smart state

Web3 security easier than ever

locus

Locus Finance
Smart contract audit report

November 27, 2023

Table of contents

Table of contents

Table of contents

Methodology

- Manual code analysis
- Best code practices
- ERC20/BEP20 compliance (if applicable)
- Locked ether
- Pool Asset Security (backdoors in the underlying ERC-20)
- FA2 compliance (if applicable)
- Logical bugs & code logic issues
- Error handling issues
- General Denial Of Service(DOS)
- Cryptographic errors
- Weak PRNG / Random number generators issues
- Protocol and header parsing errors
- Private data leaks
- Using components with known vulnerabilities
- Unchecked call return method
- Code with no effects
- Unused vars
- Use of deprecated functions
- Authorization issues
- Re-entrancy
- Arithmetic Overflows / Underflows
- Hidden Malicious Code
- External Contract Referencing
- Short Address/Parameter Attack
- Race Conditions / Front Running
- Uninitialized Storage Pointers
- Floating Points and Precision
- Signatures Replay

Summary

This audit encompasses the examination of contracts of the staking system and the internal logic of the Locus Finance project - a decentralized asset management platform offering tokenized profit-generating vaults of strategies.

Disclaimer

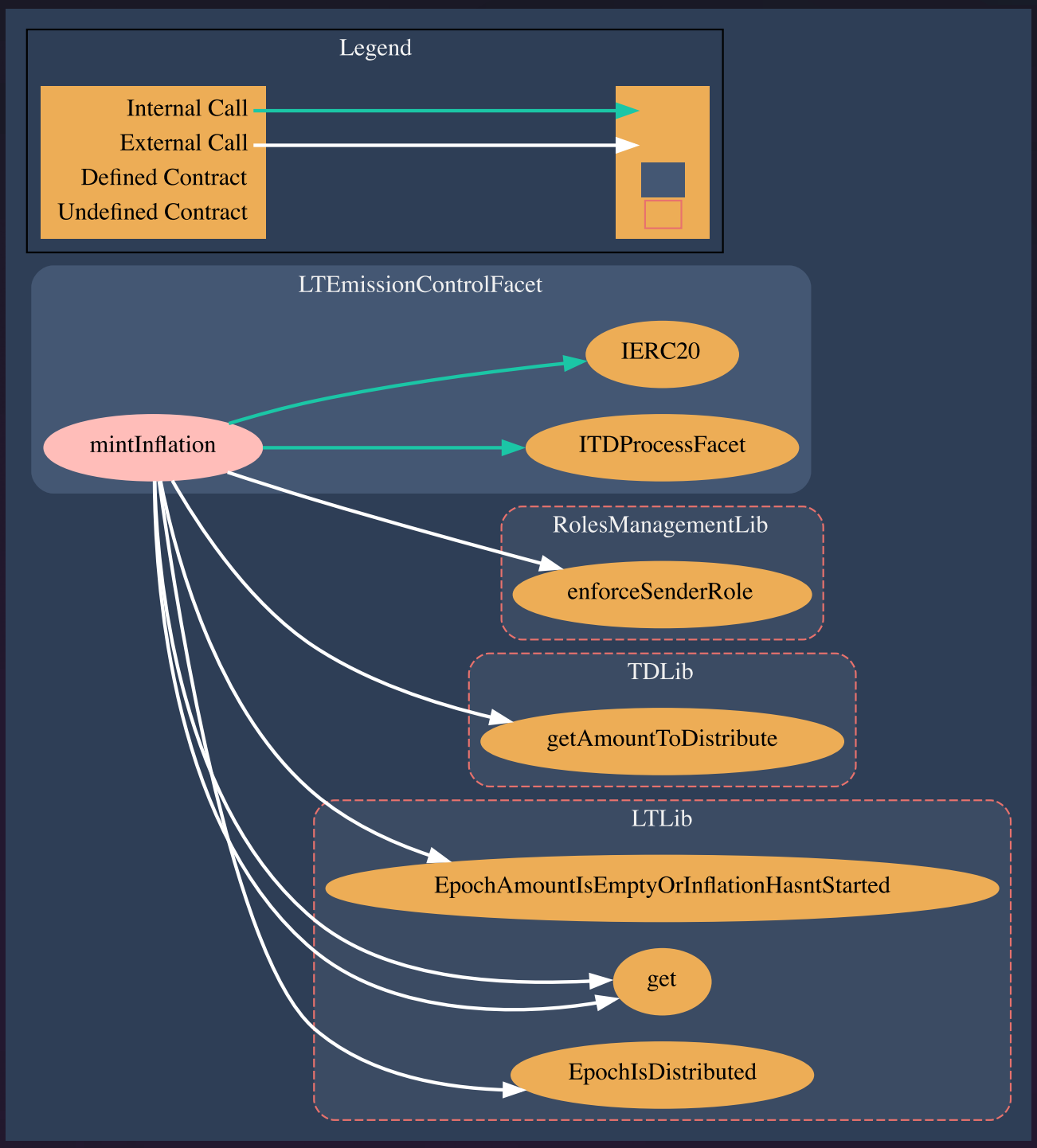
An audit does not provide any warranties regarding the code security. We presume that a single audit cannot be considered totally sufficient and always recommend several independent audits and a public bug bounty program to ensure code security.
Please do not consider this report as investment and / or financial advice of any kind.

Vulnerabilities found by type

Info	0
Warning	0
Warning	0
Total	0

1.1 Structure of contract:

LTEmissionControlFacet.sol



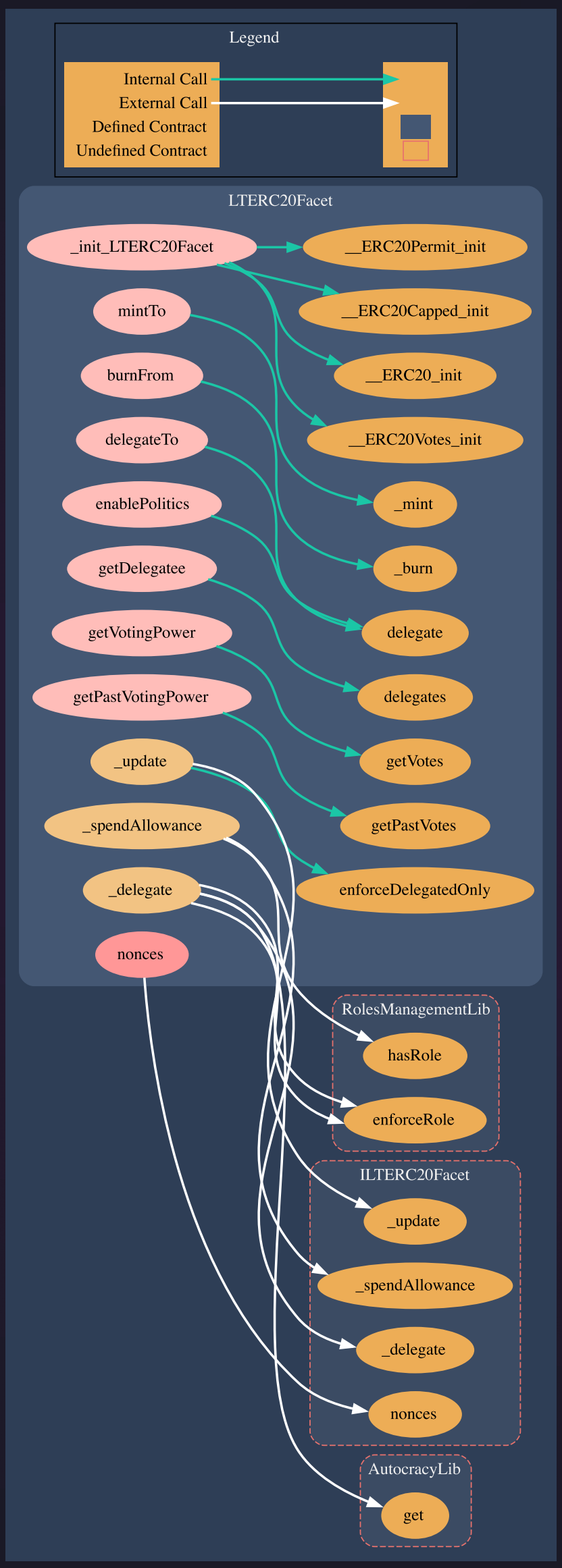
pic.1.1 LTEmissionControlFacet.sol

1.2 LTEmissionControlFacet.sol contract methods analysis:

mintInflation()
Vulnerabilities not detected

2.1 Structure of contract:

LTERC20Facet.sol



pic.2.1 LTERC20Facet.sol

2.2 LTERC20Facet.sol contract methods analysis:

`_init_LTERC20Facet()`
Vulnerabilities not detected

```
mintTo(
        address account,
        uint256 amount
    )
```

Vulnerabilities not detected

```
burnFrom(
        address account,
        uint256 amount
    )
```

Vulnerabilities not detected

```
_spendAllowance(
        address owner,
        address spender,
        uint256 value
    )
```

Vulnerabilities not detected

`_delegate(address account, address delegatee)`
Vulnerabilities not detected

2.2 LTERC20Facet.sol contract methods analysis:

_update(**address from,** **address to,** **uint256 value** **)**
Vulnerabilities not detected

nonces(**address owner** **)**
Vulnerabilities not detected

delegateTo(address delegatee)
Vulnerabilities not detected

getDelegatee(address account)
Vulnerabilities not detected

enablePolitics()
Vulnerabilities not detected

getVotingPower(**address account** **)**
Vulnerabilities not detected

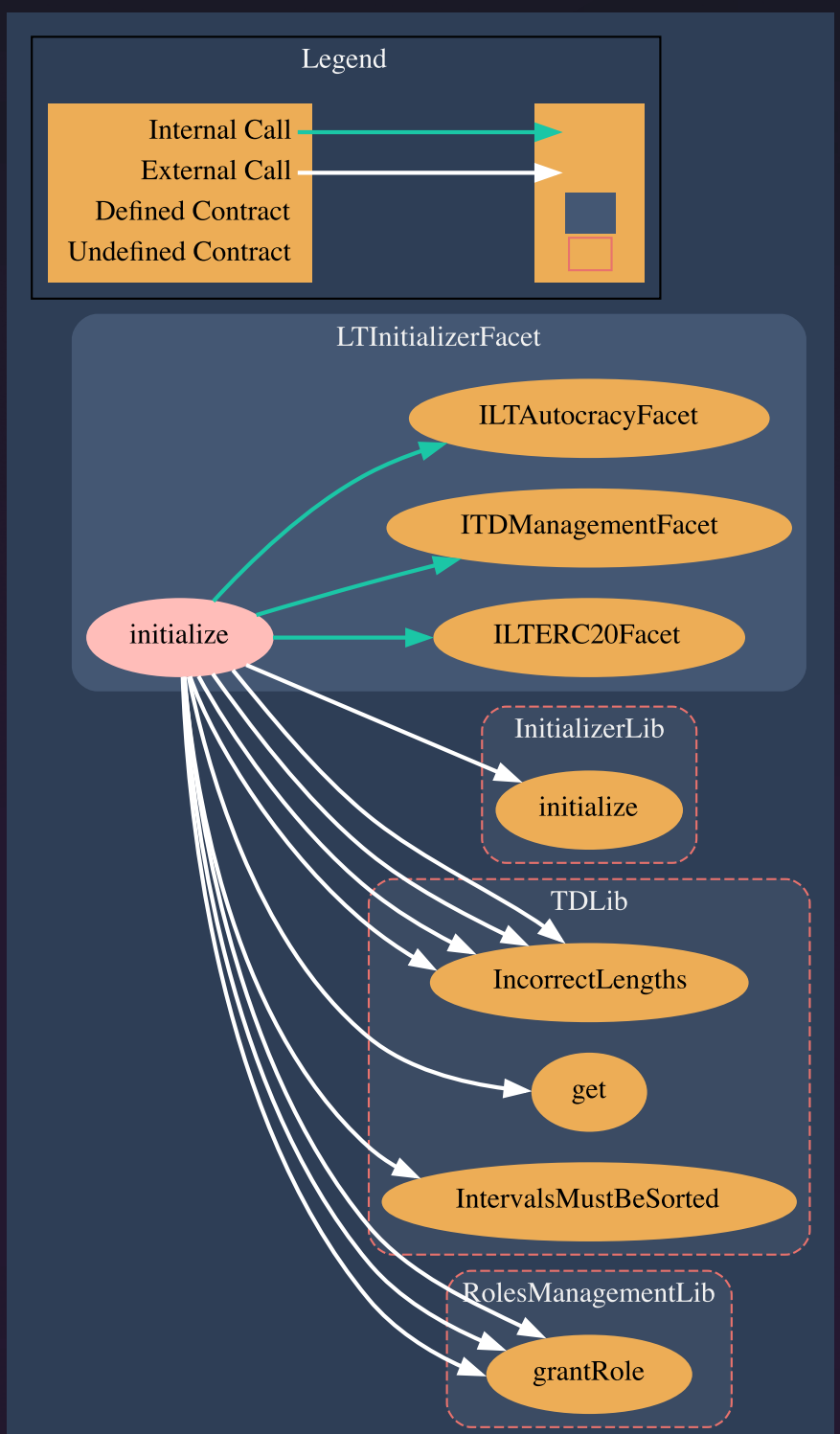
2.2 LTERC20Facet.sol contract methods analysis:

```
getPastVotingPower(
        address account,
        uint256 timepoint
    )
```

