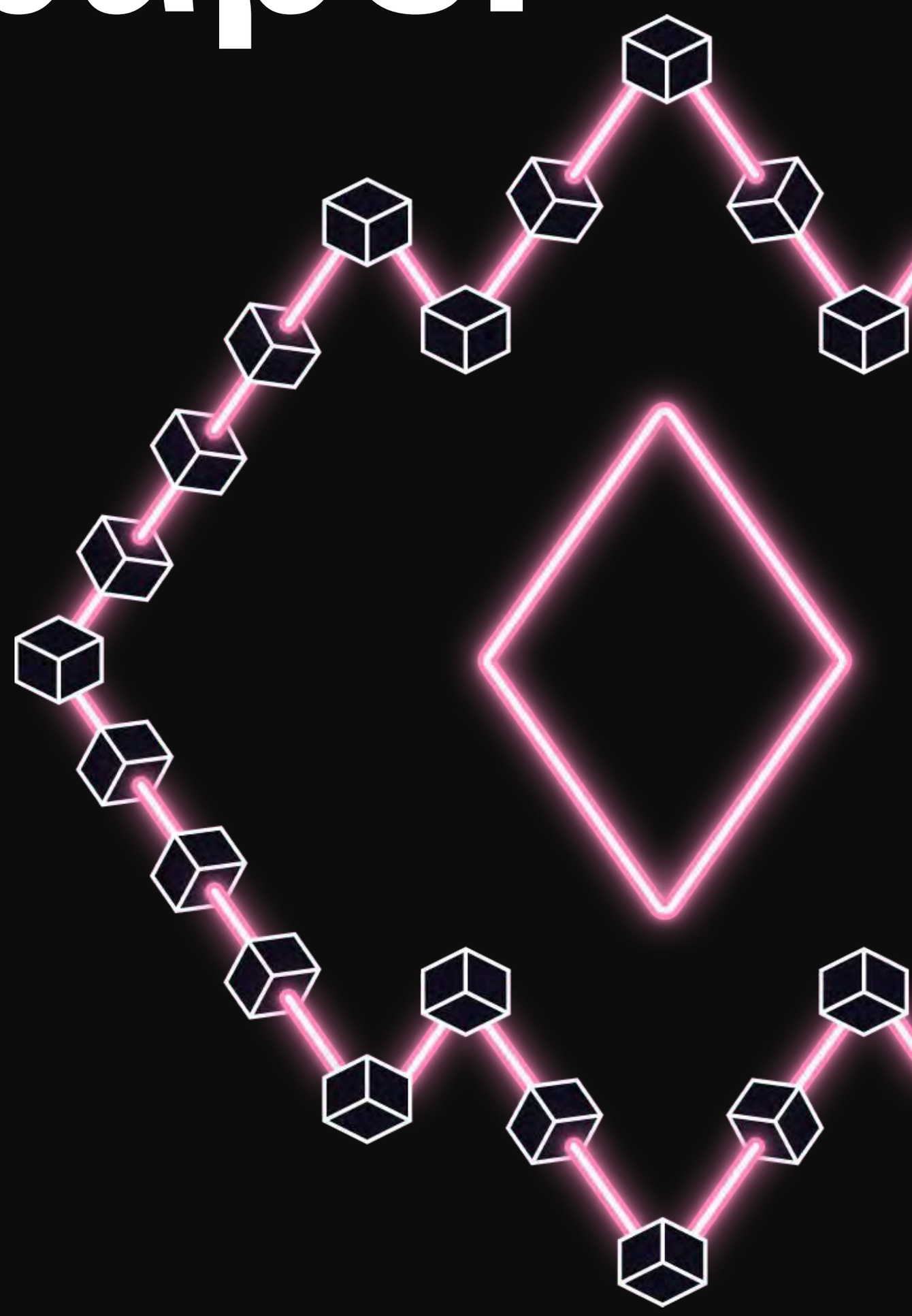
Mosaic Chain

Mosaic Chain Whitepaper



Mosaic Chain

Index

Abstract 4

Introduction 5
- Motivation 5
- Design Principles 6
- Key Contributions 7

System Overview 8
- What Mosaic Chain Is 8
- Network Maturity and Deployment Status 9
- Roles and Assets at a Glance 10

Architecture 11
- Protocol and Runtime Architecture 11
- Core On-Chain Modules 12
- Explorer, Indexing, and External Interfaces 13
- Networks and Configuration Surfaces 14

Consensus and Validator Set Mechanics 16
- Consensus Overview 16
- Sessions and Active Set Rotation 18
- Validator Subset Selection 19
- Operational Outcomes 20

NFT-Based Participation, Staking, and Economic Security 22
- Why NFTs for Validator Rights 22
- Validator NFTs 23
- Delegation NFTs (Delegator NFTs) 24
- Staking Contracts 25
- Rewards and Fees (Mechanics-Level) 26
- Slashing, Chilling, and Safety Mechanisms 27

Protocol Extensions and Native Products 29
- On-Chain Validator Reputation System (Mosaic Trust Score) 29
- Programmable Staking Vaults 32
- Cross-Chain Yield Routing via XCM (Mosaic Yield Bridge) 36

Economics (MOS) 39
MOS Utility 39
Supply and Denominations 39
Issuance and Reward Model 41
Vesting and Vesting-to-Freeze 42
Staking Incentive Pool 43

Governance and Upgrades 44
Governance Scope 44
Forkless Upgrades 45

Interoperability 46
Relationship to Polkadot 46
Parachain Status and Execution Model 46
XCM Direction 47

Validator OS and Operations 49
What Validator OS Is 49
Onboarding Flow 50
Reliability and Maintenance 51

Security Considerations 53
Key Custody and User Safety 53
Economic Security Model 54
Operational Security 55

Roadmap (Technical and Operational) 56

Mosaic Chain

Abstract

Mosaic Chain is a Substrate-based blockchain designed to support Web3 finance and privacy-first applications without sacrificing performance, security, or decentralization. The network introduces an NFT-based participation model where validator rights are represented as on-chain NFTs, enabling transparent, transferable participation while tying security to clearly defined stake value. To avoid the scalability bottlenecks that emerge when thousands of validators must coordinate continuously, Mosaic Chain uses a session-based consensus model that selects a rotating active validator set while still allowing a much larger validator population to participate over time.

Operationally, Mosaic Chain is paired with Validator OS, a purpose-built, plug-and-play validator operating system that standardizes node setup, connectivity, updates, and monitoring to reduce downtime and configuration errors. The result is a network that aims to make validator participation accessible to a wider range of operators while maintaining fast block production and predictable network behavior. This whitepaper describes the core protocol architecture, consensus and validator selection mechanics, the NFT-based staking model and its security properties, MOS token utility and economics, validator operations via Validator OS, and interoperability direction within the Polkadot ecosystem.

Introduction

Motivation

Decentralized financial systems impose stringent requirements on the infrastructure that supports them. Unlike general-purpose computation platforms where occasional latency spikes or transient inconsistencies may be tolerable, financial applications demand predictable finality, robust economic security guarantees, and low-friction participation for both operators and end-users. Stuck or delayed transactions are unacceptable in a financial context; the risk model for DeFi protocols differs fundamentally from that of social applications or IoT networks. These constraints motivate the design of a purpose-built chain rather than deployment on a general-purpose platform.

At the same time, decentralization in practice remains bottlenecked by the operational complexity of running validator infrastructure. Most proof-of-stake networks require validators to navigate complex installation procedures, manage key material across multiple accounts, monitor uptime manually, and coordinate software upgrades through off-chain channels. The result is a validator set dominated by technically sophisticated operators, often concentrated in a small number of cloud providers. This concentration undermines the decentralization guarantees that proof-of-stake systems are designed to provide.

Mosaic Chain addresses both challenges simultaneously. It provides a Substrate-based runtime optimized for financial applications with predictable block times, enforceable economic security through on-chain staking instruments, and a validator operating system that reduces the barrier to running a node from expert-level systems administration to a plug-and-play hardware setup. The network is designed from the outset to support thousands of validators without requiring all of them to participate in every consensus round, preserving both decentralization breadth and block production efficiency.

The design philosophy underlying Mosaic Chain is that finance-specific blockchain infrastructure benefits from purpose-built optimization. General-purpose chains serve many use cases simultaneously—gaming, social media, IoT, supply chain—and their design tradeoffs reflect this breadth. A chain built exclusively for financial applications can make different choices: prioritizing predictable block times over maximum throughput, emphasizing economic security legibility over smart contract expressiveness, and investing in validator accessibility over hardware scalability. These are not universally correct tradeoffs, but they are the right tradeoffs for the domain Mosaic Chain serves.

Mosaic Chain

Design Principles

The architecture of Mosaic Chain is guided by five core design principles that inform decisions at every layer of the protocol.

First, the network seeks to scale decentralization without degrading performance. Traditional consensus protocols face a fundamental tension: adding more validators increases communication overhead, which degrades throughput and latency. Mosaic Chain resolves this tension through session-based validator rotation, allowing a large total validator population (2,400+) while maintaining a compact active set (~200) for efficient block production in each session.

Second, validation must be operationally accessible. The "expert-only" barrier to running a validator node is a structural problem for decentralization, not merely a user experience inconvenience. Validator OS abstracts away installation, configuration, connectivity, monitoring, and software updates into a standardized operating layer, reducing misconfiguration risk and lowering the technical threshold for participation.

Third, security incentives must be legible and enforceable on-chain. By representing validator rights as NFTs with explicit nominal values—and by expressing delegation either as MOS-token staking contracts or, optionally, as Delegation NFTs—Mosaic Chain makes participation and economic security transparent, auditable, and enforceable on-chain. Stake amounts, commission terms, slashing conditions, and participation rights are recorded on-chain rather than managed through opaque off-chain agreements.

Fourth, the protocol must support iterative evolution via forkless upgrades. Financial infrastructure cannot tolerate the disruption and coordination costs of hard forks. Substrate's forkless runtime upgrade mechanism allows Mosaic Chain to evolve its business logic, adjust parameters, and introduce new features without requiring validators to download and install new software or risking chain splits.
Fifth, a clear interoperability path for cross-chain assets and messaging must be maintained. As a parachain in the Polkadot ecosystem, Mosaic Chain inherits access to Cross-Consensus Messaging (XCM) for trustless cross-chain communication, avoiding the security risks associated with traditional bridge architectures.

Key Contributions

Mosaic Chain introduces several distinct design elements that differentiate it from existing blockchain platforms.

NFT-based validator participation rights. Rather than representing validator status as a binary on-chain flag or a simple account balance, Mosaic Chain encodes participation rights as NFTs with explicit nominal values. This approach makes validator rights portable (subject to binding rules), auditable, and compositionally richer than traditional staking models. The NFT model introduces explicit on-chain participation rights via PoS and DPoS Validator NFTs. Delegation can be expressed directly as MOS-token staking contracts, or optionally via Delegation NFTs when a transferable, nominal-value instrument is desired. This preserves a simple path for MOS delegation while enabling richer on-chain instrument design where it adds value.

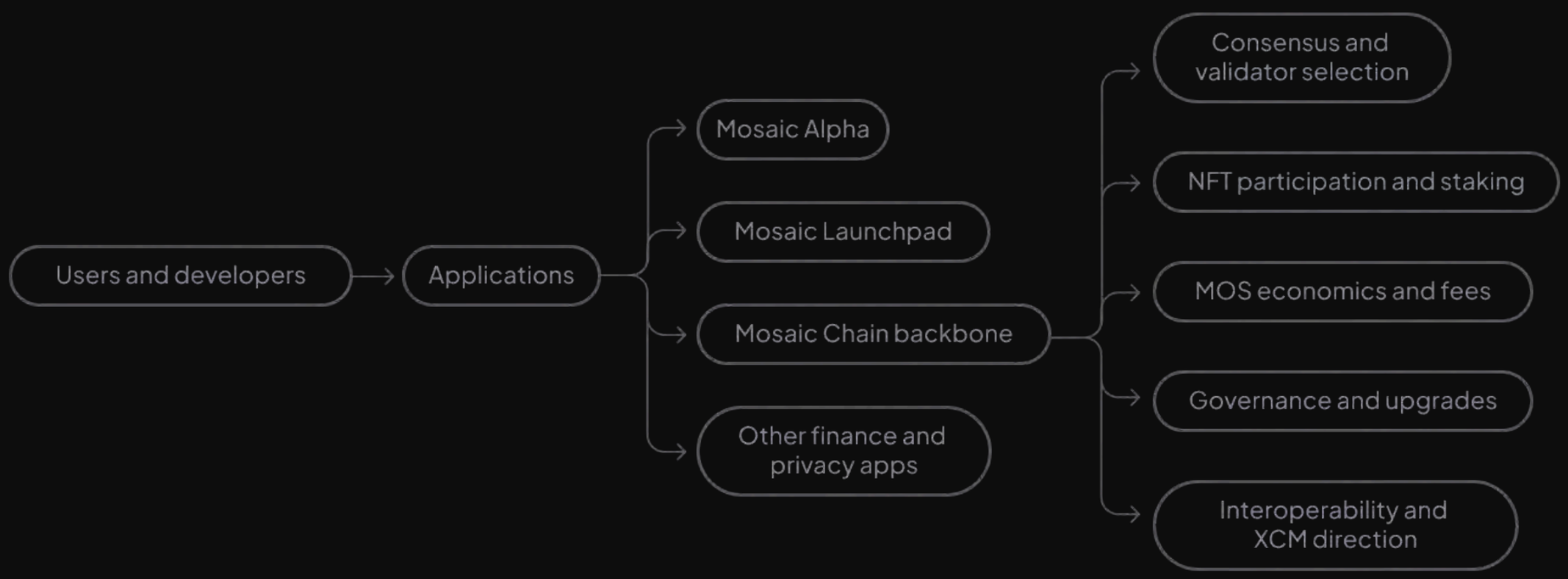
Session-based validator rotation with a large overall population. The improved bucket selection algorithm enables a network of 2,400+ validators to participate equitably over time while maintaining an active set of approximately 200 validators per session. Simulations demonstrate near-uniform selection distribution over weekly timeframes, with sufficient variance to prevent predictability.

Validator OS as a standard operating layer. Rather than treating validator operations as an external concern, Mosaic Chain includes a purpose-built Debian-based operating system that standardizes the entire validator lifecycle from hardware setup through ongoing maintenance. This design decision treats operational reliability as a protocol-level concern rather than an afterthought.

System Overview

What Mosaic Chain Is

Mosaic Chain is a custom Substrate runtime optimized for a finance-oriented ecosystem. Built on Parity Technologies' Substrate framework, it inherits the modular pallet-based architecture that separates consensus logic, application logic, and cryptographic host functions into composable components. The runtime is compiled to WebAssembly (Wasm) and stored on-chain, enabling forkless upgrades that do not require validator coordination or software reinstallation.



The core protocol, described in this paper, constitutes the "backbone" of the Mosaic ecosystem. It encompasses the consensus mechanism and validator selection algorithm, the NFT-based staking and delegation system, the MOS token and its economic model, the Validator OS operational layer, and the interoperability interface with the Polkadot relay chain. Together, these components provide the infrastructure layer on which financial applications such as Mosaic Alpha (a decentralized portfolio management platform) and Mosaic Launchpad (a Web3 fundraising platform) are built. Because user-friendliness and discoverability are core priorities, Mosaic Explorer also serves as the primary frontend for most validator and delegator workflows, backed by the decentralized indexing stack described in Section 3.3.

The distinction between backbone infrastructure and application-layer products is deliberate. Financial applications built on Mosaic Chain inherit the security, finality, and interoperability properties of the base layer without needing to reimplement them. The base layer, in turn, is designed to be minimal and stable, with new features introduced through the extensibility mechanisms described in Section 6 rather than through modifications to the core protocol.

Mosaic Chain

Network Maturity and Deployment Status

Mosaic Chain operates across multiple environments to support iterative development and safe production deployment. The devnet is used to validate runtime upgrades, parameter changes, and operational procedures under controlled conditions to test them together with partnering companies and DAOs on the Paseo network. The mainnet is the production network where MOS, validator participation rights, and protocol incentives operate under real economic constraints.

As of Feb 16, 2026, mainnet network telemetry reports the following high-level indicators via the public status dashboard:

- Total validator count: 2,027
- Current block: 1,662,769
- Node health (total reporting: 1,945):
  - Chain: 1,928 healthy, 17 unhealthy
  - Indexer: 1,770 healthy, 175 unhealthy
  - Admin: 1,937 healthy, 8 unhealthy

Validator lifecycle status at this point in time indicates that the large majority of validator participation instruments are active, with a smaller portion chilled and minimal faulted status, consistent with an operational network that is onboarding at scale while maintaining stable block production. The dashboard also reports live breakdowns of validator-side and delegator-side participation instruments, as well as session participation states (current set versus queued or out-of-session), providing ongoing transparency into how participation is distributed over time.

