SmartDec

# TokenBridge (by POA Network) Smart Contracts Security Analysis

This report is public.

Published: July 31, 2019.

Abstract . . . 3
Disclaimer . . . 3
Summary . . . 3
General recommendations . . . 3
Checklist . . . 4
Procedure . . . 5
Checked vulnerabilities . . . 6
Project overview . . . 7
Project description . . . 7
The latest version of the code . . . 7
Project architecture . . . 7
Automated analysis . . . 8
Manual analysis . . . 11
Critical issues . . . 11
Medium severity issues . . . 11
Code logic . . . 11
Overpowered owner . . . 11
Low severity issues . . . 12
Defines a return type but never explicitly returns a value . . . 12
Fallback function requires too much gas . . . 13
Upgrade code to Solidity 0.5.x . . . 13
Gas limit and loops . . . 13
Extra gas consumption . . . 14
Excessive gas consumption in a loop . . . 14
revert() vs require() . . . 14
Assert violation . . . 15
Redundant fallback function . . . 15
Redundant code . . . 15
Code style . . . 18
Code logic . . . 18
Misleading comment . . . 19
Missing input validation . . . 19
Wrong import of OpenZeppelin library . . . 20
Lack of documentation . . . 20
Missing return value of ERC20 tokens . . . 20
Private modifier . . . 20

Notes . . . . . 21
Gas limit and loops . . . . . 21
Prefer external to public visibility level . . . . . 21
Appendix . . . . . 22
Compilation output . . . . . 22
Tests output . . . . . 35
Solhint output . . . . . 48
Solium output . . . . . 50

# Abstract

In this report, we consider the security of the TokenBridge project. Our task is to find and describe security issues in the smart contracts of the platform.

# 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 smart contracts. Besides, security audit is not an investment advice.

# Summary

In this report, we considered the security of TokenBridge smart contracts. We performed our audit according to the procedure described below.

The audit showed no critical issues. However, a number of medium and low severity issues were found. They do not endanger project security.

All of the issues were addressed, some of them fixed in the latest version of the code.

# General recommendations

The contracts code is of good quality. The developers have addressed all the issues, thus we do not have any additional recommendations.

# Checklist

## Security

The audit showed no vulnerabilities.
Here by vulnerabilities we mean security issues that can be exploited by an external attacker. This does not include low severity issues, documentation mismatches, overpowered contract owner, and some other kinds of bugs.

## Compliance with the documentation

The audit showed no discrepancies between the code and the provided documentation.

## ERC20 compliance

We checked ERC20 compliance during the audit. The audit showed that **ERC677BridgeToken** contract was fully ERC20 compliant.

### ERC20 MUST

The audit showed no ERC20 “MUST” requirements violations.

### ERC20 SHOULD

The audit showed no ERC20 “SHOULD” requirements violations.

## Tests

The text below is for technical use; it details the statements made in Summary and General recommendations.

# Procedure

In our audit, we consider the following crucial features of the smart contract code:

1. Whether the code is secure.
2. Whether the code corresponds to the documentation (including whitepaper).
3. Whether the code meets best practices in efficient use of gas, code readability, etc.

We perform our audit according to the following procedure:

- automated analysis
  - we scan project's smart contracts with our own Solidity static code analyzer SmartCheck
  - we scan project's smart contracts with several publicly available automated Solidity analysis tools such as Remix, Slither and Solhint
  - we manually verify (reject or confirm) all the issues found by tools
- manual audit
  - we manually analyze smart contracts for security vulnerabilities
  - we check smart contracts logic and compare it with the one described in the documentation
  - we check ERC20 compliance
  - we run tests and check code coverage
- report
  - we report all the issues found to the developer during the audit process
  - we check the issues fixed by the developer
  - we reflect all the gathered information in the report

# Checked vulnerabilities

We have scanned TokenBridgesmart contracts for commonly known and more specific vulnerabilities. Here are some of the commonly known vulnerabilities that we considered (the full list includes them but is not limited to them):

- Reentrancy
- Front running
- DoS with (unexpected) revert
- DoS with block gas limit
- Gas limit and loops
- Locked money
- Integer overflow/underflow
- Unchecked external call
- ERC20 Standard violation
- Authentication with tx.origin
- Unsafe use of timestamp
- Using blockhash for randomness
- Balance equality
- Unsafe transfer of ether
- Fallback abuse
- Using inline assembly
- Short address attack
- Private modifier
- Compiler version not fixed
- Style guide violation
- Unsafe type deduction
- Implicit visibility level
- Use delete for arrays
- Byte array
- Incorrect use of assert/require
- Using deprecated constructions

# Project overview

## Project description

In our analysis we consider TokenBridge specification (”README.md and REWARD_MANAGEMENT.md” in the repo) and smart contracts’ code (version on commit bbb97a63c900f03a902d0e82358abac3b294e4d9).

### The latest version of the code

After the initial audit, some fixes were applied and the code was updated to the latest version (commit b66a678648ea89b6441382c7a7adefb3b0b02667).

## Project architecture

For the audit, we were provided with the truffle project. The project is an npm package and includes tests.

- The project successfully compiles with `truffle compile` command (with some warnings, see Compilation output in Appendix)
- The project successfully passes all the tests with 100% coverage

The total LOC of audited Solidity sources is 2297.

# Automated analysis

We used several publicly available automated Solidity analysis tools. Here are the combined results of SmartCheck, Solhint, Slither and Remix scanning. All the issues found by tools were manually checked (rejected or confirmed).

**True positives** are constructions that were discovered by the tools as vulnerabilities and can actually be exploited by attackers or lead to incorrect contracts operation.

**False positives** are constructions that were discovered by the tools as vulnerabilities but do not consist a security threat.

Cases when these issues lead to actual bugs or vulnerabilities are described in the next section.

| Tool | Rule | True positives | False positives |
|---|---|---|---|
| SmartCheck | Locked money | | 4 |
| | Unchecked low-level call | | 2 |
| | Overpowered role | 21 | |
| | Costly loop | 4 | 5 |
| | Unprotected SELFDESTRUCT instruction | | 1 |
| | Use of SafeMath | | 6 |
| | Hardcoded address | | 3 |
| | Prefer external to public visibility level | 76 | 21 |
| | Upgrade code to Solidity 0.5.x | 35 | 1 |
| | Extra gas consumption | 1 | 4 |
| | Revert inside the if-operator | 3 | 1 |
| | Assert violation | 1 | |
| | Compiler version not fixed | | 1 |
| | Requirement violation | 1 | 21 |

| | | | |
|---|---|---|---|
| | View-function should not change state | 1 | 8 |
| | Non-initialized return value | 3 | 2 |
| | Use of assembly | | 9 |
| | Reentrancy | 2 | 7 |
| | Weak sources of randomness from chain attributes | | 7 |
| | Redundant fallback function | 1 | |
| | Implicit visibility level | 1 | |
| | Pure-functions should not read/change state | | 2 |
| | Replace multiple return values with a struct | | 2 |
| | Deprecated constructions | | 4 |
| | Private modifier | 1 | |
| Total SmartCheck | | 151 | 111 |
| Slither | Constant function | | 1 |
| | Naming convention | | 2 |
| | Solc version | 8 | |
| | Low level calls | | 1 |
| | External function | 23 | |
| | Assembly | | 1 |
| Total Slither | | 31 | 5 |
| Remix | Constant but potentially should not be | | 22 |

| | | | |
|---|---|---|---|
| | Defines a return type but never explicitly returns a value | 4 | 1 |
| | Fallback function requires too much gas | 3 | 4 |
| | Potential Violation of Checks-Effects-Interaction pattern | | 5 |
| | Should be constant but is not | | 14 |
| | Use of inline assembly | | 12 |
| | Use of ”call” | | 2 |
| | Use of ”send” | | 1 |
| | Use of selfdestruct | | 1 |
| | Use of ”delegatecall” | | 4 |
| | Use of ”now” | | 1 |
| Total Remix | | 7 | 67 |
| Solhint | Avoid to use low level calls | | 5 |
| | Avoid to use inline assembly. It is acceptable only in rare cases | | 13 |
| | Fallback function must be simple | 2 | 2 |
| | Avoid to use ”.call.value()()” | | 1 |
| | Avoid to make time-based decisions in your business logic | | 1 |
| | Avoid multiple calls of ”send” method in single transaction | | 4 |
| Total Solhint | | 2 | 26 |
| **Total Overall** | | **191** | **209** |

# Manual analysis

The contracts were completely manually analyzed, their logic was checkedand compared with the one described in the documentation. Besides, the results of the automated analysis were manually verified. All the confirmed issues are described below.

## Critical issues

Critical issues seriously endanger smart contracts security. We highly recommend fixing them.

**The audit showed no critical issues.**

## Medium severity issues

Medium issues can influence smart contracts operation in current implementation. We highly recommend addressing them.

### Code logic

Since ERC20 Standard `transfer()` function is used for placing tokens to the Foreign bridge in `erc20-to-erc20` and `erc20-to-native` modes, there is no convenient way to limit the amount of tokens that a user can send. Thus, there could be a situation when a user sends too many tokens to the Foreign bridge. In this case, **OverdrawManagement** contract functionality is used. When validators try to validate the amount of tokens that exceeds the limit, `setTxAboveLimits()` function is called. After that, the bridge owner has to call `fixAssetsAboveLimits()` function, which emits `UserRequestForSignature()` event. Then, the validators should sign the returning transaction and return the tokens back to the user on the Foreign side. But in this case, the validators will receive the fee.

In case of `erc20-to-native` mode, **FeeManagerErcToNative** fee manager, the fee will be charged from the Home bridge contract address. In other cases (in POSDAO environment), it will lead to the bridge imbalance: amount of coins/tokens on the Home side will increase compared to the amount of tokens locked on the Foreign side. It will happen at the moment when the fee is paid after the final signature is received: the fee is calculated based on the amount of tokens that will be transferred and is minted to validators. At the same time, the full requested amount of tokens will be unlocked when confirmations are passed to the foreign side.

*This issue has been fixed in the pull request #218.*

### Overpowered owner

The owner has the following powers:

- The owner has an ability to set crucial parameters at own will.

*Comment from the developers: “It is assumed that there are three different roles:*
*- an account that could upgrade the contracts*
*- an account that could manage parameters of the bridge contracts*
*- an account that could manage parameters of the validators contracts*
*The first account is indeed has overpowered role since is able to modify the logic of the bridge contract at all. Others could introduce a limited impact. For each of three accounts it is assumed that they are a multisig wallets. And owners of the wallets are responsible for authorizing the bridge configuration actions.”*

- Users heavily rely on the owner and validators. The owner manually adds validators and then ensures that validators are the same in both Home and Foreign networks.

  *Comment from the developers: “It is the responsibility of the token bridge owner to provide bridge operations. If a current set bridge validators sabotage the bridge operations, the token bridge owner could modify the set of validators (remove old ones and one at least one new) in order to unlock funds.”*

- Staking contract can reallocate funds from any address to its own by calling `stake()` function. The owner is able to set any address as a staking contract by calling `setStakingContract()` function.

  *Comment from the developers: “The bridge contracts relies on the staking contract functionality. The staking contract is not the part of the token bridge code base.”*

- Block reward contract can mint tokens to anyone without any limits by calling `mintReward()` function. The owner is able to set any address as a block reward contract by calling `setBlockRewardContract()` function.

In the current implementation, the system depends heavily on the owner and validators of the contracts. In this case, there are scenarios that may lead to undesirable consequences for investors, e.g. if the owner’s private keys become compromised. Thus, we recommend designing contracts in a trustless manner.

# Low severity issues

Low severity issues can influence smart contracts operation in future versions of code. We recommend taking them into account.

