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

CLP smart contracts - are smart contracts for processing synth and burn operations, as well as mint and lock tokens. They are also responsible for swap processing and liquidity handling operations.

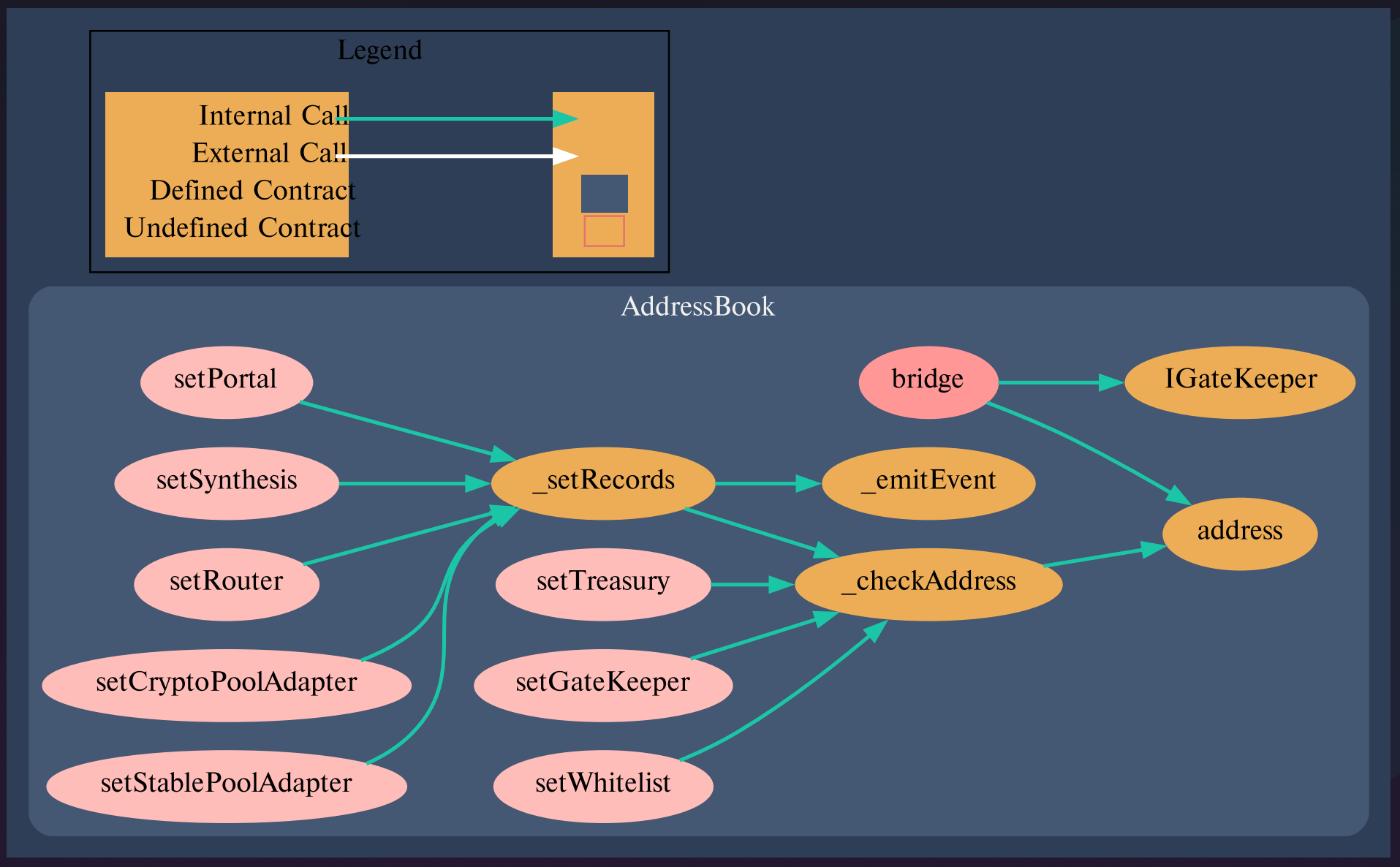
Table of contents

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

AddressBook.sol



pic.1.1 AddressBook.sol

Contract methods analysis:

bridge()
Vulnerabilities not detected

setPortal(Record[] memory records)
Vulnerabilities not detected

setSynthesis(Record[] memory records)
Vulnerabilities not detected

setRouter(Record[] memory records)
Vulnerabilities not detected

setCryptoPoolAdapter(Record[] memory records)
Vulnerabilities not detected

setStablePoolAdapter(Record[] memory records)
Vulnerabilities not detected

setTreasury(address treasury_)
Vulnerabilities not detected

setGateKeeper(address gateKeeper_)
Vulnerabilities not detected

setWhitelist(address whitelist_)
Vulnerabilities not detected

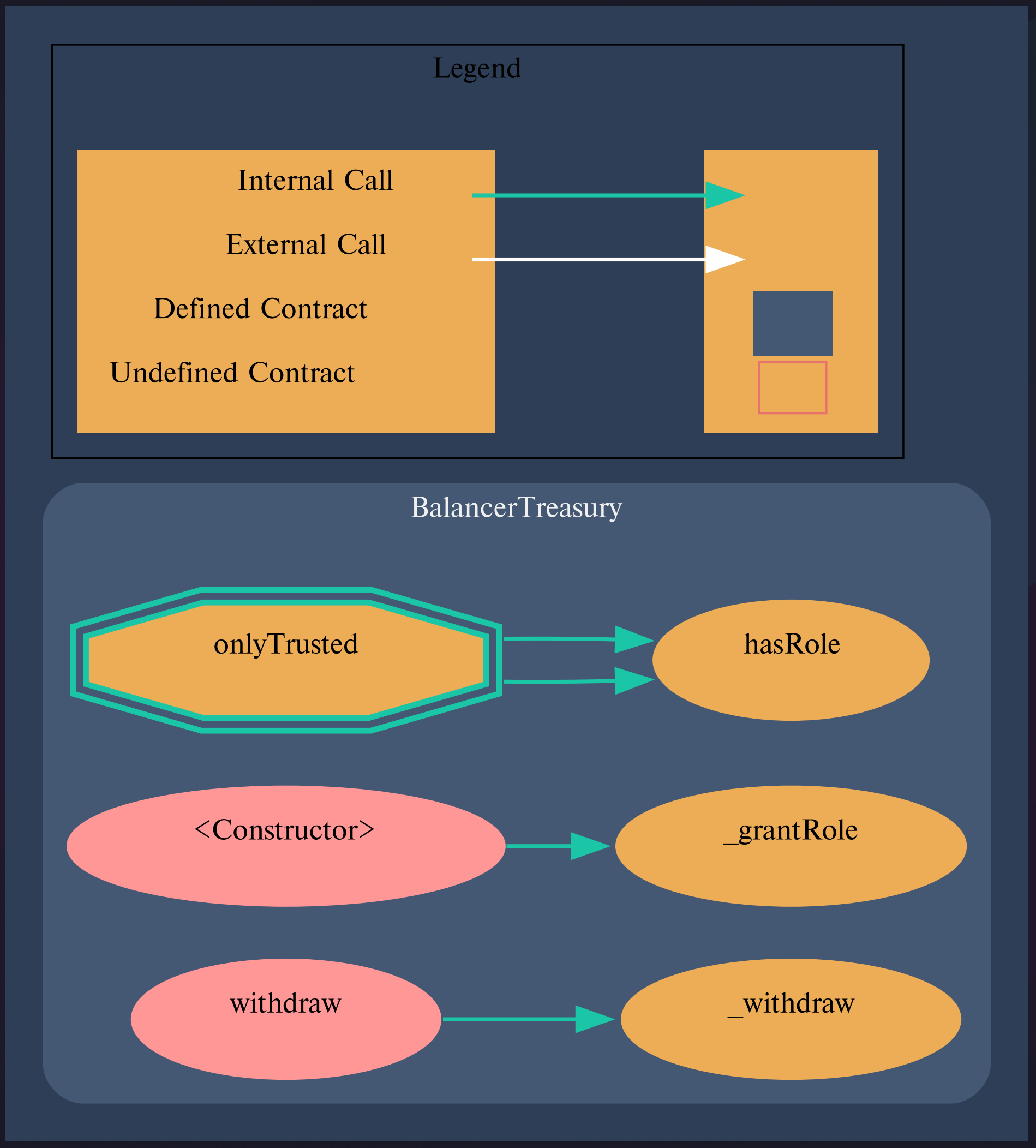
_setRecords(mapping(uint64 => address))
Vulnerabilities not detected

_emitEvent(address endPoint, uint64 chainId, RecordTypes rtype)
Vulnerabilities not detected

