Smart Contract
Security Audit
V1

CrytoRocks NFT Smart Contract

18/2/2022

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

- **Website**: https://cryptorocks.store/
- **OPenSea:** https://opensea.io/collection/cryptorocksnft
- **Twitter:** https://twitter.com/cryptorocksnft
- **Discord**: https://discord.com/invite/5t5KjK22jJ
- **Telegram**: https://t.me/cryptorocksnft
- **Instagram:** https://www.instagram.com/cryptorocks.nft/
- **Facebook**: https://www.facebook.com/cryptorocksnft
- **Medium**: https://crypto-rocks.medium.com/
- **Quora**: https://www.quora.com/profile/Crypto-Rocks
- **Platform**: Ethereum
- **Contract Address**: 0x2DA3395Fdb95bD356806477a7C3050A5960cfCD6
- **Code:**

https://rinkeby.etherscan.io/address/0x2da3395fdb95bd356806477a7c3050a5960cfcd6#code

Contracts address deployed to test net (Ethereum)

CrytoRocks NFT Smart contract on ETH test net to test important functions by the auditor.

https://rinkeby.etherscan.io/address/0x2da3395fdb95bd356806477a7c3050a5960cfcd6

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 1 notes in all solidity files of the contract

The files:

CrytoRocks.sol

File and Function Level Report

File in Scope:

Contract Name	SHA 256 hash	Contract Address
CrytoRocks.sol	**951c0082a9b4acaa714c5042643850c559274db112fceb5921b49cdf2ee5c098**	0x2DA3395Fdb95bD356806477a7C3050A5960cfCD6

- Contract: CrytoRocks
- Inherit: ERC1155, Ownable, ERC2981Base
- Observation: All passed including security check
- Test Report: passed
- Score: passed
- Conclusion: passed

Function	Test Result	Type / Return Type	Score
loteri	✓	Read / public	**Passed**
uri	✓	Read / public	**Passed**
getPrice	✓	Read / public	**Passed**
supportsInterface	✓	Read / public	**Passed**
addressMintedBalance	✓	Read / public	**Passed**
balanceOf	✓	Read / public	**Passed**
Owner	✓	Read / public	**Passed**
balanceOfBatch	✓	Read / public	**Passed**
BlanceOfTokens	✓	Read / public	**Passed**
getApprovedForAll	✓	Read / public	**Passed**
Bote	✓	Read / public	**Passed**
isMember	✓	Read / public	**Passed**
getAmountMinted	✓	Read / public	**Passed**

onlyWhitelisted	✓	Read / public	**Passed**
maxMintAmount	✓	Read / public	**Passed**
nftPerAddressLimit	✓	Read / public	**Passed**
royaltyInfo	✓	Read / public	**Passed**
addMember	✓	Write / public	**Passed**
addMemberList	✓	Write / public	**Passed**
safeTransferFrom	✓	Write / public	**Passed**
safeTransferFrom	✓	Write / public	**Passed**
setOnlyWhiteListed	✓	Write / public	**Passed**
pause	✓	Write / public	**Passed**
mint	✓	Write / payable	**Passed**
airdropGiveaway	✓	Write / public	**Passed**
transferOwnership	✓	Write / public	**Passed**
setApprovalForAll	✓	Write / public	**Passed**
transferByOwner	✓	Write / public	**Passed**
withdraw	✓	Write / public	**Passed**
GenerarGanador	✓	Write / public	**Passed**
renounceOwnership	✓	Write / public	**Passed**
removeMember	✓	Write / public	**Passed**
togglePublicSale	✓	Write / public	**Passed**
revelar	✓	Write / public	**Passed**
safeBatchTransferFrom	✓	Write / public	**Passed**
safeBatchTransferFrom	✓	Write / public	**Passed**
setPrice	✓	Write / public	**Passed**
toggleonlyWhitelisted	✓	Write / public	**Passed**
togglePrivateSale	✓	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 CrytoRocks.sol goes way above this value and currently is of size 59200 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:

#Multiple pragma statements

Line	Pragma
5	pragma solidity ^0.8.0;
46	pragma solidity ^0.8.0;
60	pragma solidity ^0.8.0;
289	pragma solidity ^0.8.0;
356	pragma solidity ^0.8.0;
380	pragma solidity ^0.8.0;
455	pragma solidity ^0.8.0;
677	pragma solidity ^0.8.0;

705	pragma solidity ^0.8.0;
831	pragma solidity ^0.8.0;
856	pragma solidity ^0.8.0;
914	pragma solidity ^0.8.0;
1427	pragma solidity ^0.8.0;
1465	pragma solidity ^0.8.0;
1501	pragma solidity ^0.8.2;

Description
There are multiple pragma statements in the code. Only the compiler version 0.8.2 will work with the code, but keeping only one pragma statement helps in maintaining readability of the code.

Remediation

Keep a single pragma statement.

Status: 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:

#Unnecessary use of SafeMath
Description

Solidity version 0.8 was released with SafeMath checks inbuilt, we can avoid using an explicit safe math library.

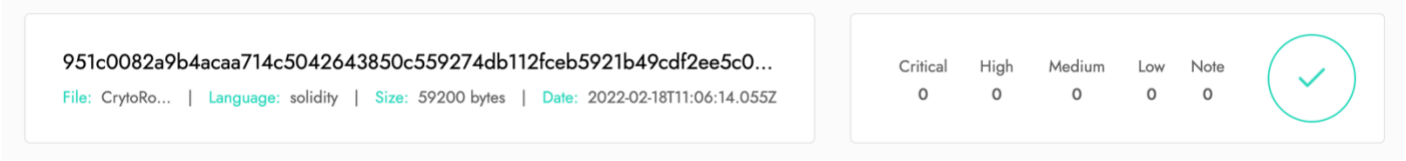
Remediation

Remove SafeMath Library to save gas fees.

Status: closed. Fix in version 2.

