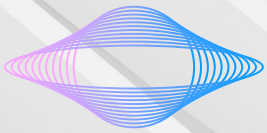

4EVERLAND

WHITEPAPER

4EVERLAND – PaaS of Web3.0

(V1.0.0)

ABSTRACT

4EVERLAND is a blockchain technology–powered cloud computing platform, designed to help with efficient Web 3.0 applications development. It features global acceleration, privacy protection, and distributed storage. Builting on consensus–driven Swarm on top of IPFS, 4EVERLAND significantly improves Data I/O efficiency and facilitates TEE–based storage challenge proofs. It achieves more stable content storage, more efficient content distribution, faster network access, file reading, and storage fraud prevention.

In addition, 4EVERLAND provides products and protocols such as DWeb hosting, decentralized gateway, decentralized domain name system(DDNS), digital marketing system, and data services to help developers access Web 3.0 in one click, which eventually leads to a globally accelerated, privacy–protected, link–perpetual distributed network and Web 3.0 infrastructure with IPFS.

Through unique cryptography and economic model, 4EVERLAND has deployed a series of nodes deployed around the world. If IPFS is a decentralized Hadoop, then 4EVERLAND is a decentralized AWS, starting with DWeb hosting, distributed storage, and gateway global acceleration, and gradually developing computing power to eventually help worldwide developers realize their aspirations in the Web 3.0 world in a highly efficient and secure way.

Mission:
To help the Internet make a smooth leap from Web 2.0 to Web 3.0

Vision:
To be the infrastructure for millions of Web3.0 developers and applications

Positioning:
A cloud computing platform for Web 3.0

1.INTRODUCTION

The Internet needs Web 3.0, a new generation of the Internet paradigm that advocates free identity, free contracts, and open assets. Blockchain technology provides a near-perfect solution to Web 3.0, with decentralized addresses as accounts, smart contracts for self-deployment and access, as well as tokens that flow autonomously and frictionlessly. The free market creates prosperity and blockchain business is growing rapidly, with the liberation of financial experience by DeFi attracting hundreds of billions of dollars. Moreover, NFT is empowering art, sports, and trendy culture that has brought millions of incremental users to the blockchain world.

However, there is still an important part of the Web 3.0 paradigm of blockchain that is missing. "I trust my Ethereum addresses, assets, and smart contracts, however, what if the DAPP interface I'm accessing disappears or is blocked some day in the future?","I have a collection of valuable NFT digital artifacts that point to a single node storage server, will the value of the NFTs disappear with the shutdown of the server?", or "What if my DEFI project is fast in Europe, but slow in Asia, affecting user experience?" Clearly, the centralized, license-based, single-node data service model cannot meet the needs of the rapidly growing number of Web 3.0 users for service security and the degree of freedom they require.

4EVERLAND is designed to solve this problem by providing a distributed, highly efficient, self-motivated, and low-cost data hosting gateway based on IPFS and the underlying technology of Ethereum. As free accounts, free contracts, and open assets are the foundation of Web 3.0, the development protocol built by 4EVERLAND will be the access layer protocol of the Web 3.0 world, liberating Web 3.0 developers from the constraints of centralization, improving the robustness and security of the service, and providing a globally accelerated, ever-lasting network service for users.

2.BACKGROUND

This section describes the key technology attributes of 4EVERLAND.

2.1 DISTRIBUTED FILE SYSTEM

The InterPlanetary File System (IPFS) is a protocol and peer–to–peer network for storing and sharing data in a distributed file system. It is a content–addressable peer–to–peer hypermedia distribution protocol where nodes in an IPFS network will constitute a distributed file system. IPFS employs the following core technologies:

- The data storage uses Merkle DAG to ensure that data is verifiable and the identical parts of files are not duplicated;

- Content addressing: all content of Merkle DAG is uniquely identified and verified by multiple hashes;

- DHT network and BitSwap protocol: IPFS uses the DHT network as a route to distribute data by exchanging data blocks between peer nodes. The BitSwap protocol is used to incentivize data preservation and ensure fairness among nodes. BitSwap is a credit system where each node keeps a ledger and peer nodes keep track of their (data exchange) balance, and the probability of sending data decreases as the debt increases.

4EVERLAND's data storage and transport layer are built on the IPFS network. Through a unique cryptographic and economic model design, a series of IPFS nodes are organized into a collaborative storage and gateway service that rewards contributing nodes and penalizes fraudulent ones.

2.2 SMART CONTRACT

A smart contract is a combination of a set of instructions on the blockchain, and a complete smart contract system consists of a smart contract programming language, a compiler, an instruction set, and a virtual machine to run them. Through smart contracts, one can define storage structures and write stored procedures inside the distributed ledger of the blockchain, hence, smart contracts make the blockchain programmable. Since the release of Ethereum smart contracts, many protocols and standards have been developed on this programmable blockchain, and they are all built on smart contracts, such as ERC20, ERC721, ENS, DAO, AMM, and various DEFIs, demonstrating how smart contracts can solve all kinds of trust problems in centralized systems in a unique and simple way. Smart contracts enable assets in 4EVERLAND to be built on a series of standards and protocols, providing a reference for 4EVERLAND to achieve decentralized governance. At the same time, AMM and various DEFI protocols will also be introduced into 4EVERLAND's core asset circulation system, ultimately making 4EVERLAND a liberal, fair and imaginative system.

2.3 DOMAIN NAME SYSTEM

DNS

The Domain Name System (DNS) is a service of the Internet that serves as a distributed database that maps domain names and IP addresses to each other, enabling people to easily access the Internet without having to remember complex IP addresses.

ENS/IPNS

A centralized domain name system cannot satisfy the emerging demands under the architecture of Web3.0. One needs a decentralized solution that maps domain names to IP addresses in a distributed ledger where its resolution does not depend on any centralized service. This enables more efficient access, such as ENS and IPNS. DNS makes it possible for content published in Web 3.0 to be accessed in the same way as in Web 2.0. Therefore, DNS will be an essential part of 4EVERLAND, which will extend the basic functionality of DNS with new record types and resolution methods to enable DNS to support content from 4EVER-STORAGE.

2.4 PROOF OF STORAGE

In the IPFS network, files are split into blocks and stored in several peer-to-peer nodes, improving the fault tolerance of the storage system. Therefore, in a decentralized file system where incentives are obtained through storage, Proof of Storage is necessary to prevent a Sybil Attack which could be utilized to improperly obtain tokens. Proof of Storage needs to confirm that a piece of content has been stored in a node's storage space for a certain period of time. There have been several storage proof solutions, such as Arweave(store files on-chain directly) and Filecoin (Proof-of-Replication-PoRep and Proof-of-Spacetime-PoSt). These solutions either have poor scalability due to large data storage synchronization or poor response time of the system due to high computing complexity. Therefore, 4EVERLAND will propose a lightweight Proof of Storage mechanism based on TEE technology that can be found in section 3.3.

2.5 LAYER 2

Ethereum is the most popular smart contract blockchain. Despite the success of its ecosystem, problems such as high gas fees and low transaction speed are increasingly emerging. As a result, Ethereum layer2 solutions such as ZK-Rollup and Optimism have been recently proposed. The main goal is to increase the throughput of the Ethereum mainnet and significantly reduce the transaction fee without sacrificing network security. Layer 2 solutions will also be considered to improve user experience and reduce transaction costs in frequent payment occasions.

