



**Amun**

**Rebalance Manager**

**SMART CONTRACT AUDIT**

**21.07.2021**

**Made in Germany by Chainsulting.de**



## Table of contents

1. Disclaimer .....	3
2. About the Project and Company .....	4
2.1 Project Overview .....	5
3. Vulnerability & Risk Level.....	6
4. Auditing Strategy and Techniques Applied .....	7
4.1 Methodology.....	7
4.2 Used Code from other Frameworks/Smart Contracts .....	8
4.3 Tested Contract Files.....	8
4.4 Metrics / CallGraph .....	9
4.5 Metrics / Source Lines & Risk .....	10
4.6 Metrics / Capabilities.....	11
4.7 Metrics / Source Unites in Scope.....	12
5. Scope of Work.....	13
5.1 Manual and Automated Vulnerability Test .....	14
5.1.1 No package.json .....	14
5.2. SWC Attacks .....	16
5.3. Associated audits with the forked codebase.....	20
5.4. Verify Claims .....	20
6. Executive Summary .....	24
7. Deployed Smart Contract.....	24



## 1. Disclaimer

The audit makes no statements or warranties about utility of the code, safety of the code, suitability of the business model, investment advice, endorsement of the platform or its products, regulatory regime for the business model, or any other statements about fitness of the contracts to purpose, or their bug free status. The audit documentation is for discussion purposes only.

The information presented in this report is confidential and privileged. If you are reading this report, you agree to keep it confidential, not to copy, disclose or disseminate without the agreement of Amun Limited. If you are not the intended receptor of this document, remember that any disclosure, copying or dissemination of it is forbidden.

Major Versions / Date	Description
0.1 (18.07.2021)	Layout
0.2 (19.07.2021)	Test Deployment
0.5 (20.07.2021)	Automated Security Testing Manual Security Testing
0.6 (20.07.2021)	Testing SWC Checks
0.7 (21.07.2021)	Verify Claims
0.9 (21.07.2021)	Summary and Recommendation
1.0 (21.07.2021)	Final document
1.1 (24.07.2021)	Added deployed contract addresses

## 2. About the Project and Company

### Company address:

Amun Limited  
Suite 202 2nd Floor  
Eden Plaza, Eden Island  
PO Box 1352, Mahe  
Seychelles

**Website:** <https://amun.com>

**Twitter:** [twitter.com/amuntokens](https://twitter.com/amuntokens)

**Medium:** <https://medium.com/amun-tokens>

**Telegram:** <https://t.me/AmunTokens>

**Discord:** <https://discord.gg/MVJmyCzahH>

**Github:** <https://github.com/amun/contracts>

AMUN

## 2.1 Project Overview

Amun DeFi is a set of Ethereum-based tools that allow users to maximize their returns across major DeFi platforms through lending and staking by helping them choose the best product at any given time. Amun DeFi supports integrations with popular protocols representing tens of billions of total locked value (TVL). From purchasing major cryptocurrencies quickly and with ease, to swapping between popular ERC-20 tokens, all the way to lending tokens at the best available rates, Amun DeFi makes it simple to monitor and interact with a user's holdings on one secure, easy-to-use platform.

Through gateways with various lending platforms and AMMs, Amun DeFi enables users to execute their desired investment strategy with relative ease. Because there are so many different platforms that users can use to achieve maximum returns, the average DeFi user might not know precisely where to begin. Further, many retail investors will be unable to implement their intended strategy due to a lack of knowledge or technical sophistication.

Amun's goal is to reduce the fragmentation of this space by offering access to all the top protocols in one platform. They intend to offer the following features to the DeFi community :

1. Dashboard that monitors your holdings and depicts your exposure across the various protocols ;
2. Ability to invest in the top liquidity pools and lending protocols ;
3. Offers our own products (autopilot strategies) that automatically moves your funds to the protocol offering the best yields ;
4. Seamlessly diversify your investments through our basket tokens.

### 3. Vulnerability & Risk Level

Risk represents the probability that a certain source-threat will exploit vulnerability, and the impact of that event on the organization or system. Risk Level is computed based on CVSS version 3.0.

