ArthStableCoin Smart Contract Preliminary Audit Report

Project Synopsis

Project Name	**MahaDao Audit**
Platform	Ethereum, Solidity
Github Repo	https://github.com/MahaDAO/arthcoin-v2/blob/master/contracts/Arth/Arth.sol
Deployed Contract	Not Deployed
Total Duration	15 Days
Timeline of Audit	**15th April 2021 to 02nd April 2021**

Contract Details

Total Contract(s)	1
Name of Contract(s)	ARTHStablecoin
Language	Solidity
Commit Hash	**932a78780b49e20bf5d69b6f7537be3e4ce963b9**

Contract Vulnerabilities Synopsis

Issues	Open Issues	Closed Issues
Critical Severity	0	0
Medium Severity	0	0
Low Severity	2	0
Informational	2	0
Total Found	4	0

Detailed Results

The contract has gone through several stages of the audit procedure that includes structural analysis, automated testing, manual code review, etc.

All the issues have been explained and discussed in detail below. Along with the explanation of the issue found during the audit, the recommended way to overcome the issue or improve the code quality has also been mentioned.

A. Contract Name: ARTHStablecoin

High Severity Issues

None Found

Medium Severity Issues

None Found

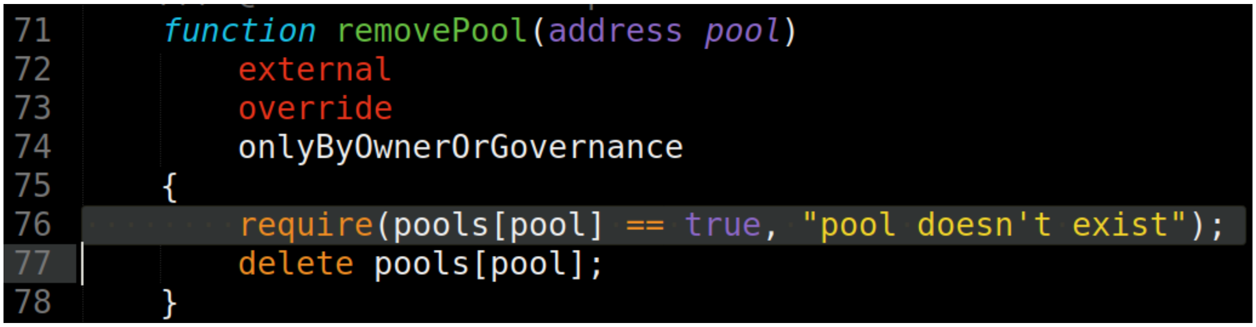
Low Severity Issues

A.1 Comparison to boolean Constant

Line no: 31, 66, 76

Description:

Boolean constants can directly be used in conditional statements or require statements.
Therefore, it's not considered a better practice to explicitly use **TRUE or FALSE** in the **require** statements.



```
71      function removePool(address pool)
72          external
73          override
74          onlyByOwnerOrGovernance
75      {
76          require(pools[pool] == true, "pool doesn't exist");
77          delete pools[pool];
78      }
```

Recommendation:

The equality to boolean constants must be removed from the above-mentioned line.

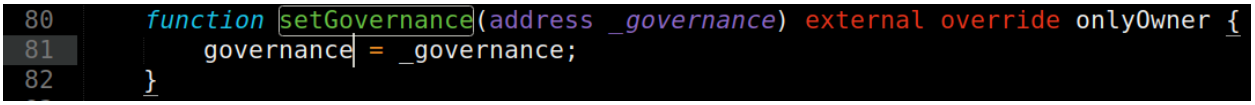
A.2 setGovernance function lacks a Zero Address Check.

Line no - 80

Description:

The **setGovernance** function doesn't validate the ***_governance*** address passed as a parameter.

It is considered a better and secure practice in solidity to ensure that the address arguments passed to a function are not Zero Addresses.



```
80      function setGovernance(address _governance) external override onlyOwner {
81          governance = _governance;
82      }
```

Recommendation:

The governance address argument must be checked with a **require** statement.

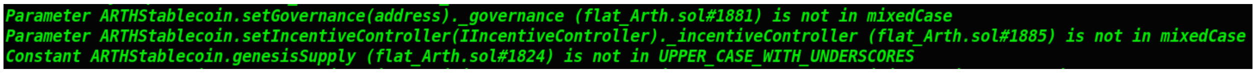
require(_governance != address(0),"Invalid address passed");

Informational

A.3 Coding Style Issues in the Contract

Explanation:
Code readability of a Smart Contract is largely influenced by the Coding Style issues and in some specific scenarios may lead to bugs in the future.

