

TRII Token Smart Contract Audit



## Summary

<b>Auditing Firm</b>	TRII DIENSTLEISTUNG SARL
<b>Client Firm</b>	TRII Token
<b>Architecture</b>	TRII DIENSTLEISTUNG Auditing Standard
<b>Language</b>	Solidity
<b>Mandatory Audit Check</b>	Static, Software, Auto Intelligent & Manual Analysis
<b>Final Report Date</b>	August 22, 2022

## Audit Summary

TRII DIENSTLEISTUNG team has performed a line-by-line manual analysis and automated review of the smart\_contract. The smart contract was analyzed mainly for common smart contract vulnerabilities, exploits, and manipulation hacks. According to the smart contract audit:

- ❖ TRII Token's smart contract source code has **LOW RISK SEVERITY**
- ❖ TRII Token's smart contract has an **ACTIVE OWNERSHIP**
- ❖ Important owner privileges – **BULK BLACKLIST, MAX TX & WALLET LIMIT, SET FEES, SET SELL FEES, SET HIGH FEES, SET COOLDOWN TIMER, SET TRADING STATUS**

The smart contract owner has 30 “Write Contract” privileges. **Centralization risk is HIGH with TRII Token.** For the detailed understanding of risk severity, source code vulnerability, and functional test, kindly refer to the audit.

🔗 Contract address: **0x0952b2b84342390E565Ff5463e6A49302d6e0E78**

🔗 Blockchain: **Binance Smart Chain**

✅ Verify the authenticity of this report on the verified organization TRII-Token @ [GitHub](#)



# Table Of Contents

## **Project Information**

Overview .....	4
----------------	---

## **TRII DIENSTLEISTUNG Audit Standard**

Audit Scope & Methodology .....	6
TRII DIENSTLEISTUNG's Risk Classification.....	8

## **Smart Contract Risk Assessment**

Static Analysis.....	9
Software Analysis .....	12
Manual Analysis.....	15
SWC Attacks .....	19
Risk Status & Radar Chart.....	21

## **Report Summary**

Auditor's Verdict .....	22
-------------------------	----

## **Legal Advisory**

Important Disclaimer.....	23
About TRII DIENSTLEISTUNG SARL.....	24

## Project Overview

TRII DIENSTLEISTUNG was consulted by TRII Token to conduct the smart contract security audit of their solidity\_source code.

### About TRII Token

TRII Token is the next Generation of a reflection token on the Binance Smart Chain (BSC)

Hold \$TRII tokens and get rewarded from each transaction, delivered automatically to your\_wallet.

Project	TRII Token
<b>Blockchain</b>	Binance Smart Chain
<b>Language</b>	Solidity
<b>Contract</b>	0x0952b2b84342390E565Ff5463e6A49302d6e0E78
<b>Website</b>	<a href="https://trii.eu">https://trii.eu</a>
<b>Telegram</b>	<a href="https://t.me/TRIIToken">https://t.me/TRIIToken</a>
<b>Twitter</b>	<a href="https://twitter.com/TRII_Token">https://twitter.com/TRII_Token</a>
<b>GitHub</b>	<a href="https://github.com/TRII-Token">https://github.com/TRII-Token</a>

## **Project Logo**



## **Solidity Source Code On Blockchain (Verified Contract Source Code).**

<https://bscscan.com/address/>

[0x0952b2b84342390E565Ff5463e6A49302d6e0E78#code](https://bscscan.com/address/0x0952b2b84342390E565Ff5463e6A49302d6e0E78#code)

Contract Name: TRII Token

Compiler Version: v0.8.7+

Optimization Enabled: Yes with 200 runs

## **SHA-1 Hash**

Solidity source code is audited at hash #e1af2e1b28fa91d8dc21268d2b1fd1cfad933af

## Audit Scope & Methodology

The scope of this report is to audit the smart contract source code of TRII Token. TRII DIENSTLEISTUNG has scanned the contract and reviewed the project for common vulnerabilities, exploits, hacks, and back-doors. Below is the list of commonly known smart contract vulnerabilities, exploits, and hacks:

Category	
Smart Contract Vulnerabilities	❖ Reentrancy
	❖ Unhandled Exceptions
	❖ Integer Overflow
	❖ Unrestricted Action
	❖ Incorrect Inheritance Order
	❖ Typographical Errors
	❖ Requirement Violation
Source Code Review	❖ Ownership Takeover
	❖ Gas Limit and Loops
	❖ Deployment Consistency
	❖ Repository Consistency
	❖ Data Consistency
Functional Assessment	❖ Token Supply Manipulation
	❖ Access Control and Authorization
	❖ Operations Trail and Event Generation
	❖ Assets Manipulation
	❖ Liquidity Access

## **TRII DIENSTLEISTUNG's Audit Standard**

The aim of TRII DIENSTLEISTUNG's standard is to analyze the smart contract and identify the vulnerabilities and the hacks in the smart contract. Mentioned are the steps used by our company to assess the smart contract:

### 1. Solidity smart contract source code reviewal:

- ❖ Review of the specifications, sources, and instructions provided to us to make sure we understand the size, scope, and functionality of the smart contract.
- ❖ Manual review of code, which is the process of reading source code line-byline to identify potential vulnerabilities.

### 2. Static, Manual, and Software analysis:

- ❖ Test coverage analysis, which is the process of determining whether the test cases are covering the code and how much code is exercised when we run those test cases.
- ❖ Symbolic execution, which is analyzing 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

## **Automated 3P frameworks used to assess the smart contract vulnerabilities**

- ❖ Slither
- ❖ Consensys MythX, Mythril
- ❖ SWC Registry
- ❖ Solidity Coverage
- ❖ Open Zeppelin Code Analyzer
- ❖ Solidity Code Compiler

## TRII DIENSTLEISTUNG's Risk Classification

Smart contracts are generally designed to manipulate and hold funds denominated in ETH/BNB. This makes them very tempting attack targets, as a successful attack may allow the attacker to directly steal funds from the contract. Below are the typical risk levels of a smart contract:

**Vulnerable:** A contract is vulnerable if it has been flagged by a static analysis tool as such. As we will see later, this means that some contracts may be vulnerable because of a false-positive.






**Exploitable:** A contract is exploitable if it is vulnerable and the vulnerability could be exploited by an external attacker. For example, if the “vulnerability” flagged by a tool is in a function which requires to own the contract, it would be vulnerable but not exploitable.

**Exploited:** A contract is exploited if it received a transaction on the main network which triggered one of its vulnerabilities. Therefore, a contract can be vulnerable or even exploitable without having been exploited.






















































Risk severity	Meaning
<b>! Critical</b>	This level vulnerabilities could be exploited easily, and can lead to asset loss, data loss, asset, or data manipulation. They should be fixed
<b>! High</b>	This level vulnerabilities are hard to exploit but very important to fix, they carry an elevated risk of smart contract manipulation, which can lead to high-risk_severity
<b>! Medium</b>	This level vulnerabilities should be fixed, as they carry an inherent risk of future_exploits, and hacks which may or may not impact the smart
<b>! Low</b>	This level vulnerabilities can be ignored. They are code style violations, and informational statements in the code. They may not affect the smart contract_execution



# Smart Contract – Static Analysis

Symbol	Meaning
	Function can be modified
	Function is payable
	Function is locked
	Function can be accessed
	Important functionality

```

| **Context** | Implementation | |||
| L | _msgSender | Internal  | | |
| L | _msgData | Internal  | | |
| |||||
| **IERC20** | Interface | |||
| L | totalSupply | External  | | NO  |
| L | balanceOf | External  | | NO  |
| L | transfer | External    | NO  |
| L | allowance | External  | | NO  |
| L | approve | External    | NO  |
| L | transferFrom | External    | NO  |
|
| |||||
| **Ownable** | Implementation | Context  |||
| L | <Constructor> | Public   | NO  |
| L | owner | Public  | NO   |
| L | renounceOwnership | Public    | onlyOwner | L
| transferOwnership | Public   | onlyOwner | L
| L | _setOwner | Private | | |
| |||||
| **Address** | Library |   |||
| L | sendValue | Internal | | |
| |||||
| **IFactory** | Interface |    |||
| L | createPair | External  | | NO  |
|
| |||||
| **IRouter** | Interface | |||
| L | factory | External   | NO    |
| L | WETH | External  | NO     |
| L | addLiquidityETH | External  | | NO  |
| L | swapExactTokensForETHSupportingFeeOnTransferTokens | External  | | NO  |
|

