INTERNET OF VALUE
STORAGE INFRASTRUCTURE

Yellow Book

V 0.1.0

Lambda

lambda.im

Lambda

CONTENTS

Lambda

ABSTRACT

Lambda is a safe, reliable and infinitely scalable decentralized storage network, which achieves the decentralized storage of data, the integrity and security verification of data, and the operation of market-oriented storage transactions based on the Lambda Chain consensus network, so as to provide infinitely scalable data storage capacity for the next-generation Internet of Value.

Storage security is the core of decentralized data storage. Lambda uses the Proof of Space Time (PoST) algorithm to ensure data storage security, and Lambda Chain applies the Verifiable Random Function (VRF) + the Byzantine Fault Tolerance consensus algorithms to ensure the operational efficiency and reliability of the consensus network. At the same time, Lambda connects storage suppliers (miners) with storage demanders (users) through the exchange (marketplace), thus completing decentralized transactions through chains. In terms of storage, Lambda has first been used to provide a distributed data distribution and storage network. The specific storage forms will be gradually improved and applied to actual business systems.

Introduction

Technical Components

Lambda mainly consists of the following components:

Consensus Network: Composed of thousands of nodes, the Lambda consensus network acts as a consensus protocol based on VRF and BFT. LAMB token transactions can be completed in the network. It also acts as a connection between storage resource providers and storage users, to provide them with functions such as matching, billing, rewards, penalties, and data verification, thus ensuring that the decentralized storage is safe, verifiable, and flexible.

Storage Network: The storage and consensus networks in the Lambda system are logically decoupled. The heterogeneous storage network can provide diversified, decentralized or other types of storage capacity, but it must follow the Lambda data integrity verification protocol.

Marketplace: Lambda supplies service providers and users with trading capacity. Users rent and pay for storage space, while storage service providers obtain the appropriate service revenue. Transactions are completed through non-high-frequency operations, after which users can obtain real-time storage service capacity, without the need for real-time matching between storage and retrieval resources. This enables real-time access to resources, making Lambda a public BlockChain with an accessible application.

Development Kit: An important technical component of Lambda's storage ecosystem, the development kit provides universal storage access capacity, including upstream and downstream data interfaces compliant with the existing cloud storage interface specification, enabling users to seamlessly enter the Lambda system. The development kit also allows other storage solutions to be connected to Lambda, making Lambda a heterogeneous storage network that provides services for different storage requirements.

Process Overview

Lambda is a decentralized storage network that issues the digital currency LAMB, which is native to the ecosystem. The consensus network composed of validators is responsible for maintaining account books and obtaining block generation rewards. Storage miners can compete to be promoted to validators. They need to pledge LAMBs to the network to become candidate nodes for block generation. Validators have a dynamic update mechanism.

Storage and Retrieval Miners: To become storage miners, miners need to complete resource preparation and pledge LAMBs equivalent to the resources stored on the network. To become retrieval miners, miners need to directly pledge a small number of LAMBs to the network. Storage and retrieval miners can obtain LAMBs by providing services. Storage miners can be promoted to validators and earn block generation rewards from the consensus network by accumulating credits and storage quantity. Storage miners who have not yet been promoted to validators also have the opportunity to obtain certain block generation rewards.

Storage and Retrieval Transactions: Storage and retrieval miners enter the built-in exchange （marketplace） after completing the pledge. Lambda's built-in exchange pools and merges storage miners based on the price of pending orders and the storage quantity. When customers place purchase orders, the system automatically completes matching based on the price, storage space and other conditions, and customers' consumption funds are integrated and hosted in the network. The consensus network carries out regular fund payment, based on the amount of services actually provided by the storage and retrieval miners.

Proof of Storage (PoS): Storage miners regularly provide PoS to obtain the storage fees paid by users. The PoS generates challenge parameters based on the random challenge fields in the previous challenge blocks. It generates the proof set over a period of time through the LAMB-PDP algorithm, then broadcasts the proof set to the network to be verified by the validators. They mark its status and apply consensus protocols to confirm that the result has been stored in the BlockChain to be used for rewards, penalties, order restoration, fee payment, etc.

Terminology

Validator

As consensus candidate nodes, validators are responsible for packaging transactions, generating consensus-based blocks, verifying PoS, and executing contracts to match storage and retrieval orders, and clear fees. There are two ways to become a validator – one is to contribute to the Lambda ecosystem in the early stage, and the other is to be promoted to a validator from a miner. However, the former kind of validator eventually has to log out.

StorageMiner

The nodes that provide storage services are called storage miners. They become miners by pledging LAMBs. Storage miners can earn storage revenue by providing storage capacity and PoS for demanders, and also have the opportunity to become potential validators.

RetrievalMiner

Retrieval miners can access the system at any time to provide resource download services for users by pledging a small number of LAMBs. Their income mainly comes from charges based on the amount of data retrieval.

LAMB-PDP

The Provable Data Integrity of Lambda is based on Provable Data Possession (PDP) and Proof of Retrievability (POR). Lambda optimizes publicly verifiable schemes so that the challenge input of PoS is generated by the parameters on the chains. Storage miners perform predefined proof functions to generate a well-proven chain of evidence, which is submitted to the network for verification by the validators that generate blocks.

Local Node

Storage miners need to join the distributed storage system, and run the Local Node to join chains and perform operations on them, such as pledging, order submission, and fee payment verification. The Local Node can be a full node participating in the complete verification of ledgers, or a light node used only to complete operations on chains.

Sector

Storage miners use sectors as the units to divide up storage resources and register them on chains, and to carry out data integrity verification. Users take order files as the units for purchasing storage space from the system, while storage miners charge fees based on the actual amount of data stored, with the payment cycle usually equivalent to N (a variable parameter) block generation cycles. A sector is a virtual data structure and the organizational form of data stored at the miners' location.

