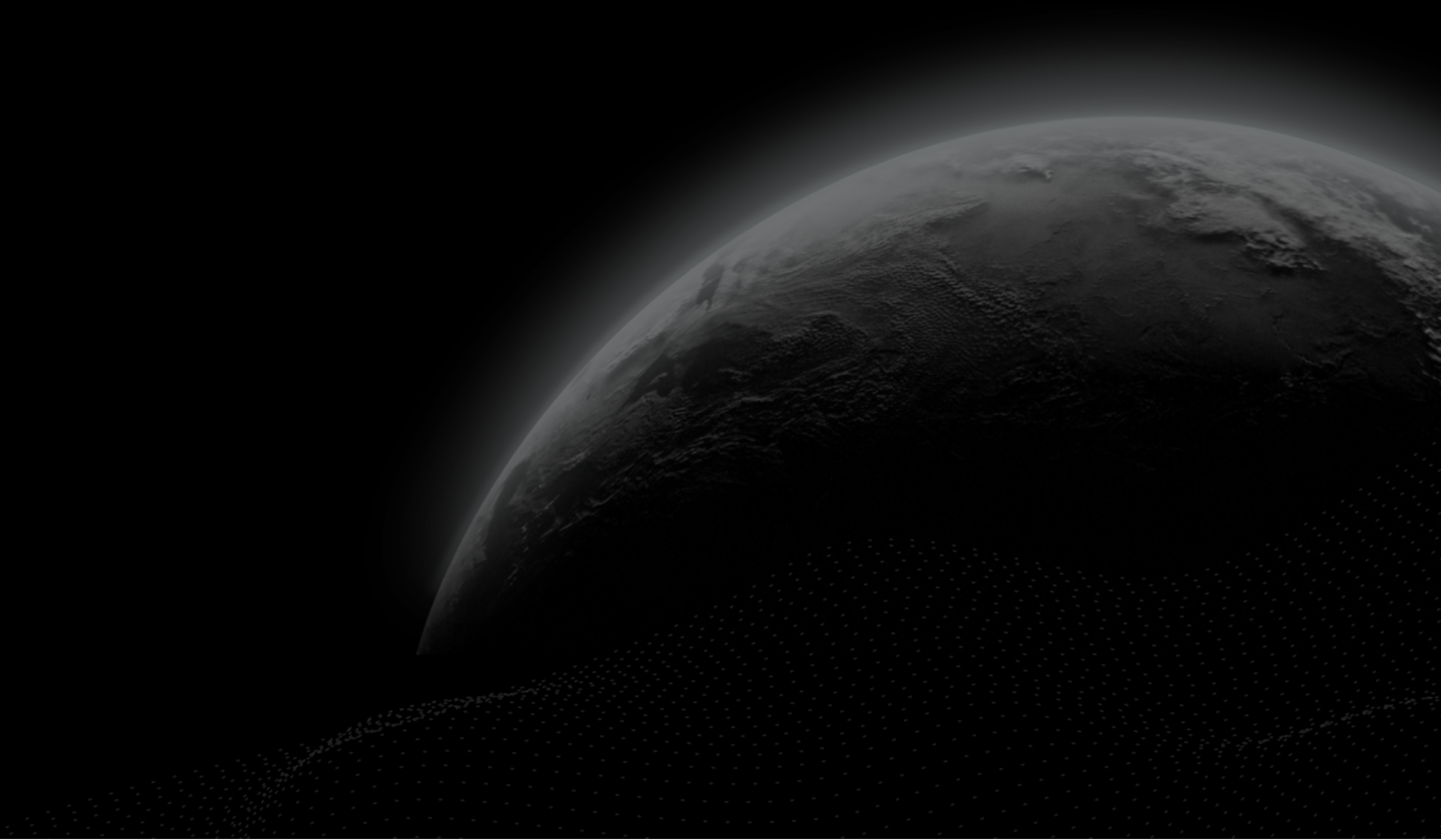
CERTIK

Security Assessment

Venus - RewardsDistributor

CertiK Assessed on Jul 10th, 2023



CertiK Assessed on Jul 10th, 2023

Venus - RewardsDistributor

The security assessment was prepared by CertiK, the leader in Web3.0 security.

