CERTIK

Preliminary Comments

TelmoCoin

Sept 11th, 2021

Table of Contents

PRELIMINARY COMMENTS

Summary

This report has been prepared for TelmoCoin to discover issues and vulnerabilities in the source code of the TelmoCoin project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Static Analysis and Manual Review techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross-referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

Additionally, this audit is based on a premise that all external smart contracts are safely implemented.

The security assessment resulted in findings that ranged from critical to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspective:

- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases;
- Provide more comments per each function for readability, especially contracts that are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

CERTIK PRELIMINARY COMMENTS

Overview

Project Summary

Project Name	TelmoCoin
Platform	Ethereum
Language	Solidity
Codebase	https://github.com/rcarrion14/TLM
Commit	7ef1930712234251b50998c3052da7a43142fdb1

Audit Summary

Delivery Date	Sept 11, 2021
Audit Methodology	Static Analysis, Manual Review
Key Components	

Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Resolved
● Critical	0	0	0	0	0	0
● Major	2	2	0	0	0	0
● Medium	0	0	0	0	0	0
● Minor	3	3	0	0	0	0
● Informational	4	4	0	0	0	0
● Discussion	4	4	0	0	0	0

PRELIMINARY COMMENTS CERTIK

Audit Scope

ID	File	SHA256 Checksum
ERC	ERC20.sol	5e239077babc72ca96141aa3e897b59285612c5a6410c8b1e4504b5fab81fa4d
TLM	TLM.sol	820cb2dd244542ca3bad4f7ec9db50128426a75ea5649c70b45806ed9dabafa4
TLT	TLMbase.sol	45a8bb840abe9f5b8a1594a324da9638438b03ce12e6c9328a8edda772711d7e
TVT	TiendaVault.sol	004fbd91f9ebcc1529e4a2b0319925f1384d2a8814feb93291336df97caa716e
VTC	Vaulteable.sol	d45031bea5b9d5be471cef6ec834cc51f37b766851ea65bf28063f19003c1c65

CERTIK PRELIMINARY COMMENTS

Understandings

Overview

TelmoCoin is a project that allows users to swap and buy the `TLM` token through the `TiendaVault` contract.

Exchange Rate

`TLM` token is backed with the `backingCoin` token that belongs to the `TiendaVault` contract. By calling the `backingCreation()` function, the `owner` deposits a fixed amount of `backingCoin` and the corresponding amount, which is calculated by formula `backingCoin amount * exchange rate`, of `TLM` token is minted to `owner`.

The exchange rate between those two tokens is initialized when the `TiendaVault` contract is deployed and never changed unless the `owner` depositing more `backingCoin` into the contract to lower it and then the value of `TLM` is increased. Because after a swap the `TLM` token is not burned, the two key state variables, `tlmWithBacking` and `tlmWithOutBacking`, are used to keep the exchange rate unaltered when the amount of `backingCoin`, which belong to `TiendaVault` contract, is changed.

Swap

By calling `backingWithdraw()` function, users can deposit `TLM` and retrieve `backingCoin` . If the caller is not the `adminWallet`, 0.5% transaction commission in `backingCoin` is transferred to the `adminWallet`.

By calling the `buy()` function, users can use `backingCoin` to buy `TLM`. If the `tlmWithoutBacking` equals zero, the `backingCoin` is directly transferred to `adminWallet` because no more `TLM` needs to be backed with `backingCoin`. If the `TLM` token, which is not backed with the `bBTC` token, is not enough for the purchase amount, the excess part of `backingCoin` is transferred to the `adminWallet`, not the `TiendaVault` contract, to keep `TLM` and `backingCoin` exchange rate unchanged.

