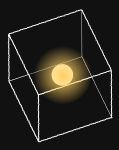
SASEUL GOLD

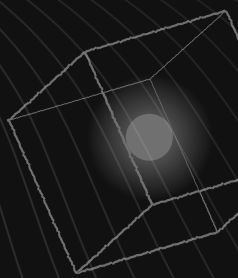
2024. 12. 20

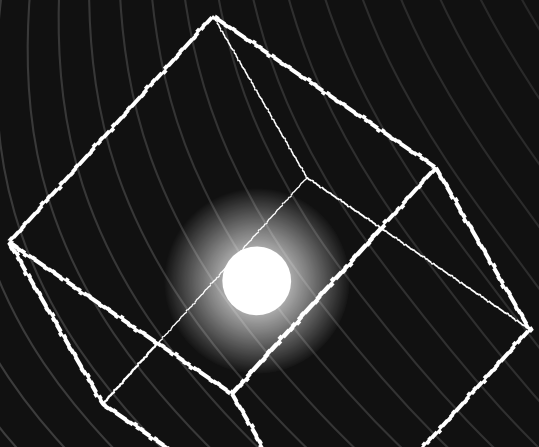
Overcoming the Limitations of Blockchain Technology

SASEUL GOLD, look beyond the possibilities.



Saseul Gold, Infinite Value Expansion through DEX





Ver 1.0.0

Author Information | saseul gold foundation

Saseul gold, look beyond the possibilities.

What is SASEUL GOLD?

Saseul Gold is a DEX-specific blockchain developed to complete the tasks that the SASEUL Network is trying to solve, referring to the direction and purpose of the SASEUL Network. It is an innovative next-generation transaction-specific blockchain that pursues the ideas presented by the existing SASEUL Network and has advanced performance, security, and scalability. While maintaining stability, Network performance, transaction processing speed, and user experience have been significantly improved. In particular, transaction bottlenecks have been greatly improved, and the system has been designed to be optimized for token exchange.

In particular, we merged the resource block and resource concepts, which were the core of the existing consensus algorithm, with the main block and introduced Proof Of Historical State (POHS) based on the Raft Consensus. This new consensus algorithm replaces it.

Through this, we simplified the system structure and maximized the efficiency and scalability of the consensus process. In addition, users are divided into a node that verifies registered blocks (Anchor Node) and a Gravity Node that performs all functions, improving the efficient use of computing resources according to the nature of the system they operate.

background

Blockchain has brought innovation to various industries by providing decentralization, transparency, and security based on distributed ledger technology. Especially in finance, payments, and supply chain management, blockchain technology maximizes efficiency through reliable data management and smart contracts that run without intermediaries. However, the current blockchain industry still has several structural limitations:

1. **Scalability Issues:**
   Most blockchains are limited in the number of transactions they can process (TPS). When network traffic surges, this leads to bottlenecks and transaction delays.
2. **High fees:**
   When there are many transactions, users must pay higher fees, a significant obstacle to actual use.
3. **Complex user experience:**
   Complex user experience: Blockchain technology is not intuitive for users, and most blockchains do not provide interoperability between networks, making them difficult for users to access quickly.

4 **Incomplete payment system:**

Incomplete payment system: Blockchain projects specializing in payments are relatively lacking, and the technical foundation to support real-time payments is often insufficient.

These limitations are preventing the expansion of blockchain's actual use and, in particular, act as a major barrier to revitalizing the payment market and token economy. Saseul Gold was created to overcome existing limitations in the blockchain industry and create a payment system optimized for actual use. Through technological innovation, we have maximized network performance and scalability and implemented a payment-oriented blockchain that anyone can use easily and quickly. Through this, Saseul Gold will provide a blockchain ecosystem that goes beyond simple blockchain and connects to the real world, establishing itself as a user-centered real-time payment platform.

Saseul Gold Development Motivation

Sasuel Gold is a project developed to improve blockchain's practical usability and technical limitations. Based on the technical foundation of the existing SASEUL blockchain, it was developed to become a blockchain specialized for DEX and payment systems and aims to solve the following problems:

1 **Blockchain optimized for payments**

Current blockchain payment systems do not satisfy fast transaction speeds and low fees. To solve these problems, Sasuel Gold sought to implement a payment-specific blockchain with high-performance transaction processing and a low fee structure.

2 **Improved scalability and performance**

To solve the performance bottleneck of blockchain and support large-scale transaction processing, Sasuel Gold supports SWAP between coins and tokens and Sasuel Gold SWAP between heterogeneous chains. It has introduced an efficient consensus algorithm that can support transactions. Sasuel Gold aims to maximize Network scalability and provide real-time payment capabilities.

3 **Maximize network efficiency**

To improve the blockchain's complex structure and resources, all infrastructure has been merged into the main block, and a single process-based system is provided through the Oracle Process. We also introduced Native Contract, which will greatly improve transaction processing performance. This speeds up transaction processing and optimizes the use of network resources.

