LWS that subscribes to these devices to send transaction topics.The LWS will verify these transactions.If the verification is successful, it will forward these transactions to the HRD core wallet through the Socket API. The P2P network interface broadcasts these transactions to the entire HRD network, and the block-producing node collects these transactions, and finally completes its operation of packaging the block chain.

LWS uses the long-connection, two-way message-based pub / sub message broker provided by AWS to release the coupling relationship with the device-side data interaction of a large number of connections, which solves the high concurrency and high scalability of the device. For block and transaction data To store queries and updates to UTXO data, LWS uses AWS's Amazon DynamoDB service to store their KV key-value data.

Considering the massive transaction data and packaged block data generated by the high concurrent TPS on the multiple branches of the HRD public chain network, and the massive UTXO data, using Amazon's ms-level response-delayed data storage service Amazon DynamoDB, it is possible for each business branch The chain creates a block database and transaction database to accelerate data retrieval

capabilities.

The LWS synchronizes the downlink blockchain data of the backbone network with the high-throughput, elastically scalable Amazon Kinesis service, using Amazon S3 distributed data with high scalability (durability), high durability (high availability), and high availability (availability) .The storage service caches a large amount of block files to the Amazon cloud, completes real-time block data collection and processing, and can be used by other LWSs with local physical addresses and even provide retrieval services to LWSs around the world.On the other hand, LWS is working with When the core wallet is out of step or data error, you can use the data in S3 to recover quickly. In addition, LWS uses AWS rules engine rules engine to convert and route messages to AWS services, and the backend uses Kinesis services to offload data to different AWS services. Or connect to the Lambda service to offload data. In an environment where the regional network transmission is uneven, you can also use AWS 'Cloud Front service to provide CDN-like functions.

Using a petabyte-scale Amazon Redshift relational data warehouse, you can store structured blockchain data, which is convenient for HRD blockchain web browsers, smart device wallet apps, and HRD blockchain development testers to debug the data view of the tracer

runtime.

LWS is developed using the high-concurrency language golang.The program uses goroutine and channel facilities to ensure that a large number of device terminals can send transactions to the core wallet backbone network simultaneously and efficiently.This enables high-speed on-chain transactions.

Mobile wallet program

The mobile light wallet program can enable terminal nodes to verify transactions without running the full wallet program. It is mainly used for IOS and Android mobile terminals, where network bandwidth and hardware performance are relatively limited to provide users with a secure wallet service.

Embedded system light wallet

The embedded system light wallet SDK provides a light wallet API for IoT smart hardware.It can access the HRD network through a terminal server.It does not need to perform heavy block synchronization and block data storage locally.It focuses on the construction and verification of transaction data related to the business. right.

Online Blockchain Browser

The online block browser cooperates with wallet nodes to display the status of the block system in real time and query historical block transaction data.

System Features

- Tree

Unlike traditional single-chain systems such as Bitcoin, HRD uses a tree structure to organize and store block data.

- High expansion

Facing the massive data transactions of the Internet of Things, the horizontal expansion of the entire system is achieved by relying on an infinitely expanded tree structure.

- High concurrency

The number of transactions per second (TPS, Transaction Per Second) of the application branch chain can reach 5,200, and the fastest 2S of a transaction can complete the on-chain confirmation, which can meet the characteristics of frequent and low-latency data transactions under the Internet of Things.

- Fast data access

Through a good system design, it provides stable interfaces and back-

end services for both streaming data and persistent data, making the data fast on the chain.

- Data notarization

The data identification and circulation records are stored on the HRD chain.Once the data identification is on the chain, it cannot be changed.It can provide data consistency proof and notarization information for the demander and provider.

- IoT value transfer platform

As the underlying infrastructure of the Internet of Things technology, it provides a stable and reliable technology platform for data transmission and value circulation on the Internet of Things.

Tree block structure

In today's common blockchain projects, all transaction information is stored in single-chain blocks, making the entire system lack sufficient flexibility in the face of growing transaction scales. In HRD, the main chain data and application data are processed. Segmentation processing is used to store system block data with a tree block structure of "safe main chain + multiple application branch chains".

The security main chain mainly stores data related to transactions and security consensus; the application side forks from any chain to generate a branch chain (fork chain), which specifically organizes and stores data related to application business. And with the expansion of transaction scale The branch chain can continue to establish child-level branch chains. In this manner similar to vertical division, the disadvantage of all transactions in the traditional single chain structure filled in the main chain block is eliminated, and the horizontal expansion of the overall system is achieved.

The greater the number of branch chains on HRD, the higher the TPS (Transaction Per Second) that the system can bear.With enough

