



**SOLIDProof**  
*Bring trust into your projects*

**Blockchain Security | Smart Contract Audits | KYC**

MADE IN GERMANY

**v1.0: 13. December, 2021**

**v1.1: 07. January, 2022**

# Audit

**Security Assessment**  
**17. January, 2022**

**For**



**BaksDAO**

Disclaimer	3
Description	5
Project Engagement	5
Logo	5
Contract Link	5
Methodology	7
Used Code from other Frameworks/Smart Contracts (direct imports)	8
Tested Contract Files	9
Source Lines	12
Risk Level	13
Capabilities	14
Scope of Work	16
Inheritance Graph	16
Verify Claims	17
Modifiers	25
CallGraph	29
Source Units in Scope	30
Critical issues	33
High issues	33
Medium issues	33
Low issues	33
Informational issues	33
Audit Comments	33
BaksDAO Test Protocol	35
SWC Attacks	37

# Disclaimer

SolidProof.io reports are not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. These reports are not, nor should be considered, an indication of the economics or value of any “product” or “asset” created by any team. SolidProof.io do not cover testing or auditing the integration with external contract or services (such as Uniswap, Uniswap, PancakeSwap etc’...)

**SolidProof.io Audits do not provide any warranty or guarantee regarding the absolute bug- free nature of the technology analyzed, nor do they provide any indication of the technology proprietors. SolidProof Audits should not be used in any way to make decisions around investment or involvement with any particular project. These reports in no way provide investment advice, nor should be leveraged as investment advice of any sort.**

SolidProof.io Reports represent an extensive auditing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology. Blockchain technology and cryptographic assets present a high level of ongoing risk. SolidProof’s position is that each company and individual are responsible for their own due diligence and continuous security. SolidProof in no way claims any guarantee of security or functionality of the technology we agree to analyze.

Version	Date	Description
1.0	13. December 2021	<ul style="list-style-type: none"><li>• Layout project</li><li>• Automated- /Manual-Security Testing</li><li>• Summary</li></ul>
1.1	07. January 2022	<ul style="list-style-type: none"><li>• Reaudit</li></ul>
1.2	17. January 2022	<ul style="list-style-type: none"><li>• Bugs were fixed</li></ul>

## **Network**

Binance Smart Chain (BEP20)

## **Website**

<https://baksdao.com/>

## **Telegram**

[https://t.me/BaksDAO\\_INT\\_COMMUNITY](https://t.me/BaksDAO_INT_COMMUNITY)

## **Twitter**

<https://twitter.com/BaksDAO>

## **Facebook**

<https://www.facebook.com/BaksDAOint/>

## **Medium**

<https://medium.com/@BaksDAO>

## **Github**

<https://github.com/BaksDAO/contracts-mirror/tree/master/contracts>

## Description

A platform to get more from your crypto assets

## Project Engagement

During the 7th of December 2021, **BaksDAO Team** engaged Solidproof.io to audit smart contracts that they created. The engagement was technical in nature and focused on identifying security flaws in the design and implementation of the contracts. They provided Solidproof.io with access to their code repository and whitepaper.

## Logo



# BaksDAO

## Contract Link

### v1.1

- Github
  - <https://github.com/BaksDAO/contracts-mirror/tree/master/contracts>
  - Commit: c4893f8f2f72b383950bd5539a50008937c83a73

### V1.2

- Github
  - <https://github.com/BaksDAO/contracts-mirror/tree/master/contracts>
  - Commit: 0e1efa0551e1593e0e02bc503c6f05cad3c606a4

# Vulnerability & Risk Level

Risk represents the probability that a certain source-threat will exploit vulnerability, and the impact of that event on the organization or system. Risk Level is computed based on CVSS version 3.0.

Level	Value	Vulnerability	Risk (Required Action)
<b>Critical</b>	9 - 10	A vulnerability that can disrupt the contract functioning in a number of scenarios, or creates a risk that the contract may be broken.	Immediate action to reduce risk level.
<b>High</b>	7 – 8.9	A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.	Implementation of corrective actions as soon as possible.
<b>Medium</b>	4 – 6.9	A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.	Implementation of corrective actions in a certain period.
<b>Low</b>	2 – 3.9	A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.	Implementation of certain corrective actions or accepting the risk.
<b>Informational</b>	0 – 1.9	A vulnerability that have informational character but is not effecting any of the code.	An observation that does not determine a level of risk

# Auditing Strategy and Techniques Applied

Throughout the review process, care was taken to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices. To do so, reviewed line-by-line by our team of expert pentesters and smart contract developers, documenting any issues as there were discovered.

## Methodology

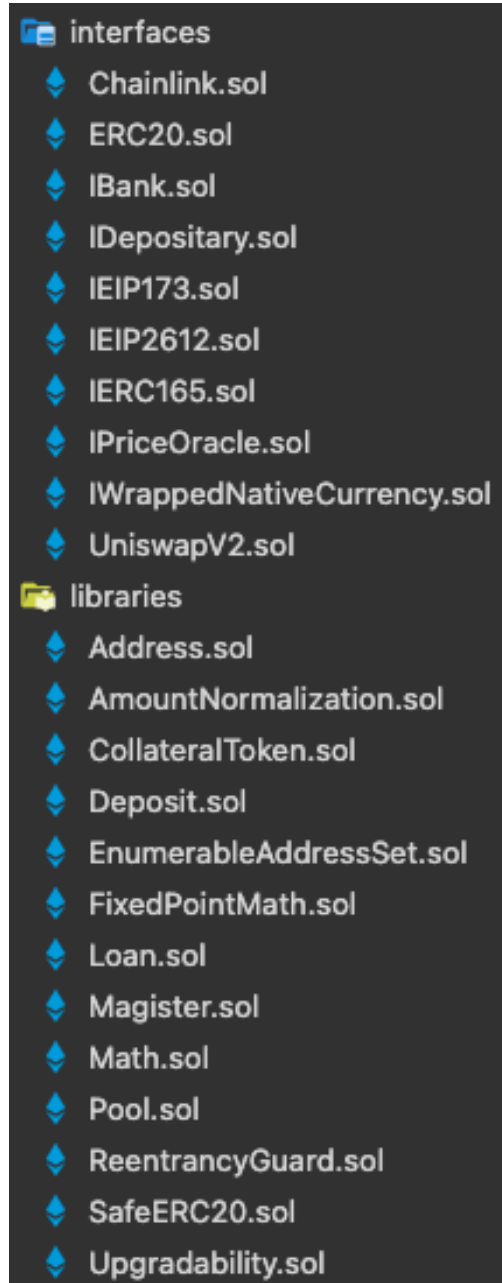
The auditing process follows a routine series of steps:

1. Code review that includes the following:
  - i) Review of the specifications, sources, and instructions provided to SolidProof to make sure we understand the size, scope, and functionality of the smart contract.
  - ii) Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
  - iii) Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to SolidProof describe.
2. Testing and automated analysis that includes the following:
  - i) Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run those test cases.
  - ii) Symbolic execution, which is analysing a program to determine what inputs causes each part of a program to execute.
3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarify, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.
4. Specific, itemized, actionable recommendations to help you take steps to secure your smart contracts.

