Shardeum

A Sharded Ethereum Compatible Smart Contract Platform

2023-11-08

Abstract: The launch of the Ethereum blockchain in 2015 demonstrated the ability to deploy Turing complete smart contracts on decentralized, peer-to-peer networks in a transparent and trustless manner. It spawned a multitude of decentralized applications, creating entirely new fields of innovations such as DeFi, AMMs, NFTs, DEXs, etc. In turn, these applications increased demand on the Ethereum network. Due to scalability issues, users faced high transaction fees and long confirmation times, resulting in limited adoption. Other smart contract platforms have been created with higher throughput levels than Ethereum but lack true scalability or decentralization. To address these challenges, we present Shardeum, a sharded, Ethereum compatible, smart contract platform. Shardeum is powered by the Shardus protocol which employs dynamic state sharding, cross-shard atomic composability, auto scaling, linear scaling and many other novel technological innovations designed to solve the scalability trilemma and keep transaction fees low even as adoption increases. We believe sustainably low transaction fees are key to bringing decentralized applications to the masses.

Purpose

This whitepaper is an evolving document that details Shardeum's current vision, roadmap, technology and future direction. Due to the dynamic nature of our development process, some aspects of our technology are continuously being refined, enhanced and adapted to serve our community better and align with our mission. We deeply value the trust, patience and support the entire community has extended during these foundational phases of development. In collaboration, we aspire to architect a groundbreaking system characterized by enduring reliability, sustainability, low fees and an innate capacity to evolve in alignment with user requirements.

Disclaimer

The content of this whitepaper is intended for informational purposes only and does not constitute financial, legal, or any other form of professional advice. Readers are advised to conduct their own independent research and consult with professionals in the relevant fields before making any decisions related to Shardeum or any associated projects.

1. **No Guarantee**: The Shardeum platform, as described in this whitepaper, is presented "as is" without any guarantees or assurances of any kind, whether expressed or implied. While every effort has been made to ensure the accuracy and completeness of the information provided, there may be errors, omissions, or inaccuracies that could affect the validity or applicability of any statement made.

2. **Future Changes and Developments**: Cryptocurrencies and blockchain technologies are rapidly evolving fields. The information contained in this whitepaper might become outdated or may not reflect the current state of the Shardeum platform or its future iterations. The Shardeum development team reserves the right to make changes or improvements to the platform without prior notice.

3. **Regulatory Uncertainty**: Cryptocurrencies and blockchain technologies are subject to a complex regulatory landscape that varies by jurisdiction. There is no guarantee that Shardeum will not become the subject of future regulatory actions or prohibitions in certain jurisdictions.

4. **Potential Risks**: Engaging with Shardeum, like all blockchain and cryptographic projects, carries inherent risks. This includes but is not limited to technological vulnerabilities, market volatility and third-party actions or interventions. It is imperative that potential users and participants understand these risks before engaging with Shardeum.

5. **Investment Warning**: This whitepaper should not be considered as a solicitation or offer to buy or sell any security or as a recommendation to invest in Shardeum. Potential investors should be aware that investments in cryptocurrencies can be high-risk and past performance is not indicative of future results.

6. **Liabilities**: The Shardeum development team, contributors and any associated parties shall not be liable for any losses or damages, whether direct or indirect, arising from the use of or reliance on the information contained in this whitepaper.

Anyone considering using Shardeum or engaging in associated activities should conduct thorough due diligence and seek advice from appropriate professionals. By accessing this whitepaper, readers accept full responsibility for any actions taken based on the information contained herein.

1 Background
- 1.1 Scalability Trilemma
- 1.2 Current Solutions
- 1.3 Sharding

2 Technology
- 2.1 Design Considerations
- 2.2 Architecture
- 2.3 Performance
- 2.4 Security

3 Tokenomics
- 3.1 SHM Coin
- 3.2 SHM Allocation
- 3.3 Sale
- 3.4 Shardus License
- 3.5 Emissions After Genesis
- 3.6 Monetary Policy

4 Ecosystem
- 4.1 DApps
- 4.2 Relayers
- 4.3 Connectors
- 4.4 Explorers
- 4.5 Oracles
- 4.6 Bridges
- 4.7 Wallets
- 4.8 Shardus

5 Community
- 5.1 OCC Principle
- 5.2 Governance
- 5.3 Growth

6 Roadmap
- 6.1 Timeline
- 6.2 Current Milestones
- 6.3 Future Directions

7 Conclusion

1 Background

The meteoric rise of Bitcoin and Ethereum has brought about a revolutionary change in how society exchanges, stores, and organizes digital assets, representing a profound transformation of the digital economy and societal infrastructure. In 2015, the Ethereum network made it possible for the first time to create fully programmable smart contracts on a decentralized network, setting off a revolution. The demand for decentralized applications has since exploded with use cases like AMMs, DeFi, NFTs and gaming bringing new users into the space who otherwise were not interested in just cryptocurrency. As the popularity of Ethereum has grown over the years, the demand for transactions on the network has reached the maximum capacity of 20 transactions per second (TPS). This has led to the transaction fees rising from under 1 cent in 2015 to about $50 in 2022. Many applications which were feasible on Ethereum are no longer feasible due to the high transaction fees. There is a growing need for a smart contract platform that can keep transaction fees low not just when the network is new, but even years later as the number of transactions continue to grow.

1.1 Scalability Trilemma

In order to keep transaction fees low in a sustainable way, a network must be able to scale to accommodate an ever-increasing number of transactions. Of course scalability is not the only important factor. A network must also maintain a high level of decentralization and security. The scalability trilemma says that as a blockchain tries to achieve scalability, decentralization and security, it will only be able to attain any two of these. With security being an essential requirement, this means that there will be a trade-off between scalability and decentralization. A lack of scalability leads to slow processing of transactions and higher transaction fees resulting in a bad user experience. A lack of decentralization is not immediately felt as a bad experience, but puts all the users at risk of losing assets should the lack of decentralization be exploited. Sharding of state data across the many nodes available in a large network can increase parallel processing and also provide a solution to the scalability trilemma.

1.2 Current Solutions

Many new networks which provide the exact same or similar smart contract functionality as Ethereum have been developed to fill the gap left by Ethereum. Among these new smart contract platforms a majority of them have sacrificed decentralization to achieve higher TPS. We intentionally use the term “higher TPS” instead of “higher scalability” because these networks are not designed to scale, but rather just raise the bar from Ethereum’s 20 TPS to a higher max TPS. Typically this is about 500 TPS. The smart contract platforms in this category do not use sharding and include networks such as: **BNB Chain**, **Solana** and **Algorand**, to name a few. As these networks approach their max TPS limit, they too will experience the same high gas fees and slow processing times as Ethereum. These platforms can only increase TPS if each node in the system is upgraded to have more compute, storage and bandwidth. This is referred to as vertical scaling.

One of the first smart contract platforms to attempt sharding was **Zilliqa**. All nodes in this platform stored the complete state and every transaction was received by every node. However, for the purpose of validating transactions the network was sharded into multiple partitions based on the address space of accounts. This is referred to as compute sharding because it divides the work of validating transactions, which is usually compute intensive. But since every node still receives every transaction and updates the state of all accounts, the network bandwidth and storage operations still become a bottleneck. Zilliqa is able to achieve a higher TPS than a system with no compute sharding, but is not truly scalable since the network and storage are not sharded.

A more scalable approach to meet the growing demand for decentralized applications is to have an interconnected system of multiple sub-chains or sidechains. Such an approach is being taken by platforms such as **Polkadot**, **Cosmos** and **Avalanche**. This approach can be referred to as functional sharding, whereby decentralized applications that need to interact with one another can be launched on the same sidechain. In the case of Polkadot, each parachain can process about 1000 TPS. Even though the TPS of a sidechain may appear low compared to networks like Solana, the ability to have multiple sidechains allows such platforms to scale and the total TPS across all sidechains can surpass that of platforms using only vertical scaling. Transactions between contracts on the same sidechain are fast and easy. However, composability between contracts on different sidechains within the same network is still difficult due to the asynchronous nature of communication between sidechains. Instead assets and messages are expected to be passed between chains to coordinate interactions. The lack of atomic composability across sidechains can make it difficult to access DeFi liquidity that is fragmented across sidechains. If a sidechain reaches its TPS capacity the only way to deal with the congestion would be to vertically scale the nodes in the sidechain or to migrate some of the popular contracts to other sidechains.

The most general approach to sharding is to divide the address space of accounts into multiple fixed size regions called shards and assign subsets of nodes in the network to different shards. This is referred to as state sharding. Such an approach is being taken by platforms such as **Near**, **Harmony** and **MultiversX** (formerly Elrond). Although **Ethereum** originally planned to implement state sharding, the new approach, proto-danksharding, shards only the data to achieve higher data availability while execution is done off chain. In a network with state sharding, transactions between contracts in the same shard are fast and easy while transactions across multiple shards require cross shard coordination and are much slower. Existing state sharded blockchains must execute transactions that affect more than one shard asynchronously and sequentially; passing the transactions to each shard that is involved. That's because transactions in such networks are grouped into blocks and consensus is done at the block level; therefore, transactions that affect multiple shards risk the possibility of being confirmed in one shard, but not confirmed or rolled back in another shard. Also, maintaining atomic processing of transactions requires additional layers of complexity. Furthermore, transactions which affect multiple shards will require additional processing time proportional to the number of shards they affect. Even with these complexities state sharding is still beneficial since the TPS of the whole network will increase proportional to the number of shards it has.

1.3 Sharding

As Vitalik Buterin, the founder of Ethereum pointed out, sharding is the solution to the scalability trilemma. The most general form of sharding referred to as state sharding divides the nodes in the network into smaller groups which store a subset of the state data and process different sets of transactions to achieve parallel processing. The transaction throughput of the network increases directly proportional to the number of shards in the network. State sharding provides a way to achieve both scalability and decentralization while maintaining security. However, many of the current platforms that employ state sharding do so in limited ways and the grouping of transactions into blocks adds complexity to the sharding protocol. No decentralized network has yet demonstrated linear scaling or auto-scaling.

Throughout the rest of this paper we will just use the word "sharding" to mean "state sharding", unless specified otherwise. Although sharding increases throughput, it introduces additional complexity. Most implementations of sharding break atomic composability which allows multiple smart contracts to be chained together in one transaction. Furthermore, sharding can potentially reduce the network's security if naively implemented. Adversaries only need to compromise the byzantine fault tolerance limits of a single shard instead of the entirety of the network to halt the shard or engage in arbitrary state changes. Adversaries can also launch other attacks, such as cross-shard takeover attacks, new data availability attacks,

sharding-specific replay attacks and other forms of attacks against sharded networks. Therefore, the sharded networks' architecture must be secure, robust and mitigate existing attack vectors to retain security.

2 Technology

Shardeum is a dynamic state sharded, EVM-based, layer 1, smart contract platform designed to achieve horizontal scaling.

2.1 Design Considerations

A central design goal of Shardeum is to ensure sustainably low transaction fees. The most popular applications on most smart contract platforms are related to asset trading because the high gas fees during times of network congestion prevent other use cases from being feasible. We believe that to onboard users in developing countries and allow more use cases of decentralized applications to evolve, the smart contract platform must provide sustainably low transaction fees.

2.1.1 User Experience

When a new network is launched, the transaction fees are typically very low because the usage is much less than the capacity of the network. Users are generally happy during this time and get the false impression that fees will continue to remain low. As the popularity of the network grows, the usage begins to approach the capacity of the network and users must bid on how much they are willing to pay to have their transaction processed. At times of peak congestion the transaction fees can increase at an exponential rate. The promise and adoption of smart contract platforms has been severely restricted by the inability of current networks to scale and meet the higher TPS demands of a growing user base.

2.1.2 Developer Experience

From a developer's perspective, a lot of time and effort is invested in building decentralized applications. Resources are invested in writing smart contracts, testing them, debugging them, getting them audited, building a friendly user interface, marketing the application, building a community and acquiring a loyal user base. These are resource-intensive endeavors that require a significant investment of time, money and other resources. When a network is new and transaction fees are low, all kinds of decentralized applications are feasible. However, as the network grows in popularity and transaction fees rise, most decentralized applications that depend on low fees will be squeezed out by DeFi applications. A smart contract that gives $1 USD worth of loyalty tokens to customers is not feasible when the transaction fees are in the dollar range. For developers, smart contract platforms that cannot ensure sustainably low transaction fees represent an existential threat to their business model.

2.1.3 Energy Efficiency

The amount of energy used by a network to process a transaction will inevitably need to be paid for by the user. If a network is to achieve sustainable low transaction fees, then the energy used to process the transactions must be kept as low as possible. Therefore, using a consensus algorithm that does not require extreme energy expenditure is a must.

2.1.4 Horizontal Scaling

Having more nodes in a network can help to increase the level of decentralization and security of the network, but after a certain point, adding more nodes does not really benefit the network. Many networks don't scale because complexity of consensus and the amount of communication increases as more nodes join

the network. In networks where each node must process every transaction, the additional nodes only add to the operating cost of the network once the decentralization and security needs are met. This will eventually result in higher transaction fees for the users. Horizontal scaling through sharding is necessary for a network to optimally use the nodes that are available, keep network operating costs low and ensure low transaction fees for users.

2.1.5 Network Size

All nodes providing resources to a network must be able to operate profitably in the long run. The transaction fees must be able to cover the operating cost of the network if it is to be sustainable in the long run. This means that the number of nodes in the network must be adjustable based on the transaction throughput of the network. At times when the TPS of a network decreases, the network must be able to reduce the number of nodes, otherwise the transaction fees will have to increase to continue paying the extra nodes. This is an important factor not considered by any of the existing networks.

2.1.6 Compatibility

An important design goal of Shardeum is to not reinvent the wheel when it comes to the programming language used for smart contracts and the virtual machine used to run the smart contracts. A choice was made to have Shardeum be EVM compatible. This not only reduced the development time, but also made the Shardeum platform easily accessible to the existing Ethereum smart contracts, developer community and ecosystem.

2.1.7 Shard Security

Economic security makes Sybil attacks expensive in unsharded networks such as Ethereum. Sharded networks that are naively designed risk a Sybil attack as the cost of taking over any one shard is much less than taking over a majority of the network. An important design goal of Shardeum is to ensure that a Sybil attack on a shard has the same economic cost as an attack on the whole network through various mechanisms such as staking, slashing, standby nodes and node rotation.

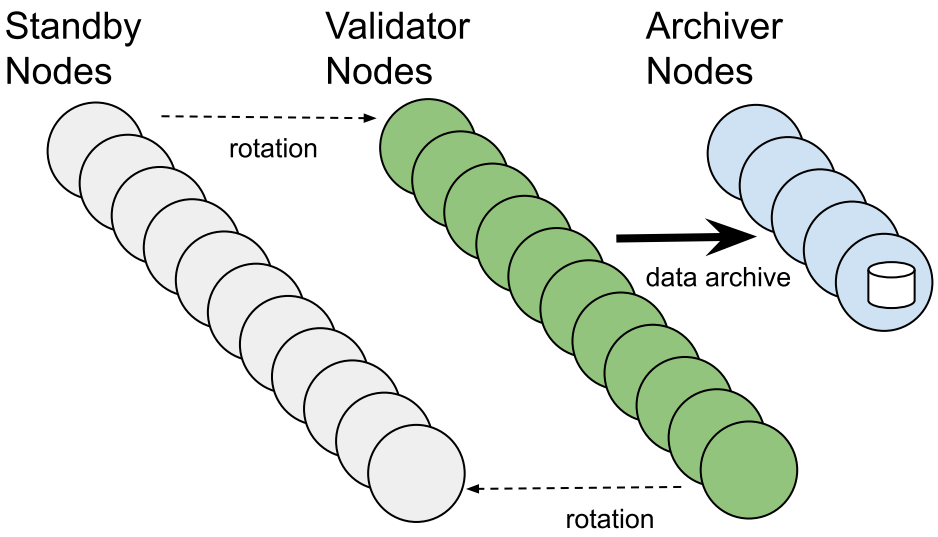
2.2 Architecture

The Shardeum network was architected based on the above design considerations. In particular, to minimize transaction costs through efficient use of available resources while maximizing network scalability, decentralization and security.