Level	Value	Vulnerability	Risk (Required Action)
Critical	9 – 10	A vulnerability that can disrupt the contract functioning in a number of scenarios, or creates a risk that the contract may be broken.	Immediate action to reduce risk level.
High	7 – 8.9	A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.	Implementation of corrective actions as soon as possible.
Medium	4 – 6.9	A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.	Implementation of corrective actions in a certain period.
Low	2 – 3.9	A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.	Implementation of certain corrective actions or accepting the risk.
Informational	0 – 1.9	A vulnerability that have informational character but is not effecting any of the code.	An observation that does not determine a level of risk

## 4. Auditing Strategy and Techniques Applied

Throughout the review process, care was taken to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices. To do so, reviewed line-by-line by our team of expert pentesters and smart contract developers, documenting any issues as there were discovered.

### 4.1 Methodology

The auditing process follows a routine series of steps:

1. Code review that includes the following:
  - i. Review of the specifications, sources, and instructions provided to Chainsulting to make sure we understand the size, scope, and functionality of the smart contract.
  - ii. Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
  - iii. Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to Chainsulting describe.
2. Testing and automated analysis that includes the following:
  - i. Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run those test cases.
  - ii. Symbolic execution, which is analysing a program to determine what inputs causes each part of a program to execute.
3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarify, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.
4. Specific, itemized, actionable recommendations to help you take steps to secure your smart contracts.

## 4.2 Used Code from other Frameworks/Smart Contracts (direct imports)

Dependency / Import Path	Source
@openzeppelin/contracts/token/ERC20/IERC20.sol	<a href="https://github.com/OpenZeppelin/openzeppelin-contracts/blob/v3.2.1-solc-0.7/contracts/token/ERC20/IERC20.sol">https://github.com/OpenZeppelin/openzeppelin-contracts/blob/v3.2.1-solc-0.7/contracts/token/ERC20/IERC20.sol</a>
@pangolindex/exchange-contracts/contracts/pangolin-periphery/interfaces/IPangolinRouter.sol	<a href="https://github.com/pangolindex/exchange-contracts/blob/main/contracts/pangolin-periphery/interfaces/IPangolinRouter.sol">https://github.com/pangolindex/exchange-contracts/blob/main/contracts/pangolin-periphery/interfaces/IPangolinRouter.sol</a>
@uniswap/v2-periphery/contracts/interfaces/IUniswapV2Router02.sol	<a href="https://github.com/Uniswap/uniswap-v2-periphery/blob/master/contracts/interfaces/IUniswapV2Router02.sol">https://github.com/Uniswap/uniswap-v2-periphery/blob/master/contracts/interfaces/IUniswapV2Router02.sol</a>
@uniswap/v3-periphery/contracts/interfaces/IQuoterV2.sol	<a href="https://github.com/Uniswap/uniswap-v3-periphery/blob/main/contracts/interfaces/IQuoterV2.sol">https://github.com/Uniswap/uniswap-v3-periphery/blob/main/contracts/interfaces/IQuoterV2.sol</a>
@uniswap/v3-periphery/contracts/interfaces/ISwapRouter.sol	<a href="https://github.com/Uniswap/uniswap-v3-periphery/blob/main/contracts/interfaces/ISwapRouter.sol">https://github.com/Uniswap/uniswap-v3-periphery/blob/main/contracts/interfaces/ISwapRouter.sol</a>

