



**Y5 Trader Token**

# **Smart Contract Security Audit Report**

---

**15.08.2022**

**Versatile Finance Audit**

**Helping Businesses Incubate Ideas  
Into Reality**

**[info@versatile.finance](mailto:info@versatile.finance)**

## Summary

**Project Name:** Y5 TRADER

**Contract Address:** 0xc41ab777c7d1d5e04414a14fc6fa4c93766fb1eb

**Client contact:** Y5 TRADER Team

**Blockchain:** Binance smart chain

**Language:** Solidity

**Project website:** <https://www.y-5.finance>

**Buy Tax:** 3%

**Sell Tax:** 3%

**Token supply:** 50,000,000

**Token ticker:** Y5TT

**Decimals:** 18

**Dividend tracker:** 0xe0ac736fcd6442d2fd46896382db9be4dc2c7d48

**Marketing fee receiver:** 0xe6f223c3bb4b38f422f520a9da236be644d01ed5

**Contract deployer address:** 0xe6f223C3bb4B38F422f520a9DA236BE644D01ED5

**Contract's current owner address:** 0xe6f223c3bb4b38f422f520a9da236be644d01ed5

## Background

Versatile Finance was commissioned by Y5 TRADER Team to perform an audit of the smart contract.

<https://bscscan.com/token/0xc41ab777c7d1d5e04414a14fc6fa4c93766fb1eb>

The purpose of this audit was to achieve the following:

- Identify potential security issues with smart contracts
- Formally check the logic behind given smart contracts.

Information in this report should be used for understanding the risk exposure of smart contracts, and as a guide to improving the security posture of smart contracts by remediating the issues that were identified.

## **What is an audit**

A smart contract audit is a comprehensive review process designed to discover logical errors, security vulnerabilities, and optimization opportunities within code. Versatile Finance manages this a step further by verifying economic logic to ensure the stability of smart contracts and highlighting privileged functionality to create a report that is easy to understand for developers and community members.

## **Techniques and Methods**

- The code quality
- Use of best practices
- Implementation of ERC-20 token standards.
- Efficient use of gas.
- Code is safe from re-entrancy and other vulnerabilities.
- Code risk issue analysis and recommendations
- Ownership privileges
- Code documentation and comments match logic and expected behavior.
- Token distribution and calculations are as per the intended behavior mentioned in the whitepaper.

The following techniques, methods, and tools were used to review all the smart contracts.

## **Structural Analysis**

We analyze the design patterns and structure of smart contracts. A thorough check is done to ensure the smart contract is structured in a way that will not have any issues.

## **Static Analysis**

A static Analysis of Smart Contracts is done to identify contract vulnerabilities. In this step, a series of automated tools and manual testings are used to test the security of smart contracts.

## **Code Review / Manual Analysis**

Manual Analysis or review of code is done to identify new vulnerabilities or verify the vulnerabilities found during the static analysis. Contracts is completely manually analyzed line by line, and the logic is checked and compared with what's mentioned in the whitepaper to make sure everything's functioned as intended.

## Gas Consumption

We check the behavior of smart contracts in production. Manual testings are done in DEXs to know how much gas gets consumed and the possibilities of optimization of code to reduce gas consumption.

## Issue Categories

Every issue in this report has been assigned a severity level. There are four levels of severity and each of them has been explained below.

### High severity issues

NO High severity issues found

A high severity issue or vulnerability means that your smart contract can be exploited. Issues on this level are critical to the smart contract's performance or functionality and we recommend these issues be fixed before moving to a live environment.

### Medium-level severity issues

NO Medium severity issues found

The issues marked as medium severity usually arises because of errors and deficiencies in the smart contract code. Issues on this level could potentially bring problems and they can still be fixed. This can put users' funds at risk and has a medium to the high probability of exploitation.

### Low-level severity issues

3 low-level severity issues found

Low-level severity issues can cause minor impact and or are just warnings that can remain unfixed for now. It would be better to fix these issues at some point in the future. These issues have a low probability of occurring or may have a minimal impact.

1. In excluding multiple accounts from the fees function there is any length validation for the accounts array if the user enters around more than 100 wallets at once, this function may fail because of the gas limitation of the block.

```
ftrace | funcSig
function excludeMultipleAccountsFromFees(
    address[] calldata accounts↑,
    bool excluded↑
) public onlyOwner {
    for (uint256 i = 0; i < accounts↑.length; i++) {
        _isExcludedFromFees[accounts↑[i]] = excluded↑;
    }

    emit ExcludeMultipleAccountsFromFees(accounts↑, excluded↑);
}
```

2. Marketing fees and rewards use the same BEP20 token address so inside the swapAndSendFee swapAndDividends tokens are swapping into the same token. Hence, both tokens can swap at once and divide according to the ratio, in the current scenario there will be an additional red candle in the chart, and also a high gas fee.

```
uint256 marketingTokens = contractTokenBalance
    .mul(marketingFee)
    .div(totalFees);
swapAndSendToFee(marketingTokens);

uint256 swapTokens = contractTokenBalance.mul(liquidityFee).div(
    totalFees
);
swapAndLiquify(swapTokens);

uint256 sellTokens = balanceOf(address(this));
swapAndSendDividends(sellTokens);
```

3. If the owner updates the dividend tracker, this function will deploy a totally new tracker and the token will lose all previous investors and their details.

```
ftrace | funcSig
function updateDividendTracker(address newAddress↑) public onlyOwner {
    require(
        newAddress↑ != address(dividendTracker),
        "BABYTOKEN: The dividend tracker already has that address"
    );

    BABYTOKENDividendTracker newDividendTracker = BABYTOKENDividendTracker(
        payable(newAddress↑)
    );

    require(
        newDividendTracker.owner() == address(this),
        "BABYTOKEN: The new dividend tracker must be owned by the BABYTOKEN token contract"
    );

    newDividendTracker.excludeFromDividends(address(newDividendTracker));
    newDividendTracker.excludeFromDividends(address(this));
    newDividendTracker.excludeFromDividends(owner());
    newDividendTracker.excludeFromDividends(address(uniswapV2Router));

    emit UpdateDividendTracker(newAddress↑, address(dividendTracker));

    dividendTracker = newDividendTracker;
}
```

## Informational

NO informational issues found

These are severity four issues that indicate an improvement request, a general question, a cosmetic or documentation error, or a request for information. There is low-to-no impact.

## Owner privileges

The owner can enable/disable anti bot feature

```
ftrace | funcSig
function setEnableAntiBot(bool _enable↑) external onlyOwner {
    enableAntiBot = _enable↑;
}
```

The owner can enable/disable swap point

```
ftrace | funcSig
function setSwapTokensAtAmount(uint256 amount↑) external onlyOwner {
    swapTokensAtAmount = amount↑;
}
```

The owner can update the dividend tracker address

```
ftrace | funcSig
function updateDividendTracker(address newAddress↑) public onlyOwner {
    require(
        newAddress↑ != address(dividendTracker),
        "BABYTOKEN: The dividend tracker already has that address"
    );

    BABYTOKENDividendTracker newDividendTracker = BABYTOKENDividendTracker(
        payable(newAddress↑)
    );

    require(
        newDividendTracker.owner() == address(this),
        "BABYTOKEN: The new dividend tracker must be owned by the BABYTOKEN token contract"
    );

    newDividendTracker.excludeFromDividends(address(newDividendTracker));
    newDividendTracker.excludeFromDividends(address(this));
    newDividendTracker.excludeFromDividends(owner());
    newDividendTracker.excludeFromDividends(address(uniswapV2Router));

    emit UpdateDividendTracker(newAddress↑, address(dividendTracker));

    dividendTracker = newDividendTracker;
}
```



The owner can update the router address

```
ftrace | funcSig
function updateUniswapV2Router(address newAddress↑) public onlyOwner {
    require(
        newAddress↑ != address(uniswapV2Router),
        "BABYTOKEN: The router already has that address"
    );
    emit UpdateUniswapV2Router(newAddress↑, address(uniswapV2Router));
    uniswapV2Router = IUniswapV2Router02(newAddress↑);
    address _uniswapV2Pair = IUniswapV2Factory(uniswapV2Router.factory())
        .createPair(address(this), uniswapV2Router.WETH());
    uniswapV2Pair = _uniswapV2Pair;
}
```

The owner can include/exclude wallets from fees

```
ftrace | funcSig
function excludeFromFees(address account↑, bool excluded↑) public onlyOwner {
    require(
        isExcludedFromFees[account↑] != excluded↑,
        "BABYTOKEN: Account is already the value of 'excluded'"
    );
    isExcludedFromFees[account↑] = excluded↑;
    emit ExcludeFromFees(account↑, excluded↑);
}
```

The owner can change marketing wallet address

```
ftrace | funcSig
function setMarketingWallet(address payable wallet↑) external onlyOwner {
    marketingWalletAddress = wallet↑;
}
```

The owner can change all fees total fees up to 25%

```
ftrace | funcSig
function setTokenRewardsFee(uint256 value↑) external onlyOwner {
    tokenRewardsFee = value↑;
    totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee);
    require(totalFees <= 25, "Total fee is over 25%");
}

ftrace | funcSig
function setLiquidityFee(uint256 value↑) external onlyOwner {
    liquidityFee = value↑;
    totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee);
    require(totalFees <= 25, "Total fee is over 25%");
}

ftrace | funcSig
function setMarketingFee(uint256 value↑) external onlyOwner {
    marketingFee = value↑;
    totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee);
    require(totalFees <= 25, "Total fee is over 25%");
}
```

The owner can add/remove a new pair address

```
ftrace | funcSig
function setAutomatedMarketMakerPair(address pair↑, bool value↑)
    public
    onlyOwner
{
    require(
        pair↑ != uniswapV2Pair,
        "BABYTOKEN: The PancakeSwap pair cannot be removed from automatedMarketMakerPairs"
    );
    _setAutomatedMarketMakerPair(pair↑, value↑);
}
```

The owner can change the maximum gas fee for reward sending

```
ftrace | funcSig
function updateGasForProcessing(uint256 newValue↑) public onlyOwner {
    require(
        newValue↑ >= 200000 && newValue↑ <= 500000,
        "BABYTOKEN: gasForProcessing must be between 200,000 and 500,000"
    );
    require(
        newValue↑ != gasForProcessing,
        "BABYTOKEN: Cannot update gasForProcessing to same value"
    );
    emit GasForProcessingUpdated(newValue↑, gasForProcessing);
    gasForProcessing = newValue↑;
}
```

The owner can change the minimum claim wait

```
ftrace | funcSig
function updateClaimWait(uint256 claimWait↑) external onlyOwner {
    dividendTracker.updateClaimWait(claimWait↑);
}
```

The owner can change the minimum tokens balance to get rewards

```
ftrace | funcSig
function updateMinimumTokenBalanceForDividends(uint256 amount↑)
    external
    onlyOwner
{
    dividendTracker.updateMinimumTokenBalanceForDividends(amount↑);
}
```







## Audit Results





















Vulnerability Category	Status
Arbitrary Jump/Storage Write	pass
BRC20 Token standards	pass
Compiler errors	pass
Latest compiler version	Low issue
Authorization of function call to untrusted contract	pass
Dependence on Predictable Variables	pass
Ether/Token Theft	pass
Gas consumption	Low issue
Safemath features	pass
Fallback usage	pass
Deprecated items	pass
Redundant code	pass
Overriding variables	pass
Flash Loans	pass
Front Running	pass
Improper Events	pass
Improper Authorization Scheme	pass
Integer Over/Underflow	pass




















Business logic issues	Low issue
Oracle issues	pass
Race Conditions	pass
Reentrancy	pass
Signature Issues	pass
Unbounded Loops	pass
Unused Code	pass
Pseudo random number generator (PRNG)	pass
Fake deposit	pass
Centralization of control	pass

**Contracts Description Table**






































Contract	Type	Bases		
L	Function Name	Visibility	M u t a b i l i t y	Modifiers
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 !
IERC20Metadata	Interface	IERC20		
L	name	External !		NO !
L	symbol	External !		NO !
L	decimals	External !		NO !
Context	Implementation			
L	_msgSender	Internal 		
L	_msgData	Internal 		
ERC20	Implementation	Context, IERC20, IERC20Metadata		
L		Public !		NO !
L	name	Public !		NO !
L	symbol	Public !		NO !
L	decimals	Public !		NO !