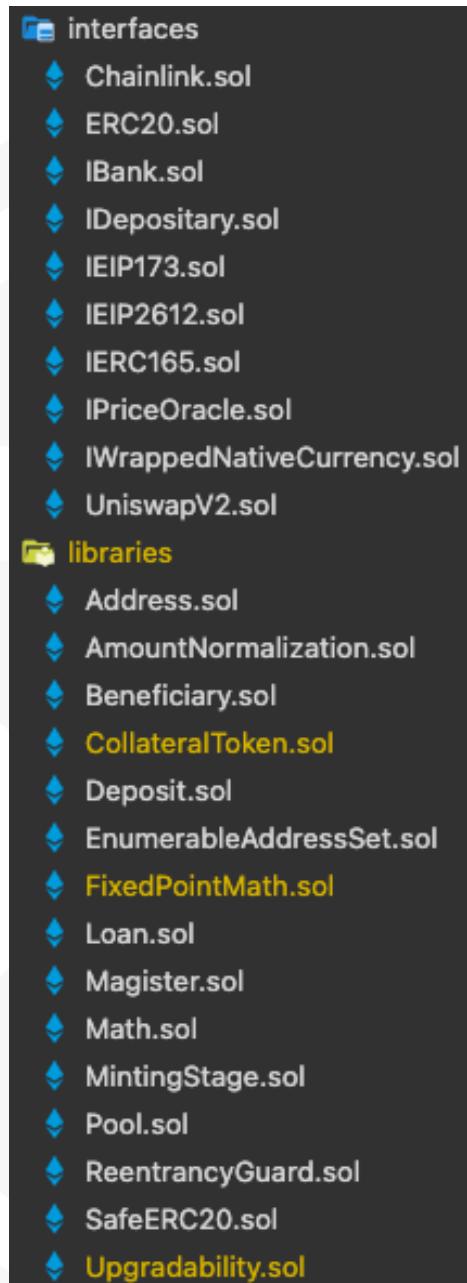
## Used Code from other Frameworks/Smart Contracts (direct imports)

Imported packages:

v1.0



v1.1





# Tested Contract Files

This audit covered the following files listed below with a SHA-1 Hash.

*A file with a different Hash has been modified, intentionally or otherwise, after the security review. A different Hash could be (but not necessarily) an indication of a changed condition or potential vulnerability that was not within the scope of this review.*

## v1.0

File Name	SHA-1 Hash
contracts/Bank.sol	119f8c95cfa5f2223e1a3095b2b26f983766c953
contracts/Core.sol	0888859f05b26d317c9089d969c1255456c33364
contracts/Voice.sol	c3505175b48d6500501a38cff4c33c37a92f622a
contracts/DevelopmentFund.sol	22b8a55f5d6acc42060e39d56ca1f093055f8b0f
contracts/ExchangeFund.sol	a0b70c7c28e582fd38c35b135ba39e44f55213be
contracts/Baks.sol	c854695fcb334ed778d875c75d3eab84a336109
contracts/BaseToken.sol	79def02ed6512525877953e116c6fc9c47fe4535
contracts/ChainlinkPriceOracle.sol	91277f57ed3cfe0089e5243f1e910c5ada28cd0e
contracts/Depository.sol	58550293d1389c074e056f573a07dc1706ffb66b
contracts/Governance.sol	b71e75981d752afdc3262afe4114e3fada84857a
contracts/libraries/ReentrancyGuard.sol	918902edcfb99b9e5966853f94cc84d6a522aa1c
contracts/interfaces/Chainlink.sol	c247bda0380df398860201f6225360a324f58b7a
contracts/libraries/Deposit.sol	a77d79889a7bd6294d3a8d041a0f645082eff996
contracts/interfaces/IERC165.sol	754045c4544e2677fbff8e619bbb6dd4f4ab0348
contracts/interfaces/EIP173.sol	d23be034f71cb447a9b3eb17ffedaa19013d3cb7
contracts/interfaces/UniswapV2.sol	3db416095fd0b486de1d2565b66aab2b6c42c0f4
contracts/interfaces/EIP2612.sol	762590cf872ae51c556f209730a6b9fda2adf921
contracts/interfaces/IDepository.sol	1cb698001233d270b0490ae4fb2437546aa6fa40
contracts/interfaces/IPriceOracle.sol	cfe142174736a6e7a604c186920da9d916c9be8a
contracts/interfaces/IWrappedNativeCurrency.sol	c71ea61b95dcf0214b504694e9029cf33ea4775d
contracts/interfaces/ERC20.sol	0aa9e66531b96d6245394c4925f02e33664001ce
contracts/interfaces/IBank.sol	74d5f67ea83b9cf63b4e233a5e2eb4e459405bb7
contracts/libraries/Address.sol	e0e6c3de29932c0810e3f502c77ad1b74edb8fed
contracts/libraries/Math.sol	1429c90bdcf252a89a6d4711b6f8d64f850473cc
contracts/libraries/AmountNormalization.sol	854bc4cf8e570c17a653e70aa0474c83a0898efc
contracts/libraries/Magister.sol	c92f545770adc6424e0ae1bd24c9fe236f523d57
contracts/libraries/SafeERC20.sol	4be13e0ddde00aa02bd56151e98d553c32ee870b
contracts/libraries/FixedPointMath.sol	18b008a953b045adb2a6eee6e7c778a3dba12385
contracts/libraries/Upgradability.sol	a10acb5cc5fba62ff2d7e4f9b6856ee16017553d
contracts/libraries/Pool.sol	ca2565ecdf7626679b15a3c51f7b61443cbd62f
contracts/libraries/EnumerableAddressSet.sol	bff81588fe35273f05bb7998a3a9b2adfb6d8a5b
contracts/libraries/CollateralToken.sol	ce741e9a579a477d37ec282206007a575cddc99f
contracts/libraries/Loan.sol	be551d43ee22b3093f35fef2c41fa0b8bc400c53

## v1.1

File Name	SHA-1 Hash
contracts/Voice.sol	c3505175b48d6500501a38cff4c33c37a92f622a
contracts/interfaces/Chainlink.sol	c247bda0380df398860201f6225360a324f58b7a
contracts/interfaces/IERC165.sol	754045c4544e2677fbff8e619bbb6dd4f4ab0348
contracts/interfaces/IEIP173.sol	d23be034f71cb447a9b3eb17ffedaa19013d3cb7
contracts/interfaces/UniswapV2.sol	3db416095fd0b486de1d2565b66aab2b6c42c0f4
contracts/interfaces/IEIP2612.sol	762590cf872ae51c556f209730a6b9fda2adf921
contracts/interfaces/IDepository.sol	a804b614b759efc031604da463862ffa0c27cfac
contracts/interfaces/IPriceOracle.sol	6e0df7359113a9eac20282ebe5dea305a990306a
contracts/interfaces/IWrappedNativeCurrency.sol	c71ea61b95dcf0214b504694e9029cf33ea4775d
contracts/interfaces/ERC20.sol	0aa9e66531b96d6245394c4925f02e33664001ce
contracts/interfaces/IBank.sol	e4a5c7c7ea5fb804eddab5f7d5c91988a994f3a7
contracts/Bank.sol	5ee7ae0b0867060f2e78487ccd96a8207e7b0ddb
contracts/Core.sol	26c1b6ead8651049ac4a73b64b46b95f3a00309c
contracts/ExchangeFund.sol	af6a256463ab670c453b95ea6a0a8803273f3ba8
contracts/DevelopmentFund.sol	5a83507b1ff34b25a7de46bb485fb0a3c73d850e
contracts/ChainlinkPriceOracle.sol	63026d26a39f567bfb6fdc97b30d1c447ec2a24d
contracts/Baks.sol	c854695fcb334ed778d875c75d3eab84a336109
contracts/BaseToken.sol	1ed0b2388d98721e6bab26c2c3937d182f69d718
contracts/Depository.sol	6451c5877442905299bb811607e40257843f97ab
contracts/Governance.sol	4b8414da0a15d01c845dc137dda60771e231c217
contracts/libraries/EnumerableAddressSet.sol	bff81588fe35273f05bb7998a3a9b2adfb6d8a5b
contracts/libraries/CollateralToken.sol	ce741e9a579a477d37ec282206007a575cddc99f
contracts/libraries/Loan.sol	0728b54dcf5813ee121e535ce608cd02f6e71a54
contracts/libraries/FixedPointMath.sol	915a7f7d19e67ff8cba895d4a9757b2745342ec5
contracts/libraries/Math.sol	1429c90bdcf252a89a6d4711b6f8d64f850473cc
contracts/libraries/AmountNormalization.sol	854bc4cf8e570c17a653e70aa0474c83a0898efc
contracts/libraries/MintingStage.sol	77e591b9f3dd1996fad917b6904aa0170848fb4d
contracts/libraries/Address.sol	96ccdc52363a0ba3d7e51a5a7f728dc62aa0a770
contracts/libraries/Magister.sol	c92f545770adc6424e0ae1bd24c9fe236f523d57
contracts/libraries/SafeERC20.sol	35efe63b37b80339e7d10bb82f5cc9ebab45751f
contracts/libraries/Upgradability.sol	b21863605b02149403b7d89c45e54028716e05b5
contracts/libraries/Pool.sol	b51e97615e91c4b89d89bf39bc6f245b02402702
contracts/libraries/Deposit.sol	a77d79889a7bd6294d3a8d041a0f645082eff996
contracts/libraries/ReentrancyGuard.sol	4c0abb7a54861c306517fa0d6bc478a5811e5b4b
contracts/libraries/Beneficiary.sol	6f31a246bd05d9564313bf6325c1c8ae2453593d

