Sukhavati: A decentralized cloud service network focused on storage

Sukhavati Labs

Abstract

Sukhavati is a decentralized cloud service network focused on storage. At the consensus layer, it provides a secure and low-consumption consensus ledger based on the Proof-of-Capacity (PoC) mechanism. At the storage layer, it makes full use of the advantages of the hardware-based Trusted Execution Environment (TEE) technology and designs a new decentralized storage verification mechanism: Efficient Proof-of-Spacetime (EPoSt), which enables a significant number of small network storage devices to participate in the network and provide trusted computing resources. Based on these infrastructures, Sukhavati will build a decentralized data access gateway that covers both Web3.0 and Web2.0 storage services and that provides unified data storage, retrieval, and management services that can meet various local compliance requirements for Web3.0 applications. The mission of Sukhavati Network is to inspire and incentivize improvements to the distributed storage ecosystem, promote it as the true infrastructure of the next-generation Internet, and expand a wide range of application scenarios for the implementation of the Web3.0 vision.

1. Introduction

1.1. Background

Bitcoin brought us a global-scale consensus mechanism and ledger. On this basis, Ethereum has added a Turing-Complete smart contract layer, and proposed the goal of a world computer that will never stop. Thus, the vision of Web3.0 has once again germinated in people's minds.

Since the birth of the Internet and the Web, we have experienced the Web1.0 era where the web was primarily made of static read-only content, and the Web2.0 era where the content on the web became more and more interactive and users could read information or write and publish their own.

What will Web3.0 look like? The meaning of Web3.0 has been described as early as 2006 by Professor Tim Berners-Lee, the inventor of the World Wide Web, when he proposed the concept of the Semantic Web. Since then, Web3.0 has gradually emerged as a movement away from the centralization of services. Later, in 2014, Dr. Gavin James Wood, co-founder of Ethereum and creator of Polkadot, described his views on the components of Web3.0[1] in his blog as follows: 1) a decentralized, encrypted information publication system; 2) an identity-based pseudonymous low-level messaging system; 3) a consensus engine; and 4) an integrated user-interface. The future is yet to come. We believe the vision of Web3.0 is to make the Internet more decentralized, verifiable and secure.

The innovation and development of blockchain have made large-scale decentralized autonomous networks, trustless value transfer, and a certain scale of verifiable computing a reality. However, there is still a key component missing from the Web3.0 ecosystem—decentralized data storage infrastructure.

In order to solve the problem of decentralized storage verification, Filecoin proposes a solution based on zero-knowledge proofs (ZKPs) to continuously verify the promised storage. The feature of this solution is that it uses pure mathematical methods so as to achieve high security consensus, but the verification process consumes too much computing power, making storage and application costs prohibitively expensive. This cost inefficiency precludes a significant number of existing small storage devices (such as home NAS servers) from participating in the decentralized storage networks. As a result, its storage distribution today remains centralized, which is unfavorable for the realization of peer-to-peer (P2P) decentralized network. With such a network topology, the cost of data transmission in the future will be just as difficult to reduce as centralized services'.

Furthermore, due to varying local laws and regulations, it is necessary that decentralized storage network nodes be able to apply local compliance restrictions to the content it stores. Existing storage projects have not paid enough attention to this important and inescapable reality, thereby inadvertently exposing miners to violation of local laws. Miners can unintentionally encounter legal troubles and financial losses.

Web3.0 applications need an infrastructure that not only has the ability to recognize variations in law, but that also provides a secure, highly available, low-cost, and easy-to-use decentralized data access service.

1.2. Introducing Sukhavati

Sukhavati is a decentralized cloud service network focused on storage, designed and implemented based on hardware-based Trusted Execution Environment (TEE) and the Substrate framework.

At the consensus layer, Sukhavati uses Proof of Capacity (PoC) as the consensus mechanism. After the initialization operation, the PoC consensus mechanism requires only a small amount of computing and IO resource consumption to maintain high security consensus, allowing most resources to be used for other meaningful work. Also, to be able to quickly establish a strong consensus, Sukhavati can make use of the capacity power of some existing PoC consensus protocols and use them as a basis for launching.

At the storage layer, Sukhavati designs a new decentralized storage verification mechanism: Efficient Proof-of-Spacetime (EPoSt) based on the Trusted Execution Environment (TEE) technology. EPoSt consists of two challenges: LivingPoSt and WinningPoSt. A storage miner must complete the challenges and publish the result to the blockchain to obtain the incentives. EPoSt makes full use of the advantages of the hardware-based TEE technology. Compared to purely cryptographic ZKP-based methods, it is 100+ times faster in terms of verification speed. As a consequence, a significant number of small devices (such as NAS servers) at the network edge can participate in Sukhavati's storage mining and provide trusted computing resources as long as its CPU supports TEE, giving full play to their advantages of low cost, P2P network transmission and being close to the source of data.

