Smart Contract
Security Audit
V1

MEFLEX NFT Marketplace Smart Contract

17/3/2023

https://saferico.com/

business@saferico.com
https://t.me/SFI_ANN

Table of Contents

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**: Ethereum
- **Name**: MEFLEX NFT Marketplace
- **Contract Address**: 0xef99c927c4e88e849c1dd1c3c0f66f7640d447d0
- **Code:**

https://goerli.etherscan.io/address/0x934228faca8df775d3ff7b079c33f0ec77a65fb3#code

Contracts address deployed to test net (Ethereum)

MEFLEX NFT Marketplace smart contract on Ethereum test net to test functions by the auditor.

https://goerli.etherscan.io/address/0xef99c927c4e88e849c1dd1c3c0f66f7640d447d0

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

The files:

MEFLEXMarketplace.sol

File and Function Level Report

File in Scope:

Contract Name	SHA 256 hash	Contract Address
MEFLEXMarketplace.sol	**5112ef811541a2a55440659ac084db225daf626b66b6d689f3076fb1fd9f23dd**	0xef99c927c4e88e849c1dd1c3c0f66f7640d447d0

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

Function	Test Result	Type / Return Type	Score
checkIfApproved	✓	Read / public	**Passed**
getAuctionDetails	✓	Read / public	**Passed**
getBidderDetails	✓	Read / public	**Passed**
getCreatorShare	✓	Read / public	**Passed**
getListing	✓	Read / public	**Passed**
getMarketShare	✓	Read / public	**Passed**
owner	✓	Read / public	**Passed**
getSellerShare	✓	Read / public	**Passed**
getOffer	✓	Read / public	**Passed**
acceptHighestBid	✓	Write / public	**Passed**
acceptOffer	✓	Write / public	**Passed**

buyItem	✓	Write / payable	Passed
cancelListing	✓	Write / public	Passed
cancelOfferedItem	✓	Write / public	Passed
renounceOwnership	✓	Write / public	Passed
claimYourPlacedBidAmount	✓	Write / public	Passed
createOffer	✓	Write / payable	Passed
endAuction	✓	Write / public	Passed
listItemForAuction	✓	Write / public	Passed
listItem	✓	Write / public	Passed
transferOwnership	✓	Write / public	Passed
placeBid	✓	Write / payable	Passed
rejectOffer	✓	Write / public	Passed
updateAuctionItem	✓	Write / public	Passed
updateListedItem	✓	Write / public	Passed

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 with Notes**
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:

#Contract code size exceeds 24576 bytes

Description

Contract implementation is too large in size to be deployed on main net. Ethereum with its spurious dragon release limited the size of the contracts deployable on main net to 24576 bytes.
The size of the contract MEFLEXMarketplace.sol goes way above this value.
You can read more here:
https://github.com/ethereum/EIPs/issues/170

Remediation

Define and use libraries for pure and view functions e.g. We can create a library which contains all the mathematical operations.

Status: Closed. The Team used to enable optimization at 200 to avoid this issue.
-

#Use of block.timestamp for comparisons

Description

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

Remediation

Avoid use of block.timestamp

Status: Acknowledged

#Multiple pragma statements

Line	Pragma
7	pragma solidity ^0.8.0;
50	pragma solidity ^0.8.13;
202	pragma solidity ^0.8.17;
209	pragma solidity ^0.8.13;
292	pragma solidity ^0.8.13;
311	pragma solidity ^0.8.0;
338	pragma solidity ^0.8.0;
473	pragma solidity ^0.8.1;
704	pragma solidity ^0.8.0;
730	pragma solidity ^0.8.0;
807	pragma solidity ^0.8.0;
836	pragma solidity ^0.8.0;
865	pragma solidity ^0.8.0;
892	pragma solidity ^0.8.0;
1334	pragma solidity ^0.8.0;
1359	pragma solidity ^0.8.0;
1474	pragma solidity ^0.8.0;
1502	pragma solidity ^0.8.0;
1562	pragma solidity ^0.8.17;
1734	pragma solidity ^0.8.0;
1799	pragma solidity ^0.8.0;
1882	pragma solidity ^0.8.0;
1815	pragma solidity ^0.8.17;

Description
There are multiple pragma statements in the code. The newest compiler version 0.8.19 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.

Very Low:

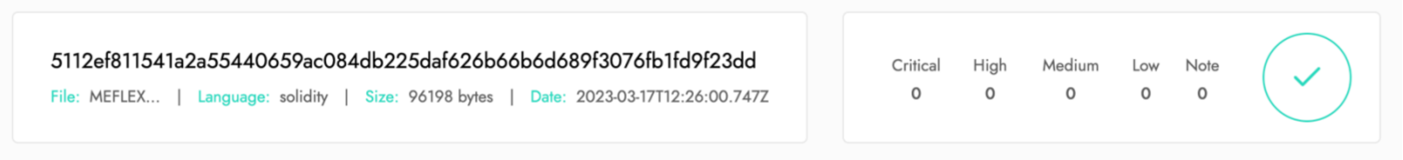
No Very Low severity vulnerabilities were found.

Notes:

No Notes were found.