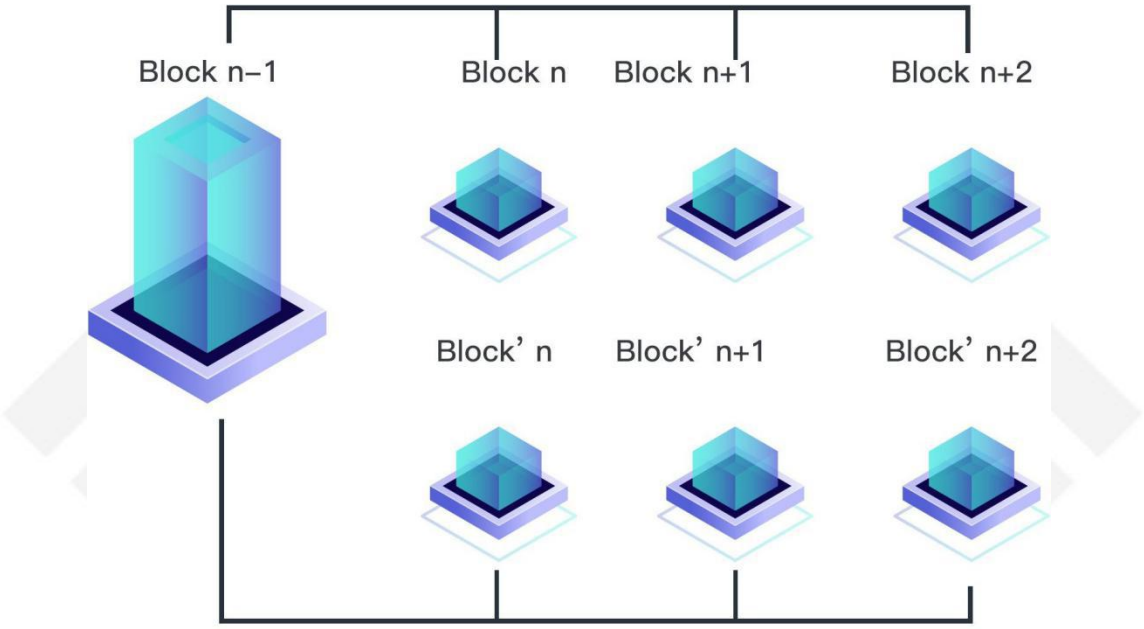
application branch chains, HRD as a whole can achieve ten or even billions of TPS.

Branch ID

The blocks of the HRD system are connected together in chronological order, forming a tree structure with multiple branches. In HRD, the security main chain and the application branch chain are collectively called "branch". Each branch will have a unique branch identifier to Marking. The security main chain uses the genesis block hash as the branch chain ID, and the application chain uses the hash of the first block after the fork point as the branch chain ID.

Before the fork, the parent chain and the branch chain have exactly the same chain structure and transactions; after the fork point, they are independent of each other and do not interfere with each other. The same token that appears before the fork point can be on the parent chain after the fork point. Create different transactions in the branch chain and send them to different addresses; the block data before the fork point can also be used in the parent chain and the branch chain. Users need to specify an anchor block when creating a transaction, which is marked here All branch chains after the block are valid. In Figure 2.1, if the anchor block is set to Block n-1, the created transaction will be included in the two branch chains; if set to Block n,

the transaction Only valid on the parent chain, new transactions can be created in the branch chain to send tokens to other addresses.



Security main chain

The security main chain is the main chain in the HRD tree structure, and all the branch chains are its "offspring" .It is used to support the security and consensus of the entire block system. In the P2P network, the synchronous broadcast message forwarding of the main chain takes precedence. The level is higher than the application branch chain. In addition to recording the main chain token transfer, the secure main chain also retains key process data negotiated by the EDPoS nodes. Subblocks cannot be inserted between the blocks of the secure main chain, and can only grow at the predetermined block interval. A considerable part of the capacity records the data of the consensus

negotiation process (take 23 EDPoS nodes as an example, the negotiation data will occupy about 115KB of each block), so the transaction capacity of the secure main chain is lower than the application branch chain. The secure main chain uses blocks The system genesis block is the starting point, and blocks are generated through the EDPoS + CPoW consensus sequence. The security main chain is used to support the security and consensus of the entire block system. All application branch nodes need to synchronize and verify the main chain block header information. After the new node is connected to the network, the main chain synchronization is first completed before the corresponding application branch synchronization is started.

Special transactions on the main chain

In the security main chain, in view of the special features, there are three types of transactions related to the consensus mechanism that are unique to the security main chain: EDPoS node voting transactions; EDPoS node registration transactions; CPoW block reward transactions.

EDPoS node voting transactions

The EDPoS node generates a Delegate template address.For the first

time, you need to send a Token to the address yourself to complete the release of the Delegate address chain.The user creates the Delegate address using the same parameters as the EDPoS node, and registers the Token at the Delegate address to complete the Token voting. Nodes can use the voting of the Delegate address as the weight to participate in the EDPoS negotiation process. When users vote for the coin deposit Delegate address, the ownership still belongs to the user and can be taken out at any time, but once it is taken out, the number of votes of the corresponding node also decreases.

EDPoS node registration transaction

EDPoS nodes need to raise enough tokens to vote in each round of negotiation, and use this to create registration transactions to register on the chain in advance and publish their initial negotiation parameters.Only nodes that have completed registration before the start of the negotiation round (more than 2% of the total votes) Allows entering the negotiation process and obtaining block rights. CPoW block reward transactions

The CPoW consensus is only used for the main chain consensus block generation by default, and the corresponding block reward is provided

to participants through this type of transaction. The role of this type of transaction is similar to the coinbase transaction in Bitcoin.

Application branch chain

In HRD, the application side sends a special type of transaction on the parent chain ------

Fork transactions are used to create application branch chains. The block generation interval of application branch chains needs to be consistent with the secure main chain. Other main parameters can be configured by the creator during the initialization of the creation branch. , Block rewards and additional offers.

The first block of the newly created branch chain (branch start block) is stored in the forked transaction. The distribution of the token of the branch chain can be defined by the creator, there are three ways:

- Create independent branches, reset the total number of tokens and the allocation method at the starting block of the branch;
- Fully inherit the fork point token distribution;
- Inherit the distribution of the fork point Token, and issue additional shares on this basis.The distribution of the additional portion is

defined in the branch starting block.

Since the fork point, the branch chain token and the parent chain are completely isolated.

Mortgage mechanism

In order to prevent malicious people from consuming resources on the parent chain through high-frequency forks, each time a branch chain is established, the parent chain Token is used for collateral, and the Token used for collateral in the fork transaction is sent to a special address to be frozen. The mortgage token is thawed in stages according to the difference between the height of the parent chain block and the starting height of the parent chain block, and the creator can transfer the thawed part of the token to other addresses after signing with his own private key. The difference between the block height and the starting height decreases, and the halving is completed every 525,600 blocks. The base number N of the mortgage token is determined by the total initial token supply of the parent chain.

