QLC CHAIN

A Multidimensional Block-Lattice Public Chain with Smart Contract Support for Network-as-a-Service

Co-author: Allen Li, Chris Zhao
Contributor: Toya Zhang, Terry Ke, Alen Breznik, Bart Coelus

Abstract

We are introducing QLC Chain, a next generation public blockchain dedicated to building Network-as-a-Service (NaaS) solutions. Among others, the three most important features of QLC Chain are: 1) Multidimensional Block Lattice structure for multiple tokens issuance; 2) Smart Contract support; 3) Dual consensus model for distributed transaction validations. In addition to provide a fast, scalable and extendable network, **QLC Chain enables everyone to operate network services and benefit from it.**

Contents

1. QLC Chain Introduction

QLC Chain is a next generation public blockchain designed for the NaaS. It deploys a multidimensional Block Lattice architecture and uses virtual machines (VM) to manage and support integrated Smart Contract functionality. Additionally, QLC Chain utilizes dual consensus: Delegated Proof of Stake (DPoS), and Shannon Consensus, which is a novel consensus developed by the QLC Chain team. Through the use of this dual consensus protocol, QLC Chain is able to deliver a high number of transactions per second (TPS), massive scalability and an inherently decentralized environment for NaaS related decentralized applications (dApp). The framework of QLC Chain will enable everyone to operate network services and benefit from it.

Network-as-a-Service (NaaS) is sometimes listed as a separate cloud provider along with Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). This factors out networking, firewalls, related security, etc.

NaaS can include flexible and extended Virtual Private Network (VPN), bandwidth on demand, custom routing, multicast protocols, security firewall, intrusion detection and prevention, Wide Area Network (WAN), content addressing and filtering, and antivirus.

1.1 Different Network Node Functions in QLC Chain

- **NAT function:** Network Address Translation Node
- **Routing function:** Route forwarding node based on content keyword/DHT/Router table
- **Storage function:** A node with saved content, which can provide contents based on retrieval request from other nodes within the network
- **Security function:** Performs firewall function and enacts security domain access rule

1.2 QLC Chain Architecture

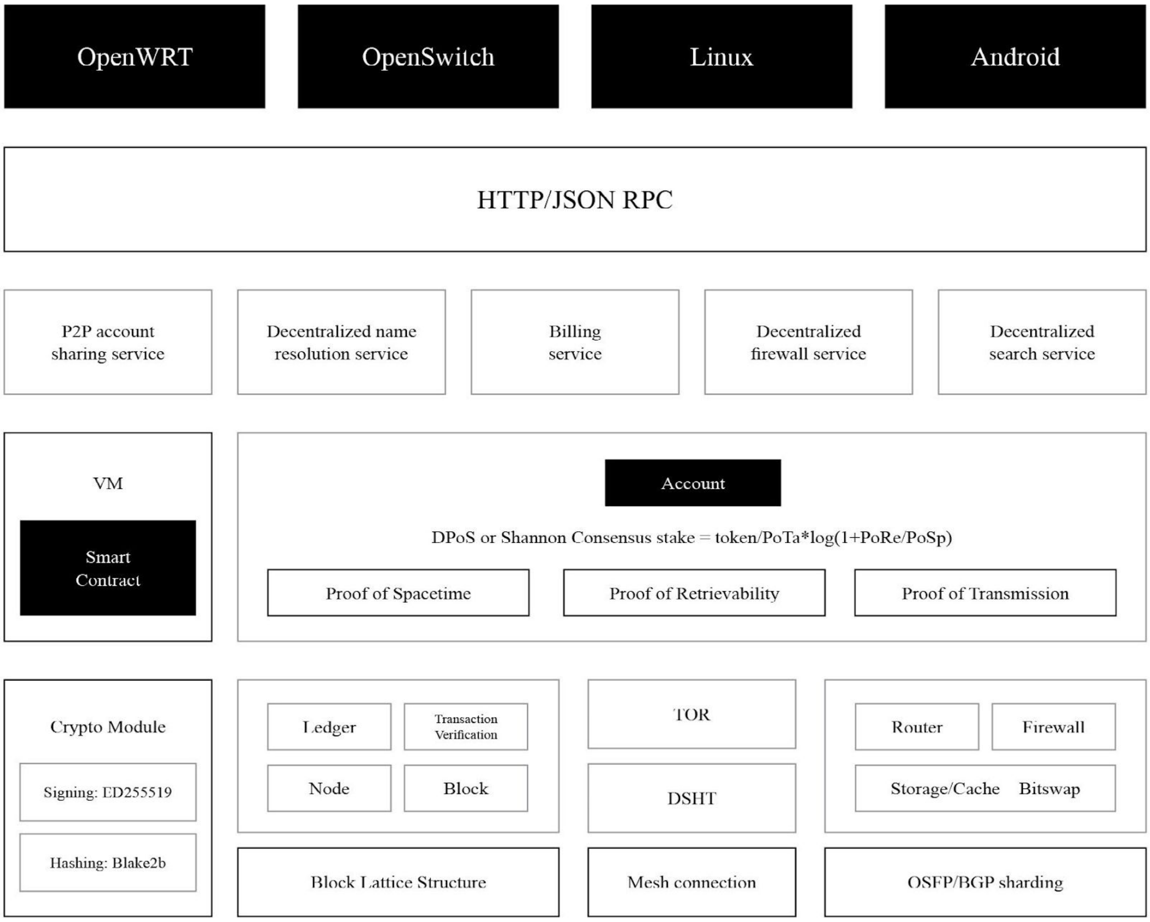


Fig. 1. QLC Chain Architecture

This paper serves to explain the details of QLC Chain, including technical structure, features and advantages.

2. Background

2.1 Block Lattice

A Block Lattice is a block architecture that was first introduced by the Nano cryptocurrency. With Block Lattice, each individual transacting account on the network possesses their own blockchain, which is controlled by the account's private keys. Under the Block Lattice structure, user's blockchain tracks the account balance, rather than a transaction amount, which allows for less intensive storage and faster transaction speed.

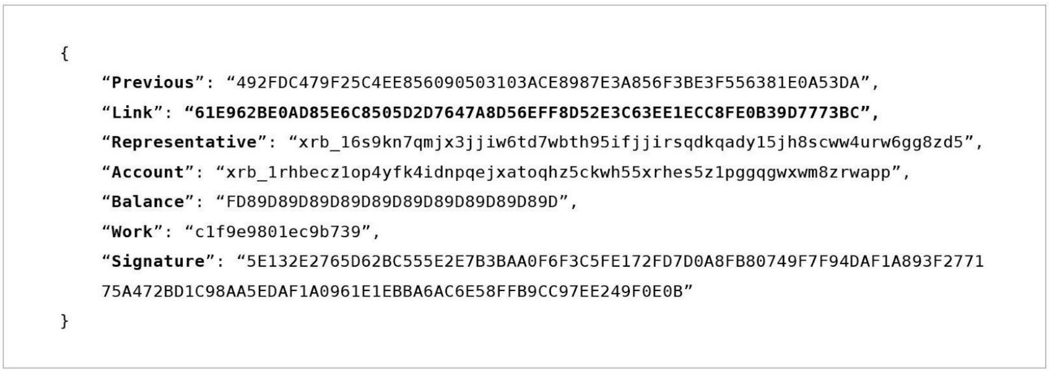
There are 4 types of Blocks in Nano: OPEN, SEND, RECEIVE and CHANGE. They are used for recording transactions. Balances are transferred between blockchains through SEND and RECEIVE blocks.

Nano further introduces the Universal Block to consolidate four types of blocks and encode all account data in every transaction. This design increases the efficiency, improves the security and simplifies codes needed in the network. There are two major improvements:

- Signature checking performance improvement: Before the change, SEND and RECEIVE do not include the account signature, so that block signatures has to be verified while running the main ledger insert process and Input/Output (I/O) operations is blocked for finding associated account. This is very time consuming. When the Universal Block contains the account information, unlimited number of block signatures can be verified without blocking on any I/O operations. Although it increases the block size, the overall performance through TPS and lower transaction processing latency is an acceptable tradeoff.

- Efficient Balance Lookups: The absence of account balance in the current design also occupies long I/O operations. With the implementation of Universal Blocks, we can know the balance simply by looking at one block, instead of searching down the chain for the last SEND Block.

The following illustrates the structure of a Universal Block:



```
{
    “Previous”: “492FDC479F25C4EE856090503103ACE8987E3A856F3BE3F556381E0A53DA”,
    “Link”: “61E962BE0AD85E6C8505D2D7647A8D56EFF8D52E3C63EE1ECC8FE0B39D7773BC”,
    “Representative”: “xrb_16s9kn7qmjx3jjiw6td7wbth95ifjjirsqdkqady15jh8scww4urw6gg8zd5”,
    “Account”: “xrb_1rhbecz1op4yfk4idnpqejxatoqhz5ckwh55xrhes5z1pggqgwxwm8zrwapp”,
    “Balance”: “FD89D89D89D89D89D89D89D89D89D89D”,
    “Work”: “c1f9e9801ec9b739”,
    “Signature”: “5E132E2765D62BC555E2E7B3BAA0F6F3C5FE172FD7D0A8FB80749F7F94DAF1A893F2771
    75A472BD1C98AA5EDAF1A0961E1EBBA6AC6E58FFB9CC97EE249F0E0B”
}
```

Fig. 2. Structure of Universal Block

Compared with the traditional blockchain architecture used by Bitcoin and Ethereum, Block Lattice delivers almost instantaneous transaction speed and unlimited scalability on low-power consuming hardware, which is highly suitable for network transmission. Block Lattice technology has proven to be stable in its more than two-year operational history. With peak transaction speeds of 7000 TPS and more than 500,000 users on more than 700 consensus nodes, this technology has outperformed most known blockchain technologies.

QLC Chain further advanced the structure by introducing multidimensional Block Lattice to support the Smart Contract and new consensus algorithm especially for network transmission services. QLC Chain team continues the development and aims to build one of highest performing network protocols for the future.

2.2 Network-as-a-Service (NaaS)

Network-as-a-Service (NaaS) is a separate traditional cloud provider along with Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). This family is depicted by following picture.

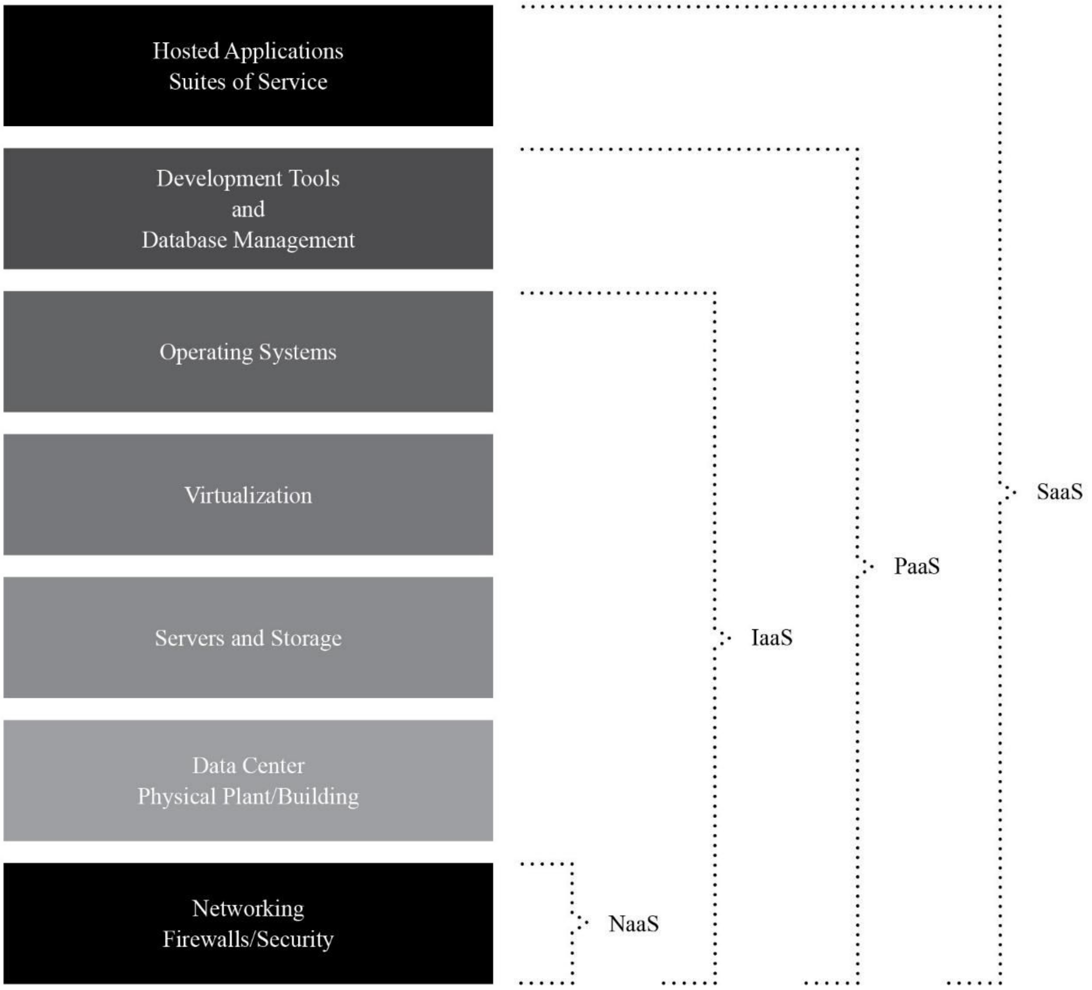


Fig. 3. IT Service Layer

NaaS can include flexible and extended Virtual Private Network (VPN), bandwidth on demand, custom routing, multicast protocols, security firewall, intrusion detection and prevention, Wide Area Network (WAN), content addressing and filtering, and antivirus.

3. QLC Chain Components

General Account:
Each public key and private key pair constructs an account. Each account has its own blockchain recording all engaged transactions. Account owner has authoritative control over transactions related to this account.

External-owned Account:
When an account owner issues one Smart Contract, the General Account becomes an External-owned Account and simultaneously generates an Internal-owned Account. The External-owned Account holds the ownership of all assets related to this Smart Contract and is able to issues more than one Smart Contracts.

Internal-owned Account:
Internal-owned Account is generated because of the issuance of a Smart Contract from the External-owned Account. This new account shares the same feature as the General Account, including Send and Receive. Sending the Public Token to the account will activate an Internal-owned Account and trigger the smart contract execution. The transaction will be recorded in a Transaction Block under the Sender Account and the Smart Contract Account.

Transaction Block:
General Block recording transactions between general accounts.

Smart Contract Block:
On top of Transaction Block, Smart Contract Block Stores Smart Contract Instance.

Transaction Ledger:
The ledger that records general transactions.

Asset Ledger:
Asset Ledger is used for network asset registration and for recording the asset exchange.

Node:
A piece of software running on a computer that conforms to the QLC Chain protocol and participates in the QLC Chain network.

Storage Node:
New added type of node for the input / output data storage in the Smart Contract Instance.

QLC Virtual Machine:
A virtual machine exclusively for QLC Chain to compile the Smart Contract into an ABI and to provide a secure environment for deployment.

4. QLC Chain: Features and Benefits

4.1 Multidimensional Block Lattice Structure

In Block Lattice structure, every account has a unique blockchain to record its own transactional information. With Smart Contract functionality, QLC Chain supports multiple tokens issuance within one account. Each account supports multiple tokens and each new token added will be mapped to a new chain within the same account, so that each account can have multiple chain. Each token has its own “OPEN Block” in every single account. Since one token/one chain is one dimension, the structure with multiple tokens creates a multidimensional Block Lattice.

Each blockchain for identical token is independent from others. The underlying structure of each token blockchain carries the Block Lattice structure and thus stays concise and agile.

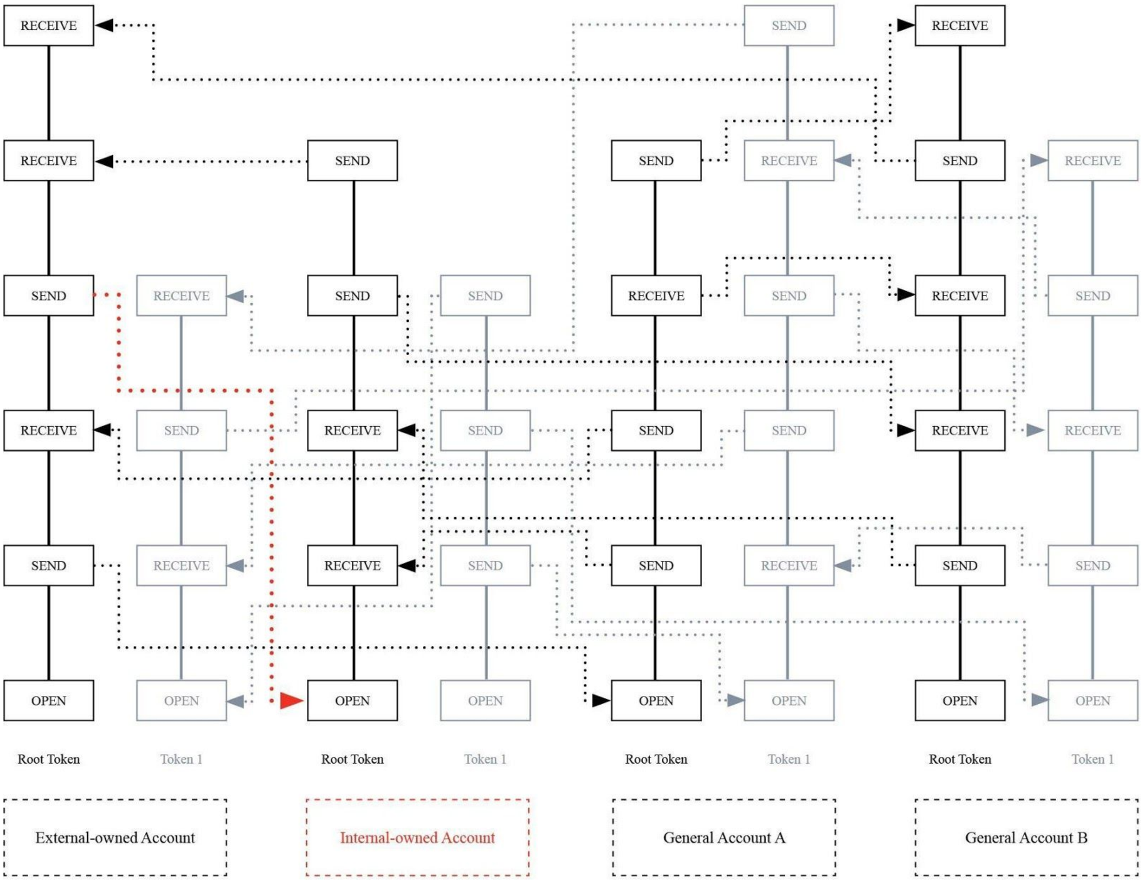


Fig. 4. Multidimensional Block Lattice Example

Multidimensional Block-Lattice structure brings QLC Chain the following benefits:

Low Transaction Validation Latency

The use of independent account-chains enables the user accounts to be updated asynchronously, without the need to involve the entire network. The dual-transaction approach leaves the process of transaction verification to the affected accounts, such as the sender and the receiver. This option eliminates the need for miners, meaning that transactions are instant and with zero fees. The network, therefore, becomes more agile.

Scalability

The QLC Chain structure allows transactions to be handled independently of the main ledger. Every transaction is also an independent block that fits into a UDP transactional packet, recorded as a unique block. This aspect eliminates issues relating to block size since nodes don't have to keep a record of all the transactions on the network. Rather, the nodes store the current blocks of every account-chain.

Therefore, by not keeping a record of the entire blockchain history, the network avoids scalability issues. This is where Block Lattice differs from mainstream blockchain. With the blockchain, a single transaction cannot be isolated and recorded on the main chain. A select number of transactions must be fitted into the block before being verified and then added to the main chain. This means increased transactions lead to a steady decline in speed, slowing down the entire network. QLC Chain use of account chains helps maintain a lighter network, reducing problems of scalability that plague blockchain.

Low Energy Consumption

The QLC Chain is built upon a dual consensus architecture: Delegated Proof of Stake (DPoS) and Shannon Consensus. Both consensuses can achieve low energy consumption because none of them requires mining activity. All energy is contributed to make effective computing. Both consensuses will be elaborated later in this paper.

Inherent Anti-Centralization

Mechanism guaranteed anti-centralization refers to the fact that each account has its own ledger, namely, the account-chain structure, and the validation conducted by delegates via an asynchronous mode. Unlike the Proof of Work (PoW) consensus used by Bitcoin, whereas the ledger generation and confirmation is completed by miner nodes; nor the Proof of Stake (PoS) synchronized mode whereas ledger generation and confirmation is completed by nodes with large number of tokens.

In addition, the structure of the anti-centralized Block Lattice requires that the transaction sender and receiver to conduct a small computational effort input - local PoW process. This process has decreased the possibility of transaction centralization, similar to how a decentralized exchange decreased the possibility of super account formation.

Another important mechanism guaranteed anti-centralization factor is Shannon consensus of the QLC Chain, which will be introduced later.

4.2 QLC Chain Smart Contract

QLC Chain enables Smart Contracts and Distributed Applications (dApps) to be built on Block-Lattice, which brings the advanced structure beyond just supporting a digital currency. By introducing Smart Contract, QLC Chain defines Smart Contract as an account which owns its account-chain and designs SMART CONTRACT Block.

