Ether Authority

www.EtherAuthority.io
audit@etherauthority.io

SMART CONTRACT

Security Audit Report

Project:	BitDrive Protocol
Website:	https://bitdrive.finance
Platform:	Binance Smart Chain
Language:	Solidity
Date:	April 20th, 2022

Table of contents

Email: audit@EtherAuthority.io

THIS IS SECURITY AUDIT REPORT DOCUMENT AND WHICH MAY CONTAIN INFORMATION WHICH IS CONFIDENTIAL. WHICH INCLUDES ANY POTENTIAL VULNERABILITIES AND MALICIOUS CODES WHICH CAN BE USED TO EXPLOIT THE SOFTWARE. THIS MUST BE REFERRED INTERNALLY AND ONLY SHOULD BE MADE AVAILABLE TO THE PUBLIC AFTER ISSUES ARE RESOLVED.

Email: audit@EtherAuthority.io

Introduction

EtherAuthority was contracted by the BitDrive team to perform the Security audit of the BitDrive Protocol smart contracts code. The audit has been performed using manual analysis as well as using automated software tools. This report presents all the findings regarding the audit performed on April 20th, 2022.

The purpose of this audit was to address the following:

- Ensure that all claimed functions exist and function correctly.
- Identify any security vulnerabilities that may be present in the smart contract.

Project Background

The BitDrive Contracts have functions like ownerMint, _baseURI, initialize, estimategetAmountOut, bitdriveSettings, biswapSettings, pancakeSettings, changesiteFee, changeAdmin, walletOfOwner, implementation, etc. The BitDrive contract inherits the ERC1967Proxy, TransparentUpgradeableProxy, ProxyAdmin, UUPSUpgradeable standard smart contracts from the OpenZeppelin library. These OpenZeppelin contracts are considered community-audited and time-tested, and hence are not part of the audit scope.

Audit scope

Name	**Code Review and Security Analysis Report for BitDrive Protocol Smart Contracts**
Platform	**BSC / Solidity**
File 1	BitdriveMiddleware.sol
File 1 MD5 Hash	7F24E1183E118E512063678EAAB93924
File 2	TransparentUpgradeableProxy.sol
File 2 MD5 Hash	4BCD82310C7DE4D23AB8353B46F797B7
File 3	GirlsDemo.sol
File 3 MD5 Hash	D8B7D81D500BBBA8FC11ADA59CD8396C
Audit Date	April 20th, 2022
Revision Date	May 7th, 2022D

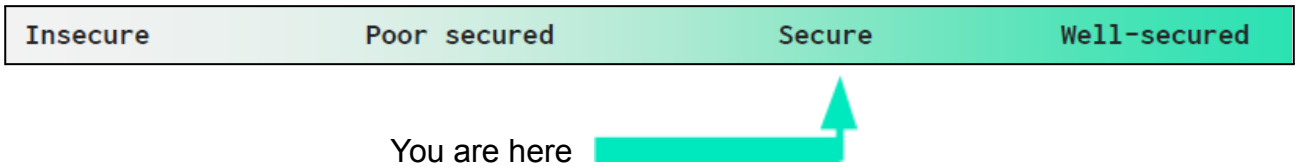
Email: audit@EtherAuthority.io

Claimed Smart Contract Features

Claimed Feature Detail	Our Observation
File 1 BitdriveMiddleware.sol • BitdriveMiddleware has functions like: initialize, swapExactTokensForETHMiddleware, etc.	**YES, This is valid.**
File 2 TransparentUpgradeableProxy.sol • This proxy contract allows the owner to change the DEX contract and to change the code logic. • TransparentUpgradeableProxy has functions like: implementation, admin, upgradeToAndCall, etc.	**YES, This is valid.**
File 3 GirlsDemo.sol • Name: GirlWithSecrets • Symbol: GWS • not Revealed URI: https://girlwithsecrets.com • Presale Cost: 0.1 ETH • Public Cost: 0.12 ETH • Maximum Supply: 3333 • Maximum Mint Amount: 5 • Public Mint Amount: unlimited	**YES, This is valid.**

Audit Summary

According to the standard audit assessment, Customer`s solidity smart contracts are **"Secured"**. Also, these contracts do contain owner control, which does not make them fully decentralized.



We used various tools like Slither, Solhint and Remix IDE. At the same time this finding is based on critical analysis of the manual audit.
All issues found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the Audit overview section. General overview is presented in AS-IS section and all identified issues can be found in the Audit overview section.

We found 0 critical, 0 high, 1 medium and 1 low and some very low level issues.
These issues are fixed / acknowledged in the revised contract code.

Investors Advice: Technical audit of the smart contract does not guarantee the ethical nature of the project. Any owner controlled functions should be executed by the owner with responsibility. All investors/users are advised to do their due diligence before investing in the project.

Technical Quick Stats

Main Category	Subcategory	Result
Contract Programming	Solidity version not specified	Passed
	Solidity version too old	Passed
	Integer overflow/underflow	Passed
	Function input parameters lack of check	Moderated
	Function input parameters check bypass	Passed
	Function access control lacks management	Passed
	Critical operation lacks event log	Passed
	Human/contract checks bypass	Passed
	Random number generation/use vulnerability	N/A
	Fallback function misuse	Passed
	Race condition	Passed
	Logical vulnerability	Passed
	Features claimed	Passed
	Other programming issues	Passed
Code Specification	Function visibility not explicitly declared	Passed
	Var. storage location not explicitly declared	Passed
	Use keywords/functions to be deprecated	Passed
	Unused code	Passed
Gas Optimization	“Out of Gas” Issue	Passed
	High consumption ‘for/while’ loop	Passed
	High consumption ‘storage’ storage	Passed
	Assert() misuse	Passed
Business Risk	The maximum limit for mintage not set	Passed
	“Short Address” Attack	Passed
	“Double Spend” Attack	Passed

Overall Audit Result: **PASSED**

Code Quality

This audit scope has 3 smart contract files. Smart contracts contain Libraries, Smart contracts, inherits and Interfaces. This is a compact and well written smart contract.

The libraries in the BitDrive Protocol are part of its 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 a specific address and its properties / methods can be reused many times by other contracts in the BitDrive Protocol.

The BitDrive team has not provided unit test scripts, which would have helped to determine the integrity of the code in an automated way.

Code parts are **not** well commented on smart contracts.

Documentation

We were given a BitDrive Protocol smart contract code in the form of a Testnet BSCScan web link. The hash of that code is mentioned above in the table.

As mentioned above, code parts are **not well** commented. So it is not easy to quickly understand the programming flow as well as complex code logic. Comments are very helpful in understanding the overall architecture of the protocol.

Use of Dependencies

As per our observation, the libraries are used in this smart contracts infrastructure that are based on well known industry standard open source projects.

Apart from libraries, its functions are used in external smart contract calls.

AS-IS overview

BitdriveMiddleware.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	__Ownable_init	internal	Passed	No Issue
3	__Ownable_init_unchained	internal	Passed	No Issue
4	owner	read	Passed	No Issue
5	onlyOwner	modifier	Passed	No Issue
6	renounceOwnership	write	access only Owner	No Issue
7	transferOwnership	write	access only Owner	No Issue
8	_transferOwnership	internal	Passed	No Issue
9	initialize	write	Passed	No Issue
10	swapExactTokensForETHMiddleware	write	Passed	No Issue
11	estimategetAmountOut	internal	Passed	No Issue
12	estimategetAmountIn	internal	Passed	No Issue
13	swapExactETHForTokensMiddleware	write	Passed	No Issue
14	swapExactTokensForTokensMiddleware	write	Passed	No Issue
15	swapTokensForExactTokensMiddleware	write	Passed	No Issue
16	bitdriveSettings	write	Function input parameters lack of check	Acknowledged
17	biswapSettings	write	Function input parameters lack of check	Acknowledged
18	pancakeSettings	write	Function input parameters lack of check	Acknowledged
19	changesiteFee	write	Function input parameters lack of check	Acknowledged
20	changeAdmin	write	Passed	No Issue

TransparentUpgradeableProxy.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	_implementation	internal	Passed	No Issue
3	ifAdmin	modifier	Passed	No Issue
4	admin	external	access if Admin	No Issue
5	implementation	external	access if Admin	No Issue
6	changeAdmin	external	access if Admin	No Issue
7	upgradeTo	external	access if Admin	No Issue
8	upgradeToAndCall	external	access if Admin	No Issue
9	_admin	internal	Passed	No Issue
10	_beforeFallback	internal	Passed	No Issue

