SOLIDITY AUDIT 

# Smart Contract Solidity Audit

## Audit Details:

**Audited project:** DOGEBLACK AI **( DBLACK AI)**
**Website: www.dogeblackai.net**
**Deployer Address:** 0xF4A0F9E3a0E77973eea14C3E173aa38F995c6c6F
**Blockchain:** Binance Smart Chain - BSC

**29/05/2023**

# 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 **DOGEBLACK AI** (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 Explorer:

**https://bscscan.com/token/0xf4a0f9e3a0e77973eea14c3e173aa38f995c6c6f**

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 a code functionality analysis, manual auditing and automated checks with Mythril and Slither and a few other tools. All issues found during the automated review were manually reviewed and important vulnerabilities are listed in the Audit Overview section.

## Contracts Description Table

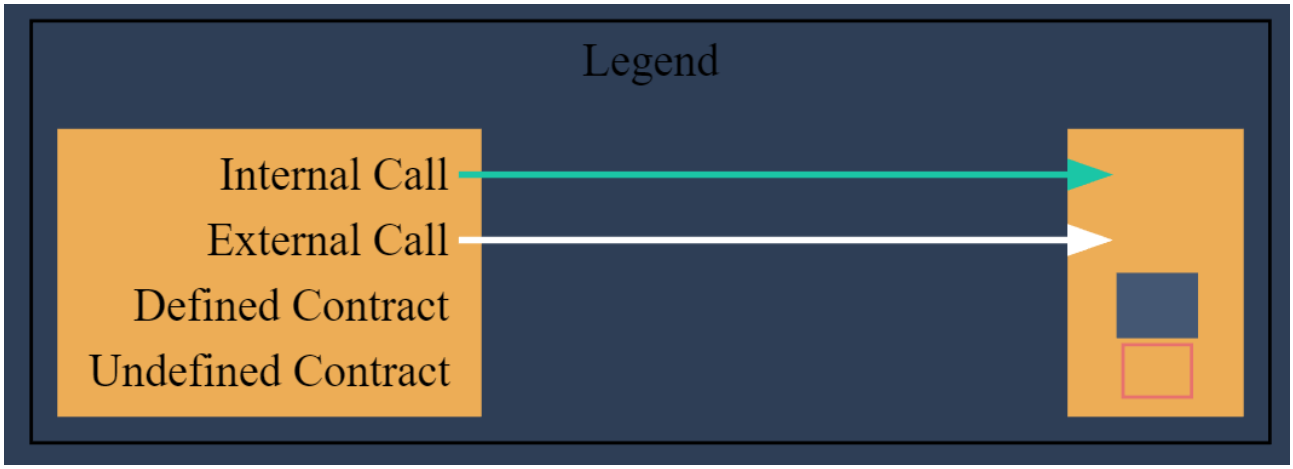
| Contract | Type | Bases | | |
|:---|:---|:---|:---|:---|
| └ | **Function Name** | **Visibility** | **Mutability** | **Modifiers** |
| **SafeMath** | Library | | | |
| └ | add | Internal 🔒 | | |
| └ | sub | Internal 🔒 | | |
| └ | sub | Internal 🔒 | | |
| └ | mul | Internal 🔒 | | |
| └ | div | Internal 🔒 | | |
| └ | div | Internal 🔒 | | |
| **IERC20** | Interface | | | |
| └ | totalSupply | External ❗ | | NO ❗ |
| └ | balanceOf | External ❗ | | NO ❗ |
| └ | transfer | External ❗ | 🔴 | NO ❗ |
| └ | allowance | External ❗ | | NO ❗ |
| └ | approve | External ❗ | 🔴 | NO ❗ |
| └ | transferFrom | External ❗ | 🔴 | NO ❗ |
| **IERC20Metadata** | Interface | IERC20 | | |
| └ | name | External ❗ | | NO ❗ |
| └ | symbol | External ❗ | | NO ❗ |
| └ | decimals | External ❗ | | NO ❗ |
| **Context** | Implementation | | | |
| └ | _msgSender | Internal 🔒 | | |
| └ | _msgData | Internal 🔒 | | |
| **ERC20** | Implementation | Context, IERC20, IERC20Metadata | | |
| └ | <Constructor> | Public ❗ | 🔴 | NO ❗ |
| └ | name | Public ❗ | | NO ❗ |
| └ | symbol | Public ❗ | | NO ❗ |
| └ | decimals | Public ❗ | | NO ❗ |
| └ | totalSupply | Public ❗ | | NO ❗ |
| └ | balanceOf | Public ❗ | | NO ❗ |
| └ | transfer | Public ❗ | 🔴 | NO ❗ |
| └ | allowance | Public ❗ | | NO ❗ |
| └ | approve | Public ❗ | 🔴 | NO ❗ |
| └ | transferFrom | Public ❗ | 🔴 | NO ❗ |
| └ | increaseAllowance | Public ❗ | 🔴 | NO ❗ |
| └ | decreaseAllowance | Public ❗ | 🔴 | NO ❗ |
| └ | _transfer | Internal 🔒 | 🔴 | |
| └ | _mint | Internal 🔒 | 🔴 | |
| └ | _burn | Internal 🔒 | 🔴 | |
| └ | _approve | Internal 🔒 | 🔴 | |
| └ | _beforeTokenTransfer | Internal 🔒 | 🔴 | |
| **ERC20Burnable** | Implementation | Context, ERC20 | | |
| └ | burn | Public ❗ | 🔴 | NO ❗ |
| └ | burnFrom | Public ❗ | 🔴 | NO ❗ |
| **Ownable** | Implementation | Context | | |
| └ | <Constructor> | Public ❗ | 🔴 | NO ❗ |
| └ | owner | Public ❗ | | NO ❗ |
| └ | renounceOwnership | Public ❗ | 🔴 | onlyOwner |
| └ | transferOwnership | Public ❗ | 🔴 | onlyOwner |

