JANZEER BLOCKCHAIN

Economics

Supply, emission & validator economics

Document v1.0 · Janzeer Network · 2026

Janzeer Economics

Janzeer is a fixed-supply, permissionless stake-free BFT Layer-1. Its entire economy rests on one invariant: exactly **100,000,000** coins exist, all pre-allocated once at genesis and never minted again. Block rewards are not new issuance — they are paid *out of* a pre-funded on-chain treasury, and a slice of every transaction fee flows back into that treasury. The result is a "C + D" emission model: a declining scheduled subsidy (C) bounded by the treasury, plus perpetual fee recycling (D), giving block producers a permanent incentive with **zero net inflation**. This page summarizes the numbers; exact byte-level facts live in the Yellow Paper.

At a glance

Parameter	Value	Notes
Max supply	**100,000,000**	fixed, never minted (scale 8)
Decimals (scale)	**8**	`0.01 → 1,000,000` base units
Genesis treasury ("bank")	**77,500,000**	key-less address `0x0000…0000`
Ecosystem/distribution pool	**12,000,000**	transparent custody (faucet/airdrop/governed)
Founder allocation	**7,500,000**	carved out of treasury when a custody address is set
Anchor validators	**3 × 1,000,000**	three bootstrap validators
Base block reward	**10**	`rewardBlock`
Halving period	**2,100,000 blocks**	`rewardHalvingPeriod` — ≈ 1 year at 15-s blocks (reward halves yearly)
Tail floor (perpetual min reward)	**0.5**	`rewardTailFloor`
Slot duration	**15 s**	12,000 ms active + 3,000 ms intermission
Fee split	**80 / 20**	80% → producer, 20% → treasury (`treasuryFeeShareBps = 2000`)
Block reward recipient	**100% → producer**	no delegation, no commission split
Validator registration	**fee 3 + deposit 2000**	deposit → treasury, **non-refundable**
Minimum amount	**0.1**	`MINIMUM_AMOUNT`
Minimum fee	**0.01**	`MINIMUM_FEE`

There is no ticker symbol in code; we refer to the unit as "the native coin".

1. Fixed supply and no minting

`MAX_SUPPLY = 100,000,000` coins are pre-allocated exactly once, at genesis, and the protocol has **no minting path**. The genesis allocation sums to *exactly* `100,000,000`, enforced by a fail-fast cap-invariant assert that refuses to boot if the numbers do not add up:

- **77,500,000** funds the treasury "bank" account at `0x0000000000000000000000000000000000000000`. This address is **key-less** (no private key derives it), so no one can spend it directly — only the protocol's reward and fee logic moves coins out of, or into, it.

- **12,000,000** seeds an ecosystem/distribution pool. Its custody must be **transparent** — a faucet, airdrop, or governed program — **not** a founder-controlled wallet (see Caveats).
- **7,500,000** is a founder allocation.
- **3 × 1,000,000** seeds the three bootstrap anchor validators.

The founder, distribution, and optional dev-fund faucet allocations are carved **out of** the treasury only when a custody address is supplied (`janzeer.genesis.founder` / `.distribution` / `janzeer.dev-fund`); if no address is given, they **fold back into** the treasury so that nothing is ever created beyond the cap.

Circulating supply is defined simply as everything not sitting in the treasury:

```
circulatingSupply = MAX_SUPPLY − bankBalance
```

As rewards are paid out, the bank shrinks and circulating supply rises; as fees recycle back, the bank refills. The total — `100,000,000` — never changes.

2. The C + D emission model

Janzeer pays producers without inflation by combining two flows.

C — declining scheduled subsidy

Each block has a scheduled reward that halves every `2,100,000` blocks (**≈ 1 year at 15-s blocks — so the reward halves roughly once a year**) but never drops below a perpetual floor:

```
scheduledReward(h) = max(0.5, 10 / 2^(h / 2,100,000))
```

Block height range	Approx. scheduled reward
0 → first halving	10 → 5
after 1st halving	~5
after 2nd halving	~2.5
after 3rd halving	~1.25
…	…
once `scheduledReward ≤ 0.5`	**0.5 forever (tail floor)**

The actually paid reward is always capped by what the bank holds:

```
paidReward(h) = min(scheduledReward(h), bankBalance)
```

If the bank ever empties, scheduled pay falls to 0 unless refilled — which is where D comes in.

D — treasury fee recycling

A fixed share of **every transaction fee returns to the treasury bank** (`treasuryFeeShareBps = 2000` → 20%). This continuously refills the bank, funding the perpetual `0.5` tail reward indefinitely. Because fees are *recycled*, not *minted*, the model is **net-zero inflation**: coins move between participants and the treasury, but the `100,000,000` cap is conserved at all times.

