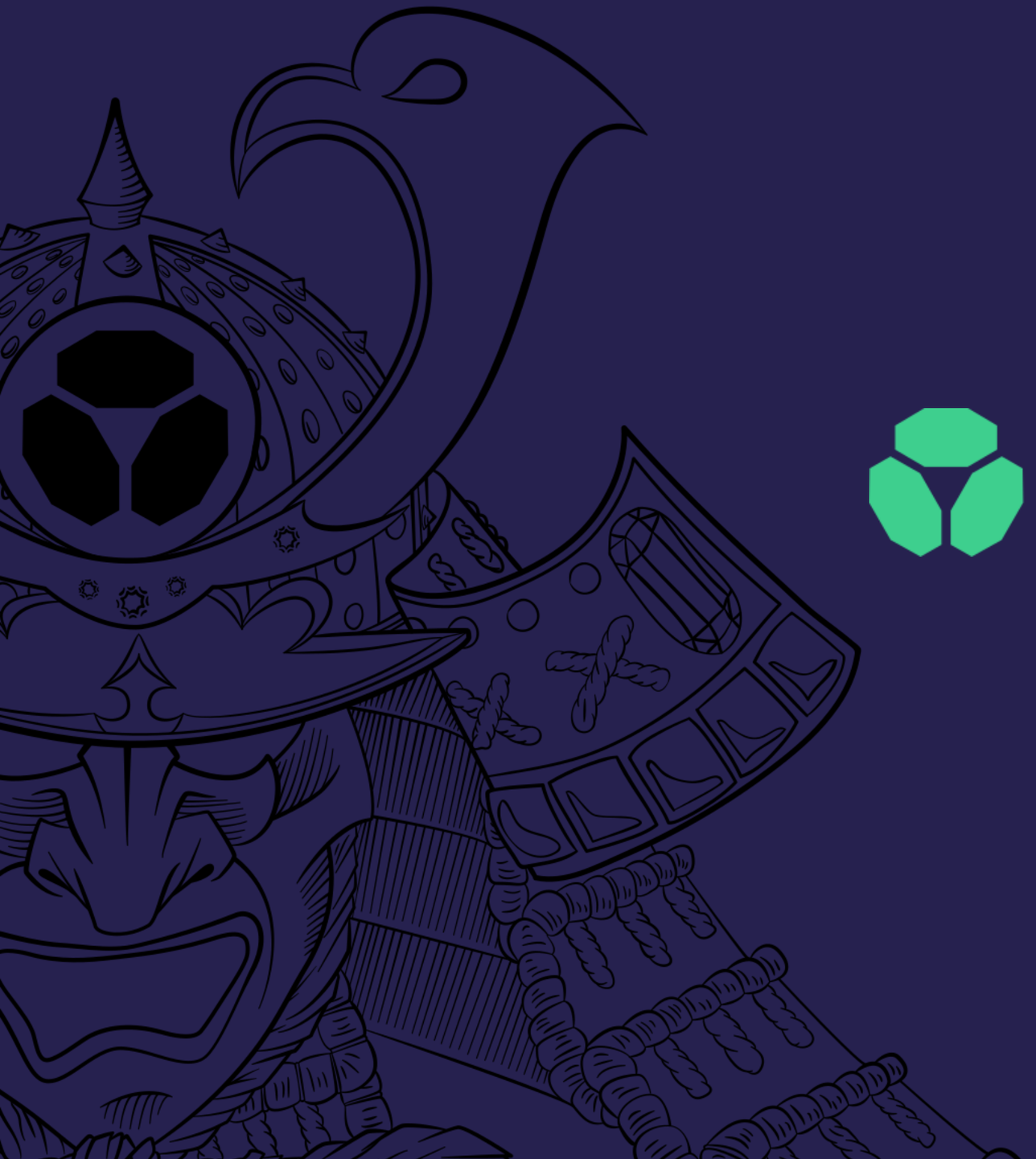


smart state

> Smart
Contract
Audit #


Feb 05
2022

[redacted]

# TABLE OF CONTENTS

Table of contents ..... 3
Methodology ..... 4
Stucture of contact Auction.sol ..... 5
Stucture of contact AuctionFactory.sol ..... 8
Stucture of contact Marketplace.sol ..... 10
Stucture of contact MyToken.sol ..... 11
Stucture of contact Staking.sol ..... 12
Stucture of contact marketplaceHelpers/Cashback.sol ..... 14
Stucture of contact marketplaceHelpers/ManageCollections.sol ..... 15
Stucture of contact stakingHelpers/Rewards.sol ..... 17
Stucture of contact stakingHelpers/SetBonus.sol ..... 19
Verification check sums ..... 21