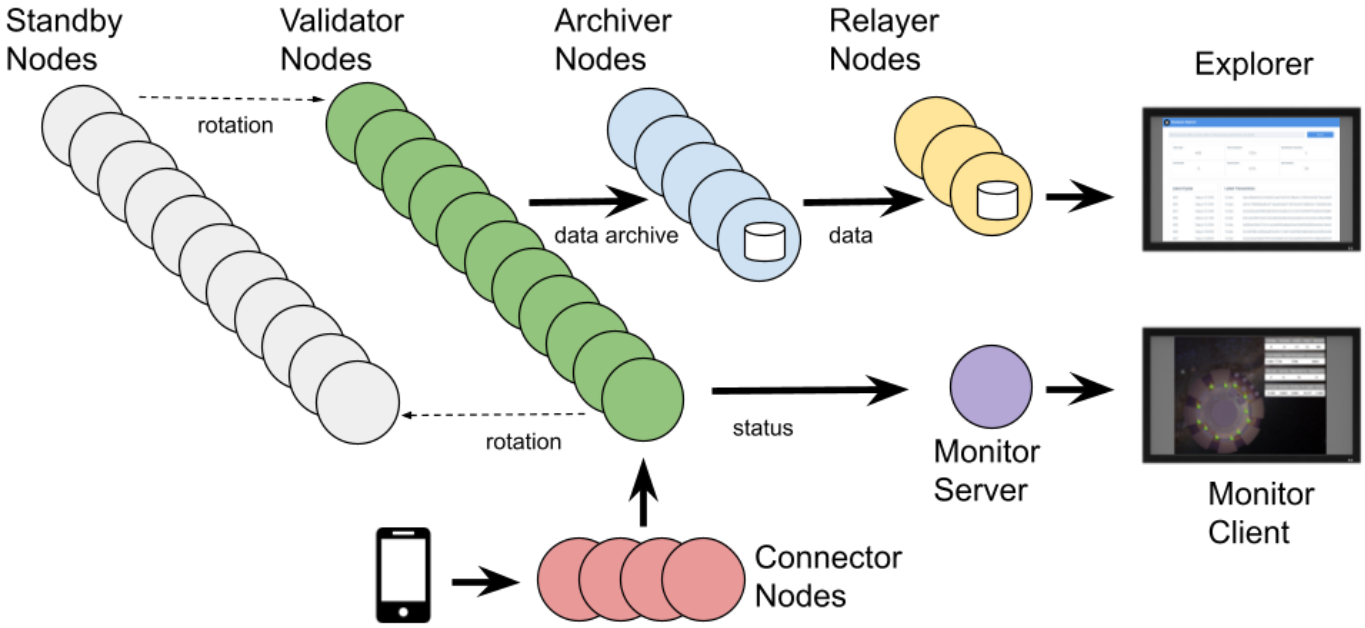
2.2.1 Network Architecture

The Shardeum network is composed of validator nodes and archiver nodes. Validator nodes that are waiting to join the network are referred to as standby nodes. The validator nodes hold the state data of the accounts they are assigned to and process incoming transactions involving the accounts within their address range to change the state. The transaction history is passed on to archiver nodes for permanent storage. The archiver nodes are not involved in any consensus and simply provide the service of storing historical network data such as transactions and receipts. Validator nodes have light storage requirements and fast syncing when joining the network as they only need to sync and store the state of the accounts they are assigned to. Validator nodes which have been active in the network for the longest time are periodically removed from the network and replaced by randomly selected standby nodes. A slow and constant rotation of validator nodes enhances the decentralization level of the network and prevents attacks from slowly adaptive adversaries. Running a validator node requires staking the native Shardeum coin called Shard with ticker SHM. The staked SHM are subject to slashing if the validator does not perform its duties as expected or is

misbehaving. However, for honest validators there is sufficient reward to easily cover the cost of operating the node and earn a good profit.



In addition, there are other types of nodes needed to move data and transactions in and out of the Shardeum network as well as monitor the health of the network. These include connector nodes, relayer nodes and a monitor server. The connector nodes provide an entry point for external wallets and clients to query and submit transactions to the network. These are the same as RPC servers in the Ethereum ecosystem. Relayer nodes communicate with archiver nodes or other relayer nodes to store and stream data produced by the network to downstream services such as the explorer. These are similar to exit nodes in the Ethereum ecosystem which are used by exchanges and explorer services. The monitor server receives status updates from active validator nodes and provides a visual view into the health of the network.



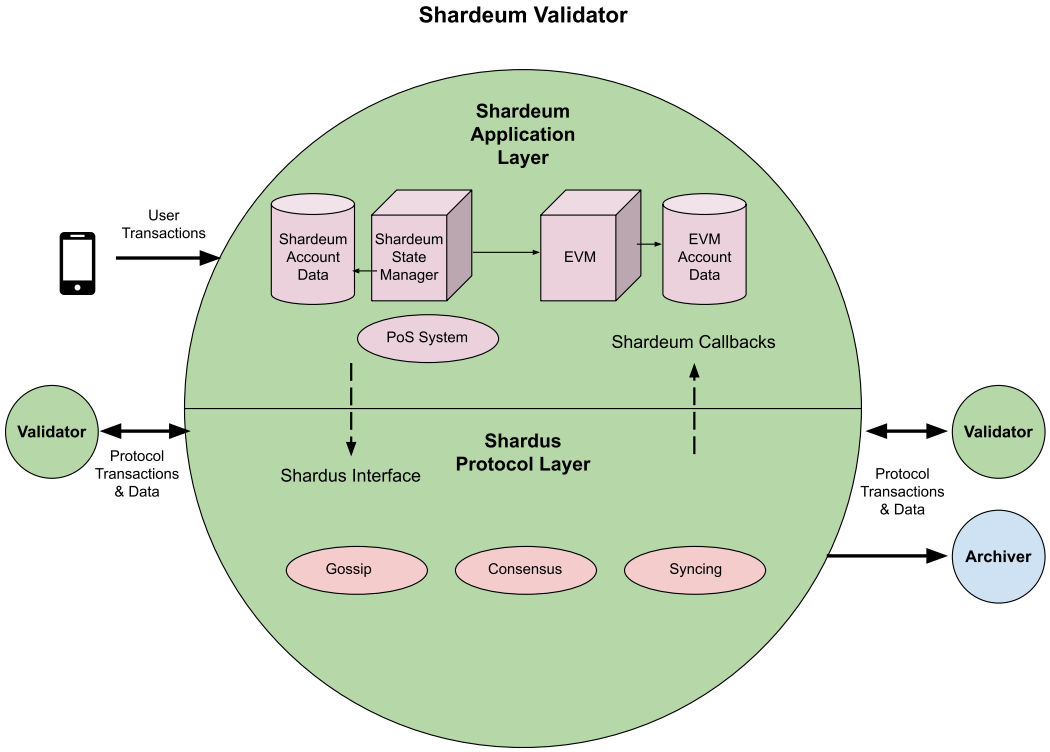
When the Shardeum mainnet is launched we expect to have a minimum of 1300 validators and 10 archivers. The number of standby validators will depend on various factors, but based on the numbers seen in the testnets we expect at least 20,000 standby validators. The number of nodes per shard will be 128. Based on usage of the network, the number of validators are expected to grow automatically to accommodate demand.

2.2.2 Validator Architecture

The architecture of a Shardeum validator node is divided into a protocol layer and an application layer. This allows for clear separation of lower level functions such as consensus, syncing, gossip, etc. from application level functions such as executing transactions and managing the application state data. A set of interface functions allows the application layer and protocol layer to communicate. The protocol layer is implemented by the Shardus software and appears as a module to the application layer. The application loads the module and invokes the methods it provides. In addition the application can register callback functions and event handlers through methods provided by the module. This allows the application layer and protocol layer to exchange data.

Account addresses are the common data that both the application and protocol layer understand. For example the endpoint on which transactions are received from users is defined by the application layer. The application layer understands the format of the transactions and can determine the involved addresses. However, to route the transaction to the validators which hold the data for these addresses, the application layer must pass the transaction along with the list of addresses to the protocol layer. The protocol layer has knowledge about the validators in the network and can route the given transaction based on the addresses provided by the application layer. The protocol layer has no knowledge about the details of the transactions or the account data and simply moves such data across the network as requested by the application layer.

The application layer is responsible for storing the account data. It also has the application logic on how to change the account data based on the instructions provided in a transaction. In the case of Shardeum, the application layer embeds the EVM and invokes it to process transactions and update the account data. The application layer also has knowledge about the staking, reward and slashing policies for the network. But it relies on the protocol layer to inform it of events that require invoking these policies. For example, if a malicious validator signs to both accept and reject a transaction during the consensus process, the protocol layer needs to notify the application layer about this so that the misbehaving validator can be slashed.



2.2.3 Shardeum Application Layer

The Shardeum application layer uses the EVM for contract execution and data storage which makes it Ethereum compatible. It also implements its own version of Proof of Stake (PoS) to protect against malicious attacks. The validator node described in the architecture section is a single piece of software that has both the Shardeum application layer and Shardus protocol layer integrated as one server process.

2.2.3.1 Shardeum State Manager

Within a conventional Ethereum client, the operation of a smart contract that involves reading, storing, or modifying state data is governed by client software containing an integrated EVM. All changes to the contract or state data must be initiated by a transaction. Shardeum also possesses an integrated EVM, akin to that in any Ethereum client. However, the state data, contract code, and account data managed by the EVM are additionally accessible to the Shardeum application layer. The Shardeum State Manager oversees any action that reads or writes this data, following the established rules of the EVM. Ordinarily, updates to the state data or contract data are prompted by a transaction. But when a node is in the process of syncing, these updates are made directly by the Shardeum State Manager.

Unlike Ethereum and most other networks, the complete state is not stored on every full node since the Shardeum network is sharded. Only a subset of the full address range will be stored on any validator. This applies to Externally Owned Accounts (EOA) and Contract Accounts (CA). The contract code is however stored by every validator.

2.2.3.2 Proof of Stake

Shardeum uses Proof of Quorum (PoQ) as its consensus mechanism, but uses PoS as its Sybil deterrence mechanism to mitigate Sybil attacks and protect the network from misbehaving and faulty nodes. In order to participate in the network, a validator node will need to lock up some stake amount which can be slashed if the node misbehaves or does not provide the expected service. The logic for staking, unstaking, rewards and penalties are handled at the Shardeum application layer, while the Shardus protocol layer notifies the application layer of critical events.

2.2.3.2.1 Staking

Certain PoS networks allow validators and sometimes delegators to become a network participant via economic ownership of tokens. Most forms of PoS have some minimum staking requirement so that network participants must have a “stake” in the network which operates as economic collateral. These forms of PoS and Delegated Proof of Stake (DPoS) networks have different sub-components, game-theory, rewards rate, penalties and staking mechanisms to align network participants in different ways resulting in vastly different outcomes. Shardeum does not have delegators as a network participant, but does have multiple other types of network participants, namely: validator nodes and archiver nodes. In order to participate in the network the validators and archivers must stake a specified amount of SHM. The minimum staking requirement for the archivers will be greater than the validators. Once a node has locked the required stake amount it can request to join the network. Since there may be many more validators wanting to join than what the network requires based on current load, the validators are put into a standby list. While a node is in the standby list or actively participating in the network, the node will not be able to unstake.

2.2.3.2.2 Unstaking

Once a node has been active in the network and becomes the oldest node it will be removed from the network. The network periodically accepts nodes from the standby list and removes nodes that have become the oldest. This period is referred to as a **cycle** and is one minute in duration. The rotation only applies to validator nodes and not archiver nodes. The archiver nodes may participate for as long as they wish once

they have been accepted into the network. An archiver node that wants to exit the network must submit a request to exit and be given permission to leave by the network. Only when a validator has been removed or an archiver has been granted permission to leave can these nodes submit a transaction to unstake.

2.2.3.2.3 Rewards

The main incentive for nodes participating in the network is to earn rewards. For validator nodes, rewards are based only on the duration the node actively participated in the network and not the time spent in standby or the amount of transactions successfully processed. Furthermore, Shardeum makes a distinction between reward accumulation and reward distribution. Even though rewards are earned while a node is active, they are not distributed until the node is removed from the network and submits a transaction to unstake. For archiver nodes the reward is also based on the time spent actively participating in the network. However, since archiver nodes are not rotated out, they are allowed to submit a transaction to receive the earned reward once a day.

2.2.3.2.4 Slashing

For every malicious action or inaction reported by the Shardus protocol layer, the Shardeum application will determine the node slashing amount and if the node needs to be removed from the network. Nodes can be penalized for misbehavior such as: leaving early, syncing too slowly, double voting, not voting, etc. Thus, nodes running on less performant hardware would be at risk of getting penalized if the node is not able to keep up with the rest of the network. Certain actions will result in the node immediately being removed from the network in addition to being slashed, while other actions such as not producing a vote when required may only result in a small penalty for each infraction. But even if the node is only slashed, it must ensure that it is always staking more than the required amount. If the sum of the initial staked amount, plus the rewards earned, minus the penalty, is less than the required stake, the node will be removed from the network. A node is allowed to initially stake more than the required stake amount and can add to the stake amount while it is participating. Thus, nodes are expected to stake more than the required amount so that a minor penalty does not cause them to be removed from the network.

2.2.4 Shardus Protocol Layer

Many of the features exhibited by the Shardeum network are inherited from the use of the Shardus protocol layer. The Shardus project has been independently developing the protocol layer since 2016. Shardus is a framework for creating linearly scalable distributed ledgers. By using the Shardus framework and adding EVM and smart contracting functionality at the application layer, Shardeum will be able to complete development much faster than otherwise possible. Listed below are some of the innovative features provided by Shardus. Since the Shardus project is in the process of filing patents, the level of detail that can currently be disclosed is limited. Nevertheless we provide details of some of the critical features that will ultimately help to reduce transaction fees.

2.2.4.1 Blockless

Blockchains have intrinsic scalability limitations due to their block-based architecture. Bitcoin bundles transactions into blocks, which are then further constricted by block size and block rate. Ethereum blocks are also constrained by block rate and their block size is indirectly limited by a parameter known as the “gas limit” as each block has a gas limit that dictates the amount of computational work that can be included in the block. Moreover, grouping transactions into blocks makes it impossible to route a particular transaction to just the set of nodes that need the transaction. A node would also receive other transactions in the block which it should not process. Processing transactions without grouping them into blocks works better for sharded networks; however, each transaction needs to be consensed upon individually. It is more efficient to do consensus on a block of transactions than on each individual transaction. This adds more processing to each transaction and at first would seem to slow down the transaction processing rate. Although this is true

for a network that is not sharded, for a network with many shards the level of parallel processing allowed by this approach achieves a much higher transaction processing rate.

2.2.4.2 PoQ Consensus

Shardus uses an energy efficient, Proof of Quorum (PoQ) consensus algorithm. All nodes in the shards that hold any data that is accessed by the transaction form a virtual transaction group. The nodes in this group are responsible for exchanging data needed to process the transaction, pausing the transaction until related transactions using the same accounts are completed, validating the transaction and then voting to either accept or reject the transaction. The actual implementation uses committees to reduce the number of votes and increase efficiency while maintaining the same level of security. At the end of the PoQ consensus for a transaction, a receipt is produced that proves either the transaction group has approved this transaction or rejected the transaction. Based on the receipt, the nodes in the transaction group can update their local state as specified by the transaction.

Consensus algorithms can be categorized as either leaderless or leader-led. All consensus algorithms in which a block is produced by a single lucky or designated node are leader-led. A key problem with leader-led consensus algorithms is that the node producing the block can determine which transactions to place in the block. Thus, it is not possible to ensure that transactions will be processed based on the order in which they were submitted to the network. For some applications like exchanges or games it is critical that transactions be processed in the order they arrived. Processing transactions in the order they arrive ensures fairness so that a single node cannot favor or discriminate against some transactions. In Shardus the set of nodes responsible for processing a transaction vote on the final state and result of the transaction. Thus, the consensus algorithm is leaderless and fair.

We choose PoQ as our consensus mechanism as we believe, when implemented correctly, this maintains long-term decentralization and has the greatest alignment in terms of incentives for network participants along with a myriad of other benefits such as a diminutive carbon footprint, no excessive waste in the form of computation or energy, broader network participation with a lower entry threshold, increased security against arbitrary transaction production and double-spends, enhanced game theory to deter and punish attackers via mechanisms such as slashing.

Proof of Work (PoW) suffers from increasing centralization concerns from the use of highly centralized mining pools, ASIC mining and excessive energy consumption concerns with an estimated annual trajectory of hundreds of terawatts used this year. DPoS also was not optimal for Shardeum as DPoS systems have a propensity to result in high levels of economic concentration amongst validators. DPoS networks also limit the number of validators that can join the network and have permissioned consensus algorithms that limit validator decentralization. Therefore we could not use a DPoS design either without compromising on fundamental beneficial properties we believe are core to the network.

2.2.4.3 Dynamic State Sharding

Instead of maintaining a complete copy of the state data, nodes in a sharded network only need to store a subset of the state data. This allows nodes to process only the transactions for which they hold data, thereby increasing the amount of parallel processing of transactions. An additional benefit of this is that there is less state data to sync when nodes are joining and reduces the sync time of a new node becoming active to a few minutes instead of hours and days. As opposed to static sharding designs, where the number of nodes in a shard and the number of shards are fixed, Shardus assigns each node to cover one or more unique address ranges in such a way that for any given address there is a well defined number of nodes holding the data for that address. Each node added to the network allows other nodes to slightly reduce the total addresses they cover while still ensuring that any given address has the specified level of redundancy.

2.2.4.4 Linear Scaling

When each node is able to immediately increase the network transaction processing rate this is referred to as linear scaling. Networks which use static sharding require a full shard number of nodes to be available before another shard can be added and so do not have smooth linear scaling. They are instead step-wise scalable. With Shardus each node added to the network increases the transaction throughput, storage capacity and bandwidth of the entire network proportionally. For example if 100 nodes provide 1000 transactions per second (TPS) then 200 nodes will provide 2000 TPS. Even adding a single node to increase the network size to 101 nodes increases the TPS to 1010. The combination of blockless transaction processing and dynamic state sharding is what allows for true linear scaling.

2.2.4.5 Auto Scaling

The backend infrastructure of large web2 services like Reddit, Uber and Facebook are designed to grow and shrink based on user demand. During a 24 hour period the number of servers needed to operate the service may see a 2x to 4x fluctuation in traffic. In the web2 world designing a service to scale infrastructure usage is critical to keeping operating costs low. In the web3 world decentralized networks have not yet begun to consider the possibility of allowing networks to adjust their size based on usage in a decentralized manner. In Shardus nodes can vote to increase or decrease the size of the network based on the load seen by the nodes. The network aggregates these votes to come to consensus on the new size it wants to attain. If it is greater than the current size, more nodes are allowed to join. If it is less, nodes are removed until the new size is reached. With this feature Shardeum will be the first network that can maintain the level of state data redundancy needed by the application while increasing and decreasing the network size to meet the throughput, storage and bandwidth demands of the users. This allows the network to minimize the operating costs since each node in the network must be paid for providing resources. This will translate to lower transaction fees for users.

2.2.4.6 Low Latency

The architecture of one network versus another can make a significant difference in the amount of time between submitting the transaction and the transaction being applied and finalized to an irreversible state. For networks that process transactions in blocks, an artificial self-imposed limit is created that is, on average, half the block frequency. In addition the possibility of blocks being rolled back if another branch of the chain becomes longer leads to an uncertain amount of time that one must wait to know that the transaction cannot be reversed. Since Shardus processes transactions as soon as they are received, the latency is minimal and limited by the server hardware, network latency and transaction compute requirements. In addition the network creates receipts to prove the final state of the transaction being accepted or rejected. This allows for low and predictable latency between the time a transaction is submitted to the time it is applied and finalized to an irreversible state. For some applications the latency of transactions reaching finality are critical to their operation. Shardeum will be able to support such applications due to transactions reaching finality within seconds as opposed to minutes or hours as is common on blockchain based networks.

