Security Assessment

Channels

Nov 15th, 2021

Table of Contents

Summary

This report has been prepared for Channels to discover issues and vulnerabilities in the source code of the Channels 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	Channels
Description	Channels
Platform	BSC
Language	Solidity
Codebase	https://github.com/ChannelsFinance/ChannelsProtocolV2
Commit	c1899219fac14a9ed6e4b67900d81986b160714b

Audit Summary

Delivery Date	Nov 15, 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	3	0	0	3	0	0
Medium	1	0	0	0	0	1
Minor	0	0	0	0	0	0
Informational	3	0	0	0	1	2
Discussion	0	0	0	0	0	0

Audit Scope

ID	File	SHA256 Checksum
CGE	CGatlingErc20Delegate.sol	3f220070d9c4cf8c8a85bcdd993046c8d3aa404f674c02f47a7ada819dd0a62c
CGD	CGatlingErc20Delegator.sol	1b582979988c8214865c2e77b0ec73c87fd616102aa640f4d6cc37e2a86ccb78
SSC	SpotSpell.sol	c4e5d9caa2bef44f6c76a4205ac8eb914b112fe8f18a6930614cd12a793cfc5b
WCC	WComptroller.sol	3a4d5dbcca9c24d19b7134fd3da74fd22b8aaf82a369cd2ec9c90d7c60019013

External Dependencies

The contract `SpotSpellV1` is serving as the underlying entity to interact with third party `UNISWAP` protocol.

The contract `CGatlingErc20Delegate` is serving as the underlying entity to interact with third party `MdexRouter`, `USDT`, `SwapMining`, `CoinWindGatling`, `MdxToken` and `CoinWindToken` protocols.

There are many files being imported for use but they are not in the scope of the audit. The scope of the audit treats these files as black boxes and assumes their functional correctness.

Details of the associated file import are as follows:

- contract `WComptroller`: `ERC1155.sol`, `IERC20.sol`, `SafeERC20.sol`, `ReentrancyGuard.sol`, `ChannelsMath.sol`, `IERC20Wrapper.sol`, `IComptroller.sol`, `ICErc20.sol`
- contract `SpotSpellV1`: `IERC20.sol`, `SafeMath.sol`, `WhitelistSpell.sol`, `ChannelsMath.sol`, `IUniswapV2Factory.sol`, `IUniswapV2Router02.sol`, `IUniswapV2Pair.sol`, `IWComptroller.sol`
- contract `CGatlingErc20Delegator`: `CTokenInterfaces.sol`
- contract `CGatlingErc20Delegate`: `CCapableErc20Delegate.sol`, `EIP20Interface.sol`

Details of the associated use are as follows:

- contract `WComptroller`: `comptroller.canSupplierIndex()`, `comptroller.claimCan()`,
- contract `SpotSpellV1`: `doTransmitBNB()`, `doTransmit()`, `doBorrow()`, `bank.getCurrentPositionInfo()`, `bank.takeCollateral()`, `bank.putCollatera()`, `doRefundBNB()`, `doRefund()`, `bank.borrowBalanceCurrent()`, `doRepay()`

We understand that the business logic of strategy and price oracle requires interaction with UNISWAP, etc. We encourage the team to constantly monitor the statuses of 3rd parties to mitigate the side effects when unexpected activities are observed.

The scope of the audit treats 3rd party entities as black boxes and assumes their functional correctness. However, in the real world, 3rd parties can be compromised and this may lead to lost or stolen assets. In addition, upgrades of 3rd parties can possibly create severe impacts, such as increasing fees of 3rd parties, migrating to new LP pools, etc.

Findings

Critical	**0** (0.00%)
Major	**3** (42.86%)
Medium	**1** (14.29%)
Minor	**0** (0.00%)
Informational	**3** (42.86%)
Discussion	**0** (0.00%)

ID	Title	Category	Severity	Status
GLOBAL-01	Incorrect naming convention utilization	Coding Style	Informational	Resolved
GLOBAL-02	Proper usage of “public” and “external” type	Coding Style	Informational	Partially Resolved
CGD-01	Centralization risk	**Centralization / Privilege**	**Major**	Acknowledged
CGE-01	Centralization risk	**Centralization / Privilege**	**Major**	Acknowledged
CGE-02	Potential subtraction overflow	Logical Issue	Medium	Resolved
CGE-03	Logical issue of `depositAmount`	Logical Issue	Informational	Resolved
WCC-01	Centralization risk	**Centralization / Privilege**	**Major**	Acknowledged

GLOBAL-01 | Incorrect naming convention utilization

Category	Severity	Location	Status
Coding Style	Informational	Global	Resolved

Description

Solidity defines a naming convention that should be followed. In general, the following naming conventions should be utilized in a Solidity file:

Constants should be named with all capital letters with underscores separating words UPPER_CASE_WITH_UNDERSCORES

refer to https://solidity.readthedocs.io/en/v0.5.17/style-guide.html#naming-conventions

Examples:

Constants like :

- contract `CGatlingErc20Delegate`: `mdexRouter`, `SwapMining`, `CoinWindGatling`, `mdx`, `cow`, `reInvestPeriod`,

Recommendation

The recommendations outlined here are intended to improve the readability, and thus they are not rules, but rather guidelines to try and help convey the most information through the names of things.

