SINSO Technology

SINSOV2(filX)

Whitepaper

Dec.2023

ABSTRACT

The main body of Sinso V1 is Sinso Gateway,a distributed content delivery network (DCDN)based on the libp2p protocol that provides L2 cache acceleration for the Filecoin ecosystem.

Sinso V1 has successfully built the main body of the DCDN secondary network,gathering over 6000 nodes.Miners from more than 10 countries and regions have participated in network construction,running a complete economic model.Sinso V2 is expected to serve as Layer2 for Filecoin,enabling computation and bandwidth expansion.

Sinso V2 will leverage the established DCDN network to encompass a comprehensive Layer2 design.This will entail integrating the previously discrete DAPP Cube and Donor Network components,while concurrently constructing the computing resource market and data trading market.As a result,a distributed edge computing network will be established to provide vital infrastructure services including computing power,bandwidth, computation verification,and incentives.

This strategic endeavor aims to facilltate the impending era of fog computing and effectively enable the scaling of computations within the Filecoin ecosystem.

Architecture

Summari ze

Sinso V2 is a decentralized edge computing system built upon IPC(InterPlanetary Consensus).The existing Sinso Gateway will undergo a reconstruction using the IPC framework to become a subnet under Filecoin.It will continue to provide cache acceleration services to the Filecoin network and serve as the data access subnet for Sinso V2.

Sinso V2 operates with either Ethereum or Filecoin as the root chain,relying on the security provided by the root chain.It functions as a Layer2 solution with access control, state verification,and task scheduling capabilities.It schedules computational tasks and validates their results within the network.Access control based on addresses is employed for data and computational tasks to ensure privacy and security.Furthermore,Sinso V2 implements a reasonable incentive mechanism to maximize the interests of data providers and computational power providers.

Architecture

- **decentralized edge computing system** :

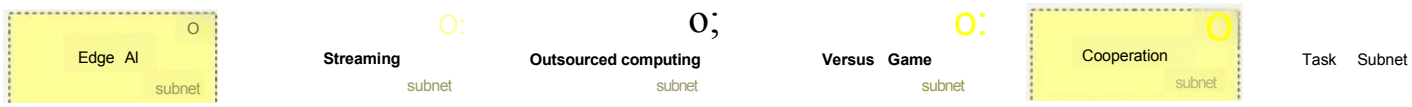
It is implemented based on IPC (InterPlanetary Consensus).

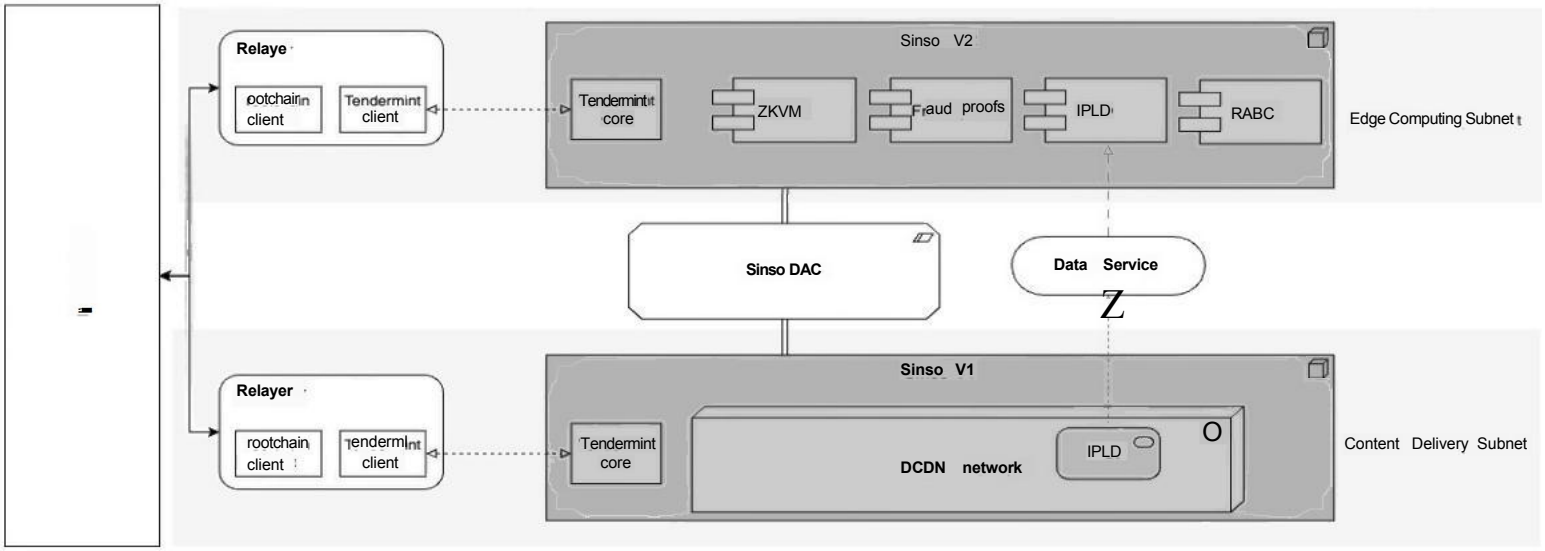
- **Security of relying on the root chain:**

Sinso V2 utilizes either Ethereum or Filecoin as the root chain.As a Layer2 solution,it possesses access control,state verification,and task scheduling functionalities.It schedules and verifies computational tasks within the network while implementing address-based access control for data and computational tasks to ensure privacy and security.Additionally,it provides a fair incentive system to maximize the interests of data providers and computational power providers.

- **Ensuring the security of the root chain (Filecoin)** ;

The communication between the subnets in the IPC architecture relies on validator nodes to maintain.These validator nodes operate simultaneously in both the parent and child subnets.The root chain communicates with the subnets through a Relayer,which forwards messages based on the block structure and consensus mechanism of the parent and child subnets.The underlying consensus mechanism for the subnets uses Tendermint,while the InterPlanetary Consensus (IPC)basic messaging and communication between subnets are managed by the Interconnection Gateway Architecture (IGA).





InterPlanetary Consensus

What is IPC?