4 Implementing interoperability and a multi-chain environment

Interoperability between multiple networks is needed for blockchain technology to spread further. Sasuel Gold supports efficient collaboration and expansion in a multi-chain environment by enabling data exchange between chains through the Bypass Operator.

Technical Vision and Goals

Sasuel Gold overcomes the limitations of blockchain technology and aims to be a payment-specialized blockchain that can be used in real life, realizing the following core values:

1 High-performance real-time payment

Transaction bottlenecks are eliminated, and real-time payments are implemented through the priority queue-based memory pool and Oracle Process.

2 scalable network

Multi-chain support and bypass operators enable data interconnection between multiple chains and maximize network scalability.

3 efficient system structure

It simplifies the complex structure of existing blockchains to reduce resource consumption and provides efficient block verification through the Proof of Historical State (POHS) consensus algorithm.

Sasuel Gold Network is an innovative blockchain-based platform that supports the development and operation of various DApps. Proper distribution of cryptocurrency coins is a very important factor in the stability and growth of the network. This white paper describes in detail the token distribution method of the Sasuel Gold Network and seeks to convey the vision and value of the network through this. The details and strategy of token distribution will contribute to the long-term development of the Sasuel Gold network.

Token Distribution Overview

The total coin issuance of Sasuel Gold Network is 3.5 billion. Of these, 500 Million coins will be pre-mined in advance and used to build initial infrastructure, provide liquidity, and promote listing on cryptocurrency exchanges.

This pre-mined coin will play a vital role in the early stages of the Sasuel Gold network. It will be used to build initial infrastructure, support the development of various DApps, and create an ecosystem. Additionally, we will be able to secure investor trust by encouraging healthy price formation through the provision of liquidity and promoting listing on exchanges.

The use of pre-mined coins is expected to contribute to the stable growth and long-term development of the Sasuel Gold network.

Initial infrastructure construction

The 500 Million coins allocated to build the initial infrastructure of the Sasuel Gold Network will play a very important role. This token will be used to support the development of various DApps and the creation of an ecosystem.

First, we plan to encourage dApp development on the Sasuel Gold Network by providing incentives to developers through Sasuel Gold coins. Developers will be rewarded with coins based on their contribution to the network, which is expected to encourage continuous development activities. We will also provide technical support and training programs to developers to improve their understanding of the network and help build their capabilities.

In addition, we plan to operate a compensation system using coins to attract early users. The more users participate and become active in the network, the more Coins they receive, which can encourage participation and activation within the ecosystem. At the same time, we plan to expand our user base through cooperation with partners and increase network utilization by providing them with various benefits.

This strategy of utilizing coins will be a key element for the stable and sustainable growth of the Sasuel Gold Network. By establishing an initial infrastructure, we will be able to build a strong DApp ecosystem and promote the participation of developers and users to achieve healthy network development.

Liquidity provision and listing

Sasuel Gold Network plans to pre-mine approximately 500 million coins out of the total issued volume of 3.5 billion and use them to promote listing on cryptocurrency exchanges and provide liquidity. By pre-mining coins, Sasuel Gold Network will seek listing on major cryptocurrency exchanges. Exchange listing plays a key role in increasing accessibility to investors and securing liquidity. The pre-mined coin will be used for fees and listing preparation costs required for the exchange listing process. Additionally, we plan to increase awareness and utilization of the Sasuel Gold Network by conducting continuous marketing and promotional activities through cooperation with exchanges.

Additionally, issued coins will be used to provide liquidity. Initially, a portion of issued coins will be provided to the market to secure sufficient liquidity and thereby induce healthy price formation. We will closely monitor market conditions and provide additional liquidity as needed. Through this, we can increase investor trust and build a healthy ecosystem.

In addition, the following measures are being prepared to protect investors and ensure reliability. First, we will ensure the reliability of transaction records by transparently disclosing all transaction details. We also plan to establish a strong security system to protect investors' assets safely. Lastly, through continuous communication with investors, we will share the current status of the network and future plans to strengthen our trust relationship.

This strategy of utilizing issued coins will support the healthy growth and long-term development of the Sasuel Gold network. It is expected that listing on exchanges and securing liquidity will increase investor accessibility and trust, and build a stable ecosystem through investor protection measures.

Conclusion

The Coin Distribution Model of the Sasuel Gold Network significantly increases the stability and growth potential of the network. Out of the total issued volume of 3.5 billion, 500 million issued coins will be used to build initial infrastructure, provide liquidity, and promote listing on major cryptocurrency exchanges, ensuring the successful launch of the Sasuel Gold Network.

Through a transparent and safe coin distribution and management strategy, we can secure the trust of investors and users, which will contribute to building a healthy ecosystem.

Sasuel Gold Network will continue to develop based on this coin distribution model and lead blockchain technology innovation. By providing a safe and reliable platform for users around the world, Sasuel Gold Network is expected to play a key role in the cryptocurrency market.

