BSGovernor Contract: Decentralized On-chain Governance

The `BSGovernor` contract is an on-chain governance system designed to facilitate decentralized decision-making for protocol upgrades, parameter adjustments, and critical operations. It combines **liquid democracy** (delegated voting power based on historical user/club performance) with **time-locked execution** for security-sensitive actions. Key objectives include:

- Enabling community-driven governance through proposals and voting.
- Restricting high-risk functions via configurable safeguards.
- Providing a transitional "interim governance" mode for bootstrapping.

Lifecycle

Proposals progress through four stages:

1. **Creation**:
    a. Users submit proposals with a target contract, function selector, and arguments.
    b. Requires voting power derived from historical leaderboard ranks.
    c. Anti-spam checks: Weekly proposal limits (4/user) and duplicate detection.
2. **Voting**:
    a. Voting period (`votingDuration`) starts; users cast votes with validated voting power.
3. **Finalization**:
    a. After the voting period, proposals are finalized if quorum and approval thresholds are met.
4. **Execution**:
    a. Approved proposals enter a **1-day timelock** before execution.
    b. Must be executed within **5 days** post-timelock to prevent expiration.

Voting System

1. Voting Power Calculation

- Derived from **historical leaderboard rank proofs**:
    - **Individual Ranks**: Proofs of being in the top `votingMaximumRank` in past `eligibleWeekCount` weeks.
    - **Club Ranks**: Proofs of leading a top-ranked club (member rank = 1, club rank <= `votingMaximumRank`). Each club leader has the right to vote to represent its club.
- Each valid proof grants **1 voting power unit**.
- **Interim Governor Bonus**: Automatically takes the maximum amount that standard users can have. (1 for individual, 1 for club per eligible week)

2. Mechanics

- **Total Eligible Votes**: Derived from historical individual/club leaderboard entries count and includes an interim owner bonus.
- **Quorum Threshold**: Minimum votes required = `(Total eligible votes * quorumThresholdPercent) / 100`.
- **Approval Threshold**: `yesVotes` must ≥ `(yesVotes + noVotes) * approvalThresholdPercent / 100`.

Key User Functions

Function	Description
`createProposal`	Submits a proposal with target, selector, and args. Requires voting power proofs.
`castVote`	Votes `yes`/`no` on active proposals using rank proofs.
`finalizeProposal`	Resolves proposals after voting ends; checks quorum/approval.
`executeProposal`	Executes approved proposals post-timelock.
`reactivateInterimGovernance`	Re-enables interim mode after 60 days of inactivity (interim owner only).

DAO Configuration

1. `quorumThresholdPercent`

- **Purpose**: Minimum percentage of total votes required for a proposal to be valid.
- **Range**: 20% – 60%.
- **Impact**: Higher values require broader participation; lower values make it easier to pass proposals.

2. `approvalThresholdPercent`

- **Purpose**: Minimum percentage of "yes" votes needed for approval.
- **Range**: 70% – 90%.
- **Impact**: Higher thresholds ensure stricter consensus; lower thresholds speed up decision-making.

3. `eligibleWeekCount`

- **Purpose**: Number of historical weeks considered for voting power.
- **Range**: 2 – 8 weeks.
- **Impact**: Shorter periods always allow the newest contributors to vote.

4. `votingMaximumRank`

- **Purpose**: Maximum rank (individual or club) eligible for voting power.
- **Range**: 100 – 1,000.
- **Impact**: Lower ranks concentrate power in top performers; higher ranks democratize influence.

5. `votingDuration`

- **Purpose**: Duration of the voting period.
- **Range**: 3 – 14 days.
- **Impact**: Shorter durations speed up governance; longer durations encourage broader participation.

6. `interimActive`

- **Purpose**: Enables/disables interim governance mode.
- **Impact**: When active, the interim owner can fast-track proposals.

7. `allowOnlyTrustedTargets`

- **Purpose**: Restricts proposal targets to trusted contracts.
- **Impact**: When enabled, only approved addresses can be targeted, reducing risks. Trusted address list is in the BS Token contract. List can be updated via setTrustedAddress

8. Restricted Function Selectors

- **Purpose**: Blocks critical functions (e.g., token transfers) from public proposals.
- **Impact**: Only the interim governor can propose/execute restricted functions, enhancing security.

Interim Governor

- **Role**: Initial owner with elevated privileges during bootstrapping.
- **Permissions**:
    - Skip voting period for non-restricted proposals (auto-approval).
    - Create proposals including restricted functions (voting required).
    - Reactivate interim mode after 60 days of inactivity.
- **Phase-Out**: Can be disabled when decentralized governance is stable (setInterimState).

Restricted Functions

- **Definition**: Critical function selectors blocked from public proposals/execution.
- **Defaults Include**:
  - Token transfers (`transfer`, `transferFrom`, approve).
  - Token burns (`burn`).
  - Rewarder treasury functions (`transferERC20ToTreasury`).
- **DAO Control**: Additional selectors can be restricted via `setRestrictedFunction`.
- **Execution**: Only the interim governor can create proposals involving restricted functions.