```

```

| **TRII** | Implementation | Context, IERC20, Ownable |||
| L | <Constructor> | Public !! | NO!! |
| L | name | Public ! | NO!! |
| L | symbol | Public ! | NO!! |
| L | decimals | Public ! | NO!! |
| L | totalSupply | Public ! | NO!! |
| L | balanceOf | Public ! | NO!! |
| L | allowance | Public ! | NO!! |
| L | approve | Public !! | NO!! |
| L | transferFrom | Public !! | NO!! |
| L | increaseAllowance | Public !! | NO!! |
| L | decreaseAllowance | Public ! | ! |
NO ! | L | transfer | Public ! | NO ! |
| L | isExcludedFromReward | Public ! | NO!! |
| L | reflectionFromToken | Public ! | NO!! |
| L | setTradingStatus | External ! | ! onlyOwner |
| L | tokenFromReflection | Public ! | NO!! |
| L | excludeFromReward | Public ! | ? onlyOwner |
| L | includeInReward | External ! | ? onlyOwner |
| L | excludeFromFee | Public !! | ? onlyOwner |
| L | includeInFee | Public !! | ? onlyOwner |
| L | isExcludedFromFee | Public !! | NO!! |
| L | setTaxes | Public ! ! | ? onlyOwner |
| L | setSellTaxes | Public ! | ! ? onlyOwner |
| L | setHighTaxes | Public ! | ? onlyOwner |
| _reflectRfi | Private ? | ? |
| L | _takeLiquidity | Private ? | ? |
| L | _takeMarketing | Private ? | ? |
| L | _takeDev | Private ? | ? |
| L | _takeTeam | Private ? | ? |
| L | _getValues | Private ? | |
| L | _getTValues | Private ? | |
| L | _getRValues1 | Private ? | |
| L | _getRValues2 | Private ? | |
| L | _getRate | Private ? | |
| L | _getCurrentSupply | Private ? | |
| L | _approve | Private ? | ? |
| L | _transfer | Private ? | ? |
| L | _tokenTransfer | Private ? | ? |
| L | swapAndLiquify | Private ? | ? lockTheSwap |
| L | addLiquidity | Private ? | ? |
| L | swapTokensForBNB | Private ? | ? |
| L | bulkExcludeFee | External ! | ? onlyOwner |
| L | updateMarketingWallet | External ! | ? onlyOwner |
| L | updateDevWallet | External ! | ? onlyOwner |
| L | updateTeamWallet | External ! | ? onlyOwner |
| L | updateCooldown | External ! | ? onlyOwner |
| L | updateSwapTokensAtAmount | External ! | ? onlyOwner |
| L | updateSwapEnabled | External ! | ? onlyOwner |
| L | updateIsBlacklisted | External ! | ? onlyOwner |

```

```

| L | bulkIsBlacklisted | External ! | ! | onlyOwner |
| L | updateAllowedTransfer | External ! | ! | onlyOwner |
| L | updateMaxTxLimit | External ! | ! | onlyOwner |
| L | updateMaxWalletlimit | External ! | ! | onlyOwner | L
| updateRouterAndPair | External ! | ! | onlyOwner | |
| L | rescueBNB | External ! | ! | onlyOwner |
| L | rescueAnyBEP20Tokens | Public ! | | onlyOwner |
| L | <Receive Ether> | External ! | | NO ! |

