everiToken

TECHNICAL WHITEPAPER

Version 3.0

Zug, Switzerland

Last Updated: February 20, 2019

DISCLAIMER

- The everiToken technical whitepaper is for informational purposes only.

- This whitepaper does not represent any expressed or implied warranty, proof, expectation, etc.

- The technical specifications or technological realization methods written in the whitepaper may change over time.

- This technology team may be disbanded or reorganized at any time, or the loss of core technicians could lead to the failure or part realization of the project.

- This whitepaper is provided “as is”. Neither the project team nor any members of the project are responsible for any content or results stemming from future use of this content.

- The token in this whitepaper does not have any practical value, only use in the virtual world, and its sole purpose is to confirm the user permissions of the token.

- Any event within the blockchain or its derivatives, which are run by the technology described in this technical whitepaper, is automatically generated by program automation, and we are not responsible for its consequences. Individuals or organizations are responsible for the inherent consequences when using the everiToken blockchain.

- All the content contained in this technical whitepaper may be used on the premise of non-commercial use, however the technical whitepaper must not be modified or altered in any way. We are not responsible for any effects resulting from the use of this content.

Content

Part I. Background and Vision

Arrival of the Token Economy

Blockchain technology turned 10 years old in February 2019. Despite its evolution over this time, one key question still persists: Is blockchain technology revolutionizing production in ways that create value for the global economy?

Let’s look at the data—at present, the assets managed on blockchains (referred to as ‘on-chain’) are basically a variety of coins/digital currencies, with a total market value of about $150 billion USD. Assets in these chains are generally characterized by high volatility and strong speculation, and fail to provide benefits for the global economy. In fact, since Satoshi Nakamoto, people have wanted to make these ‘coins’ a payment currency, but as of now they primarily serve as digital currencies and do not play a traditional currency role. A digital currency is more of a name than an actual operating currency itself.

On one hand, the right to issue currency is a political realization, and monetary power must belong to the state. Therefore, it is very difficult for cryptocurrency to replace fiat currency. Without authorization and support from the state, the so-called ‘digital currency’ is only an idealistic pursuit.

On the other hand, most global mainstream assets (tangible and intangible) are not on blockchains (referred to as ‘off-chain’), and there is limited interaction between blockchains and assets off-chain.

So, is a token just another digital currency? Not at all. The basic definition of token is “symbol, sign”, but it should more appropriately be considered a certificate rather than a digital currency. Such certificates can represent all kinds of rights and interests including shopping points, coupons, identity cards, diplomas, real estate, access keys, event tickets, and a wide variety of proofs of rights and interests. Looking back on history, proof of rights and interests has been an essential element of all civilizations of human society. Accounts, ownership, qualifications, proofs, and so on are all representative of rights and interests. As Yuval Grali stated in the *Brief History of Humankind*, “it is these ‘fictitious facts’ that are the core reasons for the wise men to stand out and to build human civilization.” If these proofs of rights and interests were all digital, electronic, and cryptographically protected to verify their authenticity and integrity, then human civilization would be revolutionized. We call this phenomenon the **token economy**.

Running certificates on-chain provides a solid foundation of trust and traceability not provided by any traditional centralized infrastructure. Therefore, if a certificate is the front-end economic unit of the token economy, then the blockchain is the back-end technology of the token economy. The two are integrally linked and co-dependent.

Competitive Analysis

As a **public blockchain** born for the token economy, everiToken currently has two main competitors, Ethereum and EOS. Our competitive advantage becomes clear when analyzing the strengths, weaknesses, opportunities, and threats within our market.

SW（Strength and Weakness):

everiToken believes that blockchain technology for the token economy should effectively manage the proof of rights and interests, primarily with regards to the following three aspects:

1. **Proof of Digital Rights and Interests**: The certificate must be a credible digital form of rights and interests, which must be backed by something of inherent and intrinsic value (whether tangible or intangible).
2. **Security, Encryption, and Authorization Management**: The certificate must be verifiable, tamper-proof, privacy-protected, supervised, protected by cryptology, and only usable by those authorized.
3. **Negotiability:** The certificate can be traded and exchanged conveniently.

According to the above requirements, we put forward a set of solutions to meet the basic needs of the token economy, to promote the management and circulation of tokens, and to build a technical foundation for the token economy. Specifically, we have realized the following three main characteristics according to the above requirements.

- **Fast, Convenient Issuance of Tokens**: Users do not need to write code and can easily issue their own tokens through our API (for apps, web pages, or third-party applications).
- **Efficient Transfer of Tokens**: Enable the transfer of tokens within seconds at the volume of hundreds of millions of tokens simultaneously.
- **Flexible Authorization Management:** A simple, elegant, and unified model to

achieve authorization management, which supports multi-person holding, private key recovery, multilevel authority, legality, government supervision, and other complex requirements without the need for extra coding.

Let's take a look at Ethereum and EOS:

Ethereum: ERC20/ERC721

The main way to achieve the token economy with Ethereum is to develop smart contracts based on the ERC20 and ERC721 protocols. Among them, ERC20 supports FTs (fungible tokens), and ERC721 supports NFTs (non-fungible tokens). However, there are some serious problems:

- **TPS**: Presently, Ethereum can only support fewer than 20 transactions per second and is unable to meet all the practical needs of token usage and circulation.
- **Cost**: The implementation of Ethereum smart contracts requires a gas fee for every step. For functions with complex business logic (such as multi-person holding, supervision, legality, etc.), the cost might be high and uncontrollable.
- **Popularize**: The realization of the token economy with Ethereum is based on smart contracts, which are not accessible for non-developers without the use of third-party applications because of their complex nature. This creates security and regulatory concerns, while preventing mass adoption.
- **Non-Standardization**: Since different smart contracts may require completely different developmental ideas, the metadata of these virtual tokens is unexchangeable and as a result isolated. This is not conducive to the ecological development of the token economy; additionally, the users cannot use a unified way to query all the different kinds of token assets that they own.

EOS

EOS launched its mainnet in June of 2018. EOS's primary aim was to remedy the problems of Ethereum by creating new solutions. However, this has created a whole new range of problems:

- **Security**: Token transactions may represent extremely precious and non-renewable real entities, and thus it is important that there are no security problems. However, overall development that continues to be based on smart contracts is limited by the proficiency level of developers, and it is difficult to ensure that all types of token developers have sufficient security awareness.

  EOS's smart contracts are based on **WebAssembly**, which is relatively new and still in the test (Beta) stage. Additionally, EOS's smart contract code is Turing complete and has excessive authority, which makes it vulnerable to unintentional security loopholes.

Most people cannot write secure smart contracts. In order to issue and transfer tokens, users must rely on third-party applications and must trust the quality of the code of that third-party. Thus, the control of assets is not in users' own hands, but rather that control is relinquished to a third-party.

