Y2 (ethyl xylene)

Arab League Y2 New Energy Fuel Laboratory

Smart Contract

Transforming non-renewable resources into renewable resources

Add: Middle East International Trade Currency

League of Arab League emblem, Y2 coin LOGO, Y2 New Energy Fuel Laboratory LOGO

Arab League Secretary General

Ahmed Aboul Gheit Signature Confirmation Y2

Summary:

When Satoshi started the Bitcoin blockchain in January 2009, he also introduced two new untested revolutionary concepts to the world. The first is bitcoin, a decentralized peer-to-peer online currency that maintains value without any asset guarantee, intrinsic value, or central issuer. So far, Bitcoin has attracted a lot of public attention. In terms of politics, it is a currency that does not have a central bank, and it has drastic price fluctuations.

However, Satoshi Nakamoto's great experiment has an equally important part of Bitcoin: the concept of blockchain based on proof of work allows people to reach a consensus on the order of transactions. Bitcoin as an application can be described as a first-to-file system: if a person has 50 BTCs and sends the 50 BTCs to both A and B, only the first confirmed transaction will be effective. There is no inherent method to determine which of the two transactions comes first. This problem has hindered the development of decentralized digital currency for many years.

Nakamoto's blockchain is the first reliable decentralized solution. Now, developers' attention begins to quickly shift to the second part of Bitcoin technology, and how blockchains are applied to areas other than money. Applications that are often mentioned include the use of digital assets on the chain to represent custom currencies and financial instruments (colored coins), ownership of certain basic physical devices (smart

assets), and non-replaceable assets such as domain names (domain name coins) And more advanced applications such as decentralized exchanges, financial derivatives, point-to-point gambling and chain identity and reputation systems.

Another important area that is often asked is "smart contracts" - systems that automatically transfer digital assets based on pre-arranged rules. For example, a person may have a storage contract in the form of "A can withdraw up to X coins per day, B can have up to Y per day, A and B can be freely extracted together, and A can withdraw the withdrawal right of B". The logical extension of this kind of contract is decentralized autonomous organizations (DAOs)—long-term smart contracts that contain the assets of an organization and encode the rules of the organization. The goal of Y2 coin is to provide a blockchain with a built-in mature Turing complete language. This language can be used to create contracts to encode arbitrary state transition functions. Users can simply implement logic using a few lines of code. Create all the systems mentioned above and many other systems that we couldn’t imagine.

table of Contents

- ○ Currency and issuance
- ○ Extensibility
- ● Overview: Decentralized Applications
- ● Conclusion
- ● Comments and Advanced Reading

history

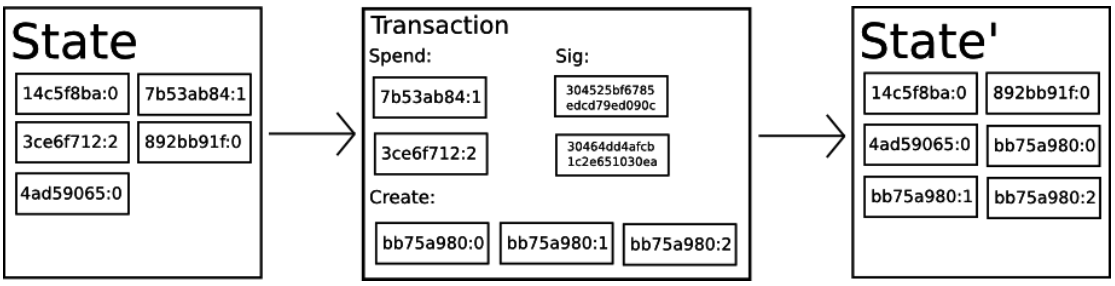
The concept of decentralized digital currency, like the alternative application of property registration, was brought up decades ago. Most anonymous electronic cash agreements in the 1980s and 1990s were based on the Chaumian blinding technique. These e-cash agreements provide highly private currency, but these protocols are not popular because they all rely on a centralized intermediary. In 1998, Wei Dai's b-money first introduced the idea of creating currency by solving computational problems and decentralized consensus, but the proposal did not give a concrete method of how to achieve decentralized consensus. In 2005, Hal Finney introduced the concept of "reusable proofs of work", which uses both b-money's thinking and Adam Back's computationally difficult hash cash (Hashcash).) Difficulties to create cryptocurrency currency. However, this concept is again lost in idealization because it relies on trusted computing as the back end. Because currency is a pre-application application, the order of transactions is crucial, so decentralized currencies need to find ways to

achieve decentralized consensus. The main obstacle for all previous bitcoin electronic currency agreements is that although the research on how to create a secure Byzantine-fault-tolerant multi-consensus system has lasted for many years, the above agreement only solved half of the problems. . These protocols assume that all participants of the system are known and generate a form of security boundary such as "If the N party participates in the system, then the system can tolerate malicious participants of N/4". However, the problem with this assumption is that, in the case of anonymity, the security boundary set by the system is vulnerable to witch attacks because an attacker can create thousands of nodes on a server or botnet, thereby unilaterally ensuring the majority Share.

Nakamoto's innovation is the introduction of a concept that combines a very simple node-based decentralized consensus protocol with a workload proof mechanism. The node obtains the right to participate in the system through the workload proof mechanism, and the transaction is packaged into "blocks" every ten minutes to create an ever-growing blockchain. A node with a lot of power has greater influence, but it is much more difficult to obtain more power than the entire network than to create one million nodes. Although the Bitcoin blockchain model is very simple, it has proved to be good enough for use. In the next five years, it will become the cornerstone of more than 200 currencies and

agreements worldwide.

Bitcoin as a state transition system



From a technical point of view, Bitcoin books can be thought of as a state transition system that includes all existing Bitcoin ownership states and "state transfer functions." The state transition function takes the current state and transaction as input and outputs a new state. For example, in the standard banking system, the status is a balance sheet. A request to transfer X USD from the A account to the B account is a transaction. The state transfer function subtracts X USD from the A account and increases the B account. X dollars. If the balance of the A account is less than X dollars, the state transition function will return an error message. So we can define the state transition function as follows:

```
APPLY(S,TX) > S’ or ERROR
```

In the banking system mentioned above, the state transition function is as follows:

```
APPLY({ Alice: $50, Bob: $50 },”send $20 from Alice to Bob”) = { Alice: $30,Bob: $70 }
```

Buts:

```
APPLY({ Alice: $50, Bob: $50 },”send $70 from Alice to Bob”) = ERROR
```

The "state" of the Bitcoin system is a collection of all Bitcoin (technically known as "unspent transaction outputs or UTXO") that has been dug up and has not been spent. Each UTXO has a face value and an owner (defined by the address of the 20-byte cryptographic public key). A transaction includes one or more inputs and one or more outputs. Each input contains a reference to an existing UTXO and a cryptographic signature created by the private key corresponding to the owner's address. Each output contains a new UTXO added to the state.

In the bitcoin system, the state transition function APPLY(S,TX)->S' can be roughly defined as follows:

1. Each input of the transaction:

- If the referenced UTXO does not exist in the current state (S), an error message is returned
- If the signature is inconsistent with the UTXO owner's signature, an error message is returned

2. If all UTXO input facet amounts are less than all UTXO output facet amounts, an error message is returned

3. Returning to the new state S', all input UTXOs are removed in the new state S', and all output UTXOs are added.

The first part of the first step prevents the sender of the transaction from spending the non-existent Bitcoin, and the second part prevents the

sender of the transaction from spending other people's Bitcoins. The second step ensures the conservation of values. Bitcoin's payment agreement is as follows. Suppose Alice wants to send Bob 11.7 BTC. In fact, Alice cannot have exactly 11.7 BTC. Assume that the minimum amount of Bitcoin she can get is: 6+4+2=12. So she can create a transaction with 3 inputs and 2 outputs. The first output has a denomination of 11.7 BTC, the owner is Bob (Bob's bitcoin address), the second output has a denomination of 0.3 BTC, and the owner is Alice himself, which is the change.