Vulnerabilities not detected

3.1 Structure of contract:

LTInitializerFacet.sol



pic.3.1 LTInitializerFacet.sol

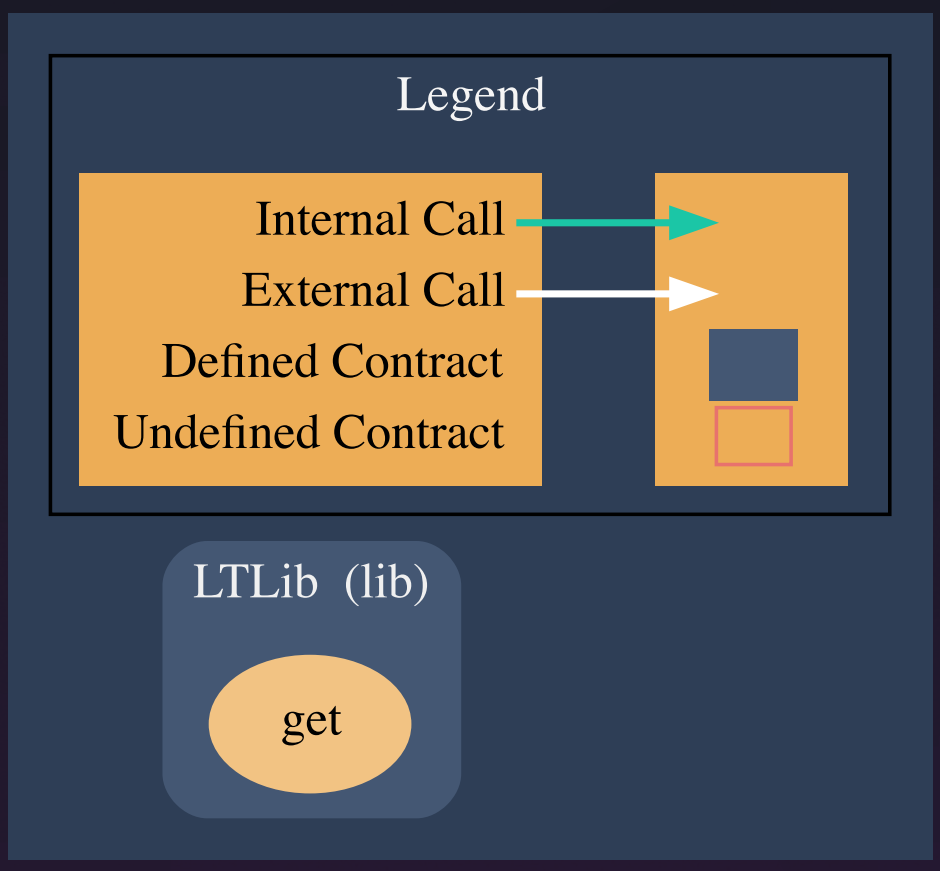
3.2 LTInitializerFacet.sol contract methods analysis:

```
initialize(
        address owner,
        address[] calldata distributionReceivers,
        uint256[] calldata distributionReceiversShares,
        uint32[] calldata distributionDurationPoints,
        uint256[][] calldata amountsPerEpochs
    )
```

Vulnerabilities not detected

4.1 Structure of contract:

LTLib.sol



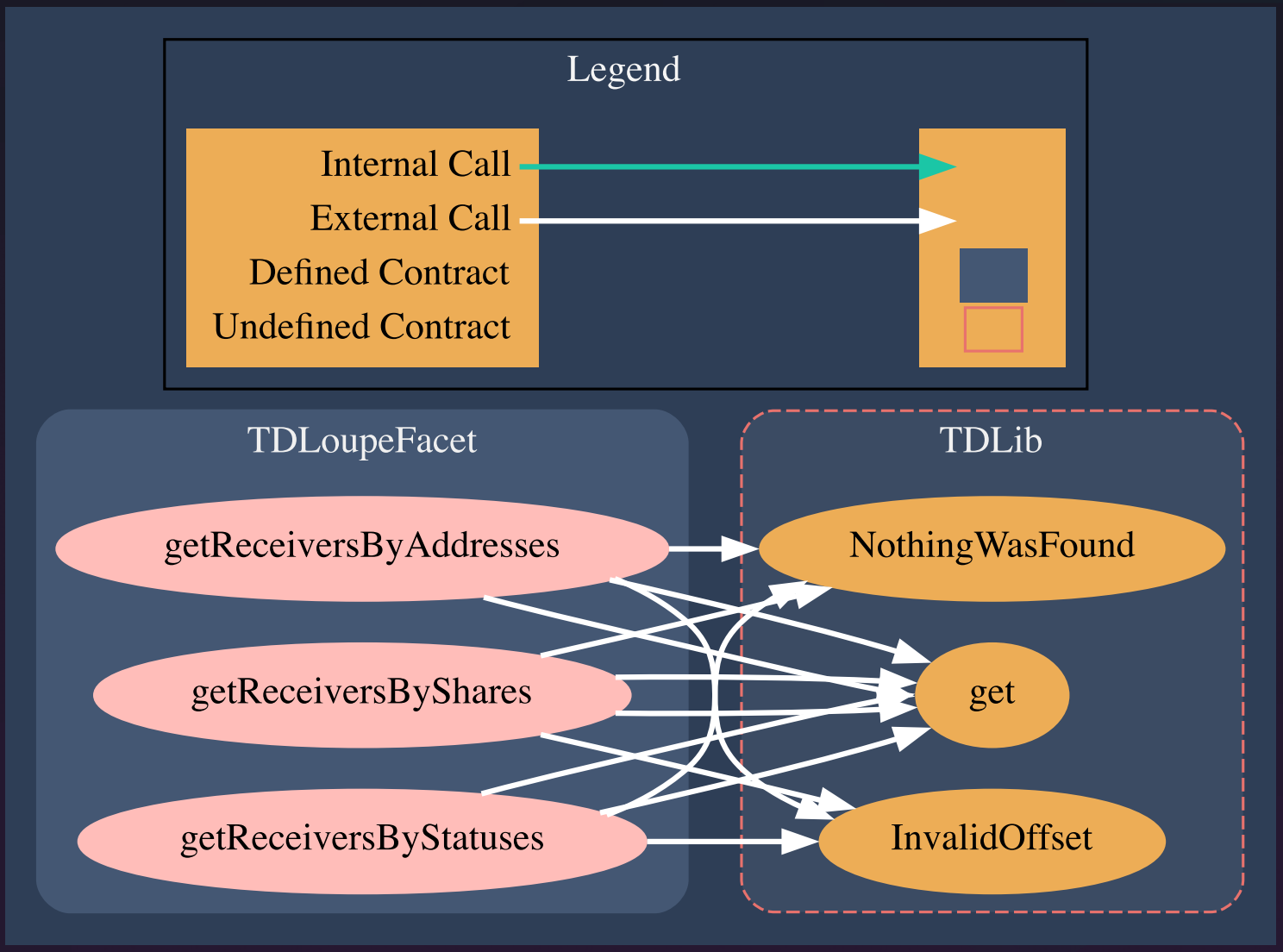
pic.4.1 LTLib.sol

4.2 LTLib.sol contract methods analysis:

get()
Vulnerabilities not detected

5.1 Structure of contract:

TDLoupeFacet.sol



pic.5.1 TDLoupeFacet.sol

5.2 TDLoupeFacet.sol contract methods analysis:

```
function getReceiversByAddresses(
        uint256 offset,
        uint256 windowSize,
        address[] memory addresses
    )
```

Vulnerabilities not detected

```
function getReceiversByShares(
        uint256 offset,
        uint256 windowSize,
        uint256[] memory shares
    )
```

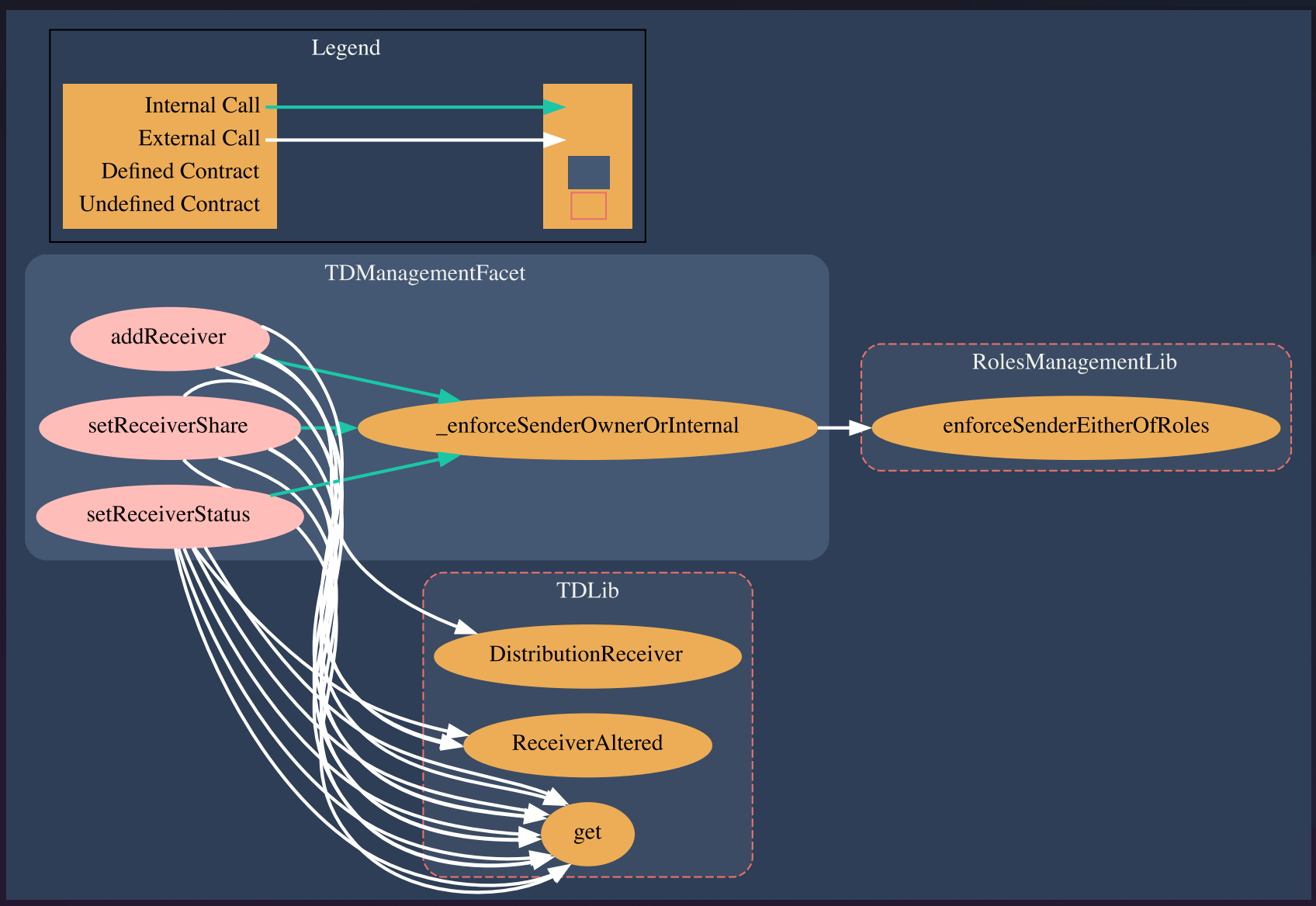
Vulnerabilities not detected

```
function getReceiversByStatuses(
        uint256 offset,
        uint256 windowSize,
        bool[] memory statuses
    )
```

Vulnerabilities not detected

6.1 Structure of contract:

TDManagementFacet.sol



pic.6.1 TDManagementFacet.sol

6.2 TDManagementFacet.sol contract methods analysis:

```
function addReceiver(
        address distributionReceiver,
        uint256 share,
        bool status
    )
```

Vulnerabilities not detected

```
function setReceiverShare(
        address distributionReceiver,
        uint256 share
    )
```