Order

Storage customers purchase storage space with sectors as the unit, i.e. /GB/C is the unit price, where C is a variable parameter of the system. Generally, there are M block generation cycles, or M days, months, or years.

Piece

Generally, large customer files need to be sharded and uploaded to storage miners, where each piece has a fixed length of L, and L is a system parameter. L can be modified later by communities, in accordance with the system requirements. If a file is shorter than L, it will be charged according to L.

ProofSet

Storage miners collect storage proof (chain data) with sectors as the units. The proof set is broadcast to the consensus network to be verified and packaged by validators and stored in BlockChains. Proof sets also require the auxiliary storage of storage miners to reduce communication traffic throughout the system.

Wallet

Lambda wallet contains the LAMB transaction function, LAMB balance query function, user purchase order history and validity period display function, miners' sector pledge and usage information, fee payment information, etc.

ConsensusNode

All validators can become consensus nodes, which are temporary roles, subject to re-election through VRF during each block generation cycle.

ValidatorTable

The validator table keeps a record of all validators that are valid for a certain period of time. The initial validators are initialized to be entered when the network starts. The subsequent addition and elimination of nodes are modified by community governance contracts, and are synchronized to the entire network through consensus.

SystemContracts

System contracts are built-in, with several default contracts in the Lambda system that are automatically executed when blocks are generated, including community governance, default exchange, fee payment, and order restoration.

SpaceID

The explicit storage identifier Space ID can be purchased in the built-in trading centre. It can only be linked to certain storage space of the miners. Space IDs have priority matching rights during order matching by the storage exchange. A Space ID represents the right to match a certain amount of storage. The total supply of Space IDs in the system is related to the total storage space of the miners, and Space IDs can be traded and transferred among miners.

Exchange

Here the Exchange refers to the storage or retrieval trading market. The Lambda system supports multiple exchanges, which are operated by various independent third parties. Operators can define their own business rules, and attract more miners and users to their exchanges to accrue more revenue.

Transactions and Contracts

Transactions

Transfer transactions: LAMBs are transferred among different accounts, covering transaction volumes, handling fees, etc.

Space pledge transactions: miners use a certain number of LAMBs to register the storage space to be sold into the network, covering space, pledge amounts, etc.

Space pledge revocation transactions: miners can cancel their pledged space and reclaim their LAMBs after reaching the minimum pledge cycle, covering pledge IDs and other data.

Validator pledge transactions: miners that meet certain criteria can pledge a certain number of LAMBs in order to be promoted to validators and participate in a consensus network.

Validator revocation transactions: validators act to revoke their consensus node identity and reclaim their LAMBs.

Space sales transactions: miners publish data about their pledged space at the exchange, including the prices, amount, and validity periods.

Traffic sales transactions: retrieval miners sell traffic at the designated exchange, including prices, etc.

Space purchase transactions: users act to purchase spaces at the designated exchange, including the space amount, prices, validity periods, security levels, etc.

Traffic purchase transactions: users act to purchase traffic through agents at the designated exchange.
Key purchase transactions: miners purchase keys to increase their credibility, and gain a higher priority for matching orders.

Encapsulation and storage resale transactions: miners can resell already stored orders to collect more storage space, thus maximizing the benefits.

Contracts

Miner pledge contracts: complete the miners' space pledges and withdrawals and conduct the corresponding security inspection.

Validator pledge contracts: complete the validators' pledges and update the validator table.

Storage exchange contracts: can be operated by different parties to match the storage demands between miners and users.

Retrieval exchange contracts: can be operated by different parties to match the retrieval demands between miners and users.

Encapsulation and storage resale contracts: complete the resale of storage space and maintain profit distribution relations.

Loan contracts: allow LAMB holders to receive interest by depositing LAMBs or providing loan services to miners or users.

Consensus Protocol

Validators

Lambda uses validators to maintain the consensus network. The number of validators in the system = the initial number of officially operated validators + the number of validators operated by other friendly third parties + the number of miners promoted == 1024. Among them, the number of miners promoted will eventually reach 1024, and the other nodes will log out in an orderly manner.

Storage miners need to pledge sectors <SIZE, LAMB> to formally become candidate miners. After orders are matched, the sectors are encapsulated and the PoSTs can be found on chains. The candidate miners officially become storage miners, and then validators when they meet certain promotion requirements.
The validator table is updated every M (an adjustable parameter) cycles.

Validator Promotion Mechanism:

1) Storage miners initiate the Validator Login transaction based on their own storage space, comprehensive credibility, resource preparation status, and willingness to apply to become validators. The following aspects of login transactions must be checked. If these conditions are not met, a certain pledge amount must be burned to prevent frequent attempts:
a) Check the pledge amount of the Validator Login.
b) Check the actual storage demand of the storage miners, which must meet these requirements:

$$Space_{M_i} > \frac{\sum_1^N Space_{M_i} * 50\%}{1024}$$

2) Carry out traversal checks on the above storage miner list, with no PoST failure during N cycles.
3) Sort the storage miner list by the amount of space, setting the value to TOP (x), where X refers to the number of validators that must be added.
4) The updated validator table is sorted by algorithm, Sort Validator (HASH (Validator ID+Block Height)).

