SOLIDITY AUDIT

Smart Contract Solidity Audit

Audit Details:

Audited project: Los Pobres Coin

Deployer Address: 0xCA46Fc1290Ec39Fecb0165A5B100BB476f3b5ab0

Blockchain: Binance Smart Chain

December 01

Audit

This document may contain confidential information about IT systems and the intellectual property of the Customer as well as information about potential vulnerabilities and methods of their exploitation. The report containing confidential information can be used internally by the Customer, or it can be disclosed publicly after all vulnerabilities are fixed - upon a decision of the Customer.

Introduction

Solidity Audit (Consultant) was contracted by Los Pobres Audit (Customer) to conduct a Smart Contract Code Review and Security Analysis. This report presents the findings of the security assessment of Customer`s smart contract . Scope The scope of the project is main net smart contracts that can be found on BSCscan:

https://bscscan.com/token/0xCA46Fc1290Ec39Fecb0165A5B100BB476f3b5ab0

We have scanned this smart contract for commonly known and more specific vulnerabilities. List of the commonly known vulnerabilities that are considered:

Category	Check Item
Code review	▪ Reentrancy ▪ Ownership Takeover ▪ Timestamp Dependence ▪ Gas Limit and Loops ▪ DoS with (Unexpected) Throw ▪ DoS with Block Gas Limit ▪ Transaction-Ordering Dependence ▪ Style guide violation ▪ Costly Loop ▪ ERC20 API violation ▪ Unchecked external call ▪ Unchecked math ▪ Unsafe type inference ▪ Implicit visibility level ▪ Deployment Consistency ▪ Repository Consistency ▪ Data Consistency
Functional review	▪ Business Logics Review ▪ Functionality Checks ▪ Access Control & Authorization ▪ Escrow manipulation ▪ Token Supply manipulation ▪ Assets integrity ▪ User Balances manipulation ▪ Kill-Switch Mechanism ▪ Operation Trails & Event Generation

Our team performed an analysis of code functionality, manual audit, and automated checks with Mythril and Slither. All issues found during automated analysis were manually reviewed, and important vulnerabilities are presented in the Audit overview section. A general overview is presented in AS-IS section, and all found issues can be found in the Audit overview section.

Security engineers found **2** medium, **1** informational issue during the audit.

Notice: the audit scope is limited and not include all files in the repository. Though, reviewed contracts are secure, we may not guarantee secureness of contracts that are not in the scope.