If we have a trusted centralized service organization, the state transition system can be easily implemented and the above functions can be simply coded accurately. However, we want to build the bitcoin system as a decentralized currency system. To ensure that everyone agrees with the order of transactions, we need to integrate the state transition system with a consensus system. Bitcoin's decentralized consensus process requires nodes in the network to constantly try to pack transactions into "blocks." The network is designed to generate a block approximately every ten minutes. Each block contains a timestamp, a random number, a reference to the previous block (ie, hash), and all transactions that occurred since the previous block was generated. List. In this way, a continuously growing blockchain is created over time. It is continuously updated to represent the latest state of the Bitcoin book.

According to this paradigm, the algorithm for checking whether a block is valid is as follows:

1. Check if the previous block referenced by the block exists and is valid.

2. Check if the time stamp of the block is later than the time stamp of the previous block and earlier than the next 2 hours.

3. Check whether the workload of the block is valid.

4. Assign the final state of the previous block to S[0].

5. Suppose TX is a block transaction list containing n transactions. For all i belonging to 0...n-1, a state transition S[i+1] = APPLY(S[i], TX[i]) is performed. If any transaction i makes a mistake in the state transition, exit the program and return an error.

6. The return is correct. The state S[n] is the final state of this block.

Essentially, every transaction in a block must provide a correct state transition. Note that "state" is not coded into blocks. It is purely an abstraction that is remembered by the check node. For any block, you can start from the state of creation and add each transaction in each block in order to (really) calculate the current state. In addition, you need to pay attention to the order in which transactions are included in the block. If there are two transactions A and B in a block, B spends UTXO created by A. If A is before B, this block is valid, otherwise, this block is invalid.

An interesting part of the block verification algorithm is the concept of

"workload proof": SHA256 hashes each block and treats the resulting hash as a 256-bit length value, which must be less than the dynamically adjusted target value. At the time of writing this book, the target number is approximately 2^190. The purpose of the proof of workload is to make the creation of the block difficult, thereby preventing the witch attacker from maliciously regenerating the blockchain. Because SHA256 is a completely unpredictable pseudo-random function, the only way to create a valid block is to simply try and error continuously, and increase the number of random numbers continuously to see if the new hash value is smaller than the target value. If the current target value is 2^192, it means that it takes 2^64 attempts to generate a valid block on average. In general, the Bitcoin network resets the target value every 2018 blocks, ensuring that an average block is generated every ten minutes.

Let us analyze what happens when a Bitcoin network has a malicious attacker. Because Bitcoin's cryptographic foundation is very secure, attackers would choose to attack parts that are not directly protected by cryptography: the order of transactions. The attacker's strategy is very simple:

1. Send 100BTC to the seller to purchase goods (especially electronic goods that do not need to be mailed).

2. Wait until the product is issued.

3. Create another transaction and send the same 100BTC to your own

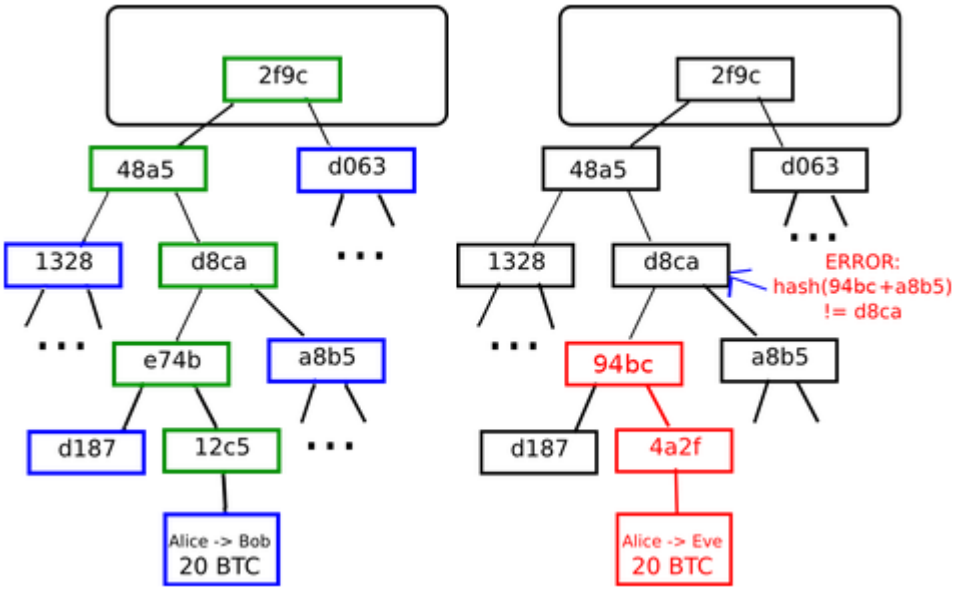
account.

4. Make the Bitcoin network believe that the transaction sent to your account is the first to be sent.

Once step (1) occurs, the transaction will be packaged into blocks within a few minutes, assuming the 270000th block. About an hour later, there will be five blocks behind this block, each of which indirectly points to the transaction to confirm the transaction. At this time the seller received the payment and shipped it to the buyer. Because we assume that this is a digital product, the attacker can receive the goods immediately. Now, the attacker creates another transaction and sends the same 100BTC to his own account. If the attacker only broadcasts this message to the entire network, this transaction will not be processed. A state transition function APPLY(S,TX) will be run and it is found that this transaction will take UTXO which is no longer in the state. Therefore, the attacker diverges the blockchain and regenerates the 270000th block from the 269999th block as the parent block, where the new transaction replaces the old transaction. Because the block data is different, this requires a proof of workload. In addition, since the new 270000th block generated by the attacker has a different hash, the original 270001 to 270005 blocks do not point to it, so the original blockchain and the attacker's new block are completely Detached. In the case of a blockchain fork, the long branch of the blockchain is considered to be an

honest blockchain. Legitimate ones will follow the original 270005 block. Only the attacker will be in the new 270000 block. . In order for the attacker to maximize his blockchain, he needs to have more maneuvering power than the entire network except him (ie, 51% attack).

Merkel Tree



Left: Only providing a small number of nodes on the Merkle tree is sufficient to give a legal proof of the branch.

Right: Any attempt to change any part of the Merkel tree will eventually lead to inconsistencies somewhere in the chain.

One of the important scalability features of the Bitcoin system is that its blocks are stored in multi-level data structures. The hash of a block is actually just the hash of the block header. The block header is the length of the root hash of the Merkel tree that contains the timestamp, the random number, the last block hash, and all block transactions are stored. About 200 bytes of data.

A Merkel tree is a binary tree consisting of a set of leaf nodes, a set of intermediate nodes, and a root node. The lowest number of leaf nodes contains the underlying data. Each intermediate node is a hash of its two child nodes. The root node is also a hash of its two child nodes and represents the top of the Merkel tree. The purpose of the Merkel tree is to allow chunks of data to be scattered: nodes can download chunk headers from one source, download other parts of their associated tree from another source, and still be able to confirm that all data is correct . This is because of the hash-diffusion: if a malicious user tries to add a fake transaction to the lower part of the tree, the resulting changes will cause changes to the upper nodes of the tree and changes to the higher-level nodes, eventually leading to the root node. The changes and changes to the block hash, so the agreement will record it as a completely different block (almost certainly with an incorrect workload proof).