## V1.2

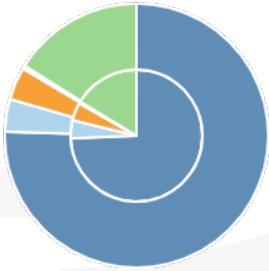
File Name	SHA-1 Hash
contracts/Governance.sol	4b8414da0a15d01c845dc137dda60771e231c217
contracts/BaseToken.sol	1ed0b2388d98721e6bab26c2c3937d182f69d718
contracts/Depository.sol	d699f00d658d73efd1b99d6af963ae89172cf916
contracts/Bank.sol	5ac27b025e300d90cf41bdfac2ad11e431c9e1ff
contracts/Core.sol	9aa7f0eba0e154e3f47f872ae14c9565a1a775e4
contracts/ExchangeFund.sol	4ffa0d716527661d1d3b765abf1afcc3348bc293
contracts/DevelopmentFund.sol	5a83507b1ff34b25a7de46bb485fb0a3c73d850e
contracts/ChainlinkPriceOracle.sol	63026d26a39f567bfb6fdc97b30d1c447ec2a24d
contracts/Baks.sol	be44ef36eab1f1ae4381a363245054e686d50b33
contracts/libraries/Address.sol	96ccdc52363a0ba3d7e51a5a7f728dc62aa0a770
contracts/libraries/MintingStage.sol	77e591b9f3dd1996fad917b6904aa0170848fb4d
contracts/libraries/AmountNormalization.sol	854bc4cf8e570c17a653e70aa0474c83a0898efc
contracts/libraries/Magister.sol	c92f545770adc6424e0ae1bd24c9fe236f523d57
contracts/libraries/SafeERC20.sol	35efe63b37b80339e7d10bb82f5cc9ebab45751f
contracts/interfaces/Chainlink.sol	c247bda0380df398860201f6225360a324f58b7a
contracts/libraries/Upgradability.sol	b21863605b02149403b7d89c45e54028716e05b5
contracts/interfaces/IERC165.sol	754045c4544e2677fbff8e619bbb6dd4f4ab0348
contracts/libraries/Math.sol	de755c84287dbc85af926659a9348faded13e6d7
contracts/interfaces/IEIP173.sol	d23be034f71cb447a9b3eb17ffedaa19013d3cb7
contracts/libraries/Pool.sol	b51e97615e91c4b89d89bf39bc6f245b02402702
contracts/interfaces/UniswapV2.sol	3db416095fd0b486de1d2565b66aab2b6c42c0f4
contracts/libraries/Deposit.sol	a77d79889a7bd6294d3a8d041a0f645082eff996
contracts/interfaces/IEIP2612.sol	762590cf872ae51c556f209730a6b9fda2adf921
contracts/libraries/FixedPointMath.sol	cffc8fa4a3a68201e687044ab9514601e80ac245
contracts/interfaces/IDepository.sol	a804b614b759efc031604da463862ffa0c27cfac
contracts/libraries/ReentrancyGuard.sol	4c0abb7a54861c306517fa0d6bc478a5811e5b4b
contracts/interfaces/IPriceOracle.sol	6e0df7359113a9eac20282ebe5dea305a990306a
contracts/interfaces/IWrappedNativeCurrency.sol	c71ea61b95dcf0214b504694e9029cf33ea4775d
contracts/libraries/Beneficiary.sol	6f31a246bd05d9564313bf6325c1c8ae2453593d
contracts/libraries/Loan.sol	0728b54dcf5813ee121e535ce608cd02f6e71a54
contracts/interfaces/ERC20.sol	0aa9e66531b96d6245394c4925f02e33664001ce
contracts/interfaces/IBank.sol	e4a5c7c7ea5fb804eddab5f7d5c91988a994f3a7
contracts/libraries/EnumerableAddressSet.sol	bff81588fe35273f05bb7998a3a9b2adfb6d8a5b
contracts/libraries/CollateralToken.sol	ce741e9a579a477d37ec282206007a575cddc99f

# Metrics

## Source Lines

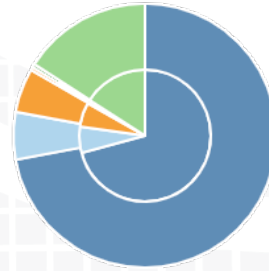
v1.0

source comment single block mixed  
empty todo blockEmpty



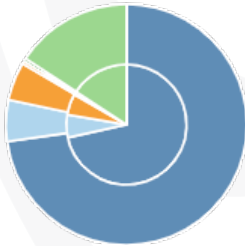
v1.1

source comment single block mixed  
empty todo blockEmpty



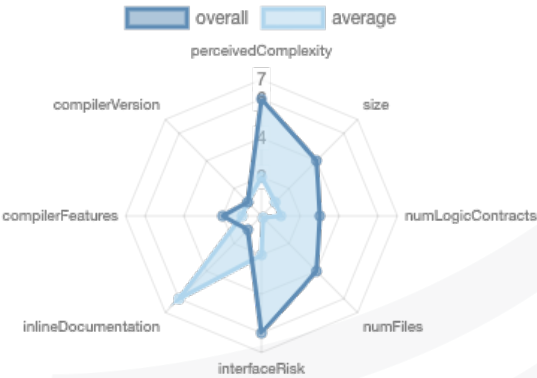
v1.2

source comment single block mixed  
empty todo blockEmpty

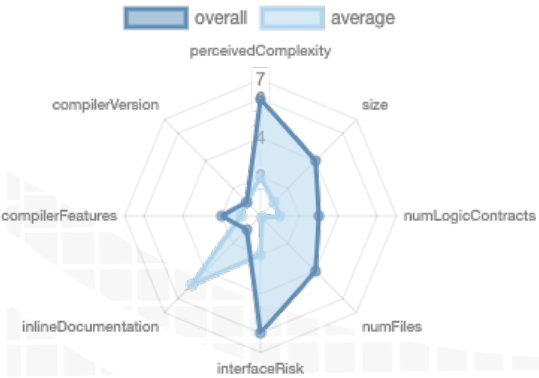


# Risk Level

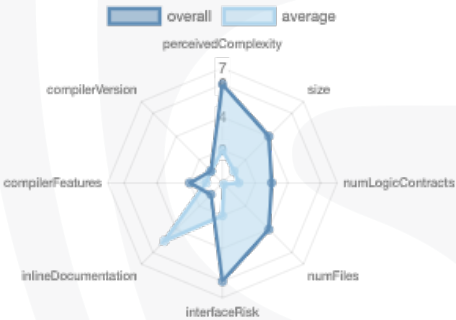
## v1.0



## v1.1



## v1.2



# Capabilities

## Components

Version	Contracts	Libraries	Interfaces	Abstract
1.0	10	12	14	5
1.1	10	14	14	5
1.2	9	14	14	5