IPC(InterPlanetary Consensus)is a framework that enables network horizontal scalability by allowing customized subnets to be deployed based on different functional requirements.IPC achieves networkscalability through the creation of subnets using permissionless blockchain subsystems.These subnets can be tailored to specific needs,providing a flexible and adaptable environment for decentralized applications and systems.

IPC(InterPlanetary Consensus)facilitates network scalabillty by organizing subnets in a hierarchical structure.A parent subnet can generate an infinite number of child subnets.Within a hierarchical subsystem,subnets can seamlessly communicate with each other,reducing the need for cross-chain bridges.

The consensus mechanism of the subnets is customizable and can leverage the security features of the parent network.The parent network can shard transaction tasks,including computational tasks.For example,a separate subnet can be created for each battle in a gaming application or for distributed computing in edge computing scenarios.

IPC is designed specifically for edge computing systems, providing a tailored solution for their unique requirements.

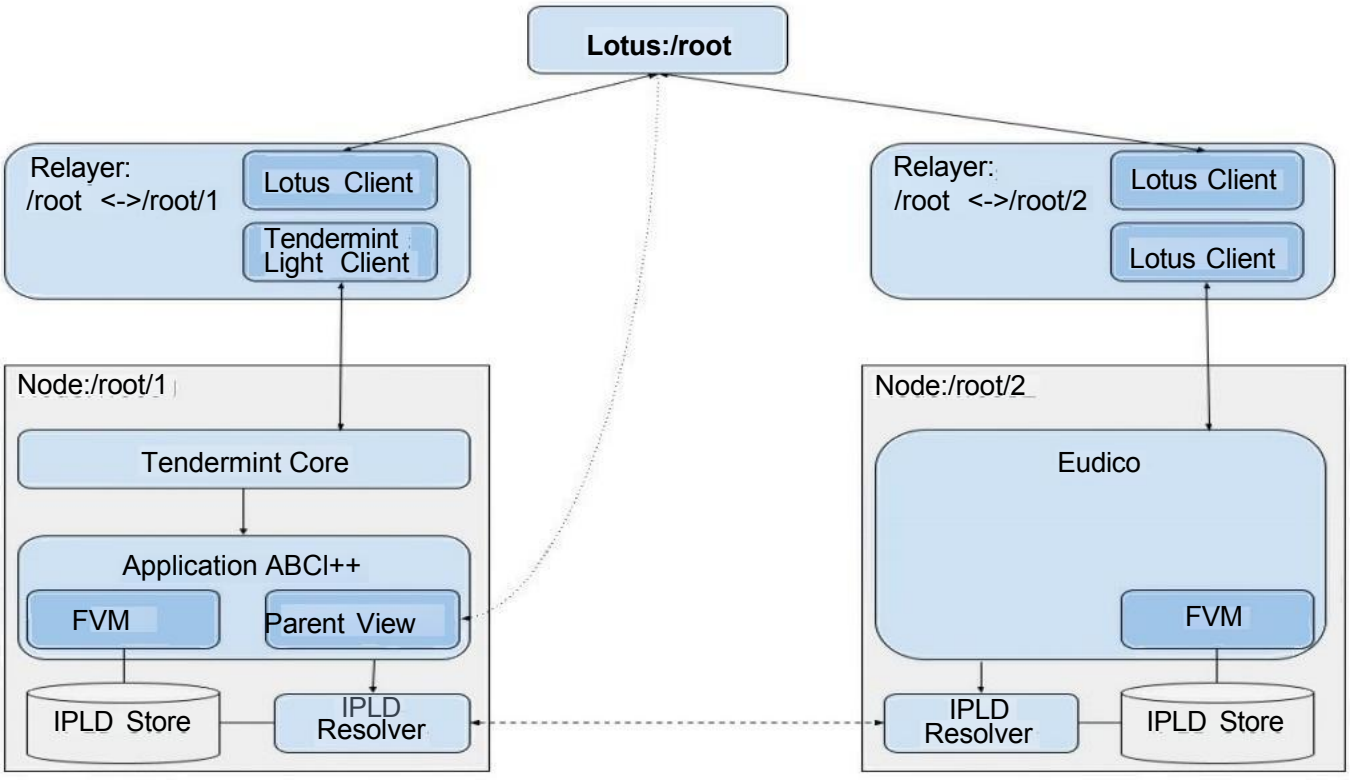
Characteristics of IPC? *(Cont.)*

IPC (InterPlanetary Consensus) is highly customizable without compromising security. One of its key features is the ability for subnets to have their own block structure and consensus algorithm while inheriting the security features of the parent subnet. Communication between subnets is native, eliminating the need for cross-chain bridges. This greatly reduces the risk of attacks and provides a protocol foundation for the infinite scalability of subnets.

The network driven by IPC will also have the capability to dynamically adjust its throughput by generating and closing temporary subnets as needed. Developers will be able to fine-tune the network according to their requirements, including customizing gas plans, native tokens, membership rules, and block times of less than 1 second. This means that the Sinso V2 edge computing network can allocate appropriate subnets based on the cost and performance requirements of each task during task scheduling. It can also reclaim resources after task completion, ensuring an optimized solution for computational tasks.

What is IPC? (Cont.)

IPC consists of a link between the parent and child nodes of the validator node,which runs on both the parent subnet and the child subnet to ensure that the checkpoint state of the child node is stored appropriately at the correct time in the parent node.The validator node is represented by the Tendermint core,ABCI++ (Application?BlockChain?Interface),Filecoin Virtual Machine (FVM),the IPC Subnet Actor (ISA),the IPC Gateway Actor (IGA),Gateway actors and Relayer. Tendermint Core is the Byzantine Fault Tolerant (BFT) consensus engine for blockchain.



IPC *(Cont.)*

IPC uses the ABCI++ interface to interact with the application running on the nodes. ABCI is a concept in Tendermint that defines the interface between the consensus engine and the application, determining when consensus can be reached during blockchain execution. The implementation of the ABCI++ interface is used to handle IPC ledger logic and transaction processing using the Filecoin Virtual Machine (or Ethereum-compatible FVM).

