CERTIK

Security Assessment

JOJO

May 25th, 2022

Table of Contents

Summary

This report has been prepared for JOJO to discover issues and vulnerabilities in the source code of the JOJO 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 contracts were implemented safely.

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	JOJO
Platform	Ethereum
Language	Solidity
Codebase	https://github.com/JOJOexchange/smart-contract-EVM
Commit	cdf04c6f21c27b58a6f995d104a9629f61bd768a

Audit Summary

Delivery Date	May 25, 2022 UTC
Audit Methodology	Static Analysis, Manual Review

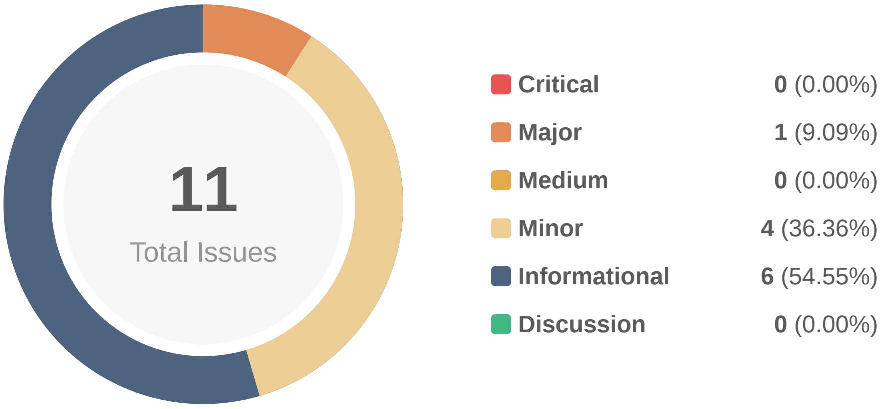
Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Mitigated	Partially Resolved	Resolved
● Critical	0	0	0	0	0	0	0
● Major	1	0	0	1	0	0	0
● Medium	0	0	0	0	0	0	0
● Minor	4	0	0	4	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
SCK	subaccount/Subaccount.sol	fcf66ff334f9751009a82361e8570ea32080039e6b5088e327bcd009bc04396b
IMP	intf/IMarkPriceSource.sol	e3fa813cd7f4fedbe170131e771ffb982f67807b5448ec2690da82a48b9a57d1
SFC	subaccount/SubaccountFactory.sol	4cf795ee1589fa4df32eb2db78acaa25591dc6d8a4bc78b6cfad870ae8e6efb2
JOC	impl/JOJOOperation.sol	eadb4242e531c7bdf265ccaba97990af31ea17ba8bee578072ef39a4974b37c5
TCK	lib/Trading.sol	78842bf8b29d347d1c5607c0386fce9efd2028c2fda3eb2cfc6a44f7a3f3c7f1
FCK	lib/Funding.sol	b7dee2bb53723622c8f4589f9ece74db2a0c0510c7518051251c9a26800a6413
TCP	lib/Types.sol	82af092ab3c112034450d3ab80679d4810fbf84f9d3792787ed93dfe516f0fc0
JOS	impl/JOJOStorage.sol	492865d560943aaeddda38ce4bdb8e8107fbe45c1adc8876a36c1034bc50eb7e
JOV	impl/JOJOView.sol	db471976ec739f9f4a16608f9c03327e9413305a3c0815798b4d55a940900513
IDC	intf/IDealer.sol	9a17657c06a7df3024ee5e087bc381140fab7d5ff4e684269737a08d74b77e78
JOO	impl/JOJOExternal.sol	4f7e6e78212ac16f582c77116f7a03e376bcc71741a9e451f94988e9cbc6b62e
LCK	lib/Liquidation.sol	71aa1796713d6f86a7ff8b79b4c71e9d79bbf9ffa3e0890420fcd08a74be43fe
ECK	utils/Errors.sol	cf8986e2ae62ecc2757a1b317e9c68c3e9b7ed4feccc14238fb4dfccf85d5490
EIP	lib/EIP712.sol	96d0f078a13b2149f2134ea76feb9856b40bcc566c8db0cf1ffd05de5a4ceb70
OCK	lib/Operation.sol	f5a9c2095a1da9c4a62bf0b25ac711bcb3669a2959e725285f9de203ea0f84af
JOJ	impl/JOJODealer.sol	e7d0e6731843522415a927cec55ab7e7c5a1a98bf0712f2c0be99e93839b7cb3
PCK	impl/Perpetual.sol	9f84693413de677dfcfe4b9fb947c65eb7fc1f4ad63270009ad7b4c9360c5aa1
SDM	utils/SignedDecimalMath.sol	0f8607bc88f34e226fe80d3fea473124898330df47d17be1455cc5c77476c12b
IPC	intf/IPerpetual.sol	ab6398213e20e3865c377456903e44362142d0dbdaf2f8e33700e2eecb687b52