_checkAddress(address checkingAddress)
Vulnerabilities not detected

Structure of contract

BalancerTreasury.sol



pic.1.2 BalancerTreasury.sol

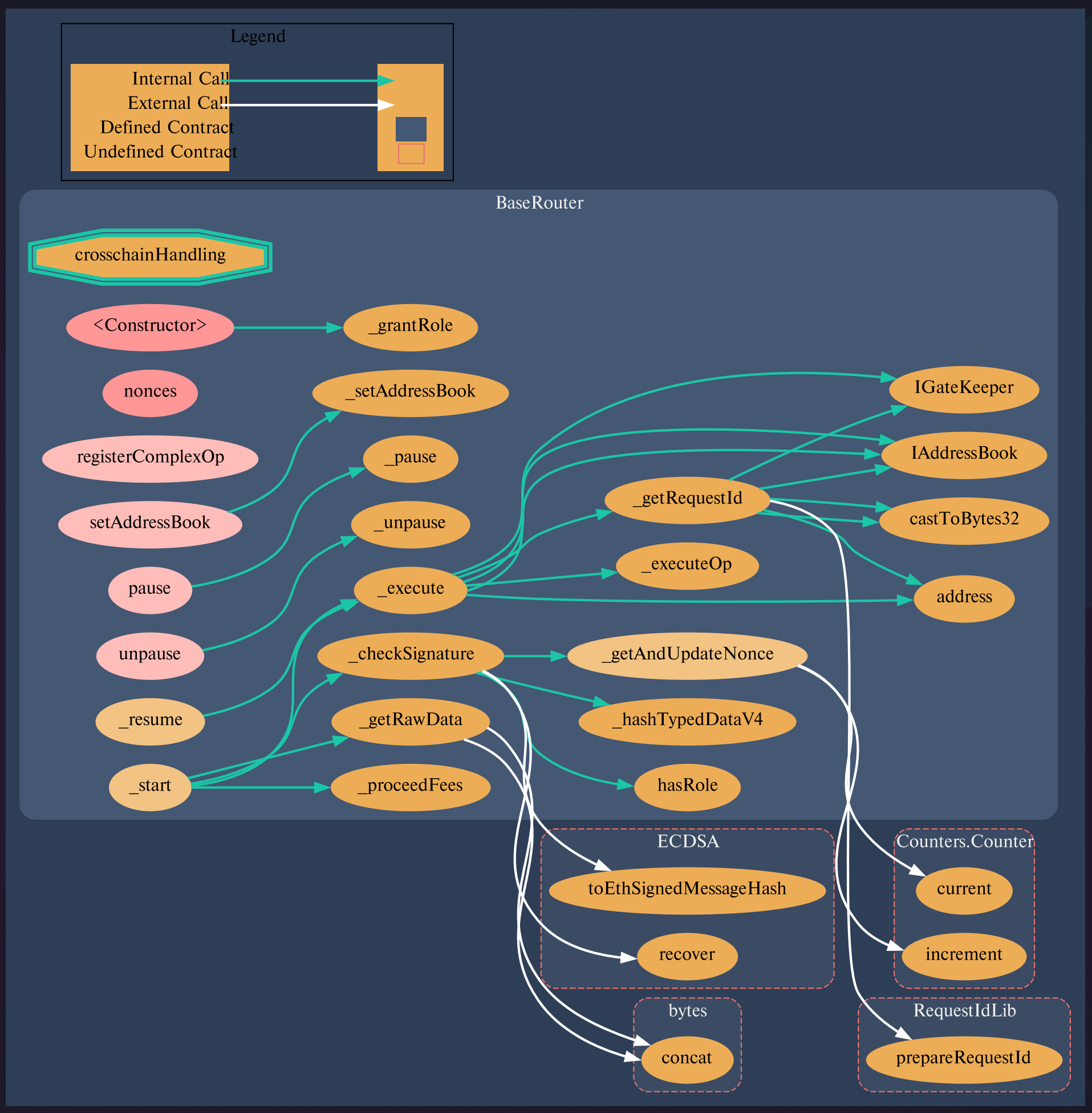
Contract methods analysis:

```
withdraw(
  string calldata reason,
  address token,
  uint256 amount,
  address to
 )
```

Vulnerabilities not detected	
TOKEN FLOW	Tokens out, onlyTrusted

Structure of contract

BaseRouter.sol



pic.1.3 BaseRouter.sol

Contract methods analysis:

nonces(address whose)
Vulnerabilities not detected

registerComplexOp(ComplexOp[] memory complexOps_)
Vulnerabilities not detected

setAddressBook(address addressBook_)
Vulnerabilities not detected

pause()
Vulnerabilities not detected

unpause()
Vulnerabilities not detected

_start(string[] calldata operations, bytes[] memory params, IRouterParams.Invoice calldata receipt)
Vulnerabilities not detected

_resume(bytes32 requestId, uint8 cPos, string[] calldata operations, bytes[] memory params)
Vulnerabilities not detected

_execute(uint256 cPos, string[] calldata operations, bytes[] memory params)
Vulnerabilities not detected

`_getAndUpdateNonce(address whose)`
Vulnerabilities not detected

`_checkSignature(`
`  address from,`
`  bytes32 operationHash,`
`  bytes memory data,`
`  IRouterParams.Invoice calldata receipt`
`  )`

Vulnerabilities not detected

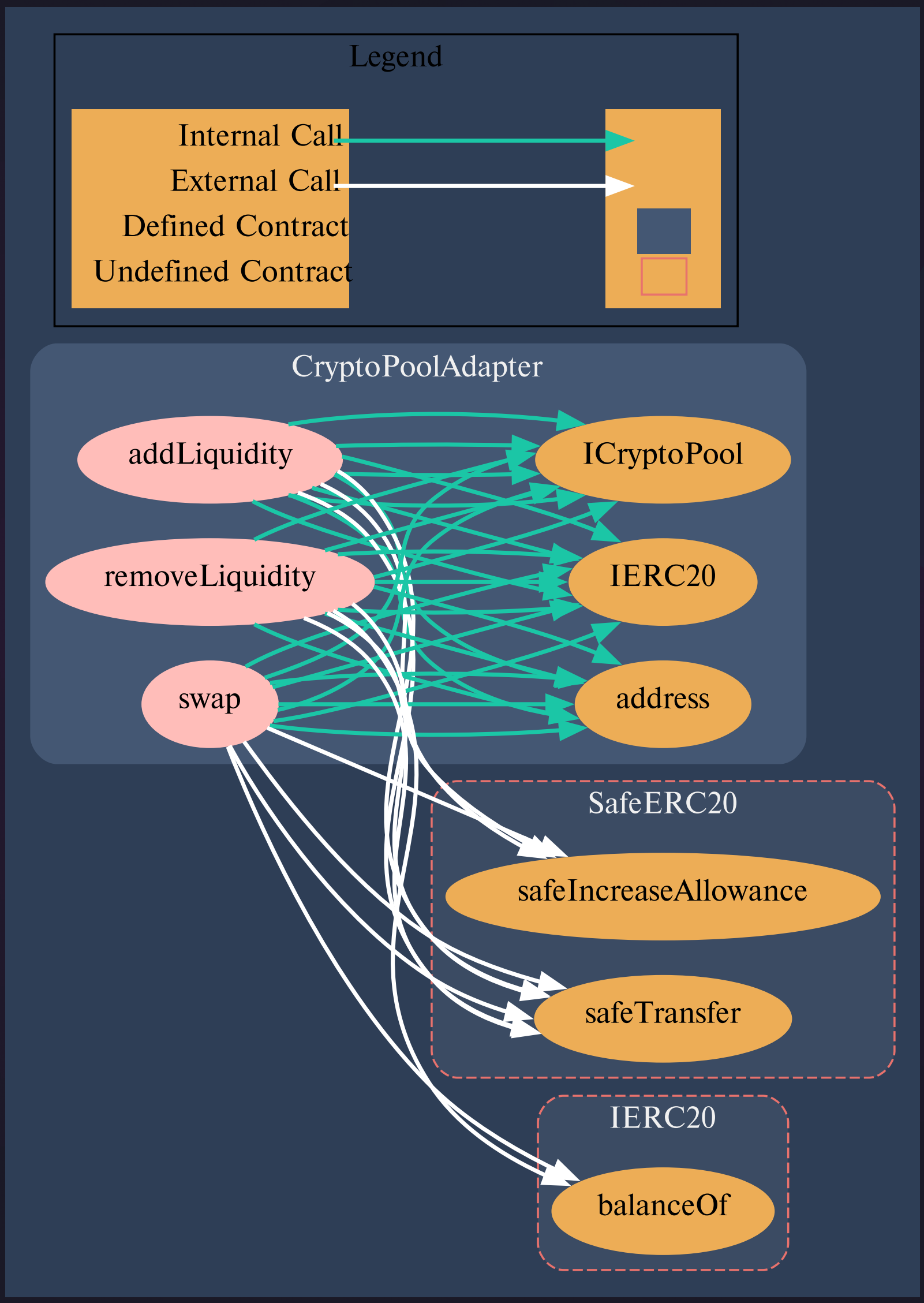
`_getRawData(`
`  string[] calldata operations,`
`  bytes[] memory params`
`  )`

Vulnerabilities not detected

`_getRequestId(address receiver, uint64 chainIdTo)`
Vulnerabilities not detected

Structure of contract

CryptoPoolAdapter.sol



pic.1.4 CryptoPoolAdapter.sol

Contract methods analysis:

```
addLiquidity(
  address pool,
  uint256 amountIn,
  uint256 coinIndex,
  address to,
  uint256 minAmountOut,
  address emergencyTo
 )
```

Vulnerabilities not detected

TOKEN FLOW	Tokens in, tokens out, public

```
swap(
  address tokenIn,
  address pool,
  uint256 i,
  uint256 j,
  address tokenOut,
  address to,
  uint256 minAmountOut,
  address emergencyTo,
  uint256 aggregationFee
 )
```

Vulnerabilities not detected

TOKEN FLOW	Tokens in, tokens out, public

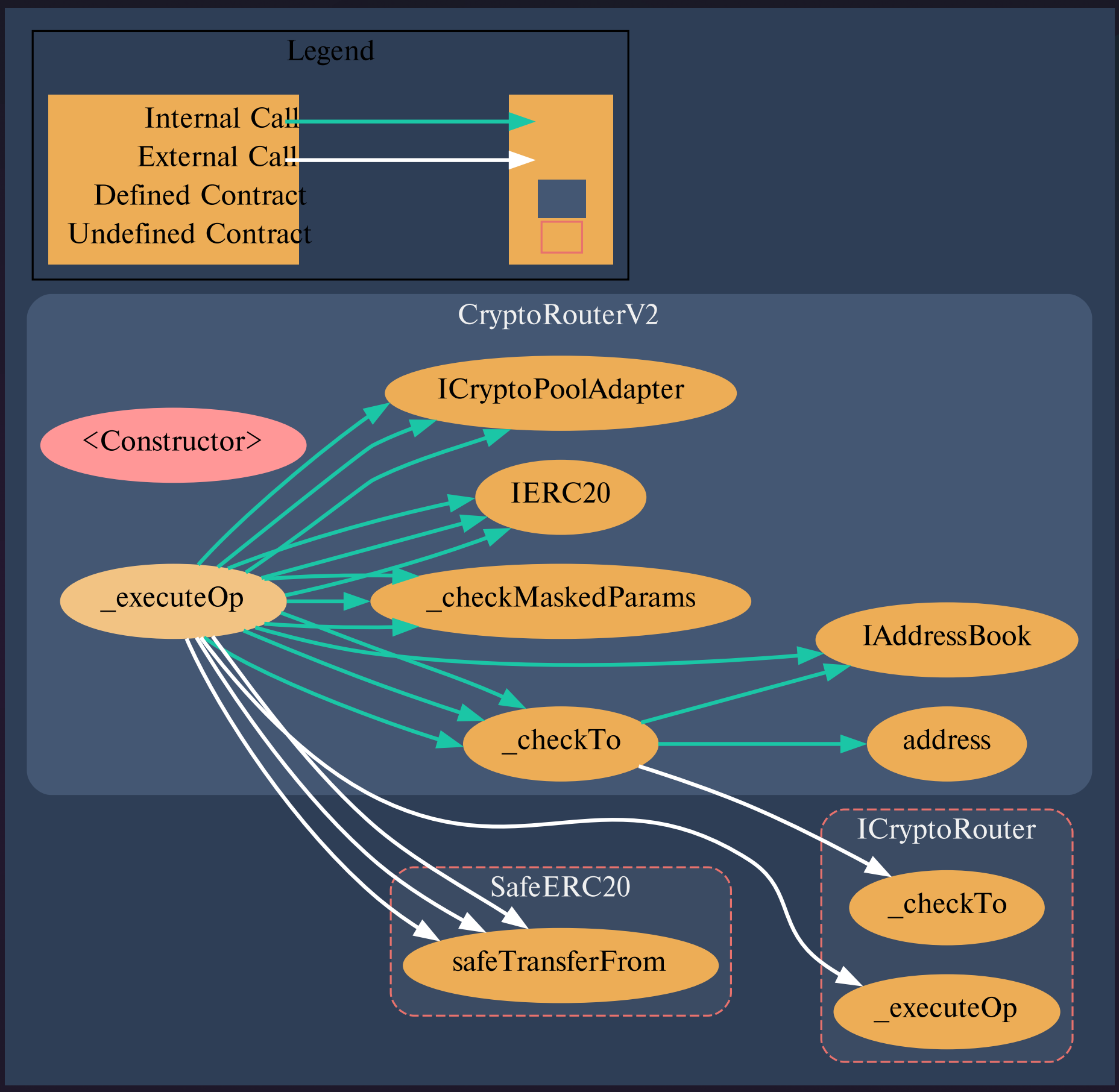
```
removeLiquidity(
  address pool,
  uint256 i,
  address to,
  address tokenOut,
  uint256 minAmountOut,
  address emergencyTo
 )
```

