Smart Contract
Security Audit
V1

Crypto Yachts NFT Smart Contract

7/3/2022

https://saferico.com/

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

Table of Contents

Background

The purpose of the audit was to achieve the following:

- Ensure that the smart contract functions as intended.
- Identify potential security issues with the smart contract.

The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Project Information

- **Platform**: Ethereum
- **Contract Address**: 0xa0d846a22dF80b30566107540c3d9B3EDAb15ac4
- **Code:**

https://rinkeby.etherscan.io/address/0xa0d846a22df80b30566107540c3d9b3edab15ac4#code

NFT Information

- Name: YACHT
- MAX Supply:
- Holders:
- Total transactions:

Contracts address deployed to test net (Ethereum)

Crypto Yachts NFT contract on ETH test net to test every function by the auditor.

https://rinkeby.etherscan.io/address/0xa0d846a22df80b30566107540c3d9b3edab15ac4

https://rinkeby.etherscan.io/address/0x4f29046e1248fe1e654330c5cdea1baa4815c719

Executive Summary

According to our assessment, the customer`s solidity smart contract is **Well-Secured**. Because the team fix all high and low issues.

Well Secured	✓
Secured	
Poor Secured	
Insecure	

Automated checks are with remix IDE. All issues were performed by the team, which included the analysis of code functionality, manual audit found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the audit overview section. The general overview is presented in the Project Information section and all issues found are located in the audit overview section.

Team found 0 critical, 1 high, 0 medium, 2 low, 0 very low-level issues and 0 notes in all solidity files of the contract

The files:

CryptoYachts.sol

File and Function Level Report

File in Scope:

Contract Name	SHA 256 hash	Contract Address
CryptoYachts.sol	**328bcc19378dd6296b5f4e365733b327ed8e2fe8963fcf2e56264167d70d9b12**	0xa0d846a22dF80b30566107540c3d9B3EDAb15ac4

- Contract: CryptoYachts
- Inherit: ERC721A, Ownable
- Observation: All passed including security check
- Test Report: passed
- Score: passed
- Conclusion: passed

Function	Test Result	Type / Return Type	Score
name	✓	Read / public	**Passed**
symbol	✓	Read / public	**Passed**
cost	✓	Read / public	**Passed**
supportsInterface	✓	Read / public	**Passed**
merkleRoot	✓	Read / public	**Passed**
balanceOf	✓	Read / public	**Passed**
Owner	✓	Read / public	**Passed**
whitelistUsed	✓	Read / public	**Passed**
walletOfOwner	✓	Read / public	**Passed**
getApprovedForAll	✓	Read / public	**Passed**
verify	✓	Read / public	**Passed**
getApproved	✓	Read / public	**Passed**

ownerOf	✓	Read / public	**Passed**
tokenURI	✓	Read / public	**Passed**
totalSupply	✓	Read / public	**Passed**
whitelistRemaining	✓	Read / public	**Passed**
stagesMaxTokensPerAddress	✓	Read / public	**Passed**
tokenPrice	✓	Read / public	**Passed**
stagesTokenPrice	✓	Read / public	**Passed**
currentStage	✓	Read / public	**Passed**
FOR_SALE_TOKENS	✓	Read / public	**Passed**
getContractInfo	✓	Read / public	**Passed**
isPublicSale	✓	Read / public	**Passed**
purchased	✓	Read / public	**Passed**
RESERVED_TOKENS	✓	Read / public	**Passed**
soldAmount	✓	Read / public	**Passed**
stageCount	✓	Read / public	**Passed**
whitlistMint	✓	Write / payable	**Passed**
approve	✓	Write / public	**Passed**
safeTransferFrom	✓	Write / public	**Passed**
safeTransferFrom	✓	Write / public	**Passed**
setCurrentStage	✓	Write / public	**Passed**
setMaxTokensPerAddress	✓	Write / public	**Passed**
mint	✓	Write / payable	**Passed**
setPrice	✓	Write / public	**Passed**
transferOwnership	✓	Write / public	**Passed**
setApprovalForAll	✓	Write / public	**Passed**
transferFrom	✓	Write / public	**Passed**

withdraw	✓	Write / public	**Passed**
setMarketRoot	✓	Write / public	**Passed**
renounceOwnership	✓	Write / public	**Passed**
setBaseURI	✓	Write / public	**Passed**
addStage	✓	Write / public	**Passed**