# METHODOLOGY

## MAIN TESTS LIST:

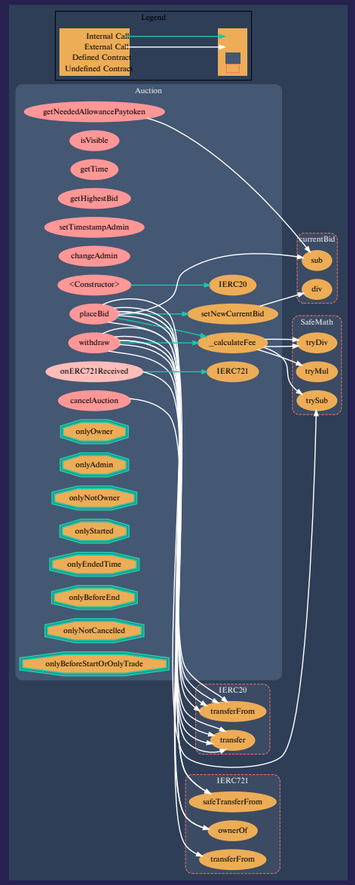
- Best code practices
- ERC20/BEP20 compliance (if applicable)
- FA2 compliance (if applicable)
- Logical bugs
- General Denial Of Service(DOS)
- Locked ether
- Private data leaks
- Using components with known vulns
- Weak PRNG
- Unsed vars
- Uncheked call return method
- Code with no effects
- Pool Asset Security (backdoors in the underlying ERC-20)
- Function visibility
- Use of deprecated functions
- Authorization issues
- Re-entrancy
- Arithmetic Over/Under Flows
- Hidden Malicious Code
- External Contract Referencing
- Short Address/ Parameter Attack
- Race Conditions / Front Running
- Uninitialized Storage Pointers
- Floating Points and Precision
- Signatures Replay

# STRUCTURE OF CONTRACT

## AUCTION.SOL

## CONTRACT METHODS ANALYSIS:

- isVisible()
  Function can be declared external
- getTime()
  Function can be declared external
- getHighestBid()
  Function can be declared external
- setTimestampAdmin(uint256 _endTimestamp)
  Function should emit an event. Function can be declared external




Pic. 1.1
Auction.sol

- changeAdmin(address newAdminAddress)
  Function should emit an event. Function can be declared external. Function should check for 0 address

- setNewCurrentBid()
  Vulnerabilities not detected

- getNeededAllowancePaytoken()
  Function can be declared external

**PAYABLE/WARNING**

- placeBid()
  Since er721Instance is fully controlled by user a malicious contract can be provided to trigger re-entrancy at line 161. This attack can be used to trigger calls at lines 157,158,159 and charge user multiple times for signle trade. NFT should be transfered before sending payment tokens and also nonReentrant modifier should be added for security.

Tokens in

- cancelAuction()
  Function can be declared external

**PAYABLE/WARNING**

- _calculateFee(uint amount)
  Should process boolean values, that SafeMath methods return

NFT out

**PAYABLE**

- withdraw()
  Function should check that auction type is Type.Auction. Check at line 199 is redundant. Typo in error at line 229. Function can be declared external

NFT out, tokens out

## PAYABLE/WARNING

- onERC721Received(
  address _operator,
  address _from,
  uint256 _tokenId,
  bytes calldata _data
  )

Function can be called by anyone, which means any user can provide a non valid er721 and ruin a deployed contract. We suggest that erc721 address and token Id should be passed in contract's constructor and verified in this method. Also a good solution is to transfer an NFT in the same tx when creating auction

