

Smart Contract Security Audit V1

xfuzion NFT Smart Contract Audit

<https://xfuzion.finance/>

Nov 28, 2023



<https://saferico.com/>

business@saferico.com

https://t.me/SFI_ANN

—

Table of Contents

Table of Contents

Background

Project Information

NFT Smart Contract Information

Executive Summary

File and Function Level Report

File in Scope:

Issues Checking Status

SWC Attack Analysis

Severity Definitions

Audit Findings

Automatic testing

Testing proves

Inheritance graph

Call graph

Source lines

Risk level

Source units in scope

Capabilities

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 Information

- **Platform:** Pulse Chain
- **Name:** xfuzion NFT
- **Language :** solidity
- **Contract Address:** 0x703e35AF0944ed7fB7da6aCdC595Da759968993F
- **Code Source:**
<https://scan.pulsechain.com/address/0x703e35AF0944ed7fB7da6aCdC595Da759968993F/contracts#address-tabs>
- **Website:** <https://xfuzion.finance/>
- **Telegram:** <https://t.me/XfuzionFinance>
- **X:** <https://x.com/xfuzionfinance?s=21&t=UppJadbfgCRdnVpOnFAakQ>
- **Youtube:** <https://youtube.com/@XfuzionFinance?si=nfNwrOVxVWNJgnq2>
- **Discord:** <https://discord.gg/FCcD8YTuxc>

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, 4 low, 0 very low-level issues and 1 note in all solidity files of the contract

The files:

xfuzionNFT.sol

Audit Score:

99% secure



File and Function Level Report

File in Scope:

Contract Name	SHA 256 hash	Contract Address
xfuzionNFT.sol	c4c7055fe362f878cfc4be9bf3 c7f4d58a8eb14b	0x703e35AF0944ed7fB7da6aCdC595Da75996 8993F

- Contract: xfuzionNFT
- Inherit: ERC721Enumerable, Ownable
- Observation: All passed including security check
- Test Report: **passed**
- Score: **passed**
- Conclusion: **passed**

Function	Test Result	Type / Return Type	Score
aprByTier	✓	Read / public	Passed
_WL	✓	Read / public	Passed
balanceOf	✓	Read / public	Passed
owner	✓	Read / public	Passed
BaseTokenURI	✓	Read / public	Passed
buyBackFeeAmount	✓	Read / public	Passed
buyBackFeeWallet	✓	Read / public	Passed
DAI	✓	Read / public	Passed
devFeeAmount	✓	Read / public	Passed
devFeeWallet	✓	Read / public	Passed
getActiveNFTBalance	✓	Read / public	Passed
isApprovedForAll	✓	Read / public	Passed
getApproved	✓	Read / public	Passed

LiquidityClaimFeeAmount	✓	Read / public	Passed
LiquidityFee	✓	Read / public	Passed
liquidityWallet	✓	Read / public	Passed
MaxNum	✓	Read / public	Passed
minted	✓	Read / public	Passed
name	✓	Read / public	Passed
ownerOf	✓	Read / public	Passed
ReferralFee	✓	Read / public	Passed
referrals	✓	Read / public	Passed
Price	✓	Read / public	Passed
NumPer Tx	✓	Read / public	Passed
saleOpen	✓	Read / public	Passed
supportsInterface	✓	Read / public	Passed
tokenInfos	✓	Read / public	Passed
tokenByIndex	✓	Read / public	Passed
symbol	✓	Read / public	Passed
tokenOfOwnerByIndex	✓	Read / public	Passed
totalSupply	✓	Read / public	Passed
tokenURI	✓	Read / public	Passed
totalSoldoutByTier	✓	Read / public	Passed
totalSold	✓	Read / public	Passed
walletOwners	✓	Read / public	Passed
walletOfOwner	✓	Read / public	Passed
mint	✓	Write / public	Passed
claim	✓	Write / public	Passed
approve	✓	Write / public	Passed
safeTransferFrom	✓	Write / public	Passed
safeTransferFrom	✓	Write / public	Passed
transferFrom	✓	Write / public	Passed

transferOwnership	✓	Write / public	Passed
renounceOwnership	✓	Write / public	Passed
setBaseURI	✓	Write / public	Passed
setPrice	✓	Write / public	Passed
setWhitelist	✓	Write / public	Passed
setApprovalForAll	✓	Write / public	Passed
setMaxNum	✓	Write / public	Passed
toggleSale	✓	Write / public	Passed

Issues Checking Status

SWC Attack Analysis

The Smart Contract Weakness Classification Registry (SWC Registry) is an implementation of the weakness classification scheme proposed in EIP-1470. It is loosely aligned to the terminologies and structure used in the Common Weakness Enumeration (CWE) for more info check

