



Multitoken Marketplace, Contract, Code Review and Security Analysis Report

Customer: OVR Platform
Prepared on: 5th December 2022
Platform: Polygon
Language: Solidity

rdauditors.com

Table of Contents

Disclaimer	2
Document	3
Introduction	7
Project Scope	8
Executive Summary	9
Code Quality	9
Documentation	11
Use of Dependencies	11
AS-IS Overview	12
Code Flow Diagram - Multitoken Marketplace	21
File: Assets3D	22
File: Auctions.sol	23
Code Flow Diagram - Slither Results Log	33
File: Auctions	54
Severity Definitions	95
Audit Findings	96
Conclusion	97
Note For Contract Users	97
Our Methodology	100
Disclaimers	102

Disclaimer

This document may contain confidential information about its systems and intellectual property of the customer as well as information about potential vulnerabilities and methods of their exploitation.

The report containing confidential information can be used internally by the customer or it can be disclosed publicly after all vulnerabilities are fixed - upon the decision of the customer.

Document

Name	Smart Contract Code Review and Security Analysis Report of OVR Multitoken Marketplace
Platform	Polygon/ Solidity
File 1	Allowed.sol
MD5 hash	f8494c82d08e611cc6985192a45af5c7
SHA256 hash	5abb457438fd82e8780b9e304862a5c4acfa27074a8b8b95dcca69b848ae99cb
File 2	Assets3DTEST.sol
MD5 hash	f1662687d9778a8de8a389f384a113f8
SHA256 hash	0bfff6cc90e7d48bb9d2608f2ef4b69cba3cd8a5b4c00040eb2a692e260f0cf
File 3	Auctions.sol
MD5 hash	60c724fbdc289617c71035ceb739b5fa
SHA256 hash	2eb910964f1e9bfff1d273cde273fa83a49fbddc2e54f0ec05916b8507df132
File 4	HeapOffers.sol

MD5 hash	1ee1d56da25185ede60557a8e0b9fe16
SHA256 hash	42d109e6f4de4b78b947c208419f2ea6936f70b781ef0a86c068f686c0585c39
File 5	HeapSales.sol
MD5 hash	430a1efa7e2a5e39f722a7a296385a2f
SHA256 hash	5b3c9dfc05308051cdc0daedb09b0e158def924cb4715b7d3c6fc909362f94ea
File 6	Offers.sol
MD5 hash	f1b4bb53111296e5d85d0cf8c39ae88b
SHA256 hash	7cbea633fc432f68c082895cd51f6dc17a0cdf95db3507228cffaff4410ce c875
File 7	OVRLandTEST.sol
MD5 hash	7b9892c59fa50d8868da8338c258004b
SHA256 hash	13df9f08db7421823aedd2c590828bc75849af88ce1cca81b0c128eaaf658326
File 8	Root.sol
MD5 hash	3b09e0fcf2ee0da45be2ef66edca8eef

SHA256 hash	a3457529d7fa1952c118e718b408da8fb908589e743594af78cef5d5ca046adb
File 9	Sales.sol
MD5 hash	1b973e5d8e7fc19d6d65ed75255a0683
SHA256 hash	aef6e9a0346070ba42291c9c6c8bfa6e2cd80e19e7618ed98aba856cc d92c5cc
File 10	Store.sol
MD5 hash	176b699020a8347217f9556c55c3b51e
SHA256 hash	9fd14dbe2a11535a639dfca04f8f6a86bed9f259befcc8a4f3a3eeb659eba76b
File 11	Token.sol
MD5 hash	82690fc716205f074c6204d141bef6e
SHA256 hash	da76a099976c1ccb338291a2e2ef8555ecccb9354eee4b36b3b53b0856a32193
File 12	TokenUtils.sol
MD5 hash	401d4c904bd8126a1e4e248455825ac9

SHA256 hash	e44a13a4c4e2f96b6c366983a384551a781302146d89f1c5dbd2a7b8037befa3
Date	5/12/2022

Introduction

RD Auditors (Consultant) were contracted by OVR Platform (Customer) to conduct a Smart Contract Code Review and Security Analysis. This report represents the findings of the security assessment of the customer's smart contract and its code review conducted between 28th November - 5th December 2022.

This contract consists of twelve files.

Project Scope

The scope of the project is a smart contract. We have scanned this smart contract for commonly known and more specific vulnerabilities, below are those considered (the full list includes but is not limited to):

- Reentrancy
- Timestamp Dependence
- Gas Limit and Loops
- DoS with (Unexpected) Throw
- DoS with Block Gas Limit
- Transaction-Ordering Dependence
- Byte array vulnerabilities
- Style guide violation
- Transfer forwards all gas
- ERC20 API violation
- Malicious libraries
- Compiler version not fixed
- Unchecked external call - Unchecked math
- Unsafe type inference
- Implicit visibility level

Executive Summary

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

You are Here



Automated checks are with smartDec, Mythril, Slither and remix IDE. All issues were performed by our team, which included the analysis of code functionality, the 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 AS-IS section and all issues found are located in the audit overview section.

We found the following;

Total Issues	0
■ Critical	0
■ High	0
■ Medium	0
■ Low	0
■ Very Low	0

Code Quality

The libraries within this smart contract are part of a logical algorithm. A library is a different type of smart contract that contains reusable code. Once deployed on the blockchain (only once), it is assigned to a specific address and its properties/methods can be reused many times by other contracts.

The OVR Platform team has provided scenario and unit test scripts, which helped to determine the integrity of the code in an automated way.

Overall, the code is well commented. Commenting can provide rich documentation for functions, return variables and more. Use of the Ethereum Natural Language Specification Format (NatSpec) for commenting is recommended.

Documentation

We were given the Multitoken MarketPlace code as a Github link:

<https://github.com/OVR-Platform/marketplace-multi-token-contracts>

The hash of that file is mentioned in the table. As mentioned above, it's well commented smart contract code, so anyone can quickly understand the programming flow as well as complex code logic.

Comments are very helpful in understanding the overall architecture of the protocol. It also provides a clear overview of the system components, including helpful details, like the lifetime of the background script.

Use of Dependencies

As per our observation, the libraries are used in this smart contract infrastructure. Those were based on well known industry standard open source projects and even core code blocks that are written well and systematically.

AS-IS Overview

MarketPlace multi_token.sol

File And Function Level Report

Contract: Allowed

Import: AccessControlUpgradeable, Structs, Enum

Inherit: AccessControlUpgradeable

Observation: Passed

Test Report: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	initialize_Allo wed	internal	Passed	All Passed	No Issue	Passed
2	OnlyAllowedA ddress	internal	Passed	All Passed	No Issue	Passed
3	addAllowedA ddress	OnlyRole	Passed	All Passed	No Issue	Passed

Contract: Assets3DTEST
Import: ERC1155, AccessControl, ERC1155Burnable, ERC1155Supply
Inherit: ERC1155, AccessControl, ERC1155Burnable, ERC1155Supply
Observation: Passed
Test Report: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	SetURI	onlyRole	Passed	All Passed	No Issue	Passed
2	totalSupply	read	Passed	All Passed	No Issue	Passed
3	name	read	Passed	All Passed	No Issue	Passed
4	symbol	read	Passed	All Passed	No Issue	Passed
5	uri	read	Passed	All Passed	No Issue	Passed
6	burn	onlyRole	Passed	All Passed	No Issue	Passed
7	burnBatch	onlyRole	Passed	All Passed	No Issue	Passed
8	mint	onlyRole	Passed	All Passed	No Issue	Passed
9	mintBatch	onlyRole	Passed	All Passed	No Issue	Passed
10	mintWithURI	onlyRole	Passed	All Passed	No Issue	Passed
11	mintBatchWithURI	onlyRole	Passed	All Passed	No Issue	Passed
12	addAdmin	write	Passed	All Passed	No Issue	Passed
13	removeAdmin	write	Passed	All Passed	No Issue	Passed
14	_beforeTokenTransfer	internal	Passed	All Passed	No Issue	Passed
15	SupportsInterface	read	Passed	All Passed	No Issue	Passed

Contract: Auctions

Inherit: Store, TokenUtils

Import: TokenUtils, Store, AuctionsUtils, Structs, Events, Utils

Observation: Passed

Test Report: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	CreateAuction	external	Passed	All Passed	No Issue	Passed
2	isApprovedForAll	internal	Passed	All Passed	No Issue	Passed
3	bid	external	Passed	All Passed	No Issue	Passed
4	completeAuction	external	Passed	All Passed	No Issue	Passed
5	cancelAuction	external	Passed	All Passed	No Issue	Passed
6	getAuctionDetails	read	Passed	All Passed	No Issue	Passed

Contract: HeapOffers

Inherit: AccessControlUpgradeable

Import: AccessControlUpgradeable

Observation: Passed

Test Report: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	initHeap	internal	Passed	All Passed	No Issue	Passed
2	insertNode	onlyRole	Passed	All Passed	No Issue	Passed
3	deleteNode	onlyRole	Passed	All Passed	No Issue	Passed
4	getMaxNode	read	Passed	All Passed	No Issue	Passed
5	getNode	read	Passed	All Passed	No Issue	Passed
6	heapifyUp	internal	Passed	All Passed	No Issue	Passed
7	heapifyDown	internal	Passed	All Passed	No Issue	Passed
8	getBestAssets	read	Passed	All Passed	No Issue	Passed
9	addAdminRole	onlyRole	Passed	All Passed	No Issue	Passed

Contract: HeapSales

Inherit: AccessControlUpgradeable

Import AccessControlUpgradeable, IERC1155Upgradeable, IStore

Observation: Passed

Test Report: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	initHeap	internal	Passed	All Passed	No Issue	Passed
2	insertNode	onlyRole	Passed	All Passed	No Issue	Passed
3	deleteNode	onlyRole	Passed	All Passed	No Issue	Passed
4	UpdateNode	onlyRole	Passed	All Passed	No Issue	Passed
5	getMinNode	read	Passed	All Passed	No Issue	Passed
6	getNode	read	Passed	All Passed	No Issue	Passed
7	heapifyUp	internal	Passed	All Passed	No Issue	Passed

8	heapifyDown	internal	Passed	All Passed	No Issue	Passed
9	getBestAssets	read	Passed	All Passed	No Issue	Passed
10	addStore	onlyRole	Passed	All Passed	No Issue	Passed
11	addAdminRole	onlyRole	Passed	All Passed	No Issue	Passed

Contract: Offers

Inherit: TokenUtils, Store, Structs, Utils, Events, Enums, IHeap

Import StoreTokenUtils

Observation: Passed

Test Report: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	_offers_init	internal	Passed	All Passed	No Issue	Passed
2	viewOffersByAs set	read	Passed	All Passed	No Issue	Passed
3	buyOrderExist	read	Passed	All Passed	No Issue	Passed
4	CreateOffer	external	Passed	All Passed	No Issue	Passed
5	deleteOffer	external	Passed	All Passed	No Issue	Passed
6	fulfillOffer	external	Passed	All Passed	No Issue	Passed

Contract: OVRLandTEST

Inherit: ERC721

Import ERC721, ERC721Enumerable, ERC721URIStorage,
ERC721Burnable, AccessControl

Observation: Passed

Test Report: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	addURIEditor	onlyRole	Passed	All Passed	No Issue	Passed
2	removeURIEditor	onlyRole	Passed	All Passed	No Issue	Passed
3	addMinter	onlyRole	Passed	All Passed	No Issue	Passed
4	removeMinter	onlyRole	Passed	All Passed	No Issue	Passed
5	addBurner	onlyRole	Passed	All Passed	No Issue	Passed
6	removeBurner	onlyRole	Passed	All Passed	No Issue	Passed
7	addAdminRole	onlyRole	Passed	All Passed	No Issue	Passed
8	removeAdminRole	onlyRole	Passed	All Passed	No Issue	Passed
9	SafeMint	onlyRole	Passed	All Passed	No Issue	Passed
10	SetOVRLandURI	onlyRole	Passed	All Passed	No Issue	Passed
11	burn	onlyRole	Passed	All Passed	No Issue	Passed
12	batchBurn	onlyRole	Passed	All Passed	No Issue	Passed
13	mint	onlyRole	Passed	All Passed	No Issue	Passed
14	batchMintLands	onlyRole	Passed	All Passed	No Issue	Passed
15	batchMintLands WithURI	onlyRole	Passed	All Passed	No Issue	Passed
16	batchSetOVRLandURI	onlyRole	Passed	All Passed	No Issue	Passed
17	_beforeTokenTransfer	internal	Passed	All Passed	No Issue	Passed
18	_burn	internal	Passed	All Passed	No Issue	Passed
19	tokenURI	read	Passed	All Passed	No Issue	Passed
20	SupportsInterface	read	Passed	All Passed	No Issue	Passed

Contract: Root
Inherit: UUPSUpgradeable, Sales, Offers, Auctions
Import UUPSUpgradeable, Sales, Offers, Auctions
Observation: Passed
Test Report: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	initialize	external	Passed	All Passed	No Issue	Passed
2	Pause	external	Passed	All Passed	No Issue	Passed
3	UnPause	external	Passed	All Passed	No Issue	Passed
4	_authorizeUpgr ade	internal	Passed	All Passed	No Issue	Passed

Contract: Sales
Inherit: Store, TokenUtils
Import IERC721Upgradeable, IERC1155Upgradeable, TokenUtils,
Store, IHeap, Events, SalesLibrary, Structs, Utils, Enums
Observation: Passed
Test Report: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	_sales_init	internal	Passed	All Passed	No Issue	Passed
2	viewSalesB yAsset	read	Passed	All Passed	No Issue	Passed

3	createSale	external	Passed	All Passed	No Issue	Passed
4	cancelSale	external	Passed	All Passed	No Issue	Passed
5	_cancelSale	internal	Passed	All Passed	No Issue	Passed
6	buy	external	Passed	All Passed	No Issue	Passed

Contract: Store

Inherit: Allowed, ReentrancyGuardUpgradeable,
PausableUpgradeable

Import ReentrancyGuardUpgradeable, PausableUpgradeable,
Allowed, Structs, Enums

Observation: Passed

Test Report: Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	_store_Init	internal	Passed	All Passed	No Issue	Passed
2	assetType	read	Passed	All Passed	No Issue	Passed

Contract: TokenUtils

Inherit: Initializable

Import IERC20Upgradeable, IERC1155Upgradeable,
IERC721Upgradeable, Initializable

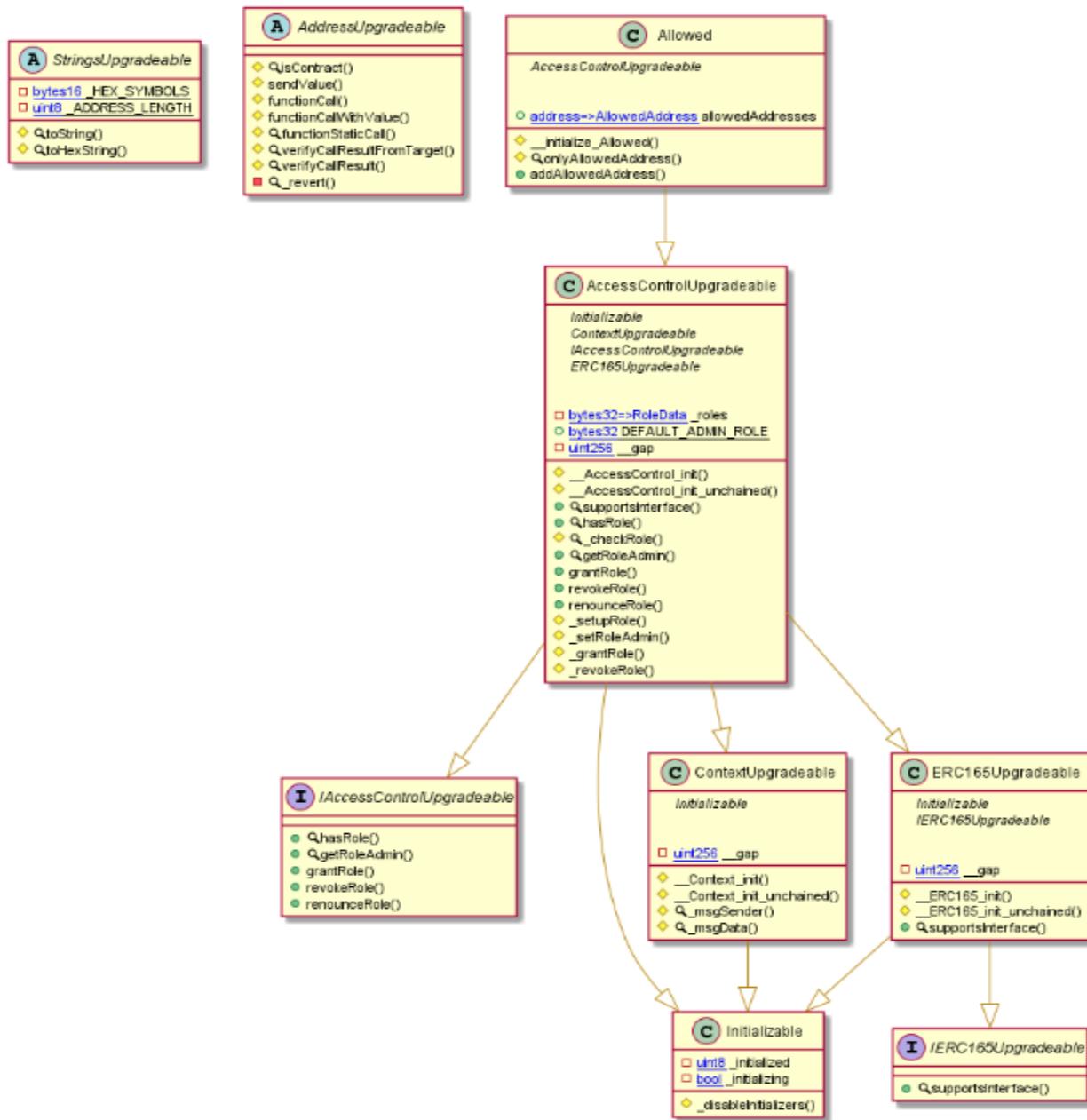
Observation: Passed

Test Report: Passed

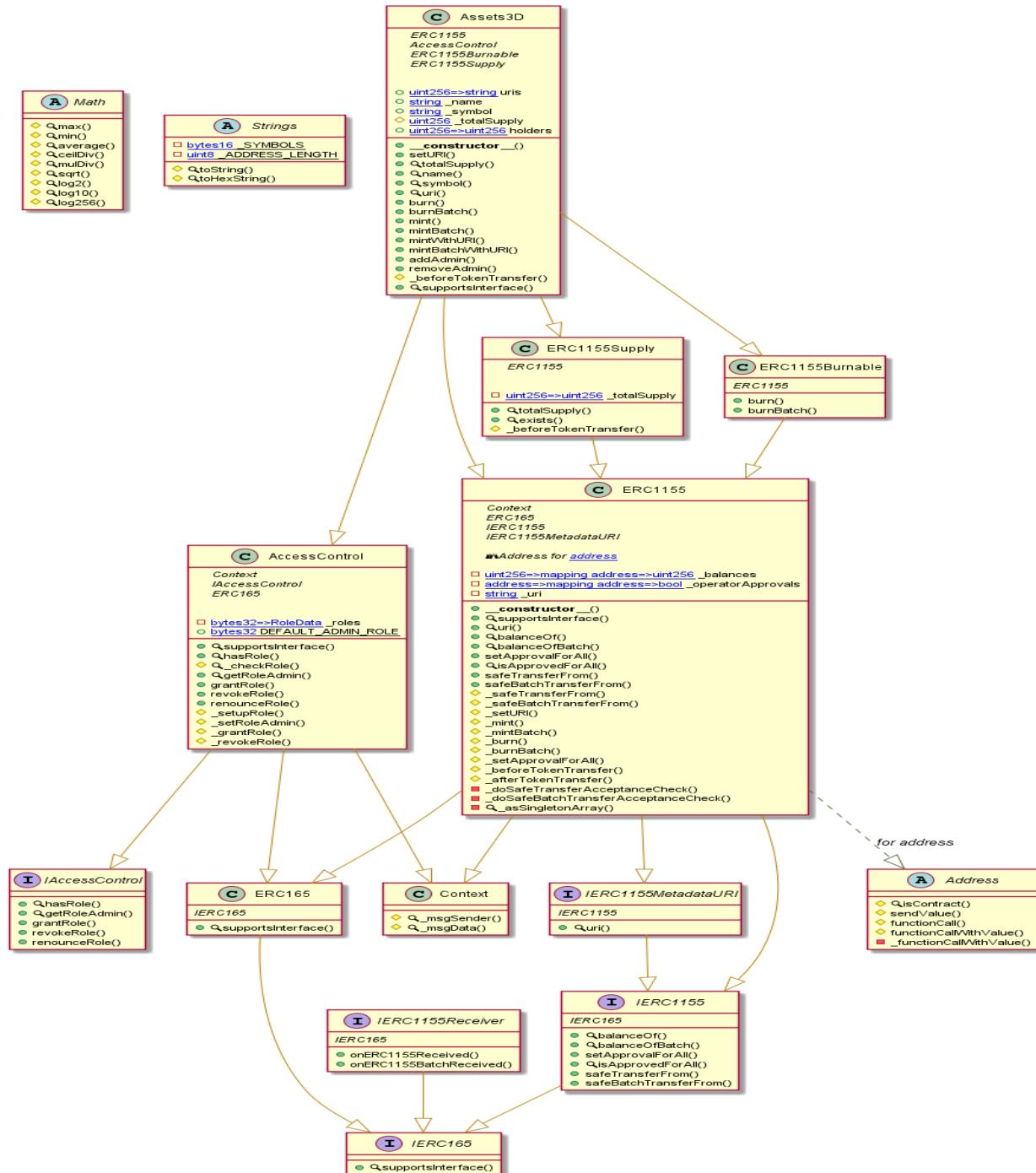
Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	_TokenUtils_init	internal	Passed	All Passed	No Issue	Passed
2	transferFromERC20	internal	Passed	All Passed	No Issue	Passed
3	transferERC20	internal	Passed	All Passed	No Issue	Passed
4	transferERC1155	internal	Passed	All Passed	No Issue	Passed
5	transferERC721	internal	Passed	All Passed	No Issue	Passed
6	transferERC1155Batch	internal	Passed	All Passed	No Issue	Passed
7	transferERC721Batch	internal	Passed	All Passed	No Issue	Passed

