



everiTOKEN

TECHNICAL WHITEPAPER

Version 2.5

© 2018, everiTOKEN

Zug, Switzerland

DISCLAIMER

- This everiToken Technical White Paper is for information purposes only.
- This White Paper does not represent any express or implied warranty, proof, expectation and so on.
- The technical specification or technology realization methods written in the White Paper may change over time.
- This technology team may be disbanded or reorganized at any time, or the loss of core technicians will lead to the failure or part realization of the project.
- This White Paper is a Tech Preview, provided “as is”. Neither the project team nor any numbers of the project is responsible for any content that is able to achieve or fail to achieve this White Paper.
- The “Token” in this White Paper does not have any practical value, only a proof that you have obtained the “Token” by means of digital encryption, and its only purpose is to confirm your permission for the “Token”.
- Any event in the blockchain or any of its derivatives, which are run by the technology described in this Technical White Paper, is automatically generated by program automation, and we are not responsible for its consequences. The consequences are responsible for individuals or organizations who use or do not use the blockchain.
- Anybody can use all the content of this Technical White Paper on the premise of non-commercial use, and do not modify or garble the Technical White Paper. But we are not responsible for any consequences caused by using the content.

Content

Part I. Background and Vision	1
The Arrival of Token Economy	1
Competitive Analysis	2
Summary	5
Part II. Technology of everiToken	6
Safe Contract	6
Database	6
Token Based	7
Security.....	15
Consensus.....	17
Other Technical Details	17
Part III. Economic Model.....	22
Fuel (EVT)	22
Pinned EVT	23
Extra EVT Issuance.....	23
Other Information.....	23
Part IV. Ecosystem	25
Tools	25
Application Scenarios	25
Timeline.....	27
Part V. Conclusions	28
Founders	29

Part I. Background and Vision

The Arrival of Token Economy

As of April 2018, blockchain technology has been around for about ten years. However, 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 the blockchain (hereinafter referred to as “on chain”) are basically a variety of tokens, with a total market value of about \$300 billion. 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 “tokens” a payment currency, but currently they primarily serve as digital assets and do not play the role of currency. Any emphasis of a “digital currency” is more of a name than a reality.

In fact, the issuance of currency is a very important right, which embodies politics, and monetary power must belong to the state. Therefore, it is very difficult for Tokens to replace 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 the blockchain (hereinafter referred to as “off chain”), and there is limited intersection between blockchains and assets off chain.

But is Token really just a token? Not really. The original meaning of Token is “symbol, signaling”, and should be more appropriately considered a certificate rather than a digital currency. Such certificates can represent all sorts of rights and interests, including identity cards, diplomas, access keys, event tickets, card coupons, and a wide variety of proofs of rights and interests. Looking back on history, all civilizations of human society can be said to be based on the proof of rights and interests. All accounts, ownership, qualification, proofs, and so on are all proof of rights and interests. As Yuval Grali said 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 are all digital, electronic, and cryptographically-protected to verify their authenticity and integrity, then human civilization will be revolutionized. We call this a “Token Economy”.

Running certificates on blockchain provides a solid foundation of trust not provided by any traditional centralized infrastructure. Therefore, if a certificate is the front-end economic unit in the Token Economy, the Blockchain is the back-end technology in the Token Economy. The two are integrally linked.

Competitive Analysis

As a public chain born for the Token Economy, we think everiToken has two main competitors, Ethereum and EOS. We will analyze from SW and OT two aspects.

SW (Strength and Weakness):

Combining the economic progress of human history and the technological progress of blockchain, we think 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 represent an 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 general 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 transfer of Tokens within seconds at the volume of hundreds of millions of Tokens simultaneously.
- **Flexible Authorization Management:** A simple and elegant 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 codings.

Let's look at Ethereum and EOS:

Ethereum: ERC20/ERC721

The main way to achieve Token Economy with Ethereum is to develop smart contracts based on the ERC20 and ERC721 protocols. Among them, ERC20 supports FT (Fungible Tokens) and ERC721 supports NFT (Non-Fungible Tokens). However, there are some serious problems.

- **TPS:** presently, Ethereum can only support fewer than 20 transactions per second, unable to meet all the practical needs of Token usage and circulation;
- **COST:** the implementation of smart contracts in Ethereum costs gas fees every step. For functions with complex business logic (such as multi-person holding, supervision, legality, etc.), the cost may be high and uncontrollable.
- **Popularize:** the realization of Token Economy in Ethereum is based on smart contracts, which are non-accessible for non-developers without the use of third-party applications.
- **Non-Standardization:** since different smart contracts may take completely different development ideas, the metadata of these virtual Tokens is unable to connect and may be 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 they own.

EOS

EOS launched its mainnet in June 2018. As EOS is aimed at addressing the problems of Ethereum, which are possible to solve via EOS solutions. However, there are still some new issues:

- **Security:**

Token transactions may correspond to extremely precious and non-renewable real entities, and thus it is important that there are no security problems. However, development based on smart contracts is limited by the 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 **Web Assembly**, which is relatively new and still in the test (Beta) stage. Additionally, EOS's smart contracts 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, third-party applications must rely on, and users must trust the quality of the code of the third-party. Thus, the control of assets is not in users' own hands, but rather depends on

the guarantees of the third party.

