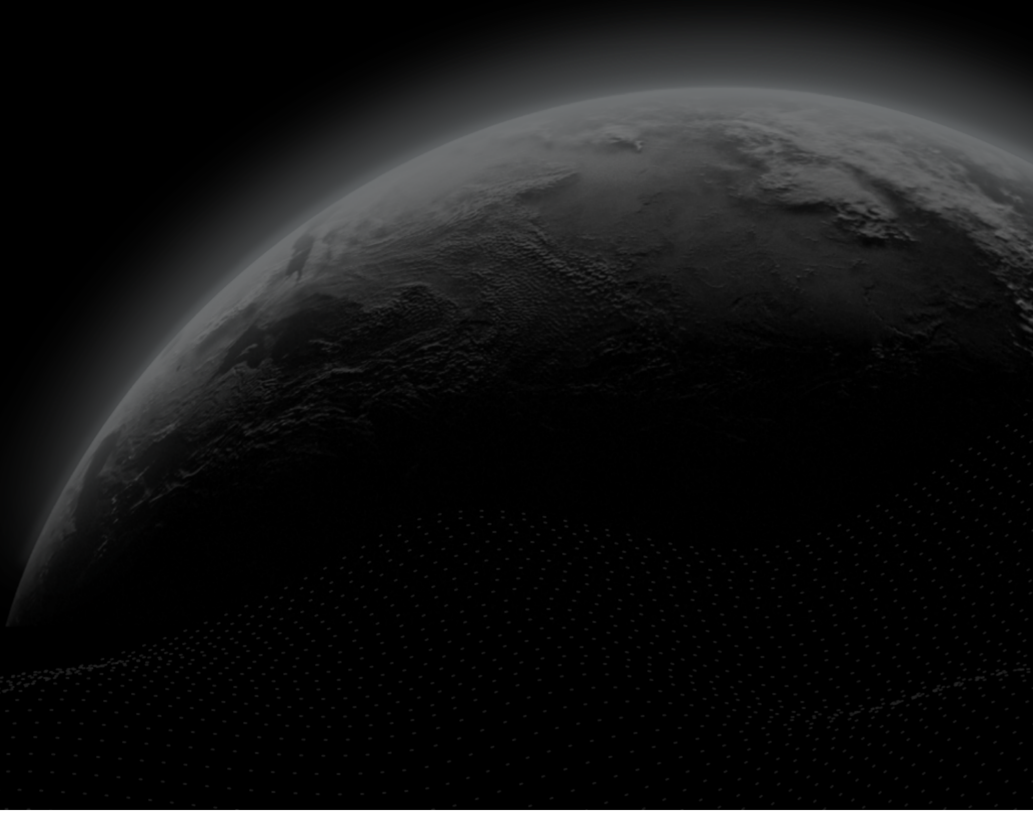
CERTIK

Security Assessment

Dpex

CertiK Verified on Jan 3rd, 2023



CertiK Verified on Jan 3rd, 2023

Dpex

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

Executive Summary

TYPES	ECOSYSTEM	METHODS
Others	Other	Manual Review, Static Analysis

LANGUAGE	TIMELINE	KEY COMPONENTS
Solidity	Delivered on 01/03/2023	N/A

CODEBASE

https://github.com/DPEX-io/dpex/

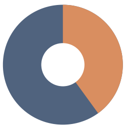
...View All

COMMITS

- f0df642cfa9b930a79d561ad1b68f9bc352ecbf1
- 845fec7a2b417ebbdb3efb97471128de2992ca35

...View All

Vulnerability Summary



5	3	2	0	0	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 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.
🟨	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.
🟦	3	Informational	3 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 | DPEX

CODEBASE | DPEX

Repository

https://github.com/DPEX-io/dpex/

Commit

- f0df642cfa9b930a79d561ad1b68f9bc352ecbf1
- 845fec7a2b417ebbdb3efb97471128de2992ca35