## Defines a return type but never explicitly returns a value

The following functions defines a return type, but never initialize the return values.

- **BasicHomeBridge.sol**, `onExecuteAffirmation()` function

  *This issue has been fixed in the pull request #203.*

- **HomeBridgeErcToErc.sol**, `_rewardableInitialize()` function

*This issue has been fixed in the pull request #221.*

- **HomeBridgeErcToErc.sol**, `_initialize()` function

  *This issue has been fixed in the pull request #221.*

We recommend adding return statements or removing return types from function declaration.

## Fallback function requires too much gas

Fallback function of the following contracts contract contains too much logic.

- **HomeBridgeErcToNative**
- **ClassicHomeBridgeNativeToErc**
- **HomeBridgeNativeToErc**

We recommend moving their functionality to separate public functions.

*The issues have been fixed in the pull request #203.*

## Upgrade code to Solidity 0.5.x

The code uses solidity compiler version `0.4.24`.

We recommend migrating code to a new version of compiler (0.5.10) since it contains several important changes. In order to update contracts to compiler version 0.5.10 the developers should follow Solidity documentation.

*Comment from the developers: "There is an issue created for this. But it will not be addressed as part of the audit since requires additional resources for implementation and testing.*

## Gas limit and loops

The following loops traverse through arrays of variable length:

1. **ERC677BridgeTokenRewardable.sol**, line 38:

```
for (uint256 i = 0; i < _receivers.length; i++)
```

*Comment from the developers: "The only place where the mintToken method is called is the following one. It is defined here that receivers array size is `stakers.length / DELEGATORS_ALIQUOT`. Number of stakers cannot be greater than 3000. `DELEGATORS_ALIQUOT` is 2. So, the maximum size of the receivers array is 1500.*

2. **RewardableValidators.sol**, line 24:

```
for (uint256 i = 0; i < _initialValidators.length; i++)
```

*This issue has been fixed in the pull request #239.*

3. **ValidatorsFeeManager.sol**, line 41:

```
while (nextValidator != F_ADDR)
```

*Comment from the developers: "There is no simple way to say what should be a limit since the method refers on the functionality implemented in a particular fee manager."*

Therefore, if there are too many items in these arrays, the execution of the corresponding functions will fail due to an out-of-gas exception.

In these cases, we recommend separating the calls into several transactions.

## Extra gas consumption

## Excessive gas consumption in a loop

`_receivers.length` variable (**ERC677BridgeTokenRewardable.sol** file, line 38) is read from the storage on every iteration of the corresponding loop. Reading from local memory requires significantly less gas compared to reading from the storage.

Thus, we recommend placing this variable into local memory in order to reduce gas consumption.

*This issue has been fixed in the pull request #242.*

## revert() vs require()

`revert()` is used in several places:

1. **BaseBridgeValidators.sol**, lines 46–48:

```
if (nextValidator == address(0) ) {     revert(); }
```

2. **BaseBridgeValidators.sol**, lines 77–79:

```
if (next == F_ADDR || next == address(0) ) {     revert(); }
```

3. **Message.sol**, line 109–111:

```
if (addressArrayContains(encounteredAddresses,
recoveredAddress)) {     revert(); }
```

We recommend using `require(condition);` instead of `if (!condition) revert();` to improve code readability and transparency.

*The issues have been fixed in the pull request #235.*

## Assert violation

The execution of `random()` function from **BaseFeeManager** contract can fail with an exception if `_count` argument is equal to zero.

```
function random(uint256 _count) public view returns(uint256)
{     return uint256(blockhash(block.number.sub(1))) %
_count; }
```

As a result, all the provided gas will be spent.

We recommend checking that considered variable is not zero in order to reduce gas costs.

*This issue has been fixed in the pull request #246.*

## Redundant fallback function

The payment rejection fallback in **HomeBridgeErcToErc** contract is redundant. Before Solidity 0.4.0, payment rejection was done manually:

```
function () { revert(); }
```

Starting from Solidity 0.4.0, contracts without a fallback function automatically revert payments, therefore, the fallback function in this contract redundant.

*This issue has been fixed in the pull request #224.*

## Redundant code

The project has the following redundant code issues:

- Both **OwnedUpgradeabilityProxy** and **ClassicEternalStorageProxy** contracts have `proxyOwner()` and `upgradeabilityOwner()` functions that return the same value. We recommend removing one of the functions in both contracts.

*This issue has been fixed in the pull request #198.*

- `claimTokens()` function from **ForeignBridgeErcToNative** contract has `onlyIfOwnerOfProxy()` modifier. However, this function calls `super()` function from **BasicBridge** contract, which also has the same modifier. Hence, this modifier is called twice. We recommend removing it from `claimTokens()` function.

  *This issue has been fixed in the pull request #198.*

- In fallback function from **HomeBridgeErcToNative** contract, `totalBurntCoins()` function is called twice: at lines 24 and 34. We recommend avoiding multiple reads of storage variables in the same function in order to decrease execution costs.

  *This issue has been fixed in the pull request #203.*

- `fireEventOnTokenTransfer()` function from **HomeBridgeErcToNative** is never used.

  *This issue has been fixed in the pull request #203.*

- `messages()` function from **BasicHomeBridge** contract is redundant since it is called only from `message()` function and its functionality can be moved there.

  *This issue has been fixed in the pull request #203.*

- `signatures()` function from **BasicHomeBridge** contract is redundant since it is called only from `signature()` function and its functionality can be moved there.

  *This issue has been fixed in the pull request #203.*

- In **BasicHomeBridge** contract, the following functions have an empty body:
  - `onExecuteAffirmation()`
  - `onSignaturesCollected()`
  - `affirmationWithinLimits()`
  - `onFailedAffirmation()`

  We recommend not implementing these functions and using `;` instead in order to make these smart contracts abstract.

  *The issues have been fixed in the pull request #203.*

- **Message.sol**, line 58:

```
recipient := and(mload(add(message, 20)),
0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF)
```

  `and` operation is redundant since this check is done automatically for variables with `address` type.

  *This issue has been fixed in the pull request #227.*

- **HomeBridgeNativeToErc** contract inherits from **RewardableHomeBridgeNativeToErc**. This means, if **HomeBridgeNativeToErc** contract is non-rewardable, it has lots of unused functions.

  *Comment from the developers: "It is done intentionally since will allow enable gathering fees from the bridge operation just by setting the Fee Manager contract without necessity to upgrade entire bridge contract."*

- `onSignatureCollected()` functions are identical in all bridges, except one. In **HomeBridgeNativeToErc** contract, this function has the following check at line 132:

```
if (fee != 0) {
```

  If the case where `fee` is equal to zero is invalid, it is not clear why there are no such checks in other bridge contracts. We recommend clarifying the possible cases for the function in the documentation.

  *Comment from the developers: "There are two modes for the native-to-erc20 bridge to work with the Fee Manager*
  *1. the fee collected on each side of the bridge: fees from home-to-foreign transfers are collected on the foreign bridge contract, fees from foreign-to-home transfers are collected on the home side.*
  *2. the fee collected on the home side only for both direction. For the mode 1 `calculateFee` invoked by `onSignatureCollected()` returns zero. So, fee will be zero as well. Such bridge mode so far is specific for the native-to-erc20 bridge only that's why the check `if (fee != 0) { ... }` exists in this version of the contract."*

- `_rewardableInitialize()` and `_initialize()` functions in **HomeBridgeErcToErc** contract always return `false`. We recommend removing return values in order to improve code readability.

  *This issue has been fixed in the pull request #221.*

- The following checks are redundant:
  - **ERC677BridgeTokenRewardable**, line 18:

```
_blockRewardContract != address(0)
```

  - **ERC677BridgeTokenRewardable**, line 23:

```
_stakingContract != address(0)
```

  Later in these lines, it is checked whether addresses are contracts. The fact that the checked address is a contract implies it is not zero address since no contracts can be deployed at `address(0)`. Thus, these checks are redundant.

*The issues have been fixed in the pull request #225.*

- The following checks at line 119, **HomeBridgeErcToNative.sol** are redundant:

```
_blockReward != address(0) && isContract(_blockReward)
```

  Since there is an external call in this line, the function will revert if `_blockReward` is not a contract or if the called function is not implemented.

  *This issue has been fixed in the pull request #225.*

- `setInitialize()` and `setFixedAssets()` functions from **HomeBridgeErcToNative** contract are always called with `true` argument.

  *This issue has been fixed in the pull request #203.*

We highly recommend removing redundant code in order to improve code readability and transparency and decrease the cost of deployment and execution.

## Code style

There are several code style issues in the project:

- Some getter functions have `get` prefix in their names while others do not have it. Moreover, `POSDAO` is a postfix in all the contracts except one — **POSDAOHomeBridgeErcToErc**. We recommend sticking to the same style when choosing names for functions or contracts.

  *This issue has been fixed in the pull request #226.*

- In **BaseBridgeValidators** contract, line 104, `"deployedAtBlock"` string should be used inside `abi.encodePacked()` function since it is suggested by Solidity documentation.

  *This issue has been fixed in the pull request #203.*

- `affirmationWithinLimits()` function is the same in all Home bridges. We recommend avoiding code duplication in order to improve code readability and reduce the chance of making a mistake when upgrading the code.

  *This issue has been fixed in the pull request #223.*

We recommend fixing these issues in order to improve code readability.

## Code logic

There are places in the project that contain code logic issues:

- **FeeManagerNativeToErc** contract is used in both networks. However, `onSignatureFeeDistribution()` function is not used in Home network and `onAffirmationFeeDistribution()` function is not used in Foreign network.

  We recommend splitting **FeeManagerNativeToErc** contract into two.

  *Comment from the developers: "It is done to keep consistent changes during improvements. The impact is a high gas usage during the deployment."*

- In **ERC677BridgeToken** contract, the logic of `transfer()` function is modified, however, `transferFrom()` function remains unchanged.

  We recommend changing the logic of `transferFrom()` function in order to make them more consistent.

  *This issue has been fixed in the pull request #220.*

## Misleading comment

Line 89 in **Message.sol** contains the following comment:

```
// message is always 84 length
```

However, the next line has the following code:

```
string memory msgLength = "104";
```

We recommend fixing this comment in order to avoid confusion.

*This issue has been fixed in the pull request #204.*

## Missing input validation

There are functions where input values are not validated correctly:

- In `hasEnoughValidSignatures()` function, there is no check that `_vs`, `_rs`, and `_ss` arrays are of equal lengths.

  *Comment from the developers: "It is not necessary to check that arrays have the same length since it will be handled during attempt to access to the corresponding elements in the loop and the call will be reverted. It will save gas for the rational validators actions and still be safe enough from security point of view."*

- In **BaseFeeManager** contract, `setHomeFee()` and `setForeignFee()` functions should have the following check:

```
require(_fee <= 1 ether);
```

We recommend implementing the mentioned checks.

*This issue has been fixed in the pull request #209.*

## Wrong import of OpenZeppelin library

In the current implementation, **OpenZeppelin** files are added to the repo. This violates **OpenZeppelin**'s MIT license, which requires the license and copyright notice to be included if its code is used. Moreover, it is more difficult and error-prone to update the code manually added to the repo.

We highly recommend using npm in order to guarantee that original **OpenZeppelin** contracts are used with no modifications. This also allows for any bug-fixes to be easily integrated into the codebase.

*This issue has been fixed in the pull request #222.*

## Lack of documentation

In the documentation, it is not described how the bridge should work in `POSDAO` environment.

We recommend amending the documentation.