Merkel's agreement is crucial to the long-term sustainability of Bitcoin. In April 2014, a full node in the Bitcoin network - a node that stores and processes all data for all blocks - required 15 GB of memory space and grew at a rate of more than 1 GB per month. At present, this storage space is acceptable for desktop computers, but mobile phones have not been able to load such huge data. Only commercial organizations and enthusiasts will act as complete nodes in the future. The simplified

payment confirmation (SPV) protocol allows another type of node to exist. Such a node is called a "light node." It downloads the block header, uses the block header to confirm the proof of workload, and then only downloads the Merkel tree "branch associated with its transaction. ". This allows the light node to securely determine the status of any Bitcoin transaction and the current balance of the account by simply downloading a small portion of the entire blockchain.

Other blockchain applications

The idea of applying the idea of blockchain to other areas has long appeared. In 2005, Nick Szabo proposed the concept of “title ownership of property”. This article describes how the development of replicated database technology can make blockchain-based systems can be used to register land ownership, including creating property rights such as ownership and illegal occupation. Detailed frameworks for concepts such as Georgia and land tax. However, unfortunately there was no practical copy database system at that time, so this protocol was not put into practice. However, since the successful decentralized consensus development of the Bitcoin system in 2009, many other applications of the blockchain began to appear rapidly.

- namecoin - Created in 2010, known as a decentralized name registration database. Decentralized protocols such as Tor, Bitcoin, and BitMessage require some method of confirming accounts so that other

people can interact with users. However, the only available identity in all existing solutions is a pseudo-random hash such as 1LW79wp5ZBqaHW1jL5TciBCrhQYtHagUWy. Ideally, people want to have an account with a name like "george". However, the problem is that if someone can create a "george" account, other people can also create a "george" account to pretend. The only solution is first-to-file. Only the first registrant can successfully register, and the second cannot register the same account again. This problem can use Bitcoin's consensus protocol. Domain Name Coin is the earliest and most successful system for implementing a name registration system using a blockchain.

- Colored coins - The purpose of colored coins is to provide people with the ability to create their own digital currency on the Bitcoin blockchain or, more importantly, the currency-digital token. According to the color currency agreement, people can issue new currencies by assigning colors to a particular Bitcoin UTXO. This protocol recursively defines other UTXOs as the same color as the transaction input UTXO. This allows the user to keep UTXOs that contain only a certain color. Sending these UTXOs is like sending ordinary bitcoins, and judging the received UTXO colors by tracing back the entire blockchain.
- Metacoins - The idea of Metacoins is to create a new protocol on the Bitcoin blockchain, using bitcoin transactions to save the currency transactions, but using a different state transfer function APPLY'. Since

the currency exchange protocol cannot block invalid currency transactions on the Bitcoin blockchain, adding a rule that if APPLY'(S,TX) returns an error, this protocol will default to APPLY'(S,TX) = S. This provides a simple solution for creating arbitrary, advanced cryptographic currency protocols that cannot be implemented in the Bitcoin system, and the development cost is very low because the problems of the network have already been handled by the bitcoin protocol.

Therefore, in general, there are two ways to establish a consensus protocol: establishing an independent network and establishing an agreement on the Bitcoin network. Although applications such as domain name coins have succeeded using the first method, the implementation of this method is very difficult, because each application needs to create a separate blockchain and establish and test all state transitions and network codes. In addition, we predict that the application of decentralized consensus technology will obey the power law distribution. Most applications are too small to guarantee the security of the free blockchain. We also notice a large number of decentralized applications, especially decentralization. Autonomous organizations need to interact with applications.

On the other hand, there are drawbacks to the Bitcoin-based approach, which does not inherit the characteristics of Bitcoin that can simplify Payment Confirmation Pay (SPV). Bitcoin can make it easier to confirm

payment because Bitcoin can use blockchain depth as a validation agent. At some point, once the ancestors of a transaction are now far enough away, they can be considered part of the legal state. In contrast, a currency exchange protocol based on the Bitcoin blockchain cannot force the blockchain to exclude transactions that do not comply with the currency exchange protocol. Therefore, the simplified payment confirmation of the safe currency protocol requires backward scanning of all blocks until the initial point of the blockchain to confirm whether a certain transaction is valid. Currently, all "light" implementations of bitcoin-based dollar currency agreements rely on trusted servers to provide data. This is only a rather suboptimal result for cryptocurrencies that eliminate the need for trust.

script

Even if the bitcoin protocol is not extended, it can achieve "smart contracts" to some extent. Bitcoin's UTXO can be owned by more than one public key, or it can be owned by more complex scripts written in a stack-based programming language. In this mode, spending such UTXO must provide data that meets the script. In fact, the basic public key ownership mechanism is also implemented by script: the script takes the elliptic curve signature as input, verifies the transaction and owns the address of this UTXO, and returns 1 if the verification is successful, otherwise it returns 0. More complex scripts are used for other different

application scenarios. For example, one can create a script (multi-signature) that requires the collection of two of the three private keys for transaction confirmation. This script is useful for corporate accounts, savings accounts, and certain commercial agents. Scripts can also be used to send rewards to users who solve computational problems. People can even create such a script "If you can provide proof that you have sent a certain amount of dog money to me for simplified confirmation payment, this Bitcoin UTXO is yours", in essence, Bitcoin system allows different passwords Learn currency decentralized exchange.

However, the bitcoin system's scripting language has some serious limitations:

- Missing Turing Completeness – This means that although the bitcoin scripting language can support multiple calculations, it cannot support all calculations. The major deficiency is the loop statement. The purpose of not supporting loop statements is to avoid infinite loops in transaction confirmation. In theory, this is an obstacle that can be overcome for scripting programmers, because any loop can be simulated with multiple iterations of the if statement, but doing so can lead to inefficiencies in script space utilization, for example, implementing a The alternative elliptic curve signature algorithm would probably require 256 repetitions of multiplication, each time requiring separate encoding.

Value-blindness. The UTXO script does not provide fine-grained control over the withdrawal amount of the account. For example, a powerful application of the oracle contract is the hedging contract. Each of A and B send bitcoins worth $1,000 to the hedging contract. 30 days later, the script sends Bitcoin worth $1,000 to A to B. Send the remaining bitcoins. Although achieving a hedge contract requires a oracle to determine how much a bitcoin value is for a dollar, this mechanism has made significant progress in reducing trust and infrastructure compared to today's fully centralized solution. However, since UTXO is indivisible, the only way to achieve this contract is to use very many UTXOs with different denominations (eg, for each k with a maximum of 30, there is a 2xk UTXO). Make the oracle predict the correct UTXO to send to A and B.

- Missing state – UTXO can only be spent or not spent, which leaves no room for multi-phase contracts or scripts that require any other internal state. This makes it difficult to implement multi-stage option contracts, decentralized exchange offers, or two-stage cryptographic commitment agreements (which are necessary to ensure that rewards are calculated). This also means that UTXO can only be used to establish simple, one-time contracts, rather than such contracts with more complex states such as decentralized organizations, making meta-protocols difficult to achieve. Binary status combined with value blind means that another important application - the withdrawal limit - is impossible to achieve.

Blockchain-blindness - UTXO does not see blockchain data, such as random numbers and the hash of the previous block. This defect deprives the scripting language of the potential value based on randomness, which severely restricts applications in other areas such as gaming.

We have examined three methods for building advanced applications on cryptocurrency: building a new blockchain, using scripts on the bitcoin blockchain, and establishing meta-coinage agreements on bitcoin blockchains. The method of establishing a new blockchain can freely implement arbitrary features, the cost being development time and fostering efforts. The method of using scripts is very easy to implement and standardize, but its ability is limited. Although the currency exchange protocol is very easy to implement, it has the disadvantage of poor scalability. In the Y2 coin system, our goal is to establish a common framework that can have all the advantages of these three modes simultaneously.