NFT in

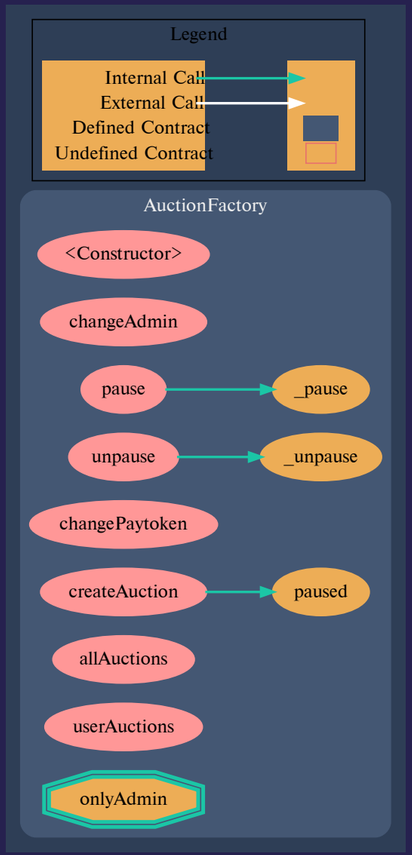
# STRUCTURE OF CONTRACT

## AUCTIONFACTORY.SOL

### CONTRACT METHODS ANALYSIS:

- changeAdmin(address newAdminAddress)
  Function should emit an event.
  Function can be declared external.
  Function should check for 0 address

- changePaytoken(address newPaytokenAddress)
  Function should emit an event.
  Function can be declared external.
  Function should check for 0 address




Pic. 1.2
AuctionFactory.sol

- createAuction(
  uint256 duration,
  uint256 buyValue,
  Type _type
  )

  Function can be optimized via cloning contracts instead of creating a new one with contstructor. See https://blog.openzeppelin.com/workshop-recap-cheap-contract-deployment-through-clones/. Function can be declared external

- allAuctions()

  Function can be declared external

- userAuctions(address userAddress)

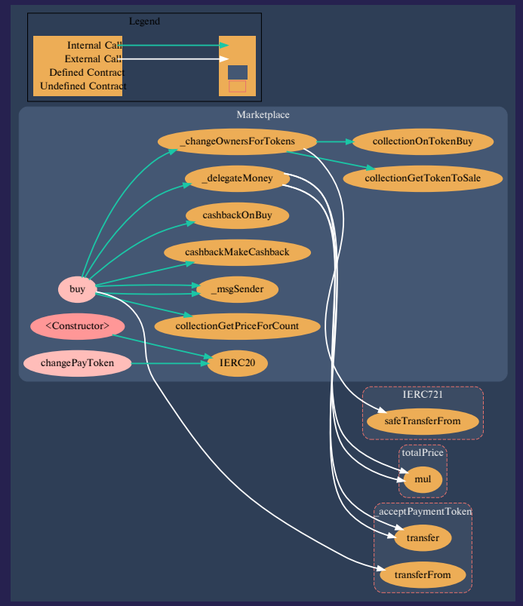
  Function can be declared external

# STRUCTURE OF CONTRACT

## MARKETPLACE.SOL

### CONTRACT METHODS ANALYSIS:

- buy(uint256 collectionIndex, uint256 count)
  Vulnerabilities not detected
- _changeOwnersForTokens(uint256 collectionIndex, address buyer, uint256 count)
  Vulnerabilities not detected
- _delegateMoney(uint256 totalPrice, uint256 unspentAmount)
  Vulnerabilities not detected
- _delegateMoney(uint256 totalPrice, uint256 unspentAmount)
  Vulnerabilities not detected
- changePayToken(address newAddress)
  Vulnerabilities not detected



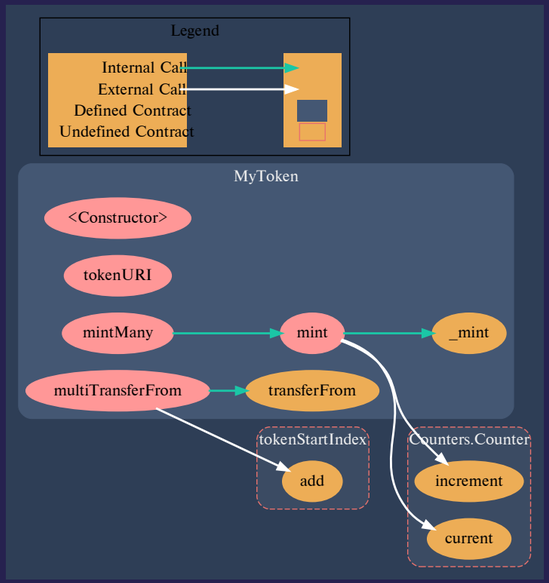

Pic. 1.3
Marketplace.sol

# STRUCTURE OF CONTRACT

## MYTOKEN.SOL

### CONTRACT METHODS ANALYSIS:

- tokenURI()
  Function can be declared external
- mint(address toAddress)
  Function can be declared external
- mintMany(address toAddress, uint256 tokenCount)
  Function can be declared external
- multiTransferFrom(address from, address[] memory adressess, uint256 tokenCount, uint256 tokenStartIndex)
  Function can be declared external



