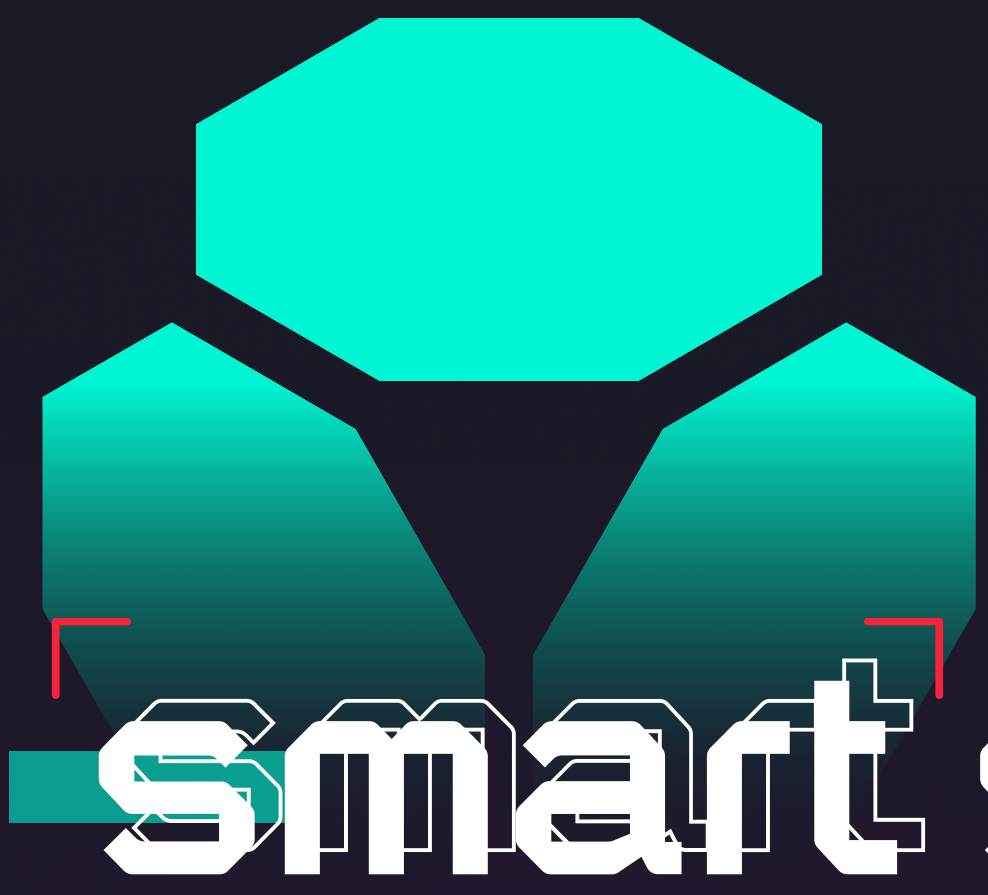


smart state

Web3 security easier than ever

EYWA
Smart contract audit report

Jun 05 2023

Disclaimer

EYWA is a system that allows different blockchain ecosystems to interact with each other. Project enable users to move their assets between different networks quickly and cheaply, and enable developers to efficiently implement cross-chain logic for their decentralized applications. The core architectural element of the EYWA ecosystem is the EYWA Cross-chain Data Protocol, which is a transport layer between blockchains. All EYWA products for DeFi users are based on this protocol.

Although at the time of this audit the core of EYWA multisig is represented by a trusted group of projects, EYWA aims for DAO, as reflected in EYWA project current documentation.

The mission of the project is to bring DeFi together. EYWA intends to make decentralized finance simple, convenient and understandable even for beginners.

CDP Smart Contracts: These smart contracts serve as a means for sending and accepting cross-chain calls. They also include a node registration contract used in the Proof of Authority (POA) consensus among oracle nodes.