Automatic Testing

1- Check for security

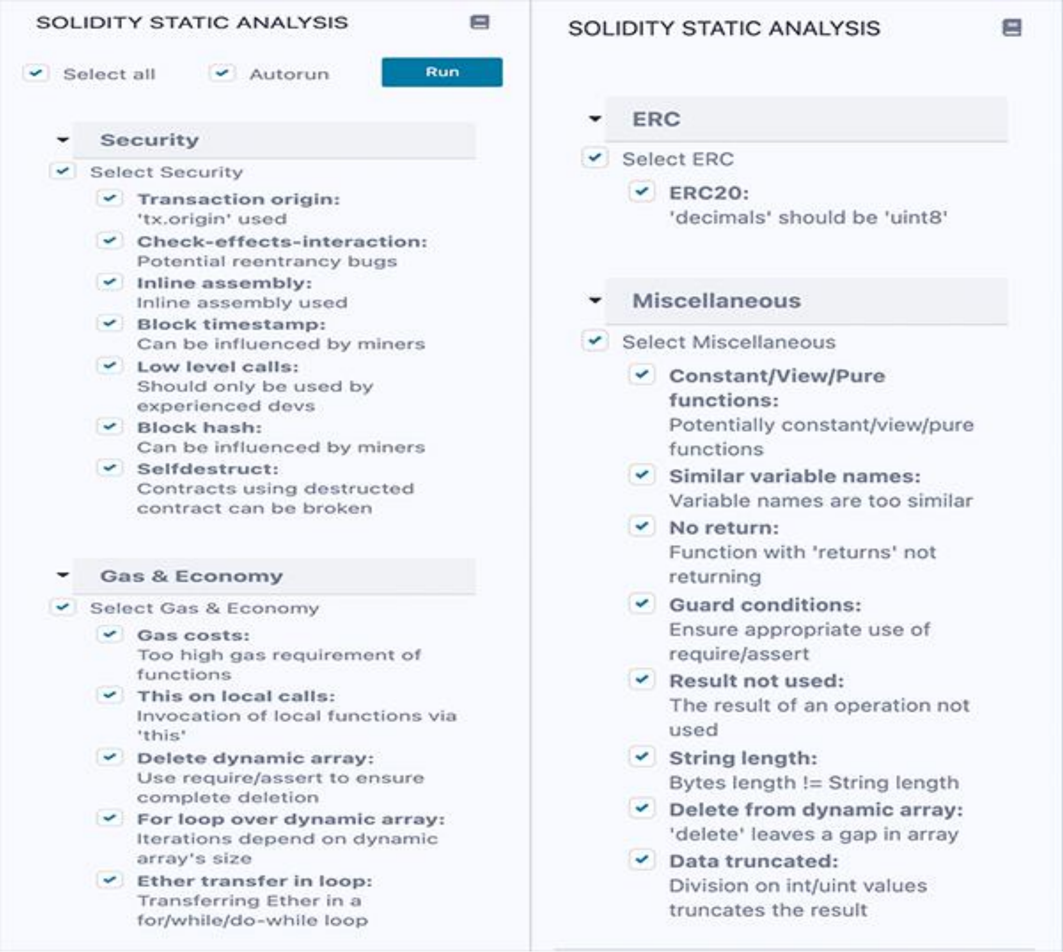


5112ef811541a2a55440659ac084db225daf626b66b6d689f3076fb1fd9f23dd

File: MEFLEX... | Language: solidity | Size: 96198 bytes | Date: 2023-03-17T12:26:00.747Z

Critical	High	Medium	Low	Note
0	0	0	0	0

2- SOLIDITY STATIC ANALYSIS



SOLIDITY STATIC ANALYSIS

☑ Select all ☑ Autorun

Security

☑ Select Security

- ☑ **Transaction origin:** 'tx.origin' used
- ☑ **Check-effects-interaction:** Potential reentrancy bugs
- ☑ **Inline assembly:** Inline assembly used
- ☑ **Block timestamp:** Can be influenced by miners
- ☑ **Low level calls:** Should only be used by experienced devs
- ☑ **Block hash:** Can be influenced by miners
- ☑ **Selfdestruct:** Contracts using destructed contract can be broken

Gas & Economy

☑ Select Gas & Economy

- ☑ **Gas costs:** Too high gas requirement of functions
- ☑ **This on local calls:** Invocation of local functions via 'this'
- ☑ **Delete dynamic array:** Use require/assert to ensure complete deletion
- ☑ **For loop over dynamic array:** Iterations depend on dynamic array's size
- ☑ **Ether transfer in loop:** Transferring Ether in a for/while/do-while loop

SOLIDITY STATIC ANALYSIS

ERC

☑ Select ERC

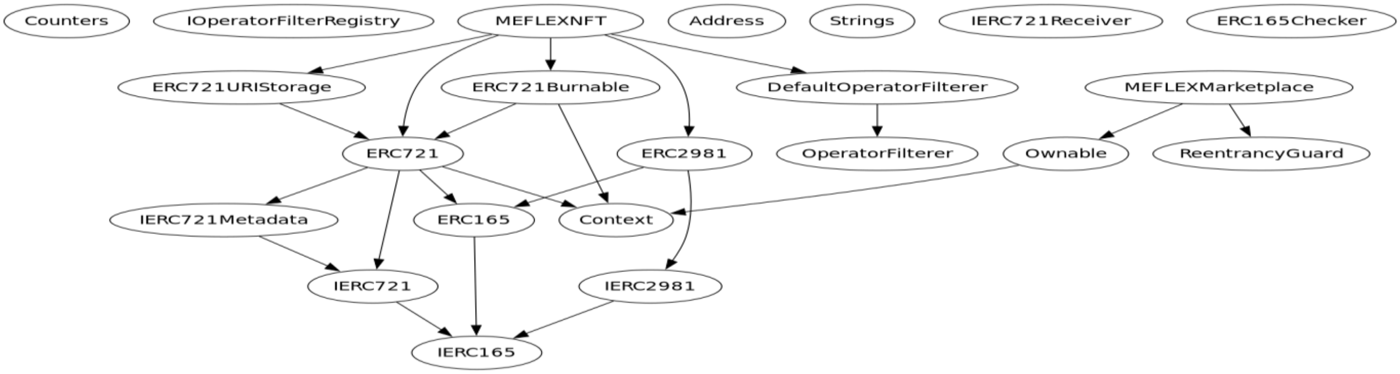
- ☑ **ERC20:** 'decimals' should be 'uint8'

