BTLink Public Chain White Paper

V1.0

BTLink is the running BTL public chain of the web3.0 blockchain Internet protocol, including the use of on-chain digital assets to represent customized currencies and financial instruments (colored coins), ownership of certain basic physical devices (smart assets), such as domain names There are no fungible assets (namecoins) and decentralized exchanges such as DEXswap, NFTswap, decentralized financial derivatives, game finance2.0-3.0 on-chain games, social finance decentralized social finance, read mining Mining, metaverse metaverse, Loan finance, Insurance finance, blockchain mall, blockchain mall, machine gun pool, peer-to-peer gaming and on-chain identity and reputation system, open interface global business docking, etc. More advanced various commercial applications.

Another important area of BTLink is "smart contracts" - systems that automatically transfer digital assets according to pre-arbitrary 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 withdraw up to Y coins per day, A and B can withdraw at will, and A can stop B's withdrawal rights". A logical extension of such contracts are Decentralized Autonomous Organizations (DAOs) - smart contracts that permanently contain an organization's assets and encode the organization's rules. The goal of BTLink is to provide a blockchain with a built-in mature Turing-complete language, in which contracts can be created to encode arbitrary state transition functions, and users only need to implement logic in a few lines of code. Such as programmable decentralized finance, decentralized lending protocols, decentralized insurance protocols, decentralized social protocols, decentralized on-chain games, decentralized exchanges, decentralized trade agreements , decentralized contract agreement, decentralized financial management system, decentralized salary performance agreement, decentralized various DAPP applications, can create all the systems mentioned above, and many more that we can’t even imagine of other systems that can be invented, innovated, created, and programmable.

Table of contents

Background of the project
The development background of blockchain
History
Bitcoin as a state transition system
Subvert the market value mining algorithm of all block BTL public chain protocols
Merkle tree
Alternative blockchain applications
Script
BTLink
Basic functions
BTLink public chain function
BTLink account
Messages and transactions
BTLink state transition function
Code execution
Blockchain and mining
Apply
Token system
Financial derivatives

Identity and reputation systems
Decentralized file storage
Decentralized Autonomous Organizations
Further applications
BTLink public chain blockchain mall
BTLink public chain blockchain encrypted storage
Virtual ecological scene
BTLink blockchain game ecosystem
BTLink decentralized exchange
BTLink encrypted social finance

Miscellaneous and Concerns
Improved Ghost Protocol implementation
Fees
Computational and Turing complete
Currency and issuance
Centralization of mining
Extensibility
Roundup: Decentralized Applications

in conclusion

Development Background

The concept of decentralized digital currency, like alternative applications like property registration, was proposed decades ago. The anonymous electronic cash protocols of the 1980s and 1990s were largely based on Chaumian blinding. These electronic cash protocols offer currencies with a high degree of privacy, but none of these protocols have caught on because they all rely on a centralized intermediary. In 1998, Wei Dai's b-money first introduced the idea of creating money by solving computational problems and decentralized consensus, but the proposal did not give a specific method on how to achieve decentralized consensus. In 2005, Hal Finney introduced the concept of "reusable proofs of work", which uses both the idea of b-money and the computationally difficult Hashcash proposed by Adam Back.) puzzle to create cryptocurrency. However, this concept again gets lost in idealization as it relies on trusted computing as a backend.

(Byzantine-fault-tolerant) research on multi-party consensus systems has been going on for many years, but the above protocol only solves half of the problem. These protocols assume that all participants of the system are known and produce a security boundary of the form "If N parties participate in the system, the system can tolerate N/4 malicious participants". The problem with this assumption, however, is that in the case of anonymity, the security perimeter set by the system is vulnerable to sybil attacks, because an attacker can create thousands of nodes on a single server or botnet, thereby unilaterally ensuring a majority share.

Satoshi Nakamoto's innovation was to introduce the idea of combining a very simple

node-based decentralized consensus protocol with a proof-of-work mechanism. Nodes gain the right to participate in the system through a proof-of-work mechanism, packing transactions into "blocks" every ten minutes, creating an ever-growing blockchain. Nodes with a lot of computing power have more influence, but getting more computing power than the entire network is much harder than creating a million nodes. Although the Bitcoin blockchain model is very rudimentary, it has been proved to be good enough in practice. In the next five years, it will become the cornerstone of more than 200 currencies and protocols around the world.

On January 3, 2009, Satoshi Nakamoto wrote on the Bitcoin genesis block: "The British Chancellor of the Exchequer has started the second round of monetary aid. This credit currency system with unlimited printing of banknotes reflects world inflation. , according to the continuous increase in prices of various living materials and various means of production, the devaluation of currency is like paper, and the people at the bottom of the world will always live in a world of high inflation. And Bitcoin is an open source constant of 21 million pieces, each transaction, The transaction is signed by the private key controlled by the trader, without centralized transaction authorization, and each transaction is confirmed by the whole network broadcast of the nodes participating in the bookkeeping. This is a great experiment and revolution in the history of human finance, currency, currency and assets The issuance and transaction of the coin are issued by each person using their own currency, and they control their own currency and assets through private keys. Each participant is the "governor of the Bitcoin central bank"; every Bitcoin transaction records The node of the account is the issuer of Bitcoin. Therefore, Bitcoin is the currency of the people of the world, and it does not belong to any centralized economy. Companies and individuals include Satoshi Nakamoto!

When Satoshi Nakamoto created the Bitcoin blockchain in 2009, he simultaneously introduced two revolutionary new concepts to the world that had not been tested. 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 worldwide attention, recognition and participation from countries, Fortune 500 companies, insurance giants, banks, consortia, and family offices. Human society has entered web2.0 from web1.0, and is now fully entering the Internet era of web3.0; from the centralized Internet of 1.0 and 2.0, it has fully entered the era of decentralized Internet 3.0! Gold, this precious metal, has been around since ancient times. Today, no matter how dynasties change, it is always the hard currency of human society, which can be exchanged for material and commodities. Because of its stable performance, scarcity and increasing mining costs, gold is the banknote among banknotes. Bitcoin is the digital gold of mankind in the next 100-1000 years and the Internet era of mankind. It can become the anchor of the belief currency of the centralized economy, and it can also become the support for various securities and bills of the world's enterprises, and it can also be used with gold. Exchange any substance in the world. Bitcoin is only 13 years old, just like a child who has just entered youth. The next 10 years will be 10 years for Bitcoin and blockchain to grow, 10 years for application, and 10 years for legislation. Politically, it is a currency without a central bank and has wild price swings.

However, Satoshi Nakamoto's great experiment had an equally important part as Bitcoin: the concept of a blockchain based on proof-of-work that allows people to agree on the order of transactions. Bitcoin as an application can be described as a first-to-file system: if someone has 1 BTC and sends that 1 BTC to both A and B, only the transaction that is confirmed first will take

effect. There is no inherent way to decide which of two transactions comes first, a problem that has hindered the development of decentralized digital currencies for many years. Satoshi Nakamoto's blockchain is the first reliable decentralized solution. Now, the attention of developers around the world is rapidly turning to the second part of Bitcoin technology, how the blockchain can be applied to the web 3.0 decentralized blockchain Internet beyond money.
Bitcoin as a state transition system

Technically, the Bitcoin ledger can be thought of as a state transition system that includes all existing Bitcoin ownership states and "state transition functions". The state transition function takes the current state and transaction as input and outputs a new state. For example, in a standard banking system, the state is a balance sheet, a request to transfer $X from account A to account B is a transaction, and the state transition function will subtract $X from account A and add to account B X dollars. If the balance of account A is less than X dollars, the state transition function will return an error message. So we can define the state transition function as follows:
<code style="font-family:Menlo, Courier, monospace, monospace, sans-serif;font-size:13.6px;margin:0px;border:none;background-color:transparent;white-space:pre-wrap; " class="hljs">APPLY(S,TX) > S' or ERROR</code>
In the banking system mentioned above, the state transition function is as follows:
<code style="font-family:Menlo, Courier, monospace, monospace, sans-serif;font-size:13.6px;margin:0px;border:none;background-color:transparent;white-space:pre-wrap; " class="hljs">APPLY({ Alice: $50, Bob: $50 },"send $20 from Alice to Bob") = { Alice: $30,Bob: $70 }</code>
but:
<code style="font-family:Menlo, Courier, monospace, monospace, sans-serif;font-size:13.6px;margin:0px;border:none;background-color:transparent;white-space:pre-wrap; " class="hljs">APPLY({ Alice: $50, Bob: $50 },"send $70 from Alice to Bob") = ERROR</code>
The "state" of the Bitcoin system is the set of all unspent bitcoins that have been mined (technically called "unspent transaction outputs, or UTXOs"). Each UTXO has a face value and owner (defined by a 20-byte address that is essentially a cryptographic public key [1]). A transaction consists of 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), return an error message
- If the signature is inconsistent with the UTXO owner's signature, return an error message

If the sum of all UTXO input denominations is less than the sum of all UTXO output denominations, return an error message
2.
Return to a new state S', in which all input UTXOs are removed and all output UTXOs are added.
3.

