

# Smart Contract Security Audit V1

## xfuzion Token Smart Contract Audit

Oct 5, 2023



<https://saferico.com/>

[business@saferico.com](mailto:business@saferico.com)

[https://t.me/SFI\\_ANN](https://t.me/SFI_ANN)

—

# Table of Contents

## **Table of Contents**

## **Background**

## **Project Information**

Token Information

Executive Summary

## **File and Function Level Report**

**File in Scope:**

## **Issues Checking Status**

Severity Definitions

Audit Findings

## **Automatic testing**

Testing proves

Inheritance graph

Call graph

## **Unified Modeling Language (UML)**

**Functions signature**

**Automatic general report**

## **Conclusion**

## **Disclaimer**

# Background

The purpose of the audit was to achieve the following:

- Ensure that the smart contract functions as intended.
- Identify potential security issues with the smart contract.

The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

## Project and Token Information

- **Platform:** Pulse Chain
- **Name:** xfuzion
- **Language :** solidity
- **Contract Address:** 0x24BC7AeDCC69813B5876934314D708D4254652ab
- **Code Source:**  
<https://scan.pulsechain.com/address/0x24BC7AeDCC69813B5876934314D708D4254652ab/contracts#address-tabs>

## Executive Summary

According to our assessment, the customer`s solidity smart contract is **Well-Secured**.

Well Secured	✓
Secured	
Poor Secured	
Insecure	

Automated checks are with remix IDE. All issues were performed by the team, which included the analysis of code functionality, manual audit found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the audit overview section. The general overview is presented in the Project Information section and all issues found are located in the audit overview section.

Team found 0 critical, 0 high, 0 medium, 2 low, 0 very low-level issues and 1 note in all solidity files of the contract

The files:

Xfuzion.sol

# File and Function Level Report

## File in Scope:

Contract Name	SHA 256 hash	Contract Address
Xfuzion.sol	753312dc898f591d13df1f8c84fbb39b6affeb40	0x24BC7AeDCC69813B5876934314D708D4254652ab

- Contract: Xfuzion
- Inherit: Context, IERC20, Ownable
- Observation: All passed including security check
- Test Report: passed
- Score: passed
- Conclusion: passed

Function	Test Result	Type / Return Type	Score
name	✓	Read / public	Passed
symbol	✓	Read / public	Passed
decimals	✓	Read / public	Passed
totalSupply	✓	Read / public	Passed
allowance	✓	Read / public	Passed
balanceOf	✓	Read / public	Passed
decimals	✓	Read / public	Passed
launched	✓	Read / public	Passed
ipulsexV2Pair	✓	Read / public	Passed
ipulsexV2Router	✓	Read / public	Passed
isExcludedFromReward	✓	Read / public	Passed
isExcludedFromFee	✓	Read / public	Passed
totalReflections	✓	Read / public	Passed
_campaignPeriod	✓	Read / public	Passed

_liquidityFee	✓	Read / public	<b>Passed</b>
_burnFee	✓	Read / public	<b>Passed</b>
_taxFee	✓	Read / public	<b>Passed</b>
getUnlockTime	✓	Read / public	<b>Passed</b>
liquidityDrive	✓	Read / public	<b>Passed</b>
ready	✓	Read / public	<b>Passed</b>
reflectionFromToken	✓	Read / public	<b>Passed</b>
router	✓	Read / public	<b>Passed</b>
swapAndLiquifyEnabled	✓	Read / public	<b>Passed</b>
startedOn	✓	Read / public	<b>Passed</b>
tokenFromReflection	✓	Read / public	<b>Passed</b>
xquantum	✓	Read / public	<b>Passed</b>
totalFees	✓	Read / public	<b>Passed</b>
approve	✓	Write / public	<b>Passed</b>
excludeFromReward	✓	Write / public	<b>Passed</b>
transferFrom	✓	Write / public	<b>Passed</b>
transfer	✓	Write / public	<b>Passed</b>
transferOwnership	✓	Write / public	<b>Passed</b>
includeInReward	✓	Write / public	<b>Passed</b>
excludeFromFee	✓	Write / public	<b>Passed</b>
includeInFee	✓	Write / public	<b>Passed</b>
decreaseAllowance	✓	Write / public	<b>Passed</b>
increaseAllowance	✓	Write / public	<b>Passed</b>
reflect	✓	Write / public	<b>Passed</b>
lock	✓	Write / public	<b>Passed</b>
unLock	✓	Write / public	<b>Passed</b>
lanuch	✓	Write / public	<b>Passed</b>

## File in Scope:

Contract Name	SHA 256 hash	Contract Address
XfuzionLiquidityDrive.sol	753312dc898f591d13df1f8c84fbb39b6affeb40	0xA8109AaDa855f906BdA4035931A57c83944337C2

- Contract: XfuzionLiquidityDrive
- Inherit: Context, Ownable
- Observation: All passed including security check
- Test Report: passed
- Score: passed
- Conclusion: passed