L	totalSupply	Public !		NO !
L	balanceOf	Public !		NO !
L	transfer	Public !		NO !
L	allowance	Public !		NO !
L	approve	Public !		NO !
L	transferFrom	Public !		NO !
L	increaseAllowance	Public !		NO !
L	decreaseAllowance	Public !		NO !
L	_transfer	Internal 		
L	_mint	Internal 		
L	_burn	Internal 		
L	_approve	Internal 		
L	_beforeTokenTransfer	Internal 		
L	_afterTokenTransfer	Internal 		
Ownable	Implementation	Context		
L		Public !		NO !
L	owner	Public !		NO !
L	renounceOwnership	Public !		onlyOwner
L	transferOwnership	Public !		onlyOwner
























L	_setOwner	Private 		
SafeMath	Library			
L	tryAdd	Internal 		
L	trySub	Internal 		
L	tryMul	Internal 		
L	tryDiv	Internal 		
L	tryMod	Internal 		
L	add	Internal 		
L	sub	Internal 		
L	mul	Internal 		
L	div	Internal 		
L	mod	Internal 		
L	sub	Internal 		
L	div	Internal 		
L	mod	Internal 		
Clones	Library			
L	clone	Internal 		
L	cloneDeterministic	Internal 		































L	predictDeterministicAddress	Internal 		
L	predictDeterministicAddress	Internal 		
IUniswapV2Factory	Interface			
L	feeTo	External 		NO 
L	feeToSetter	External 		NO 
L	getPair	External 		NO 
L	allPairs	External 		NO 
L	allPairsLength	External 		NO 
L	createPair	External 		NO 
L	setFeeTo	External 		NO 
L	setFeeToSetter	External 		NO 
IUniswapV2Router01	Interface			
L	factory	External 		NO 
L	WETH	External 		NO 
L	addLiquidity	External 		NO 
L	addLiquidityETH	External 		NO 
L	removeLiquidity	External 		NO 
L	removeLiquidityETH	External 		NO 





L	removeLiquidityWithPermit	External !		NO !
L	removeLiquidityETHWithPermit	External !		NO !
L	swapExactTokensForTokens	External !		NO !
L	swapTokensForExactTokens	External !		NO !
L	swapExactETHForTokens	External !		NO !
L	swapTokensForExactETH	External !		NO !
L	swapExactTokensForETH	External !		NO !
L	swapETHForExactTokens	External !		NO !
L	quote	External !		NO !
L	getAmountOut	External !		NO !
L	getAmountIn	External !		NO !
L	getAmountsOut	External !		NO !
L	getAmountsIn	External !		NO !
IUniswapV2Router02	Interface	IUniswapV2Router01		
L	removeLiquidityETHSupportingFeeOnTransferTokens	External !		NO !
L	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External !		NO !
L	swapExactTokensForTokensSupportingFeeOnTransferTokens	External !		NO !




L	swapExactETHForTokensSupportingFeeOnTransferTokens	External !		NO !
L	swapExactTokensForETHSupportingFeeOnTransferTokens	External !		NO !
IPinkAntiBot	Interface			
L	setTokenOwner	External !		NO !
L	onPreTransferCheck	External !		NO !
IERC20Upgradeable	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 !
IERC20MetadataUpgradeable	Interface	IERC20Upgradeable		
L	name	External !		NO !
L	symbol	External !		NO !
L	decimals	External !		NO !

Initializable	Implementation			
ContextUpgradeable	Implementation	Initializable		
L	__Context_init	Internal 		initializer
L	__Context_init_unchained	Internal 		initializer
L	_msgSender	Internal 		
L	_msgData	Internal 		
ERC20Upgradeable	Implementation	Initializable, ContextUpgradeable, IERC20Upgradeable, IERC20MetadataUpgr adeable		
L	__ERC20_init	Internal 		initializer
L	__ERC20_init_unchained	Internal 		initializer
L	name	Public 		NO 
L	symbol	Public 		NO 
L	decimals	Public 		NO 
L	totalSupply	Public 		NO 
L	balanceOf	Public 		NO 
L	transfer	Public 		NO 