*This issue has been fixed in the pull request #202.*

## Missing return value of ERC20 tokens

`claimTokens()` function from **ERC677BridgeToken** does not work with tokens that do not return `true` on `transfer()` function calls. However, some older ERC20 tokens do not provide any return value when functions such as `transferFrom()` are called.

We recommend using **SafeERC20** contract from **OpenZeppelin** library.

*This issue has been fixed in the pull request #213.*

## Private modifier

`private` modifier is used in **UpgradeabilityOwnerStorage.sol**, line 10:

```
address private _upgradeabilityOwner;
```

Contrary to a popular misconception, the private modifier does not make a variable invisible. Miners have access to all contracts' code and data. Developers must account for the lack of privacy in Ethereum.

*This issue has been fixed in the pull request #198.*

# Notes

## Gas limit and loops

The loop at **BridgeValidators.sol**, line 22 traverses through an array of variable length:

```
for (uint256 i = 0; i < _initialValidators.length; i++)
```

`_initialValidators` array is passed as `initialize()` function parameter. Therefore, if there are too many items in `_initialValidators` array, the execution of `initialize()` function will fail due to an out-of-gas exception.

We recommend keeping this problem in mind since in the current implementation function call cannot be split into several calls.

*This issue has been fixed in the pull request #239.*

## Prefer external to public visibility level

Many functions in the code have `public` visibility when they could have `external` visibility. We recommend using the latter one since it indicates that the functions are not called internally.

This analysis was performed by SmartDec.

Alexander Seleznev, Chief Business Development Officer
Boris Nikashin, Project Manager
Igor Sobolev, Analyst
Pavel Kondratenkov, Analyst
Alexander Drygin, Analyst

