# Smart Contract
# Security Audit
# V1

## Auction House Smart Contract

https://www.nftyauctions.xyz/

13/12/2022

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**: Auction House
- **Auction House Smart Contract Address**: 0xd6c1bc7aa21bbd9fd995f6147a46e7523b18456b
- **Auction House Smart Contract Code:**

https://etherscan.io/address/0xd6c1bc7aa21bbd9fd995f6147a46e7523b18456b#code

- **WhitelistRegistry Smart Contract Code**:
https://etherscan.io/address/0x23a822aebcc982bce3c73a81f4b44f03de18097d#code

- **Main net UI**: https://www.nftyauctions.xyz/
- **Test net UI:** https://testnets.nftyauctions.xyz/
- **Twitter:** https://twitter.com/NFTYauctions
- **Discord:** https://discord.com/invite/BvACuaZr52

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

The files:

AuctionHouse.sol

# File and Function Level Report

## File in Scope:

| Contract Name | SHA 256 hash | Contract Address |
|---|---|---|
| AuctionHouse.sol | **e87d0ac4fb53a0f3fb8182e9b426297524ea05b50ee23056fa59b29bb6176322** | 0xd6c1bc7aa21bbd9fd995f6147a46e7523b18456b |

- Contract: AuctionHouse
- Inherit: Ownable, ReentrancyGuard, ERC721Holder, ERC1155Holder
- Observation: All passed including security check
- Test Report: passed
- Score: passed
- Conclusion: passed

| Function | Test Result | Type / Return Type | Score |
|---|---|---|---|
| allowListings | ✓ | Read / public | **Passed** |
| assets | ✓ | Read / public | **Passed** |
| auctions | ✓ | Read / public | **Passed** |
| getAuction | ✓ | Read / public | **Passed** |
| getAuctionAssets | ✓ | Read / public | **Passed** |
| getHighestBid | ✓ | Read / public | **Passed** |
| Owner | ✓ | Read / public | **Passed** |
| INTERFACE_ID_ERC1155 | ✓ | Read / public | **Passed** |
| public_ INTERFACE_ID_ERC721 | ✓ | Read / public | **Passed** |
| isBeta | ✓ | Read / public | **Passed** |
| maxLotSize | ✓ | Read / public | **Passed** |
| penaltyFee | ✓ | Read / public | **Passed** |

| | | | |
|---|---|---|---|
| protocolFee | ✓ | Read / public | **Passed** |
| protocolFeeRecipient | ✓ | Read / public | **Passed** |
| supportsInterface | ✓ | Read / public | **Passed** |
| totalAuctionCount | ✓ | Read / public | **Passed** |
| totalBidCount | ✓ | Read / public | **Passed** |
| cancelAuction | ✓ | Write / payable | **Passed** |
| changeReservePrice | ✓ | Write / public | **Passed** |
| createAuction | ✓ | Write / public | **Passed** |
| createBid | ✓ | Write / payable | **Passed** |
| increaseBid | ✓ | Write / payable | **Passed** |
| settleAuction | ✓ | Write / payable | **Passed** |
| onERC1155BatchReceived | ✓ | Write / public | **Passed** |
| transferOwnership | ✓ | Write / public | **Passed** |
| onERC1155Received | ✓ | Write / public | **Passed** |
| onERC721Received | ✓ | Write / public | **Passed** |
| revertAuction | ✓ | Write / public | **Passed** |
| renounceOwnership | ✓ | Write / public | **Passed** |
| toggleAllowListings | ✓ | Write / public | **Passed** |
| toggleBeta | ✓ | Write / public | **Passed** |
| updateMaxLotSize | ✓ | Write / public | **Passed** |
| updatePenaltyFee | ✓ | Write / public | **Passed** |
| updateProtocolFee | ✓ | Write / public | **Passed** |
| updateProtocolFeeRecipient | ✓ | Write / public | **Passed** |
| updateWhitelistRegistry | ✓ | 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.<br>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,<br>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

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

**Very Low:**

No Very Low severity vulnerabilities were found.

