CERTIK

Code Security Assessment

STIMA | Crypto Value Standard

Mar 10th, 2022

Table of Contents

Summary

This report has been prepared for Stima to discover issues and vulnerabilities in the source code of the STIMA | Crypto Value Standard 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	STIMA | Crypto Value Standard
Platform	Etherenum
Language	Solidity
Codebase	https://github.com/STIMA-CryptoValueStandard/STIMA-CryptoValueStandard/blob/836840449aadd8d9b52f3bd2d14f94073d29da22/stima.sol https://github.com/STIMA-CryptoValueStandard/STIMA-CryptoValueStandard/blob/836840449aadd8d9b52f3bd2d14f94073d29da22/STIMA_audit.sol
Commit	

Audit Summary

Delivery Date	Mar 10, 2022 UTC
Audit Methodology	Static Analysis, Manual Review

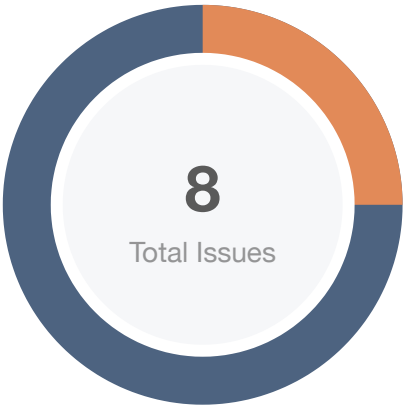
Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Mitigated	Resolved
Critical	0	0	0	0	0	0	0
Major	2	0	0	0	1	1	0
Medium	0	0	0	0	0	0	0
Minor	0	0	0	0	0	0	0
Informational	6	0	0	2	0	0	4
Discussion	0	0	0	0	0	0	0

Audit Scope

ID	File	SHA256 Checksum
ACS	@openzeppelin/contracts@4.3.2/access/AccessControl.sol	6815a22e5b2ef7e0e813961ad06afac5c9d6e7cdced9165f2cedbf11032044bd
OSC	@openzeppelin/contracts@4.3.2/access/Ownable.sol	0195650aabf5270babe540969c56f8f244342aebce89266787a3b015e41d608f
ERB	@openzeppelin/contracts@4.3.2/token/ERC20/extensions/ERC20Burnable.sol	600052c7df2ee2e969592df597ae5f78aad5822c8bee181e58b2713321efb888
ERF	@openzeppelin/contracts@4.3.2/token/ERC20/extensions/ERC20FlashMint.sol	56f366ea13b72627561596c0602145fa8ddaec900177f90e669105b7e244d228
ERS	@openzeppelin/contracts@4.3.2/token/ERC20/extensions/ERC20Snapshot.sol	cd072ee3d83f3758b507c3ea7c39b83311043247807a78a2832099a6c38440c4
ERV	@openzeppelin/contracts@4.3.2/token/ERC20/extensions/ERC20Votes.sol	09852784cfc3ed955e4d9129231fedd23c0eaad720092821bf622f41b5376ae7
ERP	@openzeppelin/contracts@4.3.2/token/ERC20/extensions/draft-ERC20Permit.sol	4caeef3716ac6c9bed65567dbc61169bc9740824e8791bfced2775dfd5a4f06b
ERC	@openzeppelin/contracts@4.3.2/token/ERC20/ERC20.sol	80e33e340442acecc4bd995b4ead9b51adc4231c8213357fca18996b945f850b
SCP	contracts/stima.sol	d5b7c012420430ba96d6b9c2b04e75c601b24fb620b4e572280c8203e98a2593

Findings



Critical	**0** (0.00%)
Major	**2** (25.00%)
Medium	**0** (0.00%)
Minor	**0** (0.00%)
Informational	**6** (75.00%)
Discussion	**0** (0.00%)