Functionally, the core backbone described in this whitepaper is already implemented and operating on mainnet, including session-based block production, the validator lifecycle and permissioning model, Validator OS onboarding and operations, and the on-chain interfaces required for staking and participation management. Because network-level metrics evolve continuously, all quantitative references in this document should be interpreted as time-stamped snapshots unless otherwise stated.

Mosaic Chain

Roles and Assets at a Glance

The Mosaic Chain protocol defines four primary roles and their associated on-chain assets. Understanding these roles and the instruments they interact with is essential to the protocol descriptions that follow.

Validators operate nodes that produce blocks and participate in consensus. Each validator must hold a Validator NFT (either PoS or DPoS type) that grants the right to participate and defines the nominal stake value associated with that validator. Validators are subject to slashing for downtime or protocol violations, and they earn block rewards proportional to their participation.

Delegators are MOS token holders who back validators with additional stake. Delegation is formalized through on-chain staking contracts that specify the stake amount, commission rate, and minimum staking period. Delegators earn rewards proportional to their contribution, net of the validator's commission. To improve user-friendliness, rewards do not need to be claimed; they are automatically distributed and compounded after every session through an on-chain background process. Delegators may delegate in two ways: (1) by delegating MOS tokens directly via staking contracts, and (2) by using Delegation NFTs whose explicit nominal values represent stake commitments as transferable on-chain instruments. Importantly, Delegation NFTs are optional: delegators can delegate MOS without minting Delegation NFTs, and Delegation NFTs are primarily used when a delegator wants an instrument-like position with explicit nominal value representation and additional composability.

The MOS token is the native asset of Mosaic Chain. It serves as the unit of account for transaction fees, staking rewards, delegation, proposal deposits for treasury/fund decisions, and (in a future upgrade) token-weighted governance voting and runtime upgrades. The maximum supply is fixed at 2,000,000,000 MOS.

Validator NFTs and Delegator NFTs are on-chain instruments that represent participation rights and stake commitments. Unlike simple balance-based staking, the NFT model makes these rights explicit, transferable (subject to binding rules), and composable with the broader on-chain state. Each NFT carries a nominal value denominated in MOS that directly contributes to the economic security of the network.

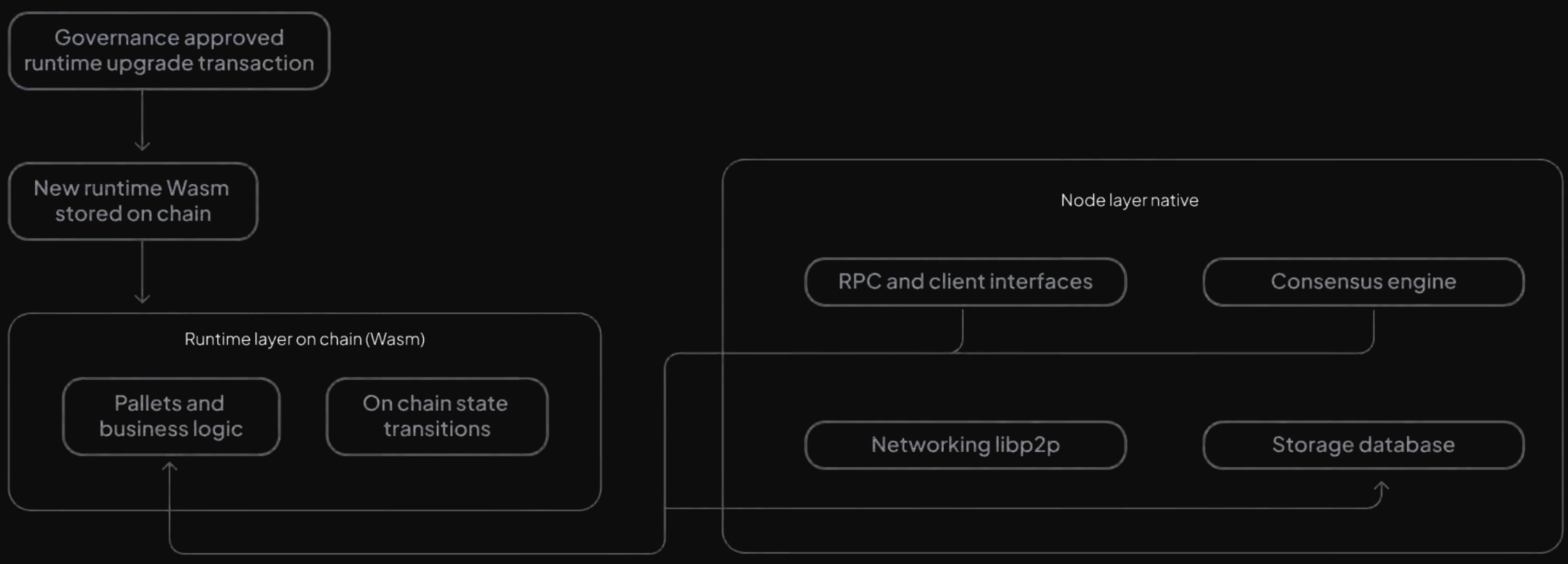
Mosaic Chain

Architecture

Protocol and Runtime Architecture

Mosaic Chain follows the standard Substrate architectural separation between the node layer and the runtime layer. This separation is fundamental to the protocol's upgrade model and operational characteristics.

The node layer encompasses the components that run as native software on validator hardware. These include the peer-to-peer networking stack (based on libp2p), the consensus engine that coordinates block production and finalization, and the persistent storage layer that maintains the blockchain state. The node layer handles peer discovery, manages transaction propagation across the network, processes Remote Procedure Call (RPC) requests from external clients, and interfaces with the runtime to execute state transitions.



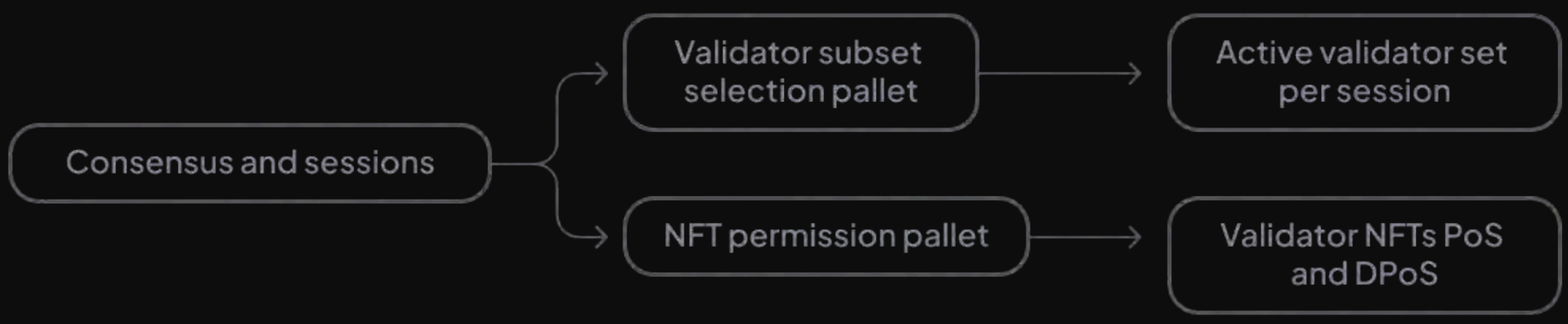
The runtime layer defines the blockchain's business logic: the rules governing state transitions, account balances, staking operations, NFT lifecycle management, and all other on-chain operations. Critically, the runtime is compiled to WebAssembly and stored as part of the chain state itself. When a validator produces a block, it executes the current on-chain runtime to validate and apply transactions. The runtime layer defines the blockchain's business logic: the rules governing state transitions, account balances, staking operations, NFT lifecycle management, and all other on-chain operations. Critically, the runtime is compiled to WebAssembly and stored as part of the chain state itself. When a validator produces a block, it executes the current on-chain runtime to validate and apply transactions. This architecture enables forkless upgrades: a governance-authority-approved runtime update is applied through an on-chain transaction, and all validators automatically begin executing the new logic from the next block without requiring any manual software update.

The separation between node and runtime also provides a clean security boundary. The node layer operates as native code with access to system resources (networking, storage, cryptographic host functions), while the runtime executes within a sandboxed WebAssembly environment with controlled access to host functions. This sandboxing ensures that runtime logic cannot directly access or corrupt node-level state, and that a bug in application-level logic cannot compromise the integrity of the networking or storage layers. For a finance-oriented chain where the cost of runtime errors is measured in economic value, this isolation is particularly important.

The runtime is composed of pallets, which are modular components analogous to libraries or packages in conventional software development. Each pallet encapsulates a specific domain of functionality with its own storage, events, errors, and callable functions. Pallets can depend on and interact with other pallets through well-defined trait interfaces, enabling composition without tight coupling. This modularity allows the Mosaic Chain team to add, modify, or replace individual protocol features without disrupting unrelated functionality.

Core On-Chain Modules

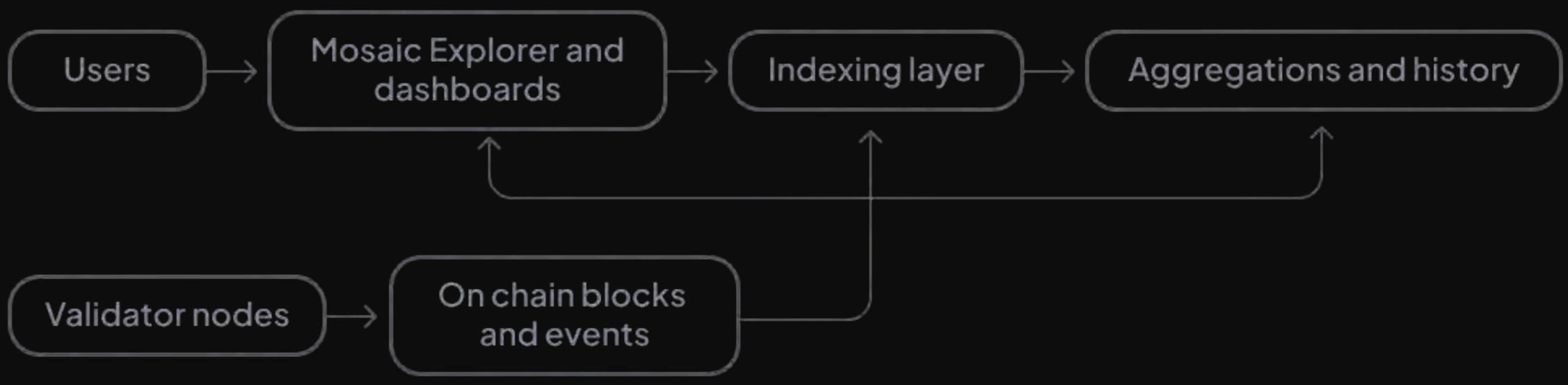
The Mosaic Chain runtime is organized around two categories of pallets: consensus-related pallets that form the core security infrastructure, and application-related pallets that provide the logic for decentralized applications.



NFT-Permission Pallet. Built on top of the standard pallet-nfts framework, this pallet manages the creation, ownership, attribute assignment, and lifecycle of permission NFTs. It supports minting new permission NFTs, binding and unbinding NFTs to and from validator accounts, and chilling/unchilling NFTs to temporarily disable or re-enable their use in consensus. The pallet enforces the invariant that only properly bound NFTs can authorize a node to participate in block production.
Staking Pallet. This pallet provides the core functionality for staking and unstaking both MOS tokens and NFTs. It manages the lifecycle of staking contracts between delegators and validators, enforces minimum staking periods and amounts, processes commission calculations, and coordinates with the slashing mechanism. The staking pallet also handles the "kick" operation that allows validators to remove delegators under defined conditions.
NFT-Delegation Pallet. This pallet manages Delegation NFTs that represent additional stakes that have a nominal value in MOS with a maximum number of sessions they can participate in. It handles the minting of Delegation NFTs with specified nominal values and participation durations, their binding to DPoS (delegation-enabled) validators, unbinding upon contract expiration, and proportional slashing when the validator to which they are bound is penalized. The pallet includes expiration logic that triggers automatically when a bound delegation reaches the end of its configured session count.
Validator-Subset-Selection Pallet. This pallet implements the improved bucket selection algorithm that determines which validators are included in the active set for each session. It manages the bucket state for each validator, calculates the selection outcome at session boundaries, and adjusts session length to match the number of selected validators. This pallet is the on-chain implementation of the selection mechanism described in detail in Section 4.
The project open-sourced core runtime components via the Mosaic Chain GitHub repository (github.com/Mosaic-Chain/mosaic-chain) to support auditability and community contribution.

Explorer, Indexing, and External Interfaces

Mosaic Explorer serves as both a blockchain explorer and the primary operational interface for validators and delegators. It is backed by a multi-technology, decentralized indexing infrastructure that provides real-time access to on-chain data.
Every Mosaic Chain node serves a dual function: participating in consensus to produce and validate blocks, and simultaneously transmitting on-chain data to the indexing layer that feeds frontend interfaces. This architectural choice ensures that data availability for the explorer and other frontends does not depend on a centralized indexer service. Instead, the data flow is inherently decentralized, with each node contributing to the overall data infrastructure.

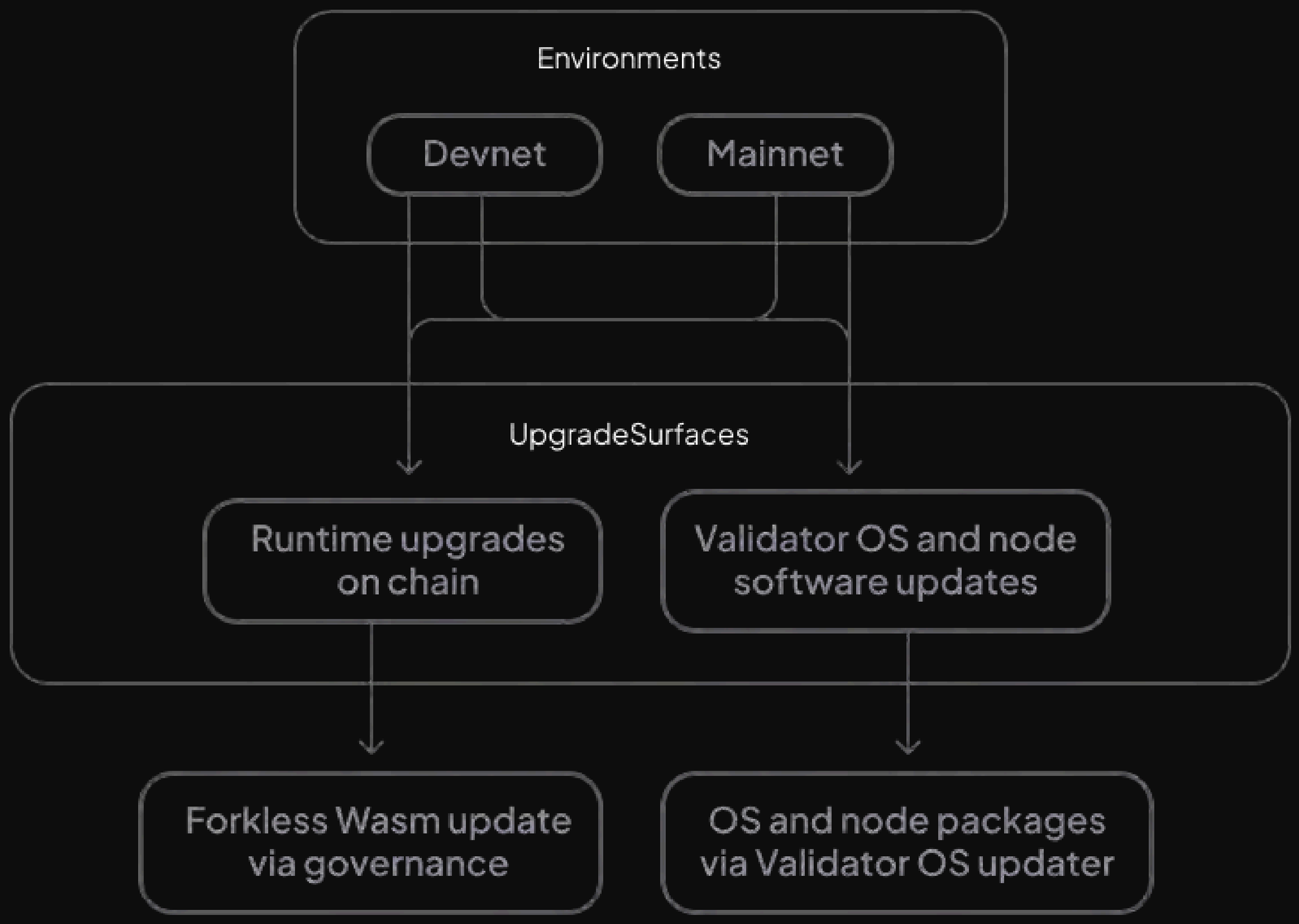


The indexer provides efficient data querying capabilities that go beyond what direct RPC calls to a full node can offer. While a full node can respond to queries about current state, the indexer maintains derived and historical data structures optimized for the specific query patterns that the explorer, validator dashboard, and wallet interfaces require. This includes aggregated staking statistics, historical reward distributions, validator performance metrics, and session participation records.
The validator frontend provides a dedicated control panel for node operators. It surfaces real-time monitoring of node health, performance metrics, staking status, commission settings, and block reward history. The public frontend offers an intuitive interface for any user to explore blockchain data, inspect validator performance, and access essential network information. Both interfaces use a gateway layer that routes requests to the appropriate backend services. Read-heavy and historical queries are served via the indexer, while live chain reads and transaction submission are served via gateway-accessible RPC nodes. This architecture preserves performant data access without requiring the frontends to target individual validator nodes directly.