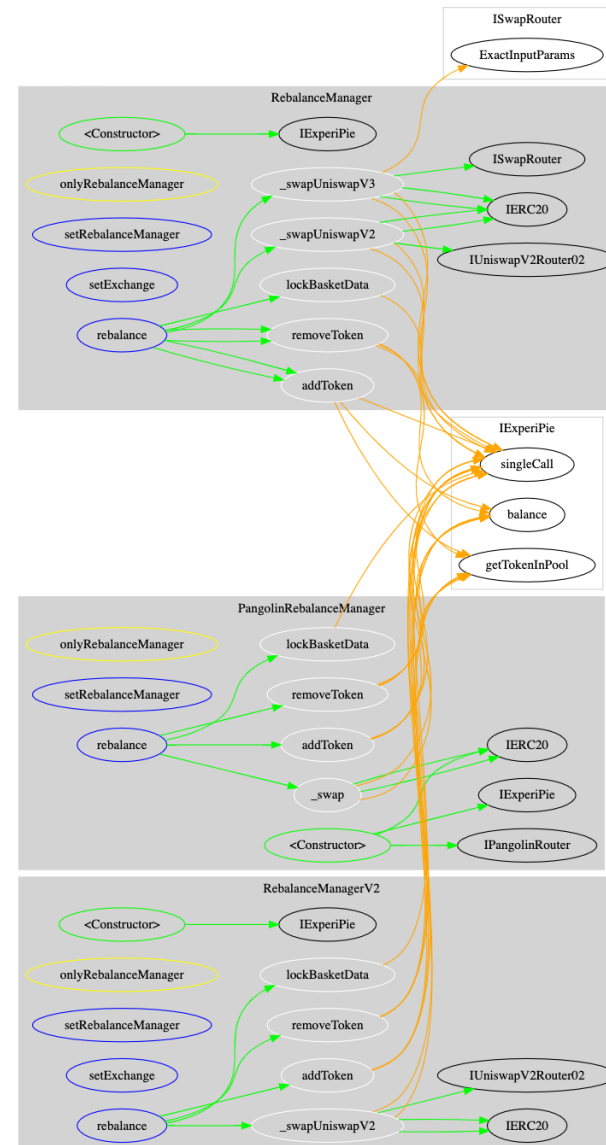
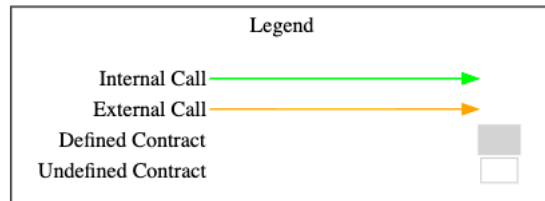
## 4.3 Tested Contract Files

The following are the MD5 hashes of the reviewed files. A file with a different MD5 hash has been modified, intentionally or otherwise, after the security review. You are cautioned that a different MD5 hash could be (but is not necessarily) an indication of a changed condition or potential vulnerability that was not within the scope of the review

File	Fingerprint (MD5)
./basket/callManagers/RebalanceManager.sol	ce4c5ef5ed27aa8704eeec013a4fd7d1
./basket/callManagers/RebalanceManagerV2.sol	591fa6ff4a96433970878fc32473c2c1
./basket/callManagers/PangolinRebalanceManager.sol	ca6dc8c0a3918af1cab36ba4c2929ca6

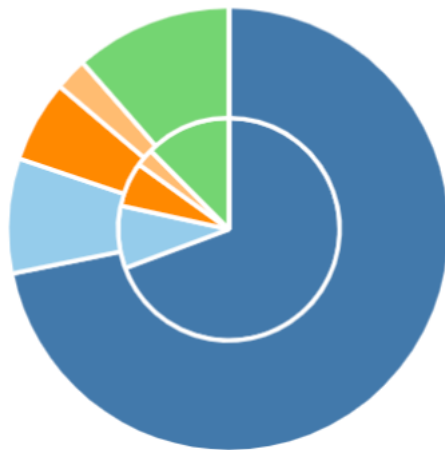


## 4.4 Metrics / CallGraph

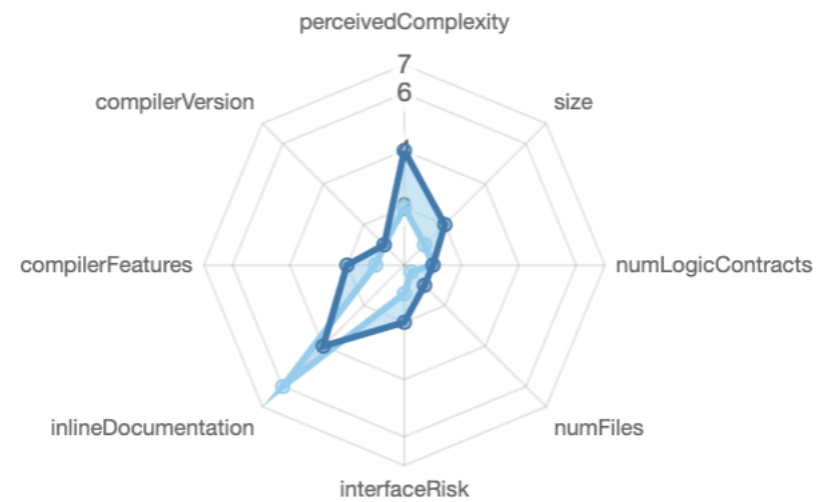


## 4.5 Metrics / Source Lines & Risk











source comment single block mixed  
empty todo blockEmpty



overall average





## 4.6 Metrics / Capabilities


Solidity Versions observed		 Experimental Features	 Can Receive Funds	 Uses Assembly	 Has Destroyable Contracts
<code>^0.7.5</code>		<code>ABIEncoderV2</code>		**** (0 asm blocks)	
 Transfers ETH	 Low-Level Calls	 DelegateCall	 Uses Hash Functions	 ECRecover	 New/Create/Create2

### Exposed Functions


This section lists functions that are explicitly declared public or payable. Please note that getter methods for public stateVars are not included.