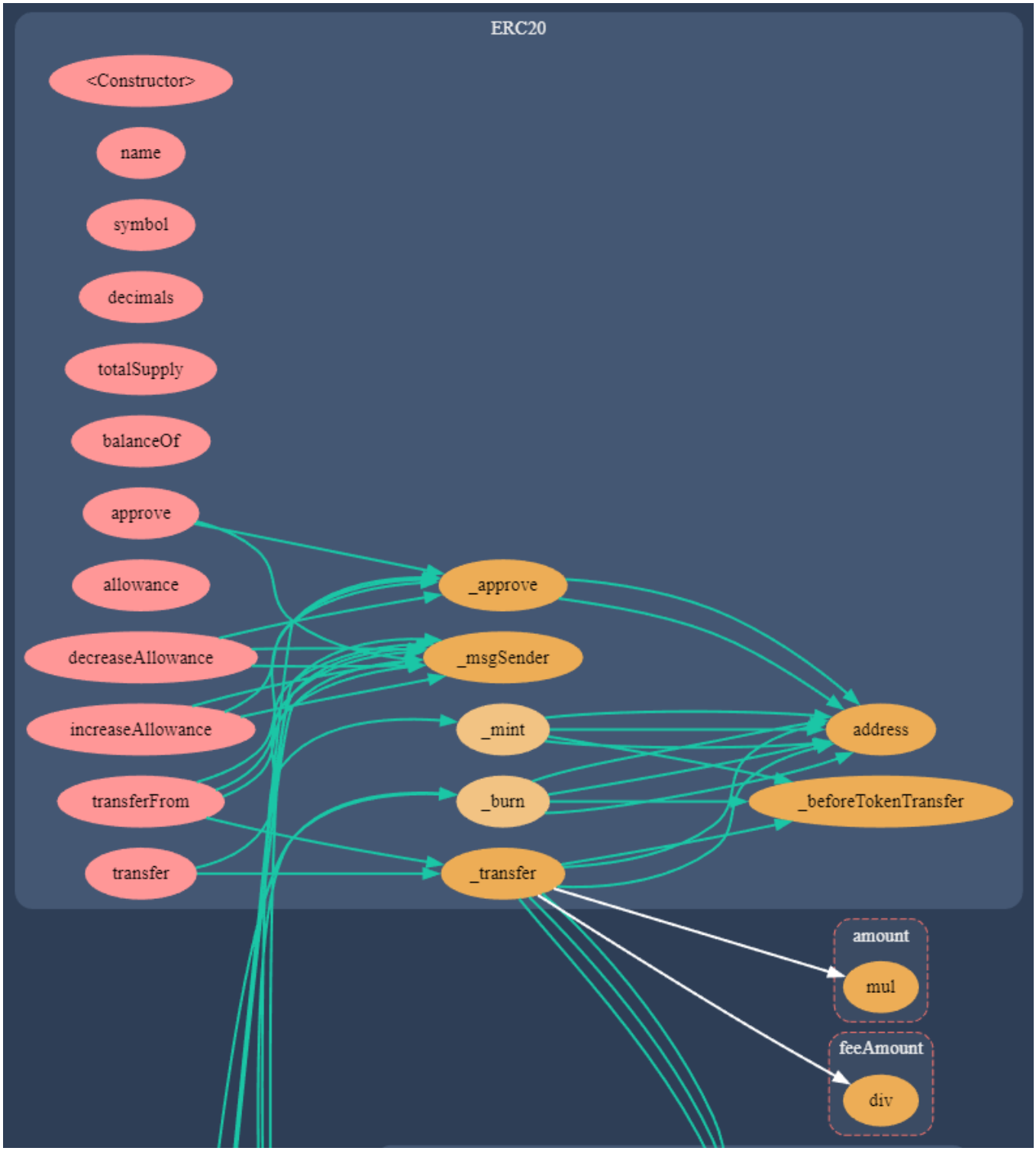
| **DogeBlackAI** | Implementation | ERC20Burnable, Ownable |||
| └ | <Constructor> | Public ❗ | 🔴 | ERC20 |
| └ | _transfer | Internal 🔒 | 🔴 | |
| └ | excludeMultipleAccountsFromFees | External ❗ | 🔴 | onlyOwner |
| └ | enableV2PairFee | External ❗ | 🔴 | onlyOwner |
| └ | updateBUY_FEE | External ❗ | 🔴 | onlyOwner |
| └ | updateSELL_FEE | External ❗ | 🔴 | onlyOwner |
| └ | recoverToken | External ❗ | 🔴 | onlyOwner |
| └ | updateAddresses | External ❗ | 🔴 | onlyOwner |