GirlsDemo.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	supportsInterface	read	Passed	No Issue
3	tokenOfOwnerByIndex	read	Passed	No Issue
4	totalSupply	read	Passed	No Issue
5	tokenByIndex	read	Passed	No Issue
6	_beforeTokenTransfer	internal	Passed	No Issue
7	_addTokenToOwnerEnumeration	write	Passed	No Issue
8	_addTokenToAllTokensEnumeration	write	Passed	No Issue
9	_removeTokenFromOwnerEnumeration	write	Passed	No Issue
10	_removeTokenFromAllTokensEnumeration	write	Passed	No Issue
11	owner	read	Passed	No Issue
12	onlyOwner	modifier	Passed	No Issue
13	renounceOwnership	write	access only Owner	No Issue
14	transferOwnership	write	access only Owner	No Issue
15	_setOwner	write	Passed	No Issue
16	_baseURI	internal	Passed	No Issue
17	ownerMint	write	Passed	No Issue
18	tokenMint	write	Passed	No Issue
19	isWhitelisted	read	Passed	No Issue
20	clearWhitelistedAddresses	write	access only Owner	No Issue
21	addWhitelistedAddresses	write	access only Owner	No Issue
22	addSingleWhitelistedAddresse	write	access only Owner	No Issue

23	removeWhitelistedAddresses	write	access only Owner	No Issue
24	getWhitelistedAddresses	write	access only Owner	No Issue
25	getWhitelistedAddressesLength	write	access only Owner	No Issue
26	isOwner	read	Passed	No Issue
27	walletOfOwner	read	Passed	No Issue
28	tokenURI	read	Passed	No Issue
29	totalMintedTokens	read	Same value return by 2 different view	Removed
30	totalBalance	read	access only Owner	No Issue
31	reveal	write	access only Owner	No Issue
32	unReveal	write	access only Owner	No Issue
33	setPresaleCost	write	access only Owner	No Issue
34	setPublicCost	write	access only Owner	No Issue
35	setPresaleOff	write	access only Owner	No Issue
36	setPresaleOn	write	access only Owner	No Issue
37	setmaxMintAmount	write	access only Owner	No Issue
38	setBaseURI	write	access only Owner	No Issue
39	setNotRevealedURI	write	access only Owner	No Issue
40	setBaseExtension	write	access only Owner	No Issue
41	pause	write	access only Owner	No Issue
42	withdraw	write	Passed	No Issue

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to token 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
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
Lowest / Code Style / Best Practice	Lowest-level vulnerabilities, code style violations and info statements can't affect smart contract execution and can be ignored.

Audit Findings

Critical Severity

No Critical severity vulnerabilities were found.

High Severity

No High severity vulnerabilities were found.

Medium

(1) Function input parameters lack of check: **BitdriveMiddleware.sol**

```
function bitdriveSettings(
    uint256 _fee,
    address _router,
    uint256 _changetype
) public {
    require(msg.sender == adminAddress, "FORBIDDEN");
    require(_changetype == 1 || _changetype == 2, "FORBIDDEN");
    if (_changetype == 1) {
        bitdrivefee = _fee;
    }
    if (_changetype == 2) {
        bitdriverouter = _router;
    }
}

function biswapSettings(
    uint256 _fee,
    address _router,
    uint256 _changetype
) public {
    require(msg.sender == adminAddress, "FORBIDDEN");
    require(_changetype == 1 || _changetype == 2, "FORBIDDEN");
    if (_changetype == 1) {
        biswapfee = _fee;
    }
    if (_changetype == 2) {
        biswaprouter = _router;
    }
}

function pancakeSettings(
    uint256 _fee,
    address _router,
    uint256 _changetype
) public {
    require(msg.sender == adminAddress, "FORBIDDEN");
    require(_changetype == 1 || _changetype == 2, "FORBIDDEN");
    if (_changetype == 1) {
        pancakefee = _fee;
    }
    if (_changetype == 2) {
        pancakerouter = _router;
    }
}
```

```
function changesiteFee(uint256 _sitefee) public {
    require(msg.sender == adminAddress, "FORBIDDEN");
    sitefee = _sitefee;
}
```

The sitefee must have some maximum limit and should be greater than biswapfee, bitdrivefee, pancakefee.

The biswapfee, bitdrivefee, pancakefee should not be greater than sitefee.

Resolution: We suggest validating the fees.

Status: Acknowledged

Low

(1) Infinite loop possibility: **GirlsDemo.sol**

```
function removeWhitelistedAddresses(address _user) public onlyOwner {
    for (uint256 i =0; i < whitelistedAddresses.length; i++) {
        if(whitelistedAddresses[i] == _user){
            whitelistedAddresses[i] = whitelistedAddresses[whitelistedAddresses.length-1];
            whitelistedAddresses.pop();
        }
    }
}
```

```
function isWhitelisted(address _user) public view returns (bool) {
    for (uint256 i = 0; i < whitelistedAddresses.length; i++) {
        if (whitelistedAddresses[i] == _user) {
            return true;
        }
    }
    return false;
}
```

If there are so many whitelisted addresses, then this logic will fail, as it might hit the block's gas limit.

Resolution: We suggest using mapping for whitelistedAddresses.

Status: Fixed

Very Low / Informational / Best practices:

(1) Same value return from two different view: **GirlsDemo.sol**

```
function totalMintedTokens() public view returns(uint256) {
  uint256 supply = totalSupply();
  return supply;
}
```

The totalMintedTokens and totalSupply are returning the same value - totalSupply.

Resolution: We suggest removing totalMintedTokens.

Status: Fixed

(2) Missing error message: **GirlsDemo.sol**

```
// public
function tokenMint(uint256 _mintAmount) public payable {
  require(!paused);
  uint256 supply = totalSupply();
  require(_mintAmount > 0);
  require(_mintAmount <= maxMintAmount);
  require(supply + _mintAmount <= maxSupply);
  if(isOwner(msg.sender)){
      require(msg.value == 0);
  }else{
      if(preSale){
          require(isWhitelisted(msg.sender));
          require(msg.value >= _mintAmount * presaleCost);
      }else{
          require(msg.value >= _mintAmount * publicCost);
      }
  }
  for (uint256 i = 1; i <= _mintAmount; i++) {
    _safeMint(msg.sender, supply + i);
  }
}
```

There are no error messages for required statements.

Resolution: We suggest adding relevant error messages to get failure of the transaction.

Status: Fixed

(3) Ignore Payable: **GirlsDemo.sol**

```
function withdraw() public payable onlyOwner {

  (bool hs, ) = payable(owner()).call{value: address(this).balance}("");
      require(hs);

}
```

```
// public
function ownerMint(uint256 _mintAmount) public onlyOwner payable {
  uint256 supply = totalSupply();
  for (uint256 i = 1; i <= _mintAmount; i++) {
    _safeMint(msg.sender, supply + i);
  }
}
```

The withdraw and ownerMint functions should not be payable. As they are only for the owner and the fund will be transferred to the owner by withdraw function.

Resolution: We suggest removing the payable keyword and define these functions as simple functions.

Status: Fixed

(4) Make variable constant: **GirlsDemo.sol**

```
uint256 public maxSupply = 500;
```

The maxSupply is unchanged in contract. So, please make it constant. It will save some gas.

Resolution: We suggest making this variable as constant to save some gas.

Status: Fixed

Centralization

This smart contract has some functions which can be executed by the Admin (Owner) only. If the admin wallet private key would be compromised, then it would create trouble. Following are Admin functions:

- ownerMint: GirlsDemo Owner can mint amount.
- addWhitelistedAddresses: GirlsDemo Owner can add multiple addresses in whitelist.
- addSingleWhitelistedAddresse: GirlsDemo Owner can add single address in whitelist.
- removeWhitelistedAddresses: GirlsDemo Owner can remove multiple addresses from whitelist.
- getWhitelistedAddresses: GirlsDemo Owner can get list of whitelisted addresses.
- getWhitelistedAddressesLength: GirlsDemo Owner can get length of whitelisted addresses.
- reveal: GirlsDemo Owner can set true status.
- unReveal: GirlsDemo Owner can set false status.
- setPresaleCost: GirlsDemo Owner can set presale cost.
- setPublicCost: GirlsDemo Owner can set public cost.
- setPresaleOff: GirlsDemo Owner can set presale off status false.
- setPresaleOn: GirlsDemo Owner can set presale on status true.
- setmaxMintAmount: GirlsDemo Owner can set maximum mint amount.
- setBaseURI: GirlsDemo Owner can set baseURI.
- setNotRevealedURI: GirlsDemo Owner can set not revealed URI.
- setBaseExtension: GirlsDemo Owner can set base extension value.
- pause: GirlsDemo Owner can set pause state.
- withdraw: GirlsDemo Owner can withdraw amount.
- admin: TransparentUpgradeableProxy admin can return the current admin.
- implementation: TransparentUpgradeableProxy admin can return the current implementation.
- changeAdmin: TransparentUpgradeableProxy admin can change the admin of the proxy.

