Sudo Execution

One of the wonders of the Cosmos SDK isgovernance . Network participants can vote on proposals to decide the future of the network. Proposals can contain messages that will be executed based on the result of the voting.

We can define a smart contract entry point that can only be called by trusted native Cosmos modules. This entry point issudo . It can not be called by users or other smart contracts but only by Cosmos modules. This means thatsudo is useful for more than just governance.

First we need a msg type:

/// SudoMsg is only exposed for internal Cosmos SDK modules to call. /// This is showing how we can expose "admin" functionality than can not be called by /// external users or contracts, but only trusted (native/Go) code in the blockchain

[derive(Serialize, Deserialize, Clone, Debug, PartialEq, JsonSchema)]

[serde(rename_all =

"snake_case")] pub

enum

SudoMsg

{ MoveFunds

{ recipient :

String , amount :

Vec < Coin

, } , } Then the entry point:

[entry_point]

pub

fn

sudo (_deps :

DepsMut , _env :

Env , msg :

SudoMsg)

->

Result < Response ,

HackError

{ match msg { SudoMsg :: MoveFunds

{ recipient , amount }

=>

{ let msg =

BankMsg :: Send

{ to_address : recipient , amount , } ; Ok (Response :: new ( ) . add_message (msg)) } } } This can be tested as normal.

When usingmulti-test you will need to add an additional call to the contract wrapper:

pub

fn

contract_template ( )

->

Box < dyn

Contract < Empty

{ let contract =

ContractWrapper :: new (crate :: contract :: execute , crate :: contract :: instantiate , crate :: contract :: query ,) ; let contract_with_sudo = contract . with_sudo (crate :: contract :: sudo) ; Box :: new (contract_with_sudo) }

Proposal

The Smart contract must be instantiated before governance can execute it.

The interface for executing a smart contract via governance is similar to any proposal.

The JSON for the message defined earlier will need to be supplied with the proposal.

wasmd tx gov submit-proposal sudo-contract [contract_addr_bech32]

[json_encoded_migration_args]

[flags] json_encoded_migration_args accepts the JSON-encodedSudoMsg :

{ "move_funds" :

{ "amount" :

"100000" , "recipient" :

"wasm126kmp3ceapx2gxrju3uruxd2d440raxaz8xa90" } } Previous Testing Next Integration * Proposal