Smart Contract Security Audit Report

Request Date: 24.05.2021
Last Revision: 26.10.2021
Version: 1.0.2

SEDAN - SERVICES AND DATA MANAGEMENT GROUP
SnT - Interdisciplinary Centre for Security, Reliability and Trust
University of Luxembourg

JFK Building - 29, avenue J. F. Kennedy, L-1855 Luxembourg

Lead Auditor

Radu State
<radu.state@uni.lu>

Co-Auditors

Christof Ferreira Torres
<christof.torres@uni.lu>

Wazen Shbair
<wazen.shbair@uni.lu>

Disclaimer

Information security threats are continually changing, with new vulnerabilities discovered on a daily basis, and no application can ever be 100% secure no matter how much security testing is conducted. This report cannot and does not protect against personal or business loss as the result of use of the applications or systems described. SEDAN offers no warranties, representations or legal certifications concerning the applications or systems it tests. All software includes defects: nothing in this document is intended to represent or warrant that security testing was complete and without error, nor does this document represent or warrant that the application tested is suitable to task, free of other defects than reported, fully compliant with any industry standards, or fully compatible with any operating system, hardware, or other application. By using this information you agree that SEDAN be held harmless in any event. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes without SEDAN's prior written consent. Notwithstanding above, this report may be disclosed to governmental or other regulatory authorities and published on the official website of the project. This report was conducted in the context of a joint research project between VNX and the Interdisciplinary Centre for Security, Reliability and Trust (SnT).

Contents

1 Executive Summary

This report summarizes the results of our smart contract security audit that was requested by VNX. This audit was conducted in order to discover issues and vulnerabilities in the source code of VNX's Commodity Token smart contracts. We used a wide range of state-of-the-art security analysis tools and a manual code review in order to assess the security of the smart contracts of VNX. The auditing process paid special attention to the testing of the smart contracts against both common and uncommon attack vectors, the assessment of the codebase for best practice and industry standards, and finally, a thorough line by line manual review of the entire codebase. Overall the audit did not reveal any critical issues. However, we found 1 major issue that should be addressed and 68 minor issues that should be considered.

Executive Summary Update - 09.08.2021

This updated version of the report summarizes the results of our second smart contract security audit that was requested by VNX. This audit was conducted in order to reassess the changes made to the source code of VNX's Commodity Token smart contracts. The reassessment of the updated smart contracts found that from the 67 minor issues reported in the first version of the audit report; 38 issues have been resolved, 10 issues have been safely ignored and 19 issues with a recommendation to be fixed, have not been fixed yet. Moreover, the major issue regarding the code size of the DGR contract still has not been fixed.

Executive Summary Update - 26.10.2021

This report has been updated according to the suggestions provided by VNX's legal department. Moreover, the report also summarizes the findings of our third round of the smart contract security audit that was requested by VNX. This third audit round was conducted in order to reassess the changes made to the source code of VNX's Commodity Token smart contracts. The reassessment of the updated smart contracts found that from the 19 minor issues reported in the second version of the audit report, 1 issue has been resolved. Moreover, the major issue regarding the code size of the DGR contract has been fixed by using shorter error messages and enabling optimization of 200 runs within the Solidity compiler.

2 Version History

Version	Date	Comments
1.0.0	04.06.2021	First version of the audit report
1.0.1	09.08.2021	Audited fixes of bugs reported in version 1.0.0
1.0.2	26.10.2021	Audited fixes of bugs reported in version 1.0.1; Updated VNX logo; Updated disclaimer; Renamed ”VNX Exchange” to ”VNX” and ”Digital Gold Receipt” to ”Commodity Token”;

3 Audit Scope

The scope of this audit is limited to the smart contracts that are related to the Commodity Token project of VNX.

3.1 Files Covered

The audit covers all the files that are listed below. These are all the files that are contained inside the "`contracts`" folder of the vnx-exchange/ethereum-contracts repository under branch code-reviews/UniLu. The file "`Migrations.sol`" and the files that are within the folder "`mocks`" and "`old stuff`" are out of scope.

Filename	**SHA-1 Hash**
AdminTransferProvider.sol	`f662047c0c65466aa2931d1fb2c83530ca608d35`
AnyTransferProvider.sol	`7d7201bfd684f81bc59edef1a22495860c23d174`
DGR.sol	`ff40a4cc5f3c29c84513a00e3c41b9f91bdd5ef6`
IRBAC.sol	`f45f8814bd084699c5cec919e80c172d0e70fe1a`
ITransferProvider.sol	`798c96359728c2beb5aa185cfed30a24e39e8571`
Manager.sol	`fad6e0ec44e6a8dbf0de8b83c8fb42afc9f95d71`
ProxyAdmin.sol	`adea8f518c0d00f2b00b7a86aba61e497379c1f2`
WhitelistTransferProvider.sol	`b9cf08055aca36ce8727e8ace423ee12d5c5f2da`

Files covered in version 1.0.0 (commit fbd4f59)

