

BACKED SMART CONTRACT **AUDIT REPORT**



by QuillAudits, June 2021

Introduction :

This Audit Report highlights the overall security of the **Backed** Smart Contract. With this report, we have tried to ensure the reliability of their smart contract by a complete assessment of their system's architecture and the smart contract codebase.

Auditing Approach and Methodologies applied :

The QuillAudits team has performed thorough testing of the project, starting with analysing the code design patterns in which we reviewed the smart contract architecture to ensure it is structured and safe use of third-party smart contracts and libraries.

Our team then performed a formal line-by-line inspection of the Smart Contract to find any potential issues like race conditions, transaction-ordering dependence, timestamp dependence, and denial of service attacks.

In the Unit testing Phase, we coded/conducted Custom unit tests written for each function in the contract to verify that each function works as expected. In Automated Testing, We tested the Smart Contract with our in-house developed tools to identify vulnerabilities and security flaws.

The code was tested in collaboration with our multiple team members, and this included -

1. Testing the functionality of the Smart Contract to determine proper logic has been followed throughout the process.

BACKED Audit Report

2. Analysing the complexity of the code by thorough, manual review of the code, line-by-line.
3. Deploying the code on testnet using multiple clients to run live tests
4. Analysing failure preparations to check how the Smart Contract performs in case of bugs and vulnerabilities.
5. Checking whether all the libraries used in the code are on the latest version.
6. Analysing the security of the on-chain data.

BACKED Audit Report

Audit Details

- Project Name: Backed
- Github Commit: [de0d6e6e8484f6147322538d0ab2a77091beb7ba](#)
- Fixes Commit [5540aec25c3263465dee9bf6a929ed16084b0c37](#)
- Languages: Solidity (Smart contract), Javascript (Unit Testing)
- Platforms and Tools: Remix IDE, Truffle, Truffle Team, Ganache, Slither, Surya

Summary of Backed Smart Contract :

QuillAudits conducted a security audit of a smart contract of Backed. Backed contracts are used to token and crowdsale contracts.

- ☐ Token contract for backed
- ☐ Crowdsale

Audit Goals

The focus of the audit was to verify that the smart contract system is secure, resilient, and working according to its specifications. The audit activities can be grouped into the following three categories:

Security: Identifying security-related issues within each contract and the system of contracts.

Sound Architecture: Evaluation of the architecture of this system through the lens of established smart contract best practices and general software best practices.

Code Correctness and Quality: A full review of the contract source code. The primary areas of focus include:

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

Security Level references :

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

High severity issues will bring problems and should be fixed.

Medium severity issues could potentially bring problems and should eventually be fixed.

Low severity issues are minor details and warnings that can remain unfixed but would be better fixed at some point in the future.

Number of issues per severity

	Low	Medium	High
Open	0	0	0
Closed	2	0	0

Manual Audit:

High severity issues:-

No High Severity Issue.

Medium Severity Issues:-

No Medium Severity Issue.

Low Severity Issues:-

1 State variable shadowing in the token contract

- Check: shadowing-state
- Severity: Low
- Confidence: High

BackedToken._owner (BackedToken.sol#) shadows:

- Ownable._owner (BackedToken.sol#)

Remove the state variable shadowing, rename _owner of the token contract.

Status: Fixed by Developer

2 Function should be declared External

- Check: external-function
- Severity: Optimization
- Confidence: High

BackedToken.unlock (BackedToken.sol#) should be declared external
Crowdsale.wallet (BackedCrowdsale.sol#) should be declared external
Crowdsale.rate (BackedCrowdsale.sol#) should be declared external

Public functions that are never called by the contract should be declared external to save gas.

Use the external attribute for functions never called from the contract.