Vulnerabilities not detected

TOKEN FLOW	Tokens in, tokens out, public

Structure of contract

CryptoRouterV2.sol



pic.1.5 CryptoRouterV2.sol

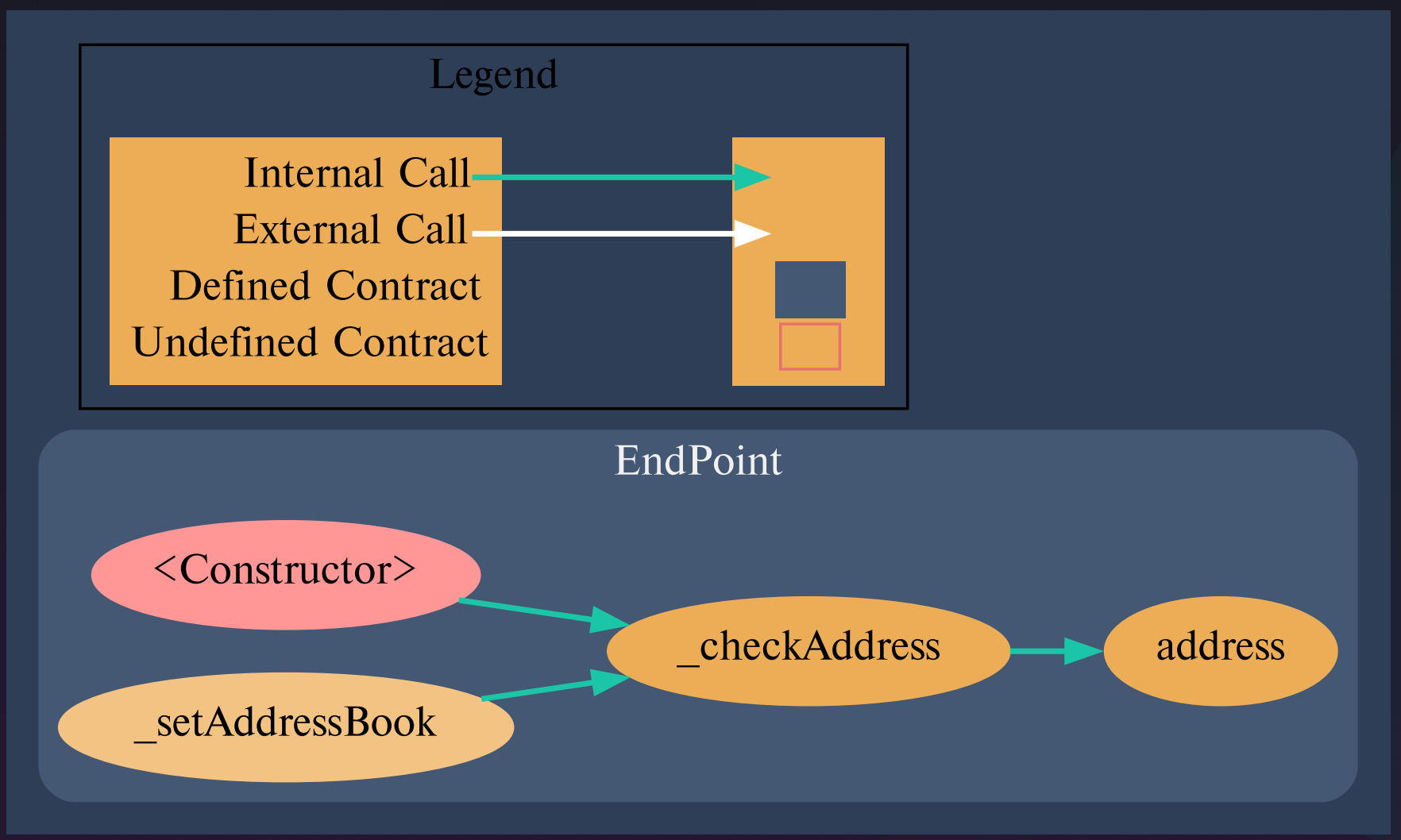
Contract methods analysis:

`_executeOp( bool isOpHalfDone, bytes32 op, bytes32 nextOp, bytes memory params, MaskedParams memory prevMaskedParams )`
Vulnerabilities not detected

`_checkTo(address to, address emergencyTo, uint64 chainId, bytes32 nextOp)`
Vulnerabilities not detected

Structure of contract

EndPoint.sol



pic.1.6 EndPoint.sol

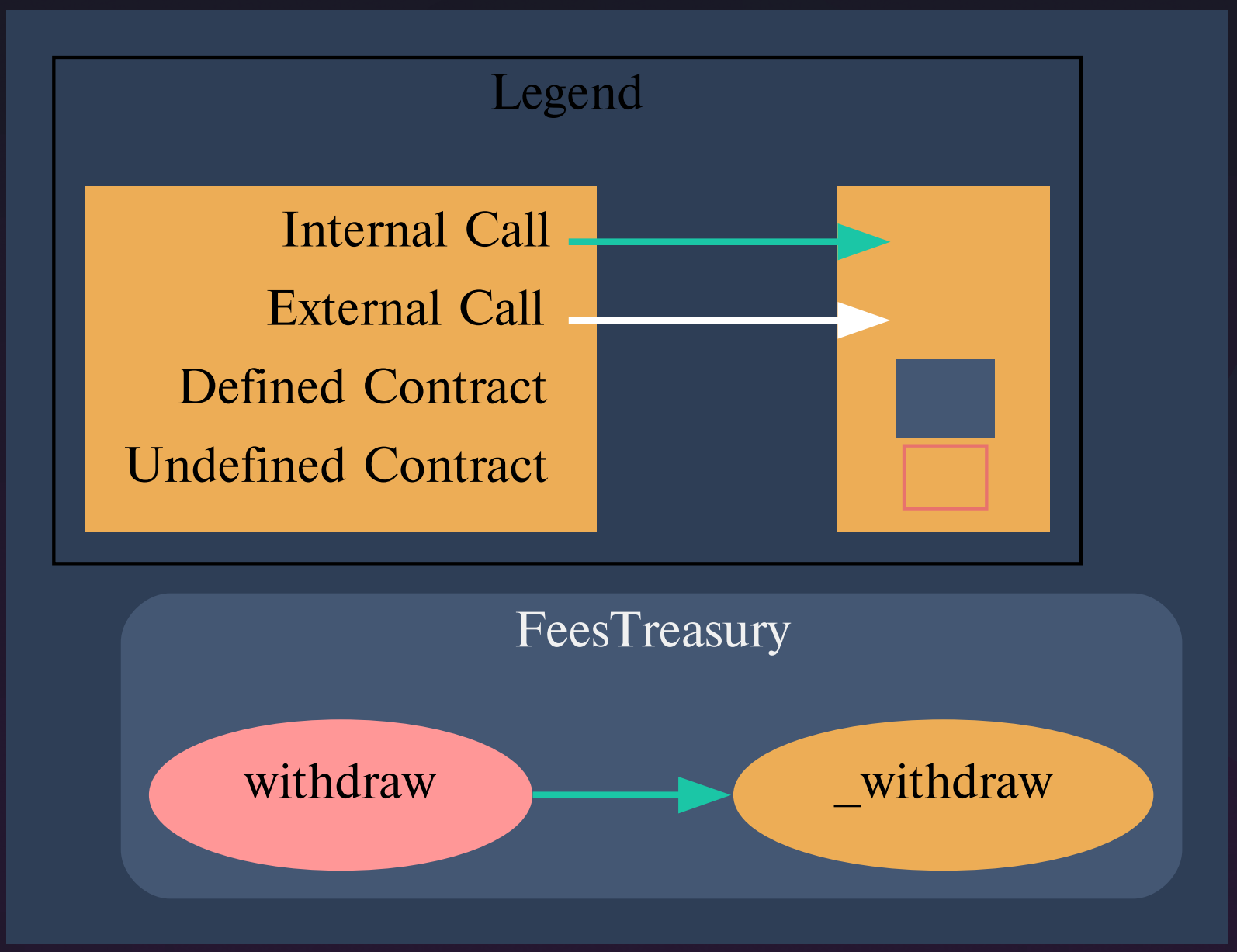
Contract methods analysis:

setAddressBook(address addressBook)
Vulnerabilities not detected

_checkAddress(address checkingAddress)
Vulnerabilities not detected

Structure of contract

FeesTreasury.sol



pic.1.7 FeesTreasury.sol

Contract methods analysis:

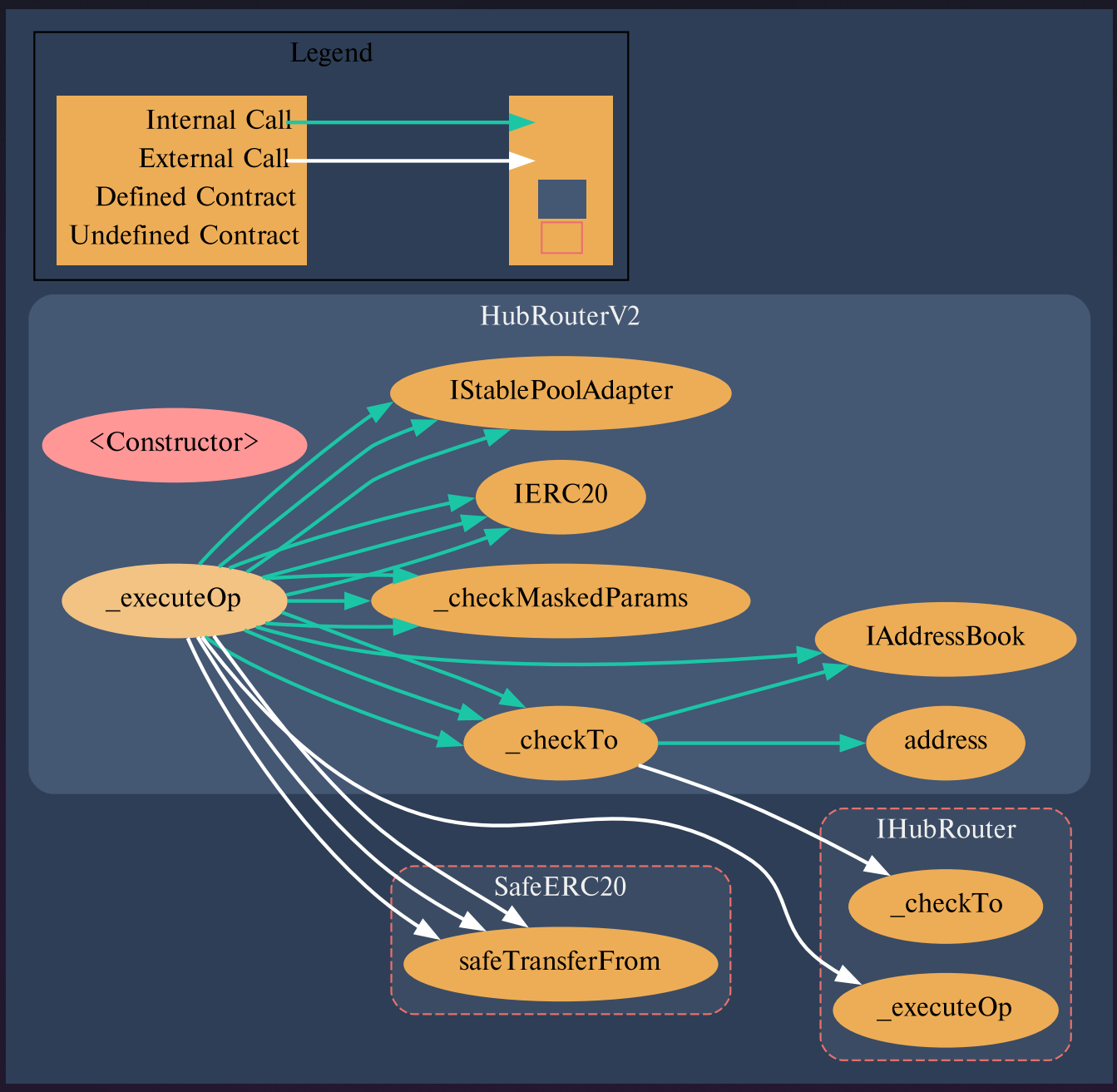
```
function withdraw(
  string calldata reason,
  address token,
  uint256 amount,
  address to
 )
```

Vulnerabilities not detected

TOKEN FLOW	Tokens out, onlyOwner

Structure of contract

HubRouterV2.sol



pic.1.8 HubRouterV2.sol

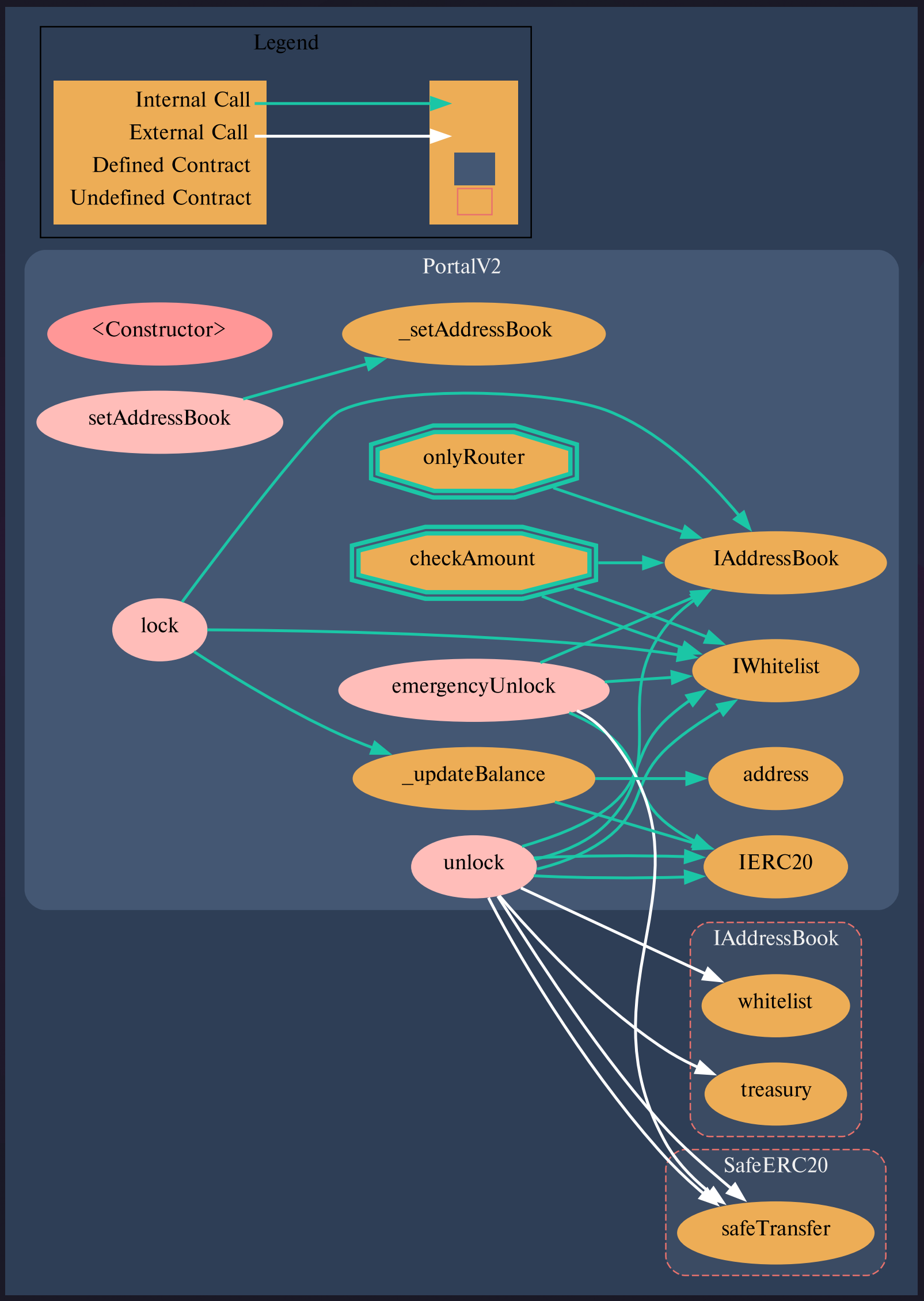
Contract methods analysis:

`_executeOp( bool isOpHalfDone, bytes32 op, bytes32 nextOp, bytes memory params, MaskedParams memory prevMaskedParams )`
Vulnerabilities not detected

`_checkTo(address to, address emergencyTo, uint64 chainId, bytes32 nextOp)`
Vulnerabilities not detected

Structure of contract

PortalV2.sol



pic.1.9 PortalV2.sol

Contract methods analysis:

```
setAddressBook(address
addressBook_)
```

Vulnerabilities not detected

```
lock(
  address token,
  uint256 amount,
  address from,
  address to
 )
```

Vulnerabilities not detected

```
unlock(
  address otoken,
  uint256 amount,
  address from,
  address to
 )
```

Vulnerabilities not detected

```
emergencyUnlock(
  address otoken,
  uint256 amount,
  address from,
  address to
 )
```

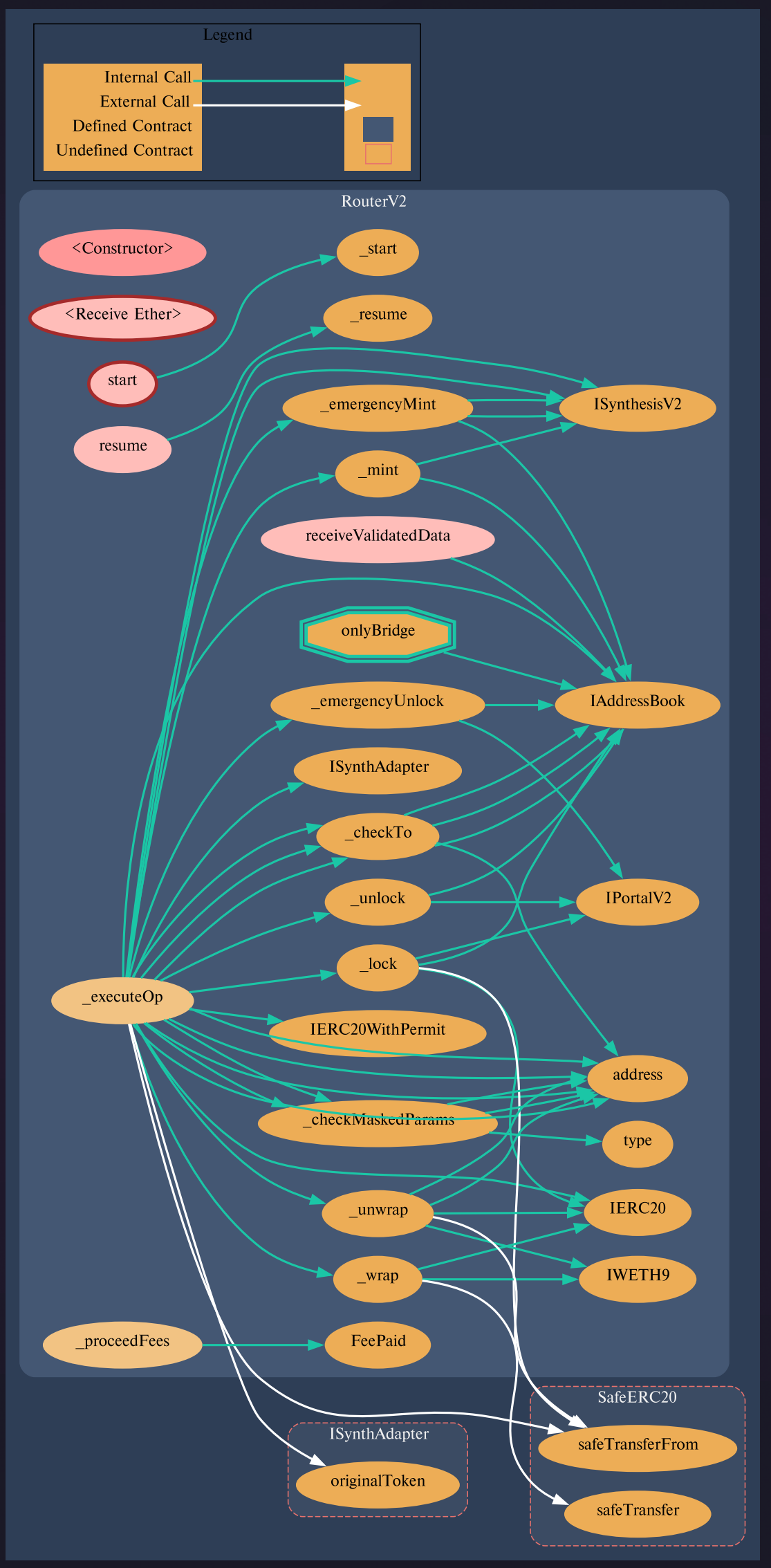
Vulnerabilities not detected

```
_updateBalance(address
token, uint256
expectedAmount)
```