 <b>Public</b>	 <b>Payable</b>				
8	0				
<b>External</b>	<b>Internal</b>	<b>Private</b>	<b>Pure</b>	<b>View</b>	
8	37	0	0	0	

### StateVariables

<b>Total</b>	 <b>Public</b>
10	9

## 4.7 Metrics / Source Unites in Scope

Type	File	Logic Contracts	Interfaces	Lines	nLines	nSLOC	Comment Lines	Complex. Score	Capabilities
	basket/callManagers/RebalanceManager.sol	1	_____	254	233	180	22	92	
	basket/callManagers/PangolinRebalanceManager.sol	1	_____	181	164	125	18	78	
	basket/callManagers/RebalanceManagerV2.sol	1	_____	179	163	122	16	64	
	<b>Totals</b>	<b>3</b>	_____	<b>614</b>	<b>560</b>	<b>427</b>	<b>56</b>	<b>234</b>	

Legend: [ ]

- **Lines**: total lines of the source unit
- **nLines**: normalized lines of the source unit (e.g. normalizes functions spanning multiple lines)
- **nSLOC**: normalized source lines of code (only source-code lines; no comments, no blank lines)
- **Comment Lines**: lines containing single or block comments
- **Complexity Score**: a custom complexity score derived from code statements that are known to introduce code complexity (branches, loops, calls, external interfaces, ...)

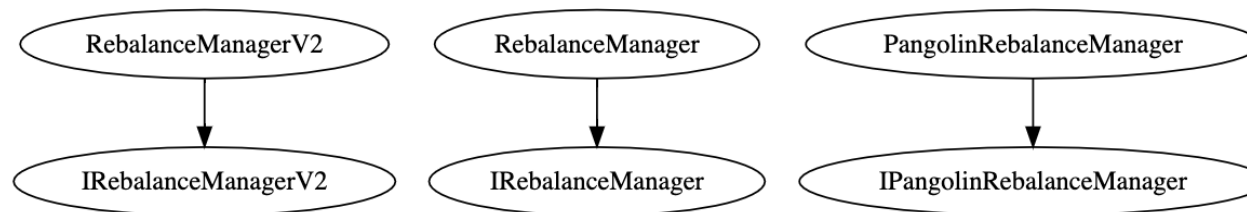
## 5. Scope of Work

The Amun Team provided us with the files that needs to be tested. The scope of the audit are the Rebalance Manager contracts.

The team put forward the following assumptions regarding the security, usage of the contracts:

- The smart contract is coded according to the newest standards and in a secure way
- Compare codebase with <https://github.com/pie-dao/PieVaults/tree/master/contracts> and point out the diffs and check them twice.
- List all audits from Pie-DAO and check if the findings effecting Amun and if they are fixed.

The main goal of this audit was to verify these claims. The auditors can provide additional feedback on the code upon the client's request.



## 5.1 Manual and Automated Vulnerability Test

### CRITICAL ISSUES

During the audit, Chainsulting's experts found **no Critical issues** in the code of the smart contract.

### HIGH ISSUES

During the audit, Chainsulting's experts found **no High issues** in the code of the smart contract.

