CERTIK

Security Assessment

AlgoFi - Governance

CertiK Verified on Sept 29th, 2022

CertiK Verified on Sept 29th, 2022

AlgoFi - Governance

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

Executive Summary

TYPES	ECOSYSTEM	METHODS
Governance	Algorand	Manual Review
LANGUAGE	**TIMELINE**	**KEY COMPONENTS**
Python	Delivered on 09/29/2022	N/A

CODEBASE
https://github.com/Algofiorg/algofi-governance-v2
...View All

COMMITS
3f3c4ddad3bbbacf32412ba2b834a0e2d9567af4
...View All

Vulnerability Summary

10	6	0	0	4	0	0
Total Findings	Resolved	Mitigated	Partially Resolved	Acknowledged	Declined	Unresolved

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 Acknowledged	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.
6	Informational	4 Resolved, 2 Acknowledged	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 | ALGOFI - GOVERNANCE

CODEBASE | ALGOFI - GOVERNANCE

Repository

https://github.com/Algofiorg/algofi-governance-v2

Commit

3f3c4ddad3bbbacf32412ba2b834a0e2d9567af4

AUDIT SCOPE | ALGOFI - GOVERNANCE

7 files audited ● 4 files with Acknowledged findings ● 2 files with Resolved findings ● 1 file without findings

ID	File	SHA256 Checksum
CKP	contracts/admin_contract.py	c50c5396e107c3da902abb1715f7b5a985a48b3c76aa2b524faa2ca77c3c39e3
GLO	contracts/global_emitter.py	33b4df526a75f50230ce66c1057eb40558744a9dc7e1a09081892ef2e9c10034
REW	contracts/rewards_manager.py	62880d5333d127fc87b84e67b35b24d184b15c4c500e399d647b7c776d1dca85
STA	contracts/staking_contract.py	7ccce1b824ef3f77f97af2568579fcefe42d7af0bcd0a9dd3fe540b6ff224487
PRP	contracts/proposal_factory.py	bceea5991532ee1cf61f703b964128619b7e9a0d3e93019a27e301a6c60a61ab
VOT	contracts/voting_escrow.py	4662f4e2a43c9bcce90882b3eefe02af84894c00b7c0c107072991cdd2396c0a
PRO	contracts/proposal.py	7e8805e2d9e116bd710bdc8f2bd480194d9431367665d4cb37d03145c24f446d

APPROACH & METHODS | ALGOFI - GOVERNANCE

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

FINDINGS | ALGOFI - GOVERNANCE

10	0	2	1	1	6
Total Findings	Critical	Major	Medium	Minor	Informational

This report has been prepared to discover issues and vulnerabilities for AlgoFi - Governance. Through this audit, we have uncovered 10 issues ranging from different severity levels. Utilizing Static Analysis techniques to complement rigorous manual code reviews, we discovered the following findings:

ID	Title	Category	Severity	Status
GLOBAL-01	**Centralization Related Risks**	**Centralization / Privilege**	**Major**	**Acknowledged**
CKP-01	**Privileged Addresses Can Claim Rewards**	**Centralization / Privilege, Volatile Code**	**Major**	**Acknowledged**
CKP-02	Proposals Validation Can Only Pass Or Revert	Business Model	Minor	Resolved
STA-01	Wrong Initialization On User States	Volatile Code	Medium	Resolved
GLOBAL-03	Unused Imports, Functions And Variables	Coding Style	Informational	Resolved
CKP-03	Typos In Comments And Codes	Coding Style	Informational	Resolved
CKP-04	Delegated Voting Power Would Not Be Removed When Undelegated	Business Model	Informational	Acknowledged
GLO-01	Sending Zero Assets In `update_rewards_manager_epoch`	Volatile Code	Informational	Acknowledged
REW-01	Inconsistency Between Documentation And Code	Inconsistency	Informational	Resolved
STA-02	Inconsistency Between Comment And Code	Coding Style	Informational	Resolved

GLOBAL-01 | CENTRALIZATION RELATED RISKS

Category	Severity	Location	Status
Centralization / Privilege	Major		Acknowledged

Description

In the contract admin_contract.py, the role `emergency_dao_address` has authority over the functions:

- `on_set_executed()`
- `on_cancel_proposal()`
- `on_set_quorum_value()`
- `on_set_super_majority()`
- `on_fast_track_proposal()`
- `on_set_voting_escrow_app_id()`
- `on_schedule_contract_update()`
- `on_increase_contract_update_delay()`
- `on_set_proposal_duration()`
- `on_set_proposal_factory_address()`
- `on_set_proposal_execution_delay()`

In the contract global_emitter.py, the role `emergency_dao_address` has authority over the functions:

- `on_schedule_contract_update()`
- `on_increase_contract_update_delay()`
- `on_update_dao_address()`
- `on_update_emergency_dao_address()`
- `on_update_rewards_manager_app_id()`
- `on_opt_in_gov_token()`
- `on_start_funding()`
- `on_halt_funding()`
- `on_restart_funding()`

In the contract proposal_factory.py, the role `emergency_dao_address` has authority over the functions:

- `on_schedule_contract_update()`
- `on_increase_contract_update_delay()`
- `on_set_proposal_template()`

- `on_set_voting_escrow_app_id()`
- `on_set_admin_app_id()`
- `on_set_minimum_ve_bank_to_propose()`
- `on_update_dao_address()`
- `on_update_emergency_dao_address()`

In the contract rewards_manager.py, the role `emergency_dao_address` has authority over the functions:

- `on_update_dao_address()`
- `on_update_emergency_dao_address()`
- `on_schedule_contract_update()`
- `on_increase_contract_update_delay()`
- `on_set_epoch_expiration_delay()`
- `on_stage_contract_opt_in()`
- `on_set_emitter_app_id()`
- `on_set_voting_escrow_app_id()`
- `on_set_gov_token_id()`
- `on_reclaim_rewards()`

In the contract staking_contract.py, the role `emergency_dao_address` has authority over the functions:

- `on_update_dao_address()`
- `on_update_emergency_dao_address()`
- `on_initialize_rewards_escrow_account()`
- `on_schedule_contract_update()`
- `on_increase_contract_update_delay()`
- `on_set_rewards_manager_app_id()`
- `on_set_voting_escrow_app_id()`
- `on_set_rewards_program()`
- `on_update_rewards_per_second()`
- `on_opt_into_asset()`
- `on_opt_into_rewards_manager()`
- `on_reclaim_rewards_assets()`

In the contract voting_escrow.py, the role `emergency_dao_address` has authority over the functions:

- `on_update_dao_address()`
- `on_update_emergency_dao_address()`
- `on_schedule_contract_update()`
- `on_increase_contract_update_delay()`

- `on_set_gov_token_id()`
- `on_set_rewards_manager_app_id()`
- `on_set_admin_contract_app_id()`

Any compromise to the `emergency_dao_address` 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 different levels:

Multi sign (⅔, ⅗) *mitigate* and avoids a single point of key management failure.

- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key being compromised;
  AND
- A medium/blog link for sharing the multi-signers addresses 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 overprivileged functionality.

Noted: The project team shall make a decision based on the current state of their project, timeline, and project resources.

Alleviation

[Algofi Team]:

This is expected. The emergency dao is a multisig that is at least 2/3. It is only used in emergency situations. Otherwise the dao is the primary means of modifying the protocol.

CKP-01 | PRIVILEGED ADDRESSES CAN CLAIM REWARDS

Category	Severity	Location	Status
Centralization / Privilege, Volatile Code	**Major**	**contracts/rewards_manager.py: 240~241, 489~490; contracts/staking_contract.py: 466~467, 633~634**	**Acknowledged**

Description

In contracts rewards_manager.py and staking_contract.py, the two functions, `on_reclaim_rewards` and `on_reclaim_rewards_assets`, can be called by `dao_address` or `emergency_dao_address`. These two functions allow the admin addresses to directly withdraw rewards without any limits.

Recommendation

Recommend adding a max reclaim threshold as the max amount can be withdrawn within one transaction, and properly testing and tuning the value of it. Also, please refer to the recommendation of GLOBAL-01 Centralization Related Risks.

Alleviation

[Algofi Team]:

Since this is desired behavior, no change will be made.

This is expected behavior. The emergency dao (a multisig) will only be used in emergencies. The dao will be able to reclaim rewards if it chooses to.