Status: Fixed by Developer

Functional test :

Function test has been done for multiple functions of three files. Results are below:

In BackToken.sol

- **transfer** transfer tokens to another address from your wallet -- > **PASS**
- **transferFrom** transfer tokens on behalf of another address --> **PASS**
- **approve** approve address to spend tokens --> **PASS**
- **IncreaseAllowance** increase allowance for address to spend tokens --> **PASS**
- **decreaseAllowance** decrease allowance for address to spend tokens --> **PASS**
- **unlock** unlock transfer of tokens by owner only --> **PASS**
- **transferOwnership** transfer contract ownership to another address --> **PASS**
- **renounceOwnership** leave account t --> **PASS**

In Crowdsale.sol

- **buyToken** send eth to contract to purchase tokens --> **PASS**
- **cap** display cap of sale --> **PASS**
- **maxAmount** max amount of wei of a single beneficiary --> **PASS**
- **minAmount** min amount of wei of a single beneficiary --> **PASS**

Automated Audit

Slither Tool Result :

```
INFO:Detectors:
ERC20Detailed.constructor.name (local variable @ BackedToken.sol#500) shadows:
- ERC20Detailed.name (function @ BackedToken.sol#509-511)
ERC20Detailed.constructor.symbol (local variable @ BackedToken.sol#500) shadows:
- ERC20Detailed.symbol (function @ BackedToken.sol#517-519)
ERC20Detailed.constructor.decimals (local variable @ BackedToken.sol#500) shadows:
- ERC20Detailed.decimals (function @ BackedToken.sol#533-535)
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#local-variable-shadowing
INFO:Detectors:
IERC20.totalSupply (BackedToken.sol#338) should be declared external
ERC20.totalSupply (BackedToken.sol#297-299) should be declared external
IERC20.balanceOf (BackedToken.sol#43) should be declared external
ERC20.balanceOf (BackedToken.sol#304-306) should be declared external
ERC20.transfer (BackedToken.sol#316-319) should be declared external
IERC20.transfer (BackedToken.sol#52) should be declared external
IERC20.allowance (BackedToken.sol#61) should be declared external
ERC20.allowance (BackedToken.sol#324-326) should be declared external
IERC20.approve (BackedToken.sol#77) should be declared external
ERC20.approve (BackedToken.sol#335-338) should be declared external
IERC20.transferFrom (BackedToken.sol#88) should be declared external
ERC20.transferFrom (BackedToken.sol#352-356) should be declared external
ERC20.increaseAllowance (BackedToken.sol#370-373) should be declared external
ERC20.decreaseAllowance (BackedToken.sol#389-392) should be declared external
ERC20Detailed.name (BackedToken.sol#509-511) should be declared external
ERC20Detailed.symbol (BackedToken.sol#517-519) should be declared external
ERC20Detailed.decimals (BackedToken.sol#533-535) should be declared external
Ownable.owner (BackedToken.sol#565-567) should be declared external
Ownable.renounceOwnership (BackedToken.sol#591-594) should be declared external
Ownable.transferOwnership (BackedToken.sol#600-602) should be declared external
BackedToken.unlock (BackedToken.sol#634-636) should be declared external
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#public-function-that-could-be-declared-as-external
INFO:Detectors:
Detected issues with version pragma in BackedToken.sol:
- pragma solidity0.5.16 (BackedToken.sol#2): it allows old versions
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#incorrect-version-of-solidity
INFO:Detectors:
Function 'Context._msgSender' (BackedToken.sol#20-22) is not in mixedCase
```