Apply branch chain to produce blocks

The security of the application branch chain depends on the

consensus mechanism of the security main chain. In HRD, there are three main ways to generate blocks of the application branch chain:

EDPoS nodes generate blocks. This method can be done without setting up a block node, and the EDPoS node can generate new blocks for branches while obtaining the block chaining right of the main chain. This method corresponds to the open business model;

The self-established node generates the block. The application can also set up the self-created block node in the application branch chain, and use the random beacon generated by the secure main chain to set up the block generation mechanism of this branch. This method corresponds to the closed business model.

Custom consensus mechanism. In addition to the above two block generation methods, the application side can also customize the consensus mechanism for the application of the generation of branch chains and sub-blocks.

Compared with the application of the branch chain and the secure main chain, in addition to generating normal branch chain blocks, in the normal block interval every minute, sub-blocks can also be generated for low-latency transactions. 2 seconds, and it is not possible to generate empty blocks. The nodes that generate subblocks are determined by independent random beacons of block blocks with

the same height as the security main chain. The subblocks have no additional block rewards, but they can obtain high transaction fee income.

Considering the sub-blocks in the application branch chain, the transaction capacity of each application branch chain can be increased by 30 times, and the tps can reach 5200. When the data service requires higher transaction capacity and concurrency, multiple can be created for the current application branch chain Branch chain to achieve high-level tps.

Branch empty block

In HRD, the application branch chain height is maintained to be highly synchronized with the block of the secure main chain.It is used to support cross-chain transaction between application branch chains and to facilitate the branch chain block to refer to the consensus result of the same height of the main chain. Therefore, in order to maintain a high degree of consistency, the application of the branch chain may generate empty blocks, and between the empty block and the next branch chain block, no sub-blocks can be generated.

When the application branch chain selects the main chain's EDPoS node to produce blocks, the main chain EDPoS node will take the

application branch chain token value as a priority reference and autonomously select the supported application branch chain. Application branches with lower value may be partially hosted The EDPoS node of the chain is excluded from the block list. At this time, in order to maintain the high synchronization of the main chain, the branch chain will automatically fill an empty block.

Branching parameters

Application of the branch chain to carry the transaction data of the actual business, the size of the sub-blocks and branch chain blocks and the main chain blockConsistent. By reducing the block generation interval of the sub-blocks to expand the branch chain tps, the basic parameters of the application branch chain are as follows:

User key and address

There are two types of HRD addresses: public key addresses and template addresses, which correspond to specific public keys and templates. The address length is fixed at 33 bytes. In the interactive interface, the encoded address is used as input / output parameters.

```
pubkey address:
encoded address = '1' + BASE32Encode(pubkey + CRC24q(pubkey))
template address:
encoded address = '2' + BASE32Encode(template ID + CRC24q(template ID))
```

Among them, BASE32Encode uses the Crockford scheme character set, but does not perform the symbol check process in this scheme.

Key and public key addresses

The HRD system uses curve25519 as the basic security algorithm.The user's private and public keys are both 32 bytes, and the private key signature is 64 bytes.curve25519 has the same security as P256, and is the most efficient asymmetric security algorithm in the same security algorithm. .Type prefix + public key as wallet public key address.

In order to ensure the security of the user's private key, the local

storage is encrypted using the chacha20 + poly1305 algorithm. The user needs to enter a password before using the private key for signing operations.

Template address

The template address consists of the type prefix + template ID. The template ID consists of a 2-byte template type + the lower 30 bytes of the parameter hash. For example, a 3-5 multi-signature template:

```
public keys:
  1: fcd74aa82a1eb098830a2fcc877735a60152b441c16b2212157c4215db074e88
  2: f1a1ced60a7ecdf83735a3380765f2ef77221f367da05bd901e885b9d799aec5
  3: c2885254a2acefaeb05bd94b0e73e483bded994b02ebd0bc6b3523c2dde558dd
  4: e2de897ad0935bbfd6cca48da2ee285c87ae784285df35513180143ec55c8450
  5: b1f1ce918f30b46aa3d2648810f6153410e44122c042998699323b982664a16f
tempate ID:
000244c03d536e6175912b3040aa876388b197c21ae55c283f182403ab610852
encoded address:
2a8463ar34gc3ya2wwmdc55xhh1hrfaj060ns2xb1ds9kvg240803k041
```

With parameter template

Today's popular blockchain systems can provide scripts or smart contracts running on different VMs (Virtual Machines, Virtual Machines), which can perform powerful and flexible function extensions to the basic ledger of the block system. However, as of now,

blocks The VM module in the system is still in its infancy.In addition to problems such as inherent security vulnerabilities, operating efficiency and usage rates have also limited the scope of smart contracts to a certain extent. The HRD system does not provide scripts and smart contract systems, but uses The process template with parameters implements commonly used scripts and smart contract functions. The corresponding template address is used to provide user function calls.

Blocks and transactions

Block

The data structure of the HRD block is designed as follows:

Elem	Type
nVersion	uint16
nType	uint16
nTimeStamp	uint32
hashPrev	uint256
hashMerkle	uint256
vchProof	vector<uint8>
txMint	CTransaction
vtx	vector < CTransaction >
vchSig	vector<uint8>

Explanation:

- The current block version is 0x0001.
- The timestamp is in UTC in seconds.
- vchProof includes serialization data of legality proof.In the security

main chain, it includes the calculation results broadcast by EDPoS nodes (including the signatures of each node) .The CPoW block also includes the workload proof parameters.In the application branch chain, the same height is included. Main chain block hash and consensus calculation results.

txMint is not signed and the signature field is empty.