Alleviation

The team heeded our advice and resolved this issue in commit `2ff45ad0a664a14600e78555dde55534e262892d`.

GLOBAL-02 | Proper usage of “public” and “external” type

Category	Severity	Location	Status
Coding Style	Informational	Global	Partially Resolved

Description

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

Examples:

Functions like :

- contract `CGatlingErc20Delegator`: `emergencyWithdraw()`, `borrowBalanceStored()`, `exchangeRateCurrent()`, `exchangeRateStored()`, `accrueInterest()`, `_setComptroller()`, `_setInterestRateModel()`,
- contract `WComptroller`: `getUnderlyingToken()`, `balanceOfBatch()`, `setApprovalForAll()`, `safeTransferFrom()`, `safeBatchTransferFrom()`, `supportsInterface()`,

Recommendation

We recommend using the “external” attribute for functions never called from the contract.

Alleviation

The team heeded our advice and partially resolved this issue in commit `fdd5a639b7d5708dfdc31dbe259b7e59b77db9f2`.

CGD-01 | Centralization risk

Category	Severity	Location	Status
Centralization / Privilege	**Major**	channels/CGatlingErc20Delegator.sol (83c4de8): 60~73	Acknowledged

Description

In the contract `CGatlingErc20Delegator`, the role `admin` has the authority over the functions shown in the diagram below.

- `_setImplementation()`: change the address of implementation to any addresses,
- `_setComptroller()`: change the address of comptroller to any addresses,
- `_setPendingAdmin()/_acceptAdmin()`: change the address of admin to any addresses,
- `_setInterestRateModel()`: update the interest rate model`,
- `_setReserveFactor()`: set a new reserve factor for the protocol,

Any compromise to the privileged account which has access to `admin` may allow the hacker to take advantage of this.

Recommendation

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 to be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., Multisignature wallets.

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

- 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.

Alleviation

The team acknowledged this issue and they will use a time-lock contract to help control the risk on their own timeframe.

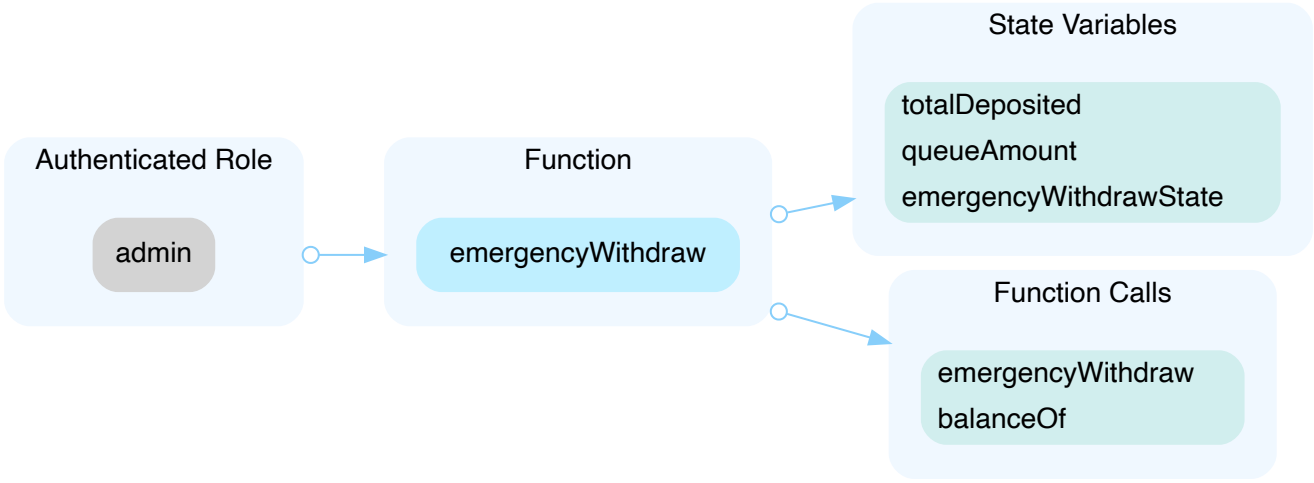
CGE-01 | Centralization risk

Category	Severity	Location	Status
Centralization / Privilege	Major	channels/CGatlingErc20Delegate.sol (83c4de8): 269~280	Acknowledged

Description

In the contract `CGatlingErc20Delegate`, the role `admin` has the authority over the functions shown in the diagram below.