Code Flow Diagram - Multitoken Marketplace

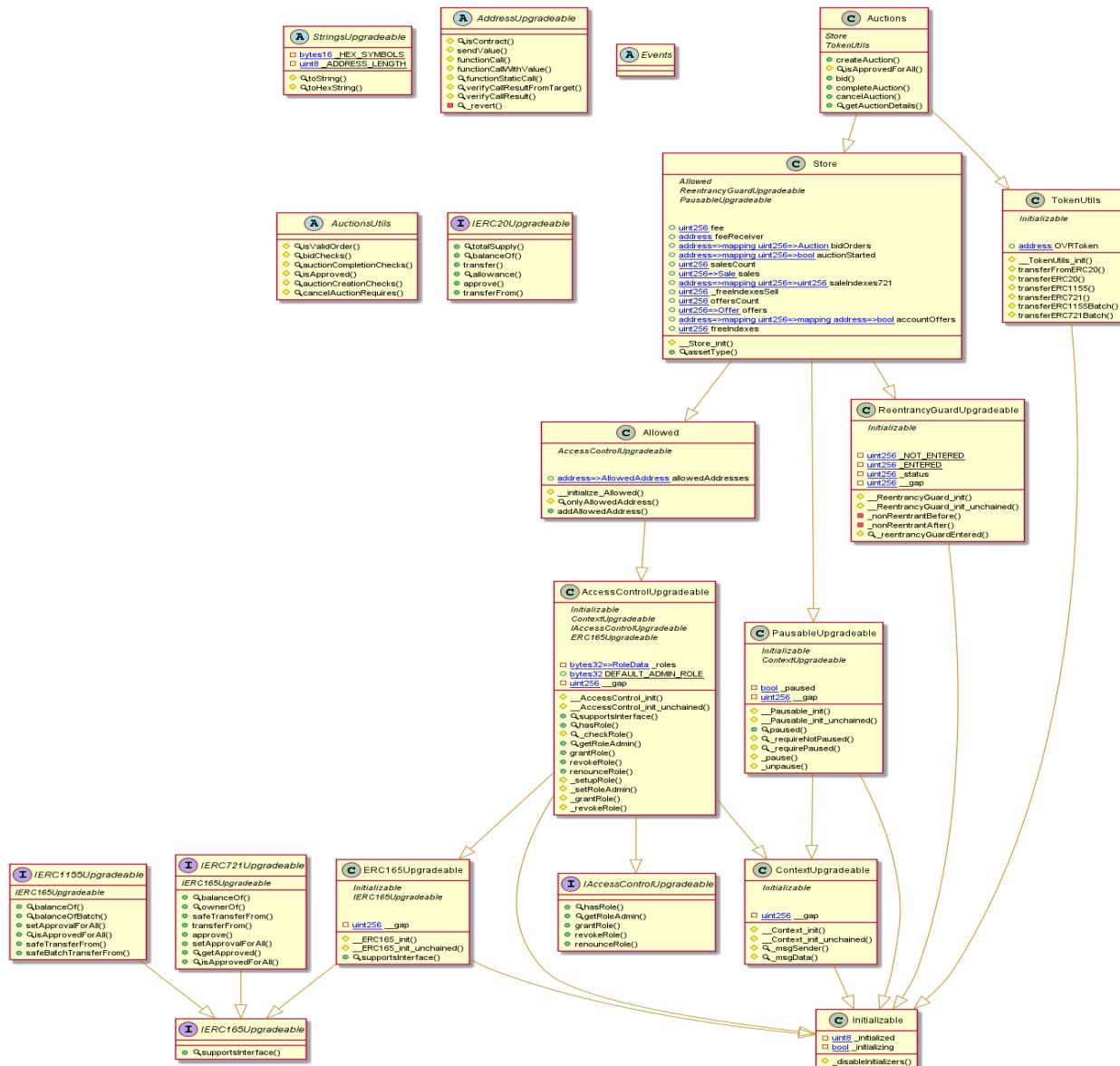
File: Allowed.sol



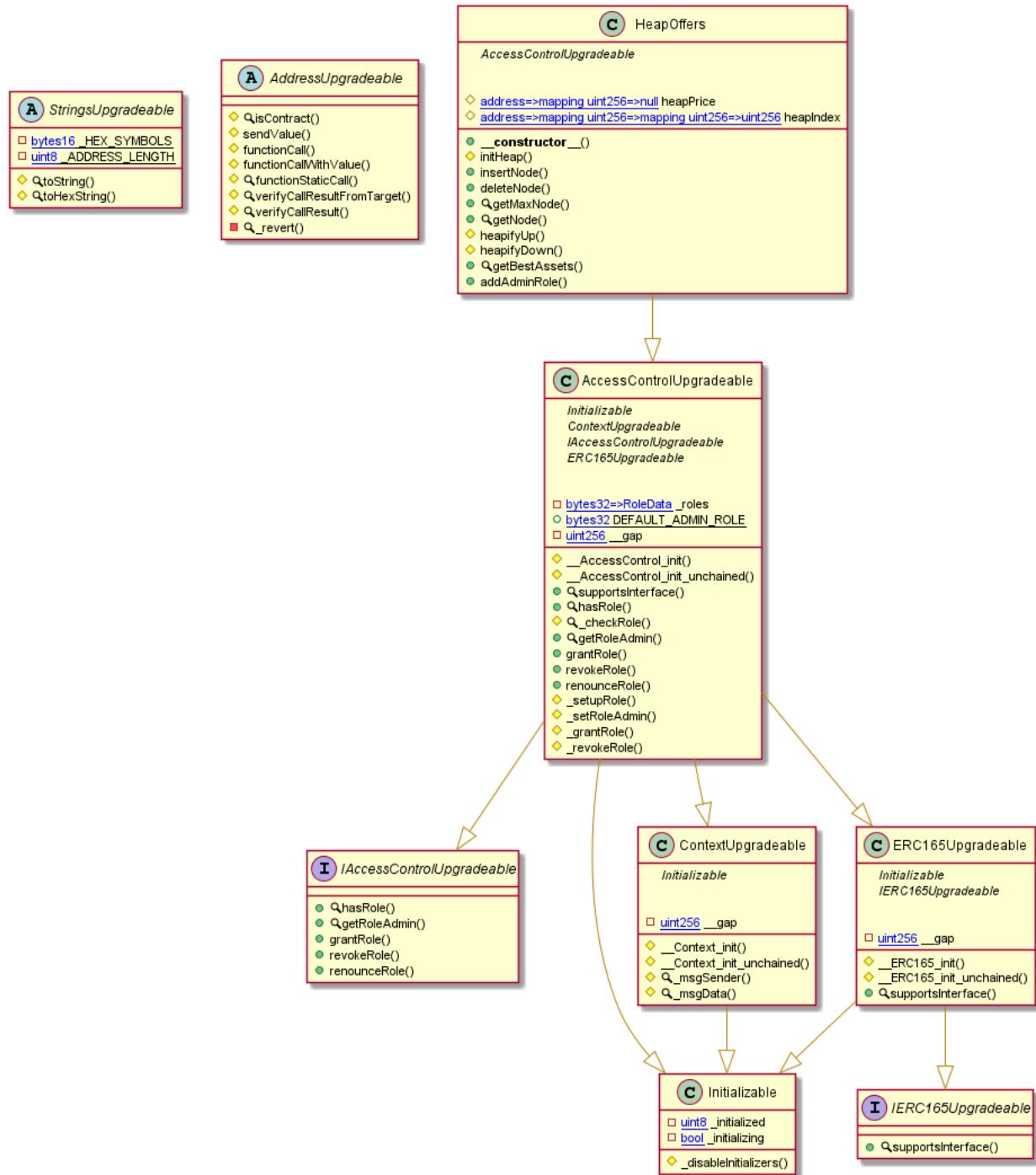
File: Assets3D



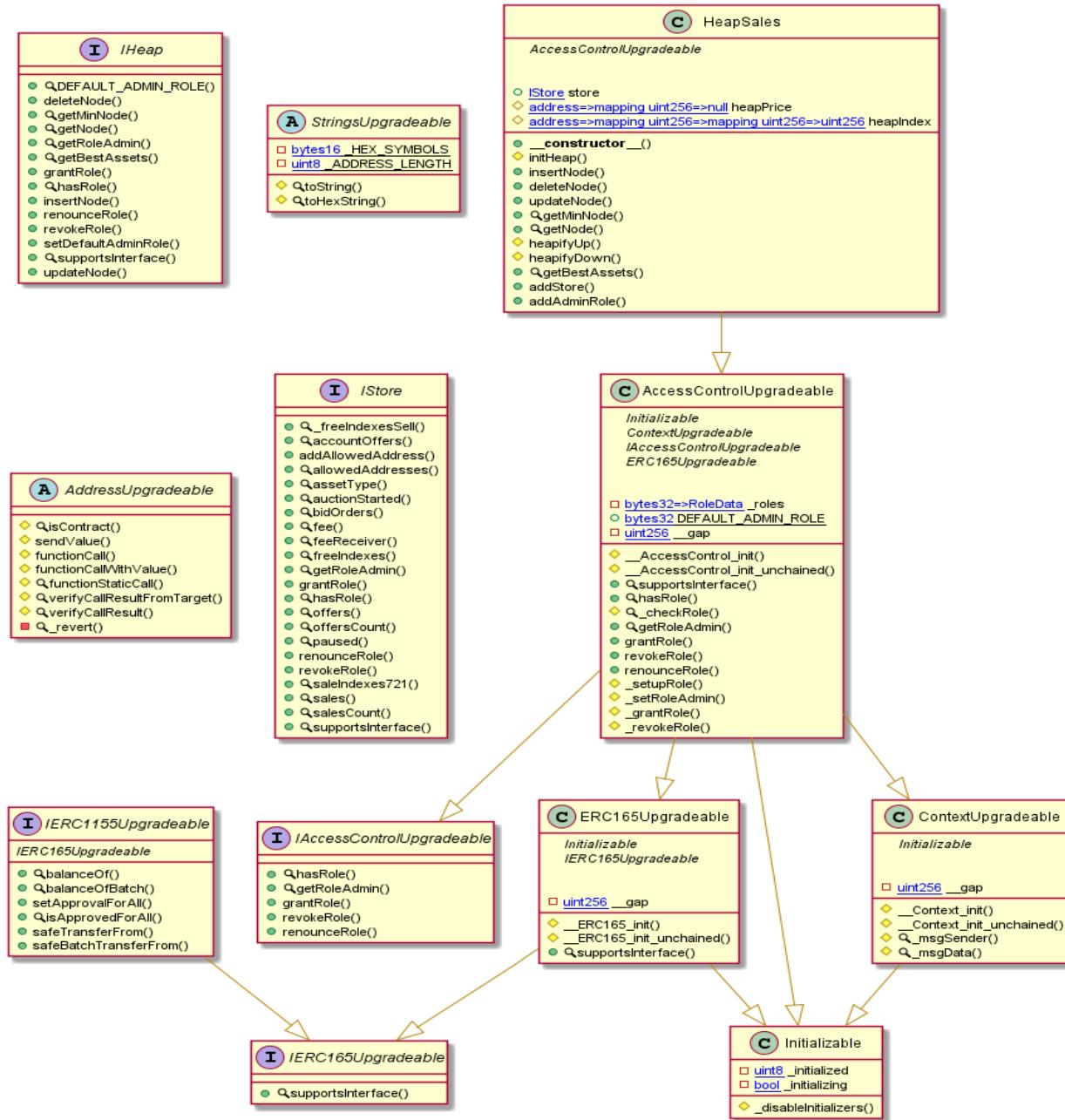
File: Auctions.sol



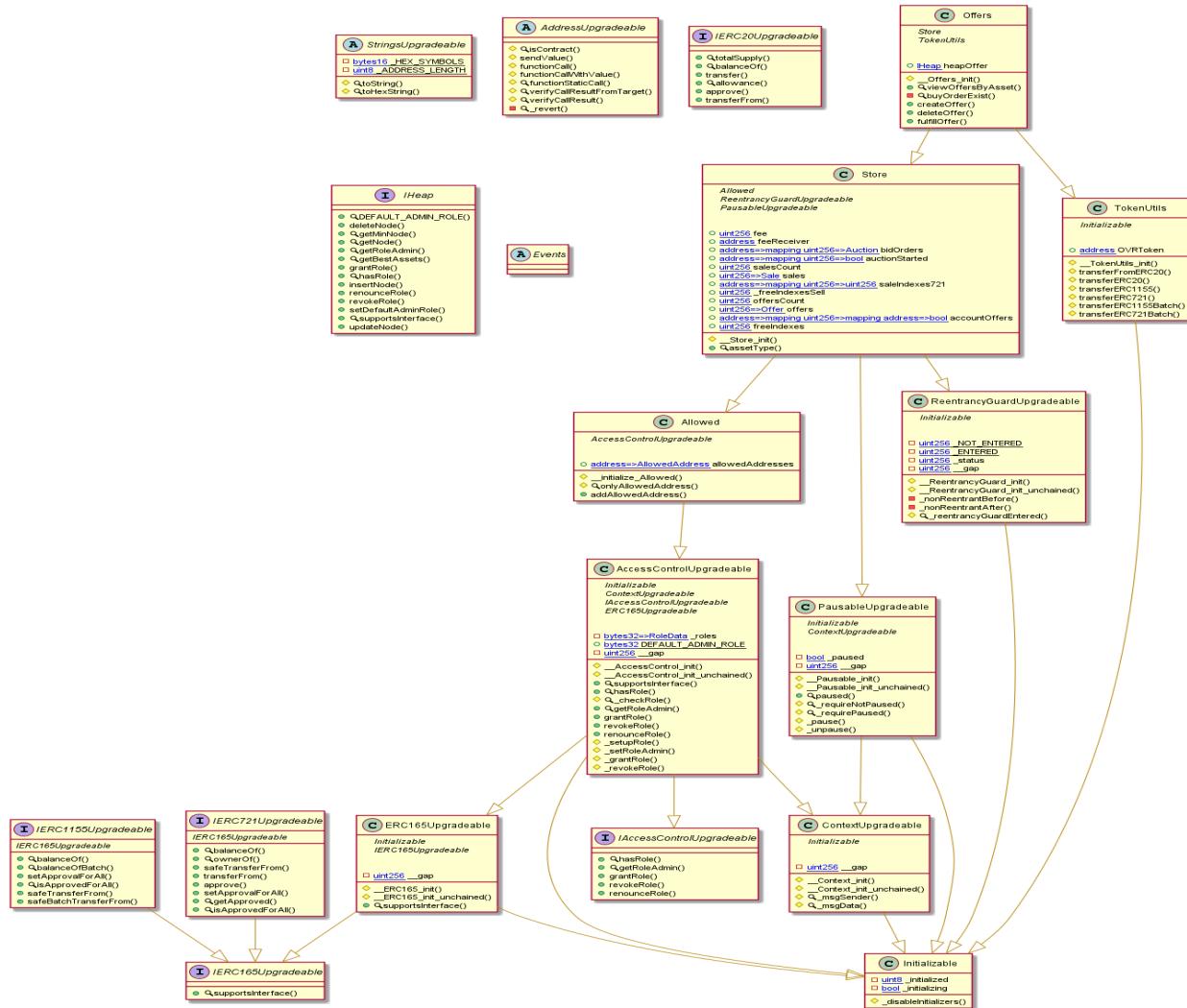
File: HeapOffers.sol



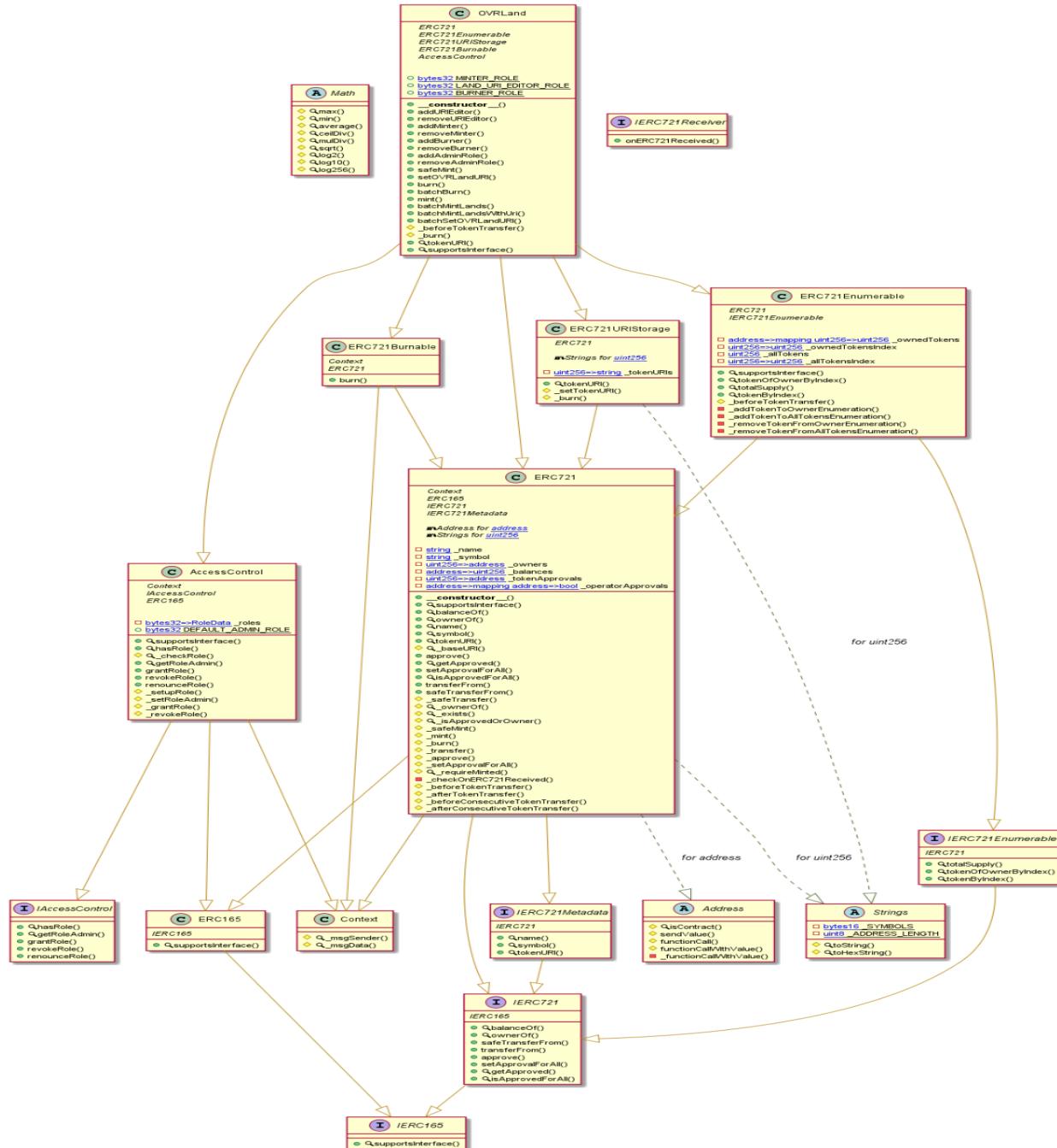
File: HeapSales



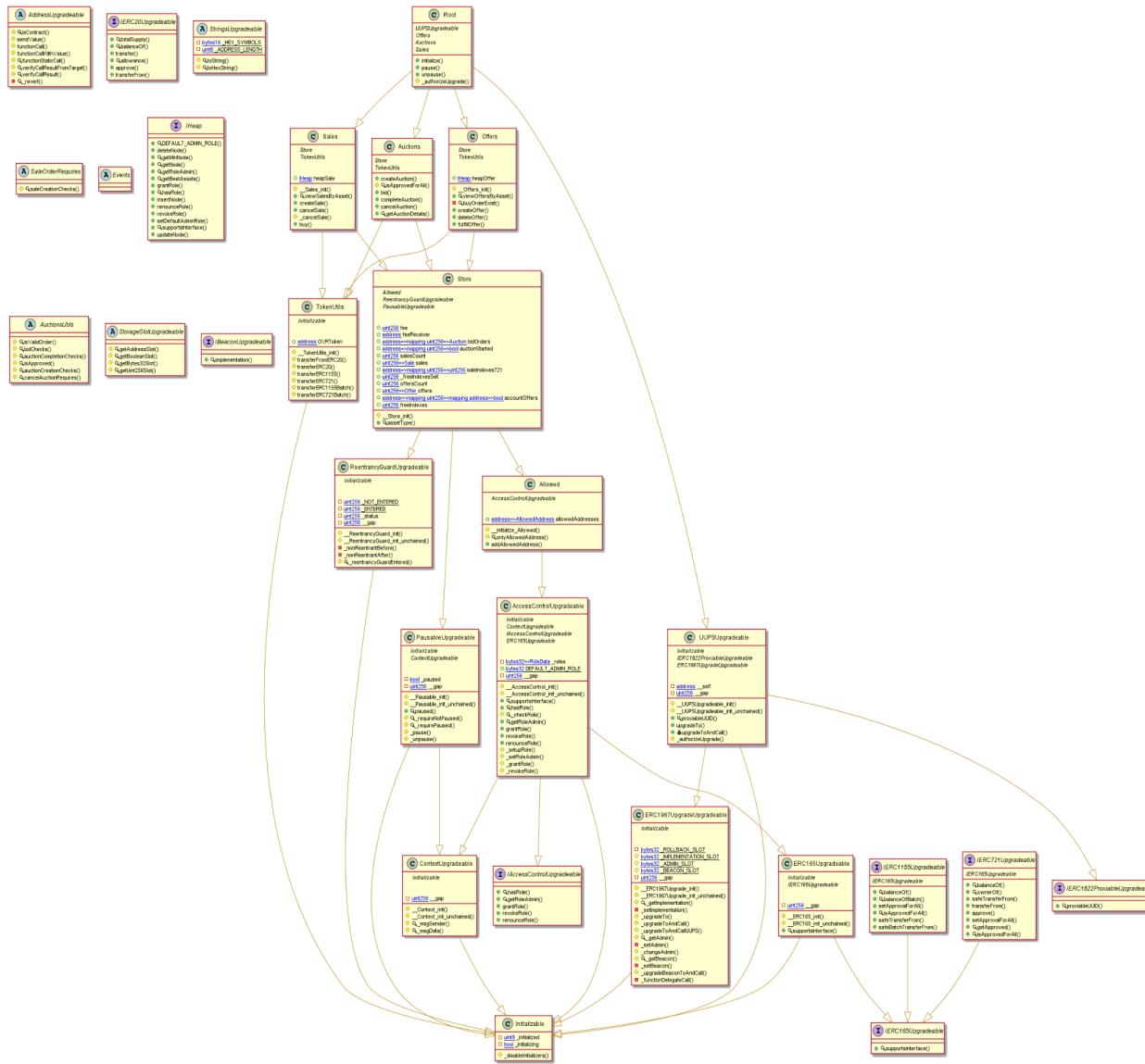
File: Offers



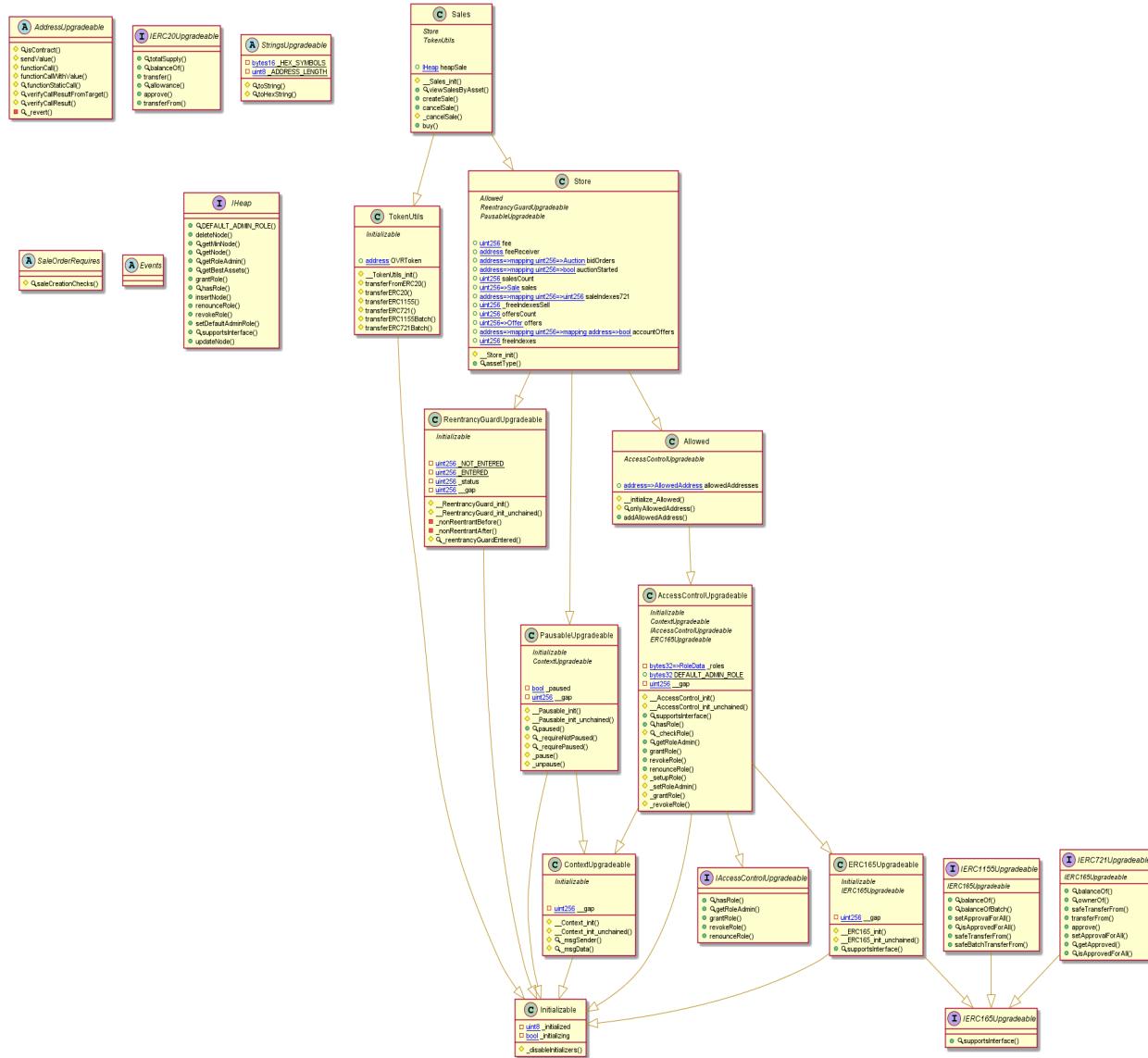
File: OVRLand



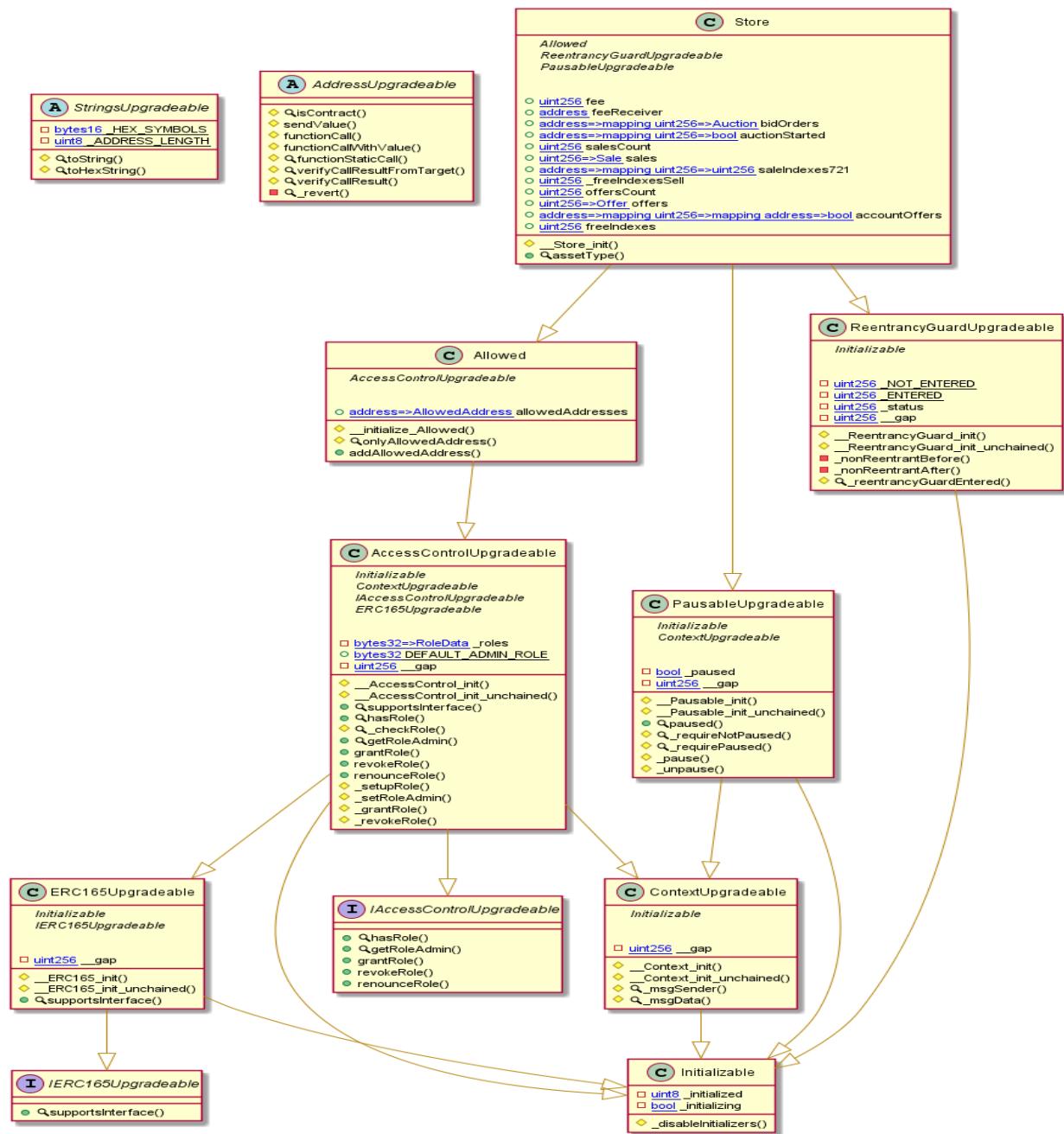
File: Root



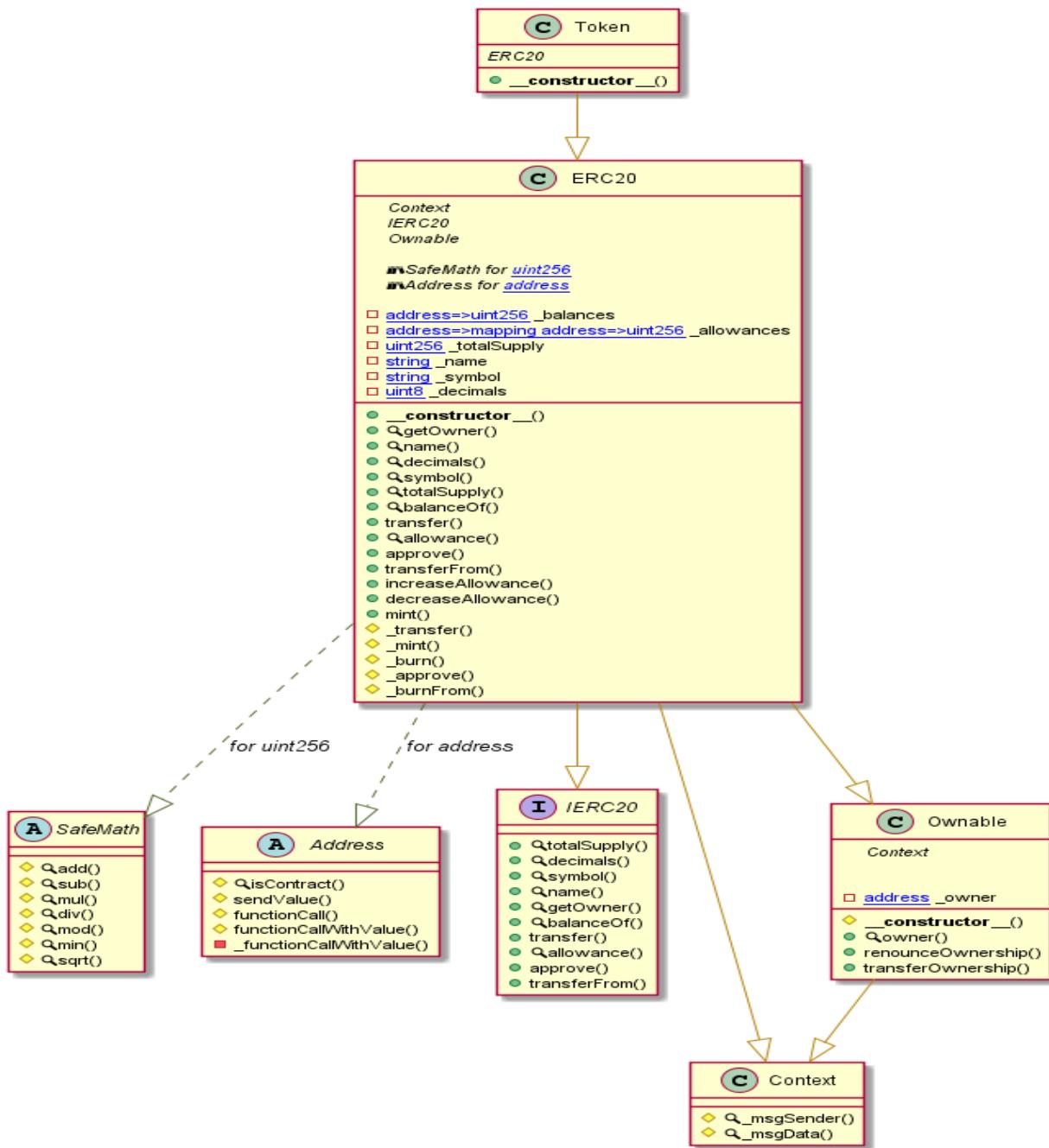
File: Sales



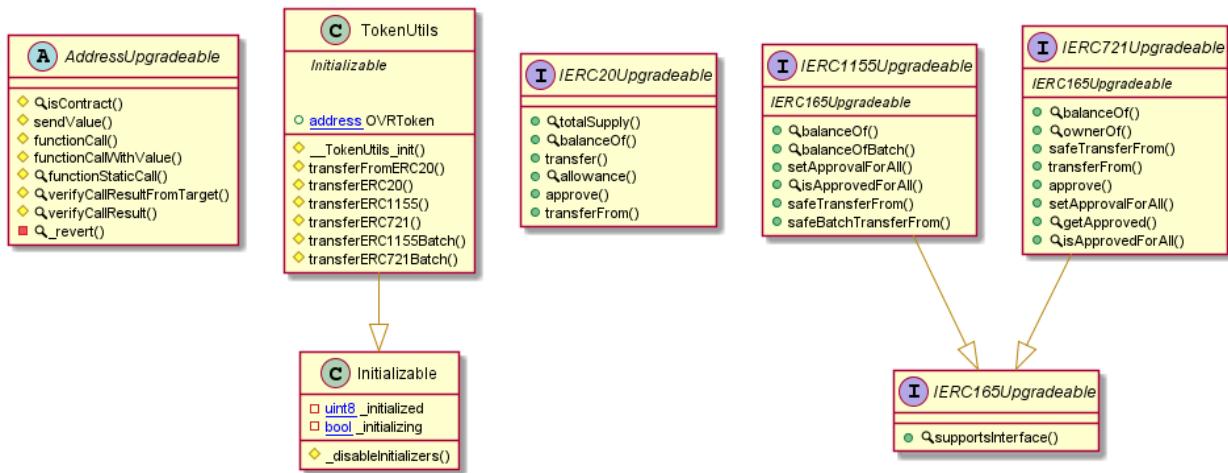
File: Store



File: Token



File: TokenUtils



Code Flow Diagram - Slither Results Log

File: Allowed.sol

```

INFO:Detectors:
AddressUpgradeable._revert(bytes,string) (Allowed.sol#200-209) uses assembly
    - INLINE ASM (Allowed.sol#202-205)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
AccessControlUpgradeable.__AccessControl_init() (Allowed.sol#313-314) is never used and should be removed
AccessControlUpgradeable.__AccessControl_init_unchained() (Allowed.sol#316-317) is never used and should be removed
AccessControlUpgradeable._setRoleAdmin(bytes32,bytes32) (Allowed.sol#381-385) is never used and should be removed
AccessControlUpgradeable._setupRole(bytes32,address) (Allowed.sol#377-379) is never used and should be removed
AddressUpgradeable._revert(bytes,string) (Allowed.sol#200-209) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (Allowed.sol#128-130) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes,string) (Allowed.sol#132-138) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (Allowed.sol#140-146) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (Allowed.sol#148-157) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (Allowed.sol#159-161) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (Allowed.sol#163-170) is never used and should be removed
AddressUpgradeable.isContract(address) (Allowed.sol#116-119) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (Allowed.sol#121-126) is never used and should be removed
AddressUpgradeable.verifyCallResult(bool,bytes,string) (Allowed.sol#188-198) is never used and should be removed
AddressUpgradeable.verifyCallResultFromTarget(address,bool,bytes,string) (Allowed.sol#172-186) is never used and should be removed
Allowed.__initialize_Allowed() (Allowed.sol#407-411) is never used and should be removed
Allowed.onlyAllowedAddress(address) (Allowed.sol#413-415) is never used and should be removed
ContextUpgradeable.__Context_init() (Allowed.sol#282-283) is never used and should be removed
ContextUpgradeable.__Context_init_unchained() (Allowed.sol#285-286) is never used and should be removed
ContextUpgradeable._msgData() (Allowed.sol#291-293) is never used and should be removed
ERC165Upgradeable.__ERC165_init() (Allowed.sol#300-301) is never used and should be removed