At the application layer, Sukhavati will build a decentralized data access gateway covering both Web3.0 and Web2.0 storage services based on the local trusted computing capability of network nodes and using the storage layer as a medium. The gateway will ultimately provide unified data storage, retrieval, and management services that can meet various local compliance requirements for Web3.0 applications.

In addition, to serve Web3.0 applications more directly, Sukhavati will develop a storage parachain for the Polkadot ecosystem, providing data access gateway services for all the applications connected to the Polkadot Relay Chain.

The mission of the Sukhavati Network is to inspire and incentivize improvements to the distributed storage ecosystem, promote it as the true infrastructure of the next-generation Internet, and expand a wide range of application scenarios for the implementation of the Web3.0 vision.

2. Consensus Mechanism

2.1. Sukhavati's Consensus Design

In order to achieve the goal of a decentralized cloud service network, Sukhavati's consensus mechanism has chosen a design that is both useful and operational in engineering—the PoC+EPoSt hybrid consensus.

2.2. Capacity-based Consensus (PoC)

The PoC (Proof-of-Capacity) mechanism is a consensus-reaching mechanism that competes for the right to bookkeeping through the investment of storage capacity. In 2014, a post called "Efficient HDD Mining" was published on BitcoinTalk, introducing the concept of PoC. The publisher of this post is the founder of BurstCoin, and BurstCoin is the first successful PoC in the history of public chains. Six years have passed, and the concept of proof of capacity has taken root in the field of blockchain consensus algorithms. The Sukhavati project conducted extensive research on the existing PoC consensus algorithm, and ultimately chose to use BurstCoin[2] and MASS[3] consensus engine as the basis for further development. In order to achieve a useful consensus mechanism, Sukhavati’s PoC consensus process achieves the following benefits:

1. Security: able to resist malicious nodes with less than 50% of total capacity power.
2. Fairness: The capacity proof provided by a miner is linearly related to the income it earns. If a miner desires more revenue, the most rational action is to expand capacity efficiently. This mechanism allows individual miners and large miners to compete on an equal basis.
3. Scalability: After completing the initialization action, only a small amount of computing and IO resources are consumed to maintain the consensus security of the entire chain, thereby preserving more resources that can be used for other meaningful work.

The Proof-of-Capacity consensus process is as follows:

1. In the initialization phase, miners first create public and private keys, and then generate a series of HashMap data files according to the protocol and save them on the hard disk.
2. During each Slot, the miner obtains a Challenge from the previous block and looks for the answer to the Challenge in the local HashMap.
3. After finding the answer, the miner calculates the quality of the answer. If the quality is greater than or equal to the current difficulty, the proof is signed and the block is generated, and then the block is broadcasted to other nodes in the network. If the quality is less than the current difficulty, the slot ends.
4. When miners in the network receive a block, they will decide how to choose the current main chain based on the total difficulty, slot duration, proof quality, and block hash.

2.3. Storage-based Consensus (EPoSt)

Proof-of-Spacetime (PoSt)[4] is a consensus-reaching mechanism by which participants have to prove that they have spent a “spacetime” resource, meaning that they have allocated storage capacity to the network over a period of time, in order to obtain the opportunity of bookkeeping.

Currently, the most widely-used PoSt is in the Filecoin[5] project. Filecoin uses a algorithm based on a very heavy cryptographic machinery: Zero-Knowledge Succinct Non-Interactive Argument of Knowledge (zk-SNARK). zk-SNARKs provides the ability to verify the computing process, which enables nodes to prove that they have kept their promises and indeed have stored the data without revealing the details of the proof and the data itself. While the result of the proof is very small and the proof verification process is relatively fast, it still requires a lot of computing resources.

The security of consensus based on zk-SNARKs only relies on mathematical theory and software implementation. Its advantage is that it has high security and has nothing to do with the execution environment. Its disadvantage is that it cannot give play to the advantages of the hardware system.

Regarding the trade-off between consensus security and efficiency, Sukhavati has made a different choice compared to Filecoin: We designed Efficient Proof-of-Spacetime (EPoSt) on the basis of hardware-based Trusted Execution Environment (TEE) technology. It consumes only a few computing resources to verify the storage and takes full advantage of the chip’s hard-to-crack feature to provide security for the consensus process.

Storage deal flow:

1. The client posts a storage order in the storage market, which includes: the IPFS URL of the data to be stored, data hash, storage duration and storage fee.
2. The miner and client come to an agreement about the terms of the deal and publish it on chain.

3. TEE remote attestation on the miner side.
4. The miner fetches the corresponding data from the client.
5. The miner calculates the data checksum, seals the data, and submits the result to the chain together with the pledge collateral. Once the order is validated, the miner will immediately gain storage power and will continuously receive storage fees during the life of the deal.