Validator Logout Mechanism:

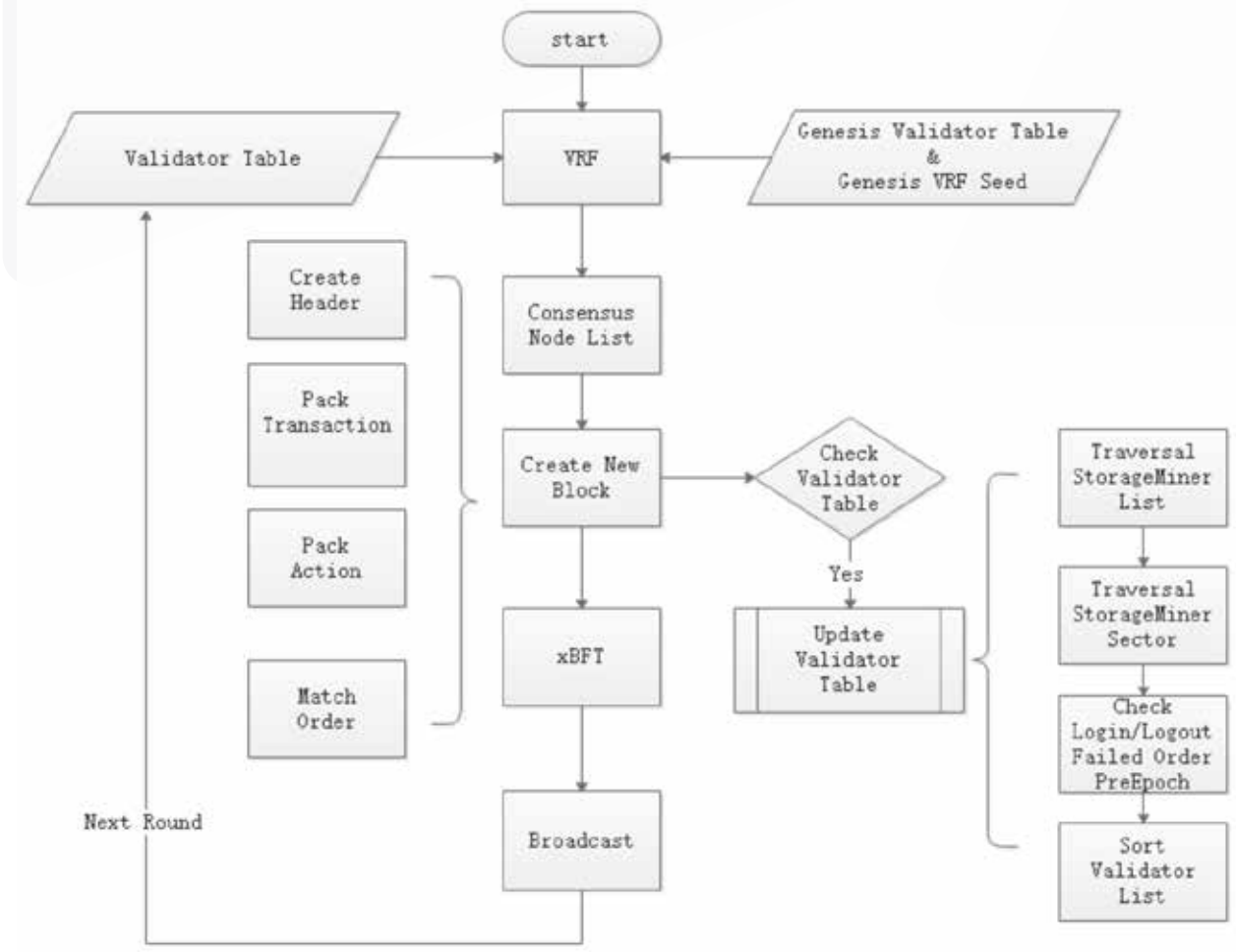
1) Carry out traversal checks on the validator table, where the last validator status should be Validator Logout.
2) The status is Login, but the Login storage capacity requirement has not been met.
3) Carry out traversal checks on the above validator table, with M sector proof failures occurring during N cycles.
4) Carry out traversal checks on the above validator table, including failure to participate in consensus or other malicious behaviour occurring during N cycles.

Lambda Consensus

Consensus input conditions:

- Height (Block Height)
- Max Block Interval
- Max Block Size
- Election Seed (VRF Random Seed)
- Validator Table
- VT Update Cycle
- Difficulty (time parameter used to dynamically adjust the block generation cycle)
- Block Reward
- Reward Curve

The consensus process is shown below:



The Lambda Consensus algorithm is based on the current validator table. For the update strategy of the valida–tor table, please see section 3.1.

The basic steps of the Lambda Consensus algorithm are as follows:

1) The VRF Parameter is made up of the new block number, the ConsensusNode Hash, the Hash, and the VRF Seed of the current block data.

2) The VRF function is used to generate a new round of VRF Seed based on the VRF Parameter.

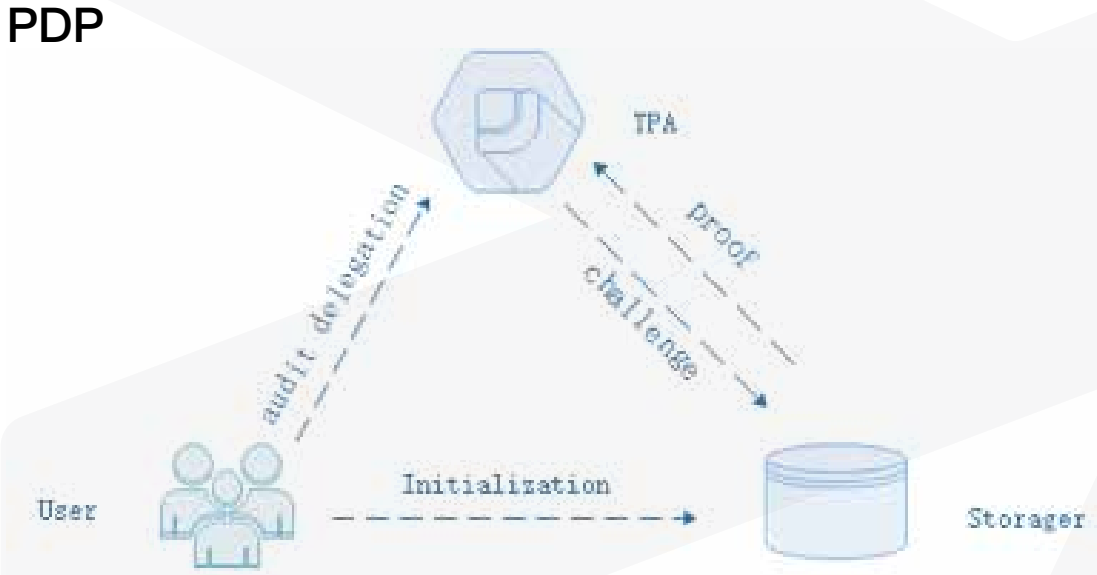
3) ConsensusNodeList = GetConsensusNodeList (ValidatorTable, Index (VRFSeed), TotalConsensusNode) is generated based on the new round of VRF Seed and the Validator Table.