Miscellaneous

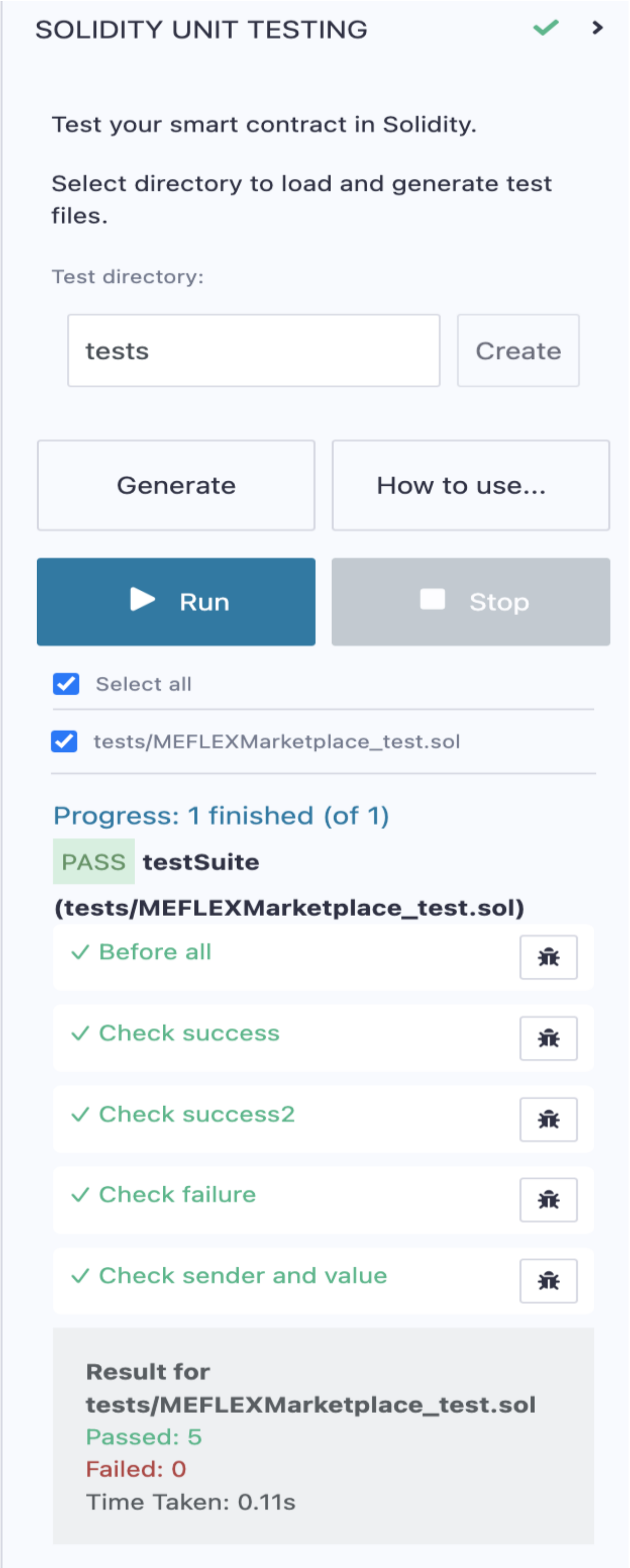
☑ Select Miscellaneous

- ☑ **Constant/View/Pure functions:** Potentially constant/view/pure functions
- ☑ **Similar variable names:** Variable names are too similar
- ☑ **No return:** Function with 'returns' not returning
- ☑ **Guard conditions:** Ensure appropriate use of require/assert
- ☑ **Result not used:** The result of an operation not used
- ☑ **String length:** Bytes length != String length
- ☑ **Delete from dynamic array:** 'delete' leaves a gap in array
- ☑ **Data truncated:** Division on int/uint values truncates the result

3- Inheritance graph



4- SOLIDITY UNIT TESTING



Test your smart contract in Solidity.

Select directory to load and generate test files.

Test directory: tests

☑ Select all

☑ tests/MEFLEXMarketplace_test.sol

Progress: 1 finished (of 1)