3.DESIGN

3.1 TECHNICAL DESIGN

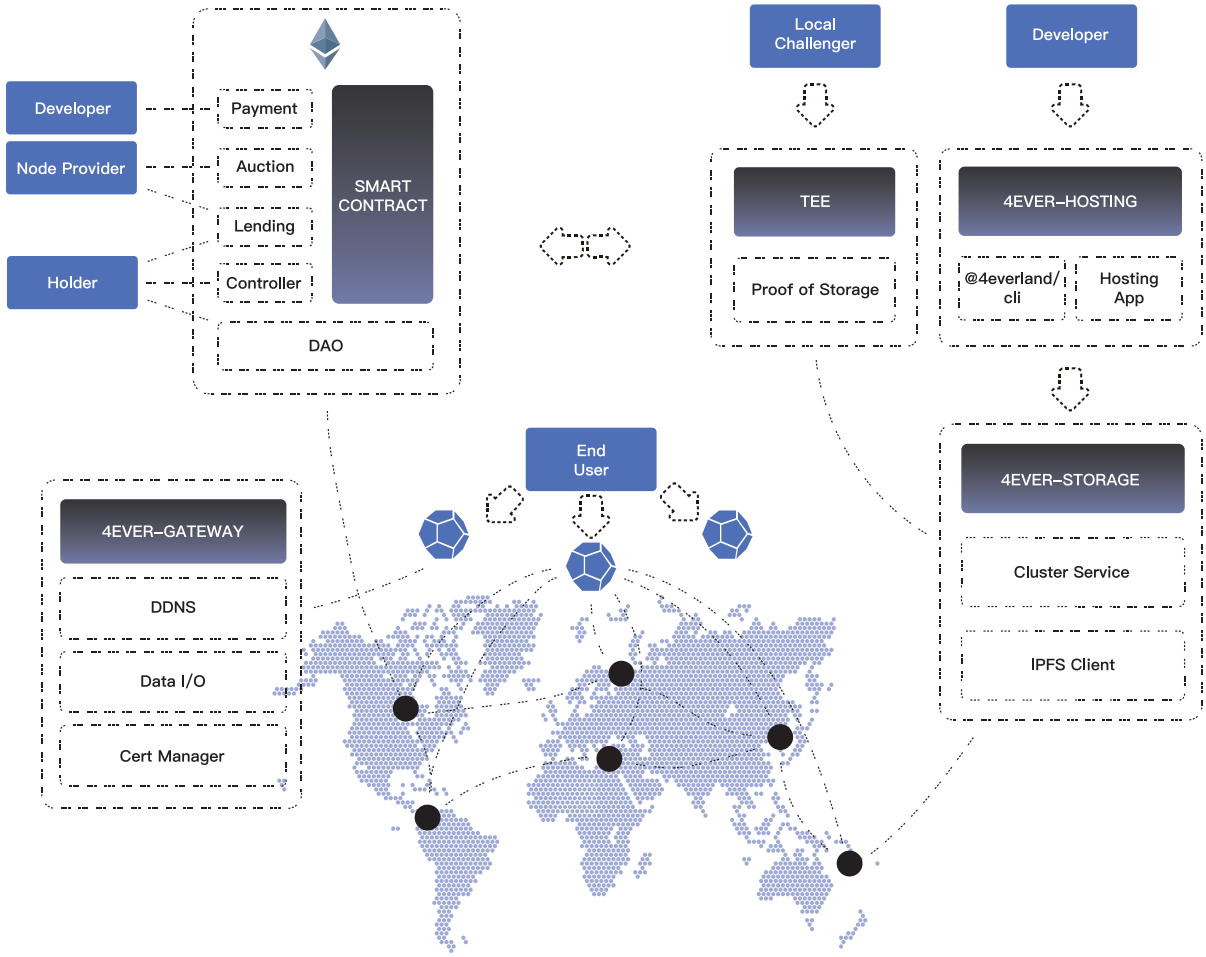

Figure 1: Overall Technical Design of 4EVERLAND

The complete technical architecture of 4EVERLAND is shown in Figure 1 and is divided into the following parts:

• **4EVER–STORAGE** (a client containing Cluster Service and IPFS nodes) is compatible with the IPFS protocol and automatically forms a cluster of nodes (Swarm) as a result of the Smart Contract modules' node auction. Nodes can synchronize content with each other and persist specific content in storage.

• **PROOF OF STORAGE** is designed for data availability, and we call it the TEE–based Proof of Storage Challenge (PoSC). The challenger can generate a PoSC by randomly combining CIDs(Content IDentifiers), and the challenged node needs to respond within a specific time to prove that the content is indeed stored. The honest node will be rewarded while the dishonest node penalized.

• **4EVER–GATEWAY** helps users to access the nearest storage provider. The gateway provider contains the following modules:

  • DDNS, or Dynamic DNS, allows developers to simply configure DNS, or uses the free domain names provided by 4EVERLAND to accelerate global content access.

  • The Data/IO module provides content compression, data cache, transfer optimization, and other functions for each site.

  • Cert Manager module provides free SSL for each site, which saves a lot of time for developers.

• **SMART CONTRACT,**which is deployed on the Ethereum blockchain, includes 4 core modules: Auction, Payment, Lending and DAO.

• **4EVER-HOSTING** is an application for developers that can be accessed from the official website of 4EVERLAND. Developers can deploy DWeb to the 4EVERLAND gateway through the visual interface (Hosting App) or the command line (@4everland/cli). All nodes in 4EVERLAND can provide persistent storage for the DWeb, and all gateways can provide acceleration for the DWeb.

3.2 4EVER-STORAGE

3.2.1 CLUSTER SERVICE

Every node follows the same peer-to-peer network protocol. These nodes will automatically form a Swarm network which can be read in the node auction contract and then synchronize data with each other. Any node can synchronize data from the Swarm network.

The difference between 4EVERLAND nodes and IPFS nodes is that 4EVERLAND nodes are collaborative, which means that 4EVERLAND follows an additional protocol and stores specific data persistently. These stored data are challenged by Proof of Storage constantly: honest behaviors will be rewarded and malicious behaviors punished.

3.2.2 NODE DATA MANAGEMENT

In the 4EVERLAND storage network, since the developers should pay the cost through Ethereum smart contracts, the node needs to know which data CID needs to be stored locally by monitoring an on-chain payment event through the payment contract address and then store it according to the transaction field related to the CID. The PoSC data of nodes are then synchronized, the data being the key factor to judge whether the node should be punished.

3.2.3 STORAGE PROVIDER CAPACITY EXPANSION

4EVERLAND uses Swarm technology to form a large storage network. Therefore, it is very important to support the scalability of node storage capacity in the 4EVERLAND network. 4EVERLAND utilizes LVM (Logical Volume Manager) technology to achieve this goal. The capacity of the entire storage space can be dynamically adjusted through PV (physical volume), where PV addition and LV management are controlled by VG (volume group). Storage providers can easily add physical disks at the PV level to expand capacity. The provider's storage management logic is as follows:

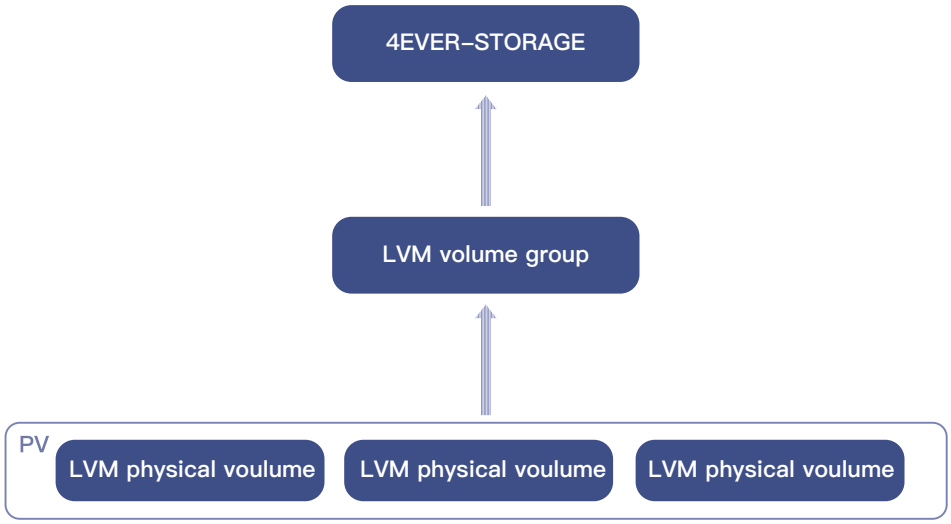

Figure 2: Logic of Storage Provider Scaling

3.3 PROOF OF STORAGE

4EVERLAND data layer is a global data synchronization network built on IPFS protocol with distributed nodes in which data is constructed and synchronized through the swarm network. Since data availability and integrity are the critical factors of the 4EVERLAND incentive component, the Proof of Storage mechanism needs to address the following issues:

a. The node synchronizes and stores data submitted by Swarm Network users.

b. The pinned data needed by users can be obtained from a 4EVERLAND node at any time.

Therefore, 4EVERLAND proposes the PoSC based on TEE technology to verify the reliability of nodes data.

The PoSC based on TEE

Data will be synchronized between nodes through the IPFS data synchronization mechanism. The purpose of the 4EVERLAND PoSC mechanism is to verify the data consistency of all nodes that are in the same swarm network. Usually, the PoSC mechanism will be used as below:

a. 4EVERLAND node's data availability should be verified before it participates in the node auction.

b. Nodes accept PoSC requests at a random moment during their running time.

To ensure that the nodes in the data network all run following the same challenge logic, the 4EVERLAND PoSC mechanism will utilize the Trusted Execution Environment technology(TEE), such as SGX of Intel. Each node stores the PoSC program in a TEE environment. In the TEE execution environment of each node, there are two main phases:

a. The TEE node running the PoSC program participates in the network.

b. The TEE network verifies and synchronizes the challenge data issued by other different nodes.

For problem a, the node's TEE program will have a mutual authentication process with the node which is already in the network by its hardware certificate information and measurable program information.

For problem b, the program running in the node's TEE environment will continuously follow the rules of PoSC. After the challenge request is finished locally, the node sends the challenge request data and result to the TEE executable network. The nodes in the other TEE trusted environments will verify the request signature and follow the same logic within the node TEE to compute and synchronize the related challenge request data locally. The detailed flow of challenge requests is described below.

Assume that there is node A in the network that will place the storage challenge StorageProof program in the TEE for execution. The storage data synchronization network is outside the trusted execution environment, and whenever the local 4EVERLAND data store synchronizes data locally by accepting pin requests, the StorageProof program of Node A will record the data related to the storage network (e.g. cid, merkle, etc.) through the OCALL interface of TEE Enclave. PoSC request will be sent to the node by a heap of random CIDs to confirm that the node has persistently stored the data. After the verification request is completed, the StorageProof program will send the data related to this challenge request to the other TEE networks of nodes B and node C. After receiving this request, the TEE Proof of Storage program of other nodes will verify the legitimacy of the challenge request using the node's public key. If the signature is legitimate, it will store the information about the challenge locally. As shown in the figure:

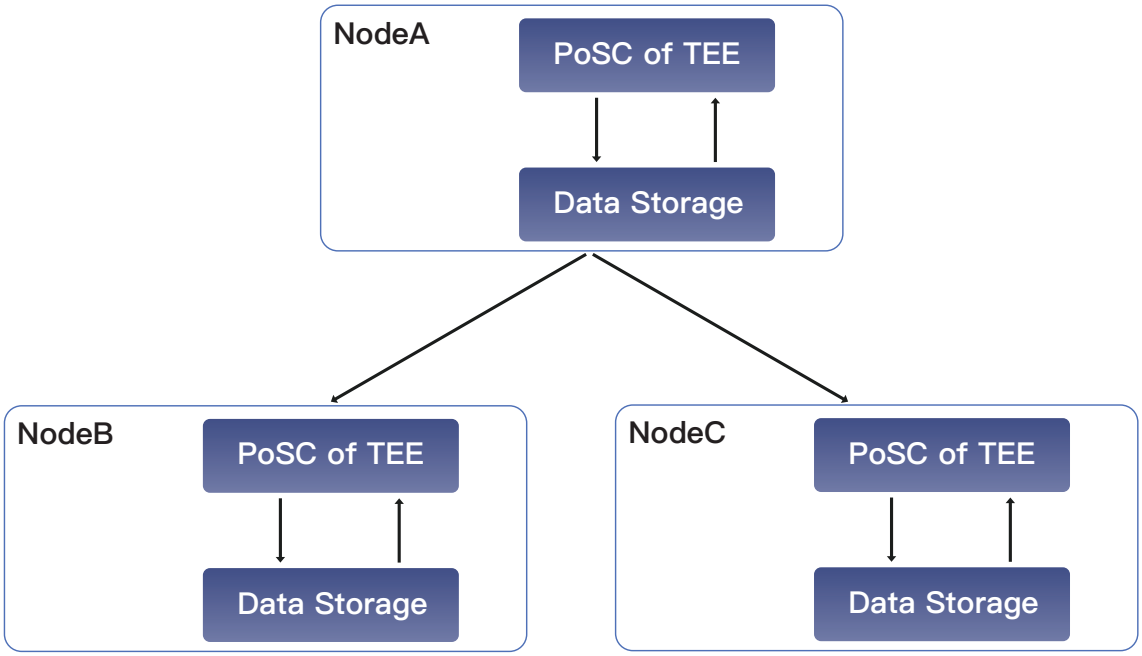

Figure 3: PoSC Architecture Based on TEE

To deal with challenge requests in the TEE network, nodes will synchronize the challenge request data with each other. 4EVERLAND will count the challenge behaviors of the node group, including the number of requests, request parameters, request results, and other information, and display these information to the community. These indicators are key factors to the ecosystem of node reward and punishment mechanism, which can be found in the white paper on node economic mechanism.