ID	Title	Category	Severity	Status
SCK-01	Unlocked Compiler Version & Version Inconsistency	Language Specific	Informational	Resolved
SCK-02	Function Visibility Optimization	Gas Optimization	Informational	Resolved
SCP-01	Missing Emit Events	Gas Optimization	Informational	Resolved
SCP-02	Centralization Related Risks	**Centralization / Privilege**	**Major**	Mitigated
SCP-03	Division Before Multiplication	Language Specific	Informational	Resolved
SCP-04	Unused `Ownable` Contract	Coding Style	Informational	Acknowledged
SCP-05	Discussion For Function `flashFee()`	Logical Issue	Informational	Acknowledged
SCP-06	Lack of Check for Function `burnFrom()`	Logical Issue	Major	Partially Resolved

SCK-01 | Unlocked Compiler Version & Version Inconsistency

Category	Severity	Location	Status
Language Specific	Informational	contracts/stima.sol: 2 @openzeppelin/contracts@4.3.2/token/ERC20/ERC20.sol: 3 @openzeppelin/contracts@4.3.2/token/ERC20/extensions/ERC20Burnable.sol: 3 @openzeppelin/contracts@4.3.2/token/ERC20/extensions/ERC20FlashMint.sol: 3 @openzeppelin/contracts@4.3.2/token/ERC20/extensions/ERC20Snapshot.sol: 3 @openzeppelin/contracts@4.3.2/token/ERC20/extensions/ERC20Votes.sol: 3 @openzeppelin/contracts@4.3.2/token/ERC20/extensions/draft-ERC20Permit.sol: 3 @openzeppelin/contracts@4.3.2/access/AccessControl.sol: 3 @openzeppelin/contracts@4.3.2/access/Ownable.sol: 3	Resolved

Description

The contract has unlocked compiler versions. 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. Examples:

`pragma solidity ^0.8.2;` `stima.sol#2` `pragma solidity ^0.8.0;` `ERC20.sol#3`

Recommendation

It is general practice to alternatively lock the compiler at a specific version rather than allow a range of compiler versions to be utilized to avoid compiler-specific bugs and thus be able to detect emerging ones. We recommend locking the compiler at the lowest possible version that supports all the capabilities required by the codebase. This will ensure that the project utilizes a compiler version that has been in use for the longest time and as such is less likely to contain yet-undiscovered bugs.

Alleviation

The client locked the compiler at `0.8.7` version.

SCK-02 | Function Visibility Optimization

Category	Severity	Location	Status
Gas Optimization	Informational	contracts/stima.sol: 32, 36, 80, 85, 89, 94, 104, 110, 114, 118 @openzeppelin/contracts@4.3.2/token/ERC20/extensions/ERC20Snapshot.sol: 105, 114 @openzeppelin/contracts@4.3.2/token/ERC20/extensions/ERC20Votes.sol: 53, 60, 74, 86, 99, 136, 143 @openzeppelin/contracts@4.3.2/token/ERC20/extensions/ERC20FlashMint.sol: 25, 58 @openzeppelin/contracts@4.3.2/token/ERC20/extensions/draft-ERC20Permit.sol: 40, 64 @openzeppelin/contracts@4.3.2/token/ERC20/extensions/ERC20Burnable.sol: 19, 34 @openzeppelin/contracts@4.3.2/token/ERC20/ERC20.sol: 61, 69, 86, 112, 131, 149, 177, 196 @openzeppelin/contracts@4.3.2/access/AccessControl.sol: 129, 142, 160 @openzeppelin/contracts@4.3.2/access/Ownable.sol: 53, 61	Resolved

Description

`public` functions that are never called by the contract could be declared `external`. When the inputs are arrays, `external` functions are more efficient than `public` functions.

Recommendation

We advise that the functions' visibility specifiers are set to `external` and the array-based arguments change their data location from `memory` to `calldata`, optimizing the gas cost of the function.

Alleviation

The client heeded our advice and resolved this issue in commit : 836840449aadd8d9b52f3bd2d14f94073d29da22.