```

# Smart Contract – Software Analysis

## Function Signatures

```

39509351 => increaseAllowance(address,uint256)
119df25f => _msgSender()
8b49d47e => _msgData()
18160ddd => totalSupply()
70a08231 => balanceOf(address)
a9059cbb => transfer(address,uint256)
dd62ed3e => allowance(address,address)
095ea7b3 => approve(address,uint256)
23b872dd => transferFrom(address,address,uint256)
8da5cb5b => owner()
715018a6 => renounceOwnership()
f2fde38b => transferOwnership(address)
fc201122 => _setOwner(address)
24a084df => sendValue(address,uint256)
c9c65396 => createPair(address,address)
c45a0155 => factory()
ad5c4648 => WETH()
f305d719 => addLiquidityETH(address,uint256,uint256,uint256,address,uint256)
791ac947 =>
swapExactTokensForETHSupportingFeeOnTransferTokens(uint256,uint256,address[],address,uint256)
06fdde03 => name()
95d89b41 => symbol()
313ce567 => decimals()
a457c2d7 => decreaseAllowance(address,uint256)
88f82020 => isExcludedFromReward(address)
4549b039 => reflectionFromToken(uint256,bool)
6ebfa4d2 => setTradingStatus(bool,uint256)
2d838119 => tokenFromReflection(uint256)
52390c02 => excludeFromReward(address)
3685d419 => includeInReward(address)
437823ec => excludeFromFee(address)
ea2f0b37 => includeInFee(address)
5342acb4 => isExcludedFromFee(address)
5fcefa42 => setTaxes(uint256,uint256,uint256,uint256,uint256)
c791fc54 => setSellTaxes(uint256,uint256,uint256,uint256,uint256)
e0b1c674 => setHighTaxes(uint256,uint256,uint256,uint256,uint256)
e8ffedee => _reflectRfi(uint256,uint256)
814bbf3f => _takeLiquidity(uint256,uint256)
916d2685 => _takeMarketing(uint256,uint256)
ff213ce2 => _takeDev(uint256,uint256)
ddcb6360 => _takeTeam(uint256,uint256)
f86adbb6 => _getValues(uint256,bool,bool,bool)
f8700692 => _getTValues(uint256,bool,bool,bool)
21080db3 => _getRValues1(valuesFromGetValues,uint256,bool,uint256)
c068776b => _getRValues2(valuesFromGetValues,bool,uint256)

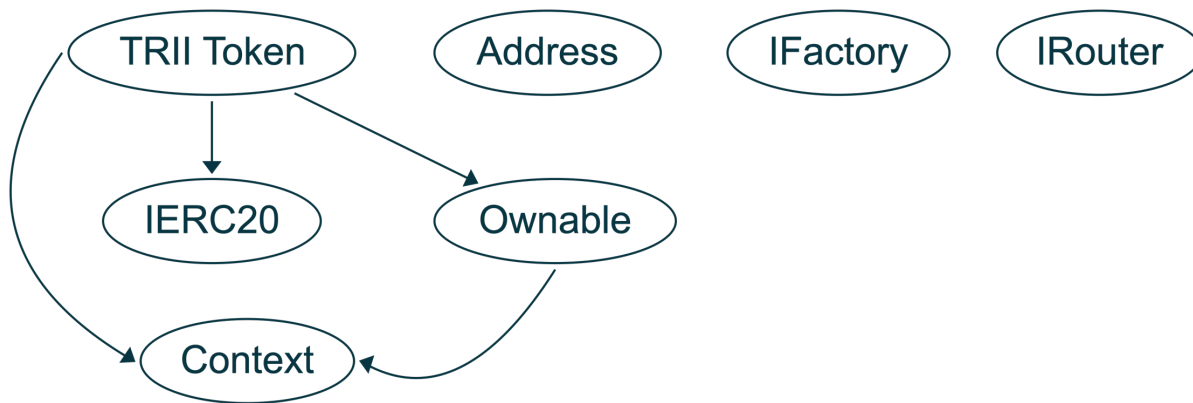
```

```

94e10784 => _getRate()
97a9d560 => _getCurrentSupply()
104e81ff => _approve(address,address,uint256)
ba4c825e => _transfer(address,address,uint256,bool)
1cfb26d9 => _tokenTransfer(address,address,uint256,bool,bool,bool)
34eb180d => swapAndLiquify(uint256,Taxes)
9cd441da => addLiquidity(uint256,uint256)
56c3726b => swapTokensForBNB(uint256)
8e2839ff => bulkExcludeFee(address[],bool)
aacebbe3 => updateMarketingWallet(address)
1816467f => updateDevWallet(address)
7cb332bb => updateTeamWallet(address)
e517f2b9 => updateCooldown(bool,uint256)
d257b34f => updateSwapTokensAtAmount(uint256)
924de9b7 => updateSwapEnabled(bool)
5b24ea5e => updateIsBlacklisted(address,bool)
13f97a8e => bulkIsBlacklisted(address[],bool)
b5c57145 => updateAllowedTransfer(address,bool)
59759f61 => updateMaxTxLimit(uint256,uint256)
d8672e51 => updateMaxWalletlimit(uint256)
40b28c2f => updateRouterAndPair(address,address)
441b1d30 => rescueBNB(uint256)
47c23092 => rescueAnyBEP20Tokens(address,address,uint256)