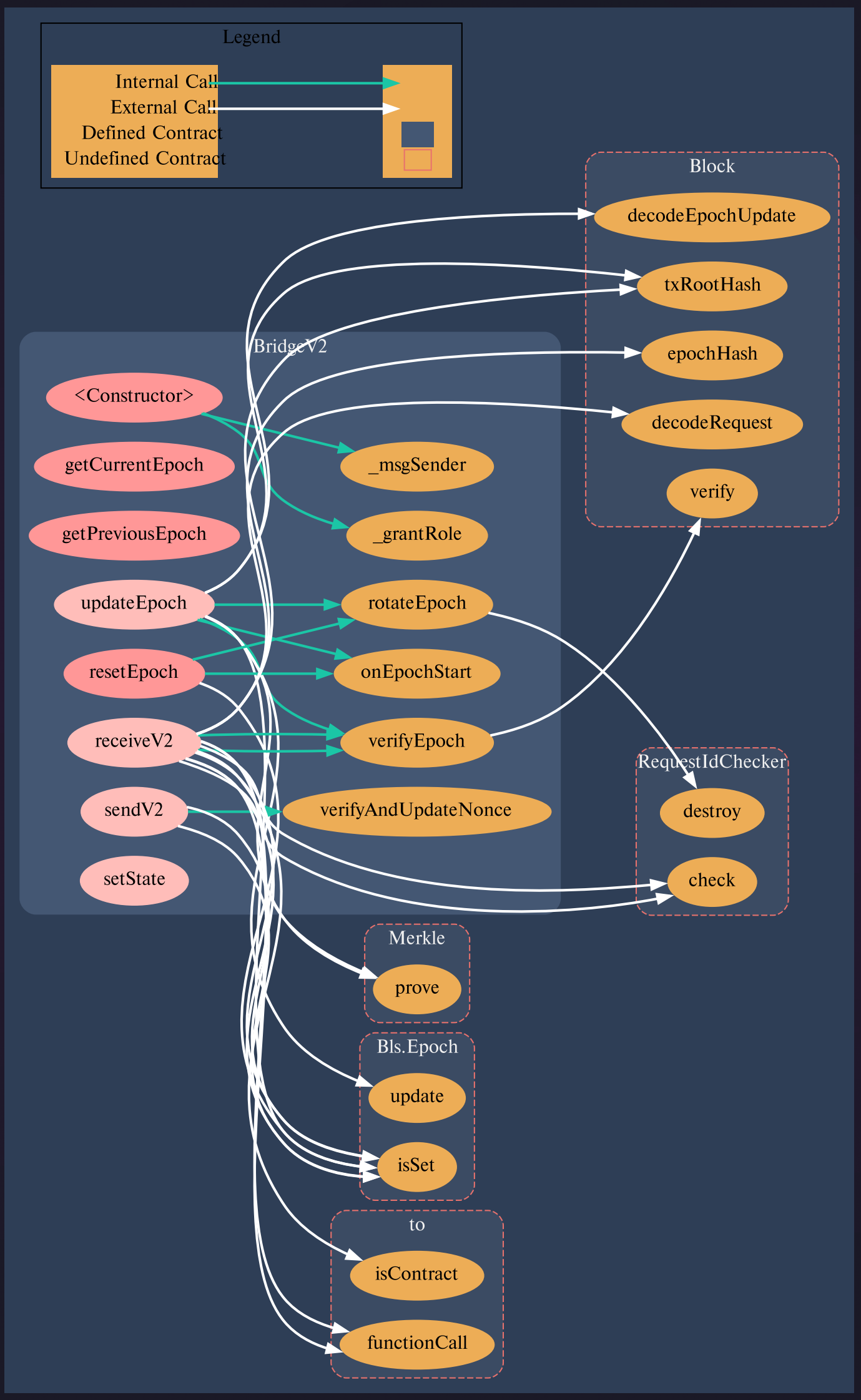
Table of contents

Methodology

- Best code practices
- ERC20/BEP20 compliance (if applicable)
- FA2 compliance (if applicable)
- Logical bugs
- General Denial Of Service(DOS)
- Locked ether
- Private data leaks
- Using components with known vulns
- Weak PRNG
- Unused vars
- Unchecked call return method
- Code with no effects
- Pool Asset Security (backdoors in the underlying ERC-20)
- Function visibility
- Use of deprecated functions
- Authorization issues
- Re-entrancy
- Arithmetic Over/Under Flows
- Hidden Malicious Code
- External Contract Referencing
- Short Address/Parameter Attack
- Race Conditions/Front Running
- Uninitialized Storage Pointers
- Floating Points and Precision
- Signatures Replay

Structure of contract

BridgeV2.sol



pic.1.1 BridgeV2.sol

Contract methods analysis:

constructor()
Vulnerabilities not detected

getCurrentEpoch()
Vulnerabilities not detected

getPreviousEpoch()
Vulnerabilities not detected

updateEpoch(ReceiveParams calldata params)
Vulnerabilities not detected

Fixed in documentation | WARNING

resetEpoch()
Acknowledged: Wallet with OPERATOR_ROLE has right to call this function which affects decentralization factor of the protocol. Consider that this wallet is managed by governance. Comment: This function is only for PoA.

