JANZEER BLOCKCHAIN

Roadmap & Governance

Shipped state · governance · candidate directions

Document v1.0 · Janzeer Network · 2026

Janzeer Roadmap & Governance

This document describes Janzeer's current on-chain governance and a set of candidate future directions. It is **informational and non-binding**. The "Shipped / current state" section reflects only what exists in node version 0.1.0; everything in "Candidate directions" is illustrative future work under consideration, not a commitment, schedule, or promise. Nothing here implies dates, funding, a token sale, a price, or any partnership. Where this document and the code disagree, the code (and the Yellow Paper distilled from it) is authoritative.

1. Shipped / current state

The following are implemented in the running node today. For exact constants and byte layouts, see the Yellow Paper; for the economic detail, see the White Paper and Economics docs.

Consensus & finality

- **Permissionless stake-free block production with two-phase BFT finality.** Validators produce blocks in deterministic round-robin slots; a PREPARE→COMMIT vote flow finalizes each block. Finality is single-block (no confirmation depth) at quorum `bftQuorum(N) = floor(2N/3) + 1` over the actual active-set size `N`. Slots are **15 s** (`time-slot-duration = 12000 ms` + `time-slot-interval = 3000 ms`).
- **Permissionless registration + registration-order election.** Anyone can join by running a node and submitting a `PromoterTx` carrying a **2000 non-refundable deposit** — no vote, no approval, no lock-up. Eligible validators are ranked by registration order (`createDate`), filled up to `consensus.promoters-count = 21`, with `consensus.rotating-producer-seats = 5` elected seats rotating fairly through a `consensus.producer-candidate-pool-size = 100` standby bench. The quorum tracks the actual set size.
- **Graceful exit.** An `ExitPromoterTx` removes a validator from the active set and slot schedule at the next epoch boundary; the registration deposit stays non-refundable.
- **Automatic anchor phase-out.** Genesis anchor validators start privileged (always-in, downtime-immune) and shed that privilege **one per epoch** (`consensus.anchor-demotions-per-epoch = 1`) once enough **proven independent** validators exist (non-anchors that produced in each of the last `consensus.validator-proven-epochs = 2` epochs), never dropping the healthy set below `consensus.safety-floor-validators = 4`; dormant anchors automatically re-engage if the independent set later collapses.
- **Epoch re-election.** Each epoch (`consensus.epoch-height = 17000` main blocks, ≈ 3 days at 15-s blocks) a GenesisBlock freezes and re-elects the active set and commits the still-privileged anchor count (`activeAnchorCount`).
- **Enforcement is live.** Equivocation (double-voting) is provable and slashable via `SlashingTx` (permanent ban at `consensus.equivocation-ban-threshold = 2` proven

offenses, plus a treasury-funded reporter bounty); offline producers are ejected from the next active set (downtime).
- **Single-tip fork choice** (longest-valid-chain) with a rollback path that requires the same `floor(2N/3)+1` quorum of distinct authenticated validators to corroborate.
- **Genesis-pinned rule-set.** The consensus-critical parameter set (`consensus.*`) is hashed into a **chain-spec digest** that is exchanged and verified in the P2P handshake; a node that edits any consensus parameter forks itself off the network instead of silently diverging.

Economics

- **Fixed supply, never minted** (`MAX_SUPPLY = 100,000,000`, scale 8). All coins pre-allocated at genesis.
- **"C + D" emission:** a declining scheduled block subsidy (C) paid out of a pre-funded, key-less treasury, plus treasury fee recycling (D) that returns a share of every fee (`treasury-fee-share-bps = 2000`, i.e. 20%) to the treasury. Net zero inflation, perpetual incentive. The subsidy starts at `reward-block = 10` and **halves yearly** (`reward-halving-period = 2,100,000` blocks ≈ 1 year) down to a perpetual floor `reward-tail-floor = 0.5`.
- **100% to the producer.** The block reward goes entirely to the producing validator's wallet — no delegation, no commission split. The reward tx is system-generated and validated against the canonical value on every node.