Changes in consensus algorithm

Change in consensus algorithm – introduction of POHS (Proof of historical state)

Sasuel Gold changed the resource block-based consensus structure to implement a new consensus algorithm called POHS (Proof Of Historical State). In the existing structure, blocks were created and verified based on network resources (blocks and resources), but this structure guarantees fast transactions in an ideal network environment and provides stable transactions in specific environments. Therefore, for the purpose of fast and reliable consensus, we implemented a consensus algorithm in a different direction that greatly improved the block creation speed.

The concept of POHS (Proof of historical state)

The formula for generating a POHS hash.

$$H_n = H(H_{n-1} || T_n)$$

H_n : n-th hash result
H_{n-1} : Previous step's hash result
T_n : The n-th state of the time or input data

- POHS is a hash generation method based on Historical proof and is an algorithm that proves temporal order during the block creation and verification process.
- Each block proves its temporal order through a unique hash, which enables fast and reliable consensus without the need for separate resource-based verification during the consensus process.
- By merging resource blocks, complex verification procedures were eliminated, and by introducing POHS, block creation speed, and network performance were dramatically improved.

smart contract

Sasuel Gold Network supports automated trading and contract execution through smart contracts. A smart contract is a contract written in code that automatically executes when certain conditions are met. These systems increase reliability and ensure transaction transparency without the intervention of intermediaries. Sasuel Gold Network's smart contracts can be used in various DApps and enable efficient transactions through direct interaction between users.

fee structure

Sasuel Gold Network's fee structure prioritizes transaction efficiency and user experience. Fees incurred when executing every transaction and smart Contract play an important role in maintaining the network's stability and security.

How fees are determined:

Sasuel Gold's transaction fees are determined by the base fee per signature. A typical transaction incurs a fee of 0.000005 SG.

When using the fee payment feature, a fee of 0.00001 SG will be incurred as two signatures are required. This means that users will incur additional costs if they choose to pay the fee.

Use of Fees:

The deducted fees enhance the network's operation and security, contributing to its continued development and stability.

It can also be used as an incentive for specific developers or projects, encouraging participation within the ecosystem.

Fee adjustment mechanism:

Fees can be adjusted dynamically based on traffic and demand on the network. This allows users to transact at a reasonable cost.

Additionally, the fee structure will continue to be improved based on user feedback and network growth.

Mempool improvements – transaction priority based processing

To maximize Sasuel Gold's transaction processing speed and efficiency, the memory pool structure has been improved from the existing simple file storage method to a Priority Queue-based processing method. This key improvement eliminates performance bottlenecks in the network and optimizes critical transactions to process faster.

Problems with the existing memory pool

1. **Simple file-based storage**
   - Mempools in many blockchain networks store transaction data as simple files and process them sequentially.
   - This method operated only in a FIFO (First-In-First-Out) manner without considering transaction priority, which delayed important transactions.
   - Transactions were lost due to synchronization issues between processes consuming and merging memory pool transaction files.

2. **performance bottleneck**
   - When transactions increased, excessive file input/output (I/O) operations occurred, slowing down the node's processing speed.
   - Even high-cost transactions could not be processed quickly during network congestion.

Improvements to priority queue-based memory pool

1 Mempool improvements – Priority Queue-based transaction processing

The memory pool's Priority Queue structure sorts transactions (Tx) according to priority (P). The priority of each transaction can be defined as follows: The memory pool's Priority Queue structure sorts transactions (Tx) according to priority (P). The priority of each transaction can be defined as follows:

Transaction priority formula

$$P(Tx) = w1 \cdot F_{gas} + w2 \cdot F_{timestamp} + w3 \cdot F_{custom}$$

F_{gas} : Transaction fee paid (Gas Fee)

$F_{timestamp}$: Weight of the transaction creation time

F_{custom}: User-defined priority (Optional)

$w1, w2, w3$: Weights of each factor (w1 + w2 + w3 = 1)

In the priority queue, transactions with high P(Tx)P(Tx)P(Tx) values are processed first. With this structure, the time complexity when sorting transactions is O(log N), where N is the number of transactions stored in the Mempool. Sasuel Gold redesigned the memory pool based on Priority Queue and implemented a processing method that considers transaction priority.

1. Apply transaction priority

- Transactions are prioritized based on fees, timestamps, and user-defined priorities.
- Transactions with high priority are processed first in the memory pool, solving the problem of essential transactions being delayed.

2. Data structure optimization

- The memory pool uses Priority Queue to keep transactions sorted. This makes transaction processing efficient with O(log N) time complexity.
- By utilizing memory-based queues instead of file-based storage, I/O operations are minimized, and processing speed is maximized.

3. Congestion management and network efficiency

- Even when the network is congested, transactions with higher fees are prioritized, allowing users to perform the transactions they need faster.
- Unnecessary transactions (low priority) are removed or placed on hold if they are not processed for a certain period of time to prevent network overload.

4. Enhanced scalability and flexibility

- The priority queue structure is highly scalable for transactions and is designed to allow for easy application of various priority criteria (e.g. network policy, user definition, etc.) in the future.