The first part of the first step prevents the sender of the transaction from spending non-existent bitcoins, and the second part prevents the sender of the transaction from spending someone else's bitcoin. The second step ensures the conservation of value. The Bitcoin payment protocol is as follows. Suppose Alice wants to send 11.7BTC to Bob. In fact, it is impossible for Alice to have exactly 11.7 BTC. Suppose, the minimum amount of bitcoins she can get is: 6+4+2=12. So, she can create a transaction with 3 inputs and 2 outputs. The face value of the first output is 11.7BTC, and the owner is Bob (Bob's bitcoin address). The face value of the second output is 0.3BTC, and the owner is Alice herself, that is, change.
mining

The state transition system can be easily implemented if we have a trusted centralized service organization, which can simply encode the above functions accurately. However, we want to build the Bitcoin system as a decentralized monetary system, and in order to ensure that everyone agrees on the order of transactions, we need to combine the state transition system with a consensus system. Bitcoin's decentralization, the consensus process requires nodes in the network to constantly try to package transactions into "blocks". The network is designed to produce a block approximately every ten minutes, and each block contains a timestamp, a nonce, a reference to the previous block (i.e. a hash), and all transactions that have occurred since the previous block was generated list. This creates a growing blockchain over time that is constantly updated to represent the latest state of the Bitcoin ledger.
Following 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 timestamp of the block is later than the timestamp of the previous block and earlier than 2 hours in the future [2].
3.
Check that the block's proof-of-work is valid.
4.

Assign the final state of the previous block to S[0].
5.

Suppose TX is the transaction list of the block, containing n transactions. For all i belonging to 0...n-1, perform state transition S[i+1] = APPLY(S[i], TX[i]). If any transaction i fails in a state transition, exit the program with an error.
6.

Returns correct, state S[n] is the final state of this block.
7.
Essentially, each transaction in the block must provide a correct state transition, it is important to note that the "state" is not encoded into the block. It is purely an abstract concept that is remembered by check nodes. For any block, starting from the genesis state, adding each transaction of each block in order, the current state can be (properly) calculated. Also, pay

attention to the order in which miners include transactions into blocks. If there are two transactions A and B in a block, B spends the UTXO created by A. If A is before B, the block is valid, otherwise, the block is invalid.

The interesting part of the block verification algorithm is the "Proof of Work" concept: each block is hashed with SHA256, and the resulting hash is treated as a 256-bit value that must be less than a constantly dynamically adjusted target value , the target value at the time of this writing is about 2^190. The purpose of proof-of-work is to make the creation of blocks difficult, thereby preventing sybil attackers 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 keep trial-and-error, increasing the value of the nonce to see if the new hash value is less than the target value. If the current target value is 2^192, it means that on average, it takes 2^64 attempts to generate a valid block. Generally speaking, the Bitcoin network resets the target value every 2016 blocks, guaranteeing that a block is generated on average every ten minutes. In order to reward miners for their computational work, each miner who successfully generates a block has the right to include a transaction in the block that sends them 25 BTC out of thin air. In addition, if the input of the transaction is greater than the output, the difference is paid to the miner as a "transaction fee". By the way, the reward for miners is the only mechanism for bitcoin issuance, there is no bitcoin in the genesis state.

To better understand the purpose of mining, let’s analyze what happens when there is a malicious attacker on the Bitcoin network. Because the cryptographic foundation of Bitcoin is very secure, attackers will choose to attack something that is not directly protected by cryptography: transaction order. The attacker's strategy is very simple:

1.

Send 1BTC to sellers to buy items (especially electronic items that don't need to be mailed).

2.

Wait until the item ships.

3.

Create another transaction to send the same 1BTC to your own account.

4.

Convince the Bitcoin network that the transaction sent to its own account was sent first.

5.

Once step (1) happens, miners will pack this transaction into a block, let's say block 270,000, in a few minutes. In about an hour, there will be five blocks following this block, each of which indirectly points to the transaction, thus confirming the transaction. At this time, the seller receives the payment and ships the goods to the buyer. Since we assume this is a digital item, the attacker can receive it instantly. Now, the attacker creates another transaction that sends the same 100BTC to his own account. If the attacker just broadcasts the news to the entire network, the transaction will not be processed. The miner will run the state transition function APPLY(S,TX) and find that this transaction will spend UTXOs that are no longer in the state. So, the attacker would fork the blockchain, regenerate the 270,000th block with the 269999th block as the parent block, and replace the old transaction with the new transaction in this block. Because the block data is different, this requires a new proof-of-work. In addition, because the new 270000th block generated by the attacker has a different hash, the original blocks 270001 to 270005 do not point to it, so the original blockchain and the attacker's new block are completely Detached. When a

blockchain fork occurs, the long branch of the blockchain is considered to be an honest blockchain, and the legitimate miners will mine along the original 270005th block, and only the attacker will mine the new block. Mining after block 270000. In order for the attacker to make his blockchain the longest, he needs to have more computing power than the entire network other than him to catch up (i.e. 51% attack).

Merkle tree

Left: Just providing a few nodes on the Merkle tree is enough to give a legal proof of the branch. Right: Any attempt to make changes to any part of the Merkle tree will eventually lead to an inconsistency somewhere on the chain.

An important scalable feature of the Bitcoin system is that its blocks are stored in a multi-level data structure. The hash of a block is actually just the hash of the block header, which is the length of the root hash of the Merkle tree that contains the timestamp, nonce, last block hash, and the Merkle tree that stores all block transactions. A piece of data about 200 bytes.

A Merkle tree is a binary tree consisting of a set of leaf nodes, a set of intermediate nodes and a root node. The bottommost set of leaf nodes contains basic data, each intermediate node is the hash of its two children, and the root node is also the hash of its two children, representing the top of the Merkle tree. The purpose of a Merkle tree is to allow block data to be transmitted piecemeal: nodes can download block headers from one source and the rest of the tree related to them from another source, and still be able to confirm that all the data is correct . The reason for this is because the hash spreads upward: if a malicious user tries to add a fake transaction in the lower part of the tree, the resulting change will result in a change in the upper node of the tree,and changes to upper nodes, which eventually lead to changes to the root node and changes to the block hash, so that the protocol records it as a completely different block (almost certainly with an incorrect proof-of-work) .

The Merkle tree protocol is arguably crucial to Bitcoin's long-term sustainability. In April 2014, a full node on the Bitcoin network — the node that stores and processes the entire data for all blocks — took up 15GB of memory and was growing at more than 1GB a month. At present, this storage space is acceptable for desktop computers, but mobile phones can no longer carry such a huge amount of data. In the future, only commercial institutions and enthusiasts will act as full nodes. The Simplified Payment Verification (SPV) protocol allows another kind of node to exist, such a node is called a "light node" that downloads block headers, confirms the proof-of-work using the block header, and then downloads only the "branch of the Merkle tree" relevant to its transaction ". This allows light nodes to securely determine the status of any Bitcoin transaction and the current balance of an account by downloading only a small portion of the entire blockchain.

Other blockchain applications

The idea of applying the ideas of blockchain to other fields has been around for a long time. In 2005, Nick Szabo came up with the concept of "Timing Property with Ownership", describing how the development of replicated database technology enables blockchain-based systems to be applied to register land titles, creating properties including, for example, property rights, illegal

misappropriation and a detailed framework for concepts such as Georgia land tax. However, unfortunately there were no practical replicated database systems at that time, so this protocol was not put into practice. However, since the successful development of the decentralized consensus of the Bitcoin system in 2009, many other applications of the blockchain have begun to emerge rapidly.

namecoin - Created in 2010, it is known as a decentralized name registration database. Decentralized protocols like Tor, Bitcoin, and BitMessage require some way of confirming accounts so others can interact with users. However, the only identities available in all existing solutions are pseudorandom hashes like 1LW79wp5ZBqaHW1jL5TciBCrhQYtHagUWy. Ideally, people would like to have an account with a name like "george". The problem, however, is that if someone can create a "george" account, others can also create a "george" account to impersonate. The only solution is the first-to-file principle, only the first registrant can successfully register, the second cannot register the same account again. This problem can take advantage of Bitcoin’ s consensus protocol. Namecoin is the earliest and most successful system to implement a name registration system using the blockchain.