Function	Test Result	Type / Return Type	Score
availableOf	✓	Read / public	Passed
claimedOf	✓	Read / public	Passed
donationsOf	✓	Read / public	Passed
owner	✓	Read / public	Passed
endedOn	✓	Read / public	Passed
getUnlockTime	✓	Read / public	Passed
token	✓	Read / public	Passed
participants	✓	Read / public	Passed
totalClaimableTokens	✓	Read / public	Passed
totalClaimedTokens	✓	Read / public	Passed
totalEthDonated	✓	Read / public	Passed
totalTxS	✓	Read / public	Passed
transferOwnership	✓	Write / public	Passed
claimTokens	✓	Write / public	Passed
lock	✓	Write / public	Passed

donate	✓	Write / public	<b>Passed</b>
end	✓	Write / public	<b>Passed</b>
renounceOwnership	✓	Write / public	<b>Passed</b>
unLock	✓	Write / public	<b>Passed</b>



## File in Scope:

Contract Name	SHA 256 hash	Contract Address
XQuantum.sol	753312dc898f591d13df1f8c84fbb39b6affeb40	0x2836f00B2223E4249A13dcCCF6D6D92DC9c8a786

- Contract: XQuantum
- Inherit: Context, Ownable
- Observation: All passed including security check
- Test Report: passed
- Score: passed
- Conclusion: passed

Function	Test Result	Type / Return Type	Score
lastRebalance	✓	Read / public	<b>Passed</b>
upperboundPercentage	✓	Read / public	<b>Passed</b>
ready	✓	Read / public	<b>Passed</b>
owner	✓	Read / public	<b>Passed</b>
getUnlockTime	✓	Read / public	<b>Passed</b>
token	✓	Read / public	<b>Passed</b>
rebalance	✓	Write / public	<b>Passed</b>
lock	✓	Write / public	<b>Passed</b>
unlock	✓	Write / public	<b>Passed</b>
transferOwnership	✓	Write / public	<b>Passed</b>
renounceOwnership	✓	Write / public	<b>Passed</b>

# Issues Checking Status

No.	Issue Description	Checking Status
1	Compiler warnings.	Passed
2	Race conditions and Reentrancy. Cross-function race conditions.	Passed
3	Possible delays in data delivery.	Passed
4	Oracle calls.	Passed
5	Design Logic.	Passed
6	Timestamp dependence.	Passed
7	Integer Overflow and Underflow.	Passed
8	DoS with Revert.	Passed
9	DoS with block gas limit.	Passed with notes
10	Methods execution permissions.	Passed
11	Economy model. If application logic is based on an incorrect economic model, the application would not function correctly and participants would incur financial losses. This type of issue is most often found in bonus rewards systems, Staking and Farming contracts, Vault and Vesting contracts, etc.	Passed
12	The impact of the exchange rate on the logic.	Passed
13	Private user data leaks.	Passed
14	Malicious Event log.	Passed
15	Scoping and Declarations.	Passed
16	Uninitialized storage pointers.	Passed
17	Arithmetic accuracy.	Passed

## Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to tokens loss etc.
High	High-level vulnerabilities are difficult to exploit; however, they also have significant impact on smart contract execution, e.g. public access to crucial functions
Medium	Medium-level vulnerabilities are important to fix; however, they can't lead to tokens lose
Low	Low-level vulnerabilities are mostly related to outdated, unused etc. code snippets, that can't have significant impact on execution
Note	Lowest-level vulnerabilities, code style violations and info statements can't affect smart contract execution and can be ignored.

# Audit Findings

## Critical:

No Critical severity vulnerabilities were found.

## High:

No High severity vulnerabilities were found.

## Medium:

No Medium severity vulnerabilities were found.

## Low:

### Use of block.timestamp for comparisons

The value of block.timestamp can be manipulated by the miner. And conditions with strict equality is difficult to achieve - block.timestamp.

```
if (balance > upperbound) {
    uint256 airdrop = balance.sub(target);

    //send airdrop to token where it will be added to
liquidity    token.transfer(address(token), airdrop);

    lastRebalance = block.timestamp;

    emit Rebalance(airdrop);
}
```

### Recommendation

Avoid use of block.timestamp.

### Status

Acknowledged.

### #Pragam version not fixed

### Description

It is a good practice to lock the solidity version for a live deployment (use 0.8.20 instead of ^0.6.8). contracts should be deployed with the same compiler version and flags that they have been tested the most with. Locking the pragma helps ensure that contracts do not accidentally get deployed using, for example, the latest compiler which may have higher risks of undiscovered bugs. Contracts may also be deployed by others and the pragma indicates the compiler version intended by the original authors.

### Remediation

Remove the ^ sign to lock the pragma version.

Status: Acknowledged. It is normal, and the team can ignore this issue

## Very Low:

No Very Low severity vulnerabilities were found.