Networks and Configuration Surfaces

Mosaic Chain operates across multiple network environments with distinct purposes and configurations. The devnet serves as the development and testing environment where new runtime features, parameter changes, and operational procedures are validated before mainnet deployment.

The mainnet is the production network where real MOS tokens, validator NFTs, and economic consequences are at stake.



Key parameters that may differ between devnet and mainnet include session length, slashing severity, minimum staking amounts, validator subset size, and various safety thresholds. The devnet typically operates with relaxed parameters to facilitate rapid iteration, while mainnet parameters are calibrated for production security and stability.

The protocol has two distinct upgrade surfaces. Runtime upgrades modify the on-chain business logic (pallets, parameters, state transitions) and are deployed through the forkless upgrade mechanism described above. OS and software updates modify the node-level software and the Validator OS environment, and are delivered through the Validator OS update system described in Section 10. These two upgrade paths are independent: a runtime upgrade does not require a Validator OS update, and vice versa. This separation ensures that chain-level logic changes and operational software maintenance can proceed on independent schedules without creating coordination bottlenecks.

Mosaic Chain

Consensus and Validator Set Mechanics

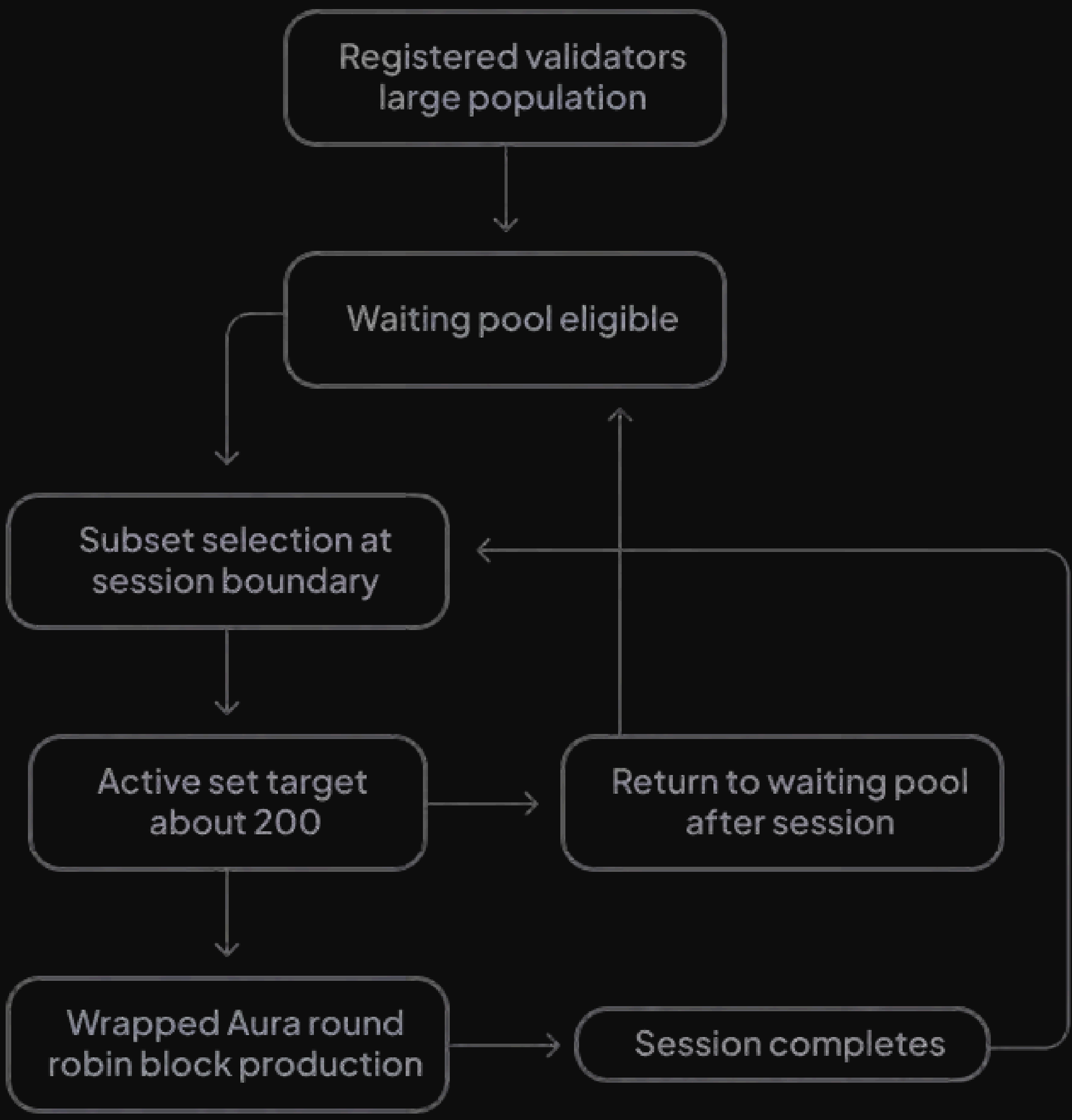
Consensus Overview

Mosaic Chain uses Proof of Authority (PoA) for block production (wrapped Aura), paired with an NPoS-style economic model for validator permissioning, delegation, rewards, and slashing. The staking/delegation model is conceptually similar to Polkadot's Nominated Proof of Stake (NPoS) scheme, but Mosaic expresses validator rights and stake commitments through permission NFTs and uses the bucket-based validator subset selection algorithm to rotate a large validator population through a compact active set.
The core challenge that the consensus design addresses is the tension between two desirable properties: broad participation (many validators contributing to decentralization) and efficient block production (fast, predictable finality with low communication overhead). Traditional consensus protocols require all validators to exchange messages during each consensus round, creating communication complexity that scales quadratically with the number of participants. For a network targeting 2,400+ validators with 6-second block times, this approach is infeasible.

Mosaic Chain resolves this tension by separating the total validator population from the active validator set. At any given time, the full network may comprise 2,400 or more registered validators, each holding a bound Validator NFT and running a synchronized node. However, only a subset of approximately 200 validators actively participates in consensus during each session. This active set rotates at session boundaries, ensuring that every validator receives an equitable opportunity to produce blocks and earn rewards over time while keeping per-session communication overhead manageable.
Block production within the active set uses a wrapped Aura (Authority Round) consensus mechanism. Aura is a simple, efficient consensus protocol in which validators take turns producing blocks in a round-robin fashion. Each validator in the active set is assigned specific slots during which it is authorized to produce a block. This deterministic assignment eliminates the energy waste and probabilistic finality of proof-of-work systems while providing predictable block times of 6 seconds.

The "wrapped" designation reflects the fact that Mosaic Chain's Aura implementation includes custom extensions to the standard Aura protocol. These extensions integrate the validator subset selection mechanism, the NFT-based permissioning system, and the session management logic into the consensus engine. The result is a consensus mechanism that inherits Aura's simplicity and efficiency while supporting the validator rotation, economic security, and operational accessibility features that are specific to Mosaic Chain.

It is worth noting how this approach compares with alternatives. Pure NPoS systems, such as Polkadot's relay chain, use election algorithms to select an active set based on stake weight, which introduces complexity around stake balancing and nominator-validator assignment. Many PoA-based networks operate with a fixed and relatively small validator set, which limits decentralization breadth. Mosaic Chain combines PoA block production with an NPoS-style delegation economy and a large registered validator population with rotating active sets determined by the bucket selection algorithm, prioritizing equitable participation while maintaining the operational efficiency of a compact active set.



Mosaic Chain

Sessions and Active Set Rotation

Time on Mosaic Chain is organized into discrete sessions, each lasting approximately one hour. A session represents a complete cycle of block production by the current active validator set. At the boundary between sessions, the validator subset selection algorithm runs to determine the next active set, and the new set begins producing blocks for the subsequent session.

The session lifecycle proceeds as follows. At the end of the current session, the validator-subset-selection pallet evaluates the bucket state for all eligible validators and produces the list of validators queued for the session after the next. At each session boundary, the chain activates the validator set that was previously queued, while simultaneously computing and queuing the subsequent set. The session length is dynamically adjusted to be an integer multiple of the number of selected validators, ensuring balanced round-robin slot assignment. As a result, each selected validator receives the same number of block production slots per session (one per round), rather than exactly one slot. Selected validators transition from the waiting state to the active state, while the previously active validators return to the waiting pool.

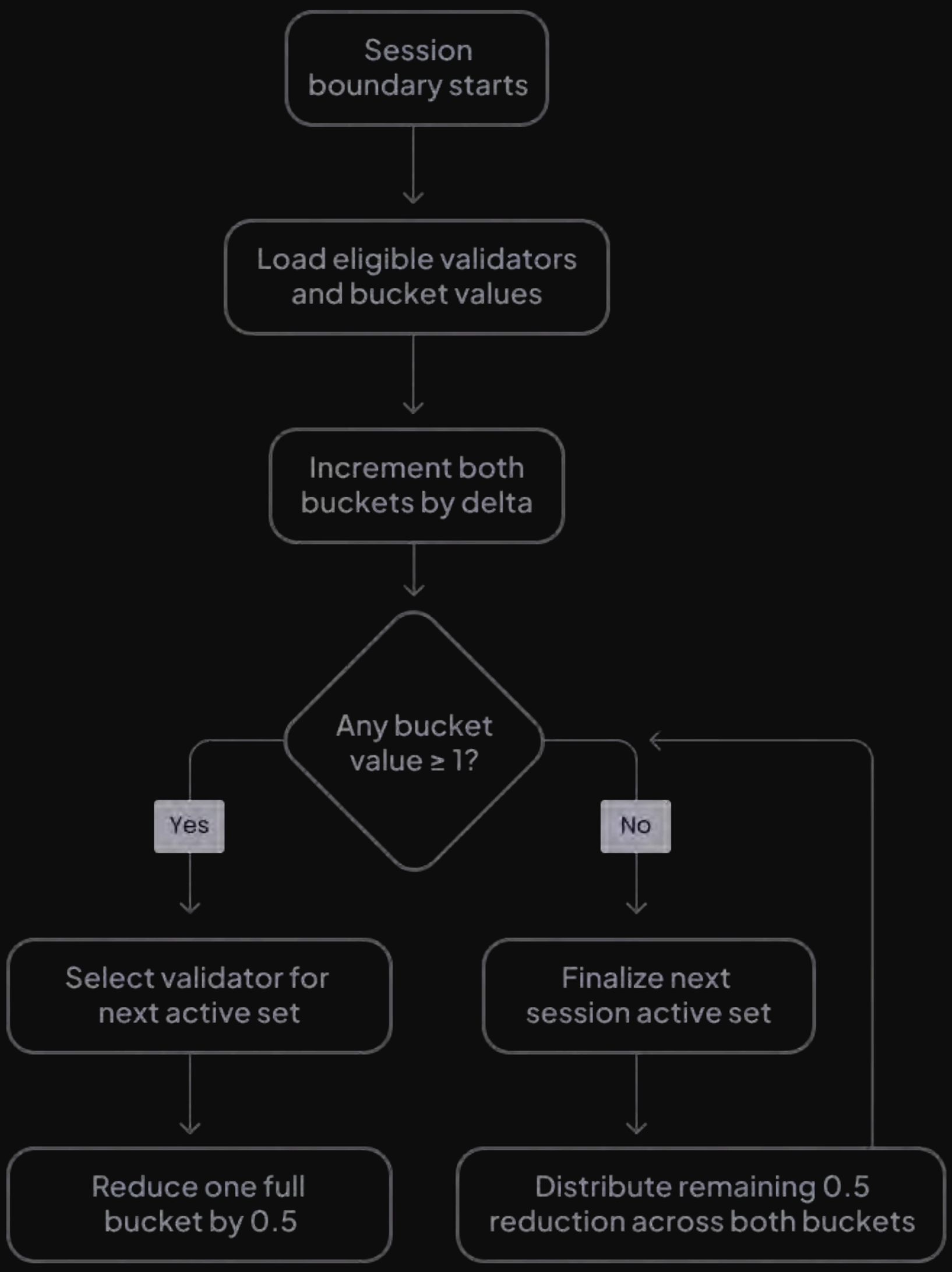
This session-based rotation provides several important properties. It ensures temporal fairness: over sufficiently long timeframes, every eligible validator is selected approximately the same number of times. It provides unpredictability: the randomized nature of the bucket selection algorithm makes it difficult for an adversary to predict which validators will be active in a given session, limiting targeted attack surfaces. And it maintains operational simplicity: the active set is fixed for the duration of a session, eliminating the complexity of mid-session validator set changes.

The one-hour session duration represents a deliberate calibration between rotation frequency and operational stability. Shorter sessions would provide more frequent rotation (and thus more granular fairness), but at the cost of increased overhead from frequent set transitions and reduced time for validators to produce their assigned blocks. Longer sessions would reduce transition overhead but increase the window during which a compromised active set could cause damage and decrease the granularity of fairness. The one-hour duration, combined with the ~200 validator active set, results in each validator producing approximately one block per session when selected, with sessions completing in roughly one hour under normal operating conditions.

Mosaic Chain

Validator Subset Selection

The validator subset selection mechanism is responsible for choosing which validators participate in each session's consensus. The mechanism must satisfy three requirements: fairness (each validator should be selected with approximately equal frequency over time), unpredictability (the next session's active set should not be predictable from public information alone), and stability (the active set size should remain close to the target of ~200 validators per session).



Mosaic Chain uses an improved bucket selection algorithm to achieve these properties. The algorithm works as follows.

Each validator is initialized with two buckets, each containing an independent uniform random value between 0 and 1. At each round (session boundary), every bucket's value is incremented by subset_size / (2 * number_of_validators), where subset_size is the target active set size and number_of_validators is the total eligible validator count. Validators with at least one bucket whose value reaches or exceeds 1.0 are selected for the next session. Upon selection, one of the validator's full buckets is decreased by 0.5, and the remaining 0.5 decrease is distributed uniformly between the validator's two buckets.

This dual-bucket mechanism produces selection behavior with desirable statistical properties. Simulation results with 1,000 validators and a target subset size of 200, run over multi-day timeframes, demonstrate near-uniform selection distribution over time while preserving enough randomness to reduce predictability. In practice, the algorithm keeps the active set size tightly clustered around the target while ensuring that no single validator is systematically favored or starved of selection opportunities across comparable time windows. This supports Mosaic Chain’s core objective: a compact per-session active set for efficient block production, paired with broad participation across a much larger validator population over time.

The subset sizes generated by the algorithm follow an approximately binomial distribution with a mean of 250 and an empirical standard deviation of 13.8, which can be effectively approximated by a Gaussian distribution. This means the active set size fluctuates slightly around the target but remains within a predictable range, ensuring consistent network throughput across sessions.

Operational Outcomes

Under normal operating conditions, the consensus mechanism produces a new block every 6 seconds, with each session's active set of approximately 200 validators rotating at around hourly boundaries. Validators joining the network undergo a registration process (described in Section 10) and enter the waiting pool, where they become eligible for selection in subsequent sessions. Validators leaving the network can voluntarily chill their NFT, which removes them from the waiting pool of eligible validators for future selection. Chilling does not retroactively remove a validator from the current session, and if the validator was already queued, it may still participate in the next session before the chill takes effect on subsequent selections.

The consensus mechanism interacts with the slashing and chilling system (described in Section 5.6) to handle validator misbehavior. When a validator in the active set goes offline during its assigned block production slot, the slot is missed and the validator is subject to slashing penalties. If a validator accumulates sufficient infractions to trigger auto-chilling, the protocol chills the validator NFT (functionally the same outcome as a manual chill): the validator is excluded from future selection from that point onward, while remaining in the current session and—if already queued—potentially also the next session. Other validators in the active set continue producing blocks normally; missed slots result in slightly delayed but not halted block production.

For validators experiencing temporary connectivity issues, the protocol defines a slacking period of 72 sessions. Unbinding is staged at the time of the delegator's transaction and committed at the end of the session. The primary role of the slacking period is early-exit protection: if a validator remains chilled for longer than the slacking period while a delegator's minimum staking period has not yet elapsed, the delegator may unbind and redelegate despite the remaining lock. This design acknowledges that brief downtime events are an operational reality, particularly for home-based validators, while still penalizing sustained unreliability that undermines network security.