**Notes:**

#Unnecessary import of IERC721Receiver library

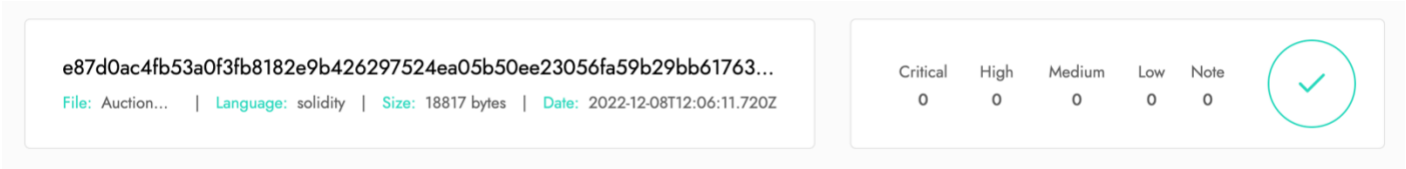
Description
The main contract inherits: Ownable, ReentrancyGuard, ERC721Holder, ERC1155Holder which is already import IERC721Receiver library, so no need to import it again in the main contract.

Remediation
Remove unnecessary library from the main contract save some gas fees.

Status: Acknowledged

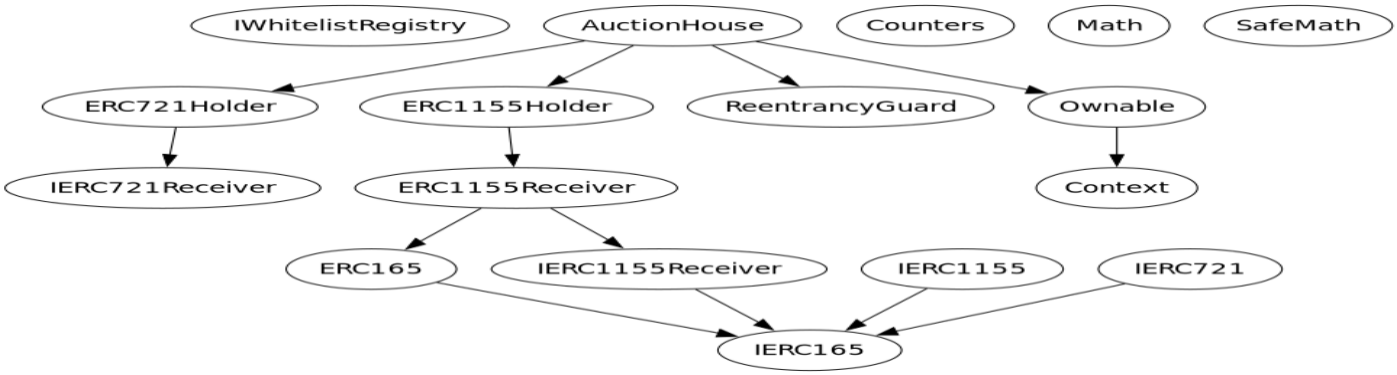
# Automatic Testing

1- Check for security

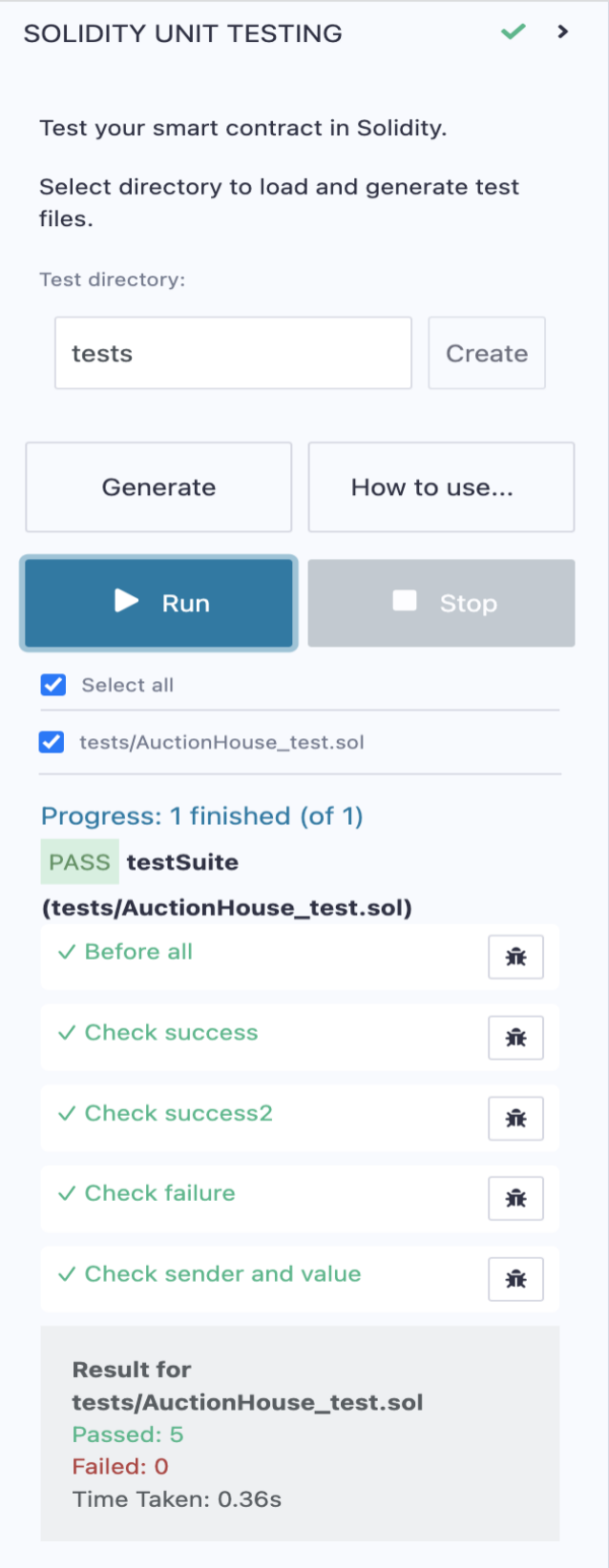



2- SOLIDITY STATIC ANALYSIS

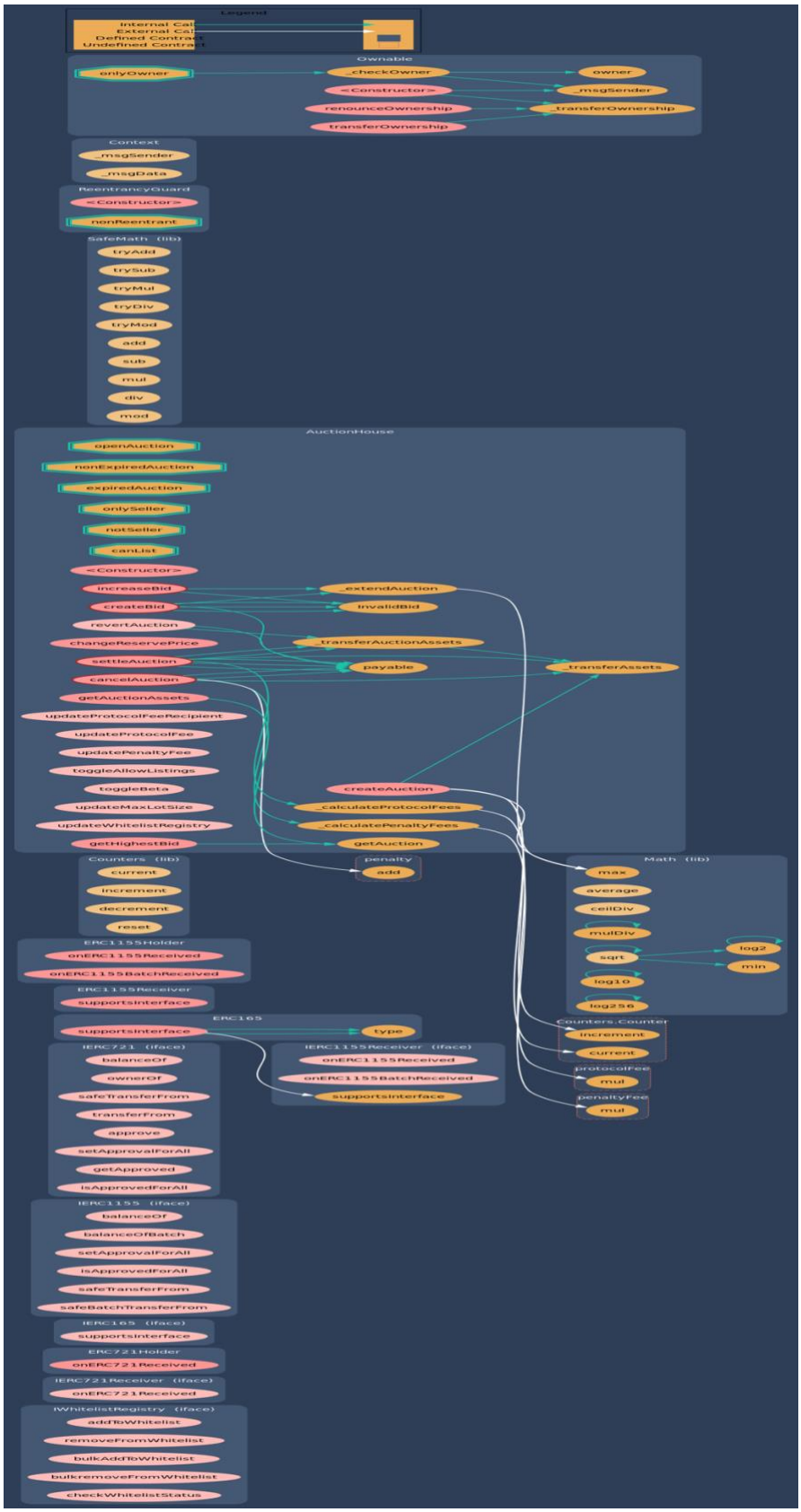