## Notes:

### Naming Conventions

#### Description

The contract follows a consistent naming convention where we are private variables with leading "\_" and public variables without it. But we have failed to comply to the condition for certain variable names "\_\_ burnFee " which is public.

```
uint256 public _taxFee = 5;
uint256 private _previousTaxFee = _taxFee;

uint256 public _burnFee = 2;

uint256 public _liquidityFee = 5;
uint256 private _previousLiquidityFee = _liquidityFee;
```

#### Recommendation

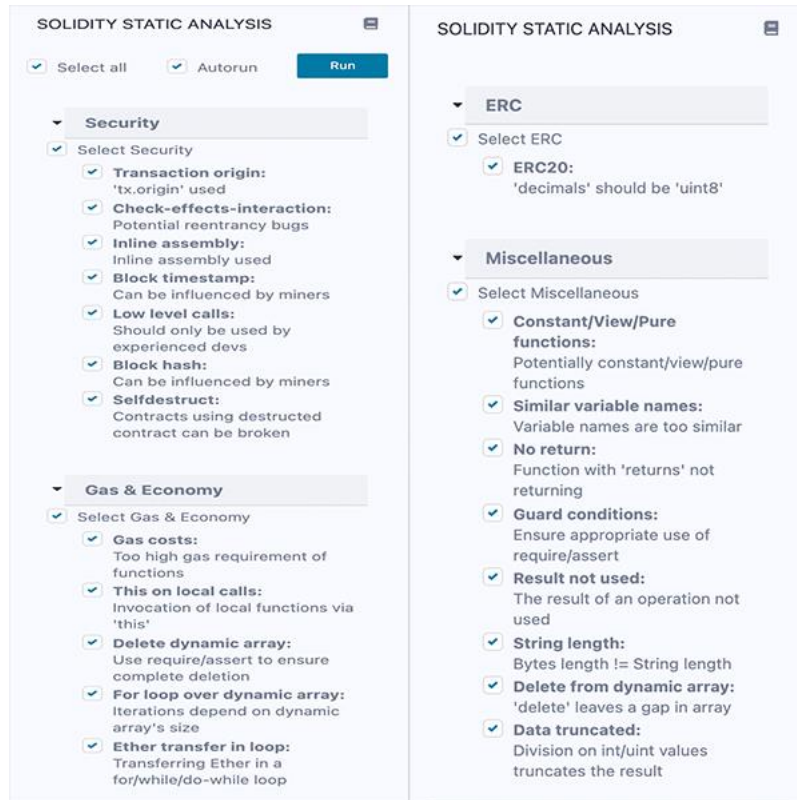
Remove "\_" from external variable names and add it to private variable names.

#### Status

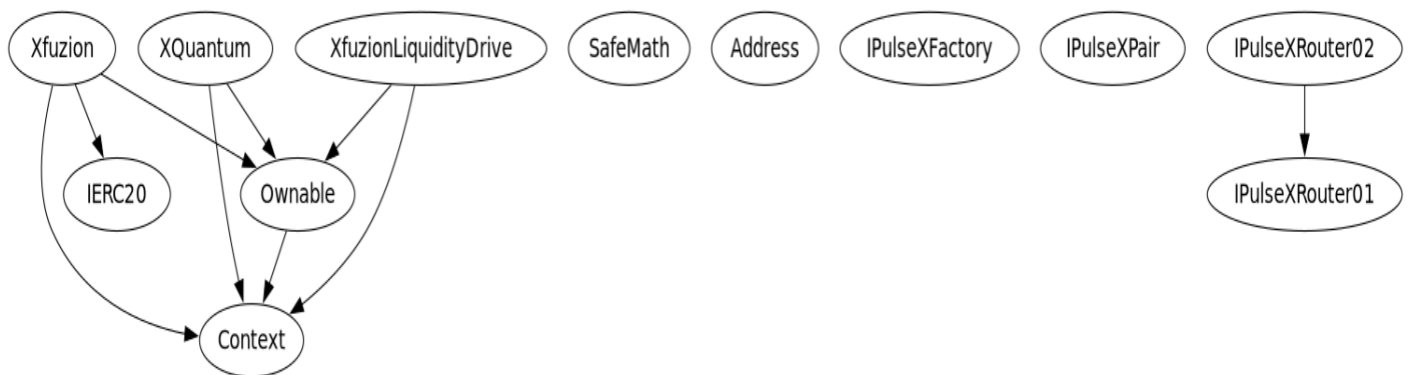
Acknowledged.

