EGO BLOCKCHAIN

A decentralized storage and 5G wireless network leveraging the AI

Artit Muhaxhiri

April 16, 2023
www.egoblockchain.com

Abstract: I know that you might question the need to introduce another layer one blockchain and add another cryptocurrency to a market already crowded with 22,932 options. However, it's important to understand that each blockchain has unique features and capabilities that can address specific challenges and demands, and the introduction of new blockchains and cryptocurrencies can pave the way for innovation and new possibilities in the industry. Our proposed solution aims to address two major problems in the industry: data storage and cryptocurrencies.

The current centralized data storage systems are vulnerable to hacking and data breaches, which can compromise sensitive information. On the other hand, Bitcoin has faced criticism for its energy-intensive Proof of Work mechanisms and the potential centralization of infrastructure. Proof of Work (PoW) mechanisms used in Bitcoin result in the energy waste involved in validating proofs, which presents no other utility. With PoW, miners benefit from economies of scale, which can lead to infrastructure centralization and work against the goal of decentralization. With Proof of Stake (PoS), centralization of infrastructure can occur out of convenience, as stakers choose to run their node on centralized cloud infrastructure, weakening the network's resilience to attacks.

To address these issues, we are creating a decentralized storage system that utilizes blockchain and EGO 5G devices to ensure data security and privacy. Micro servers will be spread worldwide to allow individuals and companies to store their data in an encrypted manner. We also utilizes AI as a supervisor to validate the nodes and ensure the integrity of the system.

Our proposed solution addresses the criticisms of blockchains and cloud services while offering improved security, a more robust ecosystem, reduced costs, passive income for storage providers, and privacy features.

1. Introduction

Many people think that Web3 is a completely new technology that operates independently of Web1 and Web2. The truth is that Web3 is built on top of the existing infrastructure provided by Web1 and Web2. This means that even if the dApp's functionality is decentralized, the frontend or user interface can still be removed or deleted, which would make the dApp inaccessible to users.

Decentralized networks have been gaining momentum in recent years as a way to provide more security, privacy, and control over data. However, the transition from Web2 to Web3 has been slow due to the lack of sufficient decentralized infrastructure, especially outside the realm of DeFi. The setup of decentralized nodes is often complicated, time-consuming, and costly, which can discourage developers and end-users from adopting decentralized applications.

We are developing a new blockchain with a main focus on decentralized storage and smart contract functionality to solve these issues. The blockchain will utilize a proof-of-spacetime and proof-of-coverage consensus mechanism to ensure security and reliability.

With new 5G micro servers to provide fast and reliable infrastructure for decentralized applications. We will enable users to interact with dApps seamlessly, without experiencing lag or delays. Furthermore, we will work in conjunction with IPFS, a protocol for distributed storage and sharing of files. This will enable developers to create decentralized applications that rely on decentralized storage, ensuring data redundancy and increasing fault tolerance. By leveraging decentralized storage, smart contracts, and fast network infrastructure, Ego Blockchain aims to solve the issue of removing the frontend of a dApp. This will help to create a more robust and scalable decentralized network that can encourage the development and adoption of decentralized applications beyond DeFi. With the elimination of infrastructure costs for developers and end-users, we hope to encourage the creation of more open-source dApps, leading to a diverse set of applications and mainstream adoption of the decentralized web. The potential benefits of decentralized networks are clear, including increased security, privacy, and control over data. We aim to provide a viable solution to the challenges that have slowed the adoption of Web3, making decentralized applications more accessible and user-friendly.

2. Decentralized storage and 5G wireless network

The Decentralized Storage Network using micro 5G servers is a distributed system that allows users to store, access and retrieve data in a decentralized and secure manner. The network enable efficient and reliable data storage and retrieval, utilizing micro 5G servers as nodes in the network.
The network will be modelled using the following equations:
$D = \{d1, d2, d3, ..., dn\}$ - set of n micro 5g servers
$P = \{p1, p2, p3, ..., pm\}$ - set of m data shards
$C = \{c1, c2, c3, ..., ck\}$ - set of k clients
In the [**Figure 1]** each data shard is encrypted and stored across multiple micro 5G servers, ensuring data redundancy and increasing fault tolerance. The number of servers that each shard is stored on can be determined by the replication factor r.
We utilize a sharding algorithm to partition the data into smaller, more manageable shards. Each shard is assigned a unique identifier, which is used to retrieve the shard from the network. The sharding algorithm will be represented using the following equation:

$$shard\_id = hash(data) \% m$$

Where hash(data) is the hash value of the data being stored, and m is the total number of shards.
Clients can access the data by sending a request to the network. The request is forwarded to the micro 5G servers storing the required data shards, which then send the requested data back to the client.
The network will be modelled using the following equation:

$$request = \{shard\_id1, shard\_id2, ..., shard\_idn\}$$

Where shard_id1 is the unique identifier of the ith shard required to fulfil the client's request. We use consensus mechanism, such as *proof-of-spacetime*, to ensure the security and reliability of the data storage and retrieval process.
The equation is: $L = H(x) \bmod N$ where L is the location where the data is stored, x is the unique identifier of the data, $H()$ is a hash function, and N is the total number of servers in the network.
Using micro 5G servers as nodes, and employing encryption/decryption, sharding, and consensus mechanisms we ensure efficient and reliable data storage and retrieval.