The Efficient Proof-of-Spacetime mechanism consists of two parts: LivingPoSt and WinningPoSt.

LivingPoSt is a mechanism used to audit the storage miner's commitment to maintain the data copy for a promised period of time. The Sukhavati network expects continuous availability of stored files. Therefore, storage miners must verify the stored files every 24 hours and submit the proofs to the blockchain. If the verification is passed, the miner can maintain the corresponding storage power. If not, the pledge collateral will be slashed and the corresponding storage power will be lost.

Storage miners can recover from faults within a limited period of time before being considered to have abandoned their storage commitments. If necessary, storage miners can also preemptively issue a "declared faults", which will reduce the penalty, but they still must resolve the faults within a reasonable timeframe.

WinningPoSt is a mechanism used to reward storage miners for their contributions. At the beginning of each Epoch, the chain will select a certain number of miners (less than or equal to the number of slots) to give them a proof challenge. If a storage miner successfully completes the challenge before the end of the Epoch, the miner will be given a chance to mine a block and get rewarded in the next Epoch. A storage miner’s probability of being selected is proportional to its storage power.

The process of LivingPoSt is as follows:

1. TEE remote attestation.
2. On each proof cycle, the storage miner performs integrity verification and challenge verification on all stored documents and submits the results to the blockchain.
3. If the verification result is consistent with the data hash in the order, it means that the storage commitment of the miner is valid and the corresponding storage power remains unchanged.
4. If the verification result is inconsistent with the data hash in the order, the corresponding storage power of the miner will be removed and the pledge collateral will be slashed.
5. Miners who fail to pass the verification will be slashed every day and cannot accept any new storage orders until they recover the faults or until a certain period of time has passed.

The process of WinningPoSt is as follows:

1. TEE remote attestation.
2. At the beginning of each Epoch, a certain number of miners are selected and given proof challenges. A miner’s probability of being selected is proportional to the share of the network’s total storage power it contributes.
3. The miners who successfully complete the challenge before the end of Epoch will be included in the block producer candidates list of the next Epoch. Those who fail to complete the challenge will forfeit their opportunity but will not be punished.
4. In the next Epoch, the priority consensus method is randomly selected at the beginning of every Nth slot: PoC or EPoSt. If EPoSt has priority, M candidates of the above list will be randomly selected, and they must complete the block production within the specific slot.
5. In a certain slot, among all blocks produced by selected candidates, only those with the same parent block set can form a new block set.
6. Storage miners who successfully mined the block can each receive a block reward. The amount of the reward is (PoC block reward + EPoSt block reward / M).
7. The block set produced by EPoSt has a weight value, which can be converted to the difficulty value of the PoC block. When the miners in the network (including PoC and EPoSt miners) receive a block, they convert the weight value of the EPoSt block set into a difficulty value, and then select the current main chain.

The EPoSt consensus mechanism requires the miner to be equipped with a CPU that supports TEE technology.

However, TEE is not a panacea. Research in recent years has shown that even if it is a hardware-based technology, it may still have vulnerabilities that could be exploited by attackers. Therefore, in addition to the LivingPoSt mechanism, EPoSt also includes a spot check challenge mechanism. A challenger can pledge a certain amount of collateral and conduct a spot check on a specific miner by sending some pieces of challenge data with insertion positions and asking for the checksum hash of combined content. If the miner passes the challenge, the challenger’s collateral will be burned. If it fails, the storage miner will be punished and the challenger will be rewarded.

2.4. Randomness

In the EPoSt consensus process, randomness is crucial to the fair and unpredictable selection of block producer candidates. Since a computer is a deterministic device, it will always have the same output for the same input. A computer's behavior is entirely predictable, by design. In general, the randomness on a computer is actually a kind of pseudo-randomness, whose resulting randomness depends on the seed input from outside. If the same seed is given, the result will always be the same. In order to prevent nodes from doing evil, randomness must be reliable and verifiable in the consensus process. The Sukhavati network uses drand network as the source of randomness.

Drand[6] is a distributed randomness beacon daemon. It was originally developed by the EPFL DEDIS team and is now under the drand organization. The drand network that Sukhavati uses is operated by a group of organizations around the world that includes: Cloudflare, EPFL, UCL, UIUC, Ethereum Foundation, Protocol Labs etc.

The drand network can provide a continuous source of randomness that is:

1. Decentralized: drand is a software run by a diverse set of reputable entities on the Internet and because a threshold of them is needed to generate randomness, there is no central point of failure.
2. Publicly verifiable and unbiased: drand periodically delivers publicly verifiable and unbiased randomness. Any third party can fetch and verify the authenticity of the randomness, ensuring that it hasn't been tampered with.
3. Private: drand nodes can also deliver encrypted randomness to be used in local applications.

