The SKALE Network- A crash course for you.

This crash course will help you gain a deep understanding of what SKALE is. Starting from the SKALE Network, we will go down to SKALE token, Protocol, validators.

This course will be delivered under the following headings-
Overview
The SKALE Network
Elastic Sidechain
Virtualized Subnodes
SKALE Admin Service
SKALE Token
SKALE Protocol

Overview

SKALE is an Ethereum- based network that provides a scalable platform for the decentralization of elastic blockchains.

It is synonymous with AWS's Platform-as-a-Service in that they both provide resources to end-users in their platform.

It's programmed to run on an undetermined number of independent nodes.

This elastic blockchain network abolishes the shortcoming associated with the traditional blockchain network.

With SKALE, you can easily develop and deploy decentralized applications(dApps) on the Ethereum network.

This cloud service comes with a complex security setup. With the latest innovation in cryptography, SKALE Network uses an almost fool-proof security outfit.

This security system is fault-tolerant and allows the network to run seamlessly even when less than one-third of its participants are corrupted.

The SKALE Network

In this section, you will gain a general understanding of the workflow in the SKALE Network. You will meet the SKALE Manager and know what SKALE Nodes are. Deeper down, you will be embraced by Elastic Sidechain and Virtualized Subnodes. And finally, we will scratch the bottom of the SKALE Network.

First off, the SKALE Network was built to solve the challenges posed by traditional blockchain networks. This includes- cost-effectiveness, configurability, scalability, latency, and so on.

SKALE offers these solutions without compromising its storage and security. The network is elastic and developed to run on the Ethereum mainnet as a sidechain so the term- Elastic Sidechain Network.

The SKALE Network is made up of the SKALE Manager and the SKALE Node.

SKALE Manager

The SKALE manager is hosted on the Ethereum mainnet. Currently, Ethereum is the only parent blockchain compatible with the SKALE network. The SKALE manager is the entry point to all smart contracts in the SKALE network.

The contract manages the following in the SKALE network- node creation and destruction, sidechain creation and destruction,

Node Creation

A node is a peer that participates in a network. To join the SKALE network as a node, the prospective node must meet minimum hardware requirements.

In order to verify that it met the standard, the prospecting node runs the SKALE daemon. Upon verification, the prospecting node will be allowed to submit a request to join the network to the SKALE manager. The request will eventually be committed to the Ethereum network.

The newly-created node will be added as a full node or as a functional node. A full node is a node that has all its resources utilized in a single sidechain. For a fractional, the resources participate in multiple sidechains.

New nodes are randomly assigned 24 peer nodes to track and record its uptime and latency. These metrics are averaged at the end of each network epoch and the value is used to determine the reward entitled to the node.

Node Destruction

This refers to a node's exit from the network. To leave the network, the node indicates and then waits for the finalization period after which it can withdraw its initial stake from the network.

If a node exits the network without waiting for the finalization period, it will be considered a dead node and its bounty will be cycled out of the chain.

You can refer to Node Creation and Node Destruction in the [SKALE Network-Whitepaper.]

Elastic Sidechain creation and destruction

End-users who want to utilize the network resources can create Elastic Sidechains. They select the desired sidechain configuration that will serve their needs and pay for the duration they will use resources in the chain. The minimum Elastic Sidechain option available contains 16 virtualized subnodes. After the selection of the desired sidechain configuration, the consumer sends a creation request to the SKALE Manager.

From here the SKALE manager receives the creation request and permits the creation of new elastic sidechains.

In the case where there are not enough resources in the network to create the desired sidechain, the transaction will be canceled and the user will be notified.

Virtualized Subnode Shuffling

This is an added optional security protocol a developer (end-user) can implement. This security measure is adopted to prevent collusion among network participants. It is facilitated by the SKALE Manager. In order to prevent consumers from determining the nodes that will be assigned to them, the SKALE network ensures that more 30% of the network resources are always available.

Elastic Sidechain Destruction

This occurs when an end-user decides to leave the network or when the user's subscription has expired.The consumer in an elastic sidechain can flag their sidechain for destruction.