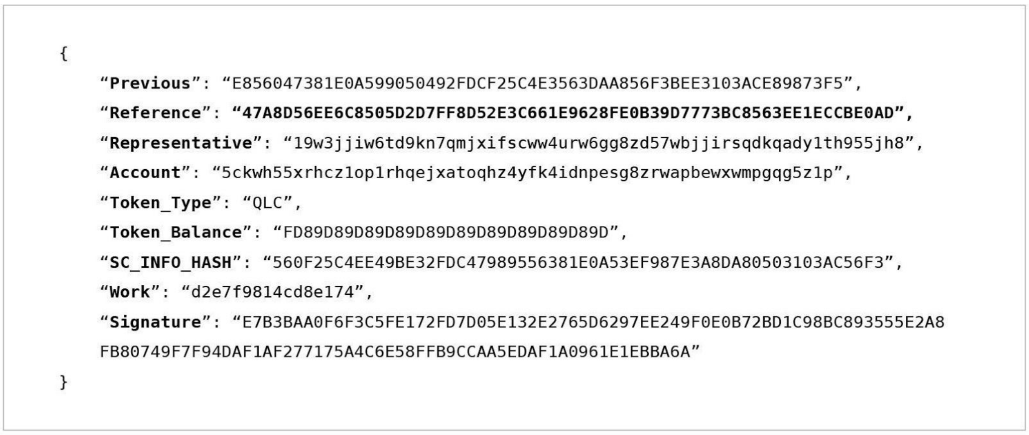
QLC Chain supports two types of Smart Contract: the Token Smart Contract - for new token issuance in the ecosystem, and the Asset Smart Contract - for digital asset registration without new token generated.

4.2.1 Explanation of SMART CONTRACT Block

QLC Chain Smart Contract contains two parts: Contract handle for Smart Contract addressing, and contract instance for saving ABI and contract signature.

The structure of block is illustrated as following:

- Smart Contract Handle: “SC_INFO_HASH” is null for ordinary transaction block. In the transaction concerning the Smart Contract, this field must not be null. The combination of “link” and “SC_INFO_ HASH” can accurately classify the Smart Contract related transaction from ordinary transaction.



```
{
    “Previous”: “E856047381E0A599050492FDCF25C4E3563DAA856F3BEE3103ACE89873F5”,
    “Reference”: “47A8D56EE6C8505D2D7FF8D52E3C661E9628FE0B39D7773BC8563EE1ECCBE0AD”,
    “Representative”: “19w3jjiw6td9kn7qmjxifscww4urw6gg8zd57wbjjirsqdkqady1th955jh8”,
    “Account”: “5ckwh55xrhcz1op1rhqejxatoqhz4yfk4idnpesg8zrwapbewxwmpgqg5z1p”,
    “Token_Type”: “QLC”,
    “Token_Balance”: “FD89D89D89D89D89D89D89D89D89D”,
    “SC_INFO_HASH”: “560F25C4EE49BE32FDC47989556381E0A53EF987E3A8DA80503103AC56F3”,
    “Work”: “d2e7f9814cd8e174”,
    “Signature”: “E7B3BAA0F6F3C5FE172FD7D05E132E2765D6297EE249F0E0B72BD1C98BC893555E2A8
    FB80749F7F94DAF1AF277175A4C6E58FFB9CCAA5EDAF1A0961E1EBBA6A”
}
```

Fig. 5. Transaction Block

- Smart Contract Instance: SC_INFO_HASH is the hash value from the Smart Contract Block illustrated in Fig. 5, which records the original data of Smart Contract ABI.

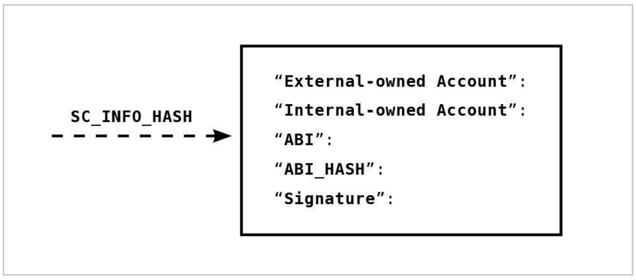


Fig. 6. Smart Contract Block (SC_INFO Block)

4.2.2. Properties of the SMART CONTRACT Block

The SMART CONTRACT Block used in QLC Chain has the following properties:

- The owner of the Smart Contract activates the account by sending Root Token to the Smart Contract account.
- By executing the file in the contract, the owner signs the block with his private key.
- The SMART CONTRACT Block is reserved for an ABI. The QLC Chain Virtual Machine compiles the Smart Contract into an ABI and further provides a secured deployment for the Smart Contract.
- The virtual machine further retrieves the ABI by loading the SMART CONTRACT Block and calling the function of the Smart Contract.
- The consensus protocol of the Smart Contract block is completely the same as which in the transaction block.
- Extend Smart Contract Block to support the Smart Contract related data storage so that the Asset Smart Contract can be saved.

4.2.3 Token Smart Contract Protocol

- The External-owned Account sends SC_INFO to other nodes.
- External-owned Account sends a SEND Block containing Publik Token transaction to activate the Internal-owned Account. Nodes in the network will verify the Smart Contract based on the SC_ INFO_HASH in SEND.
- After confirming the SEND Block, External-owned Account issues the OPEN Block in the Internal-owned Account chain and the Genesis Block of token by utilizing “Init()” in the Smart Contract and consequently broadcasts to the entire network.

- Every node updates the local account information based on the broadcast and calculates the balance of Root Token for Internal-owned Account and the balance of the new token. Thus far, the process of issuing a Token Smart Contract is completed.