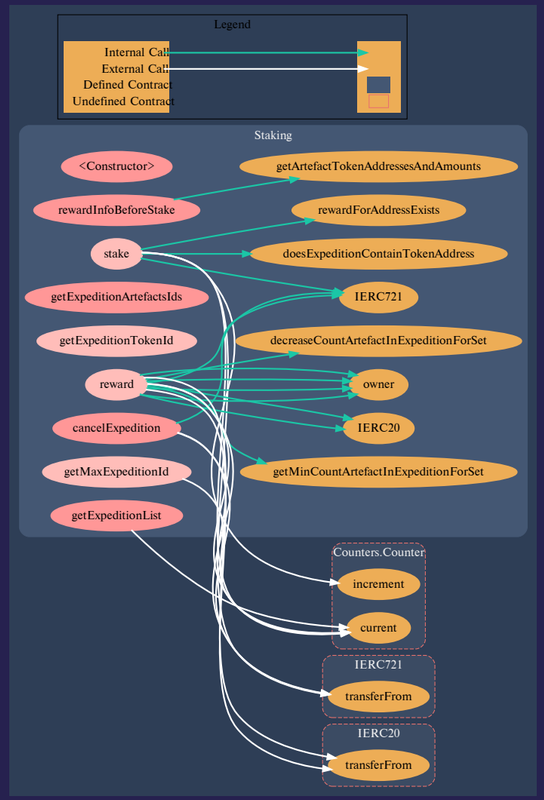

Pic. 1.4
MyToken.sol

# STRUCTURE OF CONTRACT

## STAKING.SOL

## CONTRACT METHODS ANALYSIS:

- getMaxExpeditionId()
  Vulnerabilities  not detected
- getExpeditionList()
  Function can be declared external
- getExpeditionArtefactsIds(uint256 expeditionId, address tokenAddress)
  Function can be declared external
- getExpeditionTokenId(uint256 expeditionId, address tokenAddress, uint256 index)
  Vulnerabilities  not detected




Pic. 1.5
Staking.sol

- rewardInfoBeforeStake(address artifactAddress)
  Function can be declared external

**PAYABLE/WARNING**

- stake(OneArtefact[] calldata inputArtefacts)
  Function can be optimized via combining for loops together

NFT in

**PAYABLE/WARNING**

- cancelExpedition(uint256 expeditionId)
  Function should emit an event. Function should be declared external

NFT out

**PAYABLE/WARNING**

- reward(uint256 expeditionId)
  Function should emit an event

Tokens out, NFT out

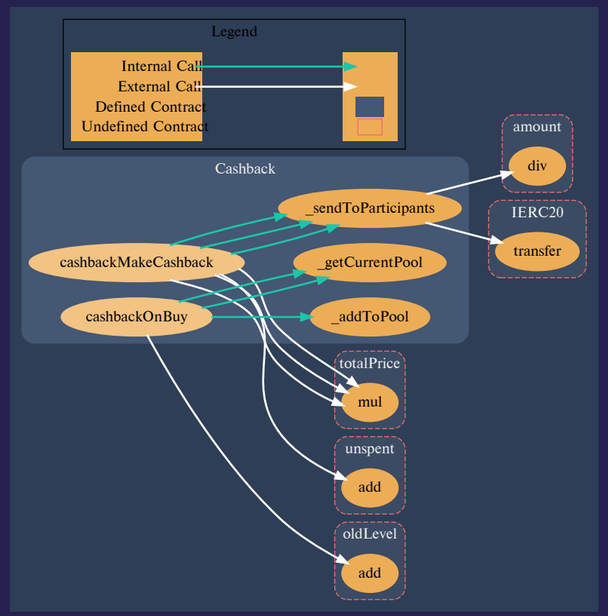
- doesExpeditionContainTokenAddress(uint256 expeditionId, address tokenAddress)
  Vulnerabilities not detected
- getMinCountArtefactInExpeditionForSet(uint256 expeditionId, uint256 setId)
  Vulnerabilities not detected
- decreaseCountArtefactInExpeditionForSet(uint256 expeditionId, uint256 setId, uint256 decreaseCount)
  Vulnerabilities not detected

# STRUCTURE OF CONTRACT

## MARKETPLACEHELP-ERS/CASHBACK.SOL

## CONTRACT METHODS ANALYSIS:

- cashbackMakeCashback(uint256 totalPrice)
  Vulnerabilities not detected
- cashbackOnBuy(address buyer, uint256 totalPrice)
  Vulnerabilities not detected
- _addToPool(uint256 poolIndex, address buyer)
  Vulnerabilities not detected
- _getCurrentPool(address buyer)
  Vulnerabilities not detected
- _sendToParticipants(address[] memory participants, uint256 amount)
  User pays too much fees due to «for» loop. We suggest that function should be optimized so users can claim their cash back by themselves instead of paying for others.