Description Report

Files Description Table

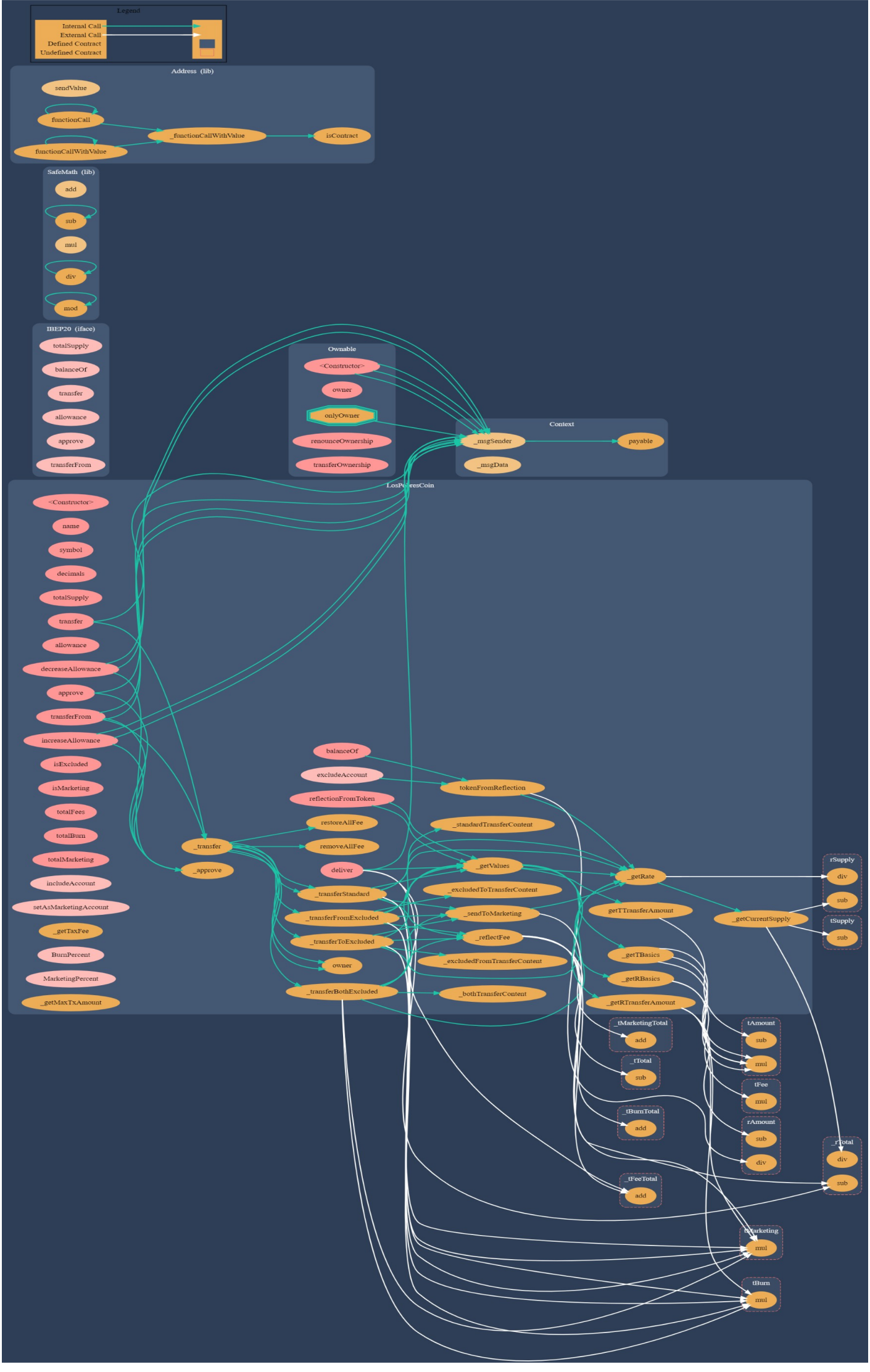
Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
Context	Implementation			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
IBEP20	Interface			
└	totalSupply	External ❗		NO❗
└	balanceOf	External ❗		NO❗
└	transfer	External ❗	🛑	NO❗
└	allowance	External ❗		NO❗
└	approve	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗
SafeMath	Library			
└	add	Internal 🔒		
└	sub	Internal 🔒		
└	sub	Internal 🔒		
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
└	mod	Internal 🔒		

Address	Library			
└	isContract	Internal 🔒		
└	sendValue	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	_functionCallWithValue	Private 🔐	🛑	
Ownable	Implementation	Context		
└	<Constructor>	Public ❗	🛑	NO❗
└	owner	Public ❗		NO❗
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
LosPobresCoin	Implementation	Context, IBEP20, Ownable		
└	<Constructor>	Public ❗	🛑	NO❗
└	name	Public ❗		NO❗
└	symbol	Public ❗		NO❗
└	decimals	Public ❗		NO❗
└	totalSupply	Public ❗		NO❗
└	balanceOf	Public ❗		NO❗
└	transfer	Public ❗	🛑	NO❗
└	allowance	Public ❗		NO❗
└	approve	Public ❗	🛑	NO❗
└	transferFrom	Public ❗	🛑	NO❗
└	increaseAllowance	Public ❗	🛑	NO❗
└	decreaseAllowance	Public ❗	🛑	NO❗
└	isExcluded	Public ❗		NO❗
└	isMarketing	Public ❗		NO❗
└	totalFees	Public ❗		NO❗
└	totalBurn	Public ❗		NO❗
└	totalMarketing	Public ❗		NO❗
└	deliver	Public ❗	🛑	NO❗

