V1.0

Sirin

Audit

Dean Eigenmann
November 16, 2017

Disclaimer

The audit makes no statements or warranties about utility of the code, safety of the code, suitability of the business model, regulatory regime for the business model, or any other statements about fitness of the contracts to purpose, or their bugfree status. The audit documentation is for discussion purposes only.

Overview

The following contracts were audited in this project. The rest were ignored because they are assumed to be secure as they stem from previously audited projects:

- SirinCrowdsale.sol

Audit Results

No major issues could be found. However there were 2 things of note:

1) The ownership of the token can never be transferred from of the crowdsale.
2) The bancor contracts are not identical to those within the bancor repository.

We suggest using the bancor contracts found in the bancor repository.

SirinCrowdsale.sol

The SirinCrowdsale contract is responsible for all the main sale logic. This contract inherits from Open Zeppelin's FinalizableCrowdsale.

The FinalizableCrowdsale inherits from Ownable and Crowdsale contracts and uses SafeMath for all uint256.

Modifiers

onlyWhileSale

Ensures the isActive function returns true.

Constructor

- Sets the start and end times for crowdsale.
- Sets the various addresses where token allocations will be transferred to.

getRate

Returns the specific token price at different times.

Finalization

This function handles the finalization of the crowdsale.

Tokens are issued to the addresses found in the presaleGranteesMap, this is iterated using the presaleGranteesMapKeys array.

The new total supply is calculated, this is done by increasing the current total supply by 60%.

Then multiple allocations take place to predefined wallets:

1. 10% of the new total supply is issued to founders.
2. 10% of the new total supply is issued to the OEM wallet.
3. 5% of the new total supply is issued to the bounties wallet.
4. 35% of the new total supply is issued to the reserve wallet.

Finally token transfers are re enabled.

For the token transfers to be enabled, the owner of the token must be the crowdsale. This is done by creating the token in the crowdsale contract. However there is currently no method to change the owner of the token to some other address once the crowdsale has finished.

getTotalFundsRaised

Returns the amount of wei raised added to the fiatRaisedConvertedToWei variable.

isActive

Ensures that the current time is between the start and end time.

addUpdateGrantee

Adds or updates a grantee to the presaleGranteesMap, also adds the address to the corresponding map of keys.

If the grantee does not exist, it is ensured that the maximum grants has not yet been reached and the address is then stored.

Finally the amount of tokens granted is then stored in the mapping, corresponding to the address.

Notes: This function can only be called when sale has already been started.

deleteGrantee

Deletes a grantee from the presaleGranteesMap, as well as deleting the address from the corresponding array storing all the addresses.

The array is holding all addresses is then shifted, this ensures there are no gaps left.

Notes: This function can only be called when sale has already been started.

setFiatRaisedConvertedToWei

Updates the fiatRaisedConvertedToWei variable to the passed parameter.