Results and Expected Effects

Through memory pool improvements, Sasuel Gold Network has achieved the following achievements:

- Increased transaction processing speed

  Important transactions are processed first, providing an environment suitable for real-time payments and large-scale transactions.
- Maximize network efficiency

  Efficiently distribute network resources even in congestion situations
- Improved user experience

  Users who pay high fees or request urgent transactions are guaranteed faster processing.

Mainpool improvements have clarified the processing priorities of transactions, eliminating network bottlenecks and making Sasuel Gold the core technology foundation that supports real-time payments and large-scale transaction processing. This allows users to enjoy a faster and more reliable payment and transaction experience.

2 RPC protocol improvement – security and speed optimization

During the development of Sasuel Gold, the existing raw HTTP request-based communication was replaced with the RPC (Remote Procedure Call) protocol. This is an important technological improvement to enhance network security and optimize communication performance.

Existing Problem – Raw HTTP-based Communication

- Overhead: HTTP requests contain additional data, such as headers and metadata, resulting in unnecessary network overhead.
- Inefficiency: Because HTTP uses text-based communication, it requires more bandwidth to transmit data.
- Security vulnerability: HTTP requests were at risk of being exposed to man-in-the-middle (MITM) attacks because data was exchanged between nodes in an unencrypted state.

RPC protocol introduction and improvements

RPC (Remote Procedure Call) is a communication protocol designed to execute function calls between remote systems or nodes as if they were local functions. Its introduction has resulted in the following improvements:

1. Minimize network overhead

- RPC eliminates the unnecessary data transfers that occur in text-based communications by serializing binary data.
- Data packet size was minimized through data serialization and deserialization, thereby reducing network bandwidth usage.

2. Optimize communication speed

- Instead of complex header processing in HTTP requests, RPC calls communicate with minimal metadata, significantly speeding up transaction transfers.
- This shortens real-time data synchronization and transaction verification time between nodes.

3. Enhanced security

- The RPC protocol supports encrypted Transport Layer Security (TLS)-based communication, allowing data to be transmitted securely.
- This fundamentally blocks security threats such as man-in-the-middle attacks, data tampering, and wiretapping.

4. Improve network efficiency

- RPC enables efficient handling of requests and responses, so it has strengths in parallel processing even when multiple nodes communicate simultaneously.
- The speed of message processing between nodes has been improved, improving the Transactions Per Second (TPS) performance of the overall network.

Results and Expected Effects

Through the introduction of the RPC protocol, Sasuel Gold Network has achieved the following achievements:

- Reduce transaction processing time
- Minimize network bandwidth usage
- Improved reliability through enhanced security
- Improved data synchronization speed between nodes

RPC improvements optimize the fundamental performance of the network while enhancing security, laying the foundation for Sasuel Gold to seamlessly support real-time payments and large-scale transaction processing. This allows users to enjoy a faster and safer trading experience.

Improved synchronization between nodes – Added Swift module

Saseul Gold introduced the Swift module, a new network service layer, to solve inter-node synchronization problems in the existing SASEUL network and maximize data transmission efficiency and stability. This module dramatically improves network scalability and performance while solving bottlenecks and packet loss that occur during data synchronization between nodes.

Problems with synchronization between existing nodes

1. excessive network traffic
    - Existing systems used simple TCP/UDP and Http communication for data synchronization between nodes.
    - Even in the process of synchronization between processes for each service in the node's local environment, packets were sent out and back, resulting in unnecessary network costs.
    - As the number of nodes increases, packet transmission becomes inefficient, resulting in synchronization delays and excessive network traffic.
2. Packet loss and inefficient retransmission
    - If some packets are lost during synchronization, the entire data must be retransmitted, increasing network overhead.
    - In particular, the longer the desynchronization between nodes, the greater the network instability.
3. Lack of scalability
    - The existing method had limitations in operating large-scale nodes because the synchronization burden increased linearly as the number of nodes increased.

Design and improvements to Swift modules

Swift modules are designed to provide efficient data synchronization strategies and layered network services to improve inter-node synchronization speed and reliability.

1. Network Layering

- The Swift module separates network communication into a transport layer and a synchronization layer.
- The transport layer compresses and divides data into chunks to prevent packet loss, and the synchronization layer efficiently synchronizes block data and state information.

2. Asynchronous streaming synchronization

- By applying an asynchronous streaming method, the synchronization process continues even if delays occur during data transfer between nodes.
- This minimizes data inconsistency issues between nodes and enables real-time synchronization.

3. Packet loss detection and retransmission optimization

- The Swift module reduces network traffic by introducing the ability to retransmit only lost packets.
- Minimize unnecessary data transmission while ensuring data consistency through an efficient ACK (Acknowledgment) mechanism.

4. Data compression and chunking

- During synchronization, data compression is performed to reduce the size of transmitted data, and the data is divided into chunks and processed simultaneously.
- This ensures efficient transfer of large amounts of data and maximizes synchronization speeds.

