Security Assessment

POXO

Jun 22nd, 2022

Table of Contents

Summary

Overview

- Project Summary
- Audit Summary
- Vulnerability Summary
- Audit Scope

Findings

- GLOBAL-01 : Centralization Related Risks
- GLOBAL-02 : Centralization Related Risks
- CON-01 : Unnecessary usage of Safemath library
- SIC-01 : The defined ICO opening time has already passed
- SIC-02 : Incorrect calculation
- SIC-03 : Incorrect calculation of current stage remaining toke amount
- SIC-04 : Potentially incorrect signature scheme
- SIC-05 : Usage of `transfer()` for sending Ether
- SIC-06 : Loss of precision for rate
- SIC-07 : Unnecessary condition checking
- STB-01 : Initial Token Distribution
- TVB-01 : Lack of input validation
- TVB-02 : Unnecessary receive() and fallback() functions
- TVB-03 : Public functions should be called without `this.` when being called internally
- TVB-04 : Internal function names should start with `_`

Appendix

Disclaimer

About

Summary

This report has been prepared for POXO to discover issues and vulnerabilities in the source code of the POXO 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.

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.

Overview

Project Summary

Project Name	POXO
Description	Sparkso ERC20 ICO
Platform	Ethereum
Language	Solidity
Codebase	https://github.com/vachmara/sparkso-token/tree/f09dc212f2308712fbd9213e2e1206062a9dfbc1/contracts
Commit	f09dc212f2308712fbd9213e2e1206062a9dfbc1

Audit Summary

Delivery Date	Jun 22, 2022 UTC
Audit Methodology	Static Analysis, Manual Review

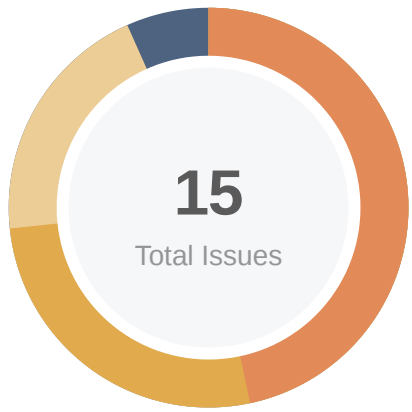
Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Mitigated	Partially Resolved	Resolved
Critical	0	0	0	0	0	0	0
Major	7	0	0	3	0	0	4
Medium	4	0	0	0	0	0	4
Minor	3	0	0	0	0	0	3
Optimization	0	0	0	0	0	0	0
Informational	1	0	0	0	0	0	1
Discussion	0	0	0	0	0	0	0

Audit Scope

ID	File	SHA256 Checksum
TVB	TokenVesting.sol	24c70f7fa06c9803f35baf653952bad43a0ad2605dafe896b7d19538bef36c24
STB	SparksoToken.sol	1d9e732a14854ab82dddf2e608c890d159d3801d333aa8002072b4a81baaadbf
SIC	SparksoICO.sol	c2e46355a7631c704665e4d3fb2291cd432c0e4fc8bb99633afcaab1d283a617

Findings



Critical	**0** (0.00%)
Major	**7** (46.67%)
Medium	**4** (26.67%)
Minor	**3** (20.00%)
Informational	**1** (6.67%)
Discussion	**0** (0.00%)

ID	Title	Category	Severity	Status
GLOBAL-01	Centralization Related Risks	**Centralization / Privilege**	**Major**	Acknowledged
GLOBAL-02	Centralization Related Risks	**Centralization / Privilege**	**Major**	Acknowledged
CON-01	Unnecessary Usage Of Safemath Library	Mathematical Operations, Gas Optimization	Minor	Resolved
SIC-01	The Defined ICO Opening Time Has Already Passed	Volatile Code	Major	Resolved
SIC-02	Incorrect Calculation	Mathematical Operations	Major	Resolved
SIC-03	Incorrect Calculation Of Current Stage Remaining Toke Amount	Mathematical Operations	Major	Resolved
SIC-04	Potentially Incorrect Signature Scheme	Logical Issue	Major	Resolved
SIC-05	Usage Of `transfer()` For Sending Ether	Volatile Code	Medium	Resolved
SIC-06	Loss Of Precision For Rate	Mathematical Operations	Medium	Resolved
SIC-07	Unnecessary Condition Checking	Logical Issue	Minor	Resolved

ID	Title	Category	Severity	Status
STB-01	Initial Token Distribution	**Centralization / Privilege**	**Major**	Acknowledged
TVB-01	Lack Of Input Validation	Logical Issue	Medium	Resolved
TVB-02	Unnecessary Receive() And Fallback() Functions	Logical Issue	Medium	Resolved
TVB-03	Public Functions Should Be Called Without `this.` When Being Called Internally	Gas Optimization	Minor	Resolved
TVB-04	Internal Function Names Should Start With `_`	Coding Style	Informational	Resolved

GLOBAL-01 | Centralization Related Risks

Category	Severity	Location	Status
Centralization / Privilege	**Major**		Acknowledged

Description

In the contract `[TokenVesting]`, the role `[owner]` has authority over the following functions:

- revoke()
- withdraw()
- release()

Any compromise to the `[owner]` account may allow a hacker to take advantage of this authority and make the contract malfunction, steal tokens from the contract.