Issues Checking Status

No.	Issue Description	Checking Status
1	Compiler warnings.	**Passed**
2	Race conditions and Reentrancy. Cross-function race conditions.	**Passed**
3	Possible delays in data delivery.	**Passed**
4	Oracle calls.	**Passed**
5	Design Logic.	**Passed**
6	Timestamp dependence.	**Passed**
7	Integer Overflow and Underflow.	**Passed**
8	DoS with Revert.	**Passed**
9	DoS with block gas limit.	**Passed with Notes**
10	Methods execution permissions.	**Passed**
11	Economy model. If application logic is based on an incorrect economic model, the application would not function correctly and participants would incur financial losses. This type of issue is most often found in bonus rewards systems, Staking and Farming contracts, Vault and Vesting contracts, etc.	**Passed**
12	The impact of the exchange rate on the logic.	**Passed**
13	Private user data leaks.	**Passed**
14	Malicious Event log.	**Passed**
15	Scoping and Declarations.	**Passed**
16	Uninitialized storage pointers.	**Passed**
17	Arithmetic accuracy.	**Passed**

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to tokens loss etc.
High	High-level vulnerabilities are difficult to exploit; however, they also have significant impact on smart contract execution, e.g. public access to crucial functions
Medium	Medium-level vulnerabilities are important to fix; however, they can’t lead to tokens lose
Low	Low-level vulnerabilities are mostly related to outdated, unused etc. code snippets, that can’t have significant impact on execution
Note	Lowest-level vulnerabilities, code style violations and info statements can’t affect smart contract execution and can be ignored.

Audit Findings

Critical:

No critical severity vulnerabilities were found.

High:

#Contract code size exceeds 24576 bytes

Description
Contract implementation is too large in size to be deployed on mainnet. Ethereum with its spurious dragon release limited the size of the contracts deployable on mainnet to 24576 bytes.
The size of the contract CryptoYachts.sol goes way above this value and currently is of size 44581 bytes.

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

Status: Closed. Fixed in version 2.

Medium:

No Medium severity vulnerabilities were found

Low:

#Pragam version not fixed

Description
It is a good practice to lock the solidity version for a live deployment (use 0.8.12 instead of ^0.8.12). contracts should be deployed with the same compiler version and flags that they have been tested the most with. Locking the pragma helps ensure that contracts do not accidentally get deployed using, for example, the latest compiler which may have higher risks of undiscovered bugs. Contracts may also be deployed by others and the pragma indicates the compiler version intended by the original authors.

Remediation
Remove the ^ sign to lock the pragma version.

Status: Closed. Fixed in version 2.

#Use of block.timestamp for comparisons

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

Remediation
Avoid use of block.timestamp

Status: Acknowledged

Very Low:

No Very Low severity vulnerabilities were found.

Notes:

No Notes were found

Automatic Testing

1- Check for security

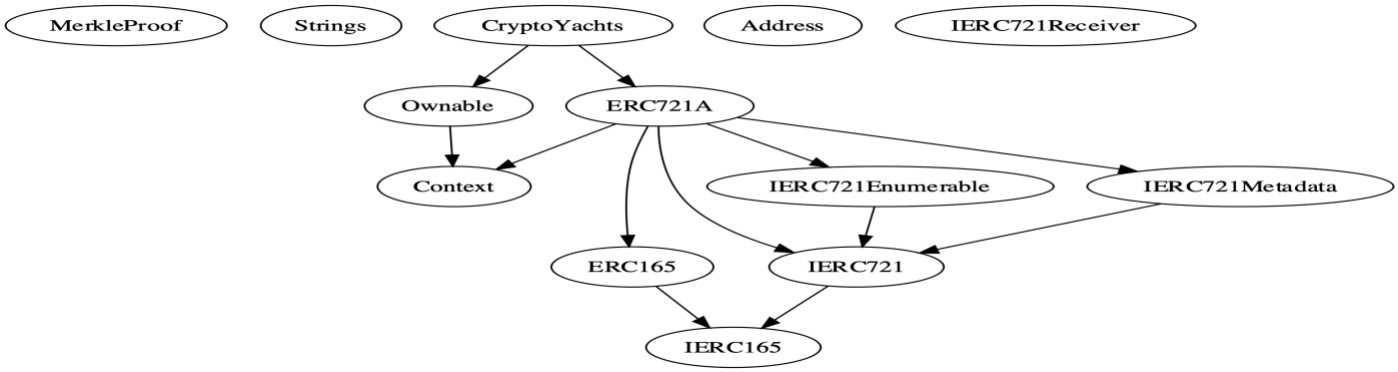
328bcc19378dd6296b5f4e365733b327ed8e2fe8963fcf2e56264167d70d9b12

File: CryptoY... | Language: solidity | Size: 44581 bytes | Date: 2022-03-07T10:47:39.064Z