# Automatic Testing

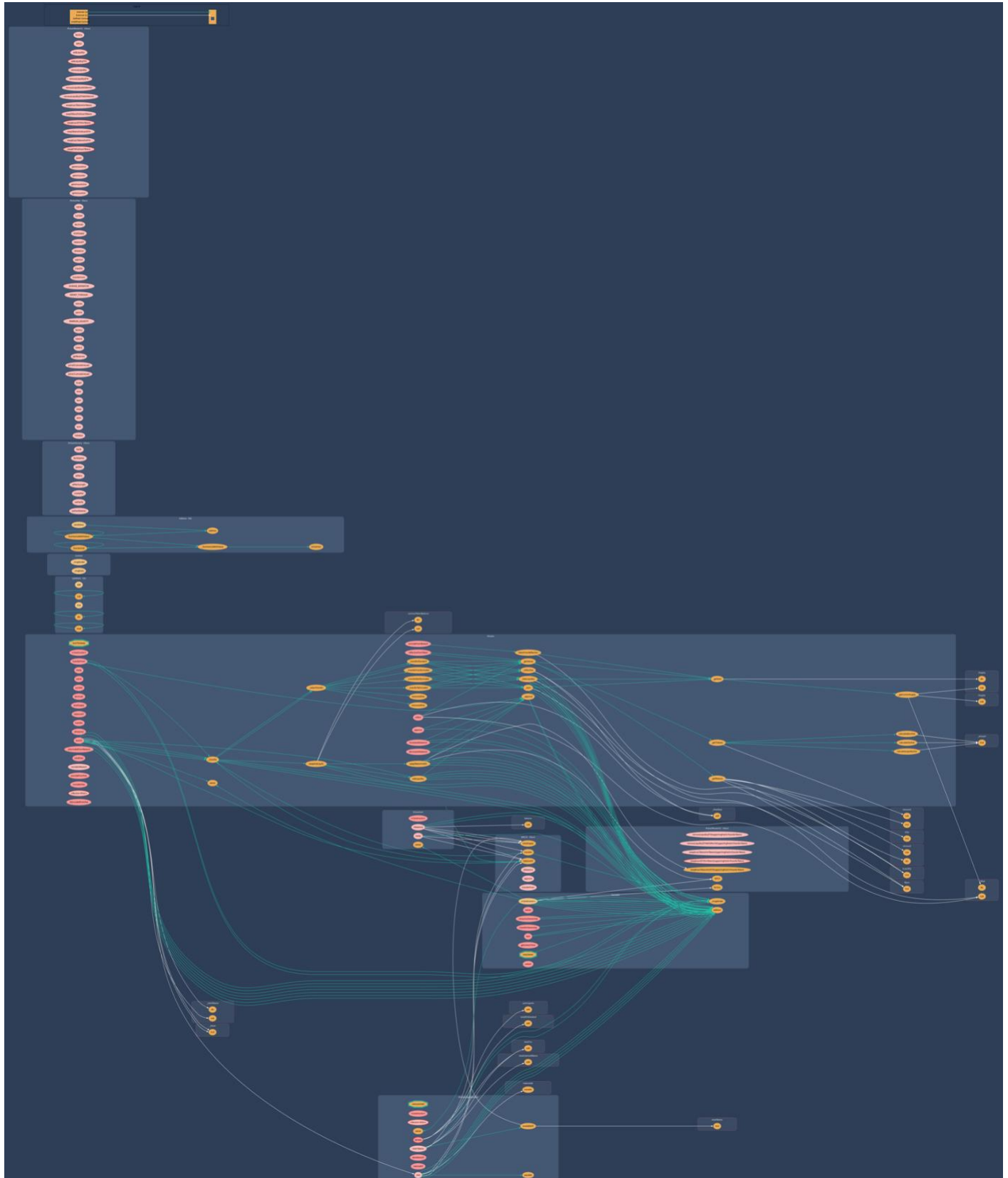
## 1- SOLIDITY STATIC ANALYSIS



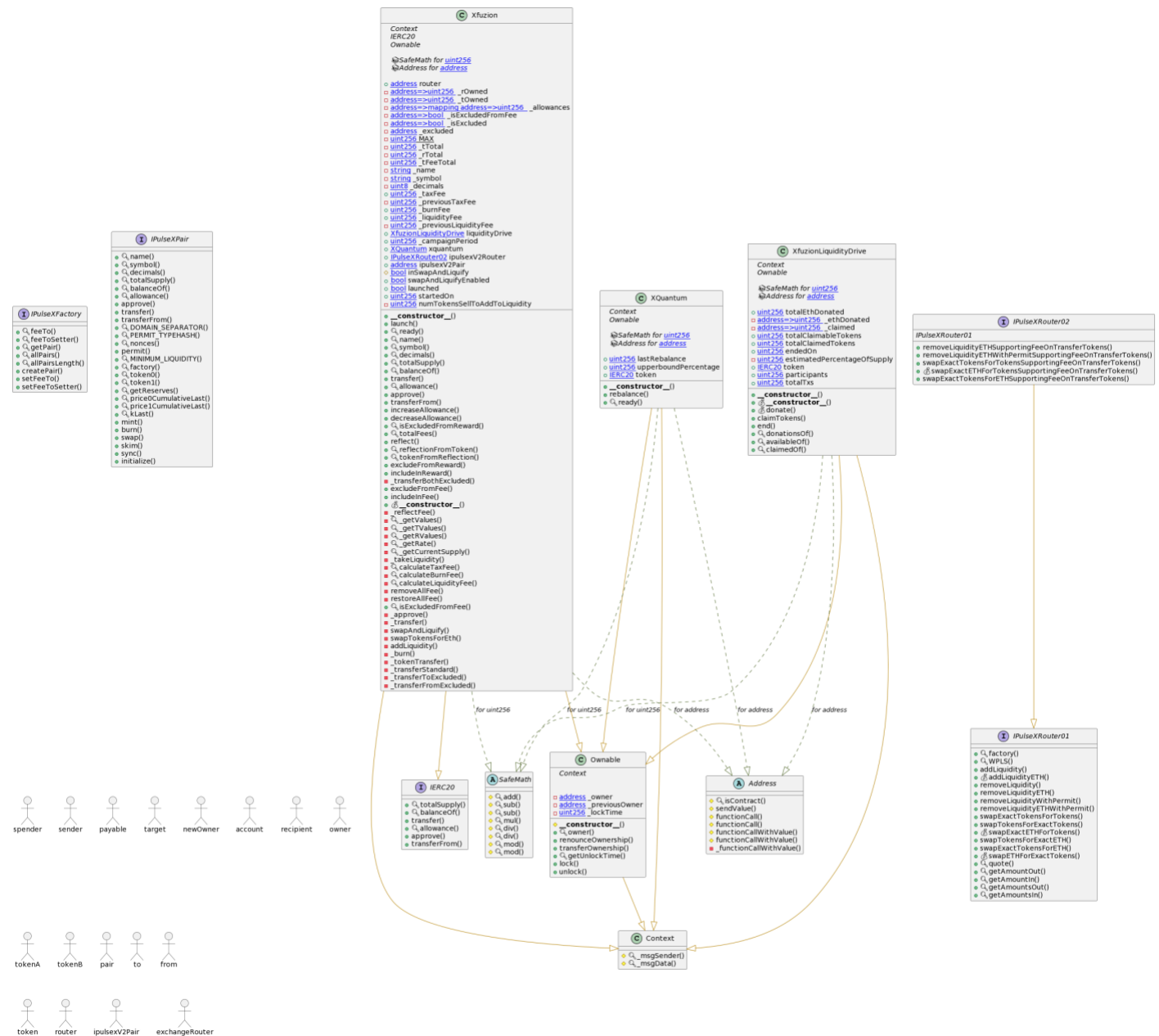
## 2- Inheritance graph



### 3- Call graph



# Unified Modeling Language (UML)





## Functions signature

Sighash		Function Signature
=====		
11902160	=>	<code>_getTValues(uint256)</code>
16279055	=>	<code>isContract(address)</code>
39509351	=>	<code>increaseAllowance(address,uint256)</code>
75128141	=>	<code>calculateTaxFee(uint256)</code>
18160ddd	=>	<code>totalSupply()</code>
70a08231	=>	<code>balanceOf(address)</code>
a9059cbb	=>	<code>transfer(address,uint256)</code>
dd62ed3e	=>	<code>allowance(address,address)</code>
095ea7b3	=>	<code>approve(address,uint256)</code>
23b872dd	=>	<code>transferFrom(address,address,uint256)</code>
771602f7	=>	<code>add(uint256,uint256)</code>
b67d77c5	=>	<code>sub(uint256,uint256)</code>
e31bdc0a	=>	<code>sub(uint256,uint256,string)</code>
c8a4ac9c	=>	<code>mul(uint256,uint256)</code>
a391c15b	=>	<code>div(uint256,uint256)</code>
b745d336	=>	<code>div(uint256,uint256,string)</code>
f43f523a	=>	<code>mod(uint256,uint256)</code>
71af23e8	=>	<code>mod(uint256,uint256,string)</code>
119df25f	=>	<code>_msgSender()</code>
8b49d47e	=>	<code>_msgData()</code>
24a084df	=>	<code>sendValue(address,uint256)</code>
a0b5ffb0	=>	<code>functionCall(address,bytes)</code>
241b5886	=>	<code>functionCall(address,bytes,string)</code>
2a011594	=>	<code>functionCallWithValue(address,bytes,uint256)</code>
d525ab8a	=>	<code>functionCallWithValue(address,bytes,uint256,string)</code>
36455e42	=>	<code>_functionCallWithValue(address,bytes,uint256,string)</code>
8da5cb5b	=>	<code>owner()</code>
715018a6	=>	<code>renounceOwnership()</code>
f2fde38b	=>	<code>transferOwnership(address)</code>
602bc62b	=>	<code>getUnlockTime()</code>
dd467064	=>	<code>lock(uint256)</code>
a69df4b5	=>	<code>unlock()</code>
017e7e58	=>	<code>feeTo()</code>
094b7415	=>	<code>feeToSetter()</code>
e6a43905	=>	<code>getPair(address,address)</code>
1e3dd18b	=>	<code>allPairs(uint256)</code>
574f2ba3	=>	<code>allPairsLength()</code>
c9c65396	=>	<code>createPair(address,address)</code>
f46901ed	=>	<code>setFeeTo(address)</code>
