If an ERC20 contract has a vulnerability, we want to redeploy the contract but hence (in most cases) also have to copy the entire storage to this new address. Most implementations simply upload all accounts and their associated balances to the chain, but this costs a lot of gas and many accounts will be untouched forever since users forgot about their tokens or they are dust amounts of tokens.

I created a proof of concept using Merkle Trees. A new ERC20 contract is uploaded to the chain with a certain Merkle Root. Users can now prove that they had an address associated with a balance by uploading the Merkle Proof to the chain. This means that: 1) users pay for gas themselves (if this is ethical, that is debatable) 2) only users who wish to get their tokens back have the incentive to go on-chain and 3) users can forever claim their tokens, the only requirement is connecting to an Ethereum archive node.

Github

Medium article