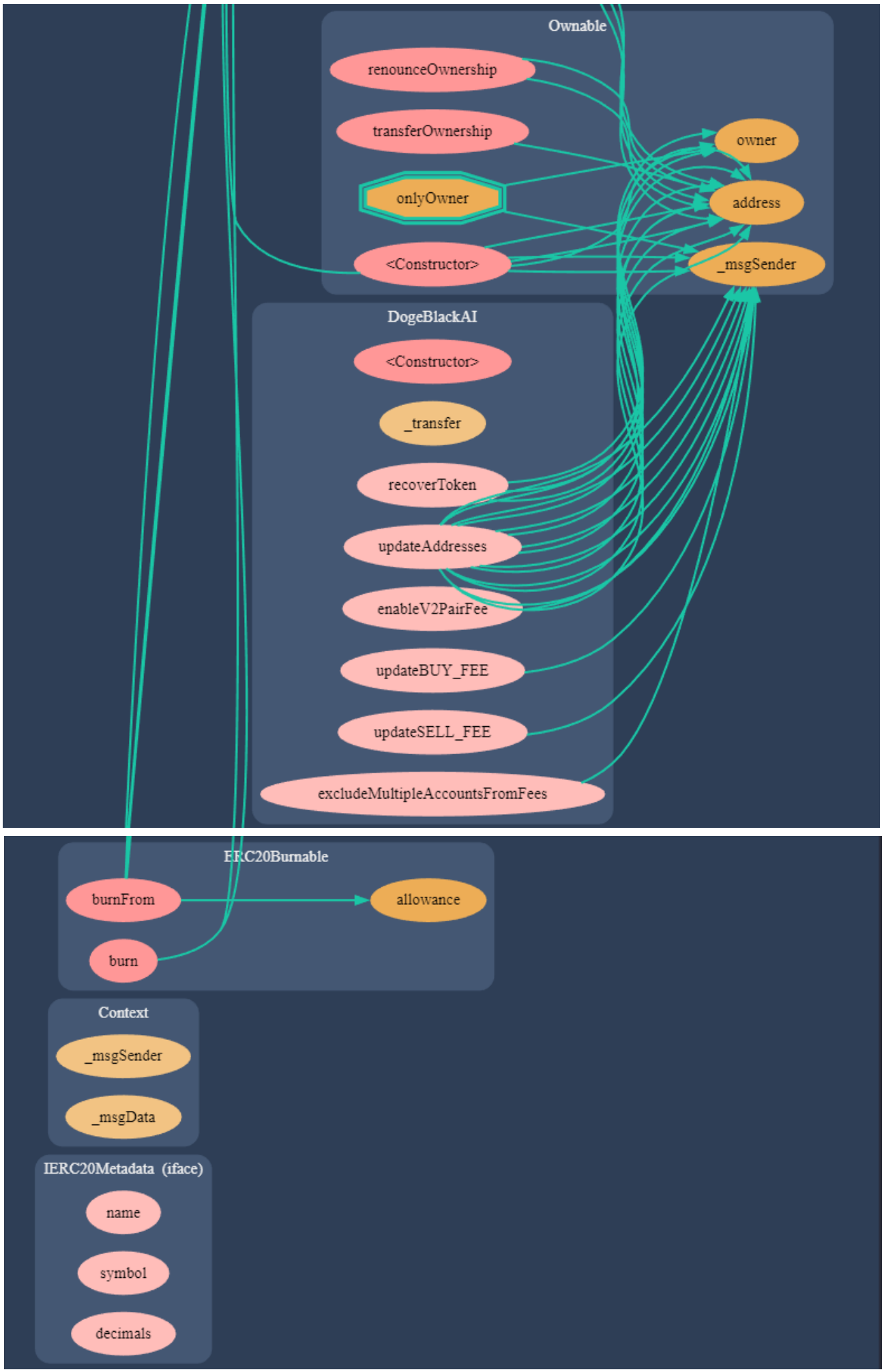
Legend

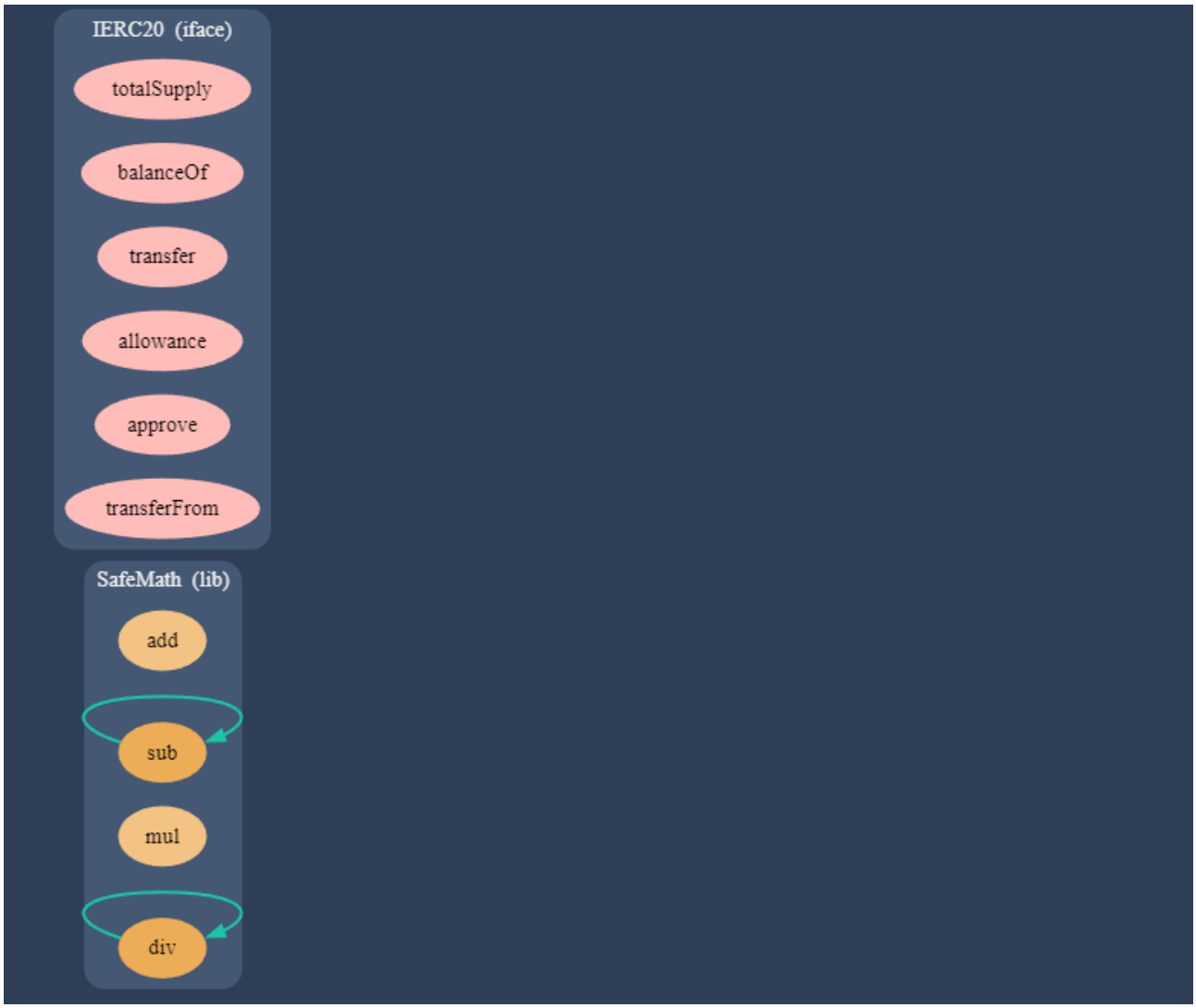
| Symbol | Meaning |
|:---:|---|
| 🔴 | Function can modify state |
| 💵 | Function is payable |