Mosaic Chain

NFT-Based Participation, Staking, and Economic Security

Why NFTs for Validator Rights

The decision to represent validator participation rights as NFTs rather than using a simple balance-based staking model is a deliberate architectural choice with several motivations. In most existing proof-of-stake networks, the relationship between token holdings and validator status is mediated entirely by protocol-level accounting: a validator bonds tokens, a staking pallet tracks the bonded amount, and the consensus mechanism queries the staking pallet to determine eligibility. This approach is functional but opaque—the staking relationship exists only as entries in protocol storage maps, not as discrete on-chain objects that can be inspected, transferred, or composed with other instruments.
Mosaic Chain's NFT-based model takes a fundamentally different approach by reifying validator rights as first-class on-chain assets. This design choice has three primary motivations.

Explicit permissioning. In balance-based staking systems, the relationship between a token balance and the right to validate is implicit and governed by protocol rules that may not be immediately legible to participants. The NFT model makes permissioning explicit: a validator can participate in consensus if and only if it holds a properly bound Validator NFT. This binary condition is easy to verify, easy to explain, and easy to enforce on-chain.
Portability and composability. Validator NFTs are on-chain assets that can be transferred, subject to binding rules that prevent transfers while the NFT is actively participating in consensus. This portability enables secondary markets for validator rights, allows validator operations to be transferred between hardware setups or operators, and creates a foundation for future composability with other on-chain instruments.

Legible stake representation. Each Validator NFT carries a nominal value denominated in MOS that represents the minimum stake the validator commits to the network's security. This nominal value is a first-class on-chain attribute rather than a derived computation, making the economic security contribution of each validator immediately visible to delegators, governance participants, and external observers.

Mosaic Chain

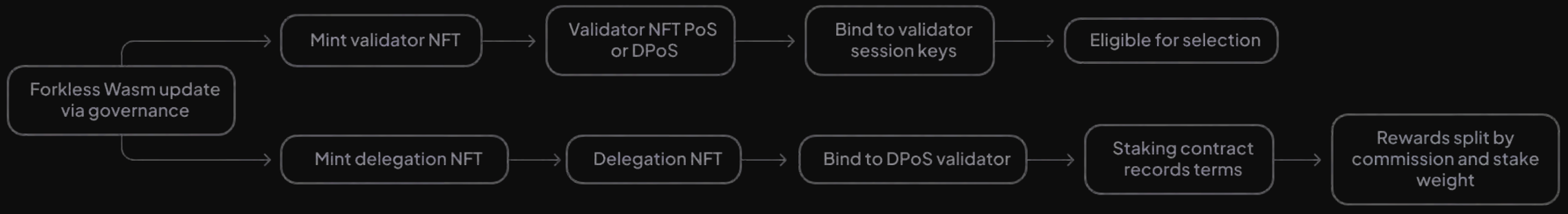
Validator NFTs

Validator NFTs come in two types, each serving a distinct role in the consensus and staking model.

PoS (Proof of Stake) NFTs are held by validators who participate in consensus using their own self-stake exclusively. A PoS NFT grants the holder the right to run a validator node and participate in the block production rotation. The nominal value of the PoS NFT represents the validator's self-stake commitment. Importantly, PoS validators cannot self-stake beyond the initial nominal value of their NFT, which prevents stake concentration and maintains a predictable security model.

DPoS (delegation-enabled) NFTs are held by validators who accept both self-stake and delegated stake from other MOS holders. Although Mosaic’s consensus engine is PoA (Aura-based slot rotation), the "DPoS" label here is shorthand for delegation-enabled validator rights under the NPoS-style economic model. In addition to the rights conferred by a standard PoS NFT, DPoS NFTs enable the validator to enter into staking contracts with delegators who hold Delegation NFTs. The total stake backing a delegation-enabled validator is the sum of the validator's own stake (the NFT nominal value) and all delegated stakes bound to that validator.

The binding and unbinding semantics of Validator NFTs are critical to the protocol's security model. A Validator NFT must be bound to a specific validator node (identified by its session keys) before the validator can participate in consensus. While bound, the NFT cannot be transferred or modified, ensuring that the validator's identity and stake commitment remain stable during active participation. Unbinding requires the validator to first chill (voluntarily exit the waiting pool of eligible validators for future selection) and wait for any active staking contracts to conclude, preventing abrupt withdrawal of stake from the network's security budget.



Mosaic Chain

Delegation NFTs (Delegator NFTs)

Delegation on Mosaic Chain does not require Delegation NFTs. Delegators can delegate MOS directly through staking contracts. Delegation NFTs are an optional instrument used when a delegator wants a transferable, explicitly nominal-value representation of a delegation position.

Delegation NFTs represent transferable, time-bounded staking instruments that allow MOS holders to contribute to network security without operating a validator node. Each Delegation NFT carries a nominal value in MOS and a maximum participation duration expressed as a maximum number of sessions it can participate in once bound to a validator. Delegation NFTs are participation instruments issued during the preliminary investment period and subsequently acquired via transfer where permitted. Their nominal value and maximum session count are attributes of the NFT; typical delegators do not mint Delegation NFTs as part of day-to-day staking. In the standard user flow, MOS holders delegate using MOS-token staking contracts, while Delegation NFTs are used when an instrument-style position is required. The NFT is then bound to a specific delegation-enabled validator through a staking contract (described in Section 5.4). Once bound, the Delegation NFT's stake contributes to the total backing of that validator, increasing the validator's weight in the network's economic security model. The participation clock begins at the moment of binding, not at minting, creating a clear timeline for the delegation commitment.

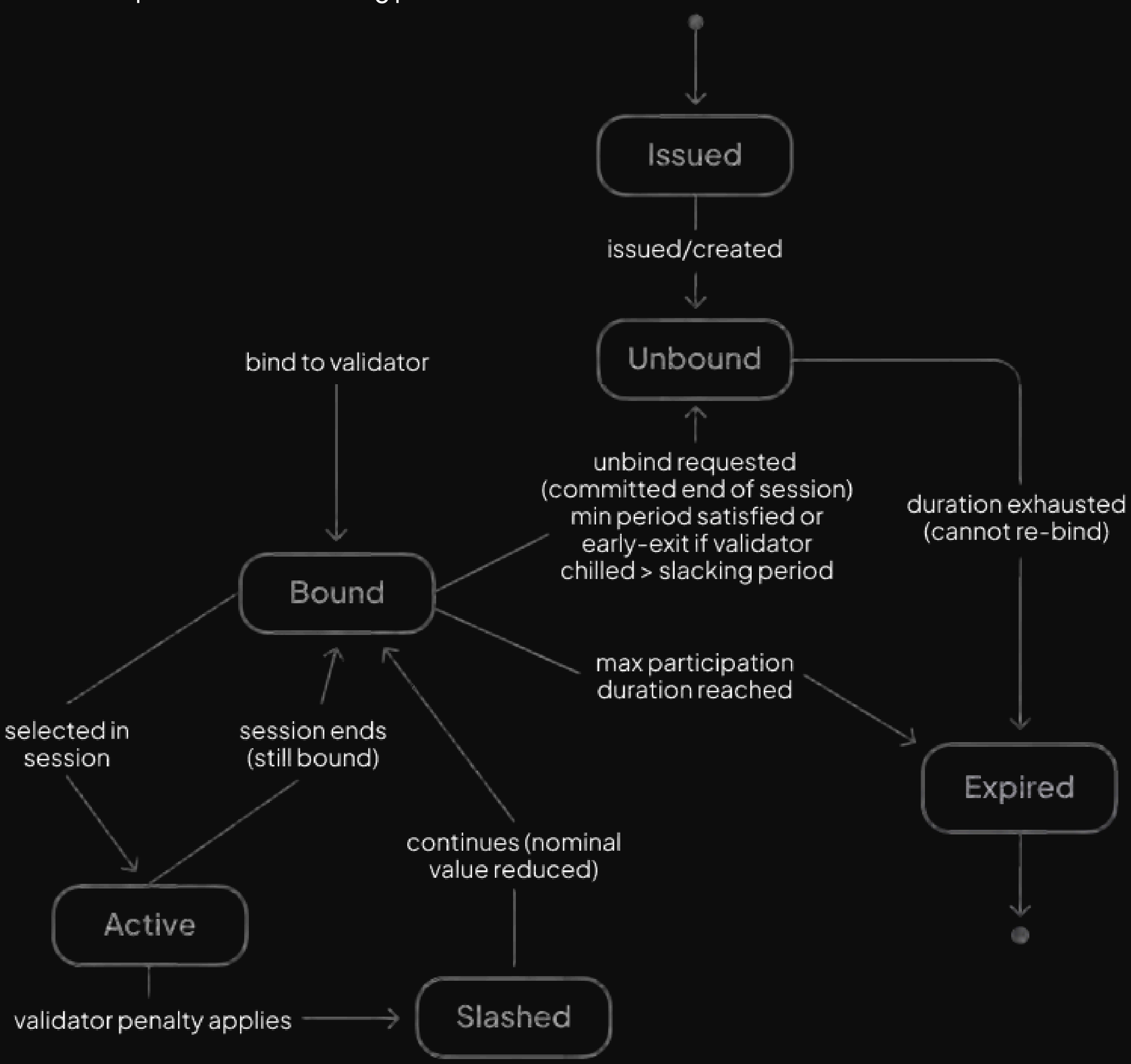
While bound, the Delegation NFT is subject to the same slashing conditions as the validator it backs. If the validator misbehaves, the nominal value of all Delegation NFTs bound to that validator is reduced proportionally. This shared-risk model ensures that delegators have economic incentives to evaluate validator quality and distribute their delegations across reliable operators, contributing to the overall health of the validator set.

After the minimum staking period has elapsed, a Delegation NFT can be unbound from the validator; separately, a Delegation NFT may expire once it reaches its maximum configured participation duration. Delegation NFTs are not redeemable for MOS and do not return any “nominal value” to the holder. If a delegator also has MOS tokens staked under the associated staking contract, those token balances follow the staking contract’s unbinding rules and reflect any slashing that applies to delegated stake. Unbinding is staged when the delegator submits the transaction and is committed at the end of the session. The slacking period is relevant only for early-exit conditions: if the validator remains chilled longer than the slacking period while a minimum staking period is still in force, delegators may unbind despite the remaining lock.

Mosaic Chain

Staking Contracts

When a delegator binds a Delegation NFT to a DPoS validator, the system creates an on-chain staking contract that records the terms of the delegation relationship. Each staking contract specifies the following parameters.

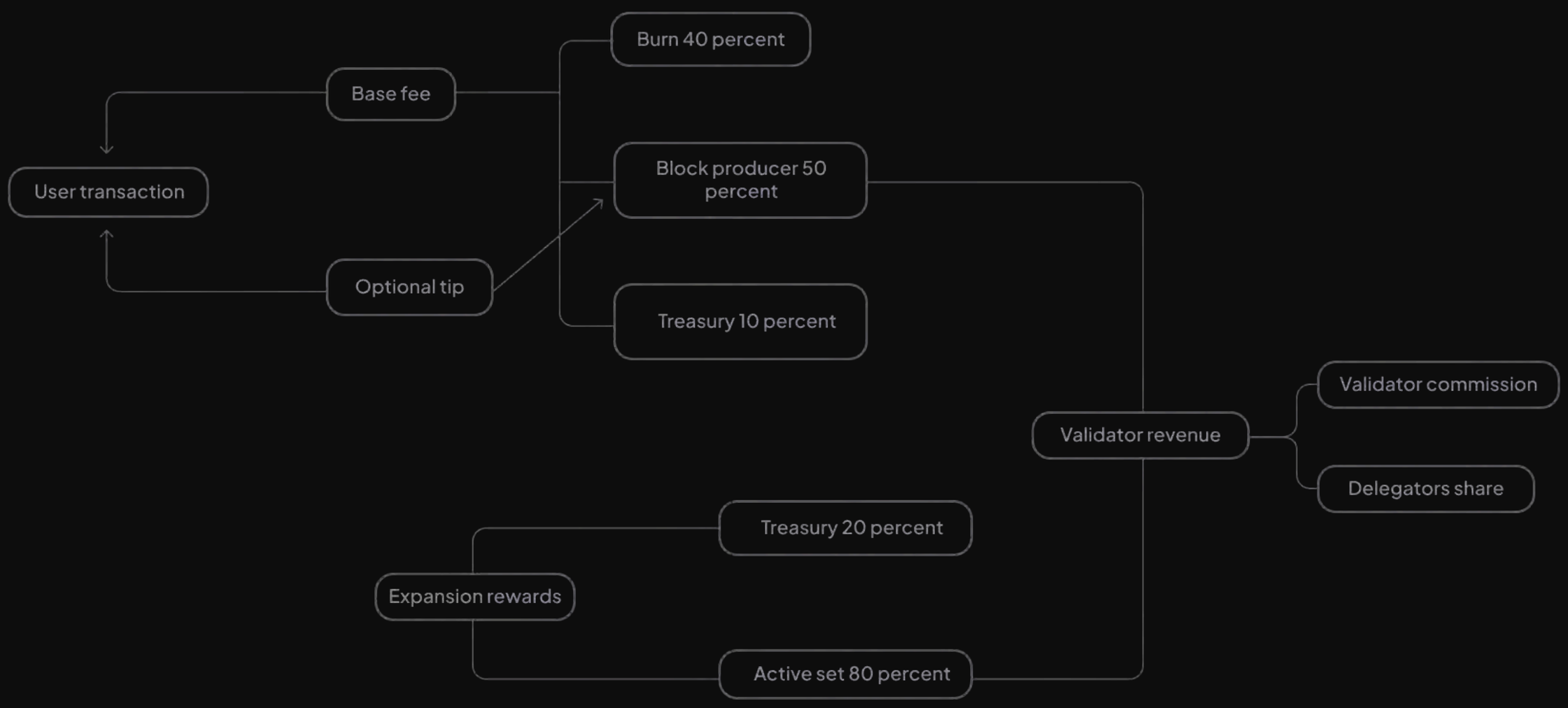


The stake amount may be provided as MOS tokens, Delegation NFT nominal value, or a combination of both, depending on the delegator’s preferred instrument design. Delegation NFTs are not required to delegate MOS. The commission rate, set by the validator (with a global minimum of 1%), determines the fraction of rewards that the validator retains before distributing the remainder to the delegator. The minimum staking period defines how long the delegation must remain active before the delegator can unbind; the global minimum is 28 days (672 sessions), though individual validators may set longer periods. The global minimum staking amount is 50 MOS, and the same minimum applies to additional delegations (top-ups) as well.
Several important behavioral rules govern staking contracts. If a delegator adds additional stake to an existing contract, the terms and lock period reset, preventing incremental additions from circumventing the minimum staking duration. Each validator can have a maximum of 1,000 active staking contracts, which bounds the computational overhead of reward distribution calculations. No single validator can accumulate more than 5% of the total MOS supply in combined self-stake and delegated stake, which prevents excessive stake concentration and limits the economic impact of any single validator's misbehavior.

These constraints are enforced at the protocol level through the staking pallet and cannot be circumvented by validator or delegator actions. The 5% cap is particularly important from a systemic risk perspective: it ensures that even a catastrophic failure of any single validator affects at most a bounded fraction of the network's total economic security.

Rewards and Fees (Mechanics-Level)

Mosaic’s economic security is reinforced by two complementary incentive streams: protocol issuance (“expansion”) paid for consensus work, and transaction-level fees paid by users for resource consumption and prioritization.



Each transaction specifies a fee budget denominated in MOS that is derived from metered execution resources (gas), plus an optional tip. The base fee is charged deterministically by the runtime at execution time; the tip is a user-provided priority fee that signals urgency when block space is scarce.

Base fee routing is enforced at the protocol level: 40% of the base fee is burned, 50% is routed to the block producer (the active validator/collator that authors the block), and 10% is routed to the on-chain treasury. Tips are routed 100% to the block producer.
Burning occurs as part of transaction execution, reducing the effective circulating supply. Treasury receipts accumulate in the treasury account and are governed by Mosaic's budget and burn processes described in the Economics section. Because the split is enforced by the runtime, fee routing is transparent, auditable, and does not rely on off-chain agreements.

Validators therefore earn a revenue stack composed of base fees, tips, and expansion rewards. Delegators receive their pro-rata share of validator earnings net of validator commission, creating a direct link between transaction throughput, block production quality, and long-run network security.

Slashing, Chilling, and Safety Mechanisms

Slashing and chilling are the protocol's primary safety mechanisms for maintaining validator reliability in a large, rotating active set. Slashing imposes an on-chain economic penalty for downtime and protocol violations, while chilling removes a validator's permission NFT from the eligible selection pool (either voluntarily or automatically) to prevent repeated faults from degrading network performance.

What triggers penalties (downtime, protocol violations). Mosaic Chain distinguishes between (1) availability faults and (2) protocol violations. All validators must submit an "I'm online" heartbeat transaction during every session. Active validators must also produce at least one block during the session when they are selected. Failure to meet these requirements is treated as downtime and may trigger slashing. Protocol violations include actions that compromise consensus integrity (for example, producing invalid blocks or other runtime-rule violations).