3.4 SMART CONTRACT

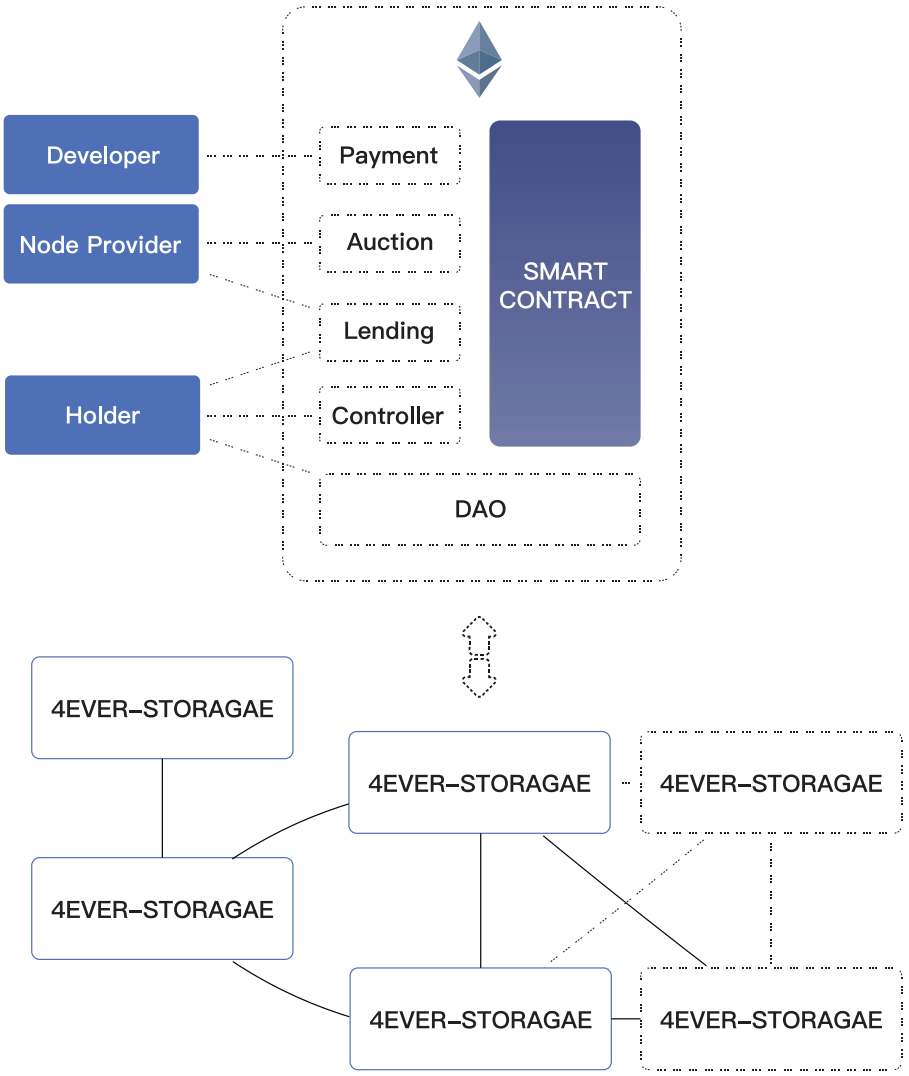

Figure 4: Smart Contract Design

The smart contract module is an important part for 4EVERLAND to realize decentralized governance. Through this module, nodes can join the 4EVERLAND network in an open, transparent, and free way, add nodes, modify node params, punish malicious nodes through a DAO, and finally, realize a safe, stable, efficient, cooperative, and autonomous node swarm.

The smart contract module of 4EVERLAND mainly consists of a payment contract, an auction contract, a lending contract, and a governance contract, which implement the 4 core functions of fee payment, node auction, lending, and community governance, respectively. The following section explains the core functional design of the smart contracts above.

3.4.1 AUCTION

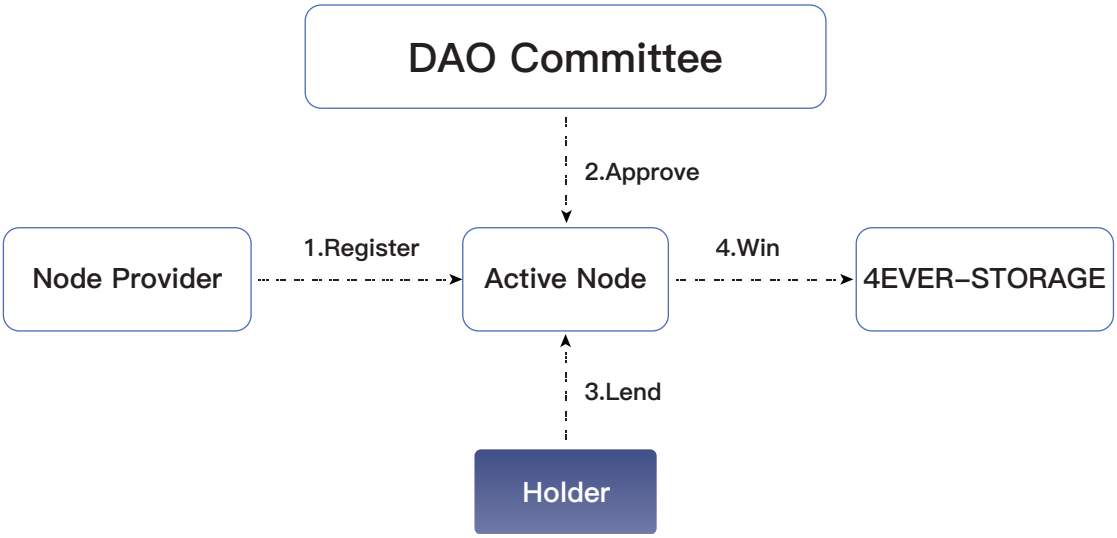

Figure 5: Auction Contract Design

Node participation mechanism:

• Any address can initiate a registration request by pledging a deposit in the Auction contract, and submitting the node name, node description, access information, and the number of coins to borrow.

• After a node is registered, it needs to complete a proof of storage to enter the active state.

• During the auction period, an available node slot can have multiple candidates. At the end of the auction period the candidate with the highest quota will become the node owner.

Reward distribution mechanism:

• The stake amount, stake period, withdrawal period, and block reward are set by the Controller contract.

• During the stake period nodes will receive block rewards, with the number of assets borrowed and interest rate being fixed. Nodes can unlock after expiration or exit of this period.

Node penalty mechanism:

• If a node misbehaves during the stake period (low Proof of Storage response rate, high response delay, etc.) the node will be punished by a vote of community users.

• The penalty for misbehavior is forfeiture of the node's staked assets and block rewards.

Node exit mechanism:

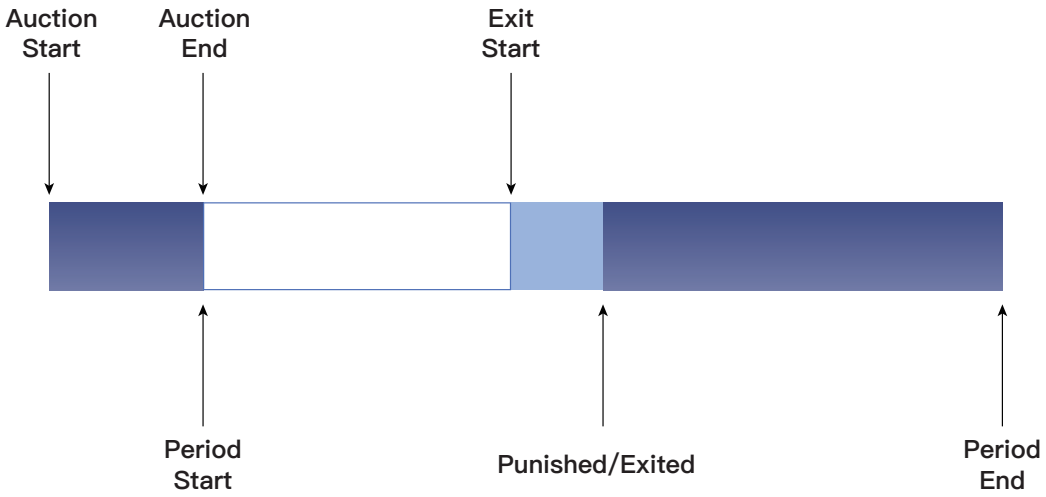

Figure 6: Node Participation and Exit

• Nodes that are not successful at the end of the auction period can exit the Auction contract at any time.