The block signature vchSig is signed using the txMint output address.The signature data segment contains all fields except vchSig.

transaction

HRD uses utxo model to record transactions, including the following data:

Elem	Type
nVersion	uint16
nType	uint16
nLockUntil	uint32
hashAnchor	uint256
vInput	vector<CTxIn>
addrTo	CDestination
nAmount	int64
nTxFee	int64
vchData	vector<uint8>
vchSig	vector<uint8>

Explanation:

- The current transaction version is 0x0001.
- hashAnchor is used to indicate the current transaction starting valid block and the corresponding branch.
- Preorder transactions in the input list require the same output address.
- The transaction includes two outputs, one is listed in the table (addrTo / nAmount), and the other is the implicit change output, the address is the same as the input address, and the amount is (Total Input- nAmount- nTxFee).
- The input address of the transaction signature is unified, and the signature data segment contains all fields except vchSig.

HRD cross-branch trading

Cross-branch transactions can be used to achieve synchronous value exchange in the case of no trust between HRD branches. In practical applications, business can often be divided according to business processes, equipment types, spatial regions and other related factors and dispersed into multiple branches. Frequent devices usually hold the same branch token and conduct data transactions in the same branch. However, as a business as a whole, the need to interact with

other branch token devices also exists objectively. In this case, cross-branch transactions can realize the support of transactions. Token exchange between chains. On the one hand, cross-branch transactions can be completed in a trustless situation, using technology principles to ensure fairness to both parties; on the other hand, cross-branch transactions are synchronized into the block between the two branch chains, ensuring that High efficiency and effectiveness. This provides good underlying technical support for applications including decentralized exchanges, token exchange gateways, and more.

Consensus mechanism

As mentioned above, the consensus mechanism adopted by the HRD system is EDPoS + CPoW, led by EDPoS, which decides the node that will get the block right next time or indicates that the next block is generated by the workload proof consensus. The EDPoS mechanism was not effectively established In the initial stage, for example, CPoW becomes the only consensus mechanism for block generation.

The consensus mechanism is explained in detail below.

EDPoS node negotiation process

EDPoS nodes use the number of tokens they hold as the weight of block production, and generate a series of fixed block generation nodes through random number calculation. After the EDPoS mechanism is established, random numbers are generated through negotiation between EDPoS nodes. The negotiation process is performed once every minute, which can be verified through weighting. Key sharing (VSS) and Byzantine fault tolerance for fair and

random calculations.

Each round of negotiation includes the following processes: 1. node registration; 2. encrypted shard data distribution; 3. secret shard publication; 4. data reconstruction and random beacon calculation.

Before each round of negotiation, each EDPoS super node needs to use the ECC algorithm to generate a set of private keys: {a0, a1, ..., at-1}, and the corresponding public keys: {A0, A1, ..., At-1}, Satisfy Ai = aiG,

(i = 0,1, ..., t-1). t is the threshold of the reconstructed data. According to the setting of the effective EDPoS super node, the maximum value of t is 50.

Node registration

The EDPoS super node broadcasts the registration information on the chain before the 16 blocks of the target block are negotiated in this round, including the encrypted polynomial coefficients {A0, A1, ..., At-1}, A0 as the node's public key for negotiation.

Encrypted data distribution

The distribution starts 16 blocks before the negotiation of the benchmark block and ends at the previous block. At the beginning of the distribution process, the calculation sequence number is allocated

according to the order and weight of the registered nodes, and the calculation method of the calculation sequence number assigned to each node is:

[token vote / (total supply X 2%)]

Node i creates a common key Kij based on the negotiated public key issued by other node j, encrypts the secret segment sij with Kij, and broadcasts the entire network. After decryption of corresponding node j, Sij can be verified with the registration information of node i.

Sij Calculation:

$$s_{ij} = \sum_{k=0}^{k<t} a_k \times j^k$$

Since the encryption public key {A0, A1,..., At-1} of node i has been published during the registration process, node j passes the followingCheck:

$$s_{ij} \cdot G = \sum_{k=0}^{k<t} j^k \times A_k$$

If the above formula holds, node i sent the correct secret fragment.

Secret shards announced

After the current block broadcast, each node will broadcast the entire network through all the secret shards that have been verified.The

nodes on the entire network can collect the decrypted node shards, which can also be verified by the above formula. Calculate valid data.

Data reconstruction and random beacon calculation

The t-secret shards of node i in the whole network can be reconstructed by Lagrange's equation

{a0, a1,..., at-1}, t nodes that cannot collect secret shards that have been verified will be removed, and they cannot enter the next stage of calculation. Repeat the above calculation process, and finally get all valid node data. This In the process, the calculation results of all reliable nodes will be the same. Through the combined calculation, a uniform beacon will be obtained across the network. Since the data used for calculation is provided randomly by each EDPoS node, it is impossible to know the other until the last step of the calculation. The data of the nodes. Cheating nodes will be removed during the checksum reconstruction phase. Without considering the 51% attack, no node can control the final calculation result, so it can be considered that the generated random beacons have true random properties.

Block allocation

In the case that the EDPoS mechanism has not been effectively established, the current block is generated by the CPoW consensus. When the EDPoS negotiation is successfully completed, the random beacon will be used to roll the dice. Assuming that the token of the EDPoS super node i is Vi, the total EDPoS Voting Vd = V0 + V1 + ... + Vn, the total token supply is S.

