CERTIK

Security Assessment

pyth2wormhole - Solana

CertiK Verified on Dec 13th, 2022

CertiK Verified on Dec 13th, 2022

pyth2wormhole - Solana

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

Executive Summary

TYPES	ECOSYSTEM	METHODS
Bridge	Ethereum | Solana | Terra	Manual Review, Static Analysis
LANGUAGE	**TIMELINE**	**KEY COMPONENTS**
Rust, Solidity	Delivered on 12/13/2022	N/A

CODEBASE
https://github.com/pyth-network/pyth2wormhole
...View All

COMMITS
b5555b80f74b88bb9f93275ab9ef293e99653f4b
...View All

Vulnerability Summary

3	1	0	0	2	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.
1	Major	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.
0	Medium		Medium risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform.
0	Minor		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.
2	Informational	1 Resolved, 1 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 | PYTH2WORMHOLE - SOLANA

CODEBASE | PYTH2WORMHOLE - SOLANA

Repository

https://github.com/pyth-network/pyth2wormhole

Commit

b5555b80f74b88bb9f93275ab9ef293e99653f4b

AUDIT SCOPE | PYTH2WORMHOLE - SOLANA

21 files audited ● 1 file with Acknowledged findings ● 20 files without findings

ID	File	SHA256 Checksum
ATR	pyth2wormhole/program/src/attest.rs	93e2f3157cc1de2de07c0fd4f750a4a8bbcd5d6884cb52342b1d4354c7fb7e42
ATS	pyth2wormhole/client/src/attestation_cfg.rs	59d6d7b87c1afc2adc12003c929421b95fedfb98d89b82c7faaddc4d46464789
BAC	pyth2wormhole/client/src/batch_state.rs	17f11174f7d1fd67c3485e5184c4b11df1cd688740d874832286f0b37a3de9f1
CLR	pyth2wormhole/client/src/cli.rs	082ebefc9867bc1ed3d319c95bc345f2f4dd3ec2e8a4b125b03b3fa11844a341
COI	pyth2wormhole/client/src/config_file.rs	288cac825c0d22025c13c1e748720e70b4fa58dfbb216050033ad78b285d2e61
LIE	pyth2wormhole/client/src/lib.rs	0b0582837b02d2059cc2d13a7693ae118b8b46961b4def50314ce59fa475f1c6
MAS	pyth2wormhole/client/src/main.rs	0509e057139cb55820462cb189bc73c351ae4f3726b999c603fbd0fd16a1540f
MOD	pyth2wormhole/client/tests/fixtures/mod.rs	539ab93d78ea9a356343baa7f08533b9389db696b263f084c35f20190ea728d8
PAS	pyth2wormhole/client/tests/fixtures/passthrough.rs	b64aedab07ef069c17ff05d6ca8197a310c26515014c88b61e396d143d6508db
PYF	pyth2wormhole/client/tests/fixtures/pyth.rs	1102819b5a5c7313314b5f1f46db9f7ce589c9dbaa60d60d4016ae5664c7d6f2
TEA	pyth2wormhole/client/tests/test_attest.rs	76fccca19ae3f058ba1a2aefaa4a7b67e0d5c9e4bffd8248cbf6510f2b2d1999
CAO	pyth2wormhole/client/Cargo.toml	24ee98d2c55f425e74a6de6728d5dd1c4eebb28351e7627187a4d0cc5eb47638
COG	pyth2wormhole/program/src/config.rs	5bf1ceea9c9a1af6e9273b460a09add048edf74dc5aac1a5aa14358731b4506b
INL	pyth2wormhole/program/src/initialize.rs	6cae76c6d165773753157f5fc05cdec6fb7c01b438574294a3c3e8349ab58122

ID	File	SHA256 Checksum
LIG	pyth2wormhole/program/src/lib.rs	dd2f11f049e5127bc093f12e3e083c5b00f3872fdc709533db7e761613465895
MIG	pyth2wormhole/program/src/migrate.rs	e6720960a811c4148b76594dc41b4109db8eb22b086cd68d42e6bfa601805d1a
SEC	pyth2wormhole/program/src/set_config.rs	f00a5450a4329ba17abd0151b4159ac55a1f2d3b1e0cba04b65bd4fb93786b04
CAP	pyth2wormhole/program/Cargo.lock	7ee7aae210c7c2f6403ab1eb315b5c4eb411a647b30b92f223fa7355c6e1e703
CAA	pyth2wormhole/program/Cargo.toml	5f32cb96bac2b3102d9445df98a8c22a0ea98d5d1c814ab7fa84322ee1a79491
CAR	pyth2wormhole/Cargo.lock	9f41c2159ab779f68eb660528137144d6cc76600e61a1b110f79dec0f0383c5e
CAG	pyth2wormhole/Cargo.toml	9d654721034cc30c98c8a694bf3f55a9941a528fbd5945fa8434ad0cef15f629