• During the stake period, nodes can initiate a withdrawal operation and will be withdrawn from the node status after the disclosure period, and the withdrawal cannot be revoked after initiation.

• During the stake period, nodes whose staked assets have been forfeited below a critical value will passively trigger an exit operation.

• After the node is exited, the remaining node stakes and borrowed assets can be unlocked and taken out. It should be noted that during the disclosure period nodes need to operate with good behavior, otherwise nodes will still be penalized.

• During the disclosure period, node slots will be reopened for auction, and after the disclosure period ends, the new node will automatically replace the old.

Node rollover:

• Active nodes can borrow assets at any time (usually when they are about to expire) to extend their term.

3.4.2 LENDING

• Users can deposit assets into a lending contract and receive different returns depending on the usage rate of the network, with all users enjoying the same rate of return.

• Nodes can borrow assets from the lending contract at the time of auction and pay different interest rates based on the usage rate.

• When the usage rate of the assets in the pool increases, the interest rate increases and vice versa.

3.4.3 CONTROLLER

The controller contract has the following functions:

- Add regional nodes
- Punish malicious nodes
- Set node block rewards
- Set the node stake period
- Set node exit period
- Set other global parameters

The above controls will eventually adopt the governance model of a DAO, which is jointly controlled by the community and the foundation.

3.4.4 Payment

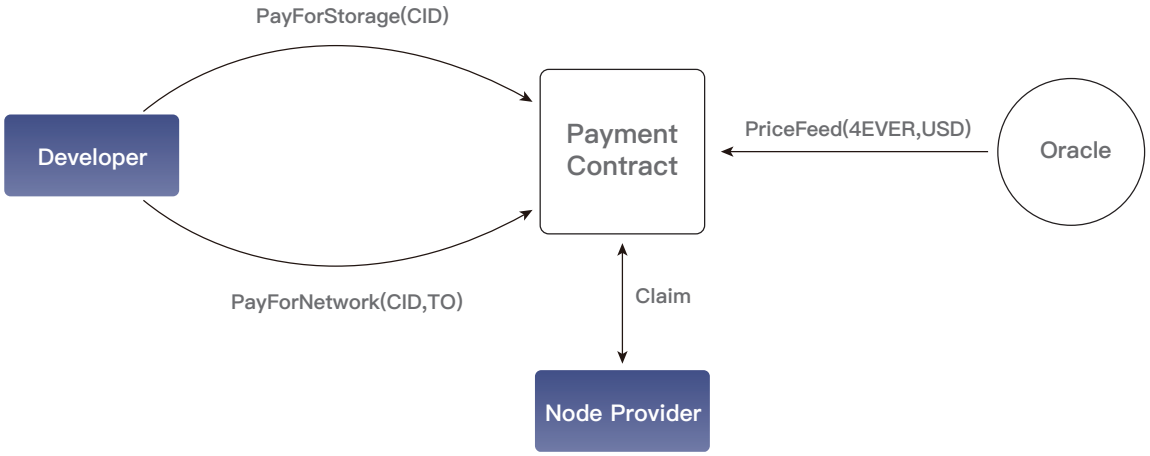

Figure 7: Payment Contract Design

During the publishing phase, developers use a payment contract to pay for content and select a specific gateway to prepay for storage and network resources. Tokens paid will be temporarily placed in the system revenue pool and wait for distribution.

After developers have completed the payment, the purchased storage and network resources will be assigned to the developer's address on Layer2. Developers can choose to approve these resources to 4EVER-HOSTING's (See 3.6) address, allowing 4EVER-HOSTING to act as a proxy for Layer2 resources.

The use of Layer2's solution can effectively reduce the fee consumption in the process of resource allocation and usage; A Layer2 solution will be chosen among rollup solutions as a trusted ledger for 4EVERLAND resources.

3.4.5 DAO

Community users can initiate governance proposals to improve the network based on 4EVERLAND public information.

Public information includes :

- Proof of Storage response rates of nodes
- Staking rates of nodes

Governance proposals include :

- Penalize malicious nodes.
- Add/reduce nodes in a region.
- Change dynamic global parameters.
- Modify block rewards.

3.5 4EVER-GATEWAY

3.5.1 DYNAMIC DOMAIN NAME SERVICE (DDNS)

DDNS is a dynamic DNS technology implemented by the 4EVERLAND team, which extends the record types and resolution methods based on the original DNS. DDNS can detect the status of the nodes in the 4EVER-Gateway, and end-user accesses are automatically assigned to the optimal 4EVER-Gateway through DDNS. Web 3.0 sites deployed in the 4EVERLAND network can be accessed in the same way as in Web 2.0.

Since the status of the swarm is public, DDNS is an open-source program that allows any person or organization to provide DDNS services to 4EVERLAND. After developers deploy their applications, they can select verifiable nodes in the swarm as DNS entry points.

DDNS is an optional service that allows users to access nodes directly, through a Web Plugin, which is a LOCAL DNS, a further advancement of DDNS, which puts DNS services directly on end devices, making the service more decentralized and giving the end device the ability to connect directly to the 4EVERLAND network and select the best gateway provider on its own.

Considering the diversity of end devices and the complex execution environment, the core functions of the Web Plugin are as simple as follows:

a. Query the nodes in the swarm and check the quality of service of the nodes.

b. Serve and forward requests from end-users.

c. Verify the integrity of the returned content.

3.5.2 DATA I/O OPTIMIZATION

The 4EVER-Gateway will optimize the DATA I/O performance based on the IPFS gateway which reads data and provides HTTP services for a specific CID, so that the efficiency of obtaining information through the 4EVER-Gateway is no less than that of a general HTTP Server. In a distributed system network like IPFS, the Resolve and Download processes are two important processes that affect the performance of data fetching. For the Resolve process, nodes need to search for data scattered in different remote nodes through the DHT table and then exchange data in the form of file blocks through the Bitswap system. Similarly, for the Download process, the file transmission rate is directly related to the node network bandwidth, disk, and other hardware configurations. Therefore, for such a data transmission mechanism, 4EVERLAND plans to improve the transmission performance and bandwidth utilization of the network data via caching and data compression technologies.

Caching Technology

For a specific CID in the IPFS network, the data can be accessed through the IPFS or IPNS gateway. This can be accomplished by using Publish and Resolve functionality of IPFS. For most users in the 4EVERLAND system, what they need to do is to use fixed-link addresses (i.e., IPNS) for access. Therefore, 4EVERLAND will build a caching layer on the IPNS side to eliminate the parsing process between IPNS addresses and IPFS addresses, allowing users to lock access to content CID faster.

Data Compression

Data compression, such as gzip compression, is a common technique used in Web 2.0/HTTP applications. Using gzip to compress a common file can save up to 60% of bandwidth. However, there is no such compression mechanism for the data in the current IPFS network, neither for data streams nor for file block transfers. Therefore, based on similar logic, 4EVERLAND will use two compression techniques to increase the efficiency of network transmission and improve the user experience:

a. Stream compression during transmission.

b. Default compression of the files uploaded by users.

3.5.3 CERT MANAGER

A certificate management module is integrated into 4EVER-Gateway by default, which can automatically create HTTPS for all sites deployed in the 4EVERLAND network.

