Jesbus Technology
Software development and integration

Audit of the Stokx token smart contract

2019 - 04 - 17

Auditor

Name: Jesse Busman
Email: jesse@jesbus.com
Country: The Netherlands
Company: Jesbus Technology

Audited smart contracts

- `SafeMath`
- `Owned`
- `Core`

Source:
https://github.com/stokxio/stokx-token-contract/blob/6df5d8e07d71f4499a7354d910c2db7fc772ae9e/contract.sol

ERC20-compliance

Function	Requirements	Compliance
`totalSupply`	• The return value of `totalSupply` must always be equal to the sum of all address balances.	• **Yes** The `totalSupply` function returns the value of `_totalSupply`. On contract construction 100000000 tokens are assigned to `_totalSupply` and to the balance of the contract deployer. The `transfer` and `transferFrom` have no effect on the total supply: they subtract and add a net total of 0 tokens. The `mint`, `mintToAddress`, `burn` and `burnFromAddress` functions modify a single user balance by the same amount as the `_totalSupply` variable.
`transfer`	• `transfer` must revert if the sender does not have enough tokens to perform the transfer they are requesting. • `transfer` must return `true` if the transfer succeeded. • `transfer` must emit the `Transfer` event if the transfer succeeded. • `transfer` must treat transfers of 0 tokens as normal transfers.	• **Yes** `safeSub` is used to prevent sending more than your balance. • **Yes** • **Yes** • **Yes**
`transferFrom`	• `transferFrom` must revert if the sender does not have enough tokens to perform the transfer they are requesting. • `transferFrom` must revert if the sender does not have approval to transfer the tokens • `transferFrom` must return true if the transfer succeeded. • `transferFrom` must emit the `Transfer` event if the transfer succeeded. • `transferFrom` must treat transfers with 0 value as normal transfers.	• **Yes** `safeSub` is used to prevent sending more than your balance. • **Yes** • **Yes** • **Yes** • **Yes**

(continues on next page)

Function	Compliance
`balanceOf`	● **Yes**
`approve`	● **Yes**
`allowance`	● **Yes**

The Stokx token smart contract is **fully compliant** with the ERC20 token standard.

Testing the Stokx token smart contract

Compiler version used: 0.5.7
Deployed on network: Ropsten testnet
Deployed by address: 0x2fe7b7Afaf9301cBC9F7A146C70BA9FAaE1Ad9Be
Deployed at address: 0x02727ca09d5a2a36b7837ff04c8ca4e83d646ea7

Addresses used during tests:

0x2fe7b7	0x2fe7b7Afaf9301cBC9F7A146C70BA9FAaE1Ad9Be
0x6aa9B	0x6aa9BCc5177928DAd3e08E5120B7952087F23419
0xe2789	0xe2789d4cc1f6B82016a58E2d67DBf728AceE8Bb9
0xc41E5e6	0xc41E5e6E2418E2b591AC762b01C7d8FFC47AB5b5
0x5750B9	0x5750B95b995C150E0F26D84d896F53BcA42AAAd8
0x7bcDd7	0x7bcDd715df242f86bB06D1a4D66A6cD8FB3309d7

Construction and ownership

Contract state before function call	**Function call**	**Expected contract state after call**	**Description**	**Observed contract state after call**
(non-existant)	From: 0x2fe7b7 constructor()	owner: 0x2fe7b7 totalSupply: 100000000 000000000000000000 balanceOf(0x2fe7b7): 100000000 000000000000000000	Contract deployment (etherscan)	owner: **0x2fe7b7** totalSupply: 100000000 000000000000000000 balanceOf(0x2fe7b7): 100000000 000000000000000000
owner: 0x2fe7b7	From: 0x6aa9B changeOwner(0xc41E5e)	owner: 0x2fe7b7 Transaction should revert	Non-owner tries to transfer ownership (etherscan)	owner: **0x2fe7b7** **Transaction reverted**
owner: 0x2fe7b7	From: 0x2fe7b7 changeOwner(0x5750B9)	owner: **0x5750B9**	Owner transfers ownership (etherscan)	owner: **0x5750B9**

(continues on next page)

Contract state before function call	Function call	Expected contract state after call	Description	Observed contract state after call
balanceOf(0xe2789): 0 balanceOf(0x5750B9): 0	From: 0xe2789 transfer(0x5750B9, 1)	balanceOf(0xe2789): 0 balanceOf(0x5750B9): 0 Transaction should revert	Address with 0 tokens tries to transfer a non-0 amount of tokens (etherscan)	balanceOf(0xe2789): **0** balanceOf(0x5750B9): **0** **Transaction reverted**
balanceOf(0x2fe7b7): 100000000 000000000000000000 balanceOf(0x5750B9): 0	From: 0x2fe7b7 transfer(0x5750B9, 115792089237316195423570985008687907853269984665640564039457584007913129639935)	balanceOf(0x2fe7b7): 100000000 000000000000000000 balanceOf(0x5750B9): 0 Transaction should revert	Address tries to trigger integer overflow by transferring 2^256 - 1 tokens (etherscan)	balanceOf(0x2fe7b7): **100000000** **000000000000000000** balanceOf(0x5750B9): **0** **Transaction reverted**
balanceOf(0x2fe7b7): 100000000 000000000000000000 balanceOf(0x5750B9): 0	From: 0x2fe7b7 transfer(0x5750B9, 500)	balanceOf(0x2fe7b7): **99999999** **999999999999999500** balanceOf(0x5750B9): **500**	Address transfers tokens (etherscan)	balanceOf(0x2fe7b7): **99999999** **999999999999999500** balanceOf(0x5750B9): **500**
allowance(0x5750B9, 0xe2789): 0	From: 0x5750B9 approve(0xe2789, 100)	allowance(0x5750B9, 0xe2789): **100**	Address approves tokens (etherscan)	allowance(0x5750B9, 0xe2789): **100**
allowance(0x5750B9, 0xe2789): 100	From: 0x5750B9 approve(0xe2789, 70)	allowance(0x5750B9, 0xe2789): **70**	Address reduces token approval (etherscan)	allowance(0x5750B9, 0xe2789): **70**
allowance(0x5750B9, 0xe2789): 70 balanceOf(0x5750B9): 500 balanceOf(0x6aa9B): 0	From: 0xe2789 transferFrom(0x5750B9, 0x6aa9B, 90)	allowance(0x5750B9, 0xe2789): 70 balanceOf(0x5750B9): 500 balanceOf(0x6aa9B): 0 Transaction should revert	Address tries to transferFrom more than they are allowed (etherscan)	allowance(0x5750B9, 0xe2789): **70** balanceOf(0x5750B9): **500** balanceOf(0x6aa9B): **0** **Transaction reverted**

(continues on next page)

Contract state before function call	Function call	Expected contract state after call	Description	Observed contract state after call
allowance(0x5750B9, 0xe2789): 70 balanceOf(0x5750B9): 500 balanceOf(0x6aa9B): 0	From: 0xe2789 transferFrom(0x5750B9, 0x6aa9B, 50)	allowance(0x5750B9, 0xe2789): **20** balanceOf(0x5750B9): **450** balanceOf(0x6aa9B): **50**	Address uses transferFrom (etherscan)	allowance(0x5750B9, 0xe2789): **20** balanceOf(0x5750B9): **450** balanceOf(0x6aa9B): **50**
balanceOf(0x6aa9B): 50	From: 0x6aa9B mint(1000)	balanceOf(0x6aa9B): 50 Transaction should revert	Non-owner tries to mint tokens (etherscan)	balanceOf(0x6aa9B): **50** **Transaction reverted**
balanceOf(0x7bcDd7): 0	From: 0x6aa9B mintToAddress(0x7bcDd7, 1000)	balanceOf(0x7bcDd7): 0 Transaction should revert	Non-owner tries to mint tokens to other address (etherscan)	balanceOf(0x7bcDd7): **0** **Transaction reverted**
balanceOf(0x6aa9B): 50	From: 0x6aa9B burn(10)	balanceOf(0x6aa9B): 50 Transaction should revert	Non-owner tries to burn tokens (etherscan)	balanceOf(0x6aa9B): **50** **Transaction reverted**
balanceOf(0x5750B9): 450	From: 0x6aa9B burnFromAddress(0x5750B9, 50)	balanceOf(0x5750B9): 450 Transaction should revert	Non-owner tries to burn tokens from other address (etherscan)	balanceOf(0x5750B9): 450 **Transaction reverted**
balanceOf(0x5750B9): 450	From: 0x5750B9 mint(1000)	balanceOf(0x5750B9): **1450**	Owner mints tokens (etherscan)	balanceOf(0x5750B9): **1450**
balanceOf(0x6aa9B): 50	From: 0x5750B9 mintToAddress(0x6aa9B, 1000)	balanceOf(0x6aa9B): **1050**	Owner mints tokens to other address (etherscan)	balanceOf(0x6aa9B): **1050**
balanceOf(0x5750B9): 1450	From: 0x5750B9 burn(1000)	balanceOf(0x5750B9): **450**	Owner burns tokens (etherscan)	balanceOf(0x5750B9): **450**