ABCI can pass checkpoint headers to the parent subnet and collect relevant signatures using the ledger. Task contracts run in the FVM of the validator nodes, and contracts running within the FVMin the subnets can provide high transaction throughput. This is suitable for various scenarios that require high throughput, such as machine learning, DAOs, gaming, and big data processing, all of which fall within the realm of edge computing.

The IPC Subnet Actor (ISA) and IPC Gateway Actor (IGA) are responsible for communication between subnets. They must adhere to the consensus of the parent and child subnets, subscribe to events, repackage messages in the appropriate format, and resend them.

access control

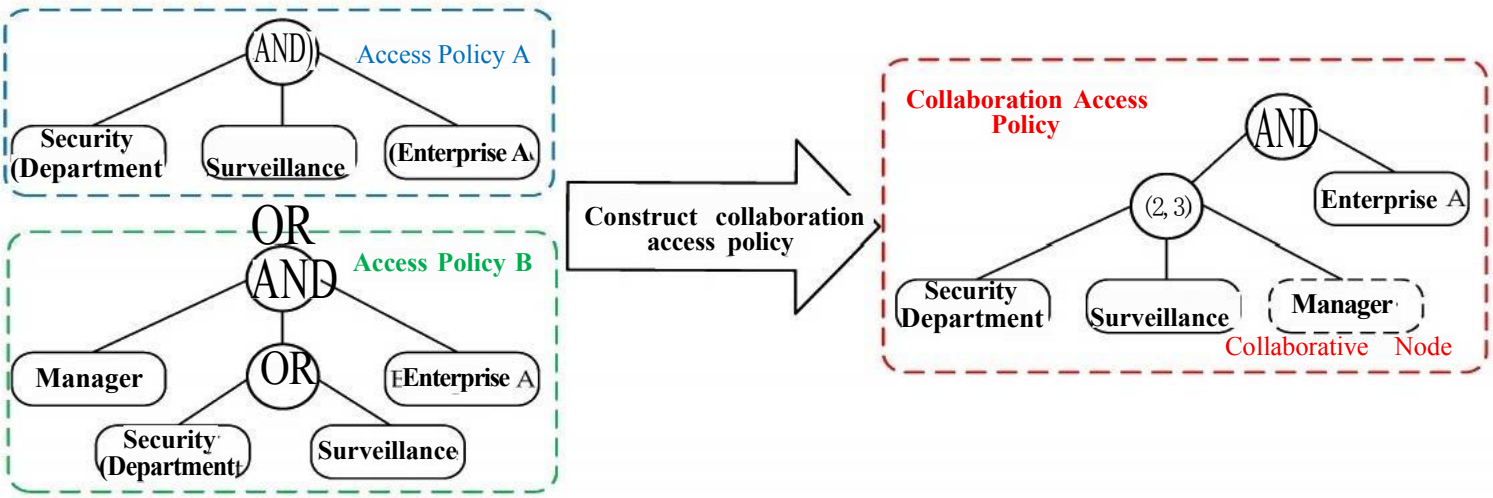
Access control is a key technology and method to ensure system security and protect user privacy. Currently, the more popular access control schemes include attribute-based and role-based access control, among which, Attribute-Based Access Control (ABAC) can be well suited for distributed architectures and achieve fine-grained data sharing. Attribute-Based Access Control.

With the dynamic and decentralised nature of edge computing, CPABE's adaptability makes it an ideal access control solution, providing a high degree of security and customisability for resources and data in edge computing environments.

Sinso V2 we choose adopt attribute-based encryption (CPABE) as the access control algorithm. In decentralised edge computing scenarios,which may involve diverse end-devices,users and resources,the flexibility of CPABE allows users to achieve fine-grained access control on the data of each participant based on their dynamic set of attributes.This feature makes it easy for data owners to define personalised access policies based on different end-devices,user identities and real-time environment changes,providing more granular and personalised privilege management.

In addition

CPABE provides a level of security that encrypts and protects data,ensuring that only users who meet the appropriate attribute criteria can decrypt and access the relevant resources.



stateful verification

In Sinso V2, we consider the integrity of the data and the integrity of the computational tasks to maintain the privacy and security of the participating parties. On the data input side, in order to prevent users from entering malicious data or attacks such as ddos, we use zkoracle to bring the data from the data input side securely into the system.

In the computing node, we consider multiple verification methods to cope with a variety of scenarios, including the use of validium for small computing scenarios and zkvm for complex computing scenarios, and their generated zk proofs are verified by the validator, and in most of the weak security requirement scenarios, We directly use Optimism mothed, where the calculator generates the fraud proofs and once the the calculator has fraudulent behaviour immediately punish it.

InterPlanetary Linked Data

ABCI:

One component of Tendermint Core is the Application BlockChain Interface(ABCI),which is an interface that defines the boundary between the replication engine (blockchain)and the state machine (application).Using the socket protocol,a consensus engine running in one process can manage application state running in another process.ABCI applications contact IPLD(InterPlanetary Linked Data)parsers and stores to read and write data,and users exchange data between subnets in IPLD format.

InterPlanetary Linked Data (PLD)is a data model for contentaddressing networks that allows for seamless retrieval of data across protocols by binding hash-based data structures together through common namespace.The entire system of Sinso is based on the libp2p protocol,and there is a free-for-all between the compute layer and the CDN layer to utilise the IPLD for addressing

Actor model *(Cont.)*

◆**Storage Node Providers:** Storage Node Providers are the builders of SINSO Getway. They profit from providing nodes, cache storage and bandwidth resource services to their users. The difference is, the storage node service provider can also act as an arithmetic service provider, who can be allowed to handle local Data.

◆**Validator Nodes:** Network validation nodes, i.e. validator nodes in Sinso Gateway, include 1. keeping accounts for storage nodes, recording the details of every transaction in the network, and providing the necessary evidence and data support for subsequent validation and auditing.2. undertaking traffic processing work to ensure unimpeded communication between all nodes in the network. At the same time, they also need to respond quickly to requests from other nodes and handle various network events and abnormalities in a timely manner to ensure the normal operation of the network.3. Monitor the heartbeat status of the network in order to detect and handle any abnormalities that may threaten network security in a timely manner.