3.6 4EVER-HOSTING

3.6.1 HOSTING APP

The Hosting App is a visual online DWeb deployment platform that helps developers quickly build, publish, and manage DWebs. The following describes a series of processes for publishing content on the Hosting App:

a. Authorize login through GitHub account.

b. Select a GitHub repository.

c. Automatically identify the framework or select the commonly used framework and customize the build command and output dir.

d. When deploying, hosting App provides a building service that automatically executes Clone, Install, Build and Upload.

e. Users are provided with a free domain name by 4EVERLAND, which is configured with DDNS by default and can access the newly released Web 3.0 through this domain name. One can access the newly released Web 3.0 site through this domain name or customize the DNS or ENS settings.

f. Connecting to an Ethereum wallet to pay bills using the 4EVER token.

3.6.2 @4everland/cli

This is a command tool that can be installed quickly via

`npm install @4everland/cli –g`

And does the following:

```
# Login,access_key can be created and managed in the backend after logging in through the hosting app
$ 4everland/cli login <access_key>

# View the current user
$ 4everland/cli whoami

# View the web3.0 sites owned by the current user
$ 4everland/cli hosts

# The command is executed in the directory where the content needs to be published, a .4everland directory will be created in the current directory and a config file will be added by default.
site_id

$ 4everland/cli upload. /dist <––site_id>
```

4. TOKEN ECONOMICS

In this section, we will focus on describing how the design of the economic model drives the effective operation of each participant. The 4EVERLAND economic model is designed to ensure the safe and stable operation of the overall network, and through a long–term dynamic incentive mechanism, to quickly develop a strong scale of the network, to reasonably distribute the benefits of each participant, and to drive each participant to work in the same direction of improving system value. The dynamic and efficient operation of the 4EVERLAND economic model enables the system to have a strong network, efficient node operation, competitive network usage fees, and globally distributed nodes that form quickly.

The governance token of 4EVERLAND is 4EVER, which is the core value carrier of the 4EVERLAND economic model, capturing the commercial value of the protocol and driving the long–term sustainability of the system.

4.1 THE PARTICIPANTS

Developer

Developers are the core users of 4EVERLAND who use the 4EVERLAND developer applications, such as publishing DWeb through Hosting App or publishing content through @4everland/cli, and pay storage and traffic fees to the network.

Storage Provider (SP)

Storage Providers participate in global Swarm construction by deploying Storage Provider programs; they participate in Storage Provider Auctions to obtain Storage Provider Slots. If they succeed in the auctions, they can obtain Storage Provider Slots and enjoy network rewards.

Gateway Provider (GP)

The gateway provider offers an access acceleration service for DWeb by deploying gateway provider, and the gateway verifies data integrity and maintains a secure connection between nodes and clients through an automatic certificate management mechanism. The gateway provider is acquired through the same auction process as the storage provider, and successful bidders are awarded a gateway provider seat and enjoy network rewards, as well as developer payments for traffic.

Challenger

Challenger challenges all Storage Providers on proofs of storage, wait for feedback from the challenged providers, announce the response of challenged providers to the community, and propose to punish inactive providers.

Token Holder

Holders can loan tokens to lending contracts for income
(the provider campaigners pay interest to the lending contract).
Holders also participate in DAO governance by holding tokens as proof.

End-User

Actual visitors for content in the 4EVERLAND can access the content published by developers using a domain name in their browser as they would in a regular Web 2.0 site, or install a Web Plugin to accelerate local network. It is also possible to use any IPFS gateway to access the content within the 4EVERLAND network.

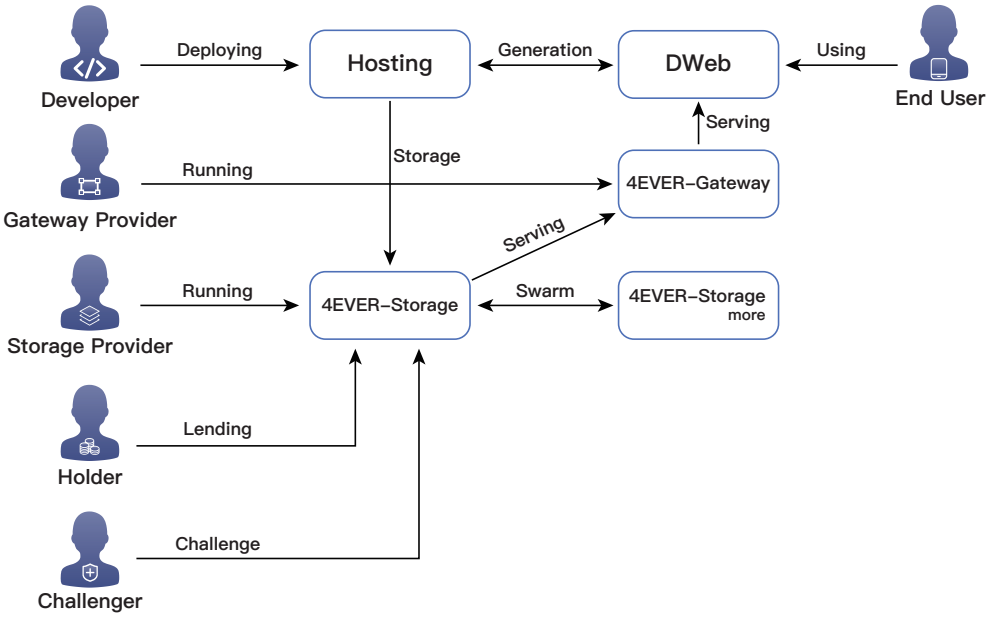

Figure 8: Interaction of Each Participant

4.2 ASSET FLOW

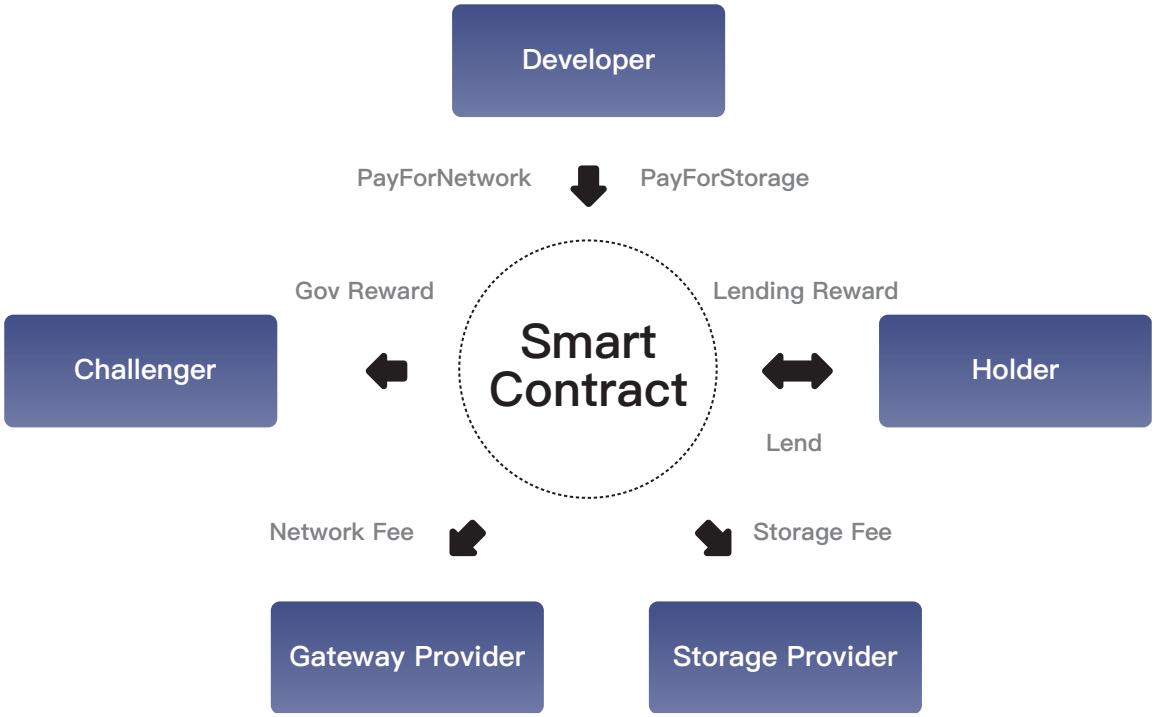

Figure 9: Flow of Asset in the 4EVERLAND Network

In the 4EVERLAND network, all payment processes are completed in the smart contract, which mainly involve:

Developers

Pay storage fees and network traffic fees to the contract;

Any token paid by developers will be swapped to 4EVER.