Towards the exhaustion of a user's subscription, they will be notified and given the option to replenish their subscription to continue using the network resources. On exhaustion of the duration paid for, and without an additional fee from the user, the sidechain will be destroyed.

Bounty Issuance

Tokens are issued at the end of each network epoch to participating nodes in the network. The number of tokens claimable by each node depends on its performance in that epoch.

A node's performance is calculated by averaging its recorded uptime and latency value as submitted by the peer nodes. Any unclaimed token due to poor performance from the node(s) will be given to the N.O.D.E foundation.

Elastic Sidechains

The SKALE network runs on the elastic sidechain. The virtualized subnode found in the elastic sidechain proposes and commits new blocks to the blockchain. To ensure that the workflow in the creation of the new Elastic Sidechain is unhindered, the virtualized subnodes adopted some protocols

These protocols are outlined and explained below.

Messaging

The messaging protocol ensures that information is passed across between nodes. It involves- Network Security Assumption and Pending Transaction Queue.

Network Security Assumption

Network security assumption is a messaging protocol that assumes all messages sent to a virtualized subnode will eventually be delivered, however long it takes.

Pending Transaction Queue

This is another messaging protocol. It allows virtualized subnodes to send messages to its peers through a dedicated outgoing queue for each peer.

With the nodes being serviced by different threads, it ensures that the failure of a particular peer to receive a message does not affect other peers.

Read more about the Elastic Sidechain in the SKALE Network Whitepaper-Elastic Sidechains.

Consensus

A consensus is a protocol that allows virtualized subnodes to make block proposals, determine the winning block proposal and commit new blocks to the blockchain. The processes involved include the following.

Block Proposal

Block proposal is made by virtualized subnodes with the aim of committing new blocks to the chain. Once a consensus is completed and a new block successfully committed, each node's TIP_ID will increase by one.

The procedures involved are found in the SKALE Network Whitepaper-Consensus.

Data Availability

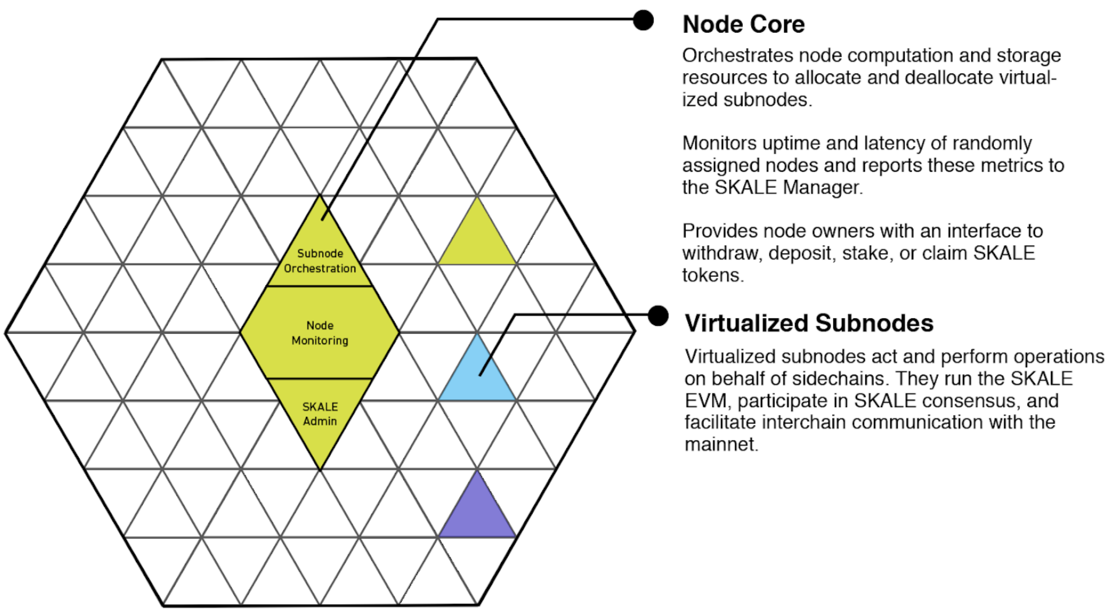
This protocol allows a virtualized subnode to communicate with other nodes about its block proposal. The data availability protocol ensures that the messages are sent to the supermajority of virtualized subnodes.