CPoW equivalent vote is Vwork = S * (1-Vd / S) ^ 3 Probability that node i gets block weight Pi = Vi / (Vd + Vwork)

CPoW get block weight probability Pwork = Vwork / (Vd + Vwork)

Repeat the above process to get the determined sequence of block generation. According to the sequence of block generation, the corresponding node completes the block generation of the current block, and records the final calculation process of the decrypted negotiation into the block to prove the legitimacy of the block generation.

After the EDPoS negotiation, the determined block generation series can be calculated and verified by all nodes. In addition to specifying the determined node for block generation, there is a certain probability that CPoW will be designated to generate blocks. As shown

in the above formula, the probability of the CPoW block being selected and the total participation in the EDPoS negotiation. Token quantity related:

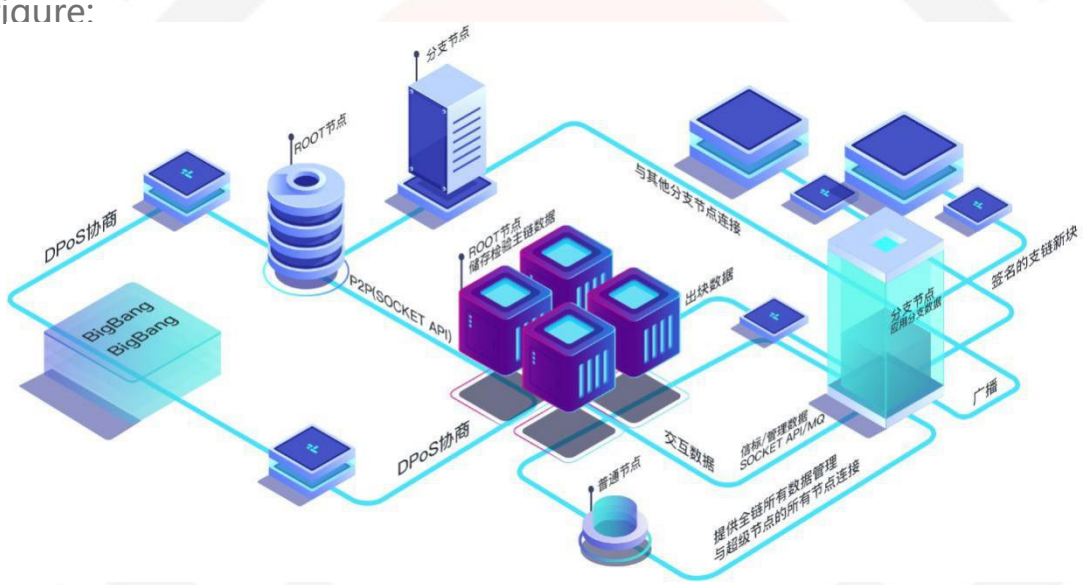
V_d / S	P_{work}
0	100%
0.25	62.8%
0.5	20%
0.75	2%
1	0%

In the initial stage, there are fewer nodes and Tokens participating in EDPoS, and the consensus mechanism is degraded to be CPoW-based.When more and more Tokens participate in the EDPoS process, the probability of CPoW generating blocks will rapidly decrease.

Network Function Unit

Network node

HRD's network nodes run core node programs for block data generation and data update synchronization between different nodes. The composition of core node programs is shown in the following figure:



The bottom layer is composed of a series of basic libraries and tool classes, providing low-level program interfaces including data storage, database access, security algorithms, application frameworks, p2p network / http, etc.

The middle layer includes block data / real-time transaction management, user wallet, and block generation structure.They are

used to verify and manage specific branch chain blocks / transaction data, built-in user wallet keys and user transaction management, and EDPoS / CPoW consensus. Block production. The upper layer P2P network layer implements the node network protocol. On the one hand, it manages the scheduling and other nodes' network connections and data requests; on the other hand, it synchronizes and exchanges data with the middle layer through the data distribution interface.

The core node program interacts with external services in two ways to provide node control and function expansion through JSON-RPC and Socket API.JSON-RPC is mainly oriented to human-computer interaction applications such as RPC client programs, wallet graphical interfaces, and wallet node management. The Socket API provides a high-speed data synchronization channel for light wallet services and distributed node deployment. These two external interfaces are aggregated by API SERVICE to provide core functions.

Light wallet service and client

LWS is the abbreviation of light wallet service, which is a bridge between the HRD public blockchain backbone network and terminal

data acquisition sensor equipment. Through it, the blocks and transaction data of the HRD core wallet are updated and cached in the LWS in a timely manner. Some high-speed memory databases and local databases.

Based on these data, it calculates the latest UTXO set of public key addresses corresponding to the keys held by different terminal devices, and publishes this information to the Amazon cloud facility through the mqtt connection to AWS 'IoT Core. Its message broker forwards to the corresponding terminal device that subscribes to this information. Correspondingly, the terminal device will package the data into the transaction after obtaining the data collected by the monitoring according to the UTXO list related to itself. mqtt is published to Amazon IoT Core.

Through the latter's message broker, it is pushed to the LWS that subscribes to these devices to send transaction topics.The LWS will verify these transactions.If the verification is successful, it will forward these transactions to the HRD core wallet through the Socket API. The P2P network interface broadcasts these transactions to the entire HRD network, and the block-producing node collects these transactions, and finally completes its operation of packaging the block chain.

The HRD client program, as part of the firmware of the IoT device,

uses the device central processor and secure computing coprocessor to process HRD transaction related calculations including transaction construction / analysis, HASH, ED25519 signature / verification. The device private key is stored to process the chip security Area, which cannot be read directly.

The utxo list synchronization and real-time update of the client are achieved through a protocol process between the lws and the HRD client.The lws responds to the HRD client to send a transaction request and broadcasts the transaction to the entire network through the connected network node.