2.2.4.7 Atomic Composability

In the original Ethereum network, it was possible to chain together smart contracts to compose new functionality that was not provided by any of the individual contracts. The ability to invoke multiple smart contracts and chain them together within one transaction is referred to as atomic composability. In the sharded Ethereum design this may not always be possible. When the state data of a network is sharded it breaks atomic composability since the contracts may be on different shards and transactions may only be allowed to invoke smart contracts that are all in the same shard. Shardus does not have this problem since the virtual transaction group formed to process each transaction independently will ensure that all nodes

have all data and contracts available to execute the transaction as if the network was not sharded. Thus, a big advantage of doing consensus on each transaction independently is that it allows for atomic composability even though the network is sharded. In Shardeum, atomic composability will not only result in a better user experience, but also lower the number of transactions a user must execute to achieve a final result. For example, instead of invoking a smart contract on one shard, moving the output to another shard and invoking another smart contract there, the user can invoke both smart contracts in one transaction and ultimately reduce total transaction fees.

2.2.4.8 High Decentralization

Validator nodes only store the current state of the network which includes mainly the data for accounts and contracts. Even that is limited to only a subset of the complete address range. The transactions and receipts are passed on to the archiver nodes for storage. Reducing the resource requirements of validator nodes means they can be run on low cost servers. This ensures that the barrier to join the network remains low and increases the decentralization of the network with many validators being run by community members. Decentralization is a property that is not binary, but is rather a scale of many levels. For a network to have a high degree of decentralization it must satisfy various factors. The number of nodes in the network must be sufficiently large; the more nodes the better. Shardeum plans to have a minimum of at least one thousand nodes. The node operators must be able to remain anonymous and independent so as to minimize the chance of collusion. The nodes should be geographically dispersed and not concentrated in a few data centers, hosting companies or countries. It is best if node operators are able to run nodes from home. Finally the nodes should be able to join and leave the network without requiring human involvement such as selection or voting. In Shardeum, nodes are selected to join by the nodes in the network following a well defined process that does not involve any human decisions that could introduce bias.

2.2.5 Archiver

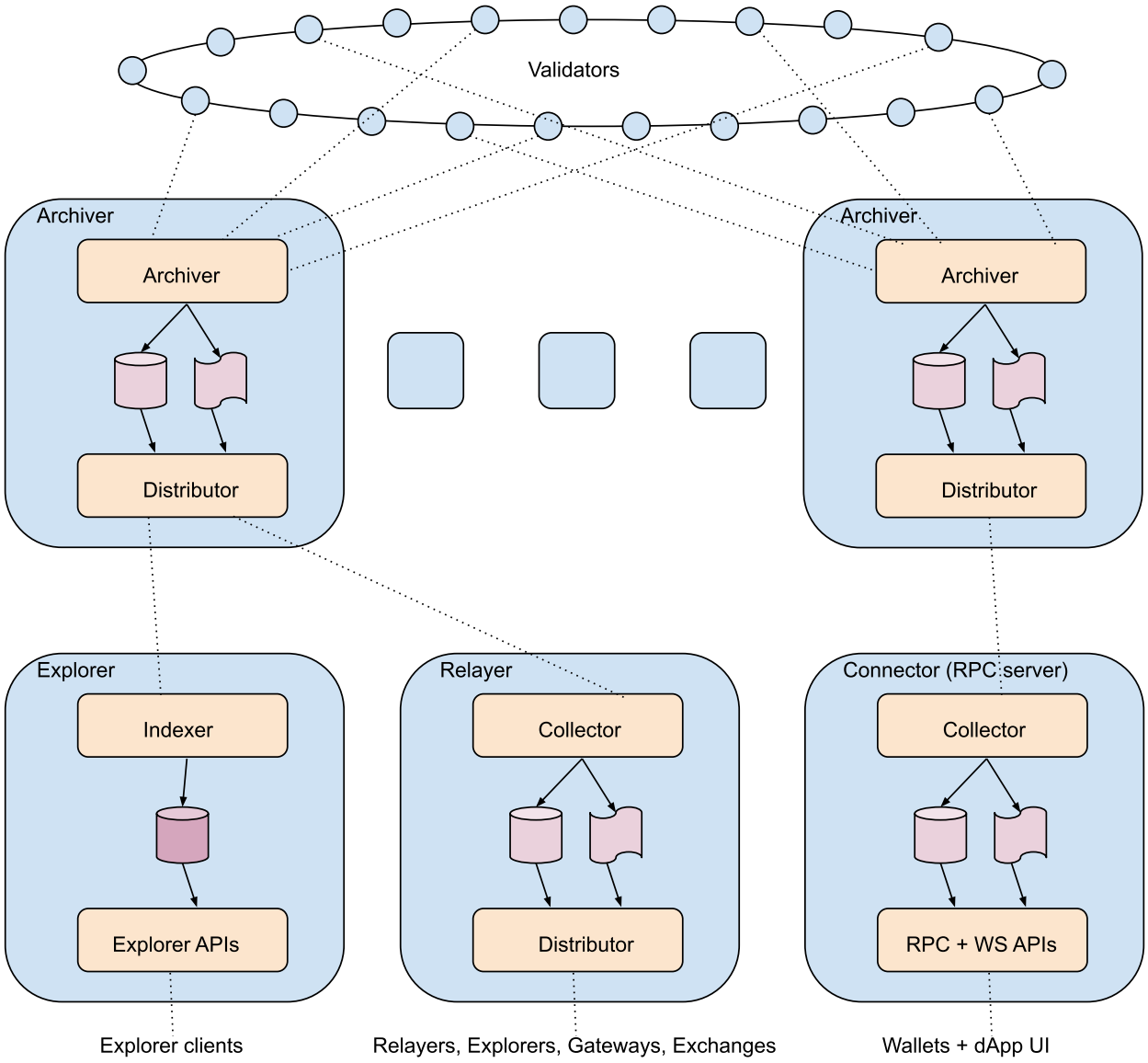
Archiver nodes store the complete state and history of the network. This allows the validator nodes to be light in terms of resource requirements. However, the archivers need to be supernodes with large amounts of storage, RAM, CPU and bandwidth. The archivers are not involved in any consensus and can be run by professional node operators.

In a sharded network like Shardeum, the state data is distributed across many validator nodes and no single validator has all the state data or knows about all the transactions. Thus, each archiver node must establish connections with many validator nodes to consolidate this distributed information. Once the complete network data is consolidated by an archiver node it can be passed on to downstream services that process and display the data to users. Examples of such services include explorers, exchanges and connectors (RPC servers in Ethereum).

Even though each archiver will hold the complete network data, there will be many archivers to provide redundancy so that even if some archivers go down, the network is not affected. In Shardeum we expect to have at least 10 archivers at mainnet launch. Since each archiver connects to about two validators in each shard, a shard size of 128 can allow up to 64 archivers. Increasing the number of archivers does not increase the performance of the network in any way, but instead adds more bandwidth load on the validators. The reason for having many archivers is only to ensure redundant storage of the complete network data.

The archiver nodes must also have an economic stake to join the network. A node can also be slashed if it leaves the network without first requesting to leave. The archiver nodes also earn a reward for participating in the network. The reward for running an archiver is expected to be about 10x more than running a validator since the hardware requirements for running an archiver will also be much higher.

In addition to earning rewards from the network, archiver nodes can also earn from providing a data subscription service. For example, exchanges will need data and events for accounts owned by the exchange and may not want to run an archiver node.



2.2.6 Relayer

Once the archiver has consolidated and stored the data from the network, a data distribution protocol helps to stream the data throughout the ecosystem. A service called the distributor running on the archiver can send the network data to downstream services. The distributor provides APIs to download historical data as well as stream the most current data. A service called a collector can connect to the distributor to get the network data and make a local copy. A relayer node runs both a collector and distributor on the same machine to fan out the network data. There can be multiple tiers of relayers helping to distribute the network data to the rest of the ecosystem. Relayers need to be supernodes with large amounts of storage, RAM, CPU and bandwidth. It is expected that relayers will be run by professional node operators. Relayers are not a direct part of the Shardeum network and do not receive any reward. It is expected that relayers will charge a subscription fee to those receiving data in order to support the service.

2.3 Performance

Although Shardeum is still in development, the Shardus technology used at the protocol level has been demonstrated to achieve linear scaling. In the Q3 2021 update event, a network of 1000 nodes running on AWS t3.medium hardware was shown to reach 5000 TPS of signed coin transfer transactions across shards. The shard size was 20 nodes; therefore with 1000 total nodes the network had 50 dynamic shards. The shard size was kept small to allow for more shards and better demonstrate linear scaling. Auto-scaling was also demonstrated in an earlier update event with the network detecting the transaction load and growing by allowing more standby nodes to become active nodes as the TPS increased. Later when the load was removed, the network shrank back down by removing some of the active nodes. The following table gives a sense of how much 5000 TPS is:

	Bitcoin	Ethereum	Polygon	BinanceSC	Nasdaq	Visa	Twitter	Emails	Google	Text	Whatsapp
Avg TPS	3	15	40	60	1,200	6,500	10,000	11,000	99,000	254,000	1,100,000
Peak TPS	8	20	120	180		30,000					

For typical smart contract transactions like those seen on the Ethereum network, the processing rate for a single processor is about 10 TPS. The Shardeum network TPS rate is given by:

$$\text{network\_tps} = \frac{node\_tps \times total\_nodes}{shard\_size}$$

For a network with a shard size of 128 nodes and a total network size of 1280 nodes, the network TPS would be 100 TPS. The Ethereum network currently has over 800,000 validators. If the Shardeum network had this many nodes, the TPS of the network would be about 62,000 TPS.

An additional benefit of sharding is the increased storage capacity of the network while the storage capacity of each node remains constant. Using the above example of 1280 nodes with a shard size of 128; if each node has a storage capacity of 100 GB the network capacity would be 1000 GB. Doubling the number of nodes to 2560 would double the network capacity to 2000 GB even though each node still has a storage capacity of 100 GB.

2.4 Security

Given that Shardeum functions as a sharded network, processing various transaction sets across its multiple shards, its security model diverges from non-sharded Layer 1 architectures: it must defend against both standard Layer 1 threats and those attacks specific to sharded architectures. To this end, we provide an overview of common attacks, Shardeum's defenses and mitigations. The list is non-exhaustive due to the evolving nature of security threats and to maintain a focused and concise document. Readers are encouraged to explore additional resources for a more comprehensive understanding of potential security challenges.

2.4.1 Sybil Attacks

Description: Sybil attacks are a type of attack in which an adversarial actor or group of adversarial actors attempt to gain control of a disproportionate percentage of a network by creating pseudonymous identities that can disrupt the network and compromise its security.

Mitigations: To counter Sybil attacks, DLTs that wish to retain decentralization and secure use a Sybil deterrence mechanism to impose a cost for network participants to join the network as they cannot use

typical centralized solutions such as identification authorities. In the case of Shardeum, nodes must stake a minimum amount of SHM. This imposes an economic burden on the attacker and helps to deter a Sybil attack. Also nodes which misbehave can be slashed and removed from the network. Losing the staked amount makes it harder for an attacker to continue to attempt additional attacks.

2.4.2 Shard Takeover Attack

Description: An attack in which an adversary fills a shard with their own nodes in order to control the shard. Once an adversary controls 33% of the nodes they can halt the shard and if they control 66% of the nodes they can forge transactions within the shard.

Mitigations: Shardeum prevents single-shard takeover attacks by not allowing nodes to select which shard they join. Nodes are randomly selected to be rotated into the active set and prevented from choosing their shard or address range. Thus, to achieve a 66% takeover of a shard would require a 66% takeover of the entire network including active and standby nodes. Since nodes are required to stake even as a standby node, the economic cost of a shard takeover attack would be very high. In addition the rotation of nodes helps to ensure that an adversary cannot build up an attack over a long period of time.

2.4.3 Nothing at Stake

Description: During a network fork, an adversary in the form of a validator node can continue to validate on a fork of a blockchain incurring no cost for validating both chains. On PoW mechanisms, miners are incentivized to not waste network resources mining on the alternate chain, but PoS validators can potentially still earn transaction fees on both versions of the network, meaning they have nothing at stake that incentivizes or disincentivizes them from supporting illegitimate forks unlike on PoW networks.

Mitigations: Shardeum does not use PoW or PoS as its consensus mechanism, it uses PoQ for consensus and PoS as a Sybil deterrence mechanism so the stake is not part of the consensus. Therefore, the network will not accept a fork based on the longest-chain rule like on PoW or based on overall weight staked like on PoS and an adversary cannot double-sign or equivocate a transaction without getting slashed. Additionally, the digital signatures used to produce a receipt via PoQ can't be forged.

2.4.4 Long Range Attacks

Description: A long-range attack is when an adversary goes back to the genesis block and forks the blockchain.

Mitigations: Long-range attacks on Shardeum are not possible as we do not use PoW or PoS as a consensus mechanism. Therefore, the network will not accept a fork based on the longest-chain rule like on PoW or based on overall weight staked like on PoS and an adversary cannot double-sign or equivocate a transaction without getting slashed. Additionally, the digital signatures used to produce a receipt via PoQ can't be forged.

2.4.5 Censorship

Description: An attack in which a validator has control of the transactions which will be included in a block and uses their status as validator to prevent the transaction going through, possibly via blacklisting.

Mitigations: By having a blockless architecture, Shardeum prevents validators from altering the order of transactions within blocks and also deciding which transactions are included in a block. Shardeum also uses leaderless consensus meaning that no validator is elected as leader and so no individual validator can

single-handedly prevent transactions being processed. A small group of validators could not put out a vote on specific accounts or transactions, but for an attacker to engage in censorship effectively they would need to control 33% of a shard which is countered by minimum stake requirements, node rotation and disallowing nodes to select which shard they join.

2.4.6 DoS or DDoS Attack

Description: Denial of service or distributed denial of service is an attack where nodes are knocked offline and fail to meet liveness requirements. A distributed denial of service is similar but with many nodes taking part to knock a node offline.

Mitigation: Nodes should have DDoS protection by running their node with an ISP with a strong DDoS protection mechanism. If a node is successfully taken down, other nodes in the shard have account range redundancy and can still validate the transaction. Another node or shard will always have an opportunity every cycle to join when nodes are downed. If a shard was knocked down via a form of DoS it would not be able to produce a receipt and the user who submitted a transaction would be informed that their transaction didn't go through resulting in certain accounts being temporarily unable to process the transaction. Nodes that stay in the network and aren't successfully taken offline will also more likely stay in the network in the long-run due to superior DDoS defenses.

2.4.7 Transaction Flooding

Description: An attack in which an adversary floods the network with valid transactions in an attempt to slow the network.

Mitigation: Shardeum prevents transaction flooding by imposing economic costs in the form of SHM gas fees. These fees are inexpensive enough to allow a beneficial user experience, but will cost an adversary a large amount should they engage in a transaction flooding attack. Additional measures such as requiring higher fees from excessively active source accounts can impose a burden on an adversary without impacting regular users.

3 Tokenomics

Shardeum employs dynamic state sharding to attain linear scalability, ensuring that every node added boosts network throughput instantly. As a result, Shardeum can enhance its TPS capacity and maintain low fees sustainably. Our simulations revealed that the economics of linear scalability required a unique approach to how the native coin SHM is issued.

3.1 SHM Coin

The native coin on Shardeum is Shard and has SHM as its ticker symbol. SHM is the foundational monetary unit of Shardeum and serves a wide range of functions. Each SHM is divisible to 18 digits, similar to ETH on the Ethereum network. We will also refer to it as the SHM token, since a tokenized version of the coin is also used by smart contracts and to be consistent with common terminology.

SHM is a utility token with various uses, including:

- Staking: Network participants can stake their SHM, increase the network's overall security and earn rewards for their active participation
- Governance: SHM is a governance token conferring stakers governance rights, allowing them to decide and vote on the future of the network. Shardeum will not use a 1 token 1 vote system

- Reward Token: SHM is given as a reward, for example: via airdrop, ecosystem rewards and network participation
- Gas Token: SHM is a gas token allowing network users to pay fees to execute the required compute units for transactions and other more complex transactions involving smart contracts on the network
- Transaction Fee Burning: All transaction (tx) fees on Shardeum are burned, allowing the network to remove the SHM from the circulating supply
- User Staking: We are considering the possibility of users staking during deflationary periods. Since Shardeum is not based on Delegated Proof of Stake (DPoS), there are no delegation pools

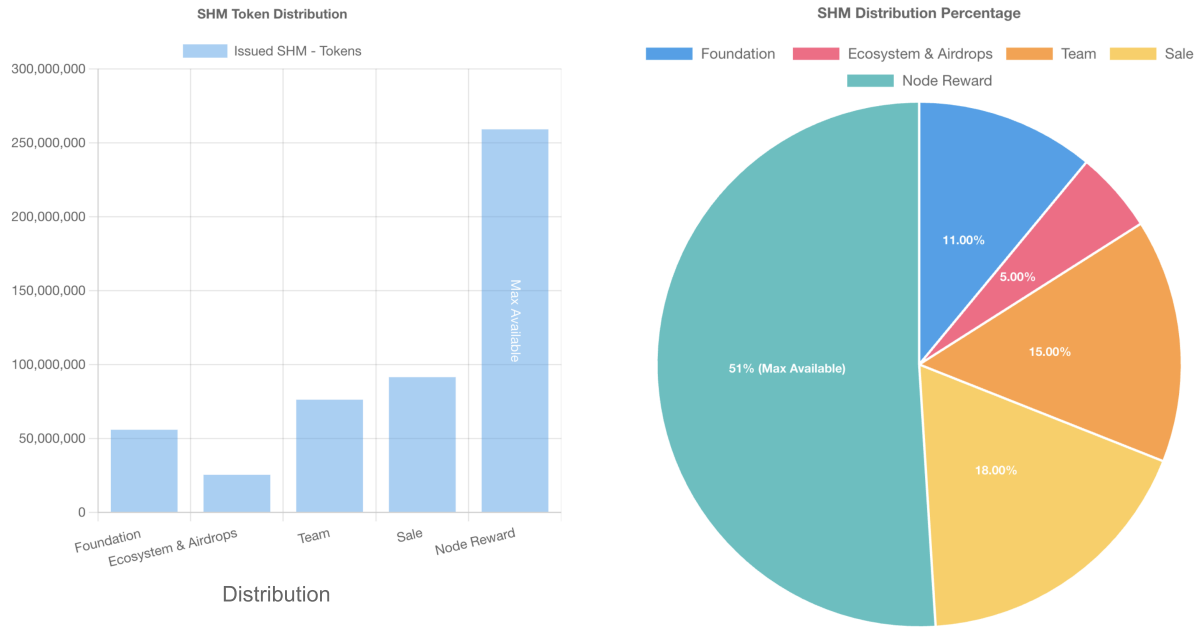
3.2 SHM Allocation

Max supply: 508 million SHM
Distribution:

- 51% Community (259,080,000 SHM) – reward to validator and archiver nodes
- 18% Sale (91,440,000 SHM) – 3 month cliff then 2 year daily linear vesting
- 15% Team (76,200,000 SHM) – 3 month cliff then 2 year daily linear vesting
- 11% Foundation (55,880,000 SHM) – unlocked at Token Generation Event (TGE)
- 5% Ecosystem (25,400,000 SHM) – unlocked at TGE

Since transaction fees are burned, the maximum SHM supply of 508 million will never be reached.