Colored coins - The purpose of colored coins is to provide services for people to create their own digital currency on the Bitcoin blockchain, or, more importantly, money in general - digital tokens. According to the Colored Coin Protocol, people can issue new coins by assigning a color to a particular Bitcoin UTXO. The protocol recursively defines other UTXOs as the same color as the transaction input UTXO. This allows users to keep UTXOs that only contain a certain color, and send these UTXOs just like sending ordinary bitcoins, by backtracking the entire blockchain to determine the color of the UTXOs received.

Metacoins - The idea of Metacoins is to create a new protocol on the Bitcoin blockchain, using Bitcoin transactions to save Metacoin transactions, but with a different state transition function APPLY'. Since the metacoin protocol cannot prevent invalid metacoin transactions on the Bitcoin blockchain, a rule is added if APPLY'(S,TX) returns an error, the protocol will default APPLY'(S,TX) = S. This provides a simple solution for creating arbitrary, advanced cryptographic currency protocols that cannot be implemented in the Bitcoin system, with very low development costs, since the mining and networking issues are already handled by the Bitcoin protocol.

So, in general, there are two ways to build a consensus protocol: building an independent network and building a protocol on the Bitcoin network. While applications like Namecoin have had success using the first approach, implementing this approach is very difficult as each application requires creating a separate blockchain and building and testing all state transitions and network code. In addition, we predict that the application of decentralized consensus technology will obey a power-law distribution, and most applications are too small to guarantee the security of a free blockchain. We also note that a large number of decentralized applications, especially decentralized Autonomous organization, which requires interaction between applications.

On the other hand, the Bitcoin-based approach suffers from the disadvantage that it does not inherit Bitcoin’ s properties of simplified confirmed payments (SPV). Bitcoin enables simplified confirmation of payments because Bitcoin can use blockchain depth as a proxy for validity confirmation. At some point, once the ancestors of a transaction are far enough away from the present, they can be considered part of a legitimate state. In contrast, a metacoin protocol based

on the Bitcoin blockchain cannot force the blockchain to exclude transactions that do not conform to the metacoin protocol. Therefore, the simplified payment confirmation of the secure metacoin protocol requires scanning all blocks backwards, up to the initial point of the blockchain, to confirm whether a transaction is valid. Currently, all "light" implementations of Bitcoin-based metacoin protocols rely on trusted servers to provide data, which is a rather suboptimal outcome for a cryptocurrency whose primary purpose is to eliminate the need for trust.

script

Even without extending the Bitcoin protocol, it enables "smart contracts" to a certain extent. Bitcoin's UTXO can be owned by more than one public key, but also by more complex scripts written in stack-based programming languages. In this mode, to spend such a UTXO, data must be provided to satisfy the script. In fact, the basic public key ownership mechanism is also implemented through a script: the script takes the elliptic curve signature as input, verifies the transaction and the address that owns this UTXO, and returns 1 if the verification is successful, and 0 otherwise. More complex scripts are used for other different application situations. For example, one can create scripts (multi-signatures) that require the collection of two of the three private keys for transaction confirmation, which is very useful for corporate accounts, savings accounts, and some commercial agents. Scripts can also be used to send rewards to users for solving computational problems. One can even create scripts like "if you can provide me a simplified confirmation of payment that you have sent me a certain amount of Dogecoin, this Bitcoin UTXO is yours", essentially the Bitcoin system allows for different ciphers Learn currency for decentralized exchange.

However, the scripting language of the Bitcoin system has some limitations:

Lack of Turing completeness - that is, although the Bitcoin scripting language can support many kinds of computations, it cannot support all computations. The main missing piece is the loop statement. The purpose of not supporting loop statements is to avoid infinite loops when transactions are confirmed. In theory, this is a surmountable obstacle for script programmers, since any loop can be simulated by repeating the if statement multiple times, but doing so would lead to inefficiencies in script space utilization, e.g. implementing a An alternative elliptic curve signature algorithm would likely require 256 repeated multiplications, each requiring separate encoding.

Value-blindness. UTXO scripts do not provide fine-grained control over the withdrawal amount of an account. For example, a powerful application of an oracle contract is a hedging contract. A and B each send $1,000 worth of bitcoins to the hedge contract. After 30 days, the script sends $1,000 worth of bitcoins to A and $1,000 to B. Send the remaining bitcoins. While implementing a hedging contract requires an oracle to determine how much a bitcoin is worth in dollars, this mechanism has been a huge step forward in reducing trust and infrastructure compared to today's fully centralized solutions. However, because UTXOs are indivisible, the only way to implement this contract is to take many UTXOs with different denominations very inefficiently (e.g. for every k up to 30, there is a 2^k UTXO) and Make the oracle pick out the correct UTXO to send to A and B.

Lack of state - UTXOs can only be spent or unspent state, which leaves no room for multi-phase contracts or scripts that require any other internal state. This makes it very difficult to implement multi-phase options contracts, decentralized exchange offers, or two-phase cryptographic commitment agreements (necessary to secure computational rewards). This also

means that UTXO can only be used to build simple, one-off contracts, rather than contracts with more complex states such as decentralized organizations, making meta-protocols difficult to implement. Binary state combined with value blindness means that another important application - withdrawal limits - is impossible to achieve.

Blockchain-blindness - UTXO cannot see the data of the blockchain, such as nonce and hash of the previous block. This flaw deprives the scripting language of its potential value based on randomness, severely limiting applications in other fields such as gaming.

We've looked at three ways to build advanced applications on cryptocurrency: building a new blockchain, using scripts on the Bitcoin blockchain, and building the Metacoin protocol on the Bitcoin blockchain. Approaches to building new blockchains are free to implement arbitrary features, at the cost of development time and nurturing effort. The approach using scripting is very easy to implement and standardize, but its capabilities are limited. Although the Metacoin protocol is very easy to implement, it has the disadvantage of poor scalability. In the Ethereum system, our aim is to build a general framework that can have all the advantages of all three modes at the same time.

BTLink

The purpose of BTLink is to integrate and improve based on the concepts of scripts, altcoins and on-chain meta-protocols, enabling developers to create any consensus-based, scalable, standardized, feature-complete, Easy to develop and collaborative applications. Ethereum enables anyone to create contracts and decentralized applications by establishing the ultimate abstract base layer - a blockchain with a Turing-complete programming language built in - and set up their freely defined ownership rules, transaction methods, and decentralized applications. State transition function. The main framework of Namecoin can be implemented in just two lines of code, and other protocols such as currency and reputation systems can be implemented in less than twenty lines of code. Smart contracts - encrypted boxes that contain value and can only be opened if certain conditions are met - can also be created on our platform, and because of Turing completeness, value-awareness, blockchain-awareness And the added power of multi-state is far more powerful than the smart contracts that Bitcoin Script can provide.

BTLink account

The BTLink public chain will be the web3.0 decentralized blockchain Internet, the earliest blockchain in the universe and on earth, and will also become the most widely used blockchain protocol in the universe and on the earth. The functional composition includes consensus mechanism: DPos or BlockDAG (consensus is determined according to the specific development situation here), smart contract support, Web3 support, smart contract integration EVM (non-WASM) support; need to use privacy encryption algorithm (by the technical party based on the current technology development to decide) and so on.

At present, millions of blockchain projects around the world are issued and run on the Etherscan, TRONscan, and BSC scan blockchains. By creating a developer-friendly underlying blockchain platform, the BTLink public chain supports allowing anyone to build in the platform and use decentralized applications that run through blockchain technology, allowing users to create complex operations as they wish , HEYTF blockchain system custom development, BTLink public chain opened 153 7537 and other 7737 service projects, rich experience.

(1) Protect the rights and interests of users from the influence of program developers

The developer of the program in the BTLink public chain has no right to interfere with the user, so the BTLink public chain can protect the rights and interests of users who use the program. In addition, the highly decentralized distributed data storage is also one of the biggest features of the BTLink public chain. The advantages of open and transparent transaction data, and the inability to tamper with the data, enable the BTLink public chain to effectively protect the data security of users.

(2) Generate network effects

An information product has an inherent need for interconnection, because people produce and use them to better collect and communicate information. With the expansion of the network scale, users can obtain more value from it, and their needs are more satisfied. The BTLink public chain is open, so it has the opportunity to be applied by many external users, and gradually generate a huge degree of network effect.