(continues on next page)

Contract state before function call	Function call	Expected contract state after call	Description	Observed contract state after call
balanceOf(0x6aa9B): 1050	From: 0x5750B9 burnFromAddress(0x6aa9B, 100)	balanceOf(0x6aa9B): **950**	Owner burns tokens from other address (etherscan)	balanceOf(0x6aa9B): **950**
balanceOf(0x6aa9B): 950 balanceOf(0x7bcDd7): 0 balanceOf(0x5750B9): 450 balanceOf(0xc41E5e): 0	From: 0x6aa9B multiTransfer([0x7bcDd7, 0x5750B9, 0xc41E5e], [30, 40, 50])	balanceOf(0x6aa9B): **830** balanceOf(0x7bcDd7): **30** balanceOf(0x5750B9): **490** balanceOf(0xc41E5e): **50**	Multi-transfer to 3 addresses (etherscan)	balanceOf(0x6aa9B): **830** balanceOf(0x7bcDd7): **30** balanceOf(0x5750B9): **490** balanceOf(0xc41E5e): **50**
transferStatus: true	From: 0x6aa9B changeTransferStatus (false)	transferStatus: true Transaction should revert	Non-owner tries to disable transfers (etherscan)	transferStatus: **true** **Transaction reverted**
transferStatus: true	From: 0x5750B9 changeTransferStatus (false)	transferStatus: **false**	Owner disables transfers (etherscan)	transferStatus: **false**
balanceOf(0x5750B9): 490	From: 0x5750B9 transfer(0xc41E5e, 90)	balanceOf(0x5750B9): 490 Transaction should revert	Transfer attempt while transfers are disabled (etherscan)	balanceOf(0x5750B9): 490 **Transaction reverted**

Test result summary:

All code paths of all functions have been tested under various conditions.
All tests have passed.

Solidity compiler version

The smart contract can only be compiled using Solidity compiler version 0.5.7:
Line 1: `pragma solidity 0.5.7;`

There are **no known bugs** in this compiler version.

Optimization

The most gas-expensive instructions of the Ethereum Virtual Machine are `SSTORE` and `CREATE`. Therefore, to effectively check whether any gas is wasted during contract execution, we calculate the minimum amount of times these instructions have to be used and compare it to the amount of times they are actually used in each non-view and non-pure contract function:

Function	SSTORE		CREATE	
	Required	Used	Required	Used
`Owned constructor`	**1**	**1**	**0**	**0**
`changeOwner`	**1**	**1**	**0**	**0**
`changeTransferStatus`	**1**	**1**	**0**	**0**
`mint`	**2**	**2**	**0**	**0**
`mintToAddress`	**2**	**2**	**0**	**0**
`burn`	**2**	**2**	**0**	**0**
`burnFromAddress`	**2**	**2**	**0**	**0**
`Core constructor`	**1**	**1**	**0**	**0**
`transfer`	**2**	**2**	**0**	**0**
`transferFrom`	**3**	**3**	**0**	**0**
`multiTransfer`	**n + 1**	**n + 1**	**0**	**0**
`approve`	**1**	**1**	**0**	**0**

The most expensive EVM instructions are used exactly as many times as they need to be used.
The Stokx smart contract **does not waste** much gas on unnecessary computations.

Summary

ERC20 compatibility	**Compatible**
Vulnerabilities or other bugs in compiler	**No known bugs or vulnerabilities**
Code legibility	**Legible**
Code optimization	**Efficient**
Test results of token contract	**All passed**
Manual analysis	**No issues found**

Conclusion

Based on our findings during this audit, we hereby issue a **positive** recommendation towards the Stokx token smart contract and its deployment on any Ethereum network in a production environment.

Date of audit: 2019-04-17
Auditing agency: Jesbus Technology
Auditor name: Jesse Busman
Auditor signature: