The Ω
White Paper

- A solution to smart contract security problems existing in Ethereum and alike
- Highly efficient consensus algorithm with no theoretical limit in scalability
- Universal and divisible token technologies to support the widest range of block chain applications
- The one and only one true land block chain

haohxu@yahoo.com

Howard Xu
V1.0 2021/6/15

Table of Contents

I. Executive Summary

In this white paper we present a next generation public block chain network. At the heart of the network is a suite of new technologies aimed solving the most pressing issues in today's block chain world, which are:

A. Smart contract security problem, which has resulted in loss of crypto assets worth over $10 billion in 2020.

B. A narrow range of applications. Contrary to its promise, block chain technology seldom finds its applications in areas outside financial realm, primarily due to its inability to handle complex forms of values.

C. Low efficiency. The fastest public block chains today can only handle hundreds of transactions per second.

To solve these problems, we have developed a new public block chain, named Ω Chain, featuring:

- A new smart contract scheme to provide smart contract users with very high level security, eliminating the possibility of coin theft in smart contracts.
- Universal and divisible token technologies that ready block chain technology for applications far beyond what block chains can support today.
- Dual chain architecture that removes any theoretical limitation on scalability. Ω Chain's transaction processing capacity scales linearly with computing power.

The Ω Chain main network went online in July, 2020 and has steadily produced about 5 million blocks so far.

II. Problems in Today's Block Chain Technology

A. Smart Contract Security

Between 2013 and 2014, Vitalik Buterin published Ethereum white paper which

suggests that block chains may be used to store programs and network nodes will execute these programs when called, and in so doing, the programs can server as agents of their creators and automate transactions without human involvements. These programs are called smart contracts. Buterin believed that this scheme is enough for secured programmed transactions. This is a multi-billion dollar mistake. In 2020, over 15 billion dollar worth of crypto asserts were lost due to security problems in block chain smart contracts.

True, contracts could be executed by programs. But not every program execution is a valid contract. For a block chain to server as a ledger for programmed transactions, transactions must be validated. As Nakamoto put it: don't trust, verify. Ethereum does not verify smart contract transactions; instead it verifies executions of smart contracts. In Ethereum, when a user calls a smart contract, nodes will execute the smart contract and as long as the smart contract returns with success, nodes will record it as a valid transaction. So what's wrong with this scheme? After all, it is the user who has initiated the smart contract call, shouldn't he be required to accept the result? That's Ethereum thinking.

Legally, contract (transaction) is meet of minds. Parties must agree on the same thing (gives and takes) for a contract to be valid. So what does a user agree to when he calls a smart contract? Does he agree to accept whatever the smart contract gives, or what was advertised by the smart contract's creator? Since most users are not programmers and don't understand what the smart contract programs do, they apparently agree to what was advertised. Ethereum nodes however are unable to

verify advertisements. It is beyond nodes' knowledge. In Ethereum, every recorded smart contract transaction is merely a statement that "the smart contract has produced such result", not a statement that "both caller and the smart contract have agreed to such result". Mistaken one for another is fatal.

To make things worse, in Ethereum, whatever values a smart contract gives to users, they are recorded as smart contract data. Ethereum nodes do not examine nor validate transfers of them. Transfers of smart contract assets are processed by smart contracts. Users do not have direct control over their asserts. Smart contracts do. That leaves the door wide open for coin thefts. At the end, Ethereum users are completely at mercy of smart contracts, Ethereum provides no security for them at all. Not transaction security as Ethereum can not guarantee user receiving what they have paid for, nor storage security as Ethereum can not guarantee that user assets will not be taken away without their consent.

B. Value Representation

In most block chains, value is represented by a single number, which may be called scalar value. Recently, NFT (non fungible token) was proposed to represent articles that can not be divided such as a piece of art work. That's all the forms of values block chains can handle today: either as non-divisible tokens or divisible numerically. Real world assets require much more forms for value representation. For example, land can not be represented by a scalar value. To describe a piece of land, one much specify it shape, location, and size. On the other hand, land is fungible.

