

Governance

KPR governance by design has a low overhead, it is not meant to be protocol intensive. The focus is simply on reviewing and approving jobs, and if absolutely required mitigate disputes or blacklist keepers.

Participants

Only bonded Keepers may participate in governance. To participate in governance a keeper must first `bond` KPR, once bonded they can `delegate` voting rights to either themselves or another party.

Managing Jobs

The core function of governance is to approve and include jobs, when liquidity is provided for a job, a proposal is automatically created to include them for review. Bonded keepers will review the contracts and decide if they should be included. It is important that jobs code defensively, assume keepers will only include your job to maximize their returns, as such maximum payment thresholds have been implemented.

```
1  /**
2   * @notice Allows governance to add new job systems
3   * @param job address of the contract for which work should be performed
4   */
5  function addJob(address job) external
```

```
1  /**
2   * @notice Allows governance to remove a job from the systems
3   * @param job address of the contract for which work should be performed
4   */
5  function removeJob(address job) external
```

Managing Keepers

As a last resort, governance has certain rights over managing Keepers, these include lodging disputes, slashing, revoking access, and resolving disputes.

Dispute Management

```
1  /**
2   * @notice allows governance to create a dispute for a given keeper
3   * @param keeper the address in dispute
4   */
5  function dispute(address keeper) external
```

```
1  /**
2   * @notice allows governance to resolve a dispute on a keeper
3   * @param keeper the address cleared
4   */
5  function resolve(address keeper) external
```

Slashing

```
1  /**
2   * @notice allows governance to slash a keeper based on a dispute
3   * @param bonded the asset being slashed
4   * @param keeper the address being slashed
5   * @param amount the amount being slashed
6   */
7  function slash(address bonded, address keeper, uint amount) public
```

Blacklist

```
1  /**
```

```
2  * @notice blacklists a keeper from participating in the network
3  * @param keeper the address being slashed
4  */
5  function revoke(address keeper) external
```