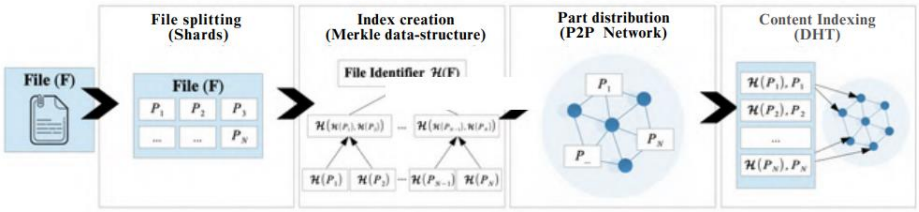


Figure 1. *Block scheme of the decentralized storage protocol for splitting a file F into N pieces Pn stored along with the hash H(Pn) in the peers.*

3. Quantum resistant key pairs

Quantum-resistant algorithms are a new generation of cryptographic algorithms designed to be secure against attacks by quantum computers. Quantum computers have the potential to break many of the traditional cryptographic algorithms that are currently in use, such as RSA and Elliptic Curve Cryptography (ECC), by using Shor's algorithm to factor large numbers and solve the discrete logarithm problem.

To address this threat, new quantum-resistant algorithms have been developed that rely on different mathematical problems that are believed to be hard even for quantum computers. One such algorithm is *sikep751*, which is based on isogenies of elliptic curves.

sikep751 is a promising candidate for post-quantum cryptography because it is believed to be secure even against attacks by quantum computers. The security of *sikep751* is based on the difficulty of finding the isogenies ϕ and ϕ' given only the public key. This problem is believed to be hard even for quantum computers, which makes *sikep751* a promising candidate for post-quantum cryptography.

Moreover, *sikep751* offers several advantages over other post-quantum cryptographic algorithms. It has relatively small key sizes and low computational requirements, making it practical for use in a wide range of applications, including IoT devices and blockchain.

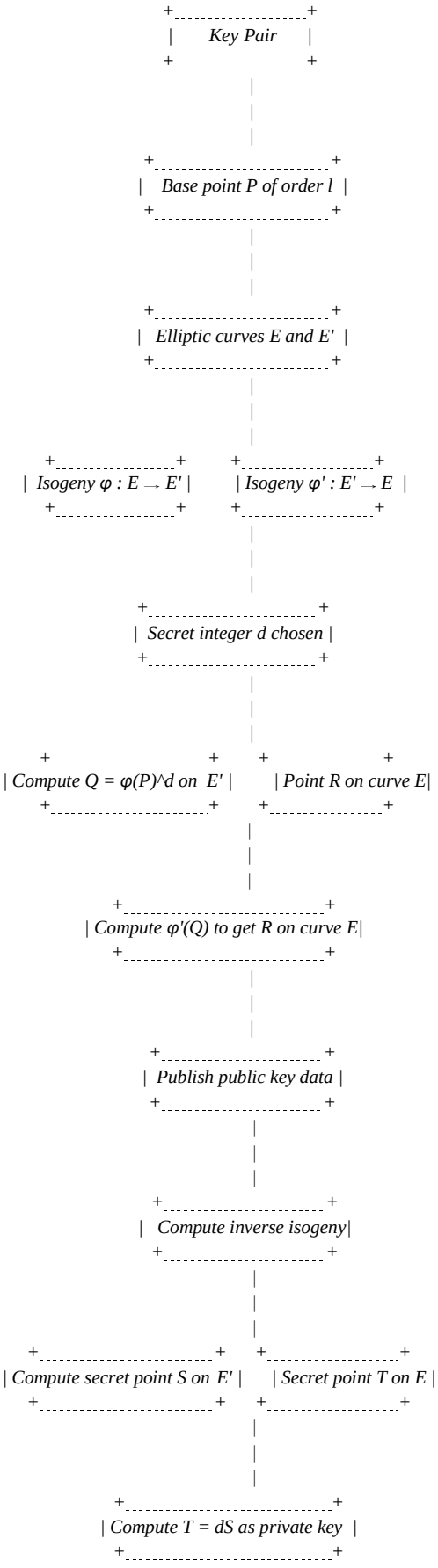
Furthermore, *sikep751* has been extensively studied and evaluated by the cryptographic community, including the National Institute of Standards and Technology (NIST), which has selected *sikep751* as one of the finalists for its post-quantum cryptography standardization process.

To generate a key pair in *sikep751*, we start by selecting a base point P of order l, where l is a prime number. We also choose two elliptic curves E and E', where E has a large number of points and E' has a small number of points. We then choose two isogenies ϕ and ϕ', where ϕ maps E to E' and ϕ' maps E' back to E.

To generate the public key, we start with a secret integer d and compute the point Q = ϕ(P)^d on the curve E'. We then compute the isogeny ϕ'(Q) to obtain a point R on the curve E. Finally, we publish the base point P, the curves E and E', and the point R as the public key.