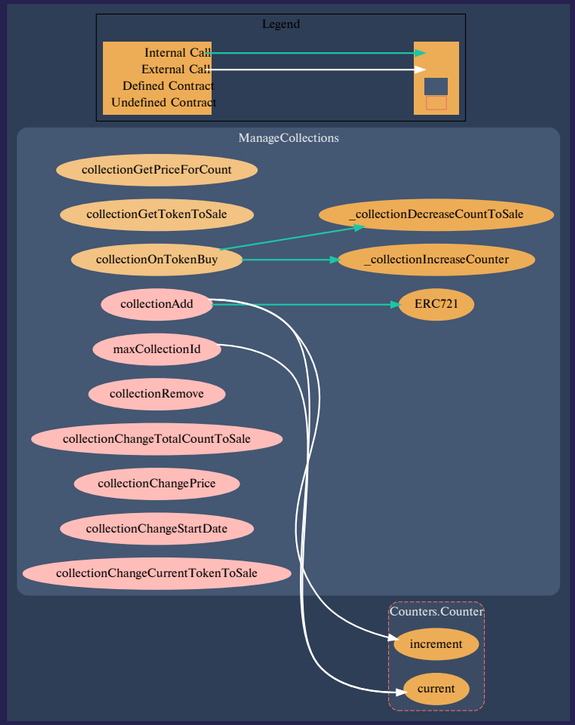

Pic. 1.6
marketplaceHelpers/Cashback.sol

# STRUCTURE OF CONTRACT

MARKETPLACEHELPERS/ MANAGECOLLECTIONS. SOL

## CONTRACT METHODS ANALYSIS:

- collectionGetPriceForCount(uint256 collectionIndex, uint256 count)
  Vulnerabilities not detected
- collectionGetTokenToSale(uint256 collectionIndex)
  Vulnerabilities not detected
- collectionOnTokenBuy(uint256 collectionIndex)
  Vulnerabilities not detected
- _collectionIncreaseCounter(uint256 collectionIndex)
  Vulnerabilities not detected
- _collectionDecreaseCountToSale(uint256 collectionIndex)
  Vulnerabilities not detected




Pic. 1.7
marketplaceHelpers/ ManageCollections.sol

- collectionAdd(
  address NFTAddress,
  address NFTOwner,
  uint256 priceInDollars,
  uint256 timestampStartOfSale,
  uint256 countToSale,
  uint256 startIndex
  )

  We suggest that this function should check for NFTOwner allowance of NFTs to this SC address. So buy function in marketplace won't revert. Function should emit an event.

- maxCollectionId()
  Vulnerabilities not detected

- collectionRemove(uint256 collectionIndex)
  Vulnerabilities not detected

- collectionChangeTotalCountToSale(uint256 collectionIndex, uint256 totalCountToSale)
  Vulnerabilities not detected

- collectionChangePrice(uint256 collectionIndex, uint256 price)
  Vulnerabilities not detected

- collectionChangeStartDate(uint256 collectionIndex, uint256 timestamp)
  Colletion start date is not utilized by Marketplace SC

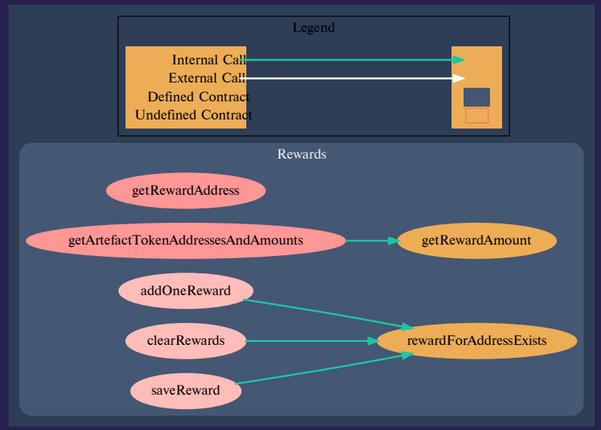
- ollectionChangeCurrentTokenToSale(uint256 collectionIndex, uint256 currentTokenToSale)
  Vulnerabilities not detected