<https://swcregistry.io/>

No.	Issue Description	Checking Status
136	Unencrypted Private Data On-Chain	Passed
135	Code With No Effects	Passed
134	Message call with hardcoded gas amount	Passed
133	Hash Collisions With Multiple Variable Length Arguments	Passed
132	Unexpected Ether balance	Passed
131	Presence of unused variables	Passed
130	Right-To-Left-Override control character (U+202E)	Passed
129	Typographical Error	Passed
128	DoS with block gas limit.	Passed
127	Arbitrary Jump with Function Type Variable	Passed
126	Insufficient Gas Griefing	Passed
125	Incorrect Inheritance Order	Passed
124	Write to Arbitrary Storage Location	Passed
123	Requirement Violation	Passed
122	Lack of Proper Signature Verification	Passed
121	Missing Protection against Signature Replay Attacks	Passed
120	Weak Sources of Randomness from Chain Attributes	Passed
119	Shadowing State Variables	Passed

118	Incorrect Constructor Name	Passed
117	Signature Malleability	Passed
116	Block values as a proxy for time	Passed
115	Authorization through tx.origin	Passed
114	Transaction Order Dependence	Passed
113	DoS with Failed Call	Passed
112	Delegatecall to Untrusted Callee	Passed
111	Use of Deprecated Solidity Functions	Passed
110	Assert Violation	Passed
109	Uninitialized Storage Pointer	Passed
108	State Variable Default Visibility	Passed
107	Reentrancy	Passed
106	Unprotected SELFDESTRUCT Instruction	Passed
105	Unprotected Ether Withdrawal	Passed
104	Unchecked Call Return Value	Passed
103	Floating Pragma	Not Passed
102	Outdated Compiler Version	Passed
101	Integer Overflow and Underflow	Passed
100	Function Default Visibility	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:

#Owner privileges (In the period when the owner isn't renounced)

Description

The owner can add / remove any address to whitelist.

The owner can change the price in the contract.

The owner can enable / disable the sale.

```
function toggleSale() external onlyOwner {
    saleOpen = !saleOpen;
}

function setWhiteList(address[] calldata _accounts, bool _value) public
onlyOwner {
    for(uint256 i = 0; i < _accounts.length; i++) {
        _WL[_accounts[i]] = _value;
    }
}

function setPrice(uint256[] memory _prices) external onlyOwner {
    for (uint256 i = 0; i < _prices.length; i++) {
        price[i] = _prices[i];
    }
}
```

Remediation

Make these functions internal in next version or the team should announce the investors before doing anything to give them time if they want to do anything.

P.S: This issue is common to the majority of those smart contracts.

Status: **Acknowledged.**

#Multiple pragma statements

Line	Pragma
5	pragma solidity ^0.8.0;
48	pragma solidity ^0.8.0;
72	pragma solidity ^0.8.0;
97	pragma solidity ^0.8.0;
256	pragma solidity ^0.8.0;
332	pragma solidity ^0.8.0;
367	pragma solidity ^0.8.0;
395	pragma solidity ^0.8.0;
466	pragma solidity ^0.8.0;
714	pragma solidity ^0.8.0;
739	pragma solidity ^0.8.0;
766	pragma solidity ^0.8.0;
1256	pragma solidity ^0.8.0;
1468	pragma solidity ^0.8.0;

Description

There are multiple pragma statements in the code. The newest compiler version 0.8.23 will work with the code, but keeping only one pragma statement helps in maintaining readability of the code.

Remediation

Keep a single pragma statement.

Status: [Acknowledged](#).

#Missing zero address validation

When the dev wants add the fees addresses when he deploys the smart contract, he has to check for the zero address to make, he didn't add the zero address. Otherwise, he will lose the fee, and the same for the owner when he adds addresses to the whitelist.

```
constructor(IERC20 _dai, address _liquidityWallet, address _devFeeWallet, address
_buybackFeeWallet) ERC721("XFuzion NFT", "XNFT") {
    DAI = _dai;
    liquidityWallet = _liquidityWallet;
    devFeeWallet = _devFeeWallet;
    buybackFeeWallet = _buybackFeeWallet;
}
function setWhiteList(address[] calldata _accounts, bool _value) public onlyOwner {
    for(uint256 i = 0; i < _accounts.length; i++) {
        _WL[_accounts[i]] = _value; } }
```

Remediation

Use the require statement to check for zero addresses.

Status: [Acknowledged](#).

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.

```
function pending(uint8 _tierType, address user) public view returns(uint256) {
    uint256 balanceOfTier = balanceOf(user);
    uint256 totalClaimableByTier = 0;
    for (uint256 i = 0; i < balanceOfTier; i++) {
        uint256 tokenId = tokenOfOwnerByIndex(user, i);
        if(tokenInfos[tokenId].tierType == _tierType &&
!tokenInfos[tokenId].isExpired) {
            uint256 diffTimeStamp = block.timestamp -
tokenInfos[tokenId].lastClaimedTime;
            uint256 claimableAmount = diffTimeStamp * price[_tierType] *
aprByTier[_tierType] / 10000 / 86400;
            if(claimableAmount + tokenInfos[tokenId].claimedAmount >
price[_tierType] * 2 ) {
                claimableAmount = price[_tierType] * 2 -
tokenInfos[tokenId].claimedAmount;
            }
            totalClaimableByTier += claimableAmount;
        }
    }
    return totalClaimableByTier;
}
```

Recommendation

Avoid use of block.timestamp.

Status

Acknowledged.

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 missed to comply to the condition for certain variable names "__WL" which is public.

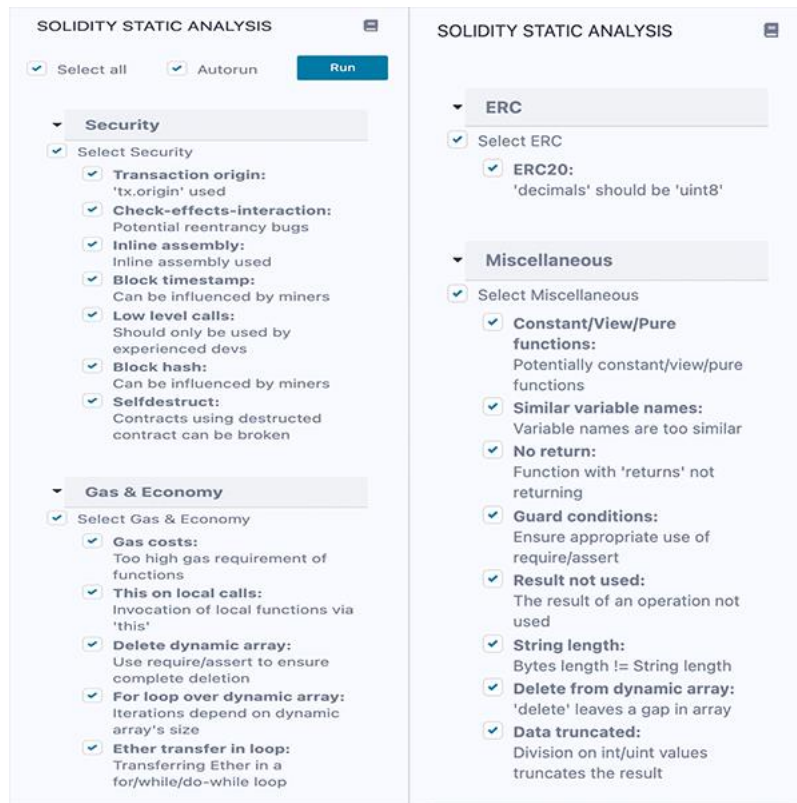
Remediation

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

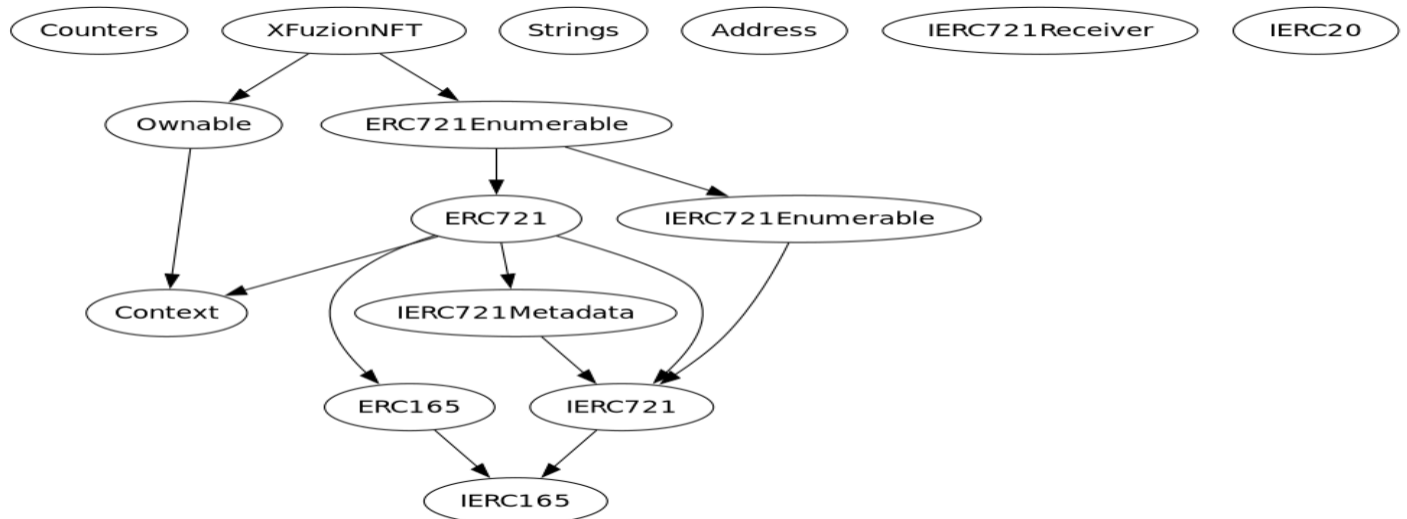
Status: Acknowledged.

Automatic Testing

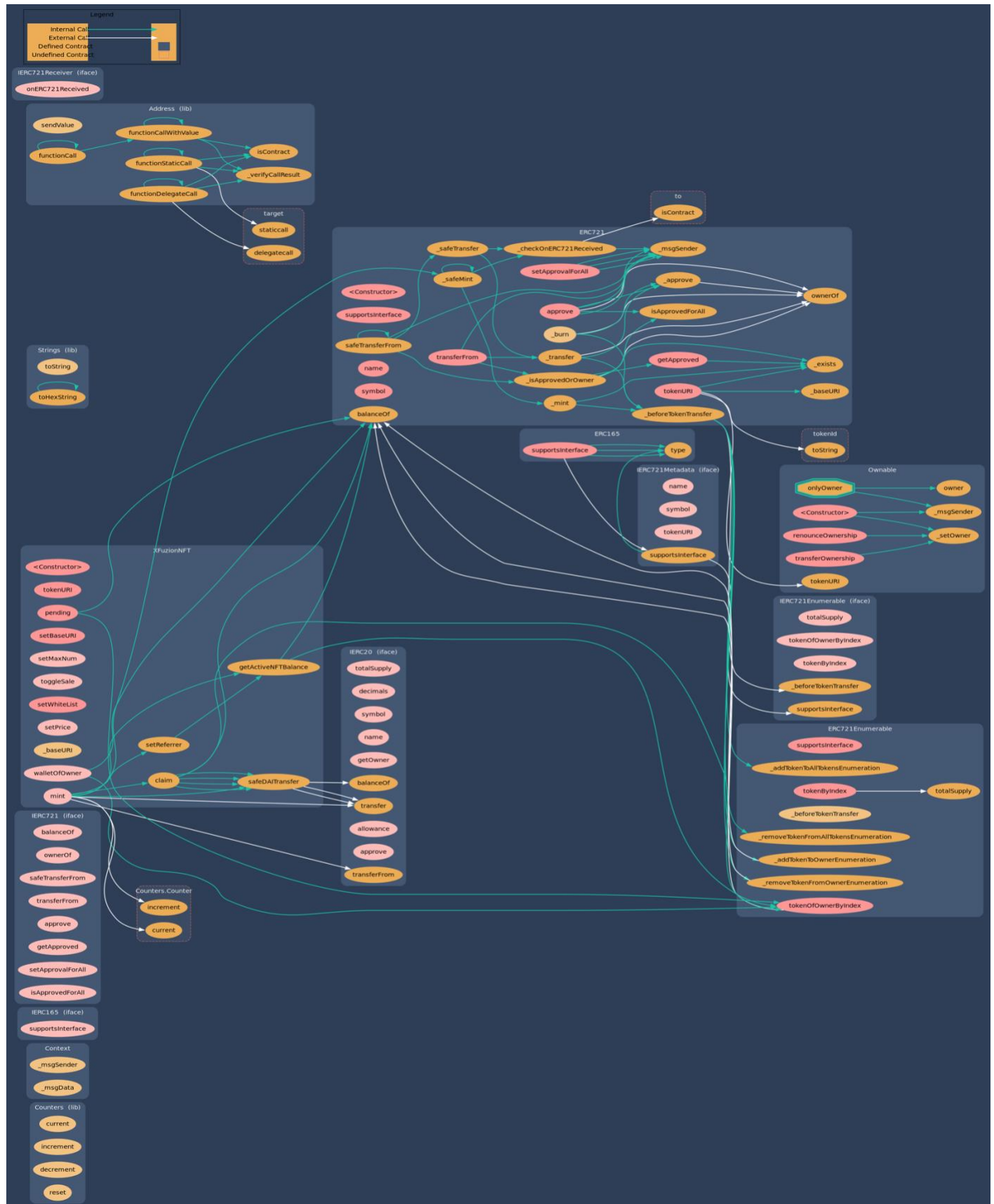
1- SOLIDITY STATIC ANALYSIS



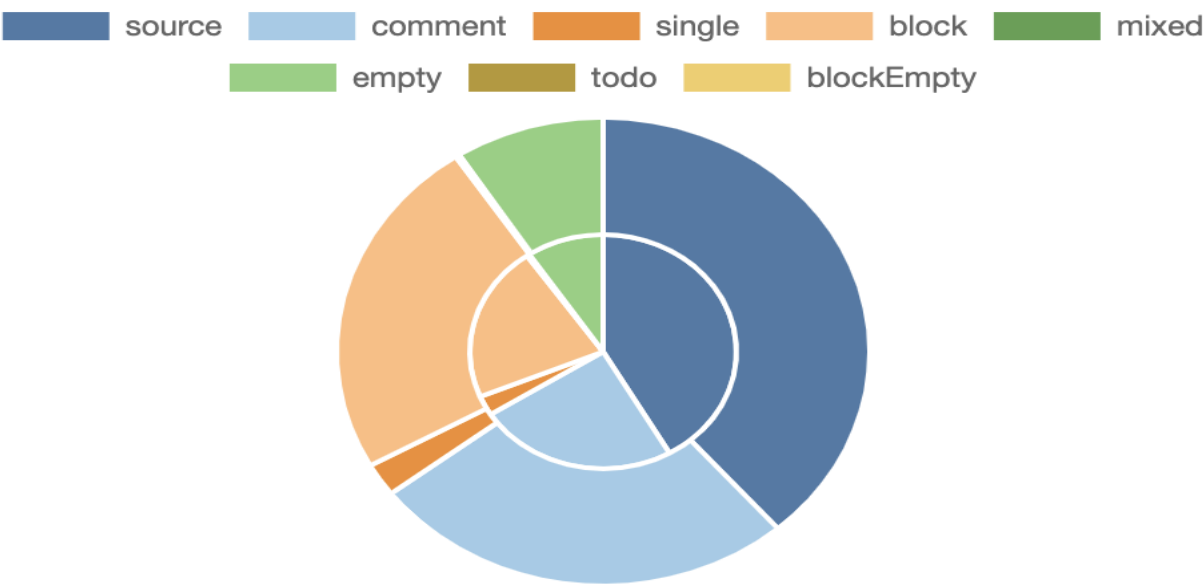
2- Inheritance graph



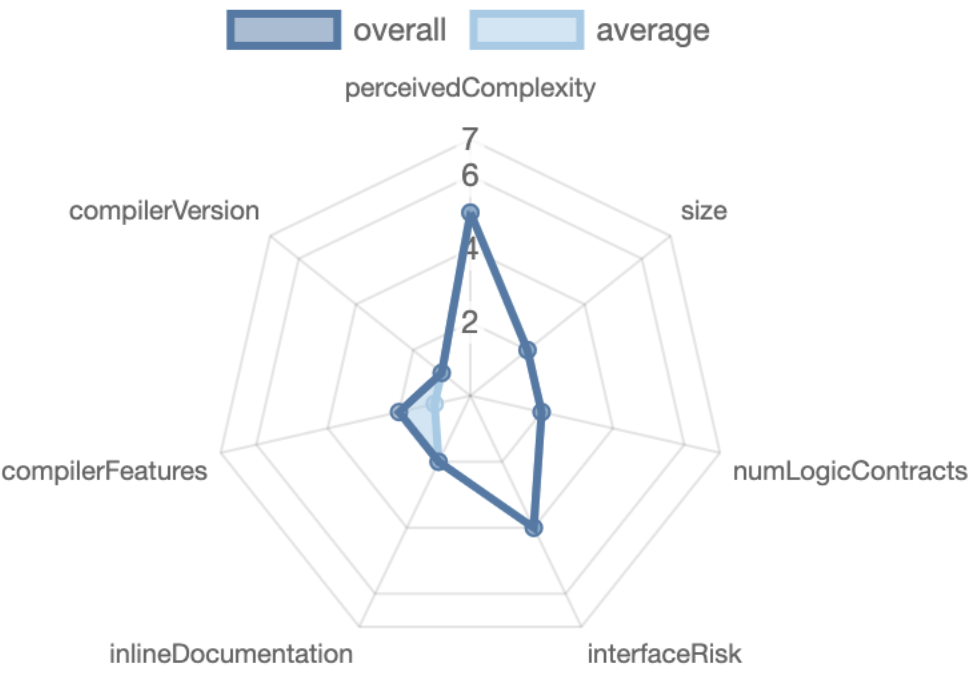
3- Call graph



Source lines



Risk level



Source units in scope

Source Units in Scope

Source Units Analyzed: 1
Source Units in Scope: 1 (100%)

Type	File	Logic Contracts	Interfaces	Lines	nLines	nSLOC	Comment Lines	Complex. Score	Capabilities
	Xfuzion NFT.sol	9	6	1708	1324	736	605	563	
	Totals	9	6	1708	1324	736	605	563	

Legend: [-]

- **Lines**: total lines of the source unit
- **nLines**: normalized lines of the source unit (e.g. normalizes functions spanning multiple lines)
- **nSLOC**: normalized source lines of code (only source-code lines; no comments, no blank lines)
- **Comment Lines**: lines containing single or block comments
- **Complexity Score**: a custom complexity score derived from code statements that are known to introduce code complexity (branches, loops, calls, external interfaces, ...)