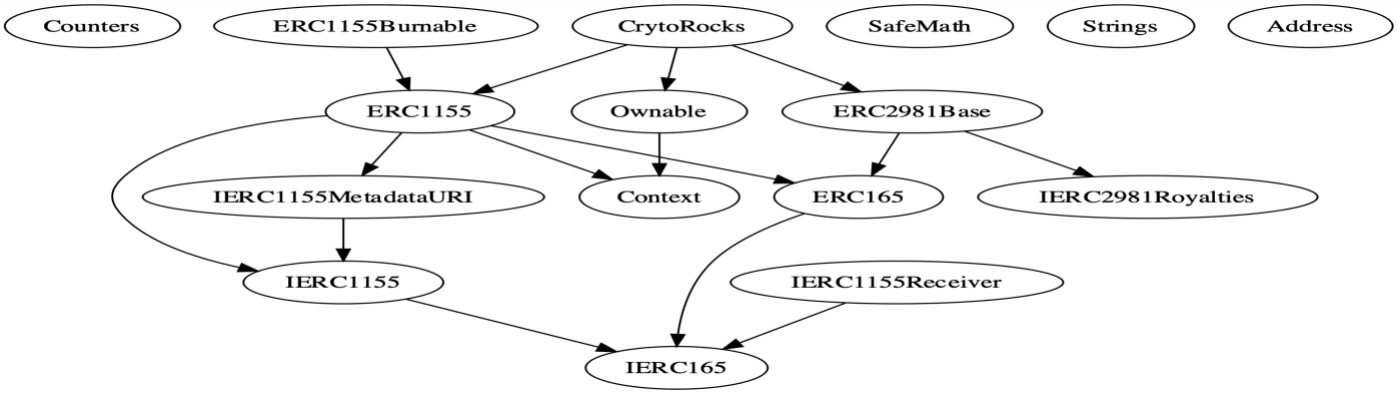
Automatic Testing

1- Check for security

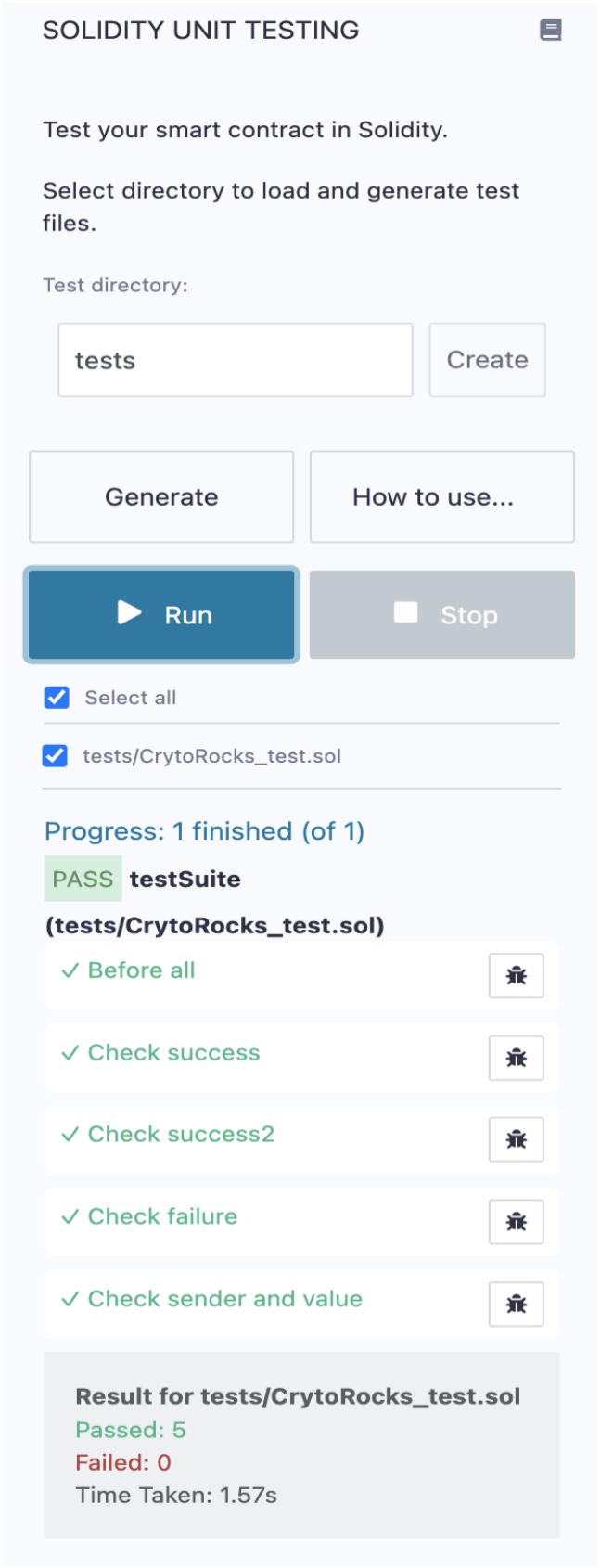


2- SOLIDITY STATIC ANALYSIS

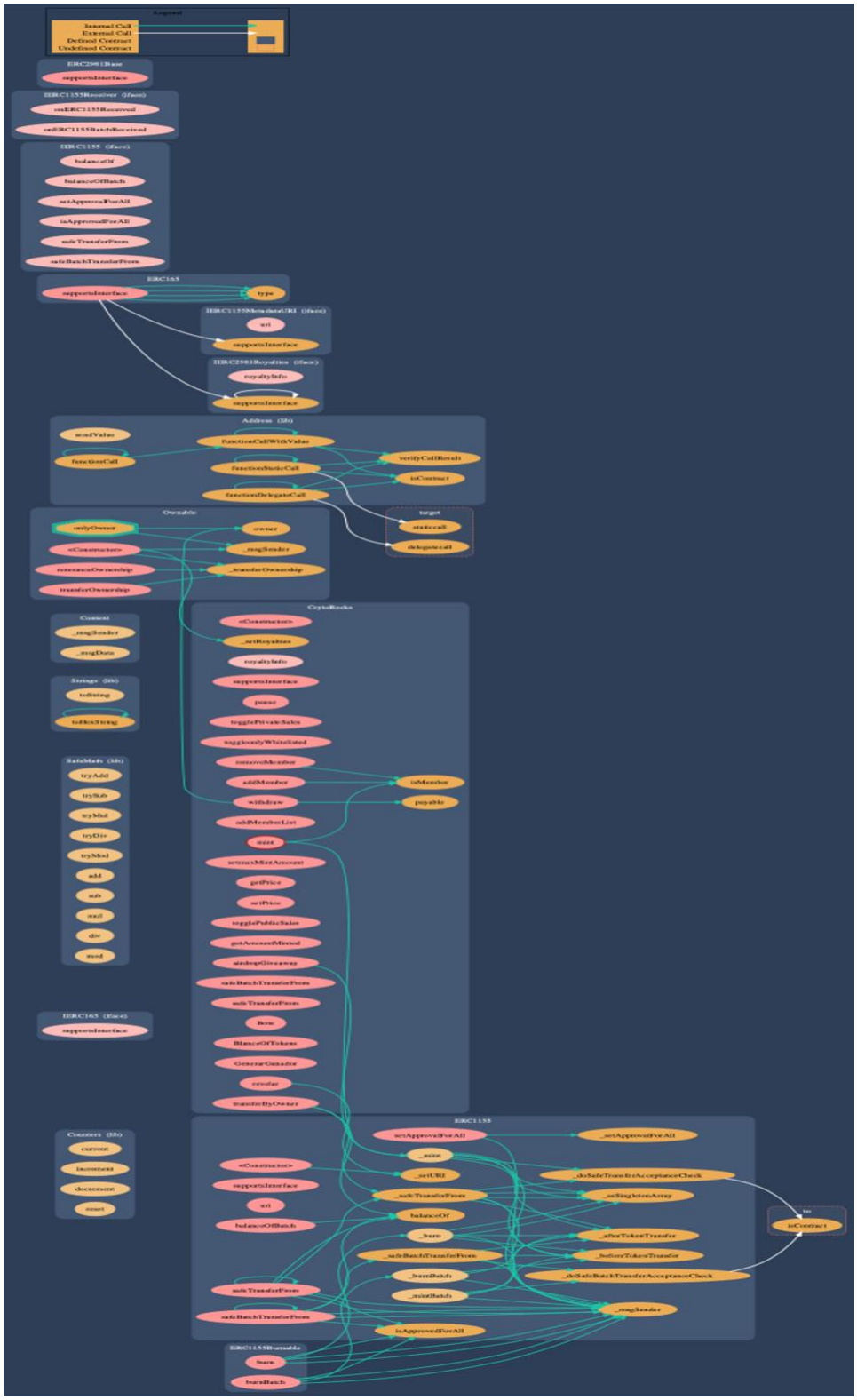
3- Inheritance graph



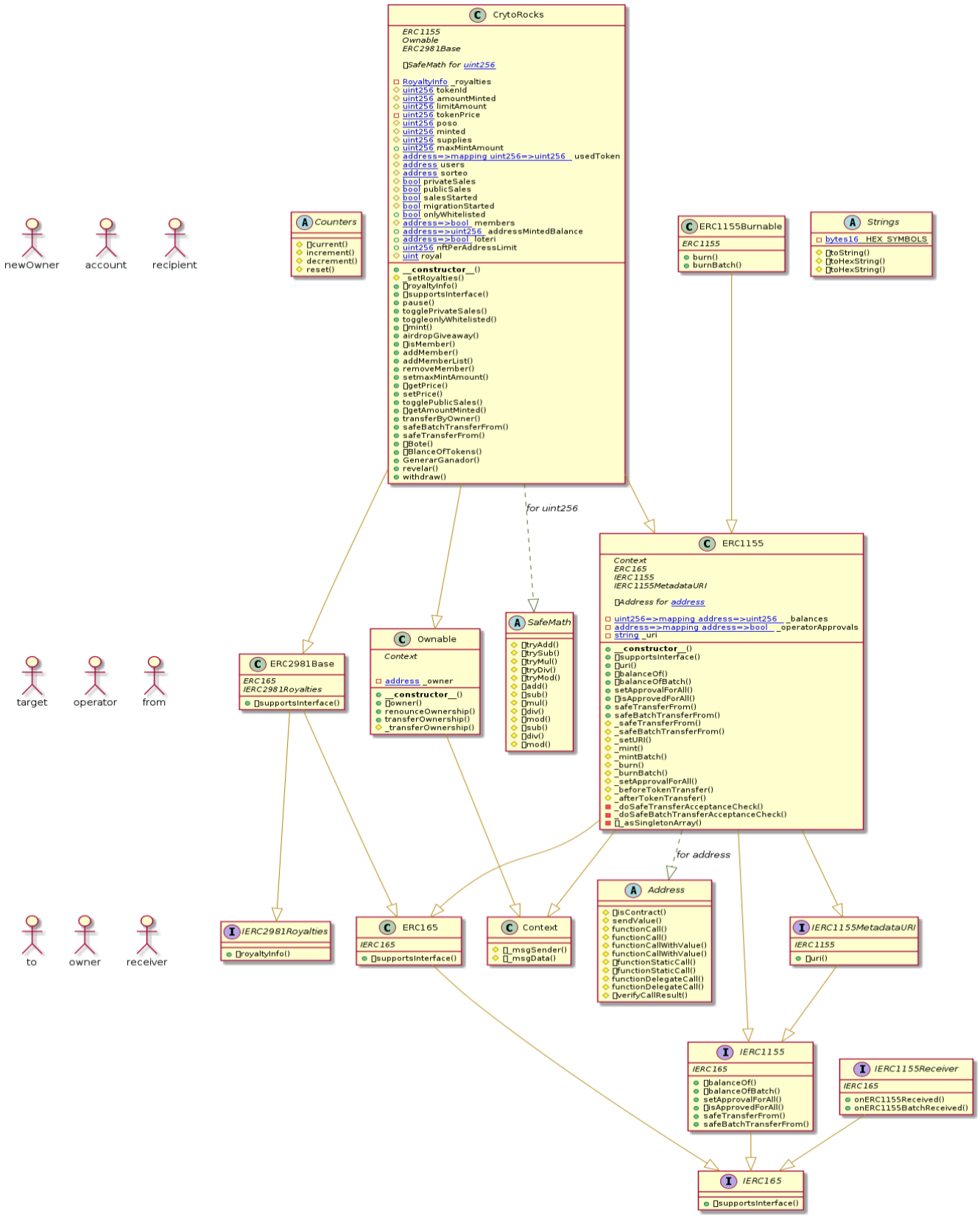
4- SOLIDITY UNIT TESTING



5- Call graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
ERC2981Base
supportsInterface
IERC1155Receiver (iface)
onERC1155Received
onERC1155BatchReceived
IERC1155 (iface)
balanceOf
balanceOfBatch