◆**Upload Nodes:** The upload node is still responsible for the task of packing and generating blocks in the SINSO Getway network. They maintain the entire blockchain network and provide communication protocol interfaces for registration. The data submitted by Upload Nodes will be through orderly format processing, which may be also combined with industry-specific standard communication protocols. Data Protocol Stacks are important upload nodes. According to the vPoS (Value Proof of Stake) consensus of the SINSO network, the Data Protocol Stack Nodes need to have storage, computing resources and workload, as a guarantee before they can stake the corresponding amount of SINSO tokens. And meanwhile, they need to stay online.
The Upload Nodes participating in the network can obtain a multiplied reward specifically forvalid data, and meanwhile bear the risk of being fined and confiscated.

Actor model *(Cont.)*

◆**Node Guarantor:** The Node Guarantor refers to the account that provides a guarantee for any one or more nodes in the SINSO Getway network. The guaranteed node can only be an upload node. Any account with SINSO tokens can become a guarantor, and its SINSO tokens can be used as a collateral asset. The guarantor obtains income by providing a guarantee for the nodes, and meanwhile shares proportionally the risk of the nodes ' being fined.

Node Guarantors play an extremely important role in the SINSO Getway network. They guarantee the stability and security for nodes and provide a source of income and a means of transaction for accounts with SINSO tokens in the network. Only through effective guarantee services and risk control can the stable and sustainable development of the SINSO Getway network be ensured.

◆**Users:** Users can use network resources in the Sinso Gateway subnet as well as computing resources in the Sinso V2 subnet.

◆**Arithmetic Node Providers:** Arithmetic node providers form the edge computing part of Sinso V2, which can be divided into terminal nodes, aggregation nodes, computation nodes, and verifier nodes. Nodes with strong computing power are often used as computing nodes or validation nodes, nodes with large bandwidth become terminal nodes, and the CDN nodes of Sinso V1 can be used directly as terminal nodes. Aggregation nodes and validator nodes are the security guarantees for the computation process, and they need to have a sufficient number of pledges in order to become such nodes. Verifier nodes need to understand the fraud proof and validity proof and other security schemes, can guarantee the integrity of the data and computation, and promote the development of system expansion.

Edge Computing

Edge computing is a decentralized computing architecture that shifts the processing of applications,data,and services from central network nodes to edge nodes in the network.Edge computing breaks down large-scale services that were previously primarily handled by central nodes into smaller and more manageable parts,which are then distributed to edge nodes for processing.Edge nodes are located closer to user terminals,enabling faster data processing and transmission speeds while reducing latency.In this architecture,data analysis and knowledge generation are closer to the source of data,making it more suitable for handling big data.

Edge Computing

The three main advantages:

AReal-time data processing:edge computing has faster response time and real-time performance because data processing and analysis tasks can be performed at the edge computing nodes,reducing the time for intermediate data transmission. This is especially important in application scenarios that require fast feedback,such as Telematics,games,etc

Improved Security:Edge computing avoids the risks of uploading all data to the cloud,such as data loss and leakage,by processing data locally.Local processing helps protect the security and privacy of the data and reduces potential risks due to network transmission.

▲**Reduced cost and energy consumption:Edge** computing reduces the need for extensive network bandwidth,lowering transmission costs and relieving network bandwidth pressure.Furthermore,edge computing is a "small-scale"processing approach,which reduces the cost of processing data on local devices while lowering energy consumption.This enhances computational efficiency.

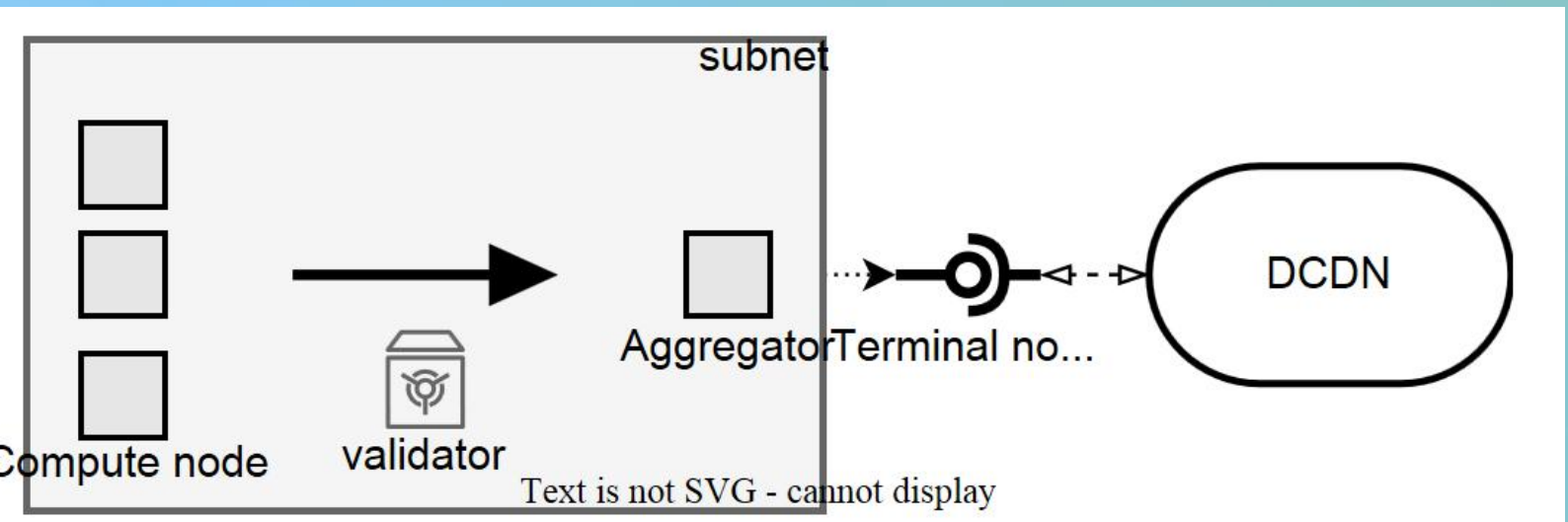
Edge Computing *(Cont.)*

Edge computing, on the other hand, can provide faster data processing and transmission capabilities. Integrating blockchain and edge computing into a single system enables reliable access to and control of networks, storage and computation distributed at the edge, thus providing large-scale network servers, data storage and validated computation methods at the proximal end in a secure manner.