- **Non-Standardization**: Like Ethereum, the metadata of different smart contracts cannot interact or cooperate together.
- **Regulation, Trust and Legality:** Due to the technical expertise required by non-standardization and code reading, it is difficult for the government to achieve regulation. Likewise, non-developers may find difficulty in deciding whether they can trust relevant programs, which makes it hard for blockchains to be accepted by ordinary people and governments.
- **Execution Efficiency**: In order to meet diverse needs, EOS's smart contract functions are complex, the system modules are numerous, and the resource scheduling and distribution are difficult. Together, this greatly increases the complexity of the system and reduces the speed of the operation. Due to the possible conflicts among different data and functions, using multithreading to increase speed is not easy, and scheduling costs are high. However, for the token economy, these complex functions are critical and must be solved.
- **Popularize**: The business needs of the global economy are complex, variable, and lack consistency. However, smart contracts take time to develop and test, which makes it difficult to solve the needs of diverse markets in a short time span. This is a hindrance to the development of the token economy.

The main difference between everiToken and others is that everiToken uses *safe contracts* while others use *smart contracts*. That means everiToken is not Turing complete and there will be some complicated application scenarios that everiToken cannot satisfy. However, everiToken can meet 99% of the demand of the token economy, and everiToken is the safest, most cost-effective, and user-friendly public chain for all people throughout the world.

OT（Opportunities and Threats）

Along with the strengths of everiToken, we have created the EvtLink standard which is used to connect payers and payees via a variety of data channels including NFC, Bluetooth, and QR code. Based on EvtLink, everiPay is a payment protocol born for **face-to-face token micropayments** using everiToken public blockchain as the core infrastructure and everiPass as its token ownership validation protocol. everiPay/everiPass includes the standard of **QR code** generation and the definition of communication protocol. We have achieved an impressive list of features with our innovations:

- **Instant Clearance**: A transaction is a settlement.
- **Decentralization**: P2P payment, no centralized platform, no one can modify the data on-chain, and everyone can participate in pricing.
- **Most Secure**: The data and content within the blockchain cannot be forged or tampered with, so as to maximize the protection and security of user property.
- **Compatible**: everiPay/everiPass support all tokens supported by everiToken, as well as currency, points, and even a key to open a door. You can use it almost everywhere with only a phone.
- **Most convenient**: Even if you can't connect to the Internet, you can complete the transaction.

Based on the above five characteristics, everiPay/everiPass can provide the world's most secure, convenient, and user-friendly service for face-to-face payments and token ownership.

Summary

Some threats still exist. As mentioned, Ethereum and EOS can be a great public chain for certain specific needs within the token economy. However, the greatest problem for Ethereum/EOS is the high entry barrier for users created by the nature of smart contracts. We have solved this problem with the development of the *safe contract*, and everiToken is now poised to support a world-wide token economy for all people.

Based on the above analysis, we have designed a new concept that is perfectly suitable and preferable for a majority of blockchain applications and propose a new public chain and ecosystem, **everiToken,** to further the development of the token economy. The assets, certificates, and vouchers of the real world can be **digitalized** by the issuance of tokens and can be easily used with unprecedented security, speed, and network compatibility.

Part II. Technology of everiToken

Safe Contract

Smart contracts, in theory, are an effective digital means of facilitating decentralized exchanges of goods or services without the need of a middleman. However, in practice, smart contracts suffer from widespread security vulnerabilities that arise from improper implementation and logical errors, giving rise to consequences such as lock-outs, leaked access, and improper terminations. As such, smart contracts often fail to provide a sufficient level of trust and may be viewed as less reliable than traditional contracts or exchanges.

everiToken introduces the novel idea of *safe contracts* via our API layer. Rather than code directly, users rely on safe contracts to facilitate processes such as the issuance and transfer of tokens. By simplifying functions to the core requirements, safe contracts ensure that all chain transactions are secure and without loopholes, as the available API functions are fully reviewed and verified. Even though safe contracts are not Turing complete, they can still achieve the majority of necessary functions via APIs, and provide flexibility to token-issuers for the completion of off-chain services.

Furthermore, safe contracts have the added benefits of increasing accessibility and TPS. Regarding the former, the inclusion of APIs makes it simple for easy integration into existing workflows without having to write chain-integration code from scratch. In regards to the latter, the usage of APIs allows for various translations types to be distinguished easily, and independent token transactions can be processed in parallel at faster speeds (10,000 TPS achieved on mainnet: December 2018).

Database

EOS utilizes a Boost.MultiIndex-based memory database (Chainbase) that supports rollback operations. The results of all contract operations exist in the memory database. In order to support rollback when branching and recovery when the contract code is abnormal, it is necessary to record extra data for rollback in every operation. In addition, all data is stored and processed in the memory database. With the increase of users and transactions over time, it is foreseeable that the demand for memory will increase significantly. This will result in a high demand for memory capacity from the block producers. Furthermore, if the program crashes or restarts, the memory data will be lost.

To restore data, we would need to repeat all operations in the blocks, leading to a long and impractical cold startup time.
While preserving EOS's memory database, we developed a token database based on RocksDB which has several benefits:

- RocksDB is a very mature, industrial-level, and key-value database which has been fully verified and is used in the core cluster of Facebook.
- RocksDB is based on LevelDB, but provides better performance and a richer functionality than LevelDB. It also enables optimization of low latency storage situations, such as Flash or SSD.
- If needed, RocksDB can be used as a memory database.
- RocksDB based architecture naturally supports version rollback and persistence, and its influence on performance is extremely low.

Our token database has RocksDB as its underlying storage engine. We have fully optimized token-related operations to maximize performance. With this technology, we can achieve rollback at a lower cost. In addition, the token database also supports optional functions such as data persistence, quantitative backup, and incremental backup to solve problems like cold startup.

Because the operations in everiToken are highly abstract, the code is fixed, and the information required for each operation is minimal. Thus, data redundancy is very low compared to general systems such as EOS, which also reduces the size of blocks.

Token Model

Overview

Born for token economy, everiToken is unique with its token-based, token management method. Tokens are different from digital currencies issued by central banks and encrypted currencies (Bitcoin or ETH).

We define a token as a proof that you have an exclusive share of the economy in an asset, a period of time, particular place, or a time-based service provided by a particular entity. Tokens are divided into two types: fungible tokens (FTs) and Non-Fungible Tokens (NFTs). There are some differences in their application scenarios and structures.

According to our analysis, non-fungible tokens may play a more extensive role in the token economy. Thus, we will begin our analysis with non-fungible tokens.

Non-Fungible Tokens

Before understanding non-fungible tokens, let's consider a large number of stones on a beach. In the real world, every stone on a beach has a different weight, appearance and rock type. There are no two exactly identical stones. Also, stones cannot be easily combined together. Therefore, we say that every stone is ‘indivisible’ and ‘not to be combined’.

An example in blockchain is the CryptoKitties, which was once a hot game in the blockchain world. Each cat has unique numbers and attributes. An NFT is similar to an individual, stone, or blockchain cat. It is naturally different and unique in the real world, as are the NFTs in our system.

Generally speaking, NFTs are divided into different categories according to their various value types. We can categorize similar kinds of NFTs to form a domain.

Concentrating on tokens allows for the high standardization feature of everiToken. All custom tokens issued by users satisfy the same structure. Specifically, each token contains one **domain name,** corresponding to a specific **domain** (that is, the classification that the token belongs to). The issuer also designates a **token name**, which must be unique within the domain. A token name usually stands for some special meaning. For example, the bar code of a product can be used as a naming rule, which includes information on the country of origin and manufacturer of the product. The uniqueness of each token is determined by the domain name together with the token name. In addition, information about the ownership is included, and each token has at least one **owner**.