What gets penalized (validator stake + delegated stake). Slashing applies to the validator's Permission NFT nominal value and to all stake delegated to that validator. This includes Delegation NFT nominal values and any MOS (or other currencies, where applicable) delegated through staking contracts. Slashing is designed to keep incentives aligned: validators bear direct economic cost, and delegators share risk proportional to the validator they chose to back.

Current slash rate and application details. The baseline slash rate is 0.1%, temporarily reduced to 0.0001% on mainnet. Permission NFT nominal value is slashed. Delegator NFT nominal values are slashed as well; within a single contract, any decrease in nominal value is applied first to the Delegation NFT that was delegated earliest, then onward until the full slashing amount is applied. This ordering helps later-delegated NFTs retain their initial nominal value to potentially preserve resale value. All currencies delegated under the contract are subject to the same slashing event.

Auto-chilling behavior and thresholds. A first fault results in slashing but does not automatically chill the validator. If the validator successfully generates a block in its next active session, its state resets to normal. If the validator is slashed again immediately after it was just slashed, the protocol auto-chills the validator to give the operator time to fix configuration or connectivity issues before re-entering the selection pool.

Slacking period and early-exit protections for delegators. The protocol provides a slacking period of 72 sessions as a grace window for temporary issues. If the validator does not return to normal operation within this window, delegators are allowed to freely cancel their staking contracts with that validator (i.e., bypassing the remaining lock period) so they can redelegate to a healthier operator.

These parameters are intended to evolve as mainnet matures. The reduced mainnet slash rate is designed to avoid disproportionate losses during large-scale onboarding and operational tuning, with the expectation that the slash rate can be increased toward the baseline as the validator set stabilizes.

Minimum staking amount. The global minimum staking amount is 50 MOS, and this minimum applies to additional delegations (top-ups) as well. This threshold reduces dust-contract overhead while keeping delegation accessible to smaller holders.

Mosaic Chain

Protocol Extensions and Native Products

The preceding sections describe Mosaic Chain's production backbone: consensus, staking, validator operations, and interoperability. This section specifies three protocol-level features planned for deployment through the forkless runtime upgrade mechanism. Each feature builds directly on existing on-chain primitives and addresses a concrete gap in the current protocol's capabilities. The features are designed to be deployed independently, though they are architecturally complementary and become more powerful in combination.

On-Chain Validator Reputation System (Mosaic Trust Score)

Problem. Delegators on Mosaic Chain currently select validators using raw metrics exposed through the Explorer: uptime percentage, commission rate, total stake, and session participation counts. These metrics are informative but fragmented. A delegator comparing two validators must manually weigh multiple data points across different time horizons, with no standardized way to assess overall reliability. For institutional delegators managing significant MOS positions across dozens of validators, this manual evaluation does not scale. The absence of a protocol-native quality signal also limits the staking vault strategies described in Section 6.2, which need a machine-readable reliability input to make automated delegation decisions.

Design. The Mosaic Trust Score is a composite, on-chain reputation metric computed by a new Reputation Pallet and stored as a first-class attribute on each Validator NFT. The score is an integer ranging from 0 to 1,000, recalculated at every session boundary using data from the preceding 672 sessions (approximately 28 days). This rolling window is long enough to smooth out isolated incidents but short enough to reflect recent behavioral changes. A validator that was reliable six months ago but has degraded over the past two weeks will see its score decline within the current window.

The score is computed from three weighted input categories. Block production fidelity (weight: 40%) measures the ratio of successfully produced blocks to assigned slots when the validator was in the active set. A validator that produces every assigned block scores maximally on this component; missed slots reduce the score proportionally. Availability and responsiveness (weight: 35%) tracks whether the validator's node was reachable and synchronized at session boundaries, as reported by the peer-to-peer health monitoring layer.

This component captures connectivity failures that do not necessarily result in missed blocks but indicate infrastructure fragility. Infraction history (weight: 25%) penalizes validators based on their slashing record within the rolling window. Each slashing event applies a decaying penalty: recent infractions carry more weight than older ones, with the decay function calibrated so that a single minor infraction loses approximately 50% of its penalty weight after 336 sessions (two weeks).

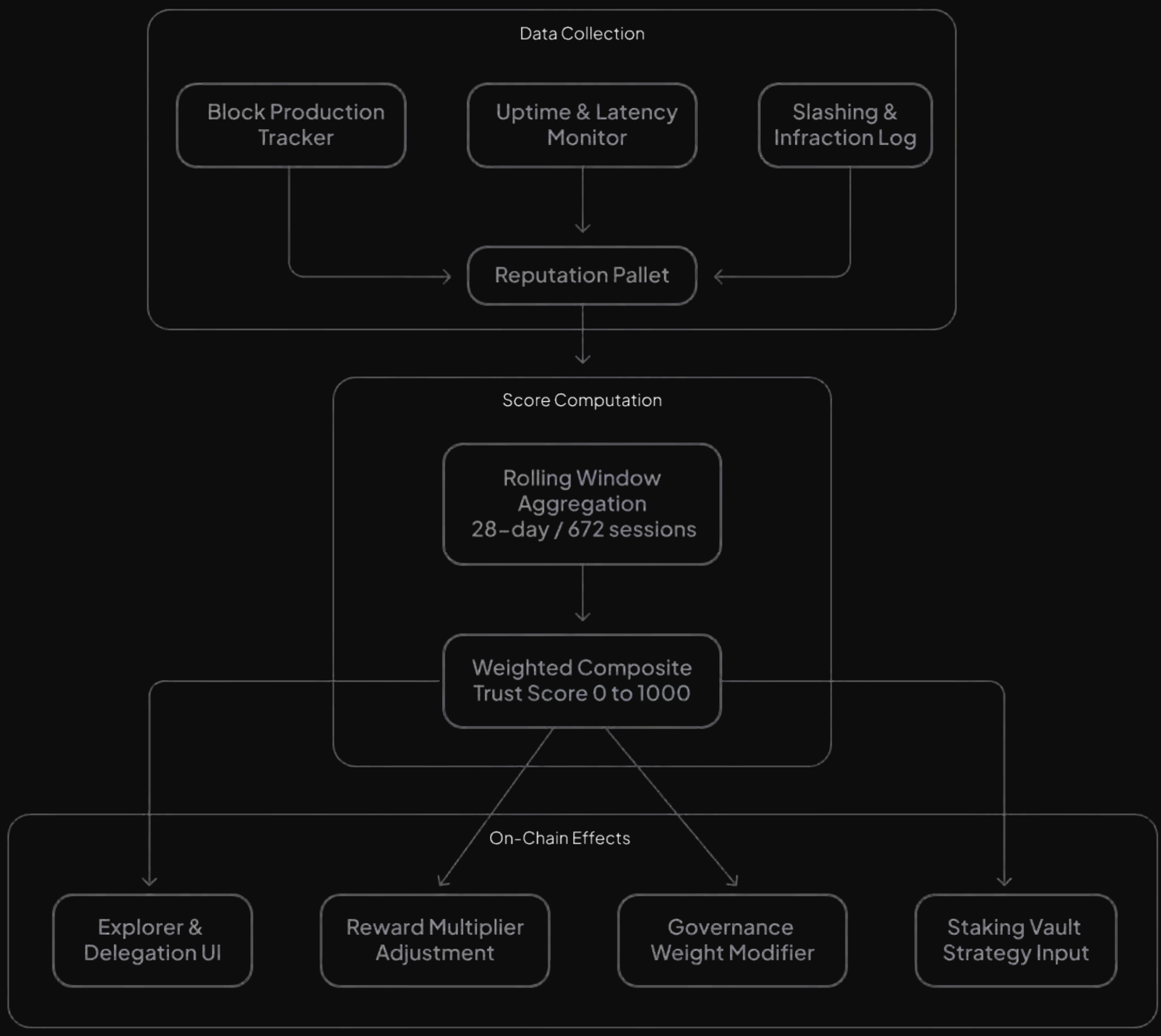
The computation runs entirely on-chain within the Reputation Pallet, using data already recorded by the existing staking, session, and slashing pallets. No off-chain oracles, indexer queries, or external data feeds are required. The pallet reads block production records from the session pallet, availability attestations from the node health reporting subsystem, and infraction records from the slashing pallet. Each input is normalized to a 0–1,000 sub-score, then combined using the fixed weights above to produce the final composite. The weights are governance-adjustable parameters, allowing the community to shift emphasis between production fidelity, availability, and infraction history as the network's priorities evolve.

On-chain primitives. The Reputation Pallet introduces a persistent storage map (ValidatorTrustScores) keyed by validator account, storing the current composite score and the per-component sub-scores. A second storage map (ReputationHistory) retains the raw input snapshots for the rolling window, enabling the pallet to incrementally update scores at session boundaries rather than recomputing from scratch. The pallet exposes a read-only RPC method (reputation_trustScore) that external clients can query for any validator, and emits a TrustScoreUpdated event at every session boundary containing the validator account and new score. The score is also written as a mutable NFT attribute on the Validator NFT through the NFT-Permission Pallet's attribute interface, making it queryable in the same way as other NFT properties.

Economic and security effects. The Trust Score creates a feedback loop between validator behavior and economic outcomes. In the initial deployment, the score is purely informational: it is displayed on the Explorer and available to staking vault contracts but does not directly affect reward calculations. In a subsequent phase (gated by a governance vote), the protocol may introduce a reward multiplier that adjusts validator rewards within a narrow band (e.g., 0.95× to 1.05×) based on the Trust Score. This multiplier is deliberately bounded to prevent extreme reward concentration among high-scoring validators, which would undermine the equitable participation model. The score also provides a natural input for the programmable staking vaults described in Section 6.2, enabling automated strategies that shift stake toward consistently reliable operators.

Failure modes and mitigations. The primary risk is score manipulation. A validator could attempt to inflate its score by strategically chilling during periods when it anticipates poor performance, thereby excluding bad data from the rolling window. The pallet mitigates this by including voluntary chilling frequency as a negative signal in the availability component: validators that chill and unchill frequently receive lower availability sub-scores, and the frequency threshold is calibrated so that legitimate maintenance windows (one or two per rolling period) are not penalized. A second risk is that the score creates a “rich get richer” dynamic where high-scoring validators attract all delegation, reducing decentralization. The bounded reward multiplier and the existing 5% per-validator stake cap structurally limit this concentration. Finally, the 28-day rolling window means that a newly onboarded validator starts with a low or default score until it accumulates sufficient operating history. The pallet addresses this by assigning new validators a neutral baseline score of 500 rather than zero, preventing cold-start penalization while still requiring performance to earn higher scores.
Interoperability touchpoints. Trust Scores are exposed through the standard RPC interface and as NFT attributes, making them accessible to any on-chain or off-chain consumer.

Cross-chain applications on other parachains could query Mosaic validator Trust Scores via XCM to make informed decisions about which Mosaic validators to rely on for cross-chain operations. The score format (integer 0–1,000 with documented component weights) is designed to be interpretable without Mosaic-specific context, facilitating integration with third-party dashboards, analytics platforms, and cross-chain aggregators. The governance mechanism retains authority over the component weights, the rolling window length, and the neutral baseline score, ensuring that the reputation system can be recalibrated as the network’s validator set grows and operational patterns mature. Future iterations may introduce additional input signals, such as geographic diversity metrics or client software diversity, provided they can be verified on-chain without trusted third-party attestations.



Programmable Staking Vaults

Problem. The existing staking model on Mosaic Chain is manual and static. A delegator binds a Delegation NFT or MOS tokens to a specific validator, earns rewards, and periodically reassesses the delegation. If a validator's performance deteriorates, the delegator must notice the decline, decide to act, initiate an unbinding transaction, wait for the minimum staking period or cooldown, and then re-delegate to a different validator. Each step requires active attention and on-chain transactions with associated fees and timing constraints. For delegators with positions spread across many validators, this maintenance burden is substantial. Earned rewards also sit idle unless the delegator manually re-stakes them, forgoing compounding. The result is that delegation on Mosaic Chain, like most proof-of-stake networks, is a "set and forget" activity for most participants, which means suboptimal capital allocation and weaker network security when stake lingers with underperforming validators.

Mosaic Chain

Design. Programmable Staking Vaults are on-chain smart contracts that accept MOS deposits from delegators and execute pre-defined staking strategies automatically. Each vault is parameterized by a strategy configuration that the vault creator (or the delegator, depending on the vault type) specifies at deployment. The vault contract holds the deposited MOS in custody, delegates it across a set of validators according to the strategy rules, and manages the full lifecycle of delegation contracts on the delegator's behalf: binding, unbinding, reward collection, re-delegation, and compounding.

Three core strategy primitives are supported, and they can be combined within a single vault. The rebalance primitive redistributes the vault's total stake across validators according to target allocation weights at each session boundary. If a vault targets a 40/30/30 split across three validators and drift causes the actual allocation to deviate beyond a configurable threshold (default: 5%), the vault triggers unbinding from over-allocated validators and re-delegation to under-allocated ones. The compound primitive automatically re-stakes earned rewards at session boundaries. Rather than accumulating rewards in an idle balance, the vault adds them to the existing delegation positions proportionally, increasing the effective yield over time. Compounding frequency is configurable: every session, every N sessions, or when accumulated rewards exceed a specified threshold. The migrate primitive monitors the Trust Scores (Section 6.1) of validators in the vault's delegation set and automatically initiates stake migration when a validator's score drops below a configurable floor. Migration respects the minimum staking period by queuing the unbinding and executing it at the earliest permitted time, then re-delegating the freed stake to the highest-scoring eligible validator not already in the vault's set.

On-chain primitives. A new Vault Pallet manages the lifecycle of staking vaults. The pallet introduces a VaultConfig storage struct that records the strategy parameters, target validator set, allocation weights, Trust Score floor, compounding frequency, and rebalance threshold. Each vault is identified by a unique on-chain account derived from the vault's creation transaction, and the vault account holds the deposited MOS. The pallet exposes extrinsics for vault creation (create_vault), deposits (deposit_to_vault), withdrawals (withdraw_from_vault), and strategy updates (update_strategy). Withdrawals are subject to a notice period equal to the longest minimum staking period among the vault's active delegation contracts, preventing instant liquidity extraction that would disrupt active delegations.
At session boundaries, the Vault Pallet's on_session_end hook evaluates each active vault against its strategy parameters. The pallet reads the current delegation state from the Staking Pallet, Trust Scores from the Reputation Pallet, and pending rewards from the reward distribution subsystem. It then computes the set of necessary actions (rebind, unbind, compound, migrate) and executes them atomically within the session transition.

This batched execution ensures that vault operations do not create mid-session state inconsistencies. The per-vault computation is bounded by the maximum number of validators in any single vault's delegation set, which is capped at 20 to keep session-boundary overhead predictable.

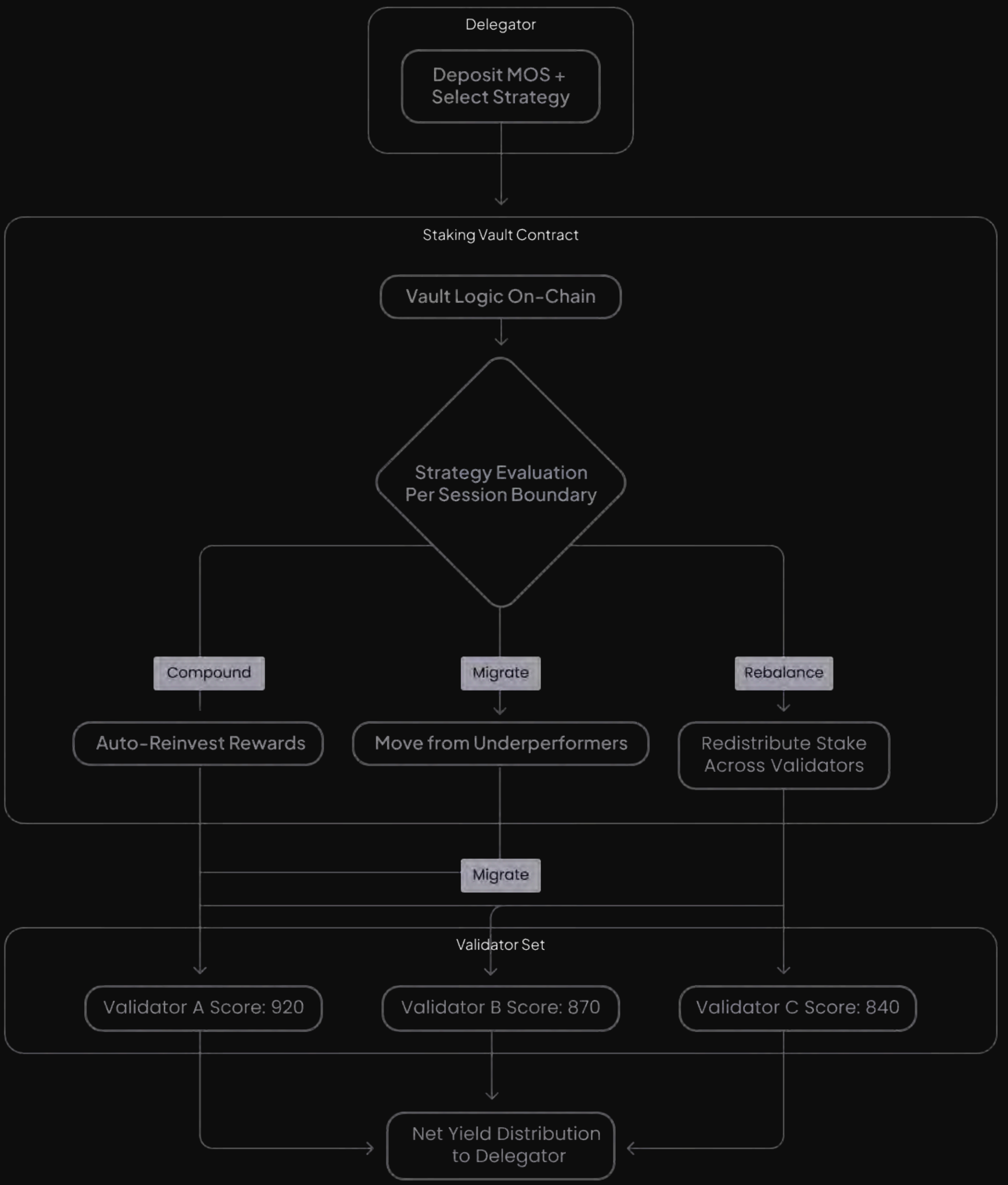
Economic and security effects. Staking vaults improve network security in two ways. First, by automating compounding, they increase the effective staked supply over time without requiring active delegator participation, raising the economic cost of consensus attacks. Second, by automating migration away from underperforming validators, they create a direct economic consequence for poor validator behavior that operates faster than manual delegator responses. Validators that maintain high Trust Scores retain vault-delegated stake; validators that slip lose it. This creates a market-like pressure toward operational quality that complements the protocol-level penalties of the slashing system.