Lands can be divided and merged. It is inadequate to represent land with NFTs. Further, land may be divided or merged according legal rights attached to it. For example, ownership of real estate includes mortgage right and tenancy right. Real estate owner may assign mortgage right to a bank and tenancy right to a tenant. When loan is paid and lease has expired, he may regain these rights. That requires division and merge of the original real estate token by rights. In fact, virtually all real world assets have legal rights attached to them and it is those rights that give values to the assets. Assets without legal right are worth nothing. Today's block chain technology is far behind in richness of value representation. The result is that much of the world is yet to benefit from block chain technology.

C. Performance

Block chains have long suffered from performance problems. Bitcoin network is only able to process 7 transactions per second, Ethereum 25. It is suggested that there exists a trilemma in block chain technology. I.e, it is impossible to improve three objectives at the same time: security, decentralism, and scalability. So far, all improvements in performance come at expense of either decentralism or security.

III. Overview of Ω Chain

Ω Chain solves these problems with following innovations:

1. **Manifest Smart Contract**: In Ω Chain smart contract call is made in an output script in a Bitcoin style UTXO data structure. It allows participants (a person or a contract) of a transaction to express their intentions by inserting inputs or

outputs to the transaction. A successful contract execution does not automatically make the transaction valid. Nodes will validate that partiers' intentions are met after contract execution. If not, the transaction will be rejected. Result of smart contract execution is expressed as UTXO outputs, not contract's internal data. This scheme solves security problems seen in Ethereum.

2. **Universal token**: Ω Chain provides a powerful and universal form of representation for all types of values: scalar values, non fungible values, fungible non scalar values. Ω Chain treats all these types of values equally and they may be exchanged through UTXO style transactions.
3. **Token division**: Ω Chain token may include a right set. User may divide a token by dividing its right set into complimentary subsets. User may also divide a token by dividing a right according to any criteria. Together with universal token, Ω Chain is able to support the widest range of applications.
4. **Land token**: In Ω Chain, values may take forms other than numbers. It could be a hash value of a polygon. Land as an important class of assets can not be represented by a number, but can by a polygon. In Ω Chain, land tokens are tokens with polygon values. In the genesis block, there is one land token representing the entire earth surface. All other land tokens are split from this genesis land token. Therefore, it uniquely represents a piece of land. This technology is instrumental to many land related applications.
5. **High performance consensus algorithm**: Ω Chain uses a new consensus

algorithm based on dual chain structure. It has no theoretical limit on scalability and has reached over 1000 TPS and 3 seconds final confirmation time without sacrificing security or decentralism.

IV. The Ω Chain Solution

A. Token

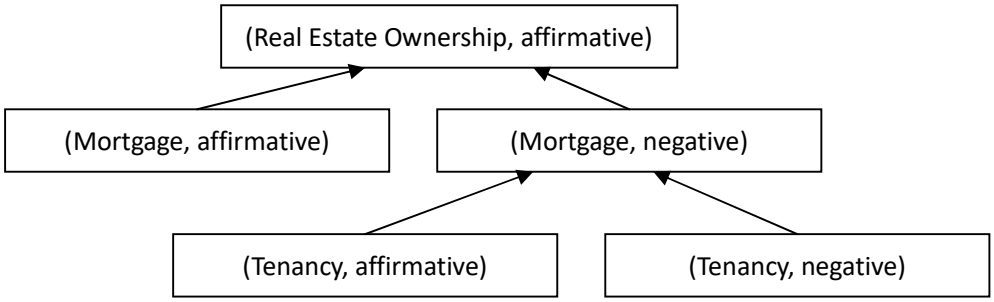
Token is a representation of value. To a large extent, block chain's application is limited by its ability to represent values. In Ω Chain, token is trinity comprising of type ID (a 64-bit number), price (a 64-bit number or a hash value), right set (a hash value representing a right definition or a set of hashes of right definitions). This structure provides a unified representation for scalar values, non fungible values, and fungible non scalar values and permits division of tokens by rights.