### MEDIUM ISSUES

During the audit, Chainsulting's experts found **no Medium issues** in the code of the smart contract

### LOW ISSUES

#### 5.1.1 No package.json

Severity: LOW

Status: ACKNOWLEDGED

File(s) affected: Package.json

Attack / Description	Code Snippet	Result/Recommendation
In the current implementation, and repository is no package.json included to correctly install the needed dependencies.	Line: 5-8 (PangolinRebalanceManager.sol) <pre>import {     IPangolinRouter } from "@pangolindex/exchange- contracts/contracts/pangolin- periphery/interfaces/IPangolinRouter.sol"; import "@openzeppelin/contracts/token/ERC20/IERC20.sol";</pre>	We highly recommend using npm and package.json in order to guarantee that the right libraries and versions are installed.  "@openzeppelin/contracts": "^3.2.1-solc-0.7", "@pangolindex/exchange-contracts": "1.0.1", "@uniswap/v3-periphery": "1.1.1", "@uniswap/v2-periphery": "1.1.0",



Line: 5-10 (RebalanceManager.sol)

```
import
"@openzeppelin/contracts/token/ERC20/IERC20.sol";

import "@uniswap/v3-
periphery/contracts/interfaces/ISwapRouter.sol";
import "@uniswap/v3-
periphery/contracts/interfaces/IQuoterV2.sol";

import "@uniswap/v2-
periphery/contracts/interfaces/IUniswapV2Router02.sol";
```

Line 5-7 (RebalanceManagerV2.sol)

```
import
"@openzeppelin/contracts/token/ERC20/IERC20.sol";

import "@uniswap/v2-
periphery/contracts/interfaces/IUniswapV2Router02.sol";
```

## 5.2. SWC Attacks

ID	Title	Relationships	Test Result
<a href="#">SWC-131</a>	Presence of unused variables	<a href="#">CWE-1164: Irrelevant Code</a>	✓
<a href="#">SWC-130</a>	Right-To-Left-Override control character (U+202E)	<a href="#">CWE-451: User Interface (UI) Misrepresentation of Critical Information</a>	✓
<a href="#">SWC-129</a>	Typographical Error	<a href="#">CWE-480: Use of Incorrect Operator</a>	✓
<a href="#">SWC-128</a>	DoS With Block Gas Limit	<a href="#">CWE-400: Uncontrolled Resource Consumption</a>	✓
<a href="#">SWC-127</a>	Arbitrary Jump with Function Type Variable	<a href="#">CWE-695: Use of Low-Level Functionality</a>	✓
<a href="#">SWC-125</a>	Incorrect Inheritance Order	<a href="#">CWE-696: Incorrect Behavior Order</a>	✓
<a href="#">SWC-124</a>	Write to Arbitrary Storage Location	<a href="#">CWE-123: Write-what-where Condition</a>	✓
<a href="#">SWC-123</a>	Requirement Violation	<a href="#">CWE-573: Improper Following of Specification by Caller</a>	✓



ID	Title	Relationships	Test Result
<a href="#">SWC-122</a>	Lack of Proper Signature Verification	<a href="#">CWE-345: Insufficient Verification of Data Authenticity</a>	✓
<a href="#">SWC-121</a>	Missing Protection against Signature Replay Attacks	<a href="#">CWE-347: Improper Verification of Cryptographic Signature</a>	✓
<a href="#">SWC-120</a>	Weak Sources of Randomness from Chain Attributes	<a href="#">CWE-330: Use of Insufficiently Random Values</a>	✓
<a href="#">SWC-119</a>	Shadowing State Variables	<a href="#">CWE-710: Improper Adherence to Coding Standards</a>	✓
<a href="#">SWC-118</a>	Incorrect Constructor Name	<a href="#">CWE-665: Improper Initialization</a>	✓
<a href="#">SWC-117</a>	Signature Malleability	<a href="#">CWE-347: Improper Verification of Cryptographic Signature</a>	✓
<a href="#">SWC-116</a>	Timestamp Dependence	<a href="#">CWE-829: Inclusion of Functionality from Untrusted Control Sphere</a>	✓
<a href="#">SWC-115</a>	Authorization through tx.origin	<a href="#">CWE-477: Use of Obsolete Function</a>	✓
<a href="#">SWC-114</a>	Transaction Order Dependence	<a href="#">CWE-362: Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')</a>	✓