The vault model also introduces a new category of ecosystem participant: the vault strategist. Strategists create vaults with specific parameter configurations and make them available for public deposits, earning a strategy fee (a small percentage of rewards generated by the vault). This creates a competitive market for staking strategy quality, where strategists are incentivized to design allocation rules that maximize risk-adjusted yield. The strategy fee is enforced on-chain by the Vault Pallet and deducted before distributing rewards to depositors.

Failure modes and mitigations. The principal risk is correlated migration: if many vaults use similar Trust Score thresholds, a temporary dip in a popular validator's score could trigger simultaneous mass unbinding, creating a cascade that destabilizes the validator's position and potentially cascades to its delegators. The pallet mitigates this with a per-session migration rate limit: no more than 10% of a validator's total delegated stake can be migrated away in any single session, smoothing exits over multiple sessions. A second risk is vault contract bugs. Because vault strategies execute automatically at session boundaries, a misconfigured strategy could repeatedly churn stake between validators, incurring unnecessary transaction costs. The pallet enforces a minimum rebalance interval (default: 24 sessions / approximately one day) and a maximum churn rate per evaluation cycle to bound this behavior. The 20-validator cap per vault further limits the computational surface for any single vault's session-boundary logic.

Interoperability touchpoints. Staking vaults are composable with the yield routing feature described in Section 6.3. A vault can be configured to route a portion of its compounded rewards through the Yield Router rather than re-staking all of them, enabling vaults that blend staking yield with cross-chain DeFi yield. Vault state (total deposits, current allocations, historical performance) is queryable through the standard RPC interface, enabling third-party portfolio trackers and analytics platforms to display vault positions alongside direct staking positions.

Mosaic Chain



Mosaic Chain

Cross-Chain Yield Routing via XCM (Mosaic Yield Bridge)

Problem. MOS stakers face a binary choice: lock tokens for staking rewards or keep them liquid for DeFi participation on other parachains. This is a capital efficiency problem common to proof-of-stake networks. Staked tokens secure the network but cannot simultaneously earn yield from lending protocols, liquidity pools, or other DeFi primitives on parachains like Acala, Moonbeam, or Astar. Liquid staking derivatives (LSDs) are the standard industry response, but they introduce counterparty risk, require external protocol trust, and create complex repricing dynamics when the derivative depegs from the underlying asset. For a finance-focused chain that emphasizes legible security and on-chain transparency, routing yield through an opaque third-party LSD protocol undercuts the design philosophy.
Design. The Mosaic Yield Bridge is a protocol-native feature that allows stakers to route a configurable fraction of their staking position to vetted DeFi yield opportunities on other Polkadot parachains, using XCM for trustless cross-chain transport. The mechanism does not unstake the tokens in the traditional sense. Instead, the Yield Router Pallet mints a synthetic receipt token (yMOS) that represents the routed portion, while the underlying MOS remains counted toward the network's economic security budget. The receipt token is non-transferable and non-composable—it exists solely as an accounting artifact within the Yield Router Pallet, not as a tradeable derivative.

The routing flow works as follows. A staker with an active staking position specifies the fraction of their staked MOS to route (subject to a protocol-enforced maximum of 30% of their position) and selects a target yield opportunity from a whitelist maintained by governance. The Yield Router Pallet verifies that the staker's remaining un-routed stake still exceeds the minimum staking threshold, then mints the corresponding yMOS receipt and initiates an XCM transfer of the specified MOS amount to the target parachain. On the target parachain, the transferred MOS is deposited into the specified DeFi protocol (e.g., a lending pool on Moonbeam, a liquidity pair on Acala, or a dApp staking contract on Astar) through a pre-configured XCM program that encodes the deposit instructions.
When the staker wishes to recall the routed funds, or when the yield opportunity's term expires, the Yield Router initiates a withdrawal through XCM. The target parachain returns the principal plus accrued yield to Mosaic Chain. The Yield Router Pallet burns the yMOS receipt, credits the returned MOS (principal plus yield) to the staker's account, and the staker can either re-stake the yield, route it to another opportunity, or withdraw it. The staker's original staking position remains intact throughout the process; the receipt token ensures that the staked balance is never double-counted for security purposes.

On-chain primitives. The Yield Router Pallet introduces several new storage structures. A YieldOpportunityRegistry stores the governance-approved whitelist of target parachains, protocols, and XCM programs, each with a risk tier and maximum routing capacity. A RoutedPositions map tracks each staker's active routing: the amount routed, the target opportunity, the yMOS receipt identifier, and the routing timestamp.

A PendingSettlements queue manages in-flight XCM transfers during the recall process. The pallet exposes extrinsics for routing (route_to_yield), recalling (recall_from_yield), and emergency withdrawal (emergency_recall), plus governance-only extrinsics for managing the opportunity whitelist (add_yield_opportunity, remove_yield_opportunity, update_risk_tier).

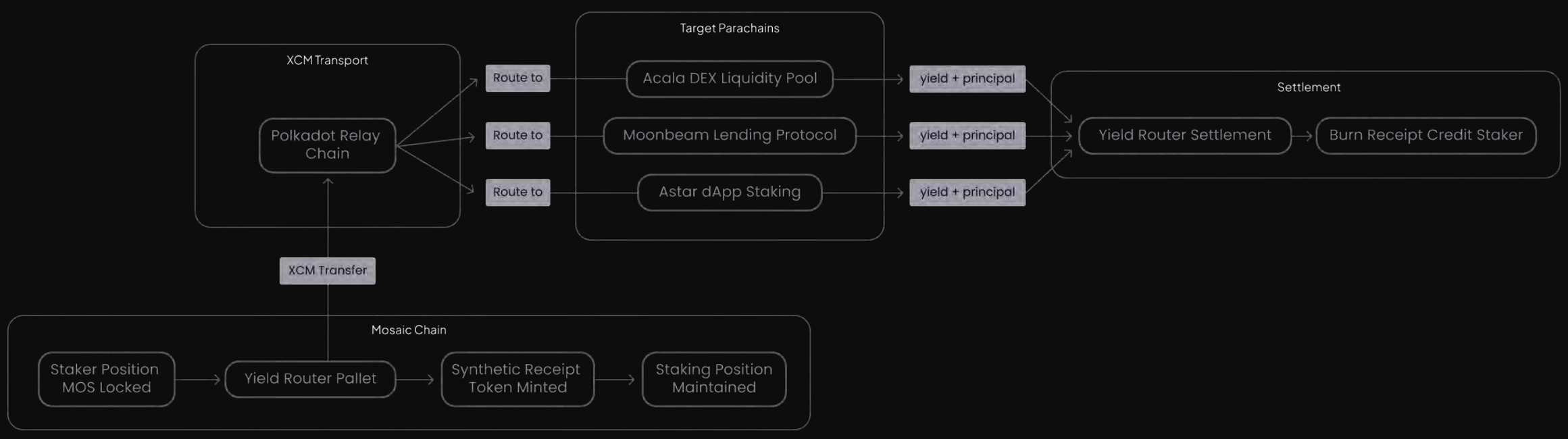
Economic and security effects. The Yield Bridge increases effective capital efficiency for MOS stakers without degrading network security. Because the routed MOS is capped at 30% of any staker's position, at least 70% of staked MOS remains fully committed to consensus security at all times. The yMOS receipt token's non-transferable, non-composable nature prevents the derivative dynamics that plague liquid staking tokens: there is no secondary market for yMOS, no depeg risk, and no leverage pathway. The 30% cap is a governance-adjustable parameter that can be tightened if the network's security budget needs strengthening or relaxed as the total staked supply grows.
For the broader Polkadot ecosystem, the Yield Bridge creates a capital flow channel from Mosaic Chain's staking base into DeFi protocols on other parachains. This increases TVL and liquidity for target protocols while giving MOS stakers access to diversified yield streams. The fee structure routes a small protocol fee (default: 2% of yield) to the Mosaic treasury, creating a new revenue stream that grows with cross-chain DeFi activity.

Failure modes and mitigations. The most serious risk is smart contract failure on the target parachain: if the DeFi protocol where MOS is deposited suffers an exploit, the routed funds could be lost. The governance-managed whitelist is the primary mitigation. Only protocols that have undergone formal security audits, have sustained mainnet operation for a minimum period (default: six months), and carry sufficient insurance or safety modules are eligible for the whitelist. Each whitelisted opportunity carries a risk tier (low, medium, high) that determines its maximum routing capacity; high-risk opportunities accept less total routed MOS. The emergency_recall extrinsic provides a governance-triggered mechanism to forcibly withdraw MOS from a target protocol if a vulnerability is detected, bypassing the normal recall flow. XCM message delivery failures are handled by the PendingSettlements queue, which retries failed transfers at configurable intervals and escalates to governance attention if retries are exhausted.

Interoperability touchpoints. This feature is inherently cross-chain and depends entirely on Polkadot's XCM infrastructure. The Yield Router constructs XCM programs for each whitelisted opportunity that encode the deposit, withdrawal, and emergency recall logic specific to the target protocol. These XCM programs are stored on-chain and auditable. As Polkadot's XCM capabilities expand (e.g., XCMv4 improvements to error handling and fee estimation), the Yield Router's XCM programs can be updated through governance without modifying the core pallet logic. Integration with the Staking Vaults (Section 6.2) enables automated yield routing as part of a vault's compounding strategy: a vault can be configured to route a percentage of compounded rewards through the Yield Bridge rather than re-staking all of them, giving vault depositors exposure to blended staking-plus-DeFi yield.

The combination of the three features described in this section—Trust Scores informing vault strategies, vaults automating delegation and compounding, and the Yield Bridge routing idle capital cross-chain—creates a vertically integrated staking infrastructure that is difficult to replicate through external third-party protocols.

Each feature functions independently, but their combined effect is a capital management stack native to the Mosaic Chain protocol.



Mosaic Chain

Economics (MOS)

MOS Utility

The Mosaic Coin (MOS) is the native token of the Mosaic Chain network, serving as the fundamental unit of value across all protocol operations. MOS fulfills five distinct utility functions within the ecosystem.

Transaction fees. All on-chain operations, including transfers, staking transactions, governance actions, and smart contract interactions, require MOS to pay for computational and storage costs. The fee model ensures that network resources are allocated efficiently and that spam or denial-of-service attacks carry an economic cost.

Staking and security. MOS tokens are locked as stake by validators (through NFT nominal values) and delegators (through Delegation NFTs and staking contracts) to secure the network. The total value of staked MOS represents the economic security budget of the network, which determines the cost of attacking the consensus mechanism.

Rewards. Block rewards and a share of transaction fees are distributed to validators and their delegators as compensation for securing the network. The reward mechanism incentivizes honest participation and penalizes behavior that undermines network reliability.
Governance. Token-weighted on-chain governance is planned for a future runtime upgrade. At present, MOS is used as a proposal deposit mechanism, while protocol and fund decisions are made by designated on-chain committees (collectives) that approve proposals by supermajority (2/3).

Treasury. A portion of block rewards and transaction fees flows into the on-chain treasury, which funds ecosystem development, marketing, security audits, and other initiatives approved through the governance process.

Supply and Denominations

The maximum supply of MOS is fixed at 2,000,000,000 tokens. This hard cap is enforced at the protocol level and cannot be modified without a governance-approved runtime upgrade. The smallest indivisible unit of MOS (analogous to satoshis in Bitcoin or wei in Ethereum) is defined as 1 tile, where 1 MOS = 10^18 tiles. All on-chain balances, staking amounts, and fee calculations operate at the tile level of precision.
Definitions (supply terms). For clarity, Mosaic Chain uses the following supply terms. “Pretokens” are tokens distributed as part of user onboarding (delivered over time via vesting). “Premined” refers to tokens minted at launch excluding pretokens. “Total supply” refers to all minted tokens (whether locked or unlocked).

Mosaic Chain

“Circulating supply” refers to minted tokens that are not locked, excluding the staking incentive pool. “Max supply” refers to the theoretical maximum token supply. A Token Generation Event is the session from which block rewards begin accruing.
Network time constants. Mosaic Chain targets a 6-second block time, organized into one-hour sessions. Under these assumptions, one on-chain year corresponds to 5,259,600 blocks.

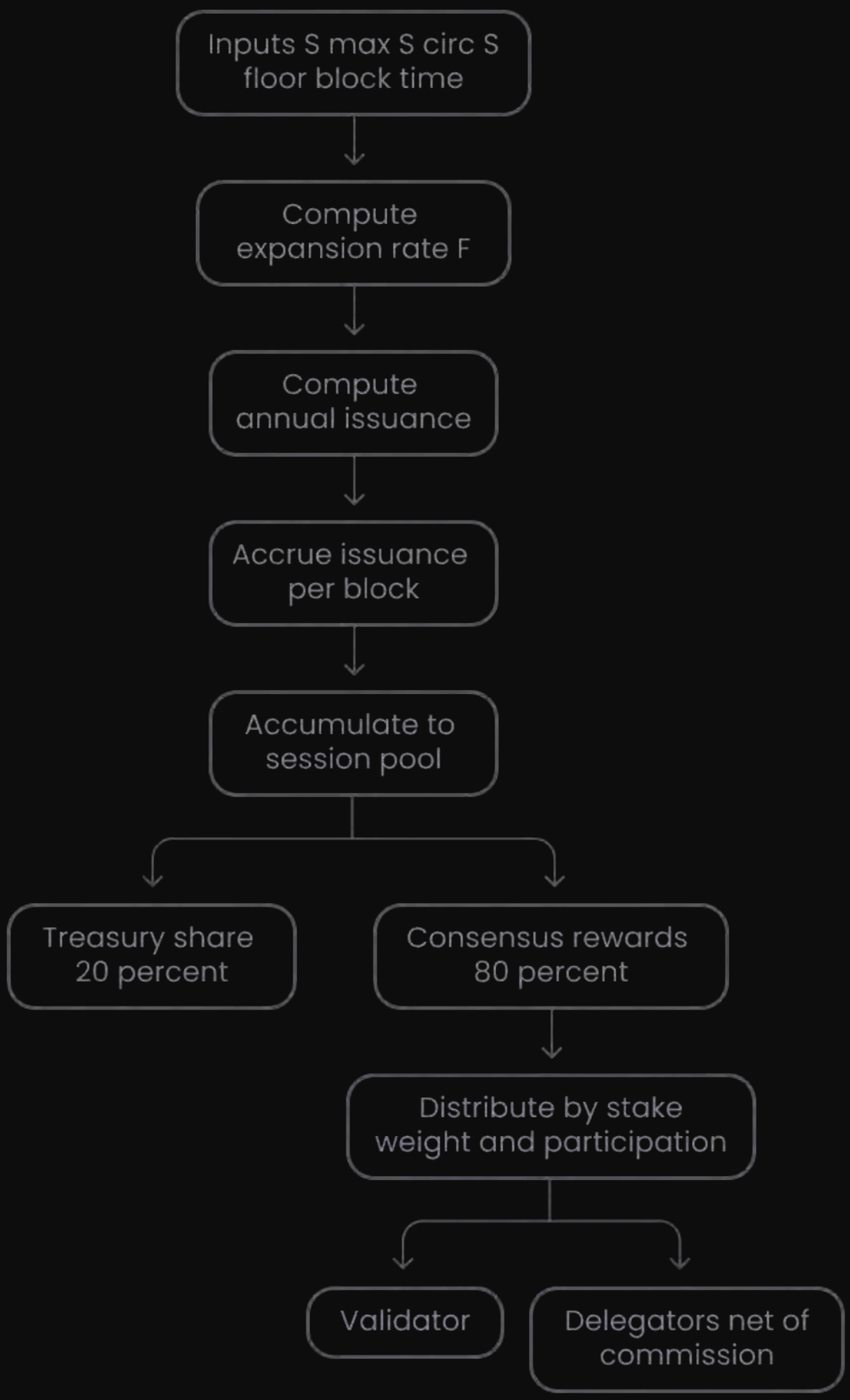
Premined token allocation. Tokens minted at launch (excluding pretokens) are reserved across the following categories:

Purpose	Amount (MOS)
Staking Incentive	~500,000,000
Pretokens	~100,000,000
Treasury	10,000,000
Development and Innovation	24,000,000
Financial and Liquidity Funds	20,000,000
Marketing and Community Building	20,000,000
Team and Advisors Support	8,000,000
Security and Compliance	4,000,000
Education and Documentation	2,400,000

The total supply at any point in time includes all minted tokens, whether locked, vested, staked, or freely circulating. The circulating supply excludes tokens that are locked in vesting schedules and the staking incentive pool. These distinctions are important for accurately assessing the network's economic dynamics and the effective liquidity available in the market.

