CERTIK

Security Assessment

HashMix

CertiK Assessed on Jun 7th, 2023

CertiK Assessed on Jun 7th, 2023

HashMix

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

Executive Summary

TYPES	ECOSYSTEM	METHODS
DeFi	Filecoin (FIL)	Manual Review, Static Analysis
LANGUAGE	**TIMELINE**	**KEY COMPONENTS**
Solidity	Delivered on 06/07/2023	N/A

CODEBASE

https://github.com/HashMixProject/hashmix-fevm/tree/0a5676a17f4d49ce6a07c3e461faa05016d18e82

...View All

Vulnerability Summary

6	0	0	0	6	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	2 Acknowledged	Major risks can include centralization issues and logical errors. Under specific circumstances, these major risks can lead to loss of funds and/or control of the project.
0	Medium		Medium risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform.
4	Minor	4 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.
0	Informational		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 | HASHMIX

CODEBASE | HASHMIX

Repository

https://github.com/HashMixProject/hashmix-fevm/tree/0a5676a17f4d49ce6a07c3e461faa05016d18e82

AUDIT SCOPE | HASHMIX

14 files audited ● 3 files with Acknowledged findings ● 11 files without findings

ID	File	SHA256 Checksum
HHM	contracts/Hashmix.sol	aab9749729ed06bf9e397f9cb8db50d5d1e7f9217fe68c0405109d029ae470ca
JRM	contracts/JumpRateModel.sol	7a1b6c07473dad6775944ee85a3def7e31e4e75612cd08e7b35b5e07ca5b4b84
WPI	contracts/WhitePaperInterestRateModel.sol	3fb88749f31ca3f00fb00de74d5d47758d5cf4db7a4f9e148abf00187f3b15f2
ENE	contracts/libs/ExponentialNoError.sol	ef79b0e99297f924296b136e84d8986170409eb233820b3e7733c2a6387707e9
LHM	contracts/libs/Leb128.sol	8601a4990b776f51ac0b65bd78d81c79b8cc229c457e78c34be6e365fcf07aae
CBH	contracts/CreditsBook.sol	1a619e662b1e00f4634694f6d1cf29e6150aec439438e799759eb0ed7c31105e
ERH	contracts/ErrorReporter.sol	97062c28c271dac34dd5b46e74bfa31d6d75f52b4c8c34653f275b2578a5e000
FMH	contracts/FeeModel.sol	ae2c56ee5cbd6faffc92073a067d5e92d2b3557ecbbebef6e3712fa235c7c263
HER	contracts/HsmERC20.sol	69af65fe97d32f71f76d3297fa4e8b250a0517281f62b5ef7289e92def365af0
IRM	contracts/InterestRateModel.sol	8ca958179765a9ef12f955a76afdd6ac8bdacbe0be61216b6329e674b5739e7d
MCH	contracts/MinerCertificate.sol	762a3e0d98a48a43647827f031e7617463476700446d223ac24a0021b57a0444
MSH	contracts/MultiSig.sol	ce6c7c27290d0e092a8a543634149a79f47c685b6ed5ff1f4ed653df20533b61
PFH	contracts/PeanutFarm.sol	db3264a2545b3a629c8b4d82e1d1a5680b7aedb1be8c216bbe9ddf742cbabec1
PHH	contracts/PeanutHull.sol	e75e06c26248061b822bfa6d2e19cdb0c528868552fcbb36197be0e86f3bdce7

APPROACH & METHODS | HASHMIX

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

FINDINGS | HASHMIX

6	0	2	0	4	0
Total Findings	Critical	Major	Medium	Minor	Informational

This report has been prepared to discover issues and vulnerabilities for HashMix. Through this audit, we have uncovered 6 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:

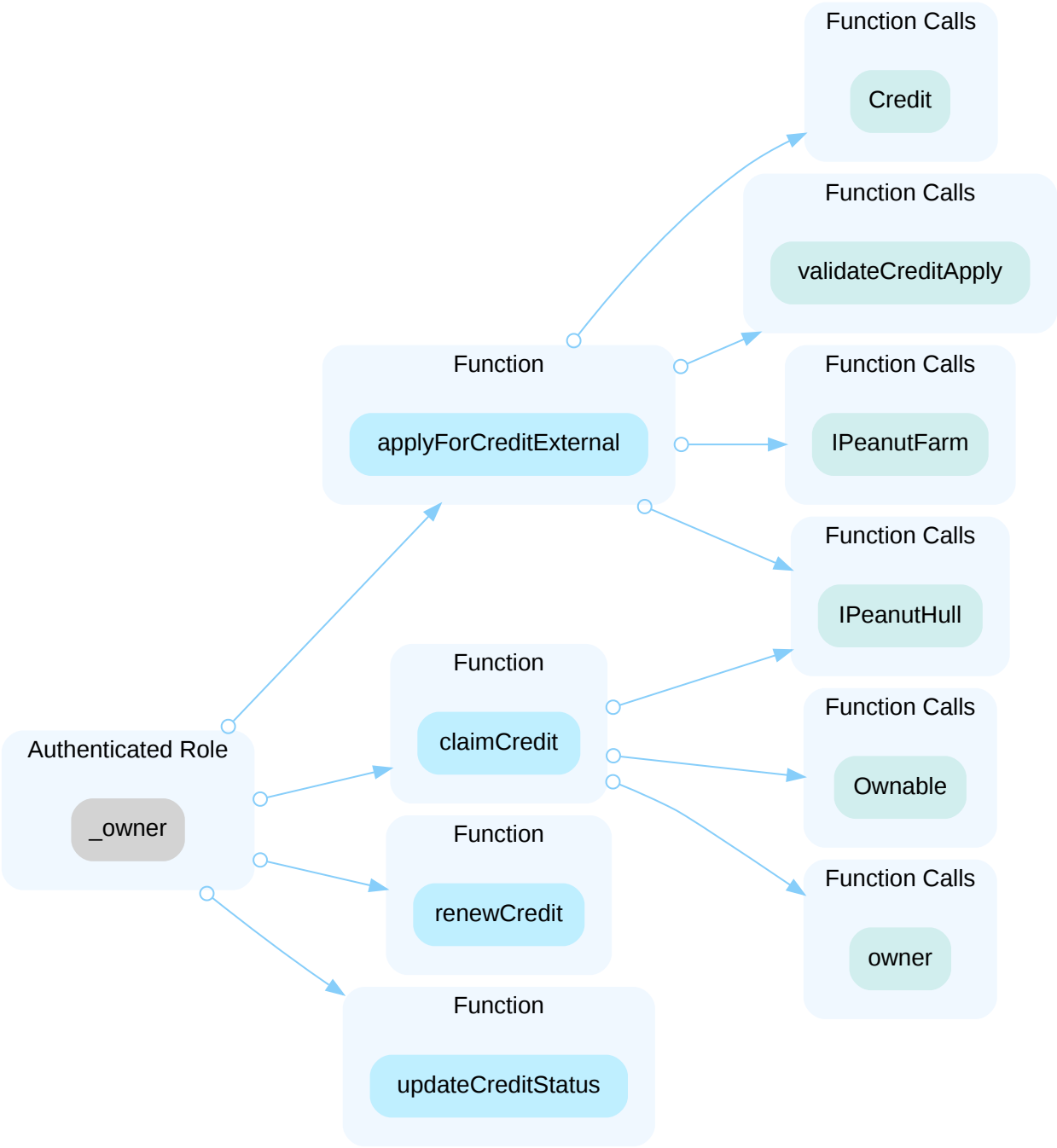
ID	Title	Category	Severity	Status
GLOBAL-01	**Centralization Related Risks**	**Centralization / Privilege**	**Major**	**Acknowledged**
GLOBAL-02	**Centralized Control Of Contract Upgrade**	**Centralization / Privilege**	**Major**	**Acknowledged**
GLOBAL-03	Out Of Scope Dependencies	Logical Issue	Minor	Acknowledged
HHM-01	Potential Unpaid Debt	Control Flow	Minor	Acknowledged
HHM-02	`harvest()` Not Support Token	Logical Issue	Minor	Acknowledged
HMP-01	Divide Before Multiply	Mathematical Operations	Minor	Acknowledged

GLOBAL-01 | CENTRALIZATION RELATED RISKS

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

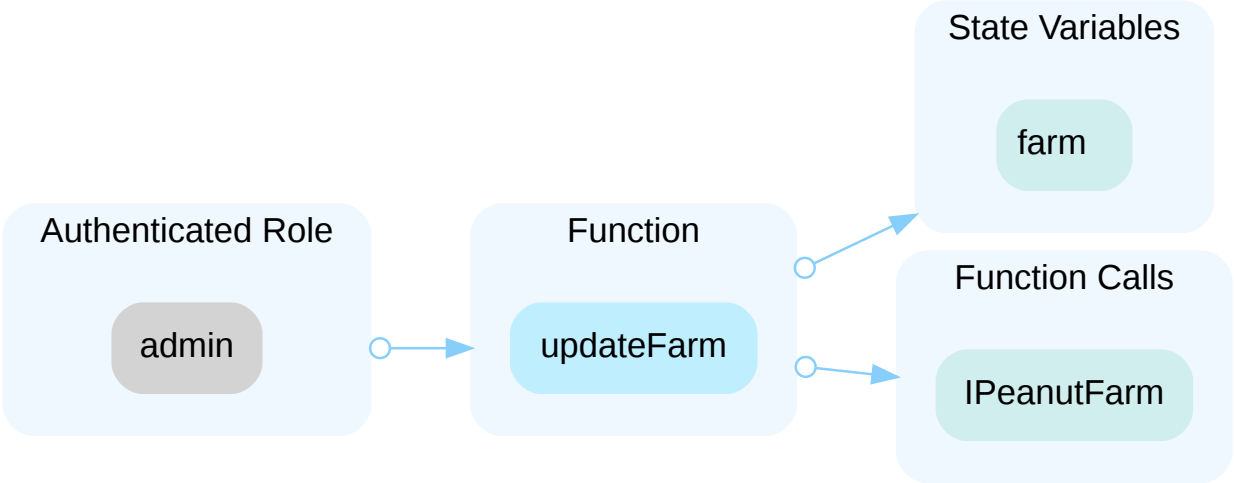
Description

In the contract `CreditsBook` 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.