All kind tokens are treated in the same way by nodes. Nodes will accept transactions involving all kind tokens, validate them according to their types, and keep books for them. While in Ethereum, Ethers are special. Nodes only keep books for Ether and validate their transfers. ERC2.0 tokens are second class citizens in Ethereum. They are validated and book-kept by respective smart contracts. A significant portion of Ethereum security originates from this model.

B. Rights and Division of Tokens

Applications may define legal rights attached to a token. A tree structure is used

to define relationships among rights. We use a trinity comprising of hash value of its parent right, content (any text of user's choice), an affirmative/negative flag to describe a single right and its relationship to other rights in the tree structure. Two rights are called siblings if they contain the same parent hash value and content, but opposite flags. In a right set, a right may be replaced by a pair of its sibling children, vice versa. Right sets can be used to describe any combination of legal rights in real world. E.g., a "real estate ownership" right may be divided into two child "mortgage" rights, one with affirmative flag and another with negative flag. The child right with negative flag may further be divided into two child "tenancy" rights, one with affirmative flag and another with negative flag.



A home owner would own a home token with (Real Estate Ownership, affirmative) right. When he takes a mortgage against the home, he may divide the token into two tokens by dividing the original (Real Estate Ownership, affirmative) right into (Mortgage, affirmative) and (Mortgage, negative) child rights, thereby creating two tokens containing respective child rights. He will give the token with (Mortgage, affirmative) right to the bank and keep the other to himself. When he rents out the home, he may divide the right into (Tenancy, affirmative) and (Tenancy, negative) child rights, thereby creating two tokens containing respective child rights. He will give the token with (Tenancy, affirmative) to the tenant and keep the other to himself. When the lease expires, home owner may merge the token the tenant returns

and the token in his hand into a token with (Mortgage, negative) right. When the loan is paid off, home owner may again merge the token the bank returns and the token in his hand into a token with (Real Estate Ownership, affirmative) right. When a buyer is going to buy the home, he would examine the home owner's home token. If the token does not contain (Real Estate Ownership, affirmative) right, he would know the home is encumbered and not to buy it. This example demonstrates that, without knowing the meanings of words such as "Mortgage" and "Tenancy", Ω Chain is still able to provide a secured mechanism for transfer of real estate values. Application of this powerful mechanism is, of course, not limited to real estate transaction. In fact, almost all real world transactions involve transfer of legal rights which can be defined and operated upon using the scheme described above.

C. Transactions

Ω Chain uses Extended UTXO transaction model, which is an extension to Bitcoin style UTXO model. An Extended UTXO transaction includes input, definition, and output sections. Like Bitcoin, a transaction input includes a reference to unspent output of a previous transaction and a signature script. A transaction output includes a token and a script. The script could be a lock script like what is in Bitcoin or a contract call script. Definition section is used to define rights or geometry entities. Once a definition is recorded, it can not be changed and may be used by any other transaction.

D. Smart Contracts

1. Problems in Present Block Chain Smart Contracts

Ethereum style smart contracts are used in today's block chains, which is characterized by that nodes will consider a smart contract transaction legal and will record it as long as execution of smart contract is successful, and the result of execution is stored as contract data. The result is that user is unable to claim his right to assets independent of smart contracts. This is not a contract model in legal sense and has caused many security problems.

Legally, a contract (transaction) is a meet of minds. Today's smart contract model does not allow validation of agreement of minds. It may be argued that user may express his intention through parameters provided in contract's call script. However, this intention can only be understood by the smart contract, not by the node running it. And there is no way for nodes to validate that the contract's understanding of these parameters is in agreement with user's understanding of them. Thus, nodes have no choice but accepting the transaction when the contract executed successfully. That means under today's smart contract model, smart contract's unilateral interpretation is taken as agreement of both parties. If a user purchase certain product, and he assumes the unit of quantity is kilograms. While the smart contract interprets the unit of quantity as pounds, the contract will execute successfully but delivers less products than user have paid for.