SCP-01 | Missing Emit Events

Category	Severity	Location	Status
Gas Optimization	Informational	contracts/stima.sol: 104	Resolved

Description

Functions that affect the status of sensitive variables should be able to emit events as notifications to customers.

Recommendation

We advise the client to add events for sensitive actions and emit them in the function.

Alleviation

The client heeded our advice and resolved this issue in commit : 836840449aadd8d9b52f3bd2d14f94073d29da22.

SCP-02 | Centralization Related Risks

Category	Severity	Location	Status
Centralization / Privilege	**Major**	contracts/stima.sol: 32, 36, 89, 94, 104	Mitigated

Description

The `SNAPSHOT_ROLE` has the responsibility to notify users about the following capabilities:

- creates a new snapshot and returns its snapshot id through `snapshot()`

The `MINTER_ROLE` has the responsibility to notify users about the following capabilities:

- mint uncapped tokens to anyone and mint part tokens to the owner through `mint()`
- release tokens to anyone through `releaseBalance()`
- burn tokens from the holder through `burnFromHold()`

The `DEFAULT_ADMIN_ROLE` has the responsibility to notify users about the following capabilities:

- set `_numenator` and `_denomenator` through `setMinterPart()`

Any compromise to the `SNAPSHOT_ROLE`/`MINTER_ROLE`/`DEFAULT_ADMIN_ROLE` account may allow a hacker to take advantage of this authority.

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.

Alleviation

`[Client]`: STIMA itself plays the role of a fiduciary third party. In order to guarantee maximum security parameters, a distribution hierarchy of fundamental roles envisaged by the smart contract has been implemented.

For this purpose, blockchain-based external systems have been introduced:

- DEFENDER: administration panel (https://docs.openzeppelin.com/defender/)
- Gnosis Safe: multi-sign wallet (https://gnosis-safe.io)

The STIMA control chain is developed on 3 layers:

- Layer 1 - CONTRACT OWNER

The ownership of the contract is entrusted to a multisign wallet, which represents the corporate shareholders, which requires a 2/3 vote to confirm transactions:
Gnosis Safe Multisig wallet:
0xeD6eE0C91eeb3b89ecbD9238c29e6d86519AC088
The decision-making process is made by the following wallets:

- 0xB6650eb6c90F52a5AE47696a7766f6B6108188D5
- 0x342db5fFF9f6f4F4a113a01B2A28B49B547A9547
- 0x309E7d2a5930b9029Db03Aa5A7DD49c1FF64555c

This role performs the following functions:

- Appoints the contract administrator (Layer2),
- Receives commissions

- Layer 2 - CONTRACT ADMIN

The administration of the contract is entrusted to a multisign wallet, which represents the corporate shareholder, which requires a 2/3 vote to confirm transactions:
Gnosis Safe Multisig wallet:
0xeD6eE0C91eeb3b89ecbD9238c29e6d86519AC088
The decision-making process is made by the following wallets:

- 0xB6650eb6c90F52a5AE47696a7766f6B6108188D5
- 0x342db5fFF9f6f4F4a113a01B2A28B49B547A9547
- 0x309E7d2a5930b9029Db03Aa5A7DD49c1FF64555c

This role performs the following functions:

- Appoint Snapshot role (Layer 3.1)
- Appoint minter role (Layer 3.2)

- Layer 3 - MINTER ROLE

The role of MINTER is entrusted to a multi-sig wallet, which requires a 2/3 vote to confirm transactions:
Gnosis Safe Multisig wallet:
0xeD6eE0C91eeb3b89ecbD9238c29e6d86519AC088
The decision-making process is made by the following wallets:

- 0xB6650eb6c90F52a5AE47696a7766f6B6108188D5
- 0x342db5fFF9f6f4F4a113a01B2A28B49B547A9547
- 0x309E7d2a5930b9029Db03Aa5A7DD49c1FF64555c

This role performs the following functions:

- MINT: generates tokens on the indicated wallet
- BURN FROM HOLD: Can only burn frozen tokens
- RELEASE BALANCE: allows defrosting tokens

All roles will be entrusted to a multi-sign wallet.