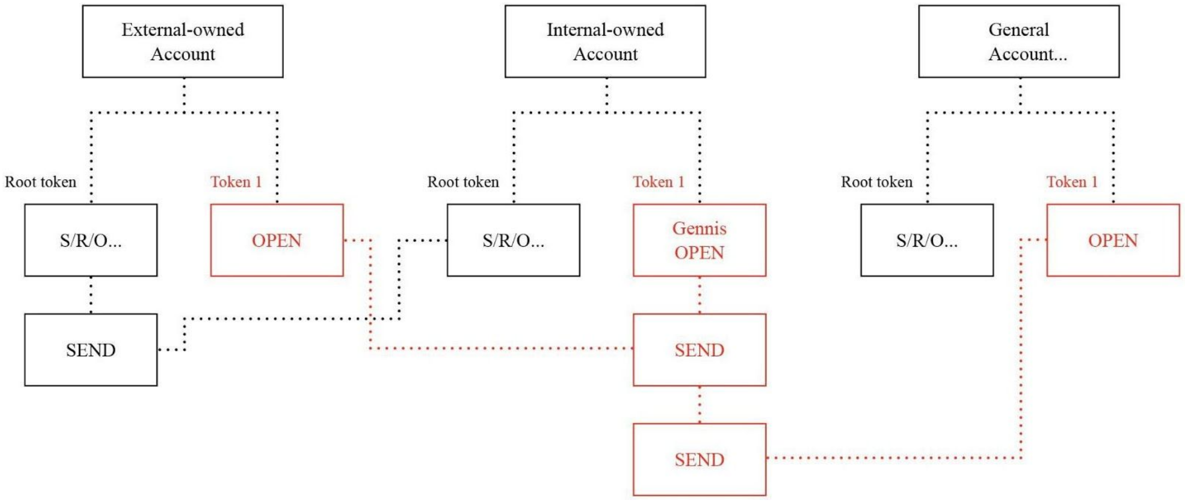


Fig. 7. Release Procedure of a Token Smart Contract

4.2.4 Asset Smart Contract Protocol

Asset Smart Contract is applicable to assets with service attributes, such as communication equipments provide communication services, automotive provide transportation services, and so on. After the Asset Smart Contract is released, the real asset will have a digital identity and a QLC Chain account that affirms the ownership relationship between real asset owner and the real asset on the blockchain. In addition, the owner can provide services with the registered asset.

Asset Smart Contract follows below principles:

- Asset Smart Contract allows asset owner to register the real asset on the blockchain without new token generated.
- Asset Owner Account produces the Asset Smart Contract Account when issuing the Asset Smart Contract.
- When Asset Smart Contracts are deployed, the Asset ledger will be generated on the QLC Chain for asset information record.
- User triggers the execution of the Smart Contract by sending the transaction to the Asset Smart Contract Account. The transaction can be completed in Public Token or other Tokens. Asset Smart Contract Account will provide corresponding services after receiving the payment.

- The block storage requirement brought by the Asset Smart Contract is stored on the storage node on the QLC Chain.
- The transaction history from the asset ledger is saved in the storage node of QLC Chain and further confirmed through the Shannon Consensus.

4.2.5 Features of QLC Chain Virtual Machine

QLC Chain VM architecture is illustrated as the following:

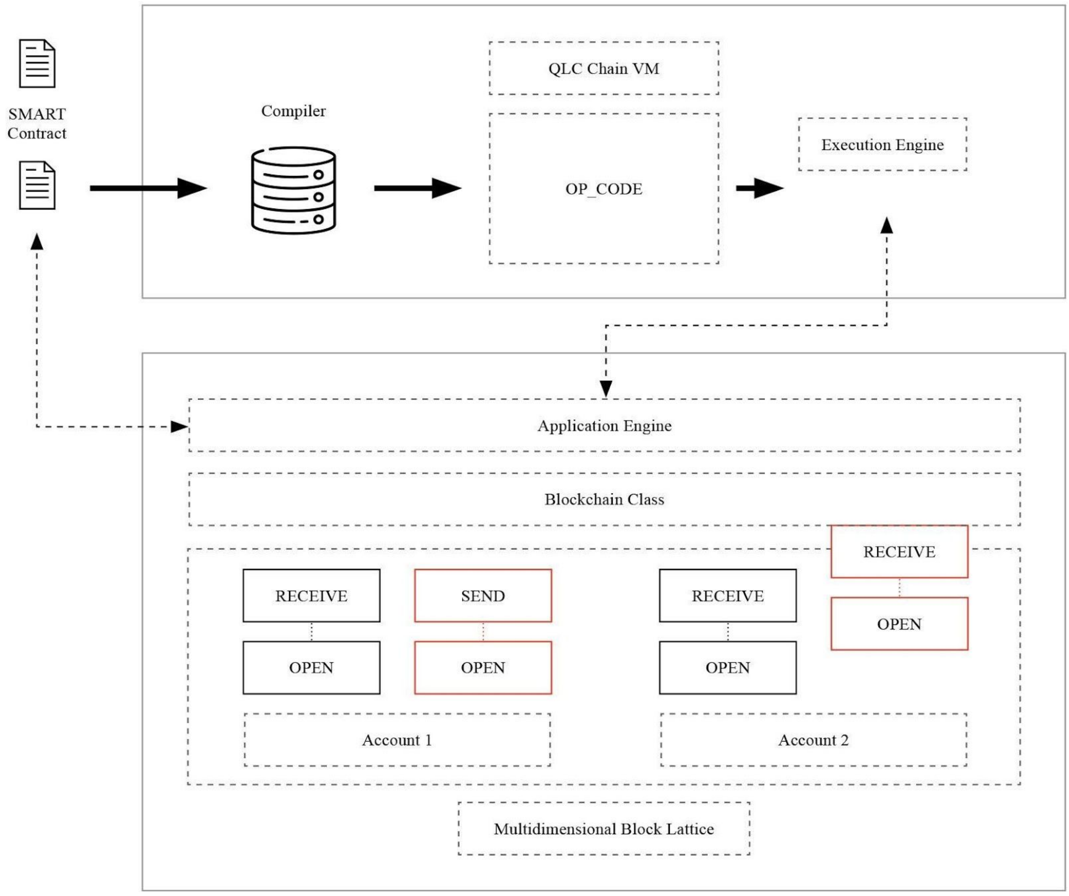


Fig. 8. QLC Chain Virtual Machine

- QLC Chain VM is Turing complete and supports native language such as Golang/C#/Java/C++.
- The compiler translates Smart Contract into the “OP_CODE” which is further converted into an ABI. The result of the Smart Contract is stored in the Block Lattice.