a2e74af6	=>	<code>setFeeToSetter(address)</code>
06fdde03	=>	<code>name()</code>
95d89b41	=>	<code>symbol()</code>
313ce567	=>	<code>decimals()</code>

```

3644e515 => DOMAIN_SEPARATOR()
30adf81f => PERMIT_TYPEHASH()
7ecebe00 => nonces(address)
d505accf => permit(address,address,uint256,uint256,uint8,bytes32,bytes32)
ba9a7a56 => MINIMUM_LIQUIDITY()
c45a0155 => factory()
0dfe1681 => token0()
d21220a7 => token1()
0902f1ac => getReserves()
5909c0d5 => price0CumulativeLast()
5a3d5493 => price1CumulativeLast()
7464fc3d => kLast()
6a627842 => mint(address)
89afcb44 => burn(address)
022c0d9f => swap(uint256,uint256,address,bytes)
bc25cf77 => skim(address)
fff6cae9 => sync()
485cc955 => initialize(address,address)
ef8ef56f => WPLS()
e8e33700 =>
addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256)
f305d719 => addLiquidityETH(address,uint256,uint256,uint256,address,uint256)
baa2abde =>
removeLiquidity(address,address,uint256,uint256,uint256,address,uint256)
02751cec => removeLiquidityETH(address,uint256,uint256,uint256,address,uint256)
2195995c =>
removeLiquidityWithPermit(address,address,uint256,uint256,uint256,address,uint256,
bool,uint8,bytes32,bytes32)
ded9382a =>
removeLiquidityETHWithPermit(address,uint256,uint256,uint256,address,uint256,bool,
uint8,bytes32,bytes32)
38ed1739 => swapExactTokensForTokens(uint256,uint256,address[],address,uint256)
8803dbee => swapTokensForExactTokens(uint256,uint256,address[],address,uint256)
7ff36ab5 => swapExactETHForTokens(uint256,address[],address,uint256)
4a25d94a => swapTokensForExactETH(uint256,uint256,address[],address,uint256)
18cbafe5 => swapExactTokensForETH(uint256,uint256,address[],address,uint256)
fb3bdb41 => swapETHForExactTokens(uint256,address[],address,uint256)
ad615dec => quote(uint256,uint256,uint256)
054d50d4 => getAmountOut(uint256,uint256,uint256)
85f8c259 => getAmountIn(uint256,uint256,uint256)
d06ca61f => getAmountsOut(uint256,address[])
1f00ca74 => getAmountsIn(uint256,address[])
af2979eb =>
removeLiquidityETHSupportingFeeOnTransferTokens(address,uint256,uint256,uint256,ad
dress,uint256)
5b0d5984 =>
removeLiquidityETHWithPermitSupportingFeeOnTransferTokens(address,uint256,uint256,
uint256,address,uint256,bool,uint8,bytes32,bytes32)