sendV2(SendParams calldata params, address from, uint256 nonce)
Vulnerabilities not detected

receiveV2(ReceiveParams[] calldata params)
Vulnerabilities not detected

Fixed in documentation | WARNING

setState(State state_)
Acnowledged: Wallet with OPERATOR_ROLE has right to call this function which affects decentralization factor of the protocol. Consider that this wallet is managed by governance. Comment: Operator should be able to start\stop send\receive messages

verifyEpoch(Bls.Epoch storage epoch, ReceiveParams calldata params)
Vulnerabilities not detected

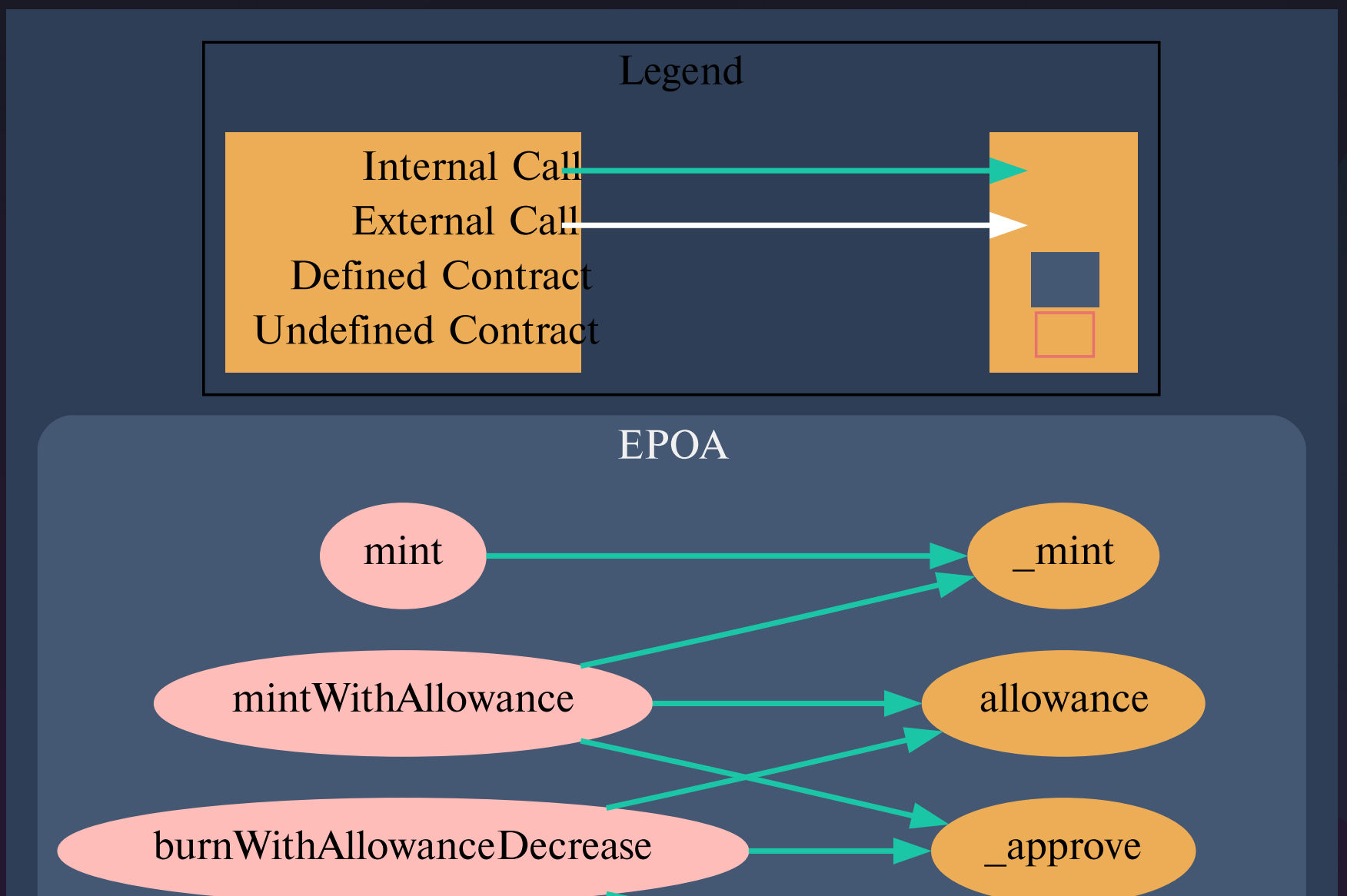
verifyAndUpdateNonce(address from, uint256 nonce)
Vulnerabilities not detected

rotateEpoch()
Vulnerabilities not detected

onEpochStart(uint64 protocolVersion_)
Vulnerabilities not detected

Structure of contract

EPOA.sol



pic.1.2 EPOA.sol

Contract methods analysis:

constructor(
address nodeRegistry_
)

