CERTIK

bot chain - DEX

Security Assessment

CertiK Assessed on Feb 17th, 2026

CertiK Assessed on Feb 17th, 2026

bot chain - DEX

The security assessment was prepared by CertiK.

Executive Summary

TYPES	ECOSYSTEM	METHODS
DEX, Staking	bot chain	Manual Review, Static Analysis

LANGUAGE	TIMELINE
Solidity	Preliminary comments published on 02/09/2026
	Final report published on 09/04/2026

Vulnerability Summary

3	0	0	3	0
Total Findings	Resolved	Partially Resolved	Acknowledged	Declined

1	Centralization	1 Acknowledged	Centralization findings highlight privileged roles & functions and their capabilities, or instances where the project takes custody of users' assets.
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.
0	Major		Major risks may include logical errors that, under specific circumstances, could result in fund losses or loss of project control.
0	Medium		Medium risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform.
1	Minor	1 Acknowledged	Minor risks can be any of the above, but on a smaller scale. They generally do not compromise the overall integrity of the project, but they may be less efficient than other solutions.
1	Informational	1 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 | BOT CHAIN - DEX

CODEBASE | BOT CHAIN - DEX

Repository

Private Repo

Audit Scope

The files in scope are listed in the appendix.

APPROACH & METHODS | BOT CHAIN - DEX

This audit was conducted for bot chain to evaluate the security and correctness of the smart contracts associated with the bot chain - DEX project. The assessment included a comprehensive review of the in-scope smart contracts. The audit was performed using a combination of Manual Review and Static Analysis.

The review process emphasized the following areas:

- Architecture review and threat modeling to understand systemic risks and identify design-level flaws.
- Identification of vulnerabilities through both common and edge-case attack vectors.
- Manual verification of contract logic to ensure alignment with intended design and business requirements.
- Dynamic testing to validate runtime behavior and assess execution risks.
- Assessment of code quality and maintainability, including adherence to current best practices and industry standards.

The audit resulted in findings categorized across multiple severity levels, from informational to critical. To enhance the project's security and long-term robustness, we recommend addressing the identified issues and considering the following general improvements:

- Improve code readability and maintainability by adopting a clean architectural pattern and modular design.
- Strengthen testing coverage, including unit and integration tests for key functionalities and edge cases.
- Maintain meaningful inline comments and documentations.
- Implement clear and transparent documentation for privileged roles and sensitive protocol operations.
- Regularly review and simulate contract behavior against newly emerging attack vectors.

SYSTEM OVERVIEW | BOT CHAIN - DEX

The BOT DEX audit covers a Uniswap V3–based DEX deployment.

BDEX Deployer (Uniswap V3 Fork)

The DEX deployment is a fork of Uniswap V3 core and periphery. **Note that for this part, only the differences between BDEX Deployer and Uniswap V3 have been audited**.

- The contracts follow the standard Uniswap V3 architecture: core pool/factory logic and periphery routers/position manager for swaps and liquidity positions.
- The deployer package includes Hardhat configuration and build tooling to compile and deploy the forked contracts.

REVIEW NOTES | BOT CHAIN - DEX

Out-of-Scope Dependencies

The codebase contains multiple components and dependencies that were not fully covered by this assessment. In particular:

- Parts of the repository are **forked or derived from upstream projects** (e.g., Uniswap V3 core/periphery). Components that were not explicitly in scope and/or appear unchanged from upstream were treated as external dependencies and assumed to behave as specified by their original maintainers.
- **Third-party libraries and build tooling** were not audited comprehensively. Any security properties of the development toolchain are outside the scope of this report.
- The security of **external contracts and external systems** interacted with by the reviewed contracts (e.g., ERC-20 tokens, WETH, bridge operators/relayers, off-chain signing infrastructure, governance processes, and key management) is out of scope. These elements were treated as operational assumptions unless the contract enforces the relevant security guarantees on-chain.

Our review focused on the on-chain behavior of the in-scope Solidity contracts within the defined scope.

Testing & Documentation

The repository includes automated tests and developer tooling, particularly around the Uniswap-derived components. However, overall **integration and edge-case coverage appears limited** across the full system, especially when considering cross-component behavior (e.g., bridge flows, privileged operations, and configuration correctness).

We recommend expanding test coverage to include stress scenarios and failure modes (e.g., misconfiguration, unexpected inputs, privilege/role transitions, and revert-path behavior). On the documentation front, inline comments exist, but higher-level documentation is limited. Adding concise system-level documentation (architecture and operational assumptions) would improve maintainability and reduce misconfiguration risk.