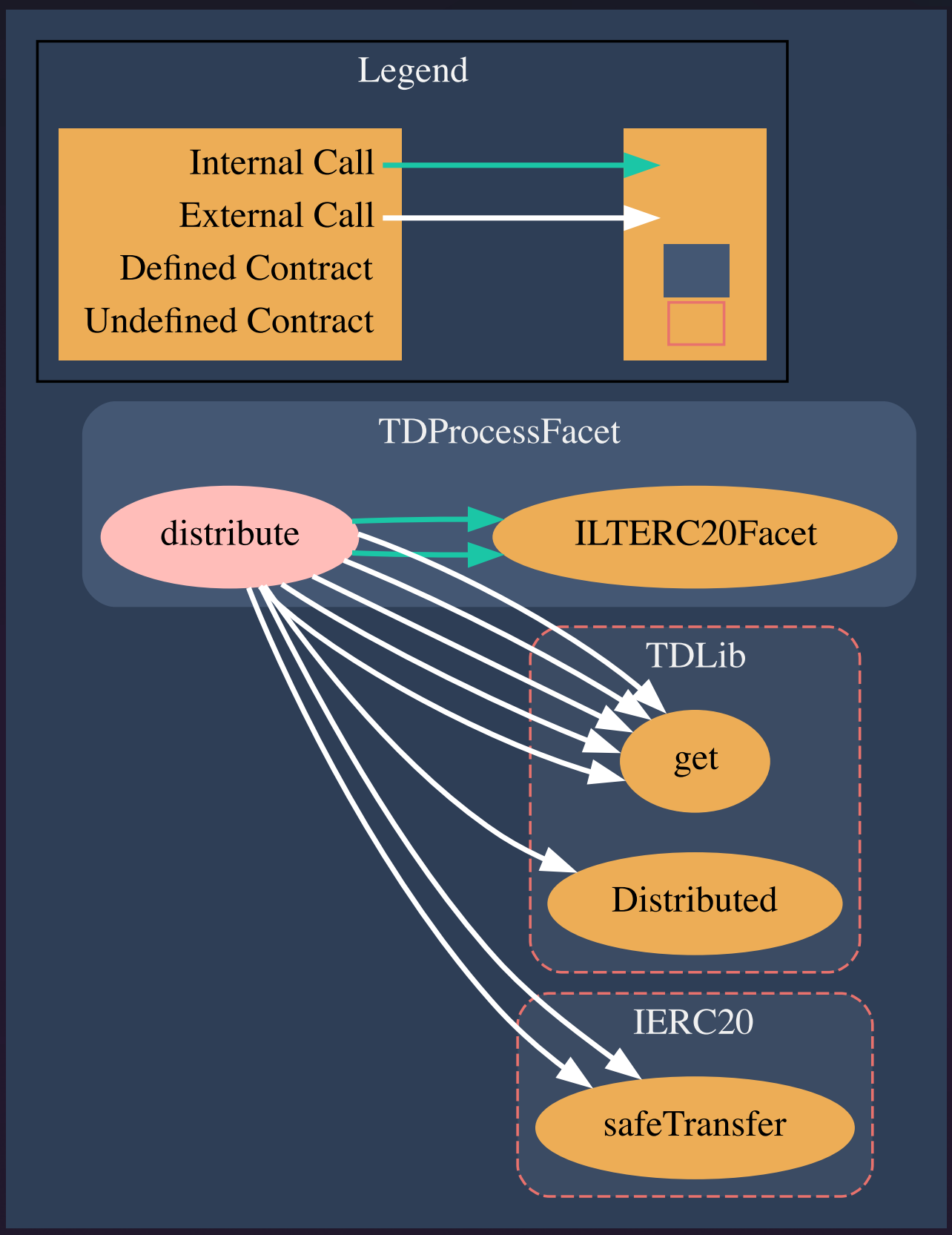
Vulnerabilities not detected

```
function setReceiverStatus(
        address distributionReceiver,
        bool status
    )
```

Vulnerabilities not detected

7.1 Structure of contract:

TDProcessFacet.sol



pic.7.1 TDProcessFacet.sol

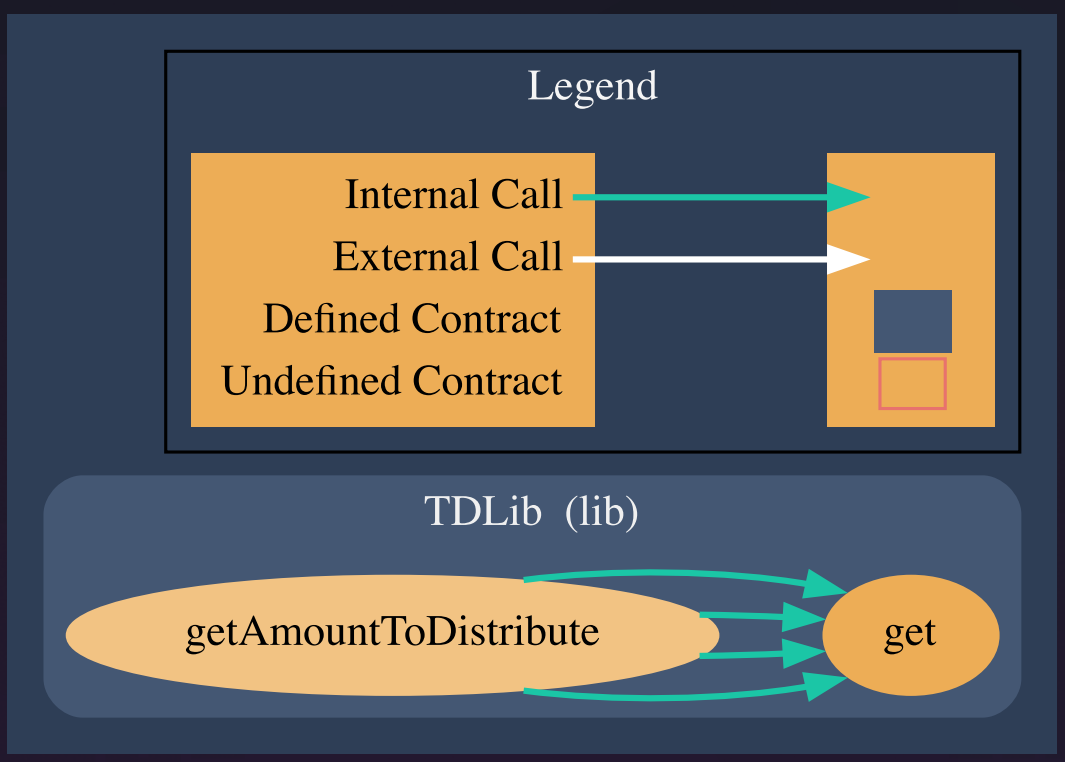
7.2 TDProcessFacet.sol contract methods analysis:

```
function distribute(
        uint256 amount,
        IERC20 token
    )
```

Vulnerabilities not detected

8.1 Structure of contract:

TDLib.sol



pic.8.1 TDLib.sol

8.2 TDLib.sol contract methods analysis:

`get()`
Vulnerabilities not detected

```
function getAmountToDistribute(
        address entity
    )
```

Vulnerabilities not detected

9. MidasClaim.sol contract methods analysis:

setNewMerkleRoot(bytes32 _newRoot)
Vulnerabilities not detected

dsetNewToken(address _newToken)
Vulnerabilities not detected

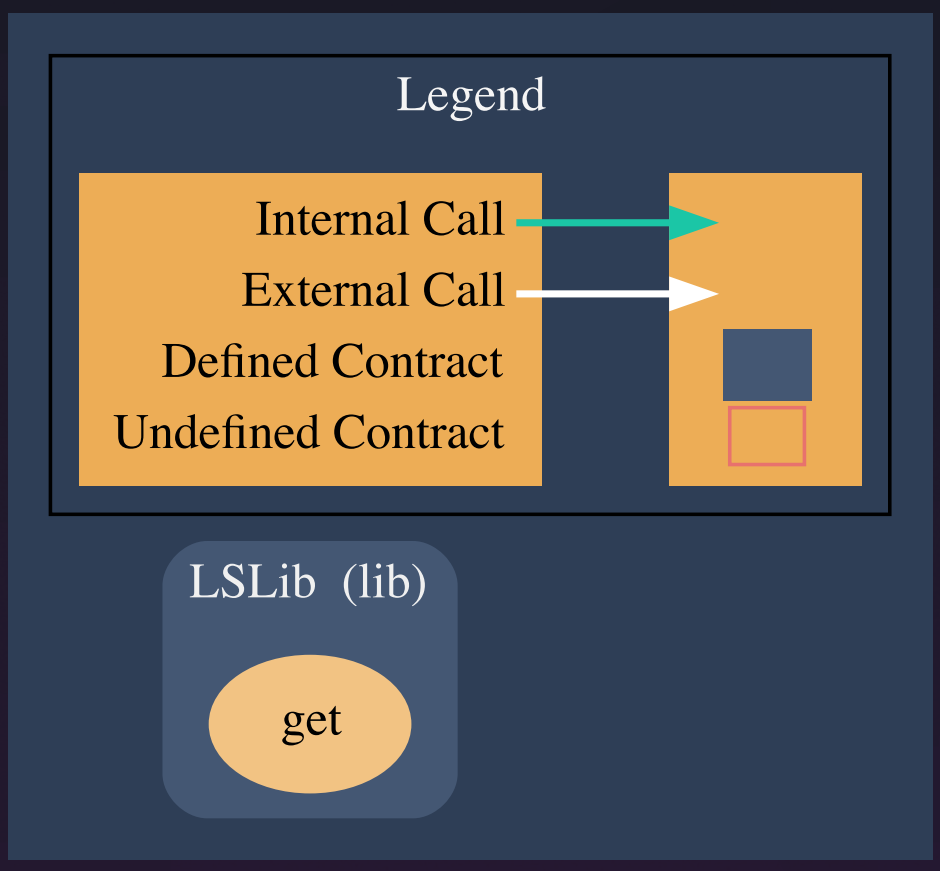
setNewTreasury(address _newTreasury)
Vulnerabilities not detected

claim(address account, uint256 amount, bytes32[] calldata merkleProof)
Vulnerabilities not detected
TOKEN FLOW: Tokens out, public