To generate the private key, we start with the integer d and compute the inverse isogeny ϕ'^(-1) from E to E'. We then compute the point S = ϕ'^(-1)(R) and multiply it by d to obtain the secret point T = dS. Finally, we keep T as the private key.

The security of *sikep751* is based on the difficulty of finding the isogenies ϕ and ϕ' given only the public key.


Key Pair
Base point P of order l
Elliptic curves E and E'
Isogeny φ : E → E'
Isogeny φ' : E' → E
Secret integer d chosen
Compute Q = φ(P)^d on E'
Point R on curve E
Compute φ'(Q) to get R on curve E
Publish public key data
Compute inverse isogeny
Compute secret point S on E'
Secret point T on E
Compute T = dS as private key

4. Encryption of Public Key

The encryption of a public key will be using *chacha20* and *nonce* for a secure and efficient way to protect sensitive information in our decentralized network. *ChaCha20* is a symmetric encryption algorithm that generates a stream of pseudo-random bits based on a key and a nonce. *Nonce* are random values that are used once to prevent the same key from generating the same output twice. To encrypt the public key, we first generate a random nonce and use it with the *ChaCha20* algorithm to encrypt the key. The resulting encrypted public key can then be signed with the private key to provide an additional layer of security and authenticity.

When transferring or receiving data or funds, a new encrypted public key is generated for every transaction. This ensures that even if one key is compromised, the attacker will not be able to access other transactions or sensitive information.

In addition to encrypt public keys, the *ChaCha20* algorithm will also be used for encryption and decryption of data. Data is encrypted with a randomly generated nonce and the shared secret key between the sender and the receiver. The encrypted data is signed with the private key of the sender to ensure that it has not been tampered with during transmission.

The use of *ChaCha20* and *nonce* provides a strong level of encryption and security in our decentralized networks, making it an ideal choice for protecting sensitive information and transactions.

Here are examples of a key pair with *sikep751* and an encrypted public key using *chacha20* and *nonce*:

Public key:

fc673855a06964291d779ae29dd3f77289366296879a4b75610ab7e5c6cb2a8225df1e73f6c2db78af23c6d7e4df1217343bb2ddb8f16 86cc561afb98c8e8e0154af019cc9815a4466c261d42d309c653fe20fd077d4d0495b1110cfe0364eece1059723d25f8572bb9326684dd f8059e70e7613fa909de96123a5440df6b275311dcfb0d914eee90ed093773e5f7749bb3019f93d7009c39b413f0a15e6ff934fbf552247 80a932464a49f01c5e01033ed8ccc27e75db0722457226c1ba84687ed6c7589ef159ffe482dd05316a6239bd08df064cc812ba3a9f713f ce4904cbb3875fb00018c61b448f432f9bde625fa6470808650d314fb39d3e20002fbd75b49089f7bafe7955f5dba430834b72b67bb885 9d0ae377c8a001a1fa6cccbc9115ea3ec8828442c9c81710a4aa6cfd7c60aab90a61ec9ab44ebe81b9fc44e5d57465c62d70cf0c8a0ffa 8684a20ded0bc7f275454d7fa4e3a56fe41751c642fea4ca63886f673ed76510139b07b208a4fc673855a06964291d779ae29dd3f7728 9366296879a4b75610ab7e5c6cb2a8225df1e73f6c2db78af23c6d7e4df1217343bb2ddb8f1686cc561afb98c8e8e0154af019cc9815a4 466c261d42d309c653fe20fd077d4d0495b1110cfe0364eece1059723d25f8572bb9326684ddf8059e70e7613fa909de96123a5440df6b 275311dcfb0d914eee90ed093773e5f7749bb3019f93d7009c39b413f0a15e6ff934fbf55224780a932464a49f01c5e01033ed8ccc27e75 db0722457226c1ba84687ed6c758

Secret key:

20273f8c199284cb4b5d133e27e782140e2966089717d362966c9e0102a6e83384acdfcfe80347c3ee1a132bee3a7b172da6fb958747c df9d2485e33cc44d7ab62c176c761b6665bb4d91e37becbc20199b54077b76bc0c8f0ba0295009003ae48ef487af252218379b284a572 a7701998770d0ef8cd46375773af4622e4115d544cf2598a2629892fe817c48925ce4785dd228e309ca23a05c6ca7ec8b007c7a877474 713d618227f569c72530e82282580d2b76407eb1b5eaa9165d706db7bcfa9d3fce369e0edb74cd780f39c885883f0386354de38b92969 e90be2ec534981f960925fc7be8929211584d98dbb0130c02288ff51e76bda752f5db3a5fde7c0e71311b6560501c989c60a0730c7cd52 1304f99ef159ffe482dd05316a6239bd08df064cc812ba3a9f713fce4904cbb3875fb00018c61b448f432f9bde625fa6470808650d314fb 39d3e20002fbd75b49089f7bafe7955f5dba430834b72b67bb8859d0ae377c8a001a1fa6cccbc9115ea3ec8828442c9c81710a4aa6cfd 7c60aab90a61ec9ab44ebe81b9fc44e5d57465c62d70cf0c8a0ffa8684a20ded0bc7f275454d7fa4e3a56fe41751c642fea4ca

Encrypted public key:

702f84bec401000000000000000000000802f84bec401000000000000000000000

To encrypt and decrypt a message we use a *ChaCha20* algorithm and *nonce*. Here are the steps involved:

- The message "This is a secret message" is converted into binary format.
- The *ChaCha20* encryption algorithm is applied to the binary message using the encrypted public key and the nonce.