# STRUCTURE OF CONTRACT

## STAKINGHELPERS/REWARDS.SOL

## CONTRACT METHODS ANALYSIS:

- getRewardAddress(address artefactAddress, uint256 index)
  Vulnerabilities not detected
- getRewardAmount(address artefactAddress, address tokenAddress)
  Vulnerabilities not detected
- getArtefactTokenAddressesAndAmounts(address artifactAddress)
  Vulnerabilities not detected
- ewardForAddressExists(address artefactAddress)
  Vulnerabilities not detected




Pic. 1.8
stakingHelpers/Rewards.sol

- addOneReward(address artefactAddress, address rewardAddress, uint256 rewardAmount)
  Function should emit an event

- clearRewards(address artefactAddress)
  Function should emit an event

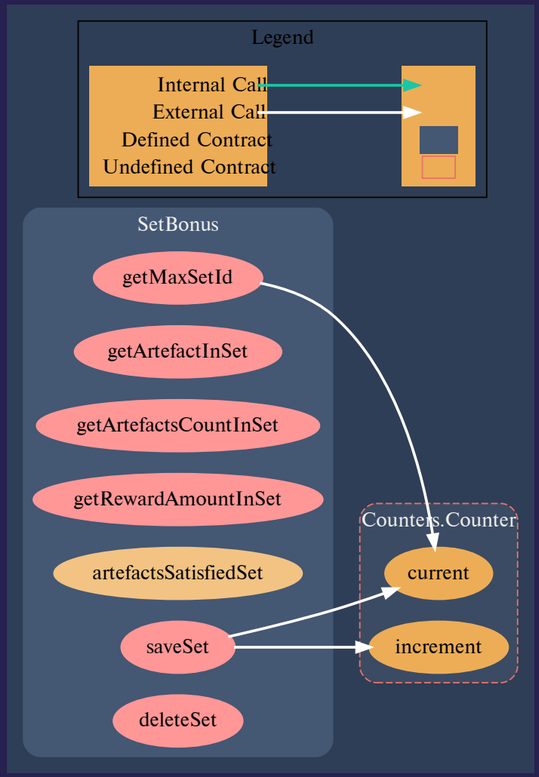
- saveReward(address artefactAddress, OneReward[] memory inputRewards)
  Vulnerabilities not detected

# STRUCTURE OF CONTRACT

## STAKINGHELPERS/ SETBONUS.SOL

## CONTRACT METHODS ANALYSIS:

- getMaxSetId()
  Vulnerabilities not detected
- getArtefactInSet(uint256 setId, uint256 artefactIndex)
  Vulnerabilities not detected
- getArtefactsCountInSet(uint256 setId)
  Vulnerabilities not detected
- getRewardAmountInSet(uint256 setId, address resourceAddress)
  Vulnerabilities not detected




Pic. 1.9
stakingHelpers/ SetBonus.sol

- artefactsSatisfiedSet(address[] memory userArtefacts, uint256 setId)
  Vulnerabilities  not detected

- saveSet(address[] memory artefactAddresses, OneReward[] memory inputRewards)
  Function should emit an event

- deleteSet(uint256 setId)
  Function should emit an event

# VERIFICATION CHECK SUMS

| Contract Name | Solc version | Optimisation | Bytecode hash (SHA 256) |
|---|---|---|---|
| Auction.sol | 0.8.0 | 200 | ed72440ed029426d462ae2aaad9762cdad18a130bb61f73101bd5f78925b0b99 |
| AuctionFactory.sol | 0.8.0 | 200 | 1e3400527e6a7544fc7bc4d58f49cea2fc45ee660f71c56da07c2c9df37e38a1 |
| Marketplace.sol | 0.8.0 | 200 | a91ef6ffa46f928473605b648688a2ca78489f84dd9ac220979ca745b93068f1 |
| MyToken.sol | 0.8.0 | 200 | 8dcedfa5bc1186db84693f1b369701a0b4a5739fb0dbc7cbc44e46f977132692 |

| Contract Name | Solc version | Optimisation | Bytecode hash (SHA 256) |
|---|---|---|---|
| Staking.sol | 0.8.0 | 200 | f6db3408cb4c396cf52333f76930b9eb6d1b7efb5ddae7c222bd11ef970cae8b |

# Get In Touch

info@smartstate.tech

smartstate.tech