3- Inheritance graph



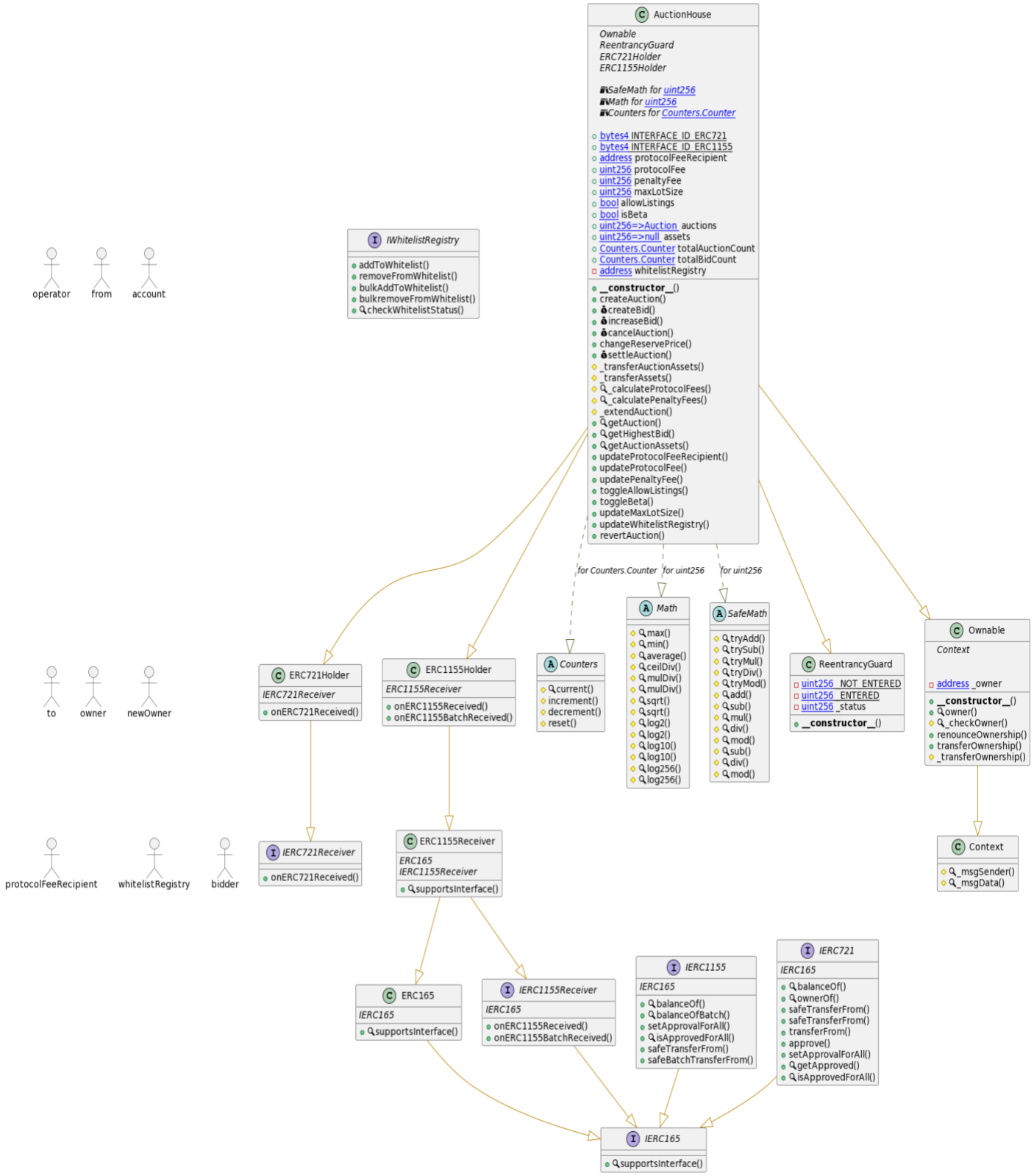
# 4- SOLIDITY UNIT TESTING



5- Call graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
Ownable
onlyOwner
_checkOwner
owner
<Constructor>
_msgSender
renounceOwnership
_transferOwnership
transferOwnership
Context
_msgSender
_msgData
ReentrancyGuard
<Constructor>
nonReentrant
SafeMath (lib)
tryAdd
trySub
tryMul
tryDiv
tryMod
add
sub
mul
div
mod
AuctionHouse
openAuction
nonExpiredAuction
expiredAuction
onlySeller
notSeller
canList
<Constructor>
increaseBid
_extendAuction
createBid
invalidBid
revertAuction
