Smart Contract
Security Audit
V1

TokenBot Smart Contract

30/6/2022

https://saferico.com/

business@saferico.com
https://t.me/SFI_ANN

Table of Contents

Background

The purpose of the audit was to achieve the following:

- Ensure that the smart contract functions as intended.
- Identify potential security issues with the smart contract.

The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Project Information

- **Name**: TokenBot
- **Ticker**: TKB
- **Max Supply**: 1,000,000,000
- **Platform**: Ethereum Network
- **Contract Address**: 0x102a96cab42c5214bb8b1b38b995969bfbfe7850
- **Code:**

https://github.com/Saferico/Smart-Contracts-for-Projects/blob/main/TokenBot.sol

Contracts address deployed to test net (ETH)

TokenBot (TKB) Token contract on ETH test net to test every function by the auditor.

https://rinkeby.etherscan.io/address/0x102a96cab42c5214bb8b1b38b995969bfbfe7850

Executive Summary

According to our assessment, the customer`s solidity smart contract is **Well Secured**.

Well Secured	✓
Secured	
Poor Secured	
Insecure	

Automated checks are with remix IDE. All issues were performed by the team, which included the analysis of code functionality, manual audit found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the audit overview section. The general overview is presented in the Project Information section and all issues found are located in the audit overview section.

Team found 0 critical, 0 high, 0 medium, 3 low, 0 very low-level issues and 2 notes in all solidity files of the contract

The files:

TokenBot.sol

File and Function Level Report

File in Scope:

Contract Name	SHA 256 hash	Contract Address
TokenBot.sol	**7ffb1c61f6045881d4526775f011d9140dc381a30c06085be0d78a0e24df2d63**	0x102a96cab42c5214bb8b1b38b995969bfbfe7850

- Contract: TokenBot
- Inherit: ERC20, ERC20Burnable, Ownable, ERC20Permit, ERC20Votes
- Observation: All passed including security check
- Test Report: passed
- Score: passed
- Conclusion: passed

Function	Test Result	Type / Return Type	Score
name	✓	Read / public	**Passed**
symbol	✓	Read / public	**Passed**
allowance	✓	Read / public	**Passed**
decimals	✓	Read / public	**Passed**
nonces	✓	Read / public	**Passed**
balanceOf	✓	Read / public	**Passed**
Owner	✓	Read / public	**Passed**
totalSuppy	✓	Read / public	**Passed**
DOMAIN_SEPARATOR	✓	Read / public	**Passed**
checkpoints	✓	Read / public	**Passed**
numCheckpoints	✓	Read / public	**Passed**

MAX_SUPPLY	✓	Read / public	Passed
getVotes	✓	Read / public	Passed
delegates	✓	Read / public	Passed
getPastTotalSupply	✓	Read / public	Passed
getPastVotes	✓	Read / public	Passed
decreaseAllowance	✓	Write / public	Passed
increaseAllowance	✓	Write / public	Passed
mint	✓	Write / public	Passed
burn	✓	Write / public	Passed
burnFrom	✓	Write / public	Passed
approve	✓	Write / public	Passed
transfer	✓	Write / public	Passed
transferFrom	✓	Write / public	Passed
delegate	✓	Write / public	Passed
transferOwnership	✓	Write / public	Passed
permit	✓	Write / public	Passed
renounceOwnership	✓	Write / public	Passed
delegateBySig	✓	Write / public	Passed

Issues Checking Status

No.	Issue Description	Checking Status
1	Compiler warnings.	**Passed**
2	Race conditions and Reentrancy. Cross-function race conditions.	**Passed**
3	Possible delays in data delivery.	**Passed**
4	Oracle calls.	**Passed**
5	Design Logic.	**Passed**
6	Timestamp dependence.	**Passed with Notes**
7	Integer Overflow and Underflow.	**Passed**
8	DoS with Revert.	**Passed**
9	DoS with block gas limit.	**Passed with Notes**
10	Methods execution permissions.	**Passed**
11	Economy model. If application logic is based on an incorrect economic model, the application would not function correctly and participants would incur financial losses. This type of issue is most often found in bonus rewards systems, Staking and Farming contracts, Vault and Vesting contracts, etc.	**Passed**
12	The impact of the exchange rate on the logic.	**Passed**
13	Private user data leaks.	**Passed**
14	Malicious Event log.	**Passed**
15	Scoping and Declarations.	**Passed**
16	Uninitialized storage pointers.	**Passed**
17	Arithmetic accuracy.	**Passed**

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to tokens loss etc.
High	High-level vulnerabilities are difficult to exploit; however, they also have significant impact on smart contract execution, e.g. public access to crucial functions
Medium	Medium-level vulnerabilities are important to fix; however, they can’t lead to tokens lose
Low	Low-level vulnerabilities are mostly related to outdated, unused etc. code snippets, that can’t have significant impact on execution
Note	Lowest-level vulnerabilities, code style violations and info statements can’t affect smart contract execution and can be ignored.

Audit Findings

Critical:

No Critical severity vulnerabilities were found

High:

No High severity vulnerabilities were found

Medium:

No Medium severity vulnerabilities were found

Low:

#Missing zero address validation

Description

When the Owner wants to mint tokens, he has to check for the zero address to make it, he didn't add the burn address. Otherwise, the mint function will act like a burn function.

```
function mint(
        address to,
        uint256 amount
    ) public onlyOwner {
        require(
            totalSupply() + amount <= MAX_SUPPLY,
            "TokenBot::mint: mint amount exceeds MAX_SUPPLY"
        );
        _mint(to, amount);
    }
```

Remediation

Use the require statement to check for zero addresses.

