SKALE

The SKALE Network

Technical Highlights

Technical Highlights of The SKALE Network

The SKALE Network is a configurable network of elastic sidechains that supports high-throughput and low-latency transactions without the high transaction costs found in public mainnets. The network offers expanded storage capabilities along with embedded connectivity and interchain messaging with the Ethereum mainnet. All of this is performed using a pooled transaction validation and security model that is efficient, scalable, and collusion-resistant.

Here are a few of the technical highlights of the network. A deeper description of these elements can be found in the SKALE Network Whitepaper.

- Zero to Near-Zero Gas Fees
- Random Node Selection/Frequent Node Rotation
- Virtualized Subnodes
- Containerized Validator Nodes
- Consensus via Asynchronous Binary Byzantine Agreement (ABBA)
- BLS Rollups
- Node Monitoring
- Ethereum Interoperability

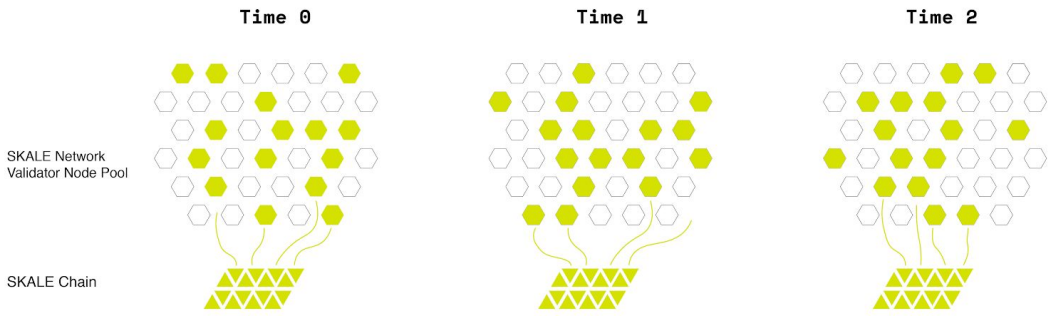
Zero to Near-Zero Gas Fees

Gas fees within the SKALE Network are zero – regardless of the size of the SKALE chain – as long as the chain is below a specific resource threshold. This zero to near-zero gas fee structure is a significant benefit in terms of building and operating decentralized applications. A large gating factor in user adoption and building out profitable use cases is the friction imposed by blockchain gas fees. Removing these costs from the equation translates into much easier go-to-market opportunities, higher adoption rates, and more successful decentralized solutions.

The SKALE chain container is allocated CPU, memory and disk sizes to proportionally perform operations with zero gas fees up to a specific level. After this level is breached, gas becomes positive. This resource switch has two benefits – one is that it prevents Denial of Service (DOS) attacks and the other is that it indicates to the user that they might need to scale up to a larger SKALE chain size. (The latter indicator is analogous to moving up a level on a cloud service such as moving from a t2.micro to an m3.large on AWS.)

Random Node Selection/Frequent Node Rotation

Validator nodes are assigned to elastic sidechains via a random process that is arbitrated by a mainnet contract. Security of chain consensus is further protected via frequent node rotation. Nodes will be removed from one or more chains on a non-deterministic schedule and new nodes added. This rotation takes place via the node cores continually checking in with the mainnet – exiting current chains and connecting with and working on new chains as so determined by the mainnet contracts and its random assignment algorithms.

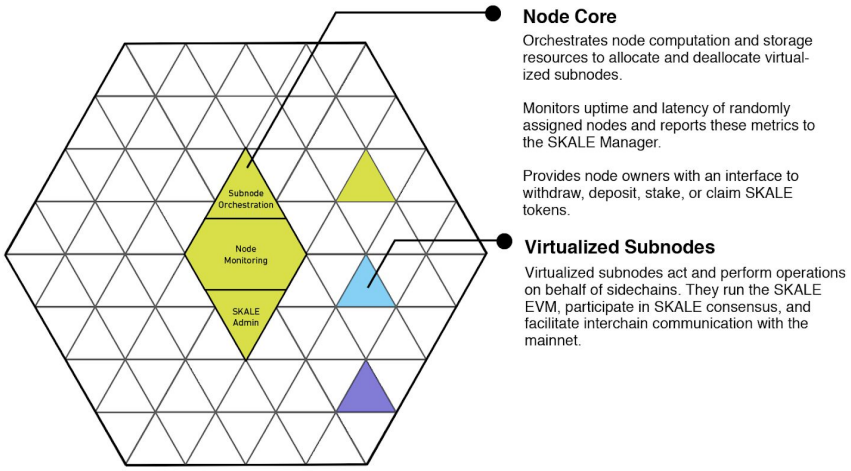


Nodes within SKALE Chains are Regularly and Randomly Rotated Thereby Leveraging the Security Pool of the Entire Network on Behalf of Each Chain

Virtualized Subnodes

Each elastic sidechain is comprised of a collective of randomly appointed virtualized subnodes which run the SKALE daemon and the SKALE consensus. Nodes in the SKALE Network are not restricted to a single chain but rather can work across multiple sidechains via the use of virtualized subnodes. This multiplex capability is made possible via a containerized subnode architecture deployed on each node in the Network. Each node is

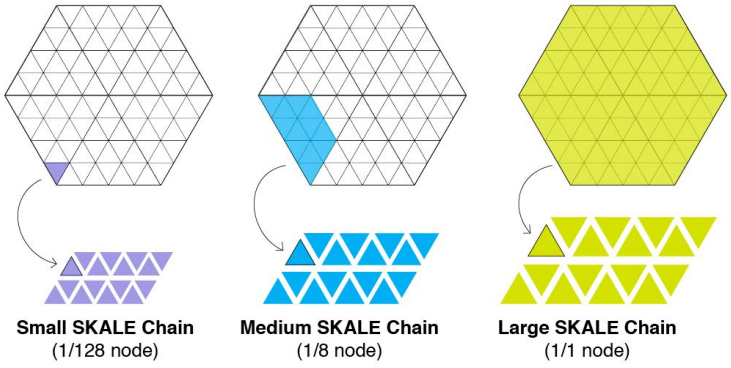
virtualized and is able to participate as a validator via this subnode architecture for an independent number of sidechains.



Validator Nodes Consist of Virtualized Subnodes and a Node Core

Containerized Validator Nodes

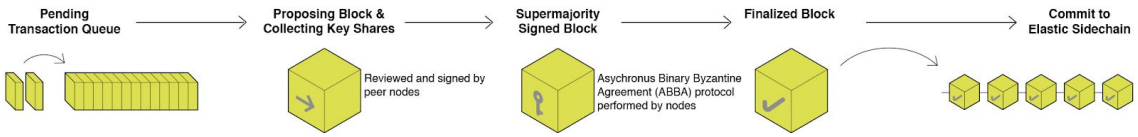
Virtualized subnodes are enabled via an innovative containerized architecture that provides industrial-grade performance and optionality for decentralized application developers – performance and flexibility that is similar to traditional centralized cloud and microservice systems. Containers are divided into several main components encapsulated via a dockerized Linux OS – allowing for each node to be hosted in an OS-agnostic manner.



Elastic sidechains can use a partial set of resources
or the full node depending on the size of the sidechain

Consensus via Asynchronous Binary Byzantine Agreement (ABBA)

The consensus model used for block creation and commitment for each elastic sidechain is a variant of the Asynchronous Binary Byzantine Agreement (ABBA) protocol. (Derived from Mostefaoui et al. although other consensus protocols can be used, as long as it satisfies certain properties.) The benefits of the ABBA protocol is that it is designed to exhibit robustness in the case of subnode downtime where each latent and/or down subnode is regarded as a slow link. Additional details on the protocol can be viewed here.



ABBA Consensus Protocol

Interchain Messaging via BLS Threshold Signatures

Each elastic sidechain supports BLS (Boneh–Lynn–Shacham) threshold signatures which is important for supporting interchain messaging. Virtualized subnodes for each chain are able to validate a transaction that was signed and committed by the subnodes in another chain through the use of that chain's group signature. This signature is made available to all other chains via publishing on the Ethereum mainnet.

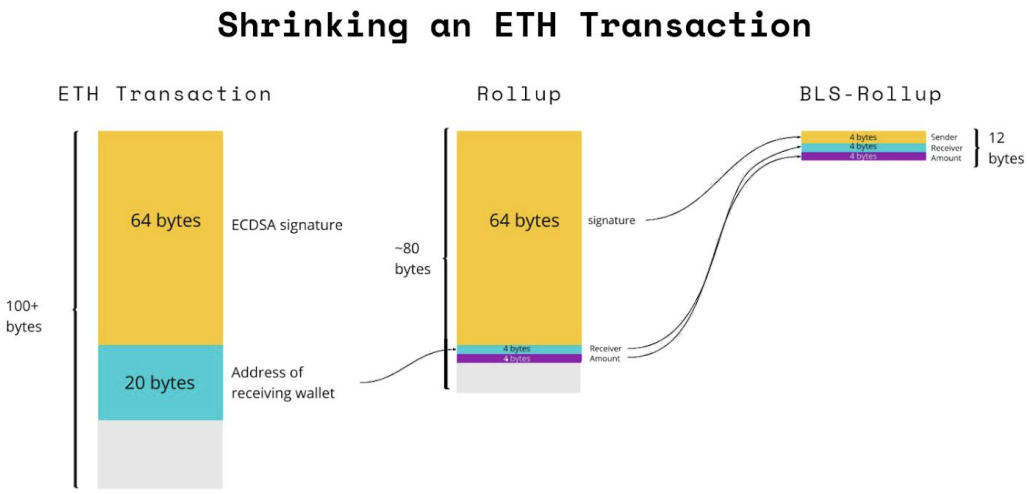
This messaging capability mirrors a microservice model whereby a sidechain is able to perform one or more specific operations and then feed these outputs directly to another chain or onto a message queue (i.e. the Ethereum Mainnet) which can then serve as inputs for other sidechains and their processing needs. SKALE's interchain messaging provides support for all the major Ethereum token standards including ETH, ERC20, ERC721, ERC777, and Dai.

BLS Rollups

Each sidechain also supports BLS Rollups which provides an efficient and secure way to use the SKALE Network to improve throughput and lower gas costs on the Ethereum mainnet. A rollup can generally be defined as a solution where transactions are published on chain, but computation and storage of transaction results is done differently to save gas. BLS Rollups works by using a crypto algorithm called aggregated BLS signatures to shrink ETH transaction sizes.

The work to integrate BLS Rollups into SKALE encompasses three phases of development with Phase 1 providing up to fifty transactions per second for ERC-20 token transfers. Other phases will further improve transaction performance metrics. A more detailed outline on how BLS rollups work and the roadmap can be found here.

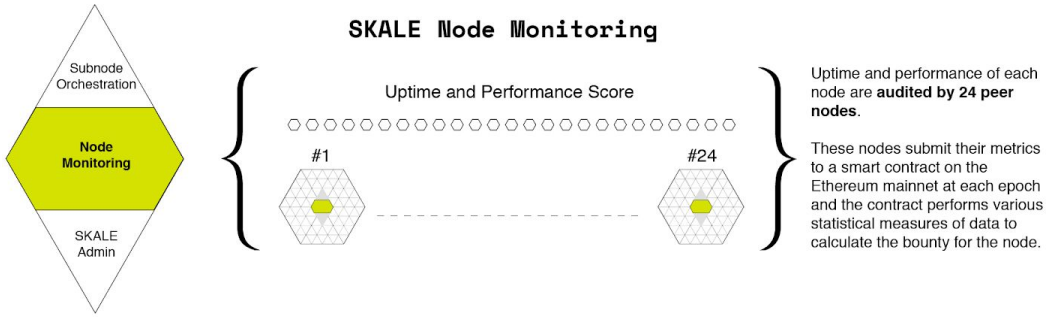
Sidenote: In addition to BLS Rollups, there are other rollup approaches including Optimistic Rollups and ZK Rollups. Optimistic rollups are problematic in that they can allow incorrect results to be published on chain (which can only be addressed via a post-transaction complaint procedure). ZK Rollups are technically better than Optimistic Rollups in that they preserve stateful correctness on chain. Problems arise, however, in that ZK-S*ark operations are compute intensive which means transactions can sometimes take hours to finalize. BLS Rollups are a more realistic solution in that it uses BLS cryptography which is lightning fast.