emergencyExit()
Vulnerabilities not detected

10.1 Structure of contract:

LSLib.sol



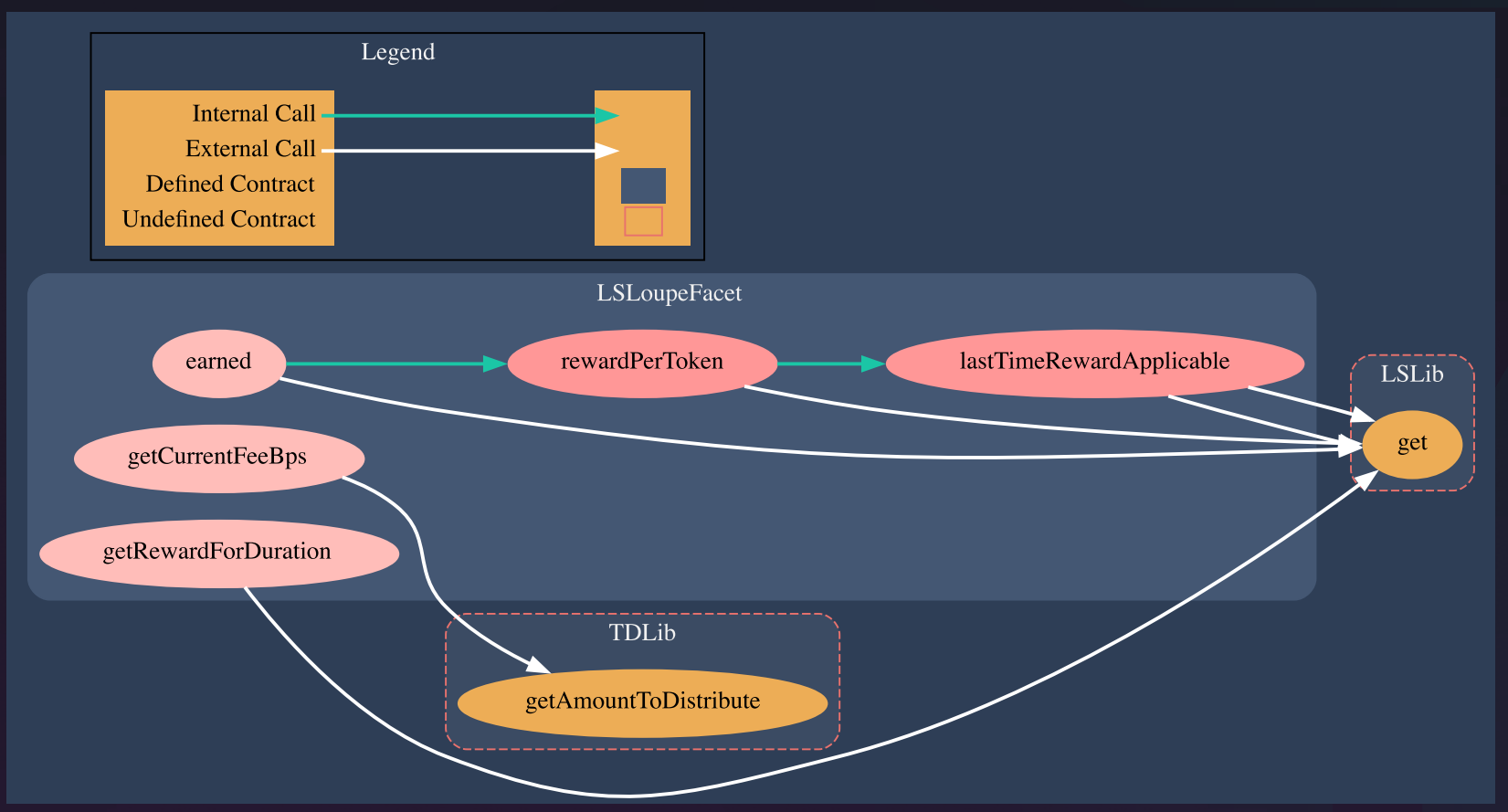
pic.10.1 LSLib.sol

10.2 LSLib.sol contract methods analysis:

get()
Vulnerabilities not detected

11.1 Structure of contract:

LSLoupeFacet.sol



pic.11.1 LSLoupeFacet.sol

11.2 LSLoupeFacet.sol contract methods analysis:

lastTimeRewardApplicable()
Vulnerabilities not detected

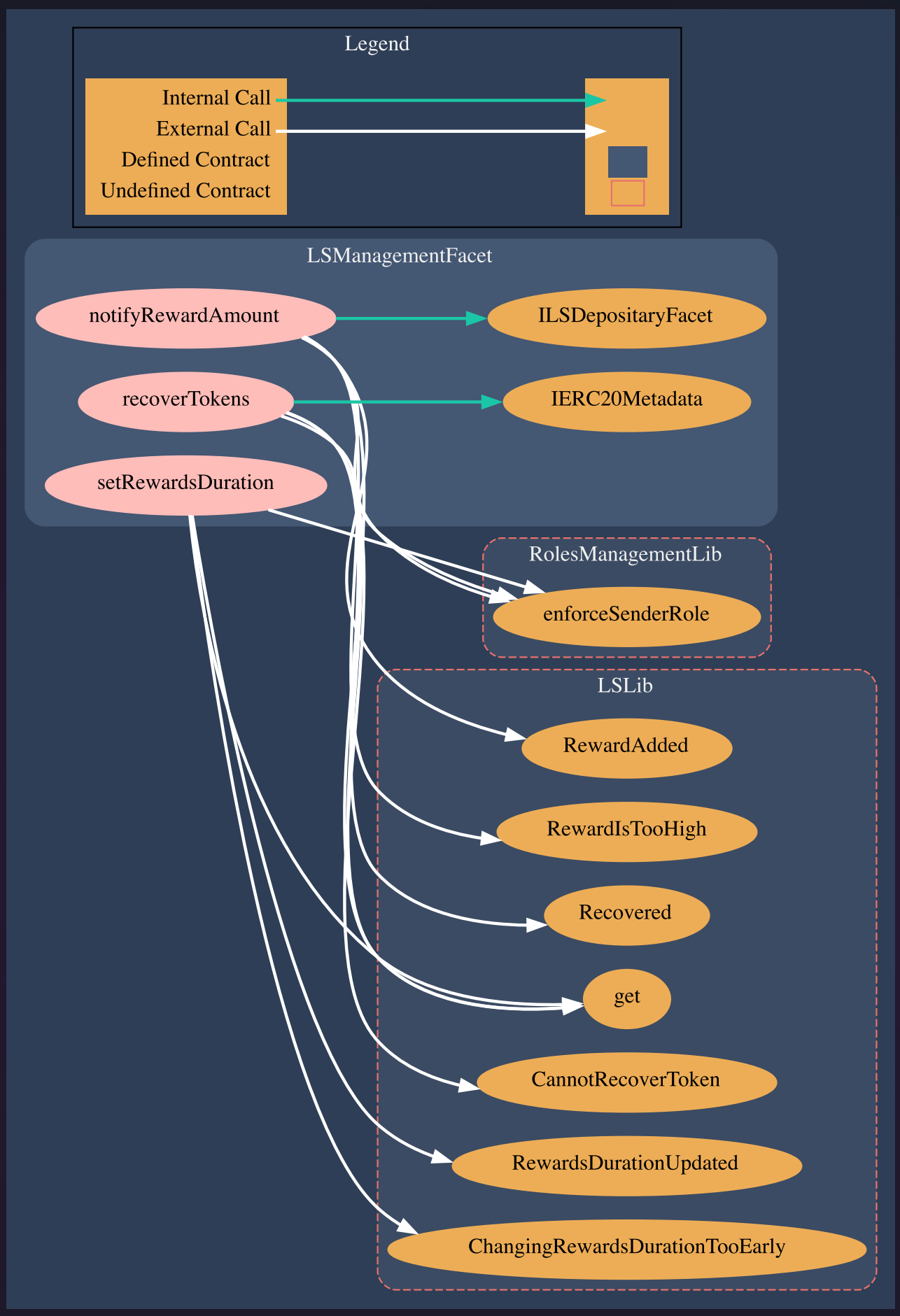
rewardPerToken()
Vulnerabilities not detected

function earned(**address account** **)**
Vulnerabilities not detected

getRewardForDuration()
Vulnerabilities not detected

12.1 Structure of contract:

LSManagementFacet.sol



pic.12.1 LSManagementFacet.sol

12.2 LSManagementFacet.sol contract methods analysis:

```
notifyRewardAmount(
        uint256 reward
    )
```

Vulnerabilities not detected

```
function recoverTokens(
        address tokenAddress,
        uint256 tokenAmount
    )
```

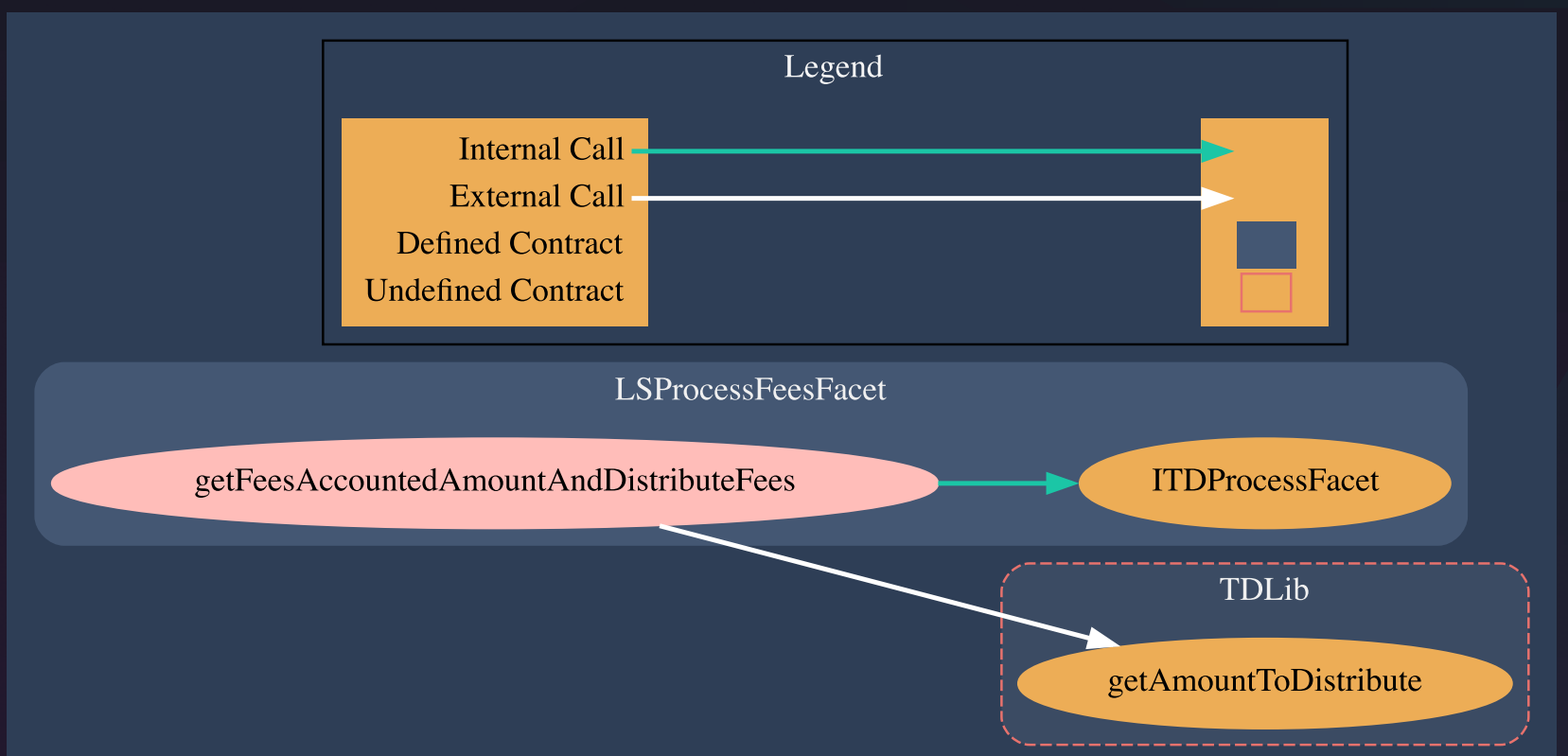
Vulnerabilities not detected

```
setRewardsDuration(
        uint256 _rewardsDuration
    )
```

Vulnerabilities not detected

13.1 Structure of contract:

LSProcessFeesFacet.sol



pic.13.1 LSProcessFeesFacet.sol

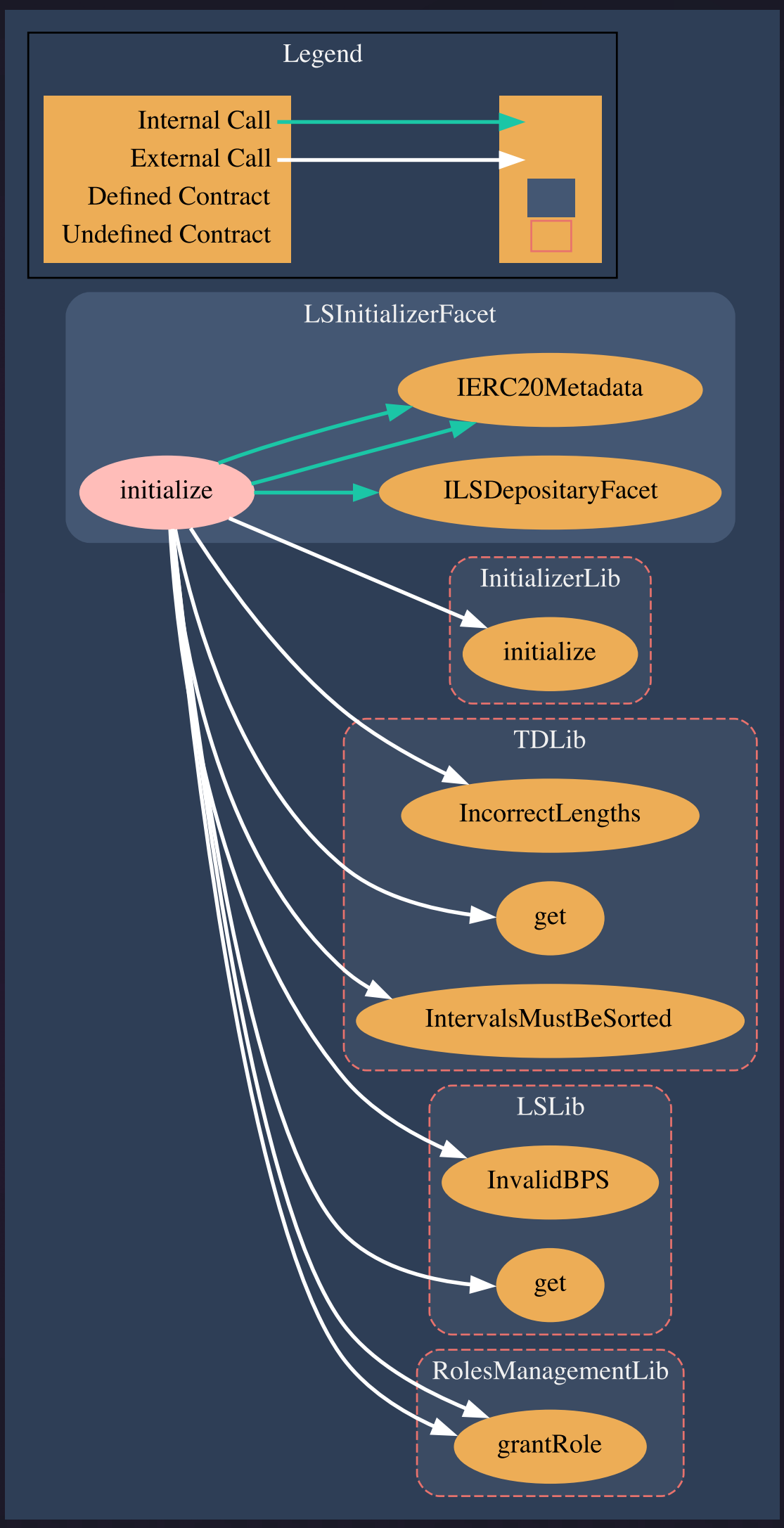
13.2 LSProcessFeesFacet.sol contract methods analysis:

```
getFeesAccountedAmountAndDistributeFees(
        uint256 reward,
        IERC20 rewardsToken
    )
```