Y2 currency

The Y2 currency is based on the digital currency of the Middle East Arab League that changes the distribution of crude oil renewable resources. The Arab League summit adopted a resolution to increase the use of Y2: international currency for circulation will be implemented in June. Y2 coin is a digital currency issued by Y2 New Energy Laboratory. Y2 is

issued to speed up the development of Y2 new fuel additive. Y2 new fuel additive is expected to be available for the first time in 2020, and will increase the existing Y2 by about 20 times, by 2035. Achieve more than 170 times; 1 coin equals 1 bottle of Y2; Save the world, defend global renewable resources (UN called the resource rescue plan, Y2 and the United States became the initiator of the rescue plan) Technical aspects have Turing completeness, value awareness (Value-awareness, blockchain-awareness, and multi-state added power are much stronger than the smart contracts that bitcoin scripts can provide.

Y2 account

In the Y2 coin system, the status is made up of objects called "accounts" (each account is a 20-byte address) and the state of transferring the value and information between the two accounts. The Y2 coin account contains four parts:

- Random number used to determine the counter that can only be processed once per transaction
- Current Y2 balance in your account
- The account's contract code, if any
- Storage of accounts (default is empty)

News and transactions

Y2 coin news is somewhat similar to bitcoin trading, but there are three important differences between the two.

First, Y2 coin messages can be created by external entities or contracts, whereas bitcoin transactions can only be created externally.

Second, Y2 coin messages can optionally contain data.

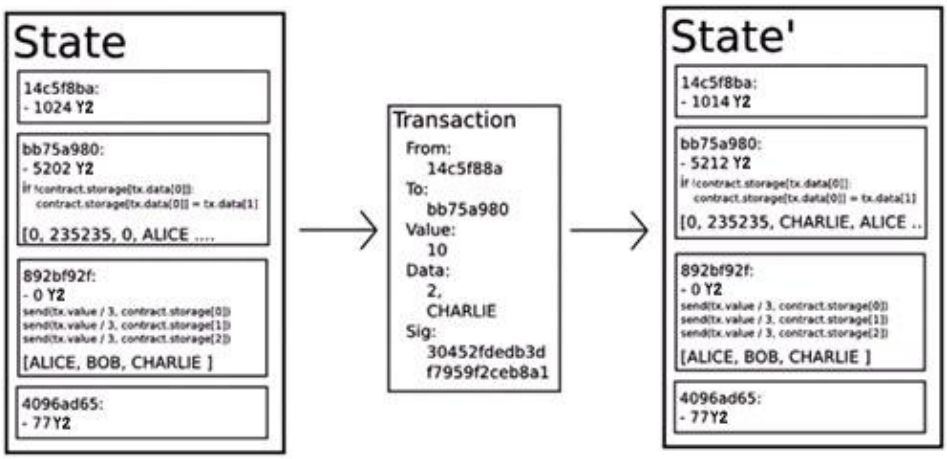
Third, if the recipient of the Y2 coin message is a contract account, you can choose to respond, which means that the Y2 coin message also contains a function concept.

The "transaction" in Y2 currency refers to a signature data packet that stores a message sent from an external account. The transaction contains the recipient of the message, the signature used to confirm the sender, the balance of the Y2 credit account, the data to be sent, and the two values known as STARTGAS and GASPRICE. To prevent exponential explosions and endless loops of code, each transaction requires restrictions on the computational steps that result from the execution of the code—including the initial message and all messages that result from the execution. STARTGAS is the limit, GASPRICE is the cost of each calculation step. If during the execution of the transaction, "the fuel is used up," all the status changes are restored to their original status, but the transaction fees already paid are not recoverable. If fuel is still left when the execution of the transaction is suspended, the fuel will be returned to the sender. The creation contract has a separate transaction type and corresponding message type; the contract's address is calculated based on the account's random number and the hash of the

transaction data.

An important consequence of the messaging mechanism is that the “primary citizen” property of the Y2 currency - the contract has the same rights as the external account, including the right to send messages and create other contracts. This allows the contract to serve multiple different roles at the same time. For example, a user can make a member of a decentralized organization (a contract) become a mediation account (another contract), customize a quantum-based certification for a paranoid use of the blue An individual with a Porter signature (third contract) and a self-signed entity using an account secured by five private keys (fourth contract) provide intermediation services. The strength of the Y2 coin platform is that decentralized organization and agency contracts do not need to care what type of account each participant of the contract is.

Y2 currency state transfer function



Y2 currency state transfer function: `APPLY(S,TX) -> S'`, It can be defined as follows:

1. Check that the format of the transaction is correct (that is, has the correct value), that the signature is valid, and that the random number matches the random number of the sender's account. If not, an error is returned.

2. Calculate the transaction fee: fee = STARTGAS * GASPRICE, and determine the sender's address from the signature. Subtract the transaction fee from the sender's account and increase the sender's random number. If the account balance is insufficient, an error is returned.

3. Set the initial value GAS = STARTGAS and subtract a certain amount of fuel value from the number of bytes in the transaction.

4. Transfer value from the sender's account to the recipient's account. If the receiving account does not yet exist, create this account. If the receiving account is a contract, run the contract code until the code runs out or the fuel runs out.

5. If the sender's account does not have enough money or if the execution of the code runs out of fuel and the value transfer fails, the original status is restored, but the transaction fee is also required to be paid and the transaction fee is added to the account.

6. Otherwise, return all remaining fuel to the sender and the spent fuel is sent to the distribution center as a transaction fee.

Code execution

The Y2 coin contract code is written in a low-level stack-based bytecode language. Code consists of a series of bytes, each byte representing an operation. In general, code execution is an infinite loop. The program counter is incremented by one (the initial value is zero) and executed until the code completes or encounters an error, STOP or RETURN. The operation can access three kinds of space for storing data:

Stack, a last-in first-out data store, 32-byte values can be pushed onto the stack.

- Memory, an infinitely extendable byte queue.
- Long term storage of contracts, storage of a secret key/value, where the secret key and value are both 32 bytes in size. Unlike the stack and memory that are reset at the end of the calculation, the stored content will be maintained for a long period of time.

The code can access the value, the sender, and the data in the received message just as the block header data is accessed. The code can also return the data's byte queue as output.

The formal execution model of PCM code is surprisingly simple. When the Y2 coin virtual machine is running, its complete calculation state can be defined by the tuple (block_state, transaction, message, code, memory, stack, pc, gas), where block_state is the global state containing all account balances and storage. . Each round of execution, by calling

the first pc (program counter) bytes of the code, the current instruction is found, and each instruction defines how it affects the tuple itself. For example, ADD pops two elements and pushes their sum to the stack, decrements gas (fuel) and adds pc to one, SSTORE pops the top two elements and inserts the second element to the first Each element defines the contract storage location, which also reduces the gas value by up to 200 and increments pc by one. Although there are many ways to optimize Y2 coins through just-in-time compilation, the basic implementation of Y2 coins can be implemented in hundreds of lines of code.

application

In general, there are two applications above Y2. The first category is the application of new energy Y2, which provides renewable energy support for all countries, institutions, and individuals that need crude oil, which can be used for sales, investment, and other related applications. The second type is financial applications, and Y2 can be transformed from resources into finance. The future will surely become a financial + energy complex favored by global financial professionals.

Token system

The token system on the chain has many applications, from subcurrencies representing assets such as US dollars or gold to company stocks, individual tokens representing smart assets, secure unforgeable

coupons, and even no relation to traditional values. A token system for reward points. Implementing a token system in Y2 coins is surprisingly easy. The key point is to understand that all currency or token systems are fundamentally a database with the following operations: subtract X units from A and add X units to B, provided that (1) A There are at least X units before the transaction and (2) the transaction is approved by A. Implementing a token system is to implement such a logic into a contract.