July 31, 2019

# Appendix

## Compilation output

```
Compiling your contracts...
===========================
> Compiling ./contracts/libraries/SafeMath.sol
> Compiling ./contracts/test/BlockReward.sol
> Compiling ./contracts/upgradeable_contracts/BaseBridgeVali
dators.sol
> Compiling ./contracts/upgradeable_contracts/BaseFeeManager
.sol
> Compiling ./contracts/upgradeable_contracts/BasicBridge.so
l
> Compiling ./contracts/upgradeable_contracts/BasicForeignBr
idge.sol
> Compiling ./contracts/upgradeable_contracts/BasicHomeBridg
e.sol
> Compiling ./contracts/upgradeable_contracts/BlockRewardFee
Manager.sol
> Compiling ./contracts/upgradeable_contracts/BridgeValidato
rs.sol
> Compiling ./contracts/upgradeable_contracts/ERC677Bridge.s
ol
> Compiling ./contracts/upgradeable_contracts/ERC677BridgeFo
rBurnableMintableToken.sol
> Compiling ./contracts/upgradeable_contracts/OverdrawManage
ment.sol
> Compiling ./contracts/upgradeable_contracts/Ownable.sol
> Compiling ./contracts/upgradeable_contracts/RewardableBrid
ge.sol
> Compiling ./contracts/upgradeable_contracts/RewardableVali
dators.sol
> Compiling ./contracts/upgradeable_contracts/ValidatorsFeeM
anager.sol
> Compiling ./contracts/upgradeable_contracts/erc20_to_erc20
/BasicForeignBridgeErcToErc.sol
> Compiling ./contracts/upgradeable_contracts/erc20_to_erc20
/FeeManagerErcToErcPOSDAO.sol
> Compiling ./contracts/upgradeable_contracts/erc20_to_erc20
/ForeignBridgeErc677ToErc677.sol
> Compiling ./contracts/upgradeable_contracts/erc20_to_erc20
/ForeignBridgeErcToErc.sol
```

```
> Compiling ./contracts/upgradeable_contracts/erc20_to_erc20
/HomeBridgeErcToErc.sol
> Compiling ./contracts/upgradeable_contracts/erc20_to_erc20
/POSDAOHomeBridgeErcToErc.sol
> Compiling ./contracts/upgradeable_contracts/erc20_to_erc20
/RewardableHomeBridgeErcToErc.sol
> Compiling ./contracts/upgradeable_contracts/erc20_to_nativ
e/FeeManagerErcToNative.sol
> Compiling ./contracts/upgradeable_contracts/erc20_to_nativ
e/FeeManagerErcToNativePOSDAO.sol
> Compiling ./contracts/upgradeable_contracts/erc20_to_nativ
e/ForeignBridgeErcToNative.sol
> Compiling ./contracts/upgradeable_contracts/erc20_to_nativ
e/HomeBridgeErcToNative.sol
> Compiling ./contracts/upgradeable_contracts/erc20_to_nativ
e/RewardableHomeBridgeErcToNative.sol
> Compiling ./contracts/upgradeable_contracts/native_to_erc2
0/ClassicHomeBridgeNativeToErc.sol
> Compiling ./contracts/upgradeable_contracts/native_to_erc2
0/FeeManagerNativeToErc.sol
> Compiling ./contracts/upgradeable_contracts/native_to_erc2
0/FeeManagerNativeToErcBothDirections.sol
> Compiling ./contracts/upgradeable_contracts/native_to_erc2
0/ForeignBridgeNativeToErc.sol
> Compiling ./contracts/upgradeable_contracts/native_to_erc2
0/HomeBridgeNativeToErc.sol
> Compiling ./contracts/upgradeable_contracts/native_to_erc2
0/RewardableForeignBridgeNativeToErc.sol
> Compiling ./contracts/upgradeable_contracts/native_to_erc2
0/RewardableHomeBridgeNativeToErc.sol

    > compilation warnings encountered:

./contracts/IBlockReward.sol:6:5: Warning: Functions in inte
rfaces should be declared external.
    function mintedTotally() public view returns (uint256);
    ^-----------------------------------------------------^
,./contracts/IBlockReward.sol:7:5: Warning: Functions in int
erfaces should be declared external.
    function mintedTotallyByBridge(address _bridge) public v
iew returns(uint256);
    ^-----------------------------------------------------
--------------------^
,./contracts/IBridgeValidators.sol:5:5: Warning: Functions i
n interfaces should be declared external.
```

```
    function isValidator(address _validator) public view ret
urns(bool);
    ^-----------------------------------------------------
----------^
,./contracts/IBridgeValidators.sol:6:5: Warning: Functions i
n interfaces should be declared external.
    function requiredSignatures() public view returns(uint25
6);
    ^-------------------------------------------------------
--^
,./contracts/IBridgeValidators.sol:7:5: Warning: Functions i
n interfaces should be declared external.
    function owner() public view returns(address);
    ^-------------------------------------------^
,./contracts/IOwnedUpgradeabilityProxy.sol:5:5: Warning: Fun
ctions in interfaces should be declared external.
    function proxyOwner() public view returns (address);
    ^-----------------------------------------------^
,./contracts/IRewardableValidators.sol:5:5: Warning: Functio
ns in interfaces should be declared external.
    function isValidator(address _validator) public view ret
urns(bool);
    ^-----------------------------------------------------
----------^
,./contracts/IRewardableValidators.sol:6:5: Warning: Functio
ns in interfaces should be declared external.
    function requiredSignatures() public view returns(uint25
6);
    ^-------------------------------------------------------
--^
,./contracts/IRewardableValidators.sol:7:5: Warning: Functio
ns in interfaces should be declared external.
    function owner() public view returns(address);
    ^-------------------------------------------^
,./contracts/IRewardableValidators.sol:8:5: Warning: Functio
ns in interfaces should be declared external.
    function validatorList() public view returns (address[])
;
    ^-------------------------------------------------------
^
,./contracts/IRewardableValidators.sol:9:5: Warning: Functio
ns in interfaces should be declared external.
    function getValidatorRewardAddress(address _validator) p
ublic view returns(address);
    ^-------------------------------------------------------
```

```
--------------------------^
,./contracts/IRewardableValidators.sol:10:5: Warning: Functi
ons in interfaces should be declared external.
    function validatorCount() public view returns (uint256);
    ^-----------------------------------------------------^
,./contracts/IRewardableValidators.sol:11:5: Warning: Functi
ons in interfaces should be declared external.
    function getNextValidator(address _address) public view
returns (address);
    ^-----------------------------------------------------
-----------------^
,./contracts/upgradeable_contracts/ERC677Bridge.sol:26:51: W
arning: Unused function parameter. Remove or comment out the
variable name to silence this warning.
    function bridgeSpecificActionsOnTokenTransfer(ERC677 _to
ken, address _from, uint256 _value) internal {
                                                  ^---------
--^
,./contracts/upgradeable_contracts/erc20_to_erc20/BasicFore
ignBridgeErcToErc.sol:47:68: Warning: Unused function parame
ter. Remove or comment out the variable name to silence this
warning.
    function onExecuteMessage(address _recipient, uint256 _a
mount, bytes32 _txHash) internal returns(bool){
                                                            ^

-------------^
,./contracts/upgradeable_contracts/erc20_to_native/ForeignBr
idgeErcToNative.sol:56:68: Warning: Unused function paramete
r. Remove or comment out the variable name to silence this w
arning.
    function onExecuteMessage(address _recipient, uint256 _a
mount, bytes32 _txHash) internal returns(bool) {
                                                            ^

-------------^
,./contracts/upgradeable_contracts/native_to_erc20/HomeBridg
eNativeToErc.sol:159:34: Warning: Unused function parameter.
Remove or comment out the variable name to silence this warn
ing.
    function onFailedAffirmation(address _recipient, uint256
_value, bytes32 _txHash) internal {
                                 ^----------------^
,./contracts/upgradeable_contracts/native_to_erc20/HomeBridg
eNativeToErc.sol:159:54: Warning: Unused function parameter.
Remove or comment out the variable name to silence this warn
```

```
ing.
    function onFailedAffirmation(address _recipient, uint256
_value, bytes32 _txHash) internal {
                                                           ^------
------^
,./contracts/upgradeable_contracts/native_to_erc20/HomeBridg
eNativeToErc.sol:159:70: Warning: Unused function parameter.
Remove or comment out the variable name to silence this warn
ing.
    function onFailedAffirmation(address _recipient, uint256
_value, bytes32 _txHash) internal {
                                                                ^
-------------^
,./contracts/upgradeable_contracts/BasicHomeBridge.sol:89:5:
Warning: Function state mutability can be restricted to pure
    function onExecuteAffirmation(address, uint256, bytes32)
internal returns(bool) {
    ^ (Relevant source part starts here and spans across mul
tiple lines).
,./contracts/upgradeable_contracts/BasicHomeBridge.sol:92:5:
Warning: Function state mutability can be restricted to pure
    function onSignaturesCollected(bytes) internal {
    ^ (Relevant source part starts here and spans across mul
tiple lines).
,./contracts/upgradeable_contracts/BasicHomeBridge.sol:160:5
: Warning: Function state mutability can be restricted to pu
re
    function affirmationWithinLimits(uint256) internal view
returns(bool) {
    ^ (Relevant source part starts here and spans across mul
tiple lines).
,./contracts/upgradeable_contracts/BasicHomeBridge.sol:164:5
: Warning: Function state mutability can be restricted to pu
re
    function onFailedAffirmation(address, uint256, bytes32)
internal {
    ^ (Relevant source part starts here and spans across mul
tiple lines).
,./contracts/upgradeable_contracts/RewardableBridge.sol:19:2
7: Warning: Function declared as view, but this expression (
potentially) modifies the state and thus requires non-payabl
e (the default) or payable.
            let result := callcode(gas, feeManager, 0x0, add
(callData, 0x20), mload(callData), 0, 32)
                          ^--------------------------------
```

```
--------------------------------------^
,./contracts/upgradeable_contracts/RewardableBridge.sol:33:2
7: Warning: Function declared as view, but this expression (
potentially) modifies the state and thus requires non-payabl
e (the default) or payable.
            let result := callcode(gas, feeManager, 0x0, add
(callData, 0x20), mload(callData), 0, 4)
                          ^--------------------------------
--------------------------------------^
,./contracts/upgradeable_contracts/RewardableBridge.sol:67:2
7: Warning: Function declared as view, but this expression (
potentially) modifies the state and thus requires non-payabl
e (the default) or payable.
            let result := callcode(gas, _impl, 0x0, add(call
Data, 0x20), mload(callData), 0, 32)
                          ^--------------------------------
----------------------------------^
,./contracts/upgradeable_contracts/erc20_to_erc20/POSDAOHome
BridgeErcToErc.sol:52:27: Warning: Function declared as view
, but this expression (potentially) modifies the state and t
hus requires non-payable (the default) or payable.
            let result := callcode(gas, feeManager, 0x0, add
(callData, 0x20), mload(callData), 0, 32)
                          ^--------------------------------
---------------------------------------^
,./contracts/upgradeable_contracts/erc20_to_native/FeeManage
rErcToNativePOSDAO.sol:17:5: Warning: Function state mutabil
ity can be restricted to pure
    function getAmountToBurn(uint256 _value) public view ret
urns(uint256) {
    ^ (Relevant source part starts here and spans across mul
tiple lines).
,./contracts/upgradeable_contracts/erc20_to_native/Rewardabl
eHomeBridgeErcToNative.sol:29:27: Warning: Function declared
as view, but this expression (potentially) modifies the stat
e and thus requires non-payable (the default) or payable.
            let result := callcode(gas, feeManager, 0x0, add
(callData, 0x20), mload(callData), 0, 32)
                          ^--------------------------------
---------------------------------------^

> Artifacts written to ./build/contracts
> Compiled successfully using:
   - solc: 0.4.24+commit.e67f0147.Emscripten.clang
```

```
Compiling your contracts...
===========================
> Compiling ./contracts/ERC677.sol
> Compiling ./contracts/ERC677BridgeToken.sol
> Compiling ./contracts/ERC677BridgeTokenRewardable.sol
> Compiling ./contracts/ERC677Receiver.sol
> Compiling ./contracts/IBlockReward.sol
> Compiling ./contracts/IBridgeValidators.sol
> Compiling ./contracts/IBurnableMintableERC677Token.sol
> Compiling ./contracts/IOwnedUpgradeabilityProxy.sol
> Compiling ./contracts/IRewardableValidators.sol
> Compiling ./contracts/Migrations.sol
> Compiling ./contracts/libraries/Message.sol
> Compiling ./contracts/libraries/SafeMath.sol
> Compiling ./contracts/test/BlockReward.sol
> Compiling ./contracts/test/Staking.sol
> Compiling ./contracts/upgradeability/ClassicEternalStorage
Proxy.sol
> Compiling ./contracts/upgradeability/EternalStorage.sol
> Compiling ./contracts/upgradeability/EternalStorageProxy.s
ol
> Compiling ./contracts/upgradeability/OwnedUpgradeabilityPr
oxy.sol
> Compiling ./contracts/upgradeability/Proxy.sol
> Compiling ./contracts/upgradeability/UpgradeabilityOwnerSt
orage.sol
> Compiling ./contracts/upgradeability/UpgradeabilityProxy.s
ol
> Compiling ./contracts/upgradeability/UpgradeabilityStorage
.sol
> Compiling ./contracts/upgradeable_contracts/BaseBridgeVali
dators.sol
> Compiling ./contracts/upgradeable_contracts/BaseFeeManager
.sol
> Compiling ./contracts/upgradeable_contracts/BasicBridge.so
l
> Compiling ./contracts/upgradeable_contracts/BasicForeignBr
idge.sol
> Compiling ./contracts/upgradeable_contracts/BasicHomeBridg
e.sol
> Compiling ./contracts/upgradeable_contracts/BlockRewardFee
Manager.sol
> Compiling ./contracts/upgradeable_contracts/BridgeValidato
rs.sol
```

```
> Compiling ./contracts/upgradeable_contracts/ERC677Bridge.s
ol
> Compiling ./contracts/upgradeable_contracts/ERC677BridgeFo