Privileged Functions

The contract contains the following privileged functions that are restricted by modifiers. We group these functions below:

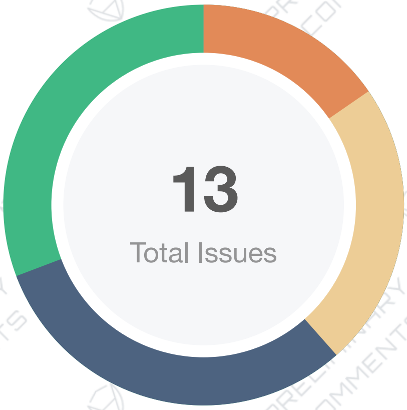
The `onlyVault` modifier used to manage the `TLM` token issue:

- `mint()` in `TLM.sol`
- `burn()` in `TLM.sol`

The `onlyOwner` modifier to initialize `TiendaVault` contract, modify exchange rate, and transfer to or out `TLM` token:

- `backingCreation()` in `TiendaVault.sol`
- `profitPayment()` in `TiendaVault.sol`
- `upForSale()` in `TiendaVault.sol`
- `outOfSale()` in `TiendaVault.sol`

Findings



🟥	**Critical**	**0** (0.00%)
🟧	**Major**	**2** (15.38%)
🟨	**Medium**	**0** (0.00%)
🟨	**Minor**	**3** (23.08%)
🟦	**Informational**	**4** (30.77%)
🟩	**Discussion**	**4** (30.77%)

ID	Title	Category	Severity	Status
GLOBAL-01	Unlocked Compiler Version	Language Specific	● Informational	Pending
GLOBAL-02	Centralization Risk	**Centralization / Privilege**	● **Major**	Pending
TLM-01	Inconsistency With The Documentation	Logical Issue	● Discussion	Pending
TVT-01	Too Many Digits	Coding Style	● Informational	Pending
TVT-02	Variable Declare As Immutable	Gas Optimization	● Minor	Pending
TVT-03	Missing Input Validation	Volatile Code	● Minor	Pending
TVT-04	Redundant Statements	Volatile Code	● Informational	Pending
TVT-05	Incorrect Boolean Expression	Logical Issue	● Discussion	Pending
TVT-06	Inconsistency With The Documentation	Logical Issue	● Discussion	Pending
TVT-07	Function Is Executed Only Once	Logical Issue	● Major	Pending
TVT-08	Check Effect Interaction Pattern Violated	Logical Issue	● Minor	Pending
TVT-09	Unclear State Variable	Logical Issue	● Discussion	Pending
TVT-10	Variable Could Be Declared As `constant`	Gas Optimization	● Informational	Pending

GLOBAL-01 | Unlocked Compiler Version

Category	Severity	Location	Status
Language Specific	● Informational	Global	Pending

Description

The contract has an unlocked compiler version. An unlocked compiler version in the contract's source code permits the user to compile it at or above a particular version. This, in turn, leads to differences in the generated bytecode between compilations due to differing compiler version numbers. This can lead to ambiguity when debugging as compiler-specific bugs may occur in the codebase that would be difficult to identify over a span of multiple compiler versions rather than a specific one.

Recommendation

We advise the compiler version is alternatively locked at the lowest version possible that the contract can be compiled at. For example, for version `v0.8.0`, the contract should contain the following line:

```
pragma solidity 0.8.0;
```

PRELIMINARY COMMENTS CERTIK

GLOBAL-02 | Centralization Risk

Category	Severity	Location	Status
Centralization / Privilege	● Major	Global	Pending

Description

The role `owner` has the authority over the listed functions:

- `backingCreation()` in `TiendaVault.sol`
- `profitPayment()` in `TiendaVault.sol`
- `upForSale()` in `TiendaVault.sol`
- `outOfSale()` in `TiendaVault.sol`

The role `vault` has the authority over the listed functions:

- `mint()` in `TLM.sol`, mint any amount of token to any account.
- `burn()` in `TLM.sol`, burn any amount token belong to any account.

Any compromise to the key role account may allow a potential hacker to take advantage of this and execute malicious acts.

Recommendation

We advise the client to carefully manage the key role account's private key to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol to be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., Multisignature wallets.

Indicatively, here are some feasible suggestions that would also mitigate the potential risk at the different levels in terms of short-term and long-term scenarios:

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key;
- Introduction of a DAO/governance/voting module to increase transparency and user involvement.

PRELIMINARY COMMENTS CERTIK

TLM-01 | Inconsistency With The Documentation

Category	Severity	Location	Status
Logical Issue	● Discussion	TLM.sol: 15	ⓘ Pending

Description

According to the documentation, `TLM` would be created with a fixed supply. We don't see the supply restriction when the `TLM` token is minted. As such who will act in the role of `vault` after the `TLM` contract is deployed on the blockchain?