The following image shows the SHM distribution as a bar chart and pie graph.



3.3 Sale

Of the 18% of SHM supply allocated for Sale, about 32% has been sold to private investors in two seed rounds.

- Seed Round (September '22) - $0.80 per SHM, $18,719,600 raised
- Strategic Round (June '23) - $1.00 per SHM, $5,916,037 raised

3.4 Shardus License

The Shardeum project will obtain a license for the Shardus software by allowing Shardus token (ULT) holders to claim 1% of the max supply (5,080,000 SHM) in proportion to the Shardus tokens they hold. More details about the claim event will be posted on the Shardeum website in the near future. The founders and several members of the team hold Shardus tokens.

3.5 Emissions After Genesis

The following accounts will receive SHM immediately at Genesis/mainnet launch:

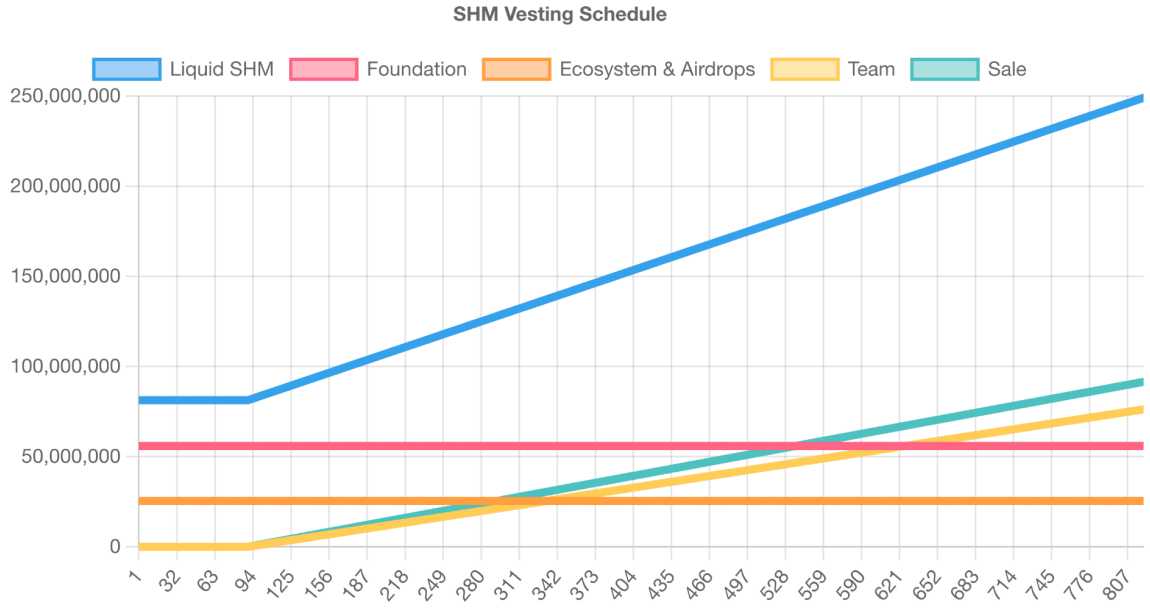
- Foundation account: 11% of 508M; 55.88M SHM; becomes available at mainnet launch
- Ecosystem account: 5% of 508M; 25.4M SHM; becomes available at mainnet launch

The following accounts will start receiving SHM 3 months (90 days) after the mainnet launch in 730 daily installments:

- Team account: 15% of 508M; 76.2M SHM; about 104,383 SHM per day after 90 days
- Sale account: 18% of 508M; 91.44M SHM, about 125,260 SHM per day after 90 days

The remaining 51% is set aside for node rewards; SHM is created when given, starting at mainnet launch, based on the node reward setting.

The graph below shows the liquid supply of SHM in the 820 days after network Genesis. The data encompasses the entire Team and Sale SHM vesting period, but does not include any of the 51% SHM supply that will be used for node reward during this time.



Liquid SHM Supply

3.6 Monetary Policy

The Shardeum monetary policy has been designed with help and feedback from the community since April 2022.

The SHM monetary policy is designed to make the supply stable, predictable, adaptable and scarce. First, we will evaluate how SHM is scarce:

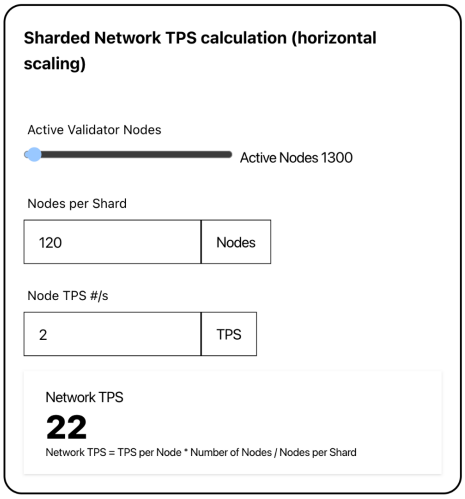
- SHM has a fixed max supply, making it a deflationary asset
- There will only ever be a maximum of 508,000,000 in existence and this parameter is inalterable even by any form of future voting
- All transaction fees are burned on Shardeum, with no transaction fees going to any miner or validator nodes, making SHM scarcer by design
- SHM tokens from slashed validators are also burned, adding to the scarcity of SHM
- This fee structure bolsters the scarcity of SHM's tokenomics as Shardeum undergoes technological maturity and sees greater adoption (with increased transaction total, smart contract deployments, etc.) SHM's supply will likely become ultra deflationary as the number of burned tokens will become higher than those given out as node rewards

In summary, Shardeum's tokenomics have been designed to be programmatically scarce at launch and for the scarcity to increase simultaneously with increased burning, linear scaling and overall adoption, making it exceptionally scarce in the future.

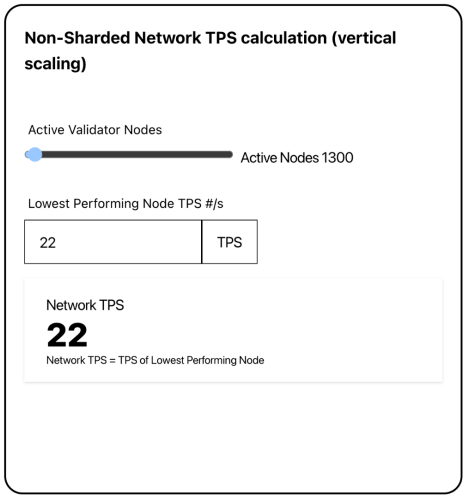
3.6.1 Design Considerations

The first step in understanding why a unique issuance method is required is to comprehend how Shardeum scales from a hardware perspective. Shardeum increases its network throughput (TPS) by increasing the number of active validator nodes. This requires a pool of standby nodes to be ready to join as needed. The rewards given to active validator nodes must be enough to ensure that the nodes will be profitable, even though they would not be earning any rewards while in standby. By increasing the reward given to active validators the network can ensure there will be sufficient nodes ready to join when traffic (TPS) increases. Not having enough standby nodes to draw from would cause the active nodes in the network to become overloaded and the network would have to reject some transactions.

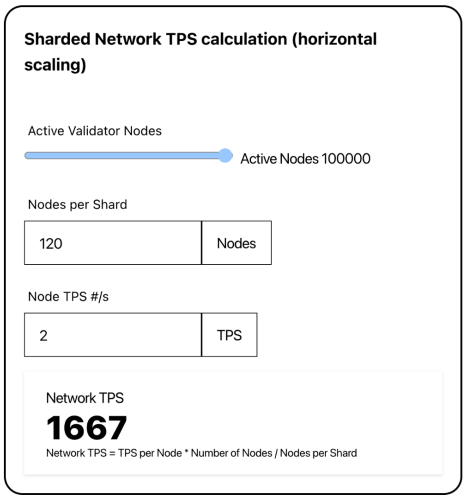
The figures below compare horizontal and vertical scaling. The two figures on the right demonstrate how non-sharded networks achieve higher throughput (TPS) by scaling vertically; this scaling method increases the CPU, RAM and network resources of each node to increase the network's capacity. This approach has two significant drawbacks. Firstly, no matter how many nodes join the network, the lowest-performing node will determine the TPS (lower right figure). The other drawback is hardware cost; these networks require high-end equipment to reach high network capacity, making it expensive to become a validator and reducing overall decentralization. Shardeum uses horizontal scaling to increase network throughput (TPS). As demonstrated by the two figures on the left; this approach adds more nodes with similar resources per node and uses parallel processing to increase the network's capacity. The network TPS is practically unlimited, providing more nodes can be added to the network (lower left figure) and due to the low hardware cost/requirement, overall decentralization increases.



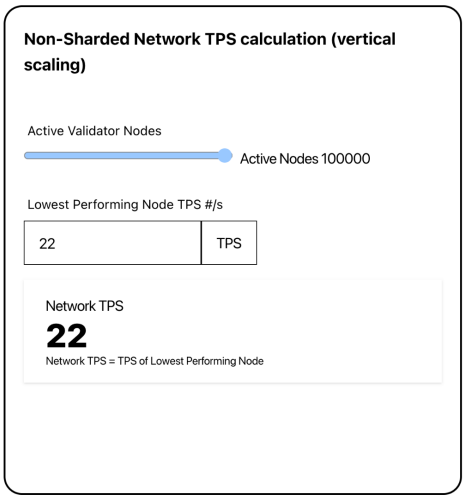
Horizontal Scaling 1300 Nodes



Vertical Scaling 1300 Nodes

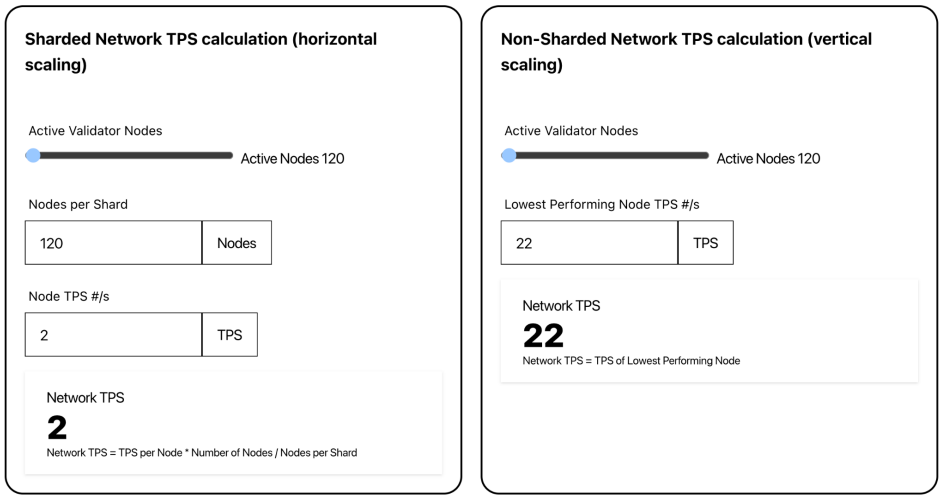


Horizontal Scaling 100k Nodes



Vertical Scaling 100k Nodes

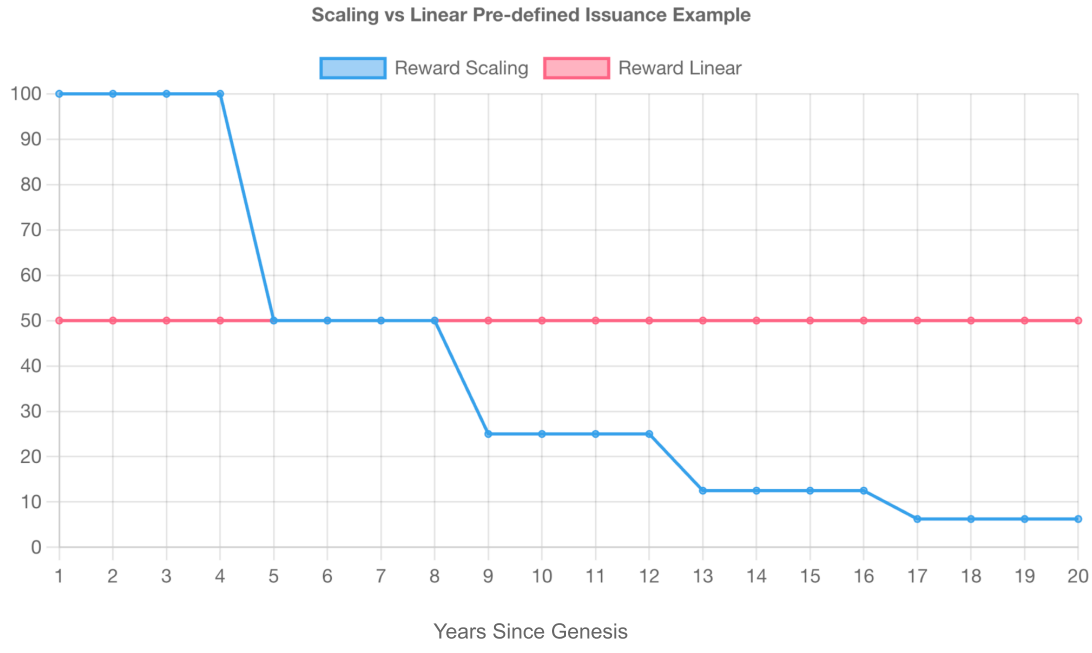
Unstable validator numbers pose a significantly higher risk for a sharded horizontally scaling network like Shardeum; this is because if the network becomes unprofitable for the node operators and the number of validators drops, so does network throughput (lower left figure), this could result in probabilistic transaction rejection as a result of TPS decreasing or the network coming to a temporary halt due to the network safety mode being triggered. These risks don't apply to non-sharded vertically scaling networks because the lowest performing node determines network throughput (TPS); if validator numbers reduce, the network throughput (TPS) stays the same (lower right figure).



Horizontal Scaling 120 Nodes Vertical Scaling 120 Nodes

3.6.2 Existing Approaches

Firstly, we analyzed pre-existing monetary policies of a multitude of Layer 1's to understand, explore and see if any of their issuance schedules were broadly applicable to Shardeum. We found that there were two prominent types of issuance schedules, namely scaled and linear issuance schedules (figure below).



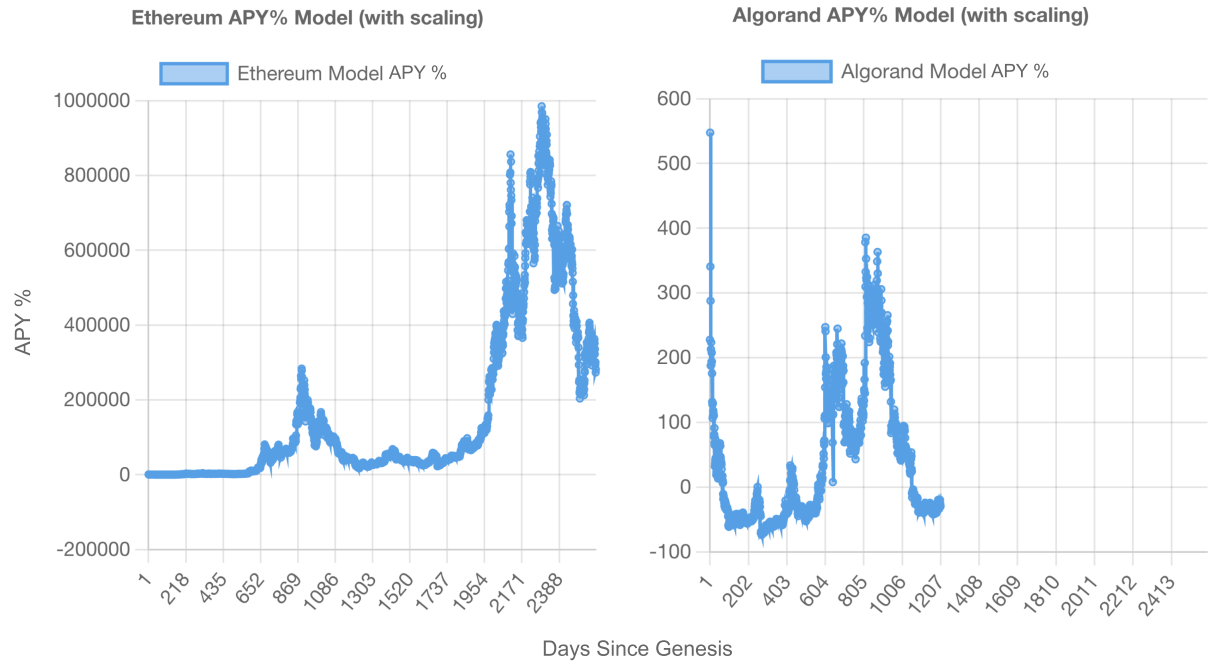
Scaled predefined issuance schedules reduce reward percentage through disinflation at different points, whether approximately every 4 years (210,000 blocks) as in the case of Bitcoin or every year with a 1.5% disinflation rate as in the case of Solana. Linear issuance schedules mint a set amount of the network's native asset over a specified duration and this is the issuance schedule of projects like Algorand. These issuance schedules did not apply to Shardeum; they caused an inefficient and unpredictable network, a lack of profitability, or too high an APY for long-term sustainability.