5. Scalability and multi-node support

- The Swift module supports the ability to distribute synchronization burden across a network's multi-node environment.
- As the number of nodes increases, network synchronization can be maintained without linear performance degradation.

Results and Expected Effects

Through the introduction of Swift modules, Saseul Gold Network has achieved the following achievements:

Improved synchronization speed:

- Real-time data synchronization between nodes enhances network-wide consistency.

Reduce network traffic:

- Reduce unnecessary network usage by optimizing data compression and retransmission of lost packets.

Maximized scalability:

- Stable synchronization is now possible even in a large-node environment.

Improved reliability:

- Asynchronous streaming ensures data remains synchronized even during network instability.

Process unification – Oracle Process implementation

To optimize Saseul Gold's network performance and maximize resource efficiency, we implemented Oracle Process, which converts the existing multi-process-based structure into a single-process-based asynchronous event system. This work is a key technical improvement that addresses inefficiencies in existing network architecture and significantly improves transaction processing speed and resource management efficiency.

Problems with existing multi-process structure

1. **Separate processes by role**
   - In the existing SASEUL network, each process was executed independently depending on the role of the node (block generation, verification, network communication, etc.).
   - We used TCP/UDP protocols for inter-process communication, which led to problems with excessive network packet generation.

2. **Waste of network resources**
   - Because each process ran individually, synchronization delays and redundant data processing resulted in excessive resource consumption.
   - Event processing speed was slowed down because the events that the node needed to process were distributed across multiple processes.

3. **Scalability and maintenance difficulties**
   - The structure that manages multiple processes and controls communication has reduced scalability as the number of nodes increases, and network maintenance and debugging have become complicated.

Design and improvements to Oracle Process

Oracle Process is a structure that integrates all roles that occur in the network (block creation, transaction verification, synchronization between nodes) into a single asynchronous event-based process. This solved network performance bottlenecks and resource waste.

1. Single process-based event processing

- Oracle Process applies an asynchronous event processing model, where one process performs all roles within a node.
- Events are processed sequentially based on an event queue, thereby preventing duplicate processing between processes and packet overload.

2. TCP/UDP communication optimization

- Existing inter-process TCP/UDP packet communication has been removed and replaced with internal event-based communication.
- Through this, network packet transmission volume was dramatically reduced and resource utilization was optimized.

3. Asynchronous event handling

- All events such as transactions, block creation, and network synchronization are processed asynchronously to maximize concurrent processing performance.
- By processing tasks in the event queue based on priority, appropriate work flow is maintained according to transaction priority and network status.

4. Resource Management Optimization

- Oracle Process efficiently manages memory and distributes CPU resources with a single process structure.
- This ensures that the node's resource usage remains consistent and provides stable performance even under high load.

5. Scalability and maintainability

- The single-process structure simplifies the codebase and makes maintaining and debugging nodes easier and more intuitive.
- It is highly scalable even in a multi-node environment, so network performance increases linearly.

Results and Expected Effects

Through the implementation of Oracle Process, Saseul Gold Network has achieved the following achievements:

Improved network performance:

- Asynchronous event queuing significantly increases concurrent processing performance and eliminates bottlenecks.

Resource efficiency:

- Memory usage and CPU load have been optimized through process unification.

Network traffic reduction:

- Packet transmission volume has been reduced by eliminating TCP/UDP communication and switching to internal event-based.

Enhanced scalability:

- Even as the number of nodes increases, network performance is maintained stably and scalability is secured.

Ease of maintenance:

- Reduce code complexity and simplify maintenance with a single process structure.

Merge resource blocks and Refine Contracts – simplify system structure

During the development of Saseul Gold, we changed the structure of Resource Block and Refine Contract, which were core concepts of the existing SASEUL network. The direction of change ultimately means merging, and these changes are not simply functional modifications, but fundamental simplification of the system structure and improvement of the consensus algorithm. By removing unnecessary complexity from existing systems, we have optimized performance, improved scalability, and improved user experience.

What are Resource Blocks and Refine Contracts?

1. Resource Block

- In the existing SASEUL consensus algorithm, block generation and verification were performed based on network resources.
- Network resources were managed by allocating the system's available resources (CPU, memory, etc.) to blocks, and resource blocks were responsible for recording and verifying the status of these resources.

2. Refine Contract

- Refine Contract is a form of contract designed for the purpose of optimizing resource blocks, and performs the function of refining and efficiently distributing network resources when creating blocks.
- This increased the complexity of the system and made maintenance difficult.

Necessity and problem solving of merge

1. Eliminate system complexity

- The existence of resource blocks and Refine Contracts has made the block creation and verification process overly complicated.
- Allocating and validating each resource incurs additional computational costs, which leads to poor network performance.
- Through merging, we simplified system logic while maximizing block creation speed and verification efficiency.

2. Improvement of consensus algorithm

- The resource block-based consensus algorithm operated centered on network resource utilization, but required several additional computation and memory resources to manage it.
- By merging resource blocks and introducing a new Proof of Historical State (POHS)-based consensus algorithm, we have made improvements to achieve simpler and faster consensus through temporal order verification.