AUDIT SCOPE | DPEX

8 files audited ● 1 file with Mitigated findings ● 7 files without findings

ID	File	SHA256 Checksum
● BTD	tokens/BaseToken.sol	ffc0a5881ae5adc2214cf2710a1ba922a2 8f6940179c25b5ca3823c9e0f73b4d
● DPE	dpex/DPEX.sol	d52b5e9193944d1e4fbcf2dc53d462082 db83d330167c36ad0fb965c56a1e52e
● CGN	libraries/GSN/Context.sol	eac5f16b2857979060cee432030681ca 9ca20f0164c98b7f7422756431e6bdea
● SMD	libraries/math/SafeMath.sol	a60c5e6a4c16e42c5c6333bae2c81600 3e755e8f979538ef65a63d40854588b2
● ERC	libraries/token/ERC20.sol	b60b5ddd0e0e0b4c39e29388fe1a6131 89ea2ac3b182c4e490e3522cfe99d0d2
● IEC	libraries/token/IERC20.sol	ef4e2497a840d900716a22e46ec10e1a9 c0da9e1aea6f7fe7769e55eb4bea341
● ADP	libraries/utils/Address.sol	32f7be26a2029f9c750526674d75bce20 3126e5f444634dedefb14bf7809489e
● MBD	tokens/MintableBaseToken.sol	8f190f687cc288278f2210d3a78255d8f c2ce8b0ea0b58c9a7b00b3fbb255ab4

APPROACH & METHODS | DPEX

This report has been prepared for Dpex to discover issues and vulnerabilities in the source code of the Dpex 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 | DPEX

Overview

Dpex has implemented a decentralized spot and perpetual exchange. The current audit scope includes the **DPEX** token contracts - the platform's utility and governance token, which can unlock various benefits for holders.

External Dependencies

The system inherits or uses a few of the depending injection contracts to fulfill the need of its business logic.

- `yieldTrackers` : contract where token holders can claim rewards from.
- Privileged roles such as minter role, admin roles and gov roles.

We assume these contracts or addresses are valid and non-vulnerable actors and implement proper logic to collaborate with the current project.

Privileged Roles

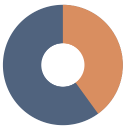
To set up the project correctly and improve overall project quality, the following roles are adopted in the codebase(More details in *GLOBAL-01 - Centralization Related Risks*):

- Governance role is adopted to set minter roles, update configurations of the contract, and set admin roles.
- Admin role is adopted to update the staking account information and recover claims.
- Minter role is adopted to mint/burn tokens from a given address.
- Handler role is adopted to transfer tokens from an arbitrary address to another one.

Any compromise of the owner's private key may allow an attacker to pause the contract.

To improve the trustworthiness of the project, dynamic runtime updates in the project should be notified to the community. Furthermore, any plan to invoke the aforementioned functions should also be considered to move to the execution queue of the `Timelock` contract.

FINDINGS | DPEX



5	0	2	0	0	3
Total Findings	Critical	Major	Medium	Minor	Informational

This report has been prepared to discover issues and vulnerabilities for Dpex. 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
BTD-01	**Minting Authority On DPEX Token**	**Centralization / Privilege**	**Major**	● **Mitigated**
GLOBAL-01	**Centralization Related Risks**	**Centralization / Privilege**	**Major**	● **Mitigated**
BTD-02	Unused Return Value	Volatile Code	Informational	● Resolved
BTD-03	Potential Denial-Of-Service Situation	Volatile Code	Informational	● Resolved
BTD-04	Potential Risk On `approve()` / `transferFrom()` Methods	Logical Issue	Informational	● Resolved

BTD-01 | MINTING AUTHORITY ON DPEX TOKEN

Category	Severity	Location	Status
Centralization / Privilege	● **Major**	**tokens/BaseToken.sol: 49**	● **Mitigated**

Description

The minter role of the **DPEX** token is able to mint/burn an unlimited amount of DPEX tokens without the consensus of the community. The concern is the minter role can distribute or burn the DPEX token arbitrarily, thus could cause tokenomics issues to the project as a whole.