2.5. Summary

The consensus design of the Sukhavati network is meant to be useful and operational. For this purpose, we designed a hybrid consensus mechanism of PoC + EPoSt based on the hardware-based TEE technology. The network in effect achieves a high hardware utilization efficiency and provides a secure consensus layer and an efficient storage layer for the whole decentralized cloud service network. Lastly, it also sets the stage for the scalability of the application layer.

3. The Sukhavati Network

3.1. Network Roles

The Sukhavati network is mainly composed of the following roles: consensus node, storage node, application node, senate node, Sukhavati DAO and application users.

3.2. Consensus Node

In the first stage of Sukhavati, there is only one role in the network: the consensus node. The first step of any new blockchain network is to establish a secure consensus. The role of the consensus node is to accomplish this mission by establishing and maintaining the foundation of the network: the consensus layer.

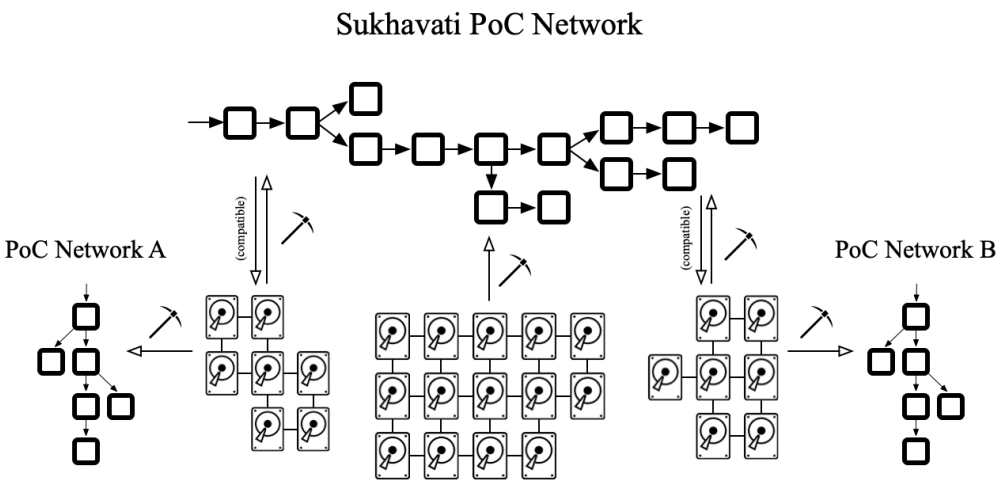

Figure 1. consensus nodes merged mining

The consensus node uses Proof of Capacity (PoC) as the consensus mechanism which consumes only a few computing resources. Also, in order to quickly establish a sufficiently strong consensus, Sukhavati's consensus nodes can make use of the capacity power of some existing PoC consensus protocols and use them as a basis for launching.

In addition to consensus reaching, the consensus node also provides functions such as value transfers and economic incentive models, as well as support for the Decentralized Autonomous Organization (DAO), which enables the decentralized governance of the network.

3.3. Storage Node

The goal of the storage node is to build the storage layer of Sukhavati decentralized cloud service network.

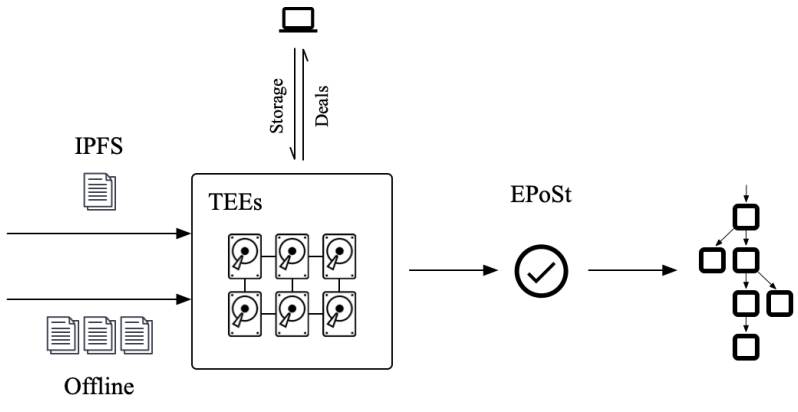

Figure 2. storage process

The storage node uses Efficient Proof-of-Spacetime (EPoSt) as the consensus mechanism based on Trusted Execution Environment (TEE) technology. When

a storage node accepts a storage deal, it will replicate the data through the IPFS network, calculate the data checksum, seal the data and submit the result to the blockchain with pledged collateral. Later during the deal life cycle, the storage node will continuously prove that it is still storing the client's data it has pledged to store.

Clients can customize the required redundancy when submitting storage orders in the storage market. If needed, clients can also send storage orders to specific storage nodes. Therefore, offline large-scale data storage can also be well supported in the Sukhavati network. For example, data sets commonly used by research institutions such as genetic sequence data, climate data, artificial intelligence and machine learning data, or enterprise data archives, large multimedia files, etc.—all such PB-scale data can be stored on the network through online transaction and offline transmission.