Filename	**SHA-1 Hash**
AdminTransferProvider.sol	`7092b72dbedeb6dd0bdd8311290c6eeb74a26349`
AnyTransferProvider.sol	`c16fbf2449b8febc30c0487ea31808f7cee55413`
DGR.sol	`8cfb26208dec9d4f14ee3a3656f4bc1a476c8fdd`
IRBAC.sol	`f45f8814bd084699c5cec919e80c172d0e70fe1a`
ITransferProvider.sol	`3e01f888b06b3a63c078f2537fb168c2a6260adb`
Manager.sol	`02971f4576bd2dffbf8aaad1bb141e619b446b39`
ProxyAdmin.sol	`10e6111f97d17d4a9e110e2f4d58ca0303ecfefd`
WhitelistTransferProvider.sol	`188c3889cd1dcadbc3c3b6e99c6b7c3b8e8714c3`

Files covered in version 1.0.1 (commit c388c47)

Filename	SHA-1 Hash
AdminTransferProvider.sol	7092b72dbedeb6dd0bdd8311290c6eeb74a26349
AnyTransferProvider.sol	c16fbf2449b8febc30c0487ea31808f7cee55413
DGR.sol	5efb1f1f8f233f7325830aaa57a6c02c62bb22a3
IRBAC.sol	f45f8814bd084699c5cec919e80c172d0e70fe1a
ITransferProvider.sol	3e01f888b06b3a63c078f2537fb168c2a6260adb
Manager.sol	02971f4576bd2dffbf8aaad1bb141e619b446b39
ProxyAdmin.sol	10e6111f97d17d4a9e110e2f4d58ca0303ecfefd
WhitelistTransferProvider.sol	6232ad0e97789920359fb0fbf47f377ac212a6c5

Files covered in version 1.0.2 (commit dc51d2a)

3.2 Audit Activities

The audit activities can be grouped into the following three broad categories:

1. **Security.** Identifying security related issues within the contract.
2. **Architecture.** Evaluating the system architecture through the lens of established smart contract best practices.
3. **Code Quality.** A full review of the contract source code. The primary areas of focus include: correctness, readability, and code complexity.

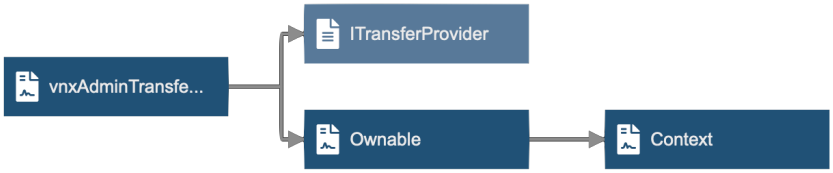
4 System Overview

The system is composed of six fully implemented smart contracts and two interface contracts. The smart contracts were developed using the Solidity programming language and Openzeppelin - a library for secure smart contract development.

4.1 AdminTransferProvider

The AdminTransferProvider contract uses the SafeMath and Ownable contracts from the standard Openzeppelin library, and implements two interfaces IRBAC and ITransferProvider.

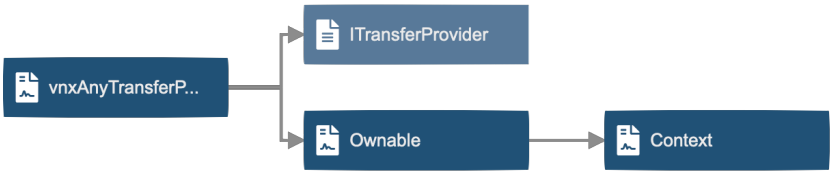
Note: the contract has no mathematical operations.



4.2 AnyTransferProvider

The AnyTransferProvider contract uses the SafeMath and Ownable contracts from the standard Openzeppelin library, and implements the ITransferProvider interface.

Note: the contract has no mathematical operations.