Executive Summary

TYPES	ECOSYSTEM	METHODS
DeFi	Binance Smart Chain (BSC)	Manual Review, Static Analysis

LANGUAGE	TIMELINE	KEY COMPONENTS
Solidity	Delivered on 07/10/2023	N/A

CODEBASE

https://github.com/VenusProtocol/isolated-pools

View All in Codebase Page

COMMITS

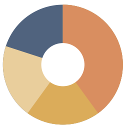
base: 1b173dc1b0a7232a02c174559535ae18c3801b9e

update1: c4590657669993c901e64a5fe9837faeef5017d0

update2: 71a36e64cf1f32d81ba9bd728f230fe488b9190b

View All in Codebase Page

Vulnerability Summary



5	3	2	0	0	0
Total Findings	Resolved	Mitigated	Partially Resolved	Acknowledged	Declined

🟥	0	Critical		Critical risks are those that impact the safe functioning of a platform and must be addressed before launch. Users should not invest in any project with outstanding critical risks.
🟧	2	Major	2 Mitigated	Major risks can include centralization issues and logical errors. Under specific circumstances, these major risks can lead to loss of funds and/or control of the project.
🟨	1	Medium	1 Resolved	Medium risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform.
🟨	1	Minor	1 Resolved	Minor risks can be any of the above, but on a smaller scale. They generally do not compromise the overall integrity of the project, but they may be less efficient than other solutions.
🟦	1	Informational	1 Resolved	Informational errors are often recommendations to improve the style of the code or certain operations to fall within industry best practices. They usually do not affect the overall functioning of the code.

TABLE OF CONTENTS | VENUS - REWARDSDISTRIBUTOR

CODEBASE | VENUS - REWARDSDISTRIBUTOR

Repository

https://github.com/VenusProtocol/isolated-pools

Commit

base: 1b173dc1b0a7232a02c174559535ae18c3801b9e

update1: c4590657669993c901e64a5fe9837faeef5017d0

update2: 71a36e64cf1f32d81ba9bd728f230fe488b9190b

AUDIT SCOPE | VENUS - REWARDSDISTRIBUTOR

2 files audited ● 1 file with Mitigated findings ● 1 file without findings

ID	Repo	File	SHA256 Checksum
● RDR	VenusProtocol/isolated-pools	RewardsDistributor.sol	e628742f940d1c9e8c5058d3d2d9497cf244b2e229616fb8e481a44e56674821
● PLL	VenusProtocol/isolated-pools	PoolLens.sol	d723429b6dea59c2380d9abda3a449333b84e793da5bd692a74f8de3c2b8fdba

APPROACH & METHODS | VENUS - REWARDSDISTRIBUTOR

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

- Testing the smart contracts against both common and uncommon attack vectors;
- 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.

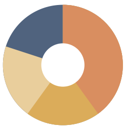
SUMMARY | VENUS - REWARDSDISTRIBUTOR

This audit concerns the changes implemented in the PR: https://github.com/VenusProtocol/isolated-pools/pull/257.

The main change introduced in this PR is to add functionality to stop supplier and borrower rewards at a given block. The contributor rewards are not affected by this and their functionality remains the same.

For more information that can be found in the previous audit see: https://skynet.certik.com/projects/venus. The previous audit can be found in the *Code Audit History* section under the title **Venus - Isolated Pools**.

FINDINGS | VENUS - REWARDSDISTRIBUTOR



5	0	2	1	1	1
Total Findings	Critical	Major	Medium	Minor	Informational

This report has been prepared to discover issues and vulnerabilities for Venus - RewardsDistributor. Through this audit, we have uncovered 5 issues ranging from different severity levels. Utilizing the techniques of Manual Review & Static Analysis to complement rigorous manual code reviews, we discovered the following findings:

ID	Title	Category	Severity	Status
RDR-04	**Centralized Control Of Contract Upgrade**	**Centralization**	**Major**	● **Mitigated**
RDR-05	**Centralization Risks**	**Centralization**	**Major**	● **Mitigated**
RDR-01	Potential Denial Of Service Attack	Logical Issue	Medium	● Resolved
RDR-02	Excess Rewards Given If Rewards Restarted	Logical Issue	Minor	● Resolved
RDR-03	Typos And Inconsistencies	Inconsistency	Informational	● Resolved

RDR-04 | CENTRALIZED CONTROL OF CONTRACT UPGRADE

Category	Severity	Location	Status
Centralization	● Major	RewardsDistributor.sol (baseRewards): 29	● Mitigated

Description

`RewardsDistributor` is an upgradeable contract. The owner can upgrade the contract without the community's commitment. If an attacker compromises the account, he can change the implementation of the contract and drain tokens from the contract as well as change the logic of the contract to return incorrect prices.