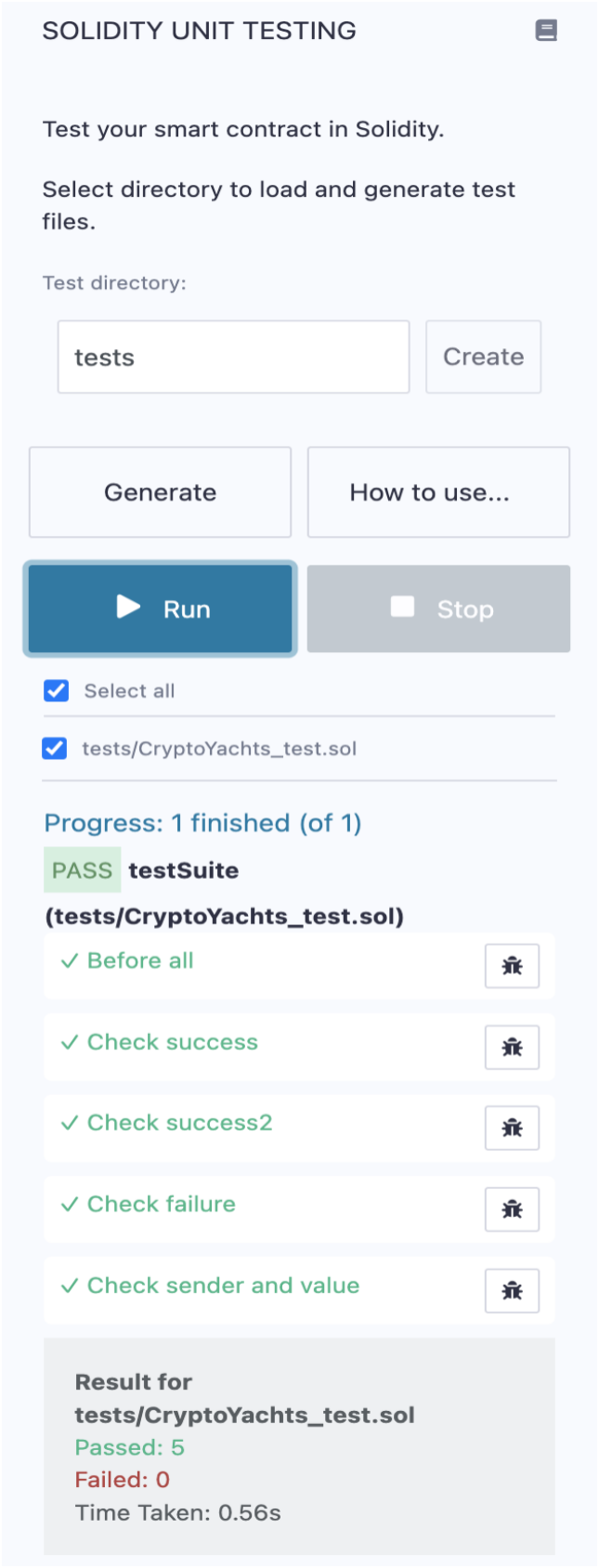
Critical	High	Medium	Low	Note
0	0	0	0	0

2- SOLIDITY STATIC ANALYSIS

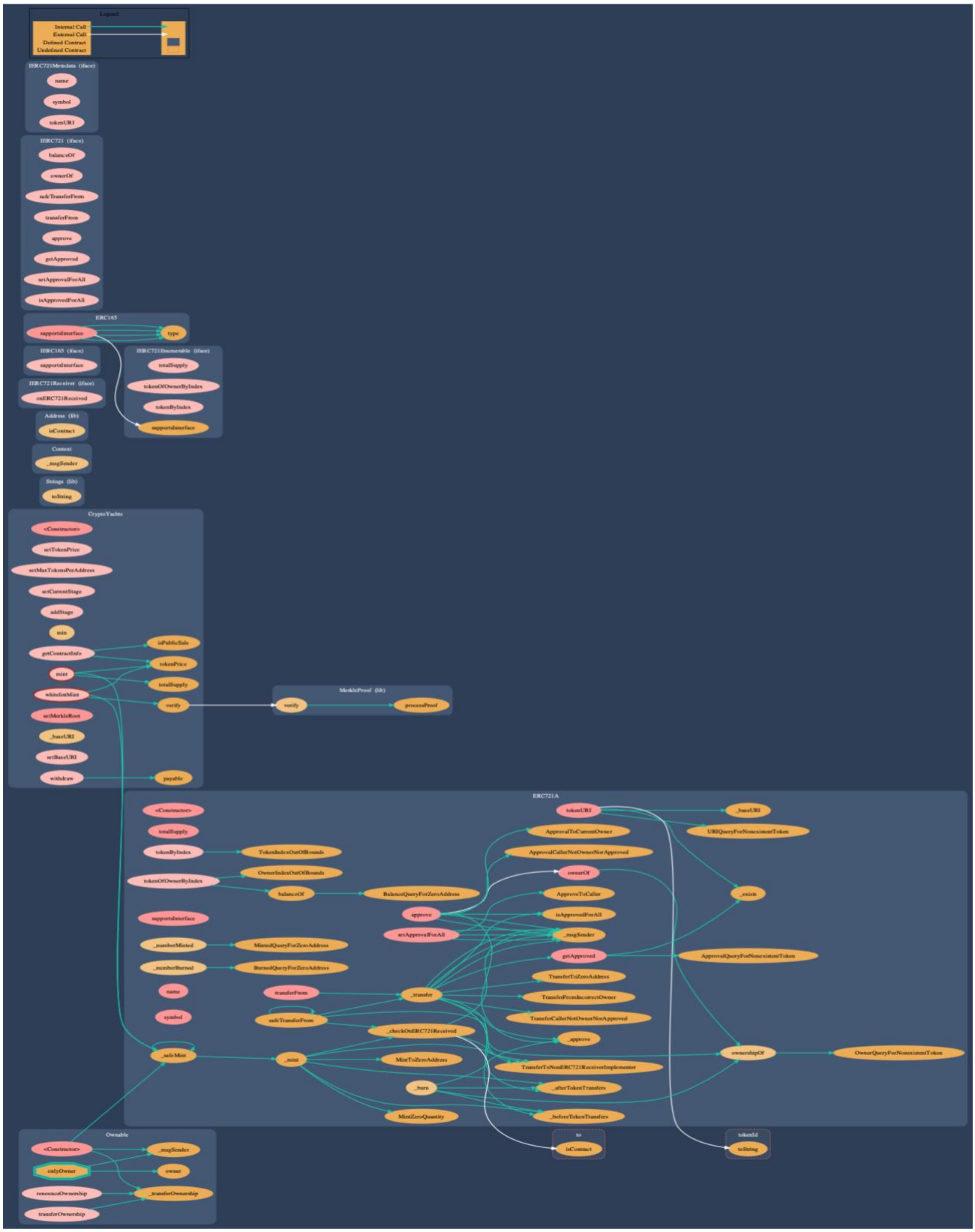
3- Inheritance graph



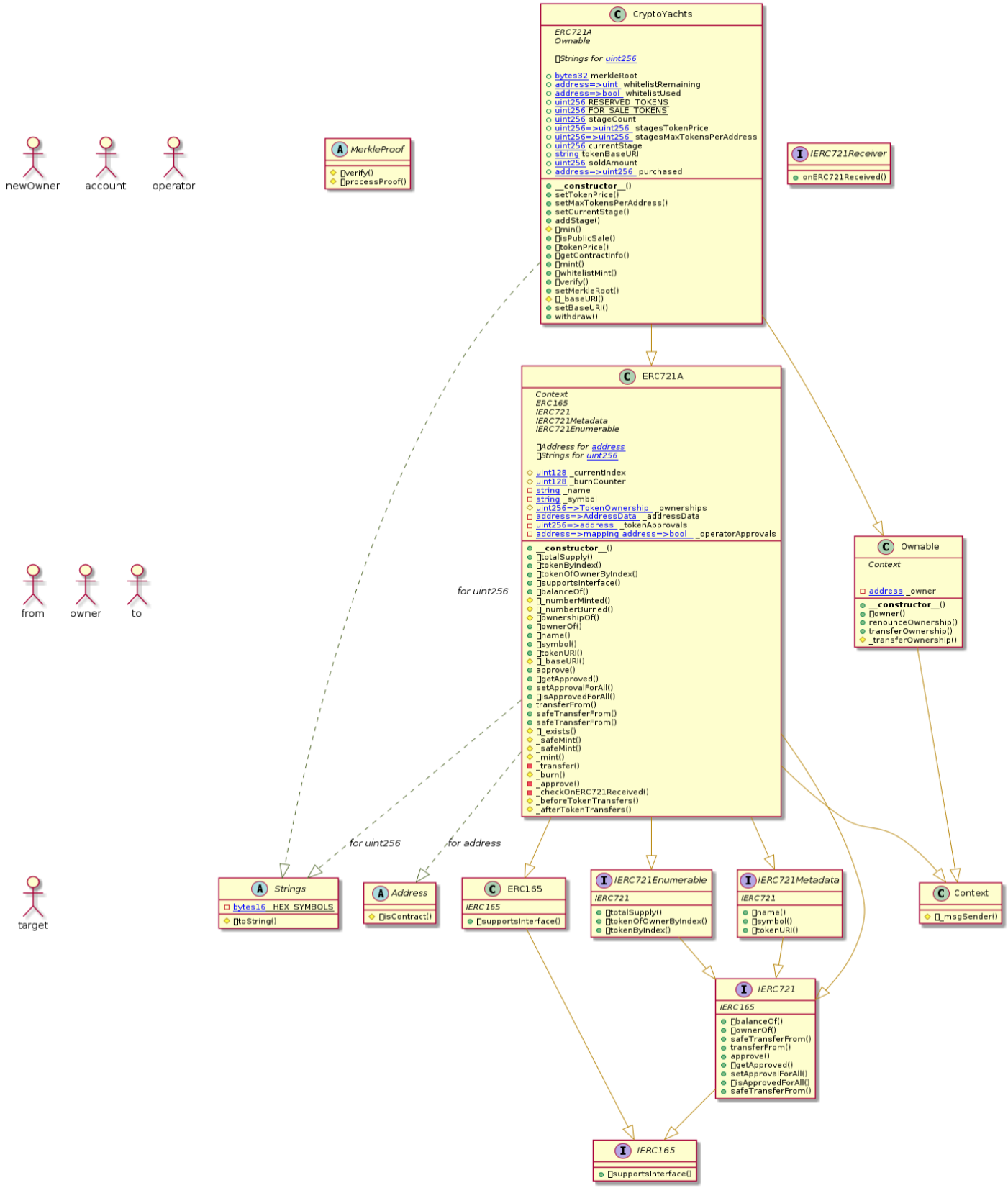
4- SOLIDITY UNIT TESTING



5- Call graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
IERC721Metadata (iface)
name
symbol
tokenURI
IERC721 (iface)
balanceOf
ownerOf
safeTransferFrom
transferFrom
approve
getApproved
setApprovalForAll
isApprovedForAll
ERC165
supportsInterface
type
IERC165 (iface)
supportsInterface
IERC721Enumerable (iface)
totalSupply
tokenOfOwnerByIndex
tokenByIndex
supportsInterface
IERC721Receiver (iface)
onERC721Received
