

Data Net Staking & Reward Distribution

Mechanism Design Specification

v0.2 — Updated with Anti-Gaming Mechanisms & EVOF Metric

1. Overview & Purpose

This document specifies the staking and reward distribution mechanism for the Reppo Protocol. It is intended as a direct specification for the smart contract engineering team. It covers the Performance Pool lifecycle, Data Net scoring model, reward distribution logic, epoch mechanics, and all anti-gaming protections.

Goal

Incentivize long-term REPPRO holding by rewarding stakers with a share of protocol revenue, while rewarding Data Net owners for running high-quality, high-usage nets — creating aligned incentives across all network participants, with robust protections against epoch gaming.

2. Performance Pool — Funding Sources

The Performance Pool is the central reserve from which all staking rewards are distributed. It is funded by two distinct protocol revenue streams:

Data Net Spin-Up Fees	100% of the one-time fee charged to operators when registering and launching a new Data Net on the protocol.
Protocol Revenue Share	10% of all publishing fees and 10% of all data access fees collected by the protocol. This is a continuous inflow that grows with network activity.

These funds accumulate continuously in the Performance Pool contract. Distribution is triggered at the end of each epoch. The current pool balance is approximately 90,000 REPPRO tokens.

3. Epoch Design

3.1 Epoch Length

Epochs are 48 hours (2 days) long. This is the core unit of time over which performance is measured and rewards are distributed.

3.2 Reward Smoothing (Anti-Gaming)

To prevent participants from gaming specific epoch windows, rewards are not calculated on a single snapshot. Each Data Net's performance score is a rolling weighted average across the most recent 3 epochs:

$$\begin{aligned} \text{SmoothedScore}(\text{net}, t) &= 0.50 * \text{RawScore}(\text{net}, t-1) \\ &+ 0.30 * \text{RawScore}(\text{net}, t-2) \\ &+ 0.20 * \text{RawScore}(\text{net}, t-3) \end{aligned}$$

This means a single outstanding epoch only partially inflates a net's score. Sustained performance across multiple epochs is required to earn top-tier rewards. The weights are governance-adjustable parameters.

3.3 New Data Net Eligibility

A Data Net that launches mid-epoch is NOT eligible for rewards in its launch epoch. It must complete one full epoch of operation before it begins accruing performance metrics that count toward reward allocation. This prevents operators from launching nets strategically to capture end-of-epoch rewards.

Rule

New Data Nets: first full epoch = observation only. Reward eligibility begins at the start of the second full epoch after registration.

3.4 Data Net Deregistration Delay

Operators cannot instantly deregister a Data Net. A deregistration request triggers a cooldown period before the net is removed from the protocol. This prevents operators from spinning up nets opportunistically, capturing rewards, and immediately exiting. The cooldown duration is governance-adjustable and recommended to be a minimum of 2 epochs (96 hours).

4. Reward Split — Owners vs. Stakers

At the end of each epoch, the total distributable amount from the Performance Pool is split between two recipient groups:

Recipient	Default Share	Rationale
REPP0 Stakers	80%	Incentivize capital deployment and long-term REPP0 holding. Larger share drives staking demand and reduces circulating supply.
Data Net Owners	20%	Compensate operators for infrastructure costs and incentivize launching and maintaining high-quality Data Nets.

Both the 80% and 20% values are stored as basis points on-chain and are governance-adjustable. See Section 8.

5. Data Net Scoring & Reward Allocation

5.1 The Four Performance Metrics

Each epoch, every eligible Data Net receives a performance score based on four on-chain metrics. All four metrics are derived entirely from on-chain data — no external oracle is required.

Metric	Default Weight	Description
EVOF (Epoch Value Over Flow)	40%	Total epoch trading volume divided by the total number of unique data sources active in the net during the epoch. A quality-efficiency signal: rewards nets that generate high value without bloat. Derived from Reppo white paper. Highest weight as it captures both activity and quality.
Total Fees Earned (epoch)	25%	Total protocol fees generated by the net during the epoch. Directly tied to economic activity and protocol revenue contribution. Fully on-chain.
Total REPP0 Staked	25%	Cumulative REPP0 staked into the net at the epoch snapshot block. Reflects community conviction and capital commitment. Fully on-chain.
Trading Volume (epoch)	10%	Total value of data transactions executed within the net during the epoch. Partially captured by EVOF already, so lower standalone weight. Included as an independent signal of raw demand.

Note All four weights must sum to 10000 basis points. They are all governance-adjustable. EVOF is defined in the Reppo white paper as: $EVOF = \text{total epoch trading volume} / \text{total unique data sources in the net during the epoch}$.