ERC165Upgradeable.__ERC165_init_unchained() (Allowed.sol#303-304) is never used and should be removed
Initializable.disableInitializers() (Allowed.sol#272-278) is never used and should be removed
StringsUpgradeable.toHexString(uint256) (Allowed.sol#85-96) is never used and should be removed
StringsUpgradeable.toString(uint256) (Allowed.sol#65-83) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

```

```

INFO:Detectors:
Pragma version0.8.4 (Allowed.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in AddressUpgradeable.sendValue(address,uint256) (Allowed.sol#121-126):
    - (success) = recipient.call{value: amount}() (Allowed.sol#124)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (Allowed.sol#148-157):
    - (success,returndata) = target.call{value: value}{data} (Allowed.sol#155)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (Allowed.sol#163-170):
    - (success,returndata) = target.staticcall(data) (Allowed.sol#168)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Function ContextUpgradeable.__Context_init() (Allowed.sol#282-283) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (Allowed.sol#285-286) is not in mixedCase
Variable ContextUpgradeable._gap (Allowed.sol#295) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init() (Allowed.sol#300-301) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init_unchained() (Allowed.sol#303-304) is not in mixedCase
Variable ERC165Upgradeable._gap (Allowed.sol#309) is not in mixedCase
Function AccessControlUpgradeable.__AccessControl_init() (Allowed.sol#313-314) is not in mixedCase
Function AccessControlUpgradeable.__AccessControl_init_unchained() (Allowed.sol#316-317) is not in mixedCase
Variable AccessControlUpgradeable._gap (Allowed.sol#401) is not in mixedCase
Function Allowed.__initialize_Allowed() (Allowed.sol#407-411) is not in mixedCase
Parameter Allowed.onlyAllowedAddress(address).address (Allowed.sol#413) is not in mixedCase
Parameter Allowed.addAllowedAddress(address,itemType).address (Allowed.sol#424) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
AccessControlUpgradeable._gap (Allowed.sol#401) is never used in Allowed (Allowed.sol#404-430)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variables

```

```

INFO:Detectors:
grantRole(bytes32,address) should be declared external:
  - AccessControlUpgradeable.grantRole(bytes32,address) (Allowed.sol#363-365)
revokeRole(bytes32,address) should be declared external:
  - AccessControlUpgradeable.revokeRole(bytes32,address) (Allowed.sol#367-369)
renounceRole(bytes32,address) should be declared external:
  - AccessControlUpgradeable.renounceRole(bytes32,address) (Allowed.sol#371-375)
addAllowedAddress(address,ItemType) should be declared external:
  - Allowed.addAllowedAddress(address,ItemType) (Allowed.sol#424-429)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:Allowed.sol analyzed (9 contracts with 75 detectors), 49 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration

```

File: Assets3D

```

INFO:Detectors:
Variable 'ERC1155._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).response (Assets3D.sol#929)' in ERC1155._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes) (Assets3D.sol#920-939) potentially used before declaration: response != IERC1155Receiver.onERC1155Received.selector (Assets3D.sol#930)
Variable 'ERC1155._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).reason (Assets3D.sol#933)' in ERC1155._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes) (Assets3D.sol#920-939) potentially used before declaration: revert(string)(reason) (Assets3D.sol#934)
Variable 'ERC1155._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes) (Assets3D.sol#951)' in ERC1155._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes) (Assets3D.sol#941-962) potentially used before declaration: response != IERC1155Receiver.onERC1155BatchReceived.selector (Assets3D.sol#953)
Variable 'ERC1155._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes).reason (Assets3D.sol#956)' in ERC1155._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes) (Assets3D.sol#941-962) potentially used before declaration: revert(string)(reason) (Assets3D.sol#957)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables
INFO:Detectors:
Reentrancy in Assets3D.mintBatchWithURI(address,uint256[],bytes,string[]) (Assets3D.sol#1157-1176):
  External calls:
    - _mintBatch(to,ids,amounts,data) (Assets3D.sol#1172)
      - IERC1155Receiver(to).onERC1155BatchReceived(operator,from,ids,amounts,data) (Assets3D.sol#950-960)
    State variables written after the call(s):
    - uris[ids[i_scope_0]] = uriStrings[i_scope_0] (Assets3D.sol#1174)
Reentrancy in Assets3D.mintWithURI(address,uint256,uint256,bytes,string) (Assets3D.sol#1143-1154):
  External calls:
    - _mint(to,id,amount,data) (Assets3D.sol#1152)
      - IERC1155Receiver(to).onERC1155Received(operator,from,id,amount,data) (Assets3D.sol#929-937)
    State variables written after the call(s):
    - uris[id] = uriString (Assets3D.sol#1153)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

```

```

INFO:Detectors:
Math.mulDiv(uint256,uint256,uint256) (Assets3D.sol#50-130) uses assembly
  - INLINE ASM (Assets3D.sol#61-65)
  - INLINE ASM (Assets3D.sol#81-88)
  - INLINE ASM (Assets3D.sol#95-104)
Strings.toString(uint256) (Assets3D.sol#345-363) uses assembly
  - INLINE ASM (Assets3D.sol#350-352)
  - INLINE ASM (Assets3D.sol#355-357)
Address.isContract(address) (Assets3D.sol#510-517) uses assembly
  - INLINE ASM (Assets3D.sol#513-515)
Address._functionCallWithValue(address,bytes,uint256,string) (Assets3D.sol#556-578) uses assembly
  - INLINE ASM (Assets3D.sol#570-573)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Assets3D.burnBatch(address,uint256[],uint256[]) (Assets3D.sol#1101-1116) has costly operations inside a loop:
  - _totalSupply = _totalSupply - values[i] (Assets3D.sol#1112)
Assets3D.mintBatch(address,uint256[],uint256[],bytes) (Assets3D.sol#1129-1140) has costly operations inside a loop:
  - _totalSupply = _totalSupply + amounts[i] (Assets3D.sol#1127)
Assets3D.mintBatchWithURI(address,uint256[],uint256[],bytes,string[]) (Assets3D.sol#1157-1176) has costly operations inside a loop:
  - _totalSupply = _totalSupply + amounts[i] (Assets3D.sol#1170)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#costly-operations-inside-a-loop
INFO:Detectors:
AccessControl._setRoleAdmin(bytes32,bytes32) (Assets3D.sol#488-492) is never used and should be removed
AccessControl._setupRole(bytes32,address) (Assets3D.sol#484-486) is never used and should be removed
Address._functionCallWithValue(address,bytes,uint256,string) (Assets3D.sol#556-578) is never used and should be removed
Address.functionCall(address,bytes) (Assets3D.sol#526-528) is never used and should be removed
Address.functionCall(address,bytes,string) (Assets3D.sol#530-536) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (Assets3D.sol#538-544) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (Assets3D.sol#546-554) is never used and should be removed
Address.sendValue(address,uint256) (Assets3D.sol#519-524) is never used and should be removed
Context._msgData() (Assets3D.sol#414-416) is never used and should be removed
Math.average(uint256,uint256) (Assets3D.sol#29-32) is never used and should be removed
Math.ceilDiv(uint256,uint256) (Assets3D.sol#40-43) is never used and should be removed
Math.log10(uint256) (Assets3D.sol#253-285) is never used and should be removed

```

```

Math.log2(uint256) (Assets3D.sol#200-236) is never used and should be removed
Math.log2(uint256,Math.Rounding) (Assets3D.sol#242-247) is never used and should be removed
Math.log256(uint256) (Assets3D.sol#304-328) is never used and should be removed
Math.log256(uint256,Math.Rounding) (Assets3D.sol#334-339) is never used and should be removed
Math.max(uint256,uint256) (Assets3D.sol#14-16) is never used and should be removed
Math.min(uint256,uint256) (Assets3D.sol#21-23) is never used and should be removed
Math.mulDiv(uint256,uint256,uint256) (Assets3D.sol#50-130) is never used and should be removed
Math.mulDiv(uint256,uint256,uint256,Math.Rounding) (Assets3D.sol#135-146) is never used and should be removed
Math.sqrt(uint256) (Assets3D.sol#153-184) is never used and should be removed
Math.sqrt(uint256,Math.Rounding) (Assets3D.sol#189-194) is never used and should be removed
Strings.toHexString(uint256) (Assets3D.sol#365-369) is never used and should be removed
Strings.toString(uint256) (Assets3D.sol#345-363) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version 0.8.4 (Assets3D.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in Address.sendValue(address,uint256) (Assets3D.sol#519-524):
  - (success) = recipient.call{value: amount}() (Assets3D.sol#522)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (Assets3D.sol#556-578):
  - (success,returnData) = target.call{value: weiValue}{data} (Assets3D.sol#564)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Variable Assets3D._totalSupply (Assets3D.sol#1052) is not in mixedCase
Variable Assets3D._name (Assets3D.sol#1050) is not in mixedCase
Variable Assets3D._symbol (Assets3D.sol#1051) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

```

```

INFO:Detectors:
renounceRole(bytes32,address) should be declared external:
  - AccessControl.renounceRole(bytes32,address) (Assets3D.sol#478-482)
uri(uint256) should be declared external:
  - Assets3D.uri(uint256) (Assets3D.sol#1080-1082)
  - ERC1155.uri(uint256) (Assets3D.sol#665-667)
balanceOfBatch(address[],uint256[]) should be declared external:
  - ERC1155.balanceOfBatch(address[],uint256[]) (Assets3D.sol#674-690)
setApprovalForAll(address,bool) should be declared external:
  - ERC1155.setApprovalForAll(address,bool) (Assets3D.sol#692-694)
safeTransferFrom(address,address,uint256,uint256,bytes) should be declared external:
  - ERC1155.safeTransferFrom(address,address,uint256,uint256,bytes) (Assets3D.sol#700-712)
safeBatchTransferFrom(address,address,uint256[],uint256[],bytes) should be declared external:
  - ERC1155.safeBatchTransferFrom(address,address,uint256[],uint256[],bytes) (Assets3D.sol#714-726)
exists(uint256) should be declared external:
  - ERC1155Supply.exists(uint256) (Assets3D.sol#984-986)
burn(address,uint256,uint256) should be declared external:
  - Assets3D.burn(address,uint256,uint256) (Assets3D.sol#1085-1098)
  - ERC1155Burnable.burn(address,uint256,uint256) (Assets3D.sol#1021-1032)
burnBatch(address,uint256[],uint256[]) should be declared external:
  - Assets3D.burnBatch(address,uint256[],uint256[]) (Assets3D.sol#1101-1116)
  - ERC1155Burnable.burnBatch(address,uint256[],uint256[]) (Assets3D.sol#1034-1045)
setURI(uint256,string) should be declared external:
  - Assets3D.setURI(uint256,string) (Assets3D.sol#1061-1066)
totalSupply() should be declared external:
  - Assets3D.totalSupply() (Assets3D.sol#1068-1070)
name() should be declared external:
  - Assets3D.name() (Assets3D.sol#1072-1074)
symbol() should be declared external:
  - Assets3D.symbol() (Assets3D.sol#1076-1078)
mint(address,uint256,uint256,bytes) should be declared external:
  - Assets3D.mint(address,uint256,uint256,bytes) (Assets3D.sol#1118-1127)
mintBatch(address,uint256[],uint256[],bytes) should be declared external:
  - Assets3D.mintBatch(address,uint256[],uint256[],bytes) (Assets3D.sol#1129-1140)
mintWithURI(address,uint256,uint256,bytes,string) should be declared external:
  - Assets3D.mintWithURI(address,uint256,uint256,bytes,string) (Assets3D.sol#1143-1154)

```

File: Auctions

```

INFO:Detectors:
AddressUpgradeable._revert(bytes,string) (Auctions.sol#200-209) uses assembly
    - INLINE ASM (Auctions.sol#202-205)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
AccessControlUpgradeable._AccessControl_init() (Auctions.sol#314-315) is never used and should be removed
AccessControlUpgradeable._AccessControl_init_unchained() (Auctions.sol#317-318) is never used and should be removed
AccessControlUpgradeable._setRoleAdmin(bytes32,bytes32) (Auctions.sol#382-386) is never used and should be removed
AccessControlUpgradeable._setupRole(bytes32,address) (Auctions.sol#378-380) is never used and should be removed
AddressUpgradeable._revert(bytes,string) (Auctions.sol#200-209) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (Auctions.sol#128-130) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes,string) (Auctions.sol#132-138) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (Auctions.sol#140-146) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (Auctions.sol#148-157) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (Auctions.sol#159-161) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (Auctions.sol#163-170) is never used and should be removed
AddressUpgradeable.isContract(address) (Auctions.sol#116-119) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (Auctions.sol#121-126) is never used and should be removed
AddressUpgradeable.verifyCallResult(bool,bytes,string) (Auctions.sol#188-198) is never used and should be removed
AddressUpgradeable.verifyCallResultFromTarget(address,bool,bytes,string) (Auctions.sol#172-186) is never used and should be removed
Allowed._initialize_Allowed() (Auctions.sol#408-412) is never used and should be removed
Allowed.onlyAllowedAddress(address) (Auctions.sol#414-416) is never used and should be removed
ContextUpgradeable._Context_init() (Auctions.sol#282-283) is never used and should be removed
ContextUpgradeable._Context_init_unchained() (Auctions.sol#285-286) is never used and should be removed
ContextUpgradeable._msgData() (Auctions.sol#291-293) is never used and should be removed
ERC165Upgradeable._ERC165_init() (Auctions.sol#300-301) is never used and should be removed
ERC165Upgradeable._ERC165_init_unchained() (Auctions.sol#303-304) is never used and should be removed
Initializable._disableInitializers() (Auctions.sol#272-278) is never used and should be removed
PausableUpgradeable._Pausable_init() (Auctions.sol#508-510) is never used and should be removed
PausableUpgradeable._Pausable_init_unchained() (Auctions.sol#512-514) is never used and should be removed

PausableUpgradeable._pause() (Auctions.sol#568-571) is never used and should be removed
PausableUpgradeable._requireNotPaused() (Auctions.sol#550-552) is never used and should be removed
PausableUpgradeable._requirePaused() (Auctions.sol#557-559) is never used and should be removed
PausableUpgradeable._unpause() (Auctions.sol#580-583) is never used and should be removed
ReentrancyGuardUpgradeable._ReentrancyGuard_init() (Auctions.sol#441-443) is never used and should be removed
ReentrancyGuardUpgradeable._ReentrancyGuard_init_unchained() (Auctions.sol#445-447) is never used and should be removed
ReentrancyGuardUpgradeable._nonReentrantAfter() (Auctions.sol#470-474) is never used and should be removed
ReentrancyGuardUpgradeable._nonReentrantBefore() (Auctions.sol#462-468) is never used and should be removed
ReentrancyGuardUpgradeable._reentrancyGuardEntered() (Auctions.sol#480-482) is never used and should be removed
Store._Store_init(address,uint256) (Auctions.sol#593-598) is never used and should be removed
StringsUpgradeable.toHexString(uint256) (Auctions.sol#85-96) is never used and should be removed
StringsUpgradeable.toString(uint256) (Auctions.sol#65-83) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version0.8.4 (Auctions.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in AddressUpgradeable.sendValue(address,uint256) (Auctions.sol#121-126):
    - (success) = recipient.call{value: amount}() (Auctions.sol#124)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (Auctions.sol#148-157):
    - (success,returndata) = target.call{value: value}(data) (Auctions.sol#155)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (Auctions.sol#163-170):
    - (success,returndata) = target.staticcall(data) (Auctions.sol#168)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Function ContextUpgradeable._Context_init() (Auctions.sol#282-283) is not in mixedCase
Function ContextUpgradeable._Context_init_unchained() (Auctions.sol#285-286) is not in mixedCase
Variable ContextUpgradeable._gap (Auctions.sol#295) is not in mixedCase
Function ERC165Upgradeable._ERC165_init() (Auctions.sol#300-301) is not in mixedCase
Function ERC165Upgradeable._ERC165_init_unchained() (Auctions.sol#303-304) is not in mixedCase
Variable ERC165Upgradeable._gap (Auctions.sol#309) is not in mixedCase
Function AccessControlUpgradeable._AccessControl_init() (Auctions.sol#314-315) is not in mixedCase
Function AccessControlUpgradeable._AccessControl_init_unchained() (Auctions.sol#317-318) is not in mixedCase
Variable AccessControlUpgradeable._gap (Auctions.sol#402) is not in mixedCase
Function Allowed._initialize_Allowed() (Auctions.sol#408-412) is not in mixedCase
Parameter Allowed.onlyAllowedAddress(address)._address (Auctions.sol#414) is not in mixedCase

```

```

Parameter Allowed.addAllowedAddress(address,ItemType)._address (Auctions.sol#425) is not in mixedCase
Function ReentrancyGuardUpgradeable._ReentrancyGuard_init() (Auctions.sol#441-443) is not in mixedCase
Function ReentrancyGuardUpgradeable._ReentrancyGuard_init_unchained() (Auctions.sol#445-447) is not in mixedCase
Variable ReentrancyGuardUpgradeable._gap (Auctions.sol#489) is not in mixedCase
Function PausableUpgradeable._Pausable_init() (Auctions.sol#508-510) is not in mixedCase
Function PausableUpgradeable._gap (Auctions.sol#512-514) is not in mixedCase
Variable PausableUpgradeable._gap (Auctions.sol#590) is not in mixedCase
Function Store._Store_init(address,uint256) (Auctions.sol#593-598) is not in mixedCase
Parameter Store._Store_init(address,uint256).feeReceiver (Auctions.sol#593) is not in mixedCase
Parameter Store._Store_init(address,uint256).fee (Auctions.sol#593) is not in mixedCase
Variable Store._freeIndexesSell (Auctions.sol#611) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
PausableUpgradeable._gap (Auctions.sol#590) is never used in Store (Auctions.sol#592-623)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variables
INFO:Detectors:
Store.offersCount (Auctions.sol#614) should be constant
Store.salesCount (Auctions.sol#608) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
INFO:Detectors:
grantRole(bytes32,address) should be declared external:
    - AccessControlUpgradeable.grantRole(bytes32,address) (Auctions.sol#364-366)
revokeRole(bytes32,address) should be declared external:
    - AccessControlUpgradeable.revokeRole(bytes32,address) (Auctions.sol#368-370)
renounceRole(bytes32,address) should be declared external:
    - AccessControlUpgradeable.renounceRole(bytes32,address) (Auctions.sol#372-376)
addAllowedAddress(address,ItemType) should be declared external:
    - Allowed.addAllowedAddress(address,ItemType) (Auctions.sol#425-430)
assetType(address) should be declared external:
    - Store.assetType(address) (Auctions.sol#620-622)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:Auctions.sol analyzed (12 contracts with 75 detectors), 75 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration

```

File: HeapOffers.sol

```

INFO:Detectors:
AddressUpgradeable._revert(bytes,string) (HeapOffers.sol#145-154) uses assembly
    - INLINE ASM (HeapOffers.sol#147-150)
HeapOffers.getBestAssets(address,uint256,uint8) (HeapOffers.sol#500-563) uses assembly
    - INLINE ASM (HeapOffers.sol#531-533)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
AccessControlUpgradeable._AccessControl_init() (HeapOffers.sol#258-259) is never used and should be removed
AccessControlUpgradeable._AccessControl_init_unchained() (HeapOffers.sol#261-262) is never used and should be removed
AccessControlUpgradeable._setRoleAdmin(bytes32,bytes32) (HeapOffers.sol#326-330) is never used and should be removed
AddressUpgradeable._revert(bytes,string) (HeapOffers.sol#145-154) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (HeapOffers.sol#73-75) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes,string) (HeapOffers.sol#77-83) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (HeapOffers.sol#85-91) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (HeapOffers.sol#93-102) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (HeapOffers.sol#104-106) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (HeapOffers.sol#108-115) is never used and should be removed
AddressUpgradeable.isContract(address) (HeapOffers.sol#61-64) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (HeapOffers.sol#66-71) is never used and should be removed
AddressUpgradeable.verifyCallResult(bool,bytes,string) (HeapOffers.sol#133-143) is never used and should be removed
AddressUpgradeable.verifyCallResultFromTarget(address,bool,bytes,string) (HeapOffers.sol#117-131) is never used and should be removed
ContextUpgradeable._Context_init() (HeapOffers.sol#227-228) is never used and should be removed
ContextUpgradeable._Context_init_unchained() (HeapOffers.sol#230-231) is never used and should be removed
ContextUpgradeable._msgData() (HeapOffers.sol#238-238) is never used and should be removed
ERC165Upgradeable._ERC165_init() (HeapOffers.sol#245-246) is never used and should be removed
ERC165Upgradeable._ERC165_init_unchained() (HeapOffers.sol#248-249) is never used and should be removed
Initializable._disableInitializers() (HeapOffers.sol#217-223) is never used and should be removed
StringsUpgradeable.toHexString(uint256) (HeapOffers.sol#30-41) is never used and should be removed
StringsUpgradeable.toString(uint256) (HeapOffers.sol#10-28) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version0.8.4 (HeapOffers.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

```

```

INFO:Detectors:
Low level call in AddressUpgradeable.sendValue(address,uint256) (HeapOffers.sol#66-71):
    - (success) = recipient.call{value: amount}() (HeapOffers.sol#69)
Low level call in AddressUpgradeable.functionCallWithValue(address,uint256,string) (HeapOffers.sol#93-102):
    - (success,returnData) = target.call{value: value}{data} (HeapOffers.sol#100)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (HeapOffers.sol#108-115):
    - (success,returnData) = target.staticcall(data) (HeapOffers.sol#113)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Function ContextUpgradeable.__Context_init() (HeapOffers.sol#227-228) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (HeapOffers.sol#230-231) is not in mixedCase
Variable ContextUpgradeable._gap (HeapOffers.sol#240) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init() (HeapOffers.sol#245-246) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init_unchained() (HeapOffers.sol#248-249) is not in mixedCase
Variable ERC165Upgradeable._gap (HeapOffers.sol#254) is not in mixedCase
Function AccessControlUpgradeable.__AccessControl_init() (HeapOffers.sol#258-259) is not in mixedCase
Function AccessControlUpgradeable.__AccessControl_init_unchained() (HeapOffers.sol#261-262) is not in mixedCase
Variable AccessControlUpgradeable._gap (HeapOffers.sol#346) is not in mixedCase
Parameter HeapOffers.addAdminRole(address)._admin (HeapOffers.sol#566) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
AccessControlUpgradeable._gap (HeapOffers.sol#346) is never used in HeapOffers (HeapOffers.sol#349-569)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variables

```

```

INFO:Detectors:
AccessControlUpgradeable._gap (HeapOffers.sol#346) is never used in HeapOffers (HeapOffers.sol#349-569)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variables
INFO:Detectors:
grantRole(bytes32,address) should be declared external:
    - AccessControlUpgradeable.grantRole(bytes32,address) (HeapOffers.sol#308-310)
revokeRole(bytes32,address) should be declared external:
    - AccessControlUpgradeable.revokeRole(bytes32,address) (HeapOffers.sol#312-314)
renounceRole(bytes32,address) should be declared external:
    - AccessControlUpgradeable.renounceRole(bytes32,address) (HeapOffers.sol#316-320)
insertNode(address,uint256,uint256,uint256) should be declared external:
    - HeapOffers.insertNode(address,uint256,uint256,uint256) (HeapOffers.sol#369-382)
deleteNode(address,uint256,uint256) should be declared external:
    - HeapOffers.deleteNode(address,uint256,uint256) (HeapOffers.sol#385-398)
getMaxNode(address,uint256) should be declared external:
    - HeapOffers.getMaxNode(address,uint256) (HeapOffers.sol#401-407)
getNode(address,uint256,uint256) should be declared external:
    - HeapOffers.getNode(address,uint256,uint256) (HeapOffers.sol#410-417)
addAdminRole(address) should be declared external:
    - HeapOffers.addAdminRole(address) (HeapOffers.sol#566-568)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:HeapOffers.sol analyzed (9 contracts with 75 detectors), 49 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration

```

File: HeapSales

```

INFO:Detectors:
HeapSales.getBestAssets(address,uint256,uint8) (HeapSales.sol#795-883) has external calls inside a loop: (seller) = store.sales[0][1].tokenIdOrderIndex (HeapSales.sol#824)
HeapSales.getBestAssets(address,uint256,uint8) (HeapSales.sol#795-883) has external calls inside a loop: IERC1155Upgradeable(toKem).balanceOfF(seller tokenId) > 0 (HeapSales.sol#826)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#calls-inside-a-loop
INFO:Detectors:
AddressUpgradeable._revert(bytes,string) (HeapSales.sol#338-347) uses assembly
    - INLINE ASM (HeapSales.sol#340-343)
HeapSales.getBestAssets(address,uint256,uint8) (HeapSales.sol#795-883) uses assembly
    - INLINE ASM (HeapSales.sol#836-838)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
AccessControlUpgradeable.__AccessControl_init() (HeapSales.sol#447-448) is never used and should be removed
AccessControlUpgradeable.__AccessControl_init_unchained() (HeapSales.sol#450-451) is never used and should be removed
AccessControlUpgradeable.setRoleAdmin(bytes32,bytes32) (HeapSales.sol#515-519) is never used and should be removed
AddressUpgradeable._revert(bytes,string) (HeapSales.sol#338-347) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (HeapSales.sol#266-268) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes,string) (HeapSales.sol#270-276) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (HeapSales.sol#278-284) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (HeapSales.sol#286-295) is never used and should be removed
Verify()
AddressUpgradeable.functionStaticCall(address,bytes) (HeapSales.sol#297-299) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (HeapSales.sol#301-308) is never used and should be removed
AddressUpgradeable.isContract(address) (HeapSales.sol#254-257) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (HeapSales.sol#259-264) is never used and should be removed
AddressUpgradeable.verifyCallResult(bool,bytes,string) (HeapSales.sol#326-336) is never used and should be removed
AddressUpgradeable.verifyCallResultFromTarget(address,bool,bytes,string) (HeapSales.sol#310-324) is never used and should be removed
move()
ContextUpgradeable.__Context_init() (HeapSales.sol#416-417) is never used and should be removed
ContextUpgradeable.__Context_init_unchained() (HeapSales.sol#419-420) is never used and should be removed
ContextUpgradeable._msgData() (HeapSales.sol#425-427) is never used and should be removed
ERC165Upgradeable.__ERC165_init() (HeapSales.sol#434-435) is never used and should be removed
ERC165Upgradeable.__ERC165_init_unchained() (HeapSales.sol#437-438) is never used and should be removed
Initializable._disableInitializers() (HeapSales.sol#406-412) is never used and should be removed
StringsUpgradeable.toHexString(uint256) (HeapSales.sol#223-234) is never used and should be removed
StringsUpgradeable.toString(uint256) (HeapSales.sol#203-221) is never used and should be removed

```

```

INFO:Detectors:
Pragma version0.8.4 (Heapsales.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in AddressUpgradeable.sendValue(address,uint256) (Heapsales.sol#259-264):
  - (success) = recipient.call{value: amount}() (Heapsales.sol#262)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (Heapsales.sol#286-295):
  - (success,returnData) = target.call{value: value}(data) (Heapsales.sol#293)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (Heapsales.sol#301-308):
  - (success,returnData) = target.staticcall(data) (Heapsales.sol#306)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Function IHeap.DEFAULT_ADMIN_ROLE() (Heapsales.sol#26) is not in mixedCase
Function ContextUpgradeable.__Context_init__() (Heapsales.sol#416-417) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained__() (Heapsales.sol#419-420) is not in mixedCase
Variable ContextUpgradeable._gap (Heapsales.sol#429) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init__() (Heapsales.sol#434-435) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init_unchained__() (Heapsales.sol#437-438) is not in mixedCase
Variable ERC165Upgradeable._gap (Heapsales.sol#443) is not in mixedCase
Function AccessControlUpgradeable.__AccessControl_init__() (Heapsales.sol#447-448) is not in mixedCase
Function AccessControlUpgradeable.__AccessControl_init_unchained__() (Heapsales.sol#450-451) is not in mixedCase
Variable AccessControlUpgradeable._gap (Heapsales.sol#535) is not in mixedCase
Function IStore._freeIndexesSell(uint256) (Heapsales.sol#539) is not in mixedCase
Parameter Heapsales.addStore(IStore)._store (Heapsales.sol#886) is not in mixedCase
Parameter Heapsales.addAdminRole(address)._admin (Heapsales.sol#891) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
AccessControlUpgradeable._gap (Heapsales.sol#535) is never used in Heapsales (Heapsales.sol#629-894)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variables

```

```

INFO:Detectors:
grantRole(bytes32,address) should be declared external:
  - AccessControlUpgradeable.grantRole(bytes32,address) (Heapsales.sol#497-499)
revokeRole(bytes32,address) should be declared external:
  - AccessControlUpgradeable.revokeRole(bytes32,address) (Heapsales.sol#501-503)
renounceRole(bytes32,address) should be declared external:
  - AccessControlUpgradeable.renounceRole(bytes32,address) (Heapsales.sol#505-509)
insertNode(address,uint256,uint256,uint256) should be declared external:
  - Heapsales.insertNode(address,uint256,uint256,uint256) (Heapsales.sol#651-664)
deleteNode(address,uint256,uint256) should be declared external:
  - Heapsales.deleteNode(address,uint256,uint256) (Heapsales.sol#667-680)
updateNode(address,uint256,uint256,uint256) should be declared external:
  - Heapsales.updateNode(address,uint256,uint256,uint256) (Heapsales.sol#683-693)
getMinNode(address,uint256) should be declared external:
  - Heapsales.getMinNode(address,uint256) (Heapsales.sol#696-702)
getNode(address,uint256,uint256) should be declared external:
  - Heapsales.getNode(address,uint256,uint256) (Heapsales.sol#705-712)
addStore(IStore) should be declared external:
  - Heapsales.addStore(IStore) (Heapsales.sol#886-888)
addAdminRole(address) should be declared external:
  - Heapsales.addAdminRole(address) (Heapsales.sol#891-893)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:Heapsales.sol analyzed (12 contracts with 75 detectors), 56 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration

```

File: offers

```

INFO:Detectors:
AddressUpgradeable._revert(bytes,string) (Offers.sol#200-209) uses assembly
    - INLINE ASM (Offers.sol#202-205)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
AccessControlUpgradeable.__AccessControl_init() (Offers.sol#313-314) is never used and should be removed
AccessControlUpgradeable.__AccessControl_init_unchained() (Offers.sol#316-317) is never used and should be removed
AccessControlUpgradeable._checkRole(bytes32) (Offers.sol#340-342) is never used and should be removed
AccessControlUpgradeable._checkRole(bytes32,address) (Offers.sol#344-357) is never used and should be removed
AccessControlUpgradeable._grantRole(bytes32,address) (Offers.sol#387-392) is never used and should be removed
AccessControlUpgradeable._revokeRole(bytes32,address) (Offers.sol#394-399) is never used and should be removed
AccessControlUpgradeable._setRoleAdmin(bytes32,bytes32) (Offers.sol#381-385) is never used and should be removed
AccessControlUpgradeable._setupRole(bytes32,address) (Offers.sol#377-379) is never used and should be removed
AddressUpgradeable._revert(bytes,string) (Offers.sol#200-209) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (Offers.sol#128-130) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes,string) (Offers.sol#132-138) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (Offers.sol#140-146) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (Offers.sol#148-157) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (Offers.sol#163-170) is never used and should be removed
AddressUpgradeable.isContract(address) (Offers.sol#116-119) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (Offers.sol#121-126) is never used and should be removed
AddressUpgradeable.verifyCallResult(bool,bytes,string) (Offers.sol#188-198) is never used and should be removed
AddressUpgradeable.verifyCallResultFromTarget(address,bool,bytes,string) (Offers.sol#172-186) is never used and should be removed
Allowed._initialize_Allowed() (Offers.sol#407-411) is never used and should be removed
Allowed.onlyAllowedAddress(address) (Offers.sol#413-415) is never used and should be removed
ContextUpgradeable.__Context_init() (Offers.sol#282-283) is never used and should be removed
ContextUpgradeable.__Context_init_unchained() (Offers.sol#285-286) is never used and should be removed
ContextUpgradeable._msgData() (Offers.sol#291-293) is never used and should be removed
ContextUpgradeable._msgSender() (Offers.sol#287-289) is never used and should be removed
ERC165Upgradeable._ERC165_init() (Offers.sol#300-301) is never used and should be removed
ERC165Upgradeable._ERC165_init_unchained() (Offers.sol#303-304) is never used and should be removed
Initializable._disableInitializers() (Offers.sol#272-278) is never used and should be removed
offers._Offers_init(uint256,address) (Offers.sol#1181-1184) is never used and should be removed
offers.buyOrderExist(uint256) (Offers.sol#1205-1207) is never used and should be removed

```

```

PausableUpgradeable.__Pausable_init() (Offers.sol#505-507) is never used and should be removed
PausableUpgradeable._Pausable_init_unchained() (Offers.sol#509-511) is never used and should be removed
PausableUpgradeable._pause() (Offers.sol#565-568) is never used and should be removed
PausableUpgradeable._requireNotPaused() (Offers.sol#547-549) is never used and should be removed
PausableUpgradeable._requirePaused() (Offers.sol#554-556) is never used and should be removed
PausableUpgradeable._unpause() (Offers.sol#577-580) is never used and should be removed
ReentrancyGuardUpgradeable.__ReentrancyGuard_init() (Offers.sol#438-440) is never used and should be removed
ReentrancyGuardUpgradeable._ReentrancyGuard_init_unchained() (Offers.sol#442-444) is never used and should be removed
ReentrancyGuardUpgradeable._nonReentrantAfter() (Offers.sol#467-471) is never used and should be removed
ReentrancyGuardUpgradeable._nonReentrantBefore() (Offers.sol#459-465) is never used and should be removed
ReentrancyGuardUpgradeable._reentrancyGuardEntered() (Offers.sol#477-479) is never used and should be removed
Store._Store_init(address,uint256) (Offers.sol#590-595) is never used and should be removed
StringsUpgradeable.toHexString(address) (Offers.sol#110-112) is never used and should be removed
StringsUpgradeable.toHexString(uint256) (Offers.sol#85-96) is never used and should be removed
StringsUpgradeable.toHexString(uint256,uint256) (Offers.sol#98-108) is never used and should be removed
StringsUpgradeable.toString(uint256) (Offers.sol#65-83) is never used and should be removed
TokenUtils._TokenUtils_init(address) (Offers.sol#950-952) is never used and should be removed
TokenUtils.transferERC1155(address,address,address,uint256,uint256) (Offers.sol#966-974) is never used and should be removed
TokenUtils.transferERC1155Batch(address,address,address,uint256[],uint256[]) (Offers.sol#985-999) is never used and should be removed
TokenUtils.transferERC20(address,uint256) (Offers.sol#962-964) is never used and should be removed
TokenUtils.transferERC721(address,address,address,uint256) (Offers.sol#976-983) is never used and should be removed
TokenUtils.transferERC721Batch(address,address,address,uint256[]) (Offers.sol#1001-1010) is never used and should be removed
TokenUtils.transferFromERC20(address,address,uint256) (Offers.sol#954-960) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version0.8.4 (Offers.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in AddressUpgradeable.sendValue(address,uint256) (Offers.sol#121-126):
    - (success) = recipient.call{value: amount}() (Offers.sol#124)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (Offers.sol#148-157):
    - (success,returndata) = target.call{value: value}{data} (Offers.sol#155)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (Offers.sol#163-170):
    - (success,returndata) = target.staticcall(data) (Offers.sol#168)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

```

```

INFO:Detectors:
Function ContextUpgradeable.__Context_init() (Offers.sol#282-283) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (Offers.sol#285-286) is not in mixedCase
Variable ContextUpgradeable._gap (Offers.sol#295) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init() (Offers.sol#300-301) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init_unchained() (Offers.sol#303-304) is not in mixedCase
Variable ERC165Upgradeable._gap (Offers.sol#309) is not in mixedCase
Function AccessControlUpgradeable.__AccessControl_init() (Offers.sol#313-314) is not in mixedCase
Function AccessControlUpgradeable.__AccessControl_init_unchained() (Offers.sol#316-317) is not in mixedCase
Variable AccessControlUpgradeable._gap (Offers.sol#401) is not in mixedCase
Function Allowed.__initialize_Allowed() (Offers.sol#407-411) is not in mixedCase
Parameter Allowed.onlyAllowedAddress(address)._address (Offers.sol#413) is not in mixedCase
Parameter Allowed.addAllowedAddress(address,ItemType)._address (Offers.sol#424) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init() (Offers.sol#438-440) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init_unchained() (Offers.sol#442-444) is not in mixedCase
Variable ReentrancyGuardUpgradeable._gap (Offers.sol#486) is not in mixedCase
Function PausableUpgradeable.__Pausable_init() (Offers.sol#505-507) is not in mixedCase
Function PausableUpgradeable.__Pausable_init_unchained() (Offers.sol#509-511) is not in mixedCase
Variable PausableUpgradeable._gap (Offers.sol#587) is not in mixedCase
Function Store.__Store_init(address,uint256) (Offers.sol#590-595) is not in mixedCase
Parameter Store.__Store_init(address,uint256)._feeReceiver (Offers.sol#590) is not in mixedCase
Parameter Store.__Store_init(address,uint256)._fee (Offers.sol#590) is not in mixedCase
Variable Store.freeIndexesSell (Offers.sol#608) is not in mixedCase
Function TokenUtils.__TokenUtils_init(address) (Offers.sol#950-952) is not in mixedCase
Parameter TokenUtils.__TokenUtils_init(address)._token (Offers.sol#950) is not in mixedCase
Variable TokenUtils.OVRToken (Offers.sol#948) is not in mixedCase
Function IHeap.DEFAULT_ADMIN_ROLE() (Offers.sol#1035) is not in mixedCase
Function Offers.__Offers_init(uint256,address) (Offers.sol#1181-1184) is not in mixedCase
Parameter Offers.__Offers_init(uint256,address)._count (Offers.sol#1181) is not in mixedCase
Parameter Offers.__Offers_init(uint256,address)._heap (Offers.sol#1181) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

```

```

INFO:Detectors:
Offers (Offers.sol#1178-1298) does not implement functions:
- AccessControlUpgradeable.__AccessControl_init() (Offers.sol#313-314)
- AccessControlUpgradeable.__AccessControl_init_unchained() (Offers.sol#316-317)
- ContextUpgradeable.__Context_init() (Offers.sol#282-283)
- ContextUpgradeable.__Context_init_unchained() (Offers.sol#285-286)
- ERC165Upgradeable.__ERC165_init() (Offers.sol#300-301)
- ERC165Upgradeable.__ERC165_init_unchained() (Offers.sol#303-304)
- PausableUpgradeable.__Pausable_init() (Offers.sol#505-507)
- PausableUpgradeable.__Pausable_init_unchained() (Offers.sol#509-511)
- ReentrancyGuardUpgradeable.__ReentrancyGuard_init() (Offers.sol#438-440)
- ReentrancyGuardUpgradeable.__ReentrancyGuard_init_unchained() (Offers.sol#442-444)
- Store.__Store_init(address,uint256) (Offers.sol#590-595)
- TokenUtils.__TokenUtils_init(address) (Offers.sol#950-952)
- Allowed.__initialize_Allowed() (Offers.sol#407-411)
- AccessControlUpgradeable._checkRole(bytes32) (Offers.sol#340-342)
- AccessControlUpgradeable._checkRole(bytes32,address) (Offers.sol#344-357)
- Initializable.disableInitializers() (Offers.sol#272-278)
- AccessControlUpgradeable._grantRole(bytes32,address) (Offers.sol#387-392)
- ContextUpgradeable._msgData() (Offers.sol#291-293)
- ContextUpgradeable._msgSender() (Offers.sol#287-289)
- ReentrancyGuardUpgradeable._nonReentrantAfter() (Offers.sol#467-471)
- ReentrancyGuardUpgradeable._nonReentrantBefore() (Offers.sol#459-465)
- PausableUpgradeable._pause() (Offers.sol#565-568)
- ReentrancyGuardUpgradeable._reentrancyGuardEntered() (Offers.sol#477-479)
- PausableUpgradeable._requireNotPaused() (Offers.sol#547-549)
- PausableUpgradeable._requirePaused() (Offers.sol#554-556)
- AccessControlUpgradeable._revokeRole(bytes32,address) (Offers.sol#394-399)
- AccessControlUpgradeable._setRoleAdmin(bytes32,bytes32) (Offers.sol#381-385)
- AccessControlUpgradeable._setupRole(bytes32,address) (Offers.sol#377-379)
- PausableUpgradeable._unpause() (Offers.sol#577-580)
- Allowed.addAllowedAddress(address,ItemType) (Offers.sol#424-429)
- Store.assetType(address) (Offers.sol#617-619)
- Offers.buyOrderExist(uint256) (Offers.sol#1205-1207)
- Offers.createOffer(address,uint256,uint256) (Offers.sol#1215-1242)
- Offers.deleteOffer(uint256) (Offers.sol#1248-1261)
- Offers.fillOffer(uint256) (Offers.sol#1267-1297)

```

```

- AccessControlUpgradeable.getRoleAdmin(bytes32) (Offers.sol#359-361)
- AccessControlUpgradeable.grantRole(bytes32,address) (Offers.sol#363-365)
- AccessControlUpgradeable.hasRole(bytes32,address) (Offers.sol#336-338)
- Allowed.onlyAllowedAddress(address) (Offers.sol#413-415)
- PausableUpgradeable.paused() (Offers.sol#540-542)
- AccessControlUpgradeable.renounceRole(bytes32,address) (Offers.sol#371-375)
- AccessControlUpgradeable.revokeRole(bytes32,address) (Offers.sol#367-369)
- AccessControlUpgradeable.supportsInterface(bytes4) (Offers.sol#332-334)
- TokenUtils.transferERC1155(address,address,address,uint256,uint256) (Offers.sol#966-974)
- TokenUtils.transferERC1155Batch(address,address,address,uint256[],uint256[]) (Offers.sol#985-999)
- TokenUtils.transferERC20(address,uint256) (Offers.sol#962-964)
- TokenUtils.transferERC721(address,address,address,uint256) (Offers.sol#976-983)
- TokenUtils.transferERC721Batch(address,address,address,uint256[]) (Offers.sol#1001-1010)
- TokenUtils.transferFromERC20(address,address,uint256) (Offers.sol#954-960)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions
INFO:Detectors:
PausableUpgradeable.__gap (Offers.sol#587) is never used in Offers (Offers.sol#1178-1298)
AccessControlUpgradeable._roles (Offers.sol#323) is never used in Offers (Offers.sol#1178-1298)
ReentrancyGuardUpgradeable._NOT_ENTERED (Offers.sol#433) is never used in Offers (Offers.sol#1178-1298)
ReentrancyGuardUpgradeable._ENTERED (Offers.sol#434) is never used in Offers (Offers.sol#1178-1298)
ReentrancyGuardUpgradeable._status (Offers.sol#436) is never used in Offers (Offers.sol#1178-1298)
PausableUpgradeable._paused (Offers.sol#500) is never used in Offers (Offers.sol#1178-1298)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variables
INFO:Detectors:
Store.salesCount (Offers.sol#605) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
INFO:Detectors:
grantRole(bytes32,address) should be declared external:
- AccessControlUpgradeable.grantRole(bytes32,address) (Offers.sol#363-365)
revokeRole(bytes32,address) should be declared external:
- AccessControlUpgradeable.revokeRole(bytes32,address) (Offers.sol#367-369)
renounceRole(bytes32,address) should be declared external:
- AccessControlUpgradeable.renounceRole(bytes32,address) (Offers.sol#371-375)
addAllowedAddress(address,ItemType) should be declared external:
- Allowed.addAllowedAddress(address,ItemType) (Offers.sol#424-429)
assetType(address) should be declared external:
- Store.assetType(address) (Offers.sol#617-619)

addAllowedAddress(address,ItemType) should be declared external:
- Allowed.addAllowedAddress(address,ItemType) (Offers.sol#424-429)
assetType(address) should be declared external:
- Store.assetType(address) (Offers.sol#617-619)
viewOffersByAsset(address,uint256,uint8) should be declared external:
- Offers.viewOffersByAsset(address,uint256,uint8) (Offers.sol#1186-1203)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:Offers.sol analyzed (19 contracts with 75 detectors), 106 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration

```

File: OVRLand

```

INFO:Detectors:
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).retval (OVRLand.sol#877)' in ERC721._checkOnERC721Received(address,address,uint256,bytes) (OVRLand.sol#870-891) potentially used before declaration: retval == IERC721Receiver.onERC721Received.selector (OVRLand.sol#878)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (OVRLand.sol#879)' in ERC721._checkOnERC721Received(address,address,uint256,bytes) (OVRLand.sol#870-891) potentially used before declaration: reason.length == 0 (OVRLand.sol#880)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (OVRLand.sol#879)' in ERC721._checkOnERC721Received(address,address,uint256,bytes) (OVRLand.sol#870-891) potentially used before declaration: revert(uint256,uint256)(32 + reason.load(uint256(reason))) (OVRLand.sol#884)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables
INFO:Detectors:
Reentrancy in OVRLand.safeMint(address,uint256,string) (OVRLand.sol#1254-1262):
  External calls:
    - _safeMint(to tokenId) (OVRLand.sol#1259)
      - IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,data) (OVRLand.sol#877-887)
  State variables written after the call(s):
    - _setTokenURI(tokenId,uri) (OVRLand.sol#1260)
      - _tokenURIs[tokenId] = _tokenURI (OVRLand.sol#1154)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2
INFO:Detectors:
Math.mulDiv(uint256,uint256,uint256) (OVRLand.sol#50-130) uses assembly
  - INLINE ASM (OVRLand.sol#61-65)
  - INLINE ASM (OVRLand.sol#81-88)
  - INLINE ASM (OVRLand.sol#95-104)
Strings.toString(uint256) (OVRLand.sol#345-363) uses assembly
  - INLINE ASM (OVRLand.sol#350-352)
  - INLINE ASM (OVRLand.sol#355-357)
Address.isContract(address) (OVRLand.sol#509-516) uses assembly
  - INLINE ASM (OVRLand.sol#512-514)
Address._functionCallWithValue(address,bytes,uint256,string) (OVRLand.sol#555-577) uses assembly
  - INLINE ASM (OVRLand.sol#569-572)
ERC721._checkOnERC721Received(address,address,uint256,bytes) (OVRLand.sol#870-891) uses assembly
  - INLINE ASM (OVRLand.sol#883-885)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