(3) Land application in actual business scenarios

Any application scenarios that require high trust, security and durability, such as asset registration, voting, management, and IoT applications in the web 3.0 era, will be affected, infiltrated, participated and interacted with the BTLink public chain on a large scale. .
In the BTLink system, state is composed of objects called "accounts" (each account consists of a 20-byte address) and state transitions that transfer value and information between the two accounts. A BTLink account consists of four parts:

Random number, a counter used to determine that each transaction can only be processed once

Account's current BTLink balance

The contract code of the account

account storage

BTL is the main crypto fuel inside BTLink and is used to pay transaction fees. Generally speaking, BTLink has two types of accounts: externally owned accounts (controlled by private keys) and contract accounts (controlled by contract code). Externally owned accounts have no code, and people can send messages from an external account by creating and signing a transaction. Whenever a contract account receives a message, code inside the contract is activated, allowing

it to read and write to internal storage, and send other messages or create contracts.
news and transactions
BTLink's message is somewhat similar to Bitcoin's transaction, but there are three important differences between the two. First, BTLink messages can be created by external entities or contracts, whereas Bitcoin transactions can only be created externally. Second, BTLink messages can optionally contain data. Third, if the recipient of the BTLink message is a contract account, it can choose to respond, which means that the BTLink message also contains the concept of functions.
BTLink's "transaction" refers to the storage of signed packets of messages sent from external accounts. The transaction contains the recipient of the message, a signature to confirm the sender, the BTLink account balance, the data to be sent, and two values called STARTGAS and GASPRICE. Each transaction uses the price of BTL to pay the gas fee on the gold standard, and the fee for each transaction is 0.05 USDT. If the process of executing the transaction "runs out of gas", all the state changes are restored to the original state, but the transaction fee that has been paid cannot be recovered. If the execution of the transaction is aborted with gas remaining, then this gas will be returned to the sender/. There are separate transaction types and corresponding message types for creating contracts; the address of the contract is calculated based on the account random number and the hash of the transaction data.
An important aspect of the messaging mechanism is BTLink's "first-class citizen" property - contracts have the same rights as external accounts, including the right to send messages and create other contracts. This enables contracts to play multiple different roles at the same time, for example, a user can make a member of a decentralized organization (one contract) an intermediary account (another contract) for a paranoid using a custom quantum proof-based blueprint The individual Porter signs (the third contract) and a co-signing entity that itself uses an account secured by five private keys (the fourth contract) provides an intermediary service. The strength of the BTLink platform lies in the decentralized organization and agency contracts, which do not need to care about what type of account each party involved in the contract has.
1. Transfer value from the sender's account to the receiver's account. If the receiving account does not already exist, create this account. If the receiving account is a contract, run the contract's code until the code runs out or runs out of gas
2. If the value transfer fails because the sender account does not have enough money or the code execution runs out of gas, restore the original state without transaction fees
code execution
The code of the BTLink contract is written in a high-level stack-based bytecode language, called "BTLink Virtual Machine Code" or "EVM Code". Code consists of a series of bytes, each byte representing an operation. In general, code execution is an infinite loop, and the program counter is incremented by one (initial value is zero) to perform an operation until the code is executed or an error, STOP or RETURN instruction is encountered. Operations can access three types of spaces where data is stored:
Stack, a last-in, first-out data storage, 32-byte values can be pushed onto the stack and popped off the stack.
Memory, infinitely scalable byte queue.
Long-term storage of contracts, a key/value storage, where both the key and the value are 32 bytes in size. Unlike the stack and memory that are reset after the calculation is over, the storage

content will be maintained for a long time.
The code can access values in the same way as block header data, the data in the sender and received messages, and the code can also return a byte queue of data as output.
The formal execution model for EVM code is surprisingly simple. When the BTLink virtual machine is running, its complete computing state can be defined by a tuple (block_state, transaction, message, code, memory, stack, pc, gas), where block_state is the global state containing all account balances and storage. At each round of execution, the current instruction is found by calling out the pc (program counter) byte of the code, each with its own definition of how it affects the tuple. For example, ADD pops two elements and pushes their sum, each gas is calculated on the gold standard, SSTORE pops the top two elements and inserts the second element into the contract store defined by the first element Location, although there are many ways to optimize BTLink by just-in-time compilation, the basic implementation of BTLink can be achieved in a few hundred lines of code.

Blockchain and Mining

While there are some differences, BTLink's blockchain is similar to the Bitcoin blockchain in many ways. Their blockchain architectures differ in that BTLink blocks not only contain transaction records and recent states, but also block serial numbers and difficulty values. The block confirmation algorithm in BTLink is as follows:
1. Check if the previous block referenced by the block exists and is valid.
2. Check if the timestamp of the block is greater than the referenced previous block and less than 1 minute.
3. Check that block number, difficulty value, transaction root, uncle root, and gas limit (many low-level concepts specific to Ethereum) are valid.
4. Check if the proof-of-work of the block is valid.
5. Assign S[0] to the STATE_ROOT of the previous block.
6. Assign TX as the transaction list of the block, there are a total of n transactions. For i belonging to 0...n-1, perform state transition S[i+1] = APPLY(S[i], TX[i]). Returns an error if an error occurs in any of the transitions, or if the program has spent more gas (gas) than GASLIMIT here.
7. Use S[n] to assign value to S_FINAL, and pay the block reward to the miner.
8. Check if S-FINAL is the same as STATE_ROOT. If they are the same, the block is valid. Otherwise, the block is invalid.
BTLink uses Greenwich Mean Time, 1 second as the unit for each block to explode, and the arrival time is generally 1 second, which is much higher than Bitcoin and other public chains. The reason is that the state is stored in a tree structure, and each additional block only needs to change a small part of the tree structure. Therefore, in general, most of the tree structure of two adjacent blocks should be the same, so data stored once can be referenced twice with pointers (ie, subtree hashes). A tree structure known as a "Patricia Tree" accomplishes this, which includes a modification of the Merkle tree concept, allowing not only to change nodes, but also to insert and delete nodes . Also, since all state information is part of the last block, there is no need to store the entire block history - this method, if applicable to the Bitcoin system, can be calculated to be 10-20 times more efficient in storage space save.
application

There are three applications on top of BTLink. The first category is financial applications, which provide users with a more powerful way to manage and participate in contracts with their money. Including sub-currency, financial derivatives, hedging contracts, savings wallets, wills, and even some kind of full-scale employment contracts. The second category is semi-financial applications, where there is money but also a heavy non-monetary aspect, a perfect example is self-enforcing bounties for solving computational problems. Finally, there are completely non-financial applications like online voting and decentralized governance.

token system

There are many applications for on-chain token systems, from sub-currencies representing assets such as USD or gold, to corporate stocks, individual tokens representing smart assets, secure unforgeable coupons, and even tokens that are completely unrelated to traditional values. A token system for reward points. Implementing a token system in BTLink 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 prior to 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 to implement a token system in Serpent language is as follows:

<code style="font-family:Menlo, Courier, monospace, monospace, sans-serif;font-size:13.6px;margin:0px;border:none;background-color:transparent;white-space:pre-wrap; " class="hljs">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</code>

This is essentially a minimal implementation of the "Banking System" state transition function described further herein. Some additional code needs to be added to provide the ability to distribute coins initially, and in some other edge cases. Ideally, a function would be added for other contracts to query the balance of an address. Will suffice. In theory, a BTLink-based token system that acts as a sub-currency could include an important feature that Bitcoin-based on-chain metacoins lack: the ability to pay transaction fees directly in that currency. The way to achieve this capability is to maintain a BTLink account in the contract that is used to pay transaction fees for the sender, by collecting the internal currency used as transaction fees and auctioning them off in a constantly running auction, The contract keeps funding the BTLink account. This way the user needs to "activate" their account with BTL, and once there is BTL in the account, it will be reused as it will be recharged each time the contract.

Financial derivatives and stable currencies

Financial derivatives are the most common application of "smart contracts" and one of the easiest to implement in code. The main challenge in implementing financial contracts is that most of them need to refer to an external price publisher; for example, a very in-demand application is a hedge against price fluctuations of BTL (or other cryptocurrencies) relative to the US dollar.

Smart contract, but the contract needs to know the price of BTL relative to USD. The easiest way to do this is through a "data feed" contract maintained by a specific institution (such as Nasdaq), which is designed so that the institution can update the contract as needed, and provides an interface that enables other contracts to pass through, Send a message to this contract to get a reply with price information.

When these key elements are in place, the hedging contract will look like this:

Wait for A to enter 10BTL. .

Wait for B to enter 10BTL.

By querying the data provisioning contract, a dollar value of 10 BTL, eg, $x, is recorded to memory.

After 30 days, A or B is allowed to "reactivate" the contract to send $x worth of BTL (re-query the data-providing contract to get the new price and calculate it) to A and send the remaining BTL to B.

Such contracts have extraordinary potential in cryptographic commerce. One of the issues that is often criticized for cryptocurrency is its price volatility; although a large number of users and merchants may need the security and convenience brought by cryptographic assets, they are not willing to face it, and assets fall in one day. Twenty or thirty percent of the value. Until now, the most common referral scheme was issuer-endorsement of assets; the idea was that issuers create a sub-currency to which they have the right to issue and redeem, giving them (offline) a unit of a specific underlying asset (e.g. Gold, USD) for one unit of sub-currency. The issuer promises when anyone returns a unit of cryptographic assets. Return a unit of related assets. This mechanism enables any non-cryptographic asset to be "upgraded" to a cryptographic asset if the issuer can be trusted.

