LemoChain

Technical Whitepaperv1.1

2018/05/10

Contents

Background

Since Bitcoin's implementation of open source P2P currency in 2009, we have witnessed a global phenomenon: an emergence of countless projects founded on achieving socio-economic development by means of decentralization and distributed ledger technologies. Arguably, the most notable of these projects being the Ethereum project, which focused on proving the potential of smart contracts whilst developing a universal platform for decentralized applications (DApps). However, despite these advances, the blockchain world still faces a host of challenges from both technical and industrial perspectives:

- **Many existing smart-contract platforms have struggled to connect with real business logic due to the technological distance from everyday business, with both Bitcoin and Ethereum architecture having limited widespread application to the common consumer.**
- **Current consensus mechanisms lack flexibility and efficiency; the exchange of value is not centered around transferability to real commercial scenarios.**
- **The compatibility problems between different blockchain platforms. For example, UTXO-based Bitcoin ecosystem isn't compatible with Account based Ethereum.**
- **Existing blockchain platforms are isolated from off-chain data. At present, most smart contracts solely accept on-chain data as the trigger condition, lacking interchangeability within the real world.**

We are committed to building a brand-new blockchain data transmission ecosystem; Lemo, as a universal Internet data value transmission protocol for future decentralized applications, digitizing and tokenizing data values, and promoting Blockchain technology is applied to real-life business scenarios.

Design Concept

1. Scalability

The scalability of the network is one of the most important elements. Without faster transaction speeds, networks it will not be able to carry large-scale applications in the future, and the establishment of an ecological vision will be meaningless. We can see that the current Bitcoin network, one of the mainstream networks, confirms an average of 300,000 transactions per day. The transaction confirmation time is at least one hour, which is far below the requirement for the settlement ability of a financial instrument. The average confirmation time of the Ethereum network is about 14 seconds. In the face of phenomenal applications, network congestion is prone to occur, and it cannot be restored in the long term, and it cannot carry large-scale applications.

LemoChain chose to use the DPoVP technology with its high response speed to solve this problem. With reference to the principle of DPoS consensus mechanism, it can provide close to 10,000 TPS transaction throughput and a confirmation speed of less than 1 second. It has reached Visa-scale transaction processing capabilities and provides sufficient growth space for the future development of LemoChain.

2. Universal

As a general data transaction blockchain, LemoChain does not favor a specific scenario. LemoChain seeks to create platform implementation for various industry solutions. At the same time, LemoChain will also provide development kits and templates to help developers quickly achieve this goal.

3. Effortless Upgrade System

No system can avoid bugs and optimizations. Even if it survives the analysis of countless hackers and scientists, it still has upgrading needs. However, the centralization of the Bitcoin network’s computing power has resulted in the mining pool having absolute right to speak. The evolution of the Bitcoin network cannot occur smoothly when there is a conflict of interest between the Bitcoin users and the mine pool and even different mining pools. Another Ethereum chain technology on behalf of Ethereum, once unable to reach consensus due to bifurcation, led to the parallel development of the two branches of ETC and ETH.

After the most thorough testing still can not avoid the emergence of a small number of bugs, LemoChain must ensure that these bugs can be quickly and easily fixed without any ambiguity.