Recommendation

Please provide more information regarding the design concept behind the `TLM` token issue.

TVT-01 | Too Many Digits

Category	Severity	Location	Status
Coding Style	● Informational	TiendaVault.sol: 15	Pending

Description

Literals with too many digits are difficult to read and review.

Recommendation

We recommend modifying as below:

```
15  uint256 ceros = 10**18;
```

PRELIMINARY COMMENTS CERTIK

TVT-02 | Variable Declare As Immutable

Category	Severity	Location	Status
Gas Optimization	● Minor	TiendaVault.sol: 12~13, 19	ⓘ Pending

Description

The linked variables should be declared as `immutable`. Immutable state variables can be assigned during contract creation but will remain constant throughout the lifetime of a deployed contract. A big advantage of immutable variables is that reading them is significantly cheaper than reading from regular state variables since will not be stored in storage. Still, values will directly insert the values into the runtime code.

Recommendation

We advise using an immutable state variable for the linked state variables.

PRELIMINARY COMMENTS CERTIK

TVT-03 | Missing Input Validation

Category	Severity	Location	Status
Volatile Code	● Minor	TiendaVault.sol: 26~32	Pending

Description

The given input is missing the check for non-zero address.

Recommendation

We advise adding the check for the passed-in values to prevent unexpected errors as below:

```
26  constructor (address tlmAddress, address backingCoinAddress, uint256 setRate, address setAdminWallet) {
27      require(address(0) != tlmAddress, "TiendaVault: set tlm to the zero address");
28      require(address(0) != backingCoinAddress, "TiendaVault: set backingCoin to the zero address");
29      require(setRate > 0, "TiendaVault: set rate to be zero");
30      require(address(0) != setAdminWallet, "TiendaVault: set adminWallet to the zero address");
31      tlm = TLM(tlmAddress);
32      backingCoin = ERC20(backingCoinAddress);
33      rate = setRate * ceros;
34      adminWallet = setAdminWallet;
35  }
```

PRELIMINARY COMMENTS CERTIK

PRELIMINARY COMMENTS

TVT-04 | Redundant Statements

Category	Severity	Location	Status
Volatile Code	● Informational	TiendaVault.sol: 64~92	ⓘ Pending

Description

The linked statements are duplicated.

Recommendation

We advise removing one of them to better prepare the code for production environments.

```
function backingWithdraw(uint256 amount) public {

    require(tlm.balanceOf(msg.sender)>= amount, "Cliente no tiene suficiente TLM");

    tlmWithBacking = tlmWithBacking - amount;
    tlmWithOutBacking = tlmWithOutBacking + amount;

    if (msg.sender== adminWallet){
        backingCoin.transfer(msg.sender, ((amount *ceros/ rate)));
    }else{
        backingCoin.transfer(msg.sender, (amount *ceros/ rate)*9950/10000);
        backingCoin.transfer(adminWallet, (amount *ceros/ rate)*50/10000);  //  la diferencia anterior se transifere al dueño
    }

    tlm.transferFrom(msg.sender, address(this), amount);

    emit backingWithdrawn(amount / rate, amount);

}
```

TVT-05 | Incorrect Boolean Expression

Category	Severity	Location	Status
Logical Issue	● Discussion	TiendaVault.sol: 126	ⓘ Pending

Description

Even if the linked boolean expression is true, the subtraction overflow may occur when the statement `tlmWithOutBacking = tlmWithOutBacking - amount * rate /ceros;`, at line 128, is executed.

Recommendation

Please provide more information about the design concept regarding the `buy()` function.

PRELIMINARY COMMENTS

TVT-06 | Inconsistency With The Documentation

Category	Severity	Location	Status
Logical Issue	● Discussion	TiendaVault.sol: 123	ⓘ Pending

Description

When users utilize bBTC to buy TLM in one transaction, 99.5% amount of TLM would be transferred to users and 0.5% amount is still kept in the TiendaVault contract. Is the 0.5% amount designated as transaction commission? Why isn't it transferred to the adminWallet? And does the adminWallet only receive the bBTC token in this project?

The documentation doesn't mention the 0.5% amount of the purchased TLM token.

Recommendation

Please provide more information about the design concept regarding the buy() function.