Token Holders

Receive income from lending 4EVER through the lending contract.

Storage Providers

Receive honest storage rewards from the contract;

Receive income from storage fees paid by developers.

Gateway Providers

Receive gateway construction rewards from the contract;

Receive income from the traffic fees paid by developers.

Challenger

Submit malicious provider information to DAO to receive governance rewards.

4.3 TOKEN DISTRIBUTION

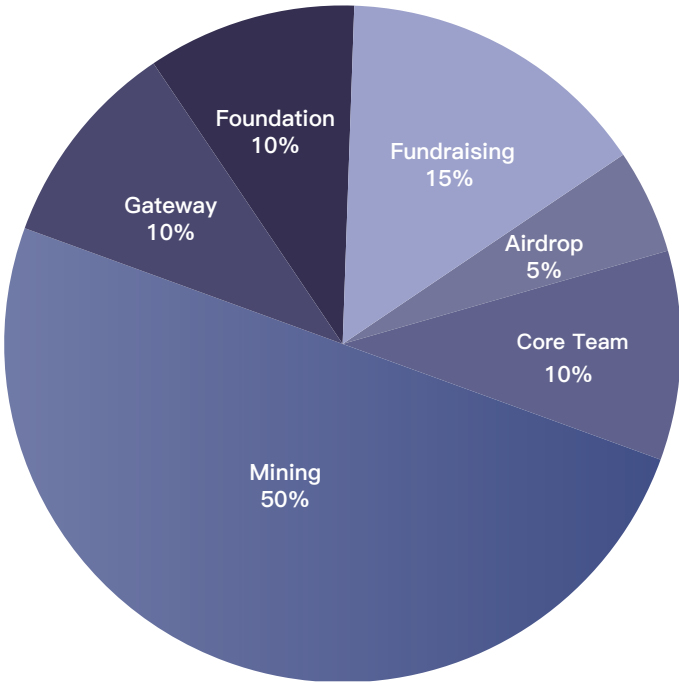

Figure 10: Token Distribution Chart

The total supply of 4EVER is 10 billion tokens whose distribution is as follows:

Storage Provider Mining:

50% (5 billion tokens), to be released linearly via storage rewards, halved in number every two years.

Gateway construction:

10% (1 billion tokens), which is used to stimulate the construction of global distributed gateway to ensure that 4EVERLAND can be built into a robust and fast gateway in the early stage.

Early Investors:

15% (1.5 billion tokens), which is allocated to early investors, and unlocked linearly over 6–24 months.

Core Team:

10% (1 billion tokens), to be used as an incentive for the core team, and unlocked linearly over 5 years.

4EVERLAND Foundation:

10% (1 billion tokens) for long-term development, operation, and brand building, unlocked linearly over 5 years.

Community Drop:

5% (500 million tokens), to be distributed to the 4EVERLAND community, to stimulate continuous participation and contribution.

4.4 TOKEN FUNCTIONS

4EVER is the core asset in 4EVERLAND, whose value is strongly positively correlated with the scale of the 4EVERLAND network.

4EVER token has the following functions:

a. Serves as the only lending and staking token for Providers during the auction and vesting period;

b. The revenue from developers will be used to buy back 4EVER tokens that are then distributed to nodes as commissions for resource services offered by Providers;

c. Serves as network transaction fee;

d. 4EVER token holders can propose and vote on network governance decisions through DAO.

4EVER captures the value of the whole network through the above functions: On one hand, the increase in storage and traffic business demand will lead to an uptick in the number of both Storage Providers and Gateway Providers, which will in turn stimulate greater Provider demand for staking and lending of 4EVER. On the other hand, developers' storage and traffic incomes will all be used to buy back 4EVER, enhancing the flow rate and liquidity of 4EVER.

4.5 ECONOMY OF NODE INCOME

The income of Storage Provider is comprised of two parts, including storage fee from developers and network storage mining reward (5 billion, which is the main source of inflation). The token economy of 4EVER consists of an increasing rate of usage and a decreasing rate of inflation. The yield from storage mining is halved every two years. Figure 10 shows the forecasted mining inflation rate of the Storage Providers in the system, which will decrease sharply from the second year, then to 3.65% in the fifth year and finally to less than 1% in the ninth year. As the product usage increases, such as for DWeb hosting, gateway acceleration, decentralized domain names, digital marketing systems, and data services, the demand for 4EVER increases, and so does the revenue of Storage Providers. In the early years of growth, the main income of Storage Providers comes from mining rewards. However, as the network grows and developers use more resources, storage fees will become the main source of income for Storage Providers (see Figure 11).