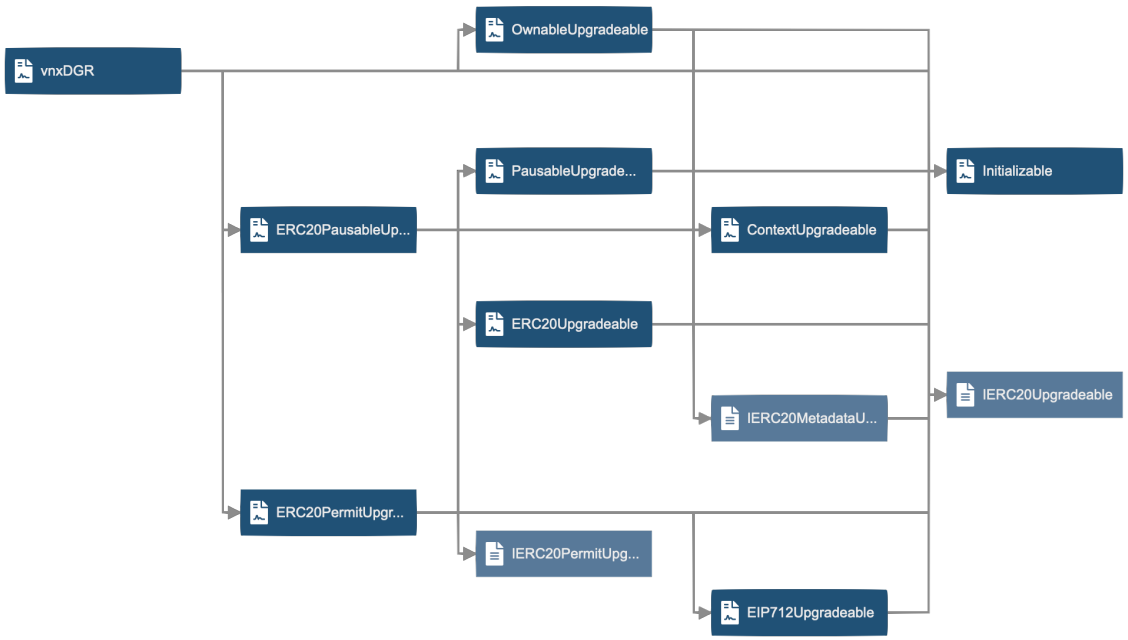
4.3 DGR

The DGR contract implements an upgradeable contract, with upgradeable SafeMath and Ownable contracts. It also provides an upgradeable ERC20

token implementation. The interface contracts ITransferProvider and IRBAC are implemented as well.

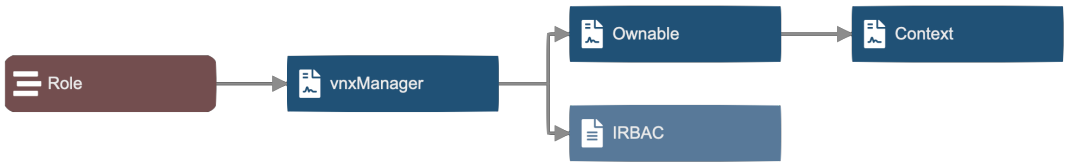
Note: the decreaseSupply function does not make use of the SafeMath subtraction function to compute the parameter for the _approve function.

```
function decreaseSupply(address _wallet, uint256 _value) public
    onlySupplyControllerRole returns (bool success) {
      require(_wallet!=address(0),"Wallet address cannot be zero");
      require(_value > 0, "Value must be positive non-zero");

      if (_wallet != _msgSender()) {
        uint256 currentAllowance=allowance(_wallet, _msgSender());
        require(currentAllowance>=_value,"ERC20: burn amount
            exceeds allowance");
        _approve(_wallet,_msgSender(),currentAllowance -_value);
            <<----
      }
      _burn(_wallet, _value);

      emit SupplyDecreased(_msgSender(), _value);
      return true;
  }
```



4.4 Manager

The Manager contract implements run-time configurable Role Based Access Control and Contract Management. It uses the Ownable contract and the IRBAC interface contract.



4.5 ProxyAdmin

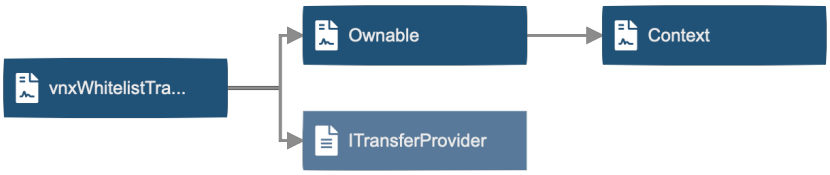
The ProxyAdmin contract overrides a number of functions of the ProxyAdmin contract that is provided by the Openzeppelin library. It also inherits functions from the TransparentUpgradeableProxy contract, and implements the IRBAC interface.

4.6 WhitelistTransferProvider

The WhitelistTransferProvider contract uses the SafeMath and Ownable contracts from the standard Openzeppelin library, and implements the ITransferProvider interface.

Note: the _addOwner function does not make use of the SafeMath subtraction function to calculate the return value.

```
function _addOwner(address _newOwner) internal returns(uint256) {
   for (uint256 i=0; i< owners.length; i++) {
     if (owners[i] == _newOwner) {
         return i;
     }
   }
   owners.push(_newOwner);
   return owners.length - 1; <<----
 }
```


Ownable
Context
vnxWhitelistTra...
ITransferProvider

5 Methodology

Our audit is composed of an automated security assessment using a collection of state-of-the-art smart contract analysis tools and a manual code review.

5.1 Tools

The following table lists the security analysis tools that were used during our audit:

Toolname	Version	Description
Solc	0.8.4	Solc is the standard Solidity compiler maintained by the Ethereum Foundation. The compiler reports compilation warnings and errors.
Mythril	0.22.19	Mythril is a mature symbolic execution tool maintained by ConSensys Dilligence for the purpose of detecting smart contract vulnerabilities.
Slither	0.8.0	Slither is a static analysis tool to find vulnerabilities within smart contracts that have been written in Solidity.