```

## Inheritance Graph



## Smart Contract – Manual Analysis

Function	Description	Tested	Verdict
<b>Total Supply</b>	provides information about the total token_supply	Yes	<b>Passed</b>
<b>Balance Of</b>	provides account balance of the owner's_account	Yes	<b>Passed</b>
<b>Transfer</b>	executes transfers of a specified number of	tokens	to a specified
<b>Approve</b>	allow a spender to withdraw a set number of	tokens	from a specified
<b>Allowance</b>	returns a set number of tokens from a spender to_the owner	Yes	<b>Passed</b>
<b>Buy Back</b>	is an action in which the project buys back its_tokens from the existing holders usually at a_market price	NA	NA
<b>Burn</b>	executes transfers of a specified number of_tokens to a burn address	NA	NA
<b>Mint</b>	executes creation of a specified number of_tokens and adds it to the circulating token supply adjusts	total supply	ly
<b>Rebase</b>	(increases or_decreases) automatically according to a token's_price fluctuations	NA	NA
<b>Blacklist</b>	stops specified wallets from interacting with the	smart	contract function
<b>Lock</b>	stops or locks all function modules of the smart_contract	NA	NA

Function	Description	Tested	Verdict
<b>Dividend</b>	executes transfers of a specified dividend token to a specified address	NA	NA
<b>Airdrop</b>	executes transfers of a specified number of tokens to a specified address	NA	NA
<b>Max Transaction</b>	a non-whitelisted wallet can only transfer a specified number of tokens	Yes	<b>! Low</b>
<b>Max Wallet</b>	a non-whitelisted wallet can only hold a specified number of tokens	Yes	<b>! Low</b>
<b>Cooldown Timer</b>	functionality to limit the number of transactions that a wallet can make	wit hin	a 24- hour
<b>Anti Bot</b>	stops some or all bot wallets from interacting with the smart contract	Yes	<b>Passed</b>
<b>Anti Snipe</b>	prevents bots from making transaction at “addLiquidity” block	NA	NA
<b>Transfer Ownership</b>	executes transfer of contract ownership to a specified wallet	Yes	<b>Passed</b>
<b>Renounce Ownership</b>	executes transfer of contract ownership to a dead address	Yes	<b>Passed</b>



## Best Practices

- ❖ Owner cannot stop or pause the smart contract.
- ❖ Owner cannot lock or burn the user assets.
- ❖ Owner cannot mint tokens after initial contract creation/deployment.
- ❖ The smart contract utilizes “SafeMath” function to avoid common smart contract vulnerabilities.

```
string private _name = "TRII Token";
library SafeMath {
function add(uint256 a, uint256 b) internal pure returns (uint256) {
    uint256 c = a + b;
    require(c >= a, "SafeMath: addition overflow");

function sub(uint256 a, uint256 b) internal pure returns (uint256) {
    return sub(a, b, "SafeMath: subtraction overflow");

    uint256 c = a * b;

    require(c / a == b, "SafeMath: multiplication overflow");

    return c;
function div(uint256 a, uint256 b) internal pure returns (uint256) {
    return div(a, b, "SafeMath: division by zero");

function mod(uint256 a, uint256 b) internal pure returns (uint256) {
    return mod(a, b, "SafeMath: modulo by zero");
```