5.2 Raw Score Formula

The raw score for Data Net i in epoch t is:

```
RawScore(i,t) = (w_evof * NormEVOF(i,t))
                + (w_fees * NormFees(i,t))
                + (w_stake * NormStake(i,t))
                + (w_volume * NormVolume(i,t))
```

Where $\text{Norm}_X(i,t)$ is each metric value for net i normalized across all eligible nets in epoch t (value divided by the sum of that metric across all nets).

5.3 Smoothed Score & Net Allocation

The smoothed score (Section 3.2) is used to determine each net's share of the reward pool:

```
NetShare(i) = SmoothedScore(i) / SUM(SmoothedScore, all
eligible nets)
NetReward(i) = TotalPoolReward * NetShare(i)
```

$\text{NetReward}(i)$ is then split: 20% to the net owner, 80% to the staker pool for that net.

6. Staking Mechanics & Anti-Gaming

6.1 Single Staking Contract

There is one canonical `StakingContract` for the entire protocol. It tracks stakes per-user, per-Data Net internally. Users can split their REPPO across multiple nets within a single contract interaction. No separate staking contracts per net.

6.2 Time-Weighted Stake (Duration-Pro-Rata)

This is the core anti-gaming mechanism. Staker rewards are NOT based on the stake balance at a single snapshot. They are based on the time-weighted average stake across the full epoch. A user who stakes at hour 46 of a 48-hour epoch only earns rewards proportional to those 2 hours — not the full epoch.

```
TimeWeightedStake(user, net, epoch) = SUM(stake_amount *
hours_held) / 48
```

This is calculated using block timestamps. Every stake and unstake event is recorded with its block timestamp. At epoch close, the contract integrates the user's stake over time. This makes last-minute whale entries economically unattractive — a whale staking 1,000,000 REppo at hour 47 earns rewards on roughly 20,833 REppo equivalent, not 1,000,000.

Key Property

Stakes automatically carry forward into the next epoch at full time weight. A user who staked at hour 46 of epoch N earns only 2/48 of that epoch's rewards, but earns full rewards in epoch N+1 onwards with no additional action required.

6.3 Staker Reward Calculation

When a user claims rewards for epoch E and net N:

```
UserTimeWeightedStake = SUM(stake_i * duration_i) /
epochLength
NetTimeWeightedTotal  = SUM(all stakers' TimeWeightedStake
for this net)
UserStakerReward      = NetReward(N) * stakerBps/10000
                       * UserTimeWeightedStake /
NetTimeWeightedTotal
```

6.4 Stake Snapshots

The contract must record a stake checkpoint on every stake and unstake event, storing the block number and timestamp. This is required to accurately compute time-weighted averages at claim time. Checkpoints are stored per user per net.

6.5 Key Contract Data Structures

```
struct StakeCheckpoint { uint256 amount; uint256
timestamp; }
mapping(address => mapping(uint256 netId =>
StakeCheckpoint[])) checkpoints
mapping(uint256 netId => address owner) netOwner
mapping(uint256 epoch => mapping(uint256 netId => uint256
reward)) epochNetRewards
mapping(uint256 netId => uint256 registrationEpoch)
netRegistrationEpoch
```

6.6 Key Contract Functions

stake(netId, amount)	Stake REPPO into a specific Data Net. Records a checkpoint with current timestamp. Net must be in eligible state (past its first full epoch).
unstake(netId, amount)	Initiate unstake. Records checkpoint. Starts cooldown timer. Stake no longer accrues time-weight after this point.
withdraw(netId)	After cooldown expires, transfers REPPO back to user wallet.
claimRewards(epoch, netId)	Claim staker rewards for a specific epoch and net. Computes time-weighted share lazily at claim time using stored checkpoints.
distributeEpoch(epoch)	Called by protocol keeper at epoch close. Reads on-chain metrics (EVOF, fees, volume, stake), computes scores, writes epoch rewards. Permissioned.
registerNet(netId)	Register a new Data Net. Records registration epoch. Net is ineligible for rewards until the following epoch boundary.
requestDeregister(netId)	Initiates deregistration cooldown. Net remains active and eligible for rewards during cooldown. Owner cannot withdraw spin-up stake until cooldown completes.
updateWeights(...)	Governance: update metric weights. Must sum to 10000 bps. Timelocked.
updateSplit(stakerBps, ownerBps)	Governance: update 80/20 split. Must sum to 10000 bps. Timelocked.

7. Unstaking & Cooldown

Cooldown Duration	7 days (governance-adjustable). Recommended minimum: 1 epoch (48 hours).
During Cooldown	REPPO is locked. Time-weight accrual stops at the unstake transaction block. Stake is NOT counted in future epoch snapshots.
After Cooldown	User calls withdraw() to receive REPPO back to their wallet.
Carry-Forward	If a user does NOT unstake, their stake automatically carries into the next epoch at full weight with no action required.
Slashing	No slashing in v1. May be introduced in a future version for adversarial or Sybil behaviour.