**GRAPH**

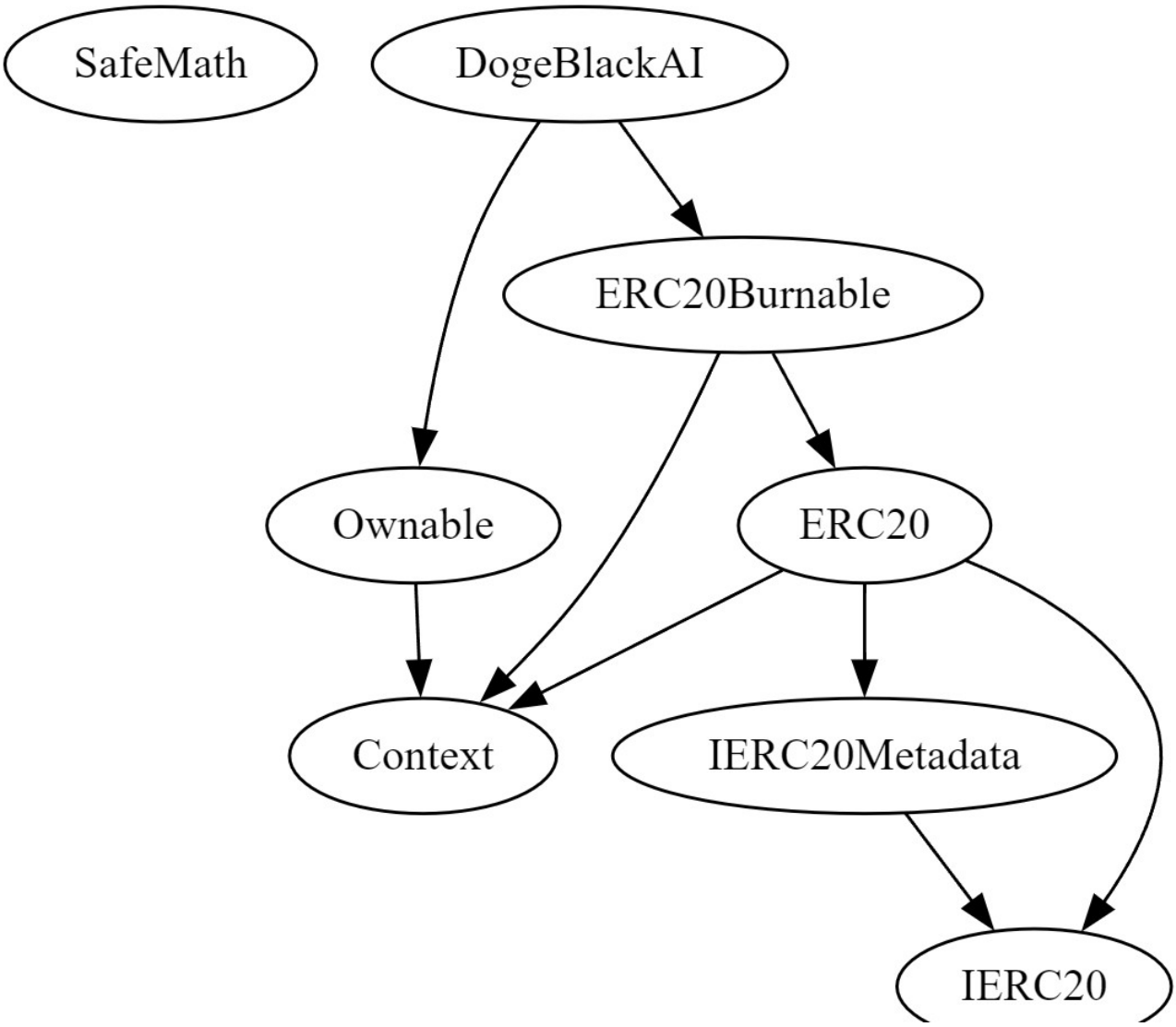



ERC20
<Constructor>
name
symbol
decimals
totalSupply
balanceOf
approve
allowance
decreaseAllowance
increaseAllowance
transferFrom
transfer
_approve
_msgSender
_mint
_burn
_transfer
address
_beforeTokenTransfer
amount
mul
feeAmount
div