Status: Closed. Fixed in version 2.

#Unnecessary import some libraries

Description

The developer import ERC20, and draft ERC20 permit libraries in the main contract and no need for that because it already imported in ERC20 Burnable and ERC20 Votes contract so its useless import just costing more ETH gas.

```
import "@openzeppelin/contracts@4.6.0/token/ERC20/ERC20.sol";
import "@openzeppelin/contracts@4.6.0/token/ERC20/extensions/draft-
ERC20Permit.sol";
```

Remediation
Remove Strings Library to save ETH gas fees.

Status: Closed. Fixed in version2.

#Use of block.timestamp for comparisons

Description
The value of block.timestamp can be manipulated by the miner. And conditions with strict equality is difficult to achieve -block.timestamp

Remediation
Avoid use of block.timestamp

Status: Acknowledged

Very Low:

No Very Low severity vulnerabilities were found.

Notes:

#Compiler version is old
Description
The compiler being used was released 9 months ago. It's recommended to use a more recent compiler version, there can be benefits like reduction in bytecode size etc.

Status: Closed. Fixed in version 2.

# Constant calculations in the contract

Description
recalculated initialization will save 2847 units of gas in deployment

```
uint256 public immutable MAX_SUPPLY = 1000000000 * 10 ** decimals();
```

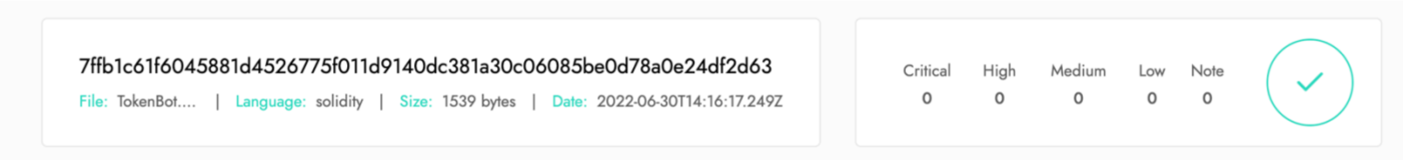
Recommendation
Replace the initialization as

```
uint256 public immutable MAX_SUPPLY = 1000000000000000000000000000;
```

Status Closed. Fixed in version 2.

Automatic Testing

1- Check for security

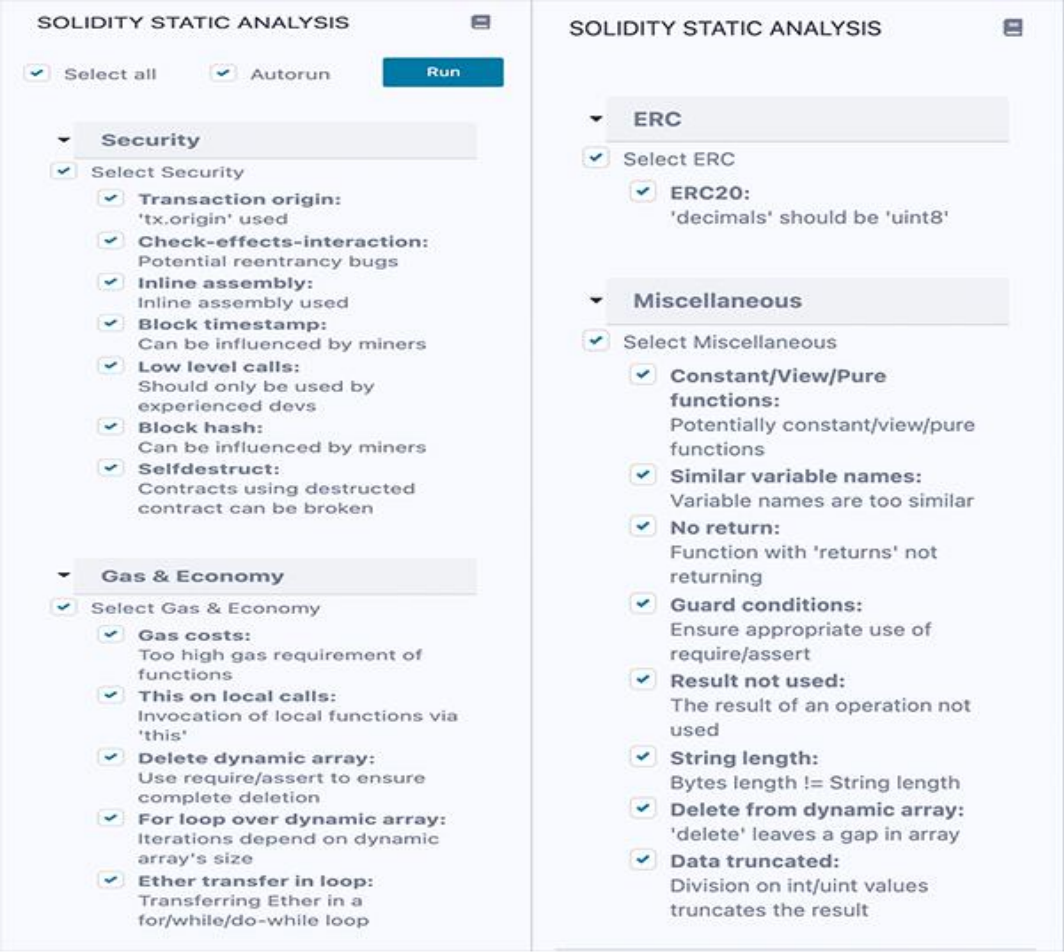


7ffb1c61f6045881d4526775f011d9140dc381a30c06085be0d78a0e24df2d63

File: TokenBot.... | Language: solidity | Size: 1539 bytes | Date: 2022-06-30T14:16:17.249Z

