CONTRACT WOLF

Blockchain Security - Smart Contract Audits

Security Assessment

March 20, 2022

Disclaimer

ContractWolf.io audits and reports should not be considered as a form of project's "advertisement" and does not cover any interaction and assessment from "project's contract" to "external contracts" such as Pancakeswap or similar.

ContractWolf does not provide any warranty on its released reports.

ContractWolf should not be used as a decision to invest into an audited project and is not affiliated nor partners to its audited contract projects.

ContractWolf provides transparent report to all its "clients" and to its "clients participants" and will not claim any guarantee of bug-free code within its **SMART CONTRACT**.

ContractWolf presence is to analyze, audit and assess the client's smart contract's code.

Each company or projects should be liable to its security flaws and functionalities.

Scope of Work

CryptoFrog team agreed and provided us with the files that needs to be tested (Github, Bscscan, Etherscan, files, etc.). The scope of the audit is the main contract.

The goal of this engagement was to identify if there is a possibility of security flaws in the implementation of the contract or system.

ContractWolf will be focusing on contract issues and functionalities along with the projects claims from smart contract to their website, whitepaper and repository which has been provided by **CryptoFrog**.

Description

CryptoFrog is an ROI project on BNB-Network with NFTs implemented. Employing the technology of blockchain and smart-contract, the whole process of making investments and getting benefits is fully transparent and secure.

Risk Level Classification

Risk Level represents the classification or the probability that a certain function or threat that can exploit vulnerability and have an impact within the system or contract.

Risk Level is computed based on CVSS Version 3.0

Level	Value	Vulnerability
Critical	9 - 10	An Exposure that can affect the contract functions in several events that can risk and disrupt the contract
High	7 - 8.9	An Exposure that can affect the outcome when using the contract that can serve as an opening in manipulating the contract in an unwanted manner
Medium	4 - 6.9	An opening that could affect the outcome in executing the contract in a specific situation
Low	0.1 - 3.9	An opening but doesn't have an impact on the functionality of the contract
Informational	0	An opening that consists of information's but will not risk or affect the contract

Auditing Approach

Every line of code along with its functionalities will undergo manual review to check its security issues, quality, and contract scope of inheritance. The manual review will be done by our team that will document any issues that there were discovered.

Methodology

The auditing process follows a routine series of steps:

1. Code review that includes the following:

    - Review of the specifications, sources, and instructions provided to ContractWolf to make sure we understand the size, scope, and functionality of the smart contract.
    - Manual review of code, our team will have a process of reading the code line-by-line with the intention of identifying potential vulnerabilities and security flaws.

2. Testing and automated analysis that includes:

    - Testing the smart contract functions with common test cases and scenarios, to ensure that it returns the expected results.

3. Best practices review, the team will review the contract with the aim to improve efficiency, effectiveness, clarifications, maintainability, security, and control within the smart contract.

4. Recommendations to help the project take steps to secure the smart contract.

Used Code from other Frameworks/Smart Contracts (Direct Imports)

Imported Packages

- **FrogPart**
  - IGeneScience
  - Math
  - Strings
  - ReentrancyGuard
  - Context
  - Ownable
  - Address
  - IERC165
  - ERC165
  - IERC1155Receiver
  - IERC1155
  - IERC1155MetadataURI
  - ERC1155
  - ERC1155Burnable
  - FrogPart
- **GeneScience**
  - Math
  - Strings
  - SafeMath
  - GeneScience
- **GenesisFrog**
  - IVault
  - IGeneScience
  - Reentrancy
  - Counters
  - Math
  - Strings

    - Context
    - Ownable
    - Address
    - IERC721Receiver
    - IERC165
    - ERC165
    - IERC721
    - IERC721Metadata
    - ERC721
    - ERC721Burnable
    - GenesisFrog
- **Marking**
    - IERC165
    - IERC1155
    - IERC721
    - Context
    - Ownable
    - Address
    - IERC20Permit
    - SafeERC20
    - SafeMath
    - MarkingType
- **MysteryBox**
    - IFrogPart
    - IGeneScience
    - IGenesisFrog
    - IVault
    - ReentrancyGuard
    - Context
    - Ownable
    - Counters
    - SafeMath
- **Vault**
    - IGenesisFrog

- IGeneScience
- ReentrancyGuard
- SafeMath
- Context
- Ownable
- Address
- IERC20Permit
- IERC20
- SafeERC20
- Vault