## Note

- ❖ Active smart contract owner: 0x84F6384E694966Ab3Fbe50cD1F65028DE658ccd9
- ❖ *Be aware that active smart contract owner privileges constitute an elevated impact to smart contract's safety and security.*
- ❖ Smart contract owner can **bulk blacklist** certain wallets from interacting with the contract function modules.
- ❖ Smart contract owner can **change transaction taxes and sell taxes**. This function module can be used to impose extraordinary transaction fees. *No arbitrary limit set.*

- ❖ Smart contract owner can *change high fees*. This function module can be used to impose extraordinary transaction fees. *No arbitrary limit set.*
- ❖ Smart contract owner can *change max transaction and wallet %*. The smart contract owner can change the value to “zero”. *No arbitrary limit set.*
- ❖ Smart contract owner can *set cooldown timer %*.

## Smart Contract – SWC Attacks

SWC ID	Description	Verdict
SWC-101	Integer Overflow and Underflow	Passed
SWC-102	Outdated Compiler Version	! Low
SWC-103	Floating Pragma	Passed
SWC-104	Unchecked Call Return Value	Passed
SWC-105	Unprotected Ether Withdrawal	Passed
SWC-106	Unprotected SELFDESTRUCT Instruction	Passed
SWC-107	Re-entrancy	Passed
SWC-108	State Variable Default Visibility	Passed
SWC-109	Uninitialized Storage Pointer	Passed
SWC-110	Assert Violation	Passed
SWC-111	Use of Deprecated Solidity Functions	Passed
SWC-112	Delegate Call to Untrusted Callee	Passed
SWC-113	DoS with Failed Call	Passed
SWC-114	Transaction Order Dependence	Passed
SWC-115	Authorization through tx.origin	Passed
SWC-116	Block values as a proxy for time	Passed
SWC-117	Signature Malleability	Passed
SWC-118	Incorrect Constructor Name	Passed

<b>SWC-119</b>	Shadowing State Variables	<b>Passed</b>
<b>SWC-120</b>	Weak Sources of Randomness from Chain Attributes	<b>Passed</b>
<b>SWC-121</b>	Missing Protection against Signature Replay Attacks	<b>Passed</b>
<b>SWC-122</b>	Lack of Proper Signature Verification	<b>Passed</b>
<b>SWC-123</b>	Requirement Violation	<b>Passed</b>
<b>SWC-124</b>	Write to Arbitrary Storage Location	<b>Passed</b>
<b>SWC-125</b>	Incorrect Inheritance Order	<b>Passed</b>
<b>SWC-126</b>	Insufficient Gas Griefing	<b>Passed</b>
<b>SWC-127</b>	Arbitrary Jump with Function Type Variable	<b>Passed</b>
<b>SWC-128</b>	DoS With Block Gas Limit	<b>Passed</b>
<b>SWC-129</b>	Typographical Error	<b>Passed</b>
<b>SWC-130</b>	Right-To-Left-Override control character (U+202E)	<b>Passed</b>
<b>SWC-131</b>	Presence of unused variables	<b>Passed</b>
<b>SWC-132</b>	Unexpected Ether balance	<b>Passed</b>
<b>SWC-133</b>	Hash Collisions With Multiple Variable Length Arguments	<b>Passed</b>
<b>SWC-134</b>	Message call with hardcoded gas amount	<b>Passed</b>
<b>SWC-135</b>	Code With No Effects (Irrelevant/Dead Code)	<b>Passed</b>
<b>SWC-136</b>	Unencrypted Private Data On-Chain	<b>Passed</b>

## Smart Contract - Risk Status & Radar Chart

Risk Severity	Status
<b>! Critical</b>	None critical severity issues identified
<b>! High</b>	None high severity issues identified
<b>! Medium</b>	None medium severity issues identified
<b>! Low</b>	6 low severity issues identified
<b>Verified</b>	54 functions and instances verified and checked

## Auditor's Verdict

TRII DIENSTLEISTUNG team has performed a line-by-line manual analysis and automated review of the smart\_contract. The smart contract was analyzed mainly for common smart contract vulnerabilities, exploits, and manipulation hacks.