3. Who earns what

Recipient	Earns	Form
Block producer (validator)	**100%** of the block reward + **80%** of every fee in the block	spendable balance
Treasury (bank)	**20%** of every fee + validator registration deposits (2000 each)	bank balance

The block reward goes **100% to the producing validator's wallet** — there is no delegation, no commission split, and no reward sharing. Producers are paid for the *work* of running the chain, not for yield on a stake. The reward transaction is **system-generated** (fee 0, nonce 0). Every node independently recomputes the canonical reward; a block whose reward tx doesn't match the expected value is rejected.

Worked example — one block reward

Suppose a validator produces a block early in the chain with `scheduledReward = 10`, and the bank has ample balance, so `paidReward = 10`.

1. The full `10` is credited to the producing validator's **spendable balance** — no split, no delegators.
2. Separately, every fee paid by transactions inside that block splits **80% to the producer's wallet, 20% to the treasury**.

For example, a block carrying transactions with `2.0` in total fees pays the producer `10 + (2.0 × 0.80) = 11.6`, and returns `2.0 × 0.20 = 0.4` to the treasury bank.

4. Validator admission (no staking)

Janzeer has **no staking, delegation, or voting**. A validator is admitted purely by running a node and registering — there is no locked stake, no bonding period, and no vote weight.

- **Register:** submit a `PromoterTx` (fee **3**) carrying a **2000** registration deposit. The deposit is sent to the treasury bank and is **non-refundable**.
- **The deposit is the Sybil guard:** together with one-seat-per-key and fair rotation, the non-refundable deposit is what makes it costly to flood the network with fake validators. It replaces the role that stake or proof-of-work plays in other chains.

- **Exit anytime:** submit an `ExitPromoterTx` (fee ≥ 0.01) to gracefully retire; the validator leaves the active set at the next epoch boundary. The registration deposit **stays gone** — exiting does not return it.
- **Selection is by registration order**, not by stake or votes: eligible validators are ranked by registration time (longest-registered first) and filled up to the seat cap. See the White Paper / Yellow Paper for the full validator lifecycle and rotation rules.

5. Fee economics and supply conservation

Transaction	Fee	Deposit / amount	Destination of fee
TransferTx	≥ 0.01	amount ≥ 0.1 (unless memo)	80% producer / 20% treasury
PromoterTx	3	2000 → treasury (non-refundable)	fee split 80/20; deposit → treasury
ExitPromoterTx	≥ 0.01	—	80% producer / 20% treasury
RewardTx (system)	0	—	bank → producer (100%)
SlashingTx (system)	0	—	reporter bounty 5 from treasury

Fees are never burned. Every fee is split — `80%` to the block producer's wallet, `20%` back to the treasury bank — so the sum of all balances plus the treasury is invariant. The minimum fee is `0.01`. Validator registration deposits (`2000`) likewise move *into* the treasury rather than out of existence. No transaction destroys coins, which is what makes the `100,000,000` cap a true conservation law rather than a soft target.

6. Native tokens: a separate ledger

Janzeer also supports optional user-defined fungible tokens (**JZT-1**), but they are **economically separate** from the native coin. A token's supply — what its issuer mints, burns, or re-caps — is its own ledger: it is **not** part of the fixed `100,000,000` coin supply, and it does not change the treasury, the C + D emission, or block rewards in any way. Token amounts never touch the coin's conservation law.

The one and only economic link to the coin is the **fee**: every token operation pays its fee **in the native coin** — `CREATE` costs exactly `5` (`fee-create-token-tx`), and every other op (`mint`, `burn`, `setcap`, `transfer`) pays at least the `0.01` `MINIMUM_FEE`. Those fees split the usual `80 / 20` (producer / treasury) like any other fee, so token activity is simply a (small) additional source of the same fee recycling (D) that funds rewards. See the White Paper for what tokens are, and the Yellow Paper for the exact `TokenTx` byte layout.

7. Honest caveats

- **Genesis allocations are configurable defaults.** The three anchor validator wallets and any committed balances are **development defaults**, overridable at genesis via `janzeer.genesis.anchors`. Production deployments set their own anchors; the headline supply parameters (cap, genesis split, reward schedule) are protocol constants and do not change.
- **The 12,000,000 ecosystem pool needs transparent custody.** It must be held and disbursed through a transparent mechanism — a faucet, an airdrop, or a governed program — **not** a founder wallet. The dev-fund faucet (`janzeer.dev-fund`) is a development convenience drawn from the treasury and must be **OFF in production**.
- **No token-sale or price claims.** This document describes only the on-chain economic mechanics. It makes no representation about price, market value, distribution sale, or any return. There is no ticker symbol in code.
- **The treasury is key-less by design.** No party — including the team — can spend the `77,500,000` bank directly; only protocol reward/fee logic moves it.
- **The registration deposit is non-refundable by design.** Retiring a validator (`ExitPromoterTx`) does not return the `2000` deposit.

Document status: version 1.0. Reflects node version 0.1.0. Exact byte-level facts (preimages, hashing, amount scaling, consensus math) live in the Yellow Paper.