5.2 Severity

We mark each finding with one of the following three severity labels:

Severity	Description
Minor	Minor issues are subjective in nature. They are typically suggestions around best practices or readability. Code maintainers should use their own judgment as to whether to address such issues.
Major	Major issues are security vulnerabilities that may not be directly exploitable or may require certain conditions in order to be exploited. All major issues should be addressed.
Critical	Critical issues are directly exploitable security vulnerabilities that need to be fixed.

5.3 Status

We mark each finding with one of the following three status labels:

Status	Description
✓	This means that the issue has been fixed.
✗	This means that the issue has not been fixed.
-	This means that our recommendation is to ignore the issue.

6 Findings

The following table summarizes all our findings:

Filename	Finding	Severity	Status
AdminTransferProvider.sol:15:3	Visibility for constructor is ignored	Minor	✓
AdminTransferProvider.sol:37:29	Unused function parameter	Minor	-
AdminTransferProvider.sol:10-52	Conformance to Solidity naming conventions	Minor	✓
AdminTransferProvider.sol:23	Conformance to Solidity naming conventions	Minor	✓
AdminTransferProvider.sol:37	Conformance to Solidity naming conventions	Minor	✓
AnyTransferProvider.sol:15:72	Unused function parameter	Minor	-
AnyTransferProvider.sol:24:29	Unused function parameter	Minor	-
AnyTransferProvider.sol:24:44	Unused function parameter	Minor	-
AnyTransferProvider.sol:24:57	Unused function parameter	Minor	-
AnyTransferProvider.sol:9-28	Conformance to Solidity naming conventions	Minor	✓
AnyTransferProvider.sol:15	Conformance to Solidity naming conventions	Minor	✓
DGR.sol:14:1	Contract code size exceeds 24576	Major	✓
DGR.sol:14-380	Conformance to Solidity naming conventions	Minor	✓
DGR.sol:91	Conformance to Solidity naming conventions	Minor	✗
DGR.sol:106	Conformance to Solidity naming conventions	Minor	✗
DGR.sol:124	Conformance to Solidity naming conventions	Minor	✗
DGR.sol:135	Conformance to Solidity naming conventions	Minor	✗
DGR.sol:147	Conformance to Solidity naming conventions	Minor	✗
DGR.sol:166	Conformance to Solidity naming conventions	Minor	✗
DGR.sol:177	Conformance to Solidity naming conventions	Minor	✗
DGR.sol:188	Conformance to Solidity naming conventions	Minor	✗
DGR.sol:198	Conformance to Solidity naming conventions	Minor	✗
DGR.sol:208	Conformance to Solidity naming conventions	Minor	✗
DGR.sol:229	Conformance to Solidity naming conventions	Minor	✗
DGR.sol:245	Conformance to Solidity naming conventions	Minor	✗
DGR.sol:266	Conformance to Solidity naming conventions	Minor	✗
DGR.sol:284-285	Conformance to Solidity naming conventions	Minor	✗
DGR.sol:314	Conformance to Solidity naming conventions	Minor	✗
DGR.sol:325	Conformance to Solidity naming conventions	Minor	✗
DGR.sol:336	Conformance to Solidity naming conventions	Minor	✗
DGR.sol:31	Conformance to Solidity naming conventions	Minor	✗
DGR.sol:135-139	Public function that could be declared external	Minor	✓
DGR.sol:147-152	Public function that could be declared external	Minor	✓
DGR.sol:166-171	Public function that could be declared external	Minor	✓
DGR.sol:177-182	Public function that could be declared external	Minor	✓
DGR.sol:188-191	Public function that could be declared external	Minor	✓
DGR.sol:198-200	Public function that could be declared external	Minor	✓