The actual storage of a storage node is converted into its storage power. Initially, when the storage power of the entire Sukhavati network has not yet reached the baseline, storage nodes can receive certain rewards according to the storage power they contribute. Later, when the storage power of the entire network reaches the baseline, the network will activate block production based on storage-based consensus. At this time, the storage nodes will begin to participate in the block production competition with their storage power.

3.4. Application Node

The application node provides local trusted computing and network transmission capabilities. Local trusted computing is a unique computing service provided by the Sukhavati network. It can complement smart contracts in actual application scenarios.

The current blockchain provides trustless computing services in the form of smart contracts. The code of the smart contract is deployed on a ledger with consensus guarantee. When a transaction invoking a smart contract is included in a block, all the nodes receiving the block will execute the same process of the smart contract with the same input and check the result of new state tree to confirm that the block producer has indeed executed the right logic. Smart contracts are very suitable in scenarios where the state of the global ledger needs to be "updated", because the execution process is included in the consensus process, ensuring the reliability of the computing.

However, in many application scenarios, there are actually a large number of processes that only involve "reading" but no "updating" to the state of the global consensus ledger[8], such as database query and analysis, content searching services, AI model training, etc. In these scenarios, clients also need to ensure that the computing process is reliable, but the process will not make any change to the global state tree. Smart contracts are not quite applicable in such scenarios.

Sukhavati has taken such demands into consideration early in the development of its network and designed the role of application node with local trusted computing capabilities[8]. The application nodes are upgraded from the storage nodes. Since the EPoSt consensus mechanism is based on TEEs, storage nodes only need to spend a small amount of computing and IO resources to maintain the storage power. With this advantage and the readily available TEEs, storage nodes can use their computing and bandwidth resources to provide trusted computing and network transmission services.

Application nodes can flexibly price their own computing and network transmission services without being affected by other nodes.

3.5. Sukhavati DAO

There are many model parameters in the Sukhavati network. Incentivizing different roles by adjusting model parameters at different stages is crucial to the development of the network. In order to realize the decentralized governance of the network and the entire project, Sukhavati should be governed by its DAO, and all stakeholders can directly participate in discussion and governance.

Sukhavati DAO will be launched after the mainnet is launched. On this decentralized governance platform, any community member can propose a network development path or modification to model parameters. Every proposal includes a detailed description and a voting period. The proposers then need to push community members to vote on their proposals. Before the voting ends, if a certain voting rate and approval ratio are reached, the proposal is passed. In order to prevent useless spam on the DAO platform, a certain amount of tokens must be staked for each proposal. Only if a certain voting rate is reached will the staked tokens be returned. Otherwise, the locked tokens will be burned after the voting ends.

We hope that the Sukhavati DAO can make the community more united and make the evolution of the network smoother.

3.6. Senate Node

Senate nodes are elected by the community through the Sukhavati DAO and are awarded for having an important impact on the development of the network at a critical moment or for contributing significant resources to the entire network. They bear the special responsibility for making proposals and supervising the healthy development of the network.

The total number of Senate nodes is limited. The election conditions of Senate nodes require that they must have a considerable amount of investment in the Sukhavati network. In return, the Senate nodes will be able to obtain a certain amount of income as the network grows. Their interests are tied to the long-term healthy development of the Sukhavati network.

The intent of this role is to elevate groups or individuals with special abilities to help the Sukhavati network connect with key resources in different fields. It also assumes the responsibility of maintaining network security. When malicious behaviors are found in the network, the Senate nodes are obliged to investigate the attacker and initiate punishment proposals.