Recommendation

We recommend transparency through providing a breakdown of the intended token-minting process in a public location. We also recommend the team make an effort to restrict the access of the corresponding private key.

Alleviation

[**DPEX**, 01/01/2023]: The team will implement TimeLock smart contract with a 4h-8h target between execution, thus protecting users from unauthorized mints. The team believes keeping the function will allow more elasticity in our protocol if the execution plan changes.

[**DPEX**, 01/03/2023]: The team has deployed this timelock smart contract at https://polygonscan.com/address/0x29d05f96e0a975ef199ee3205cceb8bdeb43d545#code to mitigated the centralization risk for mint authority. And on line 1120 implements a maximum supply authority validation which is limited by `maxTokenSupply` (`125000000000000000000000000`)

GLOBAL-01 | CENTRALIZATION RELATED RISKS

Category	Severity	Location	Status
Centralization / Privilege	● **Major**		● **Mitigated**

Description

In **DPEX** token, the role minter has authority over the following functions:

- `mint()` : Mint tokens to a given address.
- `burn()` : Burn tokens from a given address.

The governance role has authority over the following functions:

- `setMinter()` : Set a given address as Minter.
- `setGove()` : Set a given address as a governance role.
- `setYieldTrackers()` : Update the `yieldTrackers` variable.
- `addAdmin()` : Add an address as the admin.
- `removeAdmin()` : Remove an address from admin.
- `withdrawToken()` : Withdraw tokens in the contract.
- `setHandler()` : Set the state of the handler.

The admin role has the authority over the following functions:

- `addNonStakingAccount()` : Add an account as non-staking account.
- `removeNonStakingAccount()` : Remove an account as non-staking account.
- `recoverClaim()` : Recover the claim for a given account.

Additionally, once an address is set as the handler, the handler address is able to call `transferFrom()` to transfer anyone's DPEX token without approval.

Any compromise to the above-mentioned account may allow a hacker to take advantage of this authority and burn/mint tokens, thus causing unexpected results.

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 recommend carefully managing 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.

Alleviation

[**DPEX**, 01/01/2023]: The team created a gnosis-safe multisig to mitigate the centralization risk and the team will issue a time-lock once deployed.

The gnosis-safe address on the polygon is: 0xD637CB488C0ab931029d8F5E31Ac7125e1Ec7124

It has three owners, which are hardware wallets:

- M01 DPEX 33 NanoX - 0x969952e379C6F0a1cb15E1c86972965072820118
- M02 DPEX 33 NanoS 01 - 0x69C6B5E96D8EA54F7795F706C339b0057F32E99d
- M03 DPEX 33 NanoS 02 - 0xE8fC9fa37667fd9c30B7bEbE4FE68d9dd9B664e3

The policy that the team implemented on Gnosis Safe is 2/3.

BTD-02 | UNUSED RETURN VALUE

Category	Severity	Location	Status
Volatile Code	● Informational	tokens/BaseToken.sol: 103, 110	● Resolved

Description

The return value of the following external invocations `IYieldTracker(yieldTracker).claim()` is not properly handled.

```
    function recoverClaim(address _account, address _receiver) external onlyAdmin {
        for (uint256 i = 0; i < yieldTrackers.length; i++) {
            address yieldTracker = yieldTrackers[i];
            IYieldTracker(yieldTracker).claim(_account, _receiver);
        }
    }

    function claim(address _receiver) external {
        for (uint256 i = 0; i < yieldTrackers.length; i++) {
            address yieldTracker = yieldTrackers[i];
            IYieldTracker(yieldTracker).claim(msg.sender, _receiver);
        }
    }
```

Recommendation

We recommend properly handling the return values of external function calls.

Alleviation

[**DPEX**, 01/01/2023]: The team resolved this finding in commit 845fec7a2b417ebbdb3efb97471128de2992ca35 by checking claim amount for each claim operation in the batch, and adding additional function `claimByIndex` for the single index claim.