Vulnerabilities not detected

14.1 Structure of contract:

LSInitializerFacet.sol



pic.14.1 LSInitializerFacet.sol

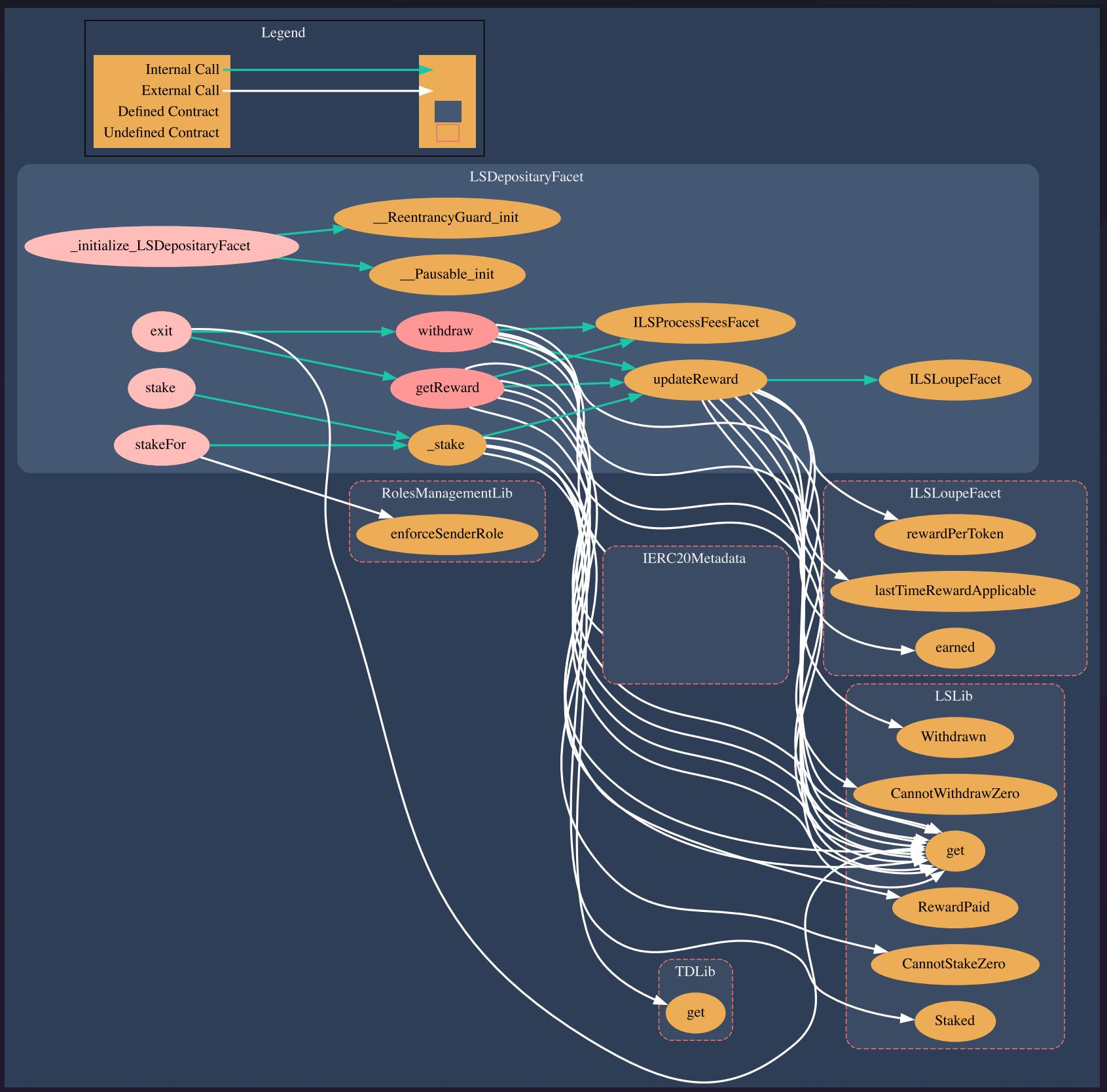
14.2 LSInitializerFacet.sol contract methods analysis:

```
initialize(
        address owner,
        address locusToken,
        address rewardDistributor,
        address rewardsToken,
        address stakingToken,
        uint32[] memory feeDurationPoints,
        uint16[] memory feeBasePoints
    )
```

Vulnerabilities not detected

15.1 Structure of contract:

LSDepositaryFacet.sol



pic.15.1 LSDepositaryFacet.sol

15.2 LSDepositaryFacet.sol contract methods analysis:

`_initialize_LSDepositaryFacet()`
Vulnerabilities not detected

```
stakeFor(
        address staker,
        uint256 amount
    )
```

Vulnerabilities not detected	
TOKEN FLOW	Tokens in, ALLOWED_TO_STAKE_FOR_ROLE

```
function stake(
        uint256 amount
    )
```

Vulnerabilities not detected	
TOKEN FLOW	Tokens in, public

```
function withdraw(
        uint256 amount
    )
```

Vulnerabilities not detected	
TOKEN FLOW	Tokens out, public

`getReward()`	
Vulnerabilities not detected	
TOKEN FLOW	Tokens out, public

15.2 LSDepositaryFacet.sol contract methods analysis:

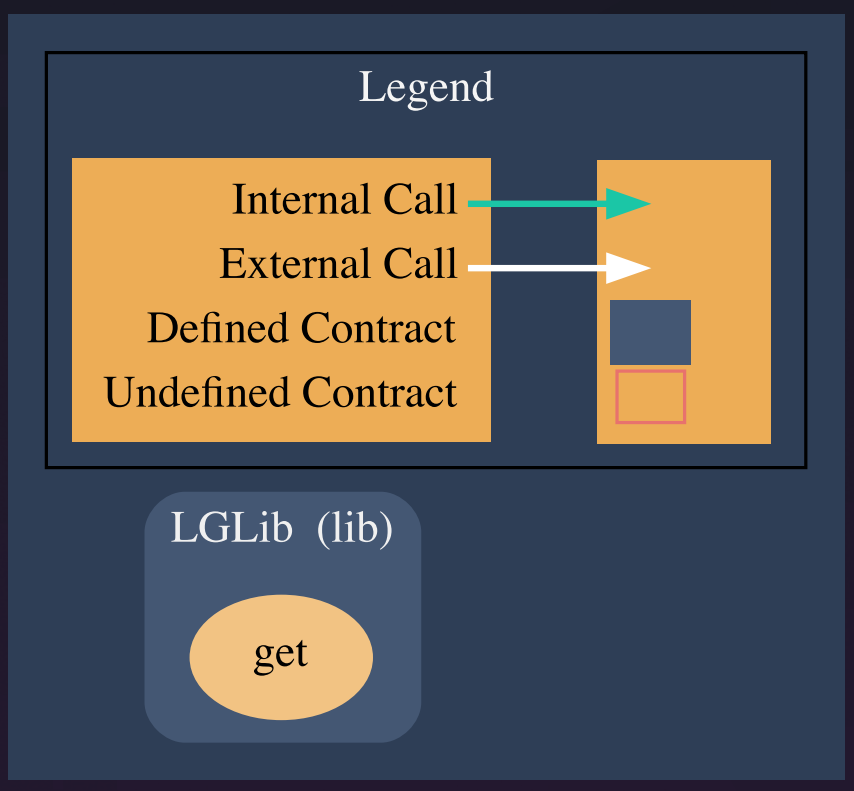
exit()	
Vulnerabilities not detected	
TOKEN FLOW	Tokens out, public

updateReward(address account)
Vulnerabilities not detected

_stake(address staker, uint256 amount)
Vulnerabilities not detected

16.1 Structure of contract:

LGLib.sol



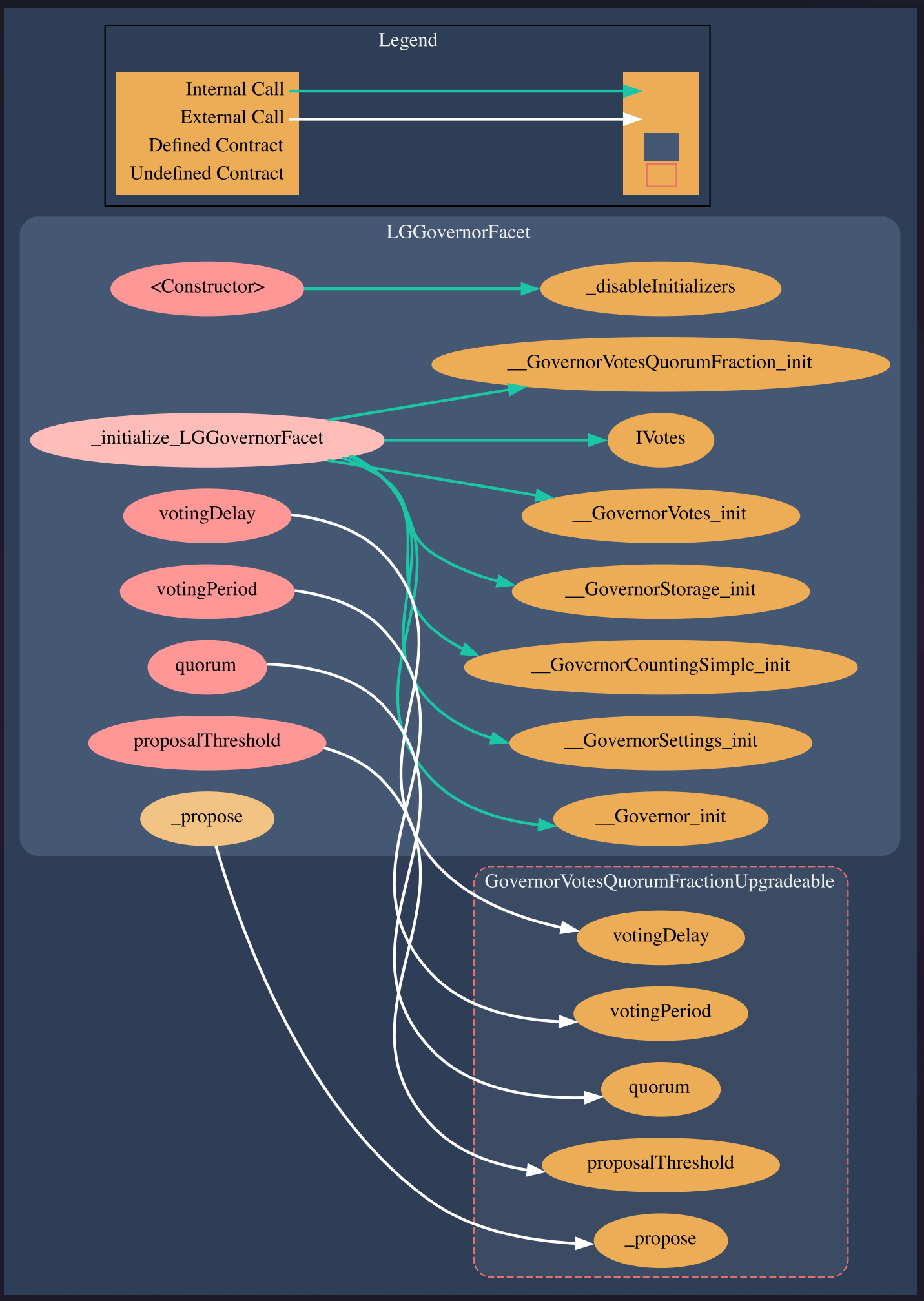
pic.16.1 LGLib.sol

16.2 LGLib.sol contract methods analysis:

get()
Vulnerabilities not detected

17.1 Structure of contract:

LGGovernorFacet.sol



pic.17.1 LGGovernorFacet.sol

17.2 LGGovernorFacet.sol contract methods analysis:

constructor()
Vulnerabilities not detected

```
_initialize_LGGovernorFacet(
        address locus,
        uint48 initialVotingDelay,
        uint32 initialVotingPeriod,
        uint256 initialProposalThresholdInLocusTokens,
        uint256 quorumFractionInPercents,
        string memory governorName
    )
```

Vulnerabilities not detected

votingDelay()
Vulnerabilities not detected

votingPeriod()
Vulnerabilities not detected

```
quorum(
        uint256 blockNumber
    )
```