3.7. Network diagram

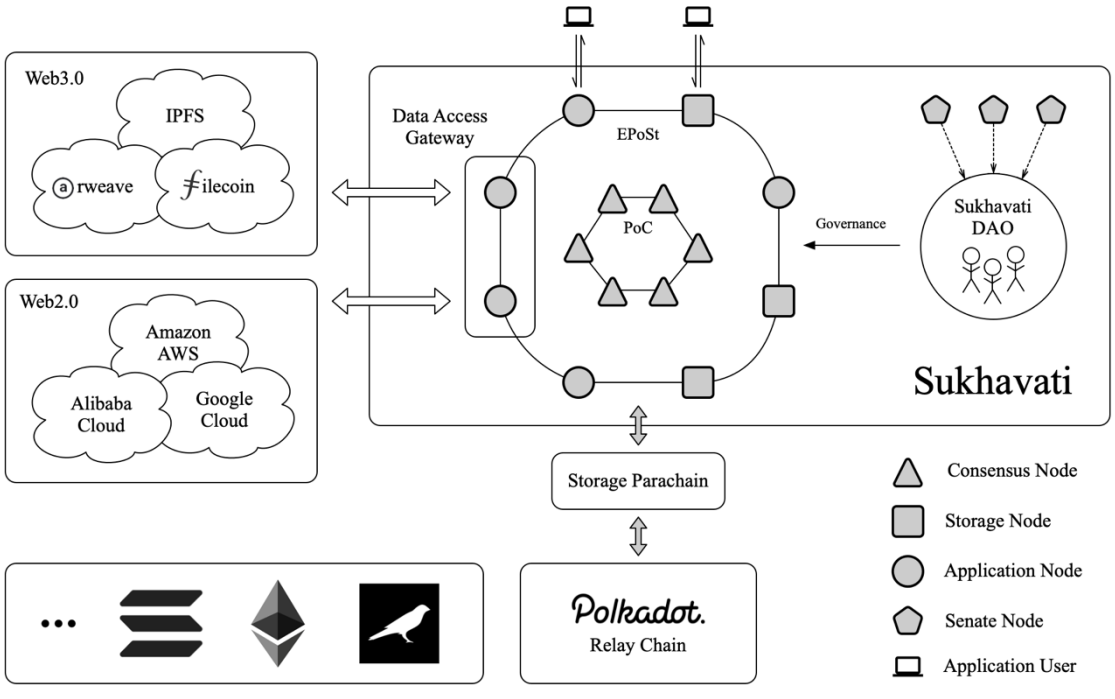

Figure 3. Sukhavati network diagram

At the application layer of Sukhavati, infrastructures such as data access gateway can be built for Web3.0 applications.

3.8. Stages of the Sukhavati Network

In order to achieve the ultimate goal of a decentralized cloud service network, Sukhavati proposes three evolutionary stages to implement the key parts of the network step by step.

In the first stage, when the mainnet is initially launched, Sukhavati will be a consensus network that uses the PoC consensus mechanism and is composed of consensus nodes. The goal of this stage is to establish a sufficiently strong and stable consensus and enable the network to have a secure and reliable ledger, initially form the governance by Sukhavati DAO and ensure that the economic model can begin to effectively incentivize participants.

The second stage of Sukhavati is a consensus and storage network. On the basis of PoC consensus, EPoSt consensus mechanism is added. The second stage of the network is composed of consensus nodes and storage nodes. The goal of this stage is to establish a usable and efficient storage layer, incentivize the storage nodes to provide storage services, realize the application of trusted computing in the Sukhavati network, and prepare for application services in the next stage.

The second stage of evolution is challenging, and it is also a key stage for the entire network. The implementation of the storage layer is further divided into two sub-stages: First, the incentives for the storage layer will be implemented. Storage nodes that contribute storage power will receive storage rewards, but they will not participate in the block production. Second, after the storage power of the entire network reaches the baseline, storage nodes will have chances to produce blocks. The purpose of baseline is to prevent the network from being attacked when the total storage power is small, and to make the network upgrade more smoothly. Before the mainnet storage nodes start to participate in the block production, a specific testnet will be launched for the exercise of the upgrade.

The third and final stage is the ultimate form of the Sukhavati network. On the basis of the consensus and storage layers, the storage nodes can be upgraded to application nodes. The goal of this stage is to provide local trusted computing and network transmission capabilities, which makes Sukhavati a true decentralized cloud service network.

4.TEE - Trusted Execution Environment

A trusted execution environment (TEE) is a secure area of the main processor. It guarantees that code and data loaded inside are protected with respect to confidentiality and integrity. The three major CPU platform vendors have different implementations: Software Guard Extensions (SGX) on the Intel platform, Secure Encrypted Virtualization (SEV) on the AMD platform, and TrustZone on the ARM platform. Among them, the SGX of the Intel platform is the most widely used TEE platform.

The following are the core building blocks of TEE[9]:

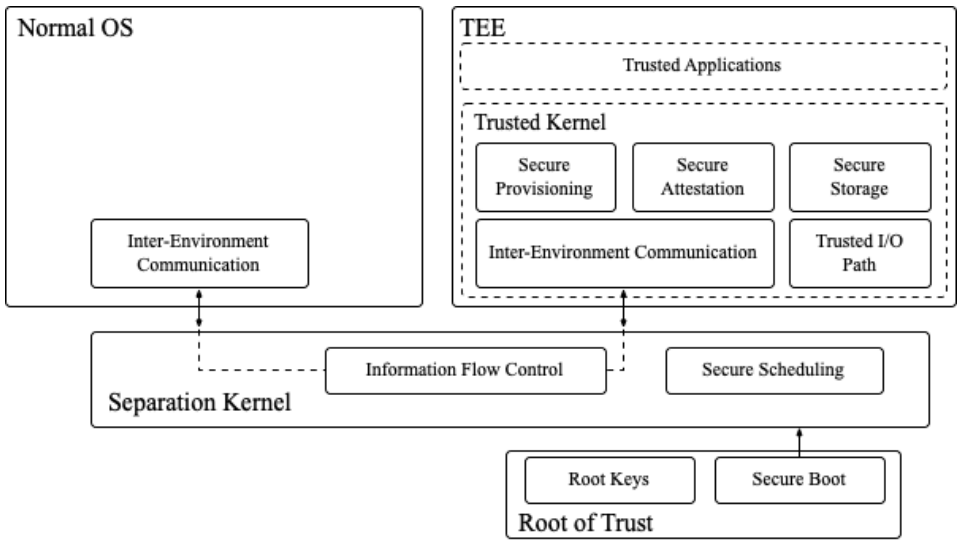

Figure 4. an overview of TEE building blocks

- Secure Boot assures that only code of a certain property can be loaded. If a modification is detected, the bootstrap process is interrupted.
- Secure Scheduling assures a “balanced” and “efficient” coordination between the TEE and the rest of the system. Indeed, it should assure that the tasks running in the TEE do not affect the responsiveness of the main OS.
- Inter-Environment Communication defines an interface allowing TEE to communicate with the rest of the system.
- Secure Storage is storage where confidentiality, integrity and freshness of stored data are guaranteed, and where only authorized entities can access the data.
- Trusted I/O Path protects authenticity, and optionally confidentiality, of communication between TEE and peripherals (e.g., keyboard or sensors). Thus, input and output data are protected from being sniffed or tampered with by malicious applications.

The remote attestation, by which a host authenticates its hardware and software configuration to a remote host, is an important mechanism of TEE to resist malicious behavior. Taking Intel SGX as an example, the basic process of remote attestation is:

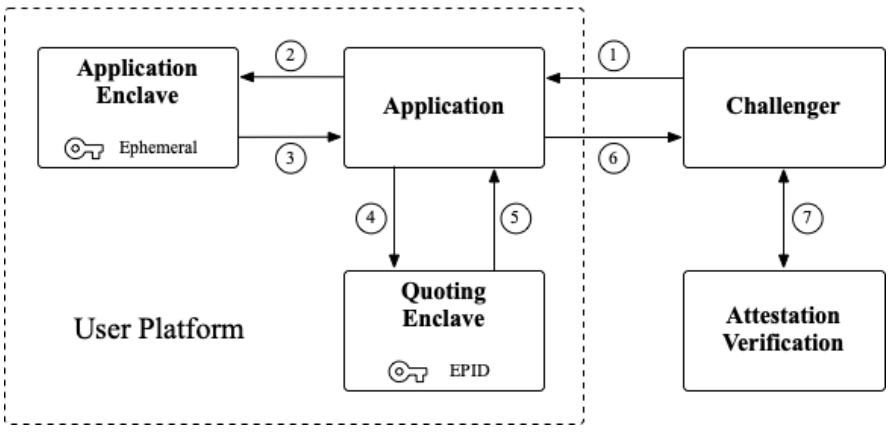

Figure 5. Intel SGX remote attestation

1. Challenger initiates a challenge to the Application of the remote computer, along with a random number nonce, which is used for the activity detection of the remote computer.
2. The Application sends the Quoting Enclave's ID, challenge and random number nonce to the Application Enclave.
3. After the Application Enclave receives the message, it initializes the log and other data according to its own program (indicated by manifest), generates a summary, puts it in the User Data, and then calls the EREPORT command to generate a report, and binds the manifest to this Application Enclave. Finally, the report is sent to Application.
4. Application forwards it to the Quoting Enclave and lets the Quoting Enclave sign it.
5. The Quoting Enclave first verifies the report, then signs it with the private key in the EPID, generates a QUOTE, and then returns the QUOTE to the Application.
6. Application sends the QUOTE and other related data to Challenger.
7. Challenger combines EPID public key and Attestation Verification Service to authenticate the QUOTE.

5. Token Economics

5.1. Token Allocation

The total supply of Sukhavati token (SKT) is 618,033,989 (approximately 618 million). The Token allocation details are as follows:

- 61.8% (381,945,005 SKT): Mining rewards. It will be released through block rewards and used to maintain the operation of the Sukhavati network. The mining rewards of the testnet will account for 1% of the total assigned amount.

- 15% (92,705,098 SKT): Strategic investors. It is allocated to investors participating in private and public offerings and will be released linearly every day for 12-24 months.

- 6.2% (38,318,107 SKT): Ecosystem builders. It is used to support the miner ecosystem and external ecosystem cooperation, and provide funds for long-term network management, partner support, academic funding and community services.

- 10% (61,803,399 SKT): Founding team. As the R&D and operating expenses of the project, it will be used for technical research, project development, and daily operations of the team, which will be linearly unlocked within 3 years.