Code Completeness

We noticed several `TODO` / `FIXME` comments remaining in the codebase. These markers may indicate incomplete hardening work, deferred changes, or leftover development notes. We recommend triaging these items by resolving or removing outdated TODO/FIXME notes and re-reviewing any security-relevant logic added or modified as a result.

FINDINGS | BOT CHAIN - DEX

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

This report has been prepared for bot chain to identify potential vulnerabilities and security issues within the reviewed codebase. During the course of the audit, a total of 3 issues were identified. Leveraging a combination of Manual Review & Static Analysis the following findings were uncovered:

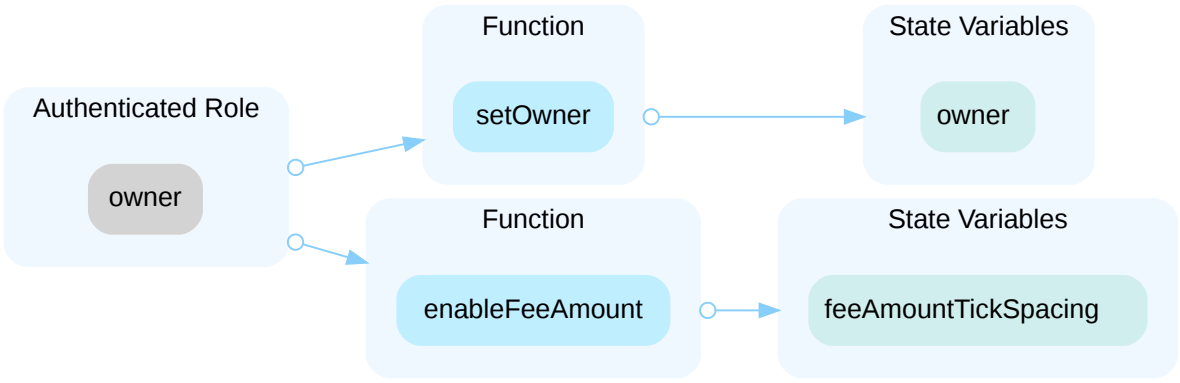
ID	Title	Category	Severity	Status
BCD-03	**Centralization Risks**	**Centralization**	**Centralization**	**Acknowledged**
BCD-04	Missing Zero Address Validation	Volatile Code	Minor	Acknowledged
BCD-05	Require Without Error Message	Coding Style	Informational	Acknowledged

BCD-03 | Centralization Risks

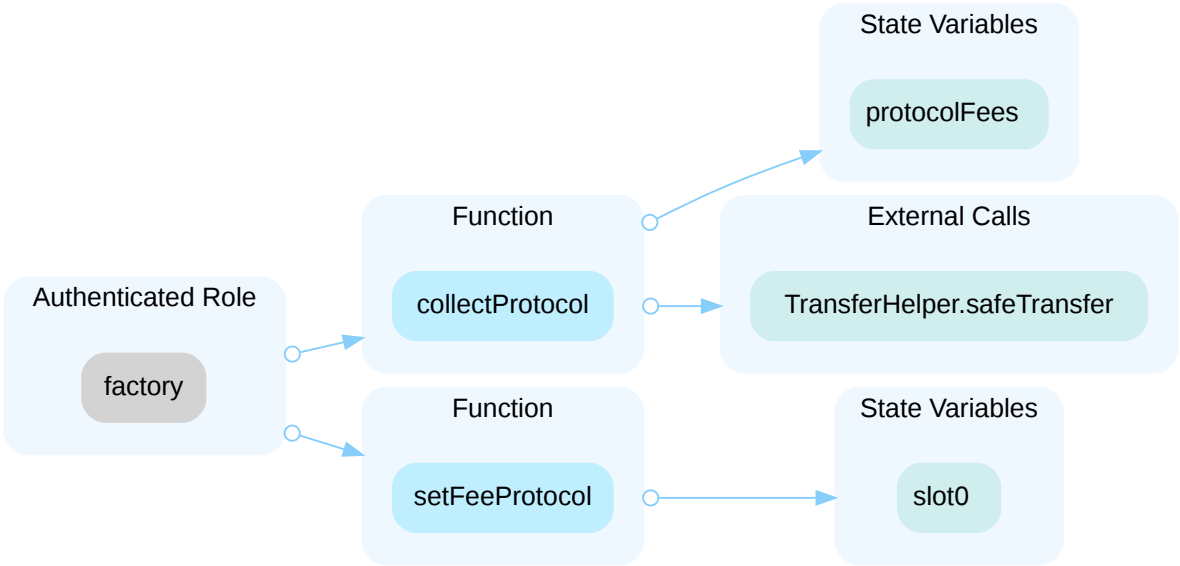
Category	Severity	Location	Status
Centralization	Centralization	contracts/v3-core/UniswapV3Factory.sol: 53, 61; contracts/v3-core/UniswapV3Pool.sol: 837, 848	Acknowledged