Vulnerabilities not detected

mint(address account, uint256 amount)

Vulnerabilities not detected

mintWithAllowance(address account, address spender, uint256 amount)

Vulnerabilities not detected

burn(uint256 amount)

Vulnerabilities not detected

burnWithAllowanceDecrease(
address account,
address spender,
uint256 amount
)

Vulnerabilities not detected

transfer(address to, uint256 amount)

Vulnerabilities not detected

transferFrom(address from, address to, uint256 amount)

Vulnerabilities not detected

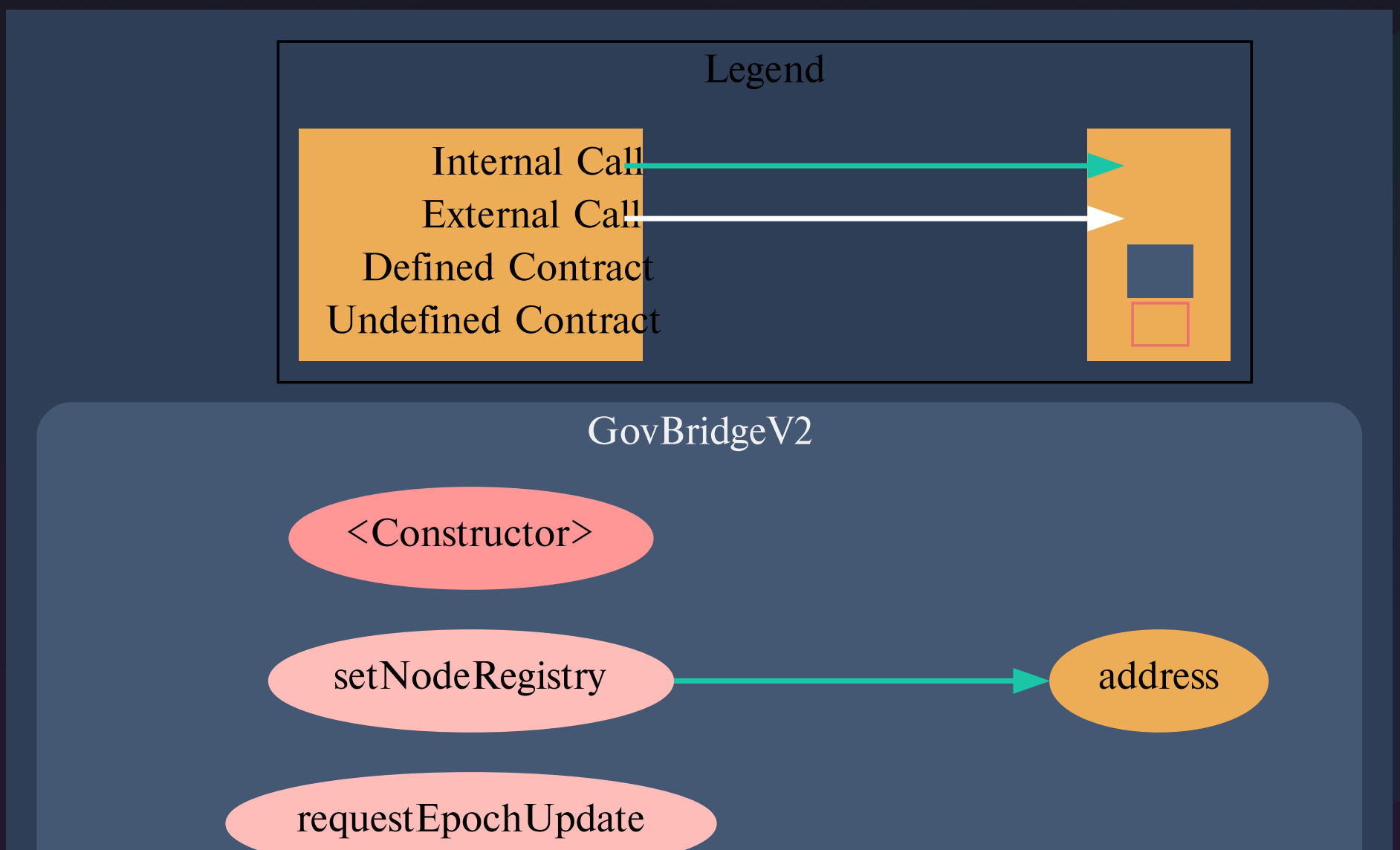
INFO

setNodeRegistry(address nodeRegistry_)