Recommendation

The risk describes the current project design and potentially makes iterations to improve in the security operation and level of decentralization, which in most cases cannot be resolved entirely at the present stage. We advise the client to carefully manage the privileged account's private key to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., multi-signature wallets.

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

Short Term:

Timelock and Multi sign (⅔, ⅗) combination *mitigate* by delaying the sensitive operation and avoiding a single point of key management failure.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key compromised;
  AND
- A medium/blog link for sharing the timelock contract and multi-signers addresses information with the public audience.

Long Term:

Timelock and DAO, the combination, *mitigate* by applying decentralization and transparency.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Introduction of a DAO/governance/voting module to increase transparency and user involvement;
  AND
- A medium/blog link for sharing the timelock contract, multi-signers addresses, and DAO information with the public audience.

Permanent:

Renouncing the ownership or removing the function can be considered *fully resolved*.

- Renounce the ownership and never claim back the privileged roles;
  OR
- Remove the risky functionality.

Noted: Recommend considering the long-term solution or the permanent solution. The project team shall make a decision based on the current state of their project, timeline, and project resources.

GLOBAL-02 | Centralization Related Risks

Category	Severity	Location	Status
Centralization / Privilege	**Major**		Acknowledged

Description

In the contract `[SparksoICO]`, the role `[owner]` has authority over the following functions:

- delayICO()
- updateICO()

Any compromise to the `[owner]` account may allow a hacker to take advantage of this authority and make the contract malfunction.

Recommendation

The risk describes the current project design and potentially makes iterations to improve in the security operation and level of decentralization, which in most cases cannot be resolved entirely at the present stage. We advise the client to carefully manage the privileged account's private key to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., multi-signature wallets.

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

Short Term:

Timelock and Multi sign (⅔, ⅗) combination *mitigate* by delaying the sensitive operation and avoiding a single point of key management failure.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key compromised;
  AND
- A medium/blog link for sharing the timelock contract and multi-signers addresses information with the public audience.

Long Term:

Timelock and DAO, the combination, *mitigate* by applying decentralization and transparency.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Introduction of a DAO/governance/voting module to increase transparency and user involvement;
  AND
- A medium/blog link for sharing the timelock contract, multi-signers addresses, and DAO information with the public audience.

Permanent:

Renouncing the ownership or removing the function can be considered *fully resolved*.

- Renounce the ownership and never claim back the privileged roles;
  OR
- Remove the risky functionality.

Noted: Recommend considering the long-term solution or the permanent solution. The project team shall make a decision based on the current state of their project, timeline, and project resources.

CON-01 | Unnecessary Usage Of Safemath Library

Category	Severity	Location	Status
Mathematical Operations, Gas Optimization	Minor	SparksoICO.sol: 15~17; TokenVesting.sol: 16	Resolved

Description

For Solidity version 0.8.x, Safemath library is no longer needed because native math operations are already safe and cheaper.

Recommendation

We advise the client to remove the usage of Safemath library.

Alleviation

Fixed in commit ea8c2a97cd890c311697cacac24043ae60d99e9d

SIC-01 | The Defined ICO Opening Time Has Already Passed

Category	Severity	Location	Status
Volatile Code	Major	SparksoICO.sol: 197	Resolved

Description

The defined ICO opening time `5th march 2022` has already passed.

Recommendation

We recommend using a new ICO opening time.

Alleviation

Fixed in commit 7b22f522c9dbfbe62c22f56b5c1ddaf888d456ff

SIC-02 | Incorrect Calculation

Category	Severity	Location	Status
Mathematical Operations	Major	SparksoICO.sol: 473	Resolved

Description

The identified calculation is incorrect and unnecessary because both ethers and Sparkso token have the same 18 decimals. `1 wei ether = rate * 1 wei Sparkso`.

Recommendation

We recommend removing the identified calculation.

Alleviation

Fixed in commit ea8c2a97cd890c311697cacac24043ae60d99e9d

SIC-03 | Incorrect Calculation Of Current Stage Remaining Toke Amount

Category	Severity	Location	Status
Mathematical Operations	Major	SparksoICO.sol: 390~394	Resolved

Description

The calculation `TOKENS_ALLOCATED[_currentStage] - (this.getVestingSchedulesTotalAmount() * 10**18)` is incorrect and will underflow. The correct steps to calculate current stage remaining toke amount is:

1. `currentStageVestingTokens = vestingSchedulesTotalAmount - (sum of allocated tokens in previous stages)`
2. `currentStageTokensRemaining = (current stage allocated tokens) - currentStageVestingTokens`

Recommendation

We recommend using the correct steps to calculate `currentStageTokensRemaining`.