Vulnerabilities not detected

proposalThreshold()
Vulnerabilities not detected

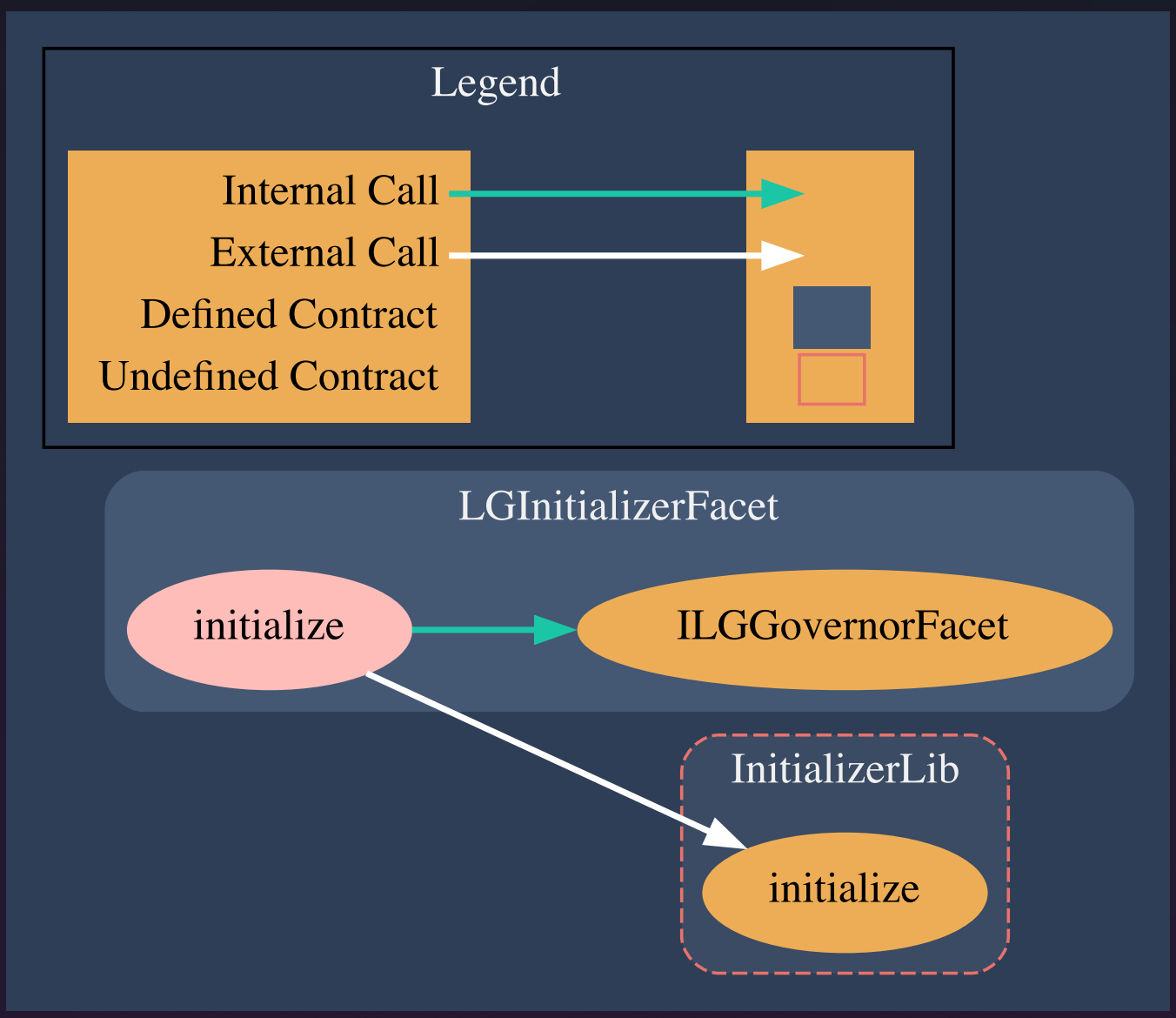
17.2 LGGovernorFacet.sol contract methods analysis:

```
_propose(
        address[] memory targets,
        uint256[] memory values,
        bytes[] memory calldatas,
        string memory description,
        address proposer
    )
```

Vulnerabilities not detected

18.1 Structure of contract:

LGInitializerFacet.sol



pic.18.1 LGInitializerFacet.sol

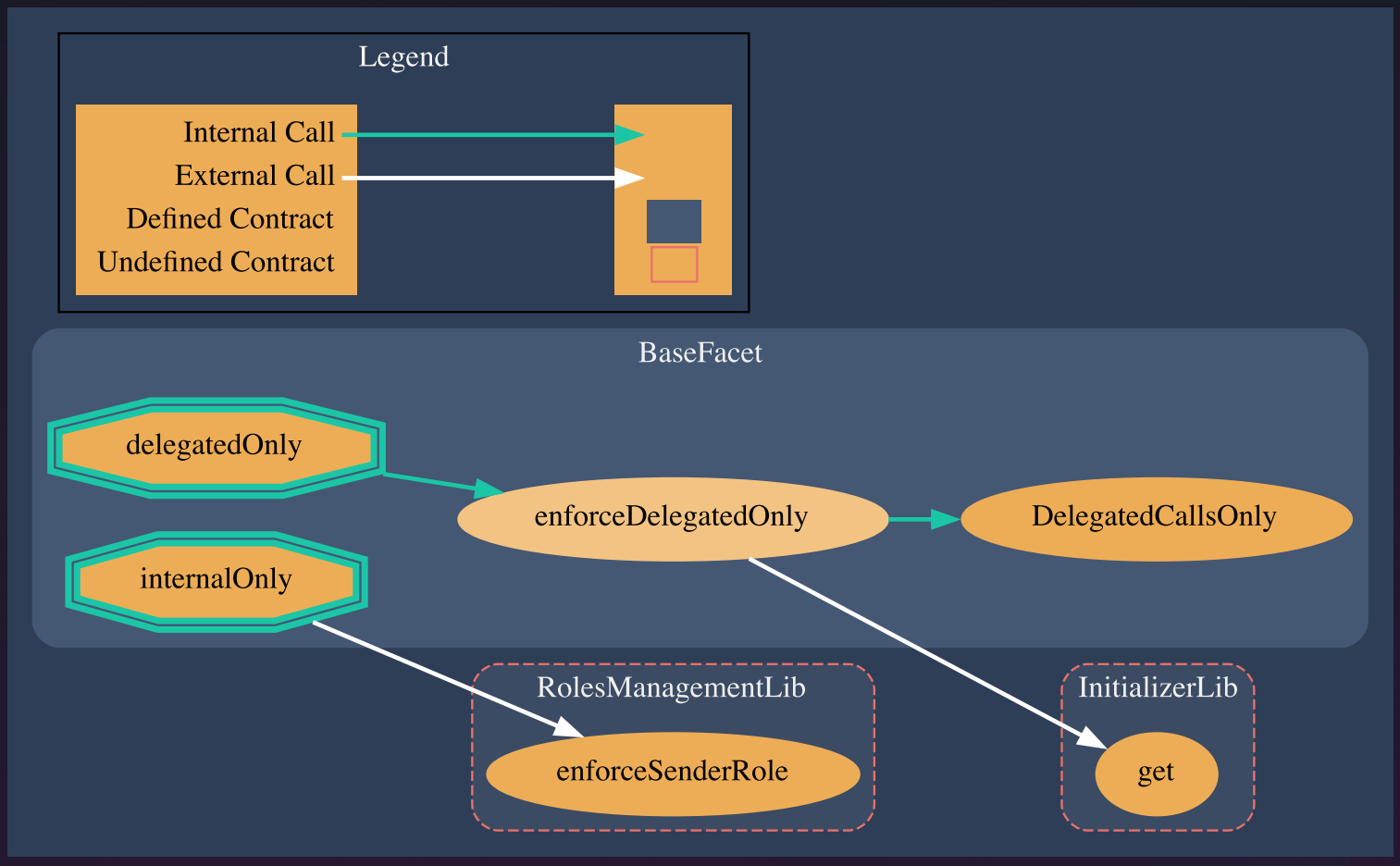
18.2 LGInitializerFacet.sol contract methods analysis:

```
initialize(
        address locus,
        uint48 initialVotingDelay,
        uint32 initialVotingPeriod,
        uint256 initialProposalThresholdInLocusTokens,
        uint256 quorumFractionInPercents
    )
```

Vulnerabilities not detected

19.1 Structure of contract:

BaseFacet.sol



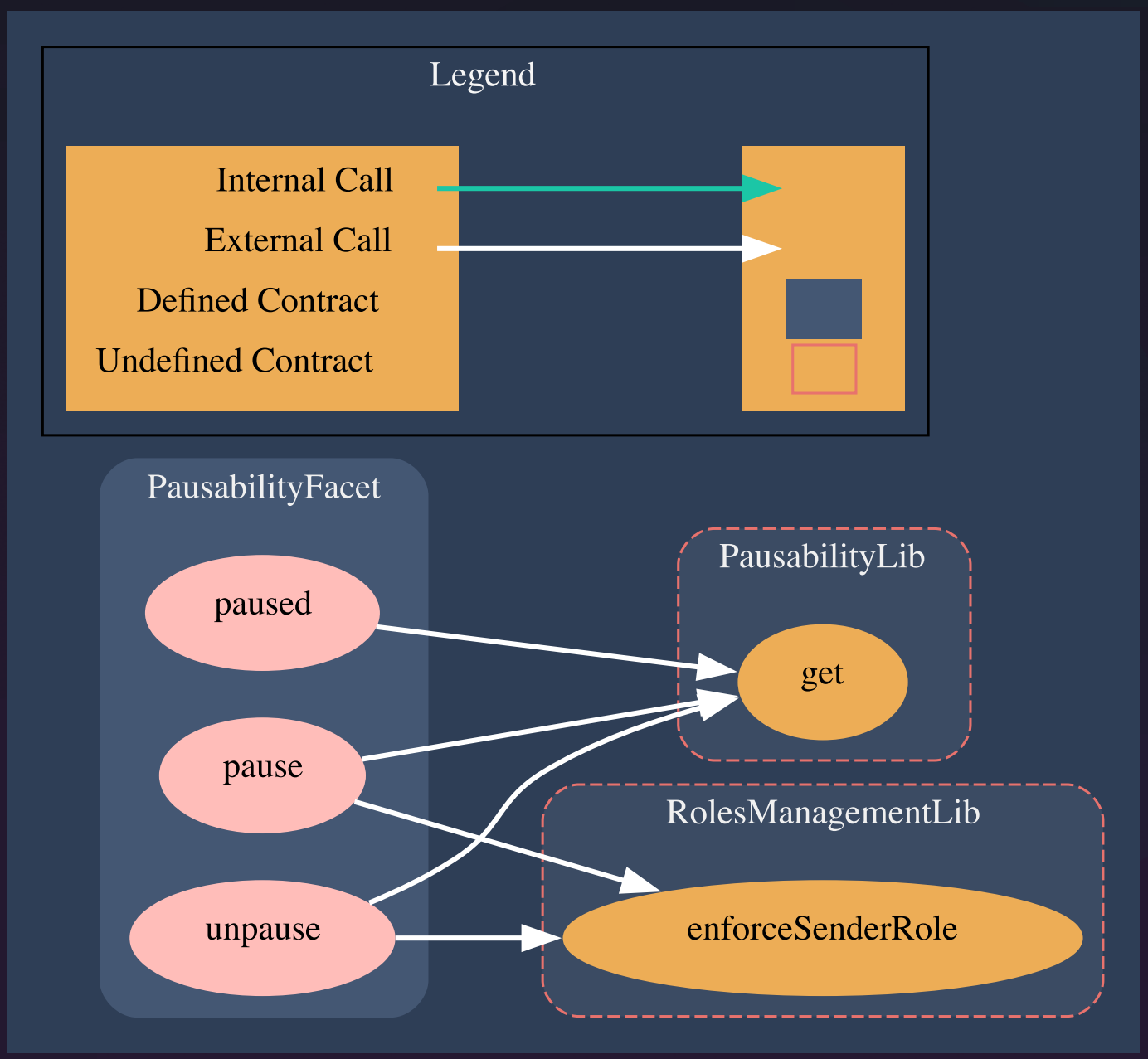
pic.19.1 BaseFacet.sol

19.2 BaseFacet.sol contract methods analysis:

enforceDelegatedOnly()
Vulnerabilities not detected

20.1 Structure of contract:

PausabilityFacet.sol



pic.20.1 PausabilityFacet.sol

20.2 PausabilityFacet.sol contract methods analysis:

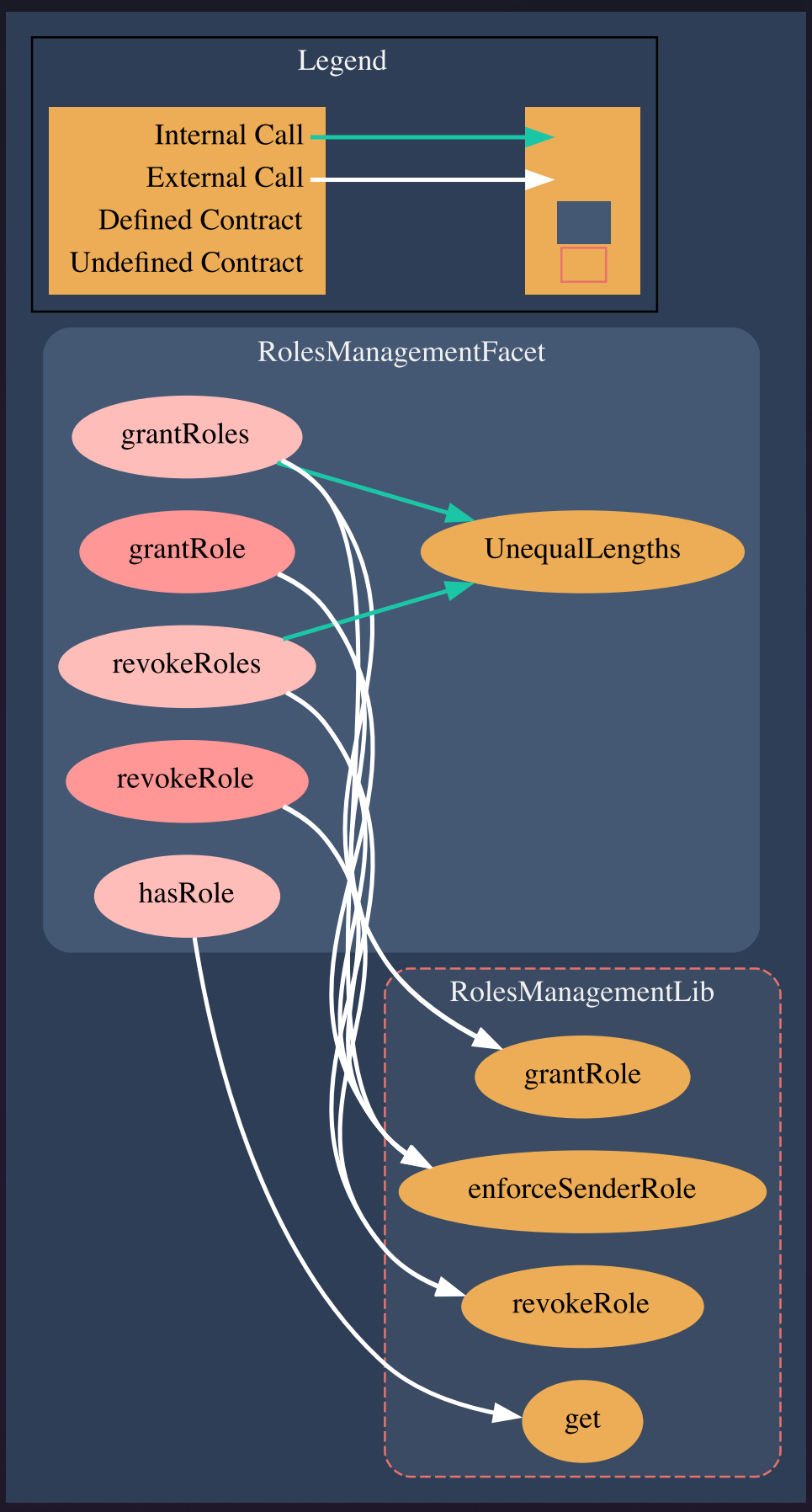
paused()
Vulnerabilities not detected

pause()
Vulnerabilities not detected

unpause()
Vulnerabilities not detected

21.1 Structure of contract:

RolesManagementFacet.sol



pic.21.1 RolesManagementFacet.sol

21.2 RolesManagementFacet.sol contract methods analysis:

grantRoles(address[] calldata people, bytes32[] calldata roles)
Vulnerabilities not detected

revokeRoles(address[] calldata people, bytes32[] calldata roles)
Vulnerabilities not detected

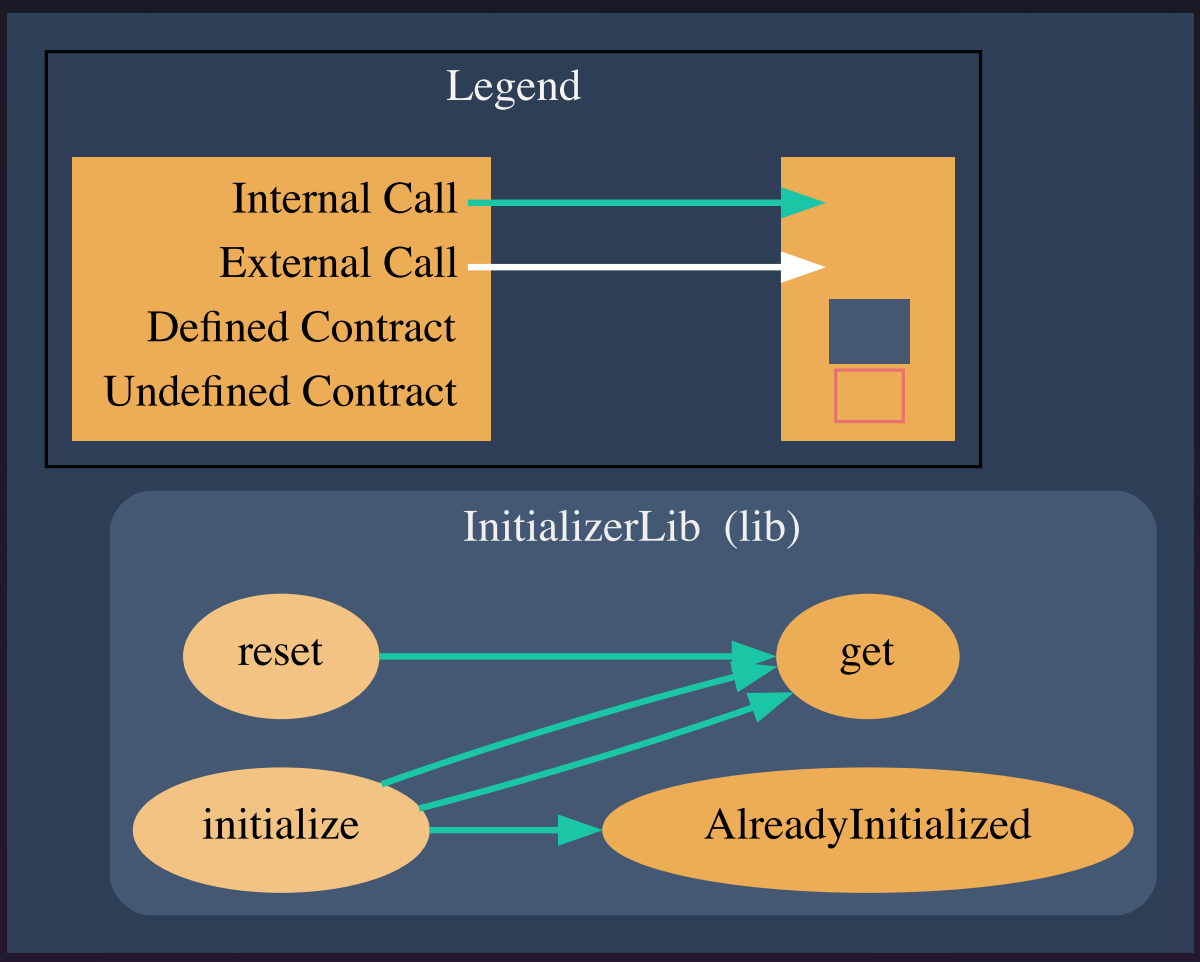
grantRole(address who, bytes32 role)
Vulnerabilities not detected