Findings



ID	Title	Category	Severity	Status
GLOBAL-01	Centralization Related Risks	**Centralization / Privilege**	● **Major**	ⓘ Acknowledged
GLOBAL-02	Third Party Dependencies	Volatile Code	● Minor	ⓘ Acknowledged
GLOBAL-03	Reliability Of Price	Logical Issue	● Minor	ⓘ Acknowledged
CON-01	Improper Usage Of `public` And `external` Type	Gas Optimization	● Informational	✓ Resolved
JOC-01	Missing Emit Events	Coding Style	● Informational	ⓘ Acknowledged
LIB-01	Functions With `_` As Name Prefix Are Not `private` Or `internal`	Coding Style	● Informational	✓ Resolved
OCK-01	Removal Of `Perpetual`	Logical Issue	● Minor	ⓘ Acknowledged
OCK-02	Lack Validation For Array Length	Logical Issue	● Informational	✓ Resolved
SCK-01	Unused State Variable	Gas Optimization	● Informational	✓ Resolved
SUB-01	Missing Zero Address Validation	Volatile Code	● Minor	ⓘ Acknowledged
TCP-01	Introduction For `primaryAsset` And `secondaryAsset`	Logical Issue	● Informational	ⓘ Acknowledged

GLOBAL-01 | Centralization Related Risks

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

Description

In the contract `JOJOOperation`, the role `owner` has authority over the following functions:

- function handleBadDebt()
- function setPerpRiskParams()
- function setFundingRateKeeper()
- function setInsurance()
- function setWithdrawTimeLock()
- function setOrderSender()
- function setSecondaryAsset()

In the contract `Perpetual`, the role `owner` has authority over the following functions:

- function changeCredit()

In the contract `Operation`, the role `fundingRateKeeper` has authority over the following functions:

- function _updateFundingRate()

In the contract `Subaccount`, the role `owner` has authority over the following functions:

- function setOperator()
- function requestWithdraw()
- function executeWithdraw()

Any compromise to these accounts 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., multisignature 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

The team response:

```
The perpetual's owner will always be JOJODealer, so no worry about it.
```

Subaccount's owner will be the one who created it. It's totally permissionless and won't influence JOJO's trading system.

FundingRateKeeper will be an EOA account managed by JOJO's team. We admit it is centralized by design.

And the owner of JOJOOperation (it is also the owner of JOJODealer) will be a 2of3 gnosis safe wallet. Will provide the address before the product launch.

GLOBAL-02 | Third Party Dependencies

Category	Severity	Location	Status
Volatile Code	● Minor		ⓘ Acknowledged

Description

The contract is serving as the underlying entity to interact with third-party protocols. 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.

Recommendation

We understand that the business logic of JOJO requires interaction with `MarkPriceSource`, etc. We encourage the team to constantly monitor the statuses of 3rd parties to mitigate the side effects when unexpected activities are observed.

Alleviation

The team response:

```
Thank's for your recommendation. The MarkPriceSource will be chainlink. And we will keep
an eye on it.

If chainlink's oracle fails, we will just shut down the whole system and switch to a
self-hold backup contract as soon as possible.
```

GLOBAL-03 | Reliability Of Price

Category	Severity	Location	Status
Logical Issue	● Minor		ⓘ Acknowledged

Description

The `mark price` comes from the external contract `MarkPriceSource`. If the hacker manipulates the price, The traders' positions may be unreasonably liquidated. For example, flash loan attacks, etc.

Recommendation

We recommend the development team guarantees that the price is reliable.

Alleviation

The team response:

```
Yes, you're right.

We will treat the oracle very seriously.
```

CON-01 | Improper Usage Of `public` And `external` Type

Category	Severity	Location	Status
Gas Optimization	● Informational	subaccount/Subaccount.sol: 49; impl/Perpetual.sol: 72; lib/Liquidation.sol: 149	✓ Resolved

Description

`public` functions that are never called by the contract could be declared as `external`. `external` functions are more efficient than `public` functions.

Recommendation

Consider using the external attribute for public functions that are never called within the contract.

Alleviation

The team has resolved this issue in commit `f2df5af85201d4fc963cc4ce624590e1f22769a4`.

JOC-01 | Missing Emit Events

Category	Severity	Location	Status
Coding Style	● Informational	impl/JOJOOperation.sol: 77	ⓘ Acknowledged

Description

There should always be events emitted in the sensitive functions that are controlled by centralization roles.

Recommendation

It is recommended emitting events for the sensitive functions that are controlled by centralization roles.

Alleviation

No Alleviation.