This produces the encrypted message: *a22c43d12bb38bdf7b526eabc34b4860c8dc9de594*

- A nonce is generated: *1ae10b594f09e26a7e902ecbd0600691*
- The message is hashed using *Blake2s*, which produces a fixed-length output called a digest.
- The digest is signed with the private key to produce a signature.
- The signature is sent along with the encrypted message.
- Upon receiving the encrypted message and signature, the receiver generates the same digest by hashing the message with *Blake2s*.

‣The received signature is verified using the encrypted public key of the sender. If the signature is valid, it means the message has not been tampered with during transmission.
‣"To decrypt the message, the decryption key, which is the same as the encrypted public key (is generated anew for every transaction), along with the same nonce, are used. The ChaCha20 decryption algorithm is then applied to the encrypted message using the key and the nonce, which produces the original message in binary format."
‣The binary message is converted back into text format, which produces the decrypted message: "*This is a secret message*".

Note: This whitepaper represents a continuous work in progress. We will endeavor to keep this document current with the latest development progress. As a result of the ongoing and iterative nature of our development process, the resulting code and implementation is likely to differ from what is represented in this paper. We invite the interested reader to peruse our GitHub repo at *https://github.com/egoblockchain* as we continue to opensource various components of the system over time.

5. Ego Protocol

- **Transport Layer**

Libp2p will be used as the transport layer protocol for *5G-CP*, providing a secure and efficient way to transfer data between nodes in our 5G network. In addition, *libp2p's* peer discovery and routing modules will improve the performance of the 5G network by enabling nodes to easily find and connect with other nodes, and to efficiently route data between nodes. With 5G-CP, data will be securely and efficiently transferred between the nodes ensuring redundant storage, prevention of deletion or tampering, and accessibility.

- **Consensus Layer**

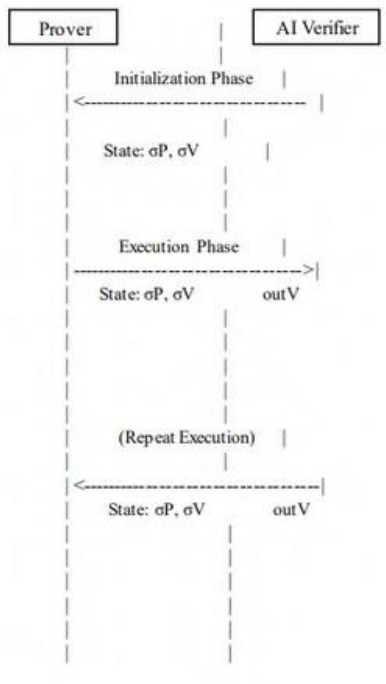
Proof-of-Coverage (PoC) is important to ensure that a particular node or device has a certain level of coverage within a network. The PoC mechanism in *5G-CP* uses 5G waves to validate that Hotspots are providing legitimate wireless coverage. The algorithm uses signal strength and signal-to-noise ratio (SNR) to determine the coverage area of a particular node or device. The formula for SNR is:
SNR = (Signal Power) / (Noise Power)
where Signal Power is the power of the signal transmitted by the device and Noise Power is the power of the noise in the system. The SNR value is an indication of the quality of the wireless signal received by the node or device.
To ensure that there are enough devices in a network to provide adequate coverage, the PoC mechanism requires devices to periodically transmit a signal indicating their presence and coverage area. The PoC mechanism will be enhanced with the use of AI, which use machine learning algorithms to analyze the coverage data collected by the witness nodes and validate the legitimacy of the coverage provided by the transmitting node. The AI will also detect and prevent any attempts to fake or manipulate coverage data, ensuring the integrity of the network.

Let S be the set of witness nodes in the network, and let N be the set of all nodes in the network. For each node n in N, let C(n) be the coverage area of node n as determined by the PoC mechanism. Let W(s, n) be a binary variable indicating whether witness node s has received a signal from node n.
Then, the AI will use the following formula to determine the legitimacy of node n's coverage area:

$$Legitimacy(n) = \left(\frac{1}{|S|}\right) * \sum(s \in S)\ W(s, n) * f(C(n), C(s))$$

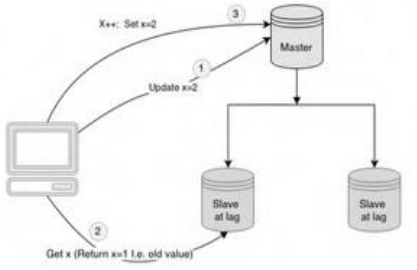
where f is a function that compares the coverage areas of node n and witness node s and returns a value between 0 and 1 indicating the similarity of the two coverage areas. This function will be based on various metrics such as distance, signal strength, SNR, etc.
The formula calculates the legitimacy of node n's coverage area by taking the average of the binary variables W(s, n) across all witness nodes s in the network. The binary variables indicate whether each witness node has received a signal from node n, and the average represents the proportion of witness nodes that have detected node n's signal.
The legitimacy value is then weighted by the similarity between node n's coverage area and each witness node's coverage area, as determined by the f function. This ensures that nodes with similar coverage areas have a greater influence on the legitimacy calculation than nodes with dissimilar coverage areas.