rBurnableMintableToken.sol
> Compiling ./contracts/upgradeable_contracts/FeeTypes.sol
> Compiling ./contracts/upgradeable_contracts/OverdrawManage
ment.sol
> Compiling ./contracts/upgradeable_contracts/Ownable.sol
> Compiling ./contracts/upgradeable_contracts/OwnedUpgradeab
ility.sol
> Compiling ./contracts/upgradeable_contracts/RewardableBrid
ge.sol
> Compiling ./contracts/upgradeable_contracts/RewardableVali
dators.sol
> Compiling ./contracts/upgradeable_contracts/Sacrifice.sol
> Compiling ./contracts/upgradeable_contracts/Validatable.so
l
> Compiling ./contracts/upgradeable_contracts/ValidatorsFeeM
anager.sol
> Compiling ./contracts/upgradeable_contracts/erc20_to_erc20
/BasicForeignBridgeErcToErc.sol
> Compiling ./contracts/upgradeable_contracts/erc20_to_erc20
/FeeManagerErcToErcPOSDAO.sol
> Compiling ./contracts/upgradeable_contracts/erc20_to_erc20
/ForeignBridgeErc677ToErc677.sol
> Compiling ./contracts/upgradeable_contracts/erc20_to_erc20
/ForeignBridgeErcToErc.sol
> Compiling ./contracts/upgradeable_contracts/erc20_to_erc20
/HomeBridgeErcToErc.sol
> Compiling ./contracts/upgradeable_contracts/erc20_to_erc20
/POSDAOHomeBridgeErcToErc.sol
> Compiling ./contracts/upgradeable_contracts/erc20_to_erc20
/RewardableHomeBridgeErcToErc.sol
> Compiling ./contracts/upgradeable_contracts/erc20_to_nativ
e/FeeManagerErcToNative.sol
> Compiling ./contracts/upgradeable_contracts/erc20_to_nativ
e/FeeManagerErcToNativePOSDAO.sol
> Compiling ./contracts/upgradeable_contracts/erc20_to_nativ
e/ForeignBridgeErcToNative.sol
> Compiling ./contracts/upgradeable_contracts/erc20_to_nativ
e/HomeBridgeErcToNative.sol
> Compiling ./contracts/upgradeable_contracts/erc20_to_nativ
e/RewardableHomeBridgeErcToNative.sol
> Compiling ./contracts/upgradeable_contracts/native_to_erc2
```

```
0/ClassicHomeBridgeNativeToErc.sol
> Compiling ./contracts/upgradeable_contracts/native_to_erc2
0/FeeManagerNativeToErc.sol
> Compiling ./contracts/upgradeable_contracts/native_to_erc2
0/FeeManagerNativeToErcBothDirections.sol
> Compiling ./contracts/upgradeable_contracts/native_to_erc2
0/ForeignBridgeNativeToErc.sol
> Compiling ./contracts/upgradeable_contracts/native_to_erc2
0/HomeBridgeNativeToErc.sol
> Compiling ./contracts/upgradeable_contracts/native_to_erc2
0/RewardableForeignBridgeNativeToErc.sol
> Compiling ./contracts/upgradeable_contracts/native_to_erc2
0/RewardableHomeBridgeNativeToErc.sol
> Compiling openzeppelin-solidity/contracts/math/SafeMath.so
l
> Compiling openzeppelin-solidity/contracts/ownership/Ownabl
e.sol
> Compiling openzeppelin-solidity/contracts/token/ERC20/Basi
cToken.sol
> Compiling openzeppelin-solidity/contracts/token/ERC20/Burn
ableToken.sol
> Compiling openzeppelin-solidity/contracts/token/ERC20/Deta
iledERC20.sol
> Compiling openzeppelin-solidity/contracts/token/ERC20/ERC2
0.sol
> Compiling openzeppelin-solidity/contracts/token/ERC20/ERC2
0Basic.sol
> Compiling openzeppelin-solidity/contracts/token/ERC20/Mint
ableToken.sol
> Compiling openzeppelin-solidity/contracts/token/ERC20/Stan
dardToken.sol

    > compilation warnings encountered:

./contracts/IBlockReward.sol:6:5: Warning: Functions in inte
rfaces should be declared external.
    function mintedTotally() public view returns (uint256);
    ^-----------------------------------------------------^
,./contracts/IBlockReward.sol:7:5: Warning: Functions in int
erfaces should be declared external.
    function mintedTotallyByBridge(address _bridge) public v
iew returns(uint256);
    ^-----------------------------------------------------
--------------------^
```

```
,./contracts/IBridgeValidators.sol:5:5: Warning: Functions i
n interfaces should be declared external.
    function isValidator(address _validator) public view ret
urns(bool);
    ^-----------------------------------------------------
----------^
,./contracts/IBridgeValidators.sol:6:5: Warning: Functions i
n interfaces should be declared external.
    function requiredSignatures() public view returns(uint25
6);
    ^-------------------------------------------------------
--^
,./contracts/IBridgeValidators.sol:7:5: Warning: Functions i
n interfaces should be declared external.
    function owner() public view returns(address);
    ^-------------------------------------------^
,./contracts/IOwnedUpgradeabilityProxy.sol:5:5: Warning: Fun
ctions in interfaces should be declared external.
    function proxyOwner() public view returns (address);
    ^-------------------------------------------------^
,./contracts/IRewardableValidators.sol:5:5: Warning: Functio
ns in interfaces should be declared external.
    function isValidator(address _validator) public view ret
urns(bool);
    ^-----------------------------------------------------
----------^
,./contracts/IRewardableValidators.sol:6:5: Warning: Functio
ns in interfaces should be declared external.
    function requiredSignatures() public view returns(uint25
6);
    ^-------------------------------------------------------
--^
,./contracts/IRewardableValidators.sol:7:5: Warning: Functio
ns in interfaces should be declared external.
    function owner() public view returns(address);
    ^-------------------------------------------^
,./contracts/IRewardableValidators.sol:8:5: Warning: Functio
ns in interfaces should be declared external.
    function validatorList() public view returns (address[])
;
    ^-----------------------------------------------------
^
,./contracts/IRewardableValidators.sol:9:5: Warning: Functio
ns in interfaces should be declared external.
```

```
    function getValidatorRewardAddress(address _validator) p
ublic view returns(address);
    ^-----------------------------------------------------
---------------------------^
,./contracts/IRewardableValidators.sol:10:5: Warning: Functi
ons in interfaces should be declared external.
    function validatorCount() public view returns (uint256);
    ^-----------------------------------------------------^
,./contracts/IRewardableValidators.sol:11:5: Warning: Functi
ons in interfaces should be declared external.
    function getNextValidator(address _address) public view
returns (address);
    ^-----------------------------------------------------
-----------------^
,./contracts/upgradeability/Proxy.sol:73:35: Warning: The "r
eturndatasize" instruction is only available for Byzantium-c
ompatible VMs. You are currently compiling for "spuriousDrag
on", where it will be interpreted as an invalid instruction.
            mstore(0x40, add(ptr, returndatasize))
                                  ^------------^
,./contracts/upgradeability/Proxy.sol:80:36: Warning: The "r
eturndatasize" instruction is only available for Byzantium-c
ompatible VMs. You are currently compiling for "spuriousDrag
on", where it will be interpreted as an invalid instruction.
            returndatacopy(ptr, 0, returndatasize)
                                   ^------------^
,./contracts/upgradeability/Proxy.sol:80:13: Warning: The "r
eturndatacopy" instruction is only available for Byzantium-c
ompatible VMs. You are currently compiling for "spuriousDrag
on", where it will be interpreted as an invalid instruction.
            returndatacopy(ptr, 0, returndatasize)
            ^-----------------------------------^
,./contracts/upgradeability/Proxy.sol:88:34: Warning: The "r
eturndatasize" instruction is only available for Byzantium-c
ompatible VMs. You are currently compiling for "spuriousDrag
on", where it will be interpreted as an invalid instruction.
            case 0 { revert(ptr, returndatasize) }
                                 ^------------^
,./contracts/upgradeability/Proxy.sol:89:35: Warning: The "r
eturndatasize" instruction is only available for Byzantium-c
ompatible VMs. You are currently compiling for "spuriousDrag
on", where it will be interpreted as an invalid instruction.
            default { return(ptr, returndatasize) }
                                  ^------------^
```

```
,./contracts/upgradeable_contracts/ERC677Bridge.sol:26:51: W
arning: Unused function parameter. Remove or comment out the
variable name to silence this warning.
    function bridgeSpecificActionsOnTokenTransfer(ERC677 _to
ken, address _from, uint256 _value) internal {
                                                  ^---------
--^
,./contracts/upgradeable_contracts/erc20_to_erc20/BasicFore
ignBridgeErcToErc.sol:47:68: Warning: Unused function parame
ter. Remove or comment out the variable name to silence this
warning.
    function onExecuteMessage(address _recipient, uint256 _a
mount, bytes32 _txHash) internal returns(bool){
                                                               ^

-------------^
,./contracts/upgradeable_contracts/erc20_to_native/ForeignBr
idgeErcToNative.sol:56:68: Warning: Unused function paramete
r. Remove or comment out the variable name to silence this w
arning.
    function onExecuteMessage(address _recipient, uint256 _a
mount, bytes32 _txHash) internal returns(bool) {
                                                               ^

-------------^
,./contracts/upgradeable_contracts/native_to_erc20/HomeBridg
eNativeToErc.sol:159:34: Warning: Unused function parameter.
Remove or comment out the variable name to silence this warn
ing.
    function onFailedAffirmation(address _recipient, uint256
_value, bytes32 _txHash) internal {
                                 ^-----------------^
,./contracts/upgradeable_contracts/native_to_erc20/HomeBridg
eNativeToErc.sol:159:54: Warning: Unused function parameter.
Remove or comment out the variable name to silence this warn
ing.
    function onFailedAffirmation(address _recipient, uint256
_value, bytes32 _txHash) internal {
                                                     ^------
------^
,./contracts/upgradeable_contracts/native_to_erc20/HomeBridg
eNativeToErc.sol:159:70: Warning: Unused function parameter.
Remove or comment out the variable name to silence this warn
ing.
    function onFailedAffirmation(address _recipient, uint256
_value, bytes32 _txHash) internal {
                                                               ^
```

```
-------------^
,./contracts/upgradeable_contracts/BasicHomeBridge.sol:89:5:
Warning: Function state mutability can be restricted to pure
    function onExecuteAffirmation(address, uint256, bytes32)
internal returns(bool) {
    ^ (Relevant source part starts here and spans across mul
tiple lines).
,./contracts/upgradeable_contracts/BasicHomeBridge.sol:92:5:
Warning: Function state mutability can be restricted to pure
    function onSignaturesCollected(bytes) internal {
    ^ (Relevant source part starts here and spans across mul
tiple lines).
,./contracts/upgradeable_contracts/BasicHomeBridge.sol:160:5
: Warning: Function state mutability can be restricted to pu
re
    function affirmationWithinLimits(uint256) internal view
returns(bool) {
    ^ (Relevant source part starts here and spans across mul
tiple lines).
,./contracts/upgradeable_contracts/BasicHomeBridge.sol:164:5
: Warning: Function state mutability can be restricted to pu
re
    function onFailedAffirmation(address, uint256, bytes32)
internal {
    ^ (Relevant source part starts here and spans across mul
tiple lines).
,./contracts/upgradeable_contracts/RewardableBridge.sol:19:2
7: Warning: Function declared as view, but this expression (
potentially) modifies the state and thus requires non-payabl
e (the default) or payable.
            let result := callcode(gas, feeManager, 0x0, add
(callData, 0x20), mload(callData), 0, 32)
                          ^--------------------------------
---------------------------------------^
,./contracts/upgradeable_contracts/RewardableBridge.sol:33:2
7: Warning: Function declared as view, but this expression (
potentially) modifies the state and thus requires non-payabl
e (the default) or payable.
            let result := callcode(gas, feeManager, 0x0, add
(callData, 0x20), mload(callData), 0, 4)
                          ^--------------------------------
---------------------------------------^
,./contracts/upgradeable_contracts/RewardableBridge.sol:67:2
7: Warning: Function declared as view, but this expression (
```

```
potentially) modifies the state and thus requires non-payabl
e (the default) or payable.
            let result := callcode(gas, _impl, 0x0, add(call
Data, 0x20), mload(callData), 0, 32)
                          ^--------------------------------
---------------------------------^
,./contracts/upgradeable_contracts/erc20_to_erc20/POSDAOHome
BridgeErcToErc.sol:52:27: Warning: Function declared as view
, but this expression (potentially) modifies the state and t
hus requires non-payable (the default) or payable.
            let result := callcode(gas, feeManager, 0x0, add
(callData, 0x20), mload(callData), 0, 32)
                          ^--------------------------------
---------------------------------------^
,./contracts/upgradeable_contracts/erc20_to_native/FeeManage
rErcToNativePOSDAO.sol:17:5: Warning: Function state mutabil
ity can be restricted to pure
    function getAmountToBurn(uint256 _value) public view ret
urns(uint256) {
    ^ (Relevant source part starts here and spans across mul
tiple lines).
,./contracts/upgradeable_contracts/erc20_to_native/Rewardabl
eHomeBridgeErcToNative.sol:29:27: Warning: Function declared
as view, but this expression (potentially) modifies the stat
e and thus requires non-payable (the default) or payable.
            let result := callcode(gas, feeManager, 0x0, add
(callData, 0x20), mload(callData), 0, 32)
                          ^--------------------------------
---------------------------------------^

> Artifacts written to ./build/spuriousDragon
> Compiled successfully using:
   - solc: 0.4.24+commit.e67f0147.Emscripten.clang
```