- **Non-Standardization:** like Ethereum, the metadata of different smart contracts cannot interact with each other or cooperate.
- **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 reduce the speed of the operation. Due to the possible conflicts among different data and functions, using multithreading execution 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, changeable, and lack consistency. However, smart contracts take time to develop and test, and may be unable meet the diverse of the market in a short time span. This will be a hindrance to the development of the Token Economy.

A 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 scenario that everiToken cannot satisfy. But we think everiToken can meet 95% of the demand in Token Economy, and everiToken is the safest, friendliest public chain and almost free for normal people.

OT (Opportunity and Thread)

Besides these strengths of everiToken, we create the EvtLink standard which is used to generate the everiPass/everiPay QR code. everiPass/everiPay is a payment protocol born for **face-to-face micropayments** using everiToken public blockchain.

everiPass/everiPay includes the standard of **QR code** generation and the definition of communication protocol. Based on everiToken **public blockchain** with five characteristics:

Instant Clearance, A transaction is a settlement.

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



Most secure, The data and content in the blockchain can not be forged or tampered with, so as to maximize the protection of user's property security.

Compatible, everiPass/everiPay support all Tokens supported by everiToken, not only currency but also tokens and points, even a key to open a door, and you can use it almost in 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, everiPass/everiPay can provide the most secure, most convenient and enjoyable services in face-to-face payment.

Some threats still exist. As I said, Ethereum and EOS can be a great public chain in the Token Economy. If the Ethereum can really enhance its TPS through some ways like sharding, It will become a strong competitor. Also smart contracts have many problems now, and over time more of the flaws in smart contracts may be solved and improved, to a certain extent, to enhance the competitiveness of Ethereum and EOS. After all, they have more attention and more users currently. That's why everiToken really pays attention to the ecosystem and the practical application.

Summary

Based on the above analysis, we have designed a new concept that is perfectly suitable for blockchain applications, and propose a new public chain and ecosystem, **everiToken**, for the development of the Token Economy. The assets, certificates, and vouchers of the real world can be **digitalized** through the issuance of Tokens, and be easily used with unprecedented security, speed, and network compatibility.

Part II. Technology of everiToken

Safe Contract

Smart contracts are, in theory, 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 termination. As such, smart contracts often fail to provide trust and may be viewed as less reliable than traditional contracts or exchange.

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

Furthermore, SafeContracts have the added benefits of increasing accessibility and TPS. With the former, the inclusion of API's makes it simple for easy integration into existing workflows without writing chain-integration code from scratch. With the latter, the usage of API's allows for various translations types to be distinguished easily, and independent token transactions can be processed in parallel at faster speeds (upwards of 5000 TPS).

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 memory. With the increase of users and transactions over time, it is foreseeable that the demand for memory will increase significantly. This raises a high demand for the memory capacity of the block producers. Furthermore, if the program crashes or restarts, the memory data will be lost. To restore data, we need to repeat all operations in the blocks before we can restore them, 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 Key-Value database which has been fully verified and is used in the core cluster of Facebook.
- RocksDB is based on LevelDB, and provides better performance and richer functionality than LevelDB. It has also made a core optimization for 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 takes RocksDB as the 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, Token database also supports optional functions such as data persistence, quantitative backup, and incremental backup to solve the problems of cold startup and so on.

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

Token Based

Overview

Born for Token Economy, a major difference in everiToken is Token Based token management method.

Tokens are different from digital currencies issued by central banks and encrypted currencies (like 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, or a particular place, or a time service provided by a particular

entity. Tokens are divided into two types: Fungible Tokens and Non-Fungible Tokens. 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 start with Non-Fungible Tokens.

Non-Fungible Tokens

Before understanding the Non-Fungible Tokens, let's consider a large number of stones on the beach. In the real world, every stone on the beach has different weight, appearance and rock type. There are no two identical stones, just like two leaves can not be found. Besides, you can't easily combine two stones together. Therefore, we say that every stone is "indivisible" and "not to be combined".

An example in block chain is the CryptoKitties, which was once a hot game in the block chain world. Each cat has unique numbers and attributes.

An NFT Token is similar to an individual of a kind of stone in the real world, or a block chain cat. They are natural different in the real world, and so are in our system.

Generally speaking, NFT Token is divided into different categories according to their different value types. We can categorize the same kind of NFT Token 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, **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 Token ID, the structure also saves the Owner of the Token and other necessary information.