ID	Title	Relationships	Test Result
<a href="#">SWC-113</a>	DoS with Failed Call	<a href="#">CWE-703: Improper Check or Handling of Exceptional Conditions</a>	✓
<a href="#">SWC-112</a>	Delegatecall to Untrusted Callee	<a href="#">CWE-829: Inclusion of Functionality from Untrusted Control Sphere</a>	✓
<a href="#">SWC-111</a>	Use of Deprecated Solidity Functions	<a href="#">CWE-477: Use of Obsolete Function</a>	✓
<a href="#">SWC-110</a>	Assert Violation	<a href="#">CWE-670: Always-Incorrect Control Flow Implementation</a>	✓
<a href="#">SWC-109</a>	Uninitialized Storage Pointer	<a href="#">CWE-824: Access of Uninitialized Pointer</a>	✓
<a href="#">SWC-108</a>	State Variable Default Visibility	<a href="#">CWE-710: Improper Adherence to Coding Standards</a>	✓
<a href="#">SWC-107</a>	Reentrancy	<a href="#">CWE-841: Improper Enforcement of Behavioral Workflow</a>	✓
<a href="#">SWC-106</a>	Unprotected SELFDESTRUCT Instruction	<a href="#">CWE-284: Improper Access Control</a>	✓
<a href="#">SWC-105</a>	Unprotected Ether Withdrawal	<a href="#">CWE-284: Improper Access Control</a>	✓
<a href="#">SWC-104</a>	Unchecked Call Return Value	<a href="#">CWE-252: Unchecked Return Value</a>	✓

ID	Title	Relationships	Test Result
<a href="#">SWC-103</a>	Floating Pragma	<a href="#">CWE-664: Improper Control of a Resource Through its Lifetime</a>	✗
<a href="#">SWC-102</a>	Outdated Compiler Version	<a href="#">CWE-937: Using Components with Known Vulnerabilities</a>	✓
<a href="#">SWC-101</a>	Integer Overflow and Underflow	<a href="#">CWE-682: Incorrect Calculation</a>	✓
<a href="#">SWC-100</a>	Function Default Visibility	<a href="#">CWE-710: Improper Adherence to Coding Standards</a>	✓

### 5.3. Associated audits with the forked codebase

Mixbytes	<a href="https://github.com/pie-dao/audits/blob/main/Mixbytes%20-%20ExperiPie%20Smart%20Contrac%202020-12-11.pdf">https://github.com/pie-dao/audits/blob/main/Mixbytes%20-%20ExperiPie Smart Contrac%202020-12-11.pdf</a>	Dec. 2020
----------	---	-----------

### 5.4. Verify Claims

#### 5.4.1 The smart contract is coded according to the newest standards and in a secure way

Status: tested and verified ✓

#### 5.4.2 Compare codebase with <https://github.com/pie-dao/PieVaults/tree/master/contracts> and point out the diffs and check them twice.

Status: tested and verified ✓

```
diff -qr ./contracts-master/contracts/basket ./PieVaults-master/contracts | sort
```

```
Files ./contracts-master/contracts/basket/callManagers/LendingManager/LendingLogicAave.sol and ./PieVaults-master/contracts/callManagers/LendingManager/LendingLogicAave.sol differ
```

```
Files ./contracts-master/contracts/basket/callManagers/LendingManager/LendingLogicAaveV2.sol and ./PieVaults-master/contracts/callManagers/LendingManager/LendingLogicAaveV2.sol differ
```

```
Files ./contracts-master/contracts/basket/callManagers/LendingManager/LendingLogicCompound.sol and ./PieVaults-master/contracts/callManagers/LendingManager/LendingLogicCompound.sol differ
```

```
Files ./contracts-master/contracts/basket/callManagers/LendingManager/LendingManager.sol and ./PieVaults-master/contracts/callManagers/LendingManager/LendingManager.sol differ
```

```
Files ./contracts-master/contracts/basket/callManagers/LendingManager/LendingRegistry.sol and ./PieVaults-master/contracts/callManagers/LendingManager/LendingRegistry.sol differ
```

```
Files ./contracts-master/contracts/basket/callManagers/LendingManager/StakeSushi.sol and ./PieVaults-master/contracts/callManagers/LendingManager/StakeSushi.sol differ
```

Files ./contracts-master/contracts/basket/callManagers/LendingManager/StakingLogicYGov.sol and ./PieVaults-master/contracts/callManagers/LendingManager/StakingLogicYGov.sol **differ**

Files ./contracts-master/contracts/basket/callManagers/RSIManager.sol and ./PieVaults-master/contracts/callManagers/RSIManager.sol **differ**

Files ./contracts-master/contracts/basket/facets/Basket/BasketFacet.sol and ./PieVaults-master/contracts/facets/Basket/BasketFacet.sol **differ**

Files ./contracts-master/contracts/basket/facets/Basket/LibBasketStorage.sol and ./PieVaults-master/contracts/facets/Basket/LibBasketStorage.sol **differ**