Address (lib)
isContract
Context
_msgSender
Strings (lib)
toString
CryptoYachts
<Constructor>
setTokenPrice
setMaxTokensPerAddress
setCurrentStage
addStage
min
getContractInfo
isPublicSale
tokenPrice
totalSupply
mint
whitelistMint
verify
setMerkleRoot
_baseURI
setBaseURI
withdraw
payable
MerkleProof (lib)
verify
processProof
ERC721A
<Constructor>
totalSupply
tokenByIndex
TokenIndexOutOfBounds
tokenOfOwnerByIndex
OwnerIndexOutOfBounds
balanceOf
BalanceQueryForZeroAddress
supportsInterface
_numberMinted
MintedQueryForZeroAddress
_numberBurned
BurnedQueryForZeroAddress
name
symbol
transferFrom
safeTransferFrom
_safeMint
_mint
_transfer
_checkOnERC721Received
MintToZeroAddress
MintZeroQuantity
_burn
approve
setApprovalForAll
tokenURI
ApprovalToCurrentOwner
ApprovalCallerNotOwnerNorApproved
ownerOf
ApproveToCaller
isApprovedForAll
_msgSender
getApproved
TransferToZeroAddress
TransferFromIncorrectOwner
TransferCallerNotOwnerNorApproved
_approve
TransferToNonERC721ReceiverImplementer
_afterTokenTransfers
_beforeTokenTransfers
_baseURI
URIQueryForNonexistentToken
_exists
ApprovalQueryForNonexistentToken
ownershipOf
OwnerQueryForNonexistentToken
to
isContract
tokenId
toString
Ownable
<Constructor>
_msgSender
onlyOwner
owner
renounceOwnership
_transferOwnership
transferOwnership

Unified Modeling Language (UML)