Function should emit an event

Structure of contract

GovBridgeV2.sol



pic.1.3 GovBridgeV2.sol

Contract methods analysis:

constructor()
Vulnerabilities not detected

Fixed in documentation	WARNING

setNodeRegistry(address nodeRegistry_)
Acknowledged: Wallet with DEFAULT_ADMIN_ROLE has right to call this function which affects decentralization factor of the protocol. Consider that this wallet is managed by governance. Comment: DEFAULT_ADMIN_ROLE controlled by DAO\Multisig

requestEpochUpdate()
Vulnerabilities not detected

Fixed in documentation	WARNING

requestProtocolVersionUpdate(uint64 version)
Acknowledged: Wallet with OPERATOR_ROLE has right to call this function which affects decentralization factor of the protocol. Consider that this wallet is managed by governance. Comment: controlled by the team (operator)

Fixed in documentation	WARNING

setEpochMinDuration(uint256 epochMinDuration_)
Acknowledged: Wallet with DEFAULT_ADMIN_ROLE has right to call this function which affects decentralization factor of the protocol. Consider that this wallet is managed by governance. Comment: DEFAULT_ADMIN_ROLE controlled by DAO\Multisig

Fixed in documentation	WARNING

setEpochMinRequestUpdateDuration(uint256 epochMinRequestUpdateDuration_)

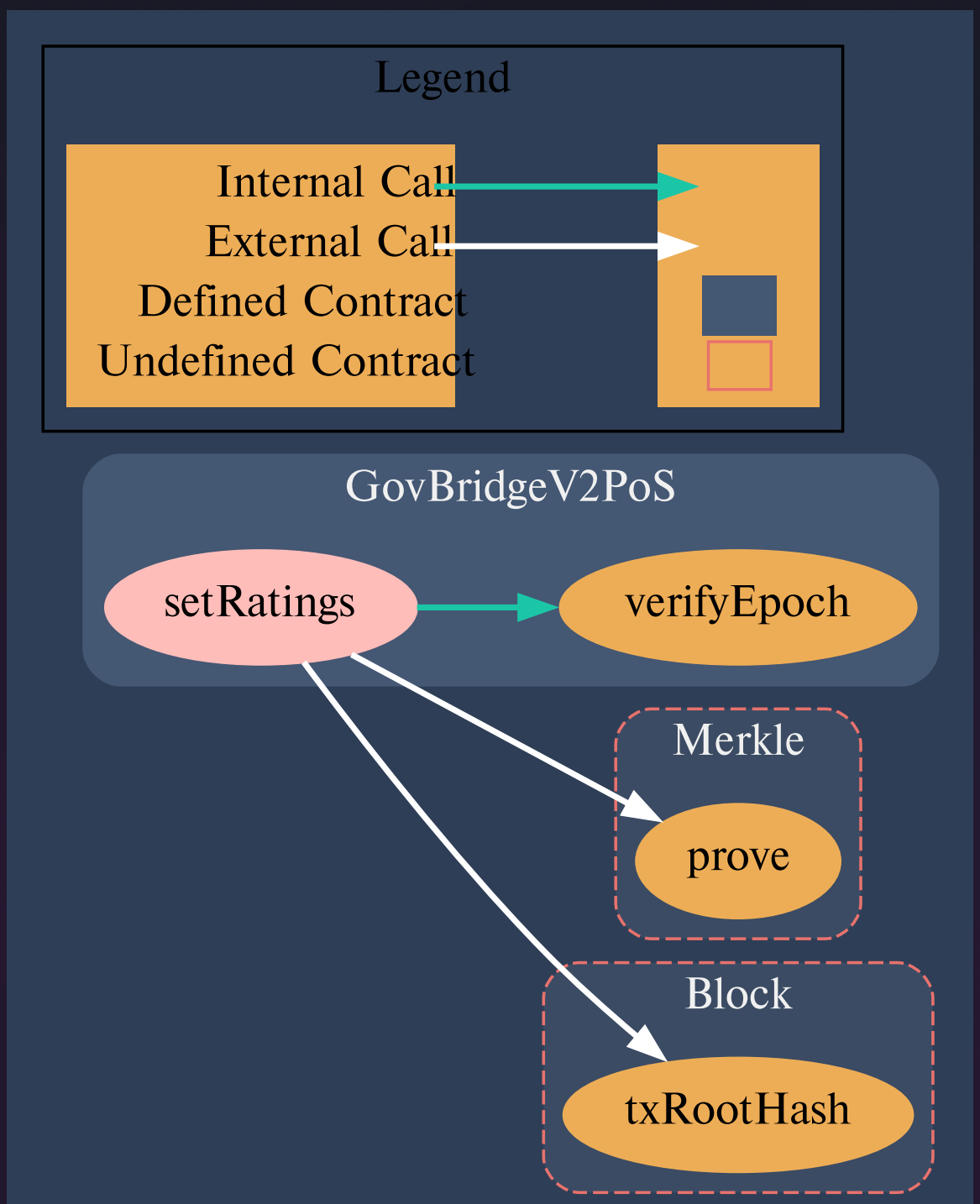
Acknowledged: Wallet with DEFAULT_ADMIN_ROLE has right to call this function which affects decentralization factor of the protocol. Consider that this wallet is managed by governance. Comment: DEFAULT_ADMIN_ROLE controlled by DAO\Multisig