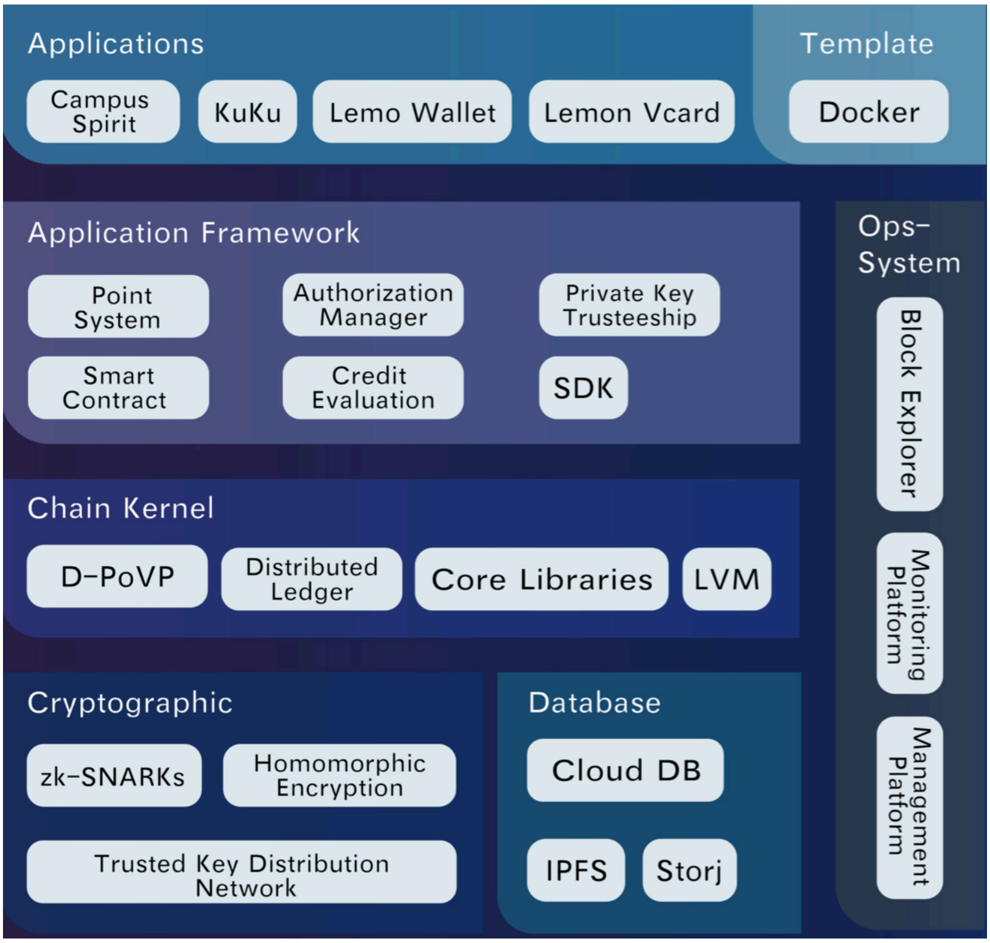
4. Security, privacy

LemoChain will focus on protecting the user’s data, transaction content, and other privacy from the blockchain core code to the upper application. Making sure no one can get this data other than the user himself. Codes that involve sensitive information will also be open sourced entirely to accept user reviews. At the same time, LemoChain cooperated with a dedicated code auditing team to ensure that the entire mechanism was able to withstand malicious attacks.

5. Development

LemoChain will build a blockchain infrastructure, provide convenient operating interfaces and development kits, share advantages with industry partners, jointly promote the development of the data exchange market, and create a win-win environment for the blockchain.

6. Systems Architecture



LemoChain shields the blockchain details at the bottom of the system and provides services through a stable interface and SDK to upper–layer applications. The modules are loosely coupled, support for replacement and upgrade, and the upper application does not require any changes.

Consensus Mechanism

The consensus mechanism has always been a hot topic in blockchain research. The prevailing view is that effective algorithms must comply with the Byzantine fault tolerance principle. And it needs to be safe, clear and irreversible in the shortest possible time, and it is easy to provide a most solid and decentralized system. In practice, the process is divided into two aspects: selecting a unique node to generate a block and making the transaction ledger irreversible.

1. Common Consensus Mechanism

The Byzantine fault-tolerance problem can be expressed as a major solution to the problem of a general trusted communication. A group of generals want to achieve a certain goal (consistent attack or retreat), and the individual actions cannot be completed. A consensus must be reached through cooperation. However, due to the presence of traitors, the generals do not know how to achieve agreement. Here ”consistency” is the main content of the discussion of General Byzantine. Currently there are many algorithms that have solved the Byzantine General problem. Here are some common algorithms

1.1 PBFT

The PBFT mechanism is represented by the IBM HyperLedger fabric. The core of a solution it describes is the state machine replica replication algorithm. First, a master node is responsible for block production; then, the received transaction data is broadcast to the entire network. Eventually each node keeps a copy of the state of the service. The total number of sets composed of all copies is denoted by N, and each copy is represented by 0 to |N|–1, as long as the number of untrusted copies is (analogous to the number of traitors) f ⩽ (|N|–1)/3, then this system can operate normally. Under this mechanism, all nodes eventually reach the same consensus and therefore do not diverge. If the master node goes offline, the backup node triggers a timeout mechanism and selects the next master node based on the node number.

The working premise of PBFT is that the nodes in the network are known in advance, and therefore are only applicable to the consortium chain or private chain. Nodes working under the PBFT mechanism need to communicate with each other. The complexity of network communication is O (n^2). The traffic volume will grow explosively as the number of nodes grows. In a public-chain environment, it will cause serious broadcast storms.

1.2 Proof of Work (PoW)

PoW is a research report posted by Nakamoto in a secret crypto-discussion group in 2008. The report describes his new ideas on cryptocurrency and the proposed bitcoin consensus algorithm. Each node in the entire system provides computing power for the entire system (abbreviation referred to as computational power). Through a competition mechanism, the nodes that have completed the most outstanding calculation work are rewarded by the system, once the distribution of newly generated currencies is completed. Simple and stable, it has withstood all kinds of attacks after attracting the attention of various hackers and scientists.

1.3 Proof of Stake (PoS)

The POS mechanism can be described as ‘virtual mining’. Since PoW mainly depends on the scarcity of computer hardware to prevent witch attacks, PoS relies mainly on tokens in the blockchain itself. The holder holds the token as a deposit in the PoS mechanism so that they become validators. The PoS algorithm randomly selects one of these verifiers and gives them the right to generate the next block. The basis for selection is how much they invest in tokens and how long they hold tokens. If, within a certain period of time, the verifier does not produce a block, a verifier will be reselected instead of generating a new block. Similar to a system that distributes interest based on the amount and timing of token possession. The actual implementation of PoS will also have some mechanisms for clearing currency age, currency decay, etc. The PoS mechanism will have the advantage of not being able to carry out force attacks because the person who launches the attack needs to hold 51% of the total currency. After the attack causes the currency value to fall, he will be the person whose total currency value is most damaged.

Under the PoS mechanism, some holders will hold large amounts of tokens for a long period of time in order to increase voting weight. As a result, the total tokens in circulation will be reduced and prices will be more vulnerable to fluctuations. Because there may be a large number of big players or mine pools holding most tokens in the entire network, the entire network may become more and more centralized as the running time increases.

1.4 Delegate Proof of Stake (DPoS)

The DPoS consensus mechanism sacrifices certain aspects of decentralization on the basis of PoS, whilst greatly accelerating the time-consuming transaction confirmation process. The main principle is to randomly select a Lemo limited number of agent nodes among all nodes, and these nodes take turns accounting and take the consensus of the agent as the consensus of the entire network. New block rewards are shared by both agents and voters. In order to avoid adverse effects on the blockchain after the malicious node becomes the agent, the DPoS mechanism needs to re-elect the agent after a certain period of time. DPoS currently has the advantages of maturity and high throughput. Only the agent node can reach a consensus to confirm the transaction, and its transaction frequency can even reach the centralized Visa settlement scale.

