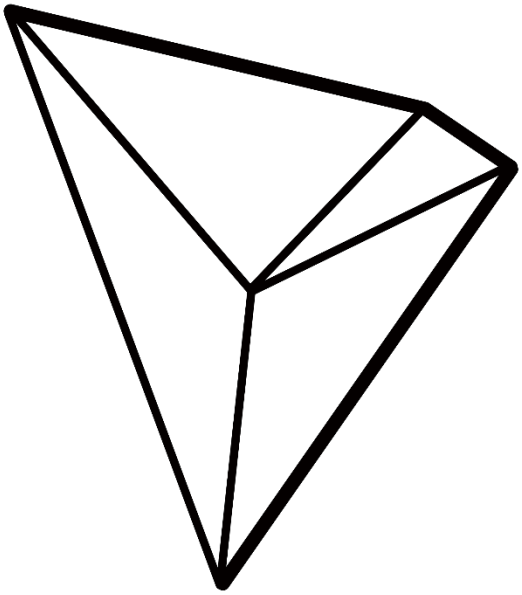
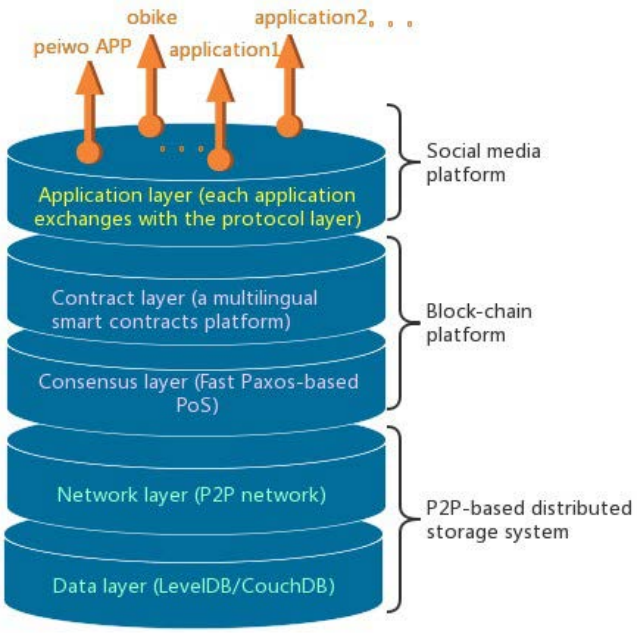


TRON

Content

I. Overall technical framework diagram



The overall technical architecture consists of three platforms:

1. Social media platform: this is an application layer;
2. Block-chain platform: a core block chain-based functional module is offered:
   (1) Contract layer: a multilingual smart contracts platform;
   (2) Consensus layer: a Fast Paxos-based PoS consensus algorithm.
3. P2P-based distributed storage system: a support located at the bottom:
   (1) Network layer: customized content-addressable P2P storage network;
   (2) Data layer: data storage is based on LevelDB/CouchDB.

A. Tron Social Media Platform

Through the use of many existing mature technologies, wave field (TRON), as a new content platform, provides security, scalability, and privacy, and simultaneously allows the participants to actively contribute to the processing capacity of their machine to build a user registration network. It also gives positive contributors the privilege to send advertisements to the whole network to incentivize (of course this group text messaging will be limited in number).

1. The user registers P2P network

Centerless, but secure user registration, is implemented through the blockchain mechanism, and the same mechanism has been applied in Bitcoin without the need for central authorization, which avoids double spend difficulties. The blockchain ensures

no duplicate registration, and the newly-registered users must obtain the confirmation of multiple blocks before taking effect, i.e. notarization. Each block is defined as:

$$Block_i = [i, H(Block_{i-1}), Nonce_i, SpamMsg_i, [UserReg_j, UserReg_{j+1}, ...]]$$

H ($Block_i$) provides Proof-of-Work to prove that the user received satisfying Nonce value in $Nonce_i$ space through violent solving; meanwhile occasional hash collision is avoided through verification. The difficulty of solving is determined by the difficulty value, and the number of blocks generated per hour is automatically set by the system, which is similar to the Bitcoin network.

$$UserReg_j = [Username_j, PUBK_j, Nonce_j]$$

New user j must broadcast $UserReg_j$ when registering online, and after receiving the broadcast message, other nodes must prove the proof-of-Work of H ($UserReg_j$), which will prevent denial of service attacks by false registration. This workload is much smaller than the workload of the blockchain; typically, a few minutes of computation can solve the problem.

The blockchain provides a mapping from the user name $Username_j$ to the user public key $PUBK_j$, a dictionary that can be publicly queried.

The node must verify the uniqueness of $Username_j$ before adding $Username_j$ to the new block, but there is an exception: if the newly-registered key is signed by the previously known public and private keys, then it may be replaced. In addition, the ID_j uniqueness and the proof-of-Work of $UserReg_j$ should be also proved when receiving the new block.

$Username_j$ also has the maximum size and the allowable character limit to protect the ID space from the hash attacks.

$SpamMsg_j$ is a broadcast message (called "Promoted" message) that sends the Promoted message as a reward to nodes that actively participate in block generation.

2. Routable DHT overlay network

The second network is a P2P overlay network similar to Kademlia, which is mainly used for resource storage and searching content, and also for direct delivery of notifications between users.

Using the user's ID as the network node ID seems like a good choice, but this leads to the exposure of the user's identity and location, breaking the system's privacy. Therefore, hashing the IP address and the node's port number to identify the node and taking it as the node's name in the DHT network can also avoid the sybil attack:

$$ID_{node\_j} = H([IP_j, port])$$

The package delivered from ID_{src} to ID_{dst} in DHT network is defined as follows:

$$Packet = [ID_{dst}, ID_{src}, SIG_j(payload), ID_j)]$$

The payload is signed through user ID_j, the ID_j may be different from other users of ID_{src} during package retransmission/refresh.

These functions constitute the third layer function of the concept model of the DHT overlay network. The above layer is the "application layer", which provides the data storage primitive to PUT and GET, PUT is defined as follows:

$$payload_{PUT} = [target, value, time, seq] \text{ where}$$
$$target = [owner, resource, restype] \text{ and } ID_{dst} = H(target)$$

Before accepting the storage request, the destination node needs to do the following rule checking:

- ID_{dst} = H（target）: ensure the correct calculation of the destination address;
- ID_{dst} is the neighbor of ID_{node} that actually receives the request;
- ID_j = H （owner）: verify when restype is "single";
- seq is larger than stored old value seqold, which is also verified when restype is "single";
- time is a valid time (i.e. not a future time value).

Restype defines resource types. There are two possible values, "single" and "multi." Single represents resources that can only key owners can update; multi represents responses from different users (i.e., replies to a post). For a single type, the node stores only a single value, and for the multi type, the new PUT request appends the value to the list. Both types of storage can set the expiration time, and the corresponding storage will be deleted from the system after the setting time, so that the expired data will be automatically cleared. Primitive data retrieval GET can also operate on two types of storage resources, other non-storage resources related to the dynamic content also can achieve similar access operations, so as to share the same API interface.

3. User contents

The k-th message of user j is defined as:

$$UserPost_{jk} = SIG_j([Username_j, k, type, MSG_k, REPLY_k])$$

MSG_k is content, k is a monotone increasing number, possible values of type include: the new posts, replies, retransmission (RT), direct messages (DM), $REPLY_k$ is an optional domain, which provides reference of the original message in response /retransmission and is defined as:

$$REPLY_k = [Username_{j'}, k']$$

Representing the original message is the k'-th message of the user j'.

The contents are simultaneously shared in two overlay networks:

1. Stored in DHT as a short-term storage value; and
2. Archived like files in BitTorrent network.

When the new content is created, the client-side must send a PUT request to the following addresses:

$$ID_{UserPost\_jk} = H([Username_j, \text{“post”} + k, \text{“single”}]) \text{ and}$$
$$ID_{swarm\_j} = H([Username_j, \text{“swarm”}, \text{“single”}]).$$

$ID_{UserPost\_jk}$ is the destination storage node’s address in the second DHT network,

providing the retrieval capability of any content.

ID_{swarm_j} is the gateway address of torrent swarm group related to the content of the user $Username_j$ in the third network, and this torrent contains all the content of a given user j, which provides quick distribution and sharing of content based BitTorrent protocol and is independent of the second DHT network. The neighbor node of ID_{swarm_j} needs to join the swarm cluster of user j to help the storage and distribution of content, provide data reliability and better data distribution performance; similarly, the neighbor node of $ID_{UserPost\_jk}$ also needs to store the same values stored by $ID_{UserPost\_jk}$.

The swarm group mechanism solves the problem of fast and efficient notifications and distribution of new content, so that the user's followers don't have to always poll the DHT network address to determine whether new content is generated.

3.1 Direct message (DM)

Users posting content can also be delivered by direct message, but only if the message receiver is a follower of user k.

$$UserPost(j \rightarrow l)_k = SIG_j\left([\text{“”}, k, \text{“dm”}, [PUBK_l(DM_k), H(DM_k)]]\right)$$

Except the for the content difference(now is [$PUBK_l(DM_k)$， $H(DM_k)$]) , there is no difference from regular posts. DM will only be received by user l who has successfully decrypted. Although other followers can also receive the message, they cannot decrypt the message, nor can they perceive who is the final receiver. Encryption is based on the ECIS elliptic curve encryption algorithm.