```

```

INFO:Detectors:
Pragma version^0.8.4 (OVRLand.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in Address.sendValue(address,uint256) (OVRLand.sol#518-523):
  - (success) = recipient.call{value: amount}() (OVRLand.sol#521)
Low level call in Address._functionCallWithValue(address,bytes,uint256,string) (OVRLand.sol#555-577):
  - (success,returnData) = target.call{value: weiValue}(data) (OVRLand.sol#563)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Parameter OVRLand.addURIEditor(address)._editor (OVRLand.sol#1204) is not in mixedCase
Parameter OVRLand.removeURIEditor(address)._editor (OVRLand.sol#1208) is not in mixedCase
Parameter OVRLand.addMinter(address)._minter (OVRLand.sol#1215) is not in mixedCase
Parameter OVRLand.removeMinter(address)._minter (OVRLand.sol#1219) is not in mixedCase
Parameter OVRLand.addBurner(address)._burner (OVRLand.sol#1223) is not in mixedCase
Parameter OVRLand.removeBurner(address)._burner (OVRLand.sol#1227) is not in mixedCase
Parameter OVRLand.addAdminRole(address)._admin (OVRLand.sol#1231) is not in mixedCase
Parameter OVRLand.removeAdminRole(address)._admin (OVRLand.sol#1237) is not in mixedCase
Parameter OVRLand.setOVRLandURI(uint256,string)._tokenId (OVRLand.sol#1269) is not in mixedCase
Parameter OVRLand.setOVRLandURI(uint256,string)._uri (OVRLand.sol#1269) is not in mixedCase
Parameter OVRLand.burn(uint256)._tokenId (OVRLand.sol#1282) is not in mixedCase
Parameter OVRLand.batchBurn(uint256[])._tokenId (OVRLand.sol#1292) is not in mixedCase
Parameter OVRLand.mint(address,uint256)._user (OVRLand.sol#1305) is not in mixedCase
Parameter OVRLand.mint(address,uint256)._tokenId (OVRLand.sol#1305) is not in mixedCase
Parameter OVRLand.batchMintLands(address[],uint256[])._to (OVRLand.sol#1319) is not in mixedCase
Parameter OVRLand.batchMintLands(address[],uint256[])._tokenId (OVRLand.sol#1319) is not in mixedCase
Parameter OVRLand.batchMintLandsWithUri(address[],uint256[],string[])._to (OVRLand.sol#1336) is not in mixedCase
Parameter OVRLand.batchMintLandsWithUri(address[],uint256[],string[])._tokenId (OVRLand.sol#1337) is not in mixedCase
Parameter OVRLand.batchMintLandsWithUri(address[],uint256[],string[])._uri (OVRLand.sol#1338) is not in mixedCase
Parameter OVRLand.batchSetOVRLandURI(uint256[],string[])._tokenId (OVRLand.sol#1355) is not in mixedCase
Parameter OVRLand.batchSetOVRLandURI(uint256[],string[])._uri (OVRLand.sol#1355) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

```

```

INFO:Detectors:
renounceRole(bytes32,address) should be declared external:
    - AccessControl.renounceRole(bytes32,address) (OVRLand.sol#478-482)
name() should be declared external:
    - ERC721.name() (OVRLand.sol#676-678)
symbol() should be declared external:
    - ERC721.symbol() (OVRLand.sol#680-682)
approve(address,uint256) should be declared external:
    - ERC721.approve(address,uint256) (OVRLand.sol#695-705)
setApprovalForAll(address,bool) should be declared external:
    - ERC721.setApprovalForAll(address,bool) (OVRLand.sol#713-715)
transferFrom(address,address,uint256) should be declared external:
    - ERC721.transferFrom(address,address,uint256) (OVRLand.sol#721-729)
safeTransferFrom(address,address,uint256) should be declared external:
    - ERC721.safeTransferFrom(address,address,uint256) (OVRLand.sol#731-737)
tokenOfOwnerByIndex(address,uint256) should be declared external:
    - ERC721Enumerable.tokenOfOwnerByIndex(address,uint256) (OVRLand.sol#968-971)
tokenByIndex(uint256) should be declared external:
    - ERC721Enumerable.tokenByIndex(uint256) (OVRLand.sol#983-986)
addURIEditor(address) should be declared external:
    - OVRLand.addURIEditor(address) (OVRLand.sol#1204-1206)
removeURIEditor(address) should be declared external:
    - OVRLand.removeURIEditor(address) (OVRLand.sol#1208-1213)
addMinter(address) should be declared external:
    - OVRLand.addMinter(address) (OVRLand.sol#1215-1217)
removeMinter(address) should be declared external:
    - OVRLand.removeMinter(address) (OVRLand.sol#1219-1221)
addBurner(address) should be declared external:
    - OVRLand.addBurner(address) (OVRLand.sol#1223-1225)
removeBurner(address) should be declared external:
    - OVRLand.removeBurner(address) (OVRLand.sol#1227-1229)
addAdminRole(address) should be declared external:
    - OVRLand.addAdminRole(address) (OVRLand.sol#1231-1235)
removeAdminRole(address) should be declared external:
    - OVRLand.removeAdminRole(address) (OVRLand.sol#1237-1244)
    
```

```

removeAdminRole(address) should be declared external:
    - OVRLand.removeAdminRole(address) (OVRLand.sol#1237-1244)
setOVRLandURI(uint256,string) should be declared external:
    - OVRLand.setOVRLandURI(uint256,string) (OVRLand.sol#1269-1274)
batchBurn(uint256[]) should be declared external:
    - OVRLand.batchBurn(uint256[]) (OVRLand.sol#1292-1296)
mint(address,uint256) should be declared external:
    - OVRLand.mint(address,uint256) (OVRLand.sol#1305-1312)
batchMintLands(address[],uint256[]) should be declared external:
    - OVRLand.batchMintLands(address[],uint256[]) (OVRLand.sol#1319-1327)
batchMintLandsWithUri(address[],uint256[],string[]) should be declared external:
    - OVRLand.batchMintLandsWithUri(address[],uint256[],string[]) (OVRLand.sol#1335-1348)
batchSetOVRLandURI(uint256[],string[]) should be declared external:
    - OVRLand.batchSetOVRLandURI(uint256[],string[]) (OVRLand.sol#1355-1363)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:OVRLand.sol analyzed (17 contracts with 75 detectors), 97 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
    
```

File: Sales

```

INFO:Detectors:
SaleOrderRequires.saleCreationChecks(uint256,uint256[],uint256[],ItemType,bool,address,address) (Sales.sol#1013-1044) has external calls inside a loop: require(bool,string)(IERC1155Upgradeable(token).balanceOf(sender,tokenId[i]) >= amount[i],s12) (Sales.sol#1037)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#calls-inside-a-loop
INFO:Detectors:
AddressUpgradeable._revert(bytes,string) (Sales.sol#90-99) uses assembly
- INLINE ASM (Sales.sol#92-95)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
AccessControlUpgradeable.__AccessControl_init() (Sales.sol#702-703) is never used and should be removed
AccessControlUpgradeable.__AccessControl_init_unchained() (Sales.sol#705-706) is never used and should be removed
AccessControlUpgradeable._checkRole(bytes32) (Sales.sol#729-731) is never used and should be removed
AccessControlUpgradeable._checkRole(bytes32,address) (Sales.sol#733-746) is never used and should be removed
AccessControlUpgradeable._grantRole(bytes32,address) (Sales.sol#776-781) is never used and should be removed
AccessControlUpgradeable._revokeRole(bytes32,address) (Sales.sol#783-788) is never used and should be removed
AccessControlUpgradeable._setRoleAdmin(bytes32,bytes32) (Sales.sol#770-774) is never used and should be removed
AccessControlUpgradeable._setupRole(bytes32,address) (Sales.sol#766-768) is never used and should be removed
AddressUpgradeable._revert(bytes,string) (Sales.sol#90-99) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (Sales.sol#18-20) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes,string) (Sales.sol#22-28) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (Sales.sol#30-36) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (Sales.sol#38-47) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (Sales.sol#49-51) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (Sales.sol#53-60) is never used and should be removed
AddressUpgradeable.isContract(address) (Sales.sol#6-9) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (Sales.sol#11-16) is never used and should be removed
AddressUpgradeable.verifyCallResult(bool,bytes,string) (Sales.sol#78-88) is never used and should be removed
AddressUpgradeable.verifyCallResultFromTarget(address,bool,bytes,string) (Sales.sol#62-76) is never used and should be removed
Allowed._initialize_Allowed() (Sales.sol#796-800) is never used and should be removed
Allowed.onlyAllowedAddress(address) (Sales.sol#802-804) is never used and should be removed
ContextUpgradeable.__Context_init() (Sales.sol#671-672) is never used and should be removed
ContextUpgradeable.__Context_init_unchained() (Sales.sol#674-675) is never used and should be removed
ContextUpgradeable._msgData() (Sales.sol#680-682) is never used and should be removed
ContextUpgradeable._msgSender() (Sales.sol#676-678) is never used and should be removed
ERC165Upgradeable._ERC165_init() (Sales.sol#689-690) is never used and should be removed
ERC165Upgradeable.__ERC165_init_unchained() (Sales.sol#692-693) is never used and should be removed

```

```

Initializable._disableInitializers() (Sales.sol#140-146) is never used and should be removed
PausableUpgradeable._Pausable_init() (Sales.sol#895-897) is never used and should be removed
PausableUpgradeable._Pausable_init_unchained() (Sales.sol#899-901) is never used and should be removed
PausableUpgradeable._pause() (Sales.sol#955-958) is never used and should be removed
PausableUpgradeable._requireNotPaused() (Sales.sol#937-939) is never used and should be removed
PausableUpgradeable._requirePaused() (Sales.sol#944-946) is never used and should be removed
PausableUpgradeable._unpause() (Sales.sol#967-970) is never used and should be removed
ReentrancyGuardUpgradeable._ReentrancyGuard_init() (Sales.sol#828-830) is never used and should be removed
ReentrancyGuardUpgradeable._ReentrancyGuard_init_unchained() (Sales.sol#832-834) is never used and should be removed
ReentrancyGuardUpgradeable._nonReentrantAfter() (Sales.sol#857-861) is never used and should be removed
ReentrancyGuardUpgradeable._nonReentrantBefore() (Sales.sol#849-855) is never used and should be removed
ReentrancyGuardUpgradeable._reentrancyGuardEntered() (Sales.sol#867-869) is never used and should be removed
SaleOrderRequires.saleCreationChecks(uint256,uint256[],uint256[],ItemType,bool,address,address) (Sales.sol#1013-1044) is never used and should be removed
Sales._Sales_init(address,uint256) (Sales.sol#1214-1217) is never used and should be removed
Sales._cancelSale(uint256) (Sales.sol#1324-1336) is never used and should be removed
Store._Store_init(address,uint256) (Sales.sol#980-985) is never used and should be removed
StringsUpgradeable.toHexString(address) (Sales.sol#648-650) is never used and should be removed
StringsUpgradeable.toHexString(uint256) (Sales.sol#623-634) is never used and should be removed
StringsUpgradeable.toHexString(uint256,uint256) (Sales.sol#636-646) is never used and should be removed
StringsUpgradeable.toString(uint256) (Sales.sol#603-621) is never used and should be removed
TokenUtils._TokenUtils_init(address) (Sales.sol#479-481) is never used and should be removed
TokenUtils.transferERC1155(address,address,address,uint256,uint256) (Sales.sol#495-503) is never used and should be removed
TokenUtils.transferERC1155Batch(address,address,address,uint256[],uint256[]) (Sales.sol#514-528) is never used and should be removed
TokenUtils.transferERC20(address,uint256) (Sales.sol#491-493) is never used and should be removed
TokenUtils.transferERC721(address,address,address,uint256) (Sales.sol#505-512) is never used and should be removed
TokenUtils.transferERC721Batch(address,address,address,uint256[]) (Sales.sol#530-539) is never used and should be removed
TokenUtils.transferFromERC20(address,address,uint256) (Sales.sol#483-489) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version0.8.4 (Sales.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

```

```
INFO:Detectors:
Low level call in AddressUpgradeable.sendValue(address,uint256) (Sales.sol#11-16):
  - (success) = recipient.call{value: amount}() (Sales.sol#14)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (Sales.sol#38-47):
  - (success,returndata) = target.call{value: value}(data) (Sales.sol#45)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (Sales.sol#53-60):
  - (success,returndata) = target.staticcall(data) (Sales.sol#58)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
```

```
INFO:Detectors:
Sales (Sales.sol#1213-1416) does not implement functions:
- AccessControlUpgradeable.__AccessControl_init() (Sales.sol#702-703)
- AccessControlUpgradeable.__AccessControl_init_unchained() (Sales.sol#705-706)
- ContextUpgradeable.__Context_init() (Sales.sol#671-672)
- ContextUpgradeable.__Context_init_unchained() (Sales.sol#674-675)
- ERC165Upgradeable.__ERC165_init() (Sales.sol#689-690)
- ERC165Upgradeable.__ERC165_init_unchained() (Sales.sol#692-693)
- PausableUpgradeable.__Pausable_init() (Sales.sol#895-897)
- PausableUpgradeable.__Pausable_init_unchained() (Sales.sol#899-901)
- ReentrancyGuardUpgradeable.__ReentrancyGuard_init() (Sales.sol#828-830)
- ReentrancyGuardUpgradeable.__ReentrancyGuard_init_unchained() (Sales.sol#832-834)
- Store.__Store_init(address,uint256) (Sales.sol#980-985)
- TokenUtils.__TokenUtils_init(address) (Sales.sol#479-481)
- Allowed.__initialize_Allowed() (Sales.sol#796-800)
- Sales._cancelSale(uint256) (Sales.sol#1324-1336)
- AccessControlUpgradeable.checkRole(bytes32) (Sales.sol#729-731)
- AccessControlUpgradeable.checkRole(bytes32,address) (Sales.sol#733-746)
- Initializable._disableInitializers() (Sales.sol#140-146)
- AccessControlUpgradeable.grantRole(bytes32,address) (Sales.sol#776-781)
- ContextUpgradeable._msgData() (Sales.sol#680-682)
- ContextUpgradeable._msgSender() (Sales.sol#676-678)
- ReentrancyGuardUpgradeable._nonReentrantAfter() (Sales.sol#857-861)
- ReentrancyGuardUpgradeable._nonReentrantBefore() (Sales.sol#849-855)
- PausableUpgradeable._pause() (Sales.sol#955-958)
- ReentrancyGuardUpgradeable._reentrancyGuardEntered() (Sales.sol#867-869)
- PausableUpgradeable._requireNotPaused() (Sales.sol#937-939)
- PausableUpgradeable._requirePaused() (Sales.sol#944-946)
- AccessControlUpgradeable._revokeRole(bytes32,address) (Sales.sol#783-788)
- AccessControlUpgradeable._setRoleAdmin(bytes32,bytes32) (Sales.sol#770-774)
- AccessControlUpgradeable._setupRole(bytes32,address) (Sales.sol#766-768)
- PausableUpgradeable._unpause() (Sales.sol#967-970)
- Allowed.addAllowedAddress(address,Itemtype) (Sales.sol#813-818)
- Store.assetType(address) (Sales.sol#1007-1009)
- Sales.buy(uint256,uint256) (Sales.sol#1344-1415)
- Sales.cancelSale(uint256) (Sales.sol#1316-1322)
```

```
- Sales.cancelSale(uint256) (Sales.sol#1316-1322)
- Sales.createSale(address,uint256[],uint256[],OrderType) (Sales.sol#1260-1310)
- AccessControlUpgradeable.getRoleAdmin(bytes32) (Sales.sol#748-750)
- AccessControlUpgradeable.grantRole(bytes32,address) (Sales.sol#752-754)
- AccessControlUpgradeable.hasRole(bytes32,address) (Sales.sol#725-727)
- Allowed.onlyAllowedAddress(address) (Sales.sol#802-804)
- PausableUpgradeable.paused() (Sales.sol#930-932)
- AccessControlUpgradeable.renounceRole(bytes32,address) (Sales.sol#760-764)
- AccessControlUpgradeable.revokeRole(bytes32,address) (Sales.sol#756-758)
- AccessControlUpgradeable.supportsInterface(bytes4) (Sales.sol#721-723)
- TokenUtils.transferERC1155(address,address,address,uint256,uint256) (Sales.sol#495-503)
- TokenUtils.transferERC1155Batch(address,address,address,uint256[],uint256[]) (Sales.sol#514-528)
- TokenUtils.transferERC20(address,uint256) (Sales.sol#491-493)
- TokenUtils.transferERC721(address,address,address,uint256) (Sales.sol#505-512)
- TokenUtils.transferERC721Batch(address,address,address,uint256[]) (Sales.sol#530-539)
- TokenUtils.transferFromERC20(address,address,uint256) (Sales.sol#483-489)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions

INFO:Detectors:
PausableUpgradeable.__gap (Sales.sol#977) is never used in Sales (Sales.sol#1213-1416)
AccessControlUpgradeable._roles (Sales.sol#712) is never used in Sales (Sales.sol#1213-1416)
ReentrancyGuardUpgradeable._NOT_ENTERED (Sales.sol#823) is never used in Sales (Sales.sol#1213-1416)
ReentrancyGuardUpgradeable._ENTERED (Sales.sol#824) is never used in Sales (Sales.sol#1213-1416)
ReentrancyGuardUpgradeable._status (Sales.sol#826) is never used in Sales (Sales.sol#1213-1416)
PausableUpgradeable._paused (Sales.sol#890) is never used in Sales (Sales.sol#1213-1416)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variables
```

```

INFO:Detectors:
Store.offersCount (Sales.sol#1001) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
INFO:Detectors:
grantRole(bytes32,address) should be declared external:
  - AccessControlUpgradeable.grantRole(bytes32,address) (Sales.sol#752-754)
revokeRole(bytes32,address) should be declared external:
  - AccessControlUpgradeable.revokeRole(bytes32,address) (Sales.sol#756-758)
renounceRole(bytes32,address) should be declared external:
  - AccessControlUpgradeable.renounceRole(bytes32,address) (Sales.sol#760-764)
addAllowedAddress(address,ItemType) should be declared external:
  - Allowed.addAllowedAddress(address,ItemType) (Sales.sol#813-818)
assetType(address) should be declared external:
  - Store.assetType(address) (Sales.sol#1007-1009)
viewSalesByAsset(address,uint256,uint8) should be declared external:
  - Sales.viewSalesByAsset(address,uint256,uint8) (Sales.sol#1221-1244)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:Sales.sol analyzed (20 contracts with 75 detectors), 108 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration

```

File: Store

```

INFO:Detectors:
AddressUpgradeable._revert(bytes,string) (Store.sol#200-209) uses assembly
  - INLINE ASM (Store.sol#202-205)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
AccessControlUpgradeable.__AccessControl_init() (Store.sol#313-314) is never used and should be removed
AccessControlUpgradeable.__AccessControl_init_unchained() (Store.sol#316-317) is never used and should be removed
AccessControlUpgradeable._setRoleAdmin(bytes32,bytes32) (Store.sol#381-385) is never used and should be removed
AccessControlUpgradeable._setupRole(bytes32,address) (Store.sol#377-379) is never used and should be removed
AddressUpgradeable._revert(bytes,string) (Store.sol#200-209) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (Store.sol#128-130) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes,string) (Store.sol#132-138) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (Store.sol#140-146) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (Store.sol#148-157) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (Store.sol#159-161) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (Store.sol#163-170) is never used and should be removed
AddressUpgradeable.isContract(address) (Store.sol#116-119) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (Store.sol#121-126) is never used and should be removed
AddressUpgradeable.verifyCallResult(bool,bytes,string) (Store.sol#188-198) is never used and should be removed
AddressUpgradeable.verifyCallResultFromTarget(address,bool,bytes,string) (Store.sol#172-186) is never used and should be removed
Allowed.__initialize_Allowed() (Store.sol#407-411) is never used and should be removed
Allowed.onlyAllowedAddress(address) (Store.sol#413-415) is never used and should be removed
ContextUpgradeable.__Context_init() (Store.sol#282-283) is never used and should be removed
ContextUpgradeable.__Context_init_unchained() (Store.sol#285-286) is never used and should be removed
ContextUpgradeable._msgData() (Store.sol#291-293) is never used and should be removed
ERC165Upgradeable.__ERC165_init() (Store.sol#300-301) is never used and should be removed
ERC165Upgradeable.__ERC165_init_unchained() (Store.sol#303-304) is never used and should be removed
Initializable._disableInitializers() (Store.sol#272-278) is never used and should be removed
PausableUpgradeable._Pausable_init() (Store.sol#507-509) is never used and should be removed
PausableUpgradeable._Pausable_init_unchained() (Store.sol#511-513) is never used and should be removed
PausableUpgradeable._pause() (Store.sol#567-570) is never used and should be removed
PausableUpgradeable._requireNotPaused() (Store.sol#549-551) is never used and should be removed
PausableUpgradeable._requirePaused() (Store.sol#556-558) is never used and should be removed
PausableUpgradeable._unpause() (Store.sol#579-582) is never used and should be removed
ReentrancyGuardUpgradeable.__ReentrancyGuard_init() (Store.sol#439-441) is never used and should be removed