2. Delegated Proof of Valuable Participation (DPoVP)

LemoChain integrated the BFT and DPoS consensus mechanism, and included the value of user contributions into the scope of incentives to develop a new DPoVP mechanism. The representative feature of this technology is to define multiple types, not just to acquire tokens in the mode of renting and selling idle computer resources, and to quantify the user's behavior through various dimensions of the scoring system, as a user's contribution to the platform, and loyalty. The measure of degree. On the one hand, it provides a standard for identifying high-quality users. On the other hand, it can also be used as a platform to motivate users. The platform contributes to the prosperity of each application ecology on LemoChain through rewards to users, and further attracts more traffic to the platform. This virtuous circle mechanism has become a major boost for LemoChain's rapid application development.

The basis of the DPoVP consensus mechanism combines the advantages of BFT's fast consensus and the DPoS's throughput capability. It adopts the rule of out-of-sequence. Once the block from the last witness is received by the two-thirds node, it can be immediately confirmed. Start production of the next block. Equivalent to the block time interval is only limited by the network transmission speed, under normal circumstances can achieve less than 1s average verification speed and average 8000TPS data throughput.

2.1 Voting

In order to maintain the independence and scalability of functions, Lemo uses smart contracts to implement voting. The node is registered as a candidate through the contract and accepts user voting. Finally, the first 21 nodes are selected as witnesses based on the voting results.

2.2 The ownership of bookkeeping rights

The bookkeeping right mainly solves the problem of whether to block oneself or when to block. Witnesses proceed to block in the lexicographic order of the address. After you block out or receive a new block, you need to recalculate the countdown of your own block. After the time is zero, a new block will be generated directly.

First we define I to indicate the blocker number. Have:

I_{me}The current node's block number;

I_{new}The block number of the block that it produced or received and passed. Height is +1 of the current chain height +1；

I_{last}The block number of the previously confirmed block before confirming the new block;

Define d as the distance of the two outlier numbers:

$$\mathrm{d}\left(\mathrm{a},\ \mathrm{b}\right) = \left(\left(\ I_a\ -\ I_b\right) +\ C\right)\ mod\ C)$$

Where C is the number of consensus nodes.

The validity of the new block needs to be verified before triggering the recalculation countdown. The countdown formula is as follows

$$\mathrm{T} = \begin{cases} (C-1)t_o & ,d\left(I_{me}, I_{new}\right) = 0 \\ 0 & ,d\left(I_{me}, I_{new}\right) = 1 \cap t_{now} - t_{new} \geq t_w \\ t_w - (t_{now} - t_{new}) & ,d\left(I_{me}, I_{new}\right) = 1 \cap t_{now} - t_{new} < t_w \\ \left(d\left(I_{me}, I_{new}\right) - 1\right)t_o - (t_{now} - t_{new}) & ,d\left(I_{me},\ I_{new}\right) > 1 \end{cases}$$

t_{now}Current time;

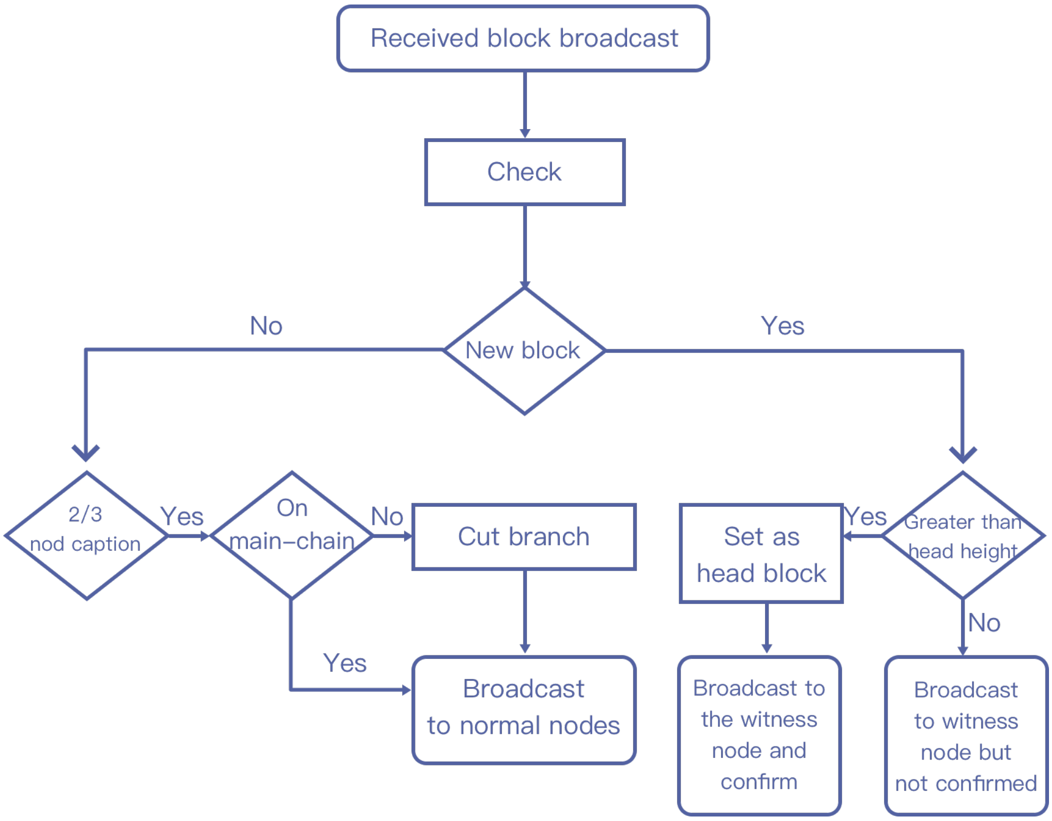
t_{new}Timestamps received in the header of the block;

t_WThe waiting time for the current node to come out of the block. This is to prevent the block from going too fast, resulting in the early transactions being less empty;

t_OThe maximum available time for the block to come out of the block, beyond which the next node should immediately block out;

2.3 Consensus

When a new block is produced, it will first be broadcast between witnesses. After two–thirds of the witness nodes are confirmed, this block will enter the "final confirmation status." At this point, the witness node broadcasts the block to the entire network. For ordinary nodes, the received block must have reached consensus and never be forked. As long as the received block can be verified, it can be safely saved.