Vulnerabilities not detected

Structure of contract

RouterV2.sol



pic.2.0 RouterV2.sol

Contract methods analysis:

receiveValidatedData(bytes4 selector, address from, uint64 chainIdFrom)
Vulnerabilities not detected

start(string[] calldata operations, bytes[] memory params, Invoice calldata receipt)
Vulnerabilities not detected
TOKEN FLOW: Tokens in, public

resume(bytes32 requestId, uint8 cPos, string[] calldata operations, bytes[] memory params)
Vulnerabilities not detected

_executeOp(bool isOpHalfDone, bytes32 op, bytes32 nextOp, bytes memory params, MaskedParams memory prevMaskedParams)
Vulnerabilities not detected

_lock(SynthParams memory p)
Vulnerabilities not detected

_unlock(SynthParams memory p)
Vulnerabilities not detected

_emergencyUnlock(SynthParams memory p)
Vulnerabilities not detected

_mint(SynthParams memory p)
Vulnerabilities not detected

_emergencyMint(SynthParams memory p)
Vulnerabilities not detected

_wrap(WrapParams memory p)
Vulnerabilities not detected

`_unwrap(WrapParams memory p)`
Vulnerabilities not detected

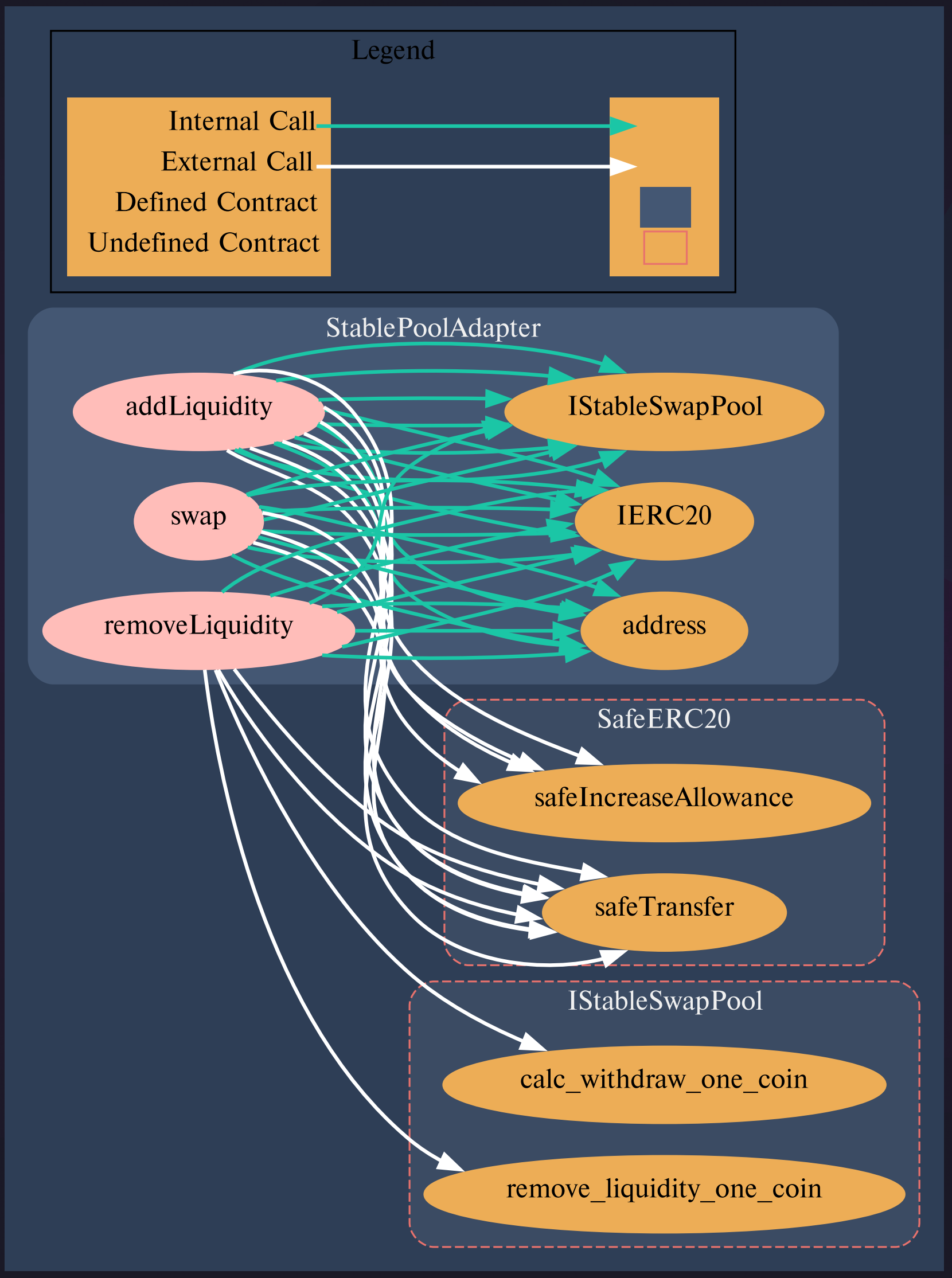
`_proceedFees(uint256 executionPrice, address accountant)`
Vulnerabilities not detected

`_checkMaskedParams( uint256 currentAmountIn, address currentFrom, address currentEmergencyTo, MaskedParams memory prevMaskedParams )`
Vulnerabilities not detected

`_checkTo(address to, address emergencyTo, uint64 chainId, bytes32 nextOp)`
Vulnerabilities not detected

Structure of contract

StablePoolAdapter.sol



pic.2.1 StablePoolAdapter.sol

Contract methods analysis:

```
addLiquidity(
  address pool,
  address tokenIn,
  uint256 amountIn,
  uint256 minAmountOut,
  uint256 coinIndex,
  address to,
  uint256 coinsCount,
  address emergencyTo
 )
```

Vulnerabilities not detected

TOKEN FLOW	Tokens in, tokens out, public

```
swap(
  address tokenIn,
  address pool,
  int128 i,
  int128 j,
  address tokenOut,
  address to,
  uint256 minAmountOut,
  address emergencyTo
 )
```

Vulnerabilities not detected

TOKEN FLOW	Tokens in, tokens out, public

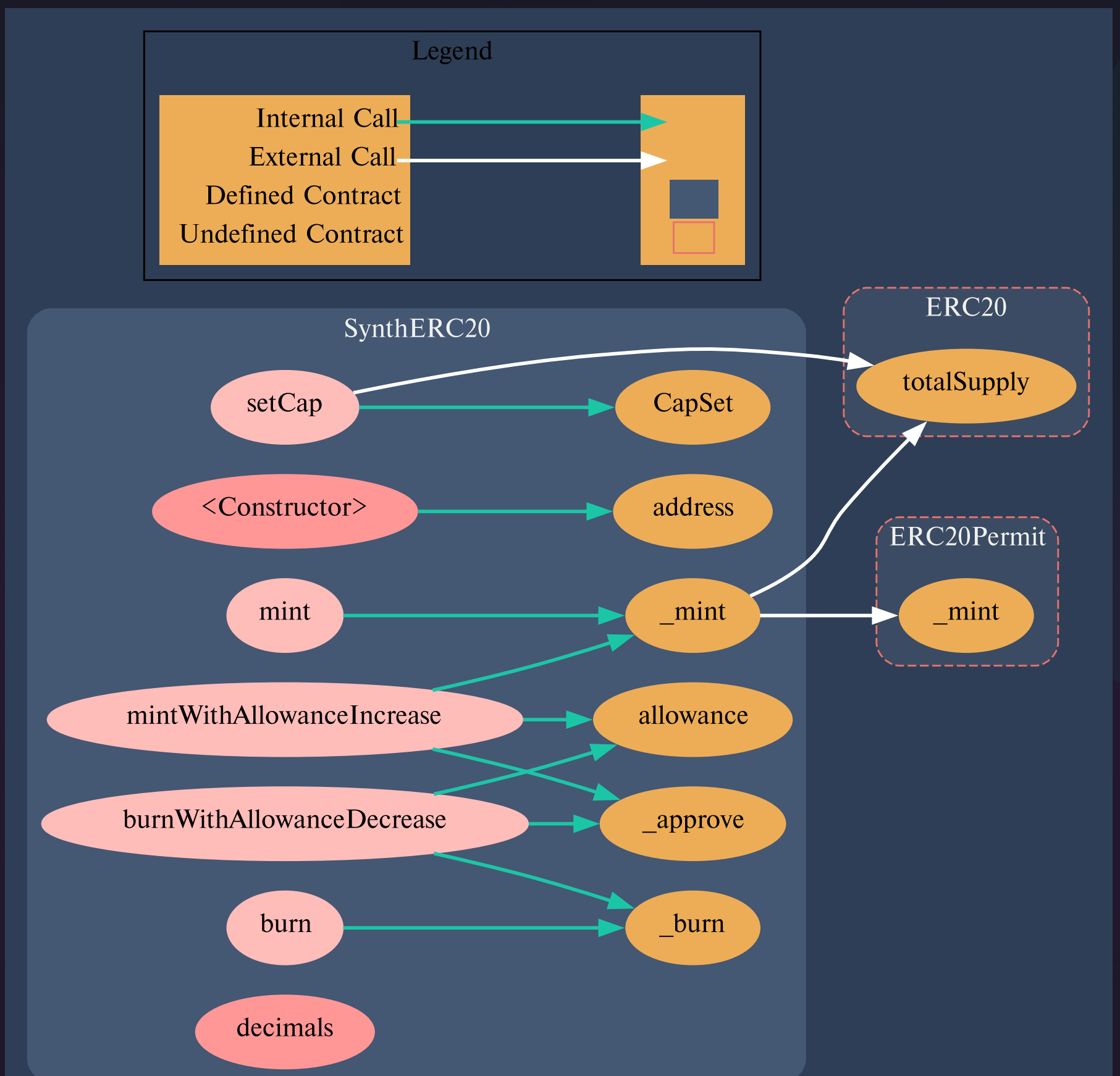
```
removeLiquidity(
  address pool,
  int128 i,
  address to,
  address tokenOut,
  uint256 minAmountOut,
  address emergencyTo
 )
```