changeReservePrice
_transferAuctionAssets
settleAuction
payable
_transferAssets
cancelAuction
getAuctionAssets
updateProtocolFeeRecipient
updateProtocolFee
updatePenaltyFee
toggleAllowListings
toggleBeta
createAuction
updateMaxLotSize
_calculateProtocolFees
updateWhitelistRegistry
_calculatePenaltyFees
getHighestBid
getAuction
Counters (lib)
current
increment
decrement
reset
penalty
add
Math (lib)
max
average
ceilDiv
mulDiv
sqrt
log2
min
log10
log256
ERC1155Holder
onERC1155Received
onERC1155BatchReceived
ERC1155Receiver
supportsInterface
ERC165
supportsInterface
type
Counters.Counter
increment
current
IERC721 (iface)
balanceOf
ownerOf
safeTransferFrom
transferFrom
approve
setApprovalForAll
getApproved
isApprovedForAll
IERC1155Receiver (iface)
onERC1155Received
onERC1155BatchReceived
supportsInterface
protocolFee
mul
penaltyFee
mul
IERC1155 (iface)
balanceOf
balanceOfBatch
setApprovalForAll
isApprovedForAll
safeTransferFrom
safeBatchTransferFrom
IERC165 (iface)
supportsInterface
ERC721Holder
onERC721Received
IERC721Receiver (iface)
onERC721Received
IWhitelistRegistry (iface)
addToWhitelist
removeFromWhitelist
bulkAddToWhitelist
bulkremoveFromWhitelist
checkWhitelistStatus

# Unified Modeling Language (UML)



## Functions signature

```
Sighash   |   Function Signature
========================
e43252d7  =>  addToWhitelist(address)
8ab1d681  =>  removeFromWhitelist(address)
6c79af10  =>  bulkAddToWhitelist(address[])
f4b45945  =>  bulkremoveFromWhitelist(address[])
178d4de5  =>  checkWhitelistStatus(address)
150b7a02  =>  onERC721Received(address,address,uint256,bytes)
01ffc9a7  =>  supportsInterface(bytes4)
00fdd58e  =>  balanceOf(address,uint256)
4e1273f4  =>  balanceOfBatch(address[],uint256[])
a22cb465  =>  setApprovalForAll(address,bool)
e985e9c5  =>  isApprovedForAll(address,address)
f242432a  =>  safeTransferFrom(address,address,uint256,uint256,bytes)
2eb2c2d6  =>  safeBatchTransferFrom(address,address,uint256[],uint256[],bytes)
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)
081812fc  =>  getApproved(uint256)
f23a6e61  =>  onERC1155Received(address,address,uint256,uint256,bytes)
bc197c81  =>  onERC1155BatchReceived(address,address,uint256[],uint256[],bytes)
ad04a8d1  =>  current(Counter)
e2bee435  =>  increment(Counter)
854ec98e  =>  decrement(Counter)
440d212a  =>  reset(Counter)
6d5433e6  =>  max(uint256,uint256)
7ae2b5c7  =>  min(uint256,uint256)
2b7423ab  =>  average(uint256,uint256)
9cb35327  =>  ceilDiv(uint256,uint256)
aa9a0912  =>  mulDiv(uint256,uint256,uint256)
1db78456  =>  mulDiv(uint256,uint256,uint256,Rounding)
677342ce  =>  sqrt(uint256)
a902bc5e  =>  sqrt(uint256,Rounding)
5456bf13  =>  log2(uint256)
2ee6af53  =>  log2(uint256,Rounding)
ebdae5f9  =>  log10(uint256)
f86799ff  =>  log10(uint256,Rounding)
36cb4c48  =>  log256(uint256)
2910b3a1  =>  log256(uint256,Rounding)
884557bf  =>  tryAdd(uint256,uint256)
a29962b1  =>  trySub(uint256,uint256)
6281efa4  =>  tryMul(uint256,uint256)
736ecb18  =>  tryDiv(uint256,uint256)
38dc0867  =>  tryMod(uint256,uint256)
771602f7  =>  add(uint256,uint256)
b67d77c5  =>  sub(uint256,uint256)
c8a4ac9c  =>  mul(uint256,uint256)
a391c15b  =>  div(uint256,uint256)
f43f523a  =>  mod(uint256,uint256)
e31bdc0a  =>  sub(uint256,uint256,string)
```

```
b745d336  =>  div(uint256,uint256,string)
71af23e8  =>  mod(uint256,uint256,string)
119df25f  =>  _msgSender()
8b49d47e  =>  _msgData()
8da5cb5b  =>  owner()
53a72975  =>  _checkOwner()
715018a6  =>  renounceOwnership()
f2fde38b  =>  transferOwnership(address)
d29d44ee  =>  _transferOwnership(address)
d42f0c15  =>  createAuction(Asset[],uint256,uint256,uint256,uint256,uint256,bool)
659dd2b4  =>  createBid(uint256)
0070c537  =>  increaseBid(uint256)
96b5a755  =>  cancelAuction(uint256)
d9ec787d  =>  changeReservePrice(uint256,uint256)
2e993611  =>  settleAuction(uint256)
ebc0e168  =>  _transferAuctionAssets(uint256,address,address)
c2b5e50b  =>  _transferAssets(Asset[],address,address)
ad25533b  =>  _calculateProtocolFees(uint256)
a5d11cf9  =>  _calculatePenaltyFees(uint256)
69f665bb  =>  _extendAuction(uint256)
78bd7935  =>  getAuction(uint256)
8f288644  =>  getHighestBid(uint256)
adc832e5  =>  getAuctionAssets(uint256)
1df47f80  =>  updateProtocolFeeRecipient(address)
4256dd78  =>  updateProtocolFee(uint256)
f1d3831e  =>  updatePenaltyFee(uint256)
188713fe  =>  toggleAllowListings(bool)
fffda358  =>  toggleBeta(bool)
56f0fe67  =>  updateMaxLotSize(uint256)
faaf3415  =>  updateWhitelistRegistry(address)
5c5e20b6  =>  revertAuction(uint256)
```