- upgradeTo: TransparentUpgradeableProxy admin can upgrade the implementation of the proxy.
- upgradeToAndCall: TransparentUpgradeableProxy admin can upgrade the implementation of the proxy, and then call a function from the new implementation as specified by `data`, which should be an encoded function call. This is useful to initialize new storage variables in the proxied contract.
- biswapSettings: BitdriveMiddleware owner can set fee, router address, changetype.
- pancakeSettings: BitdriveMiddleware owner can set fee, router address, changetype.
- changesiteFee: BitdriveMiddleware owner can change fee site.
- changeAdmin: BitdriveMiddleware owner can change admin address.
- bitdriveSettings: BitdriveMiddleware owner can set fee, router address, changetype.

To make the smart contract 100% decentralized, we suggest renouncing ownership in the smart contract once its function is completed.

Conclusion

We were given a contract code in the form of files. And we have used all possible tests based on given objects as files. We had observed some issues in the smart contracts and those are fixed / acknowledged. **So, the smart contracts are ready for the mainnet deployment**.

Since possible test cases can be unlimited for such smart contracts 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.

Smart contracts within the scope were manually reviewed and analyzed with static analysis tools. Smart Contract's high-level description of functionality was presented in the As-is overview section of the report.

Audit report contains all found security vulnerabilities and other issues in the reviewed code.

Security state of the reviewed contract, based on standard audit procedure scope, is **"Secured"**.

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 analyze 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 scrutinized 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

EtherAuthority.io Disclaimer

EtherAuthority team has analyzed this smart contract 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).

Due to the fact that the total number of test cases are unlimited, the audit makes no statements or warranties on 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 also suggest conducting a bug bounty program to confirm the high level of security of this smart contract.

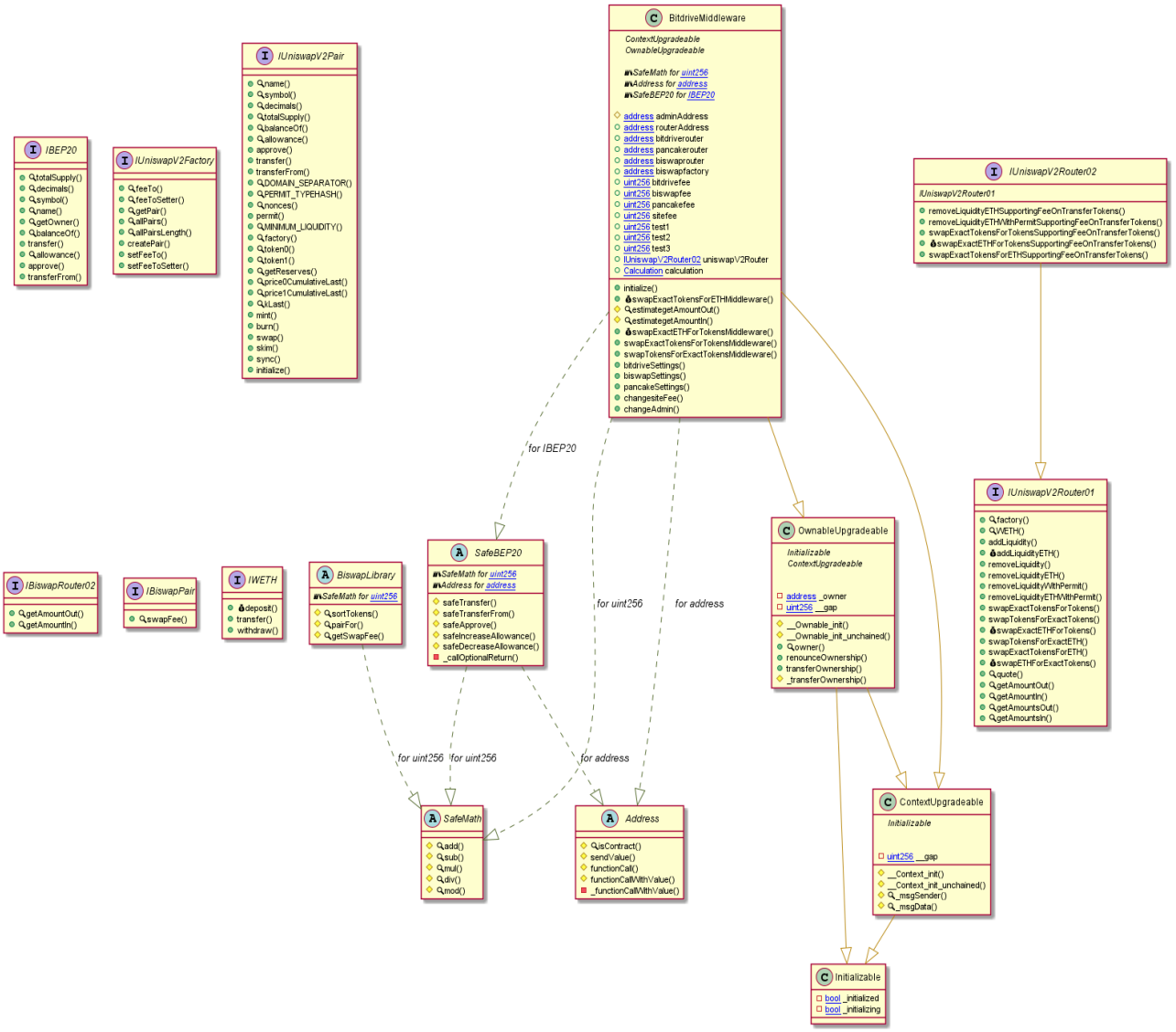
Technical Disclaimer

Smart contracts are deployed and executed on the blockchain platform. 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.