Critical	High	Medium	Low	Note
0	0	0	0	0

2- SOLIDITY STATIC ANALYSIS



SOLIDITY STATIC ANALYSIS

☑ Select all ☑ Autorun Run

Security

☑ Select Security

- ☑ **Transaction origin:** 'tx.origin' used
- ☑ **Check-effects-interaction:** Potential reentrancy bugs
- ☑ **Inline assembly:** Inline assembly used
- ☑ **Block timestamp:** Can be influenced by miners
- ☑ **Low level calls:** Should only be used by experienced devs
- ☑ **Block hash:** Can be influenced by miners
- ☑ **Selfdestruct:** Contracts using destructed contract can be broken

Gas & Economy

☑ Select Gas & Economy

- ☑ **Gas costs:** Too high gas requirement of functions
- ☑ **This on local calls:** Invocation of local functions via 'this'
- ☑ **Delete dynamic array:** Use require/assert to ensure complete deletion
- ☑ **For loop over dynamic array:** Iterations depend on dynamic array's size
- ☑ **Ether transfer in loop:** Transferring Ether in a for/while/do-while loop

SOLIDITY STATIC ANALYSIS

ERC

☑ Select ERC

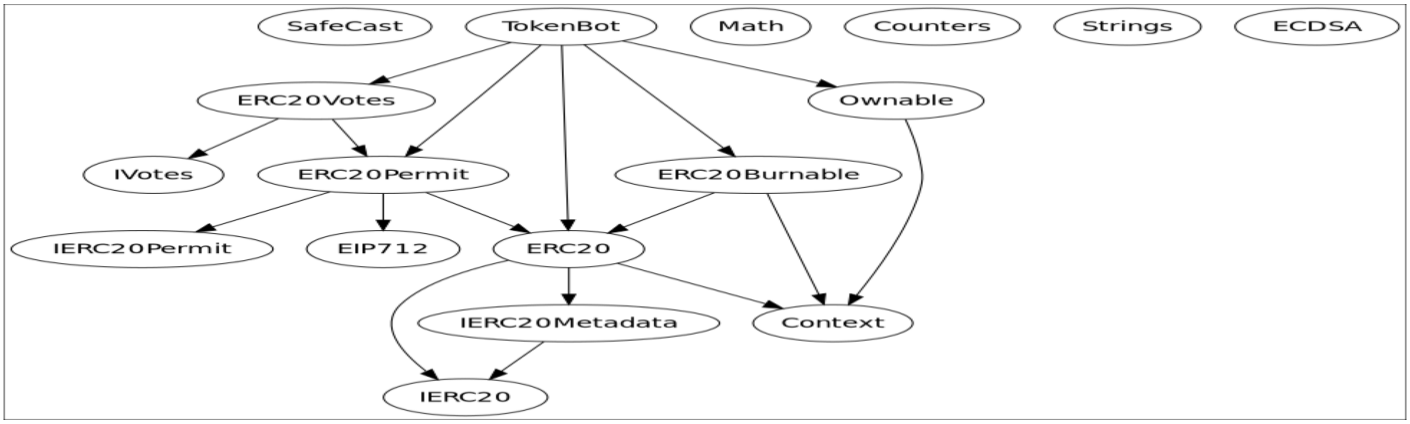
- ☑ **ERC20:** 'decimals' should be 'uint8'