4.3 Dual Consensus Protocol

QLC Chain uses dual consensus protocols for global agreement, including DPoS for transaction and Shannon Consensus for Storage Node (DPoS + Shannon Consensus).

DPoS is designed for transactions among General Accounts without incentive to representative node. However, the introduction of Smart Contract brings in data storage nodes to contribute storage, bandwidth, search capability and more that DPoS cannot efficiently satisfy the need of communication anymore. Therefore, we design a new consensus, Shannon Consensus, to serve the need of data storage in an economical and efficient way.

4.3.1 Background of Shannon Consensus

1) Economic Thoughts behind Shannon Consensus

- Traditionally speaking, PoW and PoS protocols both have inevitable drawbacks. PoW is cursed with its concentration in hashrate, while PoS is the game for the rich (nodes with more tokens are more likely to be selected for voting). Although both PoW and PoS acknowledge the issue by increasing the cost of "being evil", it is still possible that the network is compromised or manipulated by miner-alliance or large-stake-token holders and eventually discourage the fairness of the ecosystem.
- Under PoW and PoS, ledgering power is the extra benefit for "rich people". In contrast, we believe the key players should be the "middle-class" and the "poor" who are able to enhance the liquidity and dissemination of Cryptocurrency. However, getting charged the transaction fee in this process will not only evidence the Matthew Effect, but also destroy the fairness among the cryptocurrency ecosystem.
- Similarly, DPoS and BFT, modified versions of PoS, sacrifice the decentralization by selecting a group of representatives to vote, which causes the likelihood of "voting manipulation". Consequently, the effectiveness of the consensus is adversely affected under the concentration of the powerful parties.

2) The goal of QLC Chain Consensus Protocol

- Classify transmitting node and ledger node that to reduce the Matthew Effect. QLC Chain encourages more “Middle Class” nodes to participate managing the ledge and get rewarded. The “rich node” will also get rewarded by transmitting transaction for resource contribution and make secondary distribution to “Middle Class”. The network resources thus incline fairly distributed.
- The amount of ledgering reward depends on the ratio between number of tokens on hand and effective workload. The reward is zero if either number of token or effective workload is zero.
- PoW is measured by effective workload. The “noise” raised in the PoW leads to the fact that network effect is more applicable than the local behavior from individual nodes. That means QLC Chain relies on the network connection and becomes more secure.

3) Illustration of attacking scenario in QLC Chain consensus

- A number of nodes on the QLC Chain have a large amount of tokens but provide very little workload. These “rich” nodes have higher probability to be selected as the bookkeepers. In the long run, however, they inevitably spend tokens in ongoing transmission tasks. That means the bookkeeping power is gradually impaired when workload and “wealth” are downsizing.
- A number of nodes on the QLC Chain contribute intensive transmitting workload but have very few tokens. These nodes accumulate “wealth” by working hard and consequently become superior in bookkeeping power. This is fair and derived from the mechanism of market selection. We are not able to predict or prevent it.
- A number of nodes work intensively to get tokens but later speculate tokens for profit. The likelihood of being able to bookkeep the ledger falls when they have fewer tokens. The drawback of PoW and PoS in concentration of hashrate is avoided here because speculating nodes are not able to accumulate both the scale of digital/actual wealth and the stake. “Monopoly” doesn’t occur.
- A number of nodes grow up to occupy a large ledger stake by completely purchasing tokens from the open market and taking very limited workload. This situation will be developed to “Nothing in Stake” problem eventually. We can prevent that by raising the price of tokens to increase the malicious cost. Additionally, other nodes can disconnect with the malicious node to make its workload zero so that the malicious node is not able to bookkeep the ledger anymore.
- Under the innovative mechanism of Shannon Consensus, the distribution of tokens achieves mean reversion, which prevents token or hashrate from extreme centralization. Additionally, a dynamic role conversion exists between the token owner and the workload contributor: nodes with large transmitting workload are rewarded with tokens and nodes with moderate workload will be converted to ledger nodes and still be rewarded with token sharing. Neither Ledger node nor transmitting node is incentivized to take the risk of arbitrage and act malicious.