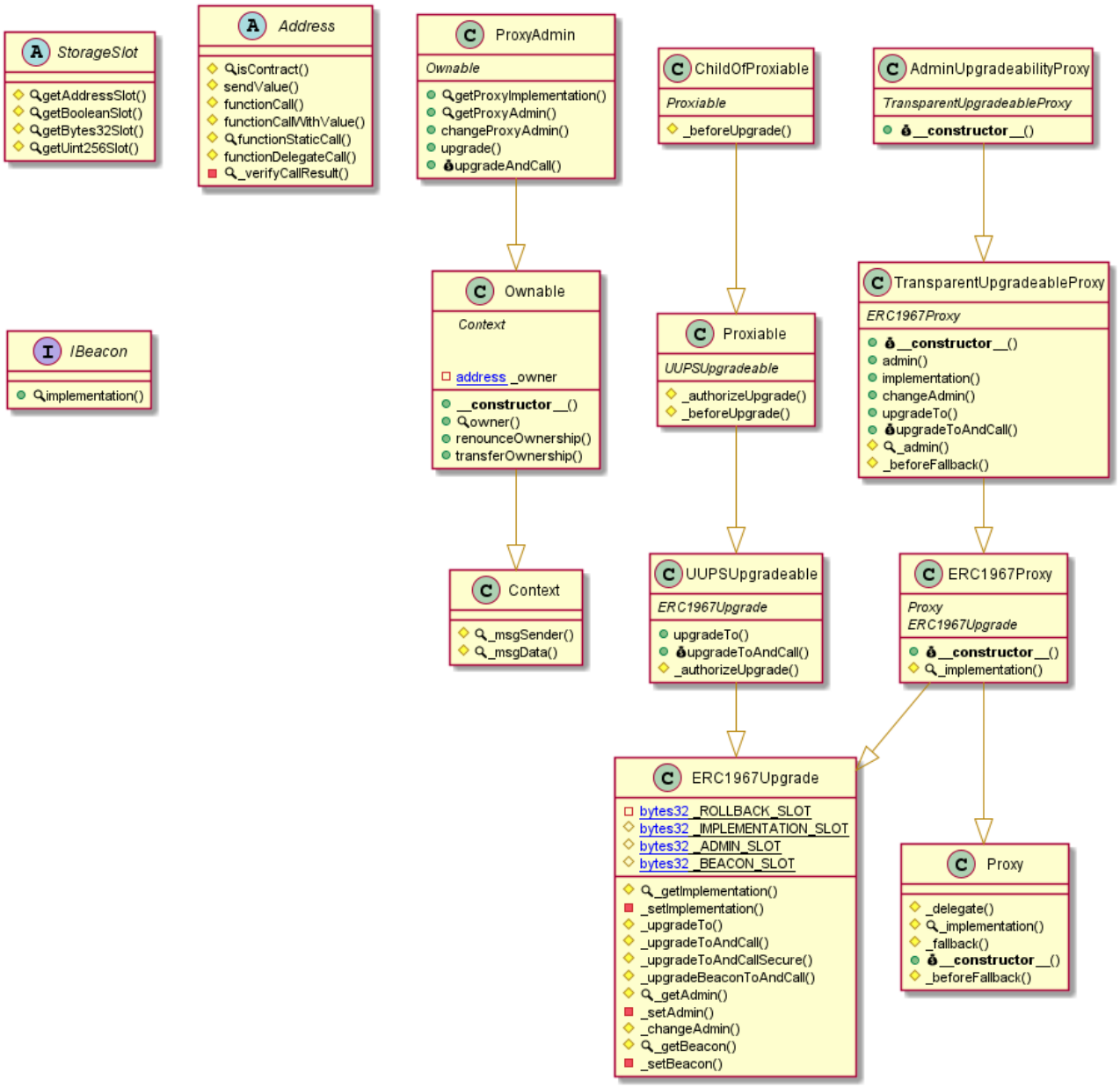
Appendix

Code Flow Diagram - BitDrive Protocol

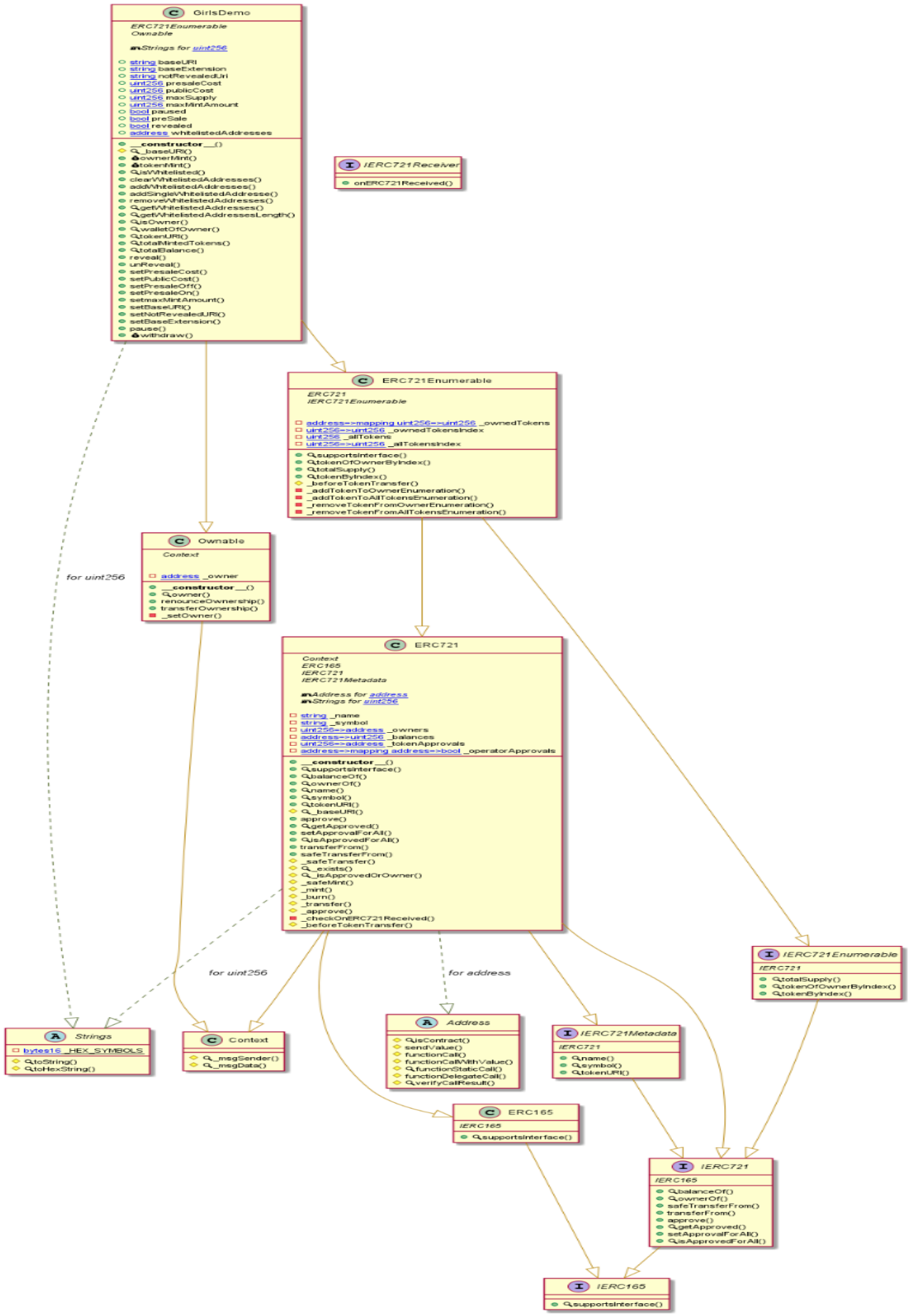
BitdriveMiddleware Diagram



TransparentUpgradeableProxy Diagram



GirlsDemo Diagram



Slither Results Log

Slither log >> BitdriveMiddleware.sol

Slither provided the following logs. We carefully checked them, and we confirm that those logs are either informative or false positive, and do not contain any security problems.

Error line numbers : 226,439,1127,1152,1169,1572,1681,1666,1655,1656,1658,1862, 1672,1678,1696,1702,1787,1863,1898

```
INFO:Detectors:
BitdriveMiddleware.swapExactTokensForETHMiddleware(address,IBEP20,address,uint256,address,address,address,uint256,uint256).uni
swapV2Router (BitdriveMiddleware.sol#1226) shadows:
        - BitdriveMiddleware.uniswapV2Router (BitdriveMiddleware.sol#1127) (state variable)
BitdriveMiddleware.swapExactETHForTokensMiddleware(address,IBEP20,address,uint256,address,address,address,uint256,uint256).uni
swapV2Router (BitdriveMiddleware.sol#1439) shadows:
        - BitdriveMiddleware.uniswapV2Router (BitdriveMiddleware.sol#1127) (state variable)
BitdriveMiddleware.swapExactTokensForTokensMiddleware(address,IBEP20,address,uint256,address,address,address,uint256).uniswapV
2Router (BitdriveMiddleware.sol#1550) shadows:
        - BitdriveMiddleware.uniswapV2Router (BitdriveMiddleware.sol#1127) (state variable)
BitdriveMiddleware.swapTokensForExactTokensMiddleware(address,IBEP20,address,uint256,address,address,address,uint256).uniswapV
2Router (BitdriveMiddleware.sol#1639) shadows:
        - BitdriveMiddleware.uniswapV2Router (BitdriveMiddleware.sol#1127) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
INFO:Detectors:
BitdriveMiddleware.bitdriveSettings(uint256,address,uint256) (BitdriveMiddleware.sol#1655-1668) should emit an event for:
        - bitdrivefee = _fee (BitdriveMiddleware.sol#1663)
BitdriveMiddleware.biswapSettings(uint256,address,uint256) (BitdriveMiddleware.sol#1670-1683) should emit an event for:
        - biswapfee = _fee (BitdriveMiddleware.sol#1678)
BitdriveMiddleware.pancakeSettings(uint256,address,uint256) (BitdriveMiddleware.sol#1685-1698) should emit an event for:
        - pancakefee = _fee (BitdriveMiddleware.sol#1693)
BitdriveMiddleware.changesiteFee(uint256) (BitdriveMiddleware.sol#1700-1703) should emit an event for:
        - sitefee = _sitefee (BitdriveMiddleware.sol#1702)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-arithmetic
INFO:Detectors:
BitdriveMiddleware.initialize(address)._admin (BitdriveMiddleware.sol#1143) lacks a zero-check on :
                - adminAddress = _admin (BitdriveMiddleware.sol#1152)
BitdriveMiddleware.bitdriveSettings(uint256,address,uint256)._router (BitdriveMiddleware.sol#1657) lacks a zero-check on :
                - bitdriverouter = _router (BitdriveMiddleware.sol#1666)
BitdriveMiddleware.biswapSettings(uint256,address,uint256)._router (BitdriveMiddleware.sol#1672) lacks a zero-check on :
                - biswaprouter = _router (BitdriveMiddleware.sol#1681)
BitdriveMiddleware.pancakeSettings(uint256,address,uint256)._router (BitdriveMiddleware.sol#1687) lacks a zero-check on :
                - pancakerouter = _router (BitdriveMiddleware.sol#1696)
BitdriveMiddleware.changeAdmin(address)._adminAddress (BitdriveMiddleware.sol#1705) lacks a zero-check on :
                - adminAddress = _adminAddress (BitdriveMiddleware.sol#1707)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
```

Error line numbers : 300,302,367,379,473,475,492,510,512,562,588

```
INFO:Detectors:
Pragma version>=0.6.4 (BitdriveMiddleware.sol#3) allows old versions
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in Address.sendValue(address,uint256) (BitdriveMiddleware.sol#367-379):
        - (success) = recipient.call{value: amount}() (BitdriveMiddleware.sol#374)
Low level call in Address._functionCallWithValue(address,bytes,uint256,string) (BitdriveMiddleware.sol#464-492):
        - (success,returndata) = target.call{value: weiValue}(data) (BitdriveMiddleware.sol#473-475)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Function ContextUpgradeable.__Context_init() (BitdriveMiddleware.sol#300-302) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (BitdriveMiddleware.sol#304) is not in mixedCase
Variable ContextUpgradeable.__gap (BitdriveMiddleware.sol#314) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init() (BitdriveMiddleware.sol#505-508) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init_unchained() (BitdriveMiddleware.sol#510-512) is not in mixedCase
Variable OwnableUpgradeable.__gap (BitdriveMiddleware.sol#562) is not in mixedCase
Function IUniswapV2Pair.DOMAIN_SEPARATOR() (BitdriveMiddleware.sol#628) is not in mixedCase
```

Error line numbers : 1155,1382,1385,1478,1488,1564,1566,1653,1655,1668,1679,1583, 1585,1698,1786,1763,1785,1708

```
swapExactTokensForETHMiddleware(address,IBEP20,address,uint256,address,address,address,uint256,uint256) should be declared ext
ernal:
        - BitdriveMiddleware.swapExactTokensForETHMiddleware(address,IBEP20,address,uint256,address,address,address,uint256,ui
nt256) (BitdriveMiddleware.sol#1155-1302)
swapExactETHForTokensMiddleware(address,IBEP20,address,uint256,address,address,address,uint256,uint256) should be declared ext
ernal:
        - BitdriveMiddleware.swapExactETHForTokensMiddleware(address,IBEP20,address,uint256,address,address,address,uint256,ui
nt256) (BitdriveMiddleware.sol#1395-1478)
swapExactTokensForTokensMiddleware(address,IBEP20,address,uint256,address,address,address,uint256) should be declared external
:
        - BitdriveMiddleware.swapExactTokensForTokensMiddleware(address,IBEP20,address,uint256,address,address,address,uint256
) (BitdriveMiddleware.sol#1480-1564)
swapTokensForExactTokensMiddleware(address,IBEP20,address,uint256,address,address,address,uint256) should be declared external
:
        - BitdriveMiddleware.swapTokensForExactTokensMiddleware(address,IBEP20,address,uint256,address,address,address,uint256
) (BitdriveMiddleware.sol#1566-1653)
bitdriveSettings(uint256,address,uint256) should be declared external:
        - BitdriveMiddleware.bitdriveSettings(uint256,address,uint256) (BitdriveMiddleware.sol#1655-1668)
biswapSettings(uint256,address,uint256) should be declared external:
        - BitdriveMiddleware.biswapSettings(uint256,address,uint256) (BitdriveMiddleware.sol#1670-1683)
pancakeSettings(uint256,address,uint256) should be declared external:
        - BitdriveMiddleware.pancakeSettings(uint256,address,uint256) (BitdriveMiddleware.sol#1685-1698)
changesiteFee(uint256) should be declared external:
        - BitdriveMiddleware.changesiteFee(uint256) (BitdriveMiddleware.sol#1700-1703)
changeAdmin(address) should be declared external:
        - BitdriveMiddleware.changeAdmin(address) (BitdriveMiddleware.sol#1705-1708)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:BitdriveMiddleware.sol analyzed (16 contracts with 75 detectors), 109 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

Slither log >> TransparentUpgradeableProxy.sol

The following slither logs identifies some issues like reentrancy, etc. We carefully checked those issues and we confirm they are false positives and do not raise any security issues.

Error line numbers : 756,681,683,674,689,583,531,517,523,529,668,586,568,24,28,47, 46,33,37,51,55,52,54,76,85,230,233

```
INFO:Detectors:
AdminUpgradeabilityProxy.constructor(address,address,bytes).admin (TransparentUpgradeableProxy.sol#756) shadows:
        - TransparentUpgradeableProxy.admin() (TransparentUpgradeableProxy.sol#691-693) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
INFO:Detectors:
Modifier TransparentUpgradeableProxy.ifAdmin() (TransparentUpgradeableProxy.sol#674-680) does not always execute _; or revertR
eference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-modifier
INFO:Detectors:
Reentrancy in ERC1967Upgrade._upgradeToAndCallSecure(address,bytes,bool) (TransparentUpgradeableProxy.sol#503-531):
        External calls:
        - Address.functionDelegateCall(newImplementation,data) (TransparentUpgradeableProxy.sol#509)
        - Address.functionDelegateCall(newImplementation,abi.encodeWithSignature(upgradeTo(address),oldImplementation)) (Trans
parentUpgradeableProxy.sol#517-523)
        Event emitted after the call(s):
        - Upgraded(newImplementation) (TransparentUpgradeableProxy.sol#529)
Reentrancy in TransparentUpgradeableProxy.constructor(address,address,bytes) (TransparentUpgradeableProxy.sol#666-669):
        External calls:
        - ERC1967Proxy(_logic,_data) (TransparentUpgradeableProxy.sol#666)
                - Address.functionDelegateCall(newImplementation,data) (TransparentUpgradeableProxy.sol#494)
                - (success,returndata) = target.delegatecall(data) (TransparentUpgradeableProxy.sol#217)
        Event emitted after the call(s):
        - AdminChanged(_getAdmin(),newAdmin) (TransparentUpgradeableProxy.sol#580)
                - _changeAdmin(admin_) (TransparentUpgradeableProxy.sol#668)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
INFO:Detectors:
StorageSlot.getAddressSlot(bytes32) (TransparentUpgradeableProxy.sol#24-28) uses assembly
        - INLINE ASM (TransparentUpgradeableProxy.sol#25-27)
StorageSlot.getBooleanSlot(bytes32) (TransparentUpgradeableProxy.sol#33-37) uses assembly
        - INLINE ASM (TransparentUpgradeableProxy.sol#34-36)
StorageSlot.getBytes32Slot(bytes32) (TransparentUpgradeableProxy.sol#42-46) uses assembly
        - INLINE ASM (TransparentUpgradeableProxy.sol#43-45)
StorageSlot.getUint256Slot(bytes32) (TransparentUpgradeableProxy.sol#51-55) uses assembly
        - INLINE ASM (TransparentUpgradeableProxy.sol#52-54)
Address.isContract(address) (TransparentUpgradeableProxy.sol#76-85) uses assembly
        - INLINE ASM (TransparentUpgradeableProxy.sol#83)
Address._verifyCallResult(bool,bytes,string) (TransparentUpgradeableProxy.sol#221-238) uses assembly
        - INLINE ASM (TransparentUpgradeableProxy.sol#230-233)
```

Error line numbers : 103,139,164,169,189,195,193,213,219,217,387,383.339,402,408, 419,438,428,440,442

```
INFO:Detectors:
Low level call in Address.sendValue(address,uint256) (TransparentUpgradeableProxy.sol#103-109):
        - (success) = recipient.call{value: amount}() (TransparentUpgradeableProxy.sol#107)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (TransparentUpgradeableProxy.sol#164-171):
        - (success,returndata) = target.call{value: value}(data) (TransparentUpgradeableProxy.sol#169)
Low level call in Address.functionStaticCall(address,bytes,string) (TransparentUpgradeableProxy.sol#189-195):
        - (success,returndata) = target.staticcall(data) (TransparentUpgradeableProxy.sol#193)
Low level call in Address.functionDelegateCall(address,bytes,string) (TransparentUpgradeableProxy.sol#213-219):
        - (success,returndata) = target.delegatecall(data) (TransparentUpgradeableProxy.sol#217)
Low level call in ProxyAdmin.getProxyImplementation(TransparentUpgradeableProxy) (TransparentUpgradeableProxy.sol#387-393):
        - (success,returndata) = address(proxy).staticcall(0x5c60da1b) (TransparentUpgradeableProxy.sol#390)
Low level call in ProxyAdmin.getProxyAdmin(TransparentUpgradeableProxy) (TransparentUpgradeableProxy.sol#402-408):
        - (success,returndata) = address(proxy).staticcall(0xf851a440) (TransparentUpgradeableProxy.sol#405)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Redundant expression "this (TransparentUpgradeableProxy.sol#254)" inContext (TransparentUpgradeableProxy.sol#248-257)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
INFO:Detectors:
renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (TransparentUpgradeableProxy.sol#294-297)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (TransparentUpgradeableProxy.sol#303-307)
getProxyImplementation(TransparentUpgradeableProxy) should be declared external:
        - ProxyAdmin.getProxyImplementation(TransparentUpgradeableProxy) (TransparentUpgradeableProxy.sol#387-393)
getProxyAdmin(TransparentUpgradeableProxy) should be declared external:
        - ProxyAdmin.getProxyAdmin(TransparentUpgradeableProxy) (TransparentUpgradeableProxy.sol#402-408)
changeProxyAdmin(TransparentUpgradeableProxy,address) should be declared external:
        - ProxyAdmin.changeProxyAdmin(TransparentUpgradeableProxy,address) (TransparentUpgradeableProxy.sol#417-419)
upgrade(TransparentUpgradeableProxy,address) should be declared external:
        - ProxyAdmin.upgrade(TransparentUpgradeableProxy,address) (TransparentUpgradeableProxy.sol#428-430)
upgradeAndCall(TransparentUpgradeableProxy,address,bytes) should be declared external:
        - ProxyAdmin.upgradeAndCall(TransparentUpgradeableProxy,address,bytes) (TransparentUpgradeableProxy.sol#440-442)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:TransparentUpgradeableProxy.sol analyzed (14 contracts with 75 detectors), 49 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

Slither log >> GirlsDemo.sol

The following Slither logs provided some issues. We carefully checked them, and we confirm that those logs are either informative or false positive, and do not contain any security problems.

Error line numbers : 1110,1111,1204,844,852,885,426,446,865,1217,257,255,441, 426,859,1229,312,138,320,326,345,338,353,364

```
INFO:Detectors:
GirlsDemo.constructor(string,string,string,string)._name (GirlsDemo.sol#1110) shadows:
        - ERC721._name (GirlsDemo.sol#498) (state variable)
GirlsDemo.constructor(string,string,string,string)._symbol (GirlsDemo.sol#1111) shadows:
        - ERC721._symbol (GirlsDemo.sol#501) (state variable)
GirlsDemo.walletOfOwner(address)._owner (GirlsDemo.sol#1204) shadows:
        - Ownable._owner (GirlsDemo.sol#1039) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
INFO:Detectors:
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).retval (GirlsDemo.sol#851)' in ERC721._checkOnERC721Rec
eived(address,address,uint256,bytes) (GirlsDemo.sol#844-865) potentially used before declaration: retval == IERC721Receiver.on
ERC721Received.selector (GirlsDemo.sol#852)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (GirlsDemo.sol#853)' in ERC721._checkOnERC721Rec
eived(address,address,uint256,bytes) (GirlsDemo.sol#844-865) potentially used before declaration: reason.length == 0 (GirlsDem
o.sol#854)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (GirlsDemo.sol#853)' in ERC721._checkOnERC721Rec
eived(address,address,uint256,bytes) (GirlsDemo.sol#844-865) potentially used before declaration: revert(uint256,uint256)(32 +
 reason,mload(uint256)(reason)) (GirlsDemo.sol#858)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables
INFO:Detectors:
Address.isContract(address) (GirlsDemo.sol#257-267) uses assembly
        - INLINE ASM (GirlsDemo.sol#263-265)
Address.verifyCallResult(bool,bytes,string) (GirlsDemo.sol#426-446) uses assembly
        - INLINE ASM (GirlsDemo.sol#438-441)
ERC721._checkOnERC721Received(address,address,uint256,bytes) (GirlsDemo.sol#844-865) uses assembly
        - INLINE ASM (GirlsDemo.sol#857-859)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
GirlsDemo.tokenURI(uint256) (GirlsDemo.sol#1217-1237) compares to a boolean constant:
        -revealed == false (GirlsDemo.sol#1229)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality
INFO:Detectors:
Address.functionCall(address,bytes) (GirlsDemo.sol#310-312) is never used and should be removed
Address.functionCall(address,bytes,string) (GirlsDemo.sol#320-326) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (GirlsDemo.sol#339-345) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (GirlsDemo.sol#353-364) is never used and should be removed
```

Error line numbers : 285,409,418,372,374,382,391,285,280,426,448,227,290,382, 391,489,418,416,1294,1297

```
Address.functionDelegateCall(address,bytes,string) (GirlsDemo.sol#409-418) is never used and should be removed
Address.functionStaticCall(address,bytes) (GirlsDemo.sol#372-374) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (GirlsDemo.sol#382-391) is never used and should be removed
Address.sendValue(address,uint256) (GirlsDemo.sol#285-290) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (GirlsDemo.sol#426-446) is never used and should be removed
Context._msgData() (GirlsDemo.sol#488-490) is never used and should be removed
ERC721._baseURI() (GirlsDemo.sol#579-581) is never used and should be removed
ERC721._burn(uint256) (GirlsDemo.sol#779-791) is never used and should be removed
Strings.toHexString(uint256) (GirlsDemo.sol#211-222) is never used and should be removed
Strings.toHexString(uint256,uint256) (GirlsDemo.sol#227-237) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version^0.8.0 (GirlsDemo.sol#4) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.0 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) (GirlsDemo.sol#285-290):
        - (success) = recipient.call{value: amount}() (GirlsDemo.sol#288)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (GirlsDemo.sol#353-364):
        - (success,returndata) = target.call{value: value}(data) (GirlsDemo.sol#362)
Low level call in Address.functionStaticCall(address,bytes,string) (GirlsDemo.sol#382-391):
        - (success,returndata) = target.staticcall(data) (GirlsDemo.sol#389)
Low level call in Address.functionDelegateCall(address,bytes,string) (GirlsDemo.sol#409-418):
        - (success,returndata) = target.delegatecall(data) (GirlsDemo.sol#416)
Low level call in GirlsDemo.withdraw() (GirlsDemo.sol#1292-1297):
        - (hs) = address(owner()).call{value: address(this).balance}() (GirlsDemo.sol#1294)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
```

Slither provided the following logs. We carefully checked them, and we confirm that those logs are either informative or false positive, and do not contain any security problems.
Error line numbers : 656,1128,1136,1157,1176,1180,1297,1268,1272,1275,1280,1280, 1288,1278,1284,554,555,560,562,572,1217,1237,586,596,610,615,627,636

```
INFO:Detectors:
Parameter ERC721.safeTransferFrom(address,address,uint256,bytes)._data (GirlsDemo.sol#656) is not in mixedCase
Parameter GirlsDemo.ownerMint(uint256)._mintAmount (GirlsDemo.sol#1128) is not in mixedCase
Parameter GirlsDemo.tokenMint(uint256)._mintAmount (GirlsDemo.sol#1136) is not in mixedCase
Parameter GirlsDemo.isWhitelisted(address)._user (GirlsDemo.sol#1157) is not in mixedCase
Parameter GirlsDemo.addSingleWhitelistedAddresse(address)._user (GirlsDemo.sol#1176) is not in mixedCase
Parameter GirlsDemo.removeWhitelistedAddresses(address)._user (GirlsDemo.sol#1180) is not in mixedCase
Parameter GirlsDemo.isOwner(address)._user (GirlsDemo.sol#1197) is not in mixedCase
Parameter GirlsDemo.walletOfOwner(address)._owner (GirlsDemo.sol#1204) is not in mixedCase
Parameter GirlsDemo.setPresaleCost(uint256)._newCost (GirlsDemo.sol#1256) is not in mixedCase
Parameter GirlsDemo.setPublicCost(uint256)._newCost (GirlsDemo.sol#1260) is not in mixedCase
Parameter GirlsDemo.setmaxMintAmount(uint256)._newmaxMintAmount (GirlsDemo.sol#1272) is not in mixedCase
Parameter GirlsDemo.setBaseURI(string)._newBaseURI (GirlsDemo.sol#1276) is not in mixedCase
Parameter GirlsDemo.setNotRevealedURI(string)._notRevealedURI (GirlsDemo.sol#1280) is not in mixedCase
Parameter GirlsDemo.setBaseExtension(string)._newBaseExtension (GirlsDemo.sol#1284) is not in mixedCase
Parameter GirlsDemo.pause(bool)._state (GirlsDemo.sol#1288) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
GirlsDemo.maxSupply (GirlsDemo.sol#1101) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
INFO:Detectors:
name() should be declared external:
        - ERC721.name() (GirlsDemo.sol#553-555)
symbol() should be declared external:
        - ERC721.symbol() (GirlsDemo.sol#560-562)
tokenURI(uint256) should be declared external:
        - ERC721.tokenURI(uint256) (GirlsDemo.sol#567-572)
        - GirlsDemo.tokenURI(uint256) (GirlsDemo.sol#1217-1237)
approve(address,uint256) should be declared external:
        - ERC721.approve(address,uint256) (GirlsDemo.sol#586-596)
setApprovalForAll(address,bool) should be declared external:
        - ERC721.setApprovalForAll(address,bool) (GirlsDemo.sol#610-615)
transferFrom(address,address,uint256) should be declared external:
        - ERC721.transferFrom(address,address,uint256) (GirlsDemo.sol#627-636)
```

Error line numbers : 641,647,926,929,1074,1083,1128,1133,1138,1155,1170,1174,1176, 1180,1187,1193,1195,1204,1215,1239,1242,1244,1246,1248,1258,122,1254

```
safeTransferFrom(address,address,uint256) should be declared external:
        - ERC721.safeTransferFrom(address,address,uint256) (GirlsDemo.sol#641-647)
tokenByIndex(uint256) should be declared external:
        - ERC721Enumerable.tokenByIndex(uint256) (GirlsDemo.sol#926-929)
renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (GirlsDemo.sol#1072-1074)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (GirlsDemo.sol#1080-1083)
ownerMint(uint256) should be declared external:
        - GirlsDemo.ownerMint(uint256) (GirlsDemo.sol#1128-1133)
tokenMint(uint256) should be declared external:
        - GirlsDemo.tokenMint(uint256) (GirlsDemo.sol#1136-1155)
clearWhitelistedAddresses() should be declared external:
        - GirlsDemo.clearWhitelistedAddresses() (GirlsDemo.sol#1166-1168)
addWhitelistedAddresses(address[]) should be declared external:
        - GirlsDemo.addWhitelistedAddresses(address[]) (GirlsDemo.sol#1170-1174)
addSingleWhitelistedAddresse(address) should be declared external:
        - GirlsDemo.addSingleWhitelistedAddresse(address) (GirlsDemo.sol#1176-1178)
removeWhitelistedAddresses(address) should be declared external:
        - GirlsDemo.removeWhitelistedAddresses(address) (GirlsDemo.sol#1180-1187)
getWhitelistedAddresses() should be declared external:
        - GirlsDemo.getWhitelistedAddresses() (GirlsDemo.sol#1189-1191)
getWhitelistedAddressesLength() should be declared external:
        - GirlsDemo.getWhitelistedAddressesLength() (GirlsDemo.sol#1193-1195)
walletOfOwner(address) should be declared external:
        - GirlsDemo.walletOfOwner(address) (GirlsDemo.sol#1204-1215)
totalMintedTokens() should be declared external:
        - GirlsDemo.totalMintedTokens() (GirlsDemo.sol#1239-1242)
totalBalance() should be declared external:
        - GirlsDemo.totalBalance() (GirlsDemo.sol#1244-1246)
reveal() should be declared external:
        - GirlsDemo.reveal() (GirlsDemo.sol#1248-1250)
unReveal() should be declared external:
        - GirlsDemo.unReveal() (GirlsDemo.sol#1252-1254)
setPresaleCost(uint256) should be declared external:
        - GirlsDemo.setPresaleCost(uint256) (GirlsDemo.sol#1256-1258)
setPublicCost(uint256) should be declared external:
```

Solidity Static Analysis

The static Analysis tool helps scan the code against hundreds of security vulnerability patterns. The tool outputs various issues, which we checked manually and confirm they do not create any negative impact and below results are false positives.

BitdriveMiddleware.sol

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: 484:16:

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: 374:27:

Gas & Economy

Gas costs:

Gas requirement of function BitdriveMiddleware.transferOwnership 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: 544:4:

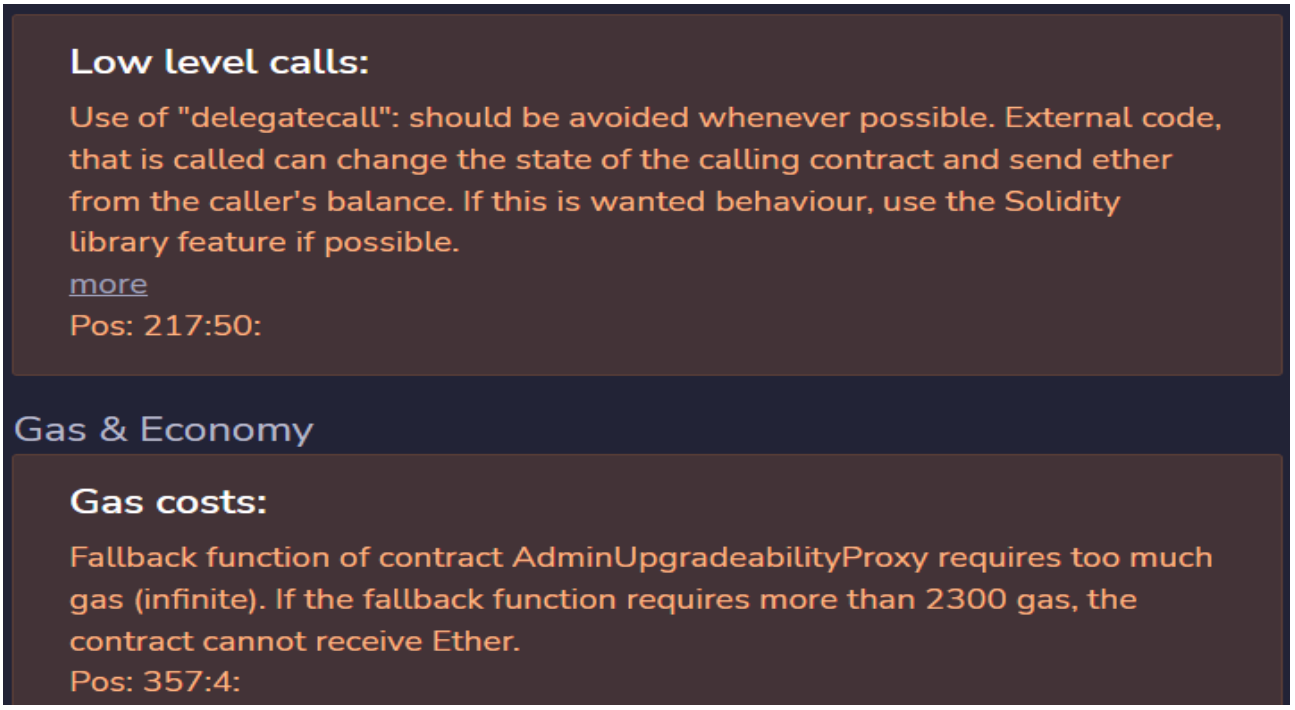
Gas costs:

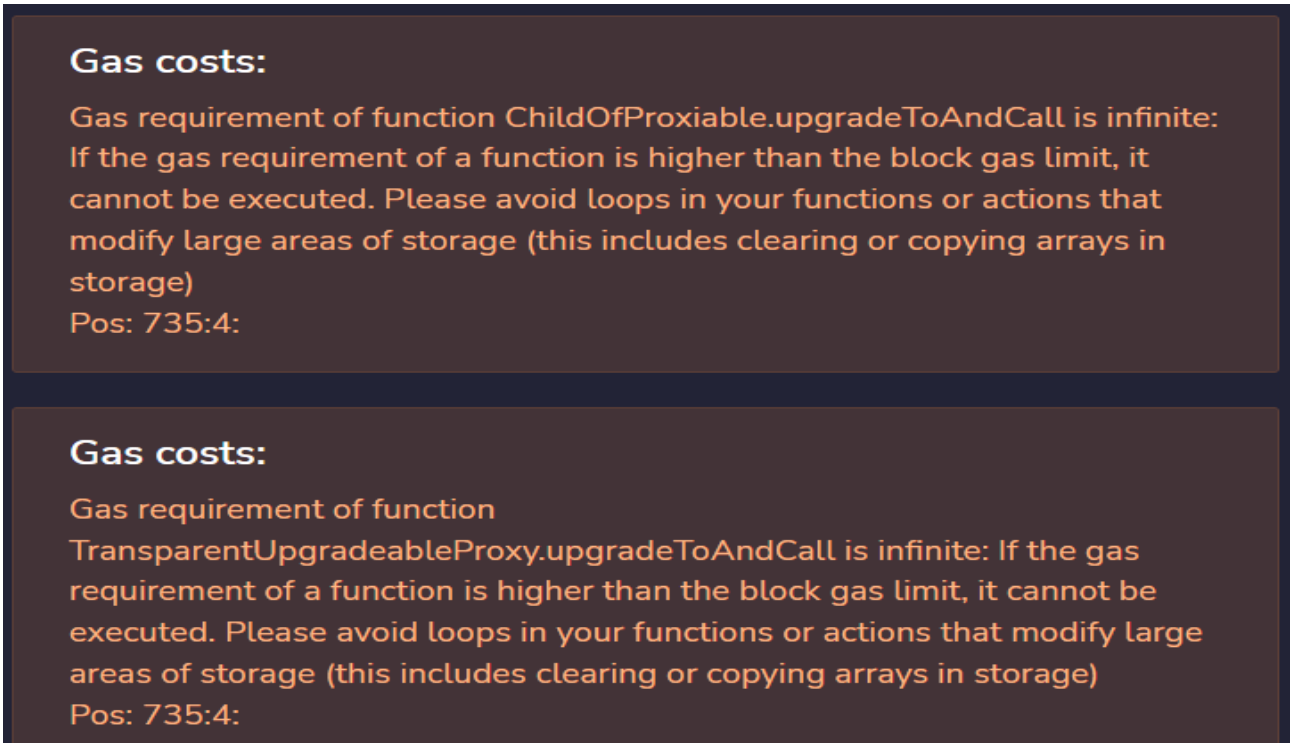
Gas requirement of function BitdriveMiddleware.initialize is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)
Pos: 1143:4:

The following issues identified by the tool, are checked manually and confirmed that they do not create any negative impact and thus below results are false positives.

The following issues identified by the tool, are checked manually and confirmed that they do not create any negative impact and thus below results are false positives.

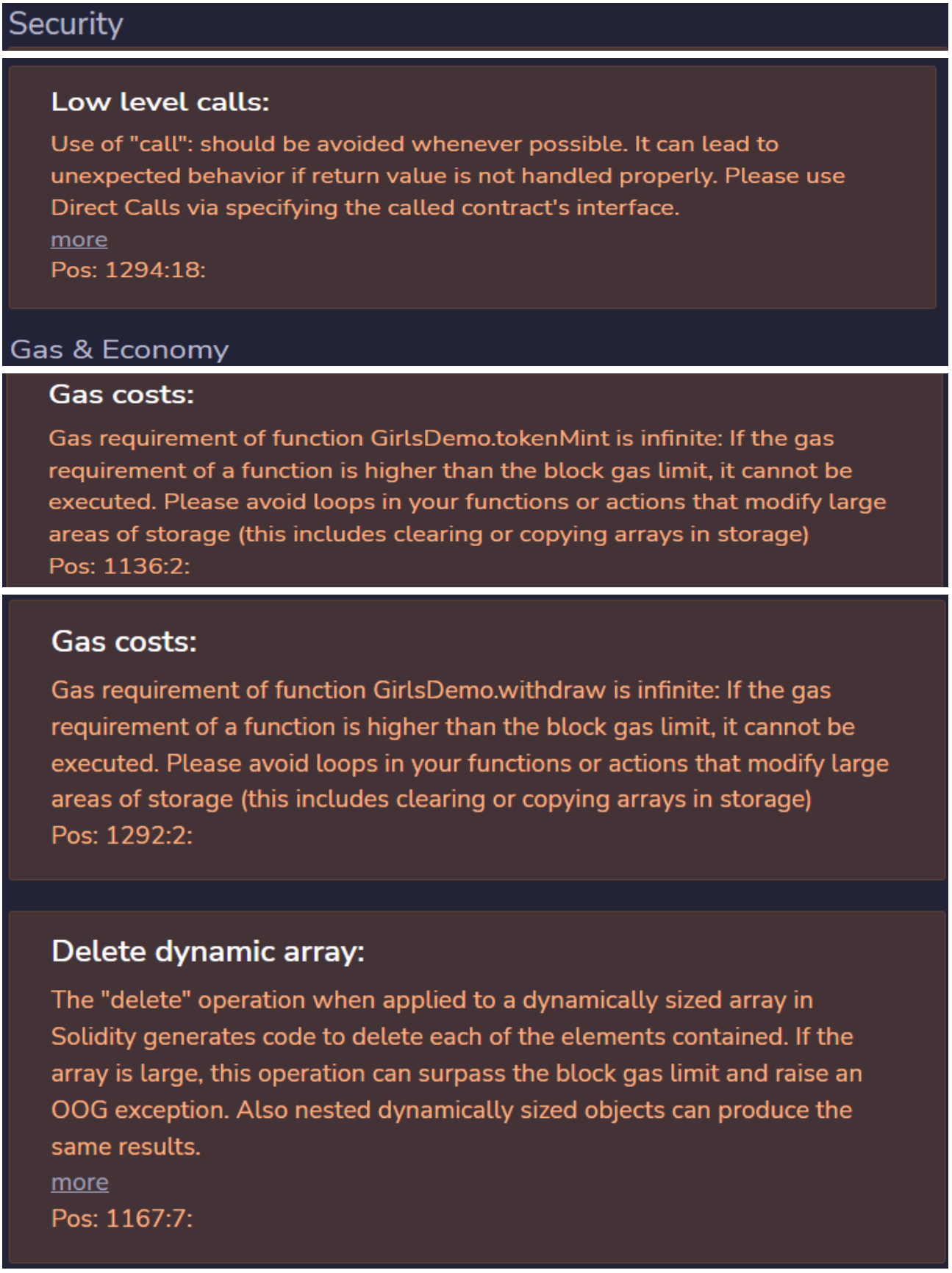
TransparentUpgradeableProxy.sol





The following issues identified by the tool, are checked manually and confirmed that they do not create any negative impact and thus below results are false positives.

GirlsDemo.sol



The following issues identified by the tool, are checked manually and confirmed that they do not create any negative impact and thus below results are false positives.

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: 1181:6:

Miscellaneous

Constant/View/Pure functions:

GirlsDemo.walletOfOwner(address) : Is constant but potentially should not be. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 1204:2:

Similar variable names:

GirlsDemo.walletOfOwner(address) : Variables have very similar names "_owner" and "_owners". Note: Modifiers are currently not considered by this static analysis.
Pos: 1209:40:

Similar variable names:

GirlsDemo.walletOfOwner(address) : Variables have very similar names "_owner" and "_owners". Note: Modifiers are currently not considered by this static analysis.
Pos: 1212:40:

Solhint Linter

Solhint Linter tool allows the code to be scanned by many different attack patterns, and logical vulnerabilities. From the code below, we can say that that tool also did not highlight any major issues.

BitdriveMiddleware.sol

```
BitdriveMiddleware.sol:3:1: Error: Compiler version >=0.6.4 does not
satisfy the r semver requirement
BitdriveMiddleware.sol:300:5: Error: Function name must be in
mixedCase
BitdriveMiddleware.sol:304:5: Error: Function name must be in
mixedCase
BitdriveMiddleware.sol:304:62: Error: Code contains empty blocks
BitdriveMiddleware.sol:505:5: Error: Function name must be in
mixedCase
BitdriveMiddleware.sol:510:5: Error: Function name must be in
mixedCase
BitdriveMiddleware.sol:628:5: Error: Function name must be in
mixedCase
BitdriveMiddleware.sol:630:5: Error: Function name must be in
mixedCase
BitdriveMiddleware.sol:661:5: Error: Function name must be in
mixedCase
BitdriveMiddleware.sol:707:5: Error: Function name must be in
mixedCase
BitdriveMiddleware.sol:1104:5: Error: Explicitly mark visibility of
state
BitdriveMiddleware.sol:1449:13: Error: Possible reentrancy
vulnerabilities. Avoid state changes after transfer.
BitdriveMiddleware.sol:1456:13: Error: Possible reentrancy
vulnerabilities. Avoid state changes after transfer.
BitdriveMiddleware.sol:1460:13: Error: Possible reentrancy
vulnerabilities. Avoid state changes after transfer.
BitdriveMiddleware.sol:1463:13: Error: Possible reentrancy
vulnerabilities. Avoid state changes after transfer.
BitdriveMiddleware.sol:1464:13: Error: Possible reentrancy
vulnerabilities. Avoid state changes after transfer.
BitdriveMiddleware.sol:1465:13: Error: Possible reentrancy
vulnerabilities. Avoid state changes after transfer.
BitdriveMiddleware.sol:1397:9: Error: Variable "_tokenContract" is
unused
```

From the below Solhint logs, we can say that that tool also did not highlight any major issues. And thus below points are false positives and can be safely ignored.

TransparentUpgradeableProxy.sol

```
TransparentUpgradeableProxy.sol:2:1: Error: Compiler version ^0.8.0
does not satisfy the r semver requirement
TransparentUpgradeableProxy.sol:52:9: Error: Avoid using inline
assembly. It is acceptable only in rare cases
TransparentUpgradeableProxy.sol:266:5: Error: Explicitly mark
visibility in function (Set ignoreConstructors to true if using
solidity >=0.7.0)
TransparentUpgradeableProxy.sol:375:49: Error: Code contains empty
blocks
TransparentUpgradeableProxy.sol:640:82: Error: Code contains empty
blocks
TransparentUpgradeableProxy.sol:649:5: Error: Explicitly mark
visibility in function (Set ignoreConstructors to true if using
solidity >=0.7.0)
TransparentUpgradeableProxy.sol:666:5: Error: Explicitly mark
visibility in function (Set ignoreConstructors to true if using
solidity >=0.7.0)
TransparentUpgradeableProxy.sol:756:5: Error: Explicitly mark
visibility in function (Set ignoreConstructors to true if using
solidity >=0.7.0)
TransparentUpgradeableProxy.sol:756:122: Error: Code contains empty
blocks
```

GirlsDemo.sol

```
GirlsDemo.sol:4:1: Error: Compiler version ^0.8.0 does not satisfy
the r semver requirement
GirlsDemo.sol:263:9: Error: Avoid using inline assembly. It is
acceptable only in rare cases
GirlsDemo.sol:416:51: Error: Avoid using low level calls.
GirlsDemo.sol:438:17: Error: Avoid using inline assembly. It is
acceptable only in rare cases
GirlsDemo.sol:518:5: Error: Explicitly mark visibility in function
(Set ignoreConstructors to true if using solidity >=0.7.0)
GirlsDemo.sol:857:21: Error: Avoid using inline assembly. It is
acceptable only in rare cases
GirlsDemo.sol:885:24: Error: Code contains empty blocks
GirlsDemo.sol:1046:5: Error: Explicitly mark visibility in function
(Set ignoreConstructors to true if using solidity >=0.7.0)
GirlsDemo.sol:1109:2: Error: Explicitly mark visibility in function
(Set ignoreConstructors to true if using solidity >=0.7.0)
GirlsDemo.sol:1294:19: Error: Avoid using low level calls.
```

Overall Software analysis result:

These software reported many false positive results and some are informational issues. So, those issues can be safely ignored.

Email: audit@EtherAuthority.io