CERTIK

Security Assessment

Sollong

CertiK Assessed on Jul 30th, 2024

CertiK Assessed on Jul 30th, 2024

Sollong

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

Executive Summary

TYPES	ECOSYSTEM	METHODS
DeFi	Solana (SOL)	Manual Review, Static Analysis
LANGUAGE	**TIMELINE**	**KEY COMPONENTS**
Rust	Delivered on 07/30/2024	N/A

CODEBASE
sollongcode
View All in Codebase Page

COMMITS
d3ad8d795cc5e7df37261e33261906fc07a925a1
View All in Codebase Page

Vulnerability Summary

10	9	0	0	1	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	1 Resolved, 1 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.
2	Minor	2 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.
5	Informational	5 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 | SOLLONG

CODEBASE | SOLLONG

Repository

sollongcode

Commit

d3ad8d795cc5e7df37261e33261906fc07a925a1

AUDIT SCOPE | SOLLONG

15 files audited ● 7 files with Acknowledged findings ● 5 files with Resolved findings ● 3 files without findings

ID	Repo	File	SHA256 Checksum
CRE	officaldevi/sollongcode	programs/sollong-preipo/src/instructions/create_financial_account.rs	31a52e5558446a2ade1b57cac249b44d91d0e6aa0e7f82109ceb571d47e74586
NEW	officaldevi/sollongcode	programs/sollong-preipo/src/instructions/new_round.rs	0a911cef68be9fb037006505a9a926e86bf2515174b936dc3c6f125cf36f1a6f
OWN	officaldevi/sollongcode	programs/sollong-preipo/src/instructions/owner.rs	50b02cb34d850fd67ba3705280a90d2691ffc37a39e68dc150b3b98acdb610d4
OWE	officaldevi/sollongcode	programs/sollong-preipo/src/instructions/owner_withdraw.rs	441009d89901818722b71aeb8e235048e81bcb5fcdadbe8e341466ad58e6509e
SET	officaldevi/sollongcode	programs/sollong-preipo/src/instructions/set_buy_share_limit.rs	fc1d5b25b26b0b093e3bb25521f8f5b6cad06b4c2c9c2b39c8c0948e4674b402
SEI	officaldevi/sollongcode	programs/sollong-preipo/src/instructions/set_time.rs	4309690eb74607627ef92a35013314275fed03f0d5cbcdfbecd188db68bc6088
WHI	officaldevi/sollongcode	programs/sollong-preipo/src/instructions/whitelist.rs	da8062577a597907a416835facd693c1ce693934992ccc5b6934815bc49e46e3
BUY	officaldevi/sollongcode	programs/sollong-preipo/src/instructions/buy.rs	28c5e3d76a2d972617f7fe3ca7f4efd4d00b5422537ecad7144d56c3c3149d99
CLO	officaldevi/sollongcode	programs/sollong-preipo/src/instructions/close_account.rs	edd7b0bd6fed131a202274a75f0dd2593f2d76c0d43ae9ad2ea7f9e329173139
CRA	officaldevi/sollongcode	programs/sollong-preipo/src/instructions/create_user_account.rs	e2900872136ff2dd83c04b4ed41977c5ed0691271d506adadb36053cf2515549
LIB	officaldevi/sollongcode	programs/sollong-preipo/src/lib.rs	5c5f23b9f6ef753f1fa8628fc77ba849513e4c87945e7ca7e565861db4334a1b
STA	officaldevi/sollongcode	programs/sollong-preipo/src/state.rs	5daded9eb709221ccf0fd69a3a20e303347ac376169dd34edae513f738abd288

