AERON

Standard of aviatvion safety powered by blockchain.
Suggested improvements:

0 Update compliler's version, to the moment 0.4.17

1 Change all constructions like if (_to == 0x0) revert(); to require - as this error doesn't require additional gas

2 if (_value <= 0) revert(); - useless construction, as _value -uint256 is unsigned type of data and cannot be less than 0. Also check if balance is more than requested _value is not required, as SafeMath will return an error if the when a negative number is received

3 Inherit from ERC20 and realize method allowance and event Approval for the services which work with smart contracts based on ERC20

4 Change constructions SafeMath.safeSub SafeMath.safeAdd to balanceOf[msg.sender].sub(_value) balanceOf[msg.sender].add(_value). To make it add using SafeMath for uint to the class; - it adds methods SafeMath to all types of uint 8-256

5 Make method public balanceOf, save balances in map balances

Questions:

1. Why do you need methods freeze / unfreeze?
2. Do you need paranoia mode - change of smart contract's owner?
3. There is burn method, and no additional emission is expected?

GitHub