8. Governance-Controlled Parameters

All critical mechanism parameters are stored on-chain and adjustable through Reppo governance. All changes are subject to a minimum 48-hour timelock before taking effect.

Parameter	Default Value	Notes
<code>stakerBps</code>	8000 (80%)	Staker share of Performance Pool. Must sum to 10000 with <code>ownerBps</code> .
<code>ownerBps</code>	2000 (20%)	Owner share of Performance Pool.
<code>w_evof</code>	4000 (40%)	Weight for EVOF metric. All four weights must sum to 10000.
<code>w_fees</code>	2500 (25%)	Weight for fees earned metric.
<code>w_stake</code>	2500 (25%)	Weight for total REPPPO staked metric.
<code>w_volume</code>	1000 (10%)	Weight for trading volume metric.
<code>epochLength</code>	48 hours	Length of each epoch in seconds (172800).
<code>smoothingWeights</code>	[0.50, 0.30, 0.20]	Rolling average weights for epochs t-1, t-2, t-3.
<code>cooldownPeriod</code>	7 days	Unstaking cooldown duration.
<code>deregisterCooldown</code>	4 epochs (192h)	Minimum delay before a Data Net can be fully deregistered.
<code>newNetEligibilityDelay</code>	1 epoch (48h)	Epochs a new net must complete before earning rewards.
<code>timelockDelay</code>	48 hours	Minimum delay before governance param changes take effect.

9. Worked Example

Assume a 100,000 REPPPO distribution epoch with three eligible Data Nets A, B, and C. After normalization and smoothing, their `SmoothedScores` are 0.50, 0.30, and 0.20 respectively.

Data Net A allocation	50,000 REPPPO → Owner: 10,000 Staker pool: 40,000
Data Net B allocation	30,000 REPPPO → Owner: 6,000 Staker pool: 24,000
Data Net C allocation	20,000 REPPPO → Owner: 4,000 Staker pool: 16,000

For a user staked in Data Net A:

- User staked 1,000 REPPPO at the START of the epoch (full 48 hours). Time-weighted stake = 1,000.
- Total time-weighted stake across all stakers in Net A = 10,000 REPPPO.
- User's share = $1,000 / 10,000 = 10\%$. User earns $40,000 \times 10\% = 4,000$ REPPPO.

Same user, late entry scenario:

- User instead staked 1,000 REPPO at hour 46 of 48. Time-weighted stake = $1,000 \times (2/48) = 41.67$ REPPO equivalent.
- Assuming same total net stake, user earns $40,000 \times (41.67/10,041.67) =$ approximately 166 REPPO.
- In epoch N+1, the same user earns the full 4,000 REPPO with no additional action needed.

Anti-Gaming Effect

The whale who dumps 1,000,000 REPPO at hour 47 earns rewards on ~20,833 REPPO equivalent that epoch — not 1,000,000. The economics of gaming are broken by time-weighting alone, with no hard cutoffs or blacklists required.

10. Summary of Anti-Gaming Protections

Attack Vector	Protection Mechanism
Late-epoch whale staking	Time-weighted stake: rewards proportional to hours held, not final balance.
Staking, claiming, exiting	Unstaking cooldown (7 days default) prevents rapid stake-claim-exit loops.
Launching nets to capture rewards	New net eligibility delay: 1 full epoch must complete before earning.
Spinning up and shutting down nets	Deregistration cooldown: minimum 4 epochs before net can be removed.
Inflating volume in one epoch	3-epoch rolling average: single epoch spikes have 50% weight at most.
High volume but low quality data	EVOF metric (40% weight) penalizes nets with bloated sources relative to volume.

11. Open Questions for Engineering

The following items require decisions before final implementation:

- EVOF on-chain availability: Confirm that the unique source count per net per epoch is emitted as an on-chain event today, or define the event that needs to be added.
- Checkpoint gas optimization: Time-weighted stake requires storing checkpoints per user per net. Evaluate whether a compressed checkpoint structure or a Merkle-based approach is needed to keep gas costs acceptable at scale.

- Batch claim: Should users be able to claim rewards across multiple epochs and nets in a single transaction to reduce gas overhead?
- Owner reward address: Is the reward payout address the net deployer address, or a configurable address set at registration? Recommend making it configurable.
- Performance Pool floor: What happens if the pool runs dry mid-epoch? Define a minimum reserve threshold and whether distribution is delayed or partial.
- EVOF weight confirmation: The 40% weight for EVOF is a starting recommendation. Team should validate this against simulated epoch data before mainnet.

Reppo Labs internal engineering specification v0.2. Subject to change based on smart contract feasibility and governance input.