3.2 User content torrent/tracker rule

1. In hashing space, the online neighbor nodes within a certain distance from ID_{swarm_j} need to join the corresponding Swarm;

2. When ID_{swarm_j}'s neighbor receives new content from the DHT network, it must work as a gateway for BitTorrent network to incorporate content into a file-like archive structure;

3. BitTorrent tracker is read-only multi-value list storage. Its hash address is calculated as:

$$ID_{tracker\_j} = H\left([Username_j, \text{“tracker”}, \text{“multi”}]\right)$$

4. The follower of user j should join in the corresponding swarm to receive real-time content update, so as to obtain the address of the initial Peer through the primitive GET query of $ID_{tracker\_j}$;

5. $ID_{tracker\_j}$ is different from other storage values because it is read-only, which prevents the tracker attack and contains the swarm members' privacy. The list of IP addresses is obtained through the swarm protocol, which requires the online neighbor node of $ID_{tracker\_j}$ to join the swarm.

6. Swarm members can only know each other through IP address, and BitTorrent does not provide any information about the user name.

7. There is no need for the hash of all user contents, because the contents (including

DM) have been signed to verify the integrity of the content;

8. The added value k when generating new content is broadcasted directly through flooding in the Swarm;

9. Members of the Swarm will exchange the content lists, where members can choose to save or request only the most recent content;

10. The seed node is the node selected to archive the content;

11. A content publishers (user j) can choose not to be member of the swarm group (to protect privacy and hide IP address);

12. If the publisher chooses to be a swarm member, it does not have to follow the ID_{swarm_j} gateway mechanism, which will22 expose its own IP address;

13. Even if the publisher becomes a swarm member, it may not have to act as a seed node;

14. The new block generation rate will impact the user's posting speed, and if a new block is generated every 10 minutes, on average, 288 contents blocks can be released daily.

4. User mention mechanism

If the new content refers to user j, the client-side also has to send a notification to ID_j, including the entire message content, to be routed through the DHT network.

The mention mechanism is the only function in the system that needs to be addressed by user ID_j instead of ID_{nodej}, which may expose users' privacy information. An alternative implementation mechanism is as follows:

$$ID_{mention\_j} = H\left([Username|_j, \text{“mention”}]\right)$$

The user name is hidden hash and a new address for receiving and accumulating all mention is calculated, $ID_{mention\_j}$ neighbor nodes will also participate in the storage of mention, providing maximum reliability and storage performance. A bad thing about this approach is that the user needs to poll this address periodically to determine if any new mention is received.

The mention mechanism requires the collaboration of the client-side, and if it does not send notification messages to the network, the user will not perceive that he has been mentioned.

5. Explicit message request

User l can request a specific explicit message from user j without joining Swarm group through directly retrieving the corresponding contents from $ID_{UserPost\_jk}$ address of the second DHT network, it supports functions of "message upstream".

6. Message downstream

A downstream trace of the message (such as a reply/RT query for a specific content query) is relatively difficult to resolve, and a possible solution is to send a notification to a storage address of a multi-valued list.

$$ID_{replies\_jk} = H\left([Username_j, \text{“replies”} + k, \text{“multi”}]\right)$$

The stored value is the copy of all responses, which also requires the client sides to work together.

7. Hash tag

Like the mention mechanism, the hash tag detects in the context of the new message, and the copy of the message is sent to a specific multi-value list storage address:

$$ID_{hashtag_t} = H\left([hashtag_t, \text{“hashtag”}, \text{“multi”}]\right)$$

This is similar to a message downstream mechanism, but the difference is: the hash tag creates a new Swarm group; and $ID_{hashtagt}$ neighbors must also join this virtual Swarm. It is called virtual because the Swarm group does not share any file content and is only used to realize the broadcast function for users who want to monitor the hash tag.

8. Content search

A search for any content that appears can be realized by extending the implementation of the hash tag to build a similar mechanism for the content that appears. In order to reduce overhead and network transmission, corresponding restrictions must be attached, such as restricting content size and excluding prepositions. In addition, it can significantly reduce the storage overhead and the system implementation complexity if the unified storage of content containing the same content is stored in a temporary multi-valued list address. The address calculation is as follows:

$$ID_{word_w} = H\left([word_w, \text{“word”}, \text{“multi”}]\right)$$

TRON content provides the following security, extensibility and privacy features:

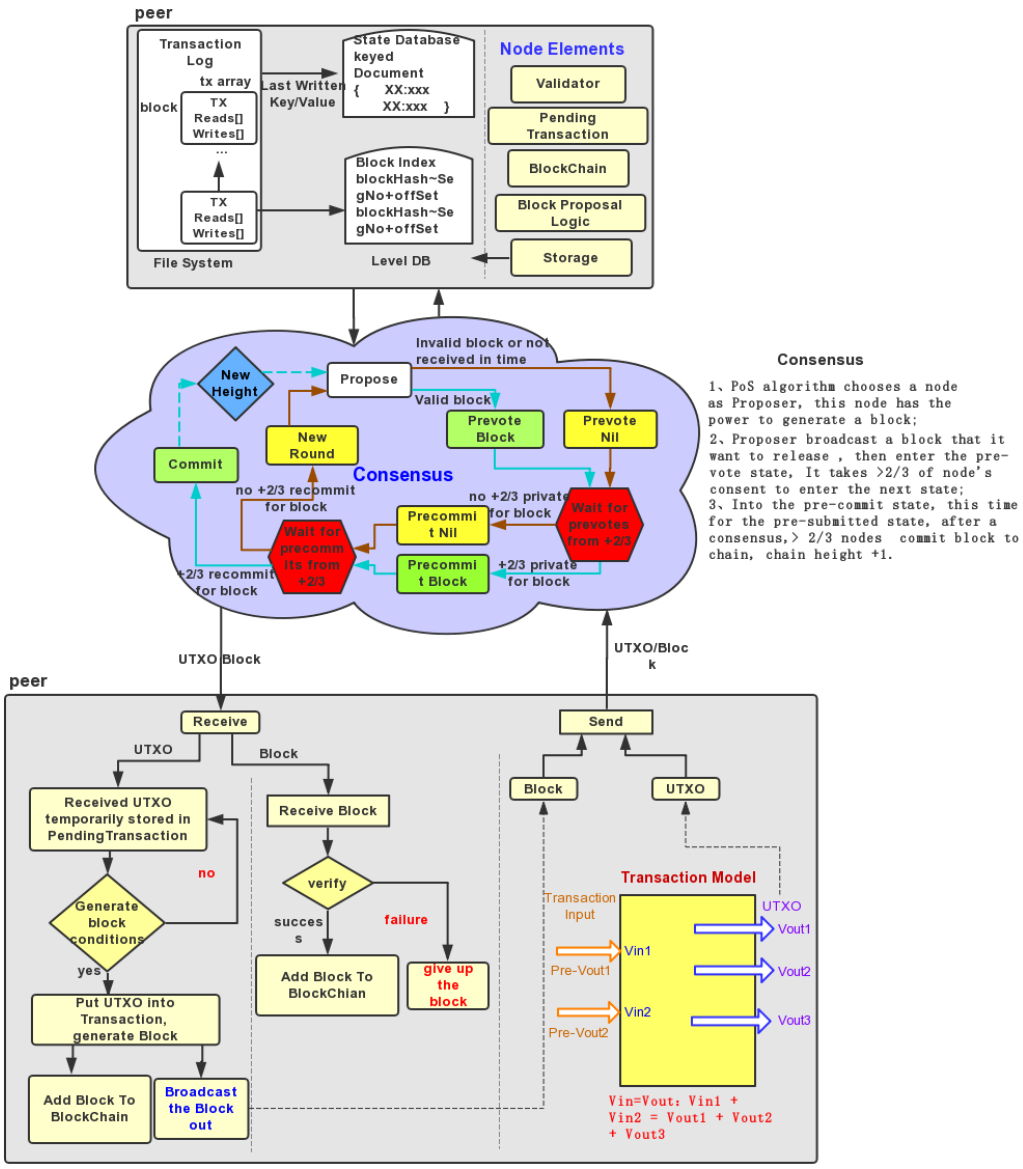
1. The architecture itself provides elastic extensions, and no single company, government or organization can close it;
2. The distributed user registration mechanism is as secure as Bitcoin transactions, providing non-centralized content authentication;
3. Users are more eager to register early so they can select their favorite user name.
4. The common user naming method and discarding the long encryption hash allow users to have a better use experience.
5. Public key substitution mechanism allows users to change their key pair when security is threatened;
6. The main functions of other blogging systems are included, such as user name search, message tracing, mention, encrypted message, hash tag and content search.
7. The ability to send notifications to and request resources from specific user via DHT routing, whether the user is online or not;

8. The architecture provides incentives for participating nodes to have the privilege of sending broadcast messages;

9. Users' public content and hashtags can be accessed through read-only web interface, which does not break the security of the system;

10. Resource-constrained client sides can be optimized, for example, by not storing all blockchains but only the hash values of the blocks. In order to search for a particular user, they can inquire network which block contains the user's registration, and the client-side only have to download the desired block without reducing security and verify data integrity through some branches of the Merkle Tree.

B. Blockchain platform



1. Introduction

TRON contains consensus engine, ABCI, UTXO, smart contracts and other modules. Consensus engine is the core, application connects with consensus engine by ABCI to form a Byzantine fault-tolerant state machine, which can be implemented in any programming language.

TRON blockchain platform has the following characteristics:

1. Scalability: TRON blockchaincan be extended through the side chain, which means that not only currency transactions, legally binding contracts and certificates, audio and video files can be stored in the blockchain database;

2. Decentralization: Without an agency, all nodes have the same rights and obligations, any node stopping working will not affect the overall operation of the system.