Description

Capabilities

Components

FrogPart

Version	Contracts	Libraries	Interfaces	Abstract
1.0	1	3	5	5

GeneScience

Version	Contracts	Libraries	Interfaces	Abstract
1.0	1	3	0	0

GenesisFrog

Version	Contracts	Libraries	Interfaces	Abstract
1.0	2	4	5	4

Marking

Version	Contracts	Libraries	Interfaces	Abstract
1.0	2	3	5	2

MysteryBox

Version	Contracts	Libraries	Interfaces	Abstract
1.0	1	2	4	2

Vault

Version	Contracts	Libraries	Interfaces	Abstract
1.0	1	3	4	3

Exposed Functions

FrogPart

Version	Public	Private	External	Internal
1.0	16	6	25	44

GeneScience

Version	Public	Private	External	Internal
1.0	12	2	0	32

GenesisFrog

Version	Public	Private	External	Internal
1.0	19	5	32	54

Marking

Version	Public	Private	External	Internal
1.0	5	11	29	35

MysteryBox

Version	Public	Private	External	Internal
1.0	9	6	30	21

Vault

Version	Public	Private	External	Internal
1.0	5	10	36	35

State Variables

FrogPart

Version	Total	Public
1.0	15	6

GeneScience

Version	Total	Public
1.0	5	3

GenesisFrog

Version	Total	Public
1.0	15	4

Marking

Version	Total	Public
1.0	6	4

MysteryBox

Version	Total	Public
1.0	14	8

Vault

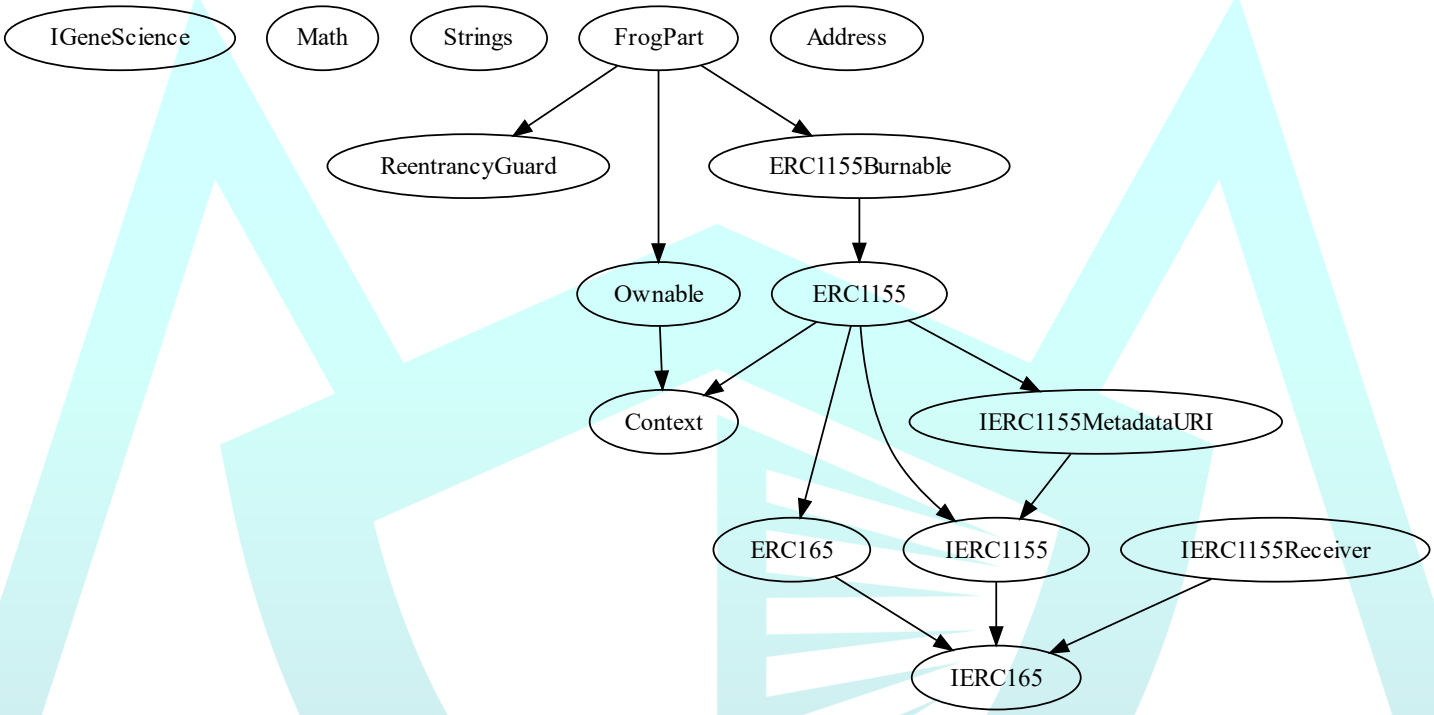
Version	Total	Public
1.0	12	4

Capabilities

Version	Solidity Versions Observed	Experimental Features	Can Receive Funds	Uses Assembly	Has Destroyable Contracts
1.0	v0.8.19		Yes	Yes	No

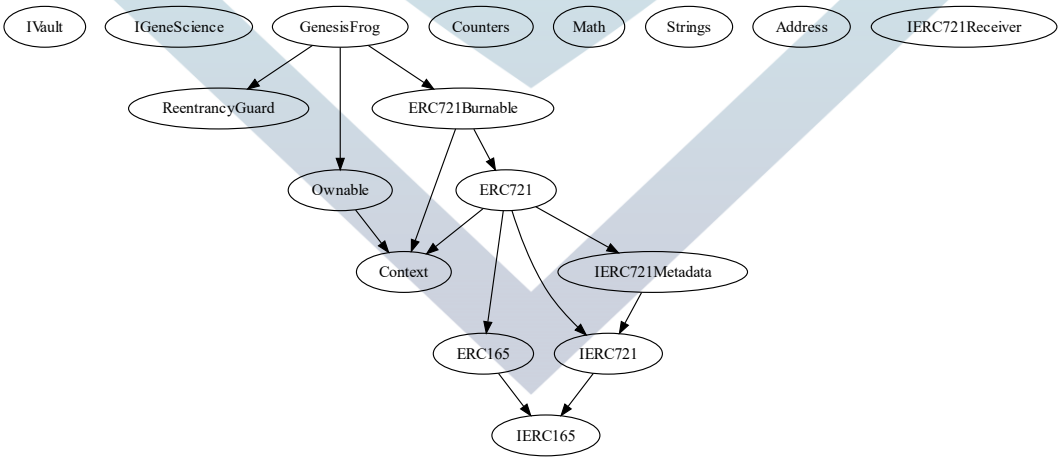
Inheritance Graph

FrogPart

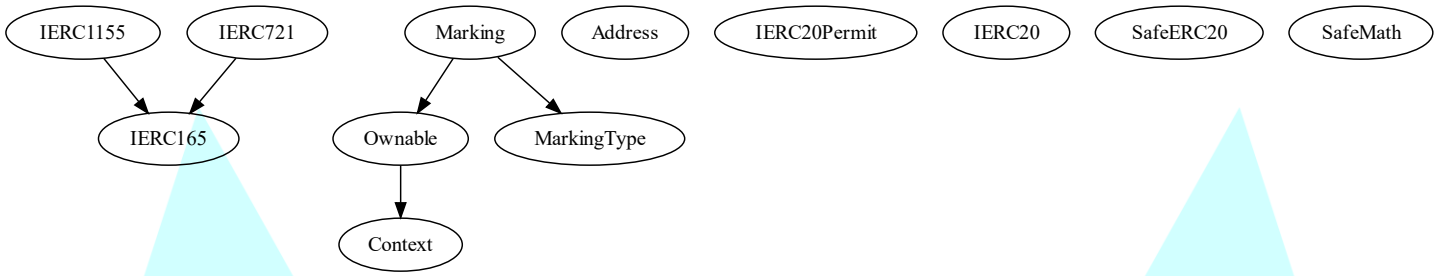


GeneScience

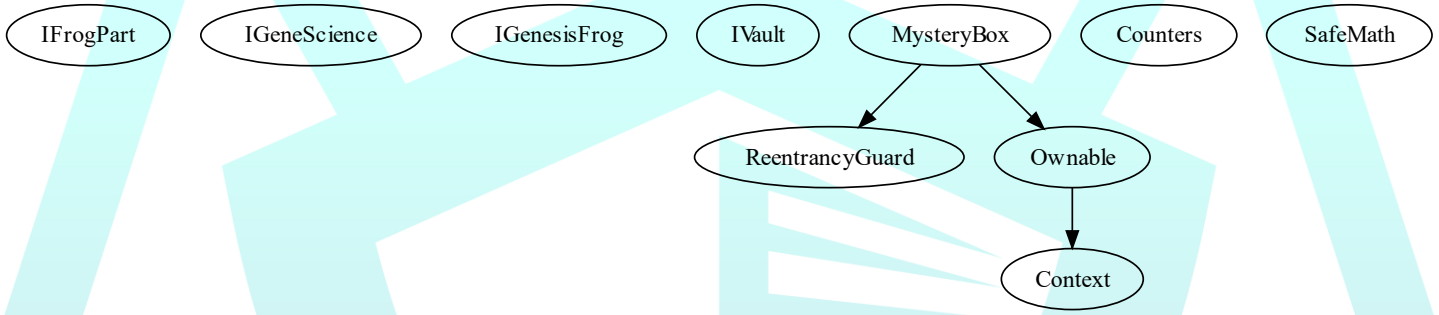
GenesisFrog



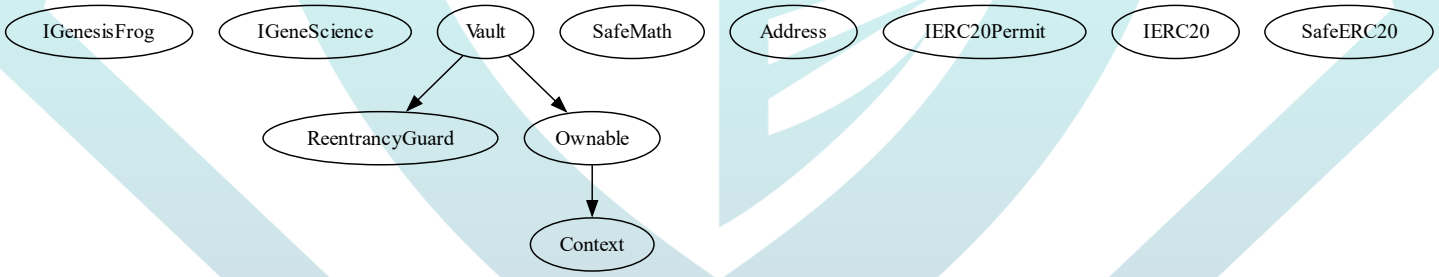
Marking



MysteryBox



Vault



Correct implementation of Token Standard

Tested	Verified
✓	✓

Overall Checkup (Smart Contract Security)

Tested	Verified
✓	✓

Function	Description	Exist	Tested	Verified
TotalSupply	Information about the total coin or token supply	✓	✓	✓
BalanceOf	Details on the account balance from a specified address	✓	✓	✓
Transfer	An action that transfers a specified amount of coin or token to a specified address	✓	✓	✓
TransferFrom	An action that transfers a specified amount of coin or token from a specified address	✓	✓	✓
Approve	Provides permission to withdraw specified number of coin or token from a specified address	✓	✓	✓

Verify Claims

Statement	Exist	Tested	Owner
Renounce Ownership	✓	✓	✓
Mint	✓	✓	✓
Burn	✓	✓	✓
Block	✓	✓	✓
Pause	–	–	–

Legend

Attribute	Symbol
Verified / Can	✓
Verified / Cannot	✗
Unverified / Not checked	⚑
Not Available	–

Write Functions of Contract

FrogPart

1. addManager (0x2d06177a)
2. born (0x56f427f9)
3. burn (0xf5298aca)
4. burnBatch (0x6b20c454)
5. burnPartBatch (0xec16892c)
6. mint (0x40c10f19)
7. removeManager (0xac18de43)
8. renounceOwnership (0x715018a6)
9. safeBatchTransferFrom (0x2eb2c2d6)
10. safeTransferFrom (0xf242432a)
11. setApprovalForAll (0xa22cb465)
12. transferOwnership (0xf2fde38b)

GenesisFrog

1. approve (0x095ea7b3)
2. born (0xf46cb4fc)
3. burn (0x42966c68)
4. burnFrog (0xfc2c1aac)
5. mixPartGenes (0x4a9cd8b0)
6. renounceOwnership (0x715018a6)
7. safeTransferFrom (0x42842e0e)
8. safeTransferFrom (0xb88d4fde)
9. setApprovalForAll (0xa22cb465)
10. setVault (0x6817031b)
11. transferFrom (0x23b872dd)
12. transferOwnership (0xf2fde38b)
13. upgradeFrog (0x14846314)

Marking

1. cancel (0xbbbeaf00)
2. execute (0x7c2a9c5b)
3. renounceOwnership (0x715018a6)
4. setOperate (0x01478e80)
5. setSigner (0x6c19e783)
6. transferOwnership (0xf2fde38b)

MysteryBox

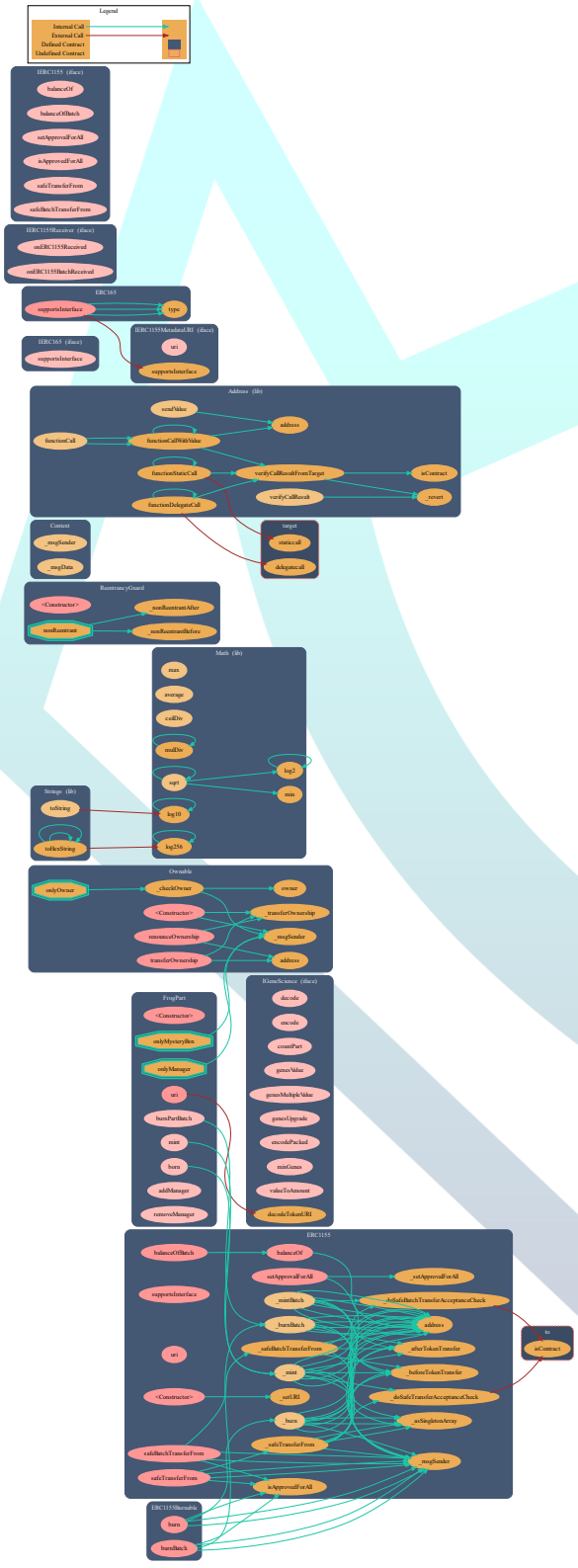
1. mixFrog (0x57a00155)
2. openBox (0xedb5ca1e)
3. openFreeBox (0x91de75bb)
4. renounceOwnership (0x715018a6)
5. setFreeSigner (0x0dee2aa3)
6. setFrogPart (0x20e8e42c)
7. setGenesisFrog (0x0448f15f)
8. setStartDate (0x82d95df5)
9. setVault (0x6817031b)
10. transferOwnership (0xf2fde38b)

Vault

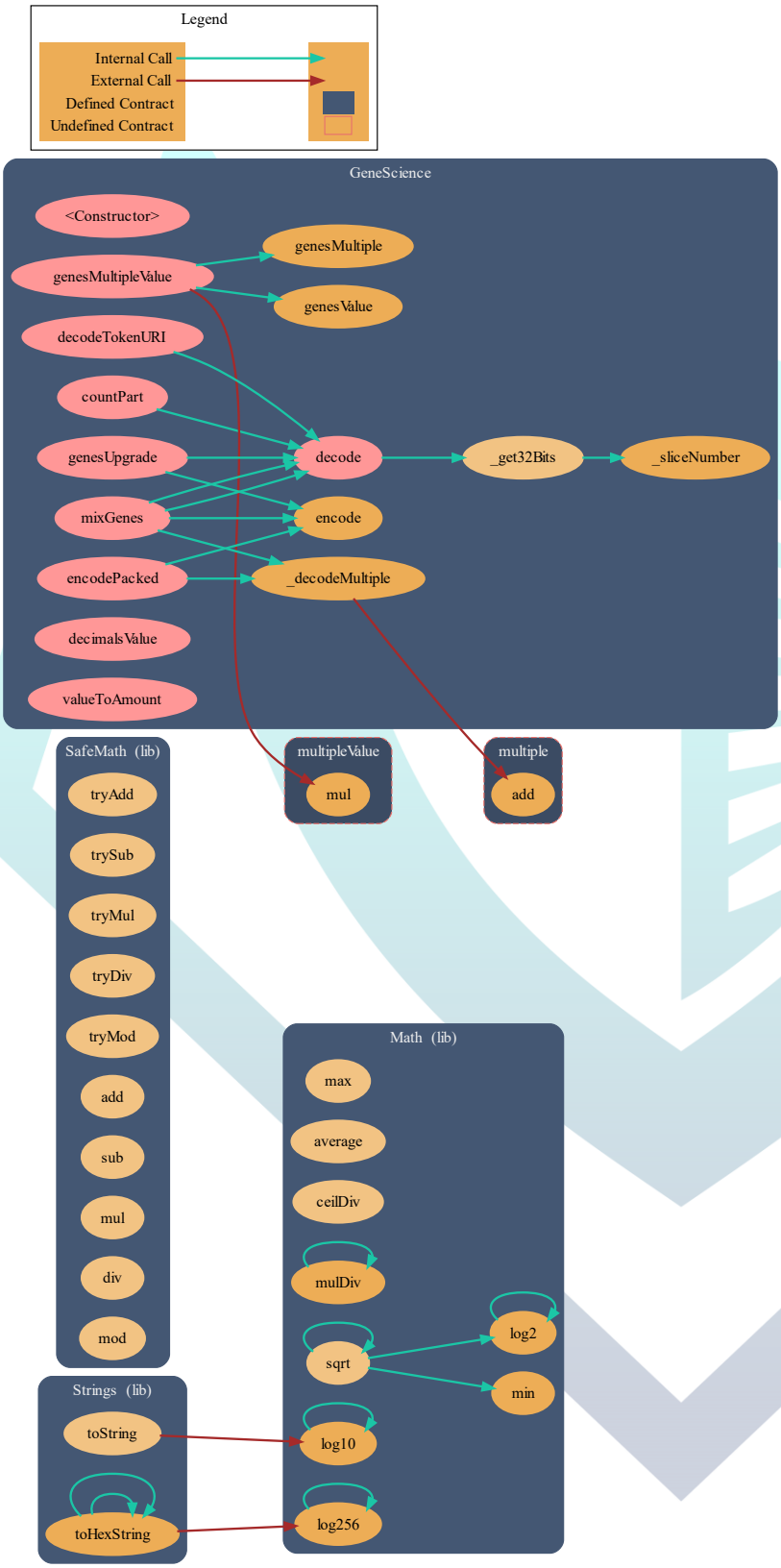
1. addBatchBlack (0x06fcfe45)
2. addFrog (0xc720f77a)
3. burnFrog (0xfc2c1aac)
4. claimBatch (0x62abebce)
5. claimByMystery (0x379d4463)
6. deposit (0xf45346dc)
7. removeBatchBlack (0x6db5c0a0)
8. renounceOwnership (0x715018a6)
9. setDailyPercent (0x6c6b4566)
10. setDeveloper (0xff70fa49)
11. transferOwnership (0xf2fde38b)
12. upgradeFrog (0x14846314)
13. withdrawBudget (0x2cc81a61)

Call Graph

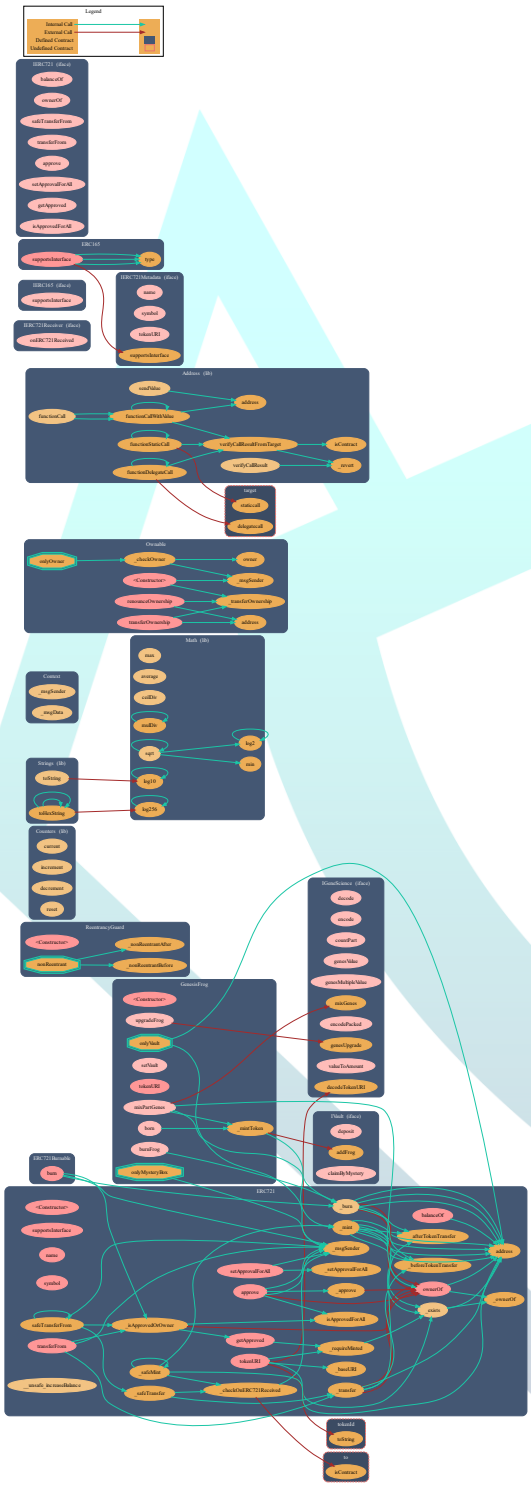
FrogPart



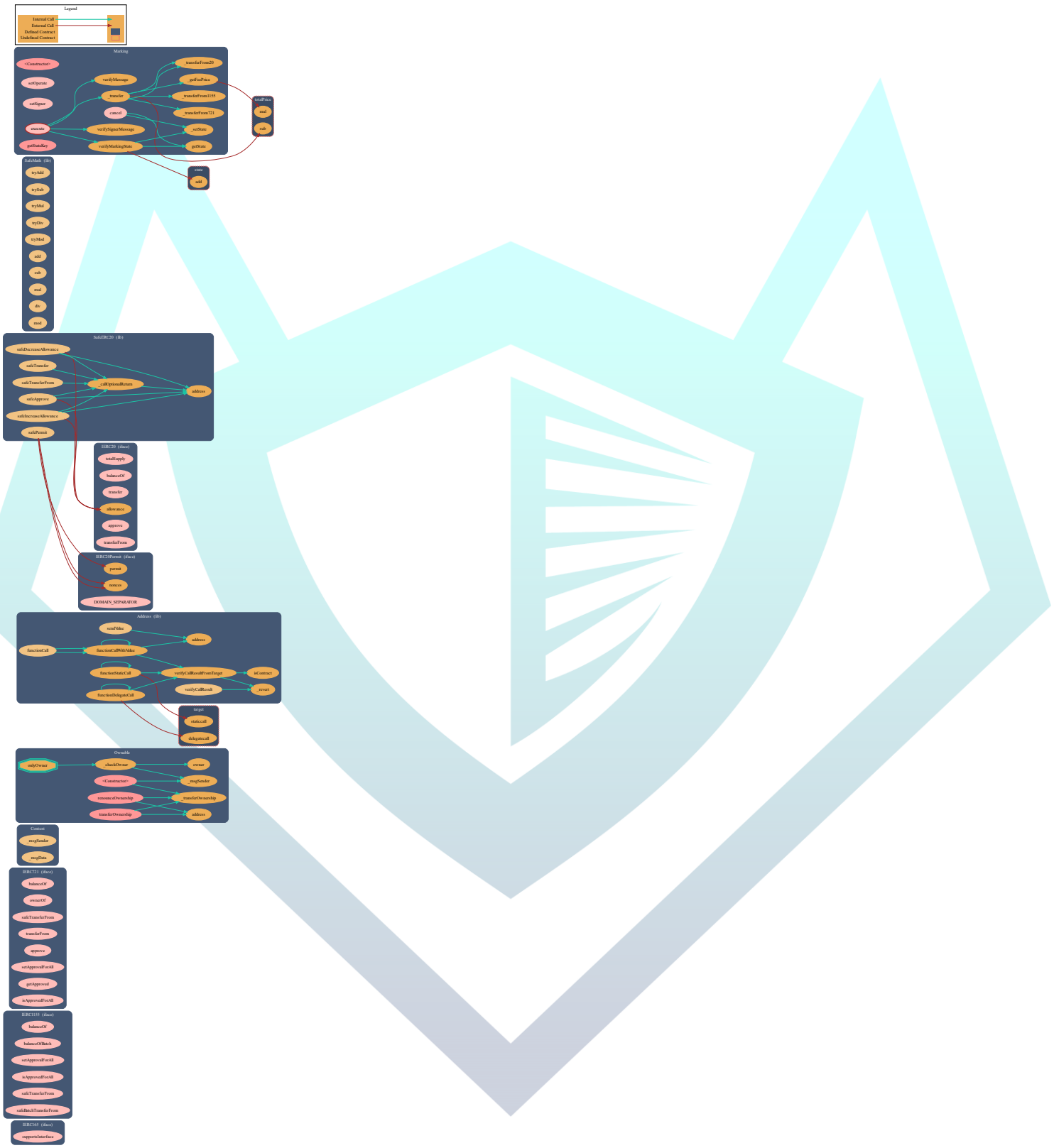
GeneScience



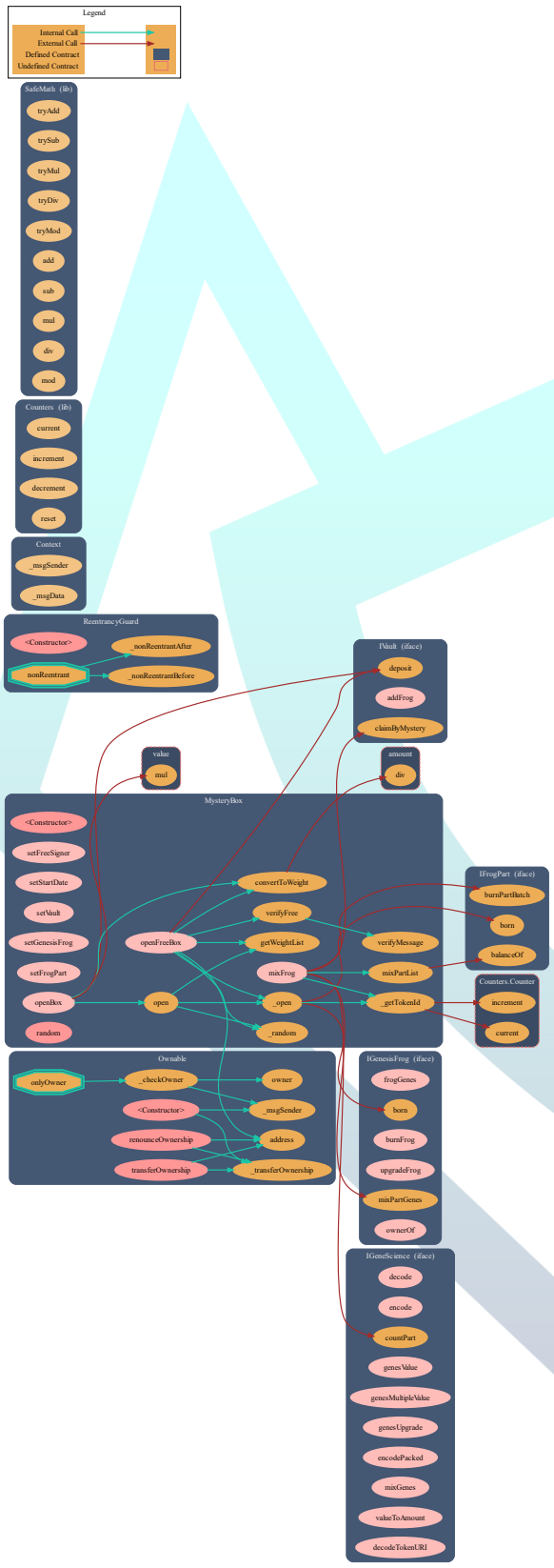
GenesisFrog



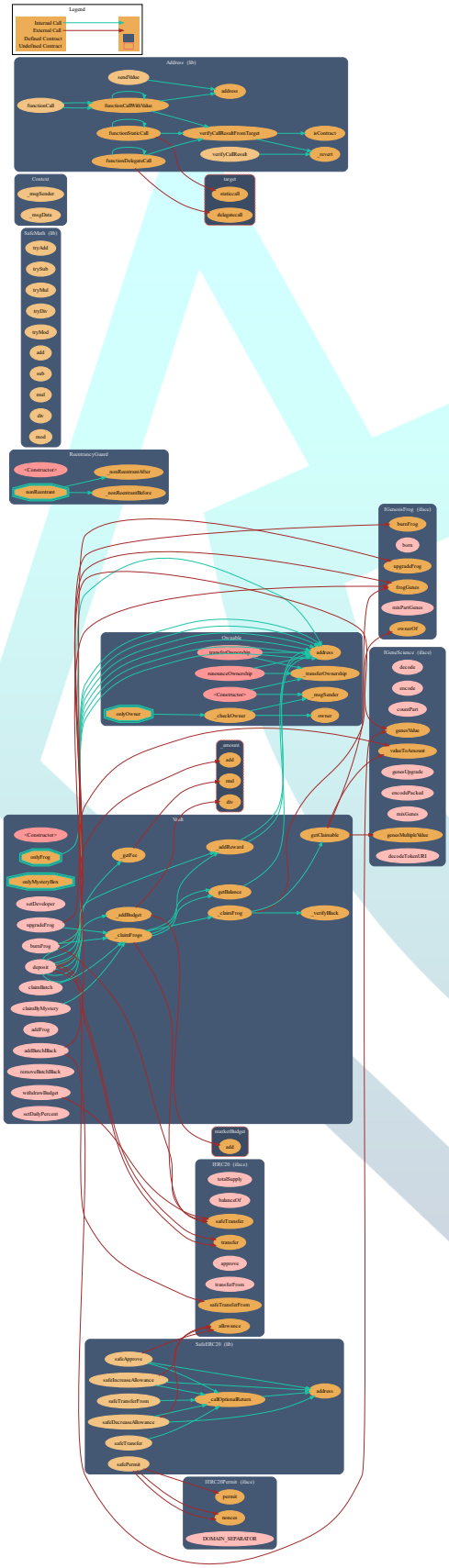
Marking


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
Marking
<Constructor>
setOperate
setSigner
execute
getStateKey
verifyMessage
_transfer
_cancel
verifySignerMessage
verifyMarkingState
transferFrom20
getFeePrice
transferFrom1155
_transferFrom721
setState
getState
totalPrice
mul
sub
state
add
SafeMath (lib)
tryAdd
trySub
tryMul
tryDiv
tryMod
add
sub
mul
div
mod
SafeERC20 (lib)
safeDecreaseAllowance
safeTransfer
safeTransferFrom
safeApprove
safeIncreaseAllowance
safePermit
_callOptionalReturn
address
IERC20 (iface)
totalSupply
balanceOf
transfer
allowance
approve
transferFrom
IERC20Permit (iface)
permit
nonces
DOMAIN_SEPARATOR
Address (lib)
sendValue
functionCall
functionCallWithValue
functionStaticCall
functionDelegateCall
verifyCallResultFromTarget
verifyCallResult
isContract
revert
target
staticcall
delegatecall
Ownable
onlyOwner
_checkOwner
<Constructor>
renounceOwnership
transferOwnership
owner
_msgSender
_transferOwnership
address
Context
_msgSender
_msgData
IERC721 (iface)
balanceOf
ownerOf
safeTransferFrom
transferFrom
approve
setApprovalForAll
getApproved
isApprovedForAll
IERC1155 (iface)
balanceOf
balanceOfBatch
setApprovalForAll
isApprovedForAll
safeTransferFrom
safeBatchTransferFrom
IERC165 (iface)
supportsInterface

MysteryBox


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
SafeMath (lib)
tryAdd
trySub
tryMul
tryDiv
tryMod
add
sub
mul
div
mod
Counters (lib)
current
increment
decrement
reset
Context
_msgSender
_msgData
ReentrancyGuard
<Constructor>
nonReentrant
_nonReentrantAfter
_nonReentrantBefore
IVault (iface)
deposit
addFrog
claimByMystery
value
mul
amount
div
MysteryBox
<Constructor>
setFreeSigner
setStartDate
setVault
setGenesisFrog
setFrogPart
openBox
random
convertToWeight
verifyFree
openFreeBox
getWeightList
mixFrog
open
_open
_random
verifyMessage
mixPartList
_getTokenId
IFrogPart (iface)
burnPartBatch
born
balanceOf
Counters.Counter
increment
current
Ownable
onlyOwner
_checkOwner
<Constructor>
renounceOwnership
transferOwnership
owner
_msgSender
address
_transferOwnership
IGenesisFrog (iface)
frogGenes
born
burnFrog
upgradeFrog
mixPartGenes
ownerOf
IGeneScience (iface)
decode
encode
countPart
genesValue
genesMultipleValue
genesUpgrade
encodePacked
mixGenes
valueToAmount
decodeTokenURI