In practice, however, issuers are not always trustworthy, and in some cases the banking system is too fragile or not honest enough for such a service to exist. Financial derivatives offer an alternative. There will no longer be a single issuer providing reserves to back an asset, but instead a decentralized market of speculators betting that the price of a cryptographic asset will rise. Unlike the issuer, the speculator side has no bargaining power because the hedging contract freezes their reserves in the contract. Note that this approach is not completely decentralized, as a trusted data source for price information is still required, although it is still arguable that this is still reducing infrastructure requirements (unlike issuers, a price issuer does not require licenses and seems to be classified as free speech) and a great advance in reducing the risk of potential fraud.

Identity and Reputation Systems

The earliest altcoin, Namecoin, attempted to use a Bitcoin-like blockchain to provide a name registration system where users could register their names, along with other data, in a public database. The most common use case is to associate a domain name like "bitcoin.org" (or in Namecoin, "bitcoin.bit") with a domain name system that corresponds to an IP address. Other use cases include email verification systems, and potentially more advanced reputation systems. Here is the base contract that provides a name registration system similar to Namecoin in BTLink:

<code style="font-family:Menlo, Courier, monospace, monospace, sans-serif;font-size:13.6px;margin:0px;border:none;background-color:transparent;white-space:pre-wrap; " class="hljs">if !contract.storage[tx.data[0]]:
contract.storage[tx.data[0]] = tx.data[1]</code>

The contract is very simple; it is a database in the BTLink network that can be added, but not modified or removed. Anyone can register a name as a value and never change it. A more complex name registration contract would contain "function clauses" that allow other contracts to query, and a mechanism for the "owner" of a name (i.e. the first registrant) to modify data or transfer ownership. You can even add reputation and trust web capabilities on top of it.

decentralized storage

A number of on-chain distributed file storage companies have emerged over the past few years, Dropbox, which seeks to allow users to upload backups of their hard drives, provide backup storage services and allow users to access them for a monthly fee. However, this file storage market is sometimes relatively inefficient at this point; a cursory look at existing services shows that, especially at the "Mystic Valley" 20-200GB level, which has neither free space nor discounts for enterprise users, mainstream The monthly price of file storage costs means paying the cost of the entire hard drive in one month. BTLink contracts allow the development of a decentralized storage ecosystem, so that users can reduce the cost of file storage by renting out their own hard drives or unused network space for a small profit.

The fundamental building block of such a facility is what we call a "decentralized Dropbox contract". The contract works as follows. First, someone divides the data that needs to be uploaded into chunks, encrypts each chunk for privacy, and builds a Merkle tree from that. Then create a contract with the following rules, every N blocks, the contract will draw a random index from the Merkle tree (using the hash of the previous block that can be accessed by the contract code to provide randomness), and give the first An entity BTLink supports a proof of ownership of a block at a specific index in the tree with a simplified verification payment (SPV)-like. When a user wants to re-download his file, he can restore the file using a micropayment channel protocol (e.g. paying 1 sabo per 32kbytes); the most cost-efficient method is for the payer not to publish the transaction until the end, but instead Replace the original transaction with a slightly more cost-effective transaction with the same nonce after every 32k bytes.

An important feature of this protocol is that, although it appears that one person trusts many random nodes who are not prepared to lose the file, he can divide the file into many small pieces by secret sharing, and then monitor the contract to know that each small piece has returned. Saved by a node. If a contract is still paying, that provides evidence that someone is still keeping the file.

Decentralized Autonomous Organization (DAO)

The concept of "decentralized autonomous organization (DAO)" in the usual sense 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 code. Members collectively decide how the organization allocates funds. The method of distributing funds might be a bounty, salary, or a more attractive mechanism such as rewarding work with an internal currency. This essentially replicates the legal sense of a traditional corporation or non-profit organization for enforcement using only cryptographic blockchain technology. Much of the discussion around DAOs thus far has revolved around the "capitalist" model of a "decentralized autonomous corporation (DAC)" with dividend-receiving shareholders and tradable shares; as an alternative, one is described as An entity called a “decentralized autonomous community” would give all members equal rights in

decision-making and would require a 67% majority to add or remove members. The rule that everyone can only have one membership needs to be enforced by the group.

Below is an outline of how to implement DO in code. The simplest design is a piece of code that can modify itself if two-thirds of the members agree. Although the code is theoretically immutable, it is still easy to bypass the barrier to make the code modifiable by placing the code backbone in a separate contract and pointing the address of the contract call to a modifiable store. There are three transaction types in a simple implementation of such a DAO contract, distinguished by the data provided by the transaction:

[0,i,K,V] Registers the proposed change at index i to the contents of storage address index K to v.

[0,i] register to vote on proposal i.

[2,i] Confirm proposal i if there are enough votes.

The contract then has specific terms for each item. It will maintain a record of all open storage changes and a who voted table. There is also a table of all members. When a two-thirds majority agrees to any change to the stored content, a final transaction will implement the change. A more complex framework would add built-in electoral functionality to enable things like sending transactions, adding or removing members, or even providing voting delegates like delegated democracy (i.e. anyone can delegate another person to vote on their behalf, and this delegation Relationships are transitive, so if A delegates B and then B delegates C then C will decide A's vote). This design will allow the DAO to grow organically as a decentralized community, enabling people to finally hand over the task of picking the right candidates to experts who, unlike current systems, can easily emerge over time as community members change their lines and disappear.

An alternative model is a decentralized company, where any account can have zero to more shares, and decisions require a two-thirds majority of shares to agree. A complete framework would include asset management functionality - the ability to submit orders to buy or sell shares and the ability to accept such orders (provided there is an order matching mechanism in the contract). Representatives still exist in an appointment-based democracy, giving rise to the concept of a "board of directors".

More advanced organizational governance mechanisms may be implemented in the future; now a Decentralized Organization (DO) can be described starting from a Decentralized Autonomous Organization (DAO). The distinction between DOs and DAOs is vague, a rough dividing line is whether governance can be achieved through a political-like process or an "automatic" process, a good intuitive test is the "no common language" criterion: if two members do not Can the same language organization still function? Obviously, a simple traditional shareholding company will fail, while a protocol like the Bitcoin protocol is likely to succeed, Robin Hansen's "futarchy", a mechanism for organized governance through prediction markets is a real A good example of what an "autonomous" style of governance might look like. Note that one does not need to assume that all DAOs are superior to all DOs; autonomy is just a paradigm that has great advantages in some specific scenarios, but may not be feasible elsewhere, and many semi-DAOs may exist.

Further applications 1. Savings wallet. Suppose Alice wants to keep her funds safe, but she is worried about losing or being hacked by her private key. She puts BTL into a contract with Bob, which is a bank as follows: ``` Alice alone can withdraw up to 1% of her funds per day. Bob alone can withdraw up to 1% of his funds per day, but Alice can create a transaction with her private

key to cancel Bob's withdrawal permission. Together Alice and Bob can withdraw funds at will. Generally speaking, 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 withdraw the money. If Bob behaves maliciously, she can turn off his withdrawal permission. `````` 2. Crop insurance. One can easily create a financial derivatives contract with weather conditions rather than any price index as data input. If a farmer in New York buys a financial derivative that pays inversely based on the rainfall in New York, then if there is a drought, the farmer will automatically receive the payout funds and if there is enough rainfall he will be happy because he crops will be good. 3. A decentralized data publisher. For difference-based financial contracts, it is in fact possible to decentralize data publishers through the "Schelling Point" protocol. Schelling points work as follows: N parties provide input values to the system for a specified data (eg BTL/USDT price), all values are sorted, and each node providing a value between 25% and 75% will Rewarded, everyone has an incentive to provide the answer that others will provide, and the answer that a large number of players can really agree with is obviously the correct answer by default, which constructs a number that can theoretically provide many values, including BTL/USDT prices, Berlin temperature Even a decentralized protocol for the outcome of a particularly difficult computation. 4. Multi-signature smart contracts. Bitcoin allows for multi-signature based transaction contracts, for example, funds can be used by collecting 3 out of 5 private keys. BTL can be made more detailed. For example, if you collect 4 of 5 private keys, you can spend all the funds. If there are only 3 private keys, you can spend up to 10% of the funds per day. If you only have 2 private keys, you can only spend 0.5% of the funds per day. In addition, the multi-signature in BTL is asynchronous, which means that both parties can register signatures on the blockchain at different times, and the transaction will be automatically sent after the last signature is in place. 5. Cloud computing. EVM techniques can also be used to create a verifiable computing environment that allows users to invite others to perform computations and then selectively ask for proof that the computations were done correctly at certain randomly chosen checkpoints. This makes it possible to create a cloud computing marketplace where any user can participate with their desktop, laptop or dedicated server, where on-site inspections and security deposits can be used to ensure that the system is trustworthy (ie no node can be deceived profit). Although such a system may not be suitable for all tasks; for example, tasks that require advanced interprocess communication cannot be easily accomplished on a large cloud of nodes. However, some other tasks are easily parallelized; projects like SETI@home, folding@home and genetic algorithms are easily carried out on such a platform. 6. Peer-to-peer gambling. Any number of peer-to-peer gaming protocols can be moved to BTL’s blockchain, such as Frank Stajano and Richard Clayton’s Cyberdice. The simplest gambling protocol is in fact a simple contract to bet on the difference between the hash value and the guess value of the next block, from which more complex gambling protocols can be created to achieve near-zero fees and Fraud-free gaming services. 7. Prediction markets. Whether with oracles or Schelling, prediction markets will be easy to implement, and prediction markets with Schelling may prove to be the first mainstream “futarchy” application as a decentralized organization management protocol. 8. On-chain decentralized market, based on identity and reputation system.