The factors affecting the speed of transaction confirmation only depend on two parts: the duration of the consensus between the witness nodes, and the time–consuming propagation of the transaction and the final consensus block, between the common node and the witness node. The process of confirming and synchronizing the witness nodes to each other accelerates the broadcast proliferation of the block, which can further increase the speed of reaching consensus in the harsh network environment.

3. Risk response capabilities

In LemoChain, two-thirds of witnesses are required to confirm that they can agree on a single block. Therefore, the number of malicious nodes that can be tolerated cannot exceed one third of the total number of people. In fact, as long as there is an honest node in some cases, the normal operation of LemoChain can be guaranteed. The following analysis of several typical risk scenarios.

3.1 “Nothing At Stake” Attack

Blocking in the DPoVP network requires no proof of workload, so the amount of computation is very low. Witnesses can confirm and block on all forks, regardless of which fork wins, they can all benefit. The blocking program does not verify the forked block at all, or aims to launch a dual-flower attack. This method does not require additional computing power and other costs, the equivalent of the bifurcation of the blockchain indifferent.

Only 2/3 node consensus needs to be reached in the DPoS to broadcast to ordinary nodes. Therefore, as long as no more than one-third of the witness nodes block out of all branches, no consensus can be reached on the abnormal block. If these witness nodes only randomly select a branch to confirm and block, it will reduce the consensus efficiency, resulting in no branch to reach a consensus. In fact, in this case, as long as there is an honest witness node, a longer blockchain can be obtained after the block is generated at the node, and thus convergence is quickly achieved and consensus is reached.

3.2 Block risk at the same time

Assume that the communication is normal at a certain moment, A node is out of the block, node B should be out of block at 10 seconds, and node C should be out of block at 20 seconds.

A quickly blocks and broadcasts, but fails to synchronize to Node B and synchronizes only to Node C. The C–node recalculated block time may be 10.3 seconds. This causes B and C nodes to block out in a very short period of time and broadcast to other witness nodes, resulting in a fork, unable to reach a two–thirds consensus.

According to the time calculation formula, the witness node does not recalculate (shorten) the block–out time when it receives discontinuous blocks, and it assists in broadcasting the confirmation information of other nodes. The nodes make decisions after trying to collect all the blocks on all branches.

The forking selection rule is to select the longest chain preferentially, and when the same length is selected, the lexicographic order of the block hash at the bifurcation is preferentially selected to be the front chain.

All nodes select the forked chain according to the same rule. After reaching two–thirds of the consensus, all the blocks in the chain enter the “final confirmation state” and are broadcast to ordinary nodes. Since bifurcation occurs at the consensus node, it does not have any effect on ordinary nodes that only receive access to the ”final status” block.

3.3 Consensus Network Splitting Risk

Assume that 21 consensus nodes have 11 in China and 10 in the United States. Due to the special circumstances such as the interruption of the optical cable, the network is split into two parts that cannot communicate with each other, each of which generates a bifurcation chain. Here are two networks named C and A.

Each node will continue to count down and out of blocks, but it will never receive more than two-thirds of the node consensus. The witness no longer broadcasts blocks to ordinary nodes. Taking the C network as an example, in the case of ignoring block time and network transmission time-consuming, the time for each node to cycle (generate $C_C$block) is linearly related to the number of A network nodes, $C_A \times t_o$, the average block interval is:

$$\frac{C_A \times t_o}{C_C}$$

Obviously, at the same time witnessing more nodes will produce more blocks. After the recovery of the optical cable, the witness network is connected and the new block can be broadcast to all nodes. Each node pulls the complete branch chain along the parent block hash, and selects the final chain according to the longest chain principle. Thus, two-thirds of the consensus reached in each block of the chain began to be broadcast to ordinary nodes. The process of confirming the block in this case will be stagnant for a while, but there is no security risk for the transaction on the chain.

3.4 Early consensus attack

Malicious witness nodes can broadcast unconsensual blocks to ordinary nodes. This will lead to ordinary nodes diverging in the short term. When the malicious witness node does not exceed 1/3 of the case, the ordinary node needs to wait for 1/3 node to get out of the block and then identify the correct chain according to the longest chain principle. In the case of 21 witness nodes with a timeout of 10 seconds, the fork time can last only 3.5 to 70 seconds. The malicious witness node will then expose itself and be voted out.

4. Data Storage

Lemo aims to create a decentralized data rights and circulation platform; the safe storage, encrypted transmission, and copyright attribution of participant data are crucial to the success of such a platform. The blockchain's security largely depends on it being mirrored by a large number of nodes and being 100% available. The storage of large, variable files on the chain will result in very high cost. For example, there is a high-performance blockchain application that processes 1 million transactions per second. Each transaction generates 100 bytes of records, and the consumed storage space will increase at more than 100MB/s. In order to maintain practicality, it is necessary to periodically truncate transaction records on the blockchain and save a blockchain state snapshot. However, the complete transaction record will still be copied to each node, causing unnecessary backup overhead. Therefore, it is a practical and nonextensible decentralized file storage solution to store large-size data in the blockchain.