The basic code for implementing a token system with the Serpent language is as follows:

```
from = msg.sender

to = msg.data[0]

value = msg.data[1]

if contract.storage[from] >= value:

contract.storage[from] = contract.storage[from] value

contract.storage[to] = contract.storage[to] + value
```

This is essentially a minimal implementation of the "bank system" state transition function that will be further described in this paper. Need to add some additional code to provide the ability to distribute the currency in the initial and other edge cases, ideally add a function to let other contracts to query the balance of an address. Will suffice. In theory, a token system that acts as a child currency based on Y2 coins may

include an important function lacking on the bitcoin-based chain currency: the ability to use the currency to pay transaction fees directly.

Financial derivatives and currency with stable value

Financial derivatives are the most common application of "smart contracts" and one of the easiest to implement with code. The main challenge in achieving financial contracts is that most of them need to refer to an external price issuer; for example, a very demanding application is a smart contract to hedge Y2 (or other cryptocurrency) against dollar prices. However, the contract needs to know the price of Y2 against the U.S. dollar. The easiest way is through a "data offer" contract maintained by a specific organization (such as Nasdaq) that is designed to enable the agency to update the contract as needed and provide an interface that allows other contracts to send a contract through Message to the contract to get a reply containing price information.

When these key elements are all in place, the hedging contract looks like the following:

Wait for A to enter 1000 Y2 currency. .

Wait for B to enter 1000 Y2 currency.

By querying the data provision contract, the dollar value of 1000 Y2 coins, for example x dollars, is recorded to memory.

After 30 days, A or B is allowed to "reactivate" the contract to send Y2

coins worth $x (requery the data to provide the contract for a new price and calculate) to A and send the remaining Y2 coins to B.

Such contracts have extraordinary potential in cryptography business. One problem with cryptocurrency often criticized is its price volatility; although a large number of users and businesses may need the security and convenience of cryptographic assets, they are less willing to face a 23% decline in assets in one day. The situation of value. Until now, the most common recommended solution was publisher endorsement of assets.

In practice, however, the issuer is not always trustworthy, and in some cases the banking system is too fragile or not honest enough to make such services impossible. Financial derivatives provide an alternative solution. There will no longer be a separate issuer that will provide reserves to support an asset, but instead will be a decentralized market made up of speculators who are willing to raise the price of a cryptographic asset. Unlike issuers, speculators have no right to bargain because hedging contracts freeze their reserves in contracts. Note that this approach is not completely decentralized because there is still a need for a trusted data source that provides price information, although it is still controversial that it is still reducing infrastructure requirements (unlike publishers, a price publisher does not need Licensing and it seems to be classifiable as a free speech) and a huge step forward in

reducing potential fraud risk.

Identity and reputation system

The earliest alternative currency, the domain name coin, tried to use a bitcoin-like blockchain to provide a name registration system where users could register their names with other data in a common database. The most common use case is a domain name system that has a domain name like "bitcoin.org" (or "bitcoin.bit" in domain name currency) and an IP address. Other application examples include email verification systems and potentially more advanced reputation systems. Here is the basic contract for providing a name registration system similar to the domain name currency in Y2 currency:

```
if !contract.storage[tx.data[0]]:

contract.storage[tx.data[0]] = tx.data[1]
```

The contract is very simple; it is a database in a Y2 coin network that can be added but cannot be modified or removed. Anyone can register a name as a value and never change. A more complex name registration contract will contain a "functional clause" that allows other contracts to be queried, as well as a mechanism to allow a name's "owner" (ie, the first registrant) to modify the data or transfer ownership. You can even add reputation and trust network features to it.

Decentralized storage

In the past few years there have been some popular online file storage

startups, most notably Dropbox, which seek to allow users to upload their hard drive backups, provide backup storage services and allow users to access them to charge users monthly. At this point, however, the file storage market is sometimes relatively inefficient; a cursory observation of existing services shows that, in particular, at the level of the Mystic Valley 20-200GB, which has neither free space nor enterprise-level user discounts. The monthly cost of file storage costs means the cost of paying the entire hard disk in one month. The Y2 bill allows the development of a decentralized storage ecosystem so that users can lease their own hard drives or unused network space to get a small amount of revenue, thereby reducing the cost of file storage.

The basic component of such a facility is what we call the "decentralized Dropbox contract." The contract works as follows. First, someone divides the data that needs to be uploaded into chunks, encrypts each piece of data to protect privacy, and builds a Merkel tree. Then create a contract with the following rules. For every N blocks, the contract will extract a random index from the Merkel tree (using the hash of the previous block that can be accessed by the contract code to provide randomness), and then give the first Entity X Y2 coins to support proof of ownership of a block with a similar simplified verification payment (SPV) at a specific index in the tree. When a user wants to download his file again, he can use the micropayment channel protocol (for example, pay 1 Saab per

32kbytes) to recover the file; the most cost-effective method is to pay the person who does not publish the last transaction, but Replace the original transaction after every 32k bytes with a slightly more cost-effective transaction with the same random number.

An important feature of this agreement is that although it seems like one person trusts many random nodes that are not ready to lose files, he can secretly share the files into many small blocks, and then through the monitoring contract that each small block is still Saved by a node. If a contract is still paying, then evidence is provided that someone is still saving the document.

Decentralized Autonomous Organization (DAO)

In the usual sense, the concept of decentralized autonomous organization (DAO) refers to a virtual entity with a certain number of members or shareholders, relying on, for example, a 67% majority to decide to spend money and modify the code. Members will collectively decide how the organization allocates funds. The method of allocating funds may be rewards, wages or more attractive mechanisms such as rewarding work with internal currency. This simply uses cryptographic block chain technology to fundamentally replicate the legal significance of traditional companies or non-profit organizations for enforcement. So far, many discussions surrounding DAO have centered on a “capitalist” model of “decentralized autonomous corporation” with dividend-shared

shareholders and tradable shares; as an alternative, one is described as The "decentralized autonomous community" entity will enable all members to have equal rights in decision-making and will require 67% majority consent when adding or subtracting members. Everyone can only have one membership. This rule needs to be enforced by the group. Here is an outline of how to implement DO using code. The simplest design is a code that can self-modify if two-thirds of the members agree. Although the code is theoretically unchangeable, the code can be easily modified by placing the code skeleton in a separate contract and pointing the contract call's address to a changeable store. There are three transaction types in the simple implementation of such a DAO contract, distinguished by the data provided by the transaction:

- [0,i,K,V] The registration index is i's recommendation to change the content of the storage address index K to v.
- [0,i] Registers the vote for suggestion i.
- [2,i] Confirm recommendation i if there are enough votes.

Then the contract has specific terms for each item. It will maintain a record of all open storage changes and a table of who voted. There is also a table of all members. When two-thirds majority consent is obtained for any change in stored content, a final transaction will execute this change. A more complex framework will add built-in voting functions to send transactions, increase or decrease members, or even

provide voting representatives such as appointed democracy (that is, anyone can entrust another person to vote on their own behalf, and this kind of delegation Relationships can be passed, so if A delegates B then B delegates C then C will decide A's vote). This design will enable DAO to grow organically as a decentralized community, so that people can ultimately hand over the task of selecting suitable candidates to experts, unlike the current system. As community members constantly change their team time, experts will easily emerge. And disappear.

An alternative model is to decentralize the company, where any account can have 0 to more shares, and a decision requires a two-thirds majority of shares to agree. A complete framework will include asset management functions - the ability to submit orders for buying and selling shares and the ability to accept such orders (provided there is an order matching mechanism in the contract). Delegates still exist in the form of appointment democracy, creating the concept of “board”.