DGR.sol:208-213	Public function that could be declared external	Minor	✓
DGR.sol:229-236	Public function that could be declared external	Minor	✓
DGR.sol:245-258	Public function that could be declared external	Minor	✓
DGR.sol:284-300	Public function that could be declared external	Minor	✓
DGR.sol:314-319	Public function that could be declared external	Minor	✓
DGR.sol:325-330	Public function that could be declared external	Minor	✓
DGR.sol:364-367	Public function that could be declared external	Minor	✓
DGR.sol:372-375	Public function that could be declared external	Minor	✓
Manager.sol:64:3	Visibility for constructor is ignored	Minor	✓
Manager.sol:75:13	Unnamed return variable can remain unassigned	Minor	✓
Manager.sol:12-176	Conformance to Solidity naming conventions	Minor	✓
Manager.sol:73	Conformance to Solidity naming conventions	Minor	✓
Manager.sol:89	Conformance to Solidity naming conventions	Minor	✓
Manager.sol:117	Conformance to Solidity naming conventions	Minor	✓
Manager.sol:129	Conformance to Solidity naming conventions	Minor	✓
Manager.sol:153	Conformance to Solidity naming conventions	Minor	✓
Manager.sol:40	Unused state variable	Minor	✓
ProxyAdmin.sol:17:5	Visibility for constructor is ignored	Minor	✓
ProxyAdmin.sol:13-73	Conformance to Solidity naming conventions	Minor	✓
ProxyAdmin.sol:47-49	Public function that could be declared external	Minor	-
ProxyAdmin.sol:58-60	Public function that could be declared external	Minor	-
ProxyAdmin.sol:70-72	Public function that could be declared external	Minor	-
WhitelistTransferProvider.sol:14:3	Visibility for constructor is ignored	Minor	✓
WhitelistTransferProvider.sol:45:72	Unused function parameter	Minor	-
WhitelistTransferProvider.sol:71:29	Unused function parameter	Minor	-
WhitelistTransferProvider.sol:9-90	Conformance to Solidity naming conventions	Minor	✓
WhitelistTransferProvider.sol:22	Conformance to Solidity naming conventions	Minor	✓
WhitelistTransferProvider.sol:29	Conformance to Solidity naming conventions	Minor	✓
WhitelistTransferProvider.sol:35	Conformance to Solidity naming conventions	Minor	✓
WhitelistTransferProvider.sol:45	Conformance to Solidity naming conventions	Minor	✓
WhitelistTransferProvider.sol:71	Conformance to Solidity naming conventions	Minor	✓

6.1 AdminTransferProvider

6.1.1 Solidity Compiler

Visibility for constructor is ignored	
AdminTransferProvider.sol:15:3	Minor

Description

If you want the contract to be non-deployable, making it ”abstract” is sufficient.

```
constructor(address _rbac) public
```

Recommendation

Remove the visibility keyword public from the constructor.

Unused function parameter	
AdminTransferProvider.sol:37:29	Minor

Description

Remove or comment out the variable name `_from` to silence this warning.

```
function considerTransfer(address _from, address _to,
    uint256 _value) external pure override returns(bool)
```

Recommendation

Ignore warning in this case since the variable name has to be included in order to be able to override the original function implementation.

6.1.2 Mythril

No issues were detected.

6.1.3 Slither

Conformance to Solidity naming conventions	
`AdminTransferProvider.sol:10-52`	Minor

Description

Contract `vnxAdminTransferProvider` is not in CapWords.

Recommendation

Try to follow the Solidity naming conventions.

Conformance to Solidity naming conventions	
`AdminTransferProvider.sol:23`	Minor

Description

Parameter `_from` in `approveTransfer` is not in mixedCase.

Parameter `_to` in `approveTransfer` is not in mixedCase.

Parameter `_value` in `approveTransfer` is not in mixedCase.

Parameter `_spender` in `approveTransfer` is not in mixedCase.

Recommendation

Try to follow the Solidity naming conventions.

Conformance to Solidity naming conventions	
`AdminTransferProvider.sol:37`	Minor

Description

Parameter `_to` in `considerTransfer` is not in mixedCase.

Parameter `_value` in `considerTransfer` is not in mixedCase.

Recommendation

Try to follow the Solidity naming conventions.

6.2 AnyTransferProvider

6.2.1 Solidity Compiler

Unused function parameter	
AnyTransferProvider.sol:15:72	Minor

Description

Remove or comment out the variable name `_spender` to silence this warning.

```
function approveTransfer(address _from, address _to, uint256
    _value, address _spender) external override returns(bool
    )
```

Recommendation

Ignore warning in this case since the variable name has to be included in order to be able to override the original function implementation.

Unused function parameter	
AnyTransferProvider.sol:24:29	Minor

Description

Remove or comment out the variable name `_from` to silence this warning.

```
function considerTransfer(address _from, address _to,
    uint256 _value) external pure override returns(bool)
```

Recommendation

Ignore warning in this case since the variable name has to be included in order to be able to override the original function implementation.

Unused function parameter	
AnyTransferProvider.sol:24:44	Minor

Description

Remove or comment out the variable name `_to` to silence this warning.

```
function considerTransfer(address _from, address _to,
    uint256 _value) external pure override returns(bool)
```

Recommendation

Ignore warning in this case since the variable name has to be included in order to be able to override the original function implementation.

Unused function parameter	
AnyTransferProvider.sol:24:57	Minor

Description

Remove or comment out the variable name `_value` to silence this warning.

```
function considerTransfer(address _from, address _to,
    uint256 _value) external pure override returns(bool)
```

Recommendation

Ignore warning in this case since the variable name has to be included in order to be able to override the original function implementation.

6.2.2 Mythril

No issues were detected.

6.2.3 Slither

Conformance to Solidity naming conventions	
AnyTransferProvider.sol:9-28	Minor

Description

Contract `vnxAnyTransferProvider` is not in CapWords.

Recommendation

Try to follow the Solidity naming conventions.

Conformance to Solidity naming conventions	
AnyTransferProvider.sol:15	Minor

Description

Parameter `_from` in `approveTransfer` is not in mixedCase.