Capabilities

Components

2	3	6	4

Exposed Functions

This section lists functions that are explicitly declared public or payable. Please note that getter methods for public stateVars are not included.

59	0

External	Internal	Private	Pure	View
32	87	7	4	46

StateVariables

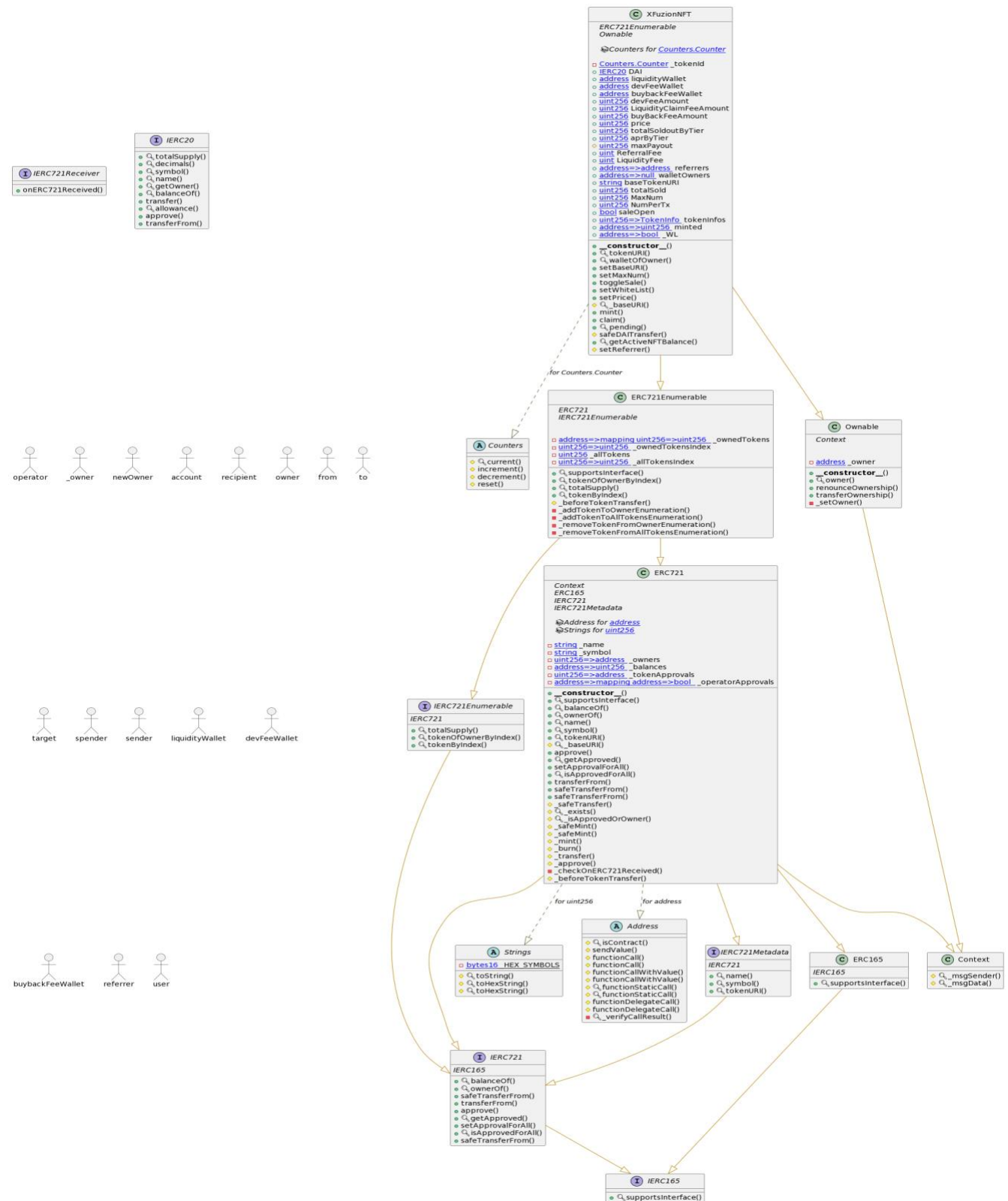
Total	
36	22

Capabilities

Solidity Versions observed	Experimental Features	Can Receive Funds	Uses Assembly	Has Destroyable Contracts
^0.8.0			yes (3 asm blocks)	

Transfers ETH	Low-Level Calls	DelegateCall	Uses Hash Functions	ECRecover	New/Create/Create2
yes		yes			

Unified Modeling Language (UML)