Miscellaneous and Concerns

Implementation of the Improved Ghost Protocol

The “Greedy Heaviest Observed Subtree” (GHOST) protocol is an innovation introduced in December 2013 by Yonatan Sompolinsky and Aviv Zohar. The motivation proposed by the ghost protocol is that the current fast-confirming blockchain suffers from low security due to the high invalidation rate of blocks; because the block takes a certain time (set to t) to spread to the entire network, if miner A digs out a Block Then miner B happens to mine another block before A's block spreads to B, and miner B's block is invalidated and does not contribute to network security. In addition, there is also the problem of centralization: if A is a mining pool with 30% of the computing power of the entire network and B has 10% of the computing power, A will face the risk of generating invalid blocks 70% of the time while B is in the Void blocks are being generated 90% of the time. Therefore, if the invalidation rate is high, A will be more efficient simply because of the higher share of computing power. Combining these two factors, a blockchain with a fast block generation rate is likely to lead to a mining pool that has the ability to actually control mining The hashrate share of the process.

As Sompolinsky and Zohar describe, ghost protocol solves the first problem of reducing network security by including stale blocks when calculating which chain is "longest"; that is, not only a block The parent block and earlier ancestor blocks of the ancestor block, the obsolete descendant blocks of the ancestor block (called "uncle blocks" in BTLink terminology) are also added to calculate which block has the maximum amount of work to support it. prove. We go beyond the protocol described by Sompolinsky and Zohar to solve the second problem - centralization tendency, BTL pays 65% of the reward for stale blocks that contribute to the confirmation of new blocks as "uncle blocks", and they are included in the calculation The "nephew block" will receive 20% of the reward, however, transaction fees are not rewarded to the uncle block.

BTLink implements a simplified version of the ghost protocol that only goes down to layer 5. Its characteristic is that the obsolete block can only be used as an uncle block by the second to fifth generation descendant blocks of its parents, not the descendant blocks that are more distantly related (for example, the sixth generation descendant block of the parent block. block, or the third-generation descendant of the grandfather block) is included in the calculation. There are several reasons for this. First, an unconditional ghost protocol would introduce too much complexity into calculating which uncle of a given block is legal. Second, the unconditional ghost protocol with the compensation used by BTLink deprives miners of the incentive to mine on the main chain rather than an open attacker's chain. Finally, calculations show that the five-layer ghost protocol with incentives achieves more than 95% efficiency even with a block time of 15s, while miners with 25% of computing power gain less than 3% from centralization.

cost

Because each transaction posted to the blockchain incurs the cost of downloading and verifying, a standardized mechanism including transaction fees is needed to prevent spamming. The default approach Bitcoin uses is purely voluntary transaction fees, relying on miners to act as gatekeepers and set dynamic minimum fees. Because this approach is “market-based,” enabling miners and transaction senders to determine prices based on supply and demand, this approach has been well-received in the Bitcoin community. The problem with this logic, however, is that transaction processing is not a market; while it is tempting to intuitively interpret transaction processing as a service provided by miners to senders, the fact that a miner collects transactions requires every

transaction in the network. It is processed by each node, so the largest part of the cost of transaction processing is borne by the third party rather than the miner who decides whether to include the transaction. Thus, a tragedy of the commons is very likely.

However, when given a particular imprecise simplifying assumption, the loophole of this market-based mechanism miraculously removes its own influence. The argument is as follows.

Suppose:

1.

A transaction takes k steps, providing a reward kR to any miner that includes the transaction, where R is set by the transaction publisher, and both k and R are visible to miners in advance (roughly).

2.

3.

The cost for each node to process each operation is C (that is, the efficiency of all nodes is the same).

4.

5.

There are N mining nodes, each with the same computing power (that is, 1/N of the computing power of the entire network).

6.

7.

There is no full node that does not mine.

8.

Miners are willing to mine when the expected reward is greater than the cost. This way, since the miner has a 1/N chance to process the next block, the expected return is kR/N , and the miner's processing cost is simply kC. This is when kR/N > kC, i.e. R > NC. Miners are willing to include transactions. Note that R is the per-step fee provided by the transaction sender and is the lower bound on the miner's benefit from processing the transaction. NC is the cost for the entire network to process an operation. Therefore, miners are only motivated to include transactions where the benefits outweigh the costs.

However, these assumptions have several important deviations from reality:

1. Because the extra verification time delays the broadcast of the block and thus increases the chance of the block becoming a dead block, the miner processing the transaction pays a higher cost than other validating nodes.

2. Full nodes without mining exist.

3. In practice, the distribution of computing power may end up being extremely uneven.

4. Speculators, political enemies, and lunatics do exist with their mission to disrupt the network, and they can smartly set up contracts to make their costs much lower than other validators.

Point 1 above drives miners to include fewer transactions, and point 2 increases NC; so the effects of these two points at least partially cancel each other out. Points 3 and 4 are the main problem; as a solution we simply build A floating upper limit: no block can contain more operations than BLK_LIMIT_FACTOR times the long-term exponential moving average.

specifically:

<code style="font-family:Menlo, Courier, monospace, monospace, sans-serif;font-size:13.6px;margin:0px;border:none;background-color:transparent;white-space:pr

e-wrap; " class="hljs">blk.oplimit = floor((blk.parent.oplimit * (EMAFACTOR − 1) + floor(parent.opcount * BLK_LIMIT_FACTOR)) /EMA_FACTOR)

BLK_LIMIT_FACTOR and EMA_FACTOR are constants set to 65536 and 1.5 for now, but may be adjusted after further analysis. </code>

Computation and Turing completeness

It is important to emphasize that the BTLink virtual machine is Turing complete; this means that the EVM code can implement any imaginable computation, including infinite loops. EVM code implements loops in two ways. First, the JUMP instruction allows the program to jump back to some place earlier in the code, and the JUMPI instruction, which allows conditional statements like while x < 27: x = x * 2, implements conditional jumps. Second, contracts can call other contracts, potentially implementing loops through recursion. This naturally leads to the question: can a malicious user have to shut down by forcing miners and full nodes into an infinite loop? This problem arises because of a problem in computer science called the halting problem: there is no way to know in general whether a given program will end in a finite amount of time.

As described in the state transitions section, our scheme solves the problem by setting a maximum number of computation steps to run for each transaction, and if it exceeds the computation is reverted but still incurs a fee. Messages work the same way. To illustrate the motivation behind this scheme, consider the following example:

An attacker creates a contract that runs in an infinite loop, then sends a transaction that activates the loop to the miner, who will process the transaction, running the infinite loop until the gas runs out. Even if the transaction is stopped halfway through running out of gas, the transaction is still correct (returned to the original place) and the miner still earns the fee calculated for each step from the attacker.

An attacker creates a very long infinite loop with the intention of forcing the miner to keep calculating for such a long time that several blocks have been generated before the end of the calculation and the miner cannot collect transactions to earn fees. However, the attacker needs to issue a STARTGAS value to limit the number of steps that can be performed, so the miner will know in advance that the computation will take too many steps.

An attacker sees a contract that contains a format such as send(A, contract.storage[A]); contract.storage[A] = 0 and sends a contract with only enough fee to perform the first step but not enough to perform the second step transactions (i.e. withdrawals without reducing account balances). Contract authors don't need to worry about defending against similar attacks, since all changes are reverted if execution stops midway.

A financial contract works by extracting the median of nine dedicated data publishers to minimize risk, an attacker takes over one of the data providers, and then designs this variable address call mechanism as described in the DAO chapter to be changeable 's data provider instead runs an infinite loop in an attempt to force any attempt to claim funds from this financial contract will be terminated by running out of gas. However, the financial contract can set a gas limit in the message to prevent such problems.