A more serious problem is that in Ethereum style smart contract model, it is the

smart contracts that keep and manage user assets, not users themselves. In Ethereum white paper, it was said that smart contracts are agents of their creators. By keeping and managing user assets, smart contract exceeds their role of creators' agents and take the position of neutral book recorder. A conflict of interests exists between these roles. It creates a condition that a smart contract creator is able to steal from his customers as some recent DEFI contracts have shown. Even if the creator is without malice, programs could have bugs. Hacker may steal user assets under smart contract's control.

This type smart contract breaches the trustless nature of block chain. Any smart contract has only one creator. While block chain may guarantee that smart contract codes are identical in every node, it can't validate security of the code. If smart contracts act exactly as agents of their creators, this wouldn't be a problem. When smart contracts also play the role of customers' book keepers, the ledgers they keep are not decentralized ledgers because all the book keepers are in fact one and have the same interest. Such ledgers are no different than traditional distributed ledger. Logically, when block chain can not provide security, people seek security outside block chain. Therefore we see rise of smart contract code audit industry. It is ironical that block chains that brand themselves decentralized and trustless rely on centralized code audit companies to provide trust. In most recent DEFI thefts incidents, the codes that enabled their creators to steal customer assets are all certified by auditors. This evidences the total failure of Ethereum style smart contract model.

2. Manifest Smart Contracts

Ω Chain provides a new smart contract model, named Manifest Smart Contracts. Under this model, users make offers and express their expectations in the form of UTXO inputs and outputs. Smart contracts attempts to fulfill users' expectations and may express their expectations by filling transaction with additional UTXO inputs and outputs. In validating transactions, nodes examine that both sides' expectations are met by comparing inputs and outputs of the final UTXO. This model includes:

- User submits a UTXO style partial transaction;
- Node runs Virtual Machine to execute smart contract calls in the UTXO outputs;
- Smart contract inserts input, output, or definition to the transaction;
- Node validates integrity of the transaction. Only integral transaction will be recorded in blocks.

We shall use an example to illustrate this model. Assuming a smart contract issues α coins, and advertises that user may trade 2 ω coins for 5 α coins. A user wishing to make the trade will submit a partial below to any node:

Input	Output
UTXO（2ω）	2ω + smart contract call script
	5α + user's lock script

This transaction means that user will take out 2 ω coins, give them to the smart contract and want to receive 5 α coins in return. Since the transaction includes a smart contract call script, node will execute the smart contract. During execution, smart contract adds an input to the transaction. The input references an unspent output with a 5α-coins token belonging to the smart contract. The smart contract then returns

successfully.

Input	Output
UTXO（2ω）	2ω + smart contract call script
UTXO（5α）	5α + user's lock script

Node validates integrity of the final transaction. I.e., check whether totals input and total outputs of every coin type are equal. In this case, since the 5 α coins input provided by the smart contract equals to the 5 α coins output asked by the user, the contract is valid. However, if the input supplied by the smart contract is 4 α coins or 6 α coins, the transaction will be held invalid. Neither party will suffer a loss.

Because user receives an UTXO output, he does not rely on smart contract to keep a book. Even if the smart contract may still keep a record, the record has no external effect and does not affect user's ownership of the 5 α coins he received. Therefore, neither smart contract creator nor hacker could steal it through the smart contract program.

Of course, hackers may still explore security loopholes in smart contract to cause it issuing unwarranted tokens to him. This will result in loss to the smart contract creators, but not their customers. This type security risk is the same as someone losing Bitcoins for failure to keep private key safe. He only has himself to blame.