More advanced organizational governance mechanisms may be implemented in the future; now a decentralized organization (DO) can start with decentralized autonomous organizations (DAO). The difference between DO and DAO is ambiguous. A general dividing line is whether governance can be achieved through a politically-like process or an "automated" process. A good intuitive test is the "no universal language" standard: if two members are not Does the same language organization

still work? Obviously, a simple traditional holding company will fail, and a bitcoin agreement like this is likely to succeed. Robin Hansen's “futarchy”, a mechanism for organizing governance by predicting the market, is a real A good example of what "autonomous" governance might look like. Note that one does not need to assume that all DAOs are superior to all DOs; autonomy is only a paradigm that has great advantages in some specific scenarios but may not be feasible in other places. Many semi-DAOs may exist.

Further application

1. Saving purse. Suppose Alice wants to make sure her money is safe, but she fears losing or being hacked to steal her private key. She put Y2 into a contract with Bob. As shown below, this contract is a bank:

Alice can withdraw up to 1% of funds on a daily basis.

Bob can extract up to 1% of funds each day, but Alice can use her private key to create a transaction to cancel Bob's withdrawal rights.

Alice and Bob can withdraw funds arbitrarily.

In general, 1% per day is enough for Alice. If Alice wants to withdraw more, she can contact Bob for help. If Alice's private key is stolen, she can immediately find Bob to transfer her funds to a new contract. If she loses her private key, Bob can slowly raise the money. If Bob shows malice, she can turn off his withdrawal rights.

2. Crop insurance. A person can easily create a financial derivative

contract using weather conditions instead of any price index as data entry. If an Iowa farmer buys a financial derivative that reverses payments based on Iowa's rainfall, then if a drought occurs, the farmer will automatically receive payment and if there is enough rainfall he will Very happy because his crop will be very good.

A decentralized data publisher. For differences-based financial contracts, it is in fact possible to decentralize the data publisher by passing the "Xerin point" protocol. The working principle of Xie Lindian is as follows: N-party provides input values to the system for a specified data (for example, Y2/USD price), all values are sorted, and each node that provides a value between 25% and 75% will have To get rewards, everyone has the incentive to provide answers that others will provide. The answer that a large number of players can actually agree on is clearly the correct answer by default. This constructs a theoretically available number of values, including Y2/USD prices, Berlin's temperature Even a de-centralized protocol with the results of a particularly difficult calculation.

4. Multi-signature smart contracts. Bitcoin allows multi-signature-based trading contracts. For example, 5 private keys can be used to collect funds. Y2 coins can be more elaborated. For example, 5 private keys can be used to collect 4 full funds. If only 3 accounts spend 10% of the funds each day, only 2 can only spend 0.5% of the funds each day. . In addition,

the multi-signature in the Y2 coin is asynchronous, meaning that both parties can register the signature on the blockchain at different times, and the last signature is automatically sent after the signature is in place.

5. Cloud computing. PCM technology can also be used to create a verifiable computing environment that allows users to invite others to perform calculations and then selectively request evidence that is calculated correctly at a randomly selected checkpoint. This makes it possible to create a cloud computing market where any user can participate with their desktops, laptops or dedicated servers. On-site inspections and security deposits can be used to ensure that the system is trustworthy (ie no nodes can be deceived Lee). Although such a system may not be suitable for all tasks; for example, tasks that require advanced interprocess communication are not easily accomplished on a large node cloud. However, some other tasks make it easy to implement parallelism; SETI@home, folding@home, and gene algorithms are very easy to implement on such platforms.

6. Point-to-point gambling. Any number of peer-to-peer gambling agreements can be moved to block Y2 coins, such as Frank Stajano and Richard Clayton's Cyberdice. The simplest gambling agreement is in fact such a simple contract that is used to bet on the difference between the hash value and the guess value of the next block, whereby more complex gambling protocols can be created to achieve near-zero cost and No

fraudulent gambling service.

7. Forecast the market. Whether it is a god or a Shilin coin, predicting the market will be easy to achieve. The forecast market with Schering currency may prove to be the first mainstream application of "futarchy" as a decentralized organization management agreement.

8. The chain goes to a centralized market, based on identity and reputation systems.

Miscellaneous and attention

Improved ghost protocol implementation

The "Greedy Heaviest Observed Subtree" (GHOST) protocol was an innovation introduced by Yonatan Sompolinsky and Aviv Zohar in December 2013. The motivation behind the Ghost Protocol is that the current fast-acknowledged blockchain is plagued by low security because of the high block rate of blocks; because blocks need to spend a certain amount of time (set as t) spreading to the entire network, if the rejection rate is high, it will be simple. Because of the higher share of computing power, it is more efficient.

As Sompolinsky and Zohar describe, the ghost protocol solves the first problem of reducing network security by including the waste block when calculating which chain is “longest”; that is, not only a block The parent block and the earlier ancestor block, the ancestral block's voided descendant block (called "tertile block" in Y2 coin terminology) is also

added to calculate which block has the maximum workload to support it. prove. We surpassed the agreement described by Sompolinsky and Zohar to solve the second problem - centralization tendency, Y2 coins pay 87.5% of the rewards for the waste block that contributes to the confirmation of the new block with the status of "unclear block" and include them. The calculated "dumpster block" will receive 12.5% of the reward, but the transaction fee is not awarded to the uncle block.

The Y2 coin implements a simplified version of the ghost protocol that only goes down to the fifth floor. Its characteristic is that the waste block can only be the second block of the parent's second generation to the fifth generation's descendant block as the status of the uncle block, rather than the later generation's block (such as the sixth generation of the parent block). The block, or the third generation descendant block of the grandfather block, is included in the calculation. There are several reasons for this. First, the unconditional ghost protocol will give excess complexity to the calculation of which tertile block of a given block is legitimate.

cost

Because each transaction released to the blockchain takes up the cost of downloading and verification, there is a need for a regulatory mechanism that includes transaction fees to guard against spamming transactions.

However, when given a special, less precise assumption of simplification, the loopholes in this market-based mechanism miraculously eliminated its influence. The argument is as follows. Assumptions:

1. A trade brings k steps to provide rewards kR to anyone who includes the trade, where R is set by the trader, and both k and R are visible (roughly) in advance.

2. The cost per node to process each step is C (ie, the efficiency of all nodes is consistent).

3. There are N nodes, each with the same computing power (ie, 1/N of the total network power).

However, there are several important deviations from these assumptions and the actual situation:

1, because the extra verification time delays the broadcasting of the block and thus increases the chance of the block becoming a waste block, processing the transaction costs more than other verification nodes.

2. The distribution of force in practice may end up being extremely uneven.

3, speculators who destroy the network as their own, political opponents and madmen do exist, and they can intelligently set up contracts so that their costs are much lower than other verification nodes.

The first point above drove the inclusion of fewer transactions, and the second point added NC; therefore the impact of these two points at least

partially offset each other. Points 3 and 4 are the main issues; as a solution we simply built a Floating upper limit: No block can contain more than BLK_LIMIT_FACTOR times the long-term exponential moving average. specifically:

```
blk.oplimit = floor((blk.parent.oplimit * (EMAFACTOR - 1) + floor(pare
nt.opcount * BLK_LIMIT_FACTOR)) /EMA_FACTOR)

BLK_LIMIT_FACTOR and EMA_FACTOR are temporarily set to 65536 and 1.5 c
onstants, but may be adjusted after further analysis.
```

Computation and Turing Complete

The JUMP instruction allows the program to jump back somewhere in the code, and JUMPI instructions that allow conditional statements such as while x < 27: x = x * 2 implement conditional jumps. Second, the contract can call other contracts, with the potential to achieve a loop through recursion. This naturally leads to a problem: Can a malicious user be forced to shut down by forcing the entire node to enter an infinite loop? This problem arises because of a problem in computer science called an outage problem: there is no way to know in a general sense whether a given program can finish running in a limited amount of time.