onEpochStart(uint64 protocolVersion_)

Vulnerabilities not detected

Structure of contract

GovBridgeV2Pos.sol



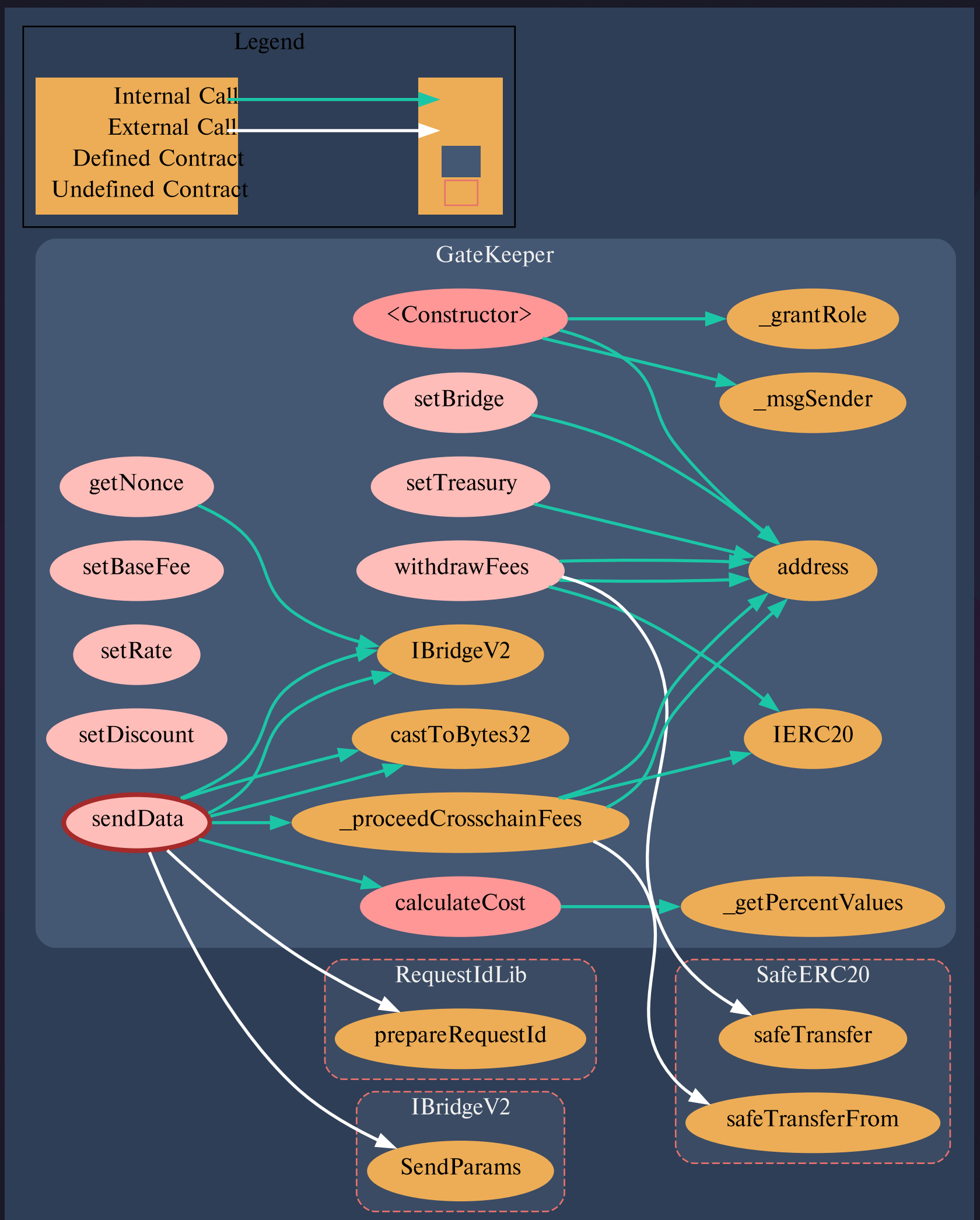
pic.1.4 GovBridgeV2Pos.sol

Contract methods analysis:

setRatings(ReceiveParams calldata params)
Vulnerabilities not detected

Structure of contract

GateKeeper.sol



pic.1.5 GateKeeper.sol

Contract methods analysis:

constructor(address bridge_)
Vulnerabilities not detected

setBridge(address bridge_)
Vulnerabilities not detected

setTreasury(address treasury_)
Vulnerabilities not detected