BTD-03 | POTENTIAL DENIAL-OF-SERVICE SITUATION

Category	Severity	Location	Status
Volatile Code	● Informational	tokens/BaseToken.sol: 103, 110, 220	● Resolved

Description

In the function `recoverClaim()` / `claim()` , user can claim their rewards. However, if any of the `claim()` invocation failed/reverted, it will cause users to be unable to claim the reward in time.

Recommendation

In the short term, ensure all the `yeildTracker` contracts works properly as expected.

In the long term, adding function allows users to choose the index of the `yeildTracker` contract they want to claim.

Alleviation

[**DPEX**, 01/01/2023]: The team resolved this finding in commit 845fec7a2b417ebbdb3efb97471128de2992ca35 by adding additional function `claimByIndex` for the single index claim to avoid failure in the batch claim.

BTD-04 | POTENTIAL RISK ON `approve()` / `transferFrom()` METHODS

Category	Severity	Location	Status
Logical Issue	● Informational	tokens/BaseToken.sol: 138	● Resolved

Description

The `BaseToken` implementation is vulnerable to a known ERC20 race condition issue, which could lead to token theft. When a user calls `approve()` for a second time on a spender that has already been allowed, the spender could call `transferFrom()` to transfer the previous value and still receive the authorization to transfer the new value.

Exploit scenario:

1. Alice calls `approve(Bob, 100)` to allow Bob to spend 100 tokens.
2. Alice changes her mind and calls `approve(Bob, 50)` .
3. Bob observes the second `approve(Bob, 50)` function call and calls `transferFrom(Alice, Bob, 100)` before the second `approve(Bob, 50)` call.
4. The above scenario can be achieved by front-running. In this case, Bob can transfer another 50 tokens from Alice and in total, he transferred 150 tokens from Alice.

Recommendation

We would advise using OpenZeppelin ERC20 implementation as it includes `increaseAllowance()` and `decreaseAllowance()` methods. These functions only change the allowance by a certain value instead of setting the new one. It is commonly used protection against FrontRunning of ERC20's approval issue.

Alleviation

[**DPEX**, 01/01/2023]: The team resolved this finding in commit 845fec7a2b417ebbdb3efb97471128de2992ca35 by implementing `increaseAllowance()` and `decreaseAllowance()` methods.

OPTIMIZATIONS | DPEX

ID	Title	Category	Severity	Status
BTD-05	Missing Input Validation	Volatile Code	Optimization	● Resolved

BTD-05 | MISSING INPUT VALIDATION

Category	Severity	Location	Status
Volatile Code	● Optimization	tokens/BaseToken.sol: 69~71	● Resolved

Description

In the contract `BaseToken.sol`, the function `removeAdmin()` / `addAdmin()` removes an account from the "admins" role. However, before setting `admins[_account]` as `true` or `false`, the function doesn't check if the addresses' state has been set. Therefore, it could cause extra gas costs to remove a non-existing admin account or add an existing admin account.

```
65      function addAdmin(address _account) external onlyGov {
66          admins[_account] = true;
67      }
68
69      function removeAdmin(address _account) external override onlyGov {
70          admins[_account] = false;
71      }
```

Recommendation

We recommend checking if the account is not an admin before actually removing the account. For example,

```
69      function removeAdmin(address _account) external override onlyGov {
70          require(admins[_account], "BaseToken: _account not marked");
71          admins[_account] = false;
72      }
```

Alleviation

[**DPEX**, 01/01/2023]: The team resolved this finding in commit 845fec7a2b417ebbdb3efb97471128de2992ca35 by checking account status before the role update.

APPENDIX | DPEX

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.
Logical Issue	Logical Issue findings detail a fault in the logic of the linked code, such as an incorrect notion on how block.timestamp works.
Volatile Code	Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that may result in a vulnerability.

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.