Overview

Being an Interchain Secured network, Neutron does not use standard Cosmos SDK governance module. Neutron governance is based onDAO DAO contracts, with some modifications. It consists of two parts:

1. The Neutron DAO,
2. Multiple subDAOs.

For privileged actions (e.g., changing network parameters and making software update proposals) Neutron uses theadmin-module fork managed by the Informal team. This module allows to specify a list of admin addresses that are able to submit proposals that are automatically executed. Neutron DAO smart contract address is added as an admin during genesis, allowing the DAO to manage the network as it sees fit.

Neutron DAO

The Neutron DAO supportssingle-choice andmultiple-choice proposals by registering the corresponding proposal contracts in the core contract, along with a special type ofoverrule proposals (see below). In the future, additional types of proposals might be introduced (e.g., gauges).

Each type of proposal can only be submitted through a dedicatedpre-propose contract (separate pre-propose contracts for single-, multi-choice and overrule proposals exist), which manages deposits and makes sure that only DAO members can submit proposals.

Voting Power Registry

Instead of a single voting power module, Neutron DAO core contract interacts with theVoting Power Registry contract that keeps track of multipleVoting Vaults (see below).

Voting vaults

A voting vault is a smart contract that implements the DAO DAO voting module interface, namely, it is capable of:

1. Providing the total voting power at a given height,
2. Providing the voting power of an address at a given height.

The overall voting power of a given address is a sum of the voting powers that the address has in all of the registered voting vaults.

There are two types of Voting Vaults:

1. Real vaults,
2. Virtual vaults.

An example of areal vault is the Neutron Vault, which allows its users to directly bond and unbond NTRN tokens. (This is done without locking them, i.e., you can bond and unbond tokens with this vault with no unbonding period.)

In most cases, however, a Voting Vault does not directly store user funds; in this sense, such voting vaults can be called"virtual" vaults.**. For example, the Lockdrop vault does not allow users to directly bond or unbond LP tokens; instead it implements a relatively complicated query to multiple contracts to determine the amount of NTRN tokens that correspond to a certain amount of LP tokens at a given height.

Note: The voting power is based exclusively on the amount of NTRN tokens, regardless of the type of the vault. Below is the list of Voting Vaults that will be available at launch:

1. Neutron Vault;
2. Credits Vault
3. (virtual) — keeps track of the NTRN tokens that are vested in theCredits
4. contract.You can not add
5. tokens or remove tokens from this vault directly
6. ;
7. Lockdrop Vault
8. (virtual) — keeps track of the NTRN tokens that are locked in theLockdrop
9. contract. You can not add
10. tokens or remove tokens from this vault directly;
11. LP Vesting Vault
12. (virtual) — keeps track of the NTRN tokens that are vested in theLP Vesting
13. contract. You can not
14. add tokens or remove tokens from this vault directly;

15. Investors Vault
16. (virtual) — keeps track of the NTRN tokens that are vested in the earlybackers vesting contract
17. . You
18. can not add tokens or remove tokens from this vault directly.

Overrule proposals

N.B.: you need to read the subDAOs design below to understand this section.

TheOverrule proposal type has a low threshold (0.01 of the total voting power). It only allows to call theoverrule_proposal() method of a subDAO proposal contract.

Re-voting should be disabled for such proposals (execute immediately after the threshold is reached).

subDAOs

The Neutron DAO creates subDAOs by executing Neutron DAO proposals that containInstantiate messages for the subDAO contracts. At the launch time, only theMultisig-type subDAO will be available, which is similar to the Neutron DAO, but uses thecw4 voting module implementation for voting power (that's where the multisig logic is implemented).

One important feature of the subDAOs is that their proposals can be overruled by the Neutron DAO within a specified timelock period.

Timelocks & Overrules

For more info on Overrules, checkOverrules page. Proposals approved by a subDAO are timelocked. During the timelock period, the Neutron DAO can overrule any proposal by creating a new Overrule proposal; this proposal has a lower threshold than the regular proposals, and is executed immediately after reaching the threshold.

The timelock mechanism is implemented as follows. When creating a proposal, the user sends a regular proposal message to the subDAO pre-propose contract. This contract wraps the messages to be executed in aTimelockProposal message that is defined by theTimelock contract . When the proposal passes, the subDAO core contract does not execute the original messages; instead, it sends them wrapped in aTimelockProposal message to the Timelock contract.

The Timelock contract has 3 handlers:

- execute_timelock_proposal(proposal_id, msgs)
- : timelocks the given proposal messages, (permissioned, only by subDAO
- core contract);
- execute_execute_proposal(proposal_id)
- : executes the proposal if the timelock period has passed (permissionless);
- execute_overrule_proposal(proposal_id)
- : overrules the proposal (permissioned, only by the Neutron DAO).

When aTimelockProposal message is processed by the Timelock contract, the submission time is recorded in the state. The Timelock contract has a parametertimelock_period that defines how much time needs to pass before the proposal can be executed.

Important notes

1. The wasmd-level admin of the Timelock contract is the Neutron DAO core contract;
2. The owner of the Timelock contract is the Neutron DAO core contract;
3. The Timelock contract is instantiated by the pre-propose contract;
4. The subDAO address is queried from the pre-propose module during instantiation.

Security subDAO

There is a specialSecurity subDAO that can only executepause() methods on the following contracts:

1. All other subDAOs;
2. Reserve
3. contract;
4. Distribution
5. contract;

The Security subDAO implements a modified version of the single-choice proposal that only allows to sendpause() messages to smart contracts. Previous Neutron Core Releases Next Overrules