

Token Economics

Band Token and Use Cases

BandChain currently utilizes its native BAND token as the sole token on its network. The chain then uses the promise of receiving reward tokens as an incentive for validators to produce new blocks and submit responses to data requests. Additionally, any network participants can use the tokens in 3 ways:

1. Bond tokens to become validators
2. Delegate tokens to another validator to earn a portion of the collected fees and inflationary rewards

By using this website, you agree to our [Cookie Policy](#).



DOCUMENTATION

◆ Introduction

◆ **Whitepaper**

Introduction

Terminology

System Overview

Token Economics

Search

ON THIS PAGE

Band Token and Use Cases

Inflation

Validators and Stakers

Community Fund Pool

Slashing

Inflation

BandChain applies an inflationary model on the BAND token to incentivize network participation by the token holders. The desired outcome of this model is that token holders will opt to stake their coins on the network, rather than solely focusing on trading, or doing nothing with it at all. The specific inflation parameters currently mirrors that of the Cosmos network; namely, the annual inflation rate ranges from 7% to 20%, and is adjusted to target to have 66% of the total supply of BAND token staked.

To illustrate how inflation incentivizes staking, imagine we have a network participant with a certain amount of holding. With inflation, if they choose to not use their coins to participate in the network's activities, they will find that the percentage of their holding with respect to the total supply decreases over time. However, if they decide to stake their coins, they will be given a share of coins proportional to the inflation, meaning their total token holding ratio will now remain relatively unchanged.

Validators and Stakers

Similar to in other Cosmos-based blockchain, one of the roles of Bandchain's validators is to provision new blocks and process transactions. By performing those tasks, they earn BAND tokens as a reward. In the case of block provisioning, the reward comes from the tokens newly minted on that block. Conversely, the reward for processing transactions comes from the [fee](#) that each validator chooses to set. Note that a percentage of the total block reward is diverted to the [community fund pool](#).

Those who do not wish to become validators themselves can still earn a portion of the validator rewards by becoming delegators. This is done through staking their holding on the network's validators. By doing so, they will share the revenue earned by those validators.

The amount of reward each validator receives is based on the total amount of tokens staked to them. Before this revenue is distributed to their delegators, a validator can apply a commission.



However, while delegators share the revenue of their validators, they also share the [risks](#). Thus it is imperative for potential delegators to understand those risks, and that being a delegator is not a passive task. Some actions that a delegator should perform are:

- **Perform due diligence on the validators you wish to stake on before committing:** If a validator you staked on misbehaves, a portion of the validator's staking, including those of their delegators, are [slashed](#). Therefore, it is advisable for delegators to carefully consider their staking choices.
- **Actively monitor the validators you've committed to:** Delegators should ensure that the validators they delegate to behave correctly, meaning that they have good uptime, do not double sign or get compromised, and participate in governance.
- **Participate in network governance:** Delegators are expected to participate in network governance activities. A delegator's

cast their vote, they will inherit the vote of the validators they staked on. If they do vote, they instead override the vote of those validators. Delegators therefore act as an important counterbalance to their validators.

Community Fund Pool

Two percent of the total block rewards are diverted to the community fund pool. The funds are intended to promote long-term sustainability of the ecosystem. These funds can be distributed in accordance with the decisions made by the governance system.

Slashing

If a validator misbehaves, their delegated stake will be partially slashed. There are three reasons why a validator may get slashed; excessive downtime, double signing, and unresponsiveness. The first two are derived



Excessive Downtime

One of the validators' main functions is to propose and subsequently commit new blocks onto the chain. Thus, if a validator has not participated in more than

`[MIN_SIGNED_PER_WINDOW]` of the last

`[SIGNED_BLOCK_WINDOW]` block proposals and commits, we will consider that they are not properly performing their function. As a consequence, we will slash the total value staked to them by `[SLASH_FRACTION_DOWNTIME]`.

Double Signing

During a double signing, the block proposer in the consensus round attempts to create two conflicting blocks and broadcast them to the network. If there are any other validators who stand to profit from this double block-proposal, they will vote then for both blocks.

If the votes passed, nodes on the network would see 2 different blocks at the same height, each with different contents and hashes. From this

The consequence of this is there will now be two versions of the “truth”.

To prevent such attempts at double signing, Cosmos, and by extension BandChain, is implemented so that any validators that double-sign are slashed, with the slashed amount being `[SLASH_FRACTION_DOUBLESIGNING]` of all tokens bonded to them.

Unresponsiveness

Finally, we also slash validators if they consistently fail to respond to data requests. If a validator failed to respond to `[CONSECUTIVE_UNRESPONSIVE_REQUESTS]` consecutive requests, they will be slashed an amount equal to `[SLASH_FRACTION_UNRESPONSIVENESS]`

Gas and Network Fee

In the Cosmos SDK, gas is a unit that is used to track the consumption of resources during

By using this website, you agree to our [Cookie Policy](#). 

during read/write operations, or whenever a

computationally expensive operation is performed. The purpose of gas is twofold:

1. To prevent blocks from consuming excessive resources, and to ensure that the block will be finalized
2. To prevent abuse from a malicious actor on the user side

Gas consumed during the execution of a message is priced by both transaction senders and validators.

First, the gas price is set by the transaction sender. This results in a fee, the value of which is equal to the gas value multiplied by the price.

Meanwhile, each block validator can subjectively set the minimum gas fee that must be reached for them to process the transaction and choose whatever transactions it wants to include in the block that it is proposing, as long as the total gas limit is not exceeded.

During periods when there is a high amount of transactions that is waiting to be processed, a

By using this website, you agree to our [Cookie Policy](#). 

try to get their transaction included in the

upcoming block. They do this by trying to have a higher proposed fee than other senders. Note that all pending transactions will eventually be sent, regardless of the fee amount proposed by the sender. The amount of gas the sender is proposing to pay only determines the likelihood that their transaction will be sent first.

Gas Calculation

This section we will compare gas usage on each message type to compare how many fee that user need to pay to do these actions.

Message	Estimated fee
MsgSend	~70k
MsgDelegate	~120k
MsgWithdrawReward	~100k
MsgTransfer(IBC)	~85k

By using this website, you agree to our [Cookie Policy](#). ×

Message	Estimated fee
MsgRequestData*	>500k upto 5m
MsgUpdateClient + MsgReceivePacket (Oracle request packet)*	>560k upto 5m
MsgCreateDataSource / MsgEditDataSource**	30k - 100k
MsgCreateOracleScript / MsgEditOracleScript**	500k - 2m

(*) The gas is used on MsgRequestData or process oracle request packet based on complexity of oracle script and how many validator need to query data on this request.

(**) Based on size of data source and oracle script mostly data source should be smaller than oracle script.



Found an Issue?

Help us improve this page by suggesting edits on GitHub.

By using this website, you agree to our [Cookie Policy](#).



PREVIOUS



**System
Overview**

NEXT



**Protocol
Messages**



BAND PROTOCOL



bandprotocol.com

This website is maintained by [Band Protocol](#). The contents and opinions of this website are those of Band Protocol.

By using this website, you agree to our [Cookie Policy](#).