Read the full details on Consensus here.

SKALE Virtualized Subnodes

The nodes in the SKALE network are known as virtualized subnodes.

The Elastic Sidechain contains randomly selected virtualized subnodes that run the SKALE daemon and participate in consensus. The containerized virtualized subnode architecture deployed on each node in the SKALE Network allows each node to run multiple Elastic Sidechains simultaneously.



Each container offers one of the following services-

SKALE Admin Service

This service serves as the human interface between virtualized subnodes and the SKALE manager. It allows the SKALE nodes to see which virtualized subnodes they are participating in, as well as deposit, withdraw, stake, and claim tokens. There is no interface that allows virtualized subnodes to enter or leave a sidechain. Instead they are selected randomly and assigned to Elastic Sidechains.

Node Monitoring Service

Each node in the network monitors all other nodes and tracks their performance. The average performance at the end of the network epoch determines the amount of token each node can claim.

Virtualized Subnode Orchestration Service

This service represents node computation and storage resources using a dynamically created virtualized subnode image that consists of SKALE daemon, catchup agent, and the transfer agent for interchain messaging.

It also functions in the recovery of failed virtualized subnodes and removal of assigned resources to a decommissioned virtualized subnode.

Attacks and Faults

A Network is always prone to attack and SKALE is no exception. SKALE adopted strategies to mitigate attacks and enhance quick recovery of downed nodes without hindering the consensus.

The strategies are outlined below as follows-

Reboots and Crashes

This protocol ensures that messages sent to a rebooting node will eventually be delivered once the node is online. This is done without obstructing the consensus.

On the other hand, when a node is hard crashed as a result of bugs or hardware issues, the peer nodes will continue sending messages to it. To prevent message overflow, messages older than one hour will be dropped.

Consensus will always move on provided that the number of Byzantine nodes in the network is less than 1/3 of the participating nodes.

Catchup Agent

Catchup agent ensures that nodes' blockchain and their proposal database are synced with the network.

When a node recovers from catchup, it will be synced by the catchup agent while it participates in the current consensus at the same time. However, the recovering node will not be able to propose new blocks because it does not have the hash of the last committed block which is used to make block proposals.

This protocol makes it very easy for hard-crashed nodes to rejoin the network.

Security Incident Response

Fool-proof networks are not yet practical. The SKALE network uses the Security Incident Response to respond to any attack on the network.

Since the SKALE network runs on Elastic Sidechains, security compromise in the network is likely to involve the Elastic Sidechains.

Refer to the Security Incident Response in the SKALE Whitepaper to see the step by step procedure that takes place when a bug is detected in the network.

SKALE Token

The SKALE token is a multi-use token that grants people the right to act in the network as one of the following-

1. Developer- Developers/entities can rent Elastic Sidechains with the desired configuration in the network. They pay a subscription fee that allows them to utilize the resources in the Elastic Sidechain. The developers pay for the duration they want to use the resources by staking a predetermined amount of SKALE tokens.

2. Validator- Validators get paid for their node's uptime and latency with the SKALE tokens.

3. Delegators- SKALE token holders can stake their tokens with validators and earn rewards at the end of the delegation period.

4. Government- When the SKALE network was launched, the N.O.D.E Foundation selected representatives to serve the community. You can find more information about the SKALE government here.

Network Bounties and Delegation Workflow

In this section, you will understand the processes involved in the SKALE network bounty as well as terminologies used to describe them.

Developers who wish to use the Elastic Sidechain resources have to choose their desired configuration. This includes the size and duration of the chain. They also have the option to enable virtualized subnode shuffling. After that, they go ahead to stake SKALE tokens into the Ethereum mainnet through the smart contract.

For validators to get paid, tokens enter the bounty pool from the mainnet. The token economy will also be inflated resulting in the creation of new tokens. These new tokens also enter the bounty pool for payments to the validators.

At the end of each network epoch, validators claim SKALE tokens based on their performances.

You can read up the terminologies here.

SKALE PROTOCOL

SKALE Protocol are processes adopted by the SKALE network to ensure smooth workflow in the network.