APPROACH & METHODS | PYTH2WORMHOLE - SOLANA

This report has been prepared for Wormhole to discover issues and vulnerabilities in the source code of the pyth2wormhole - Solana 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:

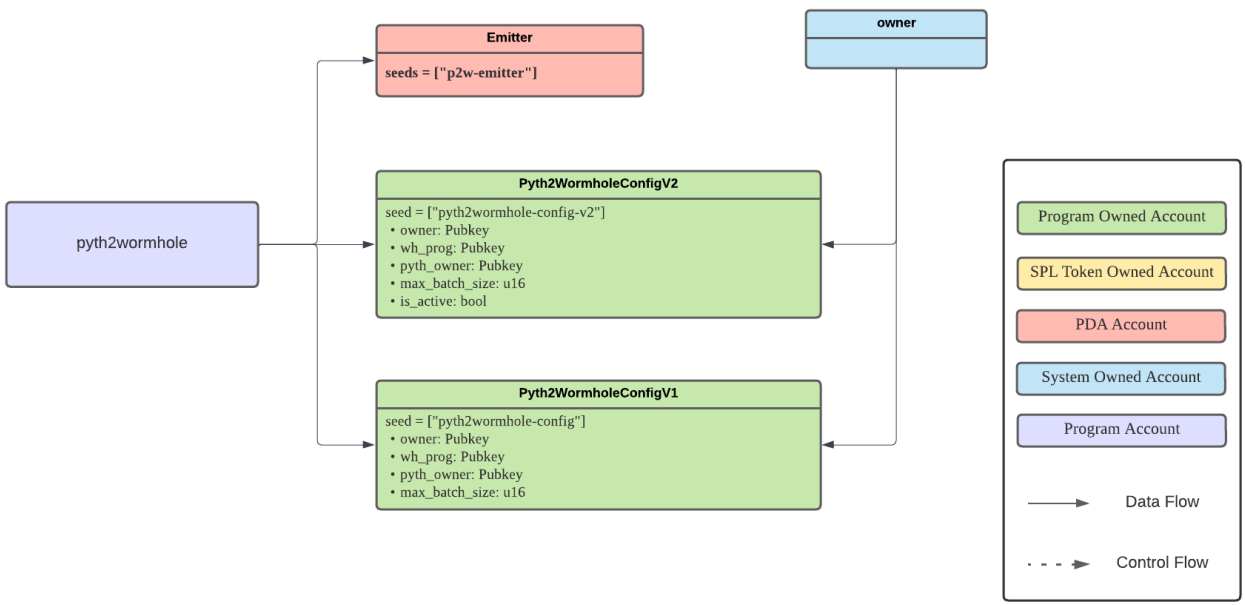
- Testing the smart contracts against both common and uncommon attack vectors;
- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases;
- Provide more comments per each function for readability, especially contracts that are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

REVIEW NOTES | PYTH2WORMHOLE - SOLANA

Understandings

This program is the main Pyth implementation that currently exists as an on-chain contract on Solana. In order to expose these prices cross-chain, the Pyth2Wormhole Solana contract acts as a sender for Pyth prices. At regular intervals, the Pyth contract will observe the current Pyth price for selected products and produce an attestation which is then relayed over Wormhole to be consumed by the various P2W receiver contracts.

Account Relationship



External Dependencies

The project mainly contains the following dependencies:

Dependency	Version
borsh	0.9.3
pyth-client	0.2.2
serde	1
serde_derive	1
serde_json	1
solana-program	1.10.13

It should also be noted here that the code dependencies are being actively developed in the current auditing version. It is necessary to keep the dependencies up-to-date to avoid potential vulnerabilities.

The on-chain program can be upgradeable after the initial deployment based on Solana's features. Also, based on the unique rent mechanism in Solana, the balance in accounts should be carefully set.

Privileged Functions

The program contains a privileged role `current_owner` that has the right to configure and update the whole program. Specifically, it has the authority over the following functions:

- `set_config()` will alter the current settings of pyth2wormhole.
- `migrate()` will migrate the settings of pyth2wormhole.

Additionally, if the program is upgradeable, the upgrade authority account can upgrade the account, thus causing unexpected consequences. The upgrade authority should be carefully managed.

To improve the trustworthiness of the project, dynamic runtime updates in the project should be notified to the community.

FINDINGS | PYTH2WORMHOLE - SOLANA

3	0	1	0	0	2
Total Findings	Critical	Major	Medium	Minor	Informational

This report has been prepared to discover issues and vulnerabilities for pyth2wormhole - Solana. Through this audit, we have uncovered 3 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	**Centralization Related Risks**	**Centralization / Privilege**	**Major**	**Acknowledged**
GLOBAL-02	Versioning Issues	Language Specific	Informational	Acknowledged
ATR-01	Lack Of Validation Between Product And Price	Logical Issue	Informational	Resolved

GLOBAL-01 | CENTRALIZATION RELATED RISKS

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

Description

In program pyth2wormhole, the role `owner` has authority over the following functions:

- `migrate()` will migrate on-chain configuration from an older format.
- `set_config()` will update current settings of `pyth2wormhole`.