- 7% (43,262,379 SKT): Sukhavati Foundation. Used for Sukhavati's ecosystem development, commercial cooperation, marketing, exchange cooperation, etc.

5.2. Incentive Model

Sukhavati's block rewards are divided into four parts: consensus rewards, collateral rewards, internal ecosystem incentives and Web3.0 cooperation incentives.

The consensus rewards are obtained by miners through PoC mining and EPoSt mining. The collateral rewards will be allocated to eligible token stakers according to the rules. The internal ecosystem incentives will be rewarded to different contributors at different stages of the network according to the needs of ecosystem development. The Web3.0 cooperation incentives are reserved for cooperative contributors who help promote the development of the entire Web3.0.

6. Use Cases

6.1. Data access gateway as Web3.0 storage infrastructure

At present, there are many storage projects in the Web3.0 ecosystem. Most of them are incompatible with each other and suitable for different demand scenarios. Based on Sukhavati's storage layer and application layer, a universal data access gateway can be built to connect various protocol islands in the ecosystem and provide unified storage services for Web3.0 applications.

For the data storage of DApps, the contents can be distributed to different storage networks according to specific demands through the gateway, while helping them to manage storage status and resource scheduling. For example, for NFT crypto artworks, they can be permanently stored on the Arweave network, and cached in the storage layer of the Sukhavati network; for cloud storage service data, it can be stored on the Filecoin network; and if the file to be stored on Filecoin is very large, and the seal operation takes a long time, one can first save the file to the Sukhavati data access gateway, and then the gateway is responsible for saving it on the Filecoin network. After seal is completed, the relevant application will be notified.

Similarly, the Sukhavati data access gateway can also provide content delivery services for data stored in decentralized storage networks. Utilizing the cost efficiency of the Sukhavati storage layer and the advantages of P2P network transmission, it can provide the same experience as reputable centralized services for applications such as video-on-demand services, live streaming, and large file downloads.

6.2. A bridge to Web3.0 for Web2.0 applications

Thanks to Sukhavati's local trusted computing capability, traditional Internet Web 2.0 applications can connect to the Web 3.0 ecosystem by deploying themselves on the application nodes and leverage the Sukhavati network to transfer value.

6.3. AWS Lambda-like serverless cloud computing service

Any application node in the Sukhavati network can provide Function as a Service (FaaS) computing services similar to AWS Lambda to the public. Clients can use the service by storing data and deploying the code snippet to the same application nodes. With the support of TEEs and the Sukhavati network, clients can enjoy secure computing services at a much cheaper price.

6.4. Support for 5G network and edge computing

In the era of 5G, due to its high bandwidth, low latency, and massive connections, edge computing will be a crucial part of the 5G platform. The Sukhavati network's consensus ledger, low-cost and highly reliable storage, local trusted computing capability and peer-to-peer network topology can provide strong support for the applications of the 5G network and edge computing.

References

[1] Gavin Wood, “ĐApps: What Web 3.0 Looks Like”, URL: https://gavwood.com/dappsweb3.html, 17 April 2014.

[2] S. Gauld, F. Von Ancoina, and R. Stadler, “The burst dymaxion an arbitrary scalable, energy efficient and anonymous transaction network based on colored tangles, ” in Proc. CryptoGuru PoC SIG, 2017.

[3] MASS, URL: http://www.massnet.org/.

[4] Tal Moran, Ilan Orlov, “Simple Proofs of Space-Time and Rational Proofs of Storage”, in: Advances in Cryptology – CRYPTO, 2019.

[5] Protocol Labs, “Filecoin: A Decentralized Storage Network”, URL: https://filecoin.io/filecoin.pdf, July 19, 2017.

[6] Drand, URL: https://github.com/drand/drand.

[7] Raymond Cheng, Fan Zhang, Jernej Kos, Warren He, Nicholas Hynes, Noah M. Johnson, Ari Juels, Andrew Miller, and Dawn Song. 2018. Ekiden: A Platform for Confidentiality-Preserving, Trustworthy, and Performant Smart Contract Execution. ArXiv e-prints (2018). http://arxiv.org/abs/1804.05141.

[8] Marcus Brandenburger, Christian Cachin, “Challenges for Combining Smart Contracts with Trusted Computing”, SysTEX '18: Proceedings of the 3rd Workshop on System Software for Trusted ExecutionJanuary 2018 Pages 20–21, https://doi.org/10.1145/3268935.3268944.

[9] Mohamed Sabt, Mohammed Achemlal, Abdelmadjid Bouabdallah. Trusted Execution Environment: What It is, and What It is Not. 14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, Aug 2015, Helsinki, Finland. ff10.1109/Trustcom.2015.357.hal-01246364

[10] Markus Jakobsson, “Secure Remote Attestation”, in: Computer Science, 2018.