CDNs must evolve from traditional IO-intensive systems centred on caching services to edge computing systems, architecting content computing networks to address future connectivity challenges posed by traditional IoT and Depin In this process, CDN networks that only provide storage and bandwidth resources are no longer sufficient to provide complete services, and must be iterated to further provide arithmetic resources.SInso V2 responds to the development of Depin, and builds an arithmetic provider network based on the SInso Gateway, which originally provided storage and bandwidth resources, i.e., a decentralised edge computing network.

Components for Edge Computing

The centralised edge computing has developed a more complete component architecture,in our design,the role of the cloud core is removed from the decentralised edge computing,the role is responsible for the deployment of policies and complex tasks.The main components in the edge computing model of Sinso V2 are end nodes, aggregation nodes,computation nodes,and **verification nodes.**



Component*(Cont.)*

◆**End Node:** The end node is the data provider, the data can be from multiple sources, either directly from the CDN subnet according to the computation task requirements, or it may be the on-chain data, the node is responsible for running the chainlink environment, which processes the on-chain data and then securely transmits it to the computation node. The end node may also be the data generating node, in the depin scenario, the end node may consist of various IoT devices (e.g., sensors, RFID tags, cameras, smartphones, etc.)
It mainly fulfils the function of collecting raw data and reporting it, as well as uploading it to the cdn network for storage and transmitting it to the computing nodes for computation.

◆**Aggregation node:** It is responsible for distributing tasks and aggregating computation results. Aggregation node receives the tasks in the network, designs the computation strategy and schedules the tasks according to the requirements of the tasks, requests the data links from the terminal nodes and distributes them to the corresponding computation node roles after encrypting the permissions from the terminal nodes, and ABAC can make only the nodes that meet the requirements to receive the computation tasks. After the completion of the edge computing task, merge in the aggregation node, and the aggregation node for analysis and processing before the final delivery.

system flow

In Sinso V2's edge computing network,all computational tasks are performed by computing power providling nodes,each of which is assigned roles based on its performance and pledge.

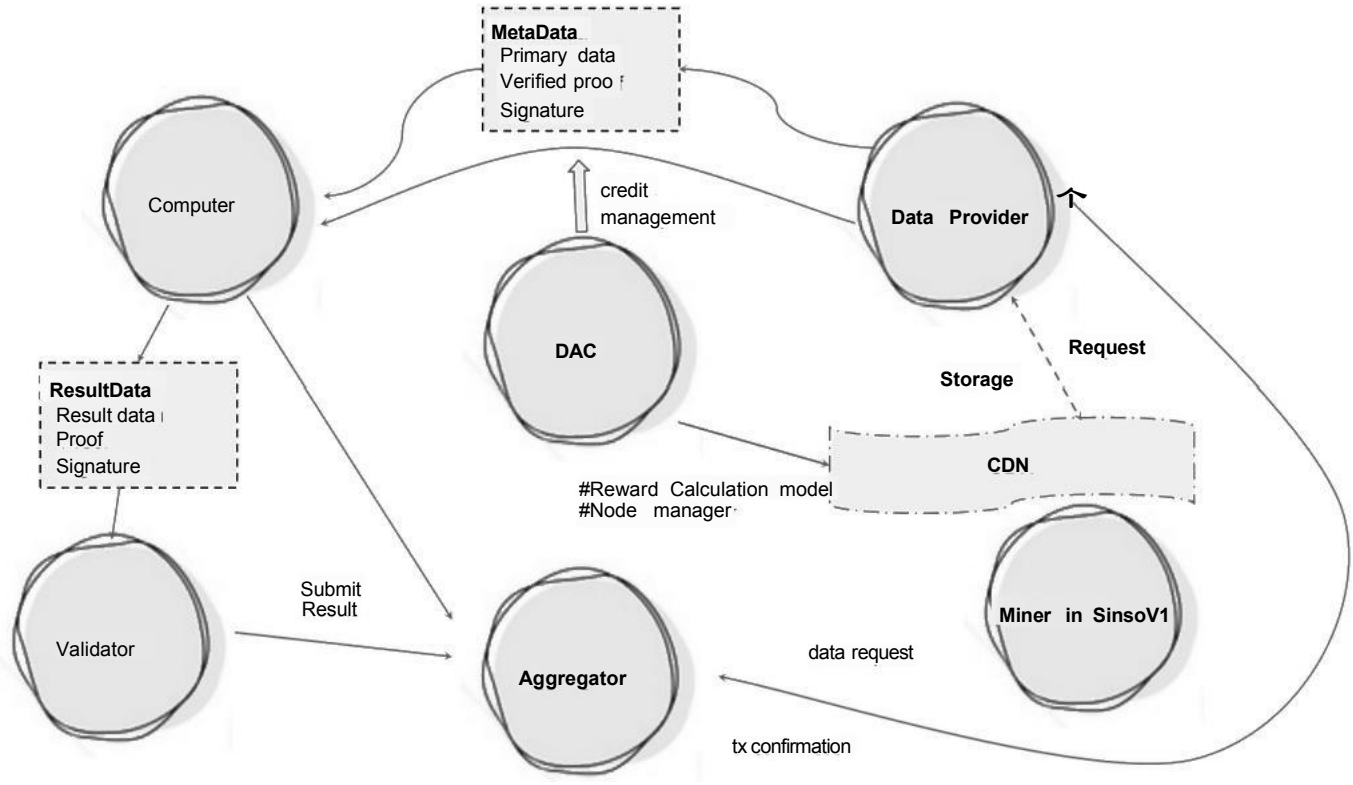
The aggregator is the IPC architecture that assumes the role of the verifier node,which is responsible for the generation of the computational subnet and the connection with the sinsov2 subnet.The aggregator nodegenerates the corresponding appropriate IPC subnet information to form a new computational subnet during the task scheduling process according to the task size and security requirements.When the task is large,the aggregator can even choose to generate multiple computational subnets,e.g,one computational node,oneverifier,and one terminal node into one subnet.