revokeRole(address who, bytes32 role)
Vulnerabilities not detected

hasRole(address who, bytes32 role)
Vulnerabilities not detected

22.1 Structure of contract:

InitializerLib.sol



pic.22.1 InitializerLib.sol

22.2 InitializerLib.sol contract methods analysis:

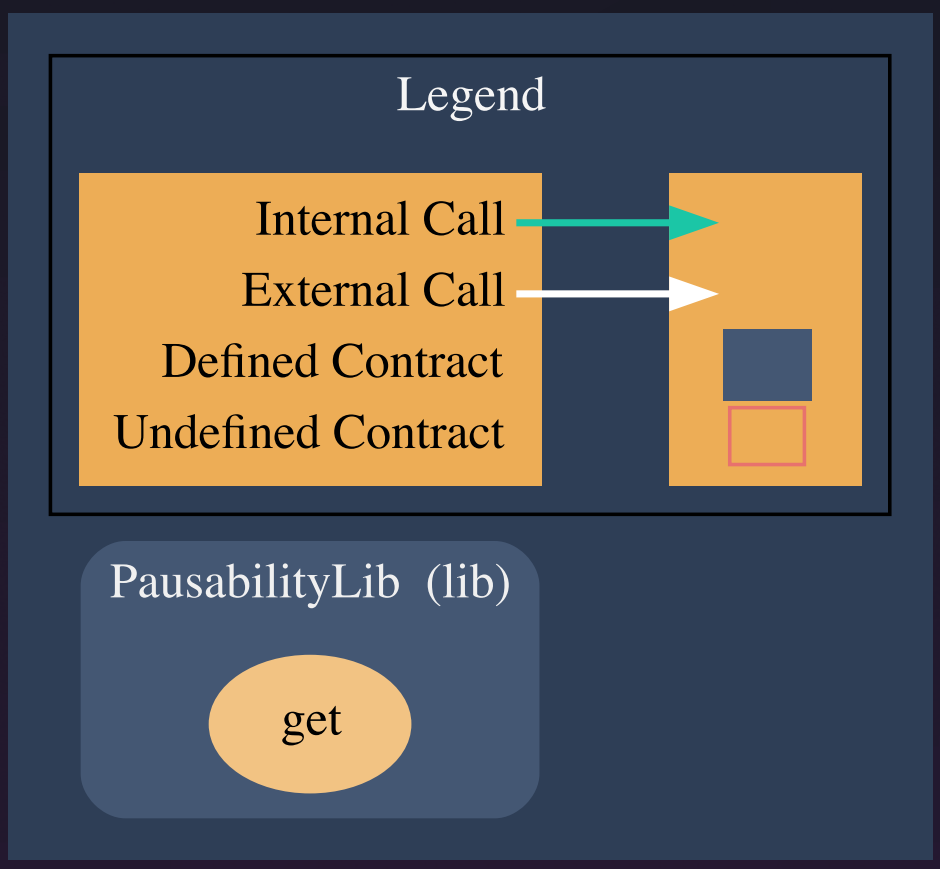
get()
Vulnerabilities not detected

reset()
Vulnerabilities not detected

initialize()
Vulnerabilities not detected

23.1 Structure of contract:

PausabilityLib.sol



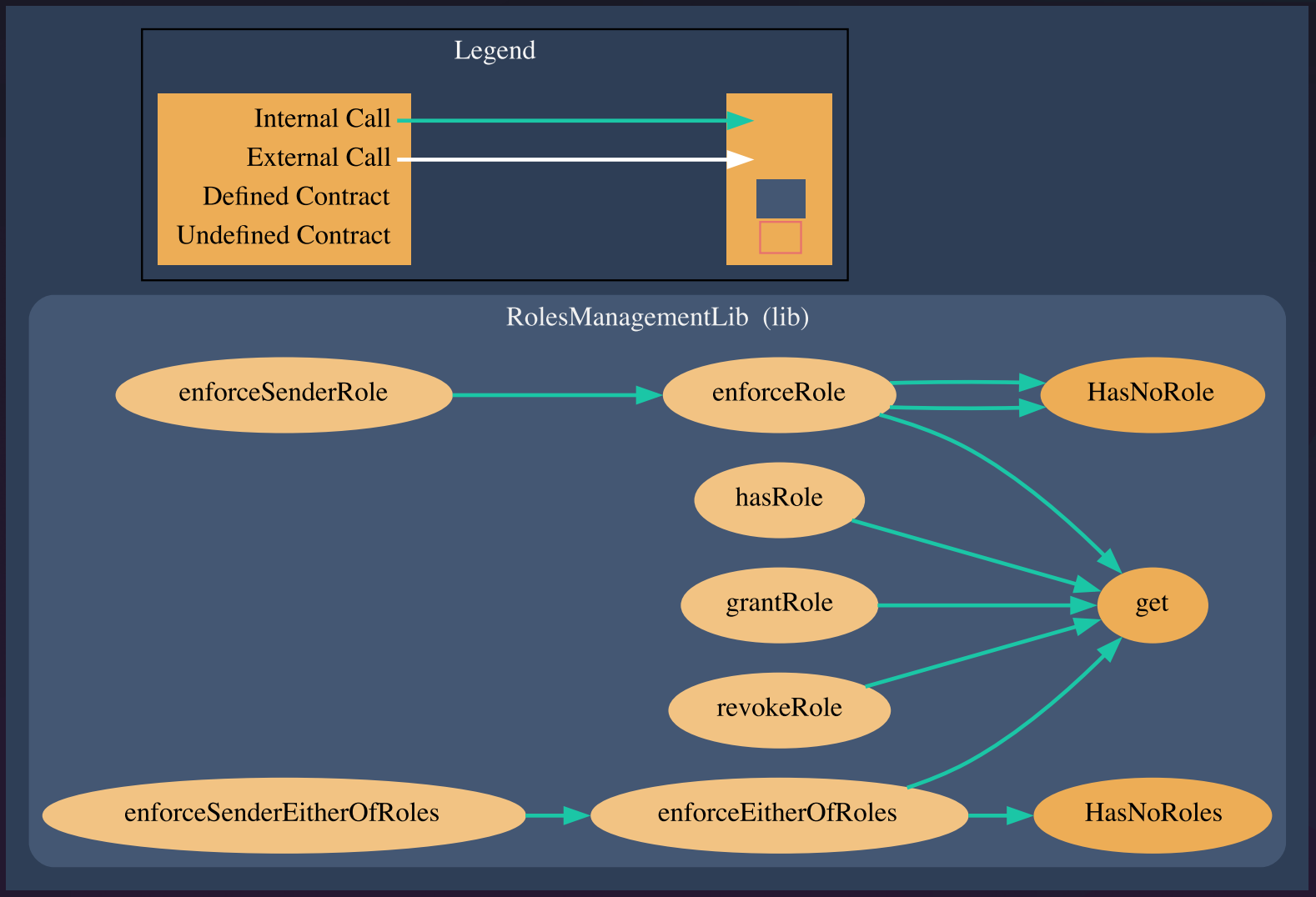
pic.23.1 PausabilityLib.sol

23.2 PausabilityLib.sol contract methods analysis:

get()
Vulnerabilities not detected

24.1 Structure of contract:

RolesManagementLib.sol



pic.24.1 RolesManagementLib.sol

24.2 RolesManagementLib.sol contract methods analysis:

get()
Vulnerabilities not detected

enforceRole(address who, bytes32 role)
Vulnerabilities not detected

hasRole(address who, bytes32 role)
Vulnerabilities not detected

enforceSenderRole(bytes32 role)
Vulnerabilities not detected

grantRole(address who, bytes32 role)
Vulnerabilities not detected

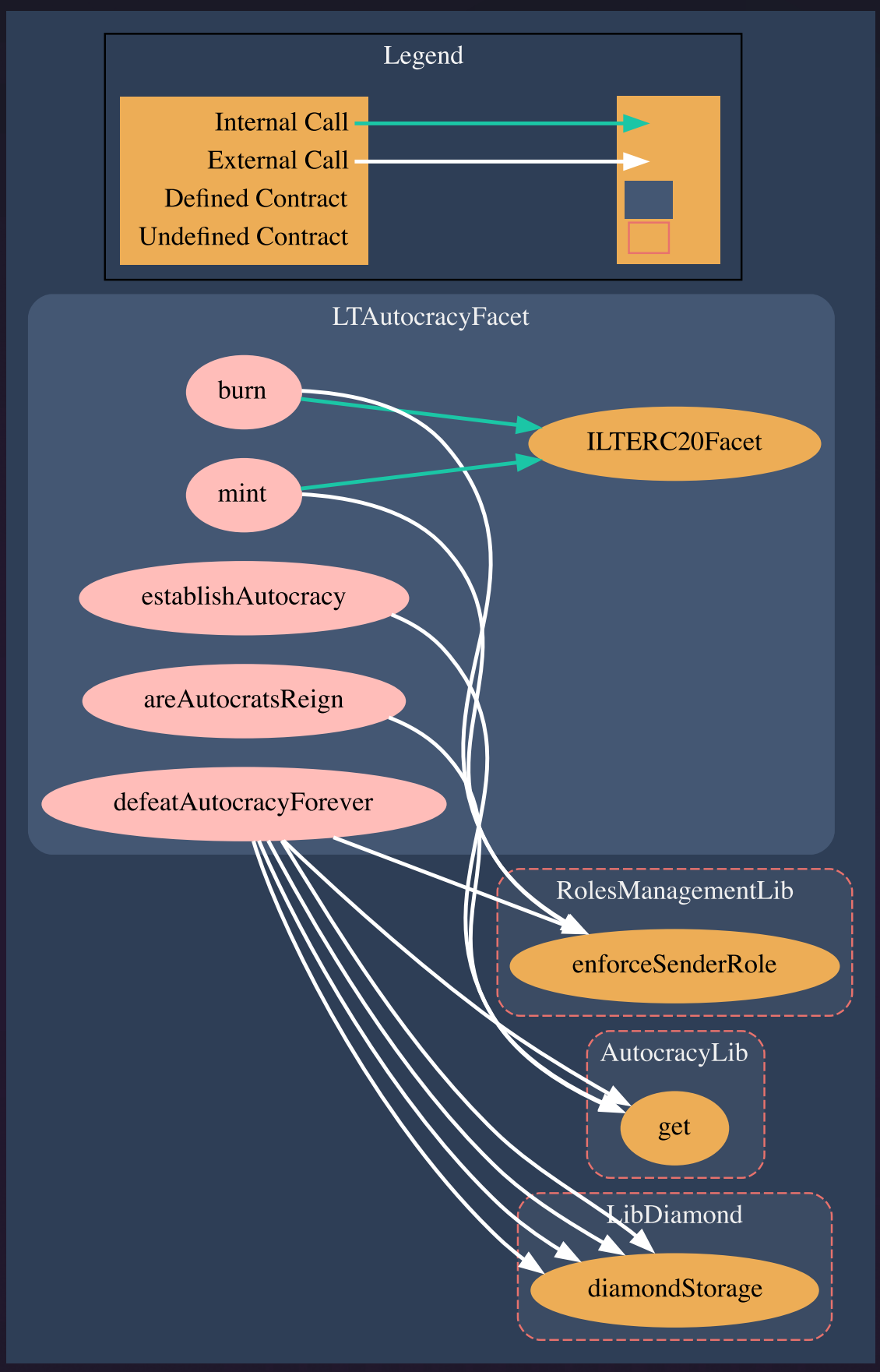
revokeRole(address who, bytes32 role)
Vulnerabilities not detected

enforceEitherOfRoles(address who, bytes32[] memory roles)
Vulnerabilities not detected

enforceSenderEitherOfRoles(bytes32[] memory roles)
Vulnerabilities not detected

25.1 Structure of contract:

LTAutocracyFacet.sol



pic.25.1 LTAutocracyFacet.sol

25.2 LTAutocracyFacet.sol contract methods analysis:

burn(uint256 amount)
Vulnerabilities not detected

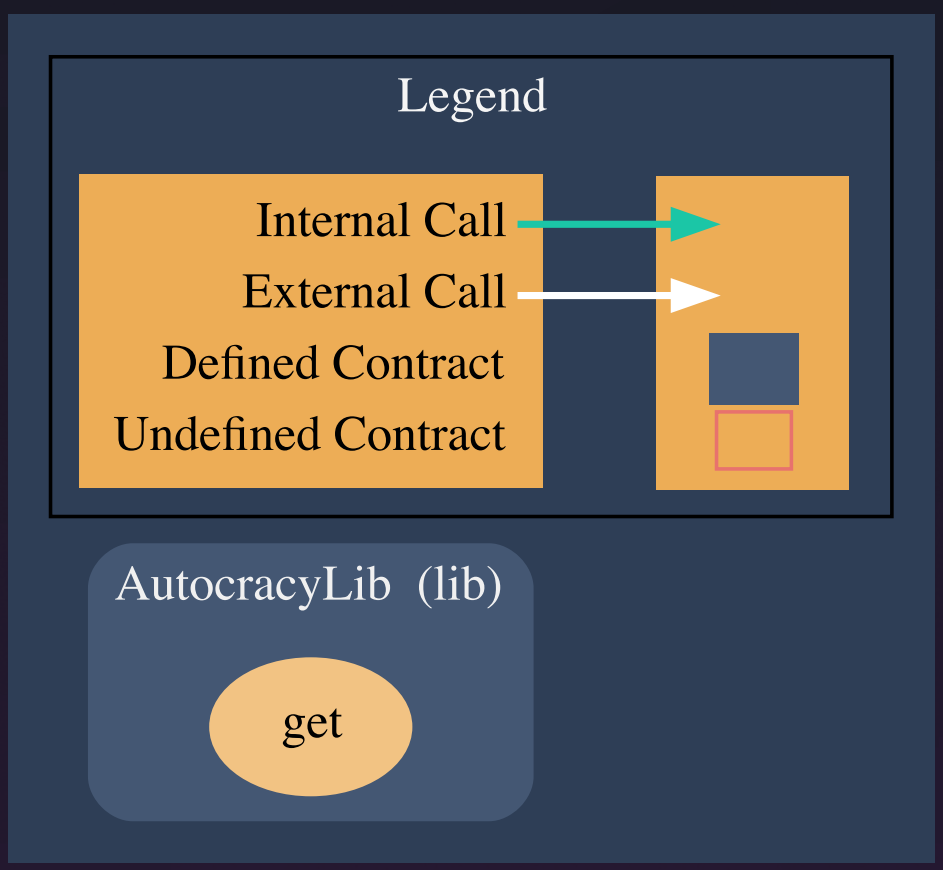
mint(address who, uint256 amount)
Vulnerabilities not detected

establishAutocracy()
Vulnerabilities not detected

defeatAutocracyForever()
Vulnerabilities not detected

26.1 Structure of contract:

AutocracyLib.sol



pic.26.1 AutocracyLib.sol

26.2 AutocracyLib.sol contract methods analysis:

get()
Vulnerabilities not detected

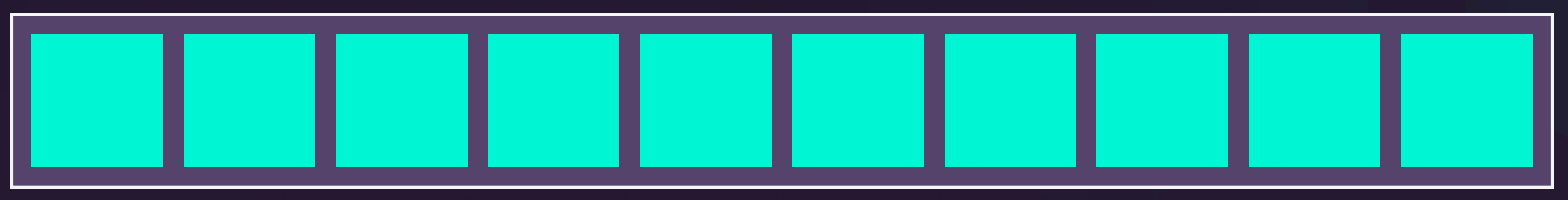
Verification checksums

Contract name	Bytecode hash(SHA-256)
LTEmissionControlFacet.sol	5f30a470e95f9c33756b4b7b2224390e963f2ba523c33457c9c5ee2bef50901c
LTERC20Facet.sol	daebf0f77a42f118c62da10b6c326bc7886a83a85c5d4d893d04f187473065dd
LTInitializerFacet.sol	68b7fc84583dadf0a5f1c6973f55054045c2cd71dff369d239b71e243a3e77ea
LTLib.sol	94cd4a196f2894b1f60a5bb8bb4565f61044b95c565e5751c5fa229ff4a68158
TDLoupeFacet.sol	6f82995a4b2d990ca5258c4d670e2a8d20b00dd2982a777cd5f6b51c43456c35
TDManagementFacet.sol	73572ff000fa442ce21036533b0fcd82a08a2f674f1032bb8c261b99b48a52a9
TDProcessFacet.sol	ea6548bc89683ea271760e5d02b6600e5795e0fc85357521237ba039513a9dd1
TDLib.sol	786659912dc4efac55a00d0fc499fc9669cc6f8d75c9b32db01e4eaa9330b40c
MidasClaim.sol	8b0be532130be3ef6bb30d4f8cffb21f94e42508c74f144ae7657f7f22e02c27
LSLib.sol	a83d83a078d3f3fdb80ff5bcbd6b7df8ad3e4e845e49717637ce6e176762a39e
LSLoupeFacet.sol	1f6f6663ab6a092e0ed184661f66854871d7f1341d8efbaf3dc30b3754f02a4c
LSManagementFacet.sol	3fe949569fa4d9c983f3cf57b0940960bae7fbd34de67ceda60a2604b20d11cf
LSProcessFeesFacet.sol	cf77381c108800057719ec4ac7d949bbddc68219d4e3bb5f215b7ee628cf7d81
LSInitializerFacet.sol	831d4bc9a5c54687445704f6627379bed299d8e9084ec576239af66b157b1b80

Verification checksums

Contract name	Bytecode hash(SHA-256)
LSDepositaryFacet.sol	9c6f5012e1d7294c372d9f639ae873fe110204e625c04119da1707d8a2746953
LGLib.sol	b6effa197c9a8e9583f08b05152d1f613e541036cd38e06dc94c611d231343bd
LGGovernorFacet.sol	e9a897c35a2f6f1565bb0e6c1bec450fcfe809126b50e5e2d8be8a6b548089df
LGInitializerFacet.sol	6d705c8a345b4a67ad12f5c6e780d1047fb62d36542d673a7924b384c042c418
BaseFacet.sol	07dc17df2b8d51d82e9c87a71982654b74554f638883208ad6712ff194dc71bf
PausabilityFacet.sol	7c3d7eac3bff71183937d3cbe55b89a072ad7bb11197069ed00e249063a96d73
RolesManagementFacet.sol	ca681f4d347fd35496695f1ab28a4e98b0263dd234d30536cdd171cda1727178
InitializerLib.sol	abd2f524f965daed6cec287a96a234f9bdd4bf16a7e4ca1f10fa183763c9ca22
PausabilityLib.sol	d50f6f16b0b35dfaa94d8582ab585702a3cbf05013cedbe488cef1b39139c50e
RolesManagementLib.sol	d2608d631dba0cc9fd86e004d4975f9549e7ad587020cc0e36e5ea18485f4d6c
LTAutocracyFacet.sol	af1753d23e1ce20b7450f6e3c3993f15eb8eadbd5a91d1a836c0687216868ada
AutocracyLib.sol	24e07f49dec14f2ce101a5a545b86f33a0231e19289c057f560ec0cbe95addb1

Project evaluation



10/10

Get in touch 👋

@smartstatetech

@smartstate

@SmartStateAudit

@smartstatetech

@smartstate.tech

View this report on Smartstate.tech

info@smartstate.tech

smartstate.tech