4) ConsensusNodeList is applied to verify the preselected blocks and obtain the blocks with the highest weights for voting. The weights are related to the number of validator sectors and node Hashes.

5) After voting, the BFT consensus is reached on the blocks and the blocks are broadcast.

Data Integrity

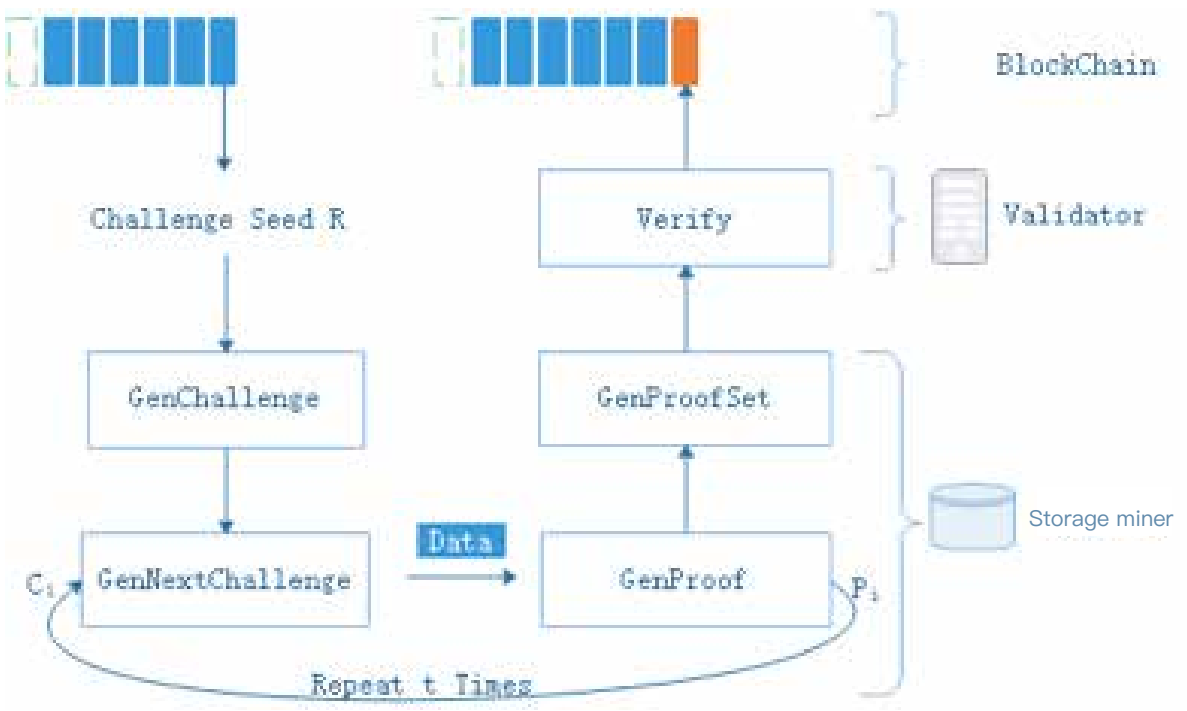
PDP



a) Users generate and sign a tag for each piece of data to be outsourced;
b) Third Party Administrators (TPAs) randomly launch challenges against one or more blocks of outsourced user data, with the challenges containing random numbers generated by the TPAs;
c) Storage miners obtain proof through a calculation based on the content of the challenged data blocks, tag information, challenge information and a random number generated by them;
d) The TPAs take the challenges, proofs and users' public keys as parameters, and through bilinear mapping functions, they verify whether the storage miners own data.