Not only users may express their expectations from a transaction, smart contracts may too. For example, in a scenario of payment on deliver, user initiates a contract call for delivery. The contract may demand payment by inserting an output for coins

paid to itself. The transaction will fail unless user provides input matching this output.

Manifest Smart Contracts allows combination of multiple smart contract calls in one transaction, while it is impossible under Ethereum. In a payment on delivery scenario, a transaction would involve three parties: a seller who wants to receive payment, a shipping company who needs confirmation of delivery, and a customer who is expected to pay the bill. Under Manifest Smart Contracts model, all these can happen in one atomic transaction. While under Ethereum, two transactions would be required, increasing the risk of fraud.

The key to smart contract security is to let users control their own assets directly without meddling of a smart contract. Some may argue that if users insist control of their assets, many DEFI applications would be impossible. After all, how could a DEFI contract lend Alice's money to Bob if Alice does not transfer her money to the DEFI contract first? This kind arguments mistake control for ownership. From a legal point of view, when Alice transfers money to a DEFI contract, she is not giving the contract ownership of the money, instead she only gives the contract control to move the money under certain condition, such as lending to Bob. Therefore, if a mechanism exists for transfer of control without transfer of ownership, the purpose of the DEFI contract could still be achieved. Ω Chain provides two mechanisms that would achieve that purpose.

Comparison of Ethereum style smart contract and Manifest Smart Contracts

	Present Smart Contract Model	Manifest Smart Contracts
Expression Of Intention	Transfer coins to smart contract; Parameters understood by smart contract.	Any token may be transferred to smart contract; Parameters understood by smart contract; User expected result.
Execution	User must accept if contract executes	Validate that final transaction meets expectations of parties. If not, transaction is invalid.
Result	Stored in smart contract's data	UTXO output, only owner may unlock
Security	Phishing contract, contract code error, misunderstanding may result in unexpected result; loss of customer assets due to hacker attack	Phishing contract, contract code error, misunderstanding will result in invalid transaction and be rejected, no customer losses. Impossible for hacker to steal customer by attacking contracts.

E. Land Token

In Ω Chain, values may take non numeric forms, one of which is land token. In Ω Chain, land is represented by a polygon with geological coordinates for its vertices. All land tokens comes from one polygon in the genesis block representing the entire earth surface. Polygons may be divided or merged, forming new polygons. Nodes verify that the combined polygons before and after division or merge are identical. Thus every polygon token uniquely represents a piece of land or sea area without conflict with other polygon tokens. Border of a country may require millions of coordinates. Ω Chain solves the problem of representing large and uncertain polygon boundaries with an innovative algorithm making it possible to process them effectively in block chain. Land tokens may have rights attached them, such as

ownership, mortgage right, tenancy right, etc. Land is an important class of assets. To be able to process lands as polygons in block chain broadens application of block chain technology to a new area.

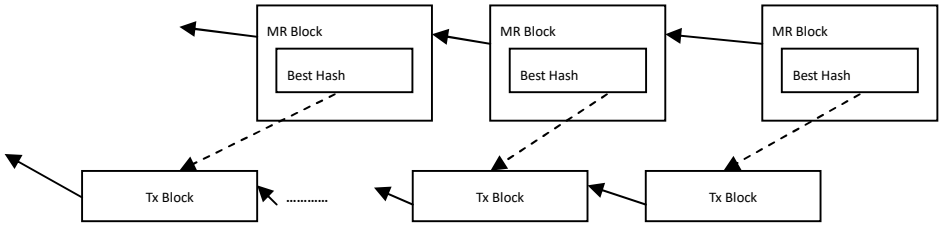
F. Ω Consensus Algorithm

In the world of block chain, there is a conjecture of trilemma which says for the three block chain objectives: decentralism, security, and scalability, it is impossible to make improvement to all of them at the same time. The root of the trilemma lies in block chain's single chain data structure. Blocks are carrier of these objectives. Due to inherent conflict among these objectives, a block can not fulfill all three objectives at the same time. By using dual chain data structure, Ω Chain is able to distribute the objectives to blocks in different chains thereby avoiding the trilemma. Ω Chain has achieved unlimited scalability without sacrificing security and decentralism.