Description

In the contract `UniswapV3Factory`, the role `owner` has authority over the functions shown in the diagram below. Any compromise to the `owner` account may allow the hacker to take advantage of this authority and potentially exploit the functions and controls assigned to this role.



In the contract `UniswapV3Pool`, the role `factory` has authority over the functions shown in the diagram below. Any compromise to the factory owner may allow the hacker to take advantage of this authority and set the pool protocol fee parameters and collect protocol fees.



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

[BOT Chain, 02/17/2026]: The team acknowledged the issue and decided not to implement the recommended change in the current engagement. Moreover, the privileged operations (e.g., ownership transfer, fee tier management, protocol fee parameter updates, and protocol fee collection) are part of the intended design inherited from the Uniswap V3 architecture, where the factory owner/factory role governs protocol-level parameters. The team will mitigate the operational risk through

strict private key management and internal controls, and may consider moving the privileged role to a multisignature and/or timelock setup in later phases.

[CertiK, 02/17/2026]: It is suggested to implement the aforementioned methods to avoid centralized failure. Also, CertiK strongly encourages the project team to periodically revisit the private key security management of all addresses related to centralized roles.

BCD-04 | Missing Zero Address Validation

Category	Severity	Location	Status
Volatile Code	Minor	contracts/v3-core/UniswapV3Factory.sol: 57; contracts/v3-periphery/NonfungiblePositionManager.sol: 79; contracts/v3-periphery/NonfungibleTokenPositionDescriptor.sol: 30; contracts/v3-periphery/V3Migrator.sol: 30; contracts/v3-periphery/base/PeripheryImmutableState.sol: 15, 16	Acknowledged

Description

The cited address input is missing a check that it is not `address(0)` .

Recommendation

We recommend adding a check the passed-in address is not `address(0)` to prevent unexpected errors.

Alleviation

[BOT Chain, 02/17/2026]: The team acknowledged the issue and decided not to implement the recommended change in the current engagement. Moreover, the contracts are forked from Uniswap V3 (and consistent with PancakeV3), where zero-address validations are not implemented, and the identified instances are mainly related to admin or deployment operations. The team prefers to maintain upstream parity and control bytecode size at this stage.

BCD-05 | Require Without Error Message

Category	Severity	Location	Status
Coding Style	Informational	contracts/v3-core/NoDelegateCall.sol: 19; contracts/v3-core/UniswapV3Factory.sol: 40, 42, 44, 45, 55, 62, 63, 67, 68; contracts/v3-core/UniswapV3Pool.sol: 113, 143, 153, 188, 197, 464, 838~841; contracts/v3-core/libraries/BitMath.sol: 14, 54; contracts/v3-core/libraries/FullMath.sol: 34, 43, 120; contracts/v3-core/libraries/LowGasSafeMath.sol: 12, 20, 28, 36, 44; contracts/v3-core/libraries/SafeCast.sol: 11, 18, 25; contracts/v3-core/libraries/SqrtPriceMath.sol: 52, 93, 112, 113, 135, 136, 164; contracts/v3-core/libraries/TickBitmap.sol: 28; contracts/v3-periphery/NonfungiblePositionManager.sol: 215, 304, 308, 383; contracts/v3-periphery/SwapRouter.sol: 62, 199; contracts/v3-periphery/base/PeripheryPaymentsWithFee.sol: 23, 44; contracts/v3-periphery/base/PoolInitializer.sol: 19; contracts/v3-periphery/lens/Quoter.sol: 43, 59; contracts/v3-periphery/lens/QuoterV2.sol: 46, 68; contracts/v3-periphery/libraries/CallbackValidation.sol: 34; contracts/v3-periphery/libraries/LiquidityAmounts.sol: 14; contracts/v3-periphery/libraries/PoolAddress.sol: 34; contracts/v3-periphery/libraries/SqrtPriceMathPartial.sol: 31	Acknowledged

Description

The `require(`) function validates conditions and throws an exception if the condition fails. It is advisable to include a descriptive error message, which will be returned to the caller, providing details about the failure.