```

```

ReentrancyGuardUpgradeable._nonReentrantAfter() (Store.sol#468-472) is never used and should be removed
ReentrancyGuardUpgradeable._nonReentrantBefore() (Store.sol#460-466) is never used and should be removed
ReentrancyGuardUpgradeable._reentrancyGuardEntered() (Store.sol#478-480) is never used and should be removed
Store._Store_init(address,uint256) (Store.sol#592-597) is never used and should be removed
StringsUpgradeable.toHexString(uint256) (Store.sol#85-96) is never used and should be removed
StringsUpgradeable.toString(uint256) (Store.sol#65-83) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version0.8.4 (Store.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in AddressUpgradeable.sendValue(address,uint256) (Store.sol#121-126):
  - (success) = recipient.call{value: amount}() (Store.sol#124)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (Store.sol#148-157):
  - (success,returnData) = target.call{value: value}{data} (Store.sol#155)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (Store.sol#163-170):
  - (success,returnData) = target.staticcall(data) (Store.sol#168)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Function ContextUpgradeable.__Context_init() (Store.sol#282-283) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (Store.sol#285-286) is not in mixedCase
Variable ContextUpgradeable._gap (Store.sol#295) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init() (Store.sol#300-301) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init_unchained() (Store.sol#303-304) is not in mixedCase
Variable ERC165Upgradeable._gap (Store.sol#309) is not in mixedCase
Function AccessControlUpgradeable.__AccessControl_init() (Store.sol#313-314) is not in mixedCase
Function AccessControlUpgradeable.__AccessControl_init_unchained() (Store.sol#316-317) is not in mixedCase
Variable AccessControlUpgradeable._gap (Store.sol#401) is not in mixedCase
Function Allowed.__initialAllowed() (Store.sol#407-411) is not in mixedCase
Parameter Allowed.onlyAllowedAddress(address)._address (Store.sol#413) is not in mixedCase
Parameter Allowed.addAllowedAddress(address,ItemType)._address (Store.sol#424) is not in mixedCase
Function ReentrancyGuardUpgradeable._ReentrancyGuard_init() (Store.sol#439-441) is not in mixedCase
Function ReentrancyGuardUpgradeable._ReentrancyGuard_init_unchained() (Store.sol#443-445) is not in mixedCase
Variable PausableUpgradeable._gap (Store.sol#487) is not in mixedCase
Function PausableUpgradeable.__Pausable_init() (Store.sol#507-509) is not in mixedCase
Function PausableUpgradeable.__Pausable_init_unchained() (Store.sol#511-513) is not in mixedCase

```

```

Variable PausableUpgradeable._gap (Store.sol#589) is not in mixedCase
Function Store._Store_init(address,uint256) (Store.sol#592-597) is not in mixedCase
Parameter Store._Store_init(address,uint256)._feeReceiver (Store.sol#592) is not in mixedCase
Parameter Store._Store_init(address,uint256)._fee (Store.sol#592) is not in mixedCase
Variable Store._freeIndexesSell (Store.sol#610) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
PausableUpgradeable._gap (Store.sol#589) is never used in Store (Store.sol#591-622)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variables
INFO:Detectors:
Store.offersCount (Store.sol#613) should be constant
Store.salesCount (Store.sol#607) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
INFO:Detectors:
grantRole(bytes32,address) should be declared external:
  - AccessControlUpgradeable.grantRole(bytes32,address) (Store.sol#363-365)
revokeRole(bytes32,address) should be declared external:
  - AccessControlUpgradeable.revokeRole(bytes32,address) (Store.sol#367-369)
renounceRole(bytes32,address) should be declared external:
  - AccessControlUpgradeable.renounceRole(bytes32,address) (Store.sol#371-375)
addAllowedAddress(address,ItemType) should be declared external:
  - Allowed.addAllowedAddress(address,ItemType) (Store.sol#424-429)
assetType(address) should be declared external:
  - Store.assetType(address) (Store.sol#619-621)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:Store.sol analyzed (12 contracts with 75 detectors), 75 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration

```

File: Token

```

INFO:Detectors:
ERC20.allowance(address,address).owner (Token.sol#276) shadows:
  - Ownable.owner() (Token.sol#206-208) (function)
ERC20._approve(address,address,uint256).owner (Token.sol#348) shadows:
  - Ownable.owner() (Token.sol#206-208) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
INFO:Detectors:
Address.isContract(address) (Token.sol#84-91) uses assembly
  - INLINE ASM (Token.sol#87-89)
Address._functionCallWithValue(address,bytes,uint256,string) (Token.sol#130-152) uses assembly
  - INLINE ASM (Token.sol#144-147)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Address.functionCallWithValue(address,bytes,uint256,string) (Token.sol#130-152) is never used and should be removed
Address.functionCall(address,bytes) (Token.sol#100-102) is never used and should be removed
Address.functionCall(address,bytes,string) (Token.sol#104-110) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (Token.sol#112-118) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (Token.sol#120-128) is never used and should be removed
Address.isContract(address) (Token.sol#84-91) is never used and should be removed
Address.sendValue(address,uint256) (Token.sol#93-98) is never used and should be removed
Context._msgData() (Token.sol#189-192) is never used and should be removed
ERC20._burn(address,uint256) (Token.sol#339-345) is never used and should be removed
ERC20._burnFrom(address,uint256) (Token.sol#359-366) is never used and should be removed
SafeMath.div(uint256,uint256) (Token.sol#38-40) is never used and should be removed
SafeMath.div(uint256,uint256,string) (Token.sol#42-51) is never used and should be removed
SafeMath.min(uint256,uint256) (Token.sol#66-68) is never used and should be removed
SafeMath.mod(uint256,uint256) (Token.sol#53-55) is never used and should be removed
SafeMath.mod(uint256,uint256,string) (Token.sol#57-64) is never used and should be removed
SafeMath.mul(uint256,uint256) (Token.sol#27-36) is never used and should be removed
SafeMath.sqrt(uint256) (Token.sol#70-81) is never used and should be removed
SafeMath.sub(uint256,uint256) (Token.sol#12-14) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version^0.8.4 (Token.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

```

```

INFO:Detectors:
ERC20.allowance(address,address).owner (Token.sol#276) shadows:
  - Ownable.owner() (Token.sol#206-208) (function)
ERC20._approve(address,address,uint256).owner (Token.sol#348) shadows:
  - Ownable.owner() (Token.sol#206-208) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
INFO:Detectors:
Address.isContract(address) (Token.sol#84-91) uses assembly
  - INLINE ASM (Token.sol#87-89)
Address._functionCallWithValue(address,bytes,uint256,string) (Token.sol#130-152) uses assembly
  - INLINE ASM (Token.sol#144-147)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Address.functionCallWithValue(address,bytes,uint256,string) (Token.sol#130-152) is never used and should be removed
Address.functionCall(address,bytes) (Token.sol#100-102) is never used and should be removed
Address.functionCall(address,bytes,string) (Token.sol#104-110) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (Token.sol#112-118) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (Token.sol#120-128) is never used and should be removed
Address.isContract(address) (Token.sol#84-91) is never used and should be removed
Address.sendValue(address,uint256) (Token.sol#93-98) is never used and should be removed
Context._msgData() (Token.sol#189-192) is never used and should be removed
ERC20._burn(address,uint256) (Token.sol#339-345) is never used and should be removed
ERC20._burnFrom(address,uint256) (Token.sol#359-366) is never used and should be removed
SafeMath.div(uint256,uint256) (Token.sol#38-40) is never used and should be removed
SafeMath.div(uint256,uint256,string) (Token.sol#42-51) is never used and should be removed
SafeMath.min(uint256,uint256) (Token.sol#66-68) is never used and should be removed
SafeMath.mod(uint256,uint256) (Token.sol#53-55) is never used and should be removed
SafeMath.mod(uint256,uint256,string) (Token.sol#57-64) is never used and should be removed
SafeMath.mul(uint256,uint256) (Token.sol#27-36) is never used and should be removed
SafeMath.sqrt(uint256) (Token.sol#70-81) is never used and should be removed
SafeMath.sub(uint256,uint256) (Token.sol#12-14) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version^0.8.4 (Token.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

```

```

increaseAllowance(address,uint256) should be declared external:
  - ERC20.increaseAllowance(address,uint256) (Token.sol#299-302)
decreaseAllowance(address,uint256) should be declared external:
  - ERC20.decreaseAllowance(address,uint256) (Token.sol#304-311)
mint(uint256) should be declared external:
  - ERC20.mint(uint256) (Token.sol#313-316)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:Token.sol analyzed (7 contracts with 75 detectors), 41 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration

```

File: TokenUtils

```

INFO:Detectors:
TokenUtils.transferERC721Batch(address,address,address,uint256[]) (TokenUtils.sol#530-539) has external calls inside a loop: IE
RC721Upgradeable(token).safeTransferFrom(from,to,ids[i]) (TokenUtils.sol#537)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#calls-inside-a-loop
INFO:Detectors:
AddressUpgradeable._revert(bytes,string) (TokenUtils.sol#90-99) uses assembly
  - INLINE ASM (TokenUtils.sol#92-98)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
AddressUpgradeable._revert(bytes,string) (TokenUtils.sol#90-99) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (TokenUtils.sol#19-20) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes,string) (TokenUtils.sol#22-28) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (TokenUtils.sol#30-36) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (TokenUtils.sol#38-47) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (TokenUtils.sol#49-51) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (TokenUtils.sol#53-60) is never used and should be removed
AddressUpgradeable.isContract(address) (TokenUtils.sol#6-9) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (TokenUtils.sol#11-16) is never used and should be removed
AddressUpgradeable.verifyCallResult(bool,bytes,string) (TokenUtils.sol#78-88) is never used and should be removed
AddressUpgradeable.verifyCallResultFromTarget(address,bool,bytes,string) (TokenUtils.sol#62-76) is never used and should be removed
Initializable._disableInitializers() (TokenUtils.sol#140-146) is never used and should be removed
TokenUtils._TokenutilsInit(address) (TokenUtils.sol#479-481) is never used and should be removed
TokenUtils.transferERC1155(address,address,address,uint256,uint256) (TokenUtils.sol#495-503) is never used and should be removed
TokenUtils.transferERC1155Batch(address,address,address,uint256[],uint256[]) (TokenUtils.sol#514-528) is never used and should be removed
TokenUtils.transferERC20(address,uint256) (TokenUtils.sol#491-493) is never used and should be removed
TokenUtils.transferERC721(address,address,address,uint256) (TokenUtils.sol#505-512) is never used and should be removed
TokenUtils.transferERC721Batch(address,address,address,uint256[]) (TokenUtils.sol#530-539) is never used and should be removed
TokenUtils.transferFromERC20(address,address,uint256) (TokenUtils.sol#483-489) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

```

```

INFO:Detectors:
Pragma version0.8.4 (TokenUtils.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in AddressUpgradeable.sendValue(address,uint256) (TokenUtils.sol#11-16):
  - (success) = recipient.call{value: amount}() (TokenUtils.sol#14)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (TokenUtils.sol#38-47):
  - (success,returndata) = target.call{value: value}(data) (TokenUtils.sol#45)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (TokenUtils.sol#53-60):
  - (success,returndata) = target.staticcall(data) (TokenUtils.sol#58)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Function TokenUtils.__TokenUtils_init(address) (TokenUtils.sol#479-481) is not in mixedCase
Parameter TokenUtils._Tokenutils_init(address)._token (TokenUtils.sol#479) is not in mixedCase
Variable TokenUtils.OVRToken (TokenUtils.sol#477) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Slither:TokenUtils.sol analyzed (7 contracts with 75 detectors), 31 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration

```

Solidity Static Analysis

File: Allowed.sol

Miscellaneous

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

[more](#)

Pos: 19:8:

File: Assets3D

Gas & Economy

Gas costs:

Gas requirement of function Assets3D._name is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 11:4:

Gas costs:

Gas requirement of function Assets3D.burn is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 46:4:

Gas costs:

Gas requirement of function Assets3D.burnBatch is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 62:4:

Gas costs:

Gas requirement of function Assets3D.mint is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 79:4:

Gas costs:

Gas requirement of function Assets3D.mintBatch is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 90:4:

Gas costs:

Gas requirement of function Assets3D.mintBatchWithURI is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 118:4:

For loop over dynamic array:

Loops that do not have a fixed number of iterations, for example, loops that depend on storage values, have to be used carefully. Due to the block gas limit, transactions can only consume a certain amount of gas. The number of iterations in a loop can grow beyond the block gas limit which can cause the complete contract to be stalled at a certain point. Additionally, using unbounded loops incurs in a lot of avoidable gas costs. Carefully test how many items at maximum you can pass to such functions to make it successful.

[more](#)

Pos: 72:8:

For loop over dynamic array:

Loops that do not have a fixed number of iterations, for example, loops that depend on storage values, have to be used carefully. Due to the block gas limit, transactions can only consume a certain amount of gas. The number of iterations in a loop can grow beyond the block gas limit which can cause the complete contract to be stalled at a certain point. Additionally, using unbounded loops incurs in a lot of avoidable gas costs. Carefully test how many items at maximum you can pass to such functions to make it successful.

[more](#)

Pos: 134:8:

Miscellaneous**Constant/View/Pure functions:**

Assets3D._beforeTokenTransfer(address,address,address,uint256[],uint256[],bytes) : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 149:4:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

[more](#)

Pos: 67:8:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

[more](#)

Pos: 125:8:

File: Auctions

Security**Check-effects-interaction:**

Potential violation of Checks-Effects-Interaction pattern in
Auctions.createAuction(address,uint256,uint256,uint256,uint256): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 13:4:

Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in
`Auctions.cancelAuction(address,uint256)`: Could potentially lead to re-entrancy vulnerability.
Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 156:4:

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.

[more](#)

Pos: 5:11:

Gas & Economy

Gas costs:

Gas requirement of function `Auctions.createAuction` is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 13:4:

Gas costs:

Gas requirement of function Auctions.getAuctionDetails is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 178:4:

For loop over dynamic array:

Loops that do not have a fixed number of iterations, for example, loops that depend on storage values, have to be used carefully. Due to the block gas limit, transactions can only consume a certain amount of gas. The number of iterations in a loop can grow beyond the block gas limit which can cause the complete contract to be stalled at a certain point. Additionally, using unbounded loops incurs in a lot of avoidable gas costs. Carefully test how many items at maximum you can pass to such functions to make it successful.

[more](#)

Pos: 70:8:

Miscellaneous**Constant/View/Pure functions:**

AuctionsUtils.bidChecks(struct Auction,uint256,uint256,address,address,bool,uint256) : Is constant but potentially should not be. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 15:4:

Similar variable names:

TokenUtils.transferERC721(address,address,address,uint256) : Variables have very similar names "to" and "id". Note: Modifiers are currently not considered by this static analysis.

Pos: 45:61:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

[more](#)

Pos: 72:8:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

[more](#)

Pos: 74:8:

Delete from dynamic array:

Using "delete" on an array leaves a gap. The length of the array remains the same. If you want to remove the empty position you need to shift items manually and update the "length" property.

[more](#)

Pos: 172:8:

Data truncated:

Division of integer values yields an integer value again. That means e.g. $10 / 100 = 0$ instead of 0.1 since the result is an integer again. This does not hold for division of (only) literal values since those yield rational constants.

Pos: 118:23:

Data truncated:

Division of integer values yields an integer value again. That means e.g. $10 / 100 = 0$ instead of 0.1 since the result is an integer again. This does not hold for division of (only) literal values since those yield rational constants.

Pos: 26:29:

File: HeapOffers.sol

Security**Check-effects-interaction:**

Potential violation of Checks-Effects-Interaction pattern in
AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 128:4:

Inline assembly:

The Contract uses inline assembly, this is only advised in rare cases. Additionally static analysis modules do not parse inline Assembly, this can lead to wrong analysis results.

[more](#)

Pos: 186:16:

Low level calls:

Use of "call": should be avoided whenever possible. It can lead to unexpected behavior if return value is not handled properly. Please use Direct Calls via specifying the called contract's interface.

[more](#)

Pos: 137:50:

Gas & Economy

Gas costs:

Gas requirement of function HeapOffers.getRoleAdmin is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 134:4:

Gas costs:

Gas requirement of function `HeapOffers.addAdminRole` is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 226:4:

For loop over dynamic array:

Loops that do not have a fixed number of iterations, for example, loops that depend on storage values, have to be used carefully. Due to the block gas limit, transactions can only consume a certain amount of gas. The number of iterations in a loop can grow beyond the block gas limit which can cause the complete contract to be stalled at a certain point. Additionally, using unbounded loops incurs in a lot of avoidable gas costs. Carefully test how many items at maximum you can pass to such functions to make it successful.

[more](#)

Pos: 124:8:

For loop over dynamic array:

Loops that do not have a fixed number of iterations, for example, loops that depend on storage values, have to be used carefully. Due to the block gas limit, transactions can only consume a certain amount of gas. The number of iterations in a loop can grow beyond the block gas limit which can cause the complete contract to be stalled at a certain point. Additionally, using unbounded loops incurs in a lot of avoidable gas costs. Carefully test how many items at maximum you can pass to such functions to make it successful.

[more](#)

Pos: 198:12:

Miscellaneous

Constant/View/Pure functions:

AccessControlUpgradeable.__AccessControl_init() : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 51:4:

Constant/View/Pure functions:

HeapOffers.getBestAssets(address,uint256,uint8) : Is constant but potentially should not be. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 160:4:

Similar variable names:

AccessControlUpgradeable.hasRole(bytes32,address) : Variables have very similar names "_roles" and "role". Note: Modifiers are currently not considered by this static analysis.

Pos: 91:15:

Similar variable names:

`HeapOffers.getBestAssets(address,uint256,uint8)` : Variables have very similar names "token" and "tokenId". Note: Modifiers are currently not considered by this static analysis.
Pos: 187:46:

No return:

`IAccessControlUpgradeable.hasRole(bytes32,address)`: Defines a return type but never explicitly returns a value.
Pos: 40:4:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

[more](#)

Pos: 162:8:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

[more](#)

Pos: 65:8:

Delete from dynamic array:

Using "delete" on an array leaves a gap. The length of the array remains the same. If you want to remove the empty position you need to shift items manually and update the "length" property.

[more](#)

Pos: 56:8:

Data truncated:

Division of integer values yields an integer value again. That means e.g. $10 / 100 = 0$ instead of 0.1 since the result is an integer again. This does not hold for division of (only) literal values since those yield rational constants.

Pos: 88:30:

Data truncated:

Division of integer values yields an integer value again. That means e.g. $10 / 100 = 0$ instead of 0.1 since the result is an integer again. This does not hold for division of (only) literal values since those yield rational constants.

Pos: 106:30:

File: HeapSales

Security

Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in `HeapSales.getBestAssets(address,uint256,uint8)`: Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 178:4:

Inline assembly:

The Contract uses inline assembly, this is only advised in rare cases. Additionally static analysis modules do not parse inline Assembly, this can lead to wrong analysis results.

[more](#)

Pos: 219:12:

Gas & Economy

Gas costs:

Gas requirement of function `HeapSales.store` is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 17:4:

For loop over dynamic array:

Loops that do not have a fixed number of iterations, for example, loops that depend on storage values, have to be used carefully. Due to the block gas limit, transactions can only consume a certain amount of gas. The number of iterations in a loop can grow beyond the block gas limit which can cause the complete contract to be stalled at a certain point. Additionally, using unbounded loops incurs in a lot of avoidable gas costs. Carefully test how many items at maximum you can pass to such functions to make it successful.

[more](#)

Pos: 142:8:

For loop over dynamic array:

Loops that do not have a fixed number of iterations, for example, loops that depend on storage values, have to be used carefully. Due to the block gas limit, transactions can only consume a certain amount of gas. The number of iterations in a loop can grow beyond the block gas limit which can cause the complete contract to be stalled at a certain point. Additionally, using unbounded loops incurs in a lot of avoidable gas costs. Carefully test how many items at maximum you can pass to such functions to make it successful.

[more](#)

Pos: 226:12:

For loop over dynamic array:

Loops that do not have a fixed number of iterations, for example, loops that depend on storage values, have to be used carefully. Due to the block gas limit, transactions can only consume a certain amount of gas. The number of iterations in a loop can grow beyond the block gas limit which can cause the complete contract to be stalled at a certain point. Additionally, using unbounded loops incurs in a lot of avoidable gas costs. Carefully test how many items at maximum you can pass to such functions to make it successful.

[more](#)

Pos: 251:8:

For loop over dynamic array:

Loops that do not have a fixed number of iterations, for example, loops that depend on storage values, have to be used carefully. Due to the block gas limit, transactions can only consume a certain amount of gas. The number of iterations in a loop can grow beyond the block gas limit which can cause the complete contract to be stalled at a certain point. Additionally, using unbounded loops incurs in a lot of avoidable gas costs. Carefully test how many items at maximum you can pass to such functions to make it successful.

[more](#)

Pos: 258:8:

Miscellaneous

Constant/View/Pure functions:

`HeapSales.initHeap(address,uint256)` : Potentially should be constant/view/pure but is not.
Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 29:4:

Similar variable names:

`HeapSales.getBestAssets(address,uint256,uint8)` : Variables have very similar names "token" and "tokenId". Note: Modifiers are currently not considered by this static analysis.

Pos: 209:61:

No return:

IStore.supportsInterface(bytes4): Defines a return type but never explicitly returns a value.