Ω Chain consists of a mining right chain and a transaction chain. The mining right chain records who has obtained the right to generate transaction blocks (right to mine). The transaction chain records user transactions. Mining right chain adopts a proof of work system with a target difficulty of one block every 10 minutes, same as Bitcoin network. Transaction chain does not use proof of work system. Miners take turns to generate and sign certain number (200) of transaction blocks according to their order and place in the mining right chain. Because at any moment, it is certain who has the right to produce transaction blocks, there is no need to use a competitive mechanism to determine a valid transaction block. Miner's signature is sufficient. Therefore, transactions could be processed very effectively and efficiency is only limited by nodes' ability to validate and pack transactions.

For security purpose, every mining right block includes a reference to a transaction block. All these references must fall within one chain without branch and in an order consistent with the blocks in mining right chain. As a result of reference,

the security generated by proof of work system in mining right chain is passed to the transaction chain, giving transactions in Ω Chain network the same level of security as transactions in Bitcoin network.



Because miner must first generate a mining right block before he may generate transaction blocks, at any moment there exists a queue of miners who have had their mining right blocks recorded and have not yet begin to generate transaction blocks. This queue is called waiting list. Ω Chain uses a POW difficulty adjusting mechanism to control the length of waiting list around 40. That means when a miner begins to generate transaction blocks, his right to mine has been confirmed about 40 times. It is impossible to overthrow such consensus. I.e, in Ω Chain it is impossible to have side chain because two miners legally claim they have right to mine. Therefore, unless a malice miner double signs two blocks at the same height, a transaction block is final once generated.

To deter malice miner, Ω Chain requires miner to provide ω coins as collaterals. To encourage miners providing more collaterals, Ω Chain adjusts proof of work target according to the amount of collateral provided. In case a breach happens, Ω Chain will use the collateral to compensate user losses. The compensation amount is set to 10000 times of transaction fees. Thus, in Ω Chain, transaction fee is a form of insurance premium.

The purpose of requiring collaterals is to deter malice miners. Ω Chain does not rely on this scheme to prevent bifurcation of chain. Because every mining right block must reference a transaction block, when side chain occurs due to double sign, the mining right chain will pick one of the branches to follow. The other branch would be discarded. Thus a highly security minded user could wait for certain number of

confirmations before concluding a transaction like a Bitcoin user would do. For large value transactions, users should take this approach. For small value transactions, users may treat recorded transactions final and recover losses in compensation pay outs.

Ω Chain also adjusts proof of work target to stimulate miner to speed up transaction process rate. Specifically, in mining right blocks, miners may report other miners' transaction processing rate they have observed. A miner's proof of work target is adjusted by comparing his averaged transaction processing rate reported by others (excluding top and bottom 25% scores) against average rate of all miners. Thus those who can process transactions faster than average would have better change to win mining right.

Since the invention of block chain, miners have invested significant amount of resources to improve their hashing power instead of ability to process transactions, because block awards are significantly more than transaction fees. On one hand, Ω Chain turns transaction fees into insurance premium encouraging users to pay more transaction fee. On the other hand, Ω Chain stimulates miners to improve transaction processing capability by rewarding high TPS miners with better chance of wining mining right. Therefore, miner would make balanced investments between improving hashing power and improving transaction processing capability. This will help Ω Chain to become a highly efficient block chain.

V. Conclusion

We have identified problems in today's block chain technology, especially the fatal security flaws in smart contracts. We have developed solutions to these problems and have demonstrated that Ω Chain is technologically superior to today's mainstream block chains, especially Ethereum. We firmly believe that Ω Chain will reshape the world of block chain and be a phenomenal success.