setBaseFee(BaseFee[] memory baseFees_)
Vulnerabilities not detected

setRate(Rate[] memory rates_)
Vulnerabilities not detected

setDiscount(address caller, uint256 discount)
Vulnerabilities not detected

calculateCost(address payToken, uint256 dataLength, uint256 chainIdTo, address sender)
Vulnerabilities not detected

_getPercentValues(uint256 amount, uint256 basePercent)
Vulnerabilities not detected

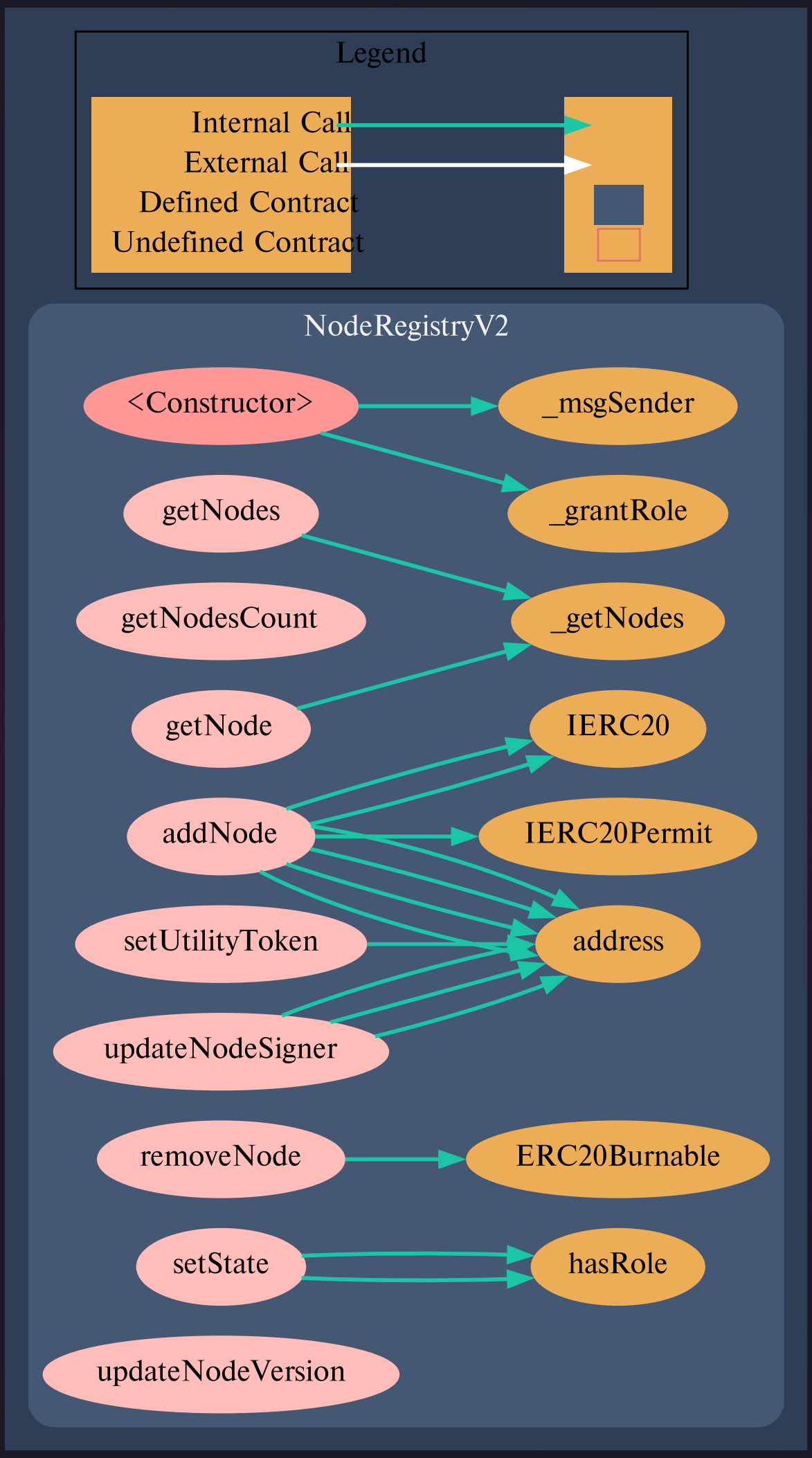
withdrawFees(address token, uint256 amount, address to)	
Vulnerabilities not detected	
TOKEN FLOW	Tokens out, ETH out, onlyOwner