System flow(cont)

The aggregator performs task scheduling and data request based on the computational task design policy (in some simple scenarios the aggregator can be not a role but an automation contract)obtained from the network.After receiving the request,the end node requests the data from the dcdn,gets the data from the nearest data source and processes it,if there is a security requirement it also needs to generate a security certificate and the end node will set the access control to the data according to the aggregator's requirement and will package the completed result into a transaction certificate and send it to the aggregate node.

The DAC monitors the data provisioning process during this process, which we call trust management.We call it trust management and if the end node provides malicious data,it will be penalised and recorded.

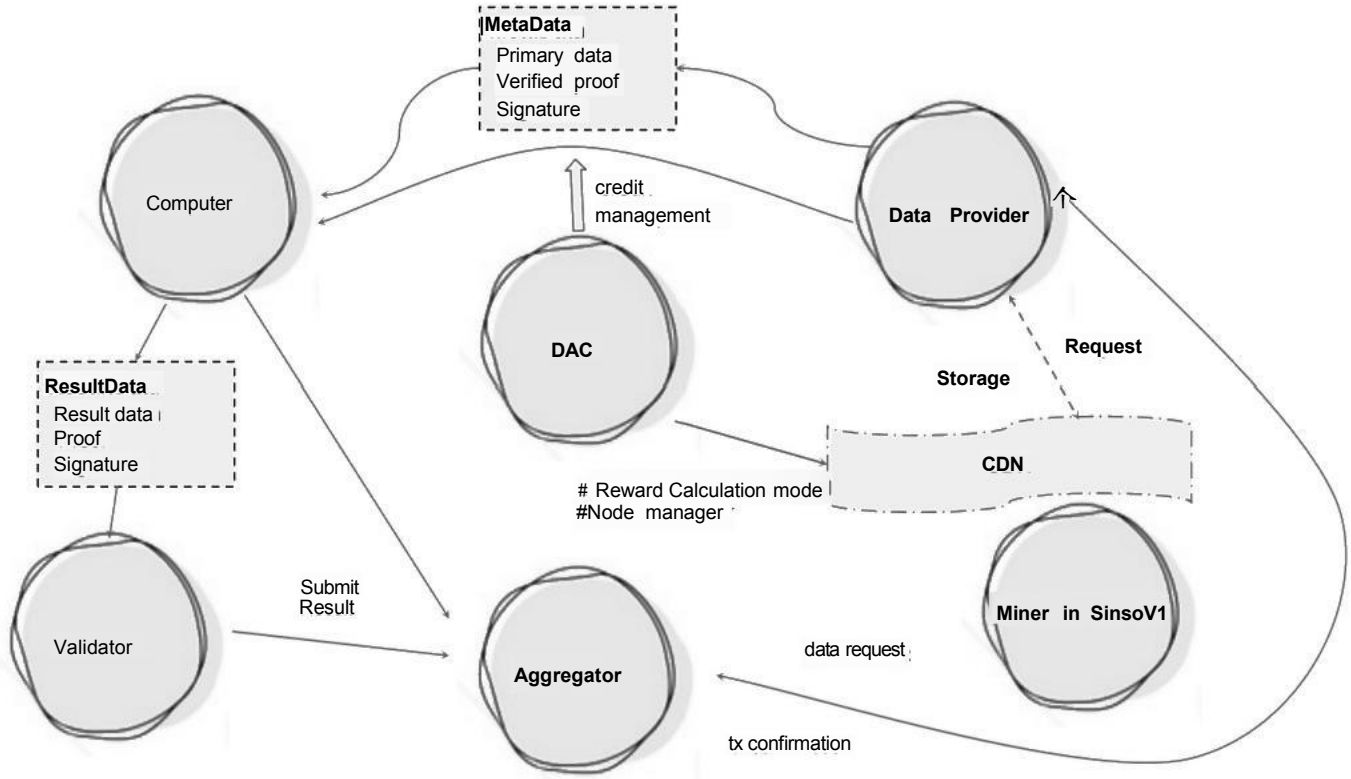
系统流程



System flow(cont)

The computation node accepts the Metadata transmitted by the terminal node and the packaged task to start the computation,generates the corresponding proof of results according to the security requirements after the computation is completed,and sends the results packaged with the computation results to the verifier,who sends the results to the aggregator after completing the verification.In the case of outsourced computation without strict security requirements,the calculator can directly accept the task,and the completion of the calculation accepts the validator to verify the completion of the task and submit it to the task publisher

系统流程



Securi ty Vali dation for Decentrali sed Edge Computing

In Sinso V2 subnetwork,we mainly consider data integrity at the data provider side, computational integrity at the edge computing side,access control based on role attributes, and security vulnerability and risk management.

Role-based access control is based on the values of the attributes of different nodes,such as pledge,performance and resources,to determine theirrespective permissions,so as to isolate different participants from each other in terms of data,computation,management and aggregation.
Sinso V2 uses ABAC to address access control, while security breaches and risk management are handled by authenticator nodes.

Data Integrity

Data integrity refers to the overall accuracy. completeness and consistency of data, specifically the absence of accidental or malicious modification,destruction or alteration of data during transmission,storage and processing.It is maintained by a set of processes,rules and standards implemented during the design phase.When data integrity is ensured,the information stored in the database will remain intact Accurate and reliable,regardless of how long it is stored or how often it is accessed

The requirement for data integrity is that the data is not subject to unauthorised tampering or can be detected quickly after tampering.

Data Integrity *(Cont.)*

Type

◆**Physical Data Integrity:** is the ability to access accurate data. This includes data access, Completeness of data, and protection against factors that could lead to data errors. We have ensured the reliability of the end nodes by being optimistic in the early days of Sinso V2, and in the future, we will propose to require the end nodes to use a secure TEE environment. Physical data integrity is more evident within the DCDN subnetwork, where the CDN node is the data caching centre and the real data provider, and the node needs to build in protections against downtime and hacking.