LAMB-PoST

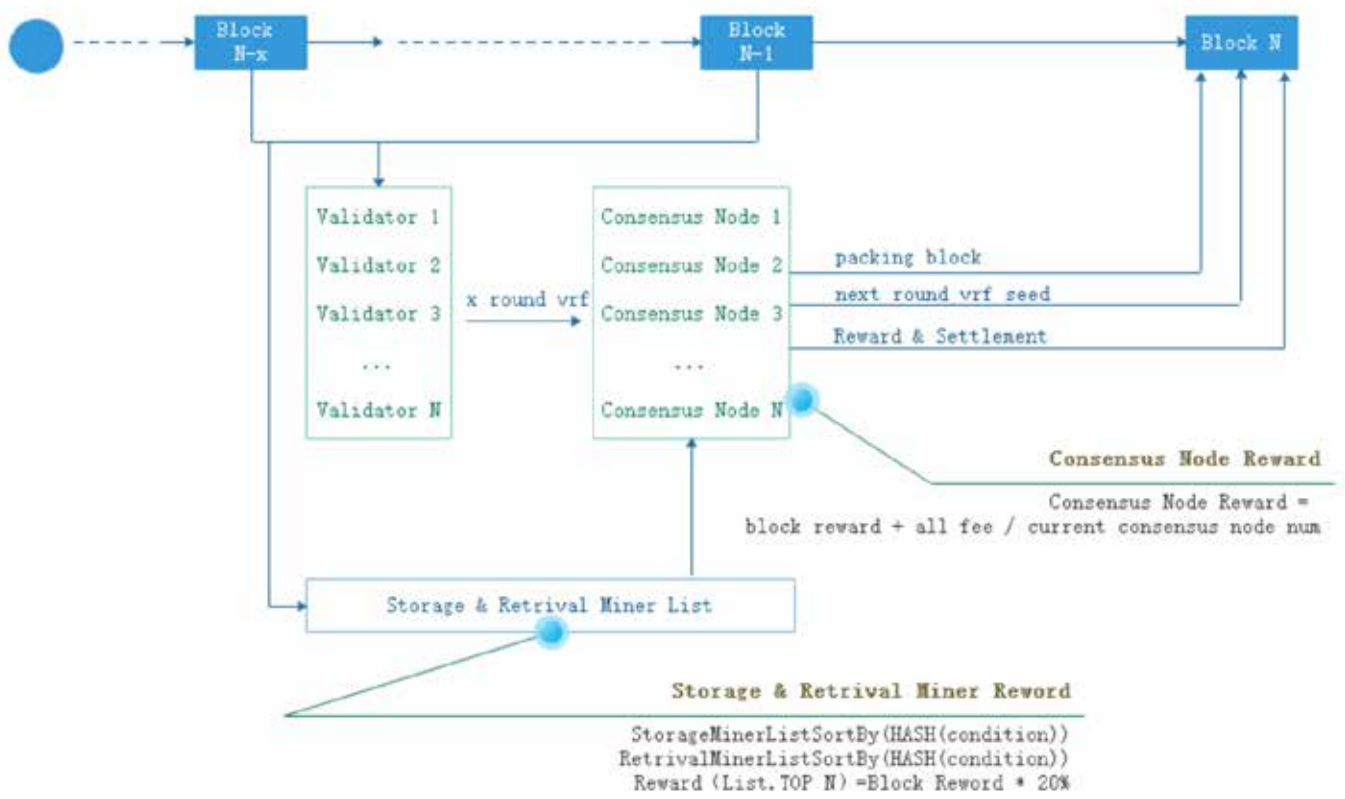
The PDP scheme faces the following problems:
1) The generation of the random challenge information depends on the TPAs, which makes it uncertain whether the random challenge value is safe or not.
2) When a challenge is launched, in order to improve the detection rate, multiple blocks of data must be detected at once, and random challenge parameters must be generated for each block, which greatly increases the communication complexity of launching a challenge.
3) Interactive challenges require highly synchronous networks, and the high frequency of the <challenge, prove> interaction causes an increase in the network load on the system.
4) As storage miners store a large number of fragmented orders, if each challenge is launched for one order, the number of challenges will greatly affect the load on the system.
5) If a piece of data is saved in multiple copies, or matched with multiple miners, this can lead to sybil attacks and the false storage of multiple copies.
6) Although we can prove which files are currently stored by generating a proof, how can we ensure that the files are stored between two detection cycles?
7) The LAMB-PDP scheme needs to solve the above problems based on the PDP algorithm, and the basic process is shown below:



1) Miners encapsulate multiple file fragments into a sector and record the tag information, public parameters, fragment/file indexes in BlockChains.
2) Miners obtain the random challenge seed "R" from the latest blocks, and calculate the Sector "S" that needs to be challenged this time. Miners must ensure that all sectors submit PoS within M cycles, to gain the corresponding revenue.
3) Miners generate challenges based on GenChallenge(R , S , Index) –> challenge.
Miners generate proofs based on GenProof(data, challenge, tag) –> proof.
4) Miners use GenNextChallenge(proof_i) to generate , and generate the Proof Set through t times of proof.
5) Miners submit the proof set to the network and wait for it to be verified and packaged by the validators.
6) Miners must continuously provide the PoST for a sector. If the PoST is not received within T cycles, the sector is considered to be lost.
7) The sector is marked as lost, the corresponding penalty is updated, and the order data contained in the sector is rematched or restored.
8) The restoration activity can be carried out by users.

Incentive Mechanism and Cost Model

Incentive Mechanism



Consensus Incentives

Consensus nodes are randomly generated in the validator tables through VRF. When a new block is created, seeds are selected for the next block consensus node and written into the new block. There are a fixed number of con–sensus nodes in each block generation cycle. If a consensus node does not participate in the consensus process, it will not participate in the distribution of rewards for block generation. If this behaviour occurs repeatedly, it is withdrawn from the validator list. The rewards for the block generation of consensus nodes are as follows:

$$R_i = [(BlockReward*50\%*K_S + BlockReward*50\%*K_d)*70\% + Fee]*\frac{Space_{C_i}}{\sum Space_{C_i}}$$

The initial value of BlockReward is 160LAMBs

R_i *refers to the rewards of the i consensus node*

C_i *refers to the i consensus node of this round*

$Space_{C_i}$ *refers to the current storage capacity of the i consensus node of this round as a storage miner*

K_s refers to the reward coefficient for the block generation of network – wide supply K_d refers to the reward coefficient for the block generation of network – wide demand

In the initial stage, each block is rewarded 160 LAMBs. After that, block rewards are reduced by half every 4 years, and theoretically, the rewards will end in about 20 years. The rewards for block generation are limited by the development of the Lambda ecology, and jointly determined by the storage supply and demand of the whole network, as follows:

$K_s coefficient\ table$

Network-wide Supply	K_sCoefficient	Total Number of Validators	Storage Miners' Reward Number	Pledge amount per capacity	Number of Rewards
<1PB	25%	64	64	5	20
1PB-10PB	50%	256	256	6	40
10PB-100PB	75%	512	512	7	60
100PB-1000PB	100%	1024	1024	8	80
>1000PB	100%	1024	1024	10	80

$K_d coefficient\ table$

Network-wide Demand	$K_d coefficie$	Total Number of Validators	Number of Rewards
<0.5PB	25%	64	20
0.5PB-5PB	50%	256	40
5PB-50PB	75%	512	60
50PB-500PB	100%	1024	80
>500PB	100%	1024	80

Example 1: The network–wide supply capacity is 5 PB, the network–wide demand capacity is 1 PB, and the election condition for validators is 5PB/2/256 = 9.7TB. 10 consensus nodes are selected each time. The samples are assumed to be [40T, 20T, 12T, 60T, 64T, 72T, 16T, 12T, 20T, 12T].