Miscellaneous

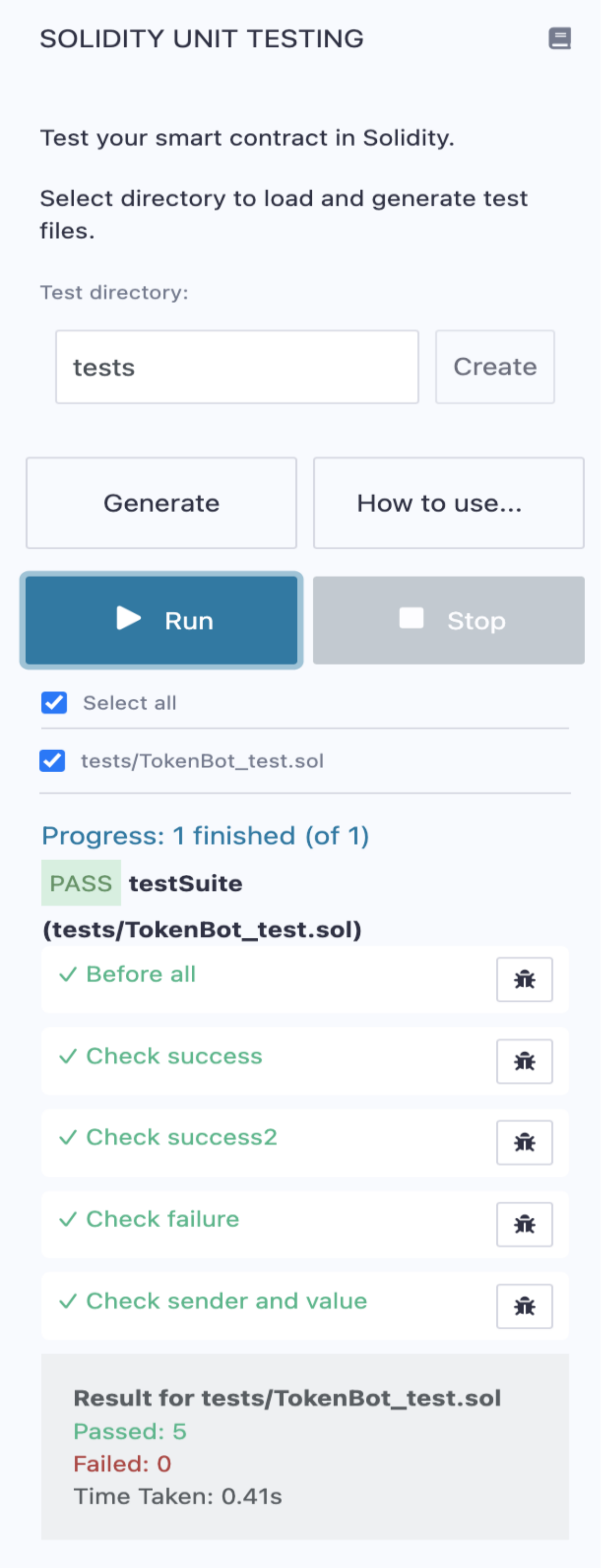
☑ Select Miscellaneous

- ☑ **Constant/View/Pure functions:** Potentially constant/view/pure functions
- ☑ **Similar variable names:** Variable names are too similar
- ☑ **No return:** Function with 'returns' not returning
- ☑ **Guard conditions:** Ensure appropriate use of require/assert
- ☑ **Result not used:** The result of an operation not used
- ☑ **String length:** Bytes length != String length
- ☑ **Delete from dynamic array:** 'delete' leaves a gap in array
- ☑ **Data truncated:** Division on int/uint values truncates the result

3- Inheritance graph



4- SOLIDITY UNIT TESTING



SOLIDITY UNIT TESTING

Test your smart contract in Solidity.

Select directory to load and generate test files.

Test directory: tests

☑ Select all

☑ tests/TokenBot_test.sol

Progress: 1 finished (of 1)