Recommendation

We recommend that the team make efforts to restrict access to the admin of the proxy contract. A strategy of combining a time-lock and a multi-signature (⅔, ⅗) wallet can be used to prevent a single point of failure due to a private key compromise. In addition, the team should be transparent and notify the community in advance whenever they plan to migrate to a new implementation contract.

Here are some feasible short-term and long-term suggestions that would mitigate the potential risk to a different level and suggestions that would permanently fully resolve the risk.

Short Term:

A combination of a time-lock and a multi signature (⅔, ⅗) wallet mitigate the risk by delaying the sensitive operation and avoiding a single point of key management failure.

- A time-lock with reasonable latency, such as 48 hours, for awareness of privileged operations;
  AND
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to a private key compromised;
  AND
- A medium/blog link for sharing the time-lock contract and multi-signers addresses information with the community.

For remediation and mitigated status, please provide the following information:

- Provide the deployed time-lock address.
- Provide the **gnosis** address with **ALL** the multi-signer addresses for the verification process.
- Provide a link to the **medium/blog** with all of the above information included.

Long Term:

A combination of a time-lock on the contract upgrade operation and a DAO for controlling the upgrade operation mitigate the contract upgrade risk by applying transparency and decentralization.

- A time-lock with reasonable latency, such as 48 hours, for community awareness of privileged operations;
  AND
- Introduction of a DAO, governance, or voting module to increase decentralization, transparency, and user involvement;
  AND
- A medium/blog link for sharing the time-lock contract, multi-signers addresses, and DAO information with the community.

For remediation and mitigated status, please provide the following information:

- Provide the deployed time-lock address.
- Provide the **gnosis** address with **ALL** the multi-signer addresses for the verification process.
- Provide a link to the **medium/blog** with all of the above information included.

Permanent:

Renouncing ownership of the `admin` account or removing the upgrade functionality can *fully* resolve the risk.

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

Note: we recommend the project team consider 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.

Alleviation

`[Venus, 07/07/2023]` : The ownership of these contracts will be transferred to 0x939bd8d64c0a9583a7dcea9933f7b21697ab6396, that is the Timelock contract used to execute the normal Venus Improvement Proposals (VIP).

For normal VIPs, the time config is: 24 hours voting + 48 hours delay before the execution.

So, this contracts will be upgraded only via a Normal VIP, involving the community in the process.