```

```

5c11d795 =>
swapExactTokensForTokensSupportingFeeOnTransferTokens(uint256,uint256,address[],ad
dress,uint256)
b6f9de95 =>
swapExactETHForTokensSupportingFeeOnTransferTokens(uint256,address[],address,uint2
56)
791ac947 =>
swapExactTokensForETHSupportingFeeOnTransferTokens(uint256,uint256,address[],addre
ss,uint256)
7d7c2a1c => rebalance()
6defbf80 => ready()
ed88c68e => donate()
48c54b9d => claimTokens()
efbe1c1c => end()
3fc30666 => donationsOf(address)
d546da90 => availableOf(address)
baa3f7ee => claimedOf(address)
01339c21 => launch()
a457c2d7 => decreaseAllowance(address,uint256)
88f82020 => isExcludedFromReward(address)
13114a9d => totalFees()
053ab182 => reflect(uint256)
4549b039 => reflectionFromToken(uint256,bool)
2d838119 => tokenFromReflection(uint256)
52390c02 => excludeFromReward(address)
3685d419 => includeInReward(address)
6ff6cdf4 => _transferBothExcluded(address,address,uint256)
437823ec => excludeFromFee(address)
ea2f0b37 => includeInFee(address)
184d894e => _reflectFee(uint256,uint256)
d4780e36 => _getValues(uint256)
65c63d72 => _getRValues(uint256,uint256,uint256,uint256,uint256)
94e10784 => _getRate()
97a9d560 => _getCurrentSupply()
c432df5e => _takeLiquidity(uint256)
6ad88269 => calculateBurnFee(uint256)
cc126a23 => calculateLiquidityFee(uint256)
301370af => removeAllFee()
e7e3e3a7 => restoreAllFee()
5342acb4 => isExcludedFromFee(address)
104e81ff => _approve(address,address,uint256)
30e0789e => _transfer(address,address,uint256)
173865ad => swapAndLiquify(uint256)
b28805f4 => swapTokensForEth(uint256)
9cd441da => addLiquidity(uint256,uint256)
6161eb18 => _burn(address,uint256)
b09bbc79 => _tokenTransfer(address,address,uint256,bool)
2852df65 => _transferStandard(address,address,uint256)
16f1cc83 => _transferToExcluded(address,address,uint256)

```

# Automatic general report

## Files Description Table

File Name	SHA-1 Hash
/Users/macbook/Desktop/smart contracts/Xfuzion.sol	753312dc898f591d13df1f8c84fbb39b6affeb40