## Exposed Functions

*This section lists functions that are explicitly declared public or payable. Please note that getter methods for public stateVars are not included.*

Version	Public	Payable
1.0	153	6
1.1	174	6

Version	External	Internal	Private	Pure	View
1.0	142	166	0	19	86
1.1	162	177	0	21	98
1.2	162	174	0	19	98

## State Variables

Version	Total	Public
1.0	85	53
1.1	93	59
1.2	92	59

## Capabilities

Version	Solidity Versions observed	Experimental Features	Can Receive Funds	Uses Assembly	Has Destroyable Contracts
1.0	0.8.10		yes	yes (13 asm blocks)	
1.1	0.8.10		yes	yes (13 asm blocks)	

Version	Transfers ETH	Low-Level Calls	DelegateCall	Uses Hash Functions	ECRecover	New/Create/Create2
1.0			yes	yes	yes	
1.1			yes	yes	yes	

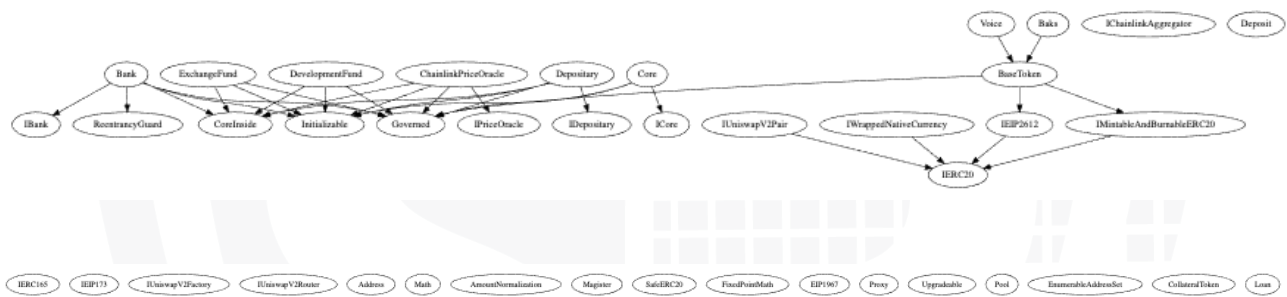
## Scope of Work

The above token Team provided us with the files that needs to be tested (Github, Bscscan, Etherscan, files, etc.). The scope of the audit is the main contract (usual the same name as team appended with .sol).

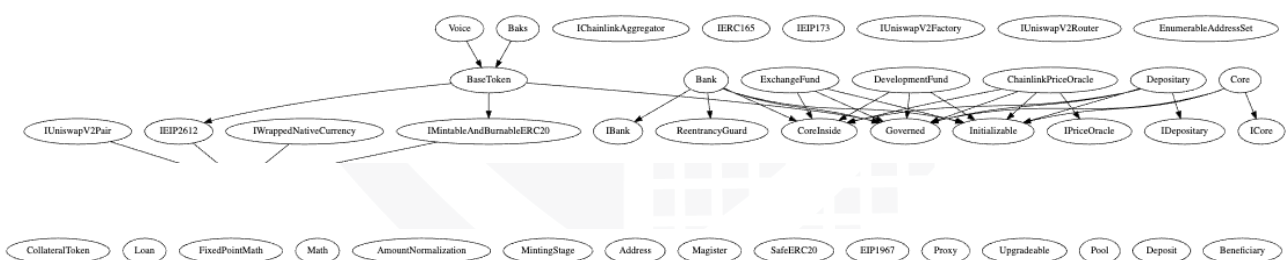
We will verify the following claims:

1. Correct implementation of Token standard
2. Deployer cannot mint any new tokens
3. Deployer cannot burn or lock user funds
4. Deployer cannot pause the contract
5. Overall checkup (Smart Contract Security)

# Inheritance Graph v1.0



## v1.1





## Verify Claims

### Correct implementation of Token standard

Tested	Verified
✓	✓

Function	Description	Exist	Tested	Verified
TotalSupply	provides information about the total token supply	✓	✓	✓
BalanceOf	provides account balance of the owner's account	✓	✓	✓
Transfer	executes transfers of a specified number of tokens to a specified address	✓	✓	✓
TransferFrom	executes transfers of a specified number of tokens from a specified address	✓	✓	✓
Approve	allow a spender to withdraw a set number of tokens from a specified account	✓	✓	✓
Allowance	returns a set number of tokens from a spender to the owner	✓	✓	✓

Write functions of contract  
v1.0

BAKS

acceptGovernance

approve

burn

mint

permit

transfer

setMinter

transferFrom

transitGovernance

BANK

acceptGovernance

borrow

borrowInNativeCurrency

deposit

depositInNativeCurrency

initialize

liquidate

listCollateralToken

onNewDeposit

rebalance

repay

salvage

setInitialLoanToValueRatio

transitGovernance

unlistCollateralToken

CHAINLINKPRICEORACLE

acceptGovernance

initialize

setAggregator

transitGovernance

CORE	DEPOSITARY	EXCHANGEFUND	VOICE
acceptGovernance		acceptGovernance	acceptGovernance
initialize	acceptGovernance	deposit	approve
setBaks		divest	burn
setBank	addPool	initialize	mint
setDepositary	blacklistMagister	invest	permit
setDevelopmentFund		listDepositableToken	transfer
setEarlyWithdrawalFee	deposit	salvage	setMinter
setEarlyWithdrawalPeriod	deposit	service	transferFrom
setExchangeFund	initialize	setSlippageTolerance	transitGovernance
setInterest		setSwapDeadline	
setLiquidationLoanToValueRatio	transitGovernance	swap	
setLiquidator		transitGovernance	
setMarginCallLoanToValueRatio	updatePool	unlistDepositableToken	
setMinimumLiquidity	whitelistMagister	withdraw	
setMinimumMagisterDepositA...	withdraw		
setMinimumPrincipalAmount			
setOperator			
setPlatformFees	DEVELOPMENTFUND		
setPriceOracle	acceptGovernance		
setRebalancingThreshold			
setServicingThreshold	initialize		
setStabilityFee			
setVoice	transitGovernance		
setWorkFee			
transitGovernance			

BANK	CORE	DEVELOPMENTFUND	DEPOSITARY
acceptGovernance	acceptGovernance	acceptGovernance	acceptGovernance
borrow	addSuperUser	initialize	addPool
borrowInNativeCurrency	initialize	transitGovernance	blacklistMagister
deposit	removeSuperUser	withdraw	deposit
depositInNativeCurrency	setBaks	EXCHANGEFUND	deposit
initialize	setBank	acceptGovernance	initialize
liquidate	setDepositary	deposit	transitGovernance
listCollateralToken	setDepositFees	divest	updatePool
migrate	setDevelopmentFund	initialize	whitelistMagister
mintVoice	setEarlyWithdrawalFee	invest	withdraw
onNewDeposit	setEarlyWithdrawalPeriod	listDepositableToken	
rebalance	setExchangeFund	salvage	
repay	setInterest	service	
setInitialLoanToValueR...	setLiquidationLoanToValueRatio	setSlippageTolerance	
transferBaksToExchan...	setLiquidator	setSwapDeadline	
transitGovernance	setMarginCallLoanToValueRatio	swap	
unlistCollateralToken	setMinimumLiquidity	transferBaksToBank	
	setMinimumMagisterDepositAmount	transitGovernance	
	setMinimumPrincipalAmount	unlistDepositableToken	
	setOperator	withdraw	
	setPlatformFees	setStabilityFee	
	setPriceOracle	setVoice	
	setRebalancingThreshold	setVoiceMintingBeneficiaries	
	setServicingThreshold	setWorkFee	
		transitGovernance	