Proof of Spacetime combined with PoC it validates nodes based on the time, space and coverage they dedicate to the network. Each node in the network is required to dedicate a certain amount of storage space and computing power to the network, which is then used to validate transactions and create new blocks.
The AI observes the nodes and monitors their performance, ensuring that they are operating as expected and dedicating the required resources to the network.
Nodes are expected to submit proof of their spacetime resources on a regular basis to the network. The AI supervisor then validates this proof and provides feedback to the network on the legitimacy of the proof submitted by each node. This feedback is used to adjust the reputation of each node, with nodes that submit valid proof being rewarded with a higher reputation and nodes that submit invalid proof being penalized.
By using the Proof of Spacetime algorithm with AI as a supervisor, we ensure that nodes are operating as expected and dedicating the required time and space to the network. This helps to maintain the security and reliability of the blockchain, while also ensuring that the nodes in the network are operating at an optimal level.



The Prover is a EGO 5G device that participates in the network and performs storage operations. The AI Verifier is responsible for validating the Prover's performance.
During the Initialization Phase, the Prover and AI Verifier establish a shared state, denoted as σP and σV, respectively.
In the Execution Phase, the Prover performs storage operations and generates an output, denoted as outV. The AI Verifier receives the output and updates its state based on the output and its current state. The updated state is denoted as σV.
The Execution Phase is repeated multiple times to ensure that the Prover's performance is consistent and reliable. The Prover generates an output each time, which is validated by the AI Verifier.

Proof of Replication (PoRep) is a consensus mechanism used to certify that a node has replicated a piece of data correctly. We utilize PoRep to ensure that each node in the network has an exact copy of the same data. We use a master node and two slave nodes to replicate the data, and each node must prove that it has stored the data correctly by providing a proof that it has replicated the data accurately. This ensures the security and integrity of the data on the network and prevents any malicious actors from tampering with it. By using PoRep, we can guarantee that the data stored on our network is accurate, secure, and accessible to all nodes in the network.



Proof-of-Burn: is used to control the number of files that can be uploaded to the EGO network, and to ensure that the uploaded files are unique. In order to upload a file, the user must first burn a certain amount of EGOC tokens as proof of their commitment to the network. The amount of tokens required for each upload can be adjusted based on network demand and other factors. By requiring users to burn tokens in order to upload files, the network ensures that there is a cost associated with each upload, which helps to prevent spam and abuse.

Proof-of-Serialization it checks if a specific set of transactions in the network have been processed in a specific order. By using Proof-of-Serialization with AI supervision, we can secure that all nodes have a consistent view of the order of transactions in the blockchain and prevent double-spending.

Proof-of-AI is a method of proving that nodes are actively running an AI algorithm as their supervisor. PoAI will be applied in the context of machine learning (ML) as a way to prove that ML algorithm is being used to improve constantly the AI services. To provide a more concrete example, the formula for PoAI as supervisor it will be like this:
Let's define the following variables first:

- N: the total number of nodes in the EGO network
- T: the total time (in seconds) that the EGO network has been active
- D: the total amount of data (in bytes) that the EGO network has processed
- S: the total amount of storage space (in bytes) available in the EGO network The

PoAI formula as supervisor can then be defined as follows:
PoAI = (P1 + P2 + P3) / 3
where:

- *P1 = (D / NT) * (1 - e^(-λ1T))* - the formula to calculate the average amount of data processed by each node per second, taking into account the total time that the node has been active in the network, with the parameter λ1 determining how quickly the node's performance decays over time.
- *P2 = (S / N) * (1 - e^(-λ2*T))* - this calculate the average amount of storage space available per node in the network, using the total time the network has been active, and the decay constant λ2, which decides how quickly the available storage space per node decays over time.
- *P3 = (C / NT) * (1 - e^(-λ3T))* - the parameter P3 represents the average amount of coverage

each node provides per second, while considering the total active time of the network. The decay constant λ3 is responsible for determining the rate at which the coverage performance of a node decreases over time. C represents the proportion of the total data processed by the network that is currently covered by the nodes. The PoAI takes into account the performance of each node based on the amount of data processed, the storage space available, and the coverage provided. It also takes into account the total time the network has been active and the decay rates of each performance metric.

- **Application Layer**

Application layer will leverage AI technology to create smart contracts in response to natural language inputs provided by users. At present, we are in the research phase, and we have yet to determine whether we will develop our own AI model, which will be trained on a vast dataset of sample smart contracts and programming languages, or utilize third-party AI providers that can comprehend user intent and generate the appropriate smart contract code.
The smart contract that are generated, will be executed by the EGO virtual machine, which will ensure that the code runs correctly on the blockchain. The virtual machine will be designed to be more user-friendly and accessible, with features such as a simpler development environment, a more intuitive programming language, and built-in debugging tools. This will enable developers and users to create and interact with smart contracts more easily and efficiently.

6. Tokenomics and Staking Rewards

The Tokenomics of the EGO blockchain has been carefully crafted to ensure a fair and sustainable ecosystem. One of the standout features of our platform is that there are no transaction fees or fees for deploying smart contracts. Instead, nodes are rewarded with newly created coins from new blocks.
To incentivize nodes to hold onto their coins, a unique system has been implemented where nodes cannot sell their newly created coins for a period of one month. In exchange for this waiting period, nodes receive a ***20%*** interest rate as a staking reward. This creates a strong incentive for nodes to hold onto their coins and contribute to the stability of the platform.
For nodes, staking bonuses are offered with an annual percentage rate (APR) starting from ***40%***. This rate increases with longer and larger staking periods, creating a strong incentive for nodes to participate in the ecosystem's growth. Similarly, other stakers receive an initial APR of ***20%***, which also increases with longer and larger staking periods.
To sustain the staking yields and ensure the platform's long-term growth, we have implemented a ***5%*** tax penalty for both buyers and sellers. Of this tax**,** ***3%*** goes towards DEX liquidation, and the remaining ***2%*** is allocated as staking rewards. This tax model ensures that the platform can provide significant staking yields, and it is expected to drive a positive trend in the platform's pricing.
The EGOCOIN token has a maximum supply of **100,000,000** coins, which are distributed among various stakeholders in a fair and balanced manner. The team is allocated **10,000,000** coins, the **40,000,000** coins allocated for miners will be gradually released over time through the creation of new blocks, ensuring a fair distribution and incentivizing participation in the network's security. Another **20,000,000** coins are reserved for the development of decentralized (DEX) and centralized (CEX) liquidations, and **20,000,000** coins are reserved for investors. The remaining **10,000,000** coins are allocated for marketing and promotional campaigns.
The EGOCOIN token is currently based on the ERC-20 standard, but we plan to migrate it to our own blockchain once it is fully developed. This allocation includes coins reserved for the development of decentralized and centralized liquidations, marketing and promotional campaigns, as well as investors, and the development team. Our tokenomics framework is designed to incentivize all stakeholders to participate in the ecosystem's growth and success while providing a fair and transparent structure. The staking bonuses for nodes and stakers, as well as the tax penalty, help sustain our staking yields and ensure the platform's long-term viability.

7. Block Details and Transaction

Our block design includes several key features that enhance the performance and reliability of the platform. Each block contains a timestamp, a hash of the previous block, and a hash function using *blake2s*. The transactions are stored in a Merkle tree structure within the block. To ensure the integrity of each block and prevent fraudulent activity, we will use a machine learning-based signature that serves as the supervisor. This will allow us to automatically detect and reject any invalid blocks, as well as identify potential security threats and take action to prevent them. Our blockchain will have a new block every 10 seconds, ensuring fast and reliable processing of transactions. Each block will have a maximum size of 1GB, allowing for large amounts of data to be stored securely. With a transactions per second (TPS) rate of approximately 100,000 without fees, our blockchain is designed to handle high volumes of transactions with ease. This high TPS rate, combined with our fast block creation time and efficient block structure, will enable our platform to support a wide range of applications and use cases with minimal latency and high throughput. To achieve high transaction throughput and fast block times, we use the combination of Proof of Spacetime (PoSt) and Proof of Coverage (PoC) consensus algorithms. PoSt ensures that miners provide valid proofs of storage, while PoC ensures that miners are evenly distributed across the network.

The following lines explains how our system works:

- The AI algorithm processes transactions and creates a block candidate.
- The candidate block is broadcast to the network, and nodes use their stored data to solve the PoSt challenge.
- Miners that solve the challenge broadcast their proof to the network.
- The AI algorithm then selects the best performing nodes based on their score to store the new block.
- The selected nodes store the new block, and the process repeats.

By using this approach, we can achieve a new block time of 10 seconds and a transaction throughput of over 100,000 transactions per second.

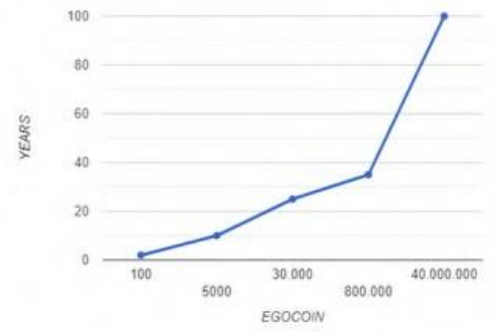
Here's also an example of how a transaction is processed and added to a block:

- Alice initiates a transaction to send 10 EGOCOIN to Bob.
- The AI algorithm receives the transaction, verifies that Alice has sufficient funds to complete the transaction and validate the transaction
- The transaction is then broadcast to the network.
- The AI algorithm selects the best performing nodes based on their score to store the transaction.
- The selected nodes store the transaction in the candidate block.
- Once enough transactions are collected, the AI algorithm creates a new block candidate and broadcasts it to the network.
- Miners use their stored data to solve the PoSt challenge, and the best performing nodes are selected to store the new block.
- The selected nodes store the new block, which includes Alice's transaction, and the process repeats.

By using this approach, we can ensure fast and secure transaction processing with a high degree of scalability and efficiency.

8. Block Reward Calculation

The Block Reward will be calculated by taking the block reward as a function of the total number of coins to be revealed, the total number of blocks produced, the number of years since the start of the network, the decay constant, and the number of transactions in the current block. The block reward decreases 50% every 2 years due to the decay constant and is also reduced based on the number of transactions in the block.



This incentivizes nodes to include as many transactions as possible in their blocks to earn higher rewards. By including the transactions per block (T/B) and the decay constant λ, we also determine how quickly the performance of a node decays over time. For the calculation we use this method, let:

- R = current block reward
- T = number of transactions in the block
- B = maximum number of transactions per block (assumed to be a constant value)
- D = total number of coins to be revealed (40,000,000 EGOC)
- Y = number of years since the start of the network
- N = total number of blocks produced in Y years (assuming 315360000 blocks per year based on our 10-second block times)

Then, the formula for calculating the block reward can be:
Which in conclusion the block reward is calculated:

$$R=(\frac{D}{N})\times(1-e^{\wedge}(-\lambda Y))\times(1-(\frac{T}{B})$$

Where:

- λ is the decay constant that determines how quickly the block reward decreases over time
- e is the mathematical constant approximately equal to 2.71828

In the early days of the network, when the number of transactions per block is low, the T/B value would be small, which means that the second part of the formula (1 - (T/B)) would be close to 1, resulting in a higher block reward.
As the number of transactions per block increases over time, the T/B value would increase, causing the second part of the formula to decrease, resulting in a lower block reward. This would incentivize nodes to continue providing their services to the network, as they would be rewarded more for their work in the early days when the network is still growing.

9. Nodes

We have only one type of node/miner that is through EGO 5G devices. When a person buys an EGO 5G device, they add the serial number to the EGO app on their phone or PC, and the device will automatically operate as a node.
To join the network, simply needs to put the sim card in the device, to connect to the 4G/5G internet and download the EGO app. Once the device is connected, it can automatically join the network and start storing data.
We know that we have to deal with different malicious of nodes which could exhibit a variety of characteristics, such as attempting to broadcast fraudulent transactions or attempting to take control of the network through a 51% attack. In the Ego blockchain, we will handle malicious nodes through a combination of consensus mechanisms such as proof of spacetime, proof of coverage, and proof of AI, as well as by implementing measures such as IP address banning and node reputation systems. To solve the 51% attack, we use a reward system that decreases by 10% when there are two or more devices in one square meter. This discourages users from clustering their devices together and gives more incentive for users to spread out their devices across a wider area, making it more difficult for any one entity to control 51% of the network.

As for Sybil attacks that act of creating multiple identities or nodes to gain control of a network or system, we have implemented several mechanisms to handle it, such as:

‣Proof of Spacetime: This mechanism ensures that each node in the network is physically located in a unique place. Each EGO 5G device has a unique GPS location, and the network uses this information to ensure that there are only two nodes per square meter.

‣Proof of Coverage: makes sure that each node is distributed evenly across the network. Every EGO 5G device provides coverage to a specific area, and the network uses this information to ensure that there is no overlapping coverage.

‣Limiting the number of nodes: The only nodes in our network are the EGO 5G devices. This ensures that there is no room for malicious nodes to join the network.

‣Monitoring network behaviour: The AI constantly monitor the network for any suspicious activity or behaviour that may indicate a Sybil attack. If AI detect any such activity, will take appropriate action to prevent it from causing any harm to the network.

In addition, we use Erlang programming language to develop our blockchain platform, which is known for its high fault tolerance and ability to handle large-scale distributed systems. This allows our network to withstand potential attacks and maintain its integrity even in the face of malicious actors.

By implementing these mechanisms, we are confident that EGO blockchain is secure against several attacks.

10. DAO

Decentralized Autonomous Organization, is a concep that allows for decentralized decision making and control. In Ego Blockchain, we will use a DAO to split the profits from selling the Ego 5G devices. This ensures that the decision-making power is distributed among the members of the community, and no single entity or individual has undue control over the network. We will operate truly decentralized by solving problems through DAO votings. The community members will be able to vote on important decisions related to the development and growth of the network, such as changes to the protocol, the allocation of resources, and the development of new features. In addition, we will use smart contracts to choose the tender winner of doing a task. This ensures transparency and fairness in the selection process, as the winner will be determined based on pre-defined criteria encoded in the smart contract. Every profit and sale from the hardware will be transparently recorded and made available on our website. This ensures that the community members are aware of the financial activities of the network, and are aware how the profits are distributed. In order to ensure fair and transparent decision-making within our ecosystem, we will implement two types of voting and the results it will be averaged to determine the final outcome. The first type of voting requires staking, which means that participants must hold a certain amount of EGOCOIN in order to vote on proposals. The second type of voting is based on passing a knowledge test about the particular subject being voted on. This allows participants to prove their expertise in the area and ensures that only those with a good understanding of the topic are able to vote on it. This will help to prevent uninformed decisions that could negatively impact our ecosystem.

