

Credible Neutrality is a property, meaning that a system is unbiased to all humans. It means that there is no owner or monarch that has or ever had any favorable position. Since the vast majority of Ethereum Tokens were deployed in such a way as to mint all of the tokens to the owner, they are not Neutral but instead they are biased, like centralized corporations.

We already saw the disasters that this has caused for Ethereum: token devs who own contracts have minted and dumped coins, they have looped their money through ICOs to print tokens for free, airdropped countless of their own token to themselves, or just kept the vast majority of their tokens in order to dump on others later. It is very difficult to make a 'neutral' token because if you do not give all of the tokens to yourself, how do they get created in the first place?

You would need to generate tokens in a manner that were neutral. There are a few ways to do this, most cause network congestion issues. For example: one could create a token that starts with zero supply and has a mint() method that could be called at any time by anyone. This would then distribute all of the tokens neutrally to those persons. However, if those tokens gained value, others could limitelessly mint() more to bring the cost down. Furthermore, it would be rate limited by the throughput of the Ethereum network bandwidth and gas costs.

In order to solve those issues, you could add a 'limiter' to the mint() method, such as a restriction that required that a Proof of Work be done in order to successfully call the method and earn the tokens. You could then have that Proof of Work difficulty automatically adjust and you could make the token rewards diminish over time in order to reach a hard cap limit. Then, you would have a totally credibly neutral token on the Ethereum Network. Everyone could credibly prove that there is no party that could corrupt, lie, scam, cheat, or steal these tokens from anyone else. There would be no ICO or Airdrops to cheat either, just pure PoW mining.

You could then take that a step further and build mining pools, so that mining users could call the mint() method through a pool (like a relay) so the pool pays the gas, not the miner. Of course the pool would also distribute shares and rewards evenly amongst the miners based on how much they contributed over a time slice.

I and many other Etheareans extensively researched this and built upon this over the last 2 years and there is a plethora of software that came out of it. Miners, Pools, contracts, etc. PM me for any resources.

For example, the etherscan tx for a batched pool payment looks like this:

<https://etherscan.io/tx/0xc674d0e4a58216f004c9c74f0563f1efb154e86ae7846c6564cc70a8f7568f06>

The etherscan tx for a mint() by a pool taking a fee looks like this:

<https://etherscan.io/tx/0x8aadf4dacd6ab85a9d7895ba79ba50711136a8028a2ce88ee980a32a9eee988e>

And the open source mining pool code that interfaces with Ethereum would look like this:

[GitHub](#)

### **[0xbitcoin/tokenpool](#)**

A pool mining coordinator that communicates with mining software using JSONRPC - 0xbitcoin/tokenpool

Furthermore, the coolest aspect: It was impossible to make neutral tokens before. However now, anyone who wants to create a credibly neutral token that is non-mined can do so by creating an 'ICO token contract' which allows users to burn existing pure-mined neutral PoW ERC20 tokens in order to then mint those new tokens. Then, those new tokens will also be competely neutral, but they can have new usecases and prices like any other new token. Those tokens would be objectively less riskier and immune to human corruption unlike tradition ICOed / airdropped tokens that are non-neutral and started all in the hands of the deployer...