## Tests output

```
  Contract: ERC677BridgeToken
     default values (198ms)
    #bridgeContract
       can set bridge contract (209ms)
       only owner can set bridge contract (298ms)
       fail to set invalid bridge contract address (185ms)
```

```
    #mint
       can mint by owner (125ms)
       no one can call finishMinting (47ms)
       cannot mint by non-owner (116ms)
    #transfer
       sends tokens to recipient (189ms)
       sends tokens to bridge contract (294ms)
       sends tokens to contract that does not contains onTok
enTransfer method (131ms)
       fail to send tokens to bridge contract out of limits
(369ms)
    #burn
       can burn (170ms)
    #transferAndCall
       calls contractFallback (439ms)
       sends tokens to bridge contract (290ms)
       fail to sends tokens to contract that does not contai
ns onTokenTransfer method (144ms)
       fail to send tokens to bridge contract out of limits
(312ms)
    #claimtokens
       can take send ERC20 tokens (285ms)
    #transfer
       if transfer called on contract, onTokenTransfer is al
so invoked (306ms)
       if transfer called on contract, still works even if o
nTokenTransfer doesnot exist (281ms)

  Contract: ERC677BridgeTokenRewardable
     default values (127ms)
    #bridgeContract
       can set bridge contract (150ms)
       only owner can set bridge contract (225ms)
       fail to set invalid bridge contract address (142ms)
    #blockRewardContract
       can set BlockReward contract (115ms)
       only owner can set BlockReward contract (178ms)
       fail to set invalid BlockReward contract address (131
ms)
    #stakingContract
       can set Staking contract (115ms)
       only owner can set Staking contract (173ms)
       fail to set invalid Staking contract address (183ms)
    #mintReward
```

```
        can only be called by BlockReward contract (117ms)
        should increase totalSupply and balances (238ms)
    #stake
        can only be called by Staking contract (208ms)
        should revert if user doesn't have enough balance (18
4ms)
        should decrease user's balance and increase Staking's
balance (231ms)
    #withdraw
        can only be called by Staking contract (285ms)
        should revert if Staking doesn't have enough balance
(268ms)
        should decrease Staking's balance and increase user's
balance (311ms)
    #mint
        can mint by owner (96ms)
        no one can call finishMinting (39ms)
        cannot mint by non-owner (91ms)
    #transfer
        sends tokens to recipient (158ms)
        sends tokens to bridge contract (256ms)
        sends tokens to contract that does not contains onTok
enTransfer method (126ms)
        fail to send tokens to bridge contract out of limits
(282ms)
        fail to send tokens to Staking contract directly (169
ms)
    #transferFrom
        fail to send tokens to Staking contract directly (192
ms)
    #burn
        can burn (149ms)
    #transferAndCall
        calls contractFallback (427ms)
        sends tokens to bridge contract (246ms)
        fail to sends tokens to contract that does not contai
ns onTokenTransfer method (130ms)
        fail to send tokens to bridge contract out of limits
(266ms)
    #claimtokens
        can take send ERC20 tokens (278ms)
    #transfer
        if transfer called on contract, onTokenTransfer is al
so invoked (292ms)
```

```
       if transfer called on contract, still works even if o
nTokenTransfer doesnot exist (263ms)

  Contract: RewardableValidators
    #initialize
       sets values (751ms)
    #addValidator
       adds validator (163ms)
       cannot add already existing validator (84ms)
       cannot add 0xf as validator address (41ms)
       cannot add 0x0 as validator address (44ms)
       cannot add 0x0 as reward address
    #removeValidator
       removes validator (155ms)
       cannot remove if it will break requiredSignatures (19
7ms)
       cannot remove non-existent validator (179ms)
    #setRequiredSignatures
       sets req signatures (112ms)
       cannot set more than  validators count (72ms)
    #upgradable
       can be upgraded via upgradeToAndCall (260ms)
    #single list remove
       should remove 0x627306090abaB3A6e1400e9345bC60c78a8BE
f57 - without Proxy (147ms)
       Removed validator should return zero address on nextV
alidator (194ms)
       should remove 0x627306090abaB3A6e1400e9345bC60c78a8BE
f57 - with Proxy (379ms)
       should remove 0xf17f52151EbEF6C7334FAD080c5704D77216b
732 - with Proxy (339ms)
       should remove 0xC5fdf4076b8F3A5357c5E395ab970B5B54098
Fef - with Proxy (451ms)
       should remove 0x821aEa9a577a9b44299B9c15c88cf3087F3b5
544 - with Proxy (336ms)
       should remove 0x0d1d4e623D10F9FBA5Db95830F7d3839406C6
AF2 - with Proxy (395ms)
    #reward address
       reward address is properly assigned (233ms)
    #Validators list
       should return validators list (151ms)

  Contract: BridgeValidators
    #initialize
```

```
       sets values (531ms)
    #addValidator
       adds validator (154ms)
       cannot add already existing validator (83ms)
       cannot add 0xf as validator address
       cannot add 0x0 as validator address (47ms)
    #removeValidator
       removes validator (142ms)
       cannot remove if it will break requiredSignatures (19
9ms)
       cannot remove non-existent validator (134ms)
    #setRequiredSignatures
       sets req signatures (109ms)
       cannot set more than  validators count (80ms)
    #upgradable
       can be upgraded via upgradeToAndCall (252ms)
    #single list remove
       should remove 0x627306090abaB3A6e1400e9345bC60c78a8BE
f57 - without Proxy (145ms)
       Removed validator should return zero address on nextV
alidator (196ms)
       should remove 0x627306090abaB3A6e1400e9345bC60c78a8BE
f57 - with Proxy (307ms)
       should remove 0xf17f52151EbEF6C7334FAD080c5704D77216b
732 - with Proxy (316ms)
       should remove 0xC5fdf4076b8F3A5357c5E395ab970B5B54098
Fef - with Proxy (308ms)
       should remove 0x821aEa9a577a9b44299B9c15c88cf3087F3b5
544 - with Proxy (311ms)
       should remove 0x0d1d4e623D10F9FBA5Db95830F7d3839406C6
AF2 - with Proxy (337ms)
    #Validators list
       should return validators list (134ms)

  Contract: HomeBridge_ERC20_to_ERC20
    #initialize
       sets variables (378ms)
       cant set maxPerTx > dailyLimit (124ms)
       can be deployed via upgradeToAndCall (246ms)
       cant initialize with invalid arguments (537ms)
    #fallback
       reverts
    #setting limits
       #setMaxPerTx allows to set only to owner and cannot b
```

```
e more than daily limit (127ms)
       #setMinPerTx allows to set only to owner and cannot b
e more than daily limit and should be less than maxPerTx (14
9ms)
    #executeAffirmation
       should allow validator to withdraw (253ms)
       should allow validator to withdraw with zero value (2
55ms)
       test with 2 signatures required (811ms)
       should not allow to double submit (156ms)
       should not allow non-authorities to execute deposit (
52ms)
       doesnt allow to deposit if requiredSignatures has cha
nged (825ms)
       works with 5 validators and 3 required signatures (57
3ms)
       should not allow execute affirmation over foreign max
tx limit (63ms)
       should fail if txHash already set as above of limits
(264ms)
       should not allow execute affirmation over daily forei
gn limit (311ms)
    #isAlreadyProcessed
       returns  (130ms)
    #submitSignature
       allows a validator to submit a signature (178ms)
       when enough requiredSignatures are collected, Collect
edSignatures event is emitted (368ms)
       works with 5 validators and 3 required signatures (64
7ms)
       attack when increasing requiredSignatures (488ms)
       attack when decreasing requiredSignatures (247ms)
    #requiredMessageLength
       should return the required message length
    #fixAssetsAboveLimits
       Should reduce outOfLimitAmount and not emit any event
(162ms)
       Should reduce outOfLimitAmount and emit UserRequestFo
rSignature (154ms)
       Should not be allow to be called by an already fixed
txHash (886ms)
       Should fail if txHash didnt increase out of limit amo
unt (123ms)
       Should fail if not called by proxyOwner (161ms)
```

```
    #OwnedUpgradeability
       upgradeabilityAdmin should return the proxy owner (19
6ms)
    #rewardableInitialize
       sets variables (772ms)
       can update fee contract (230ms)
       can update fee (347ms)
       should be able to get fee manager mode (154ms)
       should be able to set blockReward contract (273ms)
    #onTokenTransfer
       should trigger UserRequestForSignature with transfer
value (797ms)
       should trigger UserRequestForSignature with fee subtr
acted (590ms)
    #rewardable_submitSignatures
       should distribute fee to one validator (433ms)
       should distribute fee to 3 validators (594ms)
       should distribute fee to 5 validators (808ms)
    #rewardable_executeAffirmation
       should distribute fee to one validator (489ms)
       should distribute fee to 3 validators (530ms)
       should distribute fee to 5 validators (733ms)

  Contract: ForeignBridge_ERC20_to_ERC20
    #initialize
       should initialize (693ms)
    #executeSignatures
       should allow to executeSignatures (178ms)
       should allow second withdrawal with different transac
tionHash but same recipient and value (324ms)
       should not allow second withdraw (replay attack) with
same transactionHash but different recipient (262ms)
       should not allow withdraw over home max tx limit (138
ms)
       should not allow withdraw over daily home limit (316m
s)
    #withdraw with 2 minimum signatures
       withdraw should fail if not enough signatures are pro
vided (253ms)
       withdraw should fail if duplicate signature is provid
ed (122ms)
       works with 5 validators and 3 required signatures (46
9ms)
    #upgradeable
```

```
       can be upgraded (764ms)
       can be deployed via upgradeToAndCall (211ms)
    #claimTokens
       can send erc20 (637ms)
    #ForeignBridgeErc677ToErc677_onTokenTransfer
       can only be called from token contract (502ms)
       should not allow to transfer more than maxPerTx limit
(644ms)
       should only let to transfer within daily limit (741ms
)
       should not let to transfer less than minPerTx (501ms)

  Contract: HomeBridge
    #initialize
       sets variables (350ms)
       cant set maxPerTx > dailyLimit (126ms)
       can be deployed via upgradeToAndCall (244ms)
       cant initialize with invalid arguments (204ms)
       can transfer ownership (166ms)
    #fallback
       should accept native coins (1088ms)
       doesnt let you send more than max amount per tx (876m
s)
       should not let to deposit less than minPerTx (587ms)
    #setting limits
       #setMaxPerTx allows to set only to owner and cannot b
e more than daily limit (312ms)
       #setMinPerTx allows to set only to owner and cannot b
e more than daily limit and should be less than maxPerTx (28
7ms)
    #executeAffirmation
       should allow validator to executeAffirmation (144ms)
       should allow validator to executeAffirmation with zer
o value (160ms)
       test with 2 signatures required (757ms)
       should not allow to double submit (181ms)
       should not allow non-authorities to execute withdraw
(54ms)
       doesnt allow to withdraw if requiredSignatures has ch
anged (1080ms)
       force withdraw if the recepient has fallback to rever
t (148ms)
       works with 5 validators and 3 required signatures (55
6ms)
```

```
       should not allow execute affirmation over foreign max
tx limit (63ms)
       should not allow execute affirmation over daily forei
gn limit (288ms)
    #isAlreadyProcessed
       returns  (164ms)
    #submitSignature
       allows a validator to submit a signature (148ms)
       when enough requiredSignatures are collected, Collect
edSignatures event is emitted (412ms)
       works with 5 validators and 3 required signatures (56
4ms)
       attack when increasing requiredSignatures (573ms)
       attack when decreasing requiredSignatures (246ms)
    #requiredMessageLength
       should return the required message length
    #rewardableInitialize
       sets variables (873ms)
       can update fee contract (235ms)
       can update fee (201ms)
       should be able to get fee manager mode (153ms)
    #feeManager_OneDirection_fallback
       should not subtract fee from value (404ms)
    #feeManager_OneDirection_submitSignature
       should not distribute fee to validator (487ms)
    #feeManager_OneDirection_ExecuteAffirmation
       should distribute fee to validator (477ms)
       should distribute fee to 3 validators (679ms)
       should distribute fee to 5 validators (845ms)
    #feeManager_BothDirections_fallback
       should subtract fee from value (387ms)
    #feeManager_BothDirections_submitSignature
       should distribute fee to validator (563ms)
       should distribute fee to 3 validators (763ms)
       should distribute fee to 5 validators (990ms)
    #feeManager_BothDirections_ExecuteAffirmation
       should distribute fee to validator (623ms)
       should distribute fee to 3 validators (692ms)
       should distribute fee to 5 validators (946ms)

  Contract: ForeignBridge
    #initialize
       should initialize (914ms)
    #executeSignatures
```

```
       should allow to deposit (192ms)
       should reject if address is not foreign address (123m
s)
       should allow second deposit with different transactio
nHash but same recipient and value (306ms)
       should not allow second deposit (replay attack) with
same transactionHash but different recipient (226ms)
       should not allow withdraw over home max tx limit (96m
s)
       should not allow withdraw over daily home limit (357m
s)
    #executeSignatures with 2 minimum signatures
       deposit should fail if not enough signatures are prov
ided (194ms)
       deposit should fail if duplicate signature is provide
d (145ms)
       works with 5 validators and 3 required signatures (56
2ms)
    #onTokenTransfer
       can only be called from token contract (421ms)
       should not allow to burn more than the limit (490ms)
       should only let to send within maxPerTx limit (694ms)
       should not let to withdraw less than minPerTx (471ms)
    #setting limits
       #setMaxPerTx allows to set only to owner and cannot b
e more than daily limit (133ms)
       #setMinPerTx allows to set only to owner and cannot b
e more than daily limit and should be less than maxPerTx (12
8ms)
    #upgradeable
       can be upgraded (848ms)
       can be deployed via upgradeToAndCall (306ms)
       can transfer ownership (262ms)
    #claimTokens
       can send erc20 (638ms)
       also calls claimTokens on tokenAddress (605ms)
    #rewardableInitialize
       sets variables (801ms)
       can update fee contract (202ms)
       can update fee (180ms)
       should be able to get fee manager mode (309ms)
    #RewardableBridge_executeSignatures
       should distribute fee to validator (546ms)
       should distribute fee to 3 validators (592ms)
```

```
       should distribute fee to 5 validators (1020ms)

  Contract: ForeignBridge_ERC20_to_Native
    #initialize
       should initialize (1552ms)
    #executeSignatures
       should allow to executeSignatures (179ms)
       should allow second withdrawal with different transac
tionHash but same recipient and value (335ms)
       should not allow second withdraw (replay attack) with
same transactionHash but different recipient (280ms)
       should not allow withdraw over home max tx limit (146
ms)
       should not allow withdraw over daily home limit (327m
s)
    #withdraw with 2 minimum signatures
       withdraw should fail if not enough signatures are pro
vided (192ms)
       withdraw should fail if duplicate signature is provid
ed (129ms)
       works with 5 validators and 3 required signatures (49
8ms)
    #upgradeable
       can be upgraded (661ms)
       can be deployed via upgradeToAndCall (279ms)
    #claimTokens
       can send erc20 (587ms)

  Contract: HomeBridge_ERC20_to_Native
    #initialize
       sets variables (518ms)
       can update block reward contract (415ms)
       cant set maxPerTx > dailyLimit (124ms)
       can be deployed via upgradeToAndCall (418ms)
       can be upgraded keeping the state (622ms)
       cant initialize with invalid arguments (311ms)
    #rewardableInitialize
       sets variables (573ms)
       cant initialize with invalid arguments (456ms)
       can update fee contract (178ms)
       can update fee (225ms)
    #fallback
       should accept native coins (193ms)
       should accumulate burnt coins (297ms)
```

```
       doesnt let you send more than daily limit (403ms)
       doesnt let you send more than max amount per tx (434m
s)
       should not let to deposit less than minPerTx (249ms)
       should fail if not enough bridged tokens (322ms)
    #setting limits
       setMaxPerTx allows to set only to owner and cannot be
more than daily limit (144ms)
       setMinPerTx allows to set only to owner and cannot be
more than daily limit and should be less than maxPerTx (153m
s)
       setExecutionMaxPerTx allows to set only to owner and
cannot be more than execution daily limit (169ms)
       executionDailyLimit allows to set only to owner (123m
s)
    #executeAffirmation
       should allow validator to executeAffirmation (111ms)
       should allow validator to executeAffirmation with zer
o value (115ms)
       test with 2 signatures required (617ms)
       should not allow non-validator to execute affirmation
(50ms)
       should fail if the block reward contract is not set (
246ms)
       works with 5 validators and 3 required signatures (49
5ms)
       should not allow execute affirmation over foreign max
tx limit (65ms)
       should fail if txHash already set as above of limits
(198ms)
       should not allow execute affirmation over daily forei
gn limit (352ms)
    #submitSignature
       allows a validator to submit a signature (169ms)
       when enough requiredSignatures are collected, Collect
edSignatures event is emitted (336ms)
       works with 5 validators and 3 required signatures (90
0ms)
       attack when increasing requiredSignatures (518ms)
       attack when decreasing requiredSignatures (252ms)
    #requiredMessageLength
       should return the required message length
    #fixAssetsAboveLimits
       Should reduce outOfLimitAmount and not emit any event
```

```
(153ms)
       Should reduce outOfLimitAmount and emit UserRequestFo
rSignature (162ms)
       Should not be allow to be called by an already fixed
txHash (375ms)
       Should fail if txHash didnt increase out of limit amo
unt (941ms)
       Should fail if not called by proxyOwner (161ms)
    #OwnedUpgradeability
       upgradeabilityAdmin should return the proxy owner (18
6ms)
    #feeManager
       should be able to set and get fee manager contract (1
03ms)
       should be able to set and get fees (282ms)
       should be able to get fee manager mode (105ms)
    #feeManager_ExecuteAffirmation
       should distribute fee to validator (635ms)
       should distribute fee to 3 validators (960ms)
       should distribute fee to 5 validators (1554ms)
    #feeManager_fallback
       should subtract fee from value (258ms)
    #feeManager_submitSignature
       should distribute fee to validator (760ms)
       should distribute fee to 3 validators (1006ms)
       should distribute fee to 5 validators (2376ms)
    #FeeManager_random
       should return value between 0 and 3 (526ms)
    #feeManager_ExecuteAffirmation_POSDAO
       should distribute fee to validator (761ms)
       should distribute fee to 3 validators (841ms)
       should distribute fee to 5 validators (1186ms)
    #feeManager_fallback_POSDAO
       should subtract fee from value (251ms)
    #feeManager_submitSignature_POSDAO
       should distribute fee to validator (815ms)
       should distribute fee to 3 validators (1052ms)
       should distribute fee to 5 validators (1281ms)


  295 passing (3m)
```