ID	Repo	File	SHA256 Checksum
INI	officaldevi/sollongcode	programs/sollong-preipo/src/instructions/init.rs	801cbc4b9272c57af413ece3cfaaeb06af8388222a264270a79ade8a486dbab1
ERR	officaldevi/sollongcode	programs/sollong-preipo/src/errors.rs	4911835eeb161283352daf6875d3fe5852c1c1a5b100bc716d8364fdfde10c81
INT	officaldevi/sollongcode	programs/sollong-preipo/src/instructions.rs	e12d5f7257d54832cdb4833133011fbb7efd4c2f146a604cf8e27c4c00d3ef0e

APPROACH & METHODS | SOLLONG

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

SYSTEM OVERVIEW | SOLLONG

The Sollong-preipo program implements a mechanism to define pool of shares and sell them in multiple rounds. Users can buy such shares at a predetermined price in a defined time window. Participating accounts should at least buy a minimum quantity per round up to a defined maximum. Rounds may also include a whitelist in the form of a Merkle Tree in order to only allow a set of users to access the private sale.

Accounts

Solana programs store their data in on-chain accounts and their management is a key responsibilities of the implementation. The following table summarizes the accounts involved in the Sollang-preipo functionalities.

Account	Type	Description
Metadata	PDA - Global to the program and unique	Stores global program state like shares owner and current round
Financial	PDA - One per each pair (round index, financial index)	Tracks funds obtained from shares
UserData	PDA - One per each user in each round	Tracks shares bought by each user
RoundStock	PDA - One per round	Stores all the round information

REVIEW NOTES | SOLLONG

The following are consideration by the audit team that complement the security assessment.

Documentation and Tests

The codebase does not include any kind of documentation, so during the audit phase the implementation was assumed correct as-is since it was not possible to match it with any kind of specification.

Additionally, there are not any integration and unit test to assess the correct code behavior in expected and unexpected scenarios. Including extensive unit and integration tests helps validating the implemented logic and its adherence to business requirements, while minimizing the possibility of introducing bugs on already existing functionalities across different development iterations. It is strongly suggested to test the code before deploying it to production environments.

FINDINGS | SOLLONG

10	0	2	1	2	5
Total Findings	Critical	Major	Medium	Minor	Informational

This report has been prepared to discover issues and vulnerabilities for Sollong. Through this audit, we have uncovered 10 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
GLOBAL-01	**Program Upgrade Centralization Risk**	**Centralization**	**Major**	**Resolved**
INS-01	**Centralization Related Risks On Metadata** `owner`	**Centralization**	**Major**	**Acknowledged**
CLO-01	Missing Validation And Access Control On Account Close	Logical Issue	Medium	Resolved
CRE-02	Uninitialized Financial Round Index	Logical Issue	Minor	Resolved
WHI-01	Missing Validation Of Whitelist Root Presence	Logical Issue	Minor	Resolved
BUY-01	Mutiple Financial Accounts	Logical Issue	Informational	Resolved
BUY-02	Generic Error On Multiple Conditions	Coding Style	Informational	Resolved
BUY-03	Unrelated Error On Non-Whitelisted `buy` Calls	Inconsistency, Coding Style	Informational	Resolved
OWE-01	Undocumented Unchecked Account	Coding Style	Informational	Resolved
SRC-01	Typos In Codebase	Coding Style	Informational	Resolved

GLOBAL-01 | PROGRAM UPGRADE CENTRALIZATION RISK

Category	Severity	Location	Status
Centralization	**Major**		**Resolved**

Description

A Solana program can be deployed on the mainnet as:

- final: the code cannot be updated.
- upgradable: `BPFLoaderUpgradeable` is the program owner and an *upgrade authority*, a custom account, can upgrade the program code.

In case the `Sollong-preipo` program is deployed as upgradable, the upgrade authority has the privilege to update the implementation of the programs at his/her will. Any compromise to the upgrade authority account may allow a hacker to take advantage of this authority and replace the implementation of the program and therefore execute any code on the program data and funds.

Recommendation

Our recommendation depends on the team's intentions that we invite to clarify.

We recommend that the team make efforts to restrict access to the private key of the `upgrade authority` account. 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:

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 of privileged operations;
  AND
- 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 timelock 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 and multi-signers addresses, and DAO information with the public audience.