PASS **testSuite (tests/MEFLEXMarketplace_test.sol)**

✓ Before all

✓ Check success

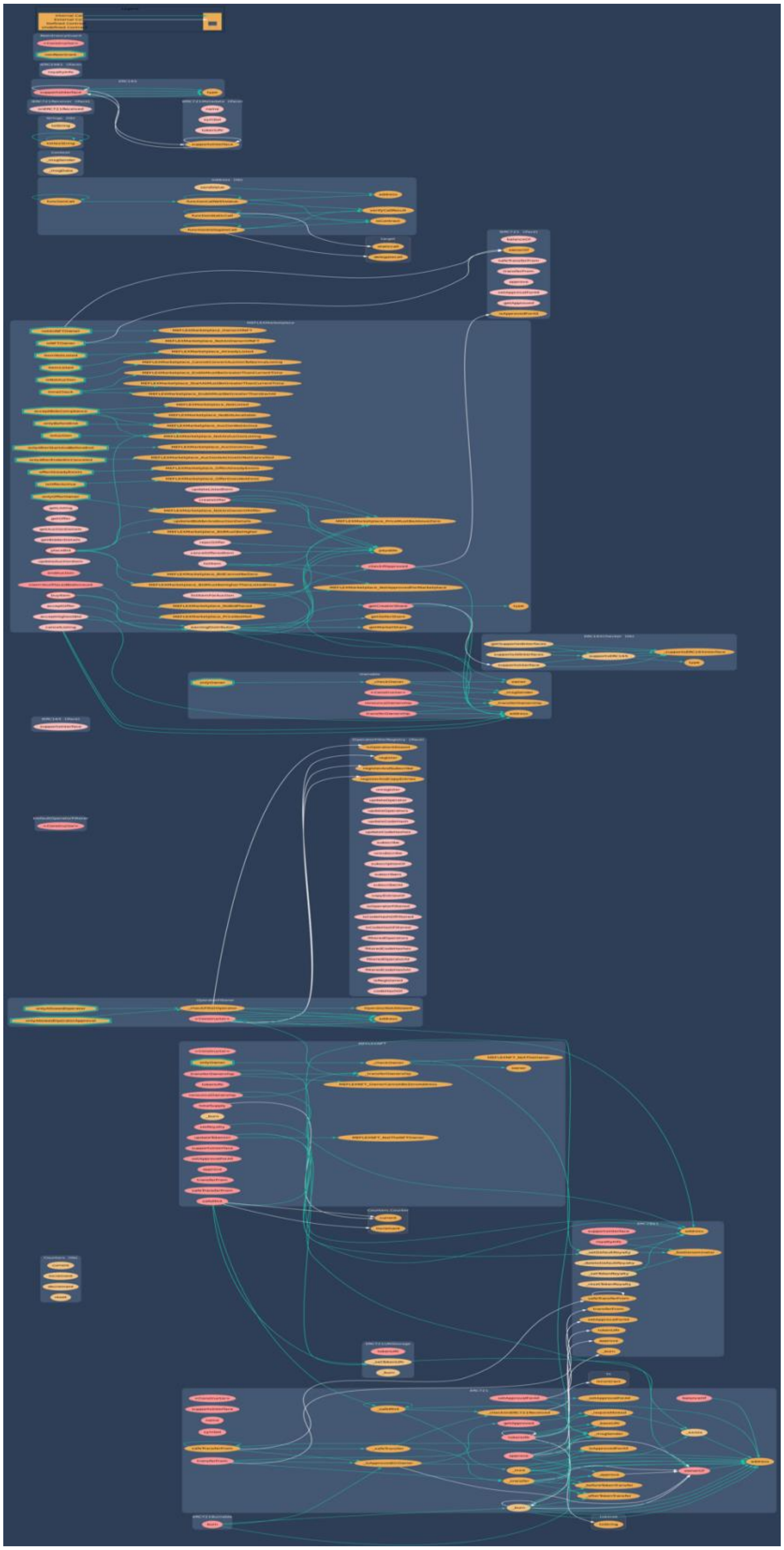
✓ Check success2

✓ Check failure

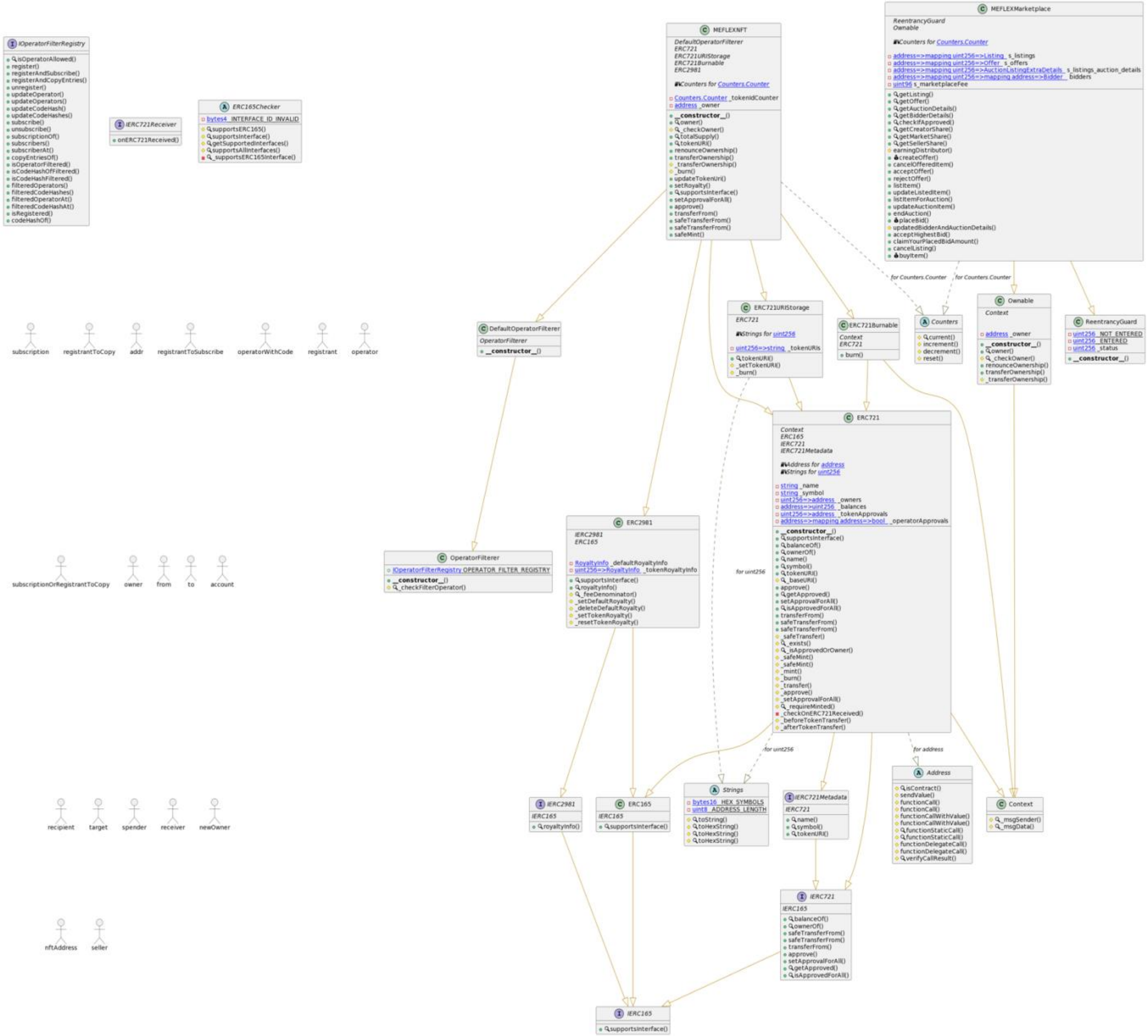
✓ Check sender and value

Result for tests/MEFLEXMarketplace_test.sol
Passed: 5
Failed: 0
Time Taken: 0.11s

5- Call graph



Unified Modeling Language (UML)



Functions signature

```
Sighash    |   Function Signature
========================
16279055  =>  isContract(address)
ad04a8d1  =>  current(Counter)
e2bee435  =>  increment(Counter)
854ec98e  =>  decrement(Counter)
440d212a  =>  reset(Counter)
c6171134  =>  isOperatorAllowed(address,address)
4420e486  =>  register(address)
7d3e3dbe  =>  registerAndSubscribe(address,address)
a0af2903  =>  registerAndCopyEntries(address,address)
2ec2c246  =>  unregister(address)
a2f367ab  =>  updateOperator(address,address,bool)
a14584c1  =>  updateOperators(address,address[],bool)
712fc00b  =>  updateCodeHash(address,bytes32,bool)
063298b6  =>  updateCodeHashes(address,bytes32[],bool)
b314d414  =>  subscribe(address,address)
34a0dc10  =>  unsubscribe(address,bool)
3c5030bb  =>  subscriptionOf(address)
5745ae28  =>  subscribers(address)
55940e51  =>  subscriberAt(address,uint256)
1e06b4b4  =>  copyEntriesOf(address,address)
e4aecb54  =>  isOperatorFiltered(address,address)
5eae3173  =>  isCodeHashOfFiltered(address,address)
6af0c315  =>  isCodeHashFiltered(address,bytes32)
c4308805  =>  filteredOperators(address)
22fa2762  =>  filteredCodeHashes(address)
3f1cc5fa  =>  filteredOperatorAt(address,uint256)
a6529eb5  =>  filteredCodeHashAt(address,uint256)
c3c5a547  =>  isRegistered(address)
bbd652c7  =>  codeHashOf(address)
3ce8f2bd  =>  _checkFilterOperator(address)
01ffc9a7  =>  supportsInterface(bytes4)
70a08231  =>  balanceOf(address)
6352211e  =>  ownerOf(uint256)
b88d4fde  =>  safeTransferFrom(address,address,uint256,bytes)
42842e0e  =>  safeTransferFrom(address,address,uint256)
23b872dd  =>  transferFrom(address,address,uint256)
095ea7b3  =>  approve(address,uint256)
a22cb465  =>  setApprovalForAll(address,bool)
081812fc  =>  getApproved(uint256)
e985e9c5  =>  isApprovedForAll(address,address)
24a084df  =>  sendValue(address,uint256)
a0b5ffb0  =>  functionCall(address,bytes)
241b5886  =>  functionCall(address,bytes,string)
2a011594  =>  functionCallWithValue(address,bytes,uint256)
d525ab8a  =>  functionCallWithValue(address,bytes,uint256,string)
c21d36f3  =>  functionStaticCall(address,bytes)
dbc40fb9  =>  functionStaticCall(address,bytes,string)
ee33b7e2  =>  functionDelegateCall(address,bytes)
57387df0  =>  functionDelegateCall(address,bytes,string)
```

```
946b5793  =>  verifyCallResult(bool,bytes,string)
119df25f  =>  _msgSender()
8b49d47e  =>  _msgData()
6900a3ae  =>  toString(uint256)
8fba8d5c  =>  toHexString(uint256)
63e1cbea  =>  toHexString(uint256,uint256)
1bb0c665  =>  toHexString(address)
150b7a02  =>  onERC721Received(address,address,uint256,bytes)
06fdde03  =>  name()
95d89b41  =>  symbol()
c87b56dd  =>  tokenURI(uint256)
743976a0  =>  _baseURI()
24b6b8c0  =>  _safeTransfer(address,address,uint256,bytes)
f8e76cc0  =>  _exists(uint256)
4cdc9549  =>  _isApprovedOrOwner(address,uint256)
b3e1c718  =>  _safeMint(address,uint256)
6a4f832b  =>  _safeMint(address,uint256,bytes)
4e6ec247  =>  _mint(address,uint256)
9b1f9e74  =>  _burn(uint256)
30e0789e  =>  _transfer(address,address,uint256)
7b7d7225  =>  _approve(address,uint256)
8c4e3f32  =>  _setApprovalForAll(address,address,bool)
a0aea85d  =>  _requireMinted(uint256)
1fd01de1  =>  _checkOnERC721Received(address,address,uint256,bytes)
cad3be83  =>  _beforeTokenTransfer(address,address,uint256)
8f811a1c  =>  _afterTokenTransfer(address,address,uint256)
2a55205a  =>  royaltyInfo(uint256,uint256)
bf8e572e  =>  _feeDenominator()
b1c1ab1b  =>  _setDefaultRoyalty(address,uint96)
36fdc63c  =>  _deleteDefaultRoyalty()
b552a471  =>  _setTokenRoyalty(uint256,address,uint96)
604ba39e  =>  _resetTokenRoyalty(uint256)
42966c68  =>  burn(uint256)
01538868  =>  _setTokenURI(uint256,string)
8da5cb5b  =>  owner()
53a72975  =>  _checkOwner()
18160ddd  =>  totalSupply()
715018a6  =>  renounceOwnership()
f2fde38b  =>  transferOwnership(address)
d29d44ee  =>  _transferOwnership(address)
d31af484  =>  updateTokenUri(uint256,string)
8f2fc60b  =>  setRoyalty(address,uint96)
d204c45e  =>  safeMint(address,string)
c398a925  =>  supportsERC165(address)
d9057007  =>  supportsInterface(address,bytes4)
77e6b4cc  =>  getSupportedInterfaces(address,bytes4[])
4b9dd904  =>  supportsAllInterfaces(address,bytes4[])
20f10ae7  =>  _supportsERC165Interface(address,bytes4)
88700d1c  =>  getListing(address,uint256)
ac71045e  =>  getOffer(address,uint256)
0cd87c68  =>  getAuctionDetails(address,uint256)
899a739d  =>  getBidderDetails(address,uint256,address)
fb8bb3ad  =>  checkIfApproved(address)
902bcb65  =>  getCreatorShare(address,address,uint256,uint256)
49b20e6b  =>  getMarketShare(uint256)
12d41864  =>  getSellerShare(uint256,uint256,uint256)
```

```
7aa7ffca  =>  earningDistributor(address,address,uint256,uint256)
d783c86d  =>  createOffer(NFTDetailsParams)
a1134129  =>  cancelOfferedItem(NFTDetailsParams)
2ae0dab5  =>  acceptOffer(NFTDetailsParams)
8912dbf0  =>  rejectOffer(NFTDetailsParams)
95a20432  =>  listItem(NFTDetailsParams,uint256)
9ee01cb2  =>  updateListedItem(NFTDetailsParams,uint256)
305fae8c  =>  listItemForAuction(NFTDetailsParams,AuctionListingParams)
0797ead9  =>  updateAuctionItem(NFTDetailsParams,AuctionListingParams)
1b648426  =>  endAuction(NFTDetailsParams)
a716279a  =>  placeBid(NFTDetailsParams)
7893a0cd  =>  updatedBidderAndAuctionDetails(address,uint256)
b9ed8d28  =>  acceptHighestBid(NFTDetailsParams)
9bcfe013  =>  claimYourPlacedBidAmount(NFTDetailsParams)
713b6727  =>  cancelListing(NFTDetailsParams)
4a8898e4  =>  buyItem(NFTDetailsParams)
```

Automatic general report

```
Files Description Table

|  File Name  |  SHA-1 Hash  |
|-------------|--------------|
| /Users/macbook/Desktop/smart contracts/MEFLEXMarketplace.sol |
a3aa291c6cebc2aab19f16cadc44643a721bdba0 |

 Contracts Description Table

|  Contract  |         Type        |       Bases      |                  |
|
|:----------:|:-------------------:|:----------------:|:----------------:|:--------
-------:|
|     └      |  **Function Name**  |  **Visibility**  |  **Mutability**  |
**Modifiers**  |
||||||
| **Counters** | Library |  |||
| └ | current | Internal 🔒 |   |  |
| └ | increment | Internal 🔒 | 🛑  |  |
| └ | decrement | Internal 🔒 | 🛑  |  |
| └ | reset | Internal 🔒 | 🛑  |  |
||||||
| **IOperatorFilterRegistry** | Interface |  |||
| └ | isOperatorAllowed | External ❗ |   |NO❗ |
| └ | register | External ❗ | 🛑  |NO❗ |
| └ | registerAndSubscribe | External ❗ | 🛑  |NO❗ |
| └ | registerAndCopyEntries | External ❗ | 🛑  |NO❗ |
| └ | unregister | External ❗ | 🛑  |NO❗ |
| └ | updateOperator | External ❗ | 🛑  |NO❗ |
| └ | updateOperators | External ❗ | 🛑  |NO❗ |
| └ | updateCodeHash | External ❗ | 🛑  |NO❗ |
| └ | updateCodeHashes | External ❗ | 🛑  |NO❗ |
| └ | subscribe | External ❗ | 🛑  |NO❗ |
| └ | unsubscribe | External ❗ | 🛑  |NO❗ |
| └ | subscriptionOf | External ❗ | 🛑  |NO❗ |
| └ | subscribers | External ❗ | 🛑  |NO❗ |
| └ | subscriberAt | External ❗ | 🛑  |NO❗ |
| └ | copyEntriesOf | External ❗ | 🛑  |NO❗ |
| └ | isOperatorFiltered | External ❗ | 🛑  |NO❗ |
| └ | isCodeHashOfFiltered | External ❗ | 🛑  |NO❗ |
| └ | isCodeHashFiltered | External ❗ | 🛑  |NO❗ |
| └ | filteredOperators | External ❗ | 🛑  |NO❗ |
| └ | filteredCodeHashes | External ❗ | 🛑  |NO❗ |
| └ | filteredOperatorAt | External ❗ | 🛑  |NO❗ |
| └ | filteredCodeHashAt | External ❗ | 🛑  |NO❗ |
| └ | isRegistered | External ❗ | 🛑  |NO❗ |
| └ | codeHashOf | External ❗ | 🛑  |NO❗ |
||||||
| **OperatorFilterer** | Implementation |  |||
| └ | <Constructor> | Public ❗ | 🛑  |NO❗ |
| └ | _checkFilterOperator | Internal 🔒 |   | |
||||||
| **DefaultOperatorFilterer** | Implementation | OperatorFilterer |||
| └ | <Constructor> | Public ❗️ | 🛑  | OperatorFilterer |
||||||
| **IERC165** | Interface |  |||
| └ | supportsInterface | External ❗️ |   |NO❗️ |
||||||
| **IERC721** | Interface | IERC165 |||
| └ | balanceOf | External ❗️ |   |NO❗️ |
| └ | ownerOf | External ❗️ |   |NO❗️ |
| └ | safeTransferFrom | External ❗️ | 🛑  |NO❗️ |
| └ | safeTransferFrom | External ❗️ | 🛑  |NO❗️ |
| └ | transferFrom | External ❗️ | 🛑  |NO❗️ |
| └ | approve | External ❗️ | 🛑  |NO❗️ |
| └ | setApprovalForAll | External ❗️ | 🛑  |NO❗️ |
| └ | getApproved | External ❗️ |   |NO❗️ |
| └ | isApprovedForAll | External ❗️ |   |NO❗️ |
||||||
| **Address** | Library |  |||
| └ | isContract | Internal 🔒 |   | |
| └ | sendValue | Internal 🔒 | 🛑  | |
| └ | functionCall | Internal 🔒 | 🛑  | |
| └ | functionCall | Internal 🔒 | 🛑  | |
| └ | functionCallWithValue | Internal 🔒 | 🛑  | |
| └ | functionCallWithValue | Internal 🔒 | 🛑  | |
| └ | functionStaticCall | Internal 🔒 |   | |
| └ | functionStaticCall | Internal 🔒 |   | |
| └ | functionDelegateCall | Internal 🔒 | 🛑  | |
| └ | functionDelegateCall | Internal 🔒 | 🛑  | |
| └ | verifyCallResult | Internal 🔒 |   | |
||||||
| **Context** | Implementation |  |||
| └ | _msgSender | Internal 🔒 |   | |
| └ | _msgData | Internal 🔒 |   | |
||||||
| **Strings** | Library |  |||
| └ | toString | Internal 🔒 |   | |
| └ | toHexString | Internal 🔒 |   | |
| └ | toHexString | Internal 🔒 |   | |
| └ | toHexString | Internal 🔒 |   | |
||||||
| **IERC721Receiver** | Interface |  |||
| └ | onERC721Received | External ❗️ | 🛑  |NO❗️ |
||||||
| **ERC165** | Implementation | IERC165 |||
| └ | supportsInterface | Public ❗️ |   |NO❗️ |
||||||
| **IERC721Metadata** | Interface | IERC721 |||
| └ | name | External ❗️ |   |NO❗️ |
| └ | symbol | External ❗️ |   |NO❗️ |
| └ | tokenURI | External ❗️ |   |NO❗️ |
||||||
| **ERC721** | Implementation | Context, ERC165, IERC721, IERC721Metadata |||
| └ | <Constructor> | Public ❗️ | 🛑  |NO❗️ |
| └ | supportsInterface | Public ❗️ |   |NO❗️ |
| └ | balanceOf | Public ❗️ |   |NO❗️ |
| └ | ownerOf | Public ❗️ |   |NO❗️ |
| └ | name | Public ❗️ |   |NO❗️ |
| └ | symbol | Public ❗️ |   |NO❗️ |
| └ | tokenURI | Public ❗️ |   |NO❗️ |
| └ | _baseURI | Internal 🔒 |   | |
| └ | approve | Public ❗️ | 🛑  |NO❗️ |
| └ | getApproved | Public ❗️ |   |NO❗️ |
| └ | setApprovalForAll | Public ❗️ | 🛑  |NO❗️ |
| └ | isApprovedForAll | Public ❗️ |   |NO❗️ |
| └ | transferFrom | Public ❗️ | 🛑  |NO❗️ |
| └ | safeTransferFrom | Public ❗️ | 🛑  |NO❗️ |
| └ | safeTransferFrom | Public ❗️ | 🛑  |NO❗️ |
| └ | _safeTransfer | Internal 🔒 | 🛑  | |
| └ | _exists | Internal 🔒 |   | |