## Solhint output

```
contracts/libraries/Message.sol
  57:9  warning  Avoid to use inline assembly. It is accepta
ble only in rare cases  no-inline-assembly
  79:9  warning  Avoid to use inline assembly. It is accepta
ble only in rare cases  no-inline-assembly

contracts/libraries/SafeMath.sol
  13:3  error  Expected indentation of 4 spaces but found 2
indent
  14:5  error  Expected indentation of 8 spaces but found 4
indent
  15:7  error  Expected indentation of 12 spaces but found 6
indent
  16:5  error  Expected indentation of 8 spaces but found 4
indent
  17:5  error  Expected indentation of 8 spaces but found 4
indent
  18:5  error  Expected indentation of 8 spaces but found 4
indent
  19:5  error  Expected indentation of 8 spaces but found 4
indent
  20:3  error  Expected indentation of 4 spaces but found 2
indent
  25:3  error  Expected indentation of 4 spaces but found 2
indent
  27:5  error  Expected indentation of 8 spaces but found 4
indent
  29:5  error  Expected indentation of 8 spaces but found 4
indent
  30:3  error  Expected indentation of 4 spaces but found 2
indent
  35:3  error  Expected indentation of 4 spaces but found 2
indent
  36:5  error  Expected indentation of 8 spaces but found 4
indent
  37:5  error  Expected indentation of 8 spaces but found 4
indent
  38:3  error  Expected indentation of 4 spaces but found 2
indent
  43:3  error  Expected indentation of 4 spaces but found 2
indent
  44:5  error  Expected indentation of 8 spaces but found 4
indent
```

```
  45:5  error  Expected indentation of 8 spaces but found 4
indent
  46:5  error  Expected indentation of 8 spaces but found 4
indent
  47:3  error  Expected indentation of 4 spaces but found 2
indent

contracts/test/BlockReward.sol
  95:17  warning  Avoid to use low level calls  avoid-low-le
vel-calls
  95:9   warning  Avoid to use low level calls  avoid-low-le
vel-calls

contracts/upgradeability/ClassicEternalStorageProxy.sol
   9:5  warning  Fallback function must be simple
o-complex-fallback
  13:9  warning  Avoid to use inline assembly. It is accepta
ble only in rare cases  no-inline-assembly

contracts/upgradeability/OwnedUpgradeabilityProxy.sol
  71:17  warning  Avoid to use ".call.value()()"  avoid-call
-value

contracts/upgradeability/Proxy.sol
  20:5  warning  Fallback function must be simple
o-complex-fallback
  23:9  warning  Avoid to use inline assembly. It is accepta
ble only in rare cases  no-inline-assembly
  78:2  error    Line length must be no more than 120 but cu
rrent length is 163     max-line-length

contracts/upgradeable_contracts/BasicBridge.sol
   84:16  warning  Avoid to make time-based decisions in you
r business logic          not-rely-on-time
  146:9   warning  Avoid to use inline assembly. It is accep
table only in rare cases  no-inline-assembly

contracts/upgradeable_contracts/BasicHomeBridge.sol
  17:2  error  Line length must be no more than 120 but curr
ent length is 126  max-line-length
  29:2  error  Line length must be no more than 120 but curr
ent length is 133  max-line-length
  59:2  error  Line length must be no more than 120 but curr
ent length is 129  max-line-length
```

```
contracts/upgradeable_contracts/RewardableBridge.sol
  18:9   warning  Avoid to use inline assembly. It is accept
able only in rare cases  no-inline-assembly
  32:9   warning  Avoid to use inline assembly. It is accept
able only in rare cases  no-inline-assembly
  53:17  warning  Avoid to use low level calls
void-low-level-calls
  53:9   warning  Avoid to use low level calls
void-low-level-calls
  59:9   warning  Avoid to use inline assembly. It is accept
able only in rare cases  no-inline-assembly
  63:2   error    Line length must be no more than 120 but c
urrent length is 122     max-line-length
  65:2   error    Line length must be no more than 120 but c
urrent length is 122     max-line-length
  66:9   warning  Avoid to use inline assembly. It is accept
able only in rare cases  no-inline-assembly
  77:17  warning  Avoid to use low level calls
void-low-level-calls
  77:9   warning  Avoid to use low level calls
void-low-level-calls
  82:17  warning  Avoid to use low level calls
void-low-level-calls
  82:9   warning  Avoid to use low level calls
void-low-level-calls

contracts/upgradeable_contracts/ValidatorsFeeManager.sol
   8:2  error  Line length must be no more than 120 but curr
ent length is 125  max-line-length
  10:2  error  Line length must be no more than 120 but curr
ent length is 131  max-line-length

 50 problems (29 errors, 21 warnings)
```