For example, the upgrades could be managed by the guardian network with an additional reasonable time-lock latency.

Permanent:

Deploying the programs as `final` can fully resolve the risk.

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.

Preliminary report only

Please provide the following information:

- Provide the account 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

Alleviation

[`CertiK` , 07/25/2024]: The team acknowledged the finding and solved the issue by introducing the renounce to the upgrade functionality in the program initialization logic in commits 74210bf74c7358ddea0b45d5833281317f9879b1 and c254ad0873a8a3efc66d25fba282de9beba6f67f.

INS-01 | CENTRALIZATION RELATED RISKS ON METADATA `owner`

Category	Severity	Location	Status
Centralization	Major	programs/sollong-preipo/src/instructions/create_financial_account.rs: 20~21; programs/sollong-preipo/src/instructions/new_round.rs: 17~18; programs/sollong-preipo/src/instructions/owner.rs: 29~30; programs/sollong-preipo/src/instructions/owner_withdraw.rs: 22~23; programs/sollong-preipo/src/instructions/set_buy_share_limit.rs: 9~10; programs/sollong-preipo/src/instructions/set_time.rs: 11~12; programs/sollong-preipo/src/instructions/whitelist.rs: 10~11, 22~23	Acknowledged

Description

In the `Sollong-preipo` program the `owner` account specified in the `Metadata` data has authority over the following functions:

- `new_round()`
- `create_financial_account()`
- `owner_withdraw()`
- `change_owner()`
- `set_buy_share_limit()`
- `set_time()`
- `set_merkel_tree_hash()`
- `set_white_list_status()`

Any compromise to the `owner` account may allow the hacker to take advantage of this authority and:

- issue new illegitimate rounds;
- initialize illegitimate financial rounds;
- withdraw funds originated from the share sale;
- change owner to an arbitrary account under its exclusive control;
- alter rounds configuration by modifying buy limits and timing;
- alter round whitelists by including and excluding any account;

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

[`CertiK`, 07/24/2024]: The team introduced a timing mechanism which allows the owner to modify the pre-ipo parameter only up to 1 day after program initialization. Then, public program functionalities can be activated and funds are planned to

be transferred to a multi-signature wallet under the project team control to mitigate key compromise problems. Changes are implemented in commit 74210bf74c7358ddea0b45d5833281317f9879b1.

Finding will be updated to the `Mitigated` status once on-chain information are available and provided by the team.

CLO-01 | MISSING VALIDATION AND ACCESS CONTROL ON ACCOUNT CLOSE

Category	Severity	Location	Status
Logical Issue	Medium	programs/sollong-preipo/src/instructions/close_account.rs: 14~18	Resolved

Description

The `close_account` instruction allows to close the `RoundStock` account in order to get back the lamports locked for its storage. However, such instruction does have neither any access control, nor any validation that the pointed `RoundStock` is still active or not (e.g. `end_timestamp` was reached, or all shares sold). In this way any user can stop the shares sale by uninitializing the respective `RoundStock` account.

Additionally the account receiving the lamports back can be arbitrarily specified, allowing any malicious user to steal that amount.

Recommendation

We recommend including validation checks that the `RoundStock` account to be closed is not an active one. Additionally, if the `to` account is left arbitrary, then an access control on the instruction caller should be provided, too, to avoid anyone collecting such lamports instead of the project team.

Alleviation

[`Sollong Team` , 07/23/2024]: The team acknowledged the finding and fixed the security issue by removing the `close_account` method from the callable instructions in commit 523ed899ceac061cb6f47a169c8f40a946213a8d.

CRE-02 | UNINITIALIZED FINANCIAL ROUND INDEX

Category	Severity	Location	Status
Logical Issue	Minor	programs/sollong-preipo/src/instructions/create_financial_account.rs: 25 ~26	Resolved

Description

Data in the `Financial` struct is initialized in the `create_financial_account` instruction, but its `index` property is not assigned any value. On contrary, the `round_index` field is assigned twice to two different values, which makes the first of the two useless.

