Admin Module

Overview

The Admin module in the Neutron network is a central governance tool, enabling the DAO to propose and execute pivotal operations. Developed using the cosmos-sdk , this module is upgraded to align with the Cosmos SDK 0.47 .

Concepts

Network Administration and Governance

Being the network's admin, our DAO is empowered to execute proposals that can significantly change the network state. These proposals can encapsulate a myriad of operations including but not limited to:

- Modifying parameters of a specific module, like adjusting transaction fees or validator incentives.
- A full spectrum of wasm
- proposals, leveraging the WebAssembly smart contract functionalities in Cosmos.
- Upgrade proposals
- , enabling seamless interoperability and data transfer between distinct blockchains.
- Execution of any sdk.msg
- , the standard message format in the Cosmos ecosystem.

Mechanism of Operation

We achieve governance through a dual mechanism approach. A message, structured as per cosmos-sdk's guidelines, is sent to the admin module using wasmbindings from MainDao. This message typically contains one of the two global types of proposals :

1) Legacy Proposal : Rooted in the x/gov module of the cosmos-sdk, certain modules like wasm can have predefined proposals. Classic examples include MigrateContract or DeleteAdmin . The intrinsic cosmos-sdk processes such proposals via a dedicated handler . The Admin module on Neutron, in line with these processes, can execute such proposals if they are part of a whitelisted group. Despite this mechanism being labeled deprecated in the recent sdk-47 update, its support remains due to certain modules (like ibc) not transitioning as per the sdk-47 guidelines.

2) Proposal : The advent of sdk-47 heralded a fresh approach. Actions, rather than being bound by predefined standards, could now be executed using the versatile sdk.Msg . This necessitated an overhaul in permission structures across modules. Modules have now integrated an authority field, signifying an address (or a group of addresses) that have been permissioned to dispatch critical messages, such as UpdateParams.

Implementation in Neutron

The Admin module in Neutron utilizes both the aforementioned proposal mechanisms. Specifically, these are executed within the msg.server of the admin module, ensuring seamless integration with the larger Neutron infrastructure. Our commitment is to keep abreast of the cosmos-sdk updates while preserving the unique governance structure that Neutron network requires.

Whitelisting

As soon as we want to control the list of proposals that may be executed via adminmodule, we have a simple whitelisting mechanism.

Challenges related to Cosmos SDK 0.47

Transitioning to Cosmos SDK 0.47 introduced several hurdles:

- Original Admin Module's Dormancy:
- Prior to V2
- , Neutron used the original Admin Module
- . This module, no longer actively maintained, needs an upgrade.
- Deprecation of x/params
- :
- With the new SDK version, the x/params
- module has been deprecated, pushing each Cosmos Module to implement custom parameter handling logic.
- Change in Governance Logic:
- The new SDK has moved away from the ProposalHandler
- logic in modules. Instead, the gov

- module can now issue direct messages to any Cosmos Module.

Enhancements & Solutions

ProposalExecuteMessage Binding

Introduced a new[ProposalExecuteMessage](#) binding, allowing the AdminModule to process any type of Cosmos message. A[signer verification](#) ensures authenticity.

Whitelisting of Executable Messages

Implemented a[whitelist mechanism](#) to ensure that only pre-approved messages are executed.

Legacy Proposals & Handler

For ensuring backward compatibility, we have retained[ClientUpdateProposal, UpgradeProposal, and ParamChangeProposal](#) . Additionally, a handler named MsgSubmitProposalLegacy has been[introduced](#) for pre-sdk47 proposals.

Revamped MsgSubmitProposal

The[MsgSubmitProposal](#) has been redesigned to handle any type of Cosmos message.

Streamlined Logic and Execution

We've removed unnecessary logic and the cache context, leading to immediate proposal execution within the msgServer/keeper and the removal of Active and Inactive queues.

Conclusion

With these updates, Neutron remains at the forefront of the evolving Cosmos ecosystem, ensuring our governance remains robust, safe and compatible.