3. Securing network scalability

- By merging resource blocks into the main block, the block creation and verification process is simplified, and the processing burden on nodes is greatly reduced.
- Accordingly, a structure was created that can be linearly expanded even as the number of nodes increases.

System structure after merging

1. Agreement process

- Due to the removal of resource blocks, unnecessary resource allocation and verification processes are eliminated when creating blocks.
- Block creation is performed through temporal order verification based on POHS (Proof of historical state), improving the performance of the entire network.

2. Improved block creation speed

- By changing the logic of resource blocks and Refine Contracts, unnecessary operations in the block creation process have been removed.
- This significantly improves transaction inclusion and block creation speeds.

3. Improved user experience and maintenance

- Simplification of system logic reduces development and maintenance costs, and improves network uptime and stability.
- Users can experience faster and more intuitive transaction processing without complex resource management concepts.

Mathematical expression – changes in system complexity

When retaining the resource block and Refine Contract, the block verification time is as follows:

$$T_{\text{verify}} = T_{\text{resource}} + T_{\text{contract}} + T_{\text{block}}$$

T_{resource}: Resource verification time

T_{contract}: Refine Contract execution time

T_{block}: Block creation time

After removing the resource block and Refine Contract:

$$T_{\text{verify(new)}} = T_{\text{block}}$$

Therefore, the total block verification time is reduced by $T_{\text{resource}} + T_{\text{contract}}$.

Node roles and structure

The Saseul Gold network adopts a dual node structure consisting of GravityNode and AnchorNode to maximize stability and efficiency. The two nodes each have clear roles andresponsibilities, and work together to perform the overall functions of the network. This provides stable blockchain operation and a strong consensus mechanism.

GravityNode plays a key role in the network and performs overall management tasks, including block generation and consensus. These nodes have additional responsibilities for maintaining network stability and performance.

Key roles:

1. **Block creation:**
   - GravityNode collects transactions and creates new blocks.
   - The generated block is delivered to AnchorNode and other GravityNodes for verification and recording.
2. **Perform consensus algorithm:**
   - GravityNode utilizes the Raft algorithm with other GravityNodes to reach consensus.
   - The Raft algorithm ensures fast and efficient consensus by electing a leader node and coordinating the final approval of blocks.
3. **Transaction and state management:**
   - GravityNode collects and manages all transactions that occur on the network and records state changes.
   - Maintains data integrity in the network by verifying the legitimacy of transactions and including them in blocks.

GravityNode is designed for high throughput and stable block generation, and is the core node responsible for consensus and block generation at the center of the network.

AnchorNode is responsible for verifying blocks created by GravityNode to ensure the reliability of the network. This node verifies and approves the GravityNode's actions based on a proof of historical state.

Key roles:

1 Block verification:

- AnchorNode verifies the validity of blocks generated by GravityNode.
- Ensures transactions within blocks comply with network rules and blocks forged or incorrect data.

2 Use historical proof:

- AnchorNode verifies block creation order and time information through history proof.
- This increases the reliability of blocks generated by GravityNode and ensures the immutability of the network.

3 Data synchronization between nodes:

- AnchorNode interacts with GravityNode and synchronizes blockchain data.
- Synchronization maintains consistency between nodes and prevents network fragmentation.

AnchorNode is a role focused on block verification and data synchronization, monitoring the activities of GravityNodes and maintaining the reliability and safety of the network.

Multi-chain support – Bypass Operator added

Saseul Gold has added Bypass Write and Bypass Read Operators to support true multi-chain. Through this, we have established a technical foundation that maximizes data exchange, interoperability, and scalability between multiple blockchains.

The existing single chain-based system had limitations in scalability because it processed all data on one chain, and collaboration or data sharing between chains was impossible. The introduction of Bypass Operator is designed to solve these limitations and smoothly manage data flow between multiple independent chains.

Existing Problems – Limitations of Single Chain System

1. **Data processing bottleneck**
   - Because all transactions and data are processed on a single chain, bottlenecks occurred in the block creation speed and transaction verification process.

2. **Data cannot be exchanged between chains**
   - The existing system did not have the ability for independent blockchains to exchange data or interoperate with each other.

3. **Lack of scalability**
   - As the number of nodes increases and the number of transactions increases, the TPS (Transactions Per Second) performance of the single chain system deteriorates and the network burden increases.

Design and role of bypass operator

Bypass Operator provides the ability to read and write data in a multi-chain environment and enables data exchange between various blockchains. This maximizes interoperability and scalability between chains. Bypass Write and Bypass Read Operator provide the Saseul Gold Network with the potential for a true multi-chain environment, enabling scalability, flexibility, and interoperability. This technology becomes an important foundation for blockchain to efficiently manage and exchange data between various chains beyond the limitations of a single chain, and is a key element in developing Saseul Gold into a future-oriented multi-chain system.

