

# Summary

The boidpower oracle is an on-chain management contract for Boid-associated devices and accounts. This service is provided by the Boid-appointed registrar and the Boid-trusted validators. Validators form a consensus on the boidpower contribution quantity by assessing the contributions of individual devices to the overall Boid network. Devices shall contribute to any or none of the valid Boid-associated computational protocols. Boid accounts to gain power through associated devices. Validator performance is tracked based off of the quality of their consensus with other validators.

## Terminology

**Device :** Computational device on the boid network

**Account :** Boid token holder

**Protocol :** Computational protocol (e.g. BOINC, RAVEN mining). Protocols are found in the protocols table at the registrar account scope and the protocol\_type index. Correct protocol type must be found by matching protocol\_name in the `protocols` table. This is done so for adherence to eosio table primary indexing specifications

**Validator :** Reporter of device performance

**Power :** Quantitative device performance

**Rating/report :** Description of device contribution

**Weight :** The amount of trust given to a specific validator for a specific protocol. A minimum weight must be reached in summation of all contributing validators in order for a report to be considered valid

**Validation :** Power report that has reached consensus

**Round :** The time frame for which a report is valid. Nearest UTC hour since epoch measured in microseconds

# Contract initialization

## **regregistrar(registrar, tokencontract)**

- Only contract owner may register new registrar
- Register registrar account and associate with the BOLD token contract

## **addprotocol(protocol\_name, description, meta, difficulty)**

- Only registrar may add new protocols
- Must first call with `addprotocol('null', 'null protocol', '', 0);`

## **regvalidator validator)**

- Only registrar may add new validators
- Register validators

## **addvalprot validator, protocol\_type, weight)**

- Only registrar may add new protocols to a validator
- Add protocol for a validator and give the validator a specific validation weight. This specifies that the validator expects to validate reports for a specific protocol\_type
- Only valid protocol type shall be added to a validator except for the null protocol

# How to register new devices

## **regdevice(owner, device\_key, registrar\_registration)**

- Only device owner or power oracle registrar may register devices
- Register a device for which validators may submit reports
- Devices shall only be associated with valid Boid accounts already present in the boidtoken contract

## **regdevprot(owner, device\_key, protocol\_type)**

- Only device owner or power oracle registrar may register new protocol for their device

# What to do as a validator

**updaterating(*validator*, *account*, *device\_key*, *round\_start*, *round\_end*, *rating*, *protocol\_type*)**

- Only validator may submit new power rating
- Submit new power rating to be validated
- Ratings may only be submitted for the closest valid UTC hour measured in microseconds (i.e. a multiple of 3600e6)
- Validator ratings are considered valid if they are not an outlier with respect to the other ratings submitted for the same round

$q1 = \text{quartile } 1$

$q3 = \text{quartile } 3$

$iqr = q3 - q1$

$\text{outlier} = (\text{rating} > q3 + iqr) \text{ or } (\text{rating} < q1 - iqr)$

- Median value for a validated report is chosen as the rating submitted to the boidtoken contract for the associated device
- Any rating for a round further along in time will cause the round to move forward and unvalidated reports to be thrown out

## TODO

- Check account exists in boidtoken contract when new account is created through `regdevice()`
- Action to payout to validators
- Change `protocol_type` to be sha256 hash of protocol name