Lending Market Governance

Here is a high level overview of how the Canto Lending Market is governed:

1. The proposals are initiated on the Network side (Cosmos runtime) and voted on by network validators as specified in the SDK governance module. Lending protocol governance uses a custom Canto proposal type equivalent to Compound V2's governance proposal type.
2. When a proposal is approved, the GovShuttle module sends the proposal type to the EVM module where it can be retrieved by a smart contract call to a specific oracle address.
3. When the proposal is retrieved, it is stored in the queue and then executed in the same manner as traditional governance on Compound V2, outlined in Compound's Governor Bravo after a proposal is approved.
4.

Technical Overview & Details

Here is a technical overview of Canto Lending Market governance:

1. User uses CLI to submit proposal.cantod tx unigov submit-proposal “proposal text here” --address="address to map contract"
2. This submits a proposal using the governance keeper.
3. Users vote on the proposal.
4. If the proposal passes, the governance module handler sends the proposal to the GovShuttle proposal handler, which triggers the GovShuttle module to call the keeper functionAppendLendingMarketProposal
5. located in theproposals.go
6. file.
7. AppendLendingMarketProposal
8. function takes in theLendingMarketProposal
9. function from earlier and returns the address at which the Map Contract is deployed.
10. Lending Market can now use theQueryProp
11. method on the map contract to return a proposal struct. Importantly, the lending market can only query proposals that have passed through GovShuttle governance.
12.

Proposals are saved to the EVM as follows:

1. DeployMapContract
2. packs the arguments contained in the proposal usingProposalStoreContract.ABI.Pack
3. Then it creates a contract address usingCreateAddress(types.ModuleAddress, nonce)
4. It then creates a byte array with the packed data and passes that into a keeper call to the EVM withCallEVMWithData(ctx, types.ModuleAddress, nil, data, true)
5. Finally it returns the address where the Map Contract (shown below) with the initial proposal is deployed.
6.

```

Copy contract map { mapping proposalID : proposal

// only GovShuttle module can call AddProposal method function AddProposal(id, proposal) { // add new proposal to mapping here }

// anyone can call this method function QueryProp(uint id) { // return proposal from mapping here using ID } }

```

1. If a GovShuttle proposal has passed before, add a proposal to the Map Contract by using:

```

Copy CallEVM(ctx, contracts.ProposalStoreContract.ABI, types.ModuleAddress, *k.mapContractAddr, true, "AddProposal", sdk.NewIntFromUint64(m.GetPropId()).BigInt(), lm.GetTitle(), lm.GetDescription(), ToAddress(m.GetAccount()), ToBigInt(m.GetValues()), m.GetSignatures(), ToBytes(m.GetCalldatas()) )

```

Previous GovShuttle Module Next Audits Last updated11 months ago On this page