CKP-02 | PROPOSALS VALIDATION CAN ONLY PASS OR REVERT

Category	Severity	Location	Status
Business Model	Minor	contracts/admin_contract.py: 687~688	Resolved

Description

The function `on_validate` calls `verify_vote_passed` and sets the variable `proposal.execution_time`. The `verify_vote_passed` function is a checker function, containing assertions of voting status, quorum and approval percentage. It will revert the transaction when a proposal does not pass via voting. In this scenario, a proposal can be either passed or never passed. There is no status like “rejected”.

Recommendation

There are no security concern in this finding. Just would like to learn if our understanding is correct and to make sure if it is allowed that a proposal does not have a "rejected" status.

We would like to learn if this is an intended design. If so, recommend properly documenting this behavior and setting up a periodical proposal cancelling workflow to avoid anachronistic proposals being approved. Otherwise, recommend setting a proposal period and if a proposal is not approved during a valid proposal period, the proposal would have a status like failed, rejected, etc.

Alleviation

[Algofi Team]:

Addressed: https://github.com/Algofiorg/algofi-governance-v2/pull/57/files

Your understanding is correct, but we will add a boolean to show if a vote has passed for clarity.

STA-01 | WRONG INITIALIZATION ON USER STATES

Category	Severity	Location	Status
Volatile Code	Medium	contracts/staking_contract.py: 487~489	Resolved

Description

In the function `on_user_opt_in`, there are two variables that have value assigned, `self.total_staked` and `self.scaled_total_staked`. It seems it should be fields of a `StakingUser` object (defined in L23,24) instead of fields of a `StakingContract` object (defined in L219, 220).

Recommendation

Recommend assigning the value to the fields of the correct class for the user opt in function call.

Alleviation

[Algofi Team]:

This has been resolved. See PR: https://github.com/Algofiorg/algofi-governance-v2/pull/53/files

GLOBAL-03 | UNUSED IMPORTS, FUNCTIONS AND VARIABLES

Category	Severity	Location	Status
Coding Style	Informational		Resolved

Description

There are unused imports, functions, and variables in the following locations:

- `proposal_factory.on_create_proposal()`:
  - `proposal_app_id_scratch`
  - `proposal_app_address`
- `voting_escrow.approval_program()`:
  - `is_delete_application`
- `config.py`:
  - `validate_token_received_by_key()`
  - `verify_txn_is_sending_algos_to_contract()`
  - `verify_txn_is_named_opt_in_application_call()`
  - `verify_txn_application_arg()`
  - `verify_txn_application()`
  - imports:

```
from enum import Enum
from algosdk.encoding import decode_address, encode_address
from base64 import b64encode, b64decode
```

Recommendation

Recommend removing unused codes for open source purpose.

Alleviation

[Algofi Team]:

Fixed: https://github.com/Algofiorg/algofi-governance-v2/pull/54/files

We will remove the unused functions, variables, and imports.

CKP-03 | TYPOS IN COMMENTS AND CODES

Category	Severity	Location	Status
Coding Style	Informational	contracts/admin_contract.py: 296~297, 392~393, 745~746; contracts/global_emitter.py: 287~288; contracts/proposal_factory.py: 343~344; contracts/rewards_manager.py: 316~317, 326~327, 343~344, 416~417, 471~472, 510~511; contracts/staking_contract.py: 182~183, 434~435, 544~545, 687~688; contracts/voting_escrow.py: 278~279, 400~401, 440~441	Resolved

Description

There are several typos in the contracts, please see the above to find the locations, and the word with typos are listed here:

- permissionless
- permissionles
- permisionless
- prorata
- initilize
- neccessary
- updat
- composability
- emergecy
- recieved

Recommendation

Recommend correcting all of the typos in the contracts to provide better readability for open source purposes.

Alleviation

[Algofi Team]:

Fixed: https://github.com/Algofiorg/algofi-governance-v2/pull/55/files

CKP-04 DELEGATED VOTING POWER WOULD NOT BE REMOVED WHEN UNDELEGATED

Category	Severity	Location	Status
Business Model	Informational	contracts/admin_contract.py: 673~674	Acknowledged

Description

In the function `on_delegated_vote`, if a `target_user` has delegated, the `vote` function would be called. Then the `vote` function would `vote_on_proposal_contract`, increment `num_proposals_opted_into` and increment either the `votes_against` or `votes_for`.

