



QuillAudits



Audit Report  
May, 2021

 POLKARARE



# Contents

Introduction	01
Audit Goals	02
Issue Categories	03
Manual Audit	04
Automated Testing	09
Summary	16
Disclaimer	17

# Introduction

This audit report highlights the overall security of the PolkaRareToken with commit hash e8fab81 & PolkaRareTokenVesting with the commit hash 65ec9d6. After the initial audit report there were some changes made at commit d0de16e0be2c. There was another revision at commit 00861ca. With this report, we have tried to ensure the reliability of the smart contract by completing the assessment of their system's architecture and smart contract codebase.

## Auditing Approach and Methodologies applied

In this audit, we consider the following crucial features of the code.

- Whether the implementation of ERC 20 standards.
- Whether the code is secure.
- Gas Optimization
- Whether the code meets the best coding practices.
- Whether the code meets the SWC Registry issue.

The audit has been performed according to the following procedure:

### Manual Audit

- Inspecting the code line by line and revert the initial algorithms of the protocol and then compare them with the specification
- Manually analyzing the code for security vulnerabilities.
- Gas Consumption and optimisation
- Assessing the overall project structure, complexity & quality.
- Checking SWC Registry issues in the code.
- Unit testing by writing custom unit testing for each function.
- Checking whether all the libraries used in the code of the latest version.
- Analysis of security on-chain data.
- Analysis of the failure preparations to check how the smart contract performs in case of bugs and vulnerability.

### Automated analysis

- Scanning the project's code base with Mythril, Slither, Echidna, Manticore, others.
- Manually verifying (reject or confirm) all the issues found by tools.
- Performing Unit testing.



- Manual Security Testing (SWC-Registry, Overflow)
- Running the tests and checking their coverage.

**Report:** All the gathered information is described in this report.

## Audit Details

**Project Name:** teamPolkaRare

**Token symbol:** PRARE

**Languages:** Solidity

**Platforms and Tools:** HardHat, Remix, VScode, solhint and other tools mentioned in the automated analysis section.

**Audit-scope:**

<https://github.com/teampolkarare/ICO-Contracts/blob/main/contracts/PolkaRareToken.sol>

<https://github.com/teampolkarare/ICO-Contracts/blob/main/contracts/PolkaRareTokenVesting.sol>

## Audit Goals

The focus of this audit was to verify whether the smart contract is secure, resilient, and working according to ERC20 specs. The audit activity can be grouped in three categories.

### Security

Identifying security related issues within each contract and the system of contract.

### Sound Architecture

Evaluating the architect of a system through the lens of established smart contract best practice and general software practice.

### Code Correctness and Quality

A full review of the contract source code. The primary areas of focus include:

- Correctness.
- Section of code with high complexity.
- Readability.
- Quantity and quality of test coverage.

# Issue Categories

Every issue in this report was assigned a severity level from the following:

## High severity issues

Issues on this level are critical to the smart contract’s performance/ functionality and should be fixed before moving to a live environment.

## Medium severity issues

Issues on this level could potentially bring problems and should eventually be fixed.

## Low severity issues

Issues on this level are minor details and warnings that can remain unfixed but would be better fixed at some point in the future.

## Informational

These are severity four issues which indicate an improvement request, a general question, a cosmetic or documentation error, or a request for information. There is low-to-no impact.

## Number of issues per severity

	High	Medium	Low	Informational
Open	0	0	2	2
Closed	0	1	1	0



# Manual Audit

## SWC Registry test

We have tested some known SWC registry issues. Out of all tests only SWC 102 and 103. Both are low priority. We have about it above already.

Serial No.	Description	Comments
<a href="#">SWC-132</a>	Unexpected Ether balance	Pass: Avoided strict equality checks for the Ether balance in a contract
<a href="#">SWC-131</a>	Presence of unused variables	Pass: No unused variables
<a href="#">SWC-128</a>	DoS With Block Gas Limit	Pass
<a href="#">SWC-122</a>	Lack of Proper Signature Verification	Pass
<a href="#">SWC-120</a>	Weak Sources of Randomness from Chain Attributes	<b>Found:</b> No random value used insufficiently (Found in Mythx also)
<a href="#">SWC-119</a>	Shadowing State Variables	Pass: No ambiguous found.
<a href="#">SWC-118</a>	Incorrect Constructor Name	Pass. No incorrect constructor name used
<a href="#">SWC-116</a>	Timestamp Dependence	Pass
<a href="#">SWC-115</a>	Authorization through tx.origin	Pass: No tx.origin found
<a href="#">SWC-114</a>	Transaction Order Dependence	Pass

Serial No.	Description	Comments
<u>SWC-113</u>	DoS with Failed Call	Pass: No failed call
<u>SWC-112</u>	Delegatecall to Untrusted Callee	Pass
<u>SWC-111</u>	Use of Deprecated Solidity Functions	Found : link
<u>SWC-108</u>	State Variable Default Visibility	Pass: Explicitly defined visibility for all state variables
<u>SWC-107</u>	Reentrancy	Pass
<u>SWC-106</u>	Unprotected SELF-DESTRUCT Instruction	Pass: Not found any such vulnerability
<u>SWC-104</u>	Unchecked Call Return Value	Pass: Not found any such vulnerability
<u>SWC-103</u>	Floating Pragma	Fixed
<u>SWC-102</u>	Outdated Compiler Version	Pass
<u>SWC-101</u>	Integer Overflow and Underflow	Pass

High level severity issues

No issues found



## Medium level severity issues

There was 1 severity issue found.

1. In the PolkaRareTokenVesting contract → marketing vesting, operationsVesting, reservesVesting, ecosystemVesting, doesn't match with <https://tokenomics.polkarare.com/>. There is an unequal amount of token released every month whereas in tokenomics its written equal amount of token will be released every month. Please have a note on this.

**Status:** Closed at [00861ca](#)

## Low level severity issues

There were 3 low severity issues found.

1. **Description** → **SWC 102: Outdated Compiler Version**

```
PolkaRareTokenVesting.sol ×
contracts > PolkaRareTokenVesting.sol
  report | graph (this) | graph | inheritance | parse | flatten | funcSigs | uml
1  // SPDX-License-Identifier: MIT
2
3  pragma solidity 0.7.4;
4
```

Using an outdated compiler version can be problematic especially if there are publicly disclosed bugs and issues that affect the current compiler version.

### Remediation

It is recommended to use a recent version of the Solidity compiler which is Version 0.8.4.

**Status:** Fixed; In the next iteration it got upgraded to the latest version.



## 2. Description: Use of "block.timestamp" → Block values as a proxy for time[line 320 in PolkarareTokenVesting.sol ] → SWC 120

```
318         returns (uint256 _availablePercentage)
319     {
320         uint256 currentTimeStamp = block.timestamp;
321         uint256 noOfDays = BokkyPooBahsDateTimeLibrary.diffDays(_initialTimestamp, current
322         uint256 noOfMonths = _daysToMonths(noOfDays);
323     }
```

Contracts often need access to time values to perform certain types of functionality. Values such as `block.timestamp`, and `block.number` can give you a sense of the current time or a time delta, however, they are not safe to use for most purposes.

In the case of `block.timestamp`, developers often attempt to use it to trigger time-dependent events. As Ethereum is decentralized, nodes can synchronize time only to some degree. Moreover, malicious miners can alter the timestamp of their blocks, especially if they can gain advantages by doing so. However, miners can't set a timestamp smaller than the previous one (otherwise the block will be rejected), nor can they set the timestamp too far ahead in the future. Taking all of the above into consideration, developers can't rely on the preciseness of the provided timestamp.

### Remediation

Developers should write smart contracts with the notion that block values are not precise, and the use of them can lead to unexpected effects

### References

- [Safety: Timestamp dependence](#)
- [Ethereum Smart Contract Best Practices - Timestamp Dependence](#)
- [How do Ethereum mining nodes maintain a time consistent with the network?](#)
- [Solidity: Timestamp dependency, is it possible to do safely?](#)
- [Avoid using block.number as a timestamp](#)

**Status: Open**



### 3. Description: Prefer external to public visibility level [“setGovernance” function in PolkaRareToken.sol & “getInitialTimestamp”, “withdrawableTokens” function in PolkaRareTokenVesting.sol ]

A function with a **public** visibility modifier that is not called internally. Changing the visibility level to **external** increases code readability. Moreover, in many cases, functions with **external** visibility modifiers spend less gas compared to functions with **public** visibility modifiers.