BACKED Audit Report

```
ERC20.totalSupply (BackedToken.sol#297-299) should be declared external
IERC20.balanceOf (BackedToken.sol#43) should be declared external
ERC20.balanceOf (BackedToken.sol#304-306) should be declared external
ERC20.transfer (BackedToken.sol#316-319) should be declared external
IERC20.transfer (BackedToken.sol#52) should be declared external
IERC20.allowance (BackedToken.sol#61) should be declared external
ERC20.allowance (BackedToken.sol#324-326) should be declared external
ERC20.approve (BackedToken.sol#77) should be declared external
ERC20.approve (BackedToken.sol#335-338) should be declared external
IERC20.transferFrom (BackedToken.sol#88) should be declared external
ERC20.transferFrom (BackedToken.sol#352-356) should be declared external
ERC20.increaseAllowance (BackedToken.sol#370-373) should be declared external
ERC20.decreaseAllowance (BackedToken.sol#389-392) should be declared external
ERC20Detailed.name (BackedToken.sol#509-511) should be declared external
ERC20Detailed.symbol (BackedToken.sol#517-519) should be declared external
ERC20Detailed.decimals (BackedToken.sol#533-535) should be declared external
Ownable.owner (BackedToken.sol#565-567) should be declared external
Ownable.renounceOwnership (BackedToken.sol#591-594) should be declared external
Ownable.transferOwnership (BackedToken.sol#600-602) should be declared external
BackedToken.unlock (BackedToken.sol#634-636) should be declared external
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#public-function-that-could-be-declared-as-external
INFO:Detectors:
Detected issues with version pragma in BackedToken.sol:
- pragma solidity0.5.16 (BackedToken.sol#2): it allows old versions
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#incorrect-version-of-solidity
INFO:Detectors:
Function 'Context._msgSender' (BackedToken.sol#20-22) is not in mixedCase
Function 'Context._msgData' (BackedToken.sol#24-27) is not in mixedCase
Function 'ERC20._transfer' (BackedToken.sol#408-415) is not in mixedCase
Function 'ERC20._mint' (BackedToken.sol#426-432) is not in mixedCase
Function 'ERC20._burn' (BackedToken.sol#445-451) is not in mixedCase
Function 'ERC20._approve' (BackedToken.sol#466-472) is not in mixedCase
Function 'ERC20._burnFrom' (BackedToken.sol#480-483) is not in mixedCase
Function 'Ownable._transferOwnership' (BackedToken.sol#607-611) is not in mixedCase
Function 'BackedToken._transfer' (BackedToken.sol#625-632) is not in mixedCase
Variable 'BackedToken._unlocked' (BackedToken.sol#617) is not in mixedCase
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#conformance-to-solidity-naming-conventions
abhi@crypticocean:~/Downloads/audit-backed-quill/backed-token-contracts-de0d6e6e8484f6147322538d0ab2a77091beb7ba/contracts$
```

```
Address.isContract uses assembly (BackedCrowdsale.sol#283-292)
- BackedCrowdsale.sol#290
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#assembly-usage
INFO:Detectors:
Crowdsale.wallet (BackedCrowdsale.sol#534-536) should be declared external
Crowdsale.rate (BackedCrowdsale.sol#541-543) should be declared external
AllowanceCrowdsale.tokenWallet (BackedCrowdsale.sol#700-702) should be declared external
AllowanceCrowdsale.remainingTokens (BackedCrowdsale.sol#708-710) should be declared external
CappedCrowdsale.cap (BackedCrowdsale.sol#744-746) should be declared external
CappedCrowdsale.capReached (BackedCrowdsale.sol#752-754) should be declared external
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#public-function-that-could-be-declared-as-external
INFO:Detectors:
Detected issues with version pragma in BackedCrowdsale.sol:
- pragma solidity0.5.0 (BackedCrowdsale.sol#2): it allows old versions
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#incorrect-version-of-solidity
INFO:Detectors:
Low level call in Address.sendValue (BackedCrowdsale.sol#322-328):
-(success) = recipient.call.value(amount)() BackedCrowdsale.sol#326
Low level call in SafeERC20.callOptionalReturn (BackedCrowdsale.sol#379-398):
-(success,returnData) = address(token).call(data) BackedCrowdsale.sol#391
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#low-level-calls
INFO:Detectors:
Function 'Context._msgSender' (BackedCrowdsale.sol#21-23) is not in mixedCase
Function 'Context._msgData' (BackedCrowdsale.sol#25-28) is not in mixedCase
Function 'Crowdsale._preValidatePurchase' (BackedCrowdsale.sol#586-590) is not in mixedCase
Function 'Crowdsale._postValidatePurchase' (BackedCrowdsale.sol#598-600) is not in mixedCase
Function 'Crowdsale._deliverTokens' (BackedCrowdsale.sol#608-610) is not in mixedCase
Function 'Crowdsale._processPurchase' (BackedCrowdsale.sol#618-620) is not in mixedCase
Function 'Crowdsale._updatePurchasingState' (BackedCrowdsale.sol#628-630) is not in mixedCase
Function 'Crowdsale._getTokenAmount' (BackedCrowdsale.sol#637-639) is not in mixedCase
Function 'Crowdsale._forwardFunds' (BackedCrowdsale.sol#644-646) is not in mixedCase
Function 'AllowanceCrowdsale._deliverTokens' (BackedCrowdsale.sol#717-719) is not in mixedCase
Function 'CappedCrowdsale._preValidatePurchase' (BackedCrowdsale.sol#761-764) is not in mixedCase
Function 'BackedCrowdsale._preValidatePurchase' (BackedCrowdsale.sol#799-814) is not in mixedCase
Function 'BackedCrowdsale._updatePurchasingState' (BackedCrowdsale.sol#822-830) is not in mixedCase
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#conformance-to-solidity-naming-conventions
INFO:Slither:BackedCrowdsale.sol analyzed (11 contracts), 33 result(s) found
abhi@crypticocean:~/Downloads/audit-backed-quill/backed-token-contracts-de0d6e6e8484f6147322538d0ab2a77091beb7ba/contracts$
```