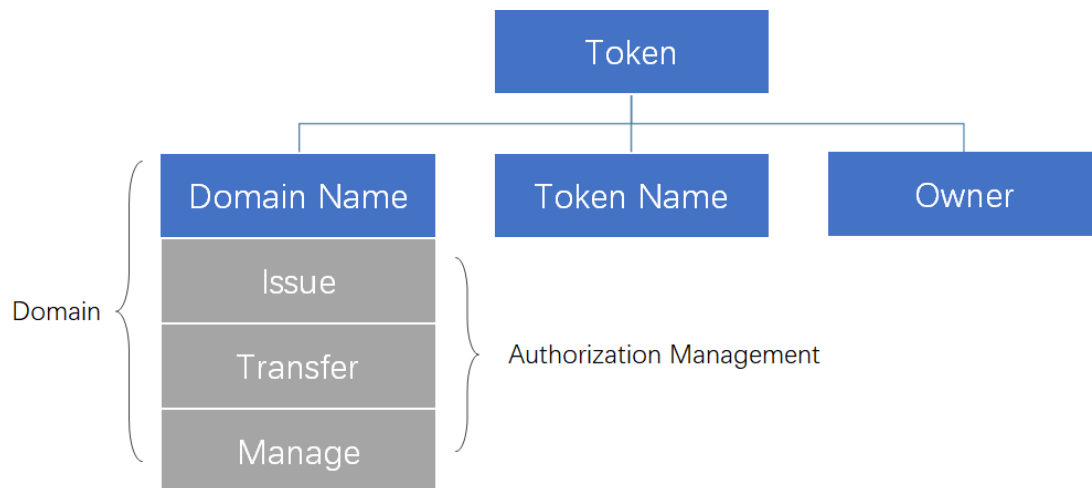


Figure 1. The Token Structure in everiTOKEN

The Domain details can be queried by the Domain Name. Each Domain also describes 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 essence of Token transformation is to change the ownership of Tokens. Every Token has an **Owner group** (there can be one or more owners). When a change of 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 in authorization management: Issue, Transfer and Manage.

- (1) **Issue** is the right to issue Token in this domain.
- (2) **Transfer** is the right to transfer Token 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 **Authorization Tree**. As the root, each permission has a threshold, and it is connected

to one or more actors.

Actors

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

A group can be a club, a company, a government department, a foundation, or even just an individual. A group contains 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 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. Therefore, the mechanism is called the **Group Autonomy**.

When a group is initiated, the system generates a group ID automatically. When the issuer designs Authorization Management in a domain, it can be invoked by directly referencing the existing group ID to its permissions system. Due to the Group Autonomy, each group can be reused conveniently.

The owner of the Token has a special group whose name is fixed to Owner and is a collection of Token owners. The special feature of this group is that each Token may be different, 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.

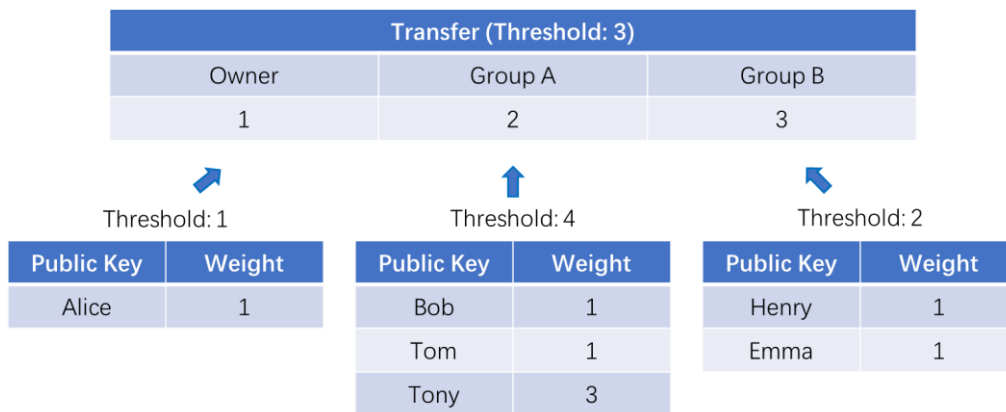


Figure 2. A Transfer Permissions

Figure 2 describes the Transfer permissions in 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 authorized by at least Bob/Tony or Tom/Tony, and Group B is 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 different domains 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 it; a concert ticket is useless after the concert, but a fixed parking space's owner may change with time.

When issuing Token, the issuer of 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 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 Token. 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 Local Property Bureau and owners.

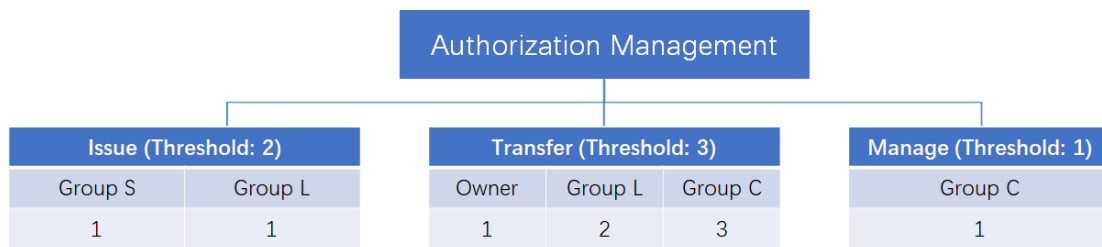


Figure 3. The Structure of Authority Management

The Issuer and initial Owner of Token in the domain is the SPV, and Group *S* represents SPV, 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 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 an accident, such as a Token owner who has passed away or lost the 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 a part of a Token's ID is lost (which is likely to happen), or if both SPV and other Token owners agree to add new Tokens, they can add new 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 the Token in the domain.