Flexible docking will support decentralized IPFS, storj file system, centralized cloud database and other programs, therefore providing users with a more diverse choice when it comes to data storage. In order to further simplify the application platform interfaces, Lemo will provide: a storage system adapter SDK, a public and private key generation package, address generation, signature verification, encryption, decryption and other functions. These will be shielded by complex signature generation rules, coding-conversion problems, and a variety of underlying error-code processing logic. The user identity management module and the private key storage module can be optionally introduced on the interface to reduce the public-private key management burden on the service application. Overall, Lemo is providing a convenient and easy to use interface for business developers.

IPFS is a global, P2P distributed version of the typical cloud system. It reduces the risk of data loss due to data center failures. IPFS's p2p network uses DHT technology, which replaces domain–based addresses with content–based addresses. IPFS is a global, P2P distributed version of the typical cloud system. It reduces the risk of data loss due to data center failures. IPFS's P2P network uses DHT technology, which replaces domain–based addresses with content–based addresses.

The centralized cloud database will be built on a world–class system of large scale cloud service providers, running as a stable, reliable, and low–cost database. It can provide 99.99999999% data reliability, 99.9% availability, up to 200gbps throughput and low latency to 1ms. Lemo will encrypt user data and open source code to secure a high–trust relationship and ensure that the privacy of user data is not violated.

5. Safe Transactions

In a traditional private data exchange scenario, the data of both parties in the transaction needs to be disclosed to each other or be matched by a trusted third party. In the current volatile and malicious business environment, this is extremely risky. The third party's right to facilitate the transaction is too large, and there is the possibility of leakage, tampering, and concealment of data between the two parties. As a result, protocols that can support joint computing and protect the privacy of participants have become increasingly more recognised. Lemo is committed to introducing Secure Multi-Party Computation (SMC) to solve this problem.

Secure multi-party computation is a collaborative computing solution that solves the problem of privacy protection among a group of non-trusted parties. SMC ensures the independence of input and the correctness of calculation; all without disclosing each input value to any of the participants. In general, a secure multi-party computing problem calculates any probability function based on any input to a distribution network. Each input party has an input on the distribution network. This distribution network needs to ensure the independence of the input and the correctness of the calculation. Also, in addition to their respective inputs, they do not disclose any other non-relevant information that can be used to derive other inputs and/or outputs.

Taking marriage and love website pairing as an example, the user's conditions and features are mapped into points in a t-dimensional space.

$$P = \left(x_1,\ x_2,\ \cdots, x_t\right), x_i \in \left[0,\ 1\right]$$

Let the target of the demand side be a, the data of the data provider is to satisfy:

$$a, b_i \in P$$

The matchmaking transaction algorithm can be summarized as the nearest neighbor algorithm NN in the t–dimensional space b , that is, the minimum distance d between a and B is found to be minimum

$$\mathrm{b}^{\backprime} = \mathrm{NN}(\mathrm{a},\ \mathrm{B}) = \min_{i=1,\cdots,n} d\left(a, b_i\right)$$

In order to protect the confidentiality of the B data, the nodes in the blockchain need to be isolated from the data a and b, and only the encrypted data can be obtained. Therefore, Lemo introduced will introduce Fully Homomorphic Encryption to perform data matching calculations. Fully homomorphic encryption can perform arbitrarily complex operations on encrypted data without a decryption key to achieve secure plaintext computations.

Let the encryption algorithm be:

$$\mathrm{E}(x) = \mathrm{c}_{\mathrm{x}}$$

The decryption algorithm is:

$$\mathrm{D}(x) = \mathrm{p}_{\mathrm{x}}$$

Limited by the performance of the homomorphic encryption algorithm, LemoChain chooses the square of the Euclidean distance to calculate the matching degree. The optimal match calculation formula is:

$$\mathrm{b}^{\backprime} = \mathrm{NN}(\mathrm{a}, \mathrm{B}) = \mathrm{D}\left(\min_{i=1,\cdots,n} d\left(c_a, c_{b_i}\right)\right) = \mathrm{D}\left(\min_{i=1,\cdots,n} \sum_{j=1}^{t} \left(c_{a_j} - c_{b_{ij}}\right)^2\right)$$

After calculating b ′ via the above formula, the inquirer obtains the best matching target. During the entire matching process, the proxy computing node and the inquirer cannot access other user’s data before encryption, and the privacy of the user data is ensured.

6. Smart Contract

A smart contract is a digital version of a traditional contract. Once written, it can be trusted by all parties, without requiring trust between those parties. The terms of the contract are final and therefore cannot be changed. This idea was proposed back in 1994 by cryptographer Nick Szabo, but the full potential was not widely recognized until the emergence of blockchain technology. Essentially, a smart contract is a computer program running on a blockchain database that can be triggered by preprogrammed conditions. Blockchain technology brings a decentralized, unchangeable and highly reliable system for an extensive range of applications. Smart contracts are one of the most important features of the blockchain and a key factor in its reputation as disruptive technology that is revolutionizing our social structure.

Lemo’s smart contract supports Java, C/C++, Python and a range of other coding languages. All smart contract source code is compiled into bytecode to run in the virtual machine.

The use of Sandbox technology has been implemented to achieve a complete isolation of affairs and limit access to computing resources, whilst maximizing performance and security.