Pos: 92:4:

Delete from dynamic array:

Using "delete" on an array leaves a gap. The length of the array remains the same. If you want to remove the empty position you need to shift items manually and update the "length" property.

[more](#)

Pos: 61:8:

Data truncated:

Division of integer values yields an integer value again. That means e.g. $10 / 100 = 0$ instead of 0.1 since the result is an integer again. This does not hold for division of (only) literal values since those yield rational constants.

Pos: 106:30:

Data truncated:

Division of integer values yields an integer value again. That means e.g. $10 / 100 = 0$ instead of 0.1 since the result is an integer again. This does not hold for division of (only) literal values since those yield rational constants.

Pos: 124:30:

File: offers

Security

Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in Offers.createOffer(address,uint256,uint256): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 50:4:

Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in Offers.deleteOffer(uint256): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 83:4:

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.

[more](#)

Pos: 5:11:

Gas & Economy

Gas costs:

Gas requirement of function Offers.heapOffer is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 14:4:

Gas costs:

Gas requirement of function Offers.viewOffersByAsset is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 21:4:

For loop over dynamic array:

Loops that do not have a fixed number of iterations, for example, loops that depend on storage values, have to be used carefully. Due to the block gas limit, transactions can only consume a certain amount of gas. The number of iterations in a loop can grow beyond the block gas limit which can cause the complete contract to be stalled at a certain point.

Additionally, using unbounded loops incurs in a lot of avoidable gas costs. Carefully test how many items at maximum you can pass to such functions to make it successful.

[more](#)

Pos: 33:8:

For loop over dynamic array:

Loops that do not have a fixed number of iterations, for example, loops that depend on storage values, have to be used carefully. Due to the block gas limit, transactions can only consume a certain amount of gas. The number of iterations in a loop can grow beyond the block gas limit which can cause the complete contract to be stalled at a certain point. Additionally, using unbounded loops incurs in a lot of avoidable gas costs. Carefully test how many items at maximum you can pass to such functions to make it successful.

[more](#)

Pos: 70:8:

Miscellaneous

Constant/View/Pure functions:

`IHeap.deleteNode(address,uint256,uint256)` : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 27:4:

Constant/View/Pure functions:

`Offers.viewOffersByAsset(address,uint256,uint8)` : Is constant but potentially should not be. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 21:4:

No return:

IHeap.hasRole(bytes32,address): Defines a return type but never explicitly returns a value.

Pos: 54:4:

No return:

IHeap.supportsInterface(bytes4): Defines a return type but never explicitly returns a value.

Pos: 72:4:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

[more](#)

Pos: 85:8:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

[more](#)

Pos: 112:11:

Delete from dynamic array:

Using "delete" on an array leaves a gap. The length of the array remains the same. If you want to remove the empty position you need to shift items manually and update the "length" property.

[more](#)

Pos: 125:8:

Data truncated:

Division of integer values yields an integer value again. That means e.g. $10 / 100 = 0$ instead of 0.1 since the result is an integer again. This does not hold for division of (only) literal values since those yield rational constants.

Pos: 106:23:

File: OVRLand

Gas & Economy**Gas costs:**

Gas requirement of function OVRLand.addURLEditor is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 28:4:

Gas costs:

Gas requirement of function OVRLand.burn is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)
Pos: 106:4:

Gas costs:

Gas requirement of function OVRLand.batchBurn is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 116:4:

Gas costs:

Gas requirement of function OVRLand.mint is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)
Pos: 129:4:

Gas costs:

Gas requirement of function OVRLand.batchMintLands is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 143:4:

Gas costs:

Gas requirement of function OVRLand.batchMintLandsWithUri is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 159:4:

Gas costs:

Gas requirement of function OVRLand.batchSetOVRLandURI is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 179:4:

For loop over dynamic array:

Loops that do not have a fixed number of iterations, for example, loops that depend on storage values, have to be used carefully. Due to the block gas limit, transactions can only consume a certain amount of gas. The number of iterations in a loop can grow beyond the block gas limit which can cause the complete contract to be stalled at a certain point. Additionally, using unbounded loops incurs in a lot of avoidable gas costs. Carefully test how many items at maximum you can pass to such functions to make it successful.

[more](#)

Pos: 117:8:

For loop over dynamic array:

Loops that do not have a fixed number of iterations, for example, loops that depend on storage values, have to be used carefully. Due to the block gas limit, transactions can only consume a certain amount of gas. The number of iterations in a loop can grow beyond the block gas limit which can cause the complete contract to be stalled at a certain point. Additionally, using unbounded loops incurs in a lot of avoidable gas costs. Carefully test how many items at maximum you can pass to such functions to make it successful.

[more](#)

Pos: 148:8:

For loop over dynamic array:

Loops that do not have a fixed number of iterations, for example, loops that depend on storage values, have to be used carefully. Due to the block gas limit, transactions can only consume a certain amount of gas. The number of iterations in a loop can grow beyond the block gas limit which can cause the complete contract to be stalled at a certain point. Additionally, using unbounded loops incurs in a lot of avoidable gas costs. Carefully test how many items at maximum you can pass to such functions to make it successful.

[more](#)

Pos: 169:8:

For loop over dynamic array:

Loops that do not have a fixed number of iterations, for example, loops that depend on storage values, have to be used carefully. Due to the block gas limit, transactions can only consume a certain amount of gas. The number of iterations in a loop can grow beyond the block gas limit which can cause the complete contract to be stalled at a certain point. Additionally, using unbounded loops incurs in a lot of avoidable gas costs. Carefully test how many items at maximum you can pass to such functions to make it successful.

[more](#)

Pos: 184:8:

Miscellaneous**Constant/View/Pure functions:**

OVRLand._beforeTokenTransfer(address,address,uint256) : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 191:4:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

[more](#)

Pos: 164:8:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

[more](#)

Pos: 183:8:

File: Root

Inline assembly:

The Contract uses inline assembly, this is only advised in rare cases. Additionally static analysis modules do not parse inline Assembly, this can lead to wrong analysis results.

[more](#)

Pos: 84:8:

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.

[more](#)

Pos: 5:11:

Low level calls:

Use of "call": should be avoided whenever possible. It can lead to unexpected behavior if return value is not handled properly. Please use Direct Calls via specifying the called contract's interface.

[more](#)

Pos: 137:50:

Gas & Economy

Gas costs:

Gas requirement of function Allowed.getRoleAdmin is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 134:4:

Gas costs:

Gas requirement of function `Store.revokeRole` is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 165:4:

Gas costs:

Gas requirement of function `Allowed.renounceRole` is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 185:4:

Gas costs:

Gas requirement of function `Offers.heapOffer` is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 14:4:

Gas costs:

Gas requirement of function `Root.viewSalesByAsset` is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 26:4:

For loop over dynamic array:

Loops that do not have a fixed number of iterations, for example, loops that depend on storage values, have to be used carefully. Due to the block gas limit, transactions can only consume a certain amount of gas. The number of iterations in a loop can grow beyond the block gas limit which can cause the complete contract to be stalled at a certain point. Additionally, using unbounded loops incurs in a lot of avoidable gas costs. Carefully test how many items at maximum you can pass to such functions to make it successful.

[more](#)

Pos: 33:8:

For loop over dynamic array:

Loops that do not have a fixed number of iterations, for example, loops that depend on storage values, have to be used carefully. Due to the block gas limit, transactions can only consume a certain amount of gas. The number of iterations in a loop can grow beyond the block gas limit which can cause the complete contract to be stalled at a certain point. Additionally, using unbounded loops incurs in a lot of avoidable gas costs. Carefully test how many items at maximum you can pass to such functions to make it successful.

[more](#)

Pos: 44:8:

For loop over dynamic array:

Loops that do not have a fixed number of iterations, for example, loops that depend on storage values, have to be used carefully. Due to the block gas limit, transactions can only consume a certain amount of gas. The number of iterations in a loop can grow beyond the block gas limit which can cause the complete contract to be stalled at a certain point. Additionally, using unbounded loops incurs in a lot of avoidable gas costs. Carefully test how many items at maximum you can pass to such functions to make it successful.

[more](#)

Pos: 31:12:

For loop over dynamic array:

Loops that do not have a fixed number of iterations, for example, loops that depend on storage values, have to be used carefully. Due to the block gas limit, transactions can only consume a certain amount of gas. The number of iterations in a loop can grow beyond the block gas limit which can cause the complete contract to be stalled at a certain point. Additionally, using unbounded loops incurs in a lot of avoidable gas costs. Carefully test how many items at maximum you can pass to such functions to make it successful.

[more](#)

Pos: 70:8:

Similar variable names:

TokenUtils.transferERC721(address,address,address,uint256) : Variables have very similar names "to" and "id". Note: Modifiers are currently not considered by this static analysis.

Pos: 45:61:

No return:

IHeap.supportsInterface(bytes4): Defines a return type but never explicitly returns a value.

Pos: 72:4:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

[more](#)

Pos: 26:12:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

[more](#)

Pos: 28:12:

Delete from dynamic array:

Using "delete" on an array leaves a gap. The length of the array remains the same. If you want to remove the empty position you need to shift items manually and update the "length" property.

[more](#)

Pos: 172:8:

Delete from dynamic array:

Using "delete" on an array leaves a gap. The length of the array remains the same. If you want to remove the empty position you need to shift items manually and update the "length" property.

[more](#)

Pos: 91:8:

Data truncated:

Division of integer values yields an integer value again. That means e.g. $10 / 100 = 0$ instead of 0.1 since the result is an integer again. This does not hold for division of (only) literal values since those yield rational constants.

Pos: 203:28:

File: sales

Security**Check-effects-interaction:**

Potential violation of Checks-Effects-Interaction pattern in
`AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string)`: Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 128:4:

Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in Sales.buy(uint256,uint256): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 149:4:

Inline assembly:

The Contract uses inline assembly, this is only advised in rare cases. Additionally static analysis modules do not parse inline Assembly, this can lead to wrong analysis results.

[more](#)

Pos: 186:16:

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.

[more](#)

Pos: 5:11:

Low level calls:

Use of "call": should be avoided whenever possible. It can lead to unexpected behavior if return value is not handled properly. Please use Direct Calls via specifying the called contract's interface.

[more](#)

Pos: 63:27:

Low level calls:

Use of "call": should be avoided whenever possible. It can lead to unexpected behavior if return value is not handled properly. Please use Direct Calls via specifying the called contract's interface.

[more](#)

Pos: 137:50:

Gas & Economy**Gas costs:**

Gas requirement of function Allowed.getRoleAdmin is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 134:4:

For loop over dynamic array:

Loops that do not have a fixed number of iterations, for example, loops that depend on storage values, have to be used carefully. Due to the block gas limit, transactions can only consume a certain amount of gas. The number of iterations in a loop can grow beyond the block gas limit which can cause the complete contract to be stalled at a certain point. Additionally, using unbounded loops incurs in a lot of avoidable gas costs. Carefully test how many items at maximum you can pass to such functions to make it successful.

[more](#)

Pos: 156:8:

For loop over dynamic array:

Loops that do not have a fixed number of iterations, for example, loops that depend on storage values, have to be used carefully. Due to the block gas limit, transactions can only consume a certain amount of gas. The number of iterations in a loop can grow beyond the block gas limit which can cause the complete contract to be stalled at a certain point. Additionally, using unbounded loops incurs in a lot of avoidable gas costs. Carefully test how many items at maximum you can pass to such functions to make it successful.

[more](#)

Pos: 173:12:

For loop over dynamic array:

Loops that do not have a fixed number of iterations, for example, loops that depend on storage values, have to be used carefully. Due to the block gas limit, transactions can only consume a certain amount of gas. The number of iterations in a loop can grow beyond the block gas limit which can cause the complete contract to be stalled at a certain point. Additionally, using unbounded loops incurs in a lot of avoidable gas costs. Carefully test how many items at maximum you can pass to such functions to make it successful.

[more](#)

Pos: 70:8:

Miscellaneous

Constant/View/Pure functions:

AccessControlUpgradeable.__AccessControl_init() : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 51:4:

Similar variable names:

TokenUtils.transferERC721(address,address,address,uint256) : Variables have very similar names "to" and "id". Note: Modifiers are currently not considered by this static analysis.
Pos: 45:57:

No return:

IHeap.hasRole(bytes32,address): Defines a return type but never explicitly returns a value.
Pos: 54:4:

No return:

IHeap.supportsInterface(bytes4): Defines a return type but never explicitly returns a value.
Pos: 72:4:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

[more](#)

Pos: 26:12:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

[more](#)

Pos: 28:12:

Delete from dynamic array:

Using "delete" on an array leaves a gap. The length of the array remains the same. If you want to remove the empty position you need to shift items manually and update the "length" property.

[more](#)

Pos: 132:12:

Data truncated:

Division of integer values yields an integer value again. That means e.g. $10 / 100 = 0$ instead of 0.1 since the result is an integer again. This does not hold for division of (only) literal values since those yield rational constants.

Pos: 203:28:

File: store

Security

Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in `AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string)`: Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 128:4:

Inline assembly:

The Contract uses inline assembly, this is only advised in rare cases. Additionally static analysis modules do not parse inline Assembly, this can lead to wrong analysis results.

[more](#)

Pos: 186:16:

Low level calls:

Use of "call": should be avoided whenever possible. It can lead to unexpected behavior if return value is not handled properly. Please use Direct Calls via specifying the called contract's interface.

[more](#)

Pos: 137:50:

Gas & Economy**Gas costs:**

Gas requirement of function Allowed.getRoleAdmin is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 134:4:

Gas costs:

Gas requirement of function Store.renounceRole is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 185:4:

Miscellaneous**Constant/View/Pure functions:**

AccessControlUpgradeable.__AccessControl_init() : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 51:4:

Constant/View/Pure functions:

ERC165Upgradeable.supportsInterface(bytes4) : Is constant but potentially should not be.

Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 32:4:

Similar variable names:

AccessControlUpgradeable.hasRole(bytes32,address) : Variables have very similar names

"_roles" and "role". Note: Modifiers are currently not considered by this static analysis.

Pos: 91:15:

No return:

IAccessControlUpgradeable.getRoleAdmin(bytes32): Defines a return type but never explicitly returns a value.

Pos: 48:4:

No return:

IERC165Upgradeable.supportsInterface(bytes4): Defines a return type but never explicitly returns a value.

Pos: 24:4:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

[more](#)

Pos: 65:8:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

[more](#)

Pos: 19:8:

File: Token

Gas & Economy

Gas costs:

Gas requirement of function ERC20.name is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 62:4:

Gas costs:

Gas requirement of function ERC20.decreaseAllowance is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 201:4:

Gas costs:

Gas requirement of function Token.decreaseAllowance is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 201:4:

Miscellaneous**Constant/View/Pure functions:**

ERC20._beforeTokenTransfer(address,address,uint256) : Potentially should be constant/view/pure but is not.

[more](#)

Pos: 358:4:

Similar variable names:

ERC20._burn(address,uint256) : Variables have very similar names "account" and "amount".

Pos: 294:49:

No return:

IERC20Metadata.decimals(): Defines a return type but never explicitly returns a value.

Pos: 27:4:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

[more](#)

Pos: 316:8:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

[more](#)

Pos: 337:12:

File: TokenUtils

Gas & Economy

For loop over dynamic array:

Loops that do not have a fixed number of iterations, for example, loops that depend on storage values, have to be used carefully. Due to the block gas limit, transactions can only consume a certain amount of gas. The number of iterations in a loop can grow beyond the block gas limit which can cause the complete contract to be stalled at a certain point. Additionally, using unbounded loops incurs in a lot of avoidable gas costs. Carefully test how many items at maximum you can pass to such functions to make it successful.

[more](#)

Pos: 70:8:

Miscellaneous

Similar variable names:

TokenUtils.transferERC1155(address,address,address,uint256,uint256) : Variables have very similar names "to" and "id". Note: Modifiers are currently not considered by this static analysis.

Pos: 36:58:

Similar variable names:

TokenUtils.transferERC1155(address,address,address,uint256,uint256) : Variables have very similar names "to" and "id". Note: Modifiers are currently not considered by this static analysis.

Pos: 36:62:

Similar variable names:

TokenUtils.transferERC721(address,address,address,uint256) : Variables have very similar names "to" and "id". Note: Modifiers are currently not considered by this static analysis.

Pos: 45:57:

Similar variable names:

TokenUtils.transferERC721(address,address,address,uint256) : Variables have very similar names "to" and "id". Note: Modifiers are currently not considered by this static analysis.

Pos: 45:61:

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to lost tokens etc.
High	High level vulnerabilities are difficult to exploit; however, they also have a significant impact on smart contract execution, e.g. public access to crucial functions.
Medium	Medium level vulnerabilities are important to fix; however, they cannot lead to lost tokens.
Low	Low level vulnerabilities are most related to outdated, unused etc. These code snippets cannot have a significant impact on execution.
Lowest Code Style/ Best Practice	Lowest level vulnerabilities, code style violations and information statements cannot 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:

No low severity vulnerabilities were found.

Very Low:

Discussion

Contract is created on the grounds of upgradeability. Our audit report will not be applicable on the upgrade part incase if it is applied in future.

Conclusion

We were given a contract file and have used all possible tests based on the given object. So it is now ready for mainnet deployment. We have used all the latest static tools and manual observations to cover maximum possible test cases to scan everything.

The security state of the reviewed contract is "**well-Secured**".

Note For Contract Users

There are several owner only functions. Those can be called by the owner's wallet only. So, if the owner's wallet is compromised, then it carries the risk of the contract becoming vulnerable.

addAllowedaddresses: Owner can add allowed addresses.

Pause/Unpause: Owner can pause or unpause

deleteNode: Owner can delete node.

UpdateNode: Owner can update node.

addStore: Owner can set store address.

addAdminRole: Owner can add admin role.

addURIEditor: Owner can add URIEditor.

removeURIEditor: Owner can remove URIEditor.

addMinter: Owner can add minter.

removeMinter: Owner can remove minter.

addBurner: Owner can add burner.

removeBurner: Owner can remove burner.

addAdminRole: Owner can add admin role.

removeAdminRole: Owner can remove admin role.

SafeMint: Owner has power to handle minting..

SetOVRLandURI: Owner has power to set OVRLand and URI.

Burn: This function is for handling burns.

Batchburn: This function is for handling batch burning tokens.

Mint: This function is for minting.

batchMintLandsWithUri: Owner has power to batch minting tokens with URI

batchSetOVRLandURI: Owner has power to set the OVRLand URI in both.

Owner has full control over the smart contract. Thus, technical auditing does not guarantee the project's ethical side.

Please do your due diligence before investing. Our audit report is never an investment advice.

Our Methodology

We like to work with a transparent process and make our reviews a collaborative effort. The goals of our security audits are to improve the quality of systems we review and aim for sufficient remediation to help protect users. The following is the methodology we use in our security audit process.

Manual Code Review

In manually reviewing all of the code, we look for any potential issues with code logic, error handling, protocol and header parsing, cryptographic errors, and random number generators. We also watch for areas where more defensive programming could reduce the risk of future mistakes and speed up future audits. Although our primary focus is on the in-scope code, we examine dependency code and behavior when it is relevant to a particular line of investigation.

Vulnerability Analysis

Our audit techniques included manual code analysis, user interface interaction, and whitebox penetration testing. We look at the project's web site to get a high level understanding of what functionality the software under review provides. We then meet with the developers to gain an appreciation of their vision of the software. We install and use the relevant software, exploring the user interactions and roles. While we do this, we brainstorm threat models and attack surfaces. We read design documentation, review other audit results, search for similar projects, examine source code dependencies, skim open issue tickets, and generally investigate details other than the implementation.

Documenting Results

We follow a conservative, transparent process for analyzing potential security vulnerabilities and seeing them through successful remediation. Whenever a potential issue is discovered, we immediately create an Issue entry for it in this document, even though we have not yet verified the feasibility and impact of the issue. This process is conservative because we document our suspicions early even if they are later shown to not represent exploitable vulnerabilities. We generally follow a process of first documenting the suspicion with unresolved questions, then confirming the issue through code analysis, live experimentation, or automated tests. Code analysis is the most tentative, and we strive to provide test code, log captures, or screenshots demonstrating our confirmation. After this we analyse the feasibility of an attack in a live system.

Suggested Solutions

We search for immediate mitigations that live deployments can take, and finally we suggest the requirements for remediation engineering for future releases. The mitigation and remediation recommendations should be scrutinised by the developers and deployment engineers, and successful mitigation and remediation is an ongoing collaborative process after we deliver our report, and before the details are made public.

Disclaimers

RD Auditors Disclaimer

The smart contracts given for audit have been analysed in accordance with the best industry practices at the date of this report, in relation to: cybersecurity vulnerabilities and issues in smart contract source code, the details of which are disclosed in this report, (Source Code); the Source Code compilation, deployment and functionality (performing the intended functions).

Because the total number of test cases are unlimited, the audit makes no statements or warranties on the security of the code. It also cannot be considered as a sufficient assessment regarding the utility and safety of the code, bugfree status or any other statements of the contract. While we have done our best in conducting the analysis and producing this report, it is important to note that you should not rely on this report only - we recommend proceeding with several independent audits and a public bug bounty program to ensure security of smart contracts.

Technical Disclaimer

Smart contracts are deployed and executed on the blockchain. The platform, its programming language, and other software related to the smart contract can have their own vulnerabilities that can lead to hacks. Thus, the audit can't guarantee explicit security of the audited smart contracts.



Email: info@rdauditors.com

Website: www.rdauditors.com