L	allowance	Public !		NO !
L	approve	Public !		NO !
L	transferFrom	Public !		NO !
L	increaseAllowance	Public !		NO !
L	decreaseAllowance	Public !		NO !
L	_transfer	Internal 		
L	_mint	Internal 		
L	_burn	Internal 		
L	_approve	Internal 		
L	_beforeTokenTransfer	Internal 		
L	_afterTokenTransfer	Internal 		
OwnableUpgradeable	Implementation	Initializable, ContextUpgradeable		
L	__Ownable_init	Internal 		initializer
L	__Ownable_init_unchained	Internal 		initializer
L	owner	Public !		NO !
L	renounceOwnership	Public !		onlyOwner
L	transferOwnership	Public !		onlyOwner
L	_setOwner	Private 		










IUniswapV2Pair	Interface			
L	name	External !		NO !
L	symbol	External !		NO !
L	decimals	External !		NO !
L	totalSupply	External !		NO !
L	balanceOf	External !		NO !
L	allowance	External !		NO !
L	approve	External !		NO !
L	transfer	External !		NO !
L	transferFrom	External !		NO !
L	DOMAIN_SEPARATOR	External !		NO !
L	PERMIT_TYPEHASH	External !		NO !
L	nonces	External !		NO !
L	permit	External !		NO !
L	MINIMUM_LIQUIDITY	External !		NO !
L	factory	External !		NO !
L	token0	External !		NO !
L	token1	External !		NO !
L	getReserves	External !		NO !
















L	price0CumulativeLast	External !		NO !
L	price1CumulativeLast	External !		NO !
L	kLast	External !		NO !
L	mint	External !		NO !
L	burn	External !		NO !
L	swap	External !		NO !
L	skim	External !		NO !
L	sync	External !		NO !
L	initialize	External !		NO !
SafeMathInt	Library			
L	mul	Internal 		
L	div	Internal 		
L	sub	Internal 		
L	add	Internal 		
L	abs	Internal 		
L	toUint256Safe	Internal 		
SafeMathUint	Library			
L	toInt256Safe	Internal 		


















IterableMapping	Library			
L	get	Public !		NO !
L	getIndexOfKey	Public !		NO !
L	getKeyAtIndex	Public !		NO !
L	size	Public !		NO !
L	set	Public !		NO !
L	remove	Public !		NO !
DividendPayingTokenInterface	Interface			
L	dividendOf	External !		NO !
L	withdrawDividend	External !		NO !
DividendPayingTokenOptionalInterface	Interface			
L	withdrawableDividendOf	External !		NO !
L	withdrawnDividendOf	External !		NO !
L	accumulativeDividendOf	External !		NO !





DividendPayingToken	Implementation	ERC20Upgradeable, OwnableUpgradeable, DividendPayingTokenI nterface, DividendPayingToken OptionalInterface		
L	__DividendPayingToken_init	Internal 🔒	⛔	initializer
L	distributeCAKEDividends	Public !	⛔	onlyOwner
L	withdrawDividend	Public !	⛔	NO !
L	_withdrawDividendOfUser	Internal 🔒	⛔	
L	dividendOf	Public !		NO !
L	withdrawableDividendOf	Public !		NO !
L	withdrawnDividendOf	Public !		NO !
L	accumulativeDividendOf	Public !		NO !
L	_transfer	Internal 🔒	⛔	
L	_mint	Internal 🔒	⛔	
L	_burn	Internal 🔒	⛔	
L	_setBalance	Internal 🔒	⛔	
BABYTOKENDividendTracker	Implementation	OwnableUpgradeable, DividendPayingToken		
L	initialize	External !	⛔	initializer
L	_transfer	Internal 🔒		

L	withdrawDividend	Public !		NO !
L	excludeFromDividends	External !		onlyOwner
L	isExcludedFromDividends	Public !		NO !
L	updateClaimWait	External !		onlyOwner
L	updateMinimumTokenBalanceFor Dividends	External !		onlyOwner
L	getLastProcessedIndex	External !		NO !
L	getNumberOfTokenHolders	External !		NO !
L	getAccount	Public !		NO !
L	getAccountAtIndex	Public !		NO !
L	canAutoClaim	Private 		
L	setBalance	External !		onlyOwner
L	process	Public !		NO !
L	processAccount	Public !		onlyOwner
BaseToken	Implementation			
AntiBotBABYTOKEN	Implementation	ERC20, Ownable, BaseToken		
L		Public !		ERC20
L	setEnabledAntiBot	External !		onlyOwner