◆**Logical Data Integrity:** is the ability to maintain data consistency and accuracy over time.

① Entity integrity: correct identification of data, including protection against duplicate or null values

② Domain Integrity: Ensuring data accuracy, including defining acceptable values

③Referential Integrity: defines how data is stored and used in the database and is only available after the

④An unauthorised change, addition or deletion can only take place

⑤ User-defined integrity: Limit user-created data to requirements

Data integrity *(Cont.)*

The Sinso DAC guarantees initial physical data integrity by setting strict node performance requirements. Mastering Ether offers two data integrity proofing schemes, authenticity proof and Trusted Execution Environment (TEE).

Based on many attestation techniques (e.g., digital signatures, zk, oracle), they efficiently transfer the required trust from the data transmitter to the attestor (the provider of the attestation method). By verifying evidence on the chain, smart contracts can verify the integrity of data before using it.

Trusted execution environment is hardware based reliability. The Trusted Execution Environment based Verified Data Feed Predicate Machine System is an accepted scheme that uses a hardware based security to verify data integrity.

Intel SGX（Software Guard eXtensions）：

Intel's SGX (Software Guard eXtensions) to ensure that responses to HTTPS queries can be verified as trustworthy. SGX also provides integrity guarantees so that applications running in the secure zone are protected by the CPU from tampering by other processes. It also provides confidentiality by guaranteeing that the state of an application running in a secure zone is unknowable to other processes. Finally, SGX determines that an application is running in the secure zone by generating a digital signature (securely determined by the hash value of its construction result), which is ultimately verified for authenticity by an on-chain smart contract.

How Sinso chose：

Sinso combines the advantages of both solutions and chooses the zkoracle solution. Due to the complexity and development difficulty of TEE, and the fact that it will make us need to increase the mandatory conditions for each participant, the TEEscheme will not be adopted in the short term.Sinso requires the end nodes to verify the authenticity of the data, which is achieved by generating verifiable proofs through zero-knowledge proofs. The end node obtains the data from CDN subnet or chain through the prophecy machine scheme, generates the authenticity proof, and then sends the original data MetaData, the verifiable proof, and the signature package to the computing node.

Given the complexity of zero-knowledge proofs, Sinso also offers a less secure scheme where end nodes can generate simpler fraud proofs, if an end node is found to have committed fraud, it will be penalised and points will be deducted from the trustworthiness management repository maintained by the DAC, credits are public and fraud may result in the end node no longer being trusted and participating in the activity.

computational integri ty

Sinso's computational integrity design focuses on the edge computing subnetwork,with the main considerations of computational accuracy and confidentiality.The accuracy of computation in turn includes the accuracy of the task and the accuracy of the result.When the computational tasks and data are received by the computational nodes,it is important to ensure that the computational process uses secure data and packaged computational tasks provided by the end nodes,due to the fact that malicious nodes can steal data and modify the running computational tasks in the computational process, thereby interfering with the accuracy of the computation.

After the computation node completes the computation task,the output result may be wrong or even with malicious content,including nodes reusing data or using incomplete data to interfere with the result,malicious nodes embedding malicious data,noise and so on in the result data,which can lead to inaccurate result data. Confidentiallty of computation mainly refers to ensuring that the computational model is not leaked because the computational tasks are provided by the aggregation nodes or task publishers and they do not want any other person to steal their computational strategies for profit,so the computational model is required to be handled confidentially after the task is published.

Computational Integrity *(Cont.)*

Zero Knowledge

Zero-knowledge proof is a cryptographic concept aimed at proving the truth of an assertion by a party without revealing specific information. A Prover can give a Verifier proves that it possesses a certain piece of information, but does not need to disclose the actual content of the information. This means that the verifier can confirm the correctness of a claim without having to be informed of the specific content of the key information.

Zero-knowledge proofs have a wide range of applications in several fields, including cryptography, blockchain technology, security protocols, and so on. In the development of blockchain in recent years, zero-knowledge proofs have received a lot of attention and have become one of Layer2's solutions, mainly used to achieve anonymous transactions, protect private data, validity verification, block compression and so on.

Computational Integrity *(Cont.)*

Zero Knowledge

The main technical frameworks include:

★ zkSNARK (Zero-Knowledge Succinct Non-interractive Argument of Knowledge)

★ zkSTARK (Zero-Knowledge Scalable Transparent Argument of Knowledge)

zk-STARKs enables a verifier to verify the truthfulness of a party's claim about a claim by generating small and verifiable proofs that require a very small amount of computation to verify the truthfulness of a party's claim about a claim without having to know the actual information content. zk-STARKs is a more novel zero-knowledge proof technique with higher scalability and suitability for interactive proofs. It allows generating verifiable proofs in larger and more complex computations while maintaining a high degree of privacy.

Computational integrity *(Cont.)*

ZKVM:

ZKVM is a virtual machine based on zero-knowledge proofs
It will be zero-knowledge proof of ZK and virtual machine VM Combined.

Important components: :

★ Compilers that compile high-level languages such as C++, Rust, etc. to generate intermediate representations (IRs) for proofing in the ZK system.

★ Instruction Set Framework ISA (Instruction Set Architecture) The instruction set is a collection of instructions that directs the CPU to perform operations.

The purpose of zkVM is: given an initial programme, an initial programme input, an initial internal machine state, proving the effective implementation of the above VM.

Computati onal integri tycont)

zkVMs four main stages

- **Setup phase:** according to the parameters (e.g. maximum number of trace rows,fixed number of columns,hash function,etc.),the Proving key and Verification key are obtained.

- **Generate Witness Phase:**(Executor)generates an execution trace (i.e.,witnesses)based on the program and program inputs.The execution trace contains:the execution of the program and additional information to help constrain the validity of the execution.The Witness generation phase also includes the work of slicing and dicing the programme for subsequent parallel proofs.

- **Proving phase:** generate proof based on execution trace and Proving key.