PASS **testSuite (tests/TokenBot_test.sol)**

✓ Before all

✓ Check success

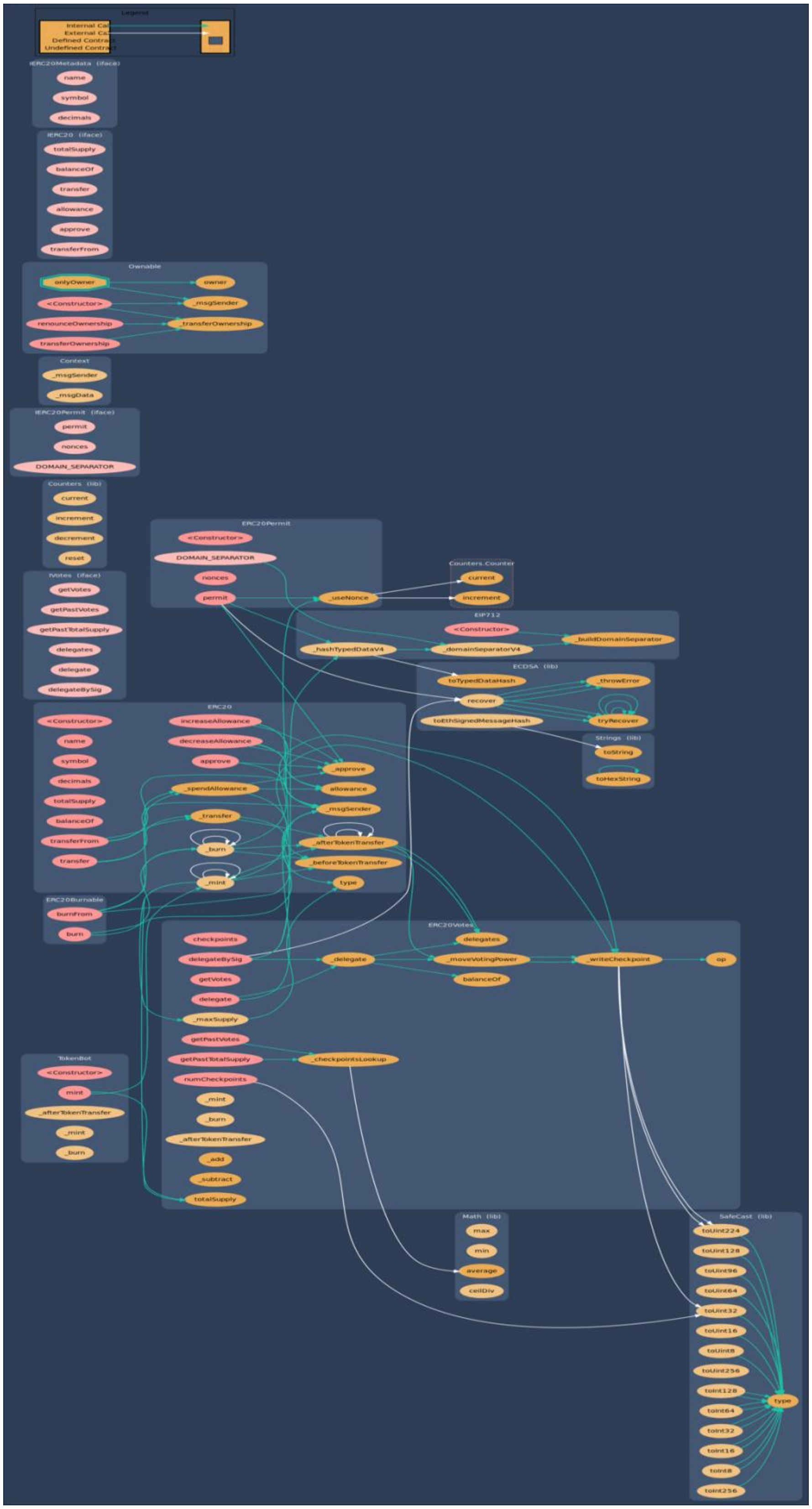
✓ Check success2

✓ Check failure

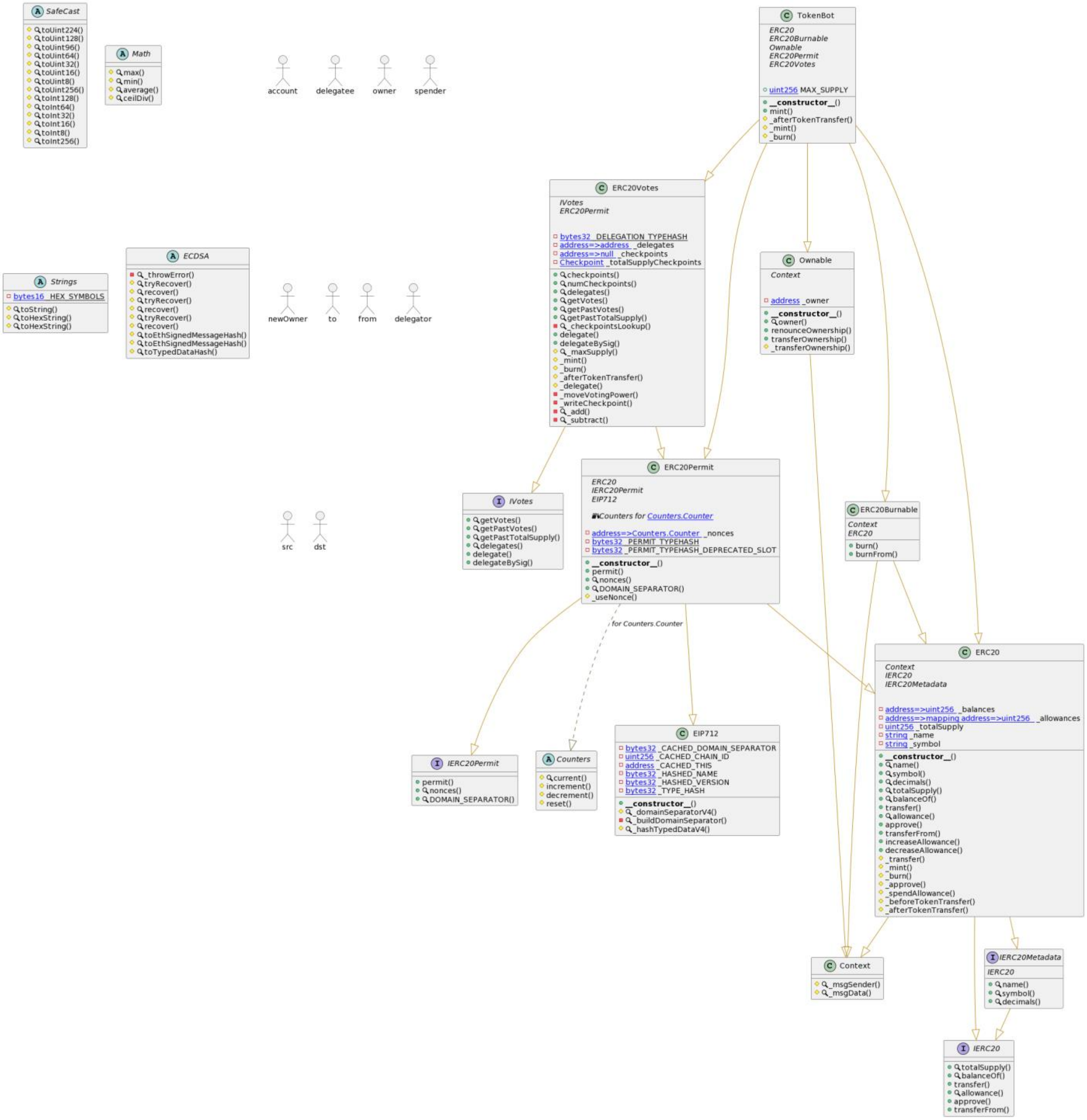
✓ Check sender and value

Result for tests/TokenBot_test.sol
Passed: 5
Failed: 0
Time Taken: 0.41s

5- Call graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
IERC20Metadata (iface)
name
symbol
decimals
IERC20 (iface)
totalSupply
balanceOf
transfer
allowance
approve
transferFrom
Ownable
onlyOwner
owner
<Constructor>
_msgSender
renounceOwnership
_transferOwnership
transferOwnership
Context
_msgSender
_msgData
IERC20Permit (iface)
permit
nonces
DOMAIN_SEPARATOR
Counters (lib)
current
increment
decrement
reset
IVotes (iface)
getVotes
getPastVotes
getPastTotalSupply
delegates
delegate
delegateBySig
ERC20Permit
<Constructor>
DOMAIN_SEPARATOR
nonces
permit
_useNonce
Counters.Counter
current
increment
EIP712
<Constructor>
_buildDomainSeparator
_hashTypedDataV4
_domainSeparatorV4
ECDSA (lib)
toTypedDataHash
_throwError
recover
toEthSignedMessageHash
tryRecover
Strings (lib)
toString
toHexString
ERC20
<Constructor>
increaseAllowance
name
decreaseAllowance
symbol
approve
_approve
decimals
_spendAllowance
allowance
totalSupply
_msgSender
balanceOf
_transfer
transferFrom
_afterTokenTransfer
_burn
transfer
_beforeTokenTransfer
_mint
type
ERC20Burnable
burnFrom
burn
ERC20Votes
checkpoints
delegates
delegateBySig
_delegate
_moveVotingPower
_writeCheckpoint
op
getVotes
balanceOf
delegate
_maxSupply
getPastVotes
getPastTotalSupply
_checkpointsLookup
numCheckpoints
_mint
_burn
_afterTokenTransfer
_add
_subtract
totalSupply
TokenBot
<Constructor>
mint
_afterTokenTransfer
_mint
_burn
Math (lib)
max
min
average
ceilDiv
SafeCast (lib)
toUint224
toUint128
toUint96
toUint64
toUint32
toUint16
toUint8
toUint256
toInt128
type
toInt64
toInt32
toInt16
toInt8
toInt256

Unified Modeling Language (UML)



Functions signature

```
Sighash   |   Function Signature
========================
39509351  =>  increaseAllowance(address,uint256)
5bb79860  =>  toUint224(uint256)
809fdd33  =>  toUint128(uint256)
1cf887fc  =>  toUint96(uint256)
2665fad0  =>  toUint64(uint256)
c8193255  =>  toUint32(uint256)
9374068f  =>  toUint16(uint256)
0cc4681e  =>  toUint8(uint256)
fdcf791b  =>  toUint256(int256)
dd2a0316  =>  toInt128(int256)
d6bd32aa  =>  toInt64(int256)
9c6f59be  =>  toInt32(int256)
cf65b4d3  =>  toInt16(int256)
f136dc02  =>  toInt8(int256)
dfbe873b  =>  toInt256(uint256)
9ab24eb0  =>  getVotes(address)
3a46b1a8  =>  getPastVotes(address,uint256)
8e539e8c  =>  getPastTotalSupply(uint256)
587cde1e  =>  delegates(address)
5c19a95c  =>  delegate(address)