Once both types of voting have taken place, we will calculate an average between the two results to determine the final outcome of the vote. This provides that both stakers and knowledgeable participants have an equal say in the decision-making process, and helps to assure that decisions are made in the best interest of our ecosystem.

Here's an example of how the voting process will work with the two different DAO votings:

‣Staking voting: In this type of voting, individuals would stake a certain amount of EGOCOINS to participate in the decision-making process.

The amount staked would be proportional to the voting power of each individual, meaning that those who stake more would have a greater say in the outcome of the vote. The formula for staking-based voting will be:

$$\textit{Total Voting Power} = \Sigma\ (\textit{Individual Stake/Total Staked})$$

For example, if there were 100 people participating in the vote and the total amount staked was 10,000 tokens, and an individual staked 500 tokens, their voting power would be:
(500/10,000) x 100 = 5% of the total voting power

‣Knowledge-based voting: In this type of voting, individuals would need to pass a test to demonstrate their understanding of the particular subject that is being voted on. Once they pass the test, they would be granted a certain amount of voting power. The formula for knowledge-based voting will be:

Voting Power = (Test Score/Max Test Score) x Base Voting Power

For example, if the base voting power was 10 and an individual scored 80% on the knowledge test with a maximum possible score of 100, their voting power would be:
(80/100) x 10 = 8

‣Combining the results: To get the final outcome of the vote, we will take an average of the results from both the staking and knowledge-based voting systems.
For example, if the staking-based voting gave a total voting power of 60% and the knowledge-based voting gave a total voting power of 40%, the final voting power will be:

Final Voting Power = (Staking Voting Power + Knowledge Voting Power)/2
Final Voting Power = (60 + 40)/2 = 50

In this example, the final voting power is 50%, meaning that the decision would be made based on the majority vote.

11. Conclusion

Our goal is to create a truly decentralized ecosystem that provides secure and reliable services to users. One of our primary objectives is to provide decentralized cloud storage that is both secure and cost-effective. With our blockchain technology, users will be able to store and access their data without worrying about security breaches or data loss. We will also provide a decentralized website that is secure, fast, and reliable.
Another key feature of our ecosystem is offline transactions and offline data storage. We will create a system that allows users to make transactions and store data even when they are offline. This is particularly useful in areas with poor internet connectivity or during natural disasters when internet access may be limited.
We are committed to provide a fee-less environment for our users. We will not charge any transaction fees for sending or receiving egocoins. Additionally, there are also no fees associated with deploying a smart contract on our platform.
Our staking rates will start at 20% APY, which is higher than many other staking platforms. This will incentivize users to hold our tokens and stake them on our platform.
Our blockchain infrastructure is designed to be easy to use for both developers and non-developers. We believe that blockchain technology should be accessible to everyone, and we have designed our ecosystem with this principle in mind.
The EGO 5G device will be created to allow for 5G internet sharing, enabling users to earn passive income by sharing their excess bandwidth with others. This will create a truly decentralized network of wireless networks, providing a more reliable and cost-effective alternative to traditional internet service providers.
With our AI technology, everyone will be able to write smart contracts and create dApps without needing to have any coding experience. This will democratize the blockchain space and allow for more people to participate in the development of decentralized applications.
Finally, we believe that anyone should be able to generate passive income with EGO 5G devices, which store data and provide wireless networks. By leveraging our blockchain technology, we aim to provide a more decentralized and equitable distribution of wealth.

References

[1] Efficient Algorithms for Supersingular Isogeny Diffie-Hellman https://www.researchgate.net/publication/301749413_Efficient_Algorithms_for_Supersingular_Isogeny_Diffie-Hellman pages 16-77 August 2016.

[2] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

[3] Juan Benet. IPFS - Content Addressed, Versioned, P2P File System. 2014.

[4] Protocol Labs. Technical Report: Proof-of-Replication. 2017.

[5] The Qogecoin Authors. qogecoin, 2021. https://github.com/qogecoin/ qogecoin, accessed May 10, 2022.

[6] Vitalik Buterin. Ethereum, 2014

[7] Tal Moran, Ilan Orlov, " Simple Proofs of Space-Time and Rational Proofs of Storage " pages 4-23.

[8]P. Labs. Filecoin: A decentralized storage network, 2017. URL: https://filecoin.io/filecoin.pdf.

[9] Helium. Proof of Coverage. https://docs.helium.com/blockchain/proof-of-coverage/ 2013

[10] Jonathan Rodriguez. Fundamentals of 5G Mobile Networks "https://pce-fet.com/common/library/books/50/6998_[Jonathan_Rodriguez]_Fundamentals_of_5G_Mobile_Net(b-ok.org).pdf" 2015.

[11] Phillip Rogaway. "Nonce-Based Symmetric Encryption". https://www.cs.ucdavis.edu/~rogaway/papers/nonce.pdf pages 4-11.