Ownable
renounceOwnership
transferOwnership
onlyOwner
<Constructor>
owner
address
_msgSender
DogeBlackAI
<Constructor>
_transfer
recoverToken
updateAddresses
enableV2PairFee
updateBUY_FEE
updateSELL_FEE
excludeMultipleAccountsFromFees
ERC20Burnable
burnFrom
allowance
burn
Context
_msgSender
_msgData
IERC20Metadata (iface)
name
symbol
decimals


IERC20 (iface)
totalSupply
balanceOf
transfer
allowance
approve
transferFrom
SafeMath (lib)
add
sub
mul
div

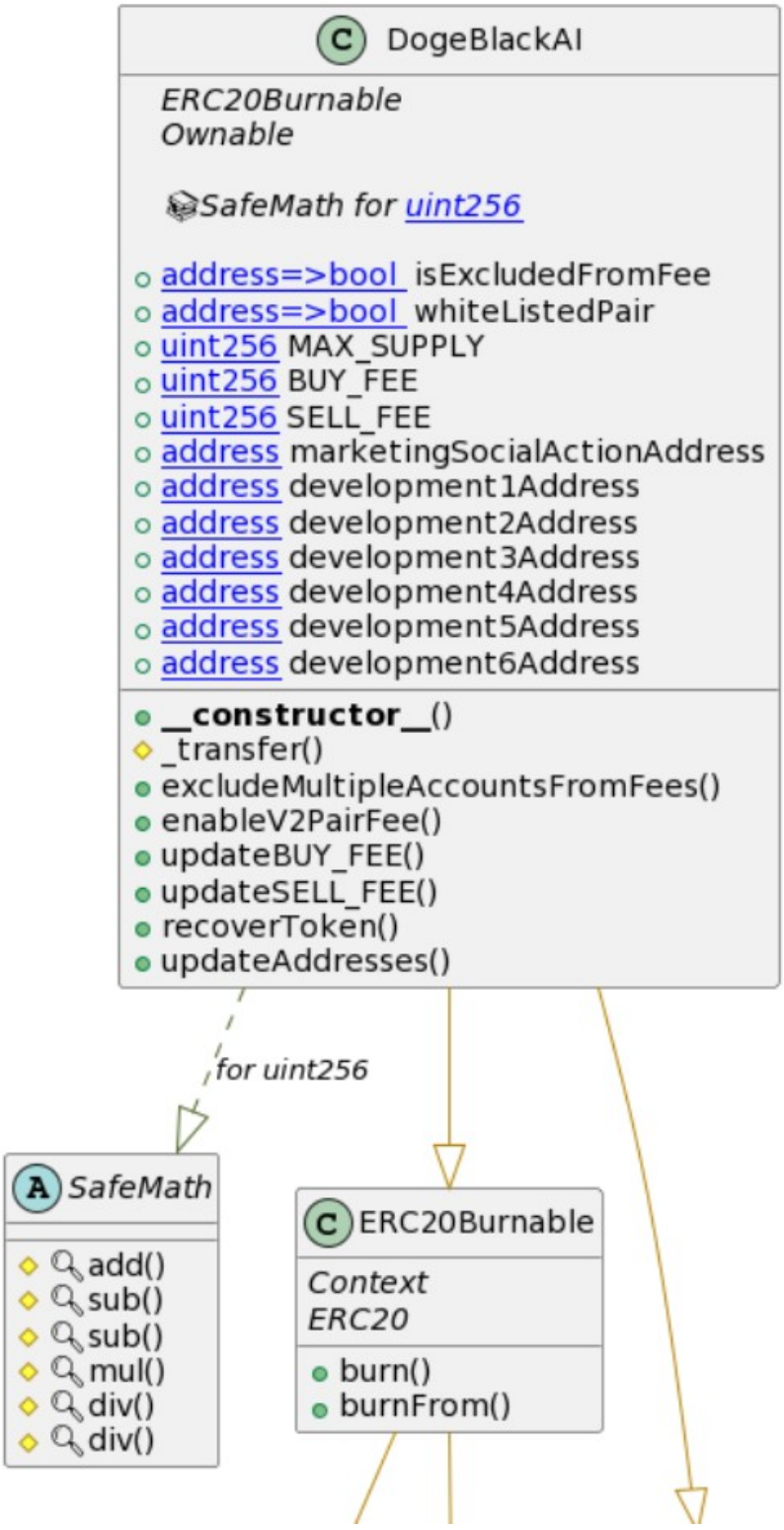
## INHERITANCE



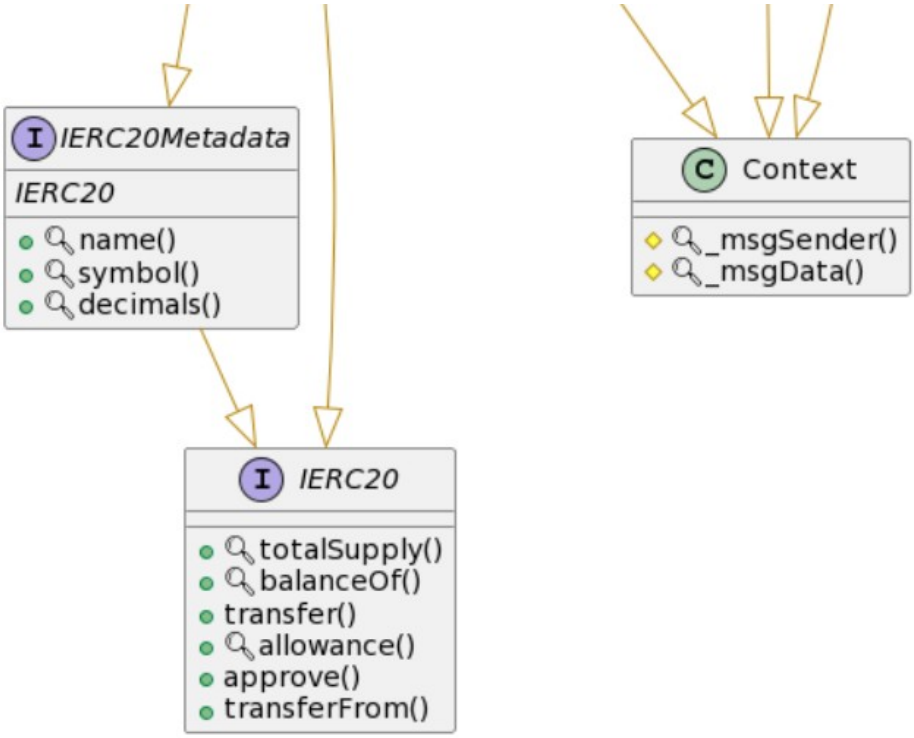
## FUNCSIGS

```
Sighash   |   Function Signature
========================
39509351  =>  increaseAllowance(address,uint256)
95392094  =>  enableV2PairFee(address,bool)
771602f7  =>  add(uint256,uint256)
b67d77c5  =>  sub(uint256,uint256)
e31bdc0a  =>  sub(uint256,uint256,string)
c8a4ac9c  =>  mul(uint256,uint256)
a391c15b  =>  div(uint256,uint256)
b745d336  =>  div(uint256,uint256,string)
18160ddd  =>  totalSupply()
70a08231  =>  balanceOf(address)