As described in the section on state transitions, our solution solves the problem by setting the maximum number of calculations to run for each transaction. If it is exceeded, the calculation is returned to the original state but the fee still remains. The news works in the same way.

An attacker sees a contract containing a contract such as send(A,contract.storage[A]); contract.storage[A] = 0 and then sends it with enough to perform the first step but not enough to perform the second step. The transaction (that is, withdrawal but does not reduce the account balance). The contract author does not need to worry about defending a similar attack because all changes are reverted if execution stops halfway.

A financial contract works by extracting the median of nine private data publishers to minimize risk. An attacker takes over one of the data providers and then designs this variable address invocation mechanism as described in the DAO section to be changeable. The data provider turned to run an infinite loop in an attempt to persuade any attempt to request funds from this financial contract will be suspended due to the depletion of fuel. However, the financial contract can set fuel limits in the message to prevent such problems.

The complete replacement for Turing is Turing incomplete, where the JUMP and JUMPI instructions do not exist and only one copy of each contract is allowed to exist within the call stack at a given time. In such a system, the above-mentioned cost system and the uncertainty of the efficiency of the solution around us may not be needed, because the cost of executing a contract will be determined by its size. In addition, Turing is incomplete or even not a big restriction. In all the contract

examples we envisioned so far, only one needs to be cycled, and even this cycle can be replaced by the repetition of 26 single-line code segments. Considering the serious trouble and limited benefits brought by Turing's completeness, why not simply use a Turing incomplete language? The fact that Turing is incomplete is far from a concise solution. why? Please consider the following contract:

```
C0: call(C1); call(C1);

C1: call(C2); call(C2);

C2: call(C3); call(C3);

…

C49: call(C50); call(C50);

C50: (run one step of a program and record the change in storage)
```

Now, send a transaction like this to A. Thus, in the 51 transactions, we have a contract that requires 250 calculations. It may try to maintain a maximum executable number of steps for each contract and call other contracts for recursion. The contract calculation may execute steps to detect such logic bombs in advance, but this will make it prohibit the creation of contracts for other contracts (because the creation and execution of the above 26 contracts can be easily put into a single contract). Another problem is that the address field of a message is a variable, so in general it may not even be possible to know in advance which of the other contracts a contract will call. Thus, we finally have a

surprising conclusion: Turing's complete management is amazingly easy, and in the absence of the same control, Turing's incomplete management is surprisingly difficult - then why not make the protocol Turing complete?

Currency and issuance

Background introduction:

The League of Arab States is a regional international organization established to strengthen cooperation and cooperation among Arab countries. Abbreviation Arab League or Arab League. In March 1945, representatives of Egypt, Iraq, Jordan, Lebanon, Saudi Arabia, Syria, and seven Arab countries in Yemen held a meeting in Cairo, adopted the "Arab League Treaty," and declared the establishment of the alliance. By 1993 there were 22 member states. The purpose is to strengthen close cooperation among member states, safeguard the independence and sovereignty of Arab countries, and coordinate their activities. In mid-November 2011, the League of Arab States suspended membership of Syria; on November 27th of the same year, the Arab League decided to impose economic sanctions on Syria immediately following the meeting of foreign ministers in Cairo, Egypt. On June 5, 2017, the Arab League headed by Saudi Arabia issued a statement announcing the exclusion of Qatar from the organization. Objectives: Close cooperation between member states, coordinate political activities among each other,

defend the independence and sovereignty of Arab countries, promote the overall interests of Arab countries, and promote the economic, financial, transportation, cultural, health, social welfare, Nationality, passport, visa, judicial, etc. Member states respect each other's political systems, and disputes between them must not be resorted to by force. The treaties and agreements concluded between member states and other countries are not binding on other countries.

There are currently 22 Arab League members: Algeria, UAE, Oman, Egypt, Palestine, Bahrain, Djibouti, Kuwait, Lebanon, Libya, Mauritania, Morocco, Saudi Arabia, Sudan, Somalia, Tunisia, Syria, Yemen, Iraq, Jordan, Ko Morrow. On November 16, 2011, the Arab League formally terminated its membership in Syria. On March 26, 2013, the Arab League decided to grant Syria the "National League" for the Syrian opposition in the Arab League seat, but it has not yet been implemented.

The distribution model is as follows:

- Through the offering, the Y2 currency will be offered at a price of 1 Y2 each, which is based on the actual price. A mechanism that aims to fund the Y2 currency research and development of Y2 new energy fuel additives and pay for developers has already been used in other cryptography. Successful use on the currency platform. Early buyers will enjoy larger discounts. The BTC and ETH (changes in net prices) from the sale will be fully used to pay developers and researchers for wages and

rewards, projects in the cryptocurrency monetary system, and Y2 new energy fuel additives. Global development, circulation, and occupancy. A total of 270 million pieces were distributed for the first time in the world and 70 million pieces were issued for the second time in the world.

First-issued country, quantity and proportion:

country	Quantity	Accounting
USA	3000000	3.75%
Korea	9000000	11.25%
China	7200000	9%
Russia	5000000	6.25%
England	8200000	10.25%
EU	3200000	4%
Japan	9000000	11.25%
France	2500000	3.125%
Saudi Arabia	5500000	6.875%
Australia	2500000	3.125%
India	5500000	6.875%
Italy	2000000	2.5%
Indonesia	2000000	2.5%
Canada	2000000	2.5%

Germany	1500000	1.875%
South Africa	1000000	1.25%
Mexico	1000000	1.25%
Turkey	1000000	1.25%
Brazil	900000	1.125%
Argentina	500000	0.625%

Arab League Secretary-General Ahmed Aboul Gheit called on everyone to join this organization that will save the world;

The Y2 New Energy Fuel Laboratory (formerly the Arab League Special Laboratory) was established in 2003 and renamed the Y2 New Energy Laboratory in 2017. It provided over 1,000 technical support for the Arab League and countries in the Middle East in 15 years.), Y2 is committed to the study of the use of new energy development, aimed at serving the Arab League member states, share the worry for the Union, Y2 in the Arab League belongs to the State Department directly under the deputy secretary-general unit.

The launch of Y2 will increase the world's energy by more than 20 times. By 2020, it will save energy use by 10-25 times. In 2030, the use of energy will be saved 170 times or more.

April 1st, 2018 - June 6th is the start of global events.

Y2 coins will be issued globally on April 7th. The Y2 exchanges are: 7 exchanges including OKEX, BitMEX, Binance, GDAX, K-net, B-net, HitBTC,

YOKI, Bit-Z, and P-net. It is not less than 6 times the cooperation price.

Issuing unit: Y2 New Energy Fuel Laboratory (Y2 الـ جديـ دة الـ طاقـ ة مخـ تـ بر لـ لوقـ ود).

Investors: Mohammad bin Salman Al Saud (Saudi Arabian), Mukesh Ambani (the richest man in India), the Quantum Fund (George Soros), and the UAE Central Bank.