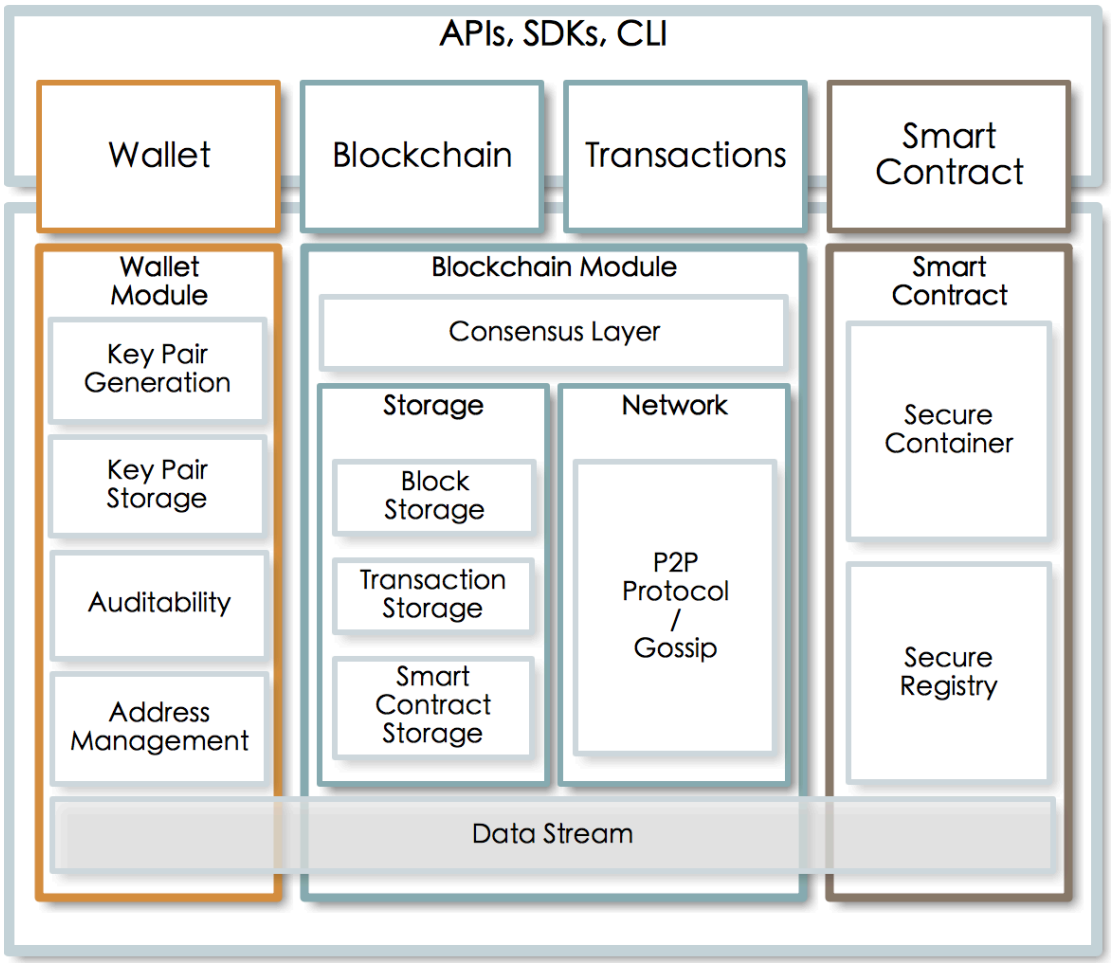
3. Trustless environment: All nodes in the system can be traded without trust. Because the operation of the database and the entire system is open and transparent, the nodes can not deceive each other;

4. Consistency: The data information between nodes is consistent;

5. Fault-tolerant: The system can accommodate 1/3 node Byzantine failure;

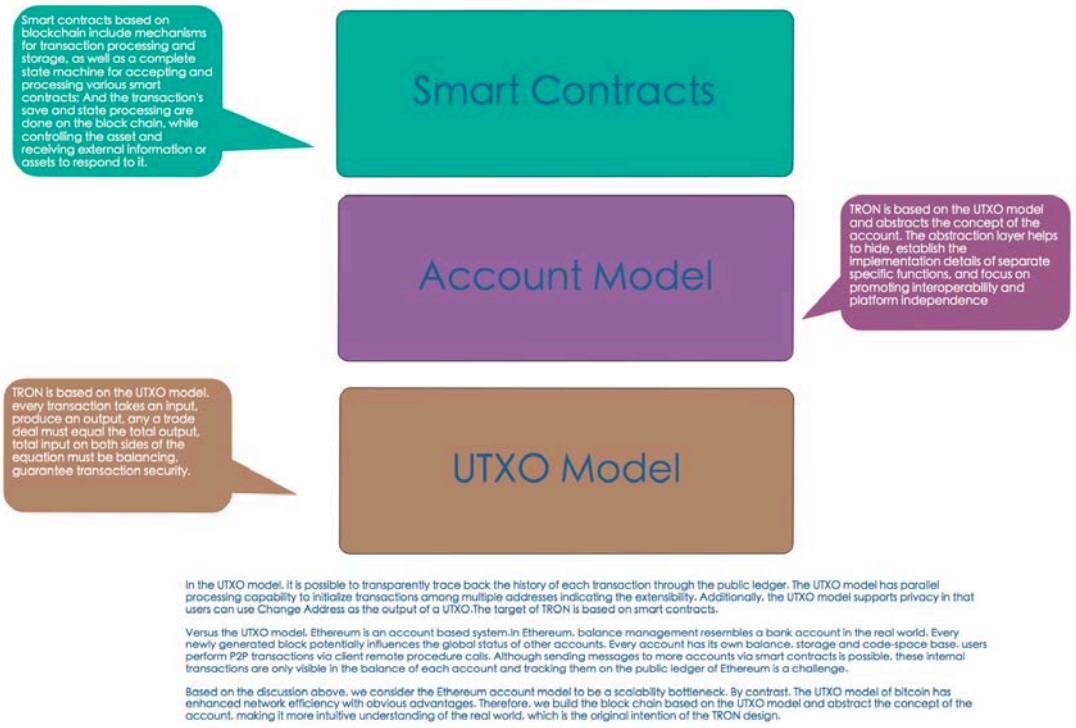
6. Scalability Account Model: UTXO Model + Account Abstraction. TRON has also made targeted improvements on the premise of UTXO's easy-to-parallel computing model. To make data easy to manage and easy to program, TRON introduces the world state-lightweight state tree concept, each of which maintains a global world state, the global state has the features of quickly find, can not be changed, easy to provide proof.

2. Software hierarchy



Software level is divided into two parts. The first part includesAPIs, SDKs and CLI, which are mainly used for calling an external provider for convenientdevelopment.The second part includesWallet Module, Blockchain Module and SmartContract Module, provides a storage interface, making the data of each modulepersistent.

3. UTXO



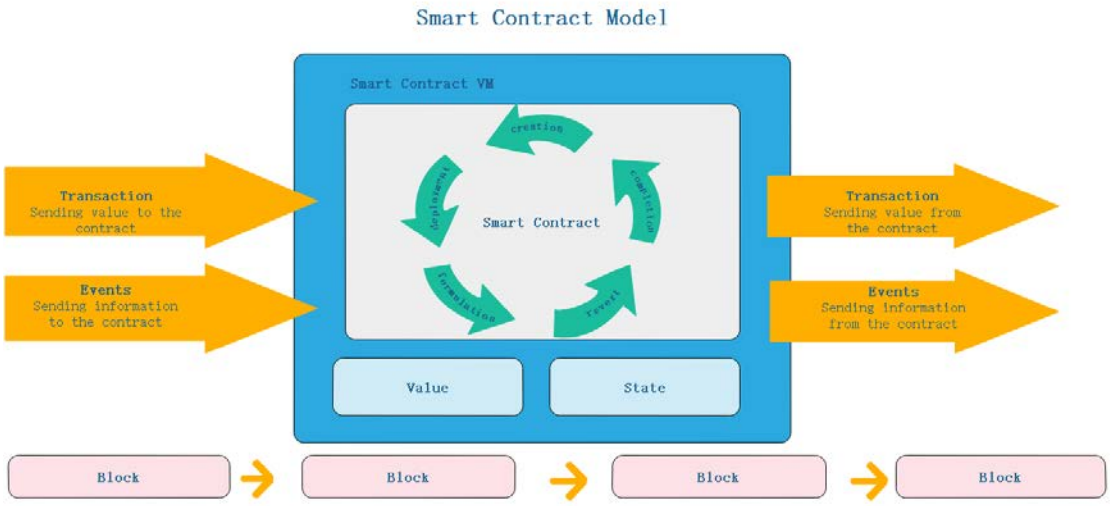
In the UTXO model, it is possible to transparently trace back the history of each transaction through the public ledger. The UTXO model has parallel processing capability to initialize transactions among multiple addresses indicating the extensibility.

Additionally, the UTXO model supports privacy protection in that users can use Change Address as the output of a UTXO. The target of TRON is based on smart contracts.

Compared with the UTXO model, Ethereum is an account based system. In Ethereum, balance management resembles a bank account in the real world. Every newly generated block potentially influences the global status of other accounts. Every account has its own balance, storage and code-space base. users perform P2P transactions via client remote procedure calls. Although sending messages to each account via smart contracts is possible, these internal transactions are only visible in the balance of each account and tracking them on the public ledger of Ethereum is a challenge.

Based on the discussion above, we consider the Ethereum account model to be a scalability bottleneck. By contrast. The UTXO model of bitcoin has enhanced network efficiency with obvious advantages. Therefore, we build the block-chain based on the UTXO model and abstract the concept of the account, making it more intuitive understanding of the real world, which is the original intention of TRON.

4. Smart contract



Certainty and Termination are two properties of a smart contract. When designing a smart contract system, non-deterministic factors need to be excluded.