Files ./contracts-master/contracts/basket/facets/Call/CallFacet.sol and ./PieVaults-master/contracts/facets/Call/CallFacet.sol **differ**

Files ./contracts-master/contracts/basket/facets/Call/LibCallStorage.sol and ./PieVaults-master/contracts/facets/Call/LibCallStorage.sol **differ**

Files ./contracts-master/contracts/basket/facets/ERC20/ERC20Facet.sol and ./PieVaults-master/contracts/facets/ERC20/ERC20Facet.sol **differ**

Files ./contracts-master/contracts/basket/facets/ERC20/LibERC20.sol and ./PieVaults-master/contracts/facets/ERC20/LibERC20.sol **differ**

Files ./contracts-master/contracts/basket/facets/ERC20/LibERC20Storage.sol and ./PieVaults-master/contracts/facets/ERC20/LibERC20Storage.sol **differ**

Files ./contracts-master/contracts/basket/facets/shared/Access/CallProtection.sol and ./PieVaults-master/contracts/facets/shared/Access/CallProtection.sol **differ**

Files ./contracts-master/contracts/basket/facets/shared/Reentry/LibReentryProtectionStorage.sol and ./PieVaults-master/contracts/facets/shared/Reentry/LibReentryProtectionStorage.sol **differ**

Files ./contracts-master/contracts/basket/facets/shared/Reentry/ReentryProtection.sol and ./PieVaults-master/contracts/facets/shared/Reentry/ReentryProtection.sol **differ**

Files ./contracts-master/contracts/basket/factories/PieFactoryContract.sol and ./PieVaults-master/contracts/factories/PieFactoryContract.sol **differ**

Files ./contracts-master/contracts/basket/interfaces/IAToken.sol and ./PieVaults-master/contracts/interfaces/IAToken.sol **differ**

Files ./contracts-master/contracts/basket/interfaces/IATokenV2.sol and ./PieVaults-master/contracts/interfaces/IATokenV2.sol **differ**

Files ./contracts-master/contracts/basket/interfaces/IAaveLendingPool.sol and ./PieVaults-master/contracts/interfaces/IAaveLendingPool.sol **differ**

Files ./contracts-master/contracts/basket/interfaces/IAaveLendingPoolV2.sol and ./PieVaults-master/contracts/interfaces/IAaveLendingPoolV2.sol **differ**

Files ./contracts-master/contracts/basket/interfaces/IBasketFacet.sol and ./PieVaults-master/contracts/interfaces/IBasketFacet.sol **differ**  
Files ./contracts-master/contracts/basket/interfaces/ICToken.sol and ./PieVaults-master/contracts/interfaces/ICToken.sol **differ**  
Files ./contracts-master/contracts/basket/interfaces/ICallFacet.sol and ./PieVaults-master/contracts/interfaces/ICallFacet.sol **differ**  
Files ./contracts-master/contracts/basket/interfaces/IERC20Facet.sol and ./PieVaults-master/contracts/interfaces/IERC20Facet.sol **differ**  
Files ./contracts-master/contracts/basket/interfaces/IExperiPie.sol and ./PieVaults-master/contracts/interfaces/IExperiPie.sol **differ**  
Files ./contracts-master/contracts/basket/interfaces/ILendingLogic.sol and ./PieVaults-master/contracts/interfaces/ILendingLogic.sol **differ**  
Files ./contracts-master/contracts/basket/interfaces/IPriceReferenceFeed.sol and ./PieVaults-master/contracts/interfaces/IPriceReferenceFeed.sol **differ**  
Files ./contracts-master/contracts/basket/interfaces/ISynthetix.sol and ./PieVaults-master/contracts/interfaces/ISynthetix.sol **differ**  
Files ./contracts-master/contracts/basket/interfaces/IXSushi.sol and ./PieVaults-master/contracts/interfaces/IXSushi.sol **differ**  
Files ./contracts-master/contracts/basket/interfaces/IYVault.sol and ./PieVaults-master/contracts/interfaces/IYVault.sol **differ**  
Files ./contracts-master/contracts/basket/test/DiamondFactoryContract.sol and ./PieVaults-master/contracts/test/DiamondFactoryContract.sol **differ**  
Files ./contracts-master/contracts/basket/test/ERC20FactoryContract.sol and ./PieVaults-master/contracts/test/ERC20FactoryContract.sol **differ**  
Files ./contracts-master/contracts/basket/test/ManualPriceReferenceFeed.sol and ./PieVaults-master/contracts/test/ManualPriceReferenceFeed.sol **differ**  
Files ./contracts-master/contracts/basket/test/MockAToken.sol and ./PieVaults-master/contracts/test/MockAToken.sol **differ**  
Files ./contracts-master/contracts/basket/test/MockATokenV2.sol and ./PieVaults-master/contracts/test/MockATokenV2.sol **differ**  
Files ./contracts-master/contracts/basket/test/MockAaveLendingPool.sol and ./PieVaults-master/contracts/test/MockAaveLendingPool.sol **differ**  
Files ./contracts-master/contracts/basket/test/MockAaveLendingPoolV2.sol and ./PieVaults-master/contracts/test/MockAaveLendingPoolV2.sol **differ**