Parameter `_to` in `approveTransfer` is not in mixedCase.

Parameter `_value` in `approveTransfer` is not in mixedCase.

Recommendation

Try to follow the Solidity naming conventions.

6.3 DGR

6.3.1 Solidity Compiler

Contract code size exceeds 24576 bytes	
DGR.sol:14:1	Major

Description

This contract may not be deployable on mainnet.

```
contract vnxDGR is Initializable, OwnableUpgradeable,
    ERC20PausableUpgradeable, ERC20PermitUpgradeable {
```

Recommendation

Check if contract is deployable by deploying the contract on a local geth node with the latest hard fork. In case the contract is not deployable, consider enabling the optimizer (with a low "runs" value!), turning off revert strings, or using libraries.

6.3.2 Mythril

No issues were detected.

6.3.3 Slither

Conformance to Solidity naming conventions	
DGR.sol:14-380	Minor

Description

Contract `vnxDGR` is not in CapWords.

Recommendation

Try to follow the Solidity naming conventions.

Conformance to Solidity naming conventions	
`DGR.sol:91`	Minor

Description

Parameter `_to` in `transfer` is not in mixedCase.

Parameter `_value` in `transfer` is not in mixedCase.

Recommendation

Try to follow the Solidity naming conventions.

Conformance to Solidity naming conventions	
`DGR.sol:106`	Minor

Description

Parameter `_from` in `transferFrom` is not in mixedCase.

Parameter `_to` in `transferFrom` is not in mixedCase.

Parameter `_value` in `transferFrom` is not in mixedCase.

Recommendation

Try to follow the Solidity naming conventions.

Conformance to Solidity naming conventions	
`DGR.sol:124`	Minor

Description

Parameter `_spender` in `approve` is not in mixedCase.

Parameter `_value` in `approve` is not in mixedCase.

Recommendation

Try to follow the Solidity naming conventions.

Conformance to Solidity naming conventions	
DGR.sol:135	Minor

Description

Parameter `_token` in `transferAnyERC20Token` is not in mixedCase.

Parameter `_to` in `transferAnyERC20Token` is not in mixedCase.

Parameter `_amount` in `transferAnyERC20Token` is not in mixedCase.

Recommendation

Try to follow the Solidity naming conventions.

Conformance to Solidity naming conventions	
DGR.sol:147	Minor

Description

Parameter `_newAssetProtectionRole` in `setAssetProtectionRole` is not in mixedCase.

Recommendation

Try to follow the Solidity naming conventions.

Conformance to Solidity naming conventions	
DGR.sol:166	Minor

Description

Parameter `_addr` in `freeze` is not in mixedCase.

Recommendation

Try to follow the Solidity naming conventions.

Conformance to Solidity naming conventions	
DGR.sol:177	Minor

Description

Parameter `_addr` in `unfreeze` is not in mixedCase.

Recommendation

Try to follow the Solidity naming conventions.

Conformance to Solidity naming conventions	
DGR.sol:188	Minor

Description

Parameter `_addr` in `reclaimTokensFromFrozenAddress` is not in mixedCase.

Recommendation

Try to follow the Solidity naming conventions.

Conformance to Solidity naming conventions	
DGR.sol:198	Minor

Description

Parameter `_addr` in `isFrozen` is not in mixedCase.

Recommendation

Try to follow the Solidity naming conventions.

Conformance to Solidity naming conventions	
DGR.sol:208	Minor

Description

Parameter `_newSupplyControllerRole` in `setSupplyControllerRole` is not in mixedCase.

Recommendation

Try to follow the Solidity naming conventions.

Conformance to Solidity naming conventions	
DGR.sol:229	Minor

Description

Parameter `_wallet` in `increaseSupply` is not in mixedCase.

Parameter `_value` in `increaseSupply` is not in mixedCase.

Recommendation

Try to follow the Solidity naming conventions.

Conformance to Solidity naming conventions	
DGR.sol:245	Minor

Description

Parameter `_wallet` in `decreaseSupply` is not in mixedCase.

Parameter `_value` in `decreaseSupply` is not in mixedCase.

Recommendation

Try to follow the Solidity naming conventions.

Conformance to Solidity naming conventions	
DGR.sol:266	Minor

Description

Parameter `_newProvider` in `changeTransferProvider` is not in mixedCase.

Recommendation

Try to follow the Solidity naming conventions.

Conformance to Solidity naming conventions	
DGR.sol:284-285	Minor

Description

Parameter `_initTransferProvider` in `initialize` is not in mixedCase.

Parameter `_rbac` in `initialize` is not in mixedCase.

Parameter `_supplyControllerRole` in `initialize` is not in mixedCase.

Parameter `_assetProtectionRole` in `initialize` is not in mixedCase.

Parameter `_owner` in `initialize` is not in mixedCase.

Parameter `_name` in `initialize` is not in mixedCase.

Parameter `_symbol` in `initialize` is not in mixedCase.

Recommendation

Try to follow the Solidity naming conventions.