setApprovalForAll
isApprovedForAll
safeTransferFrom
safeBatchTransferFrom
ERC165
supportsInterface
type
IERC1155MetadataURI (iface)
uri
supportsInterface
IERC2981Royalties (iface)
royaltyInfo
supportsInterface
Address (lib)
sendValue
functionCall
functionCallWithValue
functionStaticCall
functionDelegateCall
verifyCallResult
isContract
target
staticcall
delegatecall
Ownable
onlyOwner
<Constructor>
renounceOwnership
transferOwnership
owner
_msgSender
_transferOwnership
Context
_msgSender
_msgData
Strings (lib)
toString
toHexString
SafeMath (lib)
tryAdd
trySub
tryMul
tryDiv
tryMod
add
sub
mul
div
mod
IERC165 (iface)
supportsInterface
Counters (lib)
current
increment
decrement
reset
CryptoRocks
<Constructor>
_setRoyalties
royaltyInfo
supportsInterface
pause
togglePrivateSales
toggleOnlyWhitelisted
removeMember
addMember
withdraw
addMemberList
mint
remainMintAmount
getPrice
setPrice
toggglePublicSales
getAmountMinted
airdropGiveaway
safeBatchTransferFrom
safeTransferFrom
Role
BlanceOfToken
GenerateSeed
reader
transferByOwner
isMember
payable
ERC1155
<Constructor>
supportsInterface
uri
balanceOfBatch
setApprovalForAll
_mint
_setURI
safeTransferFrom
balanceOf
_burn
safeBatchTransferFrom
_burnBatch
_mintBatch
safeTransferFrom
safeBatchTransferFrom
isApprovedForAll
_setApprovalForAll
_doSafeTransferAcceptanceCheck
_asSingletonArray
_afterTokenTransfer
_beforeTokenTransfer
_doSafeBatchTransferAcceptanceCheck
_msgSender
to
isContract
ERC1155Burnable
burn
burnBatch