However, in the function `on_undelegate`, the voting power voted via `on_delegated_vote` is not removed. It would potentially cause the mis-calculation in `verify_vote_passed`.

Recommendation

Recommend properly documenting this mechanism and gaining sufficient community consensus, given that this is intended by design.

Alleviation

[Algofi Team]:

Delegation is forward looking only. Once a vote has been placed (either personal or delegated vote) it cannot be undone. There is no issue here.

GLO-01 | SENDING ZERO ASSETS IN `update_rewards_manager_epoch`

Category	Severity	Location	Status
Volatile Code	Informational	contracts/global_emitter.py: 88~89	Acknowledged

Description

The function `update_rewards_manager_epoch` can send assets to the reward manager and invoke `reward_manager.on_begin_next_epoch`, and it is only called in the function `on_fund`, where it is possible to have a zero fund amount to be sent to the reward manager, when funding is halted or missed.

Recommendation

Recommend adding a condition in `update_rewards_manager_epoch` to skip the zero amount asset transferring.

Alleviation

[Algofi Team]:

This is expected behavior. When the GE is paused, we need to communicate to the rewards_manager to distribute 0 assets to the opted into staking / market contracts. The RM expects an asset transfer transaction so we must send zero-value asset transfer transaction from the GE in on_fund.

REW-01 INCONSISTENCY BETWEEN DOCUMENTATION AND CODE

Category	Severity	Location	Status
Inconsistency	Informational	contracts/rewards_manager.py: 397~398	Resolved

Description

In the documentation, the group size of the function `on_vote` is `1`. However, in the code, since `PREVIOUS_TRANSACTION` is used, the group size should be at least `2`.

Recommendation

Recommending reviewing the documentation and fixing the wrong parameters to keep consistency between code implementations and documentations.

Alleviation

[Algofi Team]:

Fixed: https://algofi.gitbook.io/algofi-smart-contract-api/smart-contract-apis/algofi-governance-api/rewards-manager/user/vote

We will fix this inconsistency. The code is accurate and the smart contract docs were generated only for the purposes of this audit.

STA-02 | INCONSISTENCY BETWEEN COMMENT AND CODE

Category	Severity	Location	Status
Coding Style	Informational	contracts/staking_contract.py: 73~74, 574~575	Resolved

Description

There are some inconsistent comments:

- On L73, `update_scaled_total_staked`, it seems to be "**user_total_staked * 40% (stake_component)** + boost_multiplier * global_total_staked * 60% (boost_component)", instead of "user_total_staked * 40% + **boost_multiplier (stake_component)** * global_total_staked * 60% (boost_component)"
- On L574, `on_unstake`, it seems to be "# **decrement** global total staked", instead of "# **increment** global total staked"

Recommendation

Recommending updating the comments to keep consistency.

Alleviation

[Algofi Team]:

Fixed: https://github.com/Algofiorg/algofi-governance-v2/pull/56/files

We will resolve the code / comment inconsistencies.

OPTIMIZATIONS | ALGOFI - GOVERNANCE

ID	Title	Category	Severity	Status
GLOBAL-02	Tests Not Runnable	Coding Style	Optimization	Resolved

GLOBAL-02 | TESTS NOT RUNNABLE

Category	Severity	Location	Status
Coding Style	Optimization		Resolved

Description

The current tests are not runnable. We noticed there are some dependent packages/modules imported but not found in the target branch. e.g. `offchain_utils` in `gov/test/admin_contract_test.py`

Recommendation

Recommend properly testing the various program use cases with unit-tests and integration tests.

Alleviation

[CertiK]:

Access to the utility repo is shared and confirmed.

[Algofi Team]:

This is expected. Our test suite utilizes features from a unified utilities repo. We can provide access on request.

APPENDIX | ALGOFI - GOVERNANCE

Finding Categories

Categories	Description
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.
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.
Inconsistency	Inconsistency findings refer to functions that should seemingly behave similarly yet contain different code, such as a constructor assignment imposing different require statements on the input variables than a setter function.

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.