4) Deduction of Shannon Consensus from PoW/PoS Consensus

Theoretically, PoW satisfies the following mathematical inequation:

$$SHA3(previous\ block\ hash,\ nonce,\ time\ stamp,\ Merkel\ tree\ root)\ <\ target \qquad (1)$$

According to Satoshi Nakamoto's theory, the solution of the in-equation indicates that nodes on Bitcoin has uniform distribution

$$P(X > x)\ =\ \tfrac{1}{x}(x > 0)$$

However, due to the feasible conversion between Bitcoin and legal currency and the widespread application of ASIC chips, hashrate is artificially centralized. The actual PoW inequation in each node is:

$$\frac{SHA3(previous\ block\ hash,\ nonce,\ time\ stamp,\ Merkel\ tree\ root)}{N}\ <\ target \qquad (2)$$

where N is the coefficient of hashrate concentration in a given node. N is significantly correlated to miner's economic strength. We believe N falls under Pareto Distribution:

$$P(N > n) = \frac{x}{min(x)}^{(-k)}$$

(X is any number > min (x), min (x) is the minimum positive value of x, k is a positive parameter)

If we define N in the inequation (2) as the amount of token, we derive the following PoS inequation:

$$SHA3(previous\ block\ hash,\ nonce,\ time\ stamp,\ Merkel\ tree\ root)\ <\ target\ \times\ N \qquad (3)$$

where N is the amount of token held by a given node

However, inequation (3) doesn't comply with Nakamoto's original intention that one CPU one vote. The value of the right-hand side of the inequation is changing during mining, which transform the solution from the uniform distribution to Pareto Distraction gradually. In order to maintain the solution in uniform distribution, we have to introduce a new coefficient on the left-hand side of the inequation to neutralize the impact.

Based on our observation, the process of solving hash function is similar to a continuous process of producing entropy. The entropy synchronizes the accumulation with the concentration of hashrate.

$$H(E) = -Sigma\,[\,p(e) \times log_{2}(p(e))\,] \qquad (4)$$

We add entropy as the new coefficient to the right. In reality, the more the token held in PoS, the more active the node for transmitting data. Simultaneously, the number of bytes transmitted follows Pareto Distribution and can be presented by Shannon formula:

$$\frac{SHA3(previous\ block\ hash,\ nonce,\ time\ stamp,\ Merkel\ tree\ root) \times E}{N} < target \qquad (5)$$

We modify (5) by placing the coefficient to the right:

$$SHA3(previous\ block\ hash,\ nonce,\ time\ stamp,\ Merkel\ tree\ root) < \frac{target \times N}{E} \qquad (6)$$

where N is the number of token held and E is Shannon Coefficient

This is the Shannon Consensus deduction process.

Especially considering the network scenario, we can replace the Shannon entropy with capacity of channel transmission. We introduce a new stake coefficient that measures the marginal value of PoS in per unit of PoW

$$Stake = \frac{Token}{PoTa \times log_{2} \frac{1+PoRe}{PoSp}}$$

To entitle to manage the ledger, the following condition has to be met:

$$SHA3(previous\ block\ hash,\ nonce,\ time\ stamp,\ Merkel\ tree\ root) < nonce \times stake$$

PoTa: the total traffic in QLC Chain including upload and download of node.
PoRe: the upload traffic to other node in QLC Chain.
PoSp: the storage for data produced by transaction in QLC Chain

5) Implementation of Shannon Consensus and related algorithm

a. Global election of validation node

a1. Overlay a hash addressing mesh network on top of the conventional physical network. Mesh network will perform a global next hop which is the random next hop in TOR network. It updates in every 10 minutes and ensures the next hop is 7 bytes.

b. Account balance for vote

b1. Under the account balance , each node creates its own account balance and a ledger of the global network balance.

b2. Each account has a private key for the local ledger based on the elliptic curve cryptography. The private key is immutable.

c. Sharding model in network consensus

c1. Shard from OSPF/BGP/VLAN. Each network slice reaches individual consensus and different shards reach secondary consensus through edge network gateway.

c2. The local ledger of each node validates through Shannon Consensus within the individual network slice. Normally, step a1 is adequate to complete the process. The following attacks can also be effectively prevented by Shannon Consensus in global network:

- Double spending fork attack
- 51%+ Attack
- Sybil Attack
- Network Storm that causes election failure of voting nodes

Acknowledgement

We would like to thank everyone throughout the process who shared their advice, network, feedback and so much more to guarantee the quality of the yellow paper. Special thanks to Dr. Ueli Maurer from Information Security and Cryptography Research Group, Dr. Lixin Gao from University of Massachusetts, Dan Kolkowitz, Tech Advisor of QLC Chain and Advisor of Cryptic Labs, Professor Wan Hong from Purdue University, Da Hongfei, founder of NEO, Wang Jianbo from Cybex, Cao Ying and Wang Zuguang, Independent advisors, Li Jun, project lead of Ontology and Jack Lee from HCM.

We would also like to thank the entire blockchain community for their creativity and dedication to make blockchain technology accessible for the average users. Together, we will continue to create more practical, equitable, and efficient technologies.

References

1. LeMahieu, C. (n.d.). RaiBlocks: A Feeless Distributed Cryptocurrency Network.
   Retrieved from https://raiblocks.net/media/RaiBlocks_Whitepaper English.pdf

2. Benefits Of Universal Blocks – Nano – Medium
   Nano - https://medium.com/@nanocurrency/benefits-of-universal-blocks-3354792fe681

3. Block Lattice - Github https://github.com/nanocurrency/raiblocks/wiki/Block-lattice

4. Chantel Cary. Ovum Decision Matrix: Selecting a Real-time
   Convergent Billing and Charging Solution, 2016

5. A.M Antonopoulos. Mastering Bitcoin, 2014.

6. K. Christidis and M. Devetsikiotis. Blockchains and Smart Contracts for the Internet of Things.
   IEEE Access, 4:2292 - 2303, 2016.

7. A. Kiayias, I. Konstantinou, A. Russell, B. David, and R.
   Oliynykov. A Provably Secure Proof-of-Stake Blockchain Protocol,
   2016.

8. http://openbts.org/

9. S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008.
   [Online]. Available: http://bitcoin.org/bitcoin.pdf

10. Network-as-a-Service - https://en.wikipedia.org/wiki/Network_as_a_service