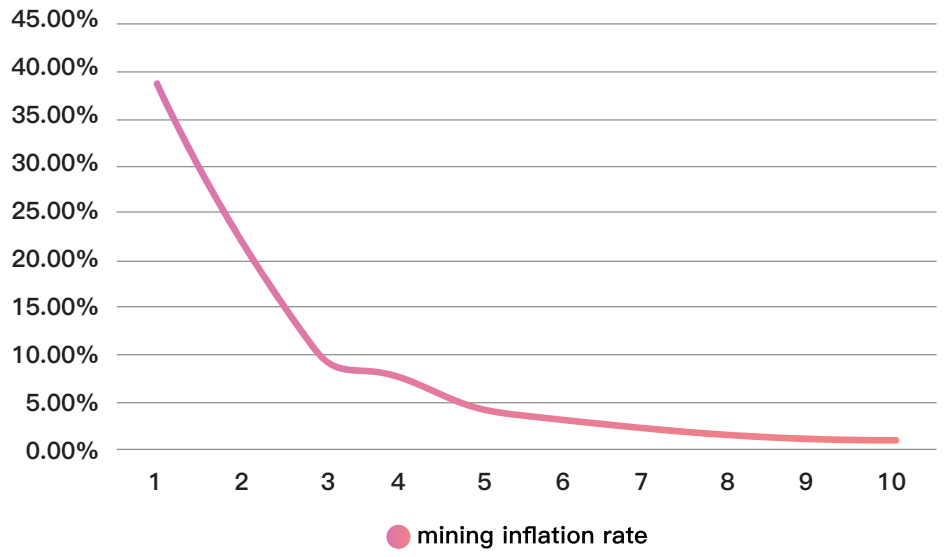

Figure 10. Storage Provider Mining Inflation Rate

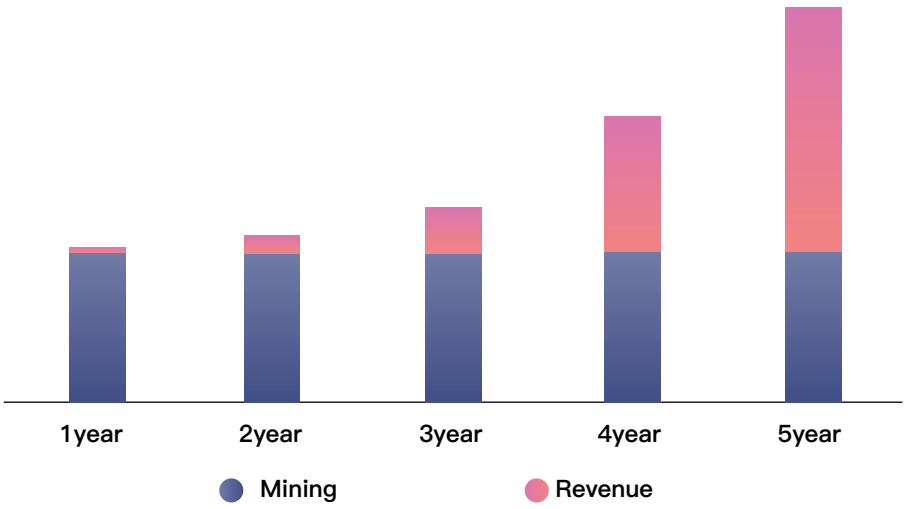

Figure 11. Forecast of Storage Provider Revenue Compositions

The income of the gateway provider is composed of two parts: The developer's share of the traffic fee and gateway construction subsidy (1 billion tokens). In the first three years, the 4EVERLAND gateway network will be rapidly built to support global high-speed access, which is funded bygateway construction subsidy. At the early phase of gateway construction , which is also the investment period, the main income of Gateway Provider comes from the gateway construction subsidy. However, as the network becomes more robust and Web3 traffic increases year by year, traffic fees will become the main source of income for Gateway Providers (Figure 12).

Figure 12. Projected Revenue Compositions of the Gateway Provider

4.6 LENDING MODEL

In order to maintain a reasonable state of asset liquidity in the network, we designed a borrowing and lending model, where:

Borrowing rate: BI

Utilization rate: U

$$BI = U * m + b$$

m and b are reconciliation parameters with different values for different borrowing usage rates, and the system's highest borrowing rate is RMAX

Range of Asset Usage Rate	Range of Interest Rate	m	b
[0, 0.5)	[0, 0.5)	0.4	0
[0.5, 0.9)	[0.2,0.2]	0	0.2
[0.9, 1]	[0.2, RMAX]	(RMAX–0.2)/0.1	m–RMAX

Ultimately, the relationship between the lending rate and the utilization rate of the pool is as follows.

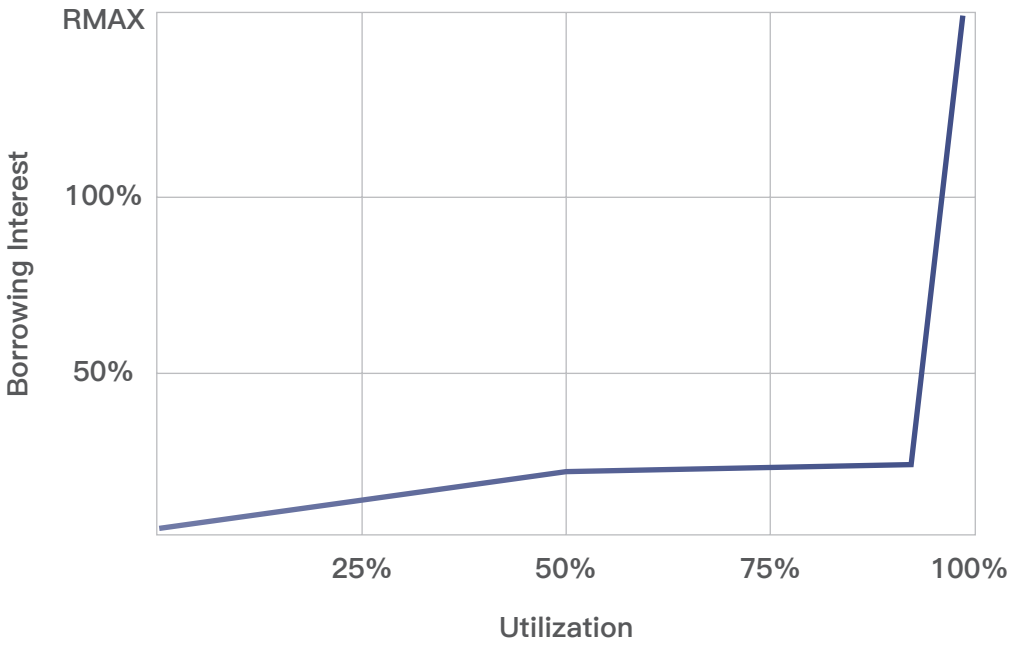

Figure 13. Relationship Between Lending Rate and Utilization Rate

The ranges of asset usage rate, ranges of interest rate, m, b, and RMAX may be adjusted in the control contract in the future. We assume:

Maximum number of nodes available for borrowing: ML

Node one-year fixed return: Y

Node stake: S

Then we get:

$$ML = (Y + S) / RMAX$$

5. ACKNOWLEDGMENTS

We would like to thank Juan Benet and Protocol Labs for creating IPFS, Vitalik Buterin, and Ethereum Foundation for launching the Ethereum project, and all those who have contributed to the 4EVERLAND open-source technology and reviewed this document.

4EVERLAND is a great place to pursue developers' ideal paradigm for Web 3.0, releasing them from the shackles of centeredness with more robust and secure services. We hope that more developers and researchers will join us along the way to build a globally accelerated, forever-linked network service for more users using 4EVERLAND, a distributed, highly efficient, self-motivated, and low-cost cloud computing network. Join us to witness a radical change in server architecture , a leapfrog upgrade based on Web3.0.

6. REFERENCES

[1] Juan Benet. IPFS–Content Addressed, Versioned,P2P File System.(DRAFT3), 2018.

[2] Protocol Labs.Filecoin:A Decentralized Strorage Network,2017.

[3] Protocol Labs. Technical Report: Proof–of–Replication. 2017.

[4] Protocol Labs. Technical Report: Power Fault Tolerance. 2017.

[5] Satoshi Nakamoto. Bitcoin: A peer–to–peer electronic cash system. 2008.

[6] Vitalik Buterin, A Next Generation Smart Contract & Decentralized Application Platform,2019.

[7] I. Baumgart and S. Mies. S/kademlia: A practicable approach towards secure key–based routing. In Parallel and Distributed Systems, 2007.

[8] B. Cohen. Incentives build robustness in bittorrent. In Workshop on Economics of Peer–to– Peer systems,2003.

[9] D. Mazieres and F. Kaashoek. Self–certifying file system. 2000.

[10] Intel Corporation. Intel® software guard extensions (Intel® SGX). Intel Labs., 2018.

[11] Intel Corporation.Secure Programming with Intel SGX and Novel Applications,2016.

[12] David Vorick and Luke Champine. Sia: simple decentralized storage, 2014.

[13] Albert–L´aszl´o Barab´asi and R´eka Albert. Emergence of scaling in random networks. science, 1999.

[14] Vitalik Buterin. What proof of stake is and why it matters. Bitcoin Magazine, August, 26, 2013.

[15] Snider, M. & K.Samani & T.Jain. Delegated Proof of Stake: Features & Tradeoffs [Z]. Multicoin Capital, 2018–03–02.