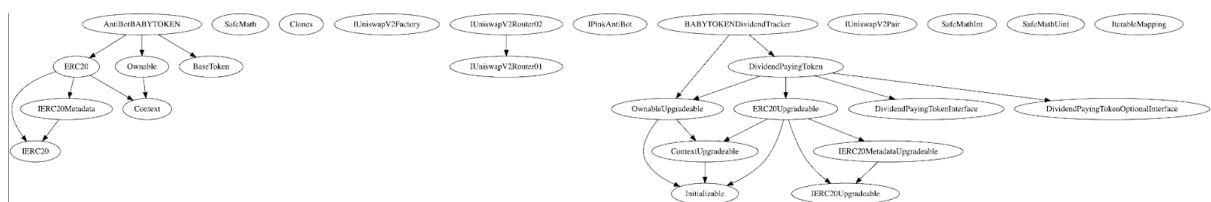
L		External !		NO !
L	setSwapTokensAtAmount	External !		onlyOwner
L	updateDividendTracker	Public !		onlyOwner
L	updateUniswapV2Router	Public !		onlyOwner
L	excludeFromFees	Public !		onlyOwner
L	excludeMultipleAccountsFromFees	Public !		onlyOwner
L	setMarketingWallet	External !		onlyOwner
L	setTokenRewardsFee	External !		onlyOwner
L	setLiquiditFee	External !		onlyOwner
L	setMarketingFee	External !		onlyOwner
L	setAutomatedMarketMakerPair	Public !		onlyOwner
L	_setAutomatedMarketMakerPair	Private 🗝		
L	updateGasForProcessing	Public !		onlyOwner
L	updateClaimWait	External !		onlyOwner
L	getClaimWait	External !		NO !
L	updateMinimumTokenBalanceFor Dividends	External !		onlyOwner
L	getMinimumTokenBalanceForDivi dends	External !		NO !
L	getTotalDividendsDistributed	External !		NO !
L	isExcludedFromFees	Public !		NO !

L	withdrawableDividendOf	Public !		NO !
L	dividendTokenBalanceOf	Public !		NO !
L	excludeFromDividends	External !		onlyOwner
L	isExcludedFromDividends	Public !		NO !
L	getAccountDividendsInfo	External !		NO !
L	getAccountDividendsInfoAtIndex	External !		NO !
L	processDividendTracker	External !		NO !
L	claim	External !		NO !
L	getLastProcessedIndex	External !		NO !
L	getNumberOfDividendTokenHolders	External !		NO !
L	_transfer	Internal 		
L	swapAndSendToFee	Private 		
L	swapAndLiquify	Private 		
L	swapTokensForEth	Private 		
L	swapTokensForCake	Private 		
L	addLiquidity	Private 		
L	swapAndSendDividends	Private 		

### Legend

Symbol	Meaning
	Function can modify state
	Function is payable

### Inheritance chart



## **Audit conclusion**

Versatile Finance team has performed in-depth testings, line by line manual code review, and automated audit of the smart contract. The smart contract was analyzed mainly for common smart contract vulnerabilities, exploits, manipulations, and hacks. According to the smart contract audit.

Smart contract functional Status: **Pass**

Number of risk issues: **3 low-level issues**

Solidity code functional issue level: **Pass**

Number of owner privileges: **11**

Centralization risk correlated to the active owner: **Low**

Smart contract active ownership: **YES**

## Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as of the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the disclaimer below – please make sure to read it in full.

DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and Versatile Finance and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers, and other representatives) (Versatile Finance) owe no duty of care towards you or any other person, nor does Versatile Finance make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties, or other terms of any kind except as set out in this disclaimer, and Versatile Finance hereby excludes all representations, warranties, conditions, and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, Versatile Finance hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against Versatile Finance, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of the use of this report, and any reliance on this report. 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.