Audit report for Paresh Masani



UI / UX / Frontend / Back end / Ecommerce / Blockchain



Technical requirments to ARAW TOKEN:

- Ticker ARAW
- Done
- Token type ERC20
- Done
- ICO token price 1 ARAW = \$0.01

Total tokens 5,000,000,000

- Done
- Available for token sale 3,500,000,000 (70%)
- Whitelist YES
- Accepts ETH

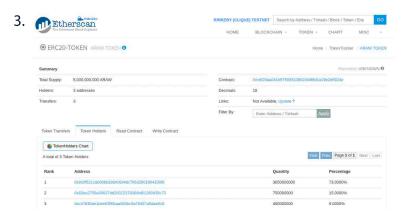
Smart contract ARAW TOKEN issues list:

1. Unused variable

uint public _totalSupply;

2. Symbol = "ARAW";
name = "ARAW TOKEN";

In requirements I didn't saw "ARAW TOKEN", I think name and symbol should be "ARAW".



Not all tokens are emitting when contract is deployed. What I mean is total supply = 5,000,000,000, but distributed only 4,850,000,000



4. Lines in token constructor:

balances[msg.sender] = 3650000000 * (10 ** decimals); balances[reservedTokensAddress] = 750000000 * (10 ** decimals); balances[foundersTokensAddress] = 450000000 * (10 ** decimals);

5. There is a contract in a project named "Pausable" which can stop all token transferring. There was no such thing in requirements and this contract might stop all token transfers include selling process, because it is implemented in transfer function. I'd strongly recommend to remove this contract from project.

```
6. require(balances[msg.sender] >= tokenLock); balances[msg.sender] = balances[msg.sender].sub(tokenLock);
```

Second line makes the same check as the 1st line.

```
7. modifier checkAfterICOLock ()
{
    if (msg.sender == owner)
    {
        __;
    }
    else if (msg.sender == reservedTokensAddress)
    {
        if (now<reservedTokensAddressLockedPeriod)
        {
            revert();
        }
        __;
    }
    else if(msg.sender == foundersTokensAddress)
    {
        if(now<foundersTokensLockedPeriod){
            revert();
        }
        __;
    }
    __;
}
</pre>
```

Modifiers should use "require" instead "if =>revert" or "if => throw". And there is no sense to check owner. So this code might look simply as that:

```
modifier checkAfterICOLock () {
  if (msg.sender == reservedTokensAddress)
    require (now >= reservedTokensAddressLockedPeriod)
  if(msg.sender == foundersTokensAddress)
    require (now >= foundersTokensLockedPeriod)
  _;
}
```



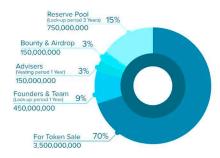
8. function privateSale and function releasePrivateLockToken.

I think there's a broken logic because "_to" can release all transferred token in any time, variable "tokenLock" makes no sense in this function.

 function releaseTokenAdvisor(uint256 percent) internal returns(bool)

This function doesn't returns anything, so removing this part(returns(bool)) doesn't affects the functionality at all.





Bounty & Airdrop part are included in owner balance as a For Token Sale. It can be done this way, but it isn't as on provided diagram.

11. In this project owner contains almost all tokens. And when you will transfer ownership, tokens will stay in previous owner. I think it's an issue, because function privateSale now is not available to be used by new owner because he hasn't tokens to call it (require(balances[msg.sender] >= tokenLock);)

The best way is create another address with name like "tokenHolder" from where tokens will be transferred.

12. function "releaseadvisorsTokensAddress" can be simplified, right now it looks like ovesized soltuion from the other project.

```
13. function releaseTokenAdvisor(uint256 percent)
internal
returns(bool)
{
    uint256 releasedTokens = (percent.mul(totalAdvisorsLockedTokens)).div(100);
    require(advisorsLockedTokensBalance >= releasedTokens);
    balances[advisorsTokensAddress] = balances[advisorsTokensAddress].add(releasedTokens);
    advisorsLockedTokensBalance = advisorsLockedTokensBalance.sub(releasedTokens);
    emit Transfer(advisorsTokensAddress, advisorsTokensAddress, balances[advisorsTokensAddress]);
}
```

Emit is wrong, require(advisorsLockedTokensBalance >= releasedTokens); is not necessary.

14. Variable "reservedTokensAddressLockedPeriod" lus is responsible for when "reservedTokens" might be transfered from "reserve wallet". here is a description of this functional:

```
if (now<reservedTokensAddressLockedPeriod)
     {
        revert();
    }</pre>
```

It could work correnctly, but during the deploy if a contract "reservedTokensAddressLockedPeriod" =0 and will get changed into "now + 1095 days" only when ICO ends. So till the end of ICO there will be an opportunity to transfer reserved tokens, and that seems to be incorrect. The same issue with foundersTokensLockedPeriod.



15. This contract can't accept ETH as noted in requirements, no whitelist.

Conclusion:

It is easyer to create a new contract than fix the current one, because technical requirments aren't done properly, and the code in my opinion is stiched from the other smart contracts and is overloaded by unnecessary lines that aren't doing anything but however, might become an issue.