The contract also includes a DAO functionality, which will be implemented as soon as the community, ecosystem, and infrastructure allow.

At this link you can find more detailed information: https://medium.com/@stima.io/centralized-decentralization-c01cd777b92c

SCP-03 | Division Before Multiplication

Category	Severity	Location	Status
Language Specific	Informational	contracts/stima.sol: 37	Resolved

Description

Mathematical operations in the aforementioned function perform divisions before multiplications. Performing multiplication before division can sometimes avoid loss of precision.

Recommendation

We advise the client to apply multiplications before divisions so that integer overflow would not happen in functions.

Alleviation

The client heeded our advice and resolved this issue in commit : 836840449aadd8d9b52f3bd2d14f94073d29da22.

SCP-04 | Unused `Ownable` Contract

Category	Severity	Location	Status
Coding Style	Informational	contracts/stima.sol: 13	Acknowledged

Description

The `Ownable` contract provides the owner tools to maintain full control over any function they choose. In the `STIMA` contract the `onlyOwner()` modifier is never used. Therefore the owner does not have have complete control of any function.

Recommendation

we advise the client to remove the `Ownable` contract.

Alleviation

`[Client]`: The reason why the ownership of the contract was retained is due to the need to represent the ownership of the company.

In order to increase security the ownership of the contract is entrusted to a multisign wallet, which represents the majority of the votes (51%) of the owners of the company, which requires a 2/2 vote to confirm transactions. This role performs the following functions:

- appoints the contract administrator
- receives minting commissions

At this link you can find more information: https://medium.com/@stima.io/centralized-decentralization-c01cd777b92c

SCP-05 | Discussion For Function `flashFee()`

Category	Severity	Location	Status
Logical Issue	Informational	contracts/stima.sol: 13	Acknowledged

Description

The function `flashFee()` in the contract `ERC20FlashMint` returns the fee applied when doing flash loans. By default this implementation has 0 fees. However, the function does not be overloaded in the contract `STIMA`.

Recommendation

We would like to confirm with the client if the current implementation aligns with the original project design.

Alleviation

`[Client]`: This feature will be used as soon as we have the pawnshop license.

In fact, any person who holds an asset with a recognized intrinsic value can pledge his asset to obtain a loan. Upon expiry of the terms for redeeming the asset by paying the debt, the asset will be permanently withheld.

SCP-06 | Lack Of Check For Function `burnFrom()`

Category	Severity	Location	Status
Logical Issue	Major	contracts/stima.sol: 53	Partially Resolved

Description

The function `_beforeTokenTransfer()` is overridden to check that accounts have the necessary balance, not into the hold. The contract `STIMA` inherits from `ERC20Burnable`, so the function `burnFrom()` is public. However, the function `burnFrom()` does not satisfy the check condition of line 53, so tokens are in the hold can be burned.

Recommendation

We advise the client to recheck the logic.

Alleviation

`[Client]`: All tokens are minted in a locked condition, to be unlocked the following conditions must be met:

- The collateral asset of which they have been minted must be delivered to a vault accredited by STIMA.
- The asset must match the conditions encountered during the minting process

The function was implemented for the following purposes:

- Reduce the amount of tokens if the asset does not match the conditions described
- Burn all tokens if the asset is found to be fake or damaged
- Burn all tokens if the asset leaves the STIMA system

This function is necessary, as STIMA plays the role of a fiduciary third party, and plays the role of guarantor on the quality of withdrawn assets and distributed tokens.

In fact, as soon as all the required conditions are present the tokens are unlocked and cannot be manipulated in any way.

A detailed explanation of the logic and use of this function can be found at this link: https://medium.com/@stima.io/stima-token-release-process-73156962a0e7

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.

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.