## Deployer cannot mint any new tokens

File	Name	Exist	Tested	Verified
Baks	cannot mint	✓	✓	✗
Voice	cannot mint	✓	✓	✗

Comments:

### v1.0

- Baks
  - onlyMinter can mint
- Voice
  - onlyMinter can mint

## Deployer cannot burn or lock user funds

File	Name	Exist	Tested	Verified
Baks	cannot lock	✓	✓	✓
	cannot burn	✓	✓	✗
Voice	cannot lock	✓	✓	✓
	cannot burn	✓	✓	✗

Deployer cannot pause the contract

File	Name	Exist	Teste d	Verifie d
All	cannot pause	-	-	-



## Overall checkup (Smart Contract Security)

Tested	Verified
✓	✓

### Legend

Attribute	Symbol
Verified / Checked	✓
Partly Verified	⚠
Unverified / Not checked	✗
Not available	—



## Modifiers

### Baks/Voice

- onlyMinter
  - Mint
  - Burn
- onlyGovernor
  - setMinter

## Bank

◆ initialize	◆ onNewDeposit
Ⓜ initializer	Ⓜ onlyDepository
◆ borrow	◆ listCollateralToken
Ⓜ tokenAllowedAsCollateral	Ⓜ onlyGovernor
◆ borrowInNativeCurrency 💰	◆ unlistCollateralToken
Ⓜ nonReentrant	Ⓜ onlyGovernor
◆ deposit	◆ setInitialLoanToValueRatio
Ⓜ onActiveLoan	Ⓜ onlyGovernor
Ⓜ notOnSubjectToLiquidation	◆ salvage
◆ depositInNativeCurrency 💰	Ⓜ onlyGovernor
Ⓜ nonReentrant	
Ⓜ onActiveLoan	
Ⓜ notOnSubjectToLiquidation	
◆ repay	
Ⓜ nonReentrant	
Ⓜ onActiveLoan	
Ⓜ notOnSubjectToLiquidation	
◆ liquidate	
Ⓜ onActiveLoan	
Ⓜ onSubjectToLiquidation	

## ChainlinkPriceOracle

- ◆ initialize
- Ⓜ initializer
- ◆ setAggregator
- Ⓜ onlyGovernor

## Core

- ◆ initialize
- Ⓜ initializer

- onlyGovernor

- ◆ setPriceOracle
- ◆ setBaks
- ◆ setVoice
- ◆ setBank
- ◆ setDepositary
- ◆ setExchangeFund
- ◆ setDevelopmentFund
- ◆ setOperator
- ◆ setLiquidator
- ◆ setInterest
- ◆ setMinimumPrincipalAmount
- ◆ setStabilityFee
- ◆ setPlatformFees
- ◆ setMarginCallLoanToValueRatio
- ◆ setLiquidationLoanToValueRatio
- ◆ setRebalancingThreshold
- ◆ setMinimumMagisterDepositAmount
- ◆ setWorkFee
- ◆ setEarlyWithdrawalPeriod
- ◆ setEarlyWithdrawalFee
- ◆ setServicingThreshold
- ◆ setMinimumLiquidity

## Depository

- initialize
- initializer

- whitelistMagister
- onlyGovernor
- blacklistMagister
- onlyGovernor
- addPool
- onlyGovernor
- updatePool
- onlyGovernor

onlyGovernor is changed to onlySuperUser modifier

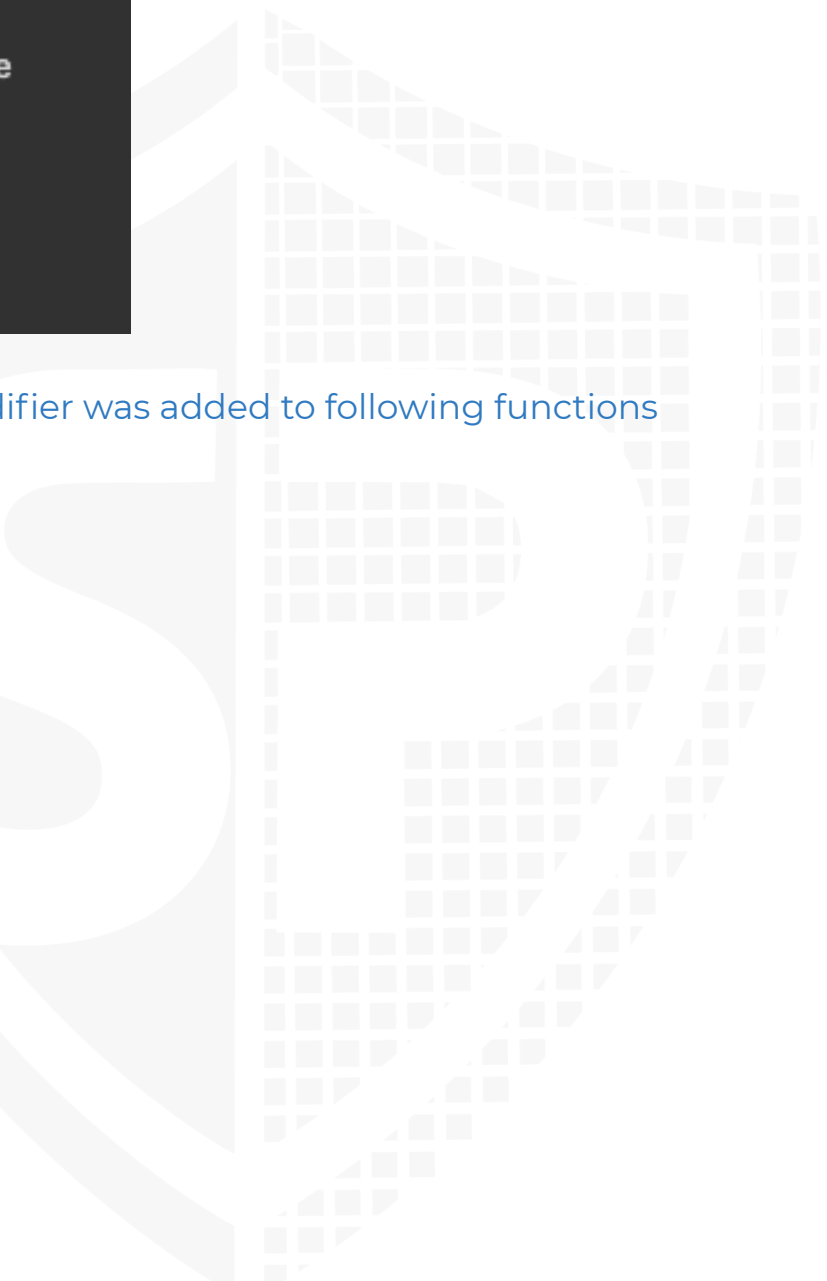
## DevelopmentFund

- Initialize
  - Initializer

## ExchangeFund

- Initialize
  - Initializer

- deposit
- tokenAllowedToBeDeposited
- swap
- tokenAllowedToBeSwapped



- ◆ listDepositToken
- Ⓜ onlyGovernor
- ◆ unlistDepositToken
- Ⓜ onlyGovernor
- ◆ setSlippageTolerance
- Ⓜ onlyGovernor
- ◆ setSwapDeadline
- Ⓜ onlyGovernor
- ◆ salvage
- Ⓜ onlyGovernor
- ◆ service