| └ | _isApprovedOrOwner | Internal 🔒 |   | |
| └ | _safeMint | Internal 🔒 | 🛑  | |
| └ | _safeMint | Internal 🔒 | 🛑  | |
| └ | _mint | Internal 🔒 | 🛑  | |
| └ | _burn | Internal 🔒 | 🛑  | |
| └ | _transfer | Internal 🔒 | 🛑  | |
| └ | _approve | Internal 🔒 | 🛑  | |
| └ | _setApprovalForAll | Internal 🔒 | 🛑  | |
| └ | _requireMinted | Internal 🔒 |   | |
| └ | _checkOnERC721Received | Private 🔐 | 🛑  | |
| └ | _beforeTokenTransfer | Internal 🔒 | 🛑  | |
| └ | _afterTokenTransfer | Internal 🔒 | 🛑  | |
||||||
| **IERC2981** | Interface | IERC165 |||
| └ | royaltyInfo | External ❗️ |   |NO❗️ |
||||||
| **ERC2981** | Implementation | IERC2981, ERC165 |||
| └ | supportsInterface | Public ❗️ |   |NO❗️ |
| └ | royaltyInfo | Public ❗️ |   |NO❗️ |
| └ | _feeDenominator | Internal 🔒 |   | |
| └ | _setDefaultRoyalty | Internal 🔒 | 🛑  | |
| └ | _deleteDefaultRoyalty | Internal 🔒 | 🛑  | |
| └ | _setTokenRoyalty | Internal 🔒 | 🛑  | |
| └ | _resetTokenRoyalty | Internal 🔒 | 🛑  | |
||||||
| **ERC721Burnable** | Implementation | Context, ERC721 |||
| └ | burn | Public ❗️ | 🛑  |NO❗️ |
||||||
| **ERC721URIStorage** | Implementation | ERC721 |||
| └ | tokenURI | Public ❗️ |   |NO❗️ |
| └ | _setTokenURI | Internal 🔒 | 🛑  | |
| └ | _burn | Internal 🔒 | 🛑  | |
||||||
| **MEFLEXNFT** | Implementation | DefaultOperatorFilterer, ERC721,
ERC721URIStorage, ERC721Burnable, ERC2981 |||
| └ | <Constructor> | Public ❗️ | 🛑  | ERC721 |
| └ | owner | Public ❗️ |   |NO❗️ |
| └ | _checkOwner | Internal 🔒 |   | |
| └ | totalSupply | Public ❗️ |   |NO❗️ |
| └ | tokenURI | Public ❗️ |   |NO❗️ |
| └ | renounceOwnership | Public ❗️ | 🛑  | onlyOwner |
| └ | transferOwnership | Public ❗️ | 🛑  | onlyOwner |
| └ | _transferOwnership | Internal 🔒 | 🛑  | |
| └ | _burn | Internal 🔒 | 🛑  | |
| └ | updateTokenUri | Public ❗️ | 🛑  |NO❗️ |
| └ | setRoyalty | Public ❗️ | 🛑  | onlyOwner |
| └ | supportsInterface | Public ❗️ |   |NO❗️ |
| └ | setApprovalForAll | Public ❗️ | 🛑  | onlyAllowedOperatorApproval |
| └ | approve | Public ❗️ | 🛑  | onlyAllowedOperatorApproval |
| └ | transferFrom | Public ❗️ | 🛑  | onlyAllowedOperator |
| └ | safeTransferFrom | Public ❗️ | 🛑  | onlyAllowedOperator |
| └ | safeTransferFrom | Public ❗️ | 🛑  | onlyAllowedOperator |
| └ | safeMint | Public ❗️ | 🛑  | onlyOwner |
||||||
| **ReentrancyGuard** | Implementation |  |||
| └ | <Constructor> | Public ❗️ | 🛑  |NO❗️ |
||||||
| **Ownable** | Implementation | Context |||
| └ | <Constructor> | Public ❗️ | 🛑  |NO❗️ |
| └ | owner | Public ❗️ |   |NO❗️ |
| └ | _checkOwner | Internal 🔒 |   | |
| └ | renounceOwnership | Public ❗️ | 🛑  | onlyOwner |
| └ | transferOwnership | Public ❗️ | 🛑  | onlyOwner |
| └ | _transferOwnership | Internal 🔒 | 🛑  | |
||||||
| **ERC165Checker** | Library |  |||
| └ | supportsERC165 | Internal 🔒 |   | |
| └ | supportsInterface | Internal 🔒 |   | |
| └ | getSupportedInterfaces | Internal 🔒 |   | |
| └ | supportsAllInterfaces | Internal 🔒 |   | |
| └ | _supportsERC165Interface | Private 🔐 |   | |
||||||
| **MEFLEXMarketplace** | Implementation | ReentrancyGuard, Ownable |||
| └ | getListing | External ❗️ |   |NO❗️ |
| └ | getOffer | External ❗️ |   |NO❗️ |
| └ | getAuctionDetails | External ❗️ |   |NO❗️ |
| └ | getBidderDetails | External ❗️ |   |NO❗️ |
| └ | checkIfApproved | Public ❗️ |   |NO❗️ |
| └ | getCreatorShare | Public ❗️ |   |NO❗️ |
| └ | getMarketShare | Public ❗️ |   |NO❗️ |
| └ | getSellerShare | Public ❗️ |   |NO❗️ |
| └ | earningDistributor | Internal 🔒 | 🛑  | |
| └ | createOffer | External ❗️ |  💵 | offerAlreadyExists notAnNFTOwner |
| └ | cancelOfferedItem | External ❗️ | 🛑  | isOfferActive onlyOfferOwner
nonReentrant |
| └ | acceptOffer | External ❗️ | 🛑  | isOfferActive isNFTOwner nonReentrant |
| └ | rejectOffer | External ❗️ | 🛑  | isOfferActive isNFTOwner nonReentrant |
| └ | listItem | External ❗️ | 🛑  | itemNotListed isNFTOwner |
| └ | updateListedItem | External ❗️ | 🛑  | itemListed isNotAuction isNFTOwner |
| └ | listItemForAuction | External ❗️ | 🛑  | itemNotListed isNFTOwner timeCheck |
| └ | updateAuctionItem | External ❗️ | 🛑  | itemListed isAuction isNFTOwner
timeCheck |
| └ | endAuction | Public ❗️ | 🛑  | itemListed isAuction isNFTOwner |
| └ | placeBid | External ❗️ |  💵 | itemListed onlyAfterStartAndBeforeEnd
```

```
notAnNFTOwner |
| └ | updatedBidderAndAuctionDetails | Internal 🔒 | 🛑  | |
| └ | acceptHighestBid | External ❗️ | 🛑  | isNFTOwner itemListed
acceptBidsCompliance nonReentrant |
| └ | claimYourPlacedBidAmount | Public ❗️ | 🛑  | onlyAfterEndedOrCanceled
notAnNFTOwner nonReentrant |
| └ | cancelListing | External ❗️ | 🛑  | isNFTOwner itemListed onlyBeforeEnd |
| └ | buyItem | External ❗️ |  💵 | itemListed notAnNFTOwner nonReentrant |

 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.