Functions signature

```
Sighash   |   Function Signature

========================

16279055  =>  isContract(address)
5a9a49c7  =>  verify(bytes32[],bytes32,bytes32)
62702a6b  =>  processProof(bytes32[],bytes32)
6900a3ae  =>  toString(uint256)
119df25f  =>  _msgSender()
8da5cb5b  =>  owner()
715018a6  =>  renounceOwnership()
f2fde38b  =>  transferOwnership(address)
d29d44ee  =>  _transferOwnership(address)
150b7a02  =>  onERC721Received(address,address,uint256,bytes)
01ffc9a7  =>  supportsInterface(bytes4)
70a08231  =>  balanceOf(address)
6352211e  =>  ownerOf(uint256)
42842e0e  =>  safeTransferFrom(address,address,uint256)
23b872dd  =>  transferFrom(address,address,uint256)
095ea7b3  =>  approve(address,uint256)
081812fc  =>  getApproved(uint256)
a22cb465  =>  setApprovalForAll(address,bool)
e985e9c5  =>  isApprovedForAll(address,address)
b88d4fde  =>  safeTransferFrom(address,address,uint256,bytes)
18160ddd  =>  totalSupply()
2f745c59  =>  tokenOfOwnerByIndex(address,uint256)
4f6ccce7  =>  tokenByIndex(uint256)
06fdde03  =>  name()
95d89b41  =>  symbol()
c87b56dd  =>  tokenURI(uint256)
4d388a98  =>  _numberMinted(address)
6ba1b8d0  =>  _numberBurned(address)
140364a1  =>  ownershipOf(uint256)
743976a0  =>  _baseURI()
f8e76cc0  =>  _exists(uint256)
b3e1c718  =>  _safeMint(address,uint256)
6a4f832b  =>  _safeMint(address,uint256,bytes)
de0d9900  =>  _mint(address,uint256,bytes,bool)
30e0789e  =>  _transfer(address,address,uint256)
9b1f9e74  =>  _burn(uint256)
f272404d  =>  _approve(address,uint256,address)
1fd01de1  =>  _checkOnERC721Received(address,address,uint256,bytes)
ef435773  =>  _beforeTokenTransfers(address,address,uint256,uint256)
08c018f7  =>  _afterTokenTransfers(address,address,uint256,uint256)
eb685c47  =>  setTokenPrice(uint256,uint256)
0e4be24c  =>  setMaxTokensPerAddress(uint256,uint256)
38c67b73  =>  setCurrentStage(uint256)
9508614b  =>  addStage()
7ae2b5c7  =>  min(uint256,uint256)
a5a865dc  =>  isPublicSale()
7ff9b596  =>  tokenPrice()
cd481e51  =>  getContractInfo(address)
a0712d68  =>  mint(uint256)
```

```
5bfe024d  =>  whitelistMint(uint256,uint256,bytes32,bytes32[])
3423e548  =>  verify(bytes32,bytes32,bytes32[])
7cb64759  =>  setMerkleRoot(bytes32)
55f804b3  =>  setBaseURI(string)
3ccfd60b  =>  withdraw()
```

Automatic general report

Files Description Table

File Name	SHA-1 Hash
/Users/macbook/Desktop/smart contracts/CryptoYachts.sol	02c0c0e0f34c5034145c935b8f0f9344881fcd3c

Contracts Description Table

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
MerkleProof	Library			
└	verify	Internal 🔒		
└	processProof	Internal 🔒		
Strings	Library			
└	toString	Internal 🔒		
Context	Implementation			
└	_msgSender	Internal 🔒		
Ownable	Implementation	Context		
└	<Constructor>	Public ❗	🛑	NO❗
└	owner	Public ❗		NO❗
└	renounceOwnership	External ❗	🛑	onlyOwner
└	transferOwnership	External ❗	🛑	onlyOwner
└	_transferOwnership	Internal 🔒	🛑	
Address	Library			
└	isContract	Internal 🔒		
IERC721Receiver	Interface			
└	onERC721Received	External ❗	🛑	NO❗
IERC165	Interface			
└	supportsInterface	External ❗		NO❗
ERC165	Implementation	IERC165		
└	supportsInterface	Public ❗		NO❗
IERC721	Interface	IERC165		
└	balanceOf	External ❗		NO❗
└	ownerOf	External ❗		NO❗
└	safeTransferFrom	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗
└	approve	External ❗	🛑	NO❗
└	getApproved	External ❗		NO❗