c3cda520  =>  delegateBySig(address,uint256,uint256,uint8,bytes32,bytes32)
6d5433e6  =>  max(uint256,uint256)
7ae2b5c7  =>  min(uint256,uint256)
2b7423ab  =>  average(uint256,uint256)
9cb35327  =>  ceilDiv(uint256,uint256)
ad04a8d1  =>  current(Counter)
e2bee435  =>  increment(Counter)
854ec98e  =>  decrement(Counter)
440d212a  =>  reset(Counter)
6900a3ae  =>  toString(uint256)
8fba8d5c  =>  toHexString(uint256)
63e1cbea  =>  toHexString(uint256,uint256)
5e2ffa14  =>  _throwError(RecoverError)
c6edd8a7  =>  tryRecover(bytes32,bytes)
19045a25  =>  recover(bytes32,bytes)
628f98cc  =>  tryRecover(bytes32,bytes32,bytes32)
bf2fe7fd  =>  recover(bytes32,bytes32,bytes32)
4d78da76  =>  tryRecover(bytes32,uint8,bytes32,bytes32)
c2bf17b0  =>  recover(bytes32,uint8,bytes32,bytes32)
918a15cf  =>  toEthSignedMessageHash(bytes32)
92bd87b5  =>  toEthSignedMessageHash(bytes)
7df7a71c  =>  toTypedDataHash(bytes32,bytes32)
7b134b4c  =>  _domainSeparatorV4()
112794f2  =>  _buildDomainSeparator(bytes32,bytes32,bytes32)
c8f1ecd8  =>  _hashTypedDataV4(bytes32)
d505accf  =>  permit(address,address,uint256,uint256,uint8,bytes32,bytes32)
7ecebe00  =>  nonces(address)
3644e515  =>  DOMAIN_SEPARATOR()
119df25f  =>  _msgSender()
8b49d47e  =>  _msgData()
8da5cb5b  =>  owner()
```

```
715018a6  =>  renounceOwnership()
f2fde38b  =>  transferOwnership(address)
d29d44ee  =>  _transferOwnership(address)
18160ddd  =>  totalSupply()
70a08231  =>  balanceOf(address)
a9059cbb  =>  transfer(address,uint256)
dd62ed3e  =>  allowance(address,address)
095ea7b3  =>  approve(address,uint256)
23b872dd  =>  transferFrom(address,address,uint256)
06fdde03  =>  name()
95d89b41  =>  symbol()
313ce567  =>  decimals()
a457c2d7  =>  decreaseAllowance(address,uint256)
30e0789e  =>  _transfer(address,address,uint256)
4e6ec247  =>  _mint(address,uint256)
6161eb18  =>  _burn(address,uint256)
104e81ff  =>  _approve(address,address,uint256)
1532335e  =>  _spendAllowance(address,address,uint256)
cad3be83  =>  _beforeTokenTransfer(address,address,uint256)
8f811a1c  =>  _afterTokenTransfer(address,address,uint256)
35d11de3  =>  _useNonce(address)
f1127ed8  =>  checkpoints(address,uint32)
6fcfff45  =>  numCheckpoints(address)
db263f39  =>  _checkpointsLookup(Checkpoint256[],uint256)
22f4596f  =>  _maxSupply()
a28a42b3  =>  _delegate(address,address)
82851b84  =>  _moveVotingPower(address,address,uint256)
5c3188b4  =>  _writeCheckpoint(Checkpoint256[],function(uint256,uint256)
3d0316c3  =>  _add(uint256,uint256)
880bf496  =>  _subtract(uint256,uint256)
42966c68  =>  burn(uint256)
79cc6790  =>  burnFrom(address,uint256)
40c10f19  =>  mint(address,uint256)
```