LIB-01 | Functions With `_` As Name Prefix Are Not `private` Or `internal`

Category	Severity	Location	Status
Coding Style	● Informational	lib/Funding.sol: 45~68, 72~92, 94~117; lib/Liquidation.sol: 52~80, 132~147, 149~226, 232~281, 283~290, 292~308; lib/Operation.sol: 36~64, 66~81, 83~89, 91~97, 99~106, 108~115, 117~126	✓ Resolved

Description

Functions with names starting with `_` should be declared as `private`/`internal`.

Recommendation

Consider changing function visibility to private or removing `_` from the start of the function name.

Alleviation

The team has resolved this issue in commit `f2df5af85201d4fc963cc4ce624590e1f22769a4`.

OCK-01 | Removal Of `Perpetual`

Category	Severity	Location	Status
Logical Issue	● Minor	lib/Operation.sol: 43~50	ⓘ Acknowledged

Description

`Perpetual` can be removed from `state.registeredPerp`. If there are open positions in the `Perpetual`, the traders' assets may suffer a loss.

Recommendation

The `owner` should make sure there are no open positions in the removed `Perpetual`.

Alleviation

The team response:

```
Acknowledged, thanks.
```

OCK-02 | Lack Validation For Array Length

Category	Severity	Location	Status
Logical Issue	● Informational	lib/Operation.sol: 71	✓ Resolved

Description

There is no validation to check whether `perpList` and `rateList` have the same length.

Recommendation

We recommended adding validation as below:

```
71      require(perpList.length > 0, "invalid array length");
72      require(perpList.length == rateList.length,"perpList and rateList length mismatch");
```

Alleviation

The team has resolved this issue in commit `f2df5af85201d4fc963cc4ce624590e1f22769a4`.

SCK-01 | Unused State Variable

Category	Severity	Location	Status
Gas Optimization	● Informational	subaccount/Subaccount.sol: 28	✓ Resolved

Description

One or more state variables are never used in the codebase.

Variable `validOperator` in `Subaccount` is never used in `Subaccount`.

File: projects/jojo/contracts/subaccount/Subaccount.sol (Line 28, Contract `Subaccount`)

```
    mapping(address => bool) validOperator;
```

Recommendation

We advise removing the unused variables.

Alleviation

The team has resolved this issue in commit `f2df5af85201d4fc963cc4ce624590e1f22769a4`.

SUB-01 | Missing Zero Address Validation

Category	Severity	Location	Status
Volatile Code	● Minor	subaccount/Subaccount.sol: 42, 43; subaccount/SubaccountFactory.sol: 30	ⓘ Acknowledged

Description

Addresses should be checked before assignment or external calls to make sure they are not zero addresses.

File: projects/jojo/contracts/subaccount/Subaccount.sol (Line 42, Function `Subaccount.init`)

```
        owner = _owner;
```

- `_owner` is not zero-checked before being used.

File: projects/jojo/contracts/subaccount/Subaccount.sol (Line 43, Function `Subaccount.init`)

```
        dealer = _dealer;
```

- `_dealer` is not zero-checked before being used.

File: projects/jojo/contracts/subaccount/SubaccountFactory.sol (Line 30, Function `SubaccountFactory.constructor`)

```
        dealer = _dealer;
```

- `_dealer` is not zero-checked before being used.

Recommendation

We advise adding a zero-check for the passed-in address value to prevent unexpected errors.

Alleviation

The team response:

> In the business logic, all subaccounts is created by subaccountFactory. So the dealer in subaccount has no need to check. Also, the owner of subaccount must be an EOA owned by someone, no need to check.
>
> We will make sure the dealer registered in subaccountFactory correct.

TCP-01 | Introduction For `primaryAsset` And `secondaryAsset`

Category	Severity	Location	Status
Logical Issue	● Informational	lib/Types.sol: 13, 15	ⓘ Acknowledged

Description

```
    netPositionValue +
        state.primaryCredit[trader] +
        int256(state.secondaryCredit[trader]) >=
        int256((exposure * strictLiqThreshold) / 10**18);
```

According to the liquidation logic, the `primaryAsset` and `secondaryAsset` have the same price. What exactly tokens they are?

Recommendation

Please provide more information about `primaryAsset` and `secondaryAsset`.

Alleviation

The team response:

```
Primary Asset is USDC. And Secondary Asset is USDJ.

USDJ is a new stable coin minted by JOJO team. It has two main purposes:

To expand liquidity: mint USDJ to trusted MMs to help them provide better liquidity.
Support multi asset collateral in the future: users can deposit any ERC20 to borrow USDJ
and use it to trade. You can think it as a mini lending protocol. Will be implemented in
V1.2

In JOJO's system, everything is settled in the term of primary asset. Secondary asset is
just something like buffer.
```

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.

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.

CERTIK