# Automatic general report

```
Files Description Table

|  File Name  |  SHA-1 Hash  |
|-------------|--------------|
| /Users/macbook/Desktop/smart contracts/AuctionHouse.sol |
f398c727b724ce82526a51153680a4e56127c0db |

 Contracts Description Table
|  Contract  |         Type        |       Bases      |                  |
|
|:----------:|:-------------------:|:----------------:|:----------------:|:--------
-------:|
|     └      |  **Function Name**  |  **Visibility**  |  **Mutability**  |
**Modifiers**  |
||||||
| **IWhitelistRegistry** | Interface |  |||
| └ | addToWhitelist | External ❗ | 🔴  |NO❗ |
| └ | removeFromWhitelist | External ❗ | 🔴  |NO❗ |
| └ | bulkAddToWhitelist | External ❗ | 🔴  |NO❗ |
| └ | bulkremoveFromWhitelist | External ❗ | 🔴  |NO❗ |
| └ | checkWhitelistStatus | External ❗ |  |NO❗ |
||||||
| **IERC721Receiver** | Interface |  |||
| └ | onERC721Received | External ❗ | 🔴  |NO❗ |
||||||
| **ERC721Holder** | Implementation | IERC721Receiver |||
| └ | onERC721Received | Public ❗ | 🔴  |NO❗ |
||||||
| **IERC165** | Interface |  |||
| └ | supportsInterface | External ❗ |  |NO❗ |
||||||
| **IERC1155** | Interface | IERC165 |||
| └ | balanceOf | External ❗ |  |NO❗ |
| └ | balanceOfBatch | External ❗ |  |NO❗ |
| └ | setApprovalForAll | External ❗ | 🔴  |NO❗ |
| └ | isApprovedForAll | External ❗ |  |NO❗ |
| └ | safeTransferFrom | External ❗ | 🔴  |NO❗ |
| └ | safeBatchTransferFrom | 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❗ |
||||||
| **ERC165** | Implementation | IERC165 |||
| └ | supportsInterface | Public ❗ |  |NO❗ |
```

```
||||||
| **IERC1155Receiver** | Interface | IERC165 |||
| └ | onERC1155Received | External ❗ | 🛑  |NO❗ |
| └ | onERC1155BatchReceived | External ❗ | 🛑  |NO❗ |
||||||
| **ERC1155Receiver** | Implementation | ERC165, IERC1155Receiver |||
| └ | supportsInterface | Public ❗ |   |NO❗ |
||||||
| **ERC1155Holder** | Implementation | ERC1155Receiver |||
| └ | onERC1155Received | Public ❗ | 🛑  |NO❗ |
| └ | onERC1155BatchReceived | Public ❗ | 🛑  |NO❗ |
||||||
| **Counters** | Library |  |||
| └ | current | Internal 🔒 |   | |
| └ | increment | Internal 🔒 | 🛑  | |
| └ | decrement | Internal 🔒 | 🛑  | |
| └ | reset | Internal 🔒 | 🛑  | |
||||||
| **Math** | Library |  |||
| └ | max | Internal 🔒 |   | |
| └ | min | Internal 🔒 |   | |
| └ | average | Internal 🔒 |   | |
| └ | ceilDiv | Internal 🔒 |   | |
| └ | mulDiv | Internal 🔒 |   | |
| └ | mulDiv | Internal 🔒 |   | |
| └ | sqrt | Internal 🔒 |   | |
| └ | sqrt | Internal 🔒 |   | |
| └ | log2 | Internal 🔒 |   | |
| └ | log2 | Internal 🔒 |   | |
| └ | log10 | Internal 🔒 |   | |
| └ | log10 | Internal 🔒 |   | |
| └ | log256 | Internal 🔒 |   | |
| └ | log256 | Internal 🔒 |   | |
||||||
| **SafeMath** | Library |  |||
| └ | tryAdd | Internal 🔒 |   | |
| └ | trySub | Internal 🔒 |   | |
| └ | tryMul | Internal 🔒 |   | |
| └ | tryDiv | Internal 🔒 |   | |
| └ | tryMod | Internal 🔒 |   | |
| └ | add | Internal 🔒 |   | |
| └ | sub | Internal 🔒 |   | |
| └ | mul | Internal 🔒 |   | |
| └ | div | Internal 🔒 |   | |
| └ | mod | Internal 🔒 |   | |
| └ | sub | Internal 🔒 |   | |
| └ | div | Internal 🔒 |   | |
| └ | mod | Internal 🔒 |   | |
||||||
| **ReentrancyGuard** | Implementation |  |||
| └ | <Constructor> | Public ❗ | 🛑  |NO❗ |
| **Context** | Implementation |  |||
| └ | _msgSender | Internal 🔒 |   | |
| └ | _msgData | Internal 🔒 |   | |
| **Ownable** | Implementation | Context |||
```

| └ | <Constructor> | Public ❗ | 🛑 |NO❗ |
| └ | owner | Public ❗ | |NO❗ |
| └ | _checkOwner | Internal 🔒 | | |
| └ | renounceOwnership | Public ❗ | 🛑 | onlyOwner |
| └ | transferOwnership | Public ❗ | 🛑 | onlyOwner |
| └ | _transferOwnership | Internal 🔒 | 🛑 | |
| **AuctionHouse** | Implementation | Ownable, ReentrancyGuard, ERC721Holder, ERC1155Holder |||
| └ | <Constructor> | Public ❗ | 🛑 |NO❗ |
| └ | createAuction | Public ❗ | 🛑 | canList |
| └ | createBid | Public ❗ | 💵 | nonReentrant openAuction notSeller nonExpiredAuction |
| └ | increaseBid | Public ❗ | 💵 |NO❗ |
| └ | cancelAuction | Public ❗ | 💵 | nonReentrant nonExpiredAuction openAuction onlySeller |
| └ | changeReservePrice | Public ❗ | 🛑 | onlySeller openAuction |
| └ | settleAuction | Public ❗ | 💵 | nonReentrant openAuction expiredAuction |
| └ | _transferAuctionAssets | Internal 🔒 | 🛑 | |
| └ | _transferAssets | Internal 🔒 | 🛑 | |
| └ | _calculateProtocolFees | Internal 🔒 | | |
| └ | _calculatePenaltyFees | Internal 🔒 | | |
| └ | _extendAuction | Internal 🔒 | 🛑 | |
| └ | getAuction | Public ❗ | |NO❗ |
| └ | getHighestBid | Public ❗ | |NO❗ |
| └ | getAuctionAssets | Public ❗ | |NO❗ |
| └ | updateProtocolFeeRecipient | External ❗ | 🛑 | onlyOwner |
| └ | updateProtocolFee | External ❗ | 🛑 | onlyOwner |
| └ | updatePenaltyFee | External ❗ | 🛑 | onlyOwner |
| └ | toggleAllowListings | External ❗ | 🛑 | onlyOwner |
| └ | toggleBeta | External ❗ | 🛑 | onlyOwner |
| └ | updateMaxLotSize | External ❗ | 🛑 | onlyOwner |
| └ | updateWhitelistRegistry | External ❗ | 🛑 | onlyOwner |
| └ | revertAuction | External ❗ | 🛑 | onlyOwner nonExpiredAuction openAuction |

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.