Any compromise to the `owner` account may allow a hacker to take advantage of this authority and manipulate the program settings.

Additionally, the Solana program could be upgradeable, and the upgrade authority is the deployer by default. Therefore, if the program is upgradable, and the upgrade authority account is compromised, it could lead to a malicious program upgrade, thus introducing centralization risk.

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 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.

Noted: Recommend considering 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

[**Pyth**, 11/24/2022]: The team acknowledged the finding, and the aforementioned actions are currently controlled by a multi-sig wallet.

GLOBAL-02 | VERSIONING ISSUES

Category	Severity	Location	Status
Language Specific	Informational		Acknowledged

Description

The `pyth2wormhole` project is built depending on the `Wormhole Solana`. The Solana module of the `pyth2wormhole` are derived from `Wormhole Solana` and remain not updated after `Wormhole Solana` commit 064228869731428e6443c761cb753cf8d1ab01e3.

For `Wormhole Solana`, some important components for `pyth2wormhole` like `solitaire` keep updating after the project branched. For example, some important updates in `solitare` are shown as below:

- Fix pre-funded account block in commit 2b56fcc7da4422f2270431f0f9c996021115acc0
- Fix initialization check in commit da479cf4c18b024a55ab3db87d027178fd319344

Additionally, it should be noticed that, in the `Wormhole Solana` commit 9aad49d631167e59790fcbc315985fd107adfbcf, `deps()` functions and `ToInstruction` macro are removed from `solitaire`, which will affect the client code that are using `to_ix` function.

Recommendation

As `pyth2wormhole`'s Solana components (like `solitaire`) are heavily dependent on `Wormhole Solana`, we advise the team to keep monitoring the updates and keep those dependencies up-to-date along with necessary modifications upon request.

Alleviation

[**Pyth**, 11/29/2022]: The team acknowledged the finding, and the team will keep tracking the code dependencies updates.

ATR-01 | LACK OF VALIDATION BETWEEN PRODUCT AND PRICE

Category	Severity	Location	Status
Logical Issue	Informational	pyth2wormhole/program/src/attest.rs: 231	Resolved

Description

During the `attest` instruction, the program will check if the given product account matches the `prod` field in the price account. It will not check the price account record in the product account data(`px_acc` field). Our concern is that the presence of multiple price accounts for a single product account(e.g. an old price account) may result in inaccurate or invalid prices.

Recommendation

We would like to check with the team about the mapping between the product account and price account, also the pair update and generation mechanism.

Alleviation

[**Pyth**, 11/29/2022]: This is a non-issue as even if this is possible, the cross-chain contracts only cache the latest price. In addition, they are identified by the `price_id` , which is unique for every `price_account` .

OPTIMIZATIONS | PYTH2WORMHOLE - SOLANA

ID	Title	Category	Severity	Status
ATR-02	Redundant Code	Gas Optimization	Optimization	Acknowledged

ATR-02 | REDUNDANT CODE

Category	Severity	Location	Status
Gas Optimization	Optimization	pyth2wormhole/program/src/attest.rs: 150	Acknowledged

Description

The function `attest()`, requires a `config` account, which is stated to be a `Derive` type.

```
67      pub config: P2WConfigAccount<'b, { AccountState::Initialized }>,
```

```
38  pub type P2WConfigAccount<'b, const IsInitialized: AccountState> =
39      P2WConfigAccountV2<'b, IsInitialized>;
```

```
  94  pub type P2WConfigAccountV2<'b, const IsInitialized: AccountState> =
  95      Derive<Data<'b, Pyth2WormholeConfigV2, { IsInitialized }>, "pyth2wormhole-
config-v2">;
```

Since `Derive` types automatically have their PDA checked through the `peel()` function acquired from the `FromAccounts` macro, it does not need to be checked again.

```
  72  impl<'a, 'b: 'a, 'c, T: Peel<'a, 'b, 'c>, const Seed: &'static str> Peel<'a,
'b, 'c>
  73      for Derive<T, Seed>
  74  {
  75      fn peel<I>(ctx: &'c mut Context<'a, 'b, 'c, I>) -> Result<Self> {
  76          // Attempt to Derive Seed
  77          let (derived, bump) = Pubkey::find_program_address(&[Seed.as_ref()],
ctx.this);
  78          match derived == *ctx.info().key {
  79              true => T::peel(ctx).map(|v| Derive(v)),
  80              _ => Err(SolitaireError::InvalidDerive(*ctx.info().key,
derived).into()),
  81          }
```

Recommendation

Recommend removing the redundant code.

Alleviation

[**Pyth**, 11/29/2022]: The team acknowledges the finding and will fix the issue in the future, which will not be included in this audit engagement.

APPENDIX | PYTH2WORMHOLE - SOLANA

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.
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.
Language Specific	Language Specific findings are issues that would only arise within Solidity, i.e. incorrect usage of private or delete.

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.