a9059cbb  =>  transfer(address,uint256)
dd62ed3e  =>  allowance(address,address)
095ea7b3  =>  approve(address,uint256)
23b872dd  =>  transferFrom(address,address,uint256)
06fdde03  =>  name()
95d89b41  =>  symbol()
313ce567  =>  decimals()
119df25f  =>  _msgSender()
8b49d47e  =>  _msgData()
a457c2d7  =>  decreaseAllowance(address,uint256)
30e0789e  =>  _transfer(address,address,uint256)
4e6ec247  =>  _mint(address,uint256)
6161eb18  =>  _burn(address,uint256)
104e81ff  =>  _approve(address,address,uint256)
cad3be83  =>  _beforeTokenTransfer(address,address,uint256)
42966c68  =>  burn(uint256)
79cc6790  =>  burnFrom(address,uint256)
8da5cb5b  =>  owner()
715018a6  =>  renounceOwnership()
f2fde38b  =>  transferOwnership(address)
c492f046  =>  excludeMultipleAccountsFromFees(address[],bool)
bc30ed06  =>  updateBUY_FEE(uint256)
6c8c44d0  =>  updateSELL_FEE(uint256)
9be65a60  =>  recoverToken(address)
8c200c4f  =>  updateAddresses(address,address,address,address,address,address,address)
```

**UML**


DogeBlackAI
ERC20Burnable
Ownable
SafeMath for uint256
address=>bool isExcludedFromFee
address=>bool whiteListedPair
uint256 MAX_SUPPLY
uint256 BUY_FEE
uint256 SELL_FEE
address marketingSocialActionAddress
address development1Address
address development2Address
address development3Address
address development4Address
address development5Address
address development6Address
__constructor__()
_transfer()
excludeMultipleAccountsFromFees()
enableV2PairFee()
updateBUY_FEE()
updateSELL_FEE()
recoverToken()
updateAddresses()
for uint256
SafeMath
add()
sub()
sub()
mul()
div()
div()
ERC20Burnable
Context
ERC20
burn()
burnFrom()


Ownable
Context
address _owner
__constructor__()
owner()
renounceOwnership()
transferOwnership()
ERC20
Context
IERC20
IERC20Metadata
address=>uint256 _balances
address=>mapping address=>uint256 _allowances
uint256 _totalSupply
string _name
string _symbol
__constructor__()
name()
symbol()
decimals()
totalSupply()
balanceOf()
transfer()
allowance()
approve()
transferFrom()
increaseAllowance()
decreaseAllowance()
_transfer()
_mint()
_burn()
_approve()
_beforeTokenTransfer()




**OVERALL**

1. Contract Inheritance:
    - The DogeBlackAI contract inherits from the ERC20Burnable and Ownable contracts.
    - The ERC20Burnable contract provides functionality for burning tokens from the sender's account.
    - The Ownable contract implements an ownership model, allowing the contract to have a single owner with special administrative powers.
2. SafeMath Library:
    - The contract makes use of the SafeMath library to prevent potential overflow and underflow issues during mathematical calculations.
3. Mappings:
    - isExcludedFromFee: It is a mapping that associates addresses with boolean values indicating whether an address is excluded from fees.
    - whiteListedPair: It is a mapping that associates address pairs with boolean values indicating whether a pair of addresses is whitelisted for fees.

4. State Variables:

- MAX_SUPPLY: It is an immutable state variable that stores the maximum supply of tokens defined at the contract creation.
- BUY_FEE and SELL_FEE: They are state variables representing the fees applied to buy and sell transactions of tokens, respectively.
- Contract Addresses for Marketing and Development Actions: They are state variables that store the addresses of contracts associated with marketing and development activities.

5. Constructor:

- The contract has a constructor that is called when the contract is created.
- The constructor takes the parameters __name, __symbol, _maxSupply, and _initialSupply.
- It checks if the initial supply (_initialSupply) is less than or equal to the maximum supply (_maxSupply).
- It sets the maximum supply (MAX_SUPPLY), adds the contract owner and the contract itself to the list of fee exemptions (isExcludedFromFee), and if the initial supply is greater than zero, mints tokens to the sender's address.

6. _transfer Function:

- It is an internal function that performs the transfer of tokens.
- It checks if the sender and recipient addresses are non-zero and if the sender's balance is sufficient for the transfer.
- It calculates the fee based on the whitelist of pairs (whiteListedPair) and fee exemptions (isExcludedFromFee).
- It divides the fee amount into seven equal parts and distributes them to the marketing and development addresses.
- It updates the balances of the sender and recipient, deducting the fee.
- It emits the transfer event.

7. Management Functions:

- excludeMultipleAccountsFromFees: Allows the contract owner to exclude or include multiple addresses in the fee exemptions list.
- enableV2PairFee: Allows the contract owner to enable or disable the fee for a specific pair of addresses.

- updateBUY_FEE and updateSELL_FEE: Allow the contract owner to update the buy and sell fees, respectively.
- recoverToken: Allows the contract owner to recover tokens accidentally sent to the contract.
- updateAddresses: Allows the contract owner to update the addresses associated with marketing and development activities.

Overall, this contract provides the basic functionality for an ERC-20 token and includes access control through the `Ownable` contract. It can be used as a starting point for creating a custom ERC-20 token with additional features.

**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.