The complete interaction process between an IoT terminal sending a transaction and LWS is as follows:

Service Req / Reply: The HRD client initiates a service request, passing information such as the protocol version, wallet address, and required branches; LWS returns the data used to construct the APIKey and a list of supported branches under the premise of providing services; the APIKey is used Sign the subsequent message.Sync Req / Reply: The HRD client initiates a synchronization request and passes the current record UTXO list HASH; the LWS compares the UTXO list HASH and pushes the corresponding UTXO list if it determines that the client is

out of sync.

Update UTXO: When the Block / Tx status is updated (new block generation, transaction broadcast), the network node will notify the LWS through the Socket API, and the LWS will push the UTXO status change to the HRD client after filtering.

SendTx Req / Reply: The HRD client constructs and signs the transaction according to the synchronized UTXO list, and broadcasts the transaction to the entire network through LWS; LWS returns the execution status.

Distributed supernode

Program's main objectives

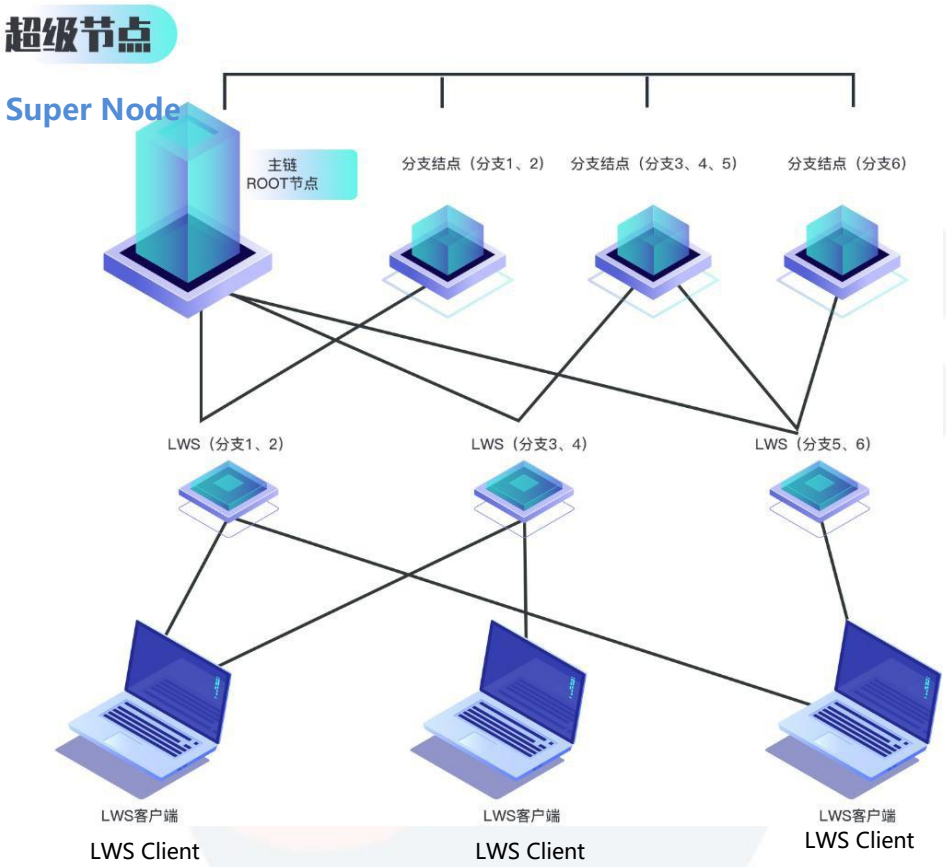
The main goal of distributed super nodes is to solve the scalability problem of EDPoS nodes of the main chain. As a multi-branched chain block system, the nodes participating in EDPoS need to synchronize blocks for all valuable branch chains. As the number of chains increases, the burden on the block-producing nodes will increase, and a single server cannot meet the scalability requirements of the system itself.Therefore, distributed super nodes are an effective solution to this problem.

An Introduction

In the distributed super node solution, the main roles are divided into root node server and branch node server according to different business.