Next Gen PoS-Based Network

This protocol is adopted to punish defaulters in the network and award good performance.

Using the Proof-of-Stake system, each participating node is required to stake a predetermined amount of SKALE tokens.

When an ill behaviour is detected in the network, the SKALE token of the defaulting node will be slashed. Such behaviours include inability to participate in assigned consensus and poor uptime and latency.

In order to track a node's participation, uptime and latency, 24 peer nodes are assigned to track each node's performance.

The average of the tracked metrics is used to reward nodes at the end of the network epoch.

Alongside promoting good behaviour, this protocol is used as an additional security. To participate in the network, every node will stake a significant amount of SKALE tokens. This raises the cost of launching a successful attack on the SKALE network.

Delegation

Delegators can stake their tokens with any node in the network provided that the node has not reached the maximum amount of tokens that can be staked or delegated.

Consensus

The SKALE network uses a variant of Moustefaoi et. al consensus because it comes with the best fit properties for a truly decentralized network.

Leaderless

The SKALE network consensus does not have a leader instead the block proposal with the supermajority keyshares gets committed to the blockchain.

This protocol makes the consensus incorruptible and transparent because all virtualized subnodes stand an equal chance of getting a winning block proposal.

Asynchronous Timing Model

With this protocol, there is no expected time of getting a response for a message sent in the network. Messages are assumed to be eventually delivered. However, messages an hour old will be dropped and redelivered to prevent message overflow.

Byzantine Fault Tolerant

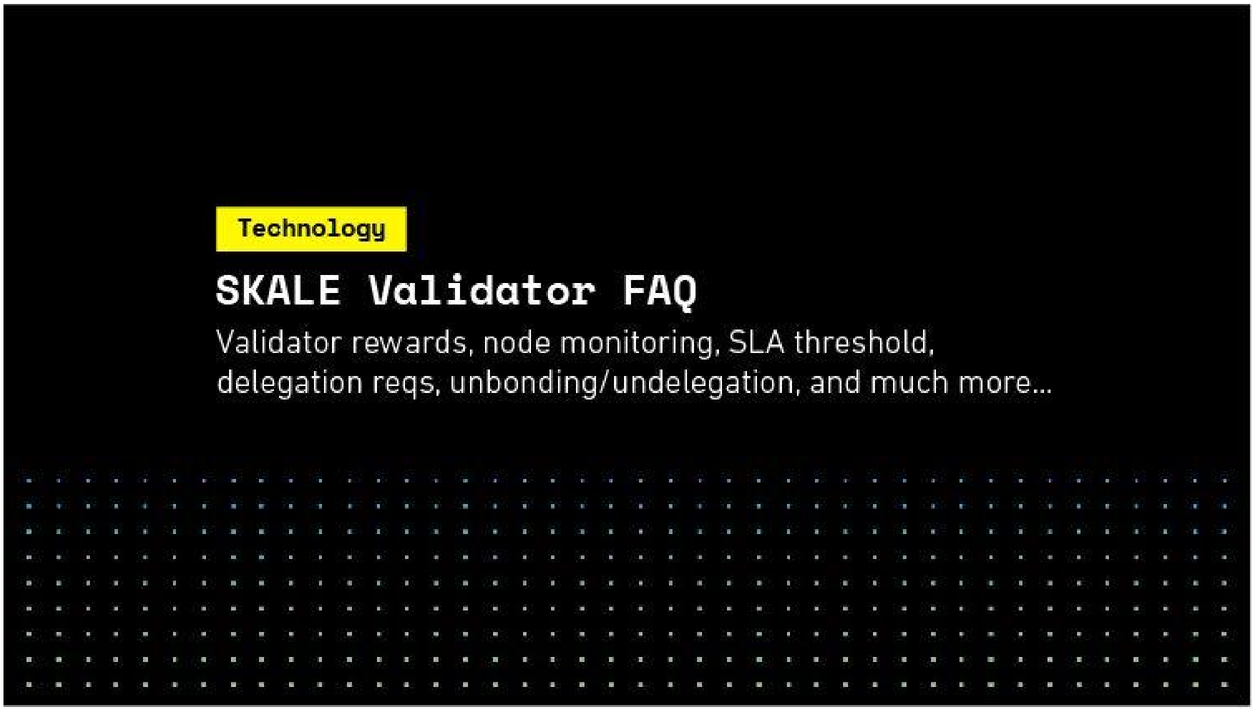
A Byzantine fault-tolerant network tolerates fault. This means that it carries on consensus even in the presence of less than one-third of corrupt nodes.