Native tokens (JZT-1)

- **Validator-issued fungible tokens.** A registered validator can create a `JZT-1` token via a single `TokenTx` (op = create/mint/burn/setcap/transfer): it sets a symbol, name, decimals, and an **adjustable** `cap`, and it alone mints, burns, or re-caps supply (`0 ≤ totalSupply ≤ cap`). Anyone holding a balance can transfer it freely. Tokens live on a **separate ledger** — a token's supply is not part of the coin's fixed `100,000,000` and does not touch emission, the treasury, or rewards — and every token op still pays its fee in the native coin (CREATE = `5` exact, other ops ≥ `0.01`). Shipped across consensus, the REST API, CLI, and wallets, with schema via Flyway `V2__native_tokens.sql`.

Wallet, API & operations

- **Non-custodial crypto.** Keys are derived and held only client-side; the node verifies signatures and has no key-generation or seed endpoints. Addresses carry an **EIP-55** mixed-case checksum at the wallet/API display boundary (all-lowercase remains the canonical consensus form). Byte-for-byte parity is proven by committed parity vectors (transfer + validator + exit).
- **Cross-chain replay protection.** A `consensus.network-id` (default `janzeer`) is bound into both the signed transaction preimage and the P2P handshake domain, so a tx signed for one network cannot be replayed on another.
- **REST API + Swagger UI** on port `7019` (versioned envelope, pagination, explorer and node-info endpoints, Actuator/Prometheus metrics).

- **Authenticated P2P** (mutual ECDSA handshake, chain-spec digest exchanged) over Netty TCP, with peer discovery and epoch-walking sync.
- **Config-driven genesis.** Treasury, anchors, and the genesis signer ceremony are configuration-driven and computed deterministically so every node derives an identical genesis; committed defaults stay intact.
- **Persistence & delivery.** Embedded H2 file by default, PostgreSQL supported; Flyway-managed schema; multi-stage Docker image; CI (test → bootJar → docker build → Trivy scan); a one-command local network (`run/nodes.sh`, launch ≥ 4 nodes for fault tolerance).

Honest caveats (current state)

- Committed anchor secrets and dev defaults are for **development** and must be rotated for any production deployment.
- The development faucet (`janzeer.dev-fund`) is **off by default** and must never be enabled in production.
- Finality requires a **live quorum** of the active set; a single node cannot finalize.

2. Governance (today)

Janzeer has **no on-chain vote and no governance token.** There is no delegation, no vote weight, and no on-chain parameter voting — all of that was removed in the stake-free redesign. This section describes the honest governance story: who gets to produce blocks, how agreement is reached, and how the rule-set is set and changed.

Who produces and finalizes blocks

- **Permissionless registration, registration-order election — no votes.** Anyone can become a validator by running a node and submitting a `PromoterTx` carrying the **2000 non-refundable deposit** (the primary Sybil guard). There is no vote, no approval, and no lock-up. Eligible validators are ranked by **registration order** (`createDate`, then id) — longest-registered first — not by any rating or vote weight.
- **The active set** = still-privileged anchor validators (from the epoch-1 genesis) + the longest-registered elected validators, filled up to `promoters-count = 21`, with `rotating-producer-seats = 5` elected seats rotating fairly through a `producer-candidate-pool-size = 100` standby bench, minus offline/banned/exited validators.
- **Graceful exit.** An `ExitPromoterTx` retires a validator at the next epoch boundary; the deposit stays non-refundable.
- **Anchors phase out automatically.** Genesis anchors shed their privilege one per epoch as proven independent validators appear, never dropping the healthy set below `safety-floor-validators = 4`, and re-engage if it later collapses — decentralization without a vote.
- **Epoch re-election.** At each epoch boundary the GenesisBlock re-freezes the active set and commits the still-privileged anchor count (`activeAnchorCount`).