As mentioned above, the **ID** of a token is uniquely determined by the domain name and the token name. The basic structure of a token is shown in Figure 1. Besides the token ID, the structure also shows the owner of the token and other necessary information.

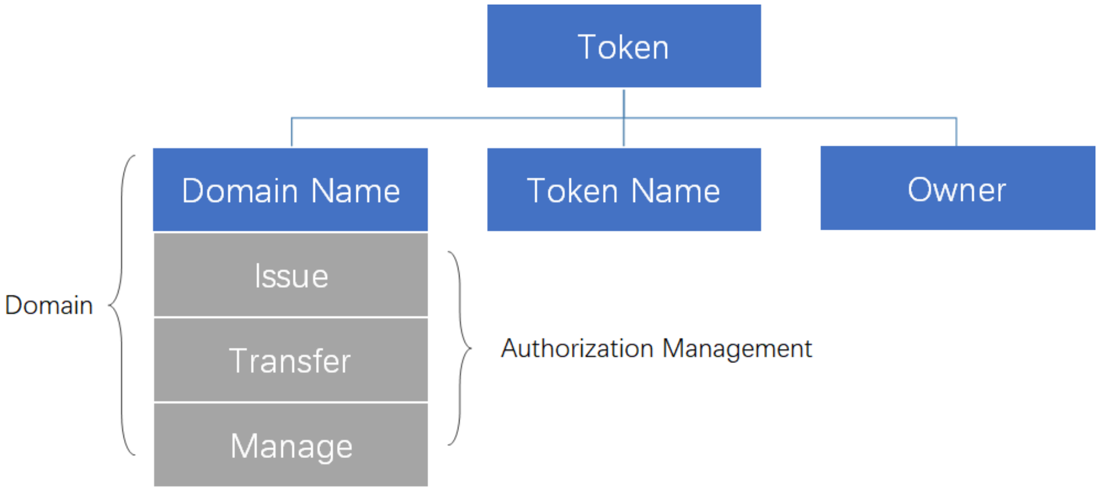


Figure 1. The token structure of everiToken

The domain details can be queried by the domain name. Each domain also displays its relative authorization management information.

Everyone has the right to issue one's own token. The token itself is of no value, and its utility is endorsed by the issuer's real credit. Once a new token is issued, it can be transferred to others through transactions.

For NFTs, the transferring of a token means changing the owner of that token. Every token has an **owner group** (there can be one or more owners). When a change in the owner group is needed, a member of the token circulation can confirm the operation by signing a digital signature, and the owner group of the token changes after the everiToken node confirms that the transaction satisfies permission requirements and synchronizes with the other nodes.

Authorization Management

The everiToken system contains three kinds of permissions regarding authorization management: Issue, Transfer and Manage.

(1) **Issue** is the right to issue tokens in this domain.

(2) **Transfer** is the right to transfer tokens in this domain.

(3) **Manage** is the right to modify the domain, including authorization management and other parameters.

Each specific authorization follows a tree structure and is therefore called an

authorization tree. As the root, each permission has a threshold and is connected to one or more actors.

Actors

Actors can be categorized into three groups: accounts, regular groups, and owner groups. Accounts are individual users, groups are clustered accounts, and an owner group is a special form of regular group.

A group can be a club, a company, a government department, a foundation, or even just an individual. A group retains the public key of the group, and the public keys and weights of each member. Operations are approved when the summed weight of all authorized members in a group approving the operation meets the required threshold of the group.

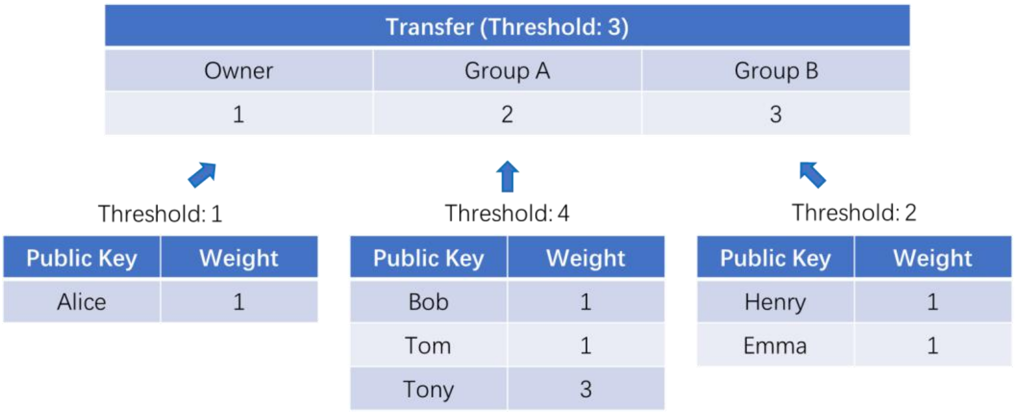
At the same time, the member that holds the public key of the group can authorize modifications on the group members and their weights. This mechanism is called **group autonomy**.

When a group is initiated, the system generates a group ID automatically. When the issuer designs authorization management for a domain, it can be invoked by directly referencing the existing group ID to its permissions system. Due to group autonomy, each group can be reused conveniently.

The owner of the token has a special group name with the fixed name '.owner' which represents a collection of a token's owners. It is special and dynamic because it always refers to the actual owners of each token, and the group's authorizing condition is that everyone agrees within the group (that is, the weight of each person in the group is 1, and the group's threshold is the number of members in the group).

Management

The authorizations are initiated by issuers of tokens, and each authorization is managed by at least one group. When the token is issued, the issuer specifies the information and relative weight of each group under each authorization and also sets a token threshold. Before executing an operation in a certain domain, the system will first verify if the operating group has enough weights, and the operation will be approved only if the weights exceed the threshold. This grouping design is suitable for many situations in the real world, and the flexible setting of weights and thresholds meets all kinds of complex needs. An example is given in Figure 2.



Transfer (Threshold: 3)		
Owner	Group A	Group B
1	2	3

Threshold: 1

Public Key	Weight
Alice	1

Threshold: 4

Public Key	Weight
Bob	1
Tom	1
Tony	3

Threshold: 2

Public Key	Weight
Henry	1
Emma	1

Figure 2. Transfer Permissions

Figure 2 describes the transfer permissions of a domain. The threshold value is 3, and there are three groups involved, namely Owner, Group A, and Group B. Based on the current set of weights for each group (1, 2, and 3, respectively), Owner and Group A need to authorize together, or Group B can authorize alone to meet the transfer threshold.

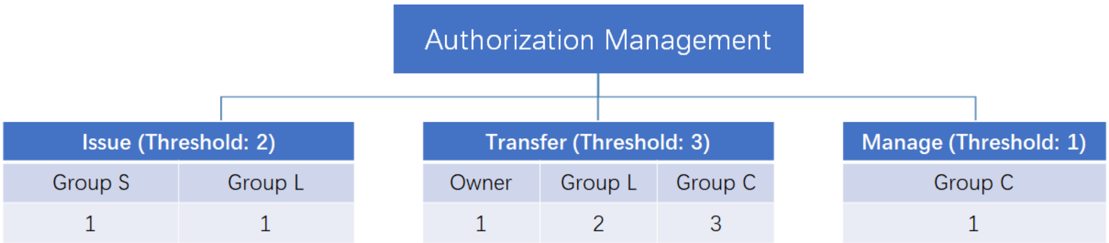
For each group, Owner is authorized by only Alice; Group A can satisfy its threshold (4) by authorizations by at least Bob/Tony or Tom/Tony; Group B must be authorized by both Henry and Emma to meet the threshold (2).