Fungible Tokens (Points)

Issuance



Everyone can issue points after registering a symbol. Users can set the total number of Tokens with this symbol or not set it. Then, users need to set the number of tokens they want to issue at this time.

Transfer

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

Other Details

Each account will record the number of tokens with different symbols it holds. There will be an independent key-values 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 specified symbol. This function is called token allowance, which can be used in token exchange.

Token-based Transaction Model

Overview

everiToken uses **token-based** transaction model for non-fungible tokens.

In short, for a token-based ledger, we create a record of token id and the ownership when it is initially issued, and then infinitely append new ownerships whenever it's transferred to others. It is a very efficient way for non-fungible tokens. Updating and querying information for a token is super fast as we have a indexed token database especially designed for this purpose.

Token-based transaction model is invented by several core team members of everiToken, and is proved to work perfect for NFTs on everiToken. everiToken could be considered as a status machine that changes its status when and only when executing actions on every irreversible block. For a blockchain based on token-based transaction model, like everiToken, it might divide the database into two parts, one is Token DB, and another is Block DB. Both Token DB and Block DB should be a versionized DB for fast rollback when a block is reversed. For example, everiToken uses RocksDB for Token DB.

Token DB

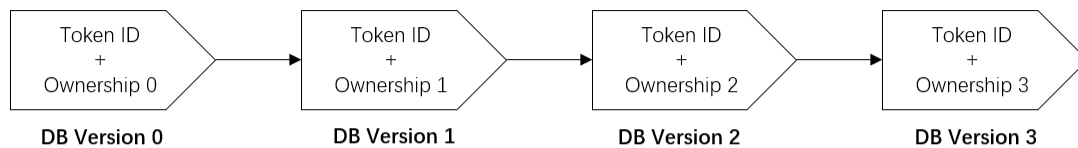
Token DB is a indexed DB for quickly finding the newest status of the blockchain such as the ownership of tokens and the balance for fungible tokens of accounts on the chain. When executing a transaction, changes the owner in DB. Token DB is a append-only database so new data will be appended, the version of DB will be increased, and old value won't be removed instantly. Old versions could be used to rollback the value if the block is reversed and will eventually be garbage collected.

Block DB

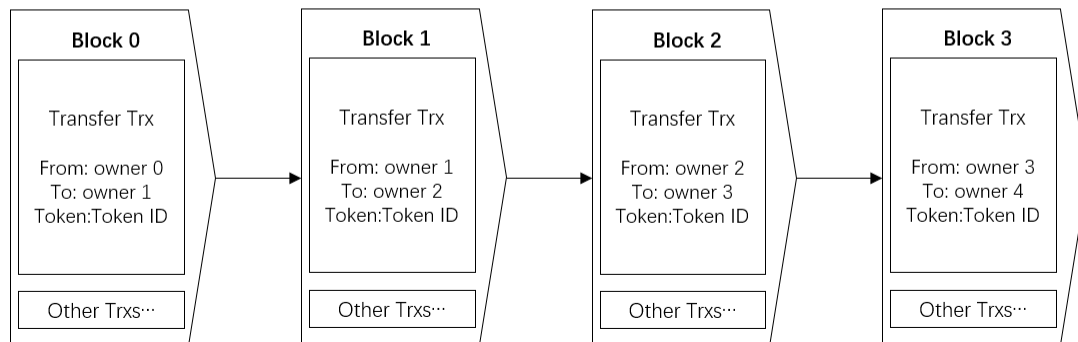
Block DB is in charge of storing all the original irreversible blocks of the chain. Each Block stores all the detail information including the name and parameters of actions executed, the signatures that is on the block and some extra information.

The following graph shows how do two kinds of databases work together for NFT tokens:

Token DB



Block DB

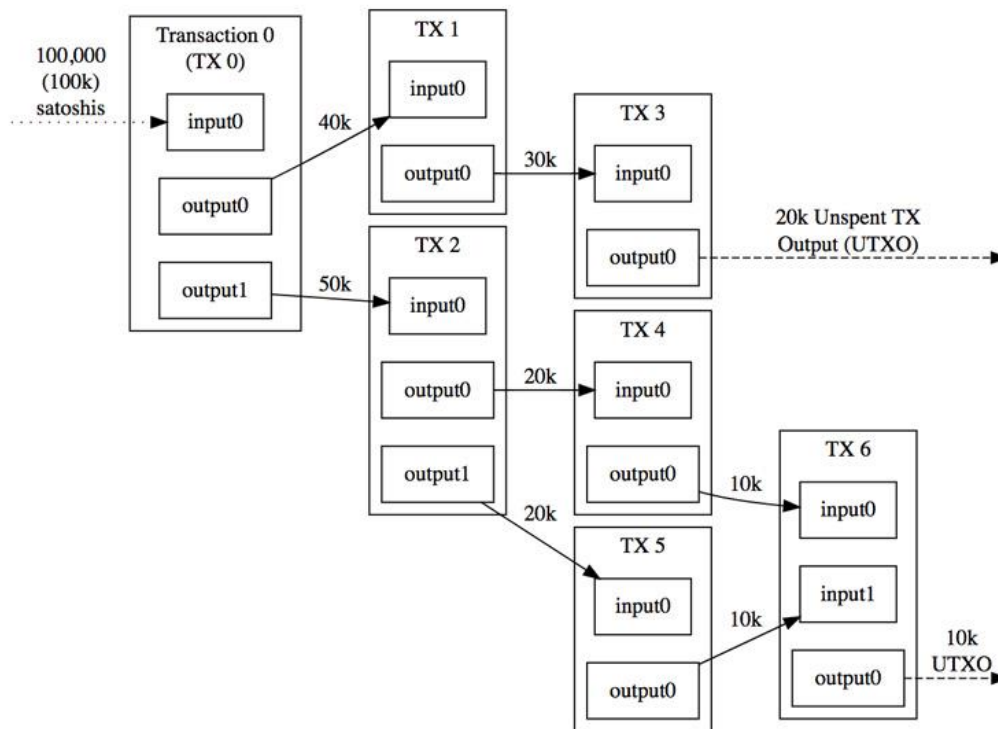


Comparing to other transaction models

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 and 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

(The picture is from bitcoin.org)

As you can see, UTX 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 NTF, it is a FT. It is useless to keep a unique id for every UTXO. (everiTOKEN moves it to NFT use.)
- UTXOs are one-off. It's a waste of computing resources and disk volume to store that huge amount of UTXOs.

b) Balance-based

Balance-based transaction model is just like what does a bank do. You create an account in a bank and then save money into the account. The bank changes the balance of your account. There is no something like UTXO. It's more efficient than UTXO because it only changes a number in the database. But obviously it is not suitable for NFTs.

Security

Focusing on Token’s related functions, everiToken streamlines unnecessary abstractions, which not only greatly increases efficiency but also has remarkable safety performance. Although the types of Token in 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 in consequence.

Script (everiSigner)

everiSigner is an offline signer. The whole signing process is done within that 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 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.

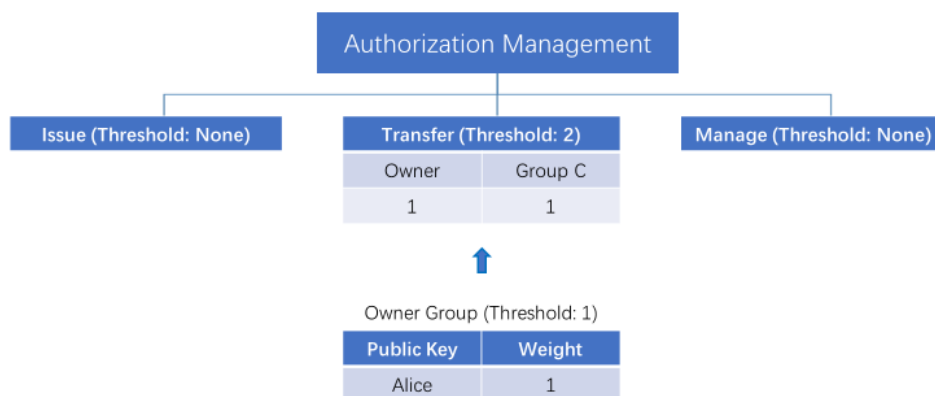


Figure 4. Company C provides get back private key Service

Of course, Group C can steal Alice’s Token, but all operations will be recorded on the chain, which will destroy C’s credibility.

Consensus

everiToken uses BFT-DPOS as its consensus algorithm. DPOS is proven to be capable of meeting the performance requirements of applications on blockchain. Under this algorithm, those who hold EVT tokens 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 exactly 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.

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.

Byzantine Fault Tolerance is added by allowing all producers to sign all blocks, so long as no producer signs two blocks with the same timestamp or the same block height. Once 11 producers have signed a block the block 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.

Other Technical Details

Basic Chain

We do not want to reinvent the wheel. Therefore, we absorb the excellent part of the existing public chain system, in order to achieve “The students surpass the teacher”. We have adopted the basic framework of the EOS as the code base, because we think



EOS is currently one of the best designed, most advanced, and practical blockchain platforms, and EOS has excellent code structure.

On this basis, we have independently developed the implementation of every operation for Token circulation in everiToken and optimized it. At the same time, focusing on the characteristics of being Token Based, we optimize 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 basic 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.
- In this process, the improvement of the basic framework will be submitted to the Github of EOS, which is in-line with the spirit of the open source community.

There are two forms of blockchain operations (Action) in EOS. One is Native code, that is the code written by C++, compiled directly into the binary code. The other is based on the execution of Web Assembly, or the code executed after the JIT compilation. We removed the second form, and implemented all codes natively.

Authorization Operation

everiToken's Authorization Operations mainly includes multi-signing, weight calculation, threshold setting and so on. Since the transfer of each Token is independent of each other, the transfer operation of different Tokens can be executed in parallel. Additionally, since each Group's permission group is independent of each other, issuance and management operations can also be executed in parallel between different groups.

Each operation is made up of a data packet + signature list. In the case of authorization verification, we only need to simply verify each signature, and there is no relation between the signatures and can be executed in parallel.