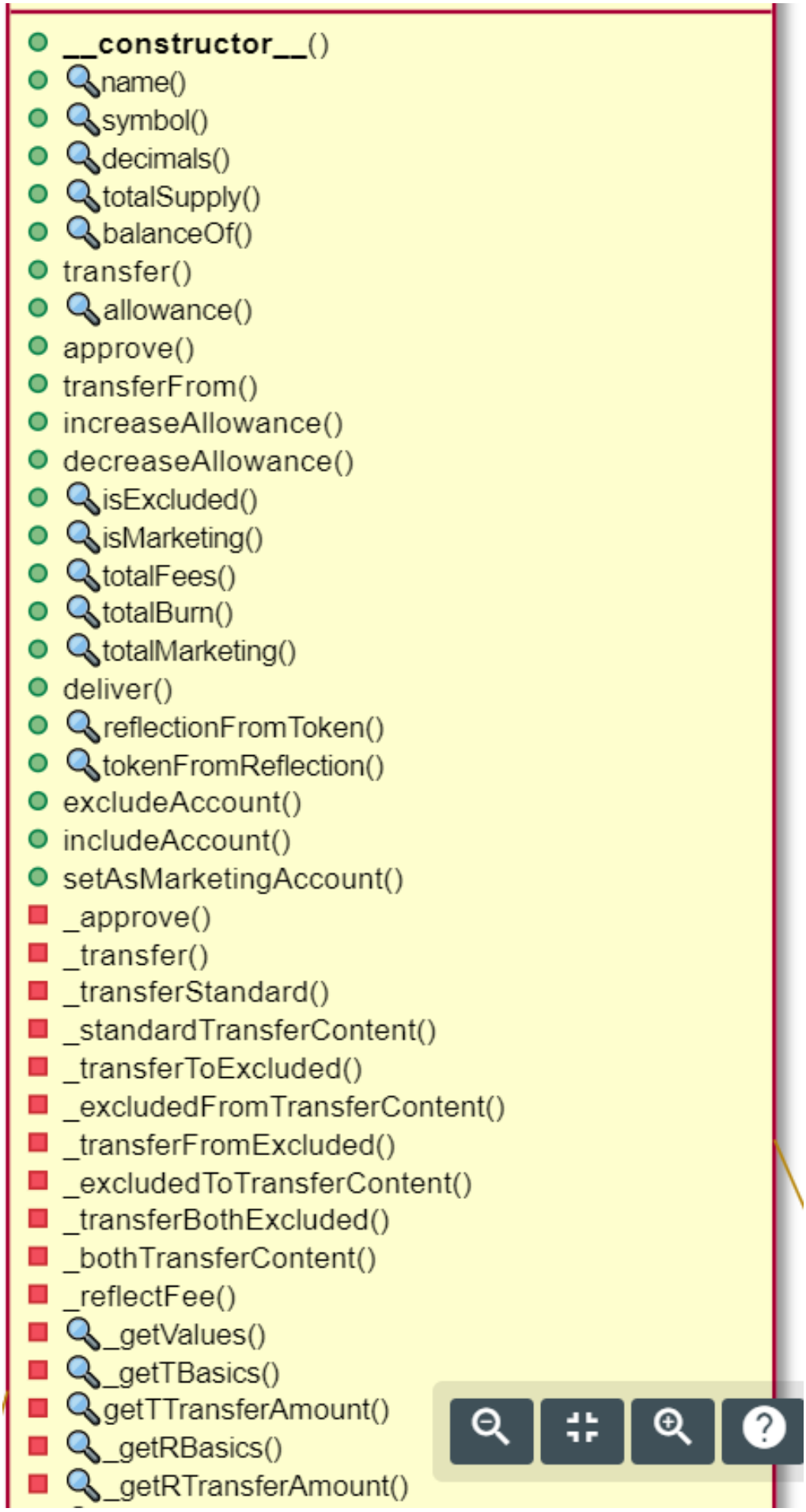
└	reflectionFromToken	Public ❗		NO❗
└	tokenFromReflection	Public ❗		NO❗
└	excludeAccount	External ❗	🛑	onlyOwner
└	includeAccount	External ❗	🛑	onlyOwner
└	setAsMarketingAccount	External ❗	🛑	onlyOwner
└	_approve	Private 🔐	🛑	
└	_transfer	Private 🔐	🛑	
└	_transferStandard	Private 🔐	🛑	
└	_standardTransferContent	Private 🔐	🛑	
└	_transferToExcluded	Private 🔐	🛑	
└	_excludedFromTransferContent	Private 🔐	🛑	
└	_transferFromExcluded	Private 🔐	🛑	
└	_excludedToTransferContent	Private 🔐	🛑	
└	_transferBothExcluded	Private 🔐	🛑	
└	_bothTransferContent	Private 🔐	🛑	
└	_reflectFee	Private 🔐	🛑	
└	_getValues	Private 🔐		
└	_getTBasics	Private 🔐		
└	getTTransferAmount	Private 🔐		
└	_getRBasics	Private 🔐		
└	_getRTransferAmount	Private 🔐		
└	_getRate	Private 🔐		
└	_getCurrentSupply	Private 🔐		
└	_sendToMarketing	Private 🔐	🛑	
└	removeAllFee	Private 🔐	🛑	
└	restoreAllFee	Private 🔐	🛑	
└	_getTaxFee	Private 🔐		
└	BurnPercent	External ❗	🛑	onlyOwner
└	MarketingPercent	External ❗	🛑	onlyOwner
└	_getMaxTxAmount	Private 🔐		