Functions signature

Function Name	Sighash	Function Signature
supportsInterface	01ffc9a7	supportsInterface(bytes4)
balanceOf	70a08231	balanceOf(address)
ownerOf	6352211e	ownerOf(uint256)
safeTransferFrom	42842e0e	safeTransferFrom(address,address,uint256)
transferFrom	23b872dd	transferFrom(address,address,uint256)
approve	095ea7b3	approve(address,uint256)
getApproved	081812fc	getApproved(uint256)
setApprovalForAll	a22cb465	setApprovalForAll(address,bool)
isApprovedForAll	e985e9c5	isApprovedForAll(address,address)
safeTransferFrom	b88d4fde	safeTransferFrom(address,address,uint256,bytes)
owner	8da5cb5b	owner()
renounceOwnership	715018a6	renounceOwnership()
transferOwnership	f2fde38b	transferOwnership(address)
totalSupply	18160ddd	totalSupply()
tokenOfOwnerByIndex	2f745c59	tokenOfOwnerByIndex(address,uint256)
tokenByIndex	4f6ccce7	tokenByIndex(uint256)
supportsInterface	01ffc9a7	supportsInterface(bytes4)
name	06fdde03	name()
symbol	95d89b41	symbol()
tokenURI	c87b56dd	tokenURI(uint256)
onERC721Received	150b7a02	onERC721Received(address,address,uint256,bytes)
supportsInterface	01ffc9a7	supportsInterface(bytes4)
balanceOf	70a08231	balanceOf(address)
ownerOf	6352211e	ownerOf(uint256)
name	06fdde03	name()
symbol	95d89b41	symbol()
tokenURI	c87b56dd	tokenURI(uint256)
approve	095ea7b3	approve(address,uint256)
getApproved	081812fc	getApproved(uint256)
setApprovalForAll	a22cb465	setApprovalForAll(address,bool)
isApprovedForAll	e985e9c5	isApprovedForAll(address,address)
transferFrom	23b872dd	transferFrom(address,address,uint256)
safeTransferFrom	42842e0e	safeTransferFrom(address,address,uint256)
safeTransferFrom	b88d4fde	safeTransferFrom(address,address,uint256,bytes)
supportsInterface	01ffc9a7	supportsInterface(bytes4)
tokenOfOwnerByIndex	2f745c59	tokenOfOwnerByIndex(address,uint256)
totalSupply	18160ddd	totalSupply()
tokenByIndex	4f6ccce7	tokenByIndex(uint256)
totalSupply	18160ddd	totalSupply()
decimals	313ce567	decimals()
symbol	95d89b41	symbol()
name	06fdde03	name()
getOwner	893d20e8	getOwner()
balanceOf	70a08231	balanceOf(address)
transfer	a9059cbb	transfer(address,uint256)
allowance	dd62ed3e	allowance(address,address)
approve	095ea7b3	approve(address,uint256)
transferFrom	23b872dd	transferFrom(address,address,uint256)
tokenURI	c87b56dd	tokenURI(uint256)
walletOfOwner	438b6300	walletOfOwner(address)
setBaseURI	55f804b3	setBaseURI(string)
setMaxNum	02e5329e	setMaxNum(uint256)
toggleSale	7d8966e4	toggleSale()
setWhiteList	e43f696e	setWhiteList(address[],bool)

```
| setPrice | e39dd79c | setPrice(uint256[]) |  
| mint | 2715eb15 | mint(uint8,uint256,address) |  
| claim | 95d4063f | claim(uint8) |  
| pending | 914ffea5 | pending(uint8,address) |  
| getActiveNFTBalance | 72753943 | getActiveNFTBalance(address) |
```

Automatic general report

Files Description Table

File Name	SHA-1 Hash
/Users/macbook/Desktop/smart contracts/Xfuzion NFT.sol	c4c7055fe362f878cfc4be9bf3c7f4d58a8eb14b