Issuance and Reward Model

Mosaic Chain follows a capped-supply issuance schedule that begins with an inflationary phase and asymptotically approaches zero issuance as circulating supply approaches the maximum supply.



Given a target block time of 6 seconds (5,259,600 blocks per on-chain year), the protocol expresses expansion as an average block reward function. Let max_supply be the maximum MOS supply and circulating_supply be the circulating supply (with a floor of 100,000,000 MOS).

The expansion factor is defined as:
F = 12.5 × sqrt((max_supply − max(circulating_supply, 100,000,000)) / (max_supply − 100,000,000))

Because rewards are distributed based on the currently active validator set each session, the protocol applies a session correction that accounts for the number of bound validators (B) and the number of active validators (A). The corrected reward rate for a session is proportional to (B × F / A), and the session payout scales by the session length (L). At the end of each session, active validators are rewarded for the blocks they produced in the prior session.

From each session's issuance pool, the protocol skims a fixed share, T = 20%, to the treasury. The remaining 80% is allocated to the active validator set as consensus rewards. Consensus rewards are distributed across the active validator set proportionally based on effective bonded stake and participation (e.g., block production and availability). Each validator then shares its rewards with delegators according to stake weight, after applying a validator-defined commission rate.
As expansion declines over time, security funding increasingly shifts toward fee-based revenue and protocol-level utility, reducing reliance on dilution while preserving predictable incentives during the network's growth phase.

Vesting and Vesting-to-Freeze

Pretokens and vesting. Pretokens are distributed as part of onboarding and are delivered over time via a vesting schedule rather than being made available immediately. The duration of the vesting schedule depends on the amount endowed and unlocks linearly over approximately 2–4 years, computed by a formula determined before onboarding begins on-chain.

Vesting-to-freeze (stakeable locked assets). Mosaic Chain supports an option to convert still-vesting pretokens into stakable MOS that remains locked until the end of the original vesting schedule. These locked tokens can be staked freely but cannot be used for transfers or other purposes until the vesting schedule completes. This mechanism allows participants to contribute to network security and earn staking rewards without increasing circulating supply.

Mosaic Chain

Staking Incentive Pool

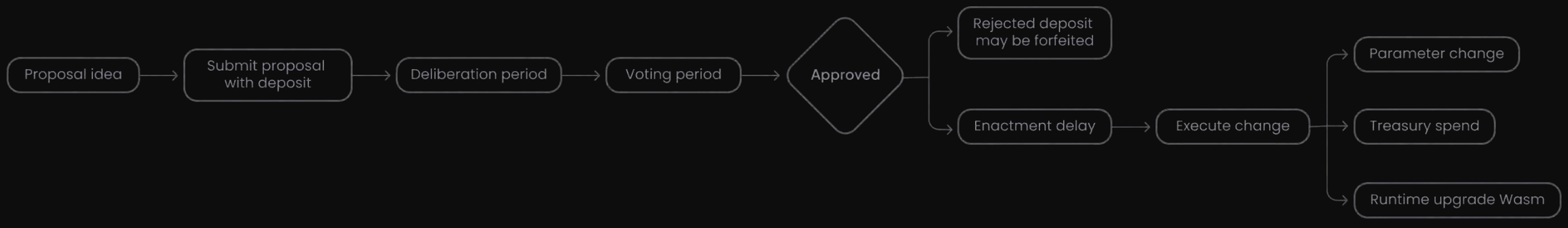
Mosaic Chain includes a staking incentive pool designed to reward long-term staking behavior. After the Token Generation Event, staking incentive rewards are intended to be claimable approximately once per year from the fourth through the eighth year. On each payout, a predefined portion of the staking incentive pool is distributed to eligible stakers based on a score that grows with both stake size and time staked. Claimed incentive rewards are received under a one-year vesting schedule and may be converted to stakable frozen assets.

Because incentive distribution parameters may evolve, the precise timing and amounts distributed per payout are governance-parameterized.

Governance and Upgrades

Governance Scope

Mosaic Chain's governance is currently committee-based. MOS holders do not yet vote on-chain to control protocol parameters, treasury spending, or upgrades. Instead, MOS is used as a proposal deposit mechanism, and proposals are evaluated and approved by designated on-chain committees (collectives). Token-weighted governance is planned for a future runtime upgrade (referred to in the roadmap as Governance 1.0). Open, token-weighted governance is not yet available on Mosaic Chain at the time of writing.
Parameter adjustments encompass changes to protocol constants such as session length, minimum staking amounts, commission rate floors, slashing percentages, and validator subset selection targets. These parameters directly affect the network's operational behavior and economic model. Under the current model, parameter changes are submitted as proposals with a MOS deposit and are approved by the relevant committee by supermajority (2/3) before enactment.



Treasury and fund spending governs the allocation of funds accumulated through block reward skimming and transaction fee routing. Proposals for spending include a requested amount, an intended purpose, and a MOS deposit. Mosaic supports a treasury plus dedicated funds; each fund is governed by its own committee rather than by token-weighted voting. A proposal is approved when the governing committee reaches the required supermajority (2/3), after which the requested amount becomes available for disbursement at the end of the current spending period (28 days / 672 sessions). To encourage active use of treasury and fund balances, periodic burn mechanisms can reduce idle balances according to configured schedules.

Runtime upgrades allow the governance process to deploy new or modified on-chain logic. Because the runtime is stored as Wasm bytecode on-chain, a successful governance vote to upgrade the runtime takes effect automatically at a specified block height. This mechanism enables the protocol to evolve its capabilities, fix bugs, and adapt to changing requirements without the coordination costs and community-splitting risks of hard forks.
The governance model supports a proposal lifecycle that includes submission with deposit, a review period, a committee voting period requiring a 2/3 supermajority, and an enactment period that provides a buffer between approval and implementation. This multi-phase process balances responsiveness with caution, ensuring that significant protocol changes receive adequate scrutiny while the network transitions toward token-weighted governance in a future upgrade.

Mosaic Chain

Forkless Upgrades

The forkless upgrade mechanism inherited from Substrate is one of Mosaic Chain's most important architectural features for a finance-oriented blockchain. Traditional blockchain upgrades through hard forks require extensive off-chain coordination: every validator must download and install new software within a narrow time window, and any validators that fail to upgrade risk being stranded on an incompatible chain fork. This process is slow, operationally risky, and can fragment the network community.

Mosaic Chain's forkless upgrades work differently. The blockchain's runtime logic is stored as WebAssembly bytecode within the chain state. When a runtime upgrade is approved by the network’s governance authority (currently committee-based), the new Wasm binary is submitted as an on-chain transaction. After that change is finalized by the relay chain and a small security window has passed, all validators execute the updated runtime automatically, because the consensus rules specify that validators must execute the most recent on-chain runtime. No software download, no manual installation, no coordination window.

This upgrade mechanism has a direct impact on the safety properties relevant to financial applications. It eliminates the risk of chain splits caused by asynchronous validator upgrades. It ensures that all participants are always executing the same protocol rules. And it enables rapid response to critical issues, since bug fixes can be deployed through the same governance-and-upgrade pipeline without waiting for a manual update cycle to propagate across the validator set.

Mosaic Chain maintains a conservative safety posture around upgrades. Runtime changes undergo thorough testing on devnet before mainnet proposal. The governance process includes deliberation and enactment periods that allow the community to evaluate proposed changes. Backward compatibility is prioritized, and state migrations (when necessary) are implemented as part of the upgrade transaction. A full rollback, while not trivially automated, is technically feasible through the same governance mechanism should a deployed upgrade prove problematic.

Mosaic Chain

Interoperability

Relationship to Polkadot

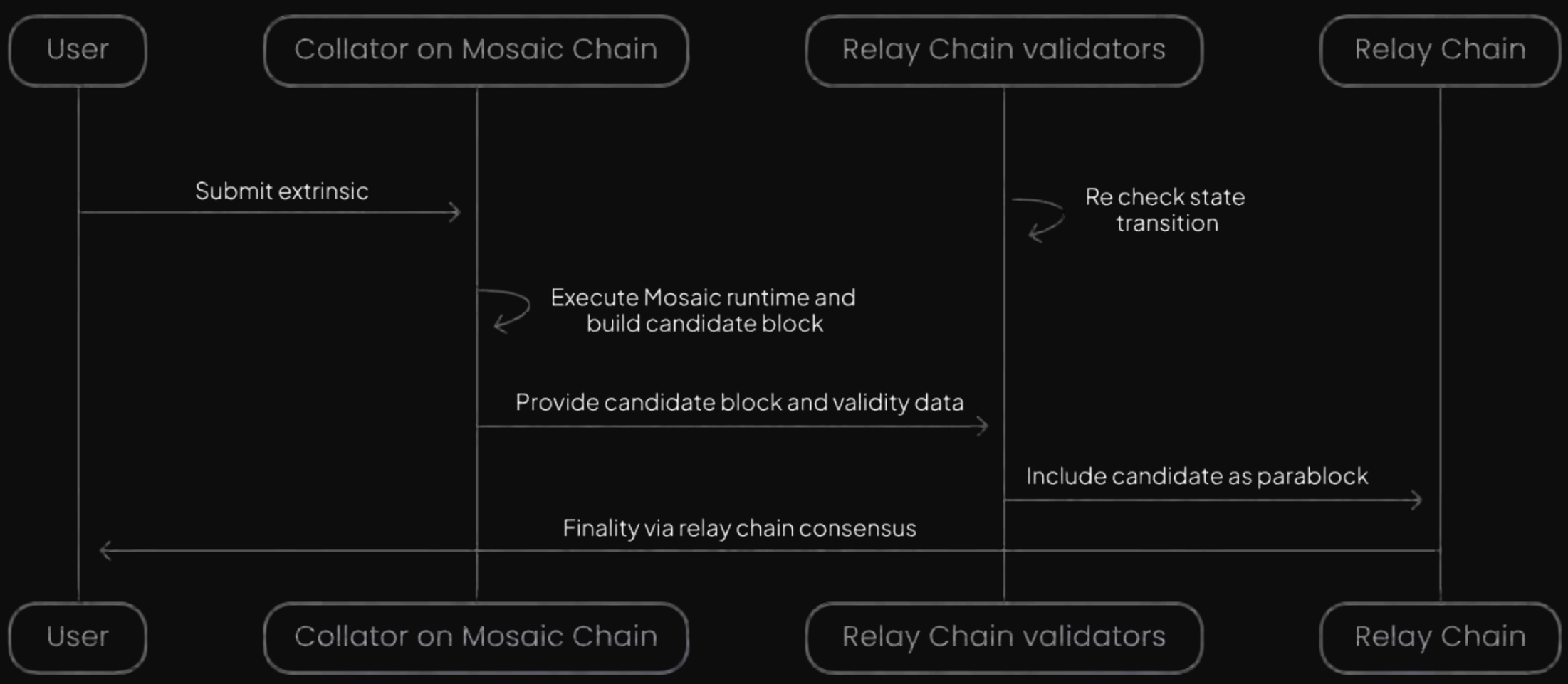
Mosaic Chain is built on Substrate and operates within the Polkadot ecosystem. Polkadot functions as a Layer 0 blockchain that provides shared security and interoperability infrastructure for connected Layer 1 chains (parachains). Unlike bridge-based interoperability models where each chain maintains independent security and relies on trusted intermediaries for cross-chain communication, Polkadot's relay chain model provides a fundamentally different security guarantee: parachain blocks ("parablocks") are validated and finalized by Polkadot's own validator set, ensuring that cross-chain messages are backed by the economic security of the entire Polkadot network.

For Mosaic Chain, this relationship provides several concrete benefits. Scalability is partially offloaded to the relay chain architecture: by connecting as a parachain, Mosaic Chain becomes part of a sharded network that distributes transaction execution across multiple blockchains, preventing any individual parachain from becoming a throughput bottleneck. Finality inherits from Polkadot's GRANDPA finalization mechanism, providing probabilistic guarantees that are substantially stronger than those available to standalone chains. Security is augmented by Polkadot's shared security model, which pools the economic security of all parachains rather than requiring each chain to bootstrap its own security independently.

Mosaic Chain won a parachain slot and will continue to operate within the Polkadot ecosystem, following protocol changes as the Polkadot network evolves. This commitment to ecosystem alignment ensures that Mosaic Chain benefits from ongoing improvements to the relay chain infrastructure, including scalability enhancements, governance upgrades, and new cross-chain communication capabilities.

Relationship to Polkadot

As a Polkadot parachain, Mosaic Chain separates local execution from global finality. Mosaic’s own validator operators run parachain infrastructure as collators: they maintain the full Mosaic state, collect user transactions (extrinsics), execute them against the Mosaic runtime, and assemble candidate blocks.
For each candidate block, the collator produces the accompanying validity data (state transition witness) required for verification by Polkadot Relay Chain validators. Relay Chain validators re-check the candidate’s state transition; if valid, the block is included on the Relay Chain and gains finality through Polkadot’s consensus (BABE for block production and GRANDPA for finality).



This model provides shared security: Mosaic enforces its own runtime rules, while Relay Chain finality makes finalized Mosaic blocks economically and cryptographically difficult to revert. Mosaic's permissioning and NPoS-style bonding mechanisms determine which operators are eligible to act as collators and how block-production rights rotate; Polkadot provides the finality and security envelope.

Parachain execution also enables horizontal scalability and interoperability. Mosaic can exchange messages and assets with other parachains and the Relay Chain using Polkadot's Cross-Consensus Messaging (XCM), allowing Mosaic-native applications to route value and calls across the broader Polkadot ecosystem without trusted intermediaries.
Runtime upgrades are delivered as on-chain Wasm updates via governance, enabling forkless evolution of execution logic while preserving chain history and the security guarantees inherited from the Relay Chain.

XCM Direction

Polkadot's Cross-Consensus Messaging (XCM) protocol enables secure, trustless communication between parachains. XCM is a messaging format and set of instructions that allows blockchains with different consensus mechanisms to exchange messages and transfer assets without relying on centralized bridges, trusted oracles, or multi-signature schemes. This eliminates the systemic risk that has led to billions of dollars in losses through traditional bridge hacks and vulnerabilities.

Mosaic Chain

For Mosaic Chain, XCM enables two categories of cross-chain interaction. Asset transfers allow MOS tokens and other assets on Mosaic Chain to be sent to and received from other Polkadot parachains, including established ecosystem participants such as Hydration, Bifrost, Acala, Moonbeam, and Astar. This capability is critical for a finance-oriented chain because DeFi applications fundamentally depend on liquidity, and cross-chain asset flows are the primary mechanism through which liquidity enters and exits a parachain ecosystem. Cross-chain messaging allows smart contracts on Mosaic Chain to invoke operations on other parachains and vice versa, enabling composable financial applications that span multiple chains.

The security model of XCM is qualitatively different from that of traditional bridges. Traditional bridge architectures typically rely on multi-signature schemes, oracle networks, or optimistic verification periods to validate cross-chain messages. Each of these approaches introduces trust assumptions and attack surfaces that have been repeatedly exploited in practice. XCM, by contrast, relies on the relay chain's own validator set to verify cross-chain messages, inheriting the full economic security of the Polkadot network. For a finance-oriented chain where the assets being transferred may represent significant value, this trustless security model is not merely a convenience but a fundamental requirement.