Bitcoin has a set of scripting engines, the instruction set is very simple and non-Turing complete, with termination, so bitcoin smart contracts are certain. The Ethereum Virtual Machine (EVM) is a runtime environment for Ethereum smart contracts. The system functions for Ethereum smart contracts are not nondeterministic, but the contract's call path can be nondeterministic and result in a scalable performance Losses, it uses meter to achieve the termination. The Hyperledger Fabric smart contract uses Docker as the execution environment. Docker is a lightweight virtualization technology, under the blockchain Docker is a “heavier” execution environment, which is where the performance bottlenecks of Fabric, currently only up to hundreds of TPS per second, which uses a timer to achieve Termination.

In order to keepwith the advantages of certainty, termination, and lightweight of virtual machines and the language flexibility of container programming, TRON is poised to develop the TRON Virtual Machine as an execution environment for its smart contracts in the future. The TVMboots very fast, occupies less resources. TRON virtual machine data manipulation instructions are directly to the array and complex data structures to provide support. These will enhance the operational performance of TRON smart contracts. The TRON Network plans to charge for the operation and storage of tokens and smart contracts to achieve economic incentives to book-keeping persons and to prevent the abuse of resources.

In the future TRON smart contract developers can use almost any high-level language they are good at for TRON smart contract development. The first language support are java, Go etc. Tron plans to provide compilers and plug-ins for these languages to compile high-level languages into the instruction sets supported by TRON virtual machines.

The TRON smart contract model shown above is a piece of code (a smart contract) that runs on a smart contract virtual machine and is deployed on a shared, replicated

ledger (blockchain). TRON has a life cycle for smart contracts management, respectively are: the establishment, deployment, development, rollback, termination. It can maintain its own status, control its own asset value and receive external information, transactions or external information and transactions to respond.

5. Consensus

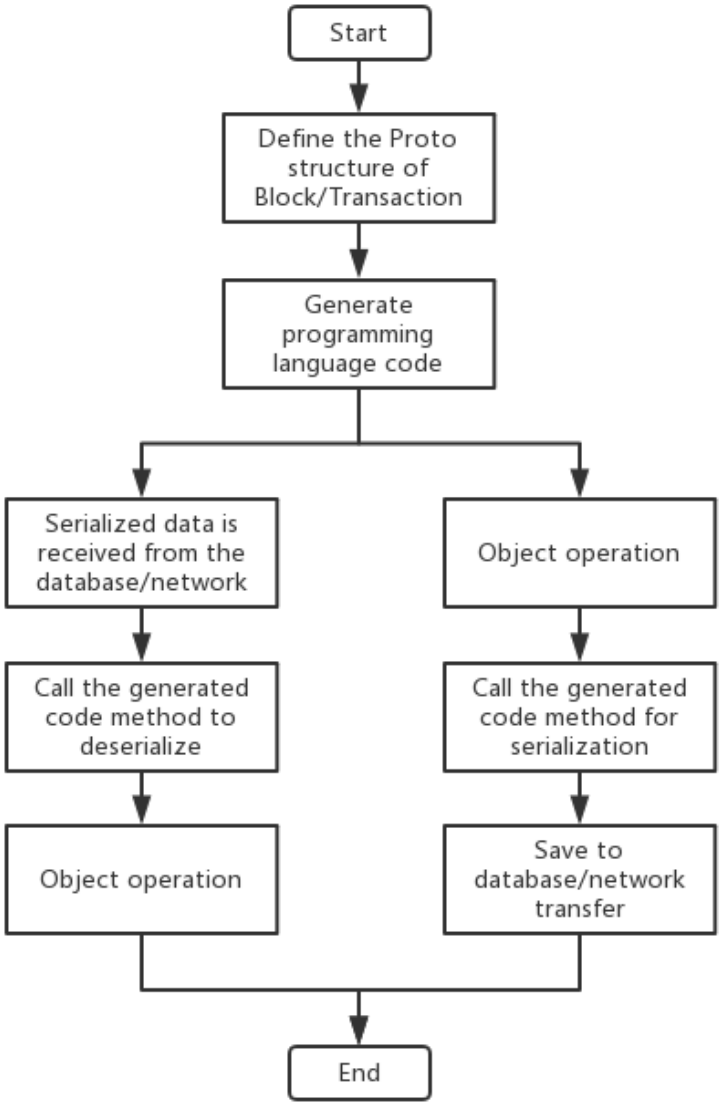
The consensus of TRON adopts a three-step strategy. The first step is to adopt a Kafka-based technology system to implement a centralized consensus algorithm. The purpose of TRON is to achieve system joint debugging and functional integration.

The second step is to use Raft-based distributed consensus mechanism to realize the centralized and distributed leapfrogging. This step gradually improves the functions of network and distribution and lays the foundation for the eventual realization of a wide distribution with no logical center.

The third step is to realize the Consensus mechanism of PoS and realize the Byzantine Fault Tolerant Consensus based on the "Margin Mechanism + Epoch Confirmation" and the compatibility consensus between PoS and PoW.

TRON is currently open source code to achieve a consensus algorithm for the first phase of the center. The second phase of the distributed consensus algorithm is under development and testing.

6. Protocol Buffer based object coding and serialization



6.1 Example

Proto code:

```
message Block {
        repeated Transaction transactions = 1;
        BlockHeaderblockHeader = 2;
}
```

Serialization:

```
Block.Builder block = Block.newBuilder()
        .setTransactions(transactions)
        .setBlockHeader(blochHeader)
        .build();
```

```
byte[] blockData = block.toByteArray();
byte[] keyData = block.getHash();
DB.saveBlock(keyData, blockData);
```

Deserialize:

```
byte[] keyData = block.getHash();
byte[] blockData = DB.getBlock(keyData);
Block block = Block.parseFrom(blockData).toBuilder().build();
```

C. Tron Content Storage

1. File Storage Protocol

TRON's bottom layer consists of a group of multi-layer protocol stacks with various implementation models for each layer and is integrated in modules. Corresponding interface standards are defined between layers, including the following five levels:

1. Name layer: a self-certified PKI namespace
2. MerkleDAG layer: data structure format
3. Exchange layer: block transmission and copy
4. Routing layer: locating peer nodes and objects
5. Network layer: establishing connection among peer nodes

1.1 Node and identification

"TRON node" is the program that can locate, release and copy MerleDAG. TRON network adopts node identification based on PKI (Public Key Infrastructure); the node is shown as NodeId, which is the Ciphertext Hash of Public Key. Node will store its public and private keys (private key will be protected by password). Users can freely establish and initiate a "new" node in every boot, which will deprive the network the profit achieved from former node running. The system has an incentive mechanism to make users maintain the same node.

```
type NodeId Multihash
type Multihash []byte
// self-describing cryptographic hash digest
type PublicKey []byte
type PrivateKey []byte
// self-describing keys
Type Node struct {
      NodeId NodeID
      PubKey PublicKey
      PriKey PrivateKey
}
```

The generation mechanism of NodeId is as follows:

```
difficulty = <integer parameter>
n = Node{}
```

```
do {
      n.PubKey, n.PrivKey = PKI.genKeyPair()
       n.NodeId = hash(n.PubKey)
      p = count_preceding_zero_bits(hash(n.NodeId))
 } while (p < difficulty)
```

When the connection is established, the nodes will exchange public keys among each other and, check if the hash of node public key is equal to the NodeId hash(other.PublicKey) equals other.NodeId of peer node, if not, the connection will be terminated.

1.2 Multihash and updatable hash

All hashes in TRON will be encoded with multihash, which is a self-describing hash format. The hash function should be used in accordance with specific security requirements. The encryption system is updatable, which means the system can switch to a stronger hash algorithm when a current hash function cannot meet more strict security requirements. But there is indeed a price to be paid, the object needs to be rehashed and the connection needs to be rebuilt. This way of not defining the length of a hash digest in advance allows for the tool used today can to work normally even if it is switched to a longer hash function tomorrow.

The hash digest value is stored in multihash format, including a short header, specified hash function and byte length of digest, For instance:

<function code><digest length><digest bytes>

The current TRON node must support the following hash algorithm: sha2-256, sha2-512 and sha3.

1.3 Network layer

Provide point-to-point reliable and unreliable transmission between two TRON nodes, and process:

1. NAT traversal—punching, port mapping and relaying;
2. Support various transport protocols—TCP, SCTP, UTP…
3. Support encryption and digital signature;
4. Multiplexing—multiplex connection, stream and protocol…

1.4 The routing layer: locating peer nodes and data

Routing layer serves for two purposes:

1. Node-routing—searching for other nodes;
2. Content routing—searching for data released to TRON.

Routing layer defines an interface, and all implementations meet or realize the interface can be linked to TRON, for instance: DHTs, mdns, snr, dns. The corresponding interface definition is as follows:

```
  FindPeer(node NodeId)
  // gets a particular peer's network address

  SetValue(key []bytes, value []bytes)
  // stores a small metadata value in DHT

  GetValue(key []bytes)
  // retrieves small metadata value from DHT

  ProvideValue(key Multihash)
  // announces this node can serve a large value

  FindValuePeers(key Multihash, min int)
  // gets a number of peers serving a large value
}
```

1.5 Block swap: transmitting content-addressable data

TRON's block swap layer is responsible for data transmission coordination. Once the nodes are aware of each other and establish connection, content-addressable blocks can be transmitted through swap agreement. Note that the term "swap" in this context does not refer to a swap in the sense of a finaical market transaction as defined in the US Commodity Exchange Act. It is a technical term of art dealing with the TRON platform code's distribution of content. The block swap layer is the interface definition, and all implementations meet or realize the interface can achieve seamless access, for instance:

1. Bitswap: current implementation, which is the generalization implementation of BitTorrent, supports the swap of any DAG;

2. HTTP: simple HTTP implementations can be used between HTTP clients and servers.

BitSwap is a block transmission agreement similar to BitTorrent—where nodes represent the expected block set with want_list, and represent the data block set they can provide with have_list. Unlike BitTorrent, block swapped by BitSwap is not limited to a single torrent. BitSwap serves as a persistent market, nodes exchange blocks through BitSwap market, the node can obtain their favored block sets, and these block sets may be completely unrelated files from the file system. Sometimes, in exchange, a node may have no blocks needed by other nodes, and it will help find blocks it needs – these get the needed block from each other, and such incentives can help the cache and distribution of rare blocks.