- ❖ TRII Token's smart contract source code has **LOW RISK SEVERITY**
- ❖ TRII Token's smart contract has an **ACTIVE OWNERSHIP**

### Note for stakeholders

- ❖ Be aware that active smart contract owner privileges constitute an elevated impact on smart contract's safety and security.
- ❖ Make sure that the project team's KYC/identity is verified by an independent firm, e.g., TRII DIENSTLEISTUNG.
- ❖ Always check if the contract's liquidity is locked. A longer liquidity lock plays an important role in project's longevity. It is recommended to have multiple liquidity providers.
- ❖ Examine the unlocked token supply in the owner, developer, or team's private wallets. Understand the project's tokenomics, and make sure the tokens outside of the LP Pair are vested or locked for a longer period of time.
- ❖ Ensure that the project's official website is hosted on a trusted platform, and is using an active SSL certificate. The website's domain should be registered for a longer period of time.

## Important Disclaimer

TRII DIENSTLEISTUNG SARL provides contract auditing and project verification services for blockchain projects. The purpose of the audit is to analyze the on-chain smart contract source code, and to provide basic overview of the project. **This report should not be transmitted, disclosed, referred to, or relied upon by any person for any purposes without TRII DIENSTLEISTUNG's prior written consent.**

TRII DIENSTLEISTUNG provides the easy-to-understand assessment of the project, and the smart contract (otherwise known as the source code). The audit makes no statements or warranties on the security of the code. It also cannot be considered as an enough assessment regarding the utility and safety of the code, bug-free status, or any other statements of the contract. While we have used all the data at our disposal to provide the transparent analysis, it is important to note that you should not rely on this report only — we recommend proceeding with several independent audits and a public bug bounty program to ensure the security of smart contracts. **Be aware that smart contracts deployed on a blockchain aren't resistant from external vulnerability, or a hack. Be aware that active smart contract owner privileges constitute an elevated impact to smart contract's safety and security. Therefore, TRII DIENSTLEISTUNG does not guarantee the explicit security of the audited smart contract.**

The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

**This report should not be considered as an endorsement or disapproval of any project or team.**

The information provided on this report does not constitute investment advice, financial advice, trading advice, or any other sort of advice and you should not treat any of the report's content as such. Do conduct your own due diligence and consult your financial advisor before making any investment decisions.

## About TRII DIENSTLEISTUNG SARL

TRII DIENSTLEISTUNG SARL has its HQ in the amazing city Casablanca in Morocco and provides IT-Services and intelligent blockchain solutions. TRII DIENSTLEISTUNG is developing an ecosystem that is seamless and responsive. Some of our services: Blockchain Security, IaaS, SaaS, PaaS, Telecommunication Technologies, IT-Security and many more. **TRII DIENSTLEISTUNG's Blockchain services are there to interconnect multiple services like Blockchain Security, under one ecosystem that is seamless, multi-chain compatible, scalable, secure, fast, responsive, and easy-to-use.**

TRII DIENSTLEISTUNG is built by a Moroccan / German team of UI experts, contributors, engineers. Our Blockchain services team currently consists of 5 core team members, and 2+ casual contributors. **TRII DIENSTLEISTUNG provides manual, static, and automatic smart contract analysis, to ensure that project is checked against known attacks and potential vulnerabilities.**

To learn more, visit <https://trii.sarl>

[Etg 1 APPT 3, 26 Av. Mers Sultan, Casablanca 20006, Maroc](#)

Royaume du Maroc Ministère de la Justice Tribunal de commerce, Casablanca

Registre du Commerce

(N° I.C.E): 002935374000013

Inscriptions Portées au registre analytique N°: 519805

TAX-ID: 519805

SOCIETE A RESPONSABILITE LIMITEE (SARL)