Many cryptocurrency fundamentalists believe that true scarce cryptocurrency shall be distributed purely by mathematical challenges, not by human hand-waving by a central bank. In other words, they believe cryptocurrency should be minted by pure mining, not by creating it from thin air and assigning it to a single account.

Traditionally, all ERC20 tokens are centrally banked, created from thin air and assigned to their respective deployer upon launch of the contract to then be distributed by this monarch in an ICO sale or airdrop. This allow for human corruption by various methods, not limited to airdropping the tokens to anonymous accounts that they secretly control (we will call these 'Secret Accounts') or by participating in the ICO using their own Secret Accounts and collected the spent ETH at the end, buying into their own ICO for free.

One obvious but under-used method for minting and generating tokens is unbiased Proof of Work via pure mining. If a token were to be distributed by pure mining like Dogecoin or Litecoin, it would be mathematically provable that the initial rules of engagement are unbiased to specific Accounts, just as those currencies are. This way there would be no 'monarch', no account that starts with all of the tokens to distribute them arbitrarily. Instead, the tokens would be distributed using pure mathematics. Whoever can provide the math solutions gets the tokens and the currency would be provably fair, as much as other pure mined cryptocurrencies are.

Ethereum EIP918 describes a sound and heavily stress-tested methodology for an auto-adjusting autonomous pure mined cryptocurrency that is also an ERC20 token. This is unheard of and a breakthrough in Ethereum technology, as simple as it may seem.

Ethereum Improvement Proposals

EIP 918: Mineable Token Standard

Details on Ethereum Improvement Proposal 918 (EIP 918): Mineable Token Standard

This is accomplished by removing the line of code from the ERC20 token contract which assigns all tokens to the deployer. Then, a method called mint() is added which is just a faucet, giving token to the caller. This mint() method is gated by a Proof of Work challenge which auto adjusts its difficulty (target number) so as to target a specific time (number of eth blocks.)