Execution Engine

In everiToken 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 can be completely stable.

everiToken divides the production processes of each block into three stages: preparation stage, transfer stage and ending stage. When creating a new domain and issuing Tokens, the system will process in preparation stage to ensure correctness of operation. Then we do parallel processing on the operation of Token owners' change and distributions, so as to ensure the best performance. Finally, we focus on exception handling at the ending stage and persist the results.

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. For example, the participants of the transaction may not be able to complete the signature at the same time. The suspended transaction allows the signature to be provided step by step until the transaction is successful.

EvtLink / everiPass / everiPay

everiPay / everiPass

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

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

Here are highlights of EvtLink / everiPass / everiPay:

- **Instant Clearance**, A transaction is a settlement.
- **Decentralization**: P2P payment, no centralized platform, no one can modify the data on chain, everyone can participate in pricing.



- **Most secure:** The data and content in the block chain can not be forged or tampered with, so as to maximize the protection of user's property security.
- **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 notification as soon as the transaction is successful.
- **Compatible:** everiPay/everiPass support all Tokens supported by everiToken, not only currency but also tokens and points, even a key to open a door, and you can use it almost everywhere, just with your phone.
- **Fast:** The everiToken has achieved high TPS, we think that a transaction can be completed within 1 - 3 seconds considering the situation of equipment and network.
- **Standardization:** Different with technologies from wallet side, EvtLink is a cross-wallet cross-chain 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, most convenient and enjoyable services in face-to-face payments.

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

Payee QR Code

Payee QR Code does not support many features comparing to everiPay, for example, payers must connect to Internet for payment, and payers & payees must input amount of money manually and they won't receive notification when payment is finished automatically.

However, payees don't need to use apps that supports this payment method. In fact, what payees needed to do is just using a wallet on their phone to check if they received the money from the payer. It is suitable for very small stores or persons.

Using everiPay instead of Payee QR Code is recommended for anyone who is able to.

How does EvtLink technically work

EvtLink is the format of QR Code used to execute everiPass / everiPay. everiToken public chain use *everipass* action and *everipay* action to execute the transaction of evtLink. it also provides a struct named `evt_link` to represent EvtLink.

Here is the technically process of payments via everiPay / everiPass:

1. The payer select a kind of token to use, and the wallet of the payer show a series of dynamic QR Codes consisting of a unique 128-bit LinkId, a signature of the payer and the symbol of the token for payment. Note that the LinkId shouldn't be changed during QR Code changing unless related transaction is executed. Else the risk of duplicate payment can't be ignored. The chain doesn't allow two actions with EvtLink with a same LinkId.
2. The wallet of payers should then continuously querying transaction id related with the LinkId by calling a API called `get_trx_id_for_link_id` until it returns a valid transacion id. After that the wallet should change the LinkId the next time it shows QR Code. And the wallet should show the transaction result by querying this transaction id. Wallets of payers don't need to send transactions directly.
3. Meanwhile, the payee scans for the QR Code using their phone, scanner or smart gateway. After EvtLink is scanned and parsed, it should be wrapped in a a action and then be pushed to the chain. After that, all the chain nodes will synchronize the result so `get_trx_id_for_link_id` will return the transaction id instead of 404.

base42 Encoding

Base42 is a encoding for binary-to-string conversion. It is similar to hexadecimal encoding, but instead use 42 as its base and correspondingly use a different alphabet. The chars in the alphabet are the same as the chars in the encoding of QR Code's alphanumeric mode so it's efficient to encode base42-encoded string to QR Code. Thereby the size of QR Code could be smaller.

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

Part III. Economic Model

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 only used to reward the resources provided by producers and prevent malicious behavior, so it has no other value.

150 million EVT (15% total) will be given to the Core Team.

400 million EVT (40% total) will be given to the Community Core contributors.

450 million EVT (45% total) are set for Placement.

All the services on everiToken will cost Service Fuel Fee.

$$ServiceFuelCost = FuelUsed \times R$$

In this formula, *FuelUsed* is the price to call specific action. The unit of the price is EVT. *R* means **Adjusting Rate**. BP nodes can independently decide at any time to make a **rate hike** when the chain is busy or under attack. They can also make a **rate cut** if the price of EVT is too high. Actual *R* is calculated as median number of 15 BP's.

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 should fail with the value of *R* from BP's response. Then the user could try again.

For example, Let the price of calling *creatingAccount* API be 2 EVT.

Usually a user is able to call *creatingAccount* API with 2 EVT.

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

We will use the middle number of all the distributions of *R* in block producers. If 3 producers suggest *R* as 1.15, 5 as 1.2, 2 as 1.1, 2 as 1.3, and 1, 1.4, 1.45 both have 1 producer. 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 fuel fee. Converting from EVT to Pinned EVT is allowed. The EVT's exchange rate against pinned EVT is always 1. **Since pinned EVT is not 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 charge fuel fee. If one decide to convert EVT to pinned EVT, the pinned EVT will automatically bind to the receiver, hence it is the named **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 EVTs and post them to specific accounts. Once completed, the pinned EVTs cannot not be transferred again.