Recommendation

We recommend reviewing the implementation and fixing the double assignment since the second of the two suggests a logic error, probably confusing `financial.index` with `financial.round_index`.

Alleviation

[`Sollong Team` , 07/23/2024]: The team acknowledged the finding and resolved the issue since the financial index is ensured to be 0 in commit 523ed899ceac061cb6f47a169c8f40a946213a8d.

WHI-01 | MISSING VALIDATION OF WHITELIST ROOT PRESENCE

Category	Severity	Location	Status
Logical Issue	Minor	programs/sollong-preipo/src/instructions/whitelist.rs: 18~27	Resolved

Description

The `set_white_list_status()` instruction allows the project owner to enable or disable the Merkle root based whitelist mechanism. However, in the case in which the `is_enabled` parameter is set to `true`, there is not any check that a Merkle root is actually present in the `RoundStock` data, so, creating an unwanted program state.

Recommendation

We recommend checking that a whitelist root is already set when the `is_enabled` parameter is set to `true`. Alternatively, if the `set_white_list_status()` operation is only meant to disable the whitelist mechanism, then the `is_enabled` parameter can be omitted and `whitelist_enabled` could be directly set to `false`.

Alleviation

[`CertiK` , 07/23/2024]: The team heeded the advice and resolved the finding by checking the presence of a non-null Merkle root in commit ac1e29907bdbeb86ae0c01057d7740799b9acf6e.

BUY-01 | MUTIPLE FINANCIAL ACCOUNTS

Category	Severity	Location	Status
Logical Issue	Informational	programs/sollong-preipo/src/instructions/buy.rs: 21~22	Resolved

Description

The `Financial` account is the recipient of users funds when a number of shares is bought. The program design allows the creation of multiple `Financial` accounts which are randomly chosen as funds destination according to the last byte of the user key performing the `buy` instruction.

Recommendation

We invite the developer team to clarify the rationale behind such a design with multiple random recipient accounts rather than having a single unique one per round.

Alleviation

[`Sollong Team` , 07/23/2024]: The team removed the funds split over multiple financial accounts in commit 523ed899ceac061cb6f47a169c8f40a946213a8d.

BUY-02 | GENERIC ERROR ON MULTIPLE CONDITIONS

Category	Severity	Location	Status
Coding Style	Informational	programs/sollong-preipo/src/instructions/buy.rs: 35~39, 81~85	Resolved

Description

The pointed `require` statement checks the following 4 conditions:

- User is buying a non-zero amount;
- Total user shares are beyond the minimum;
- Total user shares are below the maximum;
- Round has enough remaining shares. Despite checking different facts, the same error is raised if any of the mentioned validation fails. Such approach is discouraged as it lowers the code clarity and makes more difficult the interation with the program as the raised error ambigously represents 1 out of 4 possible error condition without any further details.

Recommendation

We recommend providing meaningful errors for each of the listed validations.

Alleviation

[`Sollong Team` , 07/23/2024]: The team acknowledged the finding and heeded the advice by providing dedicated errors for each condition in commit 523ed899ceac061cb6f47a169c8f40a946213a8d

BUY-03 | UNRELATED ERROR ON NON-WHITELISTED `buy` CALLS

Category	Severity	Location	Status
Inconsistency, Coding Style	Informational	programs/sollong-preipo/src/instructions/buy.rs: 31	Resolved

Description

The `FunctionCallError` error reports a message inviting non-whitelisted users to use the `buy` instruction instead of the `buy_from_whitelist` one. However, the same error is also used for the `buy` instruction itself when `buy_from_whitelist` is supposed to be used in its place. So, the error message in the `buy` case does not reflect the actual error condition.