BACKED Audit Report

```
BackedCrowdsale.sol
INFO:Detectors:
Address.isContract (BackedCrowdsale.sol#283-292) is declared view but contains assembly code
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#constant-functions-changing-the-state
INFO:Detectors:
Crowdsale.constructor.rate (local variable @ BackedCrowdsale.sol#504) shadows:
- Crowdsale.rate (function @ BackedCrowdsale.sol#541-543)
Crowdsale.constructor.wallet (local variable @ BackedCrowdsale.sol#504) shadows:
- Crowdsale.wallet (function @ BackedCrowdsale.sol#534-536)
Crowdsale.constructor.token (local variable @ BackedCrowdsale.sol#504) shadows:
- Crowdsale.token (function @ BackedCrowdsale.sol#527-529)
AllowanceCrowdsale.constructor.tokenWallet (local variable @ BackedCrowdsale.sol#692) shadows:
- AllowanceCrowdsale.tokenWallet (function @ BackedCrowdsale.sol#700-702)
CappedCrowdsale.constructor.cap (local variable @ BackedCrowdsale.sol#736) shadows:
- CappedCrowdsale.cap (function @ BackedCrowdsale.sol#744-746)
BackedCrowdsale.constructor.rate (local variable @ BackedCrowdsale.sol#778) shadows:
- Crowdsale.rate (function @ BackedCrowdsale.sol#541-543)
BackedCrowdsale.constructor.wallet (local variable @ BackedCrowdsale.sol#779) shadows:
- Crowdsale.wallet (function @ BackedCrowdsale.sol#534-536)
BackedCrowdsale.constructor.cap (local variable @ BackedCrowdsale.sol#780) shadows:
- CappedCrowdsale.cap (function @ BackedCrowdsale.sol#744-746)
BackedCrowdsale.constructor.token (local variable @ BackedCrowdsale.sol#781) shadows:
- Crowdsale.token (function @ BackedCrowdsale.sol#527-529)
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#local-variable-shadowing
INFO:Detectors:
Address.isContract uses assembly (BackedCrowdsale.sol#283-292)
- BackedCrowdsale.sol#290
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#assembly-usage
INFO:Detectors:
Crowdsale.wallet (BackedCrowdsale.sol#534-536) should be declared external
Crowdsale.rate (BackedCrowdsale.sol#541-543) should be declared external
AllowanceCrowdsale.tokenWallet (BackedCrowdsale.sol#700-702) should be declared external
AllowanceCrowdsale.remainingTokens (BackedCrowdsale.sol#708-710) should be declared external
CappedCrowdsale.cap (BackedCrowdsale.sol#744-746) should be declared external
CappedCrowdsale.capReached (BackedCrowdsale.sol#752-754) should be declared external
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#public-function-that-could-be-declared-as-external
INFO:Detectors:
Detected issues with version pragma in BackedCrowdsale.sol:
- pragma solidity^0.5.0 (BackedCrowdsale.sol#2): it allows old versions
```