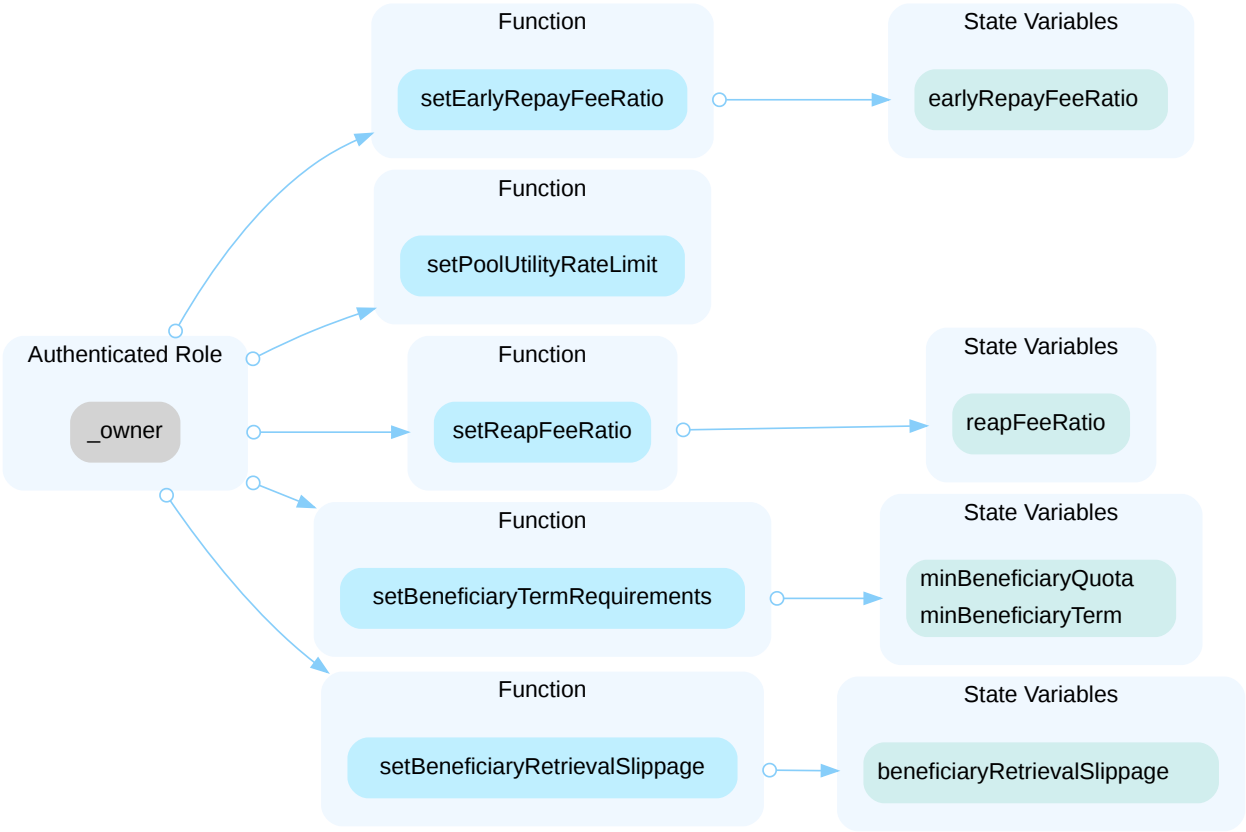


The contract `CreditsBook` 's owner is the contract `Hashmix` .

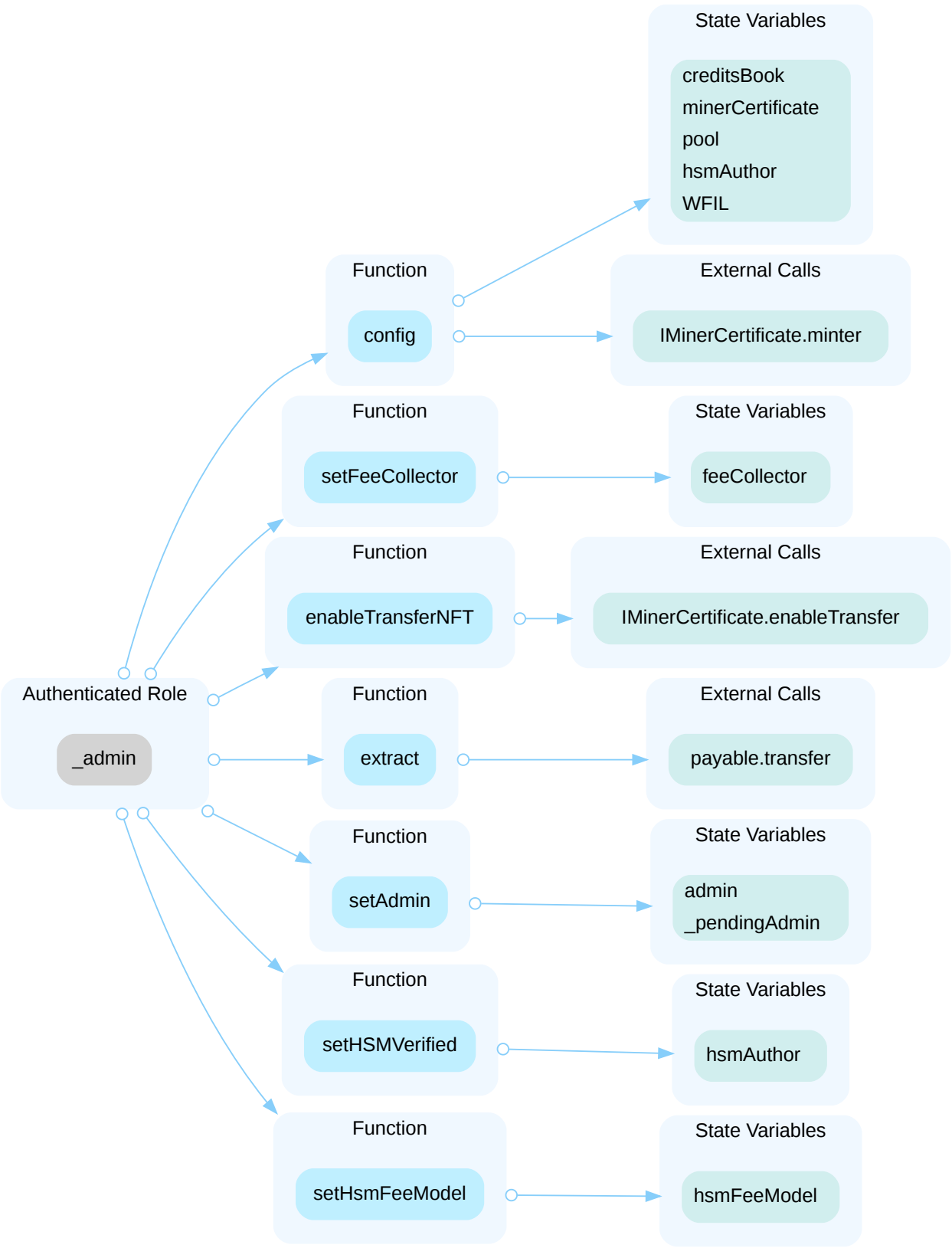
In the contract `CreditsBook` the role `admin` has authority over the functions shown in the diagram below. Any compromise to the `admin` account may allow the hacker to take advantage of this authority.