Recommendation

We recommend providing a dedicated error for the `buy` instruction when it is called on a whitelist round.

Alleviation

[`Sollong Team` , 07/23/2024]: The team acknowledged the finding and heeded the advice by providing dedicated errors for each instruction in commit 523ed899ceac061cb6f47a169c8f40a946213a8d

OWE-01 | UNDOCUMENTED UNCHECKED ACCOUNT

Category	Severity	Location	Status
Coding Style	Informational	programs/sollong-preipo/src/instructions/owner_withdraw.rs: 14~15	Resolved

Description

In Anchor, when accounts are not validated through the `Account` struct, a dedicated comment should be included to describe the field and the reason why checks were disabled using the `/// CHECK:` expression. If the practice is not followed, the Anchor build command issues a compilation error.

Recommendation

We recommend describing the `to` field with the suggested Anchor expression to document its usage.

Alleviation

[`Sollong Team` , 07/23/2024]: The team acknowledged the finding and heeded the advice by including the `/// CHECK:` comment, as required by Anchor, in commit 523ed899ceac061cb6f47a169c8f40a946213a8d

SRC-01 | TYPOS IN CODEBASE

Category	Severity	Location	Status
Coding Style	Informational	programs/sollong-preipo/src/instructions/buy.rs: 77; programs/sollong-preipo/src/instructions/whitelist.rs: 6, 12; programs/sollong-preipo/src/lib.rs: 35, 36; programs/sollong-preipo/src/state.rs: 29	Resolved

Description

The pointed locations present the following typos:

- merkel is supposed to be merkle

Recommendation

We recommend fixing the mentioned typos to enhance code clarity.

Alleviation

[`Sollong Team`, 07/23/2024]: The team acknowledged the finding and heeded the advice by fixing the pointed typos in commit 523ed899ceac061cb6f47a169c8f40a946213a8d

OPTIMIZATIONS | SOLLONG

ID	Title	Category	Severity	Status
INS-02	Unnecessary Space Allocation	Gas Optimization	Optimization	Resolved

INS-02 | UNNECESSARY SPACE ALLOCATION

Category	Severity	Location	Status
Gas Optimization	Optimization	programs/sollong-preipo/src/instructions/create_financial_account.rs: 14; programs/sollong-preipo/src/instructions/create_user_account.rs: 14	Resolved

Description

The `financial` field in the `CreateFinancial` accounts list represents the account for the data in the `Financial` struct. The account is initialized by reserving 64 bytes of on-chain space, while the `Financial` struct only uses 3 `u8` and 1 `u64` (size 8) thus requiring 11 bytes for the struct itself plus 8 for the Anchor account discriminator. The total required space is 19 bytes while the initialization asks for 64.

The `user_data` field in the `CreateUserData` accounts list represents the account for the data in the `UserData` struct. The account is initialized by reserving 64 bytes of on-chain space, while the `UserData` struct only uses 2 `u8`, 1 `PubKey` (size 32) and 1 `u32` thus requiring 38 bytes for the struct itself plus 8 for the Anchor account discriminator. The total required space is 46 bytes while the initialization asks for 64.

Recommendation

We recommend reserving the required space only for the initialized accounts in order to save on-chain space and, consequently, token for its rent.

Alleviation

[`Sollong Team` , 07/23/2024]: The team acknowledged the finding and heeded the advice by reserving the strictly necessary on-chain space for the mentioned accounts in commit 523ed899ceac061cb6f47a169c8f40a946213a8d

APPENDIX | SOLLONG

Finding Categories

Categories	Description
Gas Optimization	"Gas" is used here as generic term in DLT world, that can differ from chain to chain. Finding indicates that computational, storage resources can be saved, for benefit of users and efficiency of chain. Also in some cases, being not resourceful may lead to DoS attacks.
Coding Style	Coding Style findings may not affect code behavior, but indicate areas where coding practices can be improved to make the code more understandable and maintainable.
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.