Secondly, we ran simulations on the network to see how some of these pre-existing tokenomics models applied to SHM, confirming they were not applicable due to the aforementioned reasons and that the optimal SHM model would always be flexible enough to reach supply equilibrium.

Thirdly, the unique scalability properties of Shardeum, necessitate a unique tokenomics model and monetary policy. Shardeum can automatically assemble and disassemble shards to expand and contract the network to increase throughput or storage capacity, as Shardeum uses dynamic state sharding, autoscaling and linear scaling to solve the blockchain trilemma.

These unique scalability properties necessitated the creation of standby nodes and active nodes. The ratio between these nodes is called the S:A ratio and is critical to the efficient and secure operation of the network. The S:A ratio is the number of standby nodes to the number of active nodes. Standby nodes are randomly selected each cycle and rotated into the active set to start consensus; simultaneously, active nodes participating in the active set for the longest time are removed and rotated back into the standby set, becoming standby nodes again. This kind of rotation increases node and network decentralization. A higher S:A ratio ensures the network can scale more effectively, enhances decentralization and mitigates the risk of a majority takeover or a Sybil attack. If the S:A ratio gets too high, negligible benefit to the network occurs and can lead to the network becoming inefficient. Conversely, the network is more efficient if the S:A ratio is lower, but if the ratio falls too low, it poses the most significant risk because the network would lose its ability to scale.

The graphs below demonstrate how following a predefined issuance schedule with a horizontally scaling architecture results in massive fluctuations in APY%. In the bull case (Ethereum), the native asset increases in value rapidly, causing the network to become wildly profitable for node operators and resulting in the network becoming inefficient (delivering a higher APY% than needed). In the bear case (Algorand), the price of the native asset decreases, which results in the network becoming unprofitable for node providers (delivering negative APY% returns when hardware expenses are considered). The likely outcome would be a massive reduction in node operators; this causes significant problems for horizontally scaling networks, which could mean the network loses its ability to scale (increase TPS).

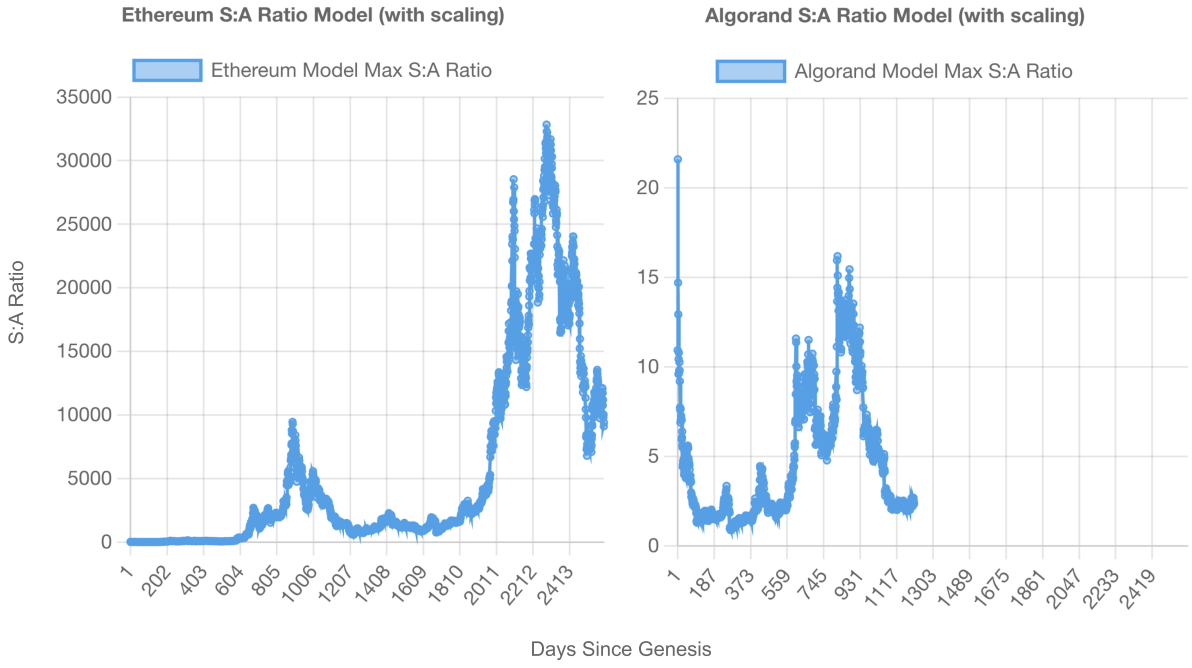


Ethereum APY%

Algorand APY%

*note Algorand data complete from Genesis to current day

In reality, in the Shardeum network, APY% would likely always find some form of equilibrium; when the network is more profitable, it would have a much higher S:A ratio. Likewise, the S:A ratio would reduce if the network becomes less profitable. The graphs below highlight how, depending on native token price action, the network could end up in the bull case (Ethereum) with an S:A ratio that is unnecessarily high and wasteful of computing resources. In the bear case (Algorand), the S:A ratio could become so low it compromises security and could cause the network to lose its ability to scale (increase TPS).



Ethereum S:A Ratio Model

Algorand S:A Ratio Model

*note Algorand data complete from Genesis to current day

An active node can only collect the reward after it has been cycled out of the active set into a standby node and is no longer active. Giving a reward to standby nodes would incentivize them to try and collect the reward without actually participating in the network and going active. Thus, standby nodes do not receive a reward.

We deduced the ideal issuance approach would be adaptable enough to:

- Ensure running a validator node on Shardeum is profitable, providing reasonable hardware costs
- Ensure the SHM supply is always heading toward equilibrium (SHM burned = SHM issued), enabling the network to reward node operators forever
- Keep the S:A ratio high enough that the network always retains its ability to scale and never sacrifices security
- Keep the S:A ratio low enough so that the network never issues more SHM than required to secure the network and avoid the unnecessary waste of natural and computational resources
- Actively promote network efficiency by adjusting the validator node reward to ensure operators receive just enough APY% to match market rates

All pre-existing monetary policies achieved only some of the above. Therefore, we opted for a bespoke approach called a capped elastic supply model.

3.6.3 Capped Elastic Supply

A capped elastic supply model is a tokenomics model in which the token issuance schedule is elastic and not predefined. Token issuance can be inflationary, deflationary, or disinflationary, depending on what is optimal in the current micro or macroeconomic environment. A capped elastic supply model allows optimal token issuance in the overall economic climate but will never exceed its maximum supply (508 million).

3.6.3.1 Adjustable Parameters

The literature above shows that controlling the S:A ratio and finding supply equilibrium is critical to the network's smooth yet efficient operation. We understand that attempting to predict countless variables to create a predefined issuance schedule is impossible. Instead, we introduce a small number of adjustable network parameters that can initially be overseen by the FDAO.

FDAO - We refer to the Foundation and DAO collectively as the FDAO.

The adjustable parameters are:

Tx Fee $ - This is the target fee for a token transfer transaction. The complexity of the transaction determines the fee paid. For example, simple SHM transfer transactions will cost less; more complex transactions, such as AMM transactions, will cost more.

Adjusting this parameter can either increase or decrease the network income (daily tx volume * tx fee) because all tx fees are burned; this parameter can cause the supply to be inflationary or deflationary and steer it towards equilibrium.

Node Reward $/hr - This defines how much each active node in the network receives as dollars per hour. Although it is specified in $, it is paid out in SHM.

Adjusting this parameter can increase or decrease the network operating cost (active nodes * node reward). Because the node reward parameter is how new SHM are issued; this parameter can cause the supply to be inflationary or deflationary and steer it towards equilibrium.

The secondary impact this parameter can have is either increasing or decreasing the S:A ratio. If the node reward per hour increases, the network can sustain a higher number of nodes at a given APY%, increasing the S:A ratio. Conversely, if the node reward per hour is reduced, the network will not be able to sustain the current number of nodes at a given APY%, causing the S:A ratio to reduce.

Stake Amount $ - The amount of SHM a node must stake to join the network. It is specified in $ but staked in SHM based on the Stable Price. Some or all of the stake can be lost if the node misbehaves or falls behind in processing; this ensures that operators run nodes on suitable hardware.

Adjusting this parameter can either increase or decrease a node's overall APY(%)/year (100 * income * 365 / stake amount); this will either increase or reduce the S:A ratio as running a node becomes more or less profitable.

3.6.3.2 Effects on Supply

It's essential to understand how the relationship between the tx fee $ and node reward $/hr allows the supply to be elastic. This relationship can be in one of three states:

If Network Revenue < Network Expense then Supply Inflation
where

Network Revenue = daily tx volume * tx fee $
Network Expense = active nodes * node reward $/hr * 24

Supply Inflation: If the network's daily revenue (from transaction fees) is less than its daily expenses (the cost of operating nodes), the network is not generating enough income to cover its costs (pay node operators). In this case, the network will issue more SHM to node operators than it is burning from transaction fees. The network will become inflationary, increasing the supply of SHM in circulation.

If Network Revenue > Network Expense then Supply Deflation

Supply Deflation: If the network's daily revenue (from transaction fees) is greater than its daily expenses (the cost of operating nodes), the network is generating more income than it needs to cover its costs (pay node operators). In this scenario, because all the transaction fees are burned, the network will become deflationary, reducing the supply of SHM in circulation.

If Network Revenue = Network Expense then Supply Equilibrium

Supply Equilibrium: When the network's daily revenue (from transaction fees) equals its daily expenses (the cost of operating nodes), it operates in a balanced financial state. In this case, the network is neither inflating nor deflating its token supply; it's maintaining a stable supply (Equilibrium).

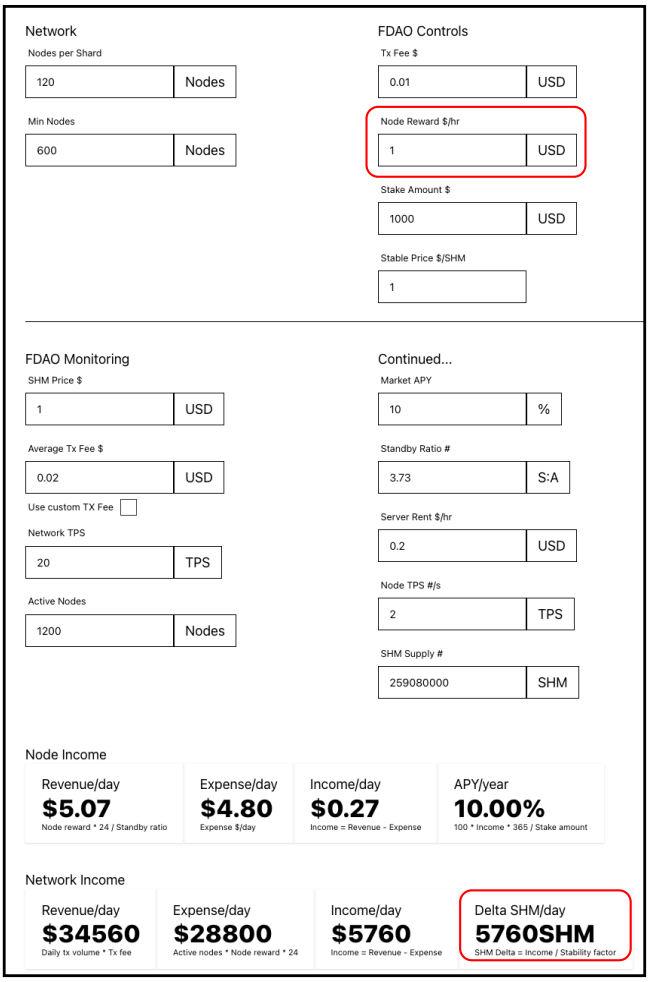
If the parameters overseen by the FDAO can change which of these states the network is in, it will ensure smooth and efficient operation.

3.6.3.3 Static Simulations

The left figure below shows the network has a positive income and delta; this means it generates more income from transaction fees than it spends paying nodes for validating the network. In this case, the network is deflationary because it will burn more SHM from the transaction fees than it will be issuing to the node operators.

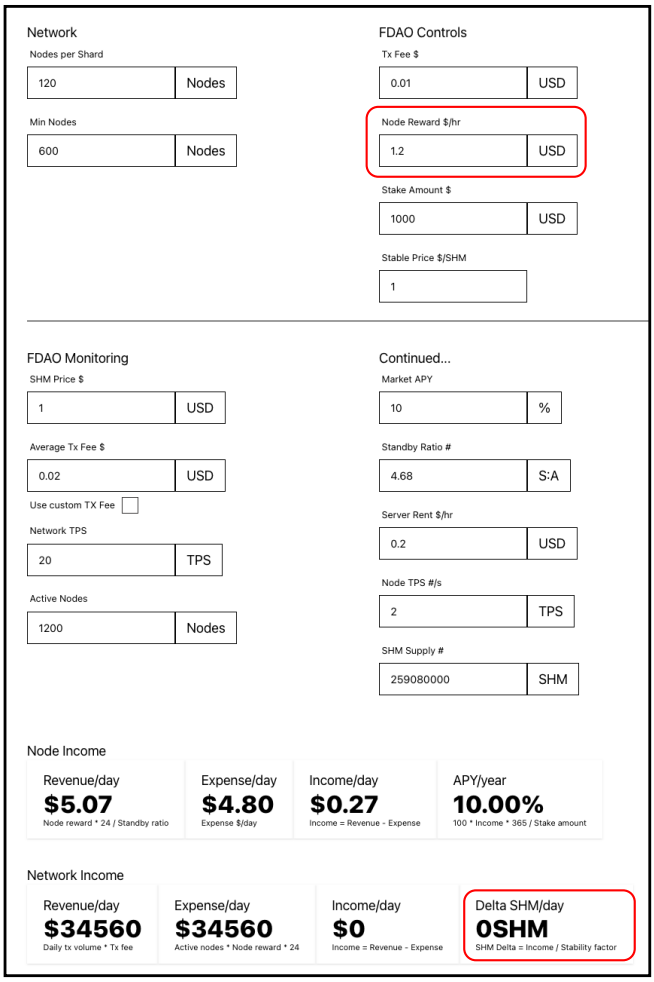
The right figure below demonstrates how changing a single FDAO-overseen variable can move the network into a different issuance state. The node reward $/hr variable was adjusted from $1 to $1.2, balancing the revenue and expenditure of the network, resulting in the network finding supply equilibrium.

Supply Deflation



Network Parameters Deflation

Supply Equilibrium

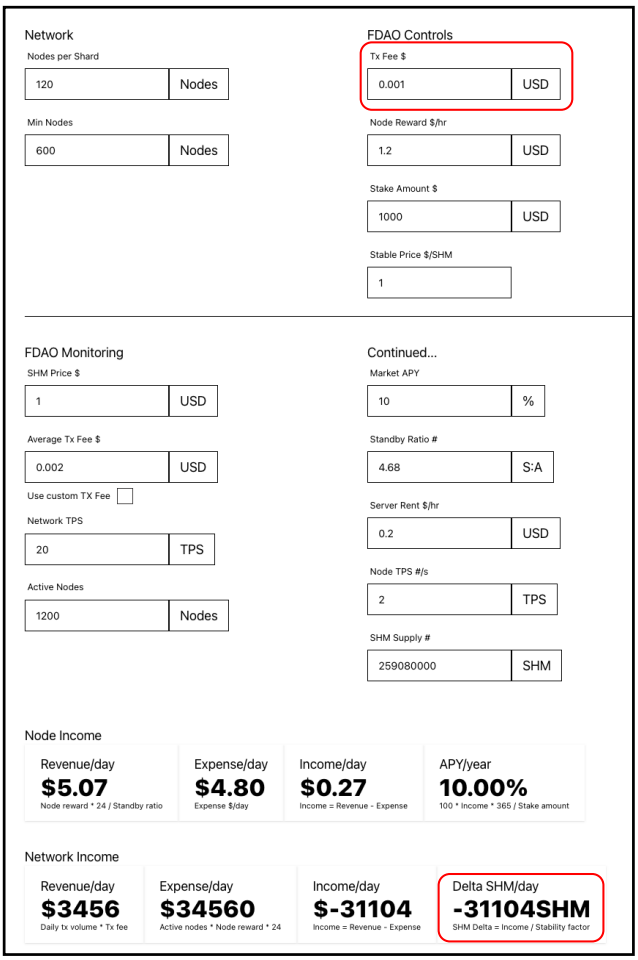


Network Parameters Equilibrium

The left figure below demonstrates how a change to another key FDAO-overseen variable (tx fee $) can impact the issuance state. In the example below, the network is moved from the equilibrium state in the above right figure to an inflationary state with a minor change to the tx fee $; after the change, the network is issuing more SHM via node reward than it's burning from the received transaction fees.