Vulnerabilities not detected

TOKEN FLOW	Tokens in, tokens out, public

Structure of contract

SynthERC20.sol



pic.2.2 SynthERC20.sol

Contract methods analysis:

setCap(uint256 cap_)

Vulnerabilities not detected

mint(address account, uint256 amount)

Vulnerabilities not detected

mintWithAllowanceIncrease(
address account,
address spender,
uint256 amount
)

Vulnerabilities not detected

burn(address account, uint256 amount)

Vulnerabilities not detected

burnWithAllowanceDecrease(
address account,
address spender,
uint256 amount
)

Vulnerabilities not detected

decimals()

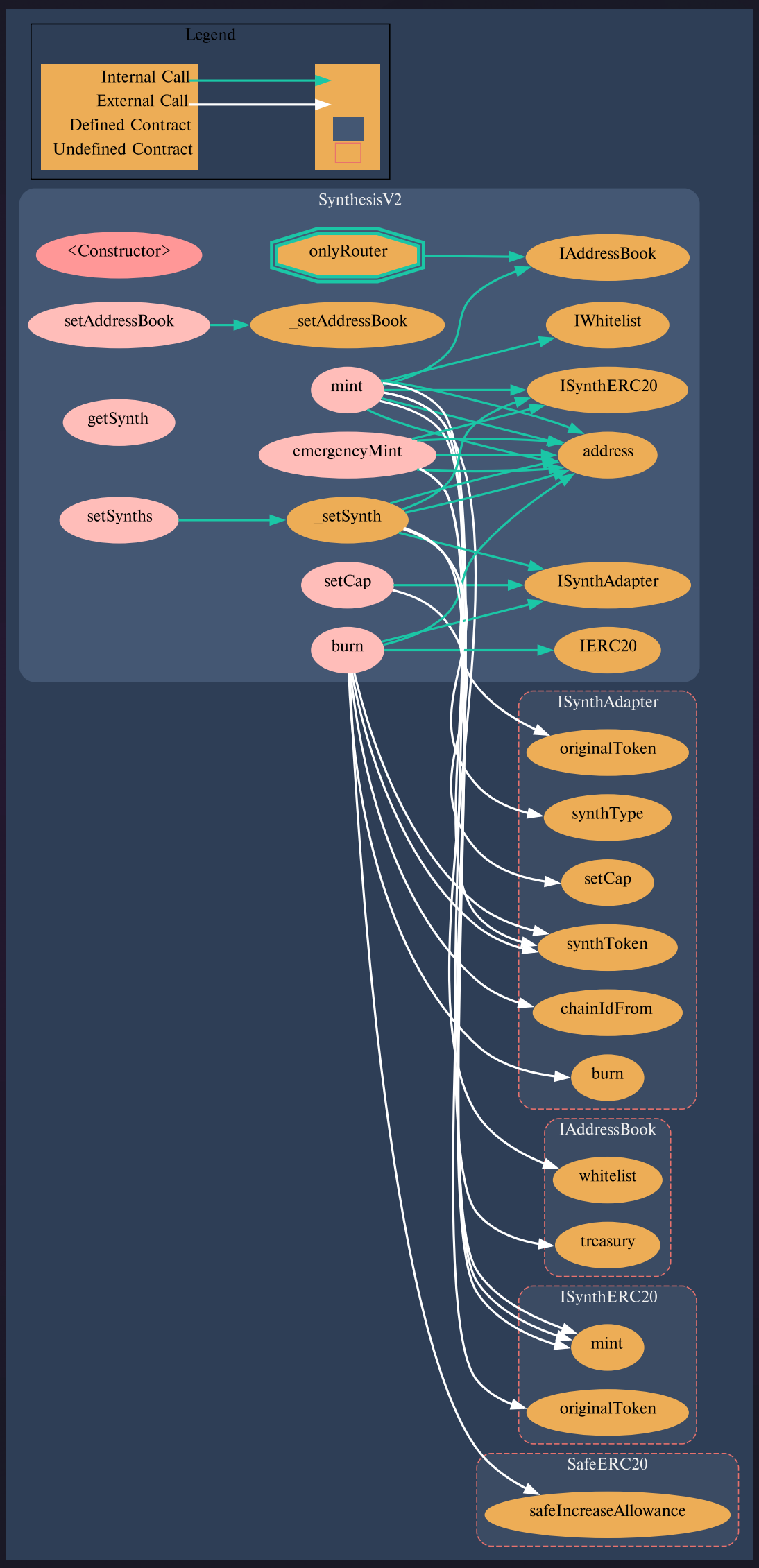
Vulnerabilities not detected

_mint(address account, uint256 amount)

Vulnerabilities not detected

Structure of contract

SynthesisV2.sol



pic.2.3 SynthesisV2.sol

Contract methods analysis:

`setAddressBook(address addressBook_)`
Vulnerabilities not detected

`setCap(address token, uint256 cap_)`
Vulnerabilities not detected

`getSynth(address otoken_)`
Vulnerabilities not detected

`mint(`
`  address otoken,`
`  uint256 amount,`
`  address from,`
`  address to`
`)`

Vulnerabilities not detected

`emergencyMint(`
`  address stoken,`
`  uint256 amount,`
`  address from,`
`  address to`
`)`

Vulnerabilities not detected

`burn(`
`  address stoken,`
`  uint256 amount,`
`  address from,`
`  address to,`
`  uint64 chainIdTo`
`)`

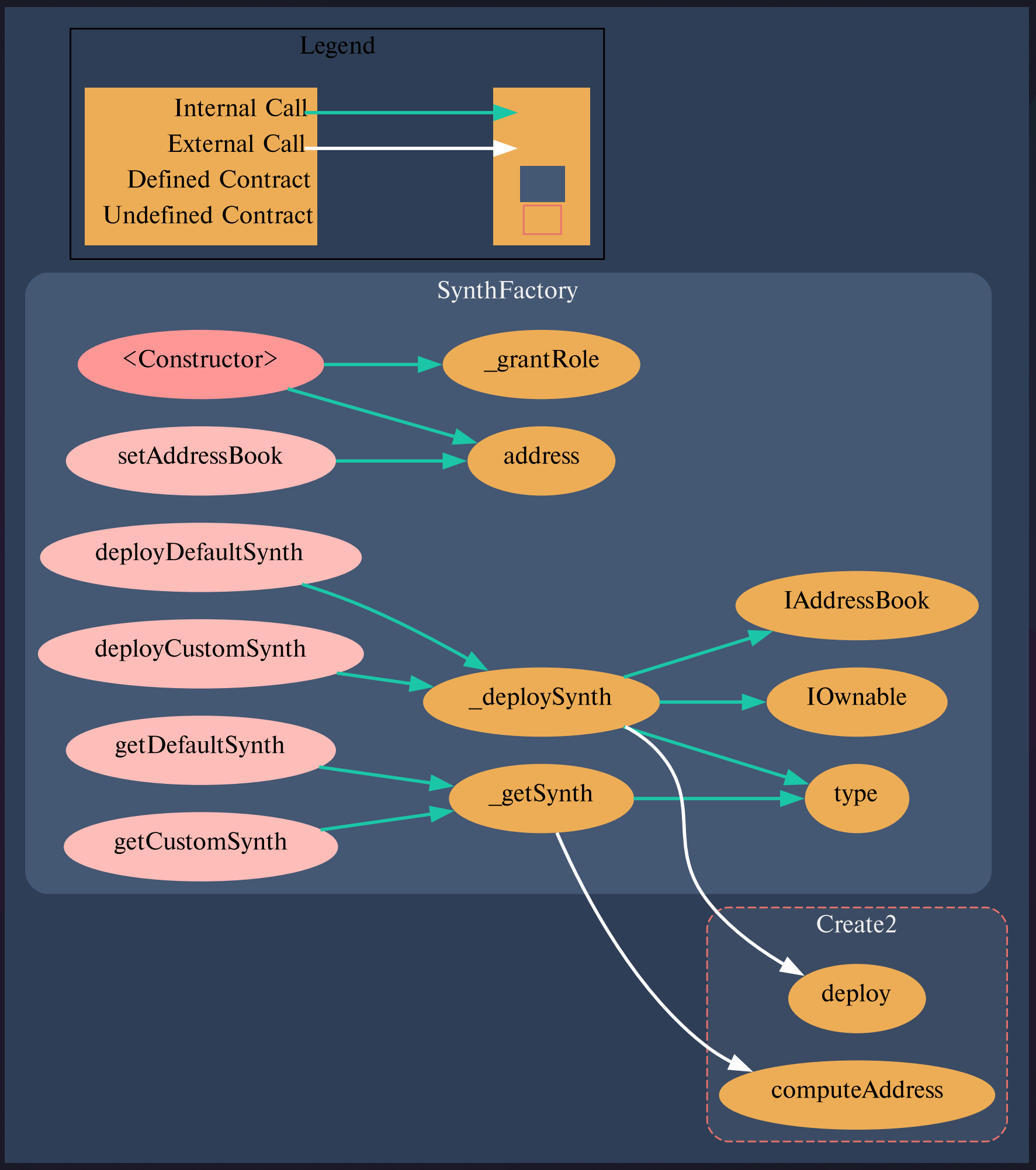
Vulnerabilities not detected

`setSynths(address[] calldata stokens)`
Vulnerabilities not detected

`_setSynth(address stoken_)`
Vulnerabilities not detected

Structure of contract

SynthFactory.sol



pic.2.4 SynthFactory.sol

Contract methods analysis:

```
setAddressBook(address
addressBook_)
```

Vulnerabilities not detected

```
getCustomSynth(
  address originalToken_,
  uint8 decimals_,
  string memory name_,
  string memory symbol_,
  uint64 chainIdFrom_,
  string memory
  chainSymbolFrom_
 )
```

Vulnerabilities not detected

```
getDefaultSynth(
  address originalToken_,
  uint8 decimals_,
  string memory
  originalName_,
  string memory
  originalSymbol_,
  uint64 chainIdFrom_,
  string memory
  chainSymbolFrom_
 )
```

Vulnerabilities not detected

```
deployCustomSynth(
  address originalToken_,
  uint8 decimals_,
  string memory name_,
  string memory symbol_,
  uint64 chainIdFrom_,
  string memory
  chainSymbolFrom_
 )
```