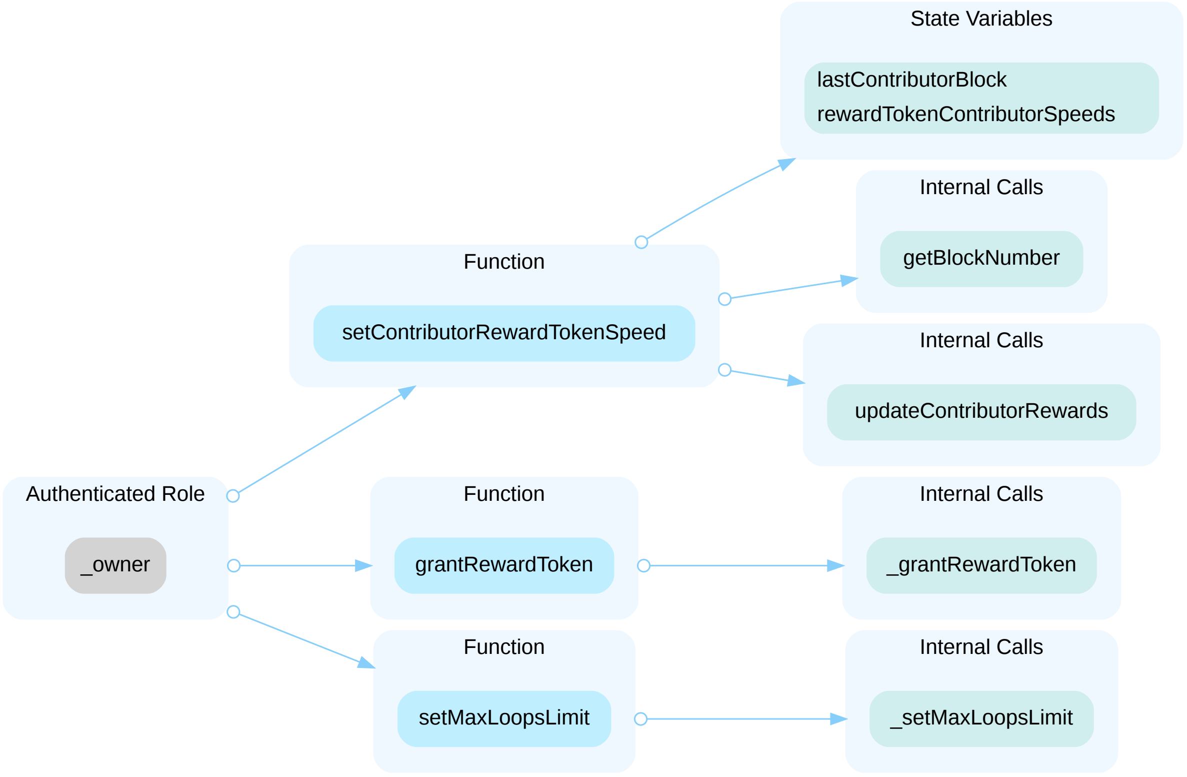
RDR-05 | CENTRALIZATION RISKS

Category	Severity	Location	Status
Centralization	● **Major**	**RewardsDistributor.sol (baseRewards): 155, 194, 202, 212, 218, 233, 256, 273, 287, 303**	● **Mitigated**

Description

In the contract `RewardsDistributor` the role `_owner` has authority over the functions shown in the diagram below. Any compromise to the `_owner` account may allow the hacker to take advantage of this authority and do the following:

- Change the contributor reward token speed to any value;
- Change the max loops, which limits that amount of `vToken` that `claimRewardToken()` can be called on at one time;
- Grant any amount of reward tokens, provided enough are held by the contract, to any user.



In the contract `RewardsDistributor` the role `DEFAULT_ADMIN_ROLE` of the Access Control Manager can grant addresses the privilege to call the following functions:

- `setRewardTokenSpeeds()`
- `setLastRewardingBlocks()`

Any compromise to the `DEFAULT_ADMIN_ROLE` or these privileged functions may allow the hacker to take advantage of this authority and do the following:

- change the reward token speed to any value;
- set the last rewarding blocks to either stop rewards early or to lengthen the amount of blocks rewards are given for.

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

`[Venus, 07/10/2023]` : The owner of the RewardsDistributor contracts will be 0x939bd8d64c0a9583a7dcea9933f7b21697ab6396, that is the Timelock contract used to execute the normal Venus Improvement Proposals (VIP).

For normal VIPs, the time config is: 24 hours voting + 48 hours delay before the execution.

So, only the community, via a VIP will be able to execute the mentioned protected functions.

We'll use the AccessControlManager (ACM) deployed at https://bscscan.com/address/0x4788629abc6cfca10f9f969efdeaa1cf70c23555

In this ACM, only 0x939bd8d64c0a9583a7dcea9933f7b21697ab6396 (Normal) has the DEFAULT_ADMIN_ROLE. And this contract is a Timelock contract used during the Venus Improvement Proposals.

Only the Normal Timelock (0x939bd8d64c0a9583a7dcea9933f7b21697ab6396) will be granted to execute the following functions in the RewardsDistributor contract:

- `setRewardTokenSpeeds()`
- `setLastRewardingBlocks()`

RDR-01 | POTENTIAL DENIAL OF SERVICE ATTACK

Category	Severity	Location	Status
Logical Issue	● Medium	RewardsDistributor.sol (baseRewards): 514~516, 518, 549~551, 553	● Resolved

Description

If the `supplyState.lastRewardingBlock` and `borrowState.lastRewardingBlock` are set to blocks before the current block, then calls to `updateRewardTokenBorrowIndex()` and `updateRewardTokenSupplyIndex()` will revert due to underflow, which can be used to perform a denial of service attack.

Scenario

Assume that for a market the `supplyState.lastRewardingBlock` is currently 300, the current block is 200, and `supplyState.block = 199`.

- A entity that has access to `setLastRewardingBlocks()` updates `supplyState.lastRewardingBlock` to be 100 for this market.
- Any other user then attempts an action that will call the `preMintHook()`, `preRedeemHook()`, `preSeizeHook()`, or `preTransferHook()` of the comptroller.
- These hooks will then call `rewardsDistributor.updateRewardTokenSupplyIndex` which will then perform the following logic:

```
        uint32 blockNumber = safe32(getBlockNumber(), "block number exceeds 32
bits");

        if (supplyState.lastRewardingBlock > 0 && blockNumber >
supplyState.lastRewardingBlock) {
            blockNumber = supplyState.lastRewardingBlock;
        }

        uint256 deltaBlocks = sub_(uint256(blockNumber),
uint256(supplyState.block));
```

- As the `supplyState.lastRewardingBlock` was set to be `100`, which is less than the current block number of `200`, `blockNumber` will be set to `100`.
- Thus `deltaBlocks` will take `100` minus the `supplyState.block = 199` and revert due to underflow.