Any user has the right to issue tokens, but the target scenarios of tokens in each domain are different. For example, the transfer of property must be reviewed by government related agencies with strict supervision; the chain's membership cards and coupons need the company's brand to endorse them; a concert ticket is useless after the concert, but a fixed parking space's owner may change with time.

When issuing tokens, the issuer of the token can implement authorization management by designing permissions in the domain. The following scenario demonstrates the convenience of authorization management.

Figure 3 shows how complex problems can be solved by using everiToken's authorization management mechanism.

A company has built a new office building and hopes to issue 1000 tokens bearing the property rights of the building. The company sets up a SPV (Special Purpose Vehicle) to issue and maintain these tokens. In real life, the token issuance and transfer of the property needs to be examined and approved by the local property bureau. They must be issued in conformity with local standards, and then the token details (total, issuer, authority management structure, etc.) can be displayed on its official platform. On top of that, the central property department has the highest authority to limit and manage the local property bureau and owners.



Authorization Management

Issue (Threshold: 2)	
Group S	Group L
1	1

Transfer (Threshold: 3)		
Owner	Group L	Group C
1	2	3

Manage (Threshold: 1)
Group C
1

Figure 3. The Structure of Authority Management

Group S represents the SPV, the issuer, and the initial owner of the token in the domain. **Group L** represents the local property bureau, and **Group C** represents the central property department.

In most cases, the transfer of a token only needs the authorization of the owner and the local property bureau (combined weight of 3, meeting the threshold). In this process, the transfer operation is audited by the local property bureau. In the case of an accident, such as a token owner who has passed away or lost a private key, the central property department may transfer the ownership of the token to the legal heir after a court or relevant department’s judgment or review.

If part of a token’s ID is lost (which is likely to happen), or if both the SPV and other token owners agree to add new tokens, they can add these tokens by getting the issuing authority to meet the actual needs. Furthermore, the authorization management structure is also suitable for handling extreme cases. For example, if the central property department needs to temporarily freeze the spread of this type of token, it can change the threshold of transfer permissions through the management permissions it holds, thus freezing the circulation of all tokens in the domain.

Fungible Tokens

Issuance

Everyone can issue fungible tokens after registering with a unique symbol, such as EVT. Users can set the total number of circulated tokens with this symbol. Then, users can decide the number of tokens that they want to immediately issue.

Transfer

Everyone with their own private key can transfer their tokens to others.

Other Details

Each account will record the number of tokens held along with the associated symbols. There will be an independent key-value record to store basic information of tokens with different symbols. Users can also allow another private key to have the right to transfer specified numbers of tokens with a specified symbol. This function is called **token allowance**, and it can be utilized in token exchange.

Token-Based Transaction Model

Overview

everiToken employs the **token-based transaction model** in regards to all tokens within our system.

In short, for each token in a token-based ledger, we create independent data space to store the full history of a token's ownership. In this way it's very easy to do sharding and multi-core paralleling because the data space of a given token has no relationship with other tokens. As a result, operations of various tokens can be done easily in a parallel manner without conflict. This enables a super high performance and constantly improving TPS by easily sharding or adding more CPU cores.
The token-based transaction model was invented by several core team members of everiToken and has been proven to work perfect for NFTs on everiToken.

A blockchain based on the token-based transaction model, like everiToken, might divide the database into two parts, one being Token DB and the other Block DB. The first one is where the token-based transaction model operates, storing and managing data spaces of all the non-fungible tokens. Block DB, the second one, stores the original blocks.

Both Token DB and Block DB should be a multi-versioned database for fast rollback when a block is reversed. For example, everiToken uses Rocks DB as the underlying database system of Token DB.

Both Token DB and Block DB are append-only databases. So, whenever someone updates a record, the new value with the increased version will be added to the database. However, the record containing the old version won't be removed.

Token DB

Token DB is an indexed database for quickly searching and changing the newest status of the blockchain such as the ownership of tokens and the account balance of fungible tokens on the chain.

Token DB could be considered a key-value database. The key indicates the ID of the tokens, and the value represents current ownerships of tokens. Since the database is append-only, there will be many values for each key, but only the latest value represents the current ownership status of the token, while the others are only for historical reference and rollback. For each token, there will be an independent data space that includes all the ownership history, just like a separate chain.

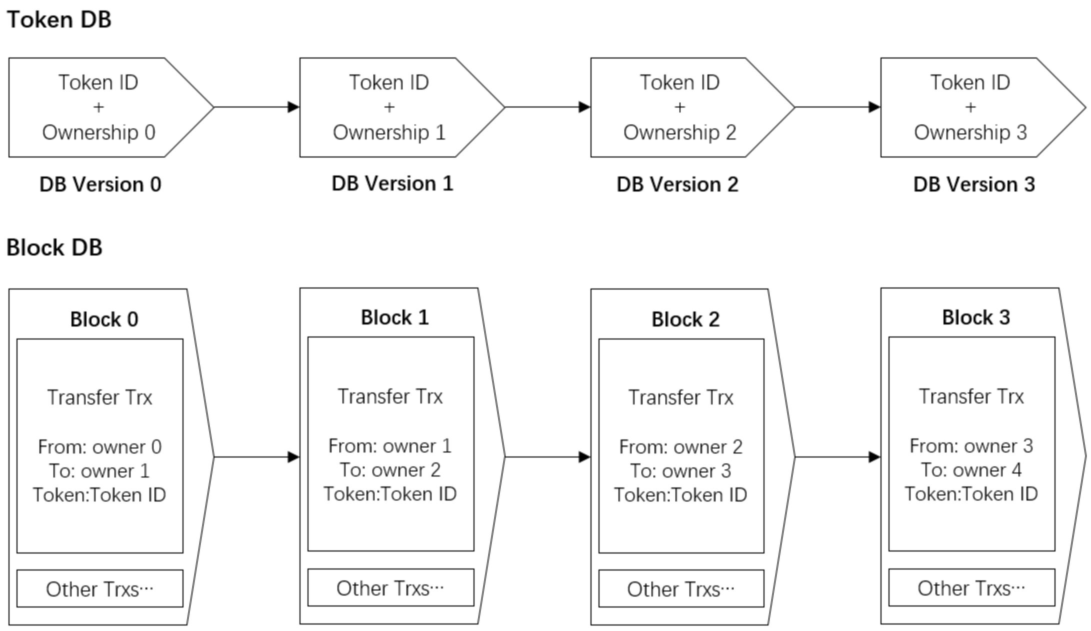
The first value of the chain is the initial ownership. For example, when one executes a transaction, the new ownership will be appended in the database. Old versions could be used to roll back the value if the block needs to be reversed and eventually will be garbage collected.

Because each token has an independent data space, sharding becomes very easy. For example, if we have two computers for one node, we could let each computer process half of the tokens. If there were 100 tokens, the first computer would process tokens 1 - 50, and the second for tokens 51 – 100. Because changing the owner of one token will not impact other tokens, the two computers could process in a parallel manner.