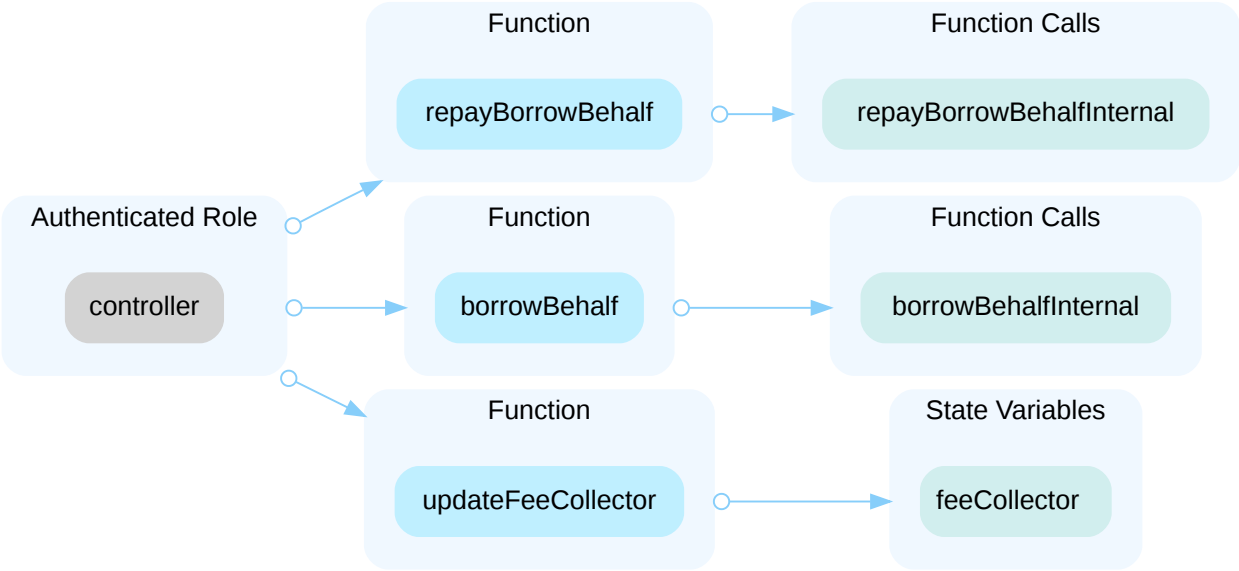
In the contract `FeeModel` the role `_owner` has authority over the functions shown in the diagram below. Any compromise to the `_owner` account may allow the hacker to take advantage of this authority.



In the contract `Hashmix` the role `admin` has authority over the functions shown in the diagram below. Any compromise to the `admin` account may allow the hacker to take advantage of this authority.

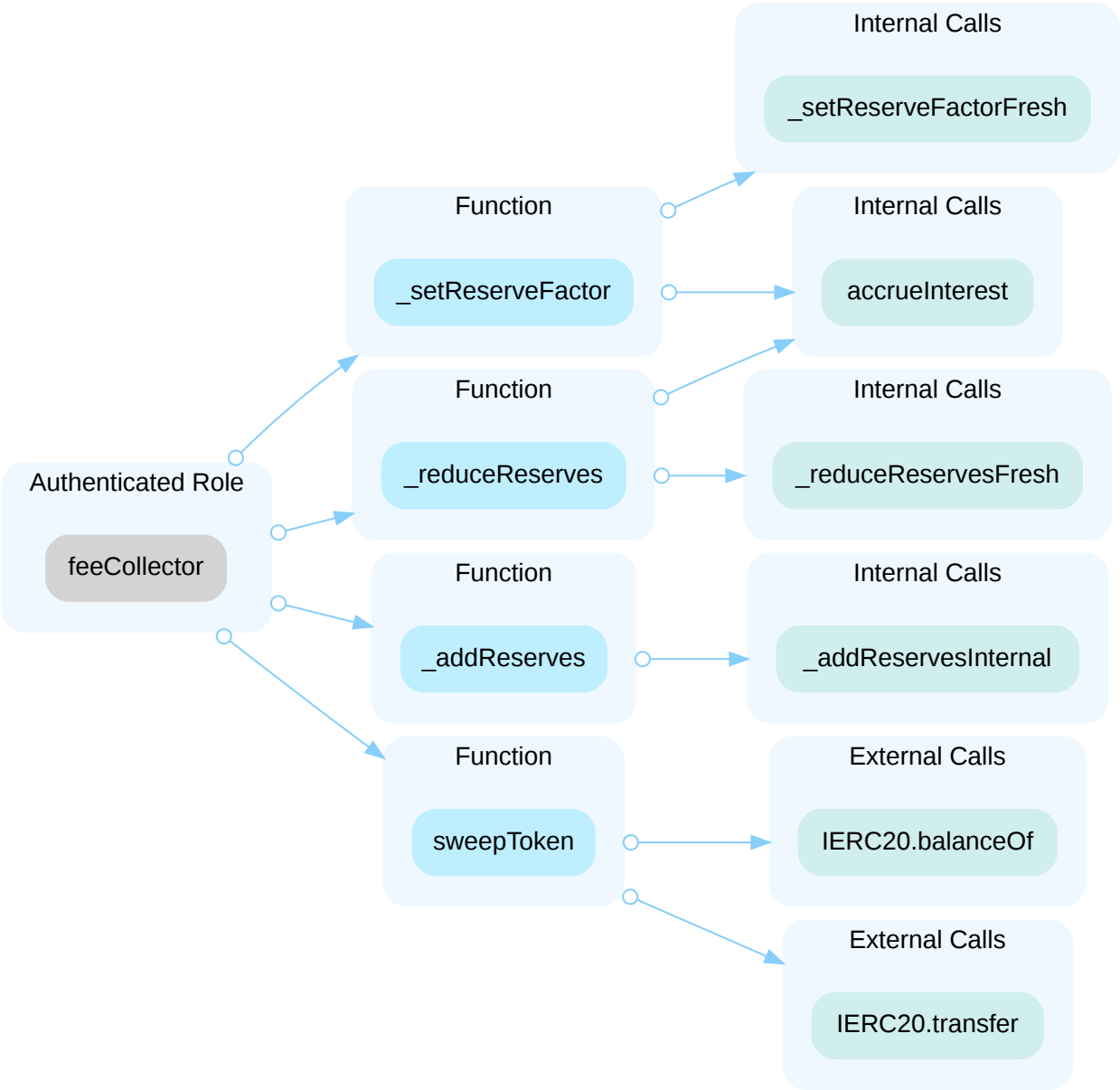


In the contract `HsmERC20` the role `controller` has authority over the functions shown in the diagram below. Any compromise to the `controller` account may allow the hacker to take advantage of this authority.

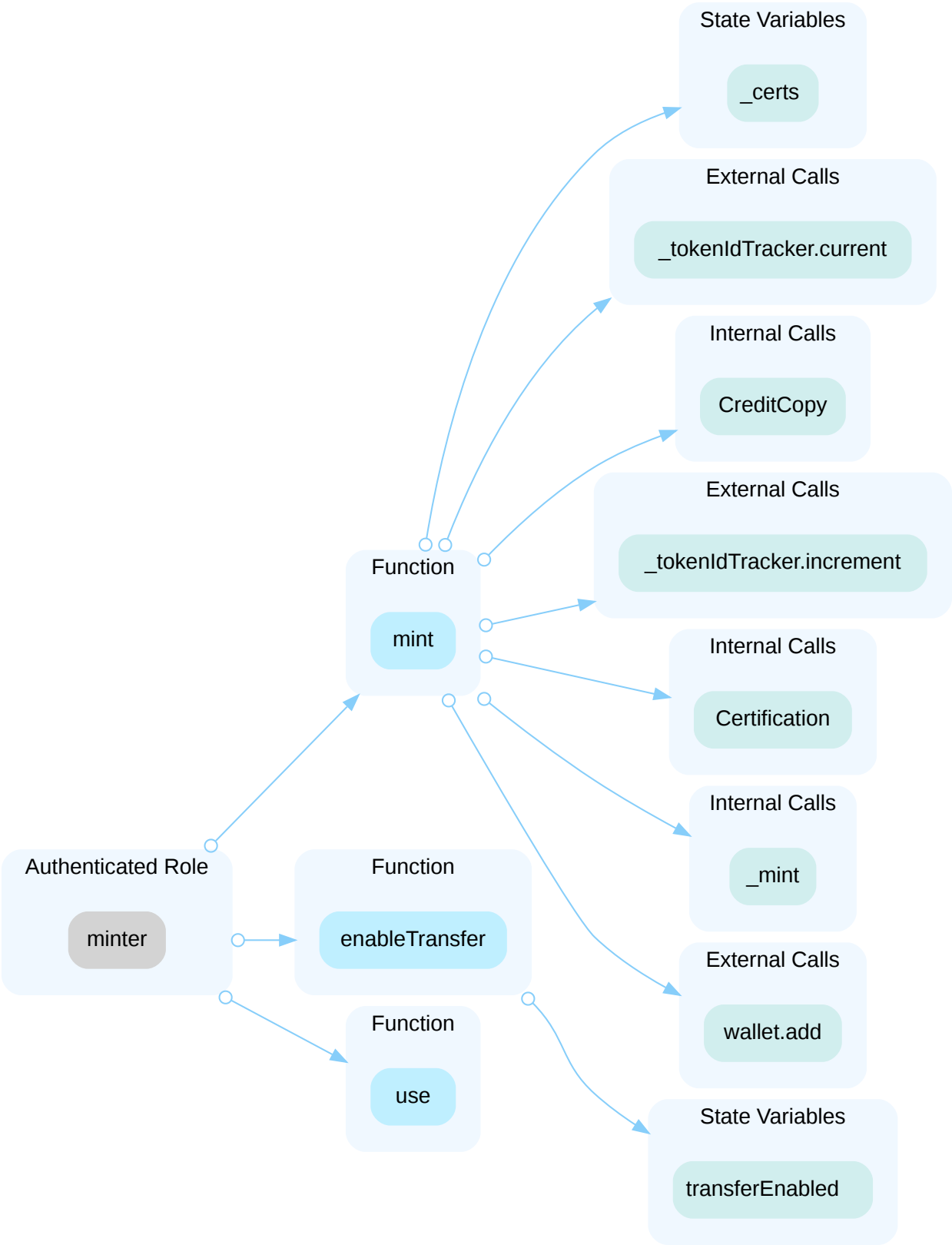


The contract `HsmERC20` 's controller is the contract `Hashmix` .

In the contract `HsmERC20` the role `feeCollector` has authority over the functions shown in the diagram below. Any compromise to the `feeCollector` account may allow the hacker to take advantage of this authority.

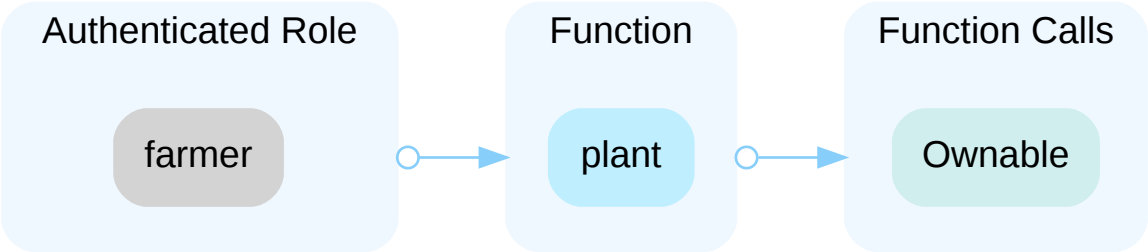


In the contract `MinerCertificate` the role `minter` has authority over the functions shown in the diagram below. Any compromise to the `minter` account may allow the hacker to take advantage of this authority.



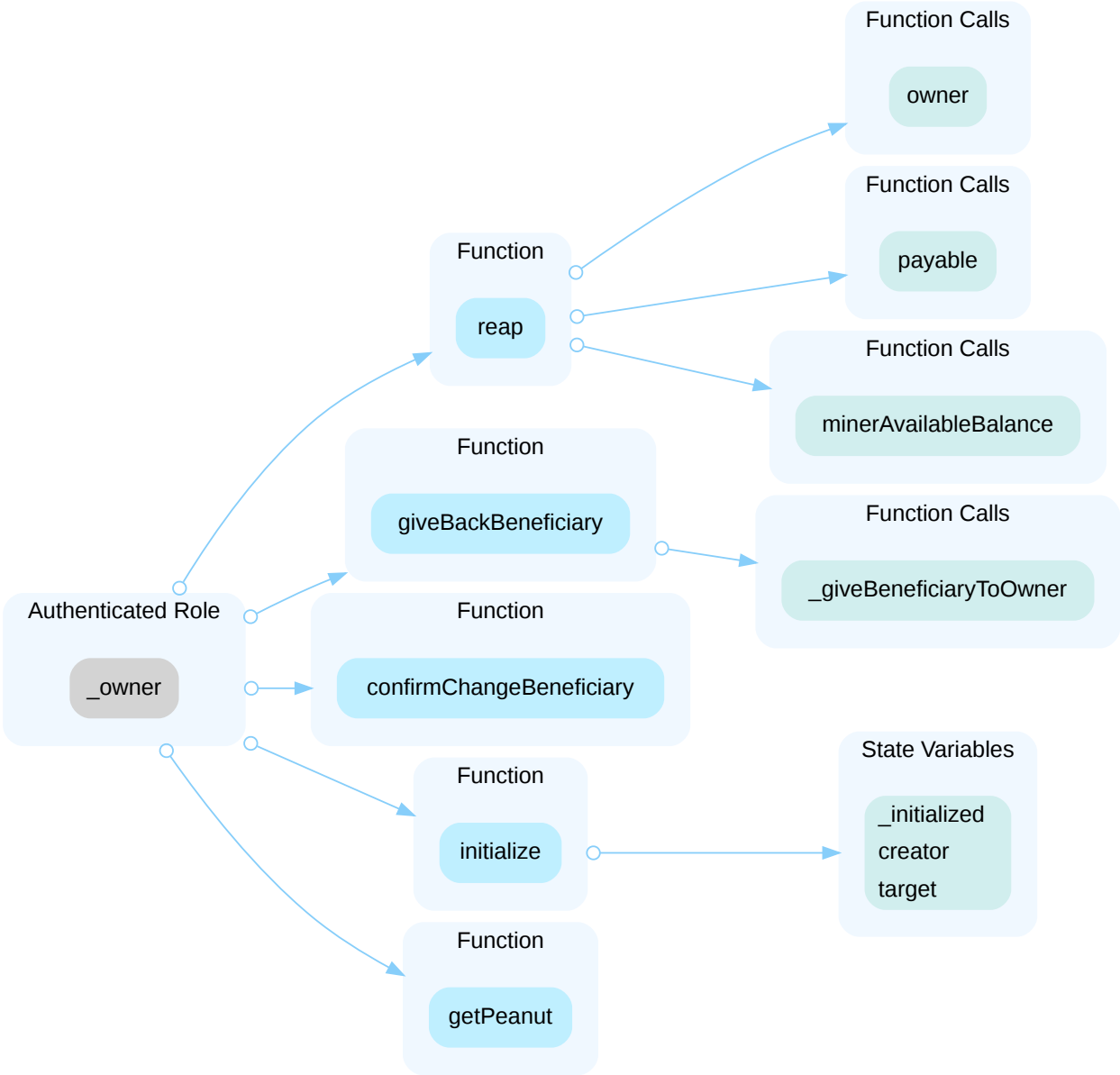
The contract `MinerCertificate`'s minter is the contract `Hashmix`.

In the contract `PeanutFarm` the role `farmer` has authority over the functions shown in the diagram below. Any compromise to the `farmer` account may allow the hacker to take advantage of this authority.



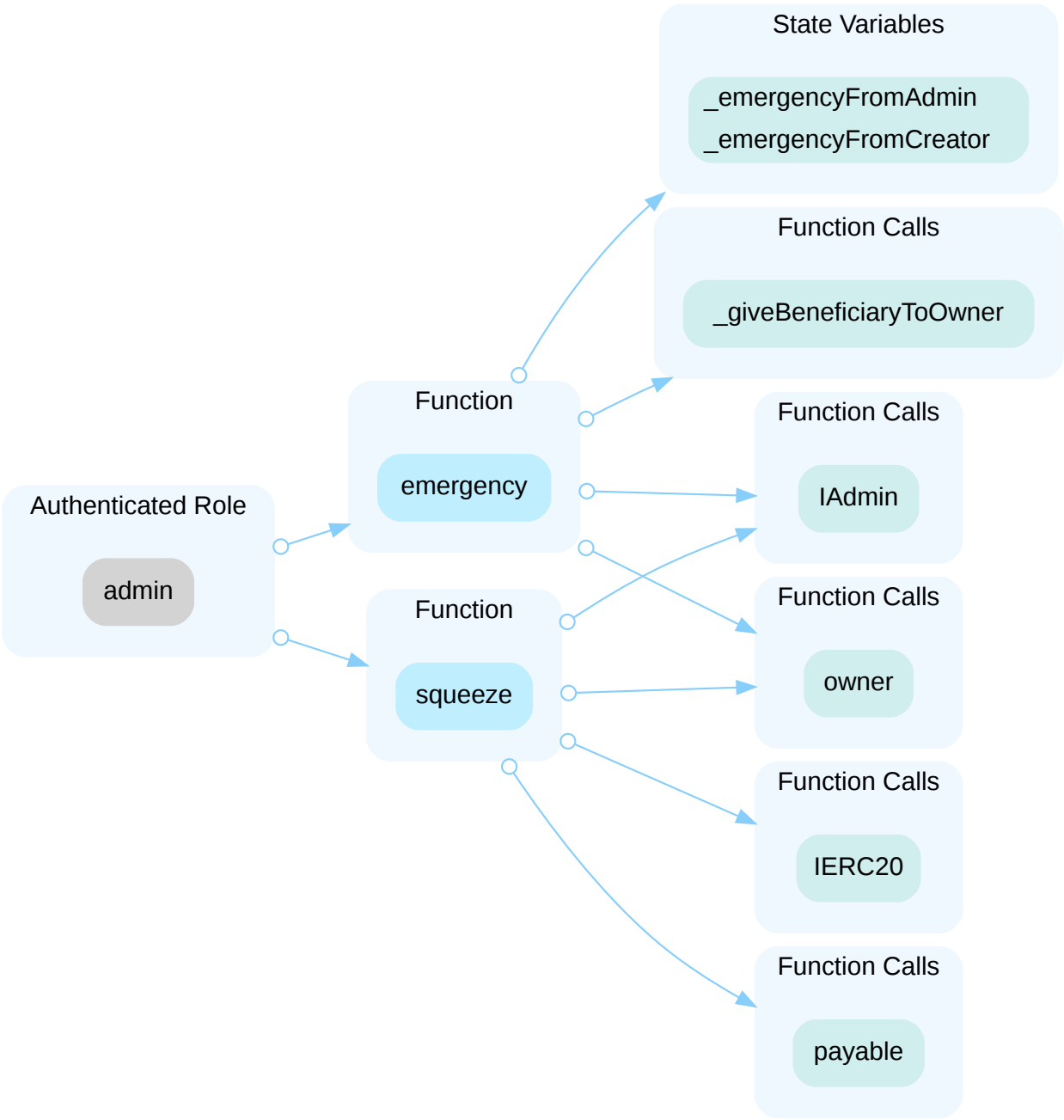
The contract `PeanutFarm` 's farmer is the contract `CreditsBook` .

In the contract `PeanutHull` 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.