This demonstrates how a user with access to `setLastRewardingBlocks()` can perform a denial of service. For example this could be done to prevent accounts from becoming liquidated, which could cause the protocol to incur bad debt.

Similarly this can be done for `borrowState.lastRewardingBlock` to perform a denial of service on actions that call the comptrollers `preBorrowHook()` or `preRepayHook()` .

Recommendation

We recommend checking that the input `supplyLastRewardingBlock` and `borrowLastRewardingBlock` are greater than the current block in the `_setLastRewardingBlock()` function.

Alleviation

`[CertiK, 07/07/2023]` : The client made the recommended changes in commit: c4590657669993c901e64a5fe9837faeef5017d0.

RDR-02 | EXCESS REWARDS GIVEN IF REWARDS RESTARTED

Category	Severity	Location	Status
Logical Issue	● Minor	RewardsDistributor.sol (baseRewards): 514~518	● Resolved

Description

If the last reward blocks for a market are reached and the last reward blocks are updated to restart giving rewards, then any user that remained a borrower or supplier in the market will receive rewards for the time the rewards were not active.

Scenario

For simplicity assume that a user has been a borrower in a market since block 0, rewards from the market have been active since block 0, the `borrowState.lastRewardingBlock` of the market is 100, the market is active so that the rewards are updated frequently, the reward token borrow speed remains a nonzero constant, and the current block is 150.

- Rewards for this market want to be restarted for another 100 blocks, so in block 150 the `borrowState.lastRewardingBlock` is set to 250 by calling `setLastRewardingBlocks()` .
- Assume no actions update the reward token borrow index until the user then calls `claimRewardToken()` in block 151, which will call `_updateRewardTokenBorrowIndex()` .
- `borrowState.block` will be `100` as it is only ever set to the `blockNumber` , which is set to the `borrowState.lastRewardingBlock` if the current block exceeds it.
- However, for this block `borrowState.lastRewardingBlock = 250` so that `blockNumber` is the current block number of 151.
- Thus `deltaBlocks = 151 - 100 = 51` , updating the reward token borrow index to allocate rewards for 51 blocks.
- As the reward token speed remained a nonzero constant for all blocks, this allocates rewards for the 50 blocks that no rewards were active allowing the user to withdraw rewards for 151 blocks, when the rewards should only be for 101 blocks.

Recommendation

We recommend ensuring that if the `lastRewardingBlock` is reached and rewards are to be restarted, that no rewards will be given for the blocks between the `lastRewardingBlock` and the block the `lastRewardingBlock` is updated to restart rewards.

Alleviation

`[CertiK, 07/10/2023]` : The client added checks to prevent rewards from being restarted if the `lastRewardingBlock` had been set and reached in commits:

- c4590657669993c901e64a5fe9837faeef5017d0;
- 71a36e64cf1f32d81ba9bd728f230fe488b9190b.

RDR-03 | TYPOS AND INCONSISTENCIES

Category	Severity	Location	Status
Inconsistency	● Informational	RewardsDistributor.sol (baseRewards): 247	● Resolved

Description

In the comments above `setLastRewardingBlocks()` , it states "The markets whose REWARD TOKEN rewarding block to update". When it is more accurate to say "The markets whose REWARD TOKEN last rewarding block to update".

Recommendation

We recommend fixing the typos/inconsistencies mentioned above.

Alleviation

`[CertiK, 07/07/2023]` : The client made the recommended changes in commit: 2e0f01d209a6b252200957ca75023e001731dba9.

APPENDIX | VENUS - REWARDSDISTRIBUTOR

Finding Categories

Categories	Description
Inconsistency	Inconsistency findings refer to different parts of code that are not consistent or code that does not behave according to its specification.
Logical Issue	Logical Issue findings indicate general implementation issues related to the program logic.
Centralization	Centralization findings detail the design choices of designating privileged roles or other centralized controls over the code.

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 | CERTIK

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.

CertiK | **Securing** the **Web3** World

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.