The function definition of “setGovernance” in **PolkaRareToken.sol** and the function definition of “getInitialTimestamp” and “withdrawableTokens” of **PolkaRareTokenVesting.sol** is marked as “public”. However, it is never directly called by another function in the same contract or in any of its descendants. Consider to mark it as “external” instead

#### Recommendations:

Use the **external** visibility modifier for functions never called from the contract via internal call. Reading [Link](#).

**Status: Open**

## Informational

### 1. Coding Style

There were many coding style issues found by solhint. It needed to be fixed as part of best practice.

### 2. Pausable function missing

A pausable contract provides extra layer of security in the token contract during emergency. The current token contract doesn't include it. Reading [link](#).



# Automated Testing

We have used multiple automated testing frameworks. This makes code more secure common attacks. The results are below.

## Slither

Slither is a Solidity static analysis framework which runs a suite of vulnerability detectors, prints visual information about contract details, and provides an API to easily write custom analyses. Slither enables developers to find vulnerabilities, enhance their code comprehension, and quickly prototype custom analyses. After running Slither we got results below.

```
5 | import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
  | ~~~~~
Error: Source "@openzeppelin/contracts/token/ERC20/IERC20.sol" not found: File not found.
--> ./PolkaRareToken.sol:6:1:
6 | import "@openzeppelin/contracts/token/ERC20/IERC20.sol";
  | ~~~~~
Error: Source "@openzeppelin/contracts/access/Ownable.sol" not found: File not found.
--> ./PolkaRareToken.sol:7:1:
7 | import "@openzeppelin/contracts/access/Ownable.sol";
  | ~~~~~
Error: Source "@openzeppelin/contracts/math/SafeMath.sol" not found: File not found.
--> ./PolkaRareToken.sol:8:1:
8 | import "@openzeppelin/contracts/math/SafeMath.sol";
  | ~~~~~
Error: Source "@openzeppelin/contracts/token/ERC20/SafeERC20.sol" not found: File not found.
--> ./PolkaRareToken.sol:9:1:
9 | import "@openzeppelin/contracts/token/ERC20/SafeERC20.sol";
```

```
9 | import "@openzeppelin/contracts/token/ERC20/SafeERC20.sol";
  | ~~~~~
Error: Source "@openzeppelin/contracts/token/ERC20/ERC20.sol" not found: File not found.
--> ./ERC20Permit.sol:5:1:
5 | import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
  | ~~~~~
Error: Source "@openzeppelin/contracts/utils/Counters.sol" not found: File not found.
--> ./ERC20Permit.sol:6:1:
6 | import "@openzeppelin/contracts/utils/Counters.sol";
  | ~~~~~
```



```

Compilation warnings/errors on ./PolkaRareToken.sol:
Error: Source "@openzeppelin/contracts/token/ERC20/ERC20.sol" not found: File not found.
--> ./PolkaRareToken.sol:5:1:
5 | import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
  | ~~~~~

Error: Source "@openzeppelin/contracts/token/ERC20/IERC20.sol" not found: File not found.
--> ./PolkaRareToken.sol:6:1:
6 | import "@openzeppelin/contracts/token/ERC20/IERC20.sol";
  | ~~~~~

Error: Source "@openzeppelin/contracts/access/Ownable.sol" not found: File not found.
--> ./PolkaRareToken.sol:7:1:
7 | import "@openzeppelin/contracts/access/Ownable.sol";
  | ~~~~~

Error: Source "@openzeppelin/contracts/math/SafeMath.sol" not found: File not found.
--> ./PolkaRareToken.sol:8:1:
8 | import "@openzeppelin/contracts/math/SafeMath.sol";
  | ~~~~~

Error: Source "@openzeppelin/contracts/token/ERC20/SafeERC20.sol" not found: File not found.