## Contracts Description Table

Contract	Type	Bases		
:-----: :-----: :-----: :-----: :-----:				
-----:				
L	**Function Name**	**Visibility**	**Mutability**	
**Modifiers**				
**IERC20**   Interface				
L	totalSupply	External !	NO!	
L	balanceOf	External !	NO!	
L	transfer	External !	⊗   NO!	
L	allowance	External !	NO!	
L	approve	External !	⊗   NO!	
L	transferFrom	External !	⊗   NO!	
**SafeMath**   Library				
L	add	Internal 🔒		
L	sub	Internal 🔒		
L	sub	Internal 🔒		
L	mul	Internal 🔒		
L	div	Internal 🔒		
L	div	Internal 🔒		
L	mod	Internal 🔒		
L	mod	Internal 🔒		
**Context**   Implementation				
L	_msgSender	Internal 🔒		
L	_msgData	Internal 🔒		
**Address**   Library				
L	isContract	Internal 🔒		
L	sendValue	Internal 🔒	⊗	
L	functionCall	Internal 🔒	⊗	
L	functionCall	Internal 🔒	⊗	
L	functionCallWithValue	Internal 🔒	⊗	
L	functionCallWithValue	Internal 🔒	⊗	

```

| L | _functionCallWithValue | Private | 🔒 | 🔒 | |
| | | | |
| **Ownable** | Implementation | Context | | |
| L | <Constructor> | Internal | 🔒 | 🔒 | |
| L | owner | Public | ! | |NO! |
| L | renounceOwnership | Public | ! | 🔒 | onlyOwner |
| L | transferOwnership | Public | ! | 🔒 | onlyOwner |
| L | getUnlockTime | Public | ! | |NO! |
| L | lock | Public | ! | 🔒 | onlyOwner |
| L | unlock | Public | ! | 🔒 |NO! |
| | | | |
| **IPulseXFactory** | Interface | | | |
| L | feeTo | External | ! | |NO! |
| L | feeToSetter | External | ! | |NO! |
| L | getPair | External | ! | |NO! |
| L | allPairs | External | ! | |NO! |
| L | allPairsLength | External | ! | |NO! |
| L | createPair | External | ! | 🔒 |NO! |
| L | setFeeTo | External | ! | 🔒 |NO! |
| L | setFeeToSetter | External | ! | 🔒 |NO! |
| | | | |
| **IPulseXPair** | Interface | | | |
| L | name | External | ! | |NO! |
| L | symbol | External | ! | |NO! |
| L | decimals | External | ! | |NO! |
| L | totalSupply | External | ! | |NO! |
| L | balanceOf | External | ! | |NO! |
| L | allowance | External | ! | |NO! |
| L | approve | External | ! | 🔒 |NO! |
| L | transfer | External | ! | 🔒 |NO! |
| L | transferFrom | External | ! | 🔒 |NO! |
| L | DOMAIN_SEPARATOR | External | ! | |NO! |
| L | PERMIT_TYPEHASH | External | ! | |NO! |
| L | nonces | External | ! | |NO! |
| L | permit | External | ! | 🔒 |NO! |
| L | MINIMUM_LIQUIDITY | External | ! | |NO! |
| L | factory | External | ! | |NO! |
| L | token0 | External | ! | |NO! |
| L | token1 | External | ! | |NO! |
| L | getReserves | External | ! | |NO! |
| L | price0CumulativeLast | External | ! | |NO! |
| L | price1CumulativeLast | External | ! | |NO! |
| L | kLast | External | ! | |NO! |
| L | mint | External | ! | 🔒 |NO! |
| L | burn | External | ! | 🔒 |NO! |
| L | swap | External | ! | 🔒 |NO! |
| L | skim | External | ! | 🔒 |NO! |