During the automated testing, it was found that the ARTHStablecoin contract had a few code style issues.



```
Parameter ARTHStablecoin.setGovernance(address)._governance (flat_Arth.sol#1881) is not in mixedCase
Parameter ARTHStablecoin.setIncentiveController(IIncentiveController)._incentiveController (flat_Arth.sol#1885) is not in mixedCase
Constant ARTHStablecoin.genesisSupply (flat_Arth.sol#1824) is not in UPPER_CASE_WITH_UNDERSCORES
```

Recommendation:

Therefore, it is highly recommended to fix the issues like naming convention, indentation, and code layout issues in a smart contract.

A.4 NatSpec Annotations must be included

Description:

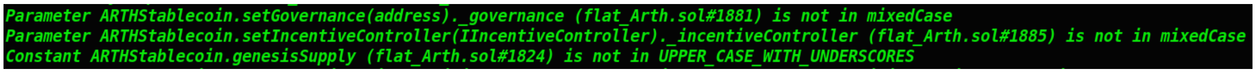
Smart contract does not include the NatSpec annotations adequately.

The Coding style issues in a Smart Contract highly influences its code readability and in some cases may lead to bugs in future.

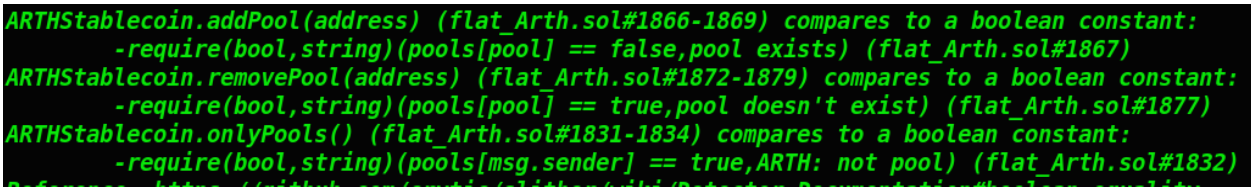
Recommendation:

It's recommended to use the **Solidity Style Guide** to fix all the issues. Moreover, the smart contract should be covered with NatSpec Annotations.

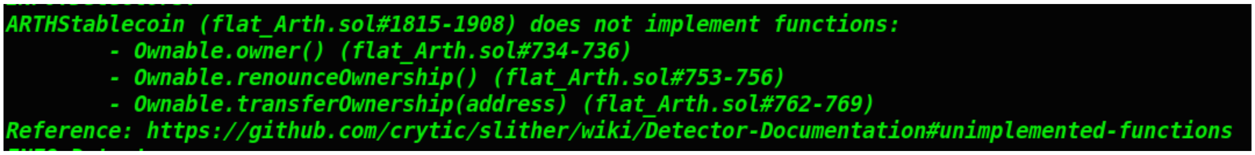
Automated Test Results



```
Parameter ARTHStablecoin.setGovernance(address)._governance (flat_Arth.sol#1881) is not in mixedCase
Parameter ARTHStablecoin.setIncentiveController(IIncentiveController)._incentiveController (flat_Arth.sol#1885) is not in mixedCase
Constant ARTHStablecoin.genesisSupply (flat_Arth.sol#1824) is not in UPPER_CASE_WITH_UNDERSCORES
```



```
ARTHStablecoin.addPool(address) (flat_Arth.sol#1866-1869) compares to a boolean constant:
        -require(bool,string)(pools[pool] == false,pool exists) (flat_Arth.sol#1867)
ARTHStablecoin.removePool(address) (flat_Arth.sol#1872-1879) compares to a boolean constant:
        -require(bool,string)(pools[pool] == true,pool doesn't exist) (flat_Arth.sol#1877)
ARTHStablecoin.onlyPools() (flat_Arth.sol#1831-1834) compares to a boolean constant:
        -require(bool,string)(pools[msg.sender] == true,ARTH: not pool) (flat_Arth.sol#1832)
```



```
ARTHStablecoin (flat_Arth.sol#1815-1908) does not implement functions:
        - Ownable.owner() (flat_Arth.sol#734-736)
        - Ownable.renounceOwnership() (flat_Arth.sol#753-756)
        - Ownable.transferOwnership(address) (flat_Arth.sol#762-769)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions
```