Any compromise to the privileged account which has access to `admin` may allow the hacker to take advantage of this.



Recommendation

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 to be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., Multisignature wallets.

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

- 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.

Alleviation

The team acknowledged this issue and they will use a time-lock contract to help control the risk on their own timeframe.

CGE-02 | Potential subtraction overflow

Category	Severity	Location	Status
Logical Issue	Medium	channels/CGatlingErc20Delegate.sol (83c4de8): 226	Resolved

Description

According to the following codes, the comments "If limit = 0 , then there is no limit" explains `poolInfo.totalAmountLimit` is 0 when there is no limit.

```
226         uint depositAmount = add_(amount, queueAmount);
227         uint depositAllowed = sub_(poolInfo.totalAmountLimit, poolInfo.totalAmount);
228         if (depositAllowed >= 1){
229             depositAllowed = sub_(depositAllowed, 1);
230         }
231 
232         // If limit = 0 , then there is no limit
233         if(poolInfo.totalAmountLimit !=0 && depositAmount > depositAllowed){
234             depositAmount = depositAllowed;
235         }
```

As a result, the subtraction `sub_(poolInfo.totalAmountLimit, poolInfo.totalAmount)` may overflow and lead the call to fail.

Recommendation

We recommend adding logic to handle this condition.

Alleviation

The team heeded our advice and resolved this issue in commit `2ff45ad0a664a14600e78555dde55534e262892d`.

CGE-03 | Logical issue of `depositAmount`

Category	Severity	Location	Status
Logical Issue	Informational	channels/CGatlingErc20Delegate.sol (83c4de8): 226~243	Resolved

Description

According to the following codes, the `depositAmount` will be calculated by the `depositAllowed` when `poolInfo.totalAmountLimit` is not 0.

```
226         uint depositAmount = add_(amount, queueAmount);
227         uint depositAllowed = sub_(poolInfo.totalAmountLimit, poolInfo.totalAmount);
228         if (depositAllowed >= 1){
229             depositAllowed = sub_(depositAllowed, 1);
230         }
231
232         // If limit = 0 , then there is no limit
233         if(poolInfo.totalAmountLimit !=0 && depositAmount > depositAllowed){
234             depositAmount = depositAllowed;
235         }
236
237         if (depositAmount > 0){
238             uint beforeDeposit =
ICoinWind(CoinWindGatling).getDepositAsset(underlying, address(this));
239             ICoinWind(CoinWindGatling).deposit(underlying, depositAmount);
240             uint afterDeposit =
ICoinWind(CoinWindGatling).getDepositAsset(underlying, address(this));
241
242             require(sub_(afterDeposit, beforeDeposit) == depositAmount, "unexpected
error occurred during deposit to CoinWind");
243         }
```

The `depositAllowed` will minus 1 when `sub_(poolInfo.totalAmountLimit, poolInfo.totalAmount)` is over 0. As `ICoinWind(CoinWindGatling)` is not in the scope of this audit. Whether this calculation is correct is unknown.

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

Recommendation

We recommend stating for this.

Alleviation

The team heeded our advice and resolved this issue in commit `2ff45ad0a664a14600e78555dde55534e262892d`.

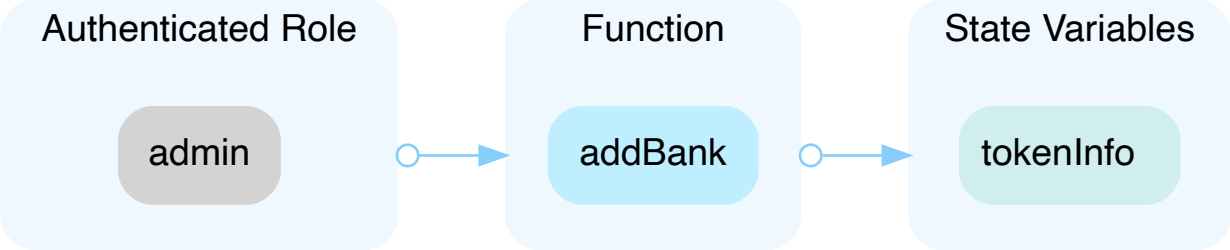
WCC-01 | Centralization risk

Category	Severity	Location	Status
Centralization / Privilege	Major	channels/WComptroller.sol (83c4de8): 68~74	Acknowledged

Description

In the contract `WComptroller`, the role `admin` has the authority over the functions shown in the diagram below.

Any compromise to the privileged account which has access to `admin` may allow the hacker to take advantage of this.



Recommendation

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 to be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., Multisignature wallets.

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

- 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.

Alleviation

The team acknowledged this issue and they will use a time-lock contract to help control the risk on their own timeframe.

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.

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.

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.