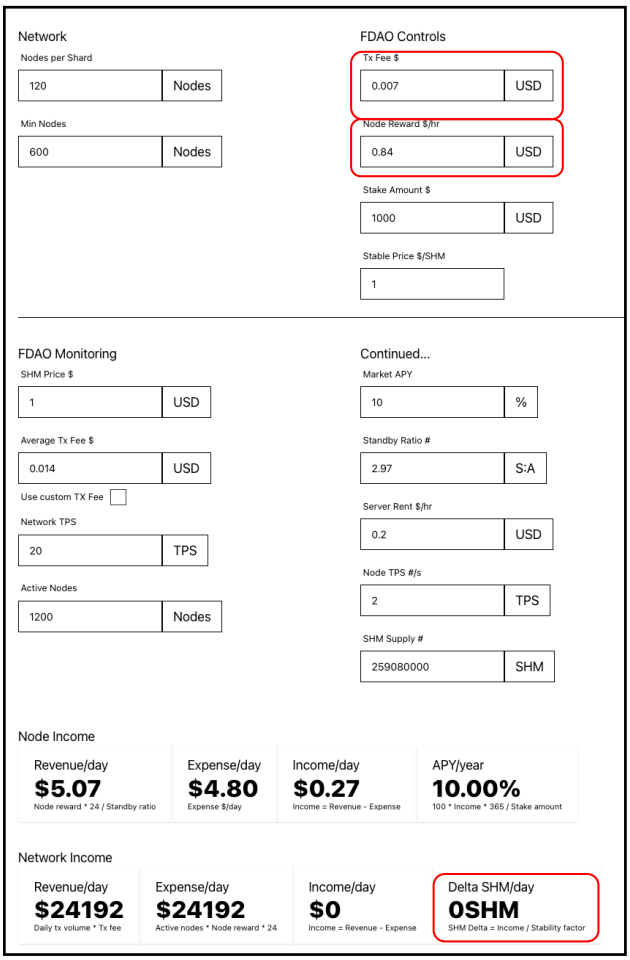
The right figure below is another example of how the FDAO-overseen variables can be changed to create a state of supply equilibrium. In this example, the tx fee $ and node reward $/hr variables are used to balance the network revenues and expenses (run more parameter scenarios here).

Supply Inflation



Network Parameters Inflation

Supply Equilibrium



Network Parameters Equilibrium 2

3.6.3.4 Model Properties

In a previous section, we discussed how no pre-existing issuance model satisfied all the ideal criteria required to guarantee the network's secure, efficient and smooth operation.

In this section, we will discuss how using a capped elastic supply model will:

1. **Ensure running a validator node on Shardeum is profitable, providing reasonable hardware costs**

If running a validator node becomes unprofitable, the S:A ratio will begin to reduce as some node operators unstake and leave the network; this reduction in S:A ratio will mean the remaining nodes will spend less time in standby and more time earning rewards for actively validating the network. If the S:A ratio reduction is too extreme, the node reward $/hr parameter can be increased to make the network more profitable and ensure an adequate S:A ratio.

2. **Ensure the SHM supply is always heading toward equilibrium (SHM burned = SHM issued), enabling the network to reward node operators forever**

We can not know demand, usage and adoption in advance; having a capped elastic supply model allows us to oversee and respond to these unpredictable factors optimally and not contribute to a misalignment of incentives, which need to be sufficiently adjustable as would be the case with other models.

The adjustable network parameters, specifically tx fee $ and node reward $/hr, can push the supply into an inflationary or deflationary state; having the ability to control this means the supply can be encouraged to find equilibrium (SHM burned = SHM issued); maximum supply (508 million) should never be reached and the network will always have SHM to issue to incentivize node operators.

3. **Keep the S:A ratio high enough that the network always retains its ability to scale and never sacrifices security**

The S:A ratio can be raised by increasing the node reward $/hr parameter. The FDAO will actively monitor the S:A ratio; if it drops low enough that security or the network's ability to scale is compromised, the increase in node reward $/hr will make the network more profitable (able to sustain more nodes at a given APY%), attracting more node operators to the network and therefore increasing the S:A ratio.

4. **Keep the S:A ratio low enough so that the network never issues more SHM than required to secure the network and avoid the pointless waste of natural and computational resources**

The S:A ratio can be lowered by decreasing the node reward $/hr parameter. The FDAO will actively monitor the S:A ratio; it is deemed inefficient if it reaches a level that no longer benefits the network from a security or scalability perspective; the decrease in node reward $/hr will make the network less profitable (unable to sustain the current number of nodes at a given APY%), resulting in nodes unstaking, leaving the network and reducing the S:A ratio.

5. **Actively promote network efficiency by rewarding the validator node operators enough APY% to stay in the network but no more**

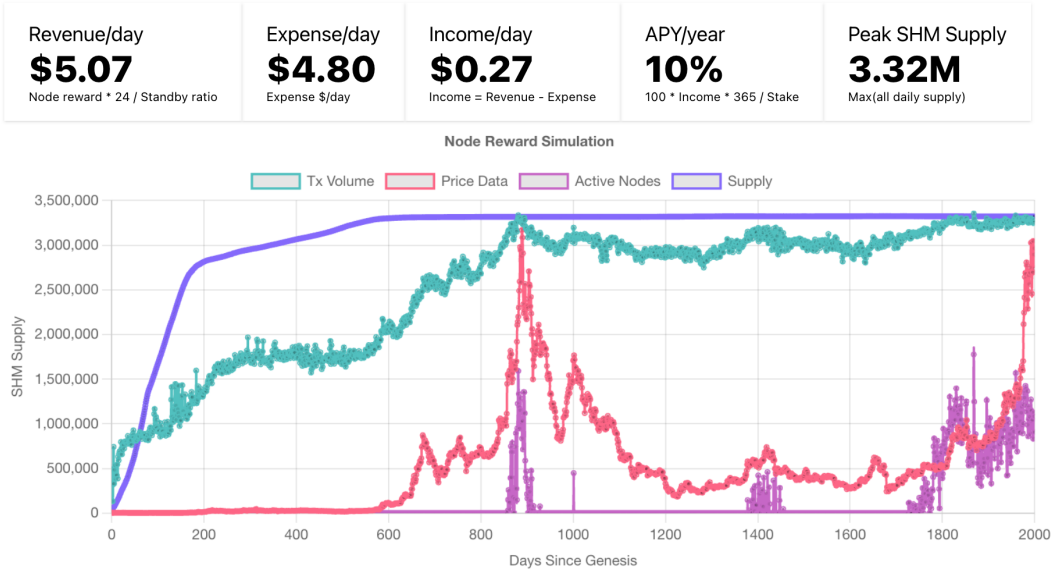
We cannot predict in advance what is "enough" APY% for node operators to continue validating the network; the micro, macro and socio-economic climate all play a significant role in how well entire asset classes perform. The S:A ratio is used as a collective representation of the current economic environment to allow us to make informed decisions on the future profitability of the network.

In the event that the S:A ratio is too high, the node reward $/hr parameter would be lowered to reduce profitability. In these circumstances, the most inefficient nodes will likely leave the network first, as they will become unprofitable sooner, resulting in a consistently efficient network.

3.6.3.5 Dynamic Simulations

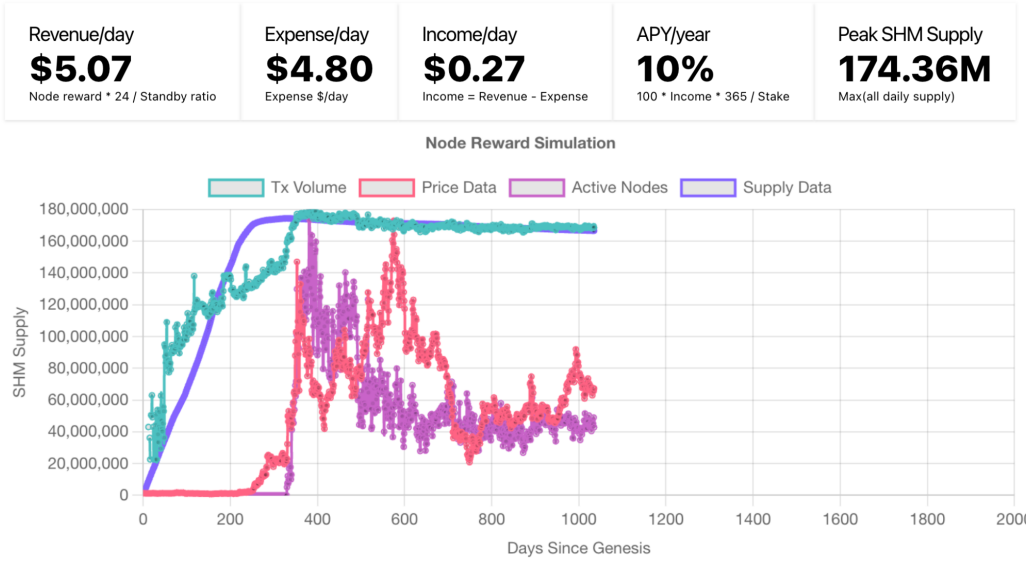
We extensively tested our model using the price action and transaction volumes of other well-known Layer 1's. The model finds equilibrium (SHM burned = SHM issued) naturally or with minor adjustments to the FDAO-overseen parameters in every circumstance.

The figure below shows how Shardeum's capped elastic supply model would react following Ethereum's price action and transaction volumes. In the first 180 days, the supply is highly inflationary as transaction volumes are low and the network still needs to reward the validator nodes (600 nodes minimum). Between 180 and 600 days since Genesis, the network transactions continue to increase; this slows supply inflation and finally results in the supply reaching equilibrium from 600 to 2000 days.

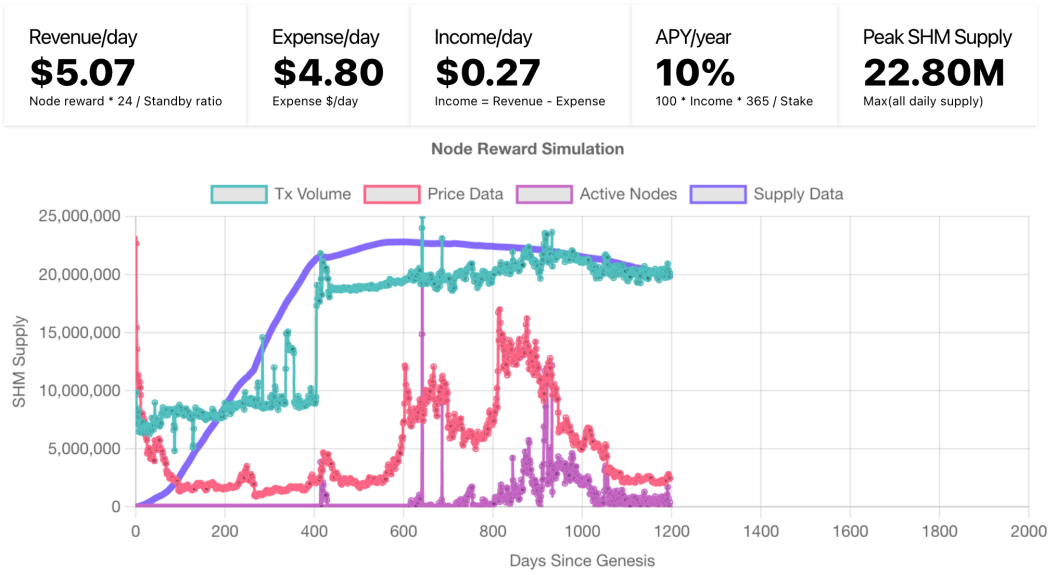


Ethereum Price vs Ethereum Transactions with capped elastic supply model

Natural equilibrium occurred in almost all our simulations. The figures below are based on the Polygon and Algorand data; both follow the same pattern as the Ethereum simulation above, achieving equilibrium as the network matures.

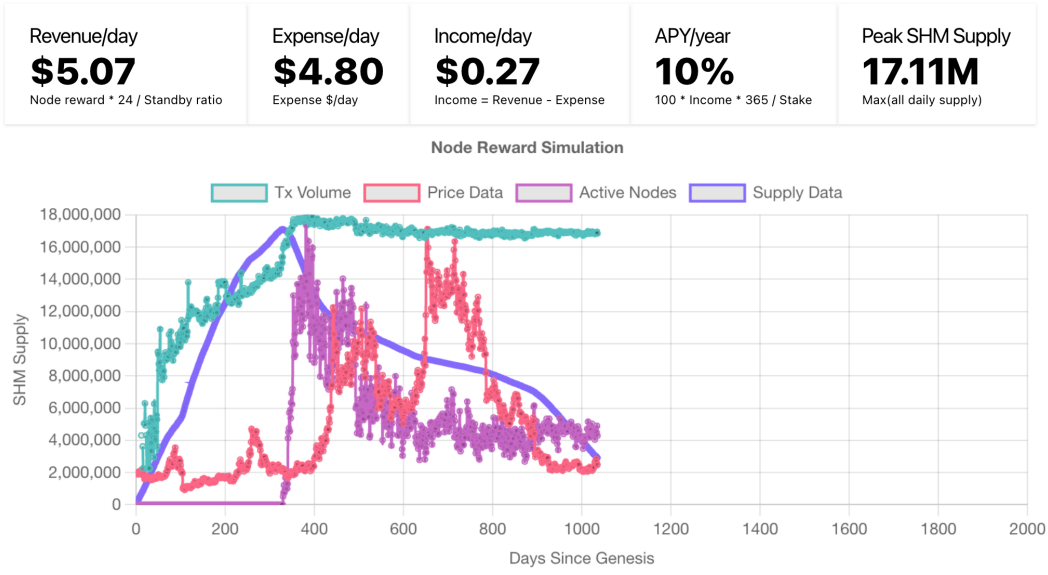


Polygon Price vs Polygon Transactions with capped elastic supply model



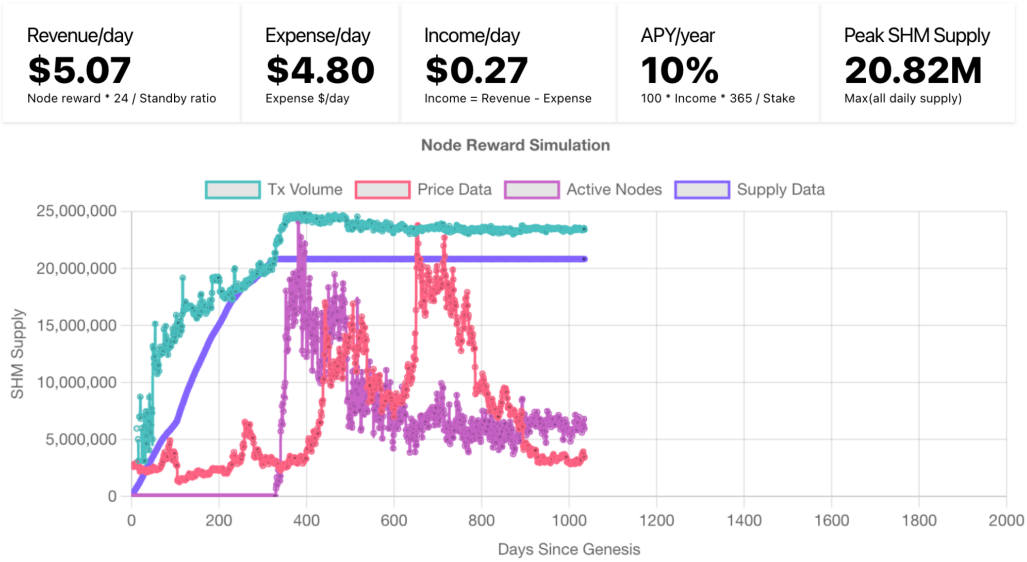
Algorand Price vs Algorand Transactions with capped elastic supply model

Occasionally, when we mixed the price and transaction volumes of different networks, we found examples where the supply did not naturally find equilibrium. The FDAO-overseen variables were tweaked in these scenarios to encourage the supply toward equilibrium. The figure below combines price action from the Algorand network and the transaction volumes from Polygon; in this model, the supply initially starts inflationary before quickly becoming highly deflationary.



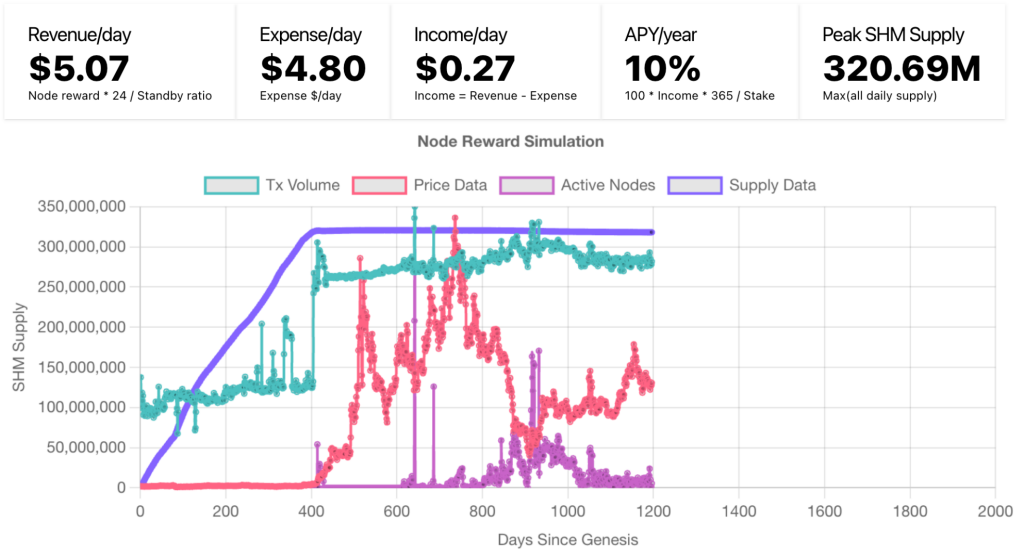
Algorand Price vs Polygon Transactions with capped elastic supply model

The following simulation is the same as above with the FDAO-overseen variable **Node Reward $/hr** changed from $1 to $1.2; this demonstrates how minor changes to these variables can steer the network toward equilibrium.



