# FourRx Smart Contract Preliminary Audit Report

## Project Synopsis

| | |
|---|---|
| **Project Name** | **FourRX** |
| **Platform** | Ethereum, Solidity |
| **Github Repo** | https://github.com/FourRX/4rx/blob/master/contracts/ICO/ICOContract.sol<br><br>https://github.com/FourRX/4rx/blob/master/contracts/4RX/FourRXToken.sol |
| **Deployed Contract** | Not Deployed |
| **Total Duration** | 4 Days |
| **Timeline of Audit** | **16 May 2021 to 20 May 2021** |

## Contract Details

| | |
|---|---|
| **Total Contract(s)** | 2 |
| **Name of Contract(s)** | FourRXToken, ICOContract |
| **Language** | Solidity |
| **Commit Hash** | **bf96812acb07dd3081e73c9a88176cc3d1a07312852a9a8221a10623ef2adc763419857a37d33797** |

**Contract Vulnerabilities Synopsis**

| Issues | Open Issues | Closed Issues |
|---|---|---|
| Critical Severity | 0 | 0 |
| Medium Severity | 0 | 0 |
| Low Severity | 6 | 0 |
| Information | 2 | 0 |
| Total Found | 8 | 0 |

# Detailed Results

The contract has gone through several stages of the audit procedure that includes structural analysis,  automated testing, manual code review etc.

All the issues have been explained and discussed in detail below. Along with the explanation of the issue found during the audit, the recommended way to overcome the issue or improve the code quality has also been mentioned.

# A. Contract Name: ICOContract

## High Severity Issues
**None Found**

## Medium Severity Issues
**None Found**

## Low Severity Issues
### A.3 Return Value of an External Call is never used Effectively
**Line no - 43, 49, 59, 93, 102**

***Explanation:***

The external calls made in the above-mentioned lines do return a boolean as well as other imperative values that could effectively indicate whether or not the external call was successful.

However, the ICO contract does not use these return values effectively in some instances of the contract.

```
59          _token.transfer(BURN_ADDRESS, balanceAfter.sub(balanceBefore).div(2));
60
```

***Recommendation:***

Effective use of all the return values from external calls must be ensured within the contract.

### A.4 External Visibility should be preferred
***Explanation:***

Those functions that are never called throughout the contract should be marked as ***external*** visibility instead of ***public*** visibility.
This will effectively result in Gas Optimization as well.

Therefore, the following function must be marked as **external** within the contract:

- ***purchase***
- ***recoverTokens***

***Recommendation:***

If the PUBLIC visibility of the above-mentioned functions is not intended, then the EXTERNAL Visibility keyword should be preferred.

### A.5 constructor does not include Zero Address Validation

**Line no: 250-253**

***Explanation:***

The **constructor** initializes some of the most imperative state variables, i.e., ***fourRXToken*** address, ***pancakeV2Pair*** *address* in the ICO contract.

However, during the automated testing of the contract, it was found that the constructor doesn't implement any Zero Address Validation Check to ensure that no zero address is passed while initializing this state variable.

```
ICOContract.constructor(address,address)._pairAddress (flat_ICO.sol#660) lacks a zero-check on :
                - pancakeV2Pair = _pairAddress (flat_ICO.sol#667)
ICOContract.recoverTokens(address,address).recipient (flat_ICO.sol#729) lacks a zero-check on :
                - recipient.transfer(address(this).balance) (flat_ICO.sol#731)
```

***Recommendation:***

Zero address validation should be implemented to ensure no invalid address passed as arguments.

## Informational

### A.6 NatSpec Annotations must be included

***Description:***

The smart contracts do not include the NatSpec annotations adequately.

***Recommendation:***

Cover by NatSpec all Contract methods.

# B. Contract Name: FourRXToken

## High Severity Issues
**None Found**

## Medium Severity Issues
**None Found**

## Low Severity Issues
### A.3 A.6 Constant declaration should be preferred
**Line no- 53 to 63**

***Description:***

State variables that are not supposed to change throughout the contract should be declared as **constant.**

***Recommendation****:*

The following state variables need to be declared as **constant**, unless the current contract design is intended.