PRELIMINARY COMMENTS

TVT-07 | Function Is Executed Only Once

Category	Severity	Location	Status
Logical Issue	● Major	TiendaVault.sol: 37~45	ⓘ Pending

Description

According to the documentation, the total supply of the `TLM` token is fixed. However, the linked function allows `owner` to mint `TLM` token without restriction. And the state variables `tlmWithOutBacking` and `tlmWithBacking` would be overwritten.

Recommendation

We advise adding the `initializer` modifier, which is linked by the following GitHub url, to the `backingCreation()` function.

https://github.com/OpenZeppelin/openzeppelin-contracts/blob/v4.3.1/contracts/proxy/utils/Initializable.sol

PRELIMINARY COMMENTS

TVT-08 | Check Effect Interaction Pattern Violated

Category	Severity	Location	Status
Logical Issue	● Minor	TiendaVault.sol: 64~92	ⓘ Pending

Description

The order of external call/transfer and storage manipulation must follow the check-effect-interaction pattern.

Recommendation

We advise checking if storage manipulation is before the external call/transfer operation refer to the following codes:

```
  64 function backingWithdraw(uint256 amount) public {
  65
  66     require(tlm.balanceOf(msg.sender)>= amount, "Cliente no tiene suficiente TLM");
  67
  68     tlmWithBacking = tlmWithBacking - amount;
  69     tlmWithOutBacking = tlmWithOutBacking + amount;
  70
  71     if (msg.sender== adminWallet){
  72         backingCoin.transfer(msg.sender, ((amount *ceros/ rate)));
  73     }else{
  74         backingCoin.transfer(msg.sender, (amount *ceros/ rate)*9950/10000);
  75         backingCoin.transfer(adminWallet, (amount *ceros/ rate)*50/10000);  //  la
diferencia anterior se transifere al dueño
  76     }
  77
  78     tlm.transferFrom(msg.sender, address(this), amount);
  79
  80     emit backingWithdrawn(amount / rate, amount);
  81
  82 }
```

Reference: check-effect-interactions

PRELIMINARY COMMENTS

TVT-09 | Unclear State Variable

Category	Severity	Location	Status
Logical Issue	● Discussion	TiendaVault.sol: 41~42	ⓘ Pending

Description

According to the linked statements, we can infer that the sum of `tlmWithBacking` and `tlmWithOutBacking` equals the total supply of the `TLM` token. Is the aforementioned inference valid during the lifecycle of this project?

Recommendation

Please provide more information about the design concept regarding the linked two state variables.

CERTIK PRELIMINARY COMMENTS

TVT-10 | Variable Could Be Declared As `constant`

Category	Severity	Location	Status
Gas Optimization	● Informational	TiendaVault.sol: 15	ⓘ Pending

Description

Variable `ceros` could be declared as `constant` since these state variables are never to be changed.

Recommendation

We advise that those state variables should be declared `constant` to save gas.

PRELIMINARY COMMENTS

Appendix

Finding Categories

Centralization / Privilege

Centralization / Privilege findings refer to either feature logic or implementation of components that act against the nature of decentralization, such as explicit ownership or specialized access roles in combination with a mechanism to relocate funds.

Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimal EVM opcodes resulting in a reduction on the total gas cost of a transaction.

Logical Issue

Logical Issue findings detail a fault in the logic of the linked code, such as an incorrect notion on how block.timestamp works.

Volatile Code

Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that may result in a vulnerability.

Language Specific

Language Specific findings are issues that would only arise within Solidity, i.e. incorrect usage of private or delete.

Coding Style

Coding Style findings usually do not affect the generated byte-code but rather comment on how to make the codebase more legible and, as a result, easily maintainable.

Checksum Calculation Method

The "Checksum" field in the "Audit Scope" section is calculated as the SHA-256 (Secure Hash Algorithm 2 with digest size of 256 bits) digest of the content of each file hosted in the listed source repository under the specified commit.

The result is hexadecimal encoded and is the same as the output of the Linux "sha256sum" command against the target file.

Disclaimer

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to you ("Customer" or the "Company") in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes, nor may copies be delivered to any other person other than the Company, without CertiK's prior written consent in each instance.

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts CertiK to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. CertiK's position is that each company and individual are responsible for their own due diligence and continuous security. CertiK's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

The assessment services provided by CertiK is subject to dependencies and under continuing development. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third-parties.