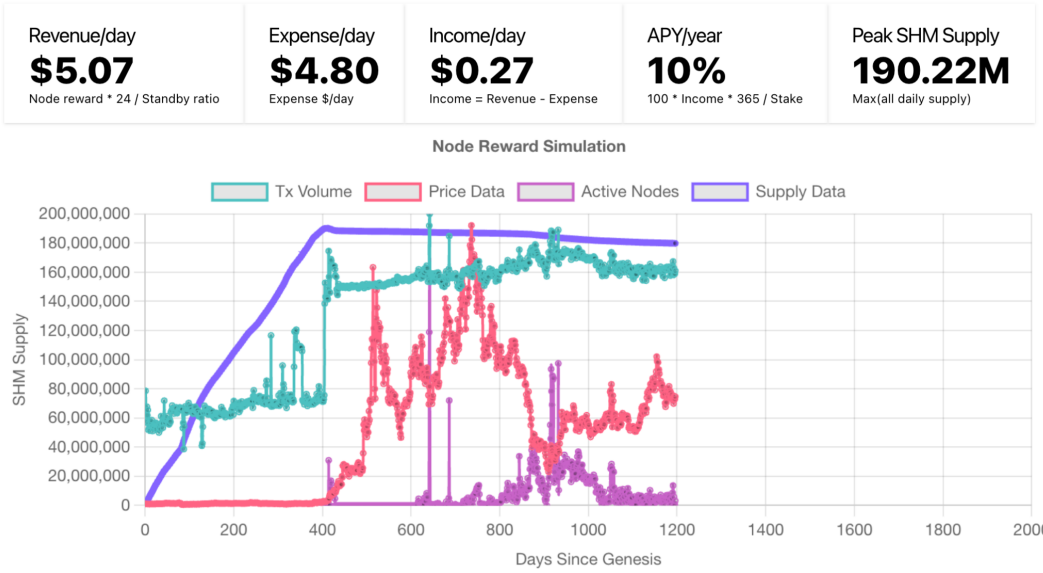
Algorand Price vs Polygon Transactions with capped elastic supply model (updated node reward per hour $)

The following figure combines Polygon price action with Algorand transaction volumes. In this model, the supply still finds equilibrium but at a level that exceeds the node reward max supply (259 million SHM).



Polygon Price vs Algorand Transactions with capped elastic supply model

The above model can be adjusted to find equilibrium at a lower supply level (figure below) by reducing the FDAO-overseen **node reward $/hr** from $1 to $0.6.



Polygon Price vs Algorand Transactions with capped elastic supply model (updated node reward per hour $)

The above simulations prove that no matter how Shardeum's price and transaction levels materialize, the issuance policy can ensure the SHM supply is always heading toward equilibrium (SHM burned = SHM issued), enabling the network to reward node operators forever (run more simulations here).

4 Ecosystem

The Shardeum project intends to incentivize ecosystem projects to build innovative new applications on the platform as well as port popular applications that have been built on other platforms. Ecosystem incentives will also be provided to develop various supporting infrastructure, software and services needed for the core network. For example, independent decentralized bridges to allow transfer of assets between Shardeum and other networks. Aside from the core validators and archivers, there are many services that are needed in the ecosystem for distributing data and making it available to users.

4.1 DApps

Decentralized Applications, commonly known as dApps, are applications that operate on smart contract platforms, eliminating the need for centralized intermediaries while providing transparency, security and immutability. In the Shardeum ecosystem, we encourage and incentivize the development of dApps that are optimized for, or can benefit from, parallel transaction processing. This is particularly beneficial for dApps with high transaction volumes or those that need to execute complex computations that can be broken down into smaller concurrent tasks.

Transactions are processed in a first come first served (FCFS) manner; this approach is beneficial as it provides predictability, fairness, prevents higher fees via bidding, offers MEV resistance and opens up a pathway for new types of dApps. For instance:

- Auction Platforms: In an auction, bids should be processed in the order they're made. FCFS ensures that if two participants place a bid at nearly the same time, the one who bid first gets precedence
- Ticketing Systems: For events with limited seats or special editions of items, a FCFS system ensures that early birds get the tickets or items without facing potential delays or out-of-order processing
- Queue-based Service Platforms: Any dApp where users queue for a service (like a virtual waiting room) would benefit from FCFS. This ensures fairness and reduces potential disputes
- Gaming: Certain online games, especially those where players compete for limited in-game resources, could use FCFS to determine the order of acquiring these resources

- Decentralized Domain Registration: As new domains or domain extensions become available, FCFS processing can fairly determine who gets a specific domain name if multiple parties are interested

Additionally, we are committed to promoting dApps that are predicated on the usage of low transaction fees, opening the door for innovative use cases previously rendered impractical on other platforms due to prohibitive costs. Such dApps include, but are not limited to:

- Loyalty point systems for merchants
- Coupon systems for product businesses
- Voting systems for small communities
- Crowdfunding platforms akin to Kickstarter
- Automated payment solutions
- Membership services reminiscent of Patreon
- Algorithmic stablecoins
- Low investment games such as lotteries

The rationale for championing this direction is multifaceted. Our incentivization of dApps reliant on low transaction fees is grounded in our commitment to ushering in a broader array of applications that could revolutionize entire existing industries and even spawn entirely new ones. Furthermore, for the first time ever, dApps can scale to levels commonly found in popular web2 services without reaching bottlenecks. Additionally, all dApps within Shardeum maintain Ethereum compatibility, offering a dual advantage: the vast developer community familiar with Ethereum can easily transition or port their projects to Shardeum while users benefit from the widespread familiarity and trust associated with Ethereum-based applications.

For the Shardeum community, these strategies herald a future of increased innovation, interoperability and inclusivity. The Ethereum compatibility not only ensures the seamless portability of dApps between platforms but also amplifies the potential for Shardeum to integrate with the broader ecosystem, enriching the user experience and fostering a cohesive, interconnected community.

4.2 Relayers

In the Shardeum infrastructure, relayers serve as an integral part of the data propagation mechanism. Once archivers have aggregated the essential data from validators, it is the responsibility of relayers to distribute this consolidated data to various services that require it, such as blockchain explorers, RPCs, exchanges and more. To enhance the efficiency of this data distribution process, there may be hierarchical tiers of relayers, designed to fan out network data extensively across multiple service providers. As the complexity and demand of the Shardeum network evolve, both archivers and relayers are envisaged to vertically scale, ensuring that the pace of data dissemination is always in step with the network's growth.

Contrary to some conventional models, Shardeum's relayers are not directly compensated by the network. This stems from the understanding that these relayers will predominantly be operated by the very services that necessitate the data, such as the explorers. Instead, to foster a vibrant ecosystem, these services are financially supported as ecosystem projects. This model lays the foundation for a novel economic dynamic: certain service providers, instead of operating their own relayers, might opt to subscribe and receive data from relayers managed by third parties.

For the Shardeum network and its community, the proposed approach presents several benefits. Firstly, by decoupling network compensation from relayer operation, it encourages service providers to optimize their infrastructure based on their specific needs, potentially leading to more efficient data distribution pathways. Additionally, the potential emergence of an economic ecosystem for data distribution fosters innovation and

diversification in service offerings. This not only enhances the robustness and adaptability of the Shardeum network but also cultivates a community where collaborative synergies are central.

4.3 Connectors

Shardeum uses Remote Procedure Call (RPC) nodes similar to Ethereum. On Shardeum, a RPC node, also called a connector node, acts as a bridge, enabling external clients such as wallets to communicate with the Shardeum networks. By offering an interface to send requests and receive responses, connector nodes allow decentralized applications to access, interact, request, query or retrieve data on distributed ledgers. For example, a client such as Metamask could submit a transaction to the network, which is first passed to the connector node and then forwarded to the active nodes for consensus and validation.

For Shardeum, integrating connector nodes is paramount. Firstly, they provide a streamlined connection point for external applications and services to interact with the Shardeum network, ensuring that these interactions remain efficient and seamless. This is especially vital for facilitating real-time data access, transaction broadcasting and smart contract execution.

Secondly, by acting as intermediaries, connector nodes alleviate the need for every application to run a full node, thereby reducing the overhead and fostering a more scalable ecosystem. Lastly, in a decentralized landscape like Shardeum, connector nodes enhance redundancy. Multiple connector nodes can be deployed across the network, ensuring that even if one faces downtime, applications can reroute their requests for uninterrupted service. In essence, connector nodes fortify Shardeum's infrastructure, promote scalability and ensure consistent and reliable network access for all integrated applications.

Shardeum also encourages third parties to operate connector nodes and will incentivize such operators through the ecosystem fund during an incubation period.

4.4 Explorers

In the realm of decentralized platforms, Shardeum recognizes the significance of building upon proven, user-familiar foundations. Our approach revolves around enhancing Layer 1 scalability and decentralization, while deliberately retaining well-established elements of the ecosystem. This means that instead of reimagining the smart contracting language, the virtual machine, or the explorer, we seek optimizations in areas that genuinely require innovation. Such a strategy is central to achieving swift adoption by both developers and users.

In line with this philosophy, the Shardeum Explorer was conceived with both user-centric and developer-centric design principles. Drawing inspiration from Etherscan, we've ensured the Shardeum Explorer offers a user experience that feels familiar to SHM users and developers previously acquainted with Ethereum. This intentional design continuity not only streamlines the transition for those migrating from Ethereum but also underscores our commitment to enhancing user experience without unnecessary alterations.

Shardeum also encourages third parties to develop Explorers and will incentivize such development through the ecosystem fund during an incubation period.

4.5 Oracles

We believe oracles are an indispensable component of the web3 technology stack and therefore are essential for Shardeum. We will use decentralized oracles networks (DONs) which will allow on-chain smart contracts

to be empowered by off-chain data such as price feeds, data feeds and hybrid smart contracts. DONs are necessary for Shardeum as they provide cryptographic truth guarantees which are both tamper-proof and immutable; unlocking the next generation of advanced decentralized applications. Here's some of the potential use cases unlocked by DONs interacting with Shardeum:

- Any stablecoins require oracles for accurate price pegging
- AMMs, DEXs and other DeFi protocols require accurate data feeds of token pairs
- Collateralization and loans also require accurate data feeds of prices to prevent users getting prematurely and unfairly liquidated
- Prediction markets and futures require oracles to provide authentic off-chain data for proof of outside events
- Mirroring of financial assets and instruments require accurate data feeds of the assets
- Commodity prices
- Election results
- Proof of events

Furthermore, beyond price feeds, data feeds and hybrid smart contracts, DONs facilitate other forms of smart contract applications using a Verifiable Random Function (VRF) which generates random numbers and can be used in other smart contracts for lottery or gaming applications.

Consequently, DONs will broaden the sphere of possible applications on Shardeum and enhance security through authentication of off-chain data, mitigating the risks of inaccurate data and any potential arbitrage opportunities that could be exploited if a centralized data feed were used. Ultimately, DONs on Shardeum expand the utility of smart contracts, facilitating the creation of advanced hybrid decentralized applications that are turbo-charged with real-world data sources that are both immutable and verified.

4.6 Bridges

Lack of interoperability between disparate networks has been a much remarked upon problem historically but recent solutions in the form of bridges, trust-minimized communication protocols such as the Interblockchain Communications Protocol (IBC) and cross-consensus messaging formats such as (XCMP) have all facilitated interoperability between heterogeneous blockchains.

Shardeum will benefit from the interoperability solutions that have already been developed. It shares the sentiment espoused by other networks that heterogeneous chains should be able to transact and interact and thus that interoperability between networks is a necessary feature.

Initially, Shardeum will accomplish interoperability through integration with multi-chain protocols and cross-chain bridges such as Axelar and Layer 0. Bridges facilitate cross-chain transactions and asset flows between disparate Layer 0, Layer 1 and Layer 2 networks. This bestows numerous benefits upon Shardeum, as well as other networks with which it interoperates, including:

- Interoperability between key Layer 0, Layer 1 and Layer 2 ecosystems
- Increased SHM utility and functionality as SHM can now be integrated into DEXs, AMMs, liquidity, treasury and other types of dApps and DeFi instruments
- Increased utility for external assets from Layer 0s, Layer 1s and Layer 2s entering the Shardeum ecosystem, enabling their integration with DEXs, AMMs, liquidity pools and other DeFi platforms on Shardeum
- Cross-chain transactions
- Cross-chain liquidity pools for DEXs, AMMs, liquidity pools and other types of DeFi primitives
- Cross-chain DAO treasuries
- Expanding Shardeum's dApps and SHM use cases

- Inherently larger user base for SHM and dApps on SHM
- Opening up multi-market liquidity opportunities on different networks
- Unlocking multi-chain future and streamlining the creation of multi-chain dApps between disparate networks

Besides Bridges, a crucial technological component is our smart contract interoperability facilitated by the EVM. This will allow existing dApps that enable synthetic and atomic swaps to be deployed on the Shardeum network, thereby reducing the need to use a bridge.

4.7 Wallets

Cryptographic wallets serve as digital interfaces that facilitate the generation, storage and management of cryptographic keys, enabling users to interact with DLTs. These keys represent a user's digital identity and ownership of assets on a network. At Shardeum, while we recognise the importance of wallets in the web3 space, we have strategically chosen not to develop our own proprietary wallet solution. Instead, we prioritize ensuring that our network is accessible via RPCs and thus making it compatible with a wide range of existing Ethereum wallet providers.

There are several critical reasons underpinning our decision. By choosing not to create a Shardeum-specific wallet, we avoid misallocating our efforts and instead can direct our resources toward mission critical areas, like enhancing the core infrastructure to optimize performance, scalability, and security. Furthermore, this approach promotes the seamless integration of Shardeum with renowned and widely adopted wallets such as Metamask. Due to the native Ethereum compatibility of Shardeum, users can leverage their favorite and most familiar wallets when interacting with our network. This level of compatibility not only lowers the barrier of entry for new users but also reduces fragmentation in the user experience.

For the Shardeum community, the implications are profound. By ensuring compatibility with well-established Ethereum wallets, users benefit from the reliability and security that the best Ethereum compatible wallets have to offer. Additionally, this compatibility reinforces the ease of transition and continuity of experience for prior Ethereum users and developers looking to explore Shardeum.

4.8 Shardus

The Shardeum project will continue to support the development of the Shardus protocol. Even after mainnet launch it is expected that upgrades and enhancements to the Shardus protocol layer will be needed. Shardeum views Shardus as a critical component of the ecosystem and will continue to support it in the future. Shardeum will also be obtaining a license to the Shardus software as mentioned in the Tokenomics section. The founders and several members of the team hold Shardus tokens.

5 Community

At the heart of every decentralized project lies its community — an intricate tapestry of developers, users, stakeholders and enthusiasts who breathe life into the protocol. The essence of decentralization, after all, is not in the mere absence of centralized control, but in the empowerment and participation of the many. Our project's vitality, innovation and resilience are directly attributable to this vibrant collective.

5.1 OCC Principle

Shardeum is being built on the principles of Open, Collaborative and Community-driven (OCC) approach. Shardeum will be an open, collaborative and community-driven project by taking an approach that's not

been tried by other projects until now. The project will maintain openness in discussions, sharing information, ideas, documents and more. Shardeum believes that cooperation among diverse projects will be necessary to enable the global adoption of decentralized applications. That means collaborating within the ecosystem to find ways to help every individual or group building on top of Shardeum. It will collaborate with anyone who is trying to achieve a similar goal of increasing decentralization. Shardeum's core belief is that community is supreme and to involve the community in key decisions. But, decisions are just part of being community-driven. An even more important aspect is to reduce the delta of project information. Whenever there is any new information to be shared we'll share that with the community as early as possible.

5.2 Governance

The practicalities of developing, launching and growing a network often mean that some level of centralization is present in the early stages. The challenge for any decentralized project is to effectively transition from this initial centralized phase to a genuinely decentralized state as it matures. During this transition, it is imperative for the network to establish clear mechanisms and frameworks that promote decentralization. Balancing the efficiency and direction provided by centralized leadership with the broader community's desires will be critical. We propose a plan on how this transition can be done smoothly as well as some of the mechanisms for how voting will work. We are open to input from the community.

5.2.1 Foundation

A Swiss based Foundation will support the early development of Shardeum. The law firm assisting with the Foundation setup is highly experienced in the cryptocurrency space as they were involved with the Foundation setup of many other prominent crypto projects including Ethereum and Cardano. The long term goal is to transition Shardeum from the Foundation and team into a DAO composed of community members.

The founding members of Shardeum are Nischal Shetty and Omar Syed. Shardeum team members, also known as committers, are distributed across the US, India and Myanmar.

- Nischal Shetty - is the Founder and CEO of WazirX, India's largest crypto exchange with over 10 million users. Nischal is a highly regarded entrepreneur with over a decade of experience building and scaling global products out of India. A software engineer by education, Nischal has also founded Crowdfire, a social media management web service with over 20 million users in the past. Nischal's previous successes have landed him on the Forbes '30 under 30' list. A passionate blockchain evangelist, Nischal has been active in the Indian crypto space since its inception. Nischal's mission has been to make crypto accessible to every Indian; he's also been advocating positive crypto regulation in India with his Twitter campaign #IndiaWantsCrypto for over 1000 days.
- Omar Syed - is a systems architect who organized the Shardus project in 2017 to build a linearly scalable distributed ledger. Over the past three decades, Omar has been involved with helping large organizations such as NASA, Yahoo and Zynga build scalable, fault-tolerant, distributed systems. Omar holds a B.S. and M.S. from Case Western Reserve University with specialization in Artificial Intelligence. Omar was involved with several start-ups including the first matrimonial website and the first stock sentiment analysis website. Omar along with his son Aamir invented the strategy board game, Arimaa and offered the Arimaa Challenge Prize to promote breakthrough research in AI. Omar's long-term vision is a world where everyone receives an unconditional universal basic income based on a stable cryptocurrency so that poverty and hunger are eliminated.

5.2.2 DAO

A Decentralized Autonomous Organization (DAO) represents the embodiment of Shardeum's commitment to decentralized governance. The DAO will act as an open, community-driven governing body where decisions are made collectively. Every stakeholder, regardless of the size of their contribution, will have a voice.

Structure: The DAO will consist of proposals and voting mechanisms. Stakeholders can propose changes or projects and after a certain vetting period, the proposals will be opened for voting.

Funding: A treasury will be established, ensuring the DAO is self-funded and can support various projects, initiatives and rewards. Periodic endowments from the network's treasury can maintain the approved projects.

Voting Mechanism: Stakeholders' voting power will be proportional to their stake or other metrics decided by the community, ensuring a balanced representation of interests. Additionally, a time-locked voting process can be established to prevent rapid, unconsidered decisions and encourage long-term planning.