```

```

| L | sync | External ! | ⬤ |NO! |
| L | initialize | External ! | ⬤ |NO! |
|||||
| **IPulseXRouter01** | Interface | |||
| L | factory | External ! | |NO! |
| L | WPLS | External ! | |NO! |
| L | addLiquidity | External ! | ⬤ |NO! |
| L | addLiquidityETH | External ! | ⬤ |NO! |
| L | removeLiquidity | External ! | ⬤ |NO! |
| L | removeLiquidityETH | External ! | ⬤ |NO! |
| L | removeLiquidityWithPermit | External ! | ⬤ |NO! |
| L | removeLiquidityETHWithPermit | External ! | ⬤ |NO! |
| L | swapExactTokensForTokens | External ! | ⬤ |NO! |
| L | swapTokensForExactTokens | External ! | ⬤ |NO! |
| L | swapExactETHForTokens | External ! | ⬤ |NO! |
| L | swapTokensForExactETH | External ! | ⬤ |NO! |
| L | swapExactTokensForETH | External ! | ⬤ |NO! |
| L | swapETHForExactTokens | External ! | ⬤ |NO! |
| L | quote | External ! | |NO! |
| L | getAmountOut | External ! | |NO! |
| L | getAmountIn | External ! | |NO! |
| L | getAmountsOut | External ! | |NO! |
| L | getAmountsIn | External ! | |NO! |
|||||
| **IPulseXRouter02** | Interface | IPulseXRouter01 |||
| L | removeLiquidityETHSupportingFeeOnTransferTokens | External ! | ⬤ |NO! |
| L | removeLiquidityETHWithPermitSupportingFeeOnTransferTokens | External ! | ⬤ |NO! |
| L | swapExactTokensForTokensSupportingFeeOnTransferTokens | External ! | ⬤ |NO! |
| L | swapExactETHForTokensSupportingFeeOnTransferTokens | External ! | ⬤ |NO! |
| L | swapExactTokensForETHSupportingFeeOnTransferTokens | External ! | ⬤ |NO! |
|||||
| **XQuantum** | Implementation | Context, Ownable |||
| L | <Constructor> | Public ! | ⬤ |NO! |
| L | rebalance | External ! | ⬤ |NO! |
| L | ready | External ! | |NO! |
|||||
| **XfuzionLiquidityDrive** | Implementation | Context, Ownable |||
| L | <Constructor> | Public ! | ⬤ |NO! |
| L | <Receive Ether> | External ! | ⬤ |NO! |
| L | donate | Public ! | ⬤ |notLaunched |
| L | claimTokens | External ! | ⬤ |NO! |
| L | end | External ! | ⬤ |onlyOwner notLaunched |
| L | donationsOf | Public ! | |NO! |
| L | availableOf | Public ! | |NO! |
| L | claimedOf | Public ! | |NO! |
|||||

```



```

| **Xfuzion** | Implementation | Context, IERC20, Ownable |||
| L | <Constructor> | Public ! |  |NO! |
| L | launch | Public ! |  |NO! |
| L | ready | Public ! | |NO! |
| L | name | Public ! | |NO! |
| L | symbol | Public ! | |NO! |
| L | decimals | Public ! | |NO! |
| L | totalSupply | Public ! | |NO! |
| L | balanceOf | Public ! | |NO! |
| L | transfer | Public ! |  |NO! |
| L | allowance | Public ! | |NO! |
| L | approve | Public ! |  |NO! |
| L | transferFrom | Public ! |  |NO! |
| L | increaseAllowance | Public ! |  |NO! |
| L | decreaseAllowance | Public ! |  |NO! |
| L | isExcludedFromReward | Public ! | |NO! |
| L | totalFees | Public ! | |NO! |
| L | reflect | Public ! |  |NO! |
| L | reflectionFromToken | Public ! | |NO! |
| L | tokenFromReflection | Public ! | |NO! |
| L | excludeFromReward | Public ! |  | onlyOwner |
| L | includeInReward | External ! |  | onlyOwner |
| L | _transferBothExcluded | Private  |  | |
| L | excludeFromFee | Public ! |  | onlyOwner |
| L | includeInFee | Public ! |  | onlyOwner |
| L | <Receive Ether> | External ! |  |NO! |
| L | _reflectFee | Private  |  | |
| L | _getValues | Private  | | |
| L | _getTValues | Private  | | |
| L | _getRValues | Private  | | |
| L | _getRate | Private  | | |
| L | _getCurrentSupply | Private  | | |
| L | _takeLiquidity | Private  |  | |
| L | calculateTaxFee | Private  | | |
| L | calculateBurnFee | Private  | | |
| L | calculateLiquidityFee | Private  | | |
| L | removeAllFee | Private  |  | |
| L | restoreAllFee | Private  |  | |
| L | isExcludedFromFee | Public ! | |NO! |
| L | _approve | Private  |  | |
| L | _transfer | Private  |  | |
| L | swapAndLiquify | Private  |  | lockTheSwap |
| L | swapTokensForEth | Private  |  | |
| L | addLiquidity | Private  |  | |
| L | _burn | Private  |  | |
| L | _tokenTransfer | Private  |  | |
| L | _transferStandard | Private  |  | |

```

```
| L | _transferToExcluded | Private  |  | |
| L | _transferFromExcluded | Private  |  | |
```

Legend

```
| Symbol | Meaning |
|:-----:|-----|
|  | Function can modify state |
|  | Function is payable |
```



# Conclusion

The contracts are written systematically. Team found no critical issues. So, it is good to go for production.

Since possible test cases can be unlimited and developer level documentation (code flow diagram with function level description) not provided, for such an extensive smart contract protocol, we provide no such guarantee of future outcomes. We have used all the latest static tools and manual observations to cover maximum possible test cases to scan Everything.

Security state of the reviewed contract is “Well Secured”.

- ✓ No volatile code.
- ✓ No high severity issues were found.

# Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as of the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against the team on the basis of what it says or doesn't say, or how team produced it, and it is important for you to conduct your own independent investigations before making any decisions. team go into more detail on this in the below disclaimer below – please make sure to read it in full.

By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and Saferico and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (Saferico s) owe no duty of care towards you or any other person, nor does Saferico make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and Saferico hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, Saferico hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against Saferico, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report. The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.