An alternative to Turing-completeness is Turing-incompleteness, where the JUMP and JUMPI instructions do not exist and only one copy of each contract is allowed to exist on the call stack at a given time. In such a system, neither the fee system described above nor the uncertainty surrounding the efficiency of our scheme may be unnecessary, since the cost of executing a

contract will be determined by its size. Also, Turing incompleteness isn't even a big limitation, of all the contract examples we've envisioned internally, only one has so far required a loop, and even that loop could be replaced by the repetition of 26 single-line code segments. Given the serious troubles and limited benefits of Turing-completeness, why not simply use a Turing-incomplete language? In fact Turing incompleteness is far from a neat solution. Why? Consider the following contract:
<code style="font-family:Menlo, Courier, monospace, monospace, sans-serif;font-size:13.6px;margin:0px;border:none;background-color:transparent;white-space:pre-wrap; " class="hljs">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)</code>
Now, send one such transaction to A, so that, out of 51 transactions, we have a contract that takes 250 steps to compute, miners might try by maintaining a maximum executable step count for each contract and for recursive calls to other The contract calculation of the contract may execute the number of steps to detect such a logic bomb in advance, but this will prevent miners from creating contracts for other contracts (since the creation and execution of the above 26 contracts can easily be put into a single contract). Another problem point is that the address field of a message is a variable, so in general it may not even be known in advance which one of the other contracts a contract will call. So, in the end we have a surprising conclusion: Turing-complete management is surprisingly easy, and Turing-incomplete management is surprisingly difficult in the absence of the same control - so why not make the protocol Turing-complete?

Currency and Issuance
The BTLink network includes its own built-in currency BTL, which plays a dual role, providing the main liquidity for various digital asset transactions, and more importantly, providing a mechanism to pay transaction fees. To facilitate and avoid future dispute periods (see the current mBTC/uBTC/satoshi debate), the names of the different denominations will be set in advance:

1: Wei
10^12: Sabo
10^15: Finney
10^18: BTL

This should be considered an extended version of the concept of "yuan" and "cent" or "bitcoin" and "satoshi", in the near future we expect "BTL" to be used for ordinary transactions, "Finney" to be used for To conduct microtransactions, "sabo" and "wei" are used for discussions about fees and protocol implementation.
The release mode is as follows:

BTL token is the native token of BTLinkBTL public chain, with 21 million open source constants

in the world, BTLink global IDO2100 nodes, each node is priced at 1000USDT, each node gets 1 BTL, one designed to raise funds for the BTLink business ecosystem , and a payment mechanism for cryptocurrency developers from Silicon Valley, Switzerland, Singapore, India, etc. The BTLinkBTL public chain running on the web3.0 decentralized blockchain Internet protocol has been developed for 12 months and will be launched globally. 2100 nodes are BTLink original shareholder nodes, 30% of the USDT obtained by IDO is used for the development costs of BTLink's various business ecosystems, 20% is for marketing, and 50% of the USDT will be added to the BTL liquidity pool.

12% of the total BTL is the web3.0 decentralized blockchain Internet protocol, the BTLink public chain, and the development of various commercial ecological applications, as well as the development cost of more cryptography around the world participating in the entire project.

For the BTL belonging to the technical team, 50% of the issued BTL will be locked every time the market value rises, and the locked BTL will increase by 10%-51% with the BTL market value each time, unlocking 1% of the remaining total. 50% of the BTL is used for the web3.0 protocol and the development of the BTLink public chain business ecosystem.

88% of the total BTL belongs to all shareholder nodes. The BTL public chain goes online to unlock 3% of the BTL of 2,100 shareholder nodes, and the remaining 97% of the BTL will increase by 10%-51% each time according to the BTL market value, unlocking 1 of the remaining total. %.

For each increase of 10-51%, multiply the number of issued BTL by 5% as the issuance amount for each market value increase, 12% is issued to the technical team, and 88% is issued to all shareholder nodes.

Market value issuance mechanism decomposition

Warren Buffett, the world's investment guru, has been denying the value of Bitcoin for many years, saying that Bitcoin is speculation and does not create social value, and the underlying algorithm of Bitcoin is block code. This is a very one-sided understanding of Buffett, because he does not understand the open source code of the blockchain, which solves the core value of the code that can be trusted and can be confirmed since the birth of mankind. Avoiding the uncontrollable human nature of a society ruled by man, the code rule world "code rule society". Like the unification, quantity, and scale of the ancient Eastern Qin Empire, it can finally be realized by human beings, a globally unified "World Blockchain Code Code". Make the world more equal and democratic; make the world more credible and harmonious. In the next century, under the world of code governance, the world will eventually realize a community with a shared future for mankind and a home for the earth!

And BTLink, it is not a simple and valuable BTL digital asset, it is a panoramic public chain integrating commercial ecological application. The web3.0 decentralized blockchain Internet protocol that can be accessed by industries all over the world can empower businesses around the world. At the same time, it leads the world economic system and traditional industries into the era of web 3.0. The pain point of the web2.0 Internet is that data is stored in centralized servers, data is occupied by centralized companies, data can be tampered with, data can be destroyed, data can be faked, data is leaked, and personal privacy is insecure. At the same time, web2.0 Internet companies have hindered the development of human civilization. Today, the web3.0 decentralized blockchain Internet protocol created by BTLink solves the above-mentioned pain points of the web2.0 Internet and solves the various drawbacks of human rule by man.

Through the non-tamperable blockchain encryption code protocol, every A citizen participating in the web3.0 Internet protocol, his own data and traffic run on the blockchain, and the participants control and control themselves through wallet addresses and private keys. All data, traffic, assets, wealth, etc. on the chain will always belong to the participants. , without the authorization of any centralized organization, can be transferred and inherited arbitrarily with the private key!

Blockchain Mall

The launch of the blockchain mall can achieve decentralization! Create the first platform for the perfect integration of consumer shopping and blockchain technology, help physical enterprises to quickly increase product sales, and also allow consumers to get the greatest benefits from shopping and value feedback. The blockchain mall uses a variety of encryption, verification methods and other means to protect the authenticity and security of data, enhance users' trust in consumption and investment consumption, and can accurately learn users' consumption behavior, consumption index, consumer literacy and other peer-to-peer services. The blockchain mall also embeds DTO smart contracts, which use the trustworthy mechanism of digital orders and smart contracts to conduct secured transactions. The trust cost and transaction cost between buyers and sellers are reduced. At the same time, the decentralization feature removes the participation of third-party institutions. , greatly shorten the time and improve the efficiency. It has created a new economic platform for digital shopping of honest encrypted e-commerce.

BTLink IPFS encrypted storage

A peer-to-peer distributed file storage protocol that connects all computers with the same file system. The traditional Internet HTTP protocol is to search for domain name addresses, but IPFS is to search for content addresses. Using IPFS, a method that subverts the HTTP protocol, can theoretically make the network faster and more secure.

The idea of IPFS is to allow files to be stored and read in a distributed manner. Now all the information on the Internet is stored in the server. If the server hangs, we will not be able to search for information. In order to prevent this from happening, IPFS technology smashes files and stores them in different hard drives. When downloading, they are read from these hard drives scattered around the world. In fact, those who have used BT download will find that IPFS is actually a BitTorrent protocol, and the development team has slightly upgraded the BitTorrent protocol.

Traditional storage vs distributed storage

Traditional storage:

A single point of failure paralyzes the entire network

Communication relies on backbone network bandwidth is expensive

Storage media is easy to be monopolized, data cannot be autonomous

Slow data upload and download

Difficult to upgrade storage space Expensive storage cost

Distributed storage:

Aggregate hundreds of millions of nodes: the system is stable and reliable

Encrypted Fragmented Data: Safe and Secure Your Data

Elastic storage capacity: high scalable performance

Nearest multi-point transmission: upload and download extremely fast

Decentralized database: breaking the data monopoly

Block technology blessing: data traceability and confirmation

Local storage has always been known for its high reliability, good stability, and rich functions, but at the same time, local storage also exposes shortcomings such as poor horizontal scalability, high price, and difficult data connectivity, which is easy to form data islands, leading to data center management and problems. Maintenance costs remain high.

With the development of cloud computing, especially the successful practice of Internet enterprise cloud data centers, the voice of distributed storage (cloud storage) replacing traditional storage arrays (local storage) is growing. Compared with local storage, distributed storage not only improves the utilization of storage space, but also achieves elastic expansion, reduces operating costs, and avoids wasting resources.

Information published by relying on IPFS will not suddenly disappear in the event of a service provider or hosting network emergency, security is increased, IPFS has no central distribution system and is fast.

The advantages of IPFS can just solve the problems of traditional centralized cloud storage data leakage, hardware damage, weak repair ability, low security, and the risk of operation termination at any time.

Distributed storage replicates the database into multiple copies through the underlying protocol of IPFS to ensure redundancy, and then divides it into multiple small parts, which are distributed and stored on many nodes in the network, so that as long as enough nodes operate normally, the data is safe.

BTLinkVR, AR, XR virtual scene ecology

Virtual reality (VR) is a new world created by computer and electronic technology. It is a simulated environment that seems to be real. Through a variety of sensing devices, according to the user's own feelings, people's natural skills are used to simulate objects in the virtual world. Conduct inspections or operations; at the same time, provide multi-channel information such as visual, auditory, tactile, etc., so that users can perceive intuitively and naturally in real time through sight, hearing, touch, etc., and immerse participants in the simulated environment. Virtual reality technology has achieved great development in the last ten years, which is mainly due to the rapid development of computer software and hardware conditions. The virtual scene modeling work in virtual reality technology is the core problem of research.

Modeling is the process of establishing a model and establishing a system model, also known as modeling. Modeling is an important means and premise of design. Design modeling is the work to be done in the early stage of design. To establish a framework is like drawing a floor plan before building a house, and a decoration drawing for decoration. Only when the modeling is completed can the design be completed step by step.

In the later stage, naked-eye holographic projection is a 3D technology that does not require glasses. The audience can see three-dimensional virtual characters. In the first stage, human "immortality" can be realized, and relatives and friends who can pass away can be "resurrected and immortalized" in the metaverse world of web3.0. , you can communicate and interact with living relatives and friends at any time. This technology is widely used in some museums. In the later stage, our BTLink will fully use the holographic stereoscopic projection technology to realize the interaction and communication between people. Instead, the projection equipment projects images from different angles onto a holographic film imported from abroad, so that you cannot see other images that do not belong to your own angle, thus realizing a true holographic stereoscopic image. Fully utilize the ecological implementation to freely adjust the timeline when

compositing, and combine multiple timelines for complex effect layers.

BTLink ecological chain game interactive ecology

GameFi = DeFi + NFT + Game Its highly decentralized features and rules allow players to realize the privatization, security and transparency of game assets. All props, resources and even characters of players in the game are monetizable and can be freely traded between players to earn profits.

1. Play-to-Earn

Play-to-Earn, referred to as P2E, is the most important element in GameFi. "Earning while playing" is the biggest difference between GameFi and the so-called traditional games.

Money in traditional games (such as gold coins in World of Warcraft, money in CS, coupons and diamonds in Glory of Kings) can only be used in the same game and cannot be exchanged for cash.

In GameFi games, the game tokens earned by players can be circulated in the market like ordinary cryptocurrencies and can be exchanged for cash or other currencies. To make money from playing games, players can continue to participate and attract more players to join. The operating model of each item in disguise is very important.

In addition, "making money while playing the game" itself has an excellent publicity effect. As long as the publisher controls the circulation of cryptocurrencies, if the price rises sharply, it will naturally attract a large number of users, become loyal players, and earn coins together. At the same time, it saves a lot of promotion costs; game publishers can extract commissions through transactions, and can also earn income. The conclusion is that as long as the game ecosystem is well established and the "metaverse" operates smoothly, a win-win situation can be created for all parties.

All props are NFT tokens; casting props into NFT has three major characteristics: it is truly owned by the player, each NFT is unique and rare, and it can be sold in other markets (eg Opensea) in addition to the built-in market in the game. .

3. Mortgage props

Based on the above attribute advantages, some game props NFT can become the collateral of other DeFi platforms, allowing players to mortgage their game assets (NFT) to borrow; or in other Dapp applications, by pledging NFT assets or game tokens Provide liquidity and make money through mortgages (such as liquidity mining and mortgage financing).

4. DAO Governance

Generally, game tokens are the most rewarding in GameFi, or the governance token of the game (Governance Token); those who hold Game token tokens can upgrade or propose improvement plans for the game by voting. DAO organization management is a common practice in decentralized applications, so that players are no longer just the role of players, and even have the right to decide the future development of the game.

BTLink Decentralized Exchange

The core of decentralization is "de-custody", and the core of decentralized exchanges is "asset de-custody". That is to say, in the process of transaction, no one or a centralized institution can

use your assets. Decentralized exchanges are different from centralized exchanges, mainly in terms of technology and governance.

From a technical point of view, decentralized exchanges implement transactions through smart contracts on the chain. Traditional centralized exchanges trade off-chain.
From a governance perspective, the governance of a decentralized exchange is open and community-driven. The governance model of centralized exchanges is the same as that of traditional companies.
When using different centralized exchanges, account registration and KYC authentication need to be repeated. With DEXs, this problem does not exist: you only need to have a wallet to use different DEXs. That is, no registration is required, just one is enough.
Here are the top five decentralized exchanges on BlockSky:

application
Transaction process
1. Account opening: Register to obtain a new address and key. The user has the private key and has absolute control over the asset. Once lost, it cannot be retrieved.
2. Recharge: The recharge is relatively simple, directly recharge from the wallet address to the new address of the decentralized exchange
3. Transaction: When a transaction is initiated, the smart contract of the decentralized exchange is directly executed to complete the transaction. During the whole process, the user always owns the ownership of the currency, and the decentralized exchange has no control.
4. Withdrawal: Users do not need authorization, manual review, and transfer from the decentralized exchange to their wallet address.

Incurring Fees Using BTL
1. When recharging, from the user's wallet address to the new address of the exchange, BTL should be used as the handling fee.
2. During the transaction, the decentralized exchange will also charge a handling fee, which is directly deducted from the currency of the transaction.
3. When withdrawing money, the exchange address is recharged to the user's wallet address, and the corresponding BTL will be used as a handling fee.

Advantage
1. The decentralized exchange model is simple, mainly matching transactions, and does not host the user's assets, which eliminates the possibility of the exchange's self-stealing.
2. The biggest difference from centralized exchanges is that all of this is realized through smart contracts, and asset custody, transaction matching, and asset settlement are all placed on the blockchain.
3. The use of smart contracts to achieve a decentralized and trustless transaction mechanism solves the risks of internal operation risks, business ethics risks, and asset theft that seriously affect the security of users' assets caused by human factors in centralized exchanges.
4. Solve the functions of the current decentralized exchange, no K-line chart, no currency price listing, no buy-to-buy contracts, etc.

5. The user's custody assets can be freely transferred without anyone's approval, and there is no need to worry about hackers stealing, losing coins and other problems, with sufficient security.

BTLink Crypto Social Finance

Each message is encrypted so that only the sender and receiver can read it, even a hacker cannot intercept it, due to the open source encryption used. You can let experts test and find problems, which can make this program more secure. This software also supports automatic deletion of sent information, which can prevent others from leaking your chat records after they get your phone.

Applicable to ios, android and windows phone, pc and apple computer, etc. In the latest encryption technology, the external end-to-end encryption function is enabled by default. The sent message is transmitted to the other party's mobile phone in the form of an encrypted code. Only if this chat tool is installed on the recipient's mobile phone, and the encrypted code can be converted into a visible plain text message through the built-in encryption key of the software , the encryption key only exists on the user's device, and no content is stored on the external server, so no one, not even the outside world, can read your information. It also provides burning after reading, which can set an automatic destruction time for the messages in the conversation. It can encrypt transfers, encrypted chat, and video transmission. Applied to major ecological sectors, such as metaverse, NFT, games, etc.

It is solved that web2.0 data is stored on a centralized server, and the data traffic of social accounts is not owned by the platform company. Because all data traffic runs on the blockchain, users register with their wallet addresses, and master the data of social accounts on the blockchain with their private keys, which can be given to and inherited by anyone with the private key.

Support BRC-20 protocol and BRC-721 protocol, enterprises and individuals can issue their own sub-chains of enterprises and individuals with one click.

Roundup: Decentralized Applications

The above contract mechanism enables anyone to run command-line applications (fundamentally speaking) on a virtual machine through consensus of the entire network, which can change a network-accessible state as its "hard disk". However, for most people, the command line interface used as the mechanism for sending transactions lacks user friendliness enough to make decentralization an attractive alternative. Finally, a complete "decentralized application" should include the underlying business logic components [whether fully implemented in BTLink or not, using BTLink in combination with other systems (such as a P2P messaging layer, one of which is planned to be put into the BTLink client) , or only other system ways], and upper-level graphical user interface components. The BTLink client is designed to be a web browser, but includes support for "BTL" Javascript API objects that can be viewed by specific web pages in the client to interact with the BTLink blockchain. From the perspective of "traditional" web pages, these 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 can use BTLink in some way to store web pages on its own.