Vulnerabilities not detected

```
deployDefaultSynth(
  address originalToken_,
  uint8 decimals_,
  string memory
  originalName_,
  string memory
  originalSymbol_,
  uint64 chainIdFrom_,
  string memory
  chainSymbolFrom_
 )
```

Vulnerabilities not detected

```
_getSynth(
  address originalToken_,
  uint8 decimals_,
  string memory name_,
  string memory symbol_,
  uint64 chainIdFrom_,
  string memory
  chainSymbolFrom_,
  ISynthAdapter.SynthType
  synthType_
 )
```

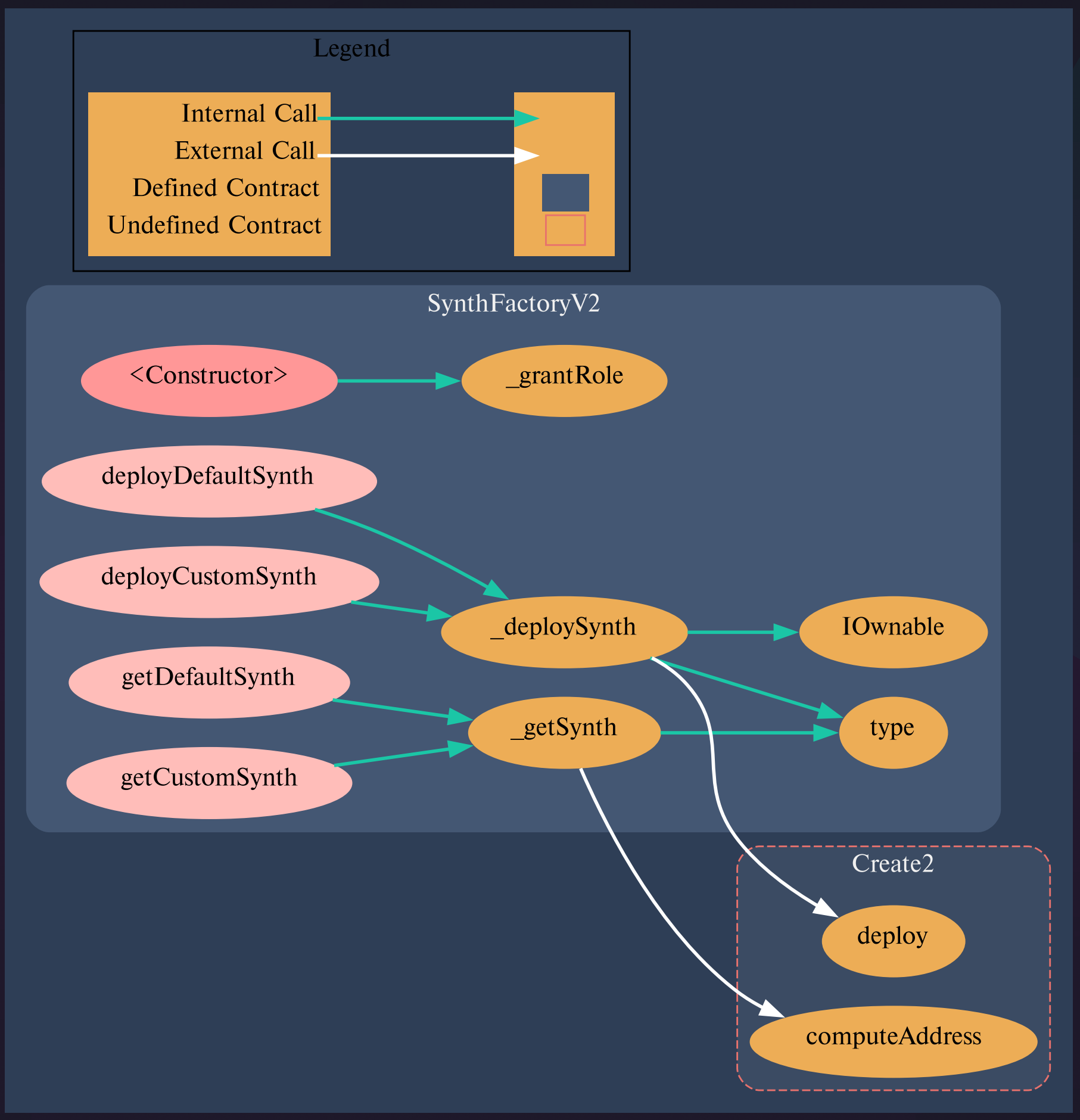
Vulnerabilities not detected

```
_deploySynth(
  address originalToken_,
  uint8 decimals_,
  string memory name_,
  string memory symbol_,
  uint64 chainIdFrom_,
  string memory
  chainSymbolFrom_,
  ISynthAdapter.SynthType
  synthType_
 )
```

Vulnerabilities not detected

Structure of contract

SynthFactoryV2.sol



pic.2.5 SynthFactoryV2.sol

Contract methods analysis:

```
getCustomSynth(
  address originalToken_,
  uint8 decimals_,
  string memory name_,
  string memory symbol_,
  uint64 chainIdFrom_,
  string memory
  chainSymbolFrom_
 )
```

Vulnerabilities not detected

```
getDefaultSynth(
  address originalToken_,
  uint8 decimals_,
  string memory
  originalName_,
  string memory
  originalSymbol_,
  uint64 chainIdFrom_,
  string memory
  chainSymbolFrom_
 )
```

Vulnerabilities not detected

```
deployCustomSynth(
  address originalToken_,
  uint8 decimals_,
  string memory name_,
  string memory symbol_,
  uint64 chainIdFrom_,
  string memory
  chainSymbolFrom_
 )
```

Vulnerabilities not detected

```
deployDefaultSynth(
  address originalToken_,
  uint8 decimals_,
  string memory
  originalName_,
  string memory
  originalSymbol_,
  uint64 chainIdFrom_,
  string memory
  chainSymbolFrom_
 )
```

Vulnerabilities not detected

```
_getSynth(
  address originalToken_,
  uint8 decimals_,
  string memory name_,
  string memory symbol_,
  uint64 chainIdFrom_,
  string memory
  chainSymbolFrom_,
  ISynthAdapter.SynthType
  synthType_
 )
```

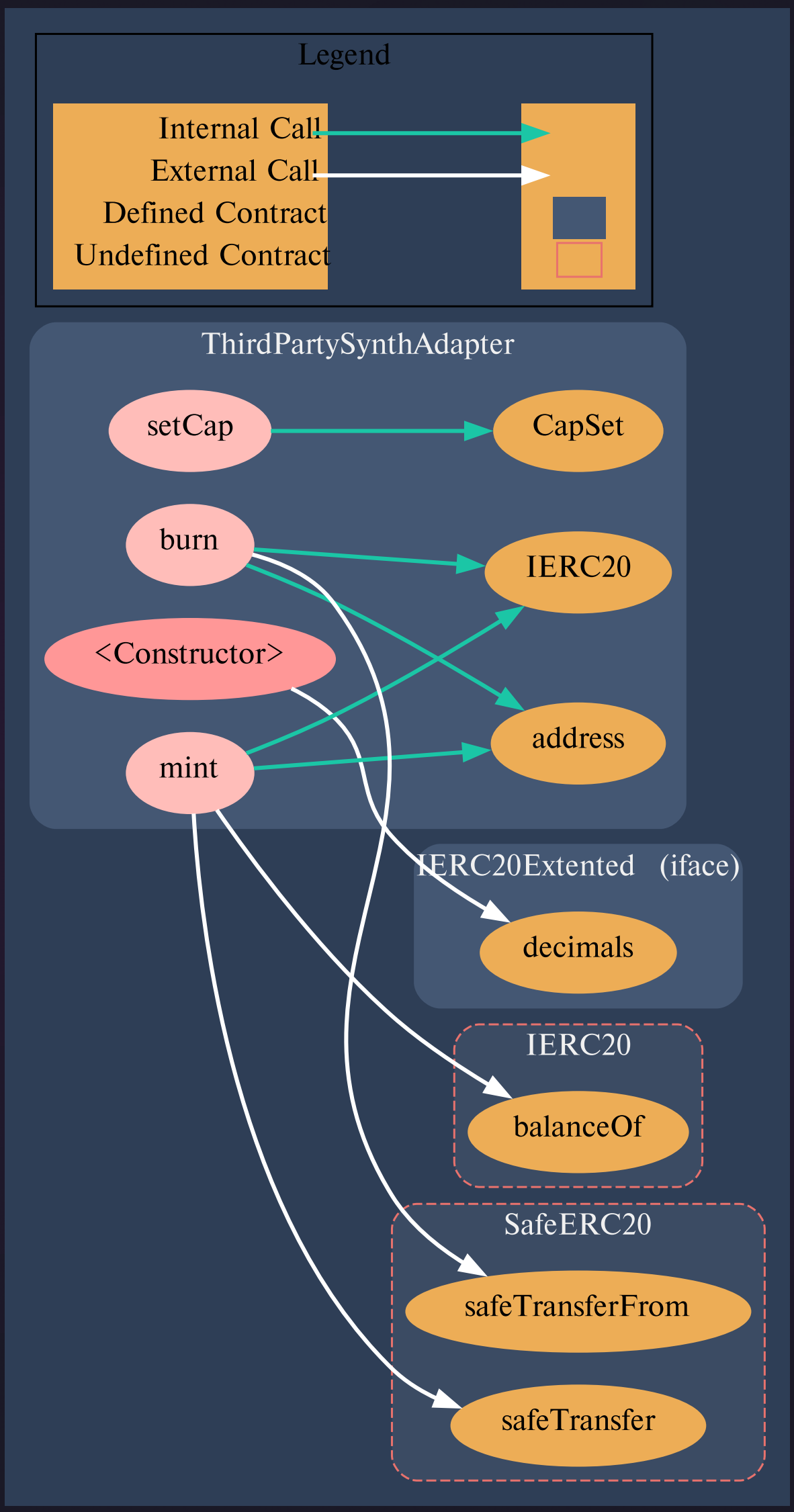
Vulnerabilities not detected

```
_deploySynth(
  address originalToken_,
  uint8 decimals_,
  string memory name_,
  string memory symbol_,
  uint64 chainIdFrom_,
  string memory
  chainSymbolFrom_,
  ISynthAdapter.SynthType
  synthType_
 )
```

Vulnerabilities not detected

Structure of contract

ThirdPartySynthAdapter.sol



pic.2.6 ThirdPartySynthAdapter.sol

Contract methods analysis:

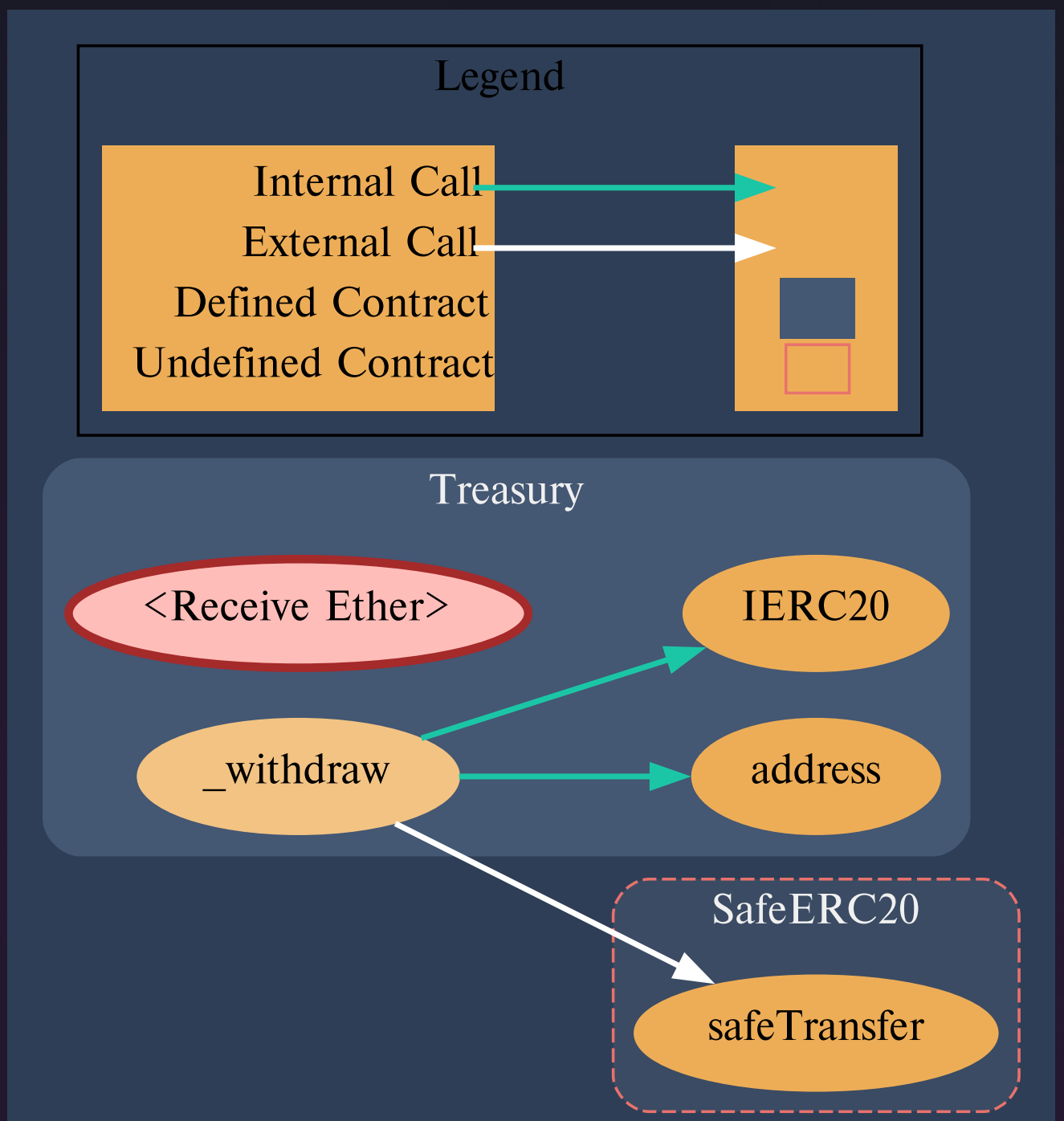
setCap(uint256 cap_)
Vulnerabilities not detected

mint(address account, uint256 amount)
Vulnerabilities not detected

burn(address account, uint256 amount)
Vulnerabilities not detected

Structure of contract

Treasury.sol



pic.2.7 Treasury.sol

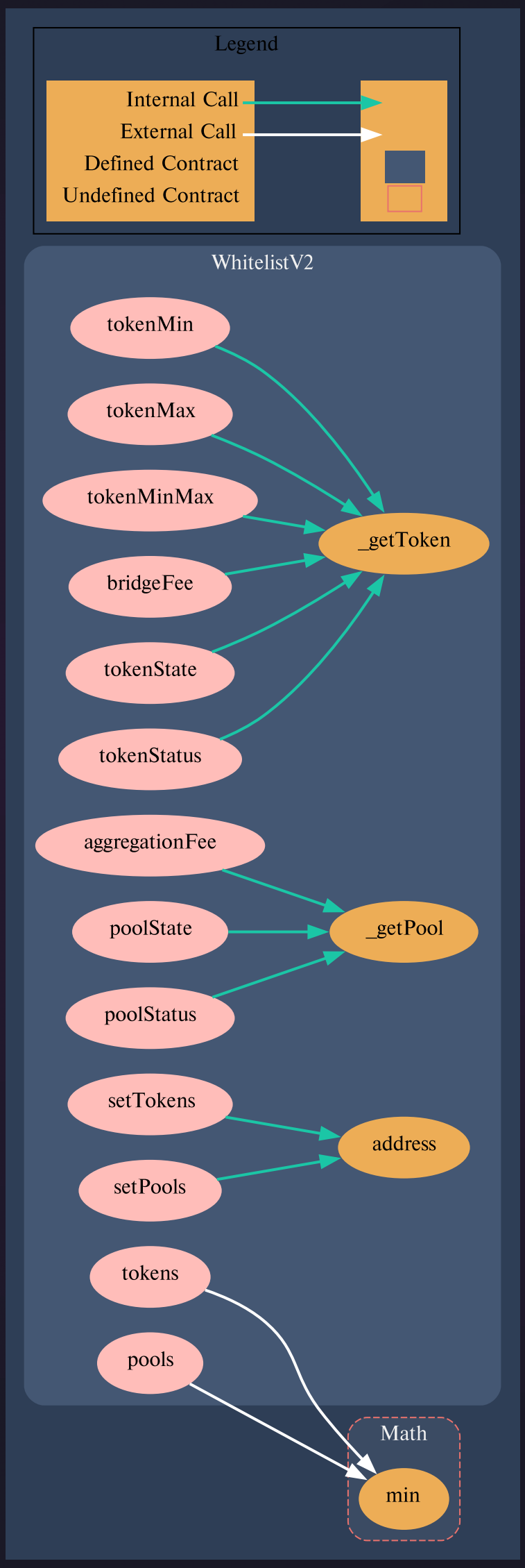
Contract methods analysis:

```
_withdraw(
  string calldata reason,
  address token,
  uint256 amount,
  address to
 )
```

Vulnerabilities not detected

Structure of contract

Whitelist.sol



pic.2.8 Whitelist.sol

Contract methods analysis:

tokenMin(address token_)
Vulnerabilities not detected

tokenMax(address token_)
Vulnerabilities not detected

tokenMinMax(address token_)
Vulnerabilities not detected

bridgeFee(address token_)
Vulnerabilities not detected

tokenState(address token_)
Vulnerabilities not detected

tokenStatus(address token_)
Vulnerabilities not detected

aggregationFee(address pool_)
Vulnerabilities not detected

poolState(address pool_)
Vulnerabilities not detected

poolStatus(address pool_)
Vulnerabilities not detected

tokens(uint256 offset, uint256 count)
Vulnerabilities not detected

pools(uint256 offset, uint256 count)
Vulnerabilities not detected

setTokens(IWhitelist.TokenStatus[] memory tokens_)
Vulnerabilities not detected

setPools(IWhitelist.PoolStatus[] memory pools_)
Vulnerabilities not detected

_getToken(address token)
Vulnerabilities not detected

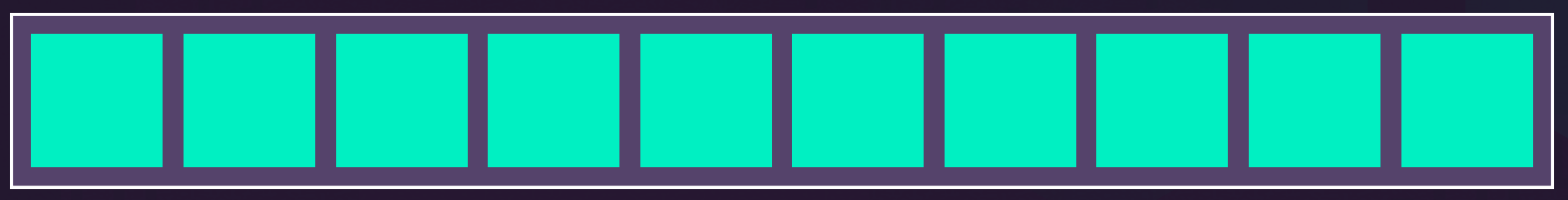
_getPool(address pool)
Vulnerabilities not detected

Verification check sums

Contract name	Bytecode hash(SHA 256)
AddressBook.sol	a12568a198a2acd26c9089d2180dbed7739db1e9e82f686c30fba9104858ba46
BalancerTreasury.sol	cfcc8ebed6259e7480e2be9c1df46b1917d12d32d0346f85af3818eb2b5db67c
BaseRouter.sol	842d5f7f72c8d10068c452a0ca4e38cae33b5a4f5726939269560526562af26c
CryptoPoolAdapter.sol	5fadb8a5cbc580d67cb5ffa95921444ccfbcd25519535bc5de4ed4f5e3bfbd27
CryptoRouterV2.sol	f3dd77c6615f20f36f11dcb6c215288ae20465f290acdec3abe19177be841f16
EndPoint.sol	5229616b8e95be7e0667c76f838868a87b16885df5f2a02ea3f8d96229054e9e
FeesTreasury.sol	3a2b6415d3bc209fc3c4bb871a09366006e99342170fe4432295335770 96ab1d
HubRouterV2.sol	48a75076fb7b0945cd62db341566e0e78a9e41b399c0586e633bc1587cabd24a
PortalV2.sol	9f43ac8e6f945527c8649df7f8cf32cda2290e66ed50b607f4b32d09710c4544
RouterV2.sol	dcf79bbfe6f7848b8983ece51789a0c9df7767fd69f4b9adfadabcd51db0091a
StablePoolAdapter.sol	00a878721931791fd1f8578f9b22d0163be2e6185c004fe0191f20404c141134
SynthERC20.sol	f7ab642c068d3b05fcfc93f24e82ef0ad53e9099ca0f9b074af1dd249f13fff8
SynthesisV2.sol	c6a149caac427c12e7395252757991d55cef9b4c33cd42a2352733b631d9c457
SynthFactory.sol	2cbecdabb4472abd