Files ./contracts-master/contracts/basket/test/MockCToken.sol and ./PieVaults-master/contracts/test/MockCToken.sol **differ**

Files ./contracts-master/contracts/basket/test/MockLendingLogic.sol and ./PieVaults-master/contracts/test/MockLendingLogic.sol **differ**

Files ./contracts-master/contracts/basket/test/MockSynthetix.sol and ./PieVaults-master/contracts/test/MockSynthetix.sol **differ**

Files ./contracts-master/contracts/basket/test/MockToken.sol and ./PieVaults-master/contracts/test/MockToken.sol **differ**

Files ./contracts-master/contracts/basket/test/MockXSushi.sol and ./PieVaults-master/contracts/test/MockXSushi.sol **differ**

Files ./contracts-master/contracts/basket/test/MockYVault.sol and ./PieVaults-master/contracts/test/MockYVault.sol **differ**

**Only in** ./PieVaults-master/contracts/callManagers/LendingManager: DepositLogicDecimalWrapper.sol

**Only in** ./PieVaults-master/contracts/interfaces: IDecimalWrapper.sol

**Only in** ./PieVaults-master/contracts/test: MockDecimalWrapper.sol

**Only in** ./contracts-master/contracts/basket/callManagers: PangolinRebalanceManager.sol

**Only in** ./contracts-master/contracts/basket/callManagers: RebalanceManageV2.sol

**Only in** ./contracts-master/contracts/basket/callManagers: RebalanceManager.sol

**Only in** ./contracts-master/contracts/basket/interfaces: IPangolinRebalanceManager.sol

**Only in** ./contracts-master/contracts/basket/interfaces: IRebalanceManager.sol

**Only in** ./contracts-master/contracts/basket/interfaces: IRebalanceManagerV2.sol

**Only in** ./contracts-master/contracts/basket/interfaces: IUniswapV2Factory.sol

**Only in** ./contracts-master/contracts/basket/interfaces: IUniswapV2Pair.sol

**Only in** ./contracts-master/contracts/basket/interfaces: IUniswapV2Router01.sol


**Only in** ./contracts-master/contracts/basket/interfaces: IUniswapV2Router02.sol

**Only in** ./contracts-master/contracts/basket/test: MockPangolinRouter.sol

**Only in** ./contracts-master/contracts/basket/test: WETH9.sol

**Only in** ./contracts-master/contracts/basket: SingleTokenJoin.sol

#### 5.4.3 List all audits from Pie-DAO and check if the findings effecting Amun and if they are fixed.

**Status:** tested and verified 

## 6. Executive Summary

Two (2) independent Chainsulting experts performed an unbiased and isolated audit of the smart contract codebase. The final debriefs took place on the July 21, 2021. The overall code quality of the project is very good, and the modifications of the forked Pie-DAO Vault contracts decrease the attack surface, as they are already used since month and have been audited.

The main goal of the audit was to verify the claims regarding the security of the smart contract and the functions. During the audit, no critical issues were found after the manual and automated security testing and the claims been successfully verified.

## 7. Deployed Smart Contract

### VERIFIED

Amun DeFi Momentum Index (DMX): <https://etherscan.io/address/0x1660F10B4D610cF482194356eCe8eFD65B15bA83#code>

Amun DeFi Index (DFI): <https://etherscan.io/address/0xA9536B9c75A9E0faE3B56a96AC8EdF76AbC91978#code>