1.5.1 BitSwap credit

The agreement must urge the nodes to be the seed because they might not have the blocks needed by other nodes. Therefore, BitSwap nodes will actively deliver blocks to other peer nodes, and the agreement must prevent the existence of greedy nodes that load little and never share their blocks. A simple system similar to credits can solve these problems.

1. An node track its number of bytes exchanged with other nodes;

2. Nodes transmit blocks to indebted nodes in form of probability, the higher the

debt of indebted node is, the lower probability of block transmission will be.

It should be noticed that if the node decides not to transmit a block to the opposite end, then it cannot transmit a block to the ignored correspondent node within the following ignore_cooldown time. This prevents the submitter from escaping from repeatedly sending blocks.

1.5.2 BitSwap strategy

The node's strategy to send blocks directly influences the performance of block swapping. It should meet the following objectives:

1. Maximize the node and the overall transaction performance;
2. Prevent greedy nodes from taking advantage of or reducing the swapping performance;
3. Efficient and exclusive to other strategies;
4. Be friendly to the credit node.

A practical strategy selection is Sigmoid function, the defined debt ratio r is:

$$r = \frac{\texttt{bytes\_sent}}{\texttt{bytes\_recv} + 1}$$

At a given r, the probability of sending to indebted node will be calculated as:

$$P\Big(\ send \mid r\ \Big) = 1 - \frac{1}{1 + exp(6 - 3r)}$$

The sending probability drops dramatically with the rising of debt ratio. Debt ratio is the measurement of credit, which is friendly to previous nodes that have swapped many blocks and unfriendly to unknown or untrusted nodes.

1.5.3 BitSwap accounts

The BitSwap node will keep accounts of block swaps, which helps node track history and avoid being cheated. When connection is established, the BitSwap nodes will exchange account information. If the information is not matched exactly, the accounts will be deleted and reinitialized, and all profits and debts will be lost. This method seems to have a loophole for malicious nodes to delete debts by way of intentionally "losing" accounts, but that is impossible because nodes cannot accumulate enough debts. Furthermore, it will lose all the previously-accumulated debt, and other nodes will consider it abnormal behavior and refuse to swap.

```
type Ledger struct {
  owner      NodeId
  partner    NodeId
  bytes_sent int
  bytes_recv int
  timestamp  Timestamp
}
```

The account information history will not influence the normal operation, and only recent account items are useful. The node can also choose store or not to store historical information.

2. Self-operation of Storage Network

TRON is a centerless storage network, which turns the storage from cloud model to market model based on algorithms and rules. The market is based on blockchain and trade in virtual currency: the miner earns TRX by providing storage for clients; on the contrary, the clients spends TRX to hire miner to store and dispatch data. Similar to Bitcoin, offered by miners, which provides a useful client-side service (unlike Bitcoin, the miners' work is only useful for blockchain consensus) and is a strong incentive to drive miners to contribute as much storage space as possible to client-side rent. The agreement will integrate these resources into a self-healing storage network for external use, and the network will realize its robustness by copying and dispersing stored content and automatically detect and repair replication errors. The client side can choose different replication parameters to protect data according to different threat degrees and levels. The storage network also provides other security guarantees for clients such as end-to-end encryption of content, and the storage provider cannot obtain the decryption key.

2.1 Proof-of-Replication (PoRep) algorithm

The server (prover, P) convinces the user (verifier, V) that its data D is replicated and stored in multiple physical storage locations.

2.1.1 Seal operation

Seal operations include:

1. getting the public key of verifier's stored data through asking verifier to prove the pseudorandom and force the data copies to be correctly stored in independent physical storage;

2. force the time required by copying the process to be longer than the expected time required by responding to a Challenge.

2.1.2 PoRep algorithm flow

Create a copy: create a copy in Setup algorithm through the Seal operation and provide the proof of successful execution.

Setup

- INPUTS:
  - prover key pair $(\mathsf{pk}_{\mathcal{P}}, \mathsf{sk}_{\mathcal{P}})$
  - prover SEAL key $\mathsf{pk}_{\mathsf{SEAL}}$
  - data $\mathcal{D}$
- OUTPUTS: replica $\mathcal{R}$, Merkle root rt of $\mathcal{R}$, proof π_{SEAL}

Storage verification: The Prove algorithm produces storage verification for the

replica. The prover receives a random challenge c and determines one leaf Rc of Merkle Tree R (root is rt), the prover produces the proof of Rc and the Merkle path to rt.

Prove
- INPUTS:
  - prover *Proof-of-Storage* key $\mathsf{pk}_{\mathsf{POS}}$
  - replica $\mathcal{R}$
  - random challenge c
- OUTPUTS: a proof π_{POS}

Verification proof: Verify algorithm checks the validity of the storage verification based on the data copy of the Markel tree root and the hash of the original data. The verification is publicly verifiable: any distributed system node interested in this data can check the validity of the storage verification.

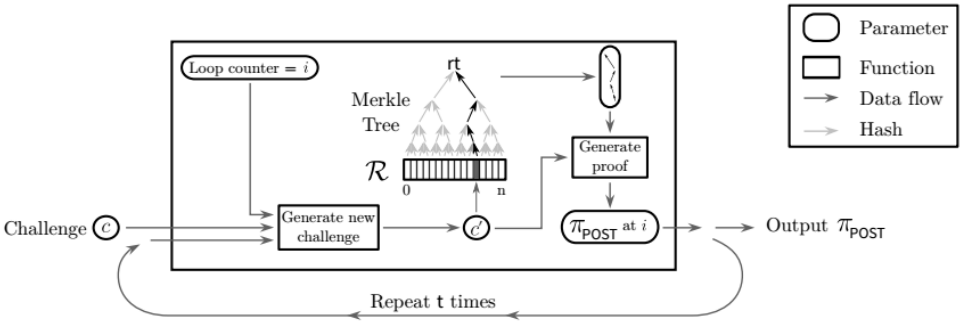
Verify
- INPUTS:
  - prover public key, $\mathsf{pk}_{\mathcal{P}}$
  - verifier SEAL and POS keys $\mathsf{vk}_{\mathsf{SEAL}}$, $\mathsf{vk}_{\mathsf{POS}}$
  - hash of data $\mathcal{D}$, $h_{\mathcal{D}}$
  - Merkle root of replica $\mathcal{R}$, rt
  - random challenge, c
  - tuple of proofs, $(\pi_{\mathsf{SEAL}}, \pi_{\mathsf{POS}})$
- OUTPUTS: bit b, equals 1 if proofs are valid

2.1.3 PoSt algorithm flow

The Setup of PoSt is similar to Verify algorithm and PoRep. The Prove algorithm produces Proof-of-Spacetime for data copy.

Prove
- INPUTS:
  - prover PoSt key $\mathsf{pk}_{\mathsf{POST}}$
  - replica $\mathcal{R}$
  - random challenge c
  - time parameter t
- OUTPUTS: a proof π_{POST}

The prover receives random challenge from verifier and orderly generates Proofs-of-Replication, then uses one proof output as the next output until t times of loop iteration, as is shown in the following picture:



PoSt. Prove mechanism displays an iterative proof that is effectively stored for a period of time.

II. Technical characteristics and contrast

A. Bitcoin vs Ethereum vs Tron Overall Technology Comparison

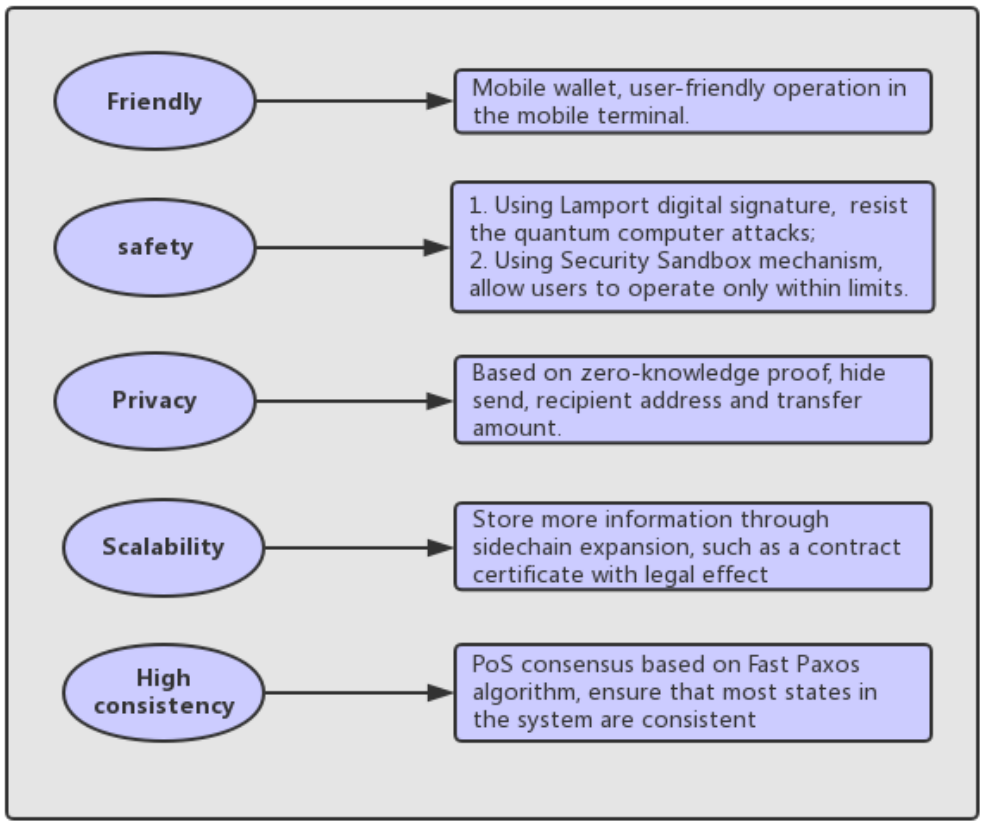
	Bitcoin	**Ethereum**	**Tron**
Consensus Algorithm	PoW	PoW	PoS similar to Tendermint
Transaction throughput	7 deal/s	25 deal/s	1500 deal/s
Block production time	10 min	15 s	15 s
Confirm the time	6 Blocks	12 Blocks	1 Blocks
Smart Contract	Simple scripting language	Based on the solidity language	A variety of programming languages
Wallet signature algorithm	ECDSA (Elliptic Curve Algorithm)	ECDSA (Elliptic Curve Algorithm)	Lamport Algorithm
Transaction Mode	Based on the UTXO transaction model	Based on Account model	Based on UTXO transaction model and local cache account information
Wallet trading platform	PC	PC	Mobile