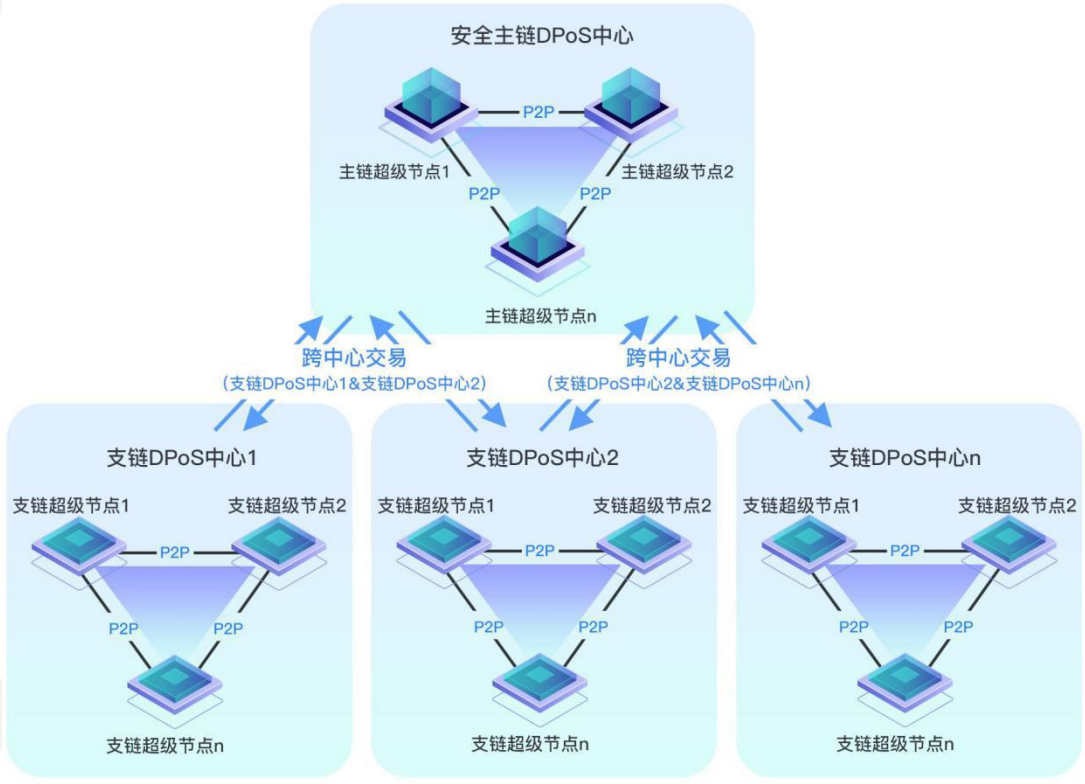
The ROOT node server is mainly responsible for consensus negotiation, main chain block generation, main chain data management, branch node management, etc .; the ROOT node server stores the relevant data of the block signing key, which is used for consensus negotiation and main chain block generation; ROOT The node server accesses the HRD network, on the one hand, it conducts EDPoS negotiation on the secure main chain, and on the other hand, it interacts with other branch nodes in the super node; on the other hand, the ROOT node server and the ROOT node server of other super nodes are connected through P2P (Socket API). Or connect with ordinary nodes through P2P (Socket API). The ROOT node server only stores the data to verify the security main chain, and the application branch data is processed by the branch node server. The branch node server and the ROOT node server are connected and interacted through the Socket API or MQ method. Security beacons, management data, and other information. Branch node servers are specifically used to organize application block data for branch chains. Each branch node server is

only responsible for one or more branch chain data, and there cannot be two branch node servers responsible for the same. Branch chain data.



The branch node server connects to the ROOT node server through the Socket API or MQ method, and receives and processes security beacons and management data. The branch node server connects to the corresponding branch nodes of other super nodes through P2P (Socket API) (the ROOT node server connects with the peer The ROOT node server obtains the connection address of the opposite branch node during the handshake negotiation and distributes the connection address to the branch node.) When EDPoS negotiates that

the current ROOT node is selected as the block-producing node, the ROOT node server constructs a new block of the main chain, and The branch chain output block context data is pushed to each branch node server and the branch node server constructs the new branch block data of each branch and signs it (the branch node stores the data related to the block signing key), and the branch node broadcasts the entire network.



HRD cross-chain template

The implementation of the HRD cross-chain template and e-commerce template does not require VM compilation, so the template runs very efficiently compared to smart contracts and scripts. At the same time, HRD transactions run fast and securely, and can be attacked without the same VM. To prevent stolen or zero tokens on the chain due to vm vulnerabilities. However, for non-smart contracts, the version update process is a bit cumbersome. After the program is released, it needs to be updated to each client synchronously, and a template automatic update module will be added at a later stage.

Cross-chain transaction template

- Create a transaction template that contains the wallet addresses of the

two parties and the two branch chain hashes participating in the transaction, as well as the block heights locked on the respective branch chains, and generate a template address for the transaction from these data.If any of these data values change, then The template address will also change. This address can be used to receive any token transferred by the user, but within the lock height, to transfer the token under the template address, the signature data of both parties can be completed. In the lock block After high, both parties involved in the transaction can transfer the Token on their own branch chain.

- Sign the cross-connect transaction template address to get the signature data.

- The two parties to the transaction can verify the signature data is correct through the public key provided by the other party, and confirm whether the signature data is the signature of the corresponding cross-chain transaction template. Users who require a low locked block will first send their signature results to the other party.

- Through the signature data provided by the other party and their own signature data, the Token in the cross-connect transaction template

can be transferred away.At this time, the user's wallet needs to be open.

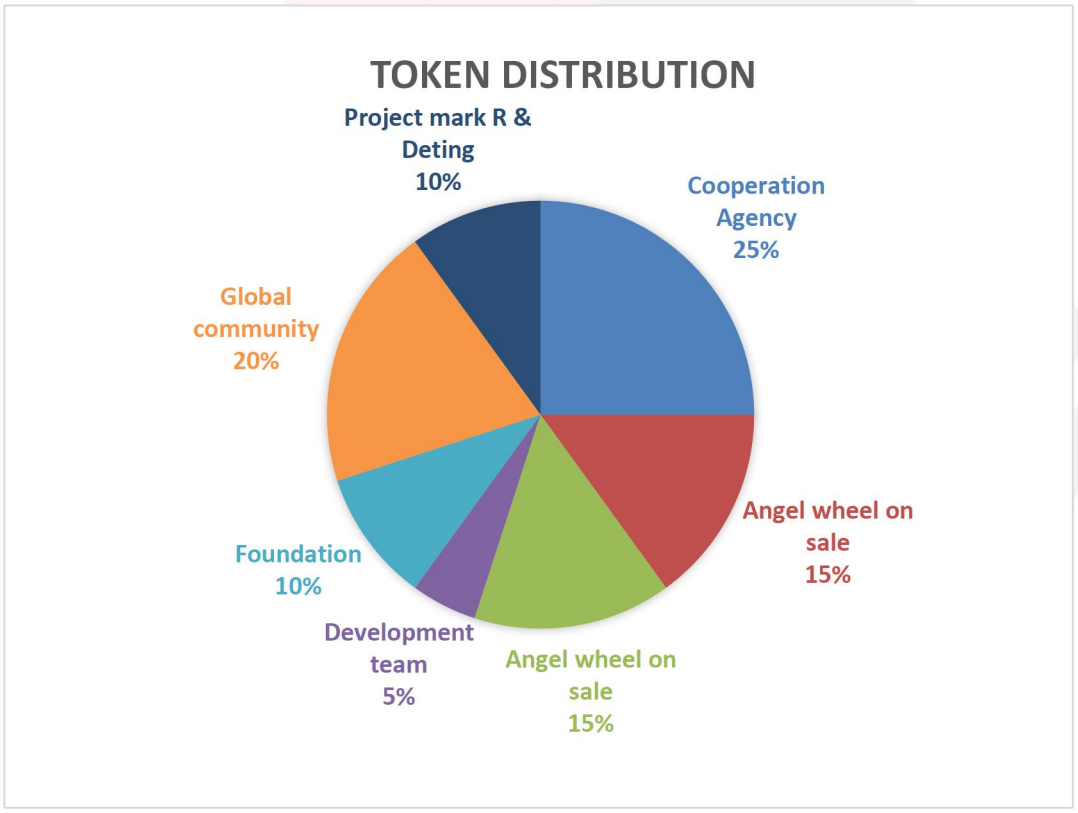
- If you do not transfer the Token within the "Trader's Priority Packing Height" after the transaction is issued, the third party will be involved in the packaging process. The third party can be an ordinary node or a super node.You can use the From and To addresses and the signature data in the template. Help with packaging and get third-party packaging rewards.
- Note: The cross-chain transaction template is a one-time use template.The code is further upgraded and can be modified into a reusable cross-connect transaction template.