Conformance to Solidity naming conventions	
DGR.sol:314	Minor

Description

Parameter `_newFeeRecipient` in `setFeeRecipient` is not in mixedCase.

Recommendation

Try to follow the Solidity naming conventions.

Conformance to Solidity naming conventions	
DGR.sol:325	Minor

Description

Parameter `_newFeeRate` in `setFeeRate` is not in mixedCase.

Recommendation

Try to follow the Solidity naming conventions.

Conformance to Solidity naming conventions	
DGR.sol:336	Minor

Description

Parameter `_value` in `getFeeFor` is not in mixedCase.

Recommendation

Try to follow the Solidity naming conventions.

Conformance to Solidity naming conventions	
DGR.sol:31	Minor

Description

Constant `feeDecimals` is not in UPPER_CASE_WITH_UNDERSCORES.

Recommendation

Try to follow the Solidity naming conventions.

Public function that could be declared external	
DGR.sol:135-139	Minor
Description	
Function `transferAnyERC20Token` should be declared external.	
Recommendation	
Change function visibility from public to external.	

Public function that could be declared external	
DGR.sol:147-152	Minor
Description	
Function `setAssetProtectionRole` should be declared external.	
Recommendation	
Change function visibility from public to external.	

Public function that could be declared external	
DGR.sol:166-171	Minor
Description	
Function `freeze` should be declared external.	
Recommendation	
Change function visibility from public to external.	

Public function that could be declared external	
DGR.sol:177-182	Minor
Description	
Function `unfreeze` should be declared external.	
Recommendation	
Change function visibility from public to external.	

Public function that could be declared external	
DGR.sol:188-191	Minor
Description	
Function `reclaimTokensFromFrozenAddress` should be declared external.	
Recommendation	
Change function visibility from public to external.	

Public function that could be declared external	
DGR.sol:198-200	Minor
Description	
Function `isFrozen` should be declared external.	
Recommendation	
Change function visibility from public to external.	

Public function that could be declared external	
DGR.sol:208-213	Minor
Description	
Function `setSupplyControllerRole` should be declared external.	
Recommendation	
Change function visibility from public to external.	

Public function that could be declared external	
DGR.sol:229-236	Minor
Description	
Function `increaseSupply` should be declared external.	
Recommendation	
Change function visibility from public to external.	

Public function that could be declared external	
`DGR.sol:245-258`	Minor
Description	
Function `decreaseSupply` should be declared external.	
Recommendation	
Change function visibility from public to external.	

Public function that could be declared external	
`DGR.sol:284-300`	Minor
Description	
Function `initialize` should be declared external.	
Recommendation	
Change function visibility from public to external.	

Public function that could be declared external	
`DGR.sol:314-319`	Minor
Description	
Function `setFeeRecipient` should be declared external.	
Recommendation	
Change function visibility from public to external.	

Public function that could be declared external	
`DGR.sol:325-330`	Minor
Description	
Function `setFeeRate` should be declared external.	
Recommendation	
Change function visibility from public to external.	

Public function that could be declared external	
`DGR.sol:364-367`	Minor
Description Function `pause` should be declared external.	
Recommendation Change function visibility from public to external.	

Public function that could be declared external	
`DGR.sol:372-375`	Minor
Description Function `unpause` should be declared external.	
Recommendation Change function visibility from public to external.	

6.4 Manager

6.4.1 Solidity Compiler

Visibility for constructor is ignored	
`Manager.sol:64:3`	Minor

Description

If you want the contract to be non-deployable, making it ”abstract” is sufficient.

```
constructor() public {
```

Recommendation

Remove the visibility keyword public from the constructor.

Unnamed return variable can remain unassigned	
`Manager.sol:75:13`	Minor

Description

Add an explicit return with value to all non-reverting code paths or name the variable.

```
returns(uint256)
```

Recommendation

Since we are overriding a function, we recommend adding a return value.

6.4.2 Mythril

No issues were detected.

6.4.3 Slither

Conformance to Solidity naming conventions	
`Manager.sol:12-176`	Minor

Description

Contract `vnxManager` is not in CapWords.

Recommendation

Try to follow the Solidity naming conventions.

Conformance to Solidity naming conventions	
`Manager.sol:73`	Minor

Description

Parameter `_roleDescription` in `addRootRole` is not in mixedCase.

Recommendation

Try to follow the Solidity naming conventions.

Conformance to Solidity naming conventions	
`Manager.sol:89`	Minor

Description

Parameter `_roleDescription` in `addRole` is not in mixedCase.

Parameter `_admin` in `addRole` is not in mixedCase.

Recommendation

Try to follow the Solidity naming conventions.

Conformance to Solidity naming conventions	
`Manager.sol:117`	Minor

Description

Parameter `_account` in `hasRole` is not in mixedCase.

Parameter `_role` in `hasRole` is not in mixedCase.

Recommendation

Try to follow the Solidity naming conventions.