└	setApprovalForAll	External ❗	🛑	NO❗
└	isApprovedForAll	External ❗		NO❗
└	safeTransferFrom	External ❗	🛑	NO❗
IERC721Enumerable	Interface	IERC721		
└	totalSupply	External ❗		NO❗
└	tokenOfOwnerByIndex	External ❗		NO❗
└	tokenByIndex	External ❗		NO❗
IERC721Metadata	Interface	IERC721		
└	name	External ❗		NO❗
└	symbol	External ❗		NO❗
└	tokenURI	External ❗		NO❗
ERC721A	Implementation	Context, ERC165, IERC721, IERC721Metadata, IERC721Enumerable		
└	<Constructor>	Public ❗	🛑	NO❗
└	totalSupply	Public ❗		NO❗
└	tokenByIndex	External ❗		NO❗
└	tokenOfOwnerByIndex	External ❗		NO❗
└	supportsInterface	Public ❗		NO❗
└	balanceOf	Public ❗		NO❗
└	_numberMinted	Internal 🔒		
└	_numberBurned	Internal 🔒		
└	ownershipOf	Internal 🔒		
└	ownerOf	Public ❗		NO❗
└	name	Public ❗		NO❗
└	symbol	Public ❗		NO❗
└	tokenURI	Public ❗		NO❗
└	_baseURI	Internal 🔒		
└	approve	Public ❗	🛑	NO❗
└	getApproved	Public ❗		NO❗
└	setApprovalForAll	Public ❗	🛑	NO❗
└	isApprovedForAll	Public ❗		NO❗
└	transferFrom	Public ❗	🛑	NO❗
└	safeTransferFrom	Public ❗	🛑	NO❗
└	safeTransferFrom	Public ❗	🛑	NO❗
└	_exists	Internal 🔒		
└	_safeMint	Internal 🔒	🛑	
└	_safeMint	Internal 🔒	🛑	
└	_mint	Internal 🔒	🛑	
└	_transfer	Private 🔐	🛑	
└	_burn	Internal 🔒	🛑	
└	_approve	Private 🔐	🛑	
└	_checkOnERC721Received	Private 🔐	🛑	
└	_beforeTokenTransfers	Internal 🔒	🛑	
└	_afterTokenTransfers	Internal 🔒	🛑	
CryptoYachts	Implementation	ERC721A, Ownable		
└	<Constructor>	Public ❗	🛑	ERC721A
└	setTokenPrice	External ❗	🛑	onlyOwner
└	setMaxTokensPerAddress	External ❗	🛑	onlyOwner
└	setCurrentStage	External ❗	🛑	onlyOwner
└	addStage	External ❗	🛑	onlyOwner
└	min	Internal 🔒		

└	isPublicSale	Public ❗		NO❗
└	tokenPrice	Public ❗		NO❗
└	getContractInfo	External ❗		NO❗
└	mint	External ❗	💵	NO❗
└	whitelistMint	External ❗	💵	NO❗
└	verify	Public ❗		NO❗
└	setMerkleRoot	Public ❗	🛑	onlyOwner
└	_baseURI	Internal 🔒		
└	setBaseURI	External ❗	🛑	onlyOwner
└	withdraw	External ❗	🛑	onlyOwner

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

Conclusion

The contracts are written systematically. Team found no critical issues. So, it is good to go for production.

Since possible test cases can be unlimited and developer level documentation (code flow diagram with function level description) not provided, for such an extensive smart contract protocol, we provide no such guarantee of future outcomes. We have used all the latest static tools and manual observations to cover maximum possible test cases to scan Everything.

Security state of the reviewed contract is “ Well Secured”.

✓ No volatile code.
✓ Not many high severity issues were found.

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as of the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against the team on the basis of what it says or doesn't say, or how team produced it, and it is important for you to conduct your own independent investigations before making any decisions. team go into more detail on this in the below disclaimer below – please make sure to read it in full.

By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and Saferico and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (Saferico s) owe no duty of care towards you or any other person, nor does Saferico make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and Saferico hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, Saferico hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against Saferico, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report. The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.