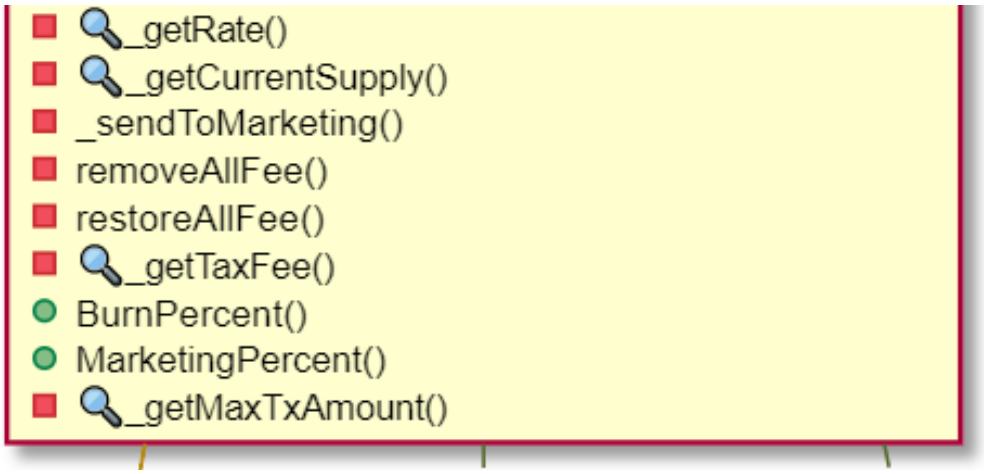
Legend

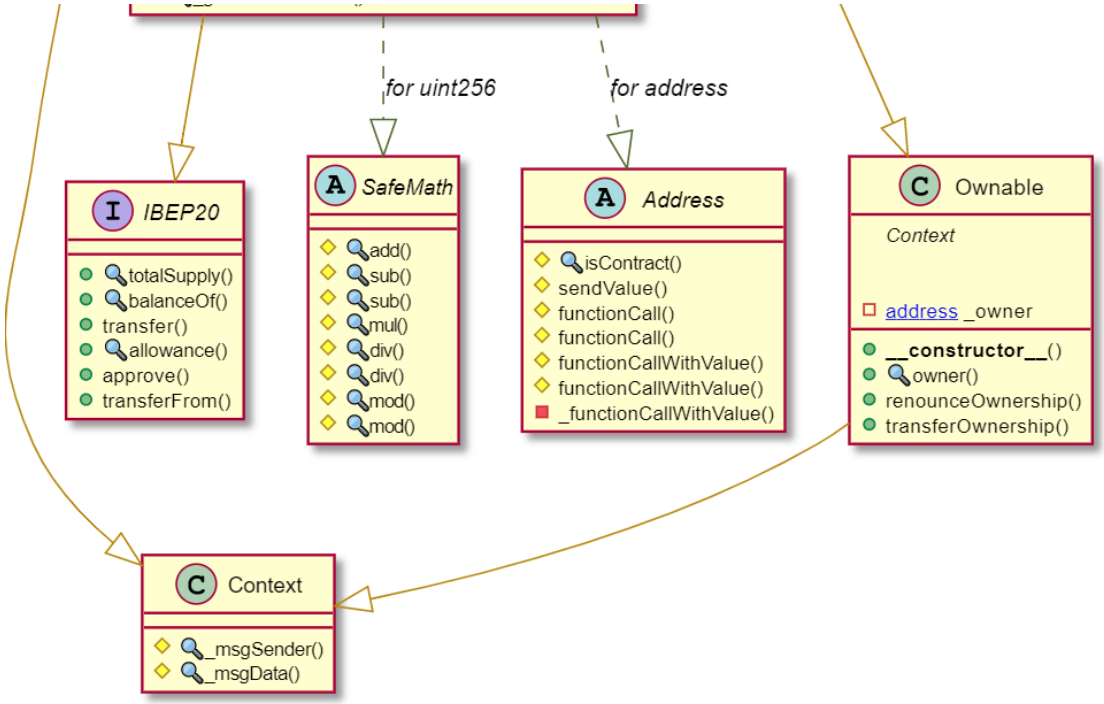
Symbol	Meaning
🛑	Function can modify state
💵	Function is payable