Conformance to Solidity naming conventions	
`Manager.sol:129`	Minor

Description

Parameter `_account` in `addBearer` is not in mixedCase.

Parameter `_role` in `addBearer` is not in mixedCase.

Recommendation

Try to follow the Solidity naming conventions.

Conformance to Solidity naming conventions	
`Manager.sol:153`	Minor
Description	
Parameter `_account` in `removeBearer` is not in mixedCase. Parameter `_role` in `removeBearer` is not in mixedCase.	
Recommendation	
Try to follow the Solidity naming conventions.	

Unused state variable	
`Manager.sol:40`	Minor
Description	
State variable `SUPERADMIN_ROLE` is never used in contract `vnxManager`.	
Recommendation	
Remove unused state variable from the contract.	

6.5 ProxyAdmin

6.5.1 Solidity Compiler

Visibility for constructor is ignored	
`ProxyAdmin.sol:17:5`	Minor
Description If you want the contract to be non-deployable, making it ”abstract” is sufficient. `constructor(uint8 _role, address _rbac) public`	
Recommendation Remove the visibility keyword public from the constructor.	

6.5.2 Mythril

No issues were detected.

6.5.3 Slither

Conformance to Solidity naming conventions	
`ProxyAdmin.sol:13-73`	Minor
Description Contract `vnxProxyAdmin` is not in CapWords.	
Recommendation Try to follow the Solidity naming conventions.	

Public function that could be declared external	
`ProxyAdmin.sol:47-49`	Minor
Description Function `changeProxyAdmin` should be declared external.	
Recommendation Ignore warning in this case since the visibility keyword has to remain public in order to be able to override the original function implementation.	

Public function that could be declared external	
`ProxyAdmin.sol:58-60`	Minor

Description

Function `upgrade` should be declared external.

Recommendation

Ignore warning in this case since the visibility keyword has to remain public in order to be able to override the original function implementation.

Public function that could be declared external	
`ProxyAdmin.sol:70-72`	Minor

Description

Function `upgradeAndCall` should be declared external.

Recommendation

Ignore warning in this case since the visibility keyword has to remain public in order to be able to override the original function implementation.

6.6 WhitelistTransferProvider

6.6.1 Solidity Compiler

Visibility for constructor is ignored	
`WhitelistTransferProvider.sol:14:3`	Minor

Description

If you want the contract to be non-deployable, making it ”abstract” is sufficient.

```
constructor() public {
```

Recommendation

Remove the visibility keyword public from the constructor.

Unused function parameter	
`WhitelistTransferProvider.sol:45:72`	Minor

Description

Remove or comment out the variable name `_spender` to silence this warning.

```
function approveTransfer(address _from, address _to, uint256
    _value, address _spender) external override returns(bool
    )
```

Recommendation

Ignore warning in this case since the variable name has to be included in order to be able to override the original function implementation.

Unused function parameter	
`WhitelistTransferProvider.sol:71:29`	Minor

Description

Remove or comment out the variable name `_from` to silence this warning.

```
function considerTransfer(address _from, address _to,
    uint256 _value) external override returns(bool)
```

Recommendation

Ignore warning in this case since the variable name has to be included in order to be able to override the original function implementation.

6.6.2 Mythril

No issues were detected.

6.6.3 Slither

Conformance to Solidity naming conventions	
`WhitelistTransferProvider.sol:9-90`	Minor
Description	
Contract `vnxWhitelistTransferProvider` is not in CapWords.	
Recommendation	
Try to follow the Solidity naming conventions.	

Conformance to Solidity naming conventions	
`WhitelistTransferProvider.sol:22`	Minor
Description	
Parameter `_ind` in `getOwnersItem` is not in mixedCase.	
Recommendation	
Try to follow the Solidity naming conventions.	

Conformance to Solidity naming conventions	
`WhitelistTransferProvider.sol:29`	Minor
Description	
Parameter `_a` in `addOwner` is not in mixedCase.	
Recommendation	
Try to follow the Solidity naming conventions.	

Conformance to Solidity naming conventions	
`WhitelistTransferProvider.sol:35`	Minor

Description

Parameter `_ind` in `removeOwner` is not in mixedCase.

Recommendation

Try to follow the Solidity naming conventions.

Conformance to Solidity naming conventions	
`WhitelistTransferProvider.sol:45`	Minor

Description

Parameter `_from` in `approveTransfer` is not in mixedCase.

Parameter `_to` in `approveTransfer` is not in mixedCase.

Parameter `_value` in `approveTransfer` is not in mixedCase.

Recommendation

Try to follow the Solidity naming conventions.

Conformance to Solidity naming conventions	
`WhitelistTransferProvider.sol:71`	Minor

Description

Parameter `_to` in `considerTransfer` is not in mixedCase.

Parameter `_value` in `considerTransfer` is not in mixedCase.

Recommendation

Try to follow the Solidity naming conventions.

wwwen.uni.lu/snt/research/sedan

JFK Building - 29, avenue J. F. Kennedy, L-1855 Luxembourg