Lemo’s smart contract virtual machine is built on a LLVM (Low Level Virtual Machine)–based compiler architecture. LLVM supports JIT (Just–In–Time Compilation) technology, which can dynamically compile and execute the generated machine code according to the users' requirements, which can significantly increase the execution speed of dynamic languages and maximize the performance of hardware. Based on LLVM's powerful three-stage design, future Lemo smart contracts will also support JavaScript and other more languages, and developers who are most comfortable with different technical backgrounds will develop smart contracts. Smart contracts include the four parts of contract registration, triggering, execution and cancellation:

6.1 Contract Registration

Contract registration is the process of storing the consensus in the blockchain after processing the user-written contract security check. Users need to consume gas according to the amount of code required to register a contract.

6.2 Contract Triggering

Contract triggering is the process of triggering contract execution through external conditions after contract registration. It supports timing triggering, event triggering, transaction triggering, and other contract triggering methods. Timing trigger refers to the process of automatically triggering the contract call after the node triggers the time consensus after meeting the preset time in the contract. Events, transactions, and other contract calls are new requests that trigger contract execution during the consensus process.

6.3 Contract Execution

Contract execution is the complete process of running the contract code in an external environment, including the contract structure mirroring environment, code execution, the implementation of state changes in the implementation of the code and exception handling of the consensus. There is a special message call named a proxy call. Except for the code of the target address being executed in the caller’s context, everything else is the same as the message call. This means that the contract can dynamically load code for other addresses at runtime. Only the code is obtained from the caller, this allows us to easily package code into libraries and reuse them in other contracts. For example, to implement a complex data structure, reusable code can be applied to contract storage.

6.4 Contract Cancellation

Cancellation is only necessary to clean up a contract that has been executed, expired or faces changes in business requirements that are no longer needed. The cleanup process requires a multi-node consensus before it can be completed. The only way to remove code from the blockchain is to have the Lemo contract perform a self-destruct operation on its address. The remaining balance under this account will be sent to the specified target, and the storage and code will be removed from the stack.

Lemo provides some of the standard contract implementations. Including the consistency check of assets, automatic integration, multi-signature, automatic settlement and other relatively simple logic of the contract. Users can invoke or adapt these contracts to suit their own business needs. It can also be completely implemented by itself.

Application Layer Services

Lemo provides some of the standard contract implementations. Including the consistency check of assets, automatic integration, multi-signature, automatic settlement and other relatively simple logic of the contract. Users can invoke or adapt these contracts to suit their own business needs. It can also be completely implemented by itself.

1. Account System

In a decentralized blockchain world, the user’s possessions can only be mastered by themselves, no one person nor organization can steal money, and there is no possibility of it being stolen by server hackers. But in fact, most users can not properly manage their accounts private key. According to Deloitte, at least 37% of users forget the password when they log in and use the ”retrieve password” feature. Forgetting the private key on the blockchain will cause the property to disappear directly, and there is no way to get this property back. The total amount of bitcoin having disappeared has reached 4 million, accounting for nearly 20% of the total amount. Users have a very strong demand for secure hosting of private keys.

LemoChain’s account system addresses the mapping of user identities to blockchain addresses, user privacy confidentiality, and regulatory audit traceability issues. It allows users to use easy–to–remember usernames and passwords for access and implements OAuth2.0 authentication mechanisms. Third–party applications that obtain user authorization can easily obtain basic Lemo White Paper 28 user information without the management logic of implementing and maintaining user–accounts by themselves. This boils down to just a few lines of code in accessing the LemoChain ecosystem.

Based on the account system, LemoChain will provide some common business unit plug–ins, which can be rapidly integrated into developers DApp applications. This greatly shortens the project development cycle.

- Online Safe. Encrypting the private key and hosting the backup online. Can only be retrieved by the user.
- Contacts. Manages and maintains many token addresses held by users, as well as address information of recent transactions.
- Points system. Supports multiple dimensions and digitizes user behavior. Accumulation and summation serve as a measure of the user's loyalty and contribution to the platform and can be used as an operation method to motivate users.
- Credit system. Through some basic real–name authentication services, the user's initial credit is evaluated, and the assessment results are continuously revised according to the user's late performance. The entire assessment result will be written as a credit record in the blockchain, providing a strong credit basis for buyers and sellers of data transaction software.
- Authority Configuration. Allows the establishment of authorization relationships
- between accounts and accounts, accounts and applications. Create higher–level data flow control logic through permissions and licensing mechanisms.

2. Online Safe

The online safe is a secure private key hosting service provided by Lemo. It is designed to ease the security burden on users. First, the local client encrypts the user's private key and uploads it to Lemo's private key coffer. When the user's private key is lost, the encrypted private key can be retrieved by providing authentication information and decrypted locally. The private and password in the entire process will not appear on the Internet, nor will it appear in the Lemo server. Private key security is guaranteed. Only users themselves can decrypt private data stored on the network.