The actual storage capacity of miner A is 40 TB data, and the rewards for block generation are [160/2*0.5+160/2*0.5]*(40/328)*0.7=8.82 LAMBs. If the samples of consensus nodes are unchanged, about 2976 LAMBs will be gained per day

Example 2: The network–wide supply capacity is 120 PB, the network–wide demand capacity is 60 PB, and the election condition for validators is 120PB/2/1024 = 117TB. 10 consensus nodes are selected each time. The samples are assumed to be [200T, 300T, 120T, 140T, 380T, 210T, 130T, 170T, 260T, 320T].

The actual storage capacity of miner A is 200 TB data, and the rewards for block generation are [160/2*1+160/2*1]*(200/2230)*0.7=10.045 LAMBs. If the samples of consensus nodes are unchanged, about 847 LAMBs will be gained per day

Miner Incentives

There are storage and retrieval miners. To ensure that the Lambda network attracts geographically distributed resources and more storage and retrieval nodes, miners can gain more returns before being promoted to validators, and storage and retrieval resources can be continuously upgraded. Non–validator miners can also get rewards for block generation from the system, with the rewards evenly distributed to N nodes (Block Reward * 30%), where N (about 5% of the number of active miners) is a system parameter to be determined.

$$R_{m_i}=(BlockReward*50\%*K_S+BlockReward*50\%*K_d)*30\%*\frac{Space_{m_i}}{\sum_0^{5\%}Space_{m_i}}$$

Requirements for maintaining the miner list:

1) The storage miners have pledged sectors, while the retrieval miners have pledged LAMBs;

2) The miners have received orders or submitted PoSTs within M block cycles, and the retrieval miners have provided retrieval services for users;

3) The list is sorted by miner ID and Block HASH, and the first N miners are rewarded.

MinerListSortBy (HASH(node id + block hash))

Example 3: The network–wide supply capacity is 5 PB, the average storage capacity of non–validator miners is 2 TB, and there are 1,250 non–validator miners.

The storage capacity of non–validator miners is 2 TB, 8640*256/1250=1769.472

[160/2*1+160/2*1]*0.3=24/256=0.09375

1769.472*0.09375=165 LAMB/Day

Cost Model

Charge Composition

Transfer Transaction Charges

Fees are charged for handling all transfer transactions in the Lambda network. The amount of these fees is determined by the current load of the network. Whether the charges are sufficient determines the packaging priority of transactions. The transaction fee is based on the factors of price and quantity. These charges are paid by the transfer initiators, and the beneficiaries of the charges are the validators in the consensus network.

Storage Order Charges

Storage order charges refers to the fees charged for conducting transactions by matching miners' sales orders and users' purchase orders. The charges are equal to a fixed percentage of the order amount, which is a system parameter determined when the main network is started. These charges are paid in settlement of the storage revenue of miners. The beneficiaries of such charges are divided into two parts: some are validators in the consensus network, while most are operators in the trading market. Operators are third parties that may be publicly registered and operated.

Retrieval Order Charges

Retrieval order charges refer to the fees charged for conducting transactions between miners and resource download users. The charges are equal to a fixed percentage of the order amount, which is a system parameter determined when the main network is started. These charges are paid in settlement of the retrieval revenue of miners. The beneficiaries of such charges are operators in the retrieval trading market. Operators are third parties that may be publicly applied for and operated.

Pledge Cost Composition

Storage Space Pledge

The total storage space pledge amount tends to be 40% of the number of Coins in circulation throughout the network. This process may take several years. With the gradual increase in miners' pledge spaces, the pledge amount per space will gradually decrease. The pledge amount per space is related to the cumulative total of miners' own pledges, and the correlation curve and calculation formula are as follows:

Incomplete storage pledges can be revoked, while closed storage pledges can only be revoked after applying for revocation and after the system rematches the order and successfully transfers the data, with a minimum pledge duration of no less than four weeks.

Validator Pledge

Validator pledges require that the node, as a miner, meets the minimum storage requirements for the validator. At this point, storage miners need to run an independent consensus node program and pledge a certain number of LAMBs of a fixed amount. The validator can revoke the pledge, and the pledged LAMBs can be refunded within four weeks. The pledge amount may also be deducted or burned in the event of behaviour such as double-signature, staying offline, or failure to vote.

Retrieval Miner Pledge

Retrieval miners pledge a certain number of LAMBs to prevent malicious attacks. The pledge amount can be revoked in full in real time after retrieval miners revoke the pending retrieval orders and the retrieval order services are completed.