Vault


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
Address (lib)
sendValue
address
functionCall
functionCallWithValue
functionStaticCall
verifyCallResultFromTarget
isContract
functionDelegateCall
verifyCallResult
_revert
Context
_msgSender
_msgData
target
staticcall
delegatecall
SafeMath (lib)
tryAdd
trySub
tryMul
tryDiv
tryMod
add
sub
mul
div
mod
ReentrancyGuard
<Constructor>
_nonReentrantAfter
nonReentrant
_nonReentrantBefore
IGenesisFrog (iface)
burnFrog
hunt
upgradeFrog
frogGenes
mintPartGenes
ownerOf
Ownable
_transferOwnership
renounceOwnership
<Constructor>
onlyOwner
address
transferOwnership
_msgSender
_checkOwner
owner
IGeneScience (iface)
decode
encode
countPart
genesValue
valueToAmount
geneUpgrade
encodePacked
mintGenes
geneMultipleValue
decodeTokenURI
_amount
add
mul
div
Vault
<Constructor>
onlyFrog
onlyMysteryBox
setDeveloper
upgradeFrog
burnFrog
deposit
claimBatch
claimMyMystery
addFrog
addBatchBlock
removeBatchBlock
withdrawBudget
setDailyPercent
_getFee
_addReward
getClaimable
getBalance
addBudget
_claimFrogs
claimFrog
_verifyBlock
marketBudget
add
IERC20 (iface)
totalSupply
balanceOf
safeTransfer
transfer
approve
transferFrom
safeTransferFrom
allowance
SafeERC20 (lib)
safeApprove
safeIncreaseAllowance
safeTransferFrom
safeDecreaseAllowance
safeTransfer
safePermit
callOptionalReturn
address
IERC20Permit (iface)
permit
nonces
DOMAIN_SEPARATOR

SWC Attacks

ID	Title	Status
SWC-136	Unencrypted Private Data On-Chain	PASSED
SWC-135	Code With No Effects	PASSED
SWC-134	Message call with hardcoded gas amount	PASSED
SWC-133	Hash Collisions with Multiple Variable Length Arguments	PASSED
SWC-132	Unexpected Ether balance	PASSED
SWC-131	Presence of unused variables	PASSED
SWC-130	Right-To Left Override control character (U+202E)	PASSED
SWC-129	Typographical Error	PASSED
SWC-128	DoS With Block Gas Limit	PASSED
SWC-127	Arbitrary Jump with Function Type Variable	PASSED
SWC-126	Insufficient Gas Griefing	PASSED
SWC-125	Incorrect Inheritance Order	PASSED
SWC-124	Write to Arbitrary Storage Location	PASSED
SWC-123	Requirement Violation	PASSED
SWC-122	Lack of Proper Signature Verification	PASSED
SWC-121	Missing Protection against Signature Replay Attacks	PASSED
SWC-120	Weak Sources of Randomness from Chain Attributes	PASSED
SWC-119	Shadowing State Variables	PASSED
SWC-118	Incorrect Constructor Name	PASSED
SWC-117	Signature Malleability	PASSED
SWC-116	Block values as a proxy for time	PASSED
SWC-115	Authorization through tx.origin	PASSED
SWC-114	Transaction Order Dependence	PASSED
SWC-113	DoS with Failed Call	PASSED
SWC-112	Delegate call to Untrusted Callee	PASSED
SWC-111	Use of Deprecated Solidity Functions	PASSED

SWC-110	Assert Violation	**PASSED**
SWC-109	Uninitialized Storage Pointer	**PASSED**
SWC-108	State Variable Default Visibility	**PASSED**
SWC-107	Reentrancy	**PASSED**
SWC-106	Unprotected SELFDESTRUCT Instruction	**PASSED**
SWC-105	Unprotected Ether Withdrawal	**PASSED**
SWC-104	Unchecked Call Return Value	**PASSED**
SWC-103	Floating Pragma	**PASSED**
SWC-102	Outdated Compiler Version	**PASSED**
SWC-101	Integer Overflow and Underflow	**PASSED**
SWC-100	Function Default Visibility	**PASSED**

Audit Result

THIS PROJECT IS AUDITED VIA LOCAL FILE AND NOT YET DEPLOYED IN LIVE NET

Low Issues

A floating pragma is set (SWC-103)	L: 1	GeneScience.sol Vault.sol FrogPart.sol Marking.sol MysteryBox.sol
State variable visibility is set (SWC-108)	L: 17	GenesisFrog.sol
State variable visibility is set (SWC-108)	L: 16, 17, 18, 26	Vault.sol
State variable visibility is set (SWC-108)	L: 23, 39	MysteryBox.sol

Findings

File: GenesisFrog, Vault.sol, MysteryBox.sol

Description:

State variable visibility is not set (SWC-108)

Suggestion:

Variables can be specified as being public, internal, or private. Explicitly define visibility for all state variables.

File: GeneScience, Vault.sol, FrogPart.sol, Marking.sol, MysteryBox.sol

Description:

A floating pragma is set (SWC-103)

Suggestion:

Specific version to ensure that the bytecode produced does not vary between builds.

Audit Comments

- FrogPart
  - Owner can renounce and transfer ownership
  - Owner can add/remove manager addresses
  - MysteryBox address can mint NFTs
  - MysteryBox address can burn NFTs
  - Manager addresses can mint NFTs
- GenesisFrog
  - Vault address can burn NFT
  - Vault address can upgrade frog NFT
  - MysteryBox address can mint frog NFT
  - MysteryBox can mix frog genes
- Marking
  - Owner can renounce and transfer ownership
  - Owner change operator address
  - Owner can change signer address
- MysteryBox
  - Owner can renounce and transfer ownership
  - Owner can change free signer address
  - Owner can update start date once
  - Owner can change vault address
  - Owner can update genesis frog address
  - Owner can update frog part address
- Vault
  - Owner can renounce and transfer ownership
  - Owner can change developer address
  - Owner can block/unblock users
  - Owner can withdraw BUSD from contract
  - Owner can update daily percent for claims
  - Vault address can add frogs NFT
  - MysteryBox address can deposit BUSD to contract and investor
  - MysteryBox can claim frogs from contract

CONTRACTWOLF

Blockchain Security - Smart Contract Audits