- **Verification phase:** generate the result Y/N of whether the verification passes or fails according to the proof and Verification key

Phase	Setup	Witness Generation	Proving	Verification
Input	Parameters	Program Inputs	Execution Trace **Proving Key**	Proof **Verification Key**
Output	**Proving Key** **Verification Key**	Execution Trace (aka Witnesses)	Proof	Y/N

Computational Integri tycont)

Validium

Validium is a scaling solution that enforces integrity of transactions using validity proofs like ZK-rollups, but doesn ’t store transaction data on the Mainnet.These”validity proofs”can come in the form of ZK-SNARKs (ZeroKnowledge Succinct Non-Interactive Argument of Knowledge)or ZK-STARKs(Zero-Knowledge Scalable Transparent ARgument of Knowledge).While offchain data availability introduces trade-offs,it can lead to massive improvements in scalability (validiums can process ~9, 000 transactions,or more,per second).

System type	Technology properties	Security guarantees	Costs
Rollup	Computation proven via fraud proofs or ZK-SNARKs, data stored on L1	You can always bring the asset back to L1	11 data availability + SNARK-proving ol redundant execution to catch errors
Validium	Computation proven via ZK-SNARKs (can’t use fraud proofs), data stored on a server or other separate system	Data availability failure can cause assets to be lost but not stolen	SNARK-proving
Disconnected	A separate chain (or server)	Trust one or a small group of people not to steal your funds or lose the keys	Very cheap

Computational Integri tyccont)

Validium

Block producers on Validium are not required to publish transaction data for transaction batches (only block headers).The Data Availability Manager in Validium certifies the availability of off-chain transaction data by signing each Validium batch.These signatures constitute a "proof of availability"that the on-chain validator contract checks before approving status updates.Some rely on trusted parties to store status data,while others use randomly assigned validators.In Sinso V2,if a Validium off-chain computational scaling scheme is used,in conjunction with zkevm,the Sinso DAC acts as the scheme's Data Availability Committee,storing copies of the state and providing proofs of data availability.

System type	Technology properties	Security guarantees	Costs
Rollup	Computation proven via fraud proofs or ZK-SNARKs,data stored on L1	You can always bring the asset back to L1	11 data availability + SNARK-proving ol redundant execution to catch errors
Validium	Computation proven via ZK-SNARKs (can't use fraud proofs),data stored on a server or other separate system	Data availability failure can cause assets to be lost but not stolen	SNARK-proving
Disconnected	A separate chain (or server)	Trust one or a small group of people not to steal your funds or lose the keys	Very cheap

Computational Integrity *(Cont.)*

Importance of DAC

The DAC plays an important role in the design of securing computational integrity, including the task of securing data integrity.The DAC is involved in the selection of the overall integrity solution for the design, as well as in the election and supervision of the validation nodes. Under the optimistic scenario, the DAC decides the penalties for nodes that break integrity, maintains a trust management mechanism for end nodes and compute nodes, and makes the trust scores of all nodes public.

Under the scheme of proof of validity, one is to choose the zero-knowledge proof scheme of the system, including the circuit approach or the VM approach or the simpler Validium scheme, and is also an important participant in Validium, and the other is to deal with the erroneous validation results submitted by the validators and the network monitoring and alerting, to deal with nodes with erroneous validation results, and to make decisions on major incidents in the network.

Computational Integrity *(Cont.)*

Security of the model

With the development of AI, AI will inevitably be combined with blockchain, which can provide privacy security and distributed computing for the training and reasoning process of AI models.

Machine learning model security and privacy requirements are summarised in three properties.

- confidentiality
- integrity
- availability

CIA models in machine learning. The confidentiality of a machine learning model requires that the machine learning system must ensure that unauthorised users do not have access to private information in the system, including both the training data of the model as well as the model's architecture, parameters, etc.; integrity requires that the predictions of the model must not deviate from the expected results; and usability requires that the machine learning system is able to provide normal services in the face of abnormal inputs or even malicious inputs.

For the time being, the computational integrity assurance scheme designed by Sinso can be compatible with the privacy and security needs of machine learning models. In the future, based on the edge computational integrity scheme, we will design a more detailed processing scheme in conjunction with the AI training and inference process.

Computational Integrity *(Cont.)*

Outsourced Computing

Outsourced Computing is a method of delegating computational tasks to a third party service provider for processing. In the edge computing subnetwork, there will be a large number of computing tasks that will be outsourced computing, mainly under the condition that the task publisher provides the computational strategy and data, and the task publisher does not want the computing power provider to steal the computational model and data provided by him.

To address the confidentiality issues arising therein, the secure outsourced computation considered by Sinso V2 takes into account various approaches such as fully homomorphic encryption, FHE, differential privacy, multiparty computation, MPC, and so on.

The validation process of machine learning model training placed on the chain will require high performance requirements and incur high costs, Sinso V2 only provides validation services for the model inference process. For some outsourced computing scenarios, including versus games, generic two-party transactions, and model inference for machine learning, Sinso V2 provides a framework for model hosting services.
, such as AIGC's Machine QA task, GPT models can be hosted directly by compute nodes to serve web3 users and protect user privacy and security during model inference.

Conclusion

- Sinso will port the overall architecture to the IPC architecture and divide the entire network into a DCDN subnetwork and an Edge Computing subnetwork,and the Sinso V1 in-role model will be retained.and given more permissions and tasks.The main development task of Sinso V2 is the Edge Computing Subnetwork,a marketplace service for computing resources developed on the basis of the DCDN network

The edge computing subnetwork accesses data from the DCDN subnetwork,the aggregation nodes of the edge computing sulbnetwork are responsible for task collection and scheduling,simple tasks are assigned separate vm runtime environments within the subnetwork,complex tasks are created by the aggregation nodes through task publishing,DAPPs are considered complex tasks in the Sinso network and run in a separate subnetwork.

Conclusion

- The terminal node is responsible for receiving the computing policy and data request from the aggregation node,obtaining the data from the DCDN subnet, combining it with the computing policy and packaging it to complete the access control to send it to the corresponding computing node,and the computing task is run by the computing node,and the verifier node is responsible for verifying the data integrity of the terminal node and the computing integrity of the computing node during the whole computation process.Sinso will become a complete set of computation scaling solution on Filecoin.


SINSOV2(filX)