Results

Some false positive errors have been reported by the tool; all other errors have been covered in issues explained above, under low-level severity issues.

Implementation Recommendations :

Function 'Context._msgSender' (BackedToken.sol#20-22) is not in mixedCase

Function 'Context._msgData' (BackedToken.sol#24-27) is not in mixedCase

Function 'ERC20._transfer' (BackedToken.sol#408-415) is not in mixedCase

Function 'ERC20._mint' (BackedToken.sol#426-432) is not in mixedCase

Function 'ERC20._burn' (BackedToken.sol#445-451) is not in mixedCase

Function 'ERC20._approve' (BackedToken.sol#466-472) is not in mixedCase

Function 'ERC20._burnFrom' (BackedToken.sol#480-483) is not in mixedCase

Function 'Ownable._transferOwnership' (BackedToken.sol#607-611) is not in mixedCase

Function 'BackedToken._transfer' (BackedToken.sol#625-632) is not in mixedCase

Variable 'BackedToken._unlocked' (BackedToken.sol#617) is not in mixedCase

Function 'Context._msgSender' (BackedCrowdsale.sol#21-23) is not in mixedCase

Function 'Context._msgData' (BackedCrowdsale.sol#25-28) is not in mixedCase

Function 'Crowdsale._preValidatePurchase' (BackedCrowdsale.sol#586-590) is not in mixedCase

Function 'Crowdsale._postValidatePurchase' (BackedCrowdsale.sol#598-600) is not in mixedCase

Function 'Crowdsale._deliverTokens' (BackedCrowdsale.sol#608-610) is not in mixedCase

Function 'Crowdsale._processPurchase' (BackedCrowdsale.sol#618-620) is not in mixedCase

Function 'Crowdsale._updatePurchasingState' (BackedCrowdsale.sol#628-630) is not in mixedCase

Function 'Crowdsale._getTokenAmount'
(BackedCrowdsale.sol#637-639) is not in mixedCase

Function 'Crowdsale._forwardFunds' (BackedCrowdsale.sol#644-646)
is not in mixedCase

Function 'AllowanceCrowdsale._deliverTokens'
(BackedCrowdsale.sol#717-719) is not in mixedCase

Function 'CappedCrowdsale._preValidatePurchase'
(BackedCrowdsale.sol#761-764) is not in mixedCase

Function 'BackedCrowdsale._preValidatePurchase'
(BackedCrowdsale.sol#799-814) is not in mixedCase

Function 'BackedCrowdsale._updatePurchasingState'
(BackedCrowdsale.sol#822-830) is not in mixedCase

Disclaimer

Quillhash audit is not a security warranty, investment advice, or an endorsement of the Backed contract. Securing smart contracts is a multistep process; therefore, running a bug bounty program as a complement to this audit is strongly recommended.

Closing Summary:

The use case of the smart contract is very well designed and Implemented. Overall, the code is well written and demonstrates effective use of abstraction, separation of concerns, and modularity. The **Backed** development team demonstrated high technical capabilities, both in the design of the architecture and in the implementation.

Some low-severity issues have been reported which, are now fixed and tested.