BLS Rollups Offer Significant Advantages Compared to Other Forms of Rollups

Node Monitoring

A Node Monitoring Service (NMS) runs on each SKALE Node and facilitates the performance tracking of a certain number of other nodes in the network. Performance tracking measures both uptime and latency through a regular process which pings each peer node and logs these measurements to a local database. At the end of each epoch, these metrics are averaged and submitted to smart contracts on the mainnet that use them to determine the payout distribution to nodes as well as flag suspect nodes for review and potential penalties.



Node Core of Each Node Analyzes and Scores the Performance of Other Nodes in the Network

Ethereum Interoperability

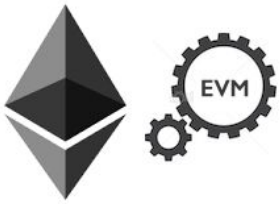
The SKALE Network is designed as a security and execution layer that ties in closely with the Ethereum network in service of its security and operational model. The smart contracts that maintain node operation all execute on the Ethereum mainnet. In addition, the validator stakes and user subscriptions not to mention the token inflation, are also maintained or controlled by smart contracts running within the Ethereum mainnet.

Solidity Support

The SKALE Network uses Solidity as its contract language sparing developers from having to learn a new language. Solidity is an object-oriented, high-level language for implementing smart contracts. It was influenced by C++, Python, and JavaScript and is designed to target the Ethereum Virtual Machine (EVM).

Ethereum EVM Compatibility

The SKALE execution model is fully compatible with the Ethereum Virtual Machine (EVM) making it so that smart contracts that run on the Ethereum mainnet can also run on the SKALE Network. There is no need to rewrite or port smart contracts. Anything written for the EVM will execute on SKALE. Developers are therefore able to migrate to SKALE elastic sidechains in a phased manner – moving smart contracts to SKALE on an individual basis as the needs and benefits dictate.



Developer Tool Support

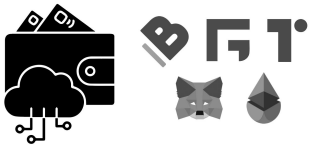
Support for Solidity and the Ethereum VM also extends to commonality with Ethereum developer tools. Developers are able to use the same tools they use when working on the Ethereum mainnet. These include connecting to the network via web3.js and web3.py as well as using tools such as Truffle and Remix.

Token Support

The SKALE Network supports all major Ethereum token standards including ETH, ERC20, ERC721, ERC777, and Dai. Interchain messaging, as well as deposit boxes and token clones, ensures the integrity and fidelity of token operations with the SKALE Network.

Common Wallet Support

The SKALE Network supports a number of major crypto wallets and browser plugins and bridges. These include Bitski, Fortmatic, Metamask, Portis, Torus. These interface components are well-regarded in the community and used by thousands of developers.



About the SKALE Network

The SKALE Network is an open-source elastic blockchain network protocol. The mission is to make it quick and easy to set up cost-effective, high-performance sidechains that run full-state smart contracts. The SKALE Network aims to deliver a performant experience to developers that offers speed and functionality without giving up security or decentralization.

You can follow the SKALE Network on Telegram (@SkaleOfficial), Twitter (@SkaleNetwork), and Discord (www.skale.chat), visit the SKALE website (www.skale.network), read developer documentation at the SKALE Developer Portal (skale.network/docs), and see the code at Github (github.com/skalenetwork).