Unified Modeling Language (UML)



Functions signature

```
Sighash   |   Function Signature
========================
16279055  =>  isContract(address)
74189861  =>  Bote()
ad04a8d1  =>  current(Counter)
e2bee435  =>  increment(Counter)
854ec98e  =>  decrement(Counter)
440d212a  =>  reset(Counter)
01ffc9a7  =>  supportsInterface(bytes4)
884557bf  =>  tryAdd(uint256,uint256)
a29962b1  =>  trySub(uint256,uint256)
6281efa4  =>  tryMul(uint256,uint256)
736ecb18  =>  tryDiv(uint256,uint256)
38dc0867  =>  tryMod(uint256,uint256)
771602f7  =>  add(uint256,uint256)
b67d77c5  =>  sub(uint256,uint256)
c8a4ac9c  =>  mul(uint256,uint256)
a391c15b  =>  div(uint256,uint256)
f43f523a  =>  mod(uint256,uint256)
e31bdc0a  =>  sub(uint256,uint256,string)
b745d336  =>  div(uint256,uint256,string)
71af23e8  =>  mod(uint256,uint256,string)
6900a3ae  =>  toString(uint256)
8fba8d5c  =>  toHexString(uint256)
63e1cbea  =>  toHexString(uint256,uint256)
119df25f  =>  _msgSender()
8b49d47e  =>  _msgData()
8da5cb5b  =>  owner()
715018a6  =>  renounceOwnership()
f2fde38b  =>  transferOwnership(address)
d29d44ee  =>  _transferOwnership(address)
24a084df  =>  sendValue(address,uint256)
a0b5ffb0  =>  functionCall(address,bytes)
241b5886  =>  functionCall(address,bytes,string)
2a011594  =>  functionCallWithValue(address,bytes,uint256)
d525ab8a  =>  functionCallWithValue(address,bytes,uint256,string)
c21d36f3  =>  functionStaticCall(address,bytes)
dbc40fb9  =>  functionStaticCall(address,bytes,string)
ee33b7e2  =>  functionDelegateCall(address,bytes)
57387df0  =>  functionDelegateCall(address,bytes,string)
946b5793  =>  verifyCallResult(bool,bytes,string)
00fdd58e  =>  balanceOf(address,uint256)
4e1273f4  =>  balanceOfBatch(address[],uint256[])
a22cb465  =>  setApprovalForAll(address,bool)
e985e9c5  =>  isApprovedForAll(address,address)
f242432a  =>  safeTransferFrom(address,address,uint256,uint256,bytes)
2eb2c2d6  =>  safeBatchTransferFrom(address,address,uint256[],uint256[],bytes)
0e89341c  =>  uri(uint256)
f23a6e61  =>  onERC1155Received(address,address,uint256,uint256,bytes)
bc197c81  =>  onERC1155BatchReceived(address,address,uint256[],uint256[],bytes)
8e8a7590  =>  _safeTransferFrom(address,address,uint256,uint256,bytes)
4f2999a7  =>  _safeBatchTransferFrom(address,address,uint256[],uint256[],bytes)
f392d4f5  =>  _setURI(string)
```

```
2ca6f4da  =>  _mint(address,uint256,uint256,bytes)
b6ea6b0a  =>  _mintBatch(address,uint256[],uint256[],bytes)
464a5ffb  =>  _burn(address,uint256,uint256)
f3323d3c  =>  _burnBatch(address,uint256[],uint256[])
8c4e3f32  =>  _setApprovalForAll(address,address,bool)
fe49010b  =>
_beforeTokenTransfer(address,address,address,uint256[],uint256[],bytes)
6024ca4a  =>
_afterTokenTransfer(address,address,address,uint256[],uint256[],bytes)
084e9e24  =>
_doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes)
e51c223d  =>
_doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],byt
es)
136ad9d4  =>  _asSingletonArray(uint256)
f5298aca  =>  burn(address,uint256,uint256)
6b20c454  =>  burnBatch(address,uint256[],uint256[])
2a55205a  =>  royaltyInfo(uint256,uint256)
40a04a5a  =>  _setRoyalties(address,uint256)
8456cb59  =>  pause()
f152aa03  =>  togglePrivateSales()
94637dd2  =>  toggleonlyWhitelisted()
a0712d68  =>  mint(uint256)
6810b3cb  =>  airdropGiveaway(address,uint256)
a230c524  =>  isMember(address)
ca6d56dc  =>  addMember(address)
00173f5c  =>  addMemberList(address[])
0b1ca49a  =>  removeMember(address)
7f00c7a6  =>  setmaxMintAmount(uint256)
98d5fdca  =>  getPrice()
91b7f5ed  =>  setPrice(uint256)
8d19c2aa  =>  togglePublicSales()
e777df20  =>  getAmountMinted()
224a1775  =>  transferByOwner(address,uint256,uint256)
23c05b17  =>  safeBatchTransferFrom(address,uint256[],uint256[])
f009bd23  =>  safeTransferFrom(address,uint256,uint256)
f81e694e  =>  BlanceOfTokens(address)
5332eee5  =>  GenerarGanador()
f20579f1  =>  revelar(string)
3ccfd60b  =>  withdraw()
```