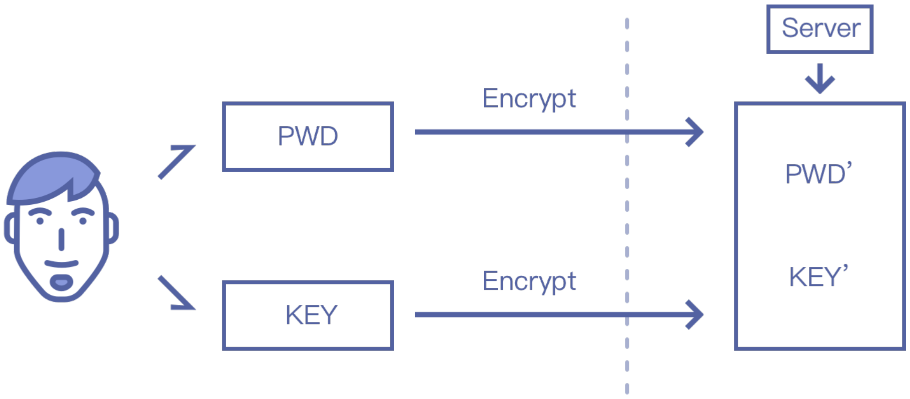


Figure 1 Hosted Private Key

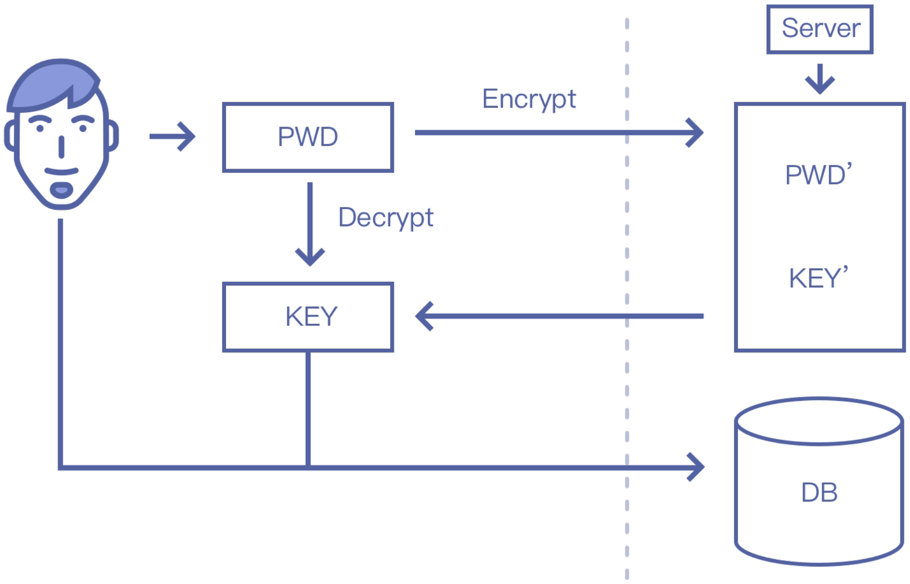


Figure 2 Retrieve the private key

To avoid possible malicious attacks causing data leakage from the server database. The key storage of the online safe deposit box will adopt three–party encryption technology, and the data will be encrypted and stored by an isolated third–party server. Even if the encrypted private key data is stolen, any available real information cannot be restored.

3. Data Exchange Template

In order to help developers realize the potential of data transaction functions of their respective industries faster, LemoChain has implemented a set of decentralized data transaction system applications based on the team's past experience in social networking and online dating. All the matching requirements are placed on the chain as transaction data, and the smart contract is used to automatically match the transaction. When the match is successful, both parties send data to decrypt each other's private key, ensuring that the user's privacy can be seen only if both parties match. The entire transaction process is open and transparent, privacy information will not be leaked to third parties, and exchanges cannot be concealed from fraud. This application shows developers the smart contract of LemoChain and the use of various services; It is the best developer learning material and can be used as a template to derive data transaction applications in other fields.

Application Prospect

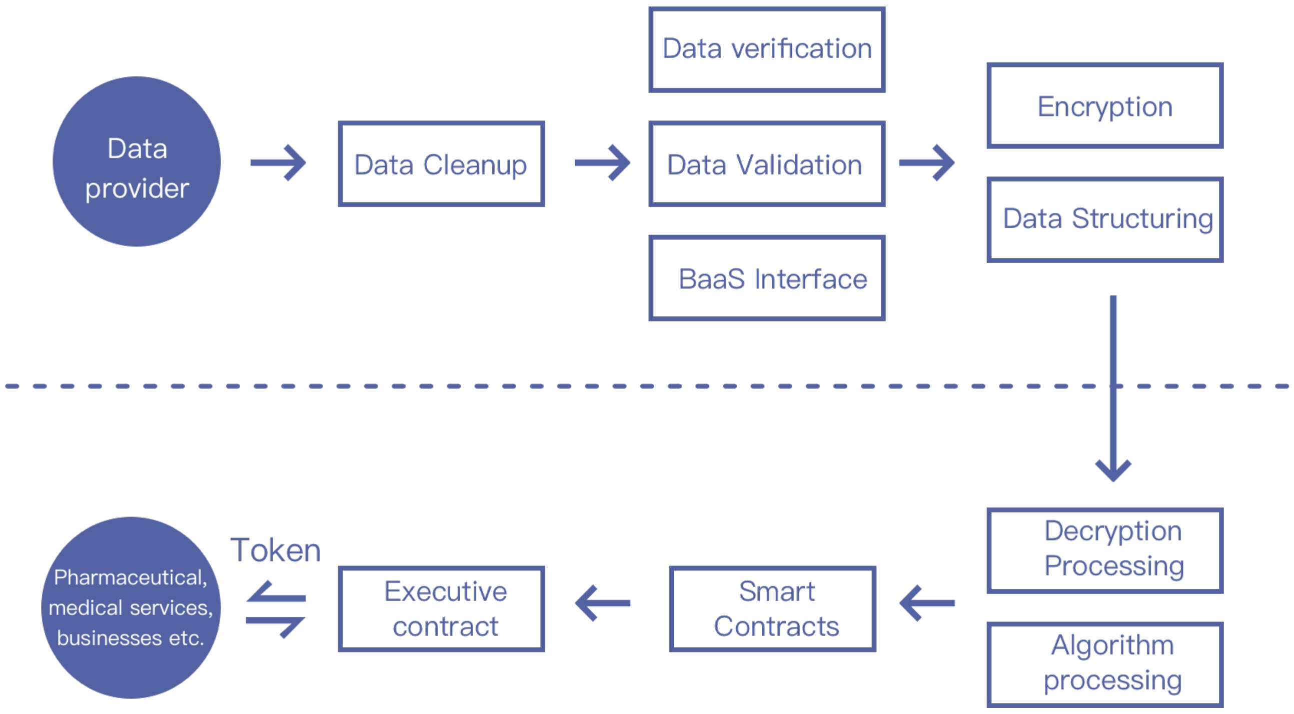
LemoChain built a platform based on a decentralized blockchain network that can support tens of millions of active users per day. Relying on decentralized and standardized data storage mechanisms to reduce participation costs in all aspects.
The eco-architecture of LemoChain is as follows:

- For developers: we open data exchange API, statistical analysis API, deep learning API
- For businesses: data transactions, algorithmic trading, enterprise DApps
- For Open Source Community: Blockchain Technology R&D Results Data Exchange

In this process, ecology determines user data.

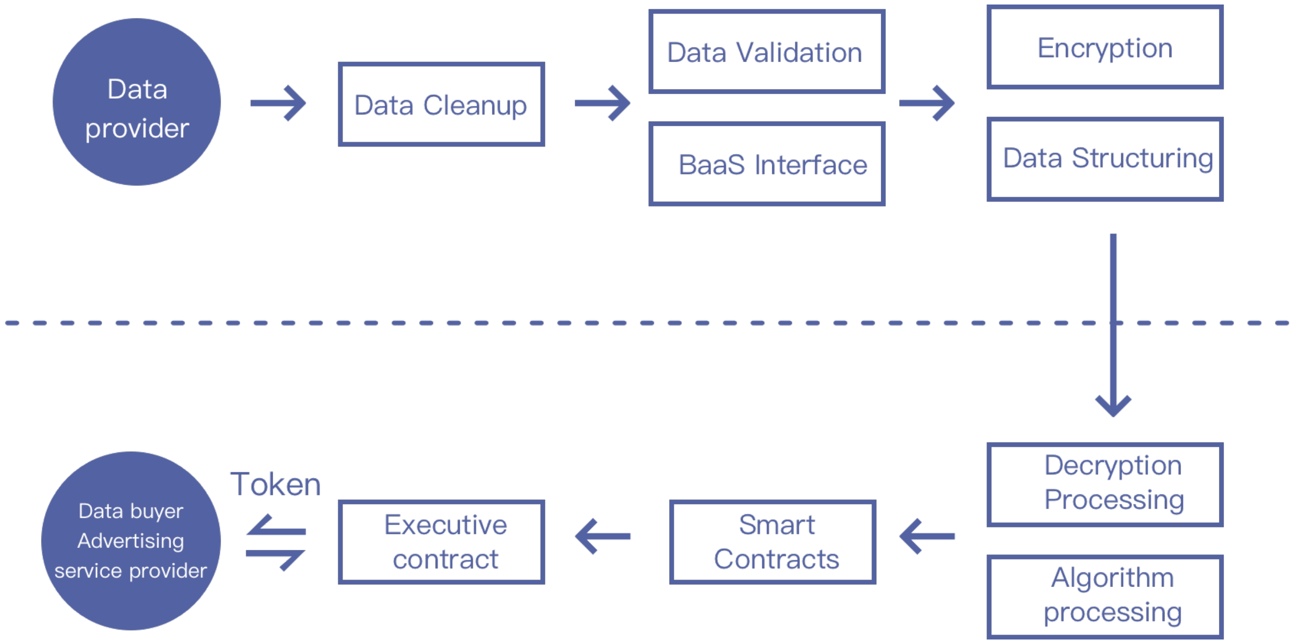
1. Application Scenario one: Health and Medical Data DAPP

Through health data DAPP, personal health data can be securely protected, data can be shared, and users can easily use it. Individuals upload their own health data to profitable data platforms and can obtain tokens. Data providers such as medical service providers and pharmaceutical companies can directly obtain data from profitable data links, and there are huge application scenarios in the direction of new drug development, testing, and precision medical care.



2. Application Scenario Two: Dating

Users submit their personal data and their friends' needs through dating and dating DApps. Individuals can encrypt and store their social data, and then digital assets can be uploaded to the profitable platform and the tokens can be obtained. Advertisers and their other dating partners need to pay tokens to obtain this data. They can be used for precise matching of dating friends and precise targeting of advertisements.



Technical Roadmap

In order to continue to promote the application and ecological establishment of LemoChain, the Foundation has made the following technical time planning:

2017.06
LemoChain Official Launch

2017.09
LemoChain Demo
Development completed

2017.12
LemoChain
Public Pre–Sale

2018.02
Establishment of Foundation
End of Pre–Sale

2018.04
LemoChain Test–network
Source Code Released

2018.08
Data Trading Platform based
on test network launched

2018.11
LemoChain publicly tests and
completes 3-month security review

2018.12
LemoChain Official Network
Launch and Launch Party

Reference

1. https://cryptovest.com/news/cryptokitties-burn-up-15-of-ethereums-gas/
2. https://bitshares.org/technology/industrial-performance-and-scalability/
3. https://blockchain.info/charts/n-transactions
4. https://etherscan.io/chart/bloktime
5. https://news.bitcoin.com/ethereum-blockchain-congested-cats/
6. GENTRY C. Fully Homomorphic Encryption Using Ideal Lattices[C]//STOC '09. [s.l.]: ACM，2009：178
7. http://fortune.com/2017/11/25/lost-bitcoins/