GovernanceFacet Code Style Findings

ON THIS PAGE

GFT-01C: Code Duplication

Type	Severity	Location
Gas Optimization	● Informational	GovernanceFacet.sol:L106-L112, L127-L133, L146-L151

Description:

The linked code commits a particular BIP using the same statements and a single literal value difference.

Example:

protocol/contracts/farm/facets/GovernanceFacet/GovernanceFacet.sol

SOL

Copy

```solidity
97  function commit(uint32 bip) external {
98      require(isNominated(bip), "Governance: Not nominated.");
99      require(!isActive(bip), "Governance: Not ended.");
100     require(!isExpired(bip), "Governance: Expired.");
101     require(
102         endedBipVotePercent(bip).greaterThanOrEqualTo(C.getGovernancePassThre
103         "Governance: Must have majority."
104     );
105
106     s.g.bips[bip].executed = true;
107
108     cutBip(bip);
109     pauseOrUnpauseBip(bip);
110
111     incentivize(msg.sender, true, bip, C.getCommitIncentive());
112     emit Commit(msg.sender, bip);
113 }
114
115 function emergencyCommit(uint32 bip) external {
116     require(isNominated(bip), "Governance: Not nominated.");
117     require(
118         block.timestamp >= timestamp(bip).add(C.getGovernanceEmergencyPeriod(
119         "Governance: Too early.");
120     require(isActive(bip), "Governance: Ended.");
```

```
    require(
        bipVotePercent(bip).greaterThanOrEqualTo(C.getGovernanceEmergencyThre
        "Governance: Must have super majority."
    );

    endBip(bip);
    s.g.bips[bip].executed = true;

    cutBip(bip);
    pauseOrUnpauseBip(bip);

    incentivize(msg.sender, false, bip, C.getCommitIncentive());
    emit Commit(msg.sender, bip);
}

function pauseOrUnpause(uint32 bip) external {
    require(isNominated(bip), "Governance: Not nominated.");
    require(diamondCutIsEmpty(bip),"Governance: Has diamond cut.");
    require(isActive(bip), "Governance: Ended.");
    require(
        bipVotePercent(bip).greaterThanOrEqualTo(C.getGovernanceEmergencyThre
        "Governance: Must have super majority."
    );

    endBip(bip);
    s.g.bips[bip].executed = true;

    pauseOrUnpauseBip(bip);

    incentivize(msg.sender, false, bip, C.getCommitIncentive());
    emit Commit(msg.sender, bip);
}
```

Recommendation:

We advise the code to be refactored to utilise a single `internal` or `private` function that accepts the literal changed as an argument instead, greatly optimising the bytecode size of the contract.

Alleviation:

The linked duplicate statements were refactored to a single `_execute` private function thereby alleviating this exhibit.

View Fix on GitHub

GFT-02C: Inefficient First Vote

Type	Severity	Location
Gas Optimization	Informational	GovernanceFacet.sol:L61

Description:

The first vote cast to a BIP by the proposer themselves is inefficiently done so as it invokes the `vote` function performing 4 redundant `require` checks.

Example:

protocol/contracts/farm/facets/GovernanceFacet/GovernanceFacet.sol

SOL

Copy

```solidity
/**
 * Proposition
**/

function propose(
    IDiamondCut.FacetCut[] calldata _diamondCut,
    address _init,
    bytes calldata _calldata,
    uint8 _pauseOrUnpause
)
    external
{
    require(canPropose(msg.sender), "Governance: Not enough Stalk.");
    require(notTooProposed(msg.sender), "Governance: Too many active BIPs.");
    require(
        _init != address(0) || _diamondCut.length > 0 || _pauseOrUnpause > 0,
        "Governance: Proposition is empty."
    );

    uint32 bipId = createBip(
        _diamondCut,
        _init,
        _calldata,
        _pauseOrUnpause,
        C.getGovernancePeriod(),
        msg.sender
    );

    emit Proposal(msg.sender, bipId, season(), C.getGovernancePeriod());

    vote(bipId);
}

/**
 * Voting
**/

function vote(uint32 bip) public {
    require(isNominated(bip), "Governance: Not nominated.");
    require(balanceOfRoots(msg.sender) > 0, "Governance: Must have Stalk.");
    require(isActive(bip), "Governance: Ended.");
    require(!voted(msg.sender, bip), "Governance: Already voted.");

    recordVote(msg.sender, bip);
    placeLock(msg.sender, bip);

    emit Vote(msg.sender, bip, balanceOfRoots(msg.sender));
}
```

Recommendation:

We advise the `propose` function to instead cast the vote directly by only using the `recordVote` and `placeLock` functions. If bytecode efficiency is desired, both functions can utilise a shared `_vote` function that is internally / privately available.

Alleviation:

The code was optimized as advised with a shared internal `_vote` function in the `VotingBooth` contract implementation.

View Fix on GitHub

PREV
FundraiserFacet.sol (FFE-C)

NEXT
InitBip0.sol (IB0-C) ›