How agreement and enforcement work

- **Agreement = BFT finality.** A block is final once it reaches the `floor(2N/3) + 1` quorum (i.e. "2f+1" when `N = 3f+1`) across PREPARE then COMMIT. Votes are bundled into a QuorumCertificate.
- **Enforcement = slashing + ejection.** Equivocation is proven on-chain and slashed (permanent ban at the offense threshold, plus reporter bounty); validators that miss their slots are ejected from the next active set.

Parameters, the rule-set, and treasury

- **The rule-set is fixed at genesis.** The economic and consensus parameters live under `consensus.*` and are frozen into a deterministic genesis. Each node computes a **chain-spec digest** (a hash over those parameters) and exchanges it in the P2P handshake; a peer whose digest differs is rejected. There is **no on-chain mechanism to change these parameters** — no vote, no governance token. Changing the rule-set means shipping a **new genesis**, i.e. a coordinated fork that operators choose to adopt. That is the honest governance model: consensus is by running the software, not by voting on-chain.
- **The treasury is key-less.** The treasury/bank account (`0x0000…0000`) holds no spendable key. No one can move it directly; only protocol reward and fee-recycling rules move funds out of (and back into) it.

3. Candidate directions (non-binding)

> **This section is illustrative.** Items below are reasonable, honestly-framed possibilities under consideration. They are **not commitments**, are **not scheduled**, and may change or be dropped. The loose phases (Now / Next / Later) express rough sequencing only — they carry no dates.

We use cautious language deliberately: each item is a *candidate* that *may* be pursued.

Now (in or near the current codebase)

- **Mobile light client.** A wallet/light client (Flutter/Kotlin) that holds keys client-side and submits signed transactions (transfer, validator registration, exit). This is designed and deferred-to-last in the protocol-upgrade notes, but **not yet built** in this repository.
- **Leader-routing of votes.** The BFT vote path is already bounded and collapsed into a QuorumCertificate; full leader-routing of consensus votes is the remaining optimization candidate.
- **Documentation & parity hardening.** Continued expansion of the bilingual docs set and the committed parity vectors.

Next (candidate, larger but adjacent)

- **Historical pruning / sync modes.** Building out the LIGHT/FULL `SyncMode` distinction so nodes can retain less history while staying verifiable, reducing footprint for smaller

operators.

- **Broader / more dynamic validator set.** Tuning or raising the active-set cap and rotation while preserving fast BFT finality (the bounded voting set is what keeps finality fast, so any change here is a careful trade-off). Any such change is a consensus parameter, so it would ship as a new genesis (coordinated fork), not an on-chain vote.
- **Observability & SDKs.** Richer metrics/dashboards on top of the existing Actuator/Prometheus surface, and client SDKs that wrap the non-custodial signing scheme.

Later (candidate, exploratory)

- **External security audit.** An independent review of the consensus, crypto, and economic code is a candidate before any production launch.
- **Explorer & UX.** A fuller block/transaction explorer and improved wallet UX building on the existing `/api/v1/explorer` surface.
- **Performance / native image.** Investigating startup and footprint improvements (e.g. native-image builds) for operators.

None of the above changes the fixed-supply, never-minted nature of the native coin, the non-custodial design, the stake-free permissionless registration model, or BFT finality; those are core properties, not roadmap items.

4. What this document does NOT promise

To be unambiguous: this roadmap states no dates, no quarters beyond the loosely-labeled Now/Next/Later phases, no funding, no token sale, no price, no listing, and no partnership. There is **no on-chain vote and no governance token** in node version 0.1.0; the consensus rule-set is fixed at genesis (pinned by the chain-spec digest) and changing it requires a coordinated fork onto a new genesis. Delegation, voting, staking, and on-chain parameter governance were **removed** in the stake-free redesign — they are not planned features.

Document status: version 1.0. Reflects node version 0.1.0. Sections 1 and 2 describe shipped behavior; Section 3 is non-binding and illustrative. Exact byte-level facts live in the Yellow Paper.