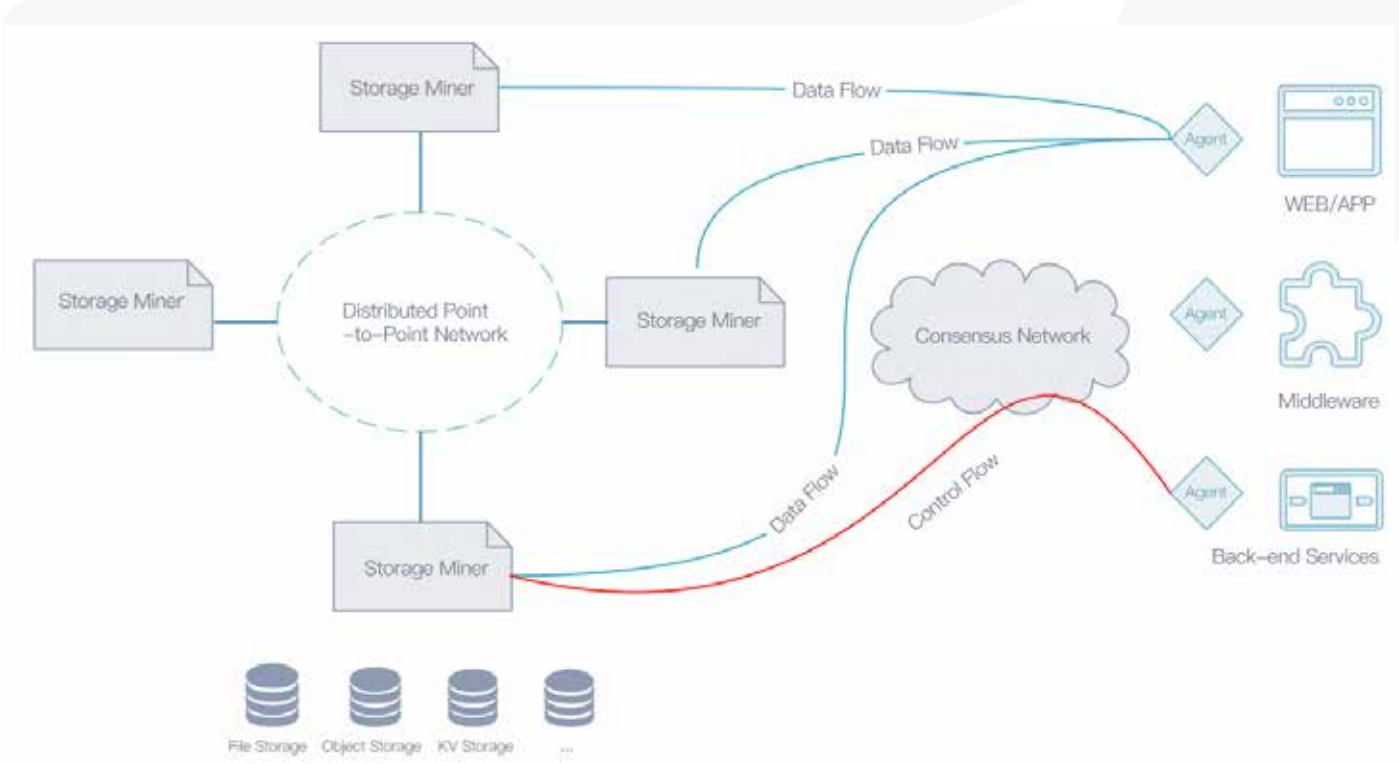
User Consumption Pledge

Users must pledge a sufficient consumption amount based on the real-time order amount, e.g. for 10GB of storage space for one year, the corresponding consumption amount must be pledged. After storage miners submit the PoS, the network will make the proportionate payment based on the users' pledged consumption. Users must store the pledged consumption for at least one week before revoking orders, and the unspent part will be returned to the users.

Storage Infrastructure

Overall Structure

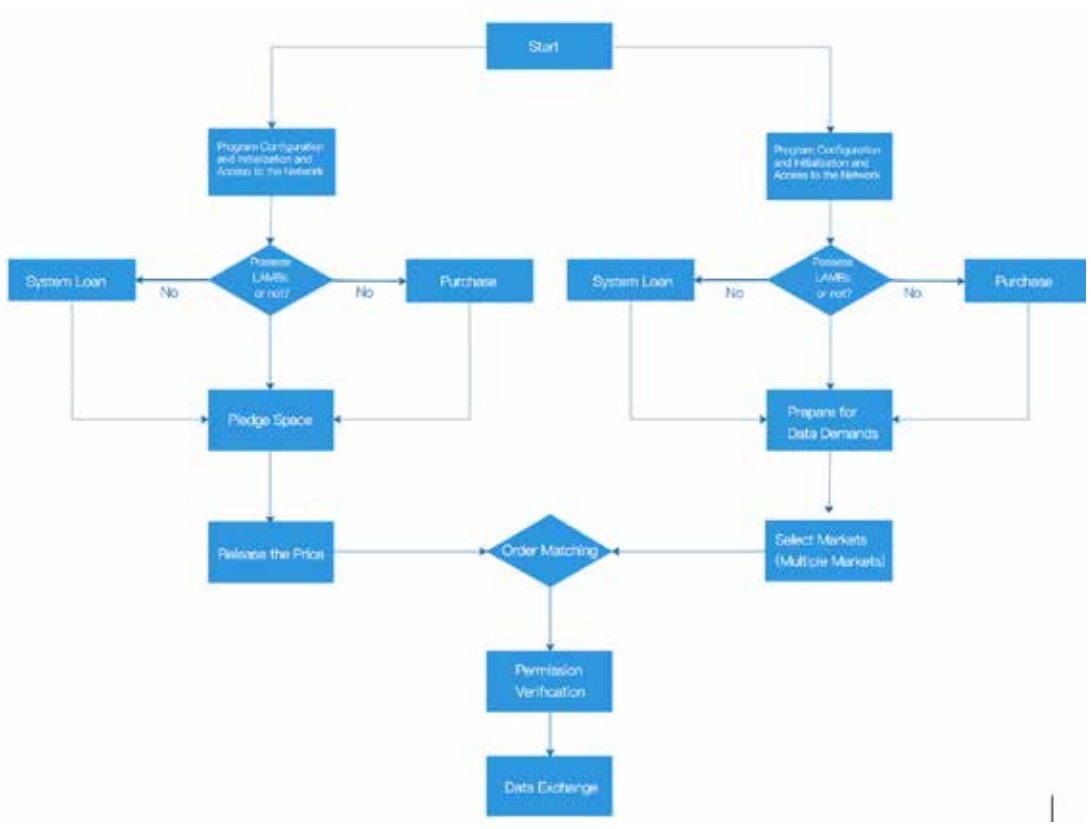
The Lambda storage network is primarily a DHT network. All storage miners are automatically discovered and added to the network through nodes. Routing information is stored throughout the DHT network and point–to–point data transmission is possible. The DHT network is the basis of the Lambda storage network, where a range of storage types can be built. However, all the technical details are encapsulated by accessing short agents, which provide the same experience for users as traditional data storage facilities. In the DHT network, only fragments of data are exchanged, and miners do not know the content and significance of the data.