## Solium output

```
contracts/ERC677BridgeToken.sol
  24:52    warning    Code contains empty block
o-empty-blocks
  26:66    warning    Visibility modifier "public" should co
me before other modifiers.    visibility-first
  27:8     warning    Provide an error message for require()
```

```
.                              error-reason
  32:8     warning    Provide an error message for require()
.                              error-reason
  39:8     warning    Provide an error message for require()
.                              error-reason
  43:12    warning    Provide an error message for require()
.                              error-reason
  59:8     warning    Provide an error message for require()
.                              error-reason
  62:16    warning    Provide an error message for revert().
error-reason
  74:19    warning    Avoid using low-level function 'call'.
security/no-low-level-calls
  83:8     error      Avoid using Inline Assembly.
ecurity/no-inline-assembly
  88:8     warning    Provide an error message for revert().
error-reason
  92:8     warning    Provide an error message for revert().
error-reason
  96:8     warning    Provide an error message for require()
.                              error-reason
  104:8    warning    Provide an error message for require()
.                              error-reason

contracts/ERC677BridgeTokenRewardable.sol
  15:58    warning    Code contains empty block
o-empty-blocks
  17:76    warning    Visibility modifier "public" should co
me before other modifiers.    visibility-first
  18:8     warning    Provide an error message for require()
.                              error-reason
  22:68    warning    Visibility modifier "public" should co
me before other modifiers.    visibility-first
  23:8     warning    Provide an error message for require()
.                              error-reason
  28:8     warning    Provide an error message for require()
.                              error-reason
  33:8     warning    Provide an error message for require()
.                              error-reason
  55:8     warning    Provide an error message for require()
.                              error-reason
  63:8     warning    Provide an error message for require()
.                              error-reason
  70:8     warning    Provide an error message for require()
.                              error-reason
```

```
  75:8     warning    Provide an error message for require()
.                              error-reason

contracts/ERC677Receiver.sol
  5:2    error    Only use indent of 4 spaces.    indentatio
n

contracts/libraries/Message.sol
  56:8      warning    Provide an error message for require(
).    error-reason
  57:8      error      Avoid using Inline Assembly.                s
ecurity/no-inline-assembly
  74:8      warning    Provide an error message for require(
).    error-reason
  100:8     warning    Provide an error message for require(
).    error-reason
  102:8     warning    Provide an error message for require(
).    error-reason
  108:12    warning    Provide an error message for require(
).    error-reason
  110:16    warning    Provide an error message for revert()
.     error-reason

contracts/libraries/SafeMath.sol
  13:2    error    Only use indent of 4 spaces.    indentati
on
  15:6    error    Only use indent of 8 spaces.    indentati
on
  20:0    error    Only use indent of 4 spaces.    indentati
on
  25:2    error    Only use indent of 4 spaces.    indentati
on
  30:0    error    Only use indent of 4 spaces.    indentati
on
  35:2    error    Only use indent of 4 spaces.    indentati
on
  38:0    error    Only use indent of 4 spaces.    indentati
on
  43:2    error    Only use indent of 4 spaces.    indentati
on
  47:0    error    Only use indent of 4 spaces.    indentati
on

contracts/test/BlockReward.sol
```

```
  25:8     warning    Provide an error message for require()
.    error-reason
  26:8     warning    Provide an error message for require()
.    error-reason
  95:8     warning    Provide an error message for require()
.    error-reason
  95:22    warning    Avoid using low-level function 'call'.
security/no-low-level-calls

contracts/test/Staking.sol
  5:25    warning    Code contains empty block    no-empty-b
locks

contracts/upgradeability/ClassicEternalStorageProxy.sol
  9:24    warning    Visibility modifier "public" should com
e before other modifiers.    visibility-first
  11:8    warning    Provide an error message for require().
error-reason
  13:8    error      Avoid using Inline Assembly.
ecurity/no-inline-assembly

contracts/upgradeability/EternalStorageProxy.sol
  13:73    warning    Code contains empty block    no-empty-
blocks

contracts/upgradeability/OwnedUpgradeabilityProxy.sol
  30:8     warning    Provide an error message for require()
.                           error-reason
  47:8     warning    Provide an error message for require()
.                           error-reason
  69:91    warning    Visibility modifier "public" should co
me before other modifiers.    visibility-first
  71:8     warning    Provide an error message for require()
.                           error-reason
  71:30    error      Consider using 'transfer' in place of
'call.value()'.               security/no-call-value

contracts/upgradeability/Proxy.sol
  20:24    warning    Visibility modifier "public" should co
me before other modifiers.    visibility-first
  22:8     warning    Provide an error message for require()
.                           error-reason
  23:8     error      Avoid using Inline Assembly.
ecurity/no-inline-assembly
```

```
contracts/upgradeability/UpgradeabilityProxy.sol
  25:8    warning    Provide an error message for require().
error-reason
  26:8    warning    Provide an error message for require().
error-reason

contracts/upgradeable_contracts/BaseBridgeValidators.sol
  21:8    warning    Provide an error message for require()
.                                                 error-reason
  22:8    warning    Provide an error message for require()
.                                                 error-reason
  36:8    error      There should be no whitespace between
"address" and the opening square bracket.    array-declarati
ons
  39:8    warning    Provide an error message for require()
.                                                 error-reason
  47:16   warning    Provide an error message for revert().
error-reason
  55:8    warning    Provide an error message for require()
.                                                 error-reason
  56:8    warning    Provide an error message for require()
.                                                 error-reason
  59:8    warning    Provide an error message for require()
.                                                 error-reason
  66:8    warning    Provide an error message for require()
.                                                 error-reason
  67:8    warning    Provide an error message for require()
.                                                 error-reason
  71:8    warning    Provide an error message for require()
.                                                 error-reason
  78:16   warning    Provide an error message for revert().
error-reason

contracts/upgradeable_contracts/BasicBridge.sol
  24:8    warning    Provide an error message for require()
.           error-reason
  34:8    warning    Provide an error message for require()
.           error-reason
  84:15   warning    Avoid using 'now' (alias to 'block.tim
estamp').    security/no-block-members
  106:8   warning    Provide an error message for require()
.           error-reason
  111:8   warning    Provide an error message for require()
.           error-reason
```

```
  116:8    warning    Provide an error message for require()
.            error-reason
  131:8    warning    Provide an error message for require()
.            error-reason
  139:8    warning    Provide an error message for require()
.            error-reason
  146:8    error      Avoid using Inline Assembly.                    s
ecurity/no-inline-assembly

contracts/upgradeable_contracts/BasicForeignBridge.sol
  21:12    warning    Provide an error message for require()
.    error-reason
  22:12    warning    Provide an error message for require()
.    error-reason
  24:12    warning    Provide an error message for require()
.    error-reason

contracts/upgradeable_contracts/BasicHomeBridge.sol
  24:12     warning    Provide an error message for require(
).    error-reason
  28:12     warning    Provide an error message for require(
).    error-reason
  41:20     warning    Provide an error message for require(
).    error-reason
  52:8      warning    Provide an error message for require(
).    error-reason
  53:8      warning    Provide an error message for require(
).    error-reason
  58:8      warning    Provide an error message for require(
).    error-reason
  63:12     warning    Provide an error message for require(
).    error-reason
  89:84     warning    Code contains empty block                   n
o-empty-blocks
  92:51     warning    Code contains empty block                   n
o-empty-blocks
  164:69    warning    Code contains empty block                   n
o-empty-blocks

contracts/upgradeable_contracts/BridgeValidators.sol
  16:8     warning    Provide an error message for require()
.    error-reason
  17:8     warning    Provide an error message for require()
.    error-reason
```

```
  19:8     warning    Provide an error message for require()
.    error-reason
  20:8     warning    Provide an error message for require()
.    error-reason
  23:12    warning    Provide an error message for require()
.    error-reason
  24:12    warning    Provide an error message for require()
.    error-reason

contracts/upgradeable_contracts/ERC677Bridge.sol
  13:8    warning    Provide an error message for require().
error-reason
  19:8    warning    Provide an error message for require().
error-reason
  20:8    warning    Provide an error message for require().
error-reason

contracts/upgradeable_contracts/OverdrawManagement.sol
  14:8    warning    Provide an error message for require().
error-reason
  18:8    warning    Provide an error message for require().
error-reason

contracts/upgradeable_contracts/Ownable.sol
  22:8    warning    Provide an error message for require().
error-reason
  39:8    warning    Provide an error message for require().
error-reason

contracts/upgradeable_contracts/OwnedUpgradeability.sol
  14:8    warning    Provide an error message for require().
error-reason

contracts/upgradeable_contracts/RewardableBridge.sol
  18:8     error      Avoid using Inline Assembly.                    s
ecurity/no-inline-assembly
  32:8     error      Avoid using Inline Assembly.                    s
ecurity/no-inline-assembly
  47:8     warning    Provide an error message for require()
.           error-reason
  53:8     warning    Provide an error message for require()
.           error-reason
  53:28    warning    Avoid using low-level function 'delega
tecall'.    security/no-low-level-calls
```

```
  59:8     error      Avoid using Inline Assembly.                    s
ecurity/no-inline-assembly
  66:8     error      Avoid using Inline Assembly.                    s
ecurity/no-inline-assembly
  77:8     warning    Provide an error message for require()
.           error-reason
  77:28    warning    Avoid using low-level function 'delega
tecall'.    security/no-low-level-calls
  82:8     warning    Provide an error message for require()
.           error-reason
  82:28    warning    Avoid using low-level function 'delega
tecall'.    security/no-low-level-calls

contracts/upgradeable_contracts/RewardableValidators.sol
  17:8     warning    Provide an error message for require()
.    error-reason
  18:8     warning    Provide an error message for require()
.    error-reason
  20:8     warning    Provide an error message for require()
.    error-reason
  21:8     warning    Provide an error message for require()
.    error-reason
  22:8     warning    Provide an error message for require()
.    error-reason
  25:12    warning    Provide an error message for require()
.    error-reason
  26:12    warning    Provide an error message for require()
.    error-reason
  27:12    warning    Provide an error message for require()
.    error-reason
  55:8     warning    Provide an error message for require()
.    error-reason

contracts/upgradeable_contracts/Validatable.sol
  12:8    warning    Provide an error message for require().
error-reason

contracts/upgradeable_contracts/ValidatorsFeeManager.sol
  38:8     warning    Provide an error message for require()
.    error-reason
  51:12    warning    Provide an error message for require()
.    error-reason

contracts/upgradeable_contracts/erc20_to_erc20/BasicForeignB
```

```
ridgeErcToErc.sol
  20:8    warning    Provide an error message for require().
error-reason
  21:8    warning    Provide an error message for require().
error-reason
  22:8    warning    Provide an error message for require().
error-reason
  23:8    warning    Provide an error message for require().
error-reason
  24:8    warning    Provide an error message for require().
error-reason
  25:8    warning    Provide an error message for require().
error-reason
  43:8    warning    Provide an error message for require().
error-reason
  57:8    warning    Provide an error message for revert().
error-reason

contracts/upgradeable_contracts/erc20_to_erc20/FeeManagerErc
ToErcPOSDAO.sol
  16:8    warning    Provide an error message for require().
error-reason
  28:8    error      Avoid using Inline Assembly.                s
ecurity/no-inline-assembly

contracts/upgradeable_contracts/erc20_to_erc20/ForeignBridge
Erc677ToErc677.sol
  24:8    warning    Provide an error message for require().
error-reason

contracts/upgradeable_contracts/erc20_to_erc20/ForeignBridge
ErcToErc.sol
  36:8    warning    Provide an error message for require().
error-reason

contracts/upgradeable_contracts/erc20_to_erc20/HomeBridgeErc
ToErc.sol
  14:0      warning    Line exceeds the limit of 145 charact
ers                           max-len
  115:8     warning    Provide an error message for require(
).                            error-reason
  135:8     warning    Provide an error message for require(
).                            error-reason
  136:8     warning    Provide an error message for require(
```

```
).                                  error-reason
  137:8     warning    Provide an error message for require(
).                                  error-reason
  138:8     warning    Provide an error message for require(
).                                  error-reason
  139:8     warning    Provide an error message for require(
).                                  error-reason
  140:8     warning    Provide an error message for require(
).                                  error-reason
  141:8     warning    Provide an error message for require(
).                                  error-reason
  159:24    warning    Visibility modifier "public" should c
ome before other modifiers.    visibility-first
  160:8     warning    Provide an error message for revert()
.                                   error-reason
  206:8     warning    Provide an error message for require(
).                                  error-reason

contracts/upgradeable_contracts/erc20_to_erc20/POSDAOHomeBri
dgeErcToErc.sol
  51:8     error      Avoid using Inline Assembly.                    s
ecurity/no-inline-assembly
  68:8     warning    Provide an error message for require()
.           error-reason
  68:28    warning    Avoid using low-level function 'delega
tecall'.    security/no-low-level-calls

contracts/upgradeable_contracts/erc20_to_native/FeeManagerEr
cToNative.sol
  24:13    warning    Consider using 'transfer' in place of
'send'.    security/no-send

contracts/upgradeable_contracts/erc20_to_native/ForeignBridg
eErcToNative.sol
  24:8    warning    Provide an error message for require().
error-reason
  25:8    warning    Provide an error message for require().
error-reason
  26:8    warning    Provide an error message for require().
error-reason
  27:8    warning    Provide an error message for require().
error-reason
  28:8    warning    Provide an error message for require().
error-reason
```

```
  29:8    warning    Provide an error message for require().
error-reason
  48:8    warning    Provide an error message for require().
error-reason
  62:8    warning    Provide an error message for require().
error-reason
  71:8    warning    Provide an error message for revert().
error-reason

contracts/upgradeable_contracts/erc20_to_native/HomeBridgeEr
cToNative.sol
  19:8     warning    Provide an error message for require()
.    error-reason
  20:8     warning    Provide an error message for require()
.    error-reason
  21:8     warning    Provide an error message for require()
.    error-reason
  24:8     warning    Provide an error message for require()
.    error-reason
  97:8     warning    Provide an error message for require()
.    error-reason
  119:8    warning    Provide an error message for require()
.    error-reason
  136:8    warning    Provide an error message for require()
.    error-reason
  137:8    warning    Provide an error message for require()
.    error-reason
  138:8    warning    Provide an error message for require()
.    error-reason
  139:8    warning    Provide an error message for require()
.    error-reason
  140:8    warning    Provide an error message for require()
.    error-reason
  141:8    warning    Provide an error message for require()
.    error-reason
  142:8    warning    Provide an error message for require()
.    error-reason
  159:8    warning    Provide an error message for require()
.    error-reason
  201:8    warning    Provide an error message for require()
.    error-reason

contracts/upgradeable_contracts/erc20_to_native/RewardableHo
meBridgeErcToNative.sol
```

```
  28:8    error    Avoid using Inline Assembly.    security/
no-inline-assembly

contracts/upgradeable_contracts/native_to_erc20/FeeManagerNa
tiveToErc.sol
  19:13    warning    Consider using 'transfer' in place of
'send'.    security/no-send

contracts/upgradeable_contracts/native_to_erc20/FeeManagerNa
tiveToErcBothDirections.sol
  22:13    warning    Consider using 'transfer' in place of
'send'.    security/no-send

contracts/upgradeable_contracts/native_to_erc20/ForeignBridg
eNativeToErc.sol
  12:0     warning    Line exceeds the limit of 145 characte
rs    max-len
  71:8     warning    Provide an error message for require()
.     error-reason
  98:8     warning    Provide an error message for require()
.     error-reason
  99:8     warning    Provide an error message for require()
.     error-reason
  100:8    warning    Provide an error message for require()
.     error-reason
  101:8    warning    Provide an error message for require()
.     error-reason
  102:8    warning    Provide an error message for require()
.     error-reason
  103:8    warning    Provide an error message for require()
.     error-reason
  140:8    warning    Provide an error message for revert().
error-reason

contracts/upgradeable_contracts/native_to_erc20/HomeBridgeNa
tiveToErc.sol
  14:8      warning    Provide an error message for require(
).          error-reason
  15:8      warning    Provide an error message for require(
).          error-reason
  16:8      warning    Provide an error message for require(
).          error-reason
  80:8      warning    Provide an error message for require(
).          error-reason
```

```
  104:8     warning    Provide an error message for require(
).          error-reason
  105:8     warning    Provide an error message for require(
).          error-reason
  106:8     warning    Provide an error message for require(
).          error-reason
  107:8     warning    Provide an error message for require(
).          error-reason
  108:8     warning    Provide an error message for require(
).          error-reason
  109:8     warning    Provide an error message for require(
).          error-reason
  110:8     warning    Provide an error message for require(
).          error-reason
  149:13    warning    Consider using 'transfer' in place of
'send'.    security/no-send
  160:8     warning    Provide an error message for revert()
.           error-reason

 24 errors, 185 warnings found.
```