B. Bitcoin vs Ethereum vs. Tron Security Technology Comparison

Bitcoin	**Ethereum**	**Tron**
No smart contracts	The EVM virtual machine has no privilege operation check mechanism	Provide a security sandbox to check the privilege operation according to the user authorization policy
ECDSA elliptic curve	The ECDSA (Elliptic	Lamport digital signature

digital signature algorithm, the current "Milkyway-2" computational power, resulting in a hash SHA256 bitcoin hash hash algorithm about 248 years, but with the development of new computing technologies such as quantum computers, the future Asymmetric encryption algorithm has a certain probability of cracking	Curve Algorithm) has the same problem with Bitcoin	algorithm, resistant to quantum computer attacks
PoW did not reach a global consensus on the block, but with the follow-up block to join the link, reducing the probability of chain bifurcation	Currently there is a bitcoin-like bifurcation problem with Ethash-based workload proofing algorithms	Consensus algorithm based on Fast Paxos variant, it can decide Peer voting weight according to the State, as long as 2/3 nodes confirm the block can reach a global consensus, there is no bifurcation problem. And it can be adjusted to meet other needs, it can also use a variety of consensus mechanisms! Such as the combination of Pos and PoW
Merkle tree for data integrity verification	Merkle Patricia Tree tree for data integrity verification	Merkle Patricia Tree tree for data integrity verification
Broadcasting Mechanism Based on Gossip Protocol	Adopt Kademlia based P2P network, but data is non-storage encrypted, the data source can be traced	Customized P2P network with data storage encryption, location transparency, source non-traceability

III. Technical solution



IV. The market size of entertainment and why TRON is necessary?

At present, the annual scale of global entertainment market has reached thousands of billions[1]. Internet Revolution makes it possible for entertainment experience to be globalized for the first time so as to generate numbers of entertainment giants at a level of trillions on the world[2].

However, factors such as fragmentation of payment settlement and an insular feature of the entertainment system, etc. have led to failures in completely releasing the potential of the global entertainment industry.

1. The market size and growth of entertainment industry

Since the 21st century, entertainment industry all over the world keeps growing at a rate of 5.6% on annual average. Among them, output value of the American entertainment industry tops the list. In some developing countries such as India, China and Brazil, etc.,

[1] https://www.statista.com/statistics/237749/value-of-the-global-entertainment-and-media-market/

[2] International Trade Administration. (2016). 2016 Top Markets Report (Media and Entertainment).

entertainment industry is making progress strongly and annual average increases in these countries all exceed 10%[3]. The reason why the entertainment industry can leap forward at full speed today is that a complete set of comprehensive recreation system centering around life style, gambling and game playing has been established for it as its foundation. The above aspects have been always the "cash cow" of the entertainment industry and their integration with network releases the giant potential of entertainment to a greater extent.

With the renovation and development of global internet technology and the awakening of consumers' self-awareness, industries falling into the category of life style are rising quickly. The 2016 Global Social Media Study Summary indicates that among nearly 7.4 billion of the global population, the number of active users of social media has reached 2.307 billion; overlord status f traditional social applications such as Facebook, etc. has been shaken to a certain degree; and, young users tend to have more diversified application selections[4]. Throughout the world, values for applications of life style, including Tinder and Instagram, etc., have been estimated to break through ten billions of dollars[5] and are forging recreational life of human beings in an unprecedented manner. In this process, broad business development opportunities are raised. Match Group, the largest online dating group of American, went public on November 19, 2015. On that day, it raised $ 0.536 billion and its aggregate market value exceeded $ 3.4 billion. Moreover, multiple well-known social apps are owned by it, such as Tinder and OkCupid, etc.. [6]Likewise, the Asia-Pacific market is booming and the development of life style apps is especially spectacular in the Asian region, particularly China. Currently, utilization frequencies of social networking software such as WeChat and QQ, etc. have ranked Top 5 among social software on the world. Emerging diversified social apps spring up one after another in a rising tendency. Taking social broadcaster software industry in China for example, Momo as a benchmarking listed company in Chinese broadcasting industry has been established for five years and its market value has arrived at $ 8 billion. Up to now, the average rate of increase in each quarter remains at 50% and above[7]. Other companies closely following it are YY (market value: $ 4 billion) and Inke (market value: 7 billion RMB), etc. successively. All of them are growing rapidly. In China, it can be said that "A Battle among Thousands of Broadcasting Companies" has been put on the stage, which attracts ten billions of venture capital funds[8]. Clearly, not only are lifestyle apps becoming an indispensable part of users' life around the world, but considerable gains can be generated commercially as they are featured with high gross margins, large flow and impressive interactions and penetration.

---

[3]Paul Bond. (2013). "Study: Global Entertainment Industry Poised to Top $2 Trillion in 2016." ***The Hollywood Reporter***, June 05.

[4] Dave Chaffey. (2016). Global social media research summary 2017. ***Smart Insights***.

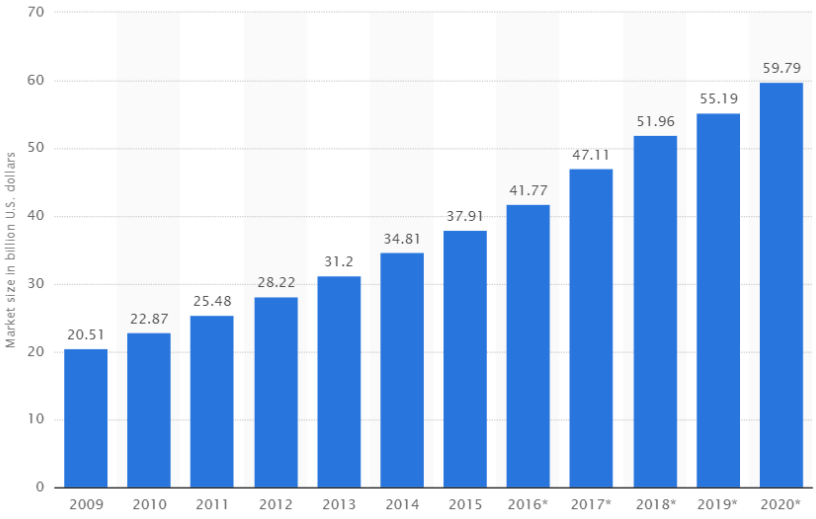
[5] Steve Schaefer. (2014). "Instagram Worth $35 Billion, Facebook Stock $91, Citi Says." ***Forbes***, Dec 19.

[6] Caitlin Stewart. (2016). The Dating Services Industry in 2016 and Beyond. ***Market Research.com***, May 23.

[7] https://www.immomo.com/newsroom

[8] Tang Xujun.(2017). Report on China's New Media Development: Bluepaper of New Media. Social Sciences Academic Press.

A the end of 1980s, gambling industry rapidly rose all over the world. To 2002, its global output value reached more than $ 900 billion and it turned into the world fourth largest industry[9]. According to the *Annual Report on World Gambling Industries*, earnings made by the global gambling industries in 2014 surpassed $ 4500.9 billion. As predicted, their revenues in 2019 can be expected to reach $ 525 billion.[10] Now, development tendency of the online gambling game industry is very prospective and the scale of such a global market almost arrives at $ 5 billion[11]. Annual income of modern gambling pioneers represented by William Hill and Ladbrokes breaks through tens of billions[12]. Additionally, Crown and MacauSlot of Asia begin to exhibit a growth trend as well. Specific to countries with great powers of traditional gambling industries, concrete industry forms there are quietly transformed into online gambling that will rise. According to statistical data published by United Kingdom Gambling Commission (UKGC) in 2016, revenue growth of offline gambling industry in UK flagged and even negative growth occurred; by contrast, year-on-year growth of 4% was realized in the online gambling industry when compared with that of the last year. As a result, the online gambling industry became the largest category of UK gambling industry.[13]



A Schematic Diagram for Market Size of Global Online Gambling Industry

Source of the Picture: statista.com

The traditional video game industry originated in the end of 1970s. From 1980s to

---

9 http://www.chinavalue.net/Media/Article.aspx?ArticleID=112204

10 Global Betting & Gaming Consultants. (2015). GBGC's Global Gambling Report 2015.

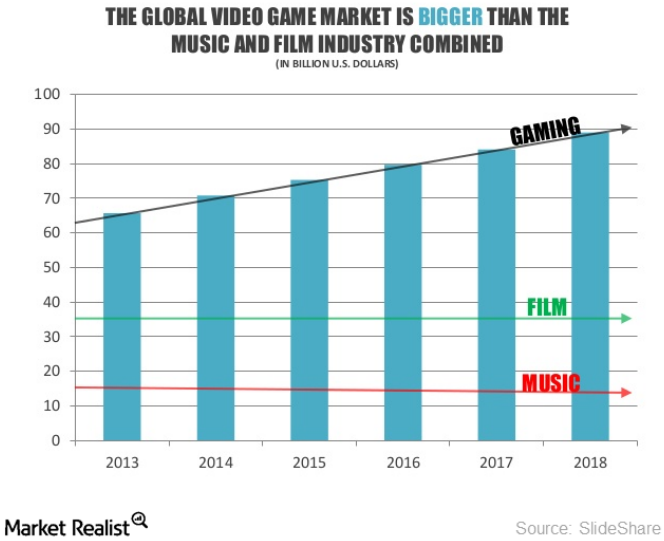
11 https://www.statista.com/app.php/statistics/270728/market-volume-of-online-gaming-worldwide/

12 http://shares.telegraph.co.uk/fundamentals/?epic=WMH;

https://markets.ft.com/data/equities/tearsheet/summary?s=LCL:LSE&mhq5j=e1

13 http://www.gamblingcommission.gov.uk/news-action-and-statistics/news/2016/New-figures-show-online-gambling-is-largest-gambling-sector-in-Britain.aspx

1990s, with the emerging of EA, Nintendo and Sony, etc., video game turned into an important basis of European, American and Japanese entertainment industries. Since the 21st century, it substantially suffers impacts of networking and a variety of new game forms take form.[14] Online games can be referred to as the earliest "cash cow" businesses in the circle of the Internet. As early as the end of the last century, some famous online game manufacturers such as Blizzard Entertainment, etc. were born in America and their values at present have broken through tens of billions[15]. In recent years, "phenomenal" mobile games have emerged in endlessly. For example, Clash of Clans launched by Supercell that is a game company in Finland and Candy Crush published by King Digital, a Swedish online game firm, and so on all have attracted extensive attention and many players, which sufficiently reflects prosperity and development of online game industry.[16] Currently, mobile online games occupy the biggest market share all over the world. In 2016, market size of global video game industry reached $ 99.6 billion; among which, that of mobile games takes $ 46.1 billion equal to a proportion of 42%. While global mobile games thrive and develop today, Asian-Pacific game market is expanding in a dramatically fast speed. According to the prediction, revenues of the game industry in Asian-Pacific Region will take a percent of 47% in the global total revenues in 2017, among which, proportion occupied by China Market will be up to 26% surpassing America to rank first throughout the world.



Simultaneously, revenue growth of mobile games in China market rises rapidly. As predicted, market size of the mobile game industry in China will arrive at $ 10 billion in 2017, which increases by 41% if compared to that in 2015 ($ 7.1 billion). Clearly, China's mobile game market booms and it is also necessary to occupy a share in such a giant emerging market.[17]

---

14 Adam Rogers. (2016). "An Investor's Guide to the US Video Gaming Industry." Market Realist, Jun 06.

15 Google Finance: Blizzard Entertainment, Inc.

16 Reportlinker. (2015). Global Online Gaming Market 2015. PR NewsWire, Sep 15.

17 Newzoo. (2016). "The Global Games Market reaches $99.6 Billion in 2016, Mobile TRONerating37%". Global Games Market

Source of the Picture: Market Realist

To sum up, integration between the entertainment industry and the Internet at the global angle of view is moving towards a new round of prosperity. At the time of locking entertainment industry development trends in developed regions such as European countries, importance should be also attached to the Asian-Pacific region as a giant emerging market rapidly growing and the key to further development of the entertainment industry. However, for the purpose of effectively grasping core markets and users of the entertainment industry, simply transferring traditional entertainment forms improved to the Internet together is far from enough. In order to stimulate the endogenous vigor, in-depth integration of information should be carried out for the entertainment industry renovated from perspectives of ideas and propositions.

2. Why TRON is necessary?

1. Information and credit sharing among entertainment systems. As for an entertainment system constructed based on the Internet, userinfo and user credit are cores of it. Due to short realization and existence cycles of the current entertainment applications and systems, a single developer cannot preserve the credit system of a user in a specific entertainment system. Consequently, each entertainment system nearly exists like an information isolated island. TRON network provides an effective means for userinfo existence and information docking between different systems in the entire entertainment market, so as to break information barriers among different applications. In this way, users are able to substantially lower information input costs required by a particular application and developers realize efficient interactions in the field of user identification to reduce expenses of repeat authentication and prevent user identity information from being disclosed or stolen by an intermediary agency.[18]

2. Token circulation among entertainment systems. Regarding traditional online entertainment industries, gross margins are high and volume of withdrawal and investment transactions is large. Nevertheless, token barriers inherent in diverse entertainment systems may cause rather high extra transaction costs to users that employ diverse sub-systems/applications of a certain system; in other words, it is extremely difficult for users to stride across different systems and applications. Barrier-free tokens based on TRON network will fill up the blank of channels that can be connected to various system settlement and payments in the entertainment network, and make absolutely trustworthy transaction information flow come true. Therefore, the trust relationship between the people in many social links and game

Report.

[18] Swan M. (2015). Blockchain: Blueprint for a new economy. " O'Reilly Media, Inc.".

sessions, etc. turns into the trust between human beings and technology.

3. In-depth amalgamation of entertainment systems. TRON network constructs a complete set of consensus mechanism to solve identification and individual credit reporting issues related to network transactions by an extremely low cost. In addition, it takes advantage of a point-to-point transactions to avoid a conventional centralized settlement structure. At the same time, TRON network plays a role of trustee among diverse users or developers to ensure authenticity and conformance of credit assure. According to the above two aspects, running efficiency of the entire globalized entertainment economy system can be enormously improved. Furthermore, the decentralized operation mode of economy itself is a typical feature of the market economy system.[19] Decrease in transaction cost will promote a substantial rise of consumer demands, which incurs the flourish of a global entertainment economy system.

V. TRON business moduel

1．lifestyle

Two cores of social contacts with strangers and entertainment are credit and user level. By sharing decentralized TRX information, social contact screening cost can be reduced unprecedentedly. By joining TRON protocol, developers and users are able to obtain TRX tokens given by the TRON foundation as a gift.

Below, examples of some application modes combined with modern entertainment systems and the corresponding application scenarios in concrete are presented.

Tinder Schema

Every newly registered user is allowed to obtain TRX. Users praise each other one-to-one by sliding to gain tokens. Each time a user is praised, he/she will earn TRX given by the system. The user possessing more TRX tokens has better credit and higher level of his/her entertainment behaviors. Thus, such a user can be favored by more other high-quality users. In addition, the system also pushes users with higher scores. Token growth limit is defined on a daily basis.

By virtue of settings, function of mutual gift presenting by sliding can be realized by TRX between users. The behavior of presenting TRX as a gift will make the corresponding user benefit from selecting more qualified and matched persons.

Online Live Streaming Schema

Every newly registered user is entitled to obtain TRX. As for anchors, they should acquire tokens by being given a reward. Each time they are presented to currencies of

[19] Davidson S, De Filippi P, Potts J. (2016). Economics of blockchain.

a certain magnitude, TRX can be obtained. As for users, they can get TRX when they present currencies of the corresponding magnitude. Anchor of a higher TRX score has a stronger performing capacity and is more popular among users. By contrast, user of a higher TRX score has a more intensive power of consumption and should be a high-paying user endowed with a glory level by the platform. TRX can be also presented as a gift among users by means of giving a reward and no upper limit has been imposed on the amount of TRX obtained in each day.

Transaction among users

Users in the user transaction schema can acquire interactive services through a social network site and get TRX provided that the transaction succeeds and both parties of the transaction are satisfied. In the case of any party is discontent or triggers reporting, TRX destruction may be incurred. Additionally, the user can also acquire TRX when the volume of a transaction reaches a certain magnitude. Likewise, no upper limit has been exerted on the amount of TRX obtained in each day.

Examples of the Application Scenario

A female user named Alice registers for an application of social contacts with strangers in TRON and it then praised by 560 men who slides right; and, such a kind of behavior is read by TRON network that thus subsequently gives Alice 56 TRX tokens as a reward.

A user Bob who notices that Alice has redeemed a point of 56 TRX send a contact request to her.

Another user Candice comes to an agreement with Bob to provide chatting services. However, Candice breaks their agreement after receive the earnest money of 200 USD in virtual currency paid by Bob. Consequently, Bob initiates a smart contract of TRX destruction against Candice. That is, Bob needs to spends 10 TRX on destroying 100 TRX owned by Candice, at the time of which, the earnest money in the transaction deposit account is destroyed as well.

Another user Douglas who has been awarded with 2,500 USD as a gift in the live streaming software receives 250 TRX presented by the system and his level is promoted at the same time. Therefore, he is entitled to multiple anchor permissions such as highlighting and broadcast starting push.

Another user Evan who has registered for social networking software in TRON injects 1,000 TRX simultaneously to achieve a higher level instantly and enjoy all social networking privileges corresponding to 1,000 TRX. Moreover, system gives priority to surrounding high-quality user push for Evan.

2. Betting

TRON makes it possible to construct decentralized online gaming platforms that are equipped with a top-up system applicable to all virtual currencies and legal tender.

Based on TRON, developers are able to freely construct online gaming platforms so as to further provide opportunities to ordinary investors to participate into the investment platform.

A gaming platform takes the responsibility to issue TRX and keep behavior records. The fundamental issuance rule is as follows.

In each case that virtual currencies of the corresponding magnitude have been spent, TRX can be issued; or, when virtual currencies of a certain value have been earned, TRX can be also issued without any upper limit.

Game developers can develop TRX gambling system to perform gaming gambling by adopting TRX as the subject.

TRX injection is accepted. Users can resell TRX obtained by themselves, which leads to address account destruction. Or, the user utilizes its own TRX to perform gaming directly as TRON network accepts varieties of virtual currencies such as BTC, Ethereum and EOS, and employs TRX as neutral currency to conduct settlement.

Examples of Application Scenarios

A user Alice who registers for a gaming application in TRON protocol wins some virtual currencies of 1,000 USD by gaming with other users and is thus awarded with 10TRX.

Another user Bob invests ETH into the gaming application in TRON protocol; thus, he gets a return of 10 ETH and is awarded with TRX.

Another user Candice lost some virtual currencies of 1,000 USD because of gaming and is thus awarded with TRX as well.

3. Game

TRON makes it possible to construct decentralized online gaming platforms. Developers are able to freely construct game platforms based on TRON so as to realize game development crowdfunding and further provide opportunities to ordinary investors to put investment into the game.

Benchmarking issuance rules of the game are as follows.

When the user spends a certain amount on purchasing within the range of the platform, he/she obtains TRX.

TRX injection is accepted. Users can resell TRX obtained by themselves, which leads to address account destruction. Or, users can directly top up TRX to obtain game props in a way of consuming TRX; besides, tokens and TRX establishing market can be issued for game applications to realize currency interconnection and interworking among different game markets.

Examples of Application Scenarios

A user Alice who registers for a game application in TRON protocol consumes some virtual currencies worth 1,000 USD and is thus rewarded with 10 TRX.

Another user Bob quit an internal battle of the game should accept credit punishment and lose TRX correspondingly.

Another user Candice successfully receives another level and reward in a new game application as he/she injects TRX obtained from gaming.

4. Entertainment File Sharing

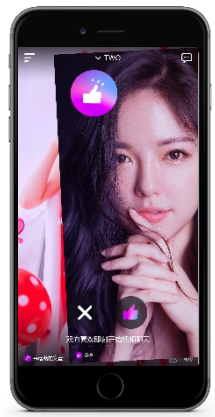
Based on rules provided by TRON network, developers can offer ease-to-use multi-platform file synchronization applications. The greatest strength of such applications lies in high speed transmission. In other words, the smart P2P technology is adopted to segment files so as to further accelerate the synchronization; at the same time, AES encryption is also conducted for such files to ensure both privacy and security. By virtue of public key generation and key reading techniques, files are transmitted and read; in this way, TRX can be awarded correspondingly after other users receive and download them. Additionally, key reading itself can be linked to virtual currencies. In the case that the other party pays the corresponding TRX or other virtual currencies, contents of the encrypted file are obtained; that is, TRON makes it possible to implement paid data transmission.

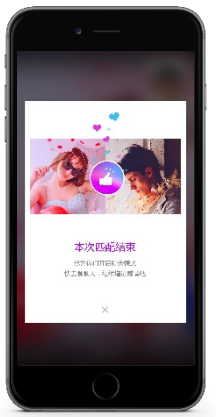
5. Mobile Application of TRX

TRON will provide official mobile end devices for TRX, including IOS and Android platforms.

TWO

TWO is a new social networking application based on TRON protocol. By sliding left or right, users of the opposite sex are screened. If both parities like each other simultaneously, they will be matched for real-time video chatting. As for users, they





can register for TWO at the mobile terminal to directly acquire TRX rewards and access TRX network to earn TRX by conducting social behaviors; but, they can be also punished by means of deducting their TRX due to destructive social behaviors.

Peiwo APP

Peiwo APP is the largest mobile audio live streaming application and owns more than 10 of millions of registered users. The number of its monthly active users has also broken through one million and the cumulative call duration goes beyond 2 billion minutes. In 2018, “Peiwo” will take the lead in being compatible with TRON network. It signifies that TRON protocol will transcend BTC and Ethereum to become the first smart contract blockchain protocol whose wallet and user quantities exceed ten million as far as all virtual currency protocols are concerned.

VI. Operating of TRON

1. Operator

As followers of Tim Berners-Lee, the TRON team firmly believes that the Internet belongs to all humankind since the day the protocol was born, instead of a profitable tool for a fraction of people. Therefore, TRON established Tron Foundation in Singapore, with the primary task to operate the TRON network publicly, fairly, transparently, and not for profit, and offer support to TRON’s development team.

Tron Foundation’s establishment was approved by Singapore’s Accounting and Corporate Regulatory Authority (ACRA) and is supervised by Singapore’s

corporate law. This Foundation is independently managed and run by a fiduciary board or management committee and is independent of the government.

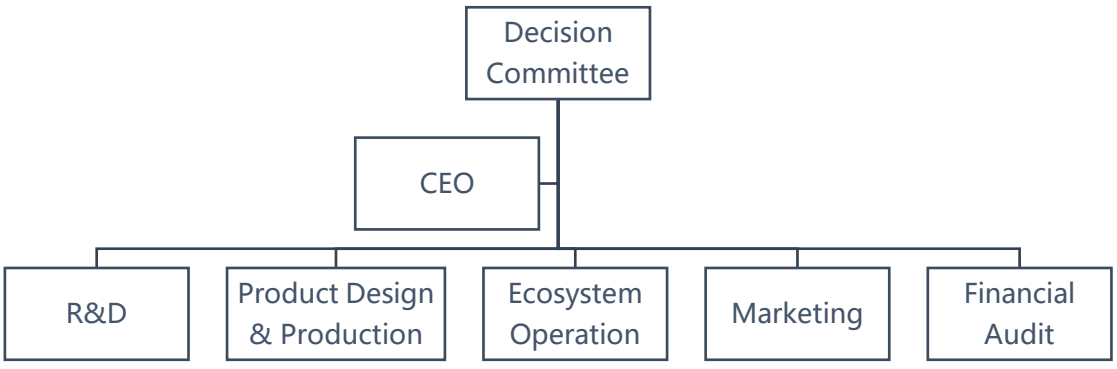
The Foundation does not have any commercial interests to support or participate in public interest or private interest activities. The “profit” earned by the Foundation is deemed surplus and will be kept as outlays for other activities instead of being distributed among its members.

Please find the BizFile of Tron Foundation as below:

https://dn-peiwo-web.qbox.me/tron/Bizfile-Tron-Foundation-2017.07.28.pdf

2. Governance structure and voting

The Foundation has set up a three-tier organization structure to ensure the reasonable use of funds and resources to promote openness, justice, and transparency; to constantly advance the rapid growth of TRON protocol; to extend the application scenarios of TRON protocol; and to attract more institutions, companies, and organizations to enter the open-source TRON ecosystem:



- Decision Committee

Decision Committee is the supreme decision-making body of the TRON Foundation and assumes the final decision. There is no seniority among the Committee members. The Committee is responsible for reviewing and approving the Foundation's major affairs, such as strategic planning, annual plan and budgeting, and vote on major issues in the TRON protocol ecosystem on behalf of the Foundation.

- CEO

The CEO is elected by the Decision Committee and is responsible for the Commission. The CEO will comprehensively organize and implement the decisions and regulations of the Decision Commission and is responsible for TRON's daily operation, reaching all targets assigned by the Commission, and reporting their implementation to the Commission/Committee on a regular basis. Moreover, the CEO has the right to establish functional departments when necessary and organize and employ managers. The CEO is responsible for the business of five departments, including R&D, product design and production, ecosystem operation, marketing and financial audit, forming a CEO-centered organizational, and management system.

- R&D Department

The R&D Department is responsible for the development and audit of underlying technology and is the Foundation's basic department. To ensure that team members exchange information and act in concert, R&D should exchange information with other departments (particularly the product design and production department) and timely adjust and communicate project details and decide the direction of future research.

- Department of Product Design and Production

The Department of Product Design and Production is responsible for enriching and perfecting product frameworks, provided by the technical department, making specific sustainable development strategies, including conducting market research

and planning product functions, and TRON's UI design and graphic design, and other works. The Department staff needs to follow trends, hotspots, community feedback, and proactively communicate with token holders and hold activities like occasional technical seminars.

- Ecosystem Operation Department

Based on what technical and product departments provide, the Ecosystem Operation Department is responsible for "the external and the internal." The Department will extend the depth of work, actively develop partners, and closely link TRON with end-users and partners, thus building an open, distributive, and privacy-protection global entertainment ecosystem. The Department will also build an ecosystem within the user community with benign interaction, free flow of information, and information symmetry.

- Marketing Department

The Marketing Department is responsible for marketing TRON's core and derivative products and services. Its duties include, but are not limit to, contacting and cooperating with media, advertising, designing user interaction, and other tasks. The Department works closely with Ecosystem Operation Department to formulate a publicity programe, based on the requirements of partners and end-users.

- Finance Department

The Finance Department is responsible for the company's financial affairs, including funds management, financial accounting, and cost control. Because digital assets feature high risks, this Department is also in charge of risk management and control and will coordiante with other Departments to analyze and evaluate projects' operational and financial risks. Because of the particularity of digital assets and tokens, it is difficult for existing institutions to supervise them in an effective way; therefore, the Decision Committee will engage professional auditors to ensure open and transparent use of TRX.