

## #

### Mint

## #

### Introduction

The incentive mechanism of POW is widely known and explicit: once a new block is produced, the block miner will acquire a certain amount of token as well as the accumulation of transaction fee in the block. As a POS blockchain network, the IRIShub has similar way to produce reward token but more complex mechanism to distribute the reward to each contributor.

In each block producing period, all POW miners compete to calculate their work proof and the fastest one will be the winner. Actually, all loser miners don't offer any positive help or collaboration to the winner miner, and they are just the competitors. So it is reasonable to grant all reward to the winner miner. However, in POS blockchain network, we can't do that. Because each block producing process is the collaboration of all validators and delegators, which means the benefit should be share by all these contributors. As for how to distribute reward token to contributors, we will document and implement it in [distribution](#) module.

The reward is composed by two parts, one is the collected transaction fee from the transactions in each block. Another one is regular inflation in each block, which will produce new tokens. The mint module is in charge of calculating the inflated token amount and add the inflated token to reward pool.

## #

### Inflation Calculation

## #

### Block Time

The block time is not the machine time, because different machines may not have exactly the same time. They must have some deviation more or less which will result in non-deterministic. So here the block time is the BFT time. Please refer to this [tendermint bft-timeopen in new window](#) for detailed description.

## #

### Inflation Rate

The inflation rate is assigned to 4% per year in genesis file. This value can be modified by governance. As for how to change the value by governance, please refer to [governance](#).

## #

### Calculation

This is the calculation equation:

$$\text{blockCostTime} = (\text{current block BFTtime}) - (\text{last block BFTtime})$$
$$\text{AnnualInflationAmount} = \text{inflationBasement} * \text{inflationRate}$$
$$\text{blockInflationAmount} = \text{AnnualInflationAmount} * \text{blockCostTime} / (\text{year})$$

The value of inflationBasement is specified in genesis file. By default its value 2000000000iris (2 billion iris, 1 iris equals  $1 * 10^6$  uiris), and its value will never be changed. Suppose blockCostTime is 5000 millisecond, and inflationRate is 4%, then the inflation amount will be 12683917 uiris (12.683917 iris)

## #

### Impact to users

The inflation calculation is automatically triggered by each block. So once a new block is produced, new tokens will be created and the loose tokens will increase accordingly. Users have no directly interface to affect this process.

There is a command line interface and one LCD restful APIs which can query total loose tokens amount.

### iris q staking pool

This is much faster, but it cannot get merkle proof and verify proof. So if you doesn't trust the connected full node, please don't use this interface.

# iris q staking pool--node

< iris-node-url> Example Output:

bonded\_tokens:"10000000000000000000" not\_bonded\_tokens:"0" /staking/pool

Please refer to LCD swagger document.

As for how to run a LCD node please refer to [LCD document](#) .