Transparency: All voting decisions and financial transactions will be recorded on the Shardeum blockchain, ensuring transparency and accountability.

5.2.3 Transition

Transitioning from a centralized Foundation to a DAO is a critical phase for Shardeum, necessitating a robust and innovative strategy. To ensure a smooth transition we use a mechanism that will gradually increase the size of the DAO and move it towards greater decentralization.

Voting Eligibility: In order to participate in the DAO an entity must be able to lock up a specified amount of SHM. Only entities that can lock up the required amount or more will be able to participate in the voting.

Initial High Threshold: At the inception of the DAO, the SHM lock up threshold will be set deliberately high; a level that only the Foundation can attain. This ensures that, in the DAO's early stages, the Foundation can guide decisions and maintain the network's stability.

Decentralization via Threshold Reduction: As Shardeum progresses and the community expands, proposals will be introduced to reduce the lock up threshold. The very act of voting on these threshold values will itself be governed by the SHM lock up mechanism. Over time, as these thresholds are voted down, a broader spectrum of stakeholders will gain influence, making the DAO more decentralized.

Staged Empowerment: This method ensures a staged empowerment of the community. As the threshold reduces, a more significant portion of the community gains the ability to influence decisions, fostering an environment of collective governance while preserving stability.

Ensuring Transition: To ensure a transition to a DAO a timeline will be established that defines a schedule for the threshold reduction.

The transition from Foundation to DAO, powered by the threshold reduction mechanism, offers a strategic path to decentralization. It balances the need for stability with the principles of democratic participation, making sure Shardeum remains resilient and community-driven.

5.2.4 Proposal Lifecycle

Shardeum's governance process will be a subset of individual processes which ensure that the created proposals are actively debated, incorporate critical feedback to extant issues, are clear and well-defined and most importantly of high quality. Shardeum's proposals will likely be hardened by rules and processes in order to raise the quality of the proposal and strengthen the proposal's case and merit before it is ratified or rejected. Subsequently, this will also minimize the likelihood that spam or poor quality proposals are raised to the community. The proposals process is as follows:

Proposal Creation: This is the genesis of a new idea. A community member or group identifies a gap or an improvement area and drafts a preliminary proposal highlighting the problem, the proposed solution and the potential impact.

Discussion and Feedback: Once the proposal is shared, it enters a phase of active debate and scrutiny by the wider community. This period allows stakeholders to question, critique and suggest modifications, ensuring diverse perspectives are considered. The entire community (including the Shardeum team and Foundation) engage in discussions and provide critical feedback on the merits and shortcomings of the rudimentary proposals.

Feedback Incorporation: Based on the community's feedback, the proposer revisits the initial draft. Necessary changes, refinements and clarifications are made to address concerns and suggestions. This stage can be iterative where the feedback is used to update the proposal and discuss it again.

Proposal Formalization: The refined proposal is then formalized, ensuring it aligns with Shardeum's objectives, is coherent and has a well-defined scope. If a proposal makes it to this stage, it should be of high quality and well formulated. Some proposals may not make it to this stage.

Voting: The formalized proposal is then opened up for community voting. Every stakeholder gets to express their support or opposition based on the token lock up mechanism.

Approval/Rejection: Once voting concludes, the proposal is either approved, if it secures the requisite support, or rejected. This decision reflects the collective voice of the community.

Implementation: For approved proposals, this phase involves putting the plan into action. It includes technical development, changes to protocols, or any other actions required to bring the proposal to life.

By meticulously following this lifecycle, Shardeum ensures that proposals are hardened by a structured process. This elevates the quality of suggestions and minimizes disruptions from spam or ill-conceived ideas, ensuring the platform's continual evolution is in line with its community's best interests.

5.2.5 Proposal Types

Shardeum will have proposal tags for certain types of proposals such as: Text Proposals, Funding Proposals, Parameter Change Proposals and Tokenomics Proposals. There will undoubtedly be more proposal tags besides the aforementioned which accurately categorize proposals so that community members and voters can immediately understand which type of proposal is being put forward. There will also be proposal templates which expedite and streamline the proposals creation process allowing users to create proposals via tweaking and editing templates, rather than the creation of complete new proposals from scratch.

Additionally, certain proposals will have predefined structures which prevent the gaming of certain elements of the governance process. For example: funding proposals in which a project or group of projects asks for a

SHM grant will only receive their grant conditionally if it's divided into quarters or months after key project milestones, deliverables and deadlines have been met in a timely fashion. If not, additional SHM will not be given.

5.2.6 Voting Policies

Based on observations and lessons learned from other DAOs, we continually strive to create a robust, fair and efficient governance model for our platform. Our commitment is to ensure that every mechanism we adopt aligns with the best interests of our community and the broader ecosystem whilst simultaneously maintaining Shardeum's security. Currently, we are considering several voting mechanisms to achieve this aim:

- Burning Mechanism: Rather than one vote per person, the more SHM an account votes with the more the vote counts. However, the SHM used to vote are burned. This ensures that only those who are genuinely interested in the proposal will vote. Also those who feel more strongly about the proposal can vote with more SHM. Parties who disagree are more likely to discuss and compromise rather than competing during the voting process and burning SHM. Someone with a large SHM balance will not be able to continue using that to drive every vote in their favor since the votes will burn the SHM and reduce the balance.
- Quadratic Voting: In addition to burning, having a mechanism where the cost of additional votes increases quadratically would allow for a more weighted representation based on the number of accounts and help mitigate the disproportionate influence of "whales" and large token holders found with 1 token 1 vote systems. However, a whale could still distribute the SHM across many accounts and subvert such a system.
- Account Reputation-Based Voting: If a form of quadratic voting is used then it would also help to introduce a mechanism to limit which accounts can vote based on not only locking up the SHM specified by the threshold, but also based on an individual account's reputation within the ecosystem. This will help mitigate a Sybil attack on quadratic voting and ensure that long-standing and potentially more invested members have a say in governance matters. Metrics such as the age of the account, liveliness of the account, etc., can be used to gauge the reputation of the account.

We welcome any constructive feedback from the community. Your insights, experiences and perspectives are invaluable as we finalize a governance model that retains security whilst championing transparency, fairness and the collective good. We pledge to keep our community informed and involved as these deliberations progress further.

5.3 Growth

The Shardeum project has been actively growing the community through grassroots efforts to increase the awareness of the project. The initial focus has been in India and is now branching out to other countries.

5.3.1 Social Media

Since its inception, Shardeum has witnessed an exponential growth trajectory, a testament to its evolving influence and presence within the web3 ecosystem. Here's a recent snapshot of our burgeoning community across various social media platforms:

- **Twitter**: 216,949 followers
- **Youtube**: 6,790 followers
- **Reddit**: 1,200 members
- **Telegram**: 10,539 members
- **Discord**: 535,715 members

Shardeum has emerged as one of the fastest-growing Layer 1 communities with over 800,000 members. At its peak, Shardeum Sphinx betanet boasted almost 40,000 community-run validators and over 75,000 validators have participated so far, making it one of the largest permissionless testnets in existence. Additionally there have been almost 11,000 dApps deployed; making Shardeum one of the most participated web3 networks in existence. The project benefits from the contributions of hundreds from over 45 countries, in roles ranging from content creation to business development and community moderation. More than 56 decentralized Shardeum communities across nations such as India, Vietnam, Turkey, Japan and Nigeria are owned and managed by enthusiasts.

To date, Shardeum has held over 250 Proof of Community events, attracting more than 15,000 participants in upwards of 90 cities across 8 countries, as it works to familiarize the masses with web3. Having primarily focused on Asia, Shardeum is now broadening its horizons with an emphasis on the USA, participating in key events like Stanford Blockchain Week and Messari Mainnet.

Looking forward, after the mainnet launch, Shardeum will persist in its global expansion and community engagement endeavors. The project plans to keep hosting events and meetups worldwide to educate and involve future users and developers. Shardeum aspires to be the premier platform for web3 development and adoption.

5.3.2 Proof of Community

Shardeum is a community-oriented, community-directed and community-driven project. In other words, the community is supreme. Shardeum has spearheaded a plethora of global community initiatives collectively known as Proof of Community (PoC) which includes online events, workshops, meetups, educational seminars and various other forms of physical and digital outreach. This is to educate and raise awareness about web3 and Shardeum as we recognize that education is the bedrock of wider web3 adoption and therefore it is imperative to raise awareness, inspire and mentor the wider community. Community members can attend their local PoC events, engage in learning, networking, mentorship, participate in bounties and wider group collaboration in order to accelerate their web3 journey.

PoC is designed to be tailored to both a web2 and web3 audience to accelerate adoption and facilitate the required education so that individuals can go deeper into web3. Our first PoC event was on 26th August 2022, in Bangalore, India, since then, we have accomplished the following:

- 250 events
- 8 countries
- 96 cities
- 15,000+ participants
- 112+ meetups
- 31+ workshops
- 103+ campus workshops
- 474 speakers
- 383 contributors
- 4.66/5 Avg rating

With a community-oriented focus, Shardeum aims to empower individuals from all across the globe to contribute to the network in various capacities including but not limited to organizing PoC events, content creation, business development, community management and more. There are over 57 self-run Shardeum communities on Twitter and Telegram created and managed by contributors from several countries including India, Turkey, Japan, Russia, Ukraine, Vietnam, Nigeria and more.

Shardeum also holds a special place for both English and non-English content creators across the world through its 'Shardeum is Borderless' program. From February to September 2023, the program received an impressive collection of over 750+ guest blogs and videos, representing contributors from 45+ countries and spanning 38 languages. The five leading languages for the guest contents so far are English, Bahasa/Indonesian, Russian, Chinese and Korean.

5.3.3 Future Growth

To foster Shardeum's growth momentum, we've envisioned strategic initiatives aimed at bolstering adoption and community engagement. First, we aim to enhance our developer outreach program, hosting hackathons and webinars to tap into emerging talent and showcase the platform's potential. Collaborative partnerships with leading educational institutions and web3 education platforms will help expand the understanding of Shardeum, its technologies and its benefits, fostering a new wave of enthusiasts and developers. Integral to our growth strategy is the cultivation of robust partnerships with Layer 0, Layer 1 and Layer 2 entities. This facilitates ecosystem expansion and interoperability, fostering a conducive environment for cross-chain collaborations and synergies. Additionally, we'll be launching targeted marketing campaigns to showcase Shardeum's unique features and benefits to a wider audience.

Such proactive steps are not just growth strategies; they are instrumental in driving real-world adoption, ensuring that Shardeum remains at the forefront of web3 innovation. Through these initiatives, we intend to not only grow our user base but also cement Shardeum's position as a thought leader, a nexus of innovation and a community-driven platform in the decentralized ecosystem.

6 Roadmap

In less than two years since inception, the Shardeum project has made significant strides in creating a brand new Layer 1 platform from scratch. While many projects start with a codebase of an already existing project and launch quickly with small incremental changes, the Shardeum project is taking the more difficult path of building a new code base from first principles. We present here a brief timeline of the project, its current milestones and future directions.

6.1 Timeline

Although the Shardeum project is only about two years old, the core technology used at the protocol layer has been in development since 2016.

6.1.1 Shardus Started Q2 2016

In April 2016, Omar Syed began working on the design of a distributed ledger that would process transactions independently without grouping them into blocks so that the ledger could more easily be sharded. Syed had come across bitcoin in 2011 while working on an economic paper titled "Sound Money Without Commodities". He was impressed by the decentralized nature of bitcoin, but having built scalable systems for large enterprises, he was concerned that the technology would not be usable for some of the global scale applications he wanted to build. By late 2017 Syed began forming a small team to work on the "Unblock Ledger" project which was later rebranded to Shardus in 2018. The first to join him was his son Aamir. In Q3 of 2021, the Shardus team demonstrated linear scaling and auto-scaling by growing a sharded coin transfer network on AWS to 1000 nodes and achieving 5000 TPS.

6.1.2 Shardeum Started Q4 2021

In October 2021, Nischal Shetty invited Syed to Dubai to discuss creating a smart contract platform based on the Shardus technology. The idea for Shardeum was born.

6.1.3 Liberty Alphanet Q1 2022

The Shardeum Liberty testnet was launched on February 2nd, 2022. The network allowed the community to deploy EVM compatible smart contracts and interact with the network using wallets and tools already available in the Ethereum ecosystem. A sharded version of the testnet with 200 nodes was demonstrated to achieve 100 TPS for token transfer transactions.

6.1.4 Sphinx Betanet Q1 2023

The Shardeum Sphinx testnet was launched on February 2nd, 2023. The network allowed community members to download and run validator nodes to join the testnet. Within 24 hours of launching the network, thousands of community nodes had joined the network and the size of the network eventually grew to over 25,000 nodes. By this time the Shardeum project had raised seed funding and started growing the team. The unexpectedly large amount of community nodes joining the network required the team to redesign the network to be even more scalable than originally planned.

6.2 Current Milestones

6.2.1 Development

The next major milestone for the Shardeum team is the launch of the mainnet. The team is working aggressively towards having a release candidate ready by January 2024 and a launch of the network in Q1 2024. A detailed roadmap tracking the projects needed for this milestone is publicly available.

6.2.2 Security

Alongside the development milestone, the Shardeum team is also working on internal and external audits of the codebase. The team is also preparing to run bug bounties on sites like Immunefi. We intend to eventually escalate the prize for reporting critical security vulnerabilities to millions of dollars.

6.2.3 Marketing

Beyond security and developmental milestones, the Shardeum team is also getting ready to see exponential growth on the marketing front. Previously, Shardeum has primarily been marketed in Asia; however, an even more global approach will be undertaken as a robust, long-term marketing strategy is crucial for the wide-scale adoption of Shardeum. Our approach is multi-faceted with more events, conferences, webinars, educational content, AMAs, feedback sessions, community engagement and digital advertising.

6.2.4 Partnerships

A major surge of partnerships and ecosystem expansion is expected to occur prior to and after the mainnet launch. The Shardeum Team will undertake strategic partnerships in both web2 and web3 in order to expand Shardeum's ecosystem and enhance its functionalities. This includes partnering with academia and enterprises, and in some cases integrating with Layer 0s, Layer 1s, Layer 2s, bridges, oracles and dApps.

6.2.5 Fundraising

The Shardeum project intends to do more fundraising in the future to grow the team as well as increase the marketing and awareness of the project. We are considering the possibility of a public SHM sale shortly before mainnet launch prior to exchange listings.

6.3 Future Directions

6.3.1 Technology Upgrades

There are various improvements planned for the Shardeum network after mainnet launch. However, none of these improvements will require rearchitecting the design of the network. Some of the planned upgrades and changes we expect in the future include:

- Non-EIP2930 transactions - In a sharded network, knowing the addresses involved in the transaction is critical for proper routing and processing of the transactions. All transactions require an EIP2930 access list. Transactions which do not will take a bit longer to process since the Shardeum connector server attempts to automatically build the access list based on static information available in the transaction. For some complex transactions this will not be possible. Such transactions will be processed by sequential execution with each shard involved in the transaction passing the state to the next shard until the transaction is completed.
- IPv6 support - Currently all nodes in the Shardeum network must be running on IPv4 addresses. Future upgrades will add support for IPv6 addresses.
- Security - The team along with white hat hackers will continue to scan the code and network policies for potential vulnerabilities and fix them as soon as they are found.
- Autoscaling on storage - Currently the Shardeum network autoscales based on transaction throughput. Future upgrades will also include the ability to autoscale based on storage demands.
- Archiver staking, reward and slashing - The archiver nodes will initially be run by the team. Future upgrades will require archivers to also stake and earn rewards as well as lose stake if slashed. Once this is implemented the community will also be able to run archiver nodes.
- Validator software in more languages - Currently the Shardeum validator is written in TypeScript using node.js. We plan to also develop the software in other languages such as Rust. The community may also develop clients in other languages.
- Upgrades to the EVM - As improvements are made to the EVM, they will be ported to the Shardeum validator.

6.3.2 Community Growth

Initially, the growth of the community will be driven by marketing efforts to increase the awareness of the Shardeum platform and what it offers. The long term growth of the Shardeum network will be based on real world use cases driving organic growth. As more dApp developers discover that Shardeum is able to maintain low transaction fees even during crypto bull markets, they will have more confidence in continuing to build on Shardeum and developing a larger user base. The atomic composability and seamless interoperability between dApps on the Shardeum network will increase the network effect and attract even more users. The low transaction fees even during bull markets will also enable new types of dApps to thrive on Shardeum as discussed in the "Ecosystem: DApps" section. Our commitment to building a platform with sustainably low transaction fees while supporting high transaction throughput will lead to a decentralized ecosystem that can attract and support a large community of users.

6.3.3 Ecosystem Growth

When the Shardeum mainnet goes live, developers will be able to port dApps they built on other EVM platforms over to Shardeum. Additionally, new dApps based on the FCFS feature and sustainably low fees on Shardeum will be developed. As existing and new types of dApps begin to be released on the Shardeum platform, more users will find the platform useful. We believe that once a critical mass of dApps and users have adopted the Shardeum platform, the network effect should help accelerate adoption and growth even further.

6.3.4 Transition to DAO

The ultimate goal of any decentralized platform is to be supported and governed by the community that uses the platform. The Shardeum team has been planning for this from the beginning and have proposed a process for a smooth transition in the “Community: Governance” section. We expect that within a few years after mainnet launch the Shardeum network will be fully transitioned to a DAO.

7 Conclusion

In the evolving landscape of smart contract platforms, Shardeum presents an innovative approach that addresses the intrinsic challenges posed by the blockchain trilemma. Through the use of a several novel innovations such as dynamic state sharding, linear scaling, autoscaling, cross shard atomic composability and Proof of Quorum and by also leveraging a sharded, blockless and EVM compatible framework, Shardeum solves the scalability trilemma between scalability, security and decentralization. Shardeum will ultimately transform the wider web3 ecosystem, ushering in an era of near unbounded scalability, unimpeachable security and unprecedented decentralization, heralding the advent of the first generation of killer applications with billions of users.