ALL SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED "AS IS" AND "AS

AVAILABLE" AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, CERTIK HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, CERTIK SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM COURSE OF DEALING, USAGE, OR TRADE PRACTICE. WITHOUT LIMITING THE FOREGOING, CERTIK MAKES NO WARRANTY OF ANY KIND THAT THE SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET CUSTOMER'S OR ANY OTHER PERSON'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULT, BE COMPATIBLE OR WORK WITH ANY SOFTWARE, SYSTEM, OR OTHER SERVICES, OR BE SECURE, ACCURATE, COMPLETE, FREE OF HARMFUL CODE, OR ERROR-FREE. WITHOUT LIMITATION TO THE FOREGOING, CERTIK PROVIDES NO WARRANTY OR UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KIND THAT THE SERVICE WILL MEET CUSTOMER'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULTS, BE COMPATIBLE OR WORK WITH ANY OTHER SOFTWARE, APPLICATIONS, SYSTEMS OR SERVICES, OPERATE WITHOUT INTERRUPTION, MEET ANY PERFORMANCE OR RELIABILITY STANDARDS OR BE ERROR FREE OR THAT ANY ERRORS OR DEFECTS CAN OR WILL BE CORRECTED.

WITHOUT LIMITING THE FOREGOING, NEITHER CERTIK NOR ANY OF CERTIK'S AGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND, EXPRESS OR IMPLIED AS TO THE ACCURACY, RELIABILITY, OR CURRENCY OF ANY INFORMATION OR CONTENT PROVIDED THROUGH THE SERVICE. CERTIK WILL ASSUME NO LIABILITY OR RESPONSIBILITY FOR (I) ANY ERRORS, MISTAKES, OR INACCURACIES OF CONTENT AND MATERIALS OR FOR ANY LOSS OR DAMAGE OF ANY KIND INCURRED AS A RESULT OF THE USE OF ANY CONTENT, OR (II) ANY PERSONAL INJURY OR PROPERTY DAMAGE, OF ANY NATURE WHATSOEVER, RESULTING FROM CUSTOMER'S ACCESS TO OR USE OF THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS.

ALL THIRD-PARTY MATERIALS ARE PROVIDED "AS IS" AND ANY REPRESENTATION OR WARRANTY OF OR CONCERNING ANY THIRD-PARTY MATERIALS IS STRICTLY BETWEEN CUSTOMER AND THE THIRD-PARTY OWNER OR DISTRIBUTOR OF THE THIRD-PARTY MATERIALS.

THE SERVICES, ASSESSMENT REPORT, AND ANY OTHER MATERIALS HEREUNDER ARE SOLELY PROVIDED TO CUSTOMER AND MAY NOT BE RELIED ON BY ANY OTHER PERSON OR FOR ANY PURPOSE NOT SPECIFICALLY IDENTIFIED IN THIS AGREEMENT, NOR MAY COPIES BE DELIVERED TO, ANY OTHER PERSON WITHOUT CERTIK'S PRIOR WRITTEN CONSENT IN EACH INSTANCE.

NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING

MATERIALS AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS.

THE REPRESENTATIONS AND WARRANTIES OF CERTIK CONTAINED IN THIS AGREEMENT ARE SOLELY FOR THE BENEFIT OF CUSTOMER. ACCORDINGLY, NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH REPRESENTATIONS AND WARRANTIES AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH REPRESENTATIONS OR WARRANTIES OR ANY MATTER SUBJECT TO OR RESULTING IN INDEMNIFICATION UNDER THIS AGREEMENT OR OTHERWISE.

FOR AVOIDANCE OF DOUBT, THE SERVICES, INCLUDING ANY ASSOCIATED ASSESSMENT REPORTS OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

About

Founded in 2017 by leading academics in the field of Computer Science from both Yale and Columbia University, CertiK is a leading blockchain security company that serves to verify the security and correctness of smart contracts and blockchain-based protocols. Through the utilization of our world-class technical expertise, alongside our proprietary, innovative tech, we're able to support the success of our clients with best-in-class security, all whilst realizing our overarching vision; provable trust for all throughout all facets of blockchain.

PRELIMINARY COMMENTS