- ***API_URL***

### A.4 External Visibility should be preferred
***Explanation:***

Those functions that are never called throughout the contract should be marked as ***external*** visibility instead of ***public*** visibility.
This will effectively result in Gas Optimization as well.

Therefore, the following function must be marked as **external** within the contract:

- ***requestPrice***
- ***fulfill***

***Recommendation:***

If the PUBLIC visibility of the above-mentioned functions is not intended, then the EXTERNAL Visibility keyword should be preferred.

### A.5 constructor does not include Zero Address Validation
**Line no: 250-253**

***Explanation:***

The **constructor** initializes one of the most imperative state variables, i.e., ***oracle address*** in the Token contract.

However, during the automated testing of the contract, it was found that the constructor doesn't implement any Zero Address Validation Check to ensure that no zero address is passed while initializing this state variable.

```
FourRXToken.constructor(address)._oracle (flat_Token.sol#1538) lacks a zero-check on :
                - oracle =  _oracle (flat_Token.sol#1543)
```

***Recommendation:***
Zero address validation should be implemented to ensure no invalid address passed as arguments.

# Informational

## A.6 NatSpec Annotations must be included

***Description:***
The smart contracts do not include the NatSpec annotations adequately.

***Recommendation:***
Cover by NatSpec all Contract methods.

# Automated Test Results

```
FourRXToken.oracle (flat_Token.sol#1532) shadows:
        - ChainlinkClient.oracle (flat_Token.sol#1281)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variable-shadowing
```

```
ICOContract.addLiquidityAndBurn(uint256,uint256) (flat_ICO.sol#673-693) ignores return value by fourRXToken.approve(address(pancakeV2Router),tokenAmount) (flat_ICO.s
ICOContract.addLiquidityAndBurn(uint256,uint256) (flat_ICO.sol#673-693) ignores return value by pancakeV2Router.addLiquidityETH{value: ethAmount}(address(fourRXToken
ount,0,0,address(this),block.timestamp) (flat_ICO.sol#681-688)
ICOContract.addLiquidityAndBurn(uint256,uint256) (flat_ICO.sol#673-693) ignores return value by _token.transfer(BURN_ADDRESS,balanceAfter.sub(balanceBefore).div(2))
.sol#691)
ICOContract.purchase() (flat_ICO.sol#711-727) ignores return value by fourRXToken.transfer(msg.sender,coinAmount) (flat_ICO.sol#725)
ICOContract.recoverTokens(address,address) (flat_ICO.sol#729-736) ignores return value by token.transfer(recipient,token.balanceOf(address(this))) (flat_ICO.sol#734)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return
```

```
        - ERC20.decreaseAllowance(address,uint256) (flat_Token.sol#459-462
addr(bytes32) should be declared external:
        - ENSResolver.addr(bytes32) (flat_Token.sol#1249)
requestPrice() should be declared external:
        - FourRXToken.requestPrice() (flat_Token.sol#1548-1561)
fulfill(bytes32,uint256) should be declared external:
        - FourRXToken.fulfill(bytes32,uint256) (flat_Token.sol#1566-1571)
```

```
Parameter FourRXToken.fulfill(bytes32,uint256)._requestId (flat_Token.sol#1566) is not in mixedCase
Parameter FourRXToken.fulfill(bytes32,uint256)._price (flat_Token.sol#1566) is not in mixedCase
Variable FourRXToken.API_URL (flat_Token.sol#1536) is not in mixedCase
```

```
Function IPancakeV2Pair.DOMAIN_SEPARATOR() (flat_ICO.sol#40) is not in mixedCase
Function IPancakeV2Pair.PERMIT_TYPEHASH() (flat_ICO.sol#41) is not in mixedCase
Function IPancakeV2Pair.MINIMUM_LIQUIDITY() (flat_ICO.sol#58) is not in mixedCase
Function IPancakeV2Router01.WETH() (flat_ICO.sol#82) is not in mixedCase
Parameter ICOContract.recoverTokens(address,address)._tokenAddress (flat_ICO.sol#7
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformar
```