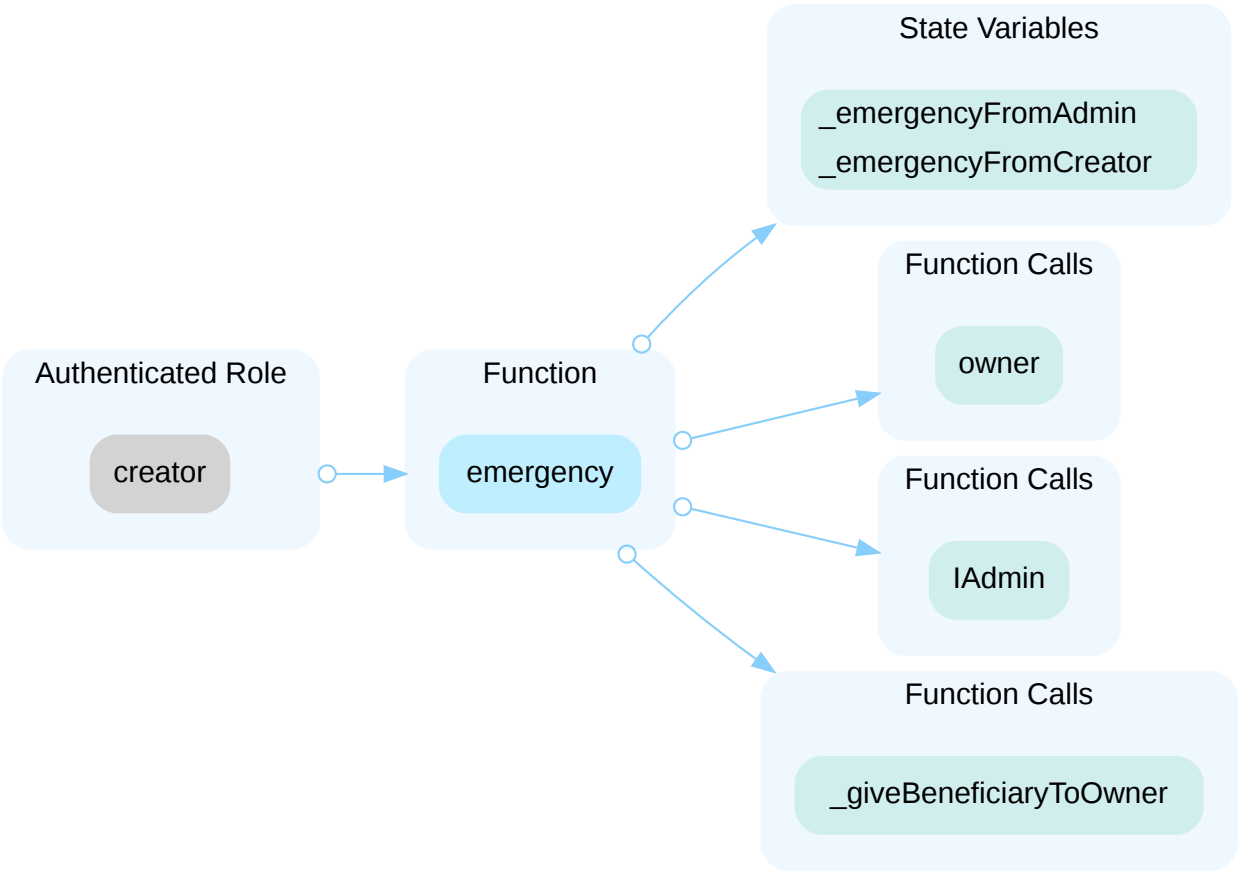


The contract `PeanutHull` 's owner is the contract `Hashmix` .

In the contract `PeanutHull` the role `admin` has authority over the functions shown in the diagram below. Any compromise to the `admin` account may allow the hacker to take advantage of this authority.



In the contract `PeanutHull` the role `creator` has authority over the functions shown in the diagram below. Any compromise to the `creator` account may allow the hacker to take advantage of this authority.



Recommendation

The risk describes the current project design and potentially makes iterations to improve in the security operation and level of decentralization, which in most cases cannot be resolved entirely at the present stage. We 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

[HashMix Team]: add multisig in commit: 9fdfef4894c7f22c2401ebe47f593c3ee87c9d38.

multisig address: 0x37878C623D87D5E99BEA602B9b72886676f3DEF0

change admin tx:

bafy2bzacecy2bo4ehup4adc5gdtrqeufhqobpyz7nrkt7jxdrsy5qctypo3gi
bafy2bzacecvzqedqqqyloxfioyh5nxdnmaz6itziejtf7u5j6xqvuhsivmsbc

GLOBAL-02 | CENTRALIZED CONTROL OF CONTRACT UPGRADE

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

Description

`CreditsBook`, `FeeModel`, and `Hashmix` are upgradeable contracts, the owner can upgrade the contract without the community's commitment. If an attacker compromises the account, he can change the implementation of the contract and drain tokens from the contract.

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

[HashMix Team]: add multisig in commit: 9fdfef4894c7f22c2401ebe47f593c3ee87c9d38

multisig address: 0x37878C623D87D5E99BEA602B9b72886676f3DEF0

change admin tx:

bafy2bzacecy2bo4ehup4adc5gdtrqeufhqobpyz7nrkt7jxdrsy5qctypo3gi
bafy2bzacecvzqedqqqyloxfioyh5nxdnmaz6itziejtf7u5j6xqvuhsivmsbc

GLOBAL-03 | OUT OF SCOPE DEPENDENCIES

Category	Severity	Location	Status
Logical Issue	Minor		Acknowledged

Description

The project HashMix serves as the underlying entity to interact with `zondax` repository. The scope of the audit treats contract that is out of scope as black boxes and assumes their functional correctness.

Recommendation

The aforementioned repository is out of the audit scope. We encourage the team to constantly monitor the status of those contracts and ensure their security and functionality correctness.

Alleviation

[HashMix Team] we are counting on zondax's correctness. that is out of our control. as far as the current version, it functions correctly.

HHM-01 | POTENTIAL UNPAID DEBT

Category	Severity	Location	Status
Control Flow	Minor	contracts/Hashmix.sol: 593	Acknowledged

Description

If `peanutHull` still has less debt left than `BeneficiaryRetrievalSlippage`, then the function `giveBackBeneficiarySafetyChecks()` checks will pass.

```
591         if (
592             debt >
593             IHashmixFeeModel(hsmFeeModel).getBeneficiaryRetrievalSlippage()
594         ) {
595             revert DebtNotClean();
596         }
```

Recommendation

We recommend that the beneficiary be given back only when the debt equals zero.

Alleviation

[HashMix Team]: this is the designed behavior.

HHM-02 | `harvest()` NOT SUPPORT TOKEN

Category	Severity	Location	Status
Logical Issue	Minor	contracts/Hashmix.sol: 508	Acknowledged

Description

The function `harvest()` is designed to retrieve the miner's mining reward and repay the debt, but if the underlying asset is token, then the call to the `harvest()` function will fail and the miner will not be able to retrieve the reward.

Recommendation

We recommend modifying the code logic to support Token.

Alleviation

[HashMix Team] harvest is now only meant for native FIL

HMP-01 | DIVIDE BEFORE MULTIPLY

Category	Severity	Location	Status
Mathematical Operations	Minor	contracts/JumpRateModel.sol: 144, 145~148; contracts/WhitePaperInterestRateModel.sol: 95, 96	Acknowledged

Description

Performing integer division before multiplication truncates the low bits, losing the precision of calculation.

```
144         uint rateToPool = (borrowRate * oneMinusReserveFactor) / BASE;
```

```
145         return
146             (utilizationRate(cash, borrows, reserves) * rateToPool) /
147             BASE /
148             PERBLOCKBASE;
```

```
 95         uint rateToPool = (borrowRate * oneMinusReserveFactor) / BASE;
```

```
 96         return (utilizationRate(cash, borrows, reserves) * rateToPool) / BASE;
```

Recommendation

We recommend applying multiplication before division to avoid loss of precision.

Alleviation

[HashMix Team]: Issue acknowledged. I won't make any changes for the current version.

APPENDIX | HASHMIX

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.
Mathematical Operations	Mathematical Operation findings relate to mishandling of math formulas, such as overflows, incorrect operations etc.
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.
Control Flow	Control Flow findings concern the access control imposed on functions, such as owner-only functions being invoke-able by anyone under certain circumstances.

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.