Celestia hardfork process

Blockchain networks often times need to upgrade with new features which require coordination work among the validators prior to activating the upgrades.

This process is called a hardfork or a network upgrade. In those events, the Celestia Labs team will be coordinating with the validators on what they need to do in order to be ready for an upcoming hardfork.

Hardforks are not backward-compatible with older versions of the network software which is why it is important that validators upgrade their software to continue validating on the network after the network upgrades.

General process

The general process can be broken down into several components:

- Hardfork specifications and features (defined by description of features and code implementation of those features).
- Binary used to add those features (a new binary release with those features will be provided by Celestia team in order for validators to upgrade their nodes to the new binary).
- A block number for when the network upgrades (even if validators upgrade their binary to be hardfork ready, the network upgrade does not happen right away, but some short time in the future at a specific block number).
- Testing of the features (happens on testnets first prior to activating on mainnet in order to ensure the network can upgrade securely).

The two testnets were hardforks are deployed on are:

- [Arabica devnet](#)
- [Mocha testnet](#)

Mocha hardfork

Celestia is planning the Mocha Hardfork upgrade on the Mamaki Testnet. This hardfork is unique as it will reset the Mamaki network to block 0 while maintaining the existing state and also will rename Mamaki to Mocha.

The new chain-id will bemocha .

You can find the[release logs for consensus nodes on the celestia-app releases page](#).

The most exciting feature included is setting the stage for Blobstream on Mocha.

Validators will need to generate 2 new keys in order to be Blobstream-ready. Note that for the Mocha Hardfork, Blobstream will not launch yet so you can swap those keys after for new ones if needed. The keys needed are:

- 1 EVM key
- 1 Celestia key

So, in order for this to happen, validators will need to maintain two new keys in order to have a successful upgrade.

Those two keys will need to be added to 2 new flags oncelestia-app :

- --evm-address
- : This flag should contain a0x
- EVM address. [][[Edit this page on GitHub]](#) Last updated: [Previous page SystemD](#) [Next page Build modular](#) []