Alleviation

Fixed in commit ea8c2a97cd890c311697cacac24043ae60d99e9d

SIC-04 | Potentially Incorrect Signature Scheme

Category	Severity	Location	Status
Logical Issue	Major	SparksoICO.sol: 276	Resolved

Description

Normally a signature can only be used once. But first, a signature is not bound to specific blockchain and specific contract address. So if the contract is intended to be deployed on multiple blockchains, a signature can be replayed on multiple blockchains. Second, a signature can be used to buy tokens multiple times. To defend against signature replay attack, the proper way should be like https://github.com/OpenZeppelin/openzeppelin-contracts/blob/v4.4.2/contracts/token/ERC20/extensions/draft-ERC20Permit.sol

Recommendation

We recommend using EIP712 and nonce/deadline for the signature scheme.

Alleviation

Fixed in commit ea8c2a97cd890c311697cacac24043ae60d99e9d

SIC-05 | Usage Of `transfer()` For Sending Ether

Category	Severity	Location	Status
Volatile Code	Medium	SparksoICO.sol: 347	Resolved

Description

After EIP-1884 was included in the Istanbul hard fork, it is not recommended to use `.transfer()` or `.send()` for transferring ether as these functions have a hard-coded value for gas costs making them obsolete as they are forwarding a fixed amount of gas, specifically `2300`. This can cause issues in case destination address is contract instead of EOA.

Recommendation

We recommend using the `sendValue()` function in library `Address` from OpenZeppelin. See https://github.com/OpenZeppelin/openzeppelin-contracts/blob/v4.4.2/contracts/utils/Address.sol

Alleviation

Fixed in commit ea8c2a97cd890c311697cacac24043ae60d99e9d

SIC-06 | Loss Of Precision For Rate

Category	Severity	Location	Status
Mathematical Operations	Medium	SparksoICO.sol: 53	Resolved

Description

Normally price information like token exchange rate is NOT integer. Using integer to represent rate will lose precision and make token amount calculation inaccurate. For example, for stage 3, the actual rate = 49245000/1708 = 28831.967. But since rate is an integer, the rate variable value in Solidity is 28831, and 1708*28831 = 49243348 < 49245000.

Recommendation

We recommend using numerator/denominator to represent ether/Sparkso rate.

Alleviation

Fixed in commit e9b2ab2e53e9f02273f9a693c6c7957c8ee41ea0

SIC-07 | Unnecessary Condition Checking

Category	Severity	Location	Status
Logical Issue	Minor	SparksoICO.sol: 331~334, 384~387	Resolved

Description

uint256 type value is always `>= 0`.

Recommendation

We recommend removing the identified condition checking.

Alleviation

Fixed in commit ea8c2a97cd890c311697cacac24043ae60d99e9d

STB-01 | Initial Token Distribution

Category	Severity	Location	Status
Centralization / Privilege	**Major**	SparksoToken.sol: 29	Acknowledged

Description

All of the [Sparkso] tokens are sent to the `wallet_` address when deploying the contract. This could be a centralization risk as the owner of `wallet_` address can distribute [Sparkso] tokens without obtaining the consensus of the community.

Recommendation

We recommend the team to be transparent regarding the initial token distribution process, and the team shall make enough efforts to restrict the access of the private key.

TVB-01 | Lack Of Input Validation

Category	Severity	Location	Status
Logical Issue	Medium	TokenVesting.sol: 284~285	Resolved

Description

The parameter `_cliff` must be less than `_duration`. Otherwise users can not withdraw vested tokens even if vesting duration has ended.

Recommendation

We recommend adding a check to make sure `_cliff < _duration`.

Alleviation

Fixed in commit ea8c2a97cd890c311697cacac24043ae60d99e9d

TVB-02 | Unnecessary Receive() And Fallback() Functions

Category	Severity	Location	Status
Logical Issue	Medium	TokenVesting.sol: 77~79	Resolved

Description

The payable receive() and fallback() functions allow ethers to be transferred to the contract by mistake; those ethers are locked in the contract and lost forever.

Recommendation

We recommend removing the receive() and fallback() functions.

Alleviation

Fixed in commit ea8c2a97cd890c311697cacac24043ae60d99e9d

TVB-03 | Public Functions Should Be Called Without `this.` When Being Called Internally

Category	Severity	Location	Status
Gas Optimization	Minor	TokenVesting.sol: 164, 296, 299	Resolved

Description

The public functions `getWithdrawableAmount()` and `computeNextVestingScheduleIdForHolder()` are called internally by the contract using message call `this.`, which is not gas efficient.

Recommendation

We recommend removing `this.`.

Alleviation

Fixed in commit ea8c2a97cd890c311697cacac24043ae60d99e9d

TVB-04 | Internal Function Names Should Start With _

Category	Severity	Location	Status
Coding Style	Informational	TokenVesting.sol: 281, 290, 344, 345	Resolved

Description

Solidity naming convention requires internal function names to start with _, which makes the code more clear and readable.

Recommendation

We advise the client to follow Solidity naming convention.

Alleviation

Fixed in commit ea8c2a97cd890c311697cacac24043ae60d99e9d

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.

Mathematical Operations

Mathematical Operation findings relate to mishandling of math formulas, such as overflows, incorrect operations etc.

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.

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.