Automatic general report

Files Description Table

File Name	SHA-1 Hash
/Users/macbook/Desktop/smart contracts/CrytoRocks.sol	6b6a097fac0c61b718f3f4a07baf524193e2c016

Contracts Description Table

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
Counters	Library			
└	current	Internal 🔒		
└	increment	Internal 🔒	🛑	
└	decrement	Internal 🔒	🛑	
└	reset	Internal 🔒	🛑	
IERC165	Interface			
└	supportsInterface	External ❗		NO❗
SafeMath	Library			
└	tryAdd	Internal 🔒		
└	trySub	Internal 🔒		
└	tryMul	Internal 🔒		
└	tryDiv	Internal 🔒		
└	tryMod	Internal 🔒		
└	add	Internal 🔒		
└	sub	Internal 🔒		
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
└	sub	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
Strings	Library			
└	toString	Internal 🔒		
└	toHexString	Internal 🔒		
└	toHexString	Internal 🔒		
Context	Implementation			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
Ownable	Implementation	Context		
└	<Constructor>	Public ❗	🛑	NO❗
└	owner	Public ❗		NO❗


└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
└	_transferOwnership	Internal 🔒	🛑	
Address	Library			
└	isContract	Internal 🔒		
└	sendValue	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	functionStaticCall	Internal 🔒		
└	functionStaticCall	Internal 🔒		
└	functionDelegateCall	Internal 🔒	🛑	
└	functionDelegateCall	Internal 🔒	🛑	
└	verifyCallResult	Internal 🔒		
ERC165	Implementation	IERC165		
└	supportsInterface	Public ❗		NO❗
IERC1155	Interface	IERC165		
└	balanceOf	External ❗		NO❗
└	balanceOfBatch	External ❗		NO❗
└	setApprovalForAll	External ❗	🛑	NO❗
└	isApprovedForAll	External ❗		NO❗
└	safeTransferFrom	External ❗	🛑	NO❗
└	safeBatchTransferFrom	External ❗	🛑	NO❗
IERC1155MetadataURI	Interface	IERC1155		
└	uri	External ❗		NO❗
IERC1155Receiver	Interface	IERC165		
└	onERC1155Received	External ❗	🛑	NO❗
└	onERC1155BatchReceived	External ❗	🛑	NO❗
ERC1155	Implementation	Context, ERC165, IERC1155, IERC1155MetadataURI		
└	<Constructor>	Public ❗	🛑	NO❗
└	supportsInterface	Public ❗		NO❗
└	uri	Public ❗		NO❗
└	balanceOf	Public ❗		NO❗
└	balanceOfBatch	Public ❗		NO❗
└	setApprovalForAll	Public ❗	🛑	NO❗
└	isApprovedForAll	Public ❗		NO❗
└	safeTransferFrom	Public ❗	🛑	NO❗
└	safeBatchTransferFrom	Public ❗	🛑	NO❗
└	_safeTransferFrom	Internal 🔒	🛑	
└	_safeBatchTransferFrom	Internal 🔒	🛑	
└	_setURI	Internal 🔒	🛑	
└	_mint	Internal 🔒	🛑	
└	_mintBatch	Internal 🔒	🛑	
└	_burn	Internal 🔒	🛑	
└	_burnBatch	Internal 🔒	🛑	
└	_setApprovalForAll	Internal 🔒	🛑	
└	_beforeTokenTransfer	Internal 🔒	🛑	
└	_afterTokenTransfer	Internal 🔒	🛑	

└	_doSafeTransferAcceptanceCheck	Private 🔐	🛑	
└	_doSafeBatchTransferAcceptanceCheck	Private 🔐	🛑	
└	_asSingletonArray	Private 🔐		
ERC1155Burnable	Implementation	ERC1155		
└	burn	Public ❗	🛑	NO❗
└	burnBatch	Public ❗	🛑	NO❗
IERC2981Royalties	Interface				
└	royaltyInfo	External ❗		NO❗
ERC2981Base	Implementation	ERC165, IERC2981Royalties		
└	supportsInterface	Public ❗		NO❗
CrytoRocks	Implementation	ERC1155, Ownable, ERC2981Base		
└	<Constructor>	Public ❗	🛑	ERC1155
└	_setRoyalties	Internal 🔒	🛑	
└	royaltyInfo	External ❗		NO❗
└	supportsInterface	Public ❗		NO❗
└	pause	Public ❗	🛑	onlyOwner
└	togglePrivateSales	Public ❗	🛑	onlyOwner
└	toggleonlyWhitelisted	Public ❗	🛑	onlyOwner
└	mint	Public ❗	💵	NO❗
└	airdropGiveaway	Public ❗	🛑	onlyOwner
└	isMember	Public ❗		NO❗
└	addMember	Public ❗	🛑	onlyOwner
└	addMemberList	Public ❗	🛑	onlyOwner
└	removeMember	Public ❗	🛑	onlyOwner
└	setmaxMintAmount	Public ❗	🛑	onlyOwner
└	getPrice	Public ❗		NO❗
└	setPrice	Public ❗	🛑	onlyOwner
└	togglePublicSales	Public ❗	🛑	onlyOwner
└	getAmountMinted	Public ❗		NO❗
└	transferByOwner	Public ❗	🛑	NO❗
└	safeBatchTransferFrom	Public ❗	🛑	NO❗
└	safeTransferFrom	Public ❗	🛑	NO❗
└	Bote	Public ❗		NO❗
└	BlanceOfTokens	Public ❗		NO❗
└	GenerarGanador	Public ❗	🛑	onlyOwner
└	revelar	Public ❗	🛑	onlyOwner
└	withdraw	Public ❗	🛑	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.