Automatic general report

```
Files Description Table

|  File Name  |  SHA-1 Hash  |
|-------------|--------------|
| /Users/macbook/Desktop/smart contracts/TokenBot.sol |
08ce9891cf56a713d33e464df69d631b3bf9da44 |

 Contracts Description Table

|  Contract  |         Type        |       Bases      |                  |
|
|:----------:|:-------------------:|:----------------:|:----------------:|:--------
-------:|
|     └      |  **Function Name**  |  **Visibility**  |  **Mutability**  |
**Modifiers**  |
||||||
| **SafeCast** | Library |  |||
| └ | toUint224 | Internal 🔒 |   | |
| └ | toUint128 | Internal 🔒 |   | |
| └ | toUint96 | Internal 🔒 |   | |
| └ | toUint64 | Internal 🔒 |   | |
| └ | toUint32 | Internal 🔒 |   | |
| └ | toUint16 | Internal 🔒 |   | |
| └ | toUint8 | Internal 🔒 |   | |
| └ | toUint256 | Internal 🔒 |   | |
| └ | toInt128 | Internal 🔒 |   | |
| └ | toInt64 | Internal 🔒 |   | |
| └ | toInt32 | Internal 🔒 |   | |
| └ | toInt16 | Internal 🔒 |   | |
| └ | toInt8 | Internal 🔒 |   | |
| └ | toInt256 | Internal 🔒 |   | |
||||||
| **IVotes** | Interface |  |||
| └ | getVotes | External ❗ |   |NO❗ |
| └ | getPastVotes | External ❗ |   |NO❗ |
| └ | getPastTotalSupply | External ❗ |   |NO❗ |
| └ | delegates | External ❗ |   |NO❗ |
| └ | delegate | External ❗ | 🛑  |NO❗ |
| └ | delegateBySig | External ❗ | 🛑  |NO❗ |
||||||
| **Math** | Library |  |||
| └ | max | Internal 🔒 |   | |
| └ | min | Internal 🔒 |   | |
| └ | average | Internal 🔒 |   | |
| └ | ceilDiv | Internal 🔒 |   | |
||||||
| **Counters** | Library |  |||
| └ | current | Internal 🔒 |   | |
| └ | increment | Internal 🔒 | 🛑  | |
| └ | decrement | Internal 🔒 | 🛑  | |
| └ | reset | Internal 🔒 | 🛑  | |
||||||
| **Strings** | Library |  |||
| └ | toString | Internal 🔒 |   | |
| └ | toHexString | Internal 🔒 |   | |
| └ | toHexString | Internal 🔒 |   | |
||||||
| **ECDSA** | Library |  |||
| └ | _throwError | Private 🔐 |   | |
| └ | tryRecover | Internal 🔒 |   | |
| └ | recover | Internal 🔒 |   | |
| └ | tryRecover | Internal 🔒 |   | |
| └ | recover | Internal 🔒 |   | |
| └ | tryRecover | Internal 🔒 |   | |
| └ | recover | Internal 🔒 |   | |
| └ | toEthSignedMessageHash | Internal 🔒 |   | |
| └ | toEthSignedMessageHash | Internal 🔒 |   | |
| └ | toTypedDataHash | Internal 🔒 |   | |
||||||
| **EIP712** | Implementation |  |||
| └ | <Constructor> | Public ❗️ | 🛑  |NO❗️ |
| └ | _domainSeparatorV4 | Internal 🔒 |   | |
| └ | _buildDomainSeparator | Private 🔐 |   | |
| └ | _hashTypedDataV4 | Internal 🔒 |   | |
||||||
| **IERC20Permit** | Interface |  |||
| └ | permit | External ❗️ | 🛑  |NO❗️ |
| └ | nonces | External ❗️ |   |NO❗️ |
| └ | DOMAIN_SEPARATOR | External ❗️ |   |NO❗️ |
||||||
| **Context** | Implementation |  |||
| └ | _msgSender | Internal 🔒 |   | |
| └ | _msgData | Internal 🔒 |   | |
||||||
| **Ownable** | Implementation | Context |||
| └ | <Constructor> | Public ❗️ | 🛑  |NO❗️ |
| └ | owner | Public ❗️ |   |NO❗️ |
| └ | renounceOwnership | Public ❗️ | 🛑  | onlyOwner |
| └ | transferOwnership | Public ❗️ | 🛑  | onlyOwner |
| └ | _transferOwnership | Internal 🔒 | 🛑  | |
||||||
| **IERC20** | Interface |  |||
| └ | totalSupply | External ❗️ |   |NO❗️ |
| └ | balanceOf | External ❗️ |   |NO❗️ |
| └ | transfer | External ❗️ | 🛑  |NO❗️ |
| └ | allowance | External ❗️ |   |NO❗️ |
| └ | approve | External ❗️ | 🛑  |NO❗️ |
| └ | transferFrom | External ❗️ | 🛑  |NO❗️ |
||||||
| **IERC20Metadata** | Interface | IERC20 |||
| └ | name | External ❗️ |   |NO❗️ |
| └ | symbol | External ❗️ |   |NO❗️ |
| └ | decimals | External ❗️ |   |NO❗️ |
||||||
| **ERC20** | Implementation | Context, IERC20, IERC20Metadata |||
| └ | <Constructor> | Public ❗️ | 🛑  |NO❗️ |
| └ | name | Public ❗️ |   |NO❗️ |
| └ | symbol | Public ❗️ |   |NO❗️ |
| └ | decimals | Public ❗️ |   |NO❗️ |
| └ | totalSupply | Public ❗️ |   |NO❗️ |
| └ | balanceOf | Public ❗️ |   |NO❗️ |
| └ | transfer | Public ❗️ | 🛑  |NO❗️ |
| └ | allowance | Public ❗️ |   |NO❗️ |
| └ | approve | Public ❗️ | 🛑  |NO❗️ |
| └ | transferFrom | Public ❗️ | 🛑  |NO❗️ |
| └ | increaseAllowance | Public ❗️ | 🛑  |NO❗️ |
| └ | decreaseAllowance | Public ❗️ | 🛑  |NO❗️ |
| └ | _transfer | Internal 🔒 | 🛑  | |
| └ | _mint | Internal 🔒 | 🛑  | |
| └ | _burn | Internal 🔒 | 🛑  | |
| └ | _approve | Internal 🔒 | 🛑  | |
| └ | _spendAllowance | Internal 🔒 | 🛑  | |
| └ | _beforeTokenTransfer | Internal 🔒 | 🛑  | |
| └ | _afterTokenTransfer | Internal 🔒 | 🛑  | |
||||||
| **ERC20Permit** | Implementation | ERC20, IERC20Permit, EIP712 |||
| └ | <Constructor> | Public ❗️ | 🛑  | EIP712 |
| └ | permit | Public ❗️ | 🛑  |NO❗️ |
| └ | nonces | Public ❗️ |   |NO❗️ |
| └ | DOMAIN_SEPARATOR | External ❗️ |   |NO❗️ |
| └ | _useNonce | Internal 🔒 | 🛑  | |
||||||
| **ERC20Votes** | Implementation | IVotes, ERC20Permit |||
| └ | checkpoints | Public ❗️ |   |NO❗️ |
| └ | numCheckpoints | Public ❗️ |   |NO❗️ |
| └ | delegates | Public ❗️ |   |NO❗️ |
| └ | getVotes | Public ❗️ |   |NO❗️ |
| └ | getPastVotes | Public ❗️ |   |NO❗️ |
| └ | getPastTotalSupply | Public ❗️ |   |NO❗️ |
| └ | _checkpointsLookup | Private 🔐 |   | |
| └ | delegate | Public ❗️ | 🛑  |NO❗️ |
| └ | delegateBySig | Public ❗️ | 🛑  |NO❗️ |
| └ | _maxSupply | Internal 🔒 |   | |
| └ | _mint | Internal 🔒 | 🛑  | |
| └ | _burn | Internal 🔒 | 🛑  | |
| └ | _afterTokenTransfer | Internal 🔒 | 🛑  | |
| └ | _delegate | Internal 🔒 | 🛑  | |
| └ | _moveVotingPower | Private 🔐 | 🛑  | |
| └ | _writeCheckpoint | Private 🔐 | 🛑  | |
| └ | _add | Private 🔐 |   | |
| └ | _subtract | Private 🔐 |   | |
||||||
| **ERC20Burnable** | Implementation | Context, ERC20 |||
| └ | burn | Public ❗️ | 🛑  |NO❗️ |
| └ | burnFrom | Public ❗️ | 🛑  |NO❗️ |
||||||
| **TokenBot** | Implementation | ERC20, ERC20Burnable, Ownable, ERC20Permit,
ERC20Votes |||
| └ | <Constructor> | Public ❗️ | 🛑  | ERC20 ERC20Permit |
| └ | mint | Public ❗️ | 🛑  | onlyOwner |
| └ | _afterTokenTransfer | Internal 🔒 | 🛑  | |
| └ | _mint | Internal 🔒 | 🛑  | |
| └ | _burn | Internal 🔒 | 🛑  | |

 Legend

|  Symbol  |  Meaning  |
|:--------:|-----------|
|    🛑    | Function can modify state |
|    💵    | Function is payable |
```

Conclusion

The contracts are written systematically. Team found no critical issues. So it is good to go for production.

Since possible test cases can be unlimited and developer level documentation (code flow diagram with function level description) not provided, for such an extensive smart contract protocol, we provide no such guarantee of future outcomes. We have used all the latest static tools and manual observations to cover maximum possible test cases to scan Everything.

Security state of the reviewed contract is " Well Secured".

✓No volatile code.
✓ No many high severity issues were found.

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as of the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against the team on the basis of what it says or doesn’t say, or how team produced it, and it is important for you to conduct your own independent investigations before making any decisions. team go into more detail on this in the below disclaimer below – please make sure to read it in full.

By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and Saferico and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (Saferico s) owe no duty of care towards you or any other person, nor does Saferico make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and Saferico hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, Saferico hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against Saferico, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report. The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.