1. **Bypass Write Operator**
   - Role: Performs the function of recording data directly to another chain.
   - How it works:
     - When executing a specific transaction or smart contract, the output data can be written to an external chain through the Bypass Write Operator.
     - In this process, the data goes through encryption and signing processes, and its integrity is guaranteed through verification of the original chain.

Mathematical model:

$$\text{Write}_{\text{bypass}} = f(T_{\text{x}}, C_{\text{i}} \rightarrow C_{\text{j}})$$

T_{x} : Transaction data

C_{i} : Source chain of the data

C_{j} : Target chain where the data will be recorded

f : Verification function that ensures data integrity

2. Bypass Read Operator

- Role: Supports reading data written on an external chain from the current chain.
- How it works:
  - When executing a transaction or smart contract, state data from another chain can be read and used through the Bypass Read Operator.
  - This enables cross-chain data referencing and expands the scope and usability of operations.

Mathematical model:

$$\text{Read}_{\text{bypass}} = g(C_j \rightarrow C_i, D)$$

C_j : External chain where the data exists

C_i : Current chain that reads the data

D : Data to be retrieved

g : Function that validates and returns the data counterpart

Major improvements to Bypass Operator

1. Data can be exchanged between chains

- Bypass Write and Read enable data recording and referencing across multiple chains. This significantly improves interoperability between blockchains.

2. Maximize scalability

- Since data can be distributed and processed in a multi-chain environment, data bottlenecks have been eliminated and TPS performance has been improved.

3. Maintain independent chains

- Each chain operates independently, but interconnection and cooperation between chains is possible through the Bypass Operator.
- Security is also maintained because data integrity and verification process are guaranteed.

4. Flexible use of smart contracts

- The scope of use has expanded as smart contracts can read and write data from multiple chains.
- This makes it possible to implement more complex and advanced business logic.

Results and Expected Effects

With the addition of Bypass Operator, Saseul Gold achieved the following achievements:

1. **Multi-chain support:**
   - Maximizes blockchain scalability by implementing data exchange and interoperability between chains.
2. **Performance improvement:**
   - Network TPS has increased by eliminating data bottlenecks and distributed processing.
3. **Smart contract scalability:**
   - Data from external chains can be referenced or recorded, increasing the flexibility of business logic.
4. **Guaranteed data integrity:**
   - Integrity and reliability were maintained during the process of verifying and recording cross-chain data.

Transition to Go Language – Key Improvements

The Go language is a statically compiled language and a language optimized for system-level programming, overcoming the performance limitations of PHP and optimizing Saseul Gold's network system.

1. high execution performance

- The Go language is executed with code converted to machine language through static compilation, which dramatically improves execution speed compared to PHP's interpreter method.
- Transaction processing and inter-node synchronization are performed much faster and more reliably.

Performance Formula:

$$T_{Go} = T_{PHP} \cdot \alpha \ (\alpha < 1 \text{ Optimization Coefficient})$$

T_{PHP} : Task processing time based on PHP

T_{Go} : Task processing time after transitioning to Go

α : Performance optimization ratio
(approximately 2 to 5 times improvement)

2. High-Performance Concurrency Processing – Goroutines

- Goroutines in the Go language provide lightweight thread-like functionality and can process thousands of tasks in parallel.
- This allows nodes to simultaneously process transactions, create blocks, and synchronize the network, significantly improving the network's Transactions Per Second (TPS).

Goroutine Concurrent Processing Formula:

$$T_{\text{total}} = \frac{T_{\text{single}}}{T_{\text{goroutine}}}$$

T_{single} : Single task processing time

$C_{\text{goroutine}}$: Number of goroutines executed in parallel

D_{total} : Optimized task time through concurrent processing

3. Memory efficiency and reliable resource management

- Go prevents memory leaks by implementing an efficient garbage collector (GC).
- Maintains stable performance while minimizing memory usage when running concurrently.

4. Enhanced scalability

- The Go language is optimized for network applications and distributed system development and is advantageous for large-scale node expansion.
- It can scale linearly without performance degradation as the number of nodes increases.

Achievements and expected effects after switching to Go

1. **2 to 5 times improvement in transaction processing speed**
   - Compared to PHP, the Go language has dramatically improved TPS performance through static compilation and parallel processing capabilities.
2. **Enhanced network stability**
   - Go's efficient memory management and concurrency support minimized network node downtime and resource consumption.
3. **Massive scalability support**
   - We implemented a linearly scalable structure without performance degradation even as the number of network nodes increases.
4. **ease of maintenance**
   - The Go language's concise syntax and static typing system make development and maintenance easier.

Switching from PHP to Go language was an essential choice to maximize Sasuel Gold's network performance and stability. The static compilation characteristics of the Go language greatly improved code execution speed, and high-performance concurrency processing based on Goroutines enabled efficient parallel processing of large-scale transactions. Additionally, Go provides excellent memory management functionality for garbage collection (GC), contributing to improving the stability of the system. Based on these technical advantages, Sasuel Gold has a structure that can expand linearly even as the number of nodes increases and has achieved large-scale network scalability.
As a result, Sasuel Gold was reborn as a high-performance distributed system, providing users with a faster and more reliable blockchain environment in terms of transaction processing speed and network stability.

Transition to Native Contract

The efficiency of system execution was improved by changing the existing system contract to a native contract. Native Contract operates in a more direct and optimized form within the system, so transaction verification and execution are performed more quickly and reliably.

Transition to Native Contract – Enhancing system performance and efficiency

During the Sasuel Gold development process, the existing System Contract was converted to a Native Contract. This transition is a key improvement to maximize the efficiency of the smart contract execution environment and further enhance the performance and scalability of the blockchain network. Native Contract is a contract that runs at the system level and supports fast execution speed, low resource consumption, and direct blockchain operation. Additionally, the performance of the network was improved by precompiling the NFT/TOKEN issuance contract.

SASEUL GOLD NETWORK ARCHITECTURE

The Sasuel Gold network has designed a hierarchical architecture for efficient and stable blockchain operation. Each layer is independent and performs clear roles and functions. This document describes the hierarchical structure of the network and the main roles of each layer.

1 Oracle Service Layer

The Oracle Service layer is the highest abstraction layer of the network and is responsible for the overall operation and management of the blockchain. This layer manages transaction verification and block creation and handles peer-to-peer communication for network synchronization. It also manages the mempool to optimize the broadcasting and processing speed of transactions.

Key features include transaction broadcasting, block creation and propagation, and synchronization between Network nodes. The Oracle Service layer plays a key role in maintaining network stability and performance.

2 Machine layer

The Machine layer manages the state of the network and provides a transaction execution environment. It manages block verification, transaction execution, and state changes, and is responsible for implementing the consensus algorithm (Proof of Historical State, POHS).

The Machine layer defines and maintains the state of the blockchain network and provides a core environment to prevent inconsistencies that may occur during execution.

3 Contract layer

The Contract layer is an intermediate layer that connects the Machine layer and the Interpreter layer and is responsible for smart contract management and execution logic.

This layer handles events that occur during contract execution, including deployment, invocation, and state management of smart contracts, and performs state rollbacks. Contracts are registered and managed through ContractRegistry, and ContractExecutor handles contract execution and state changes. Additionally, events that occur during contract execution are propagated to the network through EventEmitter. The Contract layer ensures the stability of the network by implementing smart contract validation and conflict prevention logic.

4 Interpreter layer

The Interpreter layer acts as an engine that executes smart contracts. This layer processes contracts through multiple states and manages state changes and parameter verification processes.

The Interpreter layer manages the logic of smart contract execution and accurately tracks transactions and changes to the network state. This minimizes errors that may occur during smart contract execution and maintains the state stably.

5 Operator layer

The Operator layer is the layer responsible for basic operations on data and state. It handles state reading/writing and data conversion through various operators. Each operator is used to efficiently manipulate the state of the blockchain.

This layer contributes to improving the speed of data processing and transformation, and provides the network's fundamental computations reliably.

6 Storage layer

The storage layer is a layer that manages data persistence and stores and manages network block data, state data, and mempool data. The components include StatusFile, ChainStorage, and MempoolStorage, and each component plays a role in storing block data, managing state data, and efficiently processing transaction data. The storage layer plays an essential role in maintaining the reliability and safety of network data.

7 File System layer

The File System layer provides an OS-level file system interface and is responsible for physical data storage and directory structure management.

This layer safely stores network data in hardware through file reading/writing and directory management and ensures data accessibility required by the upper layer.

The File System layer helps prevent data loss and maintain data integrity.

This hierarchical structure allows the Sasuel Gold network to operate a stable blockchain while maintaining high performance and scalability. The addition of the Contract layer makes the interaction between smart contracts and network state clearer, providing improved functionality and reliability.

This hierarchical structure allows the Saseul Gold network to operate a stable blockchain while maintaining high performance and scalability. The addition of the Contract layer makes the interaction between smart contracts and network state clearer, providing improved functionality and reliability.

Saseul Gold LEGAL RIGHTS ANALYSIS

Legal Rights This material was prepared by the Foundation Asset Room of Sasuel Gold (hereinafter referred to as "Project") and is not intended to induce or recommend investment but rather to provide information that can be used as a reference for investors' investment decisions. This material is based on our reliable content and information at the time of writing but may contain errors and different information. Accordingly, we do not guarantee accuracy or completeness under any circumstances. No one, including the publisher of this material, guarantees the value or payment of these digital assets. The value of these digital assets may fluctuate significantly depending on market conditions, technological changes, regulatory trends, etc. All statements made herein may differ from our official opinions. Additionally, our company has never provided this information to a third party prior to its disclosure. Investment in digital assets may result in original losses in some cases. We do not take any responsibility for any customer investment results based on the information provided. This material cannot, under any circumstances, be used as evidence of legal responsibility for the customer's investment results. The copyright of this material belongs to the project and cannot be copied, modified, or redistributed in any form without the consent of the project.