Block DB

Block DB is in charge of storing all the original, irreversible blocks of the chain. Each block stores all the detailed information including names, parameters of actions executed, signatures on the block, and more.

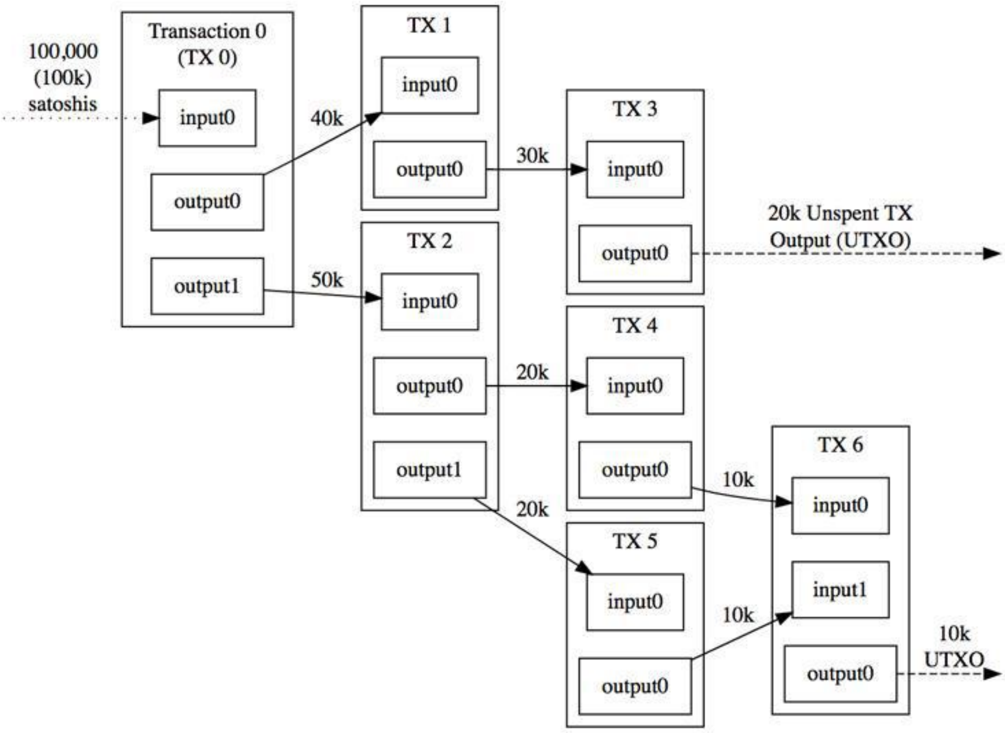
The following graph shows how two kinds of databases work together for NFTs:



Transaction Model Comparisons

a) UTXO

In the UTXO model, each token owner transfers a coin they own to another by digitally signing the hash of a previous transaction and the public key (address) of the next owner, adding these to the end of the coin. The mechanism is essentially a continual transgression of inputs and outputs where the owner of tokens actually does not directly own the tokens, but rather owns the output to a specific number of tokens that can then be signed over as an input to a new owner who then controls the new outputs.



Triple-Entry Bookkeeping (Transaction-To-Transaction Payments) As Used By Bitcoin

(Source: bitcoin.org)

As you can see, UTXO is great for avoiding double-spending as it is obvious that any input could only be used once, but it also has some disadvantages:

- BTC is not a kind of NFT, it is a FT. It is useless to keep a unique ID for every UTXO. (everiToken supports both NFT and FT)
- UTXOs are a one-off. It's a waste of computing resources and disk volume to store the huge amount of UTXOs.

b) Account-based

The account-based transaction model is just like what a bank does. You create an account at a bank and then save money into the account, changing the balance. This is completely different from the way UTXO works. It's more efficient than UTXO because it only has to update the balance in the database, not create new UTXOs. As a result, the UTXO model is not suitable for NFTs.

Moreover, the balance-based model is not good at sharding because when transferring something to another person, it requires two steps: the first is to modify the account of the old holder, and second is to modify the account of the new holder. For safety reasons, you must do two steps as one atomic operation, but in a sharding environment it's difficult and the performance level is poor. However, in the token-based transaction model there is only one step, which is to append the new ownership of the token.

Security

Focusing on the related functions of tokens, everiToken streamlines unnecessary abstractions, which not only greatly increases efficiency but also provides remarkable safety. Although the types of tokens on everiToken can be very abundant and theoretically unlimited, the unified token structure enables the system or any third-party organization to audit them following the same principles. It can be regarded that the system only recognizes one single form of smart contract, which avoids complicated auditing and security implications as a consequence.

everiToken Core Codebase

As of Spring 2019, everiToken has introduced four organizations reviewing all the core code of everiToken public chain including Hacken Proof, Chaitin, and others. Static and dynamic analysis were both included.

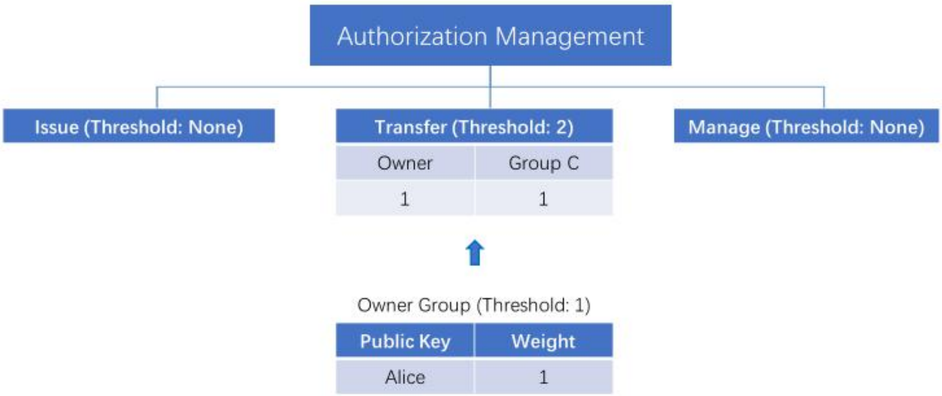
Since everiToken uses the *safe contract,* once it is proved that our core codebase is safe, then all the contracts based on everiToken are also proved to be safe.

Script (everiSigner)

everiSigner is an offline signer plugin for browsers. The whole signing process is done within this add-on so that private keys are never exposed. The website interacts with everiSigner by creating a new channel to ensure security; the website passes the content to be signed into the channel, and then everiSigner returns the signed data.

Lost private key

Based on the authorization management, third-parties can provide many services. For example, Company C specializes in password protection services, and Alice fears that she has forgotten or lost the private key to her own token. Alice can manage the transfer permission of the domain to the Owner (1), Group C (1), and set the threshold to 1. In this case, if Alice has forgotten her private key and cannot get the authorization by herself, she can still get the authorization via Group C if she proves herself to be Alice (through identity card or fingerprints) to Company C. In this way, Alice can recover the token by transferring it to a new account after verification.



Owner	Group C
1	1

Public Key	Weight
Alice	1

Figure 4. Company C provides service to retrieve key

Of course, Group C could steal Alice's token, but all operations will be recorded on the chain, which would destroy Group C's credibility.

Consensus Algorithm

everiToken uses BFT-DPOS as its consensus algorithm. DPOS is proven to be capable of meeting the performance requirements of applications on-chain. Under this algorithm, those who hold EVT may select block producers through a continuous approval voting system. Anyone may choose to participate in block production and will be given an opportunity to produce blocks, provided they can persuade token holders to vote for them.

everiToken enables blocks to be produced every 0.5 seconds, and exactly one producer is authorized to produce a block at any given point in time. If the block is not produced at the scheduled time, then the block for that time slot is skipped. When one or more blocks are skipped, there is a 0.5 or more second gap in the blockchain.

The number of block producers for everiToken public chain is dynamic. For the initial year, 15 producers are set. Then the number will be decided on by an on-chain governance committee. For convenience, we will use 15 for the whitepaper.

In everiToken, blocks are produced in rounds of 180 (12 blocks each, multiplied by 15 producers). At the start of each round, 15 unique block producers are chosen by preference of votes cast by EVT holders. The selected producers are scheduled in an order agreed upon by 11 or more producers.

If a producer misses a block and has not produced any block within the last 24 hours, they are removed from consideration until they notify the blockchain of their intention to start producing blocks again. This ensures the network operates smoothly by minimizing the number of blocks missed by not scheduling producers who are proven to be unreliable.

The Byzantine fault tolerance is used to provide extra security and safety for users by requiring all confirmations to be signed by all producers. No producer may sign two blocks with the same timestamp or the same block height. Once 11 producers have signed a block, it is deemed irreversible. Any Byzantine producer would have to generate cryptographic evidence of their treason by signing two blocks with the same timestamp or block height. Under this model, an irreversible consensus should be reachable within 1 second.

Bonus Design

Bonuses have been added with the release of everiToken 3.0 in February 2019. It's a powerful, flexible, and convenient element to combine with existing features. It’s mainly designed for the purpose of distributing profits to its stakeholders or shareholders according to a set of rules. There are two types of bonuses supported now according to the different ways of collecting profits: passive bonus and active bonus.

For the passive bonus, the profit is collected during every transaction within one fungible token. So, if the managers of one fungible token decide to set a passive bonus to it, then in every transaction not only EVT will be charged as a fuel, but also an additional fee for the fungible token will be charged as well.

There are several options to control the actual fees in one transaction. The main option is the transaction rate. The result of fees is the rate multiplied by the amount of the transaction. There are also minimum and threshold control options limiting the upper and lower bounds of the final fees. This prevents an overwhelming cost for high value transactions.

The manager of the fungible token can decide how the fees will be charged such as which party will be responsible for the fee and the method of fee attachment. The first method is like a credit card, with the payer paying *n* amount but the payee receiving less than *n* amount because the fee is subtracted from the initial amount. The second method is more like a traditional bank transaction. If you want to transfer *n* amount to another, you need to pay an additional fee for this transaction on top of the original amount.

As for the active bonus, it’s launched manually, similar to dividends of stocks. It’s decided by the fungible token manager how much bonus should be divided.

Whether the bonus is active or passive, it should have a set definition of distribution rules. Three types of rules are currently valid: fixed, percent, and remaining percent. Fixed rule is the fixed amount guaranteed for the receiver, while percent rule is calculated by the percent value multiplied times the total amount of the bonus. The remaining percent rule is separate from the fixed and percent rules, and consists of the remaining amount multiplied times the percent value.

For each rule, it’s also necessary to assign the receiver. The receiver isn’t limited to only one address, but also can be the holder of one fungible token, and each holder can receive the amount according to his balance in relation to the total supply of that fungible token. Also, the stakeholders of fungible tokens here are not limited to the fungible token used for profit, but every fungible token registered on everiToken is acceptable. So, it's possible to issue one `bonus token` solely for-profit distribution, and it will benefit from the transparency, fairness, and liquidity provided by everiToken.

During implementation, it’s required to take a snapshot of all the stakeholders’ addresses together with the balance when the receiver has more than one address. It will take much more storage because each stakeholder address will cost 34 bytes. We’ve highly optimized this situation, and in most cases each address will only cost 4 bytes to store. With one million stakeholders or more, the cost will be around 4 Mbytes versus 34 Mbytes. Due to the fine-tune optimization of our token database, the system can read and update stakeholder balances at an extraordinary low cost.

Lock Functions

Lock functions are supported on everiToken's system. It's allowed to lock both non-fungible tokens or fungible tokens for a period time. This depends on the conditions, which are set during the lock proposal. Whether or not the conditions are satisfied during the lock time, after a set period of time the unlocked assets will be transferred to different registered addresses. Currently, the lock conditions can only be adjusted by public keys, which means that during the lock time only approved keys for a given proposal can provide access.

Other Technical Details

Basic Chain

We do not want to reinvent the wheel. As a result, we have absorbed the excellent parts of the existing public chain system and improved on its weaknesses. We have adopted the basic framework of EOS because we believe this system is currently one of the most advanced, practical, and best designed blockchain platforms. We recognize that EOS has an excellent code structure, but we also see critical infrastructure and user issues that everiToken was born to remedy.

On this basis, we have independently developed the implementation of every operation for token circulation on everiToken. At the same time, we have focused on enhancing token-based characteristics by optimizing the data structure of EOS in order to get better performance.

There are many advantages to such a practice:

- EOS has a complete and well-tested framework. DPOS and other core mechanisms have been fully tested in projects like BitShare.

- Reusing the basic framework can reduce part of the workload, allowing us to focus more on optimizing operations related to everiToken.

- During this process, the improvement of the basic framework will be submitted to Github of EOS, which is in line with the spirit of the open source community.

There are two major forms of blockchain operations (actions) in EOS. One is the native code written by C++ that is compiled directly into the binary code. The other is based on the implementation of WebAssembly, which executes the code after the JIT compilation. everiToken removed the second form and implemented all codes natively.

Authorization Operation

everiToken's authorization operations mainly include multi-signing, weight calculation, threshold setting, and so on. Since the transfer of each token is independent of others, the transfer operation of different tokens can be executed in parallel. Additionally, since each group's permission status is independent, issuance and management operations can also be executed in parallel between different groups.

Each transaction is made up of a data packet plus a signature list. In the case of authorization verification, we only need to verify each signature. There is no relationship among the signatures, so authorization operations can be executed in parallel.

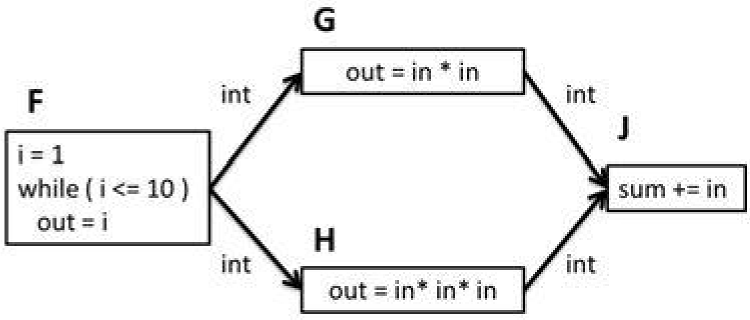
Execution Engine

In everiToken's system, since each token operation is completely independent, parallel processes do not require additional partition burdens. In addition, because the type of token operation is limited, the code is also built-in. As long as each type of operation is tested repeatedly, the system is completely stable.

The execution of one transaction can be split into several phases such as recovering signatures, authorization checks, computation, writing database, etc. All of the phases should be executed sequentially, but some phases are independent from one another across different transactions. One of these phases is called *signature-recovering*. There isn't any logical dependency on the signatures of each transaction, and each signature of a transaction is also independent. So, it's no problem to recover signatures in a fully parallel manner. Another of these phases is the *authorization checks*. It appears to be the same as recovering signatures at first glance, but imagine checking the authorization of two transactions of transferring tokens. Even though each token plays no role in another's function, if there are two transactions transferring the same token, then the system will encounter unexpected behavior if it continues to check parallelly. Because the owners of the token take part in the checking, it would be changed in the first transaction.

So, there is no way to execute some phases in parallel, but these situations can be planned carefully. What we have implemented is shown below in the *dependence graph*.

Our system parallelizes the data flow by using graph parallelism. Computations are represented by nodes and the communication channels between these computations are represented by edges.



Above is an example of how to calculate the sum of the sequences of the squares and cubes from 1 to 10. In our implementation, each node represents one phase of a transaction, and there is one scheduler who will receive transactions and split them to build the whole graph.

Suspended Transaction

A suspended transaction is a transaction that is completed after multiple delays. Ordinary non-suspended transactions are done in one go, and all the conditions must be met when the transaction is submitted. For example, all signers must sign together. However, in reality many transactions are completed with a process. The participants of the transaction may not be able to complete the signatures at the same time. The suspended transaction allows the signatures to be provided step-by-step until the transaction is successful.

everiPay / everiPass / EvtLink

everiPay / everiPass

everiPay/everiPass is a payment method born for face-to-face micropayments using everiToken public blockchain.

EvtLink includes the standard of QR code generation and the definition of communication protocol.

Here are some highlights regarding everiPay/everiPass/EvtLink:

- **Instant Clearance:** A transaction is a settlement.

- **Decentralization**: P2P payment, no centralized platform, no one can modify the data on-chain, and everyone can participate in pricing.

- **Most Secure**: The data and content in the blockchain cannot be forged or tampered with, so as to maximize the security and protection of user property.

- **Most Convenient:** Even if you can't connect to the Internet, you can complete the transaction. Payer/Payee doesn't need to input the amount of money manually. Payer and payee will receive a notification as soon as the transaction is successful.

- **Compatible**: everiPay/everiPass support all tokens issued on everiToken. Also, functional day-to-day operations such as a key to open a door are supported. The best part is that you can use it almost everywhere, simply with your phone.

- **Lightning Fast:** everiToken has achieved a very high TPS quickly, and transactions can be completed within 1 - 3 seconds depending on the quality of equipment or network.

- **Standardization**: Unique from technologies on the wallet side, EvtLink is a cross-wallet, cross-chain, and cross-app standard directly made for the whole ecosystem. You can use any apps to create or parse it.

Based on the above seven characteristics, everiPay/everiPass can provide the most secure, convenient, and user-friendly service in the face-to-face payment industry.

For everiPay/everiPass, the payee must use an app that supports parsing EvtLink and pushing transactions to everiToken. It has been made simple and easy, as we provide easy-to-use APIs and code examples for developers. It is similar to adding AliPay/WeChat support for your store, but even much easier.

Payee QR Code

A payee QR code does not support many features that everiPay provides. For example, payers must connect to the Internet to complete a payee QR code transaction, and both payers & payees must input the transaction amount manually. Also, they do not receive an automatic notification when the payment is finished.

However, payees don't need to use an app that supports this payment method. In fact, all payees need to do is simply use a wallet supported by everiToken on their phone to check if they have received the money from the payer. It is suitable for all types and sizes of vendors, as well as exchanges between people.

Using everiPay instead of a payee QR code is recommended for anyone because it is more transparent, safe, and user-friendly.

How EvtLink works?

EvtLink is the standard of binary format that represents everiPay/everiPass. everiToken public chain utilizes everiPay and everPass actions in order to execute the transaction on evtLink.

Here is the process of payments via everiPay/everiPass from a technical point of view:

1. The payer selects a kind of token to use, and then the wallet of the payer shows a series of dynamic QR Codes consisting of a unique 128-bit LinkId, a signature of the payer, and the symbol of the token used for payment. Note that the LinkId shouldn't be changed during QR Code swapping unless the related transaction is executed. This prevents the risk of a duplicate payment because the chain doesn't allow two actions on EvtLink with the same LinkId.
2. The payer's wallet application should then continuously query the transaction related to the LinkId by calling the API named 'get_trx_id_for_link_id' until it returns a valid transaction ID. The wallet should then change the LinkId the next time it shows a QR Code. Also, the wallet should show the transaction result by querying the transaction ID. Payer wallets don't need to send transactions directly.
3. Meanwhile, the payee scans the QR Code using his or her phone, scanner or smart gateway. After EvtLink is scanned and parsed, it should be wrapped within an action and then be pushed to the chain. After that, all of the chain nodes will be synchronized as a result, and the 'get_trx_id_for_link_id' will return the transaction ID.

Base42 Encoding

Base42 is an encoding algorithm for binary-to-string conversions. It is similar to hexadecimal encoding, but instead uses 42 as its base and correspondingly uses a unique alphabet sequence. The characters in the alphabet are the same as the characters in the encoding of a QR Code's alphanumeric mode, so it's efficient to pack base42-encoded string to a QR Code. This results in a smaller QR Code that allows for more convenient scanning.

On everiToken, *base42* will be used to encode EvtLink’s content.

Part III. Economic Model

Gas Fee/Fuel (EVT)

In order to avoid such attacks on the system as DDoS, to provide stake for the DPOS vote, and to give a reasonable reward to the producers, we will issue EVT as our fuel. Any operation will charge a certain EVT as a service fee, which will be a reward for the producer. The number of EVT charged will float automatically, and the fees collected are primarily to prevent malicious attacks and will not affect most users' regular use.

The method of generation and transfer of EVT is the same as that of the mainstream blockchain's encrypted currency. EVT is used to reward the resources provided by producers and prevent malicious behavior.

150 million EVT (15% total) will be given to the core team (14% for the five co-founders of everiToken and an extra 1% for core contributors).
400 million EVT (40% total) will be given to those community members who build apps based on everiToken and greatly contribute to the everiToken ecosystem by providing technology, resources, promotion, funding, and so on.
450 million EVT (45% total) are for investors of multiple rounds.

All of the services on everiToken will cost a service fuel fee.

$$ServiceFuelCost = FuelUsed \times R$$

In this formula, $FuelUsed$ is the price of a specific action. The unit of the price is EVT. R represents the **adjusting rate**. BP nodes can independently decide at any time to make a **rate hike** when the chain is too busy or under attack. They can also make a **rate cut** if the price of EVT is too high. Actual R is calculated as the median number of 15 BPs.
Users of the chain can assume R is 1 for the first time they call an API. Provided R has not been changed by the BP, the call will be completed. If R has been changed, the call will fail with the value of R from BP's response. The user would then have to try the action again.

For example, let the price of *creatingAccount* API be 2 EVT.
Usually a user is able to call *creatingAccount* API with 2 EVT.

If BPs make a rate hike to R = 1.1, then the price will change to 2.2 EVT.