The SKALE network uses Asynchronous Byzantine Fault Tolerant protocol making the network stronger than just BFT.

Threshold Signatures

Threshold Signatures are used for supermajority voting. During consensus, the block proposal with the supermajority key shares is considered to have the winning proposal. It gets committed to blockchain.
Read more about SKALE Consensus here.

About SKALE Validator- All you need to know



Validator Nodes
Validator nodes are rewarded after each network epoch based on the average performance of that node.
Validators are paid at the end of each month from tokens in the bounty pool which are obtained from subscription fees and tokens minted during inflation.

How does the SKALE Network work?
SKALE Network Sidechains are created by developers and configured to suit their business demands.
Validators on the other hand are responsible for overseeing the Sidechains, they set up nodes which run independently and simultaneously validating transactions within the Sidechian they are assigned to. While developers pay for subscription fees, validators are paid for their work with the SKALE tokens.

What benefits does virtualized subnodes and node containerization provide?
Elastic Sidechains in the network are assigned randomly selected virtualized subnodes which run the SKALE Daemon and SKALE consensus.

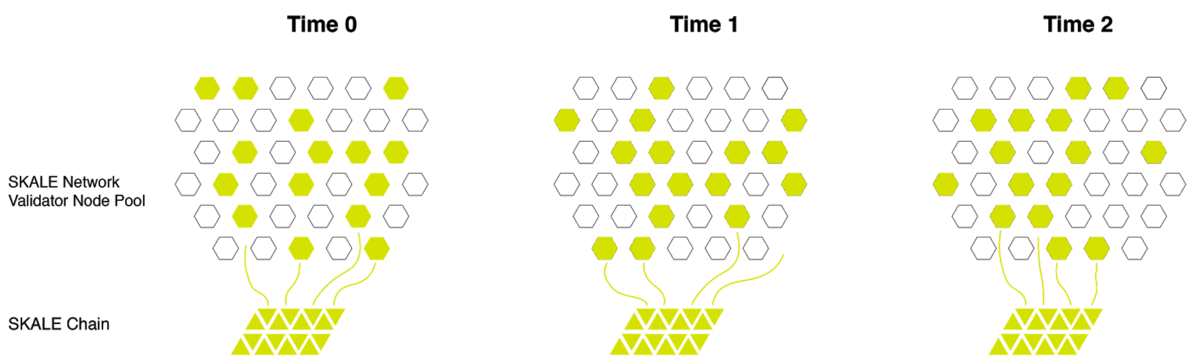
The SKALE Nodes can participate in multiple Sidechains through the virtualized subnodes. This is made possible by the containerization architecture adopted on the SKALE nodes. The SKALE are virtualized and act as a validator for an independent number of sidechains.

This subnode virtualization provides industrial-grade performance and offers more options to developers in the decentralized blockchain.

How Secure is the SKALE Network?

The validator nodes are the overseers of the network. They validate all the transactions in their respective sidechain. In other words, the security of the network largely depends on the behaviour of the validator nodes in the network.
Having a large number of validators means more eyes watching over the network.

To further enhance security in the network, the nodes in a particular network are randomly selected and continually shuffled. This enhances the transparency in the network.



How long does a validator operate in the network?

When registering a node in the network, validators choose the duration they want to operate. However, the minimum duration is 3 months.

How is validators' payout measured?

Each validator node is assigned 24 peer nodes to monitor and track its performance- participation, uptime and latency standard. At the end of the month, the average of the tracked metrics is used to entitle payouts to validators.

Refer to [SKALE Validator FAQ] for more details.

Follow us on-
[Telegram] @SkaleOfficial
[Twitter] @SkaleNetwork
[Discord] ([www.skale.chat])
[SKALE website] (www.skale.network)