Contracts Description Table

Contract	Type	Bases	
:-----: :-----: :-----: :-----:			
L	**Function Name**	**Visibility**	**Mutability**
Modifiers			
Counters Library			
L current	Internal		
L increment	Internal		
L decrement	Internal		
L reset	Internal		
Context Implementation			
L _msgSender	Internal		
L _msgData	Internal		
IERC165 Interface			
L supportsInterface	External		NO
IERC721 Interface IERC165			
L balanceOf	External		NO
L ownerOf	External		NO
L safeTransferFrom	External		NO
L transferFrom	External		NO
L approve	External		NO
L getApproved	External		NO
L setApprovalForAll	External		NO
L isApprovedForAll	External		NO
L safeTransferFrom	External		NO
Ownable Implementation Context			
L <Constructor>	Public		NO
L owner	Public		NO
L renounceOwnership	Public		onlyOwner
L transferOwnership	Public		onlyOwner
L _setOwner	Private		
IERC721Enumerable Interface IERC721			
L totalSupply	External		NO
L tokenOfOwnerByIndex	External		NO
L tokenByIndex	External		NO
ERC165 Implementation IERC165			
L supportsInterface	Public		NO

```

| **Strings** | Library | ||| |
| L | toString | Internal | 🔒 | | |
| L | toHexString | Internal | 🔒 | | |
| L | toHexString | Internal | 🔒 | | |
| ||||
| **Address** | Library | |||
| L | isContract | Internal | 🔒 | | |
| L | sendValue | Internal | 🔒 | 🔒 | |
| L | functionCall | Internal | 🔒 | 🔒 | |
| L | functionCall | Internal | 🔒 | 🔒 | |
| L | functionCallWithValue | Internal | 🔒 | 🔒 | |
| L | functionCallWithValue | Internal | 🔒 | 🔒 | |
| L | functionStaticCall | Internal | 🔒 | | |
| L | functionStaticCall | Internal | 🔒 | | |
| L | functionDelegateCall | Internal | 🔒 | 🔒 | |
| L | functionDelegateCall | Internal | 🔒 | 🔒 | |
| L | _verifyCallResult | Private | 🔒 | | |
| ||||
| **IERC721Metadata** | Interface | IERC721 | |||
| L | name | External | ! | NO! |
| L | symbol | External | ! | NO! |
| L | tokenURI | External | ! | NO! |
| ||||
| **IERC721Receiver** | Interface | | |||
| L | onERC721Received | External | ! | 🔒 | NO! |
| ||||
| **ERC721** | Implementation | Context, ERC165, IERC721, IERC721Metadata | |||
| L | <Constructor> | Public | ! | 🔒 | NO! |
| L | supportsInterface | Public | ! | NO! |
| L | balanceOf | Public | ! | NO! |
| L | ownerOf | Public | ! | NO! |
| L | name | Public | ! | NO! |
| L | symbol | Public | ! | NO! |
| L | tokenURI | Public | ! | NO! |
| L | _baseURI | Internal | 🔒 | | |
| L | approve | Public | ! | 🔒 | NO! |
| L | getApproved | Public | ! | NO! |
| L | setApprovalForAll | Public | ! | 🔒 | NO! |
| L | isApprovedForAll | Public | ! | NO! |
| L | transferFrom | Public | ! | 🔒 | NO! |
| L | safeTransferFrom | Public | ! | 🔒 | NO! |
| L | safeTransferFrom | Public | ! | 🔒 | NO! |
| L | _safeTransfer | Internal | 🔒 | 🔒 | |
| L | _exists | Internal | 🔒 | | |
| L | _isApprovedOrOwner | Internal | 🔒 | | |
| L | _safeMint | Internal | 🔒 | 🔒 | |
| L | _safeMint | Internal | 🔒 | 🔒 | |
| L | _mint | Internal | 🔒 | 🔒 | |
| L | _burn | Internal | 🔒 | 🔒 | |
| L | _transfer | Internal | 🔒 | 🔒 | |
| L | _approve | Internal | 🔒 | 🔒 | |
| L | _checkOnERC721Received | Private | 🔒 | 🔒 | |
| L | _beforeTokenTransfer | Internal | 🔒 | 🔒 | |
| ||||
| **ERC721Enumerable** | Implementation | ERC721, IERC721Enumerable | |||
| L | supportsInterface | Public | ! | NO! |
| L | tokenOfOwnerByIndex | Public | ! | NO! |
| L | totalSupply | Public | ! | NO! |



```

```

| L | tokenByIndex | Public ! | | NO! |
| L | _beforeTokenTransfer | Internal | | |
| L | _addTokenToOwnerEnumeration | Private | | |
| L | _addTokenToAllTokensEnumeration | Private | | |
| L | _removeTokenFromOwnerEnumeration | Private | | |
| L | _removeTokenFromAllTokensEnumeration | Private | | |
| | | |
| **IERC20** | Interface | | |
| L | totalSupply | External ! | | NO! |
| L | decimals | External ! | | NO! |
| L | symbol | External ! | | NO! |
| L | name | External ! | | NO! |
| L | getOwner | External ! | | NO! |
| L | balanceOf | External ! | | NO! |
| L | transfer | External ! | | NO! |
| L | allowance | External ! | | NO! |
| L | approve | External ! | | NO! |
| L | transferFrom | External ! | | NO! |
| | | |
| **XFuzionNFT** | Implementation | ERC721Enumerable, Ownable | | |
| L | <Constructor> | Public ! | | ERC721 |
| L | tokenURI | Public ! | | NO! |
| L | walletOfOwner | External ! | | NO! |
| L | setBaseURI | Public ! | | onlyOwner |
| L | setMaxNum | External ! | | onlyOwner |
| L | toggleSale | External ! | | onlyOwner |
| L | setWhiteList | Public ! | | onlyOwner |
| L | setPrice | External ! | | onlyOwner |
| L | _baseURI | Internal | | |
| L | mint | External ! | | NO! |
| L | claim | Public ! | | NO! |
| L | pending | Public ! | | NO! |
| L | safeDAITransfer | Internal | | |
| L | getActiveNFTBalance | Public ! | | NO! |
| L | setReferrer | Internal | | |

```

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.