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
Address (lib)
sendValue
functionCall
_functionCallWithValue
isContract
functionCallWithValue
SafeMath (lib)
add
sub
mul
div
mod
IBEP20 (iface)
totalSupply
balanceOf
transfer
allowance
approve
transferFrom
Ownable
<Constructor>
owner
onlyOwner
renounceOwnership
transferOwnership
Context
_msgSender
payable
_msgData
LosPolleresCoin
<Constructor>
name
symbol
decimals
totalSupply
transfer
allowance
decreaseAllowance
approve
transferFrom
increaseAllowance
isExcluded
isMarketing
totalFees
totalBurn
totalMarketing
includeAccount
setAsMarketingAccount
_getTaxFee
BurnPercent
MarketingPercent
_getMaxTxAmount
balanceOf
excludeAccount
reflectionFromToken
tokenFromReflection
restoreAllFee
removeAllFee
_standardTransferContent
_transfer
_approve
deliver
_getValues
_transferStandard
_excludedToTransferContent
_transferFromExcluded
_sendToMarketing
_transferToExcluded
_reflectFee
owner
_excludedFromTransferContent
_transferBothExcluded
_bothTransferContent
_getRate
getTTransferAmount
_getCurrentSupply
_getTBasics
_getRBasics
_getRTransferAmount
rSupply
div
sub
tSupply
sub
_tMarketingTotal
add
tAmount
sub
mul
_tTotal
sub
tFee
mul
_tBurnTotal
add
rAmount
sub
div
_rTotal
div
sub
_tFeeTotal
add
tMarketing
mul
tBurn
mul

Disclai-
mers


__constructor__()
name()
symbol()
decimals()
totalSupply()
balanceOf()
transfer()
allowance()
approve()
transferFrom()
increaseAllowance()
decreaseAllowance()
isExcluded()
isMarketing()
totalFees()
totalBurn()
totalMarketing()
deliver()
reflectionFromToken()
tokenFromReflection()
excludeAccount()
includeAccount()
setAsMarketingAccount()
_approve()
_transfer()
_transferStandard()
_standardTransferContent()
_transferToExcluded()
_excludedFromTransferContent()
_transferFromExcluded()
_excludedToTransferContent()
_transferBothExcluded()
_bothTransferContent()
_reflectFee()
_getValues()
_getTBasics()
getTTransferAmount()
_getRBasics()
_getRTransferAmount()


_getRate()
_getCurrentSupply()
_sendToMarketing()
removeAllFee()
restoreAllFee()
_getTaxFee()
BurnPercent()
MarketingPercent()
_getMaxTxAmount()


for uint256
for address
IBEP20
totalSupply()
balanceOf()
transfer()
allowance()
approve()
transferFrom()
SafeMath
add()
sub()
sub()
mul()
div()
div()
mod()
mod()
Address
isContract()
sendValue()
functionCall()
functionCall()
functionCallWithValue()
functionCallWithValue()
_functionCallWithValue()
Ownable
Context
address _owner
__constructor__()
owner()
renounceOwnership()
transferOwnership()
Context
_msgSender()
_msgData()

Solidity Audit Disclaimer

The smart contracts given for audit have been analyzed in accordance with the best industry practices at the date of this report, in relation to cybersecurity vulnerabilities and issues in smart contract source code, the details of which are disclosed in this report (Source Code); the Source Code compilation, deployment, and functionality (performing the intended functions). The audit makes no statements or warranties on security of the code. It also cannot be considered as a sufficient assessment regarding the utility and safety of the code, bugfree status or any other statements of the contract. While we have done our best in conducting the analysis and producing this report, it is important to note that you should not rely on this report only - we recommend proceeding with several independent audits and a public bug bounty program to ensure security of smart contracts.

Technical Disclaimer

Smart contracts are deployed and executed on blockchain platform. The platform, its programming language, and other software related to the smart contract can have its vulnerabilities that can lead to hacks. Thus, the audit can't guarantee the explicit security of the audited smart contracts.