A **Payer** is an account who pays for a transaction. everiToken allows users specify payers in a transaction. This is useful for creating accounts. For safety, payers should have extra signatures in the transaction. If the Payer is set, pinned EVT is not allowed for the fee.

Each domain has a special Pinned EVT balance.

The chain prefers consuming the domain's Pinned EVT balance (if not zero) during actions like transfer and destroying tokens in the domain.

Users are able to prepay for a domain's Pinned EVT balance via their EVT.

Extra EVT Issuance

We will issue extra EVT per year.

- $R = \begin{cases} 0.05 - 0.005 \times Y & (0 \leq Y \leq 5) \\ 0.02 & (6 \leq Y) \end{cases}$
- Y defines the year of current time minus the creation year of the chain.
 - That is, the value of Y will grow from 0
- Actual currency issued is calculated by $(1 + R) \times \text{currentCurrencyVolume}$.

Other Information

Block Producers(BPs)

- Count of BP: 15

Different from EOS because having more BPs increases the cost, but we want to give more EVT to the community, not BPs.

15 BPs are enough for decentralization.

- In the first year BPs can get 1% of issued EVT, the second year BP's get 0.9%

Escrow Company

everiToken knows nothing about assets or coins except for token id. The value of a token is endorsed by **escrow companies**. Those Escrow companies can make an extra signature during issuing the tokens, so everyone can trust the token if he/she trusts the company who make the signature on the token. It is just like SSL.

Part IV. Ecosystem

Tools

everiSigner add-on

everiSigner is an open source add-on that allows users to run everiToken apps directly in their browsers. everiSigner includes a secure identity vault and user interface to manage identities on different sites and verify blockchain transactions. It stores users' private keys and is compatible with EVT, ETH, EOS and related plugins.

everiWallet

As the name implies, everiWallet is a everiToken wallet based on the everiSigner add-on. 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 Browser. 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 a blockchain browser of everiToken. Anyone can search for specific information on all the present blocks generated by the nodes in everiToken TestNet (and main net in the future), including 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 chain. For users, it provides a method of verifying the authenticity of transactions.

Please visit here for more information. <https://evtscan.io/>

Application Scenarios

In the network of Token economy, a blockchain records the issuance, confirmation of transactions, accounting, reconciliation, and liquidation. From upstream to downstream, all institutions can use everiToken public chain to conduct business according to their roles, including issuers, exchanges, circulation channels, and notarization platforms as well.

Only the relevant core information of Tokens is recorded on the chain. Irrevocability and the huge cost of misbehaviors contribute to the solution of trust issues. The original modularity satisfies the majority of needs in Token economy, and thus it is significantly faster and more user-friendly.

In the short term, everiToken chooses the following three application scenarios as entry points to facilitate cooperation and landing.

Coupons

EveriToken is building up a coupon system for North American House Buyers (www.beimeigoufang.com, a housing transaction system). NAHB provides a web platform to allow sub-users (registered house sellers or agents) to issue various coupons with NAHB signatures using everiSigner including signatures. The platform also supports transactions of those coupons. APIs are provided in everiToken to assure that all the Coupon Token related operations can be easily implemented, including issuance, transfer, management and verification, and both sub-users and end-users (people who want to buy houses) can operate directly on the web side. Every single issuing record can be tracked on everiScan, so it is not possible to issue excess Coupon Tokens. A Coupon Token has high practical value. For example, its initial price may only be 100 US dollars, but it can offset a value of ten times, that is, 1000 USD, when purchasing a house. NAHB, its sub-users, and also the end-users who eventually need to buy a house, all of them will benefit from the rapid and easy circulation based on the Tokenization of traditional coupons.

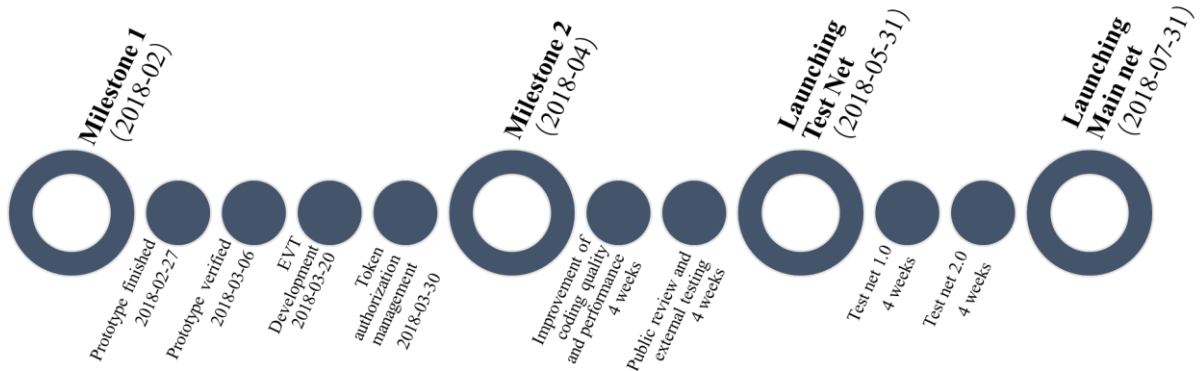
Gaming Protocols

FastX is another partner of everiToken. Addressing the slow congestion on Ethernet, FastX provides a protocol for Ethereum-based-games so that users can lock game data statuses to smart contracts. At the same time, FastX will generate the same information on a subchain. It is noteworthy that the complete game data are performed on the subchain, which requires only periodic interaction with the main chain to update those changes and conducts quick reviews via Merkle Trees. EveriToken can be used as a subchain of the game protocols due to its features of high security, convenient transfer, and simple management. All the game data changes are recorded directly on the everiToken chain. For the majority of games, it is unnecessary to keep all fields of data,

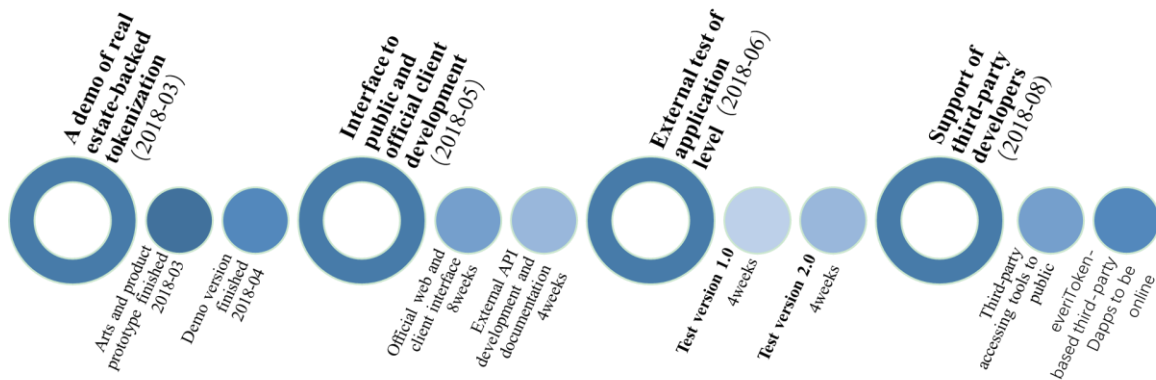
and it is reasonable to store the most important parts of the Token-related information.

Timeline

Public Chain Development Schedule



Application Layer Development Schedule



Part V. Conclusions

The Token Economy is coming, but Ethereum and EOS smart contracts are not suitable for developing Token Economy.

Aimed at the core idea of a fundamentally Token-Based blockchain technology, everiToken has built a special system that is easy for Token issuance, transfer, and verification. The system has removed Turing completeness, but it also reduces the level of abstraction within the system; improves speed, security, interoperability, stability, and supervision; and can achieve more efficient execution, allowing everyone in the world to understand, create, interact, and truly exchange value digitally.

Founders

Hengjin Cai, Chief Scientist

Dr Hengjin Cai is a professor and Ph.D. advisor in the School of Computer Science at Wuhan University since 2005, an expert-in-residence at the Global FinTech Lab, a visiting researcher at the Shenzhen Institutes of Advanced Technology of Chinese Academy of Science, and an expert committee member of the China AI and Big Data Committee of 100. He is engaged in service science, AI and blockchain technologies, and published the book BEFORE THE RISE OF MACHINES: The Beginning of the Consciousness and the Human Intelligence, winning the WU Wenjun Artificial Intelligence Science and Technology Award of 2017. He was elected in the first batch of Wuhan Yellow Crane Talents Plan, and received the President's Award for Extraordinary Contributions to Teaching at Wuhan University in 2012. As a dedicated advisor, he has led students to win more than 80 prizes in influential competitions in 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

Brady is a true believer in the global token economy based on blockchain technology. He is Beijing University of Aeronautics and Astronautics Electrical engineering undergraduate, Brandeis University financial graduate student, studied block chain strategy curriculum in Said Business School of University of Oxford. Entrepreneur, elected the third batch of Shanghai 1000 people's plan (venture group). Worked in New York Oppenheimer Funds alternative asset investment (CDO – Based asset securitization) and Japan's largest financial group, MITSUBISHI UFJ securities (Tokyo headquarters and Shanghai).

Bozhen Chen, COO

Bozhen has rich experience in government project operation, and specializes in communications and public relations. He graduated from Aston University, Bachelor of Science in Business Administration. E-commerce providers, apparel supply chain B2B services, social short videos, government e-commerce project entrepreneurs. He has rich experience in social capital and government project operation, and has strong execution, communication and public relations skills. As the permanent host of the Internet Conference, the electronic commerce public service center of Tongxiang and the director of the youth internet entrepreneurship Service Center who has won title of the country's rural youth leading leaders, 2017 the most beautiful Zhejiang youth leaders and so on.

Ceeji Cheng, CPO

Ceeji is a full-stack developer and experienced system architect with more than 10 years of software development experience, and rich entrepreneurship and management experience. He was the first-prize winner of 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 development engineer with more than 10 years of system development experience. He previously worked in Tianfeng Securities in Shanghai, then participated in private placement as a technical partner, and was responsible for quantitative trading system development.