Based on the distributed DHT network, Lambda has gradually developed different storage types, such as object storage, file storage, KV storage and partial relational storage, which also required the synchronous development of Agents to reduce the development challenge of using different storage types. Data on BlockChains are publicly accessible, which greatly limits the data usage scenarios. In order to extend the usage scenarios, Lambda provides an access control scheme based on multi–authority attribute–based encryption (MA–ABE), along with data encryption capacity and the option to remove attributes through agent encryption, thus authorizing secure access control through chains.

Business Process

IThe Lambda network marketplace connects miners with users. Miners enter the market after preparing resources, pledging, and releasing prices, while users prepare funds according to storage demand. Users can choose appro–priate trading markets before Agents automatically complete the pending storage and retrieval orders and order matching in the selected markets, then conduct permission verification and data exchange. The specific process is as follows:

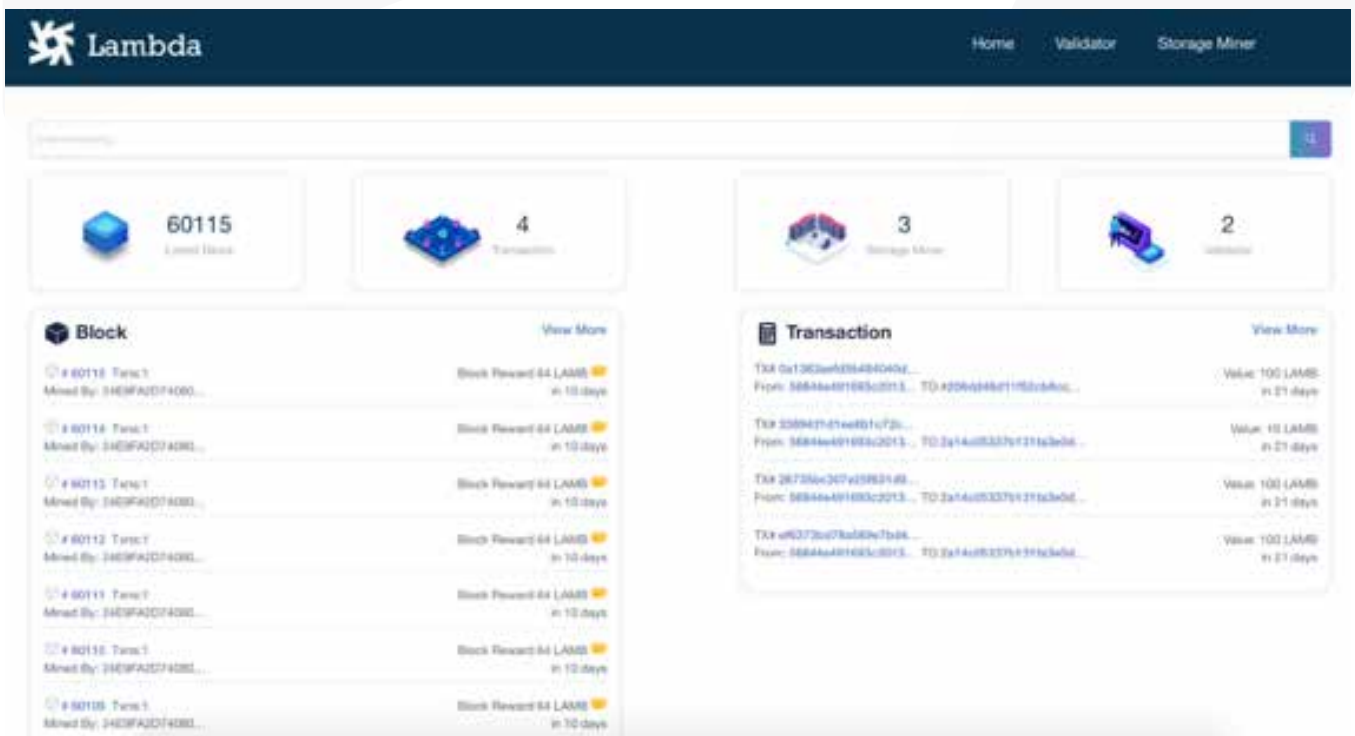


Visualization and Extension

Wallet

The Lambda wallet mainly contains a homepage view, miner view, consumer view, and validator view. Transaction functions such as balance, transfer, borrowing and lending functions can be completed in the wallet. Functions such as equipment registration, space pledging, pending sales orders, order tracking, pledge revocation, the removal of mining machines, encapsulation space transfer, and joining mining pools are available in the miner view. Operations such as tracking orders and payment information as well as acquiring access authorization are available in the consumer view. Functions including tracking the verification of validator qualification, application and revocation, and tracking block generation are available in the validator view. The wallet is an illustration for guidance, with the above functions available since its release. Lambda is also developing an Application Programming Interface (API) which can be developed and launched by third parties.

Browser



Follow–up tasks

- Detailed definition of the storage interface and the development of Agents;
- Providing capacity to store third–party operations in the retrieval market, and opening up contracting capacity;
- Improving access control and authorization;
- Adjusting and optimizing the PoS by storage type;
- Optimizing the performance of validator storage verification;
- Improving the digital retrieval market and real–time data flow.

Lambda

lambda.im