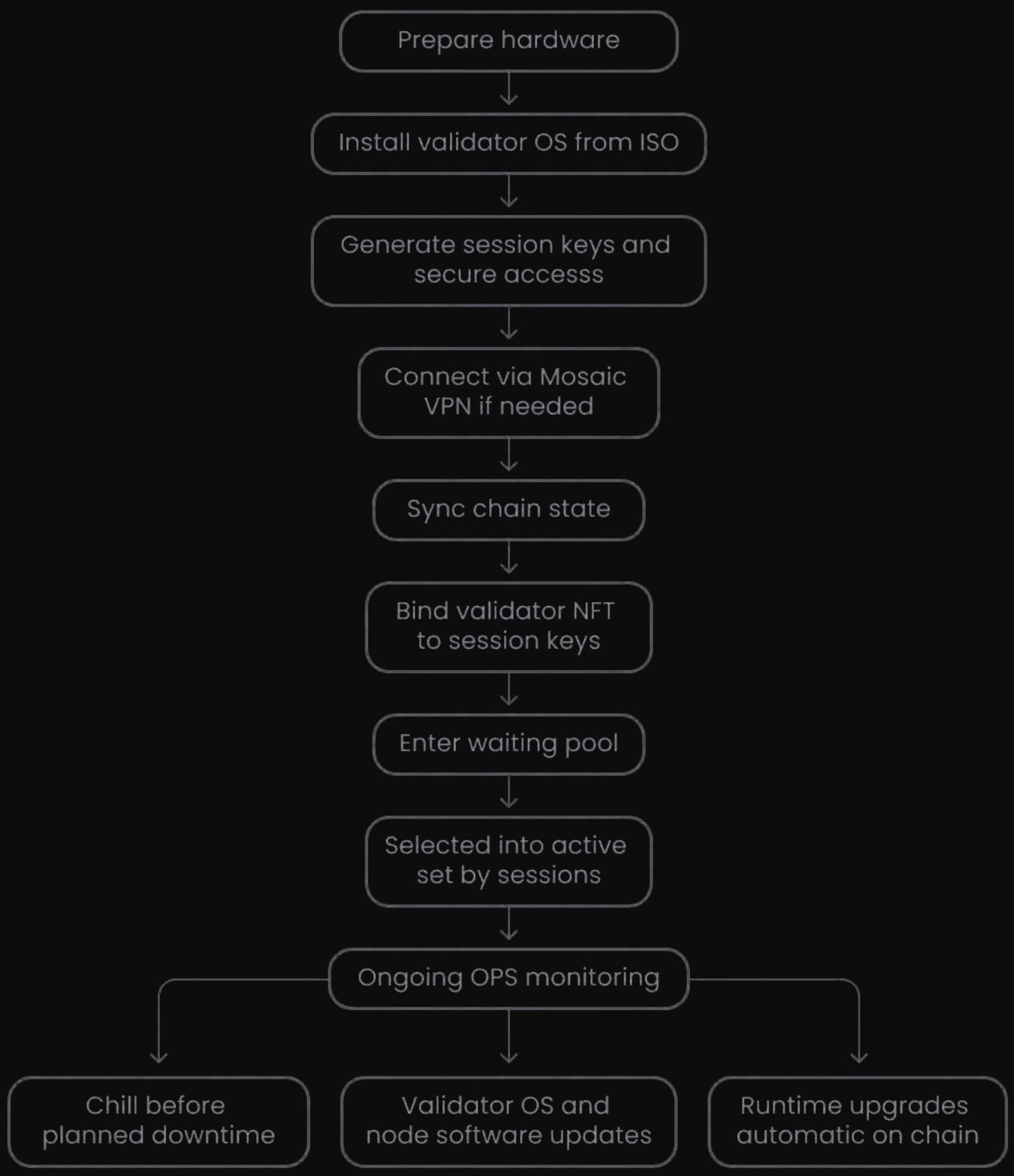
The current scope of Mosaic Chain's XCM implementation is conservative and focused on establishing reliable basic functionality. Asset transfers between Mosaic Chain and other parachains in the Polkadot ecosystem represent the initial capability. Future capabilities, to be developed and deployed through the governance-approved upgrade process, include programmatic cross-chain contract calls, cross-chain liquidity pooling, and integration with parachains that provide complementary financial infrastructure. The roadmap includes bridging MOS to major external networks such as Ethereum and Solana, initially through ecosystem bridges and subsequently through native XCM capabilities as Polkadot's external bridge infrastructure matures.

Mosaic Chain

Validator OS and Operations

What Validator OS Is

Validator OS is a purpose-built, Debian-based Linux distribution designed to standardize and simplify the entire lifecycle of validator node operations on Mosaic Chain. Unlike general-purpose operating systems where validator software must be manually installed, configured, and maintained, Validator OS provides a pre-configured environment with all necessary components, security measures, and operational tooling pre-installed.



The system is distributed as an ISO image that can be written to a USB drive (minimum 16GB) and used to format and configure validator hardware directly. The operating system includes the complete Mosaic Chain node software, networking configuration for peer-to-peer connectivity (including Mosaic VPN for nodes behind NAT), monitoring and telemetry infrastructure, and automated update mechanisms. No separate downloads or package installations are required after the initial setup.

The design philosophy behind Validator OS is to reduce the operational barrier to becoming a validator while improving reliability through standardized tooling. Instead of assuming every validator operator is a professional system administrator, Mosaic provides a purpose-built Debian-based distribution that packages the node, networking, monitoring, and update tooling into a guided, repeatable workflow. Validator OS is planned to be open-sourced so independent parties can create and maintain alternative flavors. This avoids a fragile monoculture where a single OS-level bug could impact the entire network, while still preserving a recommended “golden path” configuration for most operators.

Onboarding Flow

The validator onboarding process follows a deliberate, sequential flow designed to ensure that each validator node is properly configured and secured before joining the network. The process is intentionally strict in certain areas because some steps are irreversible and because key custody errors cannot be corrected after the fact.
The process begins with hardware preparation. A computer meeting the minimum specifications (Intel Core i5–12400 or equivalent CPU, 32GB RAM DDR4, 2×1TB NVMe SSDs, wired Ethernet connection with at least 30–40 Mbps bandwidth and ideally sub-50ms latency) must be available without any pre-installed operating system. The Validator OS ISO image is written to a USB drive, and the system is booted from the USB to begin installation.

The installer formats the target machine's storage, installs the operating system and all validator software, configures the network stack, and establishes the initial security settings. Once the operating system is installed, the node operator is guided through a step-by-step registration process that includes generating session keys, configuring two-factor authentication, connecting to the Mosaic VPN for peer discovery, and synchronizing the blockchain state from the network.

The final onboarding step is binding a Validator NFT to the node's session keys. This on-chain transaction formally registers the node as an eligible validator and, once the node is fully synchronized, places it in the waiting pool for inclusion in future active sets through the validator subset selection algorithm.

To preserve the network's decentralization properties, several configurations are explicitly unsupported in the recommended operating model: running multiple validators on a single physical computer, hosting multiple validators in the same physical location, installing validators in cloud services (VPS) or with any service provider, and hosting validators in the same server park. These configurations are not enforced via protocol-level technical checks; instead, the network monitors how much of the validator set follows these recommendations to understand decentralization health, while leaving room for legitimate edge cases where stricter enforcement would break valid use-cases.

The Mosaic VPN service is a complementary component that addresses a practical barrier to home-based validation: many residential internet connections use Network Address Translation (NAT) and do not provide publicly routable IP addresses, which are traditionally required for peer-to-peer networking. Mosaic VPN enables validator nodes behind NAT to securely connect to other nodes in the network without requiring the operator to configure port forwarding, static IP addresses, or other advanced networking settings. This service is integrated into Validator OS and configured automatically during the onboarding process.

Hardware requirements are deliberately modest and accessible. The recommended specifications—a consumer-grade Intel Core i5 processor, 32GB of RAM, and two 1TB NVMe SSDs—are achievable with off-the-shelf components at a fraction of the cost required by networks with more intensive hardware requirements. The dual SSD configuration will provide redundancy for the chain's storage layer without requiring enterprise-grade RAID hardware. The emphasis on home-based operation, combined with the accessibility of the hardware requirements, supports the project's goal of decentralization across a diverse geographical distribution of validators rather than concentration in professional data centers.

Reliability and Maintenance

Ongoing validator maintenance operates across two independent update channels. Operating system and node software updates are delivered through Validator OS's built-in update mechanism, which pulls updates from Mosaic Chain's repositories automatically. These updates may include security patches, performance improvements, networking stack updates, and new versions of the node software. Power users who prefer manual control can opt out of automatic updates and maintain their systems directly. The software stack follows standard Substrate node conventions, and code availability is expected to expand over time as components are progressively open-sourced.

Runtime upgrades, as described in Section 8.2, operate independently through the on-chain forkless upgrade mechanism. Validators do not need to take any action for runtime upgrades to take effect; the new runtime is automatically applied by the consensus engine when it encounters the upgrade transaction in the chain state.

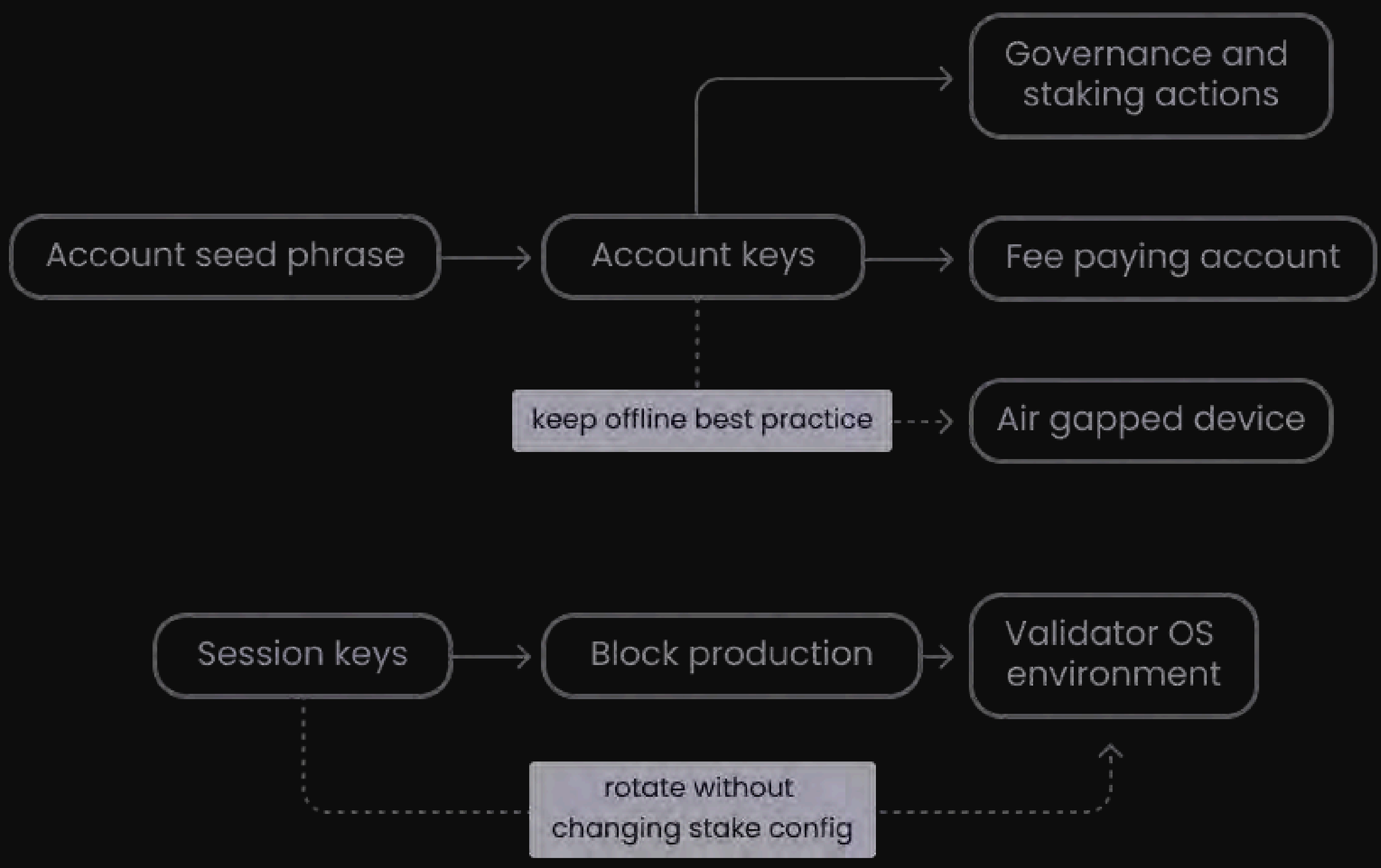
Best practices for validator maintenance include proactive chilling before planned downtime (hardware maintenance, network changes, or physical relocation), maintaining backup power supplies (UPS) and network redundancy to prevent unplanned downtime, regularly verifying that monitoring and telemetry systems are reporting correctly, and practicing secure credential management (particularly for seed phrases, which are irrecoverable if lost). The Validator OS frontend provides real-time visibility into node health, synchronization status, staking performance, and reward history, enabling operators to detect and respond to issues before they result in slashing penalties.

Mosaic Chain

Security Considerations

Key Custody and User Safety

Mosaic Chain uses standard public-key cryptography for account authentication and transaction signing. Each account is derived from a seed phrase (mnemonic) that serves as the ultimate root of authority over the account's assets and operations. The security model places responsibility for seed phrase custody entirely on the user: there is no recovery mechanism if a seed phrase is lost, and no revocation mechanism if a seed phrase is compromised. This is a fundamental property of the cryptographic design, not a limitation that can be addressed through software updates.



For validator operators, key custody has additional dimensions. The validator session keys used for block production are distinct from the account keys used for staking and governance operations. This separation allows validators to rotate session keys without changing their staking configuration, and it limits the exposure of high-value account keys to the operational attack surface of the validator node. Best practice is to generate and manage account keys on a separate, air-gapped device, and to keep session key generation confined to the Validator OS environment.

Users and validators should maintain strict account separation between operational accounts (used for daily transactions and fee payments) and high-value accounts (used for large staking positions and governance). This practice limits the blast radius of any single key compromise and creates defense-in-depth against both technical attacks and operational errors.

Mosaic Chain

Economic Security Model

The economic security of Mosaic Chain is grounded in the total value of MOS staked by validators and delegators. An attacker seeking to subvert consensus would need to control a sufficient fraction of the staked supply, which represents a direct economic cost proportional to the network's total staked value. Slashing mechanisms ensure that misbehavior is not merely unprofitable but actively destructive to the attacker's capital, creating a deterrent that scales with the value secured by the network.

The security budget of the network—the total value at risk through staking—determines the cost of a consensus attack. Because Mosaic uses PoA for block production with an NPoS-style economic model, an attacker would need to either accumulate sufficient stake influence (and/or control of permissioned participation rights) or corrupt enough validator operators to control a majority of the active set. The session-based rotation mechanism complicates targeted corruption: because the active set changes every session and is determined by a randomized algorithm, an attacker cannot predict which validators will be active in a future session and would need to compromise a large fraction of the total validator population, not merely a fraction of one active set.

The protocol's economic design incorporates several structural safeguards against concentration risk. The 5% per-validator stake cap prevents any single entity from accumulating a dominant stake position through a single validator. The session-based rotation ensures that even validators with maximum stake do not have continuous block production authority, limiting the impact of any individual validator's compromise. The delegation model, while increasing the total staked amount, also distributes monitoring responsibility across delegators who have economic incentives to evaluate validator quality.

Diversification is critical for delegators managing significant MOS positions. Because slashing applies to both validator self-stake and all delegated stake, a delegator who concentrates their entire position with a single validator accepts concentrated risk. The protocol's staking contract structure, which limits each validator to 1,000 contracts and 5% of total supply, encourages but does not enforce diversification at the individual delegator level. Delegators should evaluate validator uptime history, commission rates, and operational transparency when making delegation decisions. The Mosaic Explorer provides public visibility into these metrics, enabling informed delegation choices.

Mosaic Chain

Operational Security

Validator node security follows defense-in-depth principles. Validator OS provides a hardened base configuration that minimizes the attack surface: validator nodes serve a single purpose (validation) and cannot be used to run other programs or services. Network connectivity is managed through Mosaic VPN, which provides secure peer-to-peer connections even for nodes behind NAT or without public IP addresses. Two-factor authentication protects the validator administration interface.

Common failure modes and their mitigations include unplanned downtime (mitigated by UPS and network redundancy), misconfiguration (mitigated by Validator OS standardization), compromised credentials (mitigated by account separation and session key rotation), and hardware failure (mitigated by the dual-SSD storage configuration and the ability to restore from chain sync on replacement hardware). The protocol's slashing mechanism is calibrated to penalize sustained unreliability without catastrophically punishing isolated incidents, recognizing that operational perfection is neither achievable nor economically efficient to require.

Roadmap

Click here to explore our roadmap

