Overview

This document describes the Reserve contract for the Neutron network.

The Reserve contract holds the vested NTRNs, and sends them to theDAO treasury (core module) andDistribution contracts. This contract is owned by theNeutron DAO and is instantiated at genesis. It is responsible for the first step of tokens distribution.

Reserve contract can be configured only by theNeutron DAO . Thedistribute call is permissionless and can be called by anybody.

Reserve coins are vested based on on-chain activity: the more NTRN coins are burned while processing block fees (see above), the more tokens distributed from the Reserve.

In order to distribute coins any address can executedistribute call, this method is triggered by a transaction and distributes coins from a treasure. It starts by loading configuration parameters such as the denomination of the currency, minimum period between distributions, and the distribution rate.

The function checks if the time since the last distribution is less than the minimum period. If it is, then the function returns anContractError::TooSoonToDistribute error indicating that it is too soon to distribute. The time of the last distribution is saved for future reference. The current balance of the contract is retrieved from the interchain querier. If there are no funds, then the function returns anContractError::NoFundsToDistribute error indicating that there are no funds to distribute.

Also it calculates the amount of burned coins for the period and adjusts it using thesafe_burned_coins_for_period function to prevent arithmetic overflow. If there are no burned coins, then the function returns anContractError::NoBurnedCoins error indicating that there are no burned coins. The balance to distribute is calculated using thevesting_function .

Thevesting_function calculates the amount of coins to be distributed (released)we use following expression:burnt_tokens × multiplier burnt_tokens \times multiplier burnt_tokens

× multiplier based on the formulamultiplier = (configurable_denominator – 1 configurable_denominator) burnt_tokens ∗ x multiplier = (\frac{configurable_denominator-1}{configurable_denominator})^{burnt_tokens} * x multiplier

= (configurable_denominator

configurable_denominator – 1) burnt_tokens

∗ x , wherex x x is the current balance in the treasure andburnt_tokens {burnt_tokens} burnt_tokens is the number of burned coins in a period of time. The original formulamultiplier = x configurable_denominator multiplier = \frac{x}{configurable_denominator} multiplier

= configurable_denominator

x was optimized and current formula was used in order to make the calculation more efficient and faster. Otherwise it will be required to calculate released coins on every burned coin but this is may be time and resource consuming.

Theconfigurable_denominator configurable_denominator configurable_denominator should be set based on coins burning rate and preferred total duration of vesting. Take into account that while initially one burnt tokens equals multiple NTRN tokens made liquid, the flow of new tokens into the Reserve progressively slows down until the tokens supply is exhausted and the tokenomics becomes deflationary.

The final result is rounded up to the nearest integer and returned as aUint128 type.

The amount of coins to distribute and the amount of coins to reserve are then calculated. The distribution stats are updated and a response is created indicating the amount of coins distributed, the amount of coins reserved, and a tag indicating the action as "neutron/reserve/distribute".

Deployment

This is one of the contracts that are initialized at Neutron genesisInitialization message contains The Neutron DAO and Security DAO addresses. Previous Overview Next Messages