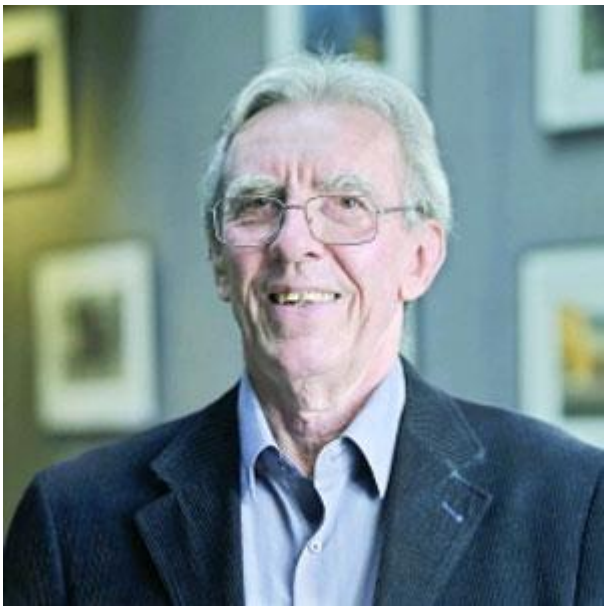
Awarded: Digital Currency Trading Permits, International Trade Circulation Permits, League of Arab League Summit Currency Committee Members, etc.

Main staff:

Y2 Lab Chief Scientist, CEO of Y2 Currency, 2011 Nobel Prize in Chemistry, Co-Chairman of the Arabic Digital Currency Foundation: Shechtman (شـ يخـ تمان)

The heir to the Crown Prince of Saudi Arabia, son of King Salman, chief strategy officer of Y2 currency: Mohammad bin Salman Al Saud (محمد بن سلمان بن سعود آل سعود)

Y2 Lab Technical Consultant, PayTabsCEO, Chief Technology Officer of Y2 Currency: Abdulaziz Al Jouf (عبد العزيز الجوف)



Y2 Lab Scientist, Nobel Prize in Chemistry 2016, Chief Operating Officer of Y2 Currency PIERRE SOVIC (مالك سوفير)

Only support: BTC, ETH (change in price)

Example: 1 BTC = 46,000 yuan, that is 1 BTC =133.33333333 Y2

Example: 1ETH = 2500 yuan, that is 1 ETH = 7.24763681 Y2

- 0.099x (x is the total amount sold) will be allocated to BTC, ETH (real-price movements) and cash contributors or other deterministic financings to participate in the early contributors before the successful development, and another 0.099x will be allocated to long-term research projects .

Release decomposition

The permanent linear growth model reduces the risk of excessive concentration of wealth in Bitcoin and gives people living in the present and future a fair chance to obtain currency, while maintaining incentives to acquire and hold Y2 coins because of long-term Look at the "money

supply growth rate" is tending to zero. We also infer that, with the passage of time, coins will always be lost due to carelessness and death. If the loss of a coin is a fixed proportion of the annual money supply, the money supply in the final total circulation will be stable. In a value equal to the annual money circulation divided by the loss rate (for example, when the loss rate is 1%, when the supply reaches 30x, 0.3x is dug out each time and 0.3x is lost each year, reaching an equilibrium).

In addition to the linear issuance method, the growth rate of supply of Bitcoin-like Y2 coins tends to be zero in the long-term.

Extensibility

Scalability issues are a common concern for Y2 coins. Like Bitcoin, Y2 coins also suffer from the fact that each transaction requires every node in the network to deal with this dilemma. Bitcoin's current blockchain size is about 20GB, which grows at a rate of 1MB per hour. If the bitcoin network processes Visa-class 2000tps transactions, it will grow at 1MB every three seconds (1GB per hour, 8TB per year). Y2 coins may also experience similar or even worse growth patterns, because there are many applications on the blockchain of Y2 coins, rather than Bitcoin as a simple currency, but Y2 coins only need to store state instead of The fact of a complete blockchain history has improved the situation.

The problem with large blockchains is the risk of centralization. If the blockchain size increases to, say, 100TB, the likely scenario would be that

only a very small number of large merchants will run full nodes, while regular users use light SPV nodes. This raises concerns about the risk of fraud at the full node partnership (for example, changing block rewards, giving themselves BTC). Light nodes will have no way to detect this fraud immediately. Of course, there may be at least one honest full node, and fraudulent information leaked through channels like Reddit a few hours later, but it's too late to let the average user make any effort to repeal the blocks that have already been generated. They will all encounter huge infeasible coordination issues of the same scale as the launch of a successful 51% attack. In Bitcoin, this is a problem now, but a change suggested by Peter Todd can alleviate this problem.

Recently, Y2 coins will use two additional strategies to address this issue. A certain number of full nodes are guaranteed. Second, and more importantly, after processing each transaction, we will include the root of an intermediate state tree in the blockchain. Even if the block verification is centralized, as long as there is an honest verification node, the centralized problem can be avoided by a verification protocol. If an incorrect block is issued, the block is either in the wrong format or the state S[n] is wrong. Since S[0] is correct, there must be the first error state S[i] but S[i-1] is correct, the validation node will provide index i, along with processing APPLY (S[i- 1], TX[i]) -> S[i] A subset of Patricia tree nodes required. These nodes will be mandated to perform this part of

the calculation to see if the resulting S[i] is consistent with the previously provided values.

In addition, it is more complicated to maliciously release an incomplete block to attack, resulting in insufficient information to determine whether the block is correct. The solution is a challenge-response protocol: the verification node challenges the target transaction index, and the light node receiving the challenge information untrusts the corresponding block until another or the verifier provides a subset of Patricia nodes as correct. evidence of.

Review: Decentralized Applications

The above contract mechanism enables any one person to establish a command line application (fundamentally speaking) through a global network consensus on a virtual machine that can change an entire network-accessible state as its "hard disk." However, for most people, the lack of adequate user-friendliness of the command line interface used as a transaction delivery mechanism makes decentralization an attractive alternative. Finally, a complete "decentralized application" should include the underlying business logic components [whether or not Y2 coins are fully implemented, using Y2 coins and other system combinations (such as a P2P message layer, one of which is planned to put Y2 coin customers End-of-the-box or other system-level] graphical user interface components. The Y2 coin client is designed as a web

browser, but includes support for "PC" Javascript API objects that can be used by specific web pages seen by the client to interact with the Y2 coin blockchain. From the perspective of "traditional" web pages, these web pages are completely static content because blockchain and other decentralized protocols will completely replace the server to handle user-initiated requests. Finally, the decentralized protocol promises to use some form of Y2 coin to store web pages.

in conclusion

The Y2 bill agreement was originally conceived as an upgraded version of cryptocurrency that provides highly advanced functions such as chain contracts, withdrawal restrictions, and financial contracts, gambling markets, etc., through a highly common language. However, what is more interesting about Y2 coins is that the Y2 coin protocol goes further than pure currency, centering around decentralized storage, decentralized computing and decentralized forecasting markets, and dozens of similar concepts to establish agreements and decentralization Applications have the potential to fundamentally improve the efficiency of the computing industry, and provide strong support for other P2P protocols by adding economic layers for the first time. Finally, there will also be a large number of applications that have nothing to do with money.

The concept of arbitrary state transitions implemented by the Y2 coin

protocol provides a platform with unique potential; unlike closed-end agreements designed for single purposes such as data storage, gambling or finance, Y2 coins are open-ended in design and We believe that it is extremely suitable as a basic layer to serve the extremely large number of financial and non-financial agreements that will emerge in the coming years.

Comments and Advanced Reading

annotation

1. An experienced reader will notice that the bitcoin address is in fact a hash of the public key of the elliptic curve, not the public key itself, but in fact it is perfectly reasonable to refer to the public key hash as a public key from a cryptographic point of view. This is because Bitcoin cryptography can be considered as a custom digital signature algorithm. The public key is composed of the hash of the elliptic curve public key. The signature is composed of the elliptic curve public key connected by the elliptic curve signature. The verification algorithm includes the use of public key. The key provides the elliptic curve public key hash to check the elliptic curve public key, and then uses the elliptic curve public key to verify the elliptic curve signature.

2. Technically speaking, the median of the first 11 blocks.

3. Internally, 2 and "CHARLIE" are numbers, the latter has a huge

base256 encoding format, the number can be from 0 to 2^256-1.