```

```

~~~~~
Error: Source "@openzeppelin/contracts/math/SafeMath.sol" not found: File not found.
--> ./PolkaRareToken.sol:8:1:
8 | import "@openzeppelin/contracts/math/SafeMath.sol";
  | ~~~~~

Error: Source "@openzeppelin/contracts/token/ERC20/SafeERC20.sol" not found: File not found.
--> ./PolkaRareToken.sol:9:1:
9 | import "@openzeppelin/contracts/token/ERC20/SafeERC20.sol";
  | ~~~~~

Error: Source "@openzeppelin/contracts/token/ERC20/ERC20.sol" not found: File not found.
--> ./ERC20Permit.sol:5:1:
5 | import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
  | ~~~~~

Error: Source "@openzeppelin/contracts/utils/Counters.sol" not found: File not found.
--> ./ERC20Permit.sol:6:1:
6 | import "@openzeppelin/contracts/utils/Counters.sol";
  | ~~~~~

```



```

5 | import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
  | ~~~~~
Error: Source "@openzeppelin/contracts/token/ERC20/IERC20.sol" not found: File not found.
--> ./PolkaRareToken.sol:6:1:
6 | import "@openzeppelin/contracts/token/ERC20/IERC20.sol";
  | ~~~~~
Error: Source "@openzeppelin/contracts/access/Ownable.sol" not found: File not found.
--> ./PolkaRareToken.sol:7:1:
7 | import "@openzeppelin/contracts/access/Ownable.sol";
  | ~~~~~
Error: Source "@openzeppelin/contracts/math/SafeMath.sol" not found: File not found.
--> ./PolkaRareToken.sol:8:1:
8 | import "@openzeppelin/contracts/math/SafeMath.sol";
  | ~~~~~
Error: Source "@openzeppelin/contracts/token/ERC20/SafeERC20.sol" not found: File not found.
--> ./PolkaRareToken.sol:9:1:
9 | import "@openzeppelin/contracts/token/ERC20/SafeERC20.sol";

```

```

8 | import "@openzeppelin/contracts/math/SafeMath.sol";
  | ~~~~~
Error: Source "@openzeppelin/contracts/token/ERC20/SafeERC20.sol" not found: File not found.
--> ./PolkaRareToken.sol:9:1:
9 | import "@openzeppelin/contracts/token/ERC20/SafeERC20.sol";
  | ~~~~~
Error: Source "@openzeppelin/contracts/token/ERC20/ERC20.sol" not found: File not found.
--> ./ERC20Permit.sol:5:1:
5 | import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
  | ~~~~~
Error: Source "@openzeppelin/contracts/utils/Counters.sol" not found: File not found.
--> ./ERC20Permit.sol:6:1:
6 | import "@openzeppelin/contracts/utils/Counters.sol";
  | ~~~~~

```

**Description:** Slither was unable to import the openzeppelin library. Library formatting can be fixed there.

**Status** → There was an upgrade done after the initial report which reduces the warnings and errors. Below are the results.



```

Compilation warnings/errors on ./PolkaRareToken.sol:
Error: Source "@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol" not found: File not found.
--> ./PolkaRareToken.sol:5:1:
5 | import "@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol";
  | ~~~~~

Error: Source "@openzeppelin/contracts/utils/math/SafeMath.sol" not found: File not found.
--> ./PolkaRareToken.sol:6:1:
6 | import "@openzeppelin/contracts/utils/math/SafeMath.sol";
  | ~~~~~

Error: Source "@openzeppelin/contracts/access/Ownable.sol" not found: File not found.
--> ./PolkaRareToken.sol:7:1:
7 | import "@openzeppelin/contracts/access/Ownable.sol";
  | ~~~~~

Error: Source "@openzeppelin/contracts/token/ERC20/ERC20.sol" not found: File not found.
--> ./PolkaRareToken.sol:8:1:
8 | import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
  | ~~~~~

Error: Source "@openzeppelin/contracts/token/ERC20/IERC20.sol" not found: File not found.

```

```

7 | import "@openzeppelin/contracts/access/Ownable.sol";
  | ~~~~~

Error: Source "@openzeppelin/contracts/token/ERC20/ERC20.sol" not found: File not found.
--> ./PolkaRareToken.sol:8:1:
8 | import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
  | ~~~~~

Error: Source "@openzeppelin/contracts/token/ERC20/IERC20.sol" not found: File not found.
--> ./PolkaRareToken.sol:9:1:
9 | import "@openzeppelin/contracts/token/ERC20/IERC20.sol";
  | ~~~~~

Error: Source "@openzeppelin/contracts/token/ERC20/ERC20.sol" not found: File not found.
--> ./ERC20Permit.sol:5:1:
5 | import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
  | ~~~~~

Error: Source "@openzeppelin/contracts/utils/Counters.sol" not found: File not found.
--> ./ERC20Permit.sol:6:1:
6 | import "@openzeppelin/contracts/utils/Counters.sol";
  | ~~~~~

```



## Manticore

Manticore is a symbolic execution tool for the analysis of smart contracts and binaries. It executes a program with symbolic inputs and explores all the possible states it can reach. It also detects crashes and other failure cases in binaries and smart contracts.

Manticore results throw the same warning which is similar to the Slither warning.

## Solhint

Solhint is an open source project created by <https://protofire.io>. Its goal is to provide a linting utility for Solidity code. Below are the results from Solhint.

```
contracts/BokkyPooBahsDateTimeLibrary.sol
31:5   warning Explicitly mark visibility of state      state-visibility
31:22  error     'SECONDS_PER_DAY' should start with _    private-vars-leading-underscore
32:5   warning Explicitly mark visibility of state      state-visibility
32:22  error     'SECONDS_PER_HOUR' should start with _    private-vars-leading-underscore
33:5   warning Explicitly mark visibility of state      state-visibility
33:22  error     'SECONDS_PER_MINUTE' should start with _  private-vars-leading-underscore
34:5   warning Explicitly mark visibility of state      state-visibility
34:21  error     'OFFSET19700101' should start with _        private-vars-leading-underscore
36:5   warning Explicitly mark visibility of state      state-visibility
36:22  error     'DOW_MON' should start with _              private-vars-leading-underscore
37:5   warning Explicitly mark visibility of state      state-visibility
37:22  error     'DOW_TUE' should start with _              private-vars-leading-underscore
38:5   warning Explicitly mark visibility of state      state-visibility
38:22  error     'DOW_WED' should start with _              private-vars-leading-underscore
39:5   warning Explicitly mark visibility of state      state-visibility
39:22  error     'DOW_THU' should start with _              private-vars-leading-underscore
40:5   warning Explicitly mark visibility of state      state-visibility
40:22  error     'DOW_FRI' should start with _              private-vars-leading-underscore
41:5   warning Explicitly mark visibility of state      state-visibility
41:22  error     'DOW_SAT' should start with _              private-vars-leading-underscore
42:5   warning Explicitly mark visibility of state      state-visibility
42:22  error     'DOW_SUN' should start with _              private-vars-leading-underscore
57:5   error     '_daysFromDate' should not start with _  private-vars-leading-underscore
62:9   warning Provide an error message for require     reason-string

42:22  error     'DOW_SUN' should start with _              private-vars-leading-underscore
57:5   error     '_daysFromDate' should not start with _  private-vars-leading-underscore
62:9   warning Provide an error message for require     reason-string
98:5   error     '_daysToDate' should not start with _    private-vars-leading-underscore
109:9  warning Variable name must be in mixedCase         var-name-mixedcase
110:9  warning Variable name must be in mixedCase         var-name-mixedcase
216:5  error     '_isLeapYear' should not start with _     private-vars-leading-underscore
233:5  error     '_getDaysInMonth' should not start with _  private-vars-leading-underscore
283:9  warning Provide an error message for require     reason-string
296:9  warning Provide an error message for require     reason-string
301:9  warning Provide an error message for require     reason-string
306:9  warning Provide an error message for require     reason-string
311:9  warning Provide an error message for require     reason-string
316:9  warning Provide an error message for require     reason-string
327:9  warning Provide an error message for require     reason-string
340:9  warning Provide an error message for require     reason-string
345:9  warning Provide an error message for require     reason-string
350:9  warning Provide an error message for require     reason-string
355:9  warning Provide an error message for require     reason-string
360:9  warning Provide an error message for require     reason-string
364:9  warning Provide an error message for require     reason-string
371:9  warning Provide an error message for require     reason-string
378:9  warning Provide an error message for require     reason-string
383:9  warning Provide an error message for require     reason-string
388:9  warning Provide an error message for require     reason-string
393:9  warning Provide an error message for require     reason-string
```



```

contracts/ERC20Permit.sol
17:20 warning Variable name must be in mixedCase var-name-mixedcase
21:9 warning Avoid to use inline assembly. It is acceptable only in rare cases no-inline-assembly

contracts/PolkaRareTokenVesting.sol
53:5 warning Explicitly mark visibility of state state-visibility
55:5 warning Explicitly mark visibility of state state-visibility
55:15 error 'ecosystemVesting' should start with _ private-vars-leading-underscore
74:5 warning Explicitly mark visibility of state state-visibility
74:15 error 'marketingVesting' should start with _ private-vars-leading-underscore
98:5 warning Explicitly mark visibility of state state-visibility
98:15 error 'teamVesting' should start with _ private-vars-leading-underscore
123:5 warning Explicitly mark visibility of state state-visibility
123:15 error 'reservesVesting' should start with _ private-vars-leading-underscore
152:5 warning Explicitly mark visibility of state state-visibility
152:15 error 'operationsVesting' should start with _ private-vars-leading-underscore
186:5 warning Explicitly mark visibility of state state-visibility
186:15 error 'miningStakingVesting' should start with _ private-vars-leading-underscore

123:15 error 'reservesVesting' should start with _ private-vars-leading-underscore
152:5 warning Explicitly mark visibility of state state-visibility
152:15 error 'operationsVesting' should start with _ private-vars-leading-underscore
186:5 warning Explicitly mark visibility of state state-visibility
186:15 error 'miningStakingVesting' should start with _ private-vars-leading-underscore
315:5 error Function has cyclomatic complexity 8 but allowed no more than 7 code-complexity

contracts/PrivateDistribution.sol
30:21 error 'vestingMonth' should start with _ private-vars-leading-underscore
42:5 warning Explicitly mark visibility of state state-visibility
42:15 error 'seedVesting' should start with _ private-vars-leading-underscore
73:5 warning Explicitly mark visibility of state state-visibility
73:15 error 'privateOneVesting' should start with _ private-vars-leading-underscore
103:5 warning Explicitly mark visibility of state state-visibility
103:15 error 'privateTwoVesting' should start with _ private-vars-leading-underscore

* 70 problems (26 errors, 44 warnings)
error Command failed with exit code 1.
info Visit https://yarnpkg.com/en/docs/cli/run for documentation about this command.

```

There were 26 errors and 44 warnings from solhint in the whole codebase. Status: There was some fix done in the next iteration. Below are the results.



```
123:5 warning Explicitly mark visibility of state state-visib
ility
152:5 warning Explicitly mark visibility of state state-visib
ility
186:5 warning Explicitly mark visibility of state state-visib
ility
310:5 error Function has cyclomatic complexity 8 but allowed no more than 7 code-comple
xity

contracts/PrivateDistribution.sol
  3:1 error Compiler version 0.8.4 does not satisfy the ^0.7.0 semver requirement compiler-ve
rsion
 41:5 warning Explicitly mark visibility of state state-visib
ility
 72:5 warning Explicitly mark visibility of state state-visib
ility
102:5 warning Explicitly mark visibility of state state-visib
ility

✖ 66 problems (22 errors, 44 warnings)

error Command failed with exit code 1.
info Visit https://yarnpkg.com/en/docs/cli/run for documentation about this command.
```



## Disclaimer

The Audit does not give any warranties on the security of the code. One audit cannot be considered enough. We always recommend proceeding with several independent audits and a public bug bounty program to ensure the security of the code. Besides, a security audit, please don't consider this report as investment advice.



## Summary

The use of smart contracts is simple and the code is relatively small. Altogether the code is written and demonstrates effective use of abstraction, separation of concern, and modularity. But there are a few issues/vulnerabilities to be tackled at various security levels, it is recommended to fix them before deploying the contract on the main network. Given the subjective nature of some assessments, it will be up to the Polkarare team to decide whether any changes should be made.





**QuillAudits**



Canada, India, Singapore and United Kingdom



[audits.quillhash.com](https://audits.quillhash.com)



[hello@quillhash.com](mailto:hello@quillhash.com)