Recommendation

We advise adding error messages to the linked **require** statements.

Alleviation

[BOT Chain, 02/17/2026]: The team acknowledged the issue and decided not to implement the recommended change in the current engagement. Moreover, the codebase follows the upstream Uniswap V3 design pattern where many `require` statements intentionally omit revert strings to reduce bytecode size and minimize gas overhead, especially in frequently executed paths. The team prefers to keep the implementation consistent with upstream and avoid introducing non-trivial diffs in this phase.

OPTIMIZATIONS | BOT CHAIN - DEX

ID	Title	Category	Severity	Status
BCD-01	Cache Multiple Accesses Of Mapping/Array Values	Gas Optimization	Optimization	Acknowledged
BCD-02	Avoid Updating Storage When The Value Hasn't Changed	Gas Optimization	Optimization	Acknowledged

BCD-01 | Cache Multiple Accesses Of Mapping/Array Values

Category	Severity	Location	Status
Gas Optimization	Optimization	contracts/v3-core/UniswapV3Factory.sol: 45, 68; contracts/v3-periphery/NonfungiblePositionManager.sol: 128	Acknowledged

Description

When a mapping value is accessed multiple times, caching it in a local `storage` or `calldata` variable can save approximately 42 gas per access. This is because it eliminates the need to recalculate the key's keccak256 hash, which costs 30 gas (Gkeccak256), along with the associated stack operations. Similarly, caching a struct from an array avoids the repetitive calculation of array offsets into memory or calldata.

Recommendation

Caching a mapping's value in a local `storage` or `calldata` variable.

Alleviation

[BOT Chain, 02/17/2026]: The team acknowledged the issue and decided not to implement the recommended change in the current engagement. Moreover, this is a non-security gas optimization suggestion, and the current implementation is intentionally kept close to upstream Uniswap V3 to minimize code divergence. The team may consider applying selective micro-optimizations in a future optimization-focused phase after core logic and upgrade plans are finalized.

BCD-02 | Avoid Updating Storage When The Value Hasn't Changed

Category	Severity	Location	Status
Gas Optimization	Optimization	contracts/v3-core/UniswapV3Factory.sol: 57	Acknowledged

Description

If the old value is equal to the new value, not re-storing the value will avoid a Gsreset (2900 gas), potentially at the expense of a Gcoldsload (2100 gas) or a Gwarmaccess (100 gas).

Recommendation

We recommend avoiding updating the value if the value is not changed.

Alleviation

[BOT Chain, 02/17/2026]: The team acknowledged the issue and decided not to implement the recommended change in the current engagement. Moreover, this is a non-security gas optimization suggestion and does not impact protocol correctness or safety. To preserve upstream parity and avoid introducing additional branching logic that changes the original code structure, the team decided not to apply this optimization in the current engagement.

APPENDIX | BOT CHAIN - DEX

Audit Scope

CertiKProject/certik-audit-projects
contracts/v3-periphery/base/PoolInitializer.sol
contracts/v3-periphery/base/PeripheryImmutableState.sol
contracts/v3-periphery/base/PeripheryPaymentsWithFee.sol
contracts/v3-periphery/NonfungiblePositionManager.sol
contracts/v3-periphery/NonfungibleTokenPositionDescriptor.sol
contracts/v3-periphery/SwapRouter.sol
contracts/v3-periphery/V3Migrator.sol
contracts/v3-core/NoDelegateCall.sol
contracts/v3-core/UniswapV3Factory.sol
contracts/v3-core/UniswapV3Pool.sol
contracts/v3-periphery/base/BlockTimestamp.sol
contracts/v3-periphery/base/ERC721Permit.sol
contracts/v3-periphery/base/LiquidityManagement.sol
contracts/v3-periphery/base/Multicall.sol
contracts/v3-periphery/base/PeripheryPayments.sol
contracts/v3-periphery/base/PeripheryValidation.sol
contracts/v3-periphery/base/SelfPermit.sol
contracts/v3-core/UniswapV3PoolDeployer.sol

Finding Categories

Categories	Description
Centralization	Centralization findings detail the design choices of designating privileged roles or other centralized controls over the code.
Volatile Code	Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases and may result in vulnerabilities.
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.
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.

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.

Elevate Your Web3 Journey

CertiK is the largest Web3 security platform combining formal verification with audits and comprehensive security solutions.