HRD Token Distribution Mechanism

In order to promote the development of the HRD ecology from the incentive level, HRD has constantly issued 2 billion tokens, which is based on the decentralized digital assets issued by Ethereum.The token is abbreviated as HRD.

Token issuer: Hong Kong Fushi Group Block chain Business Department.

The HRD token allocation ratio is as follows:



Disclaimer and Risk Statement

Disclaimer

Except as expressly stated in this white paper, HRD developers do not make any representations or guarantees (especially for their marketability and specific functions). Anyone participating in the HRD project is based on their own knowledge of HRD and this White Paper The HRD developer hereby expressly does not recognize and refuses to assume the following responsibilities:

1.Anyone who violates the anti-money laundering, anti-terrorist financing or other regulatory requirements of any country when participating in the HRD project;

2.Anyone who violates the requirements or obligations imposed by this white paper when participating in the HRD project, and the resulting failure to pay or withdraw HRD coins;

3.The development of HRD failed or was abandoned, and the resulting failure to deliver HRD coins;

4, the delay or extension of HRD development, and the resulting failure to reach the schedule of prior disclosure;

5.Errors, flaws, defects or other problems in the HRD source code;

6.HRD or HRD coins fail to achieve any specific function or are not suitable for any specific purpose;

7.Failed to disclose information about HRD development in a timely and complete manner;

8.Any participant has leaked, lost or destroyed the private key of the wallet of the digital cryptocurrency or token (especially the private key of the HRD coin wallet used by him);

9.The default, violation, infringement, collapse, paralysis, termination or suspension of service, fraud, misoperation, misconduct, negligence, negligence, bankruptcy, liquidation, dissolution or closure of the third party crowdfunding platform of HRD Coin

10.There is a difference, conflict or contradiction between the agreed content between anyone and the third party crowdfunding platform and the content of this white paper;

11.Anyone's trading or speculation on HRD;

12.The listing or delisting of HRD Coin on any exchange;

13.HRD currency is classified or regarded as a currency, securities, commercial paper, negotiable instruments, investment products or other things by any government, quasi-government agency, competent authority or public agency, which is prohibited, regulated or legal limit;

Security and management of funds

The funds received by the project should be kept and operated in accordance with the principles of transparency, auditability and efficiency. The platform's profits are kept by multi-signature wallets and reviewed by the public. For security issues, the private keys of these multi-signature wallets are worth five Trusted personal control. Any payment performed by the wallet requires the simultaneous signature of these five people. The funds received by the platform will not be used for shareholder dividends or profit distribution by the HRD developer, but will be used for HRD development, maintenance, etc. Technical work and ecosystem construction of HRD (such as investing in various applications on HRD, etc.).

Risk disclosure

There are risks in the development, maintenance, and operation of HRD, many of which are beyond the control of the developer. In addition to the other content described in this white paper, each participant of HRD should read, understand and carefully Consider the following risks before deciding whether to participate in the platform project.

Participation in this platform project should be a deliberate decision-making action, and it will be deemed that the participants have fully understood and agreed to accept the following risks:

1.The risk that HRD cannot be developed or used normally due to changes

in laws and policies or government actions, or that HRD coins are prohibited from being held or used;

2.Due to the development of cryptography or the commercialization of quantum computers, the risk that cryptocurrency-based currencies no longer have sufficient security (for example, private keys can be easily cracked);

3.The risk of development failure due to the higher difficulty of HRD technology development;

4.The risk of eth or btc obtained by this platform project, resulting in the lack of financial support for HRD development and the risk of unsustainability;

5.The source code of HRD has the risk of various faults in the HRD operation process caused by flaws, defects and vulnerabilities;

6.The source code of HRD is upgraded or modified based on community requirements, resulting in unpredictable risks;

7. The risk of HRD being "distributed denial of service" or other types of attacks during operation;

8.The risk that the HRD coins held by anyone will be stolen, forgotten or lost;

9.The HRD currency lacks a secondary trading market, the price is unstable, or there is no risk that others are willing to buy the HRD currency;

10.The development and operation of other blockchains with similar

functions or competitive relationships with HRD, so that HRD is marginalized or out of the market;

11.Risks caused by faults and defects in various applications on HRD developed by third parties.