We will use the median number of all the distributions of R in block producers. If 3 producers suggest R as 1.15, 5 producers as 1.2, 2 as 1.1, 2 as 1.3, and 1, 1.4 and 1.45 with 1 producer, then the final value of R is 1.2.

Pinned EVT

A Pinned EVT is similar to EVT but cannot be transferred. It can only be used as a fuel fee. Converting from EVT to Pinned EVT is allowed. EVT's exchange rate against the Pinned EVT is always 1. **Since Pinned EVT is not a currency**, it is safe enough to airdrop Pinned EVT to someone.

Generally, one should not convert EVT to Pinned EVT, as they are able to use EVT to pay for fuel fees. If one decides to convert EVT to Pinned EVT, the Pinned EVT will automatically bind to the receiver, hence its name **Pinned EVT.**

Pinned EVT belongs to an account and cannot be transferred to others. It is convenient and safe to airdrop Pinned EVT to users. Companies and organizations are able to convert EVT to Pinned EVT and post them to specific accounts. Pinned EVTs cannot be transferred across addresses.

A **Payer** is the account that pays for a given transaction. everiToken allows users to specify payers in a transaction. This is useful for creating accounts. For safety, payers should have extra signatures for the transaction.

Each domain has a special Pinned EVT balance.
The chain prefers consuming the domain's Pinned EVT balance (if not zero) during actions like transferring or destroying tokens in the domain.
Users are able to prepay for a domain's Pinned EVT balance via their EVT.

Extra EVT Issuance

The initial volume of EVT is 1 billion. The chain might issue extra EVT on a yearly basis. The actual issuance will be decided by everiToken's on-chain governance committee. We will not issue extra EVT until January 1, 2020 at the earliest.

Block Producers(BPs)

- Number of BPs: Dynamic

We give few permissions to BPs so it is very hard for BPs to do evil. The only evil BPs can do is DoS (Denial of Service). To balance the earnings of BPs and ensure decentralization, we use a dynamic count which is equal or greater than 15. In 2019 we will actually use 15 BPs. For the following years, the count will be decided upon by the on-chain governance committee.

Part IV. Ecosystem

Tools

everiWallet

As the name implies, everiWallet is an everiToken wallet which supports both web browsers and mobile phones. Please visit here for more information: https://www.everiwallet.com/

EVTJS

EVTJS is everiToken's API binding library for JavaScript and supports both NodeJS and browsers. It is also supported by everiSigner, so you can use this library to build web apps on everiToken easily. Please visit here for more information:
https://www.github.com/everitoken/evtjs

evtScan

evtScan is the blockchain browser of everiToken. Anyone can search for specific information on all the present blocks generated by the nodes in the everiToken mainnet. This includes the details of transactions, accounts, groups, and domains on the chain, as well as statistics and analytics. For developers, evtScan is an efficient tool to confirm whether information is properly linked to the chain. For users, it provides a method of verifying the authenticity of transactions. Please visit here for more information:
https://evtscan.io/

Decentralized On-chain Governance Committee

everiToken public chain will have a decentralized on-chain governance committee to decide important things like count of BPs and extra issuance of EVT. The future is under development, and the committee is expected to be online before 1st Jan 2020.

Escrow Company

everiToken involves itself with the assets or coins of users except for the token ID. The value of a token is endorsed by **escrow companies**. The escrow companies can sign an extra signature during the issuing of tokens, so everyone can trust the token if he/she trusts the company who makes the signature on the token. It is just like SSL.

Part V. Conclusions

The token economy is well on its way to touching every corner of the globe. Ethereum and EOS smart contracts were a good start, but they are not suitable for developing a token economy that all people of the world can utilize.

everiToken was born with the goal of creating token-based blockchain technology that benefits everyone, everywhere. We have built a revolutionary system that makes it low-cost and simple for developers, businesses, and end-users to issue, transfer, and verify the use of tokens within our system. Our safe contracts have removed Turing completeness, but as a result the abstraction and complications within the system are greatly reduced. Instead of constantly creating custom models, we created the one-size-fits-all model, becoming the preferable solution for over 99% of people. We have improved the speed, security, operability, stability, and supervision needed to create an efficient and thriving token economy, while providing a decentralized platform for all people of the world to learn, create, interact, and truly exchange value digitally. Join the token economy revolution and visit our website at www.everitoken.io

Founders

Hengjin Cai, Chief Scientist

Dr. Hengjin Cai is a professor and Ph.D. advisor at the School of Computer Science at Wuhan University since 2005. He is an expert-in-residence at the Global FinTech Lab, a visiting researcher at the Shenzhen Institute of Advanced Technology of the Chinese Academy of Science, and an expert committee member of the China AI and Big Data Committee of 100. He is engaged in SSME (service science, management, and engineering), AI & blockchain technologies, and recently published a book named *A Blockchain System with Integrated Human-Machine Intelligence*. In 2017, he won the WU Wenjun Artificial Intelligence Science and Technology Award. He received the President's Award for Extraordinary Contributions to Teaching at Wuhan University in 2012. As a dedicated advisor, he has led students in winning more than 80 prizes in influential competitions across China and the world including the Microsoft Imagine Cup, Microsoft and Morgan Stanley Cup of High-Performance Computing in Finance, Intel Cup National Collegiate Software Innovation Contest, and China College Students' Entrepreneurship Competition.

Brady Luo, CEO

in Brady-everiToken

Brady is a true believer in the global token economy based on blockchain technology. He received his undergraduate from Beijing University of Aeronautics and Astronautics in electrical engineering, a master's degree in finance from Brandeis University in the United States, and studied blockchain strategy curriculum at Said Business School at the University of Oxford. A natural and serial entrepreneur, he was elected into the third batch of the Shanghai 1000 talents plan (venture group) and sold two of his past startups. He worked as an analyst for nearly four years at US top 10 fund manager New York Oppenheimer Funds in the alternative asset investment group (New York City) and Japan's largest financial group, MITSUBISHI UFJ securities (Tokyo headquarters, Hong Kong, and Shanghai).

Bozhen Chen, COO

Bozhen has rich experience in government project operations and specializes in communications and public relations. He graduated from Aston University with a Bachelor of Science in Business Administration. Bozhen has worked with e-commerce providers, apparel supply chains, B2B services, and governmental organizations. Throughout this time, he developed a strong set of execution, communication, and public relations skills across a variety of industries and interests. He is the permanent host of the Internet Conference, leader of the Electronic Commerce Public Service Center of Tongxiang, and director of the Youth Internet Entrepreneurship Service Center. He has won numerous awards as one of China's youth leaders including Outstanding Youth in Jiaxing and the 2018 Motivated and Kind Youth awarded by the Communist Youth League of China.

Ceeji Cheng, CPO

Ceeji is a full-stack developer and experienced system architect with more than 10 years of software development, entrepreneurial, and management experience. He was the first-prize winner at the National Informatics Olympiad and previously worked at his own start-up (as CTO and co-founder).

Harry Wang, CTO

Harry is an experienced system architect and engineer with over 10 years of expertise operating in the finance and Internet industries. He previously worked at Tianfeng Securities in Shanghai before taking part in the founding of a quantitative hedge fund company as a technical partner. He developed a high-performance quantitative trading system that operates today with multiple markets and products worldwide.