sendData(bytes calldata data, address to, uint256 chainIdTo, address payToken)	
Vulnerabilities not detected	
TOKEN FLOW	Tokens in, ETH in, public

_proceedCrosschainFees(address payToken, uint256 transactionCost)
Vulnerabilities not detected

Structure of contract

NodeRegistryV2.sol



pic.1.6 NodeRegistryV2.sol

Contract methods analysis:

getNodes()
Vulnerabilities not detected

getNodes(address owner)
Vulnerabilities not detected

getNodesCount()
Vulnerabilities not detected

getNode(address signer)
Vulnerabilities not detected

setUtilityToken(address token)
Vulnerabilities not detected

addNode(Node memory node, uint256 deadline, uint8 v, bytes32 r, bytes32 s)
Vulnerabilities not detected

TOKEN FLOW	Tokens in, public

removeNode(uint64 id)
Vulnerabilities not detected

setState(uint64 id, State state)
Vulnerabilities not detected

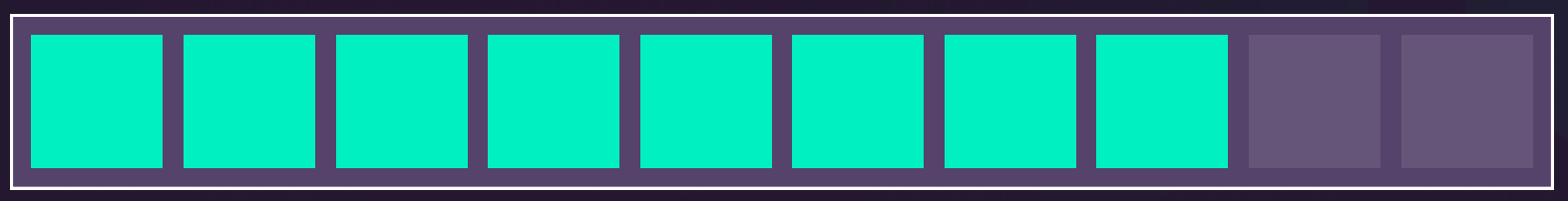
updateNodeSigner(uint64 id, address signer)
Vulnerabilities not detected

updateNodeVersion(uint64 id, uint64 version)
Vulnerabilities not detected

Verification check sums

Contract name	Bytecode hash(SHA 256)
BridgeV2.sol	a26dd74e4a0cb7ede0197fff817d7547a46727cadee1e2f2997d62ffb1525596
EPOA.sol	0744f8043eee14b352ea12cefe100f9c51339515f97e83e6384d87340a51b61b
GovBridgeV2.sol	e17ba555603f02ab7e0f122a20302591f1957d6c41972effb04eeda1bee3f63a
GovBridgeV2Pos.sol	c4dd73deda96fc4ff360b9995e4584a7122724dd28dbe254070aeef13b4fb69d
GateKeeper.sol	498b662e4988adb41cf795c84729527f7c5073ce8b53315abe32fbf3880a1f97
NodeRegistryV2.sol	b6b990dac13369f9d00d15992b6b6a5886ae1a826778868da9d9e1656113e23f
Link to source code:	https://gitlab.digiu.ai/blockchainlaboratory/eywa-cdp/-/tree/refactoring

Project evaluation



8/10

Get in touch 👋

@smartstatetech

@smartstate

@SmartStateAudit

@smartstatetech

@smartstate.tech

View this report on Smartstate.tech

info@smartstate.tech

smartstate.tech