- onlySuperUser modifier was added to following functions
  - Deposit
  - Invest
  - Divest
  - Withdraw

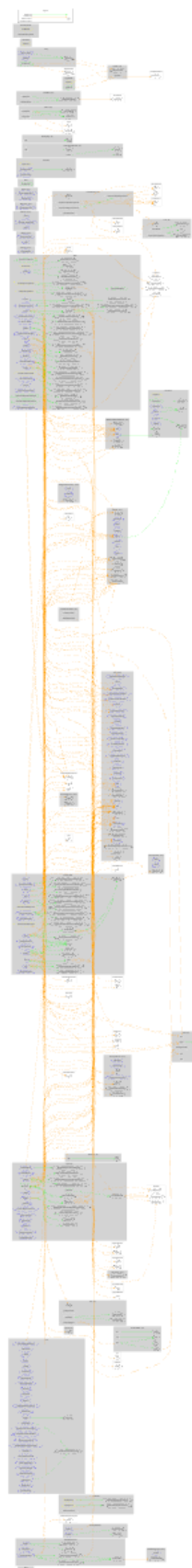
## Governance

- transitGovernance
  - onlyGovernor

# CallGraph v1.0











































# v1.1















































# Source Units in Scope

## v1.0

Type	File	Logic Contracts	Interfaces	Lines	nLines	nSLOC	Comment Lines	Complex. Score	Capabilities
	contracts/Bank.sol	1	————	605	555	400	50	342	
	contracts/Core.sol	2	1	325	263	204	22	187	————
	contracts/Voice.sol	1	————	8	8	5	1	4	————
	contracts/DevelopmentFund.sol	1	————	13	13	10	1	12	————
	contracts/ExchangeFund.sol	1	————	402	397	294	23	319	————
	contracts/Baks.sol	1	————	8	8	5	1	4	————
	contracts/BaseToken.sol	1	————	226	206	162	2	103	
	contracts/ChainlinkPriceOracle.sol	1	————	88	84	66	1	54	————
	contracts/Depository.sol	1	————	422	387	323	2	225	
	contracts/Governance.sol	1	————	52	52	42	1	23	————
	contracts/libraries/ReentrancyGuard.sol	1	————	27	27	18	1	6	————
	contracts/interfaces/Chainlink.sol	————	1	17	5	3	1	5	————
	contracts/libraries/Deposit.sol	1	————	27	27	23	1	1	————
	contracts/interfaces/ERC165.sol	————	1	6	5	3	1	3	
	contracts/interfaces/EIP173.sol	————	1	10	7	4	1	5	————
	contracts/interfaces/UniswapV2.sol	————	3	80	9	6	1	29	————
	contracts/interfaces/EIP2612.sol	————	1	21	7	4	2	9	————
	contracts/interfaces/Depository.sol	————	1	6	5	3	1	3	————
	contracts/interfaces/PriceOracle.sol	————	1	24	23	8	12	3	————
	contracts/interfaces/IWrappedNativeCurrency.sol	————	1	10	7	4	1	10	
	contracts/interfaces/ERC20.sol	————	2	35	9	6	1	26	————
	contracts/interfaces/Bank.sol	————	1	8	7	4	1	3	————
	contracts/libraries/Address.sol	1	————	60	48	39	1	34	
	contracts/libraries/Math.sol	1	————	100	100	90	2	26	————
	contracts/libraries/AmountNormalization.sol	1	————	28	24	19	1	6	————
	contracts/libraries/Magister.sol	1	————	18	18	14	1	1	————
	contracts/libraries/SafeERC20.sol	1	————	39	29	21	1	11	————
	contracts/libraries/FixedPointMath.sol	1	————	374	370	333	6	286	
	contracts/libraries/Upgradability.sol	4	————	113	113	86	1	89	
	contracts/libraries/Pool.sol	1	————	62	58	47	1	15	————
	contracts/libraries/EnumerableAddressSet.sol	1	————	51	51	40	1	10	————
	contracts/libraries/CollateralToken.sol	1	————	129	102	86	1	33	————
	contracts/libraries/Loan.sol	1	————	55	55	45	1	15	————
	<b>Totals</b>	<b>27</b>	<b>14</b>	<b>3449</b>	<b>3079</b>	<b>2417</b>	<b>145</b>	<b>1902</b>	

Type	File	Logic Contracts	Interfaces	Lines	nLines	nSLOC	Comment Lines	Complex. Score	Capabilities
	contracts/Voice.sol	1	_____	8	8	5	1	4	_____
	contracts/interfaces/Chainlink.sol	_____	1	17	5	3	1	5	_____
	contracts/interfaces/IERC165.sol	_____	1	6	5	3	1	3	
	contracts/interfaces/IEIP173.sol	_____	1	10	7	4	1	5	_____
	contracts/interfaces/UniswapV2.sol	_____	3	80	9	6	1	29	_____
	contracts/interfaces/IEIP2612.sol	_____	1	21	7	4	2	9	_____
	contracts/interfaces/IDepository.sol	_____	1	10	7	4	1	5	_____
	contracts/interfaces/IPriceOracle.sol	_____	1	24	23	5	15	3	_____
	contracts/interfaces/WRappedNativeCurrency.sol	_____	1	10	7	4	1	10	
	contracts/interfaces/ERC20.sol	_____	2	35	9	6	1	26	_____
	contracts/interfaces/IBank.sol	_____	1	10	7	4	1	5	_____
	contracts/Bank.sol	1	_____	680	630	438	66	384	
	contracts/Core.sol	2	1	444	364	291	25	236	_____
	contracts/ExchangeFund.sol	1	_____	412	407	301	23	332	_____
	contracts/DevelopmentFund.sol	1	_____	21	21	16	1	18	_____
	contracts/ChainlinkPriceOracle.sol	1	_____	92	88	66	5	54	_____
	contracts/Baks.sol	1	_____	8	8	5	1	4	_____
	contracts/BaseToken.sol	1	_____	240	220	149	29	103	
	contracts/Depository.sol	1	_____	428	393	330	2	232	
	contracts/Governance.sol	1	_____	56	56	38	9	23	_____
	contracts/libraries/EnumerableAddressSet.sol	1	_____	51	51	40	1	10	_____
	contracts/libraries/CollateralToken.sol	1	_____	129	102	86	1	33	_____
	contracts/libraries/Loan.sol	1	_____	55	55	45	1	15	_____
	contracts/libraries/FixedPointMath.sol	1	_____	379	375	329	15	286	
	contracts/libraries/Math.sol	1	_____	100	100	90	2	26	_____
	contracts/libraries/AmountNormalization.sol	1	_____	28	24	19	1	6	_____
	contracts/libraries/MintingStage.sol	1	_____	9	9	7	1	2	_____
	contracts/libraries/Address.sol	1	_____	62	50	38	4	34	
	contracts/libraries/Magister.sol	1	_____	18	18	14	1	1	_____
	contracts/libraries/SafeERC20.sol	1	_____	40	30	20	3	11	_____
	contracts/libraries/Upgradability.sol	4	_____	117	117	82	9	89	
	contracts/libraries/Pool.sol	1	_____	62	57	46	1	14	_____
	contracts/libraries/Deposit.sol	1	_____	27	27	23	1	1	_____
	contracts/libraries/ReentrancyGuard.sol	1	_____	28	28	17	3	6	_____
	contracts/libraries/Beneficiary.sol	1	_____	9	9	7	1	4	_____
	<b>Totals</b>	<b>29</b>	<b>14</b>	<b>3726</b>	<b>3333</b>	<b>2545</b>	<b>232</b>	<b>2028</b>	

Type	File	Logic Contracts	Interfaces	Lines	nLines	nSLOC	Comment Lines	Complex. Score	Capabilities
	contracts/Governance.sol	1	_____	56	56	38	9	23	_____
	contracts/BaseToken.sol	1	_____	240	220	149	29	103	
	contracts/Depositary.sol	1	_____	428	393	330	2	232	
	contracts/Bank.sol	1	_____	679	629	437	66	383	
	contracts/Core.sol	2	1	489	409	335	25	278	_____
	contracts/ExchangeFund.sol	1	_____	422	417	310	23	340	_____
	contracts/DevelopmentFund.sol	1	_____	21	21	16	1	18	_____
	contracts/ChainlinkPriceOracle.sol	1	_____	92	88	66	5	54	_____
	contracts/Baks.sol	1	_____	8	8	5	1	4	_____
	contracts/libraries/Address.sol	1	_____	62	50	38	4	34	
	contracts/libraries/MintingStage.sol	1	_____	9	9	7	1	2	_____
	contracts/libraries/AmountNormalization.sol	1	_____	28	24	19	1	6	_____
	contracts/libraries/Magister.sol	1	_____	18	18	14	1	1	_____
	contracts/libraries/SafeERC20.sol	1	_____	40	30	20	3	11	_____
	contracts/interfaces/Chainlink.sol	_____	1	17	5	3	1	5	_____
	contracts/libraries/Upgradability.sol	4	_____	117	117	82	9	89	
	contracts/interfaces/ERC165.sol	_____	1	6	5	3	1	3	
	contracts/libraries/Math.sol	1	_____	96	96	87	2	25	_____
	contracts/interfaces/EIP173.sol	_____	1	10	7	4	1	5	_____
	contracts/libraries/Pool.sol	1	_____	62	57	46	1	14	_____
	contracts/interfaces/UniswapV2.sol	_____	3	80	9	6	1	29	_____
	contracts/libraries/Deposit.sol	1	_____	27	27	23	1	1	_____
	contracts/interfaces/EIP2612.sol	_____	1	21	7	4	2	9	_____
	contracts/libraries/FixedPointMath.sol	1	_____	363	359	324	6	283	
	contracts/interfaces/Depositary.sol	_____	1	10	7	4	1	5	_____
	contracts/libraries/ReentrancyGuard.sol	1	_____	28	28	17	3	6	_____
	contracts/interfaces/PriceOracle.sol	_____	1	24	23	5	15	3	_____
	contracts/interfaces/WrappedNativeCurrency.sol	_____	1	10	7	4	1	10	
	contracts/libraries/Beneficiary.sol	1	_____	9	9	7	1	4	_____
	contracts/libraries/Loan.sol	1	_____	55	55	45	1	15	_____
	contracts/interfaces/ERC20.sol	_____	2	35	9	6	1	26	_____
	contracts/interfaces/IBank.sol	_____	1	10	7	4	1	5	_____
	contracts/libraries/EnumerableAddressSet.sol	1	_____	51	51	40	1	10	_____
	contracts/libraries/CollateralToken.sol	1	_____	129	102	86	1	33	_____
	<b>Totals</b>	<b>28</b>	<b>14</b>	<b>3752</b>	<b>3359</b>	<b>2584</b>	<b>222</b>	<b>2069</b>	

## Legend

Attribute	Description
Lines	total lines of the source unit
nLines	normalized lines of the source unit (e.g. normalizes functions spanning multiple lines)
nSLOC	normalized source lines of code (only source-code lines; no comments, no blank lines)
Comment Lines	lines containing single or block comments
Complexity Score	a custom complexity score derived from code statements that are known to introduce code complexity (branches, loops, calls, external interfaces, ...)



# Audit Results

## AUDIT PASSED

### Critical issues

- no critical issues found -

### High issues

- no high issues found -

### Medium issues

- no medium issues found -

### Low issues

Issue	File	Type	Line	Description
#1	Main	Contract doesn't import npm packages from source (like OpenZeppelin etc.)	-	We recommend to import all packages from npm directly without flatten the contract. Functions could be modified or can be susceptible to vulnerabilities

### Informational issues

- no informational issues found -

### Audit Comments

#### 13. December 2021:

- [Read whole report for more information](#)

#### 07. January 2022:

- Bank
  - New errors were added
  - MintingStage was added
  - mintVoice function was added
  - transferBaksToExchangeFund function was added
  - migrate function was added
  - getTotalValueLocked function was added
- Core

- DepositFeesUpdated event was added
- Voice minting schedule state variable was added
- setDepositFees function was added
- setVoiceMintingBeneficiaries function was added
- New modifier (onlySuperUser) was added
- Depositary
  - getTotalValueLocked function was added
- DevelopmentFund
  - withdraw function was added
- ExchangeFund
  - transferBaksToBank function was added
- MintingStage
  - New source file was added to project

## **17. January 2022:**

- Core
  - New error was added
- Bugs were fixed by team

# BaksDAO Test Protocol

Because the repository is private, we chose not to run a public test on the testnet and instead used a local Ganache installation for the subsequent tests.

## Compiling successful

Compiling 36 files with 0.8.10

Generating typings for: 44 artifacts in dir: ./src/types/ for target: ethers-v5

Successfully generated 65 typings!

Compilation finished successfully

*Note: Solidity 0.8.10 is not fully supported by Hardhat yet. Hardhat can still be used, but some features, like stack traces, might not work correctly.*

## Deployment

All contracts were deployed on local running blockchain node. No excessive gas usages were detected.

## Conclusion

The core functions such as deposit and withdraw also work manually on the blockchain as indicated in the unit tests. Corresponding events are fired. Custom claims were not specified, the basic functions that can be constructed from the whitepaper and the unit tests were tested.

The contract is safe to deploy and basic logic errors were not detected. The code is written to the best standard and sufficiently commented. Known risks were checked by the auditor and the code was scanned for other vulnerabilities as well.

All unit tests make sense and have also been checked. Logic errors could not be found here either.

## Disclaimer

We have checked and verified the code to the best of our knowledge. However, deeper logic errors cannot be excluded and Solidproof.io cannot be held liable for any damage that may occur.

## All Unit Tests successful

42 passing (5s)

### Bank

#### governed functions

- ✓ lists and unlists token as collateral (58ms)

#### initial loan-to-value ratio

- ✓ reverts when token is not allowed as collateral (44ms)
- ✓ reverts when new initial loan-to-value ratio equal or higher than margin call loan-to-value ratio
- ✓ updates initial loan-to-value ratio

#### salvation

- ✓ reverts when trying to salvage one of collateral tokens
- ✓ reverts when trying to salvage stablecoin

#### borrow

- ✓ reverts when token is not allowed as collateral
- ✓ reverts when `amount` is zero (47ms)
- ✓ reverts when user don't approve a sufficient amount of collateral tokens (40ms)
- ✓ mints correct amount of stablecoin and creates a loan (171ms)

#### loan interactions

##### deposit

- ✓ reverts on inactive loan
- ✓ reverts when `amount` is zero
- ✓ reverts when user don't approve a sufficient amount of collateral tokens
- ✓ successfully deposits

##### repay

- ✓ reverts when loan is subject to liquidation
- ✓ reverts when `amount` is zero

##### liquidation

- ✓ should revert when loan is not subject to liquidation
- ✓ should liquidate loan

### Base Token

- ✓ have correct minter address

#### metadata

- ✓ returns the correct name
- ✓ returns the correct symbol
- ✓ returns the correct number of decimals

#### total supply

- ✓ returns the total supply of tokens

### Core

#### bank

- ✓ updates stability fee
- ✓ updates margin call loan-to-value ratio
- ✓ updates liquidation loan-to-value ratio

#### platform fees

- ✓ reverts when platform fees sum up to less than one
- ✓ reverts when platform fees sum up to more than one
- ✓ updates platform fees

#### stabilization fund

- ✓ updates rebalancing threshold

#### exchange fund

- ✓ updates minimum liquidity
- ✓ updates servicing threshold

### Exchange Fund

#### governed functions

- ✓ lists and unlists depositable token (47ms)
- ✓ updates slippage tolerance
- ✓ updates swap deadline
- ✓ service (stableCoin=>token) and need addLiquidity (67ms)
- ✓ service (stableCoin=>token) and not need addLiquidity (49ms)
- ✓ service (token=>stableCoin) and need addLiquidity (83ms)
- ✓ service (token=>stableCoin) and not need addLiquidity (68ms)

### Math

- ✓ sqrt
- ✓ fpsqrt
- ✓ abs

## SWC Attacks

ID	Title	Relationships	Status
<a href="#">SW C-1 36</a>	Unencrypted Private Data On-Chain	<a href="#">CWE-767: Access to Critical Private Variable via Public Method</a>	PASSED
<a href="#">SW C-1 35</a>	Code With No Effects	<a href="#">CWE-1164: Irrelevant Code</a>	PASSED
<a href="#">SW C-1 34</a>	Message call with hardcoded gas amount	<a href="#">CWE-655: Improper Initialization</a>	PASSED
<a href="#">SW C-1 33</a>	Hash Collisions With Multiple Variable Length	<a href="#">CWE-294: Authentication Bypass by Capture-replay</a>	PASSED
<a href="#">SW C-1 32</a>	Unexpected Ether balance	<a href="#">CWE-667: Improper Locking</a>	PASSED
<a href="#">SW C-1 31</a>	Presence of unused variables	<a href="#">CWE-1164: Irrelevant Code</a>	PASSED
<a href="#">SW C-1 30</a>	Right-To-Left-Override control character (U+202E)	<a href="#">CWE-451: User Interface (UI) Misrepresentation of Critical Information</a>	PASSED
<a href="#">SW C-1 29</a>	Typographical Error	<a href="#">CWE-480: Use of Incorrect Operator</a>	PASSED
<a href="#">SW C-1 28</a>	DoS With Block Gas Limit	<a href="#">CWE-400: Uncontrolled Resource Consumption</a>	PASSED

<a href="#">SW C-1 27</a>	Arbitrary Jump with Function Type	<a href="#">CWE-695: Use of Low-Level Functionality</a>	<b>PASSED</b>
<a href="#">SW C-1 25</a>	Incorrect Inheritance Order	<a href="#">CWE-696: Incorrect Behavior Order</a>	<b>PASSED</b>
<a href="#">SW C-1 24</a>	Write to Arbitrary Storage Location	<a href="#">CWE-123: Write-what-where Condition</a>	<b>PASSED</b>
<a href="#">SW C-1 23</a>	Requirement Violation	<a href="#">CWE-573: Improper Following of Specification by Caller</a>	<b>PASSED</b>
<a href="#">SW C-1 22</a>	Lack of Proper Signature Verification	<a href="#">CWE-345: Insufficient Verification of Data Authenticity</a>	<b>PASSED</b>
<a href="#">SW C-1 21</a>	Missing Protection against Signature Replay Attacks	<a href="#">CWE-347: Improper Verification of Cryptographic Signature</a>	<b>PASSED</b>
<a href="#">SW C-1 20</a>	Weak Sources of Randomness from Chain	<a href="#">CWE-330: Use of Insufficiently Random Values</a>	<b>PASSED</b>
<a href="#">SW C-11 9</a>	Shadowing State Variables	<a href="#">CWE-710: Improper Adherence to Coding Standards</a>	<b>PASSED</b>
<a href="#">SW C-11 8</a>	Incorrect Constructor Name	<a href="#">CWE-665: Improper Initialization</a>	<b>PASSED</b>
<a href="#">SW C-11 7</a>	Signature Malleability	<a href="#">CWE-347: Improper Verification of Cryptographic Signature</a>	<b>PASSED</b>

<a href="#">SW C-11 6</a>	Timestamp Dependence	<a href="#">CWE-829: Inclusion of Functionality from Untrusted Control Sphere</a>	<b>PASSED</b>
<a href="#">SW C-11 5</a>	Authorization through tx.origin	<a href="#">CWE-477: Use of Obsolete Function</a>	<b>PASSED</b>
<a href="#">SW C-11 4</a>	Transaction Order Dependence	<a href="#">CWE-362: Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')</a>	<b>PASSED</b>
<a href="#">SW C-11 3</a>	DoS with Failed Call	<a href="#">CWE-703: Improper Check or Handling of Exceptional Conditions</a>	<b>PASSED</b>
<a href="#">SW C-11 2</a>	Delegatecall to Untrusted Callee	<a href="#">CWE-829: Inclusion of Functionality from Untrusted Control Sphere</a>	<b>PASSED</b>
<a href="#">SW C-11 1</a>	Use of Deprecated Solidity Functions	<a href="#">CWE-477: Use of Obsolete Function</a>	<b>PASSED</b>
<a href="#">SW C-11 0</a>	Assert Violation	<a href="#">CWE-670: Always-Incorrect Control Flow Implementation</a>	<b>PASSED</b>
<a href="#">SW C-1 09</a>	Uninitialized Storage Pointer	<a href="#">CWE-824: Access of Uninitialized Pointer</a>	<b>PASSED</b>
<a href="#">SW C-1 08</a>	State Variable Default Visibility	<a href="#">CWE-710: Improper Adherence to Coding Standards</a>	<b>PASSED</b>
<a href="#">SW C-1 07</a>	Reentrancy	<a href="#">CWE-841: Improper Enforcement of Behavioral Workflow</a>	<b>PASSED</b>
<a href="#">SW C-1 06</a>	Unprotected SELFDESTRUCT Instruction	<a href="#">CWE-284: Improper Access Control</a>	<b>PASSED</b>

<a href="#">SW</a> <a href="#">C-1</a> <a href="#">05</a>	Unprotected Ether Withdrawal	<a href="#">CWE-284: Improper Access Control</a>	<b>PASSED</b>
<a href="#">SW</a> <a href="#">C-1</a> <a href="#">04</a>	Unchecked Call Return Value	<a href="#">CWE-252: Unchecked Return Value</a>	<b>PASSED</b>
<a href="#">SW</a> <a href="#">C-1</a> <a href="#">03</a>	Floating Pragma	<a href="#">CWE-664: Improper Control of a Resource Through its Lifetime</a>	<b>PASSED</b>
<a href="#">SW</a> <a href="#">C-1</a> <a href="#">02</a>	Outdated Compiler Version	<a href="#">CWE-937: Using Components with Known Vulnerabilities</a>	<b>PASSED</b>
<a href="#">SW</a> <a href="#">C-1</a> <a href="#">01</a>	Integer Overflow and Underflow	<a href="#">CWE-682: Incorrect Calculation</a>	<b>PASSED</b>
<a href="#">SW</a> <a href="#">C-1</a> <a href="#">00</a>	Function Default Visibility	<a href="#">CWE-710: Improper Adherence to Coding Standards</a>	<b>PASSED</b>



The logo features the words "Solid Proofed" in a white, elegant script font. The word "Solid" is positioned above "Proofed". Behind the text is a faint, stylized shield emblem with a grid-like pattern, rendered in a darker shade of blue. The entire composition is set against a solid blue background.

Solid  
Proofed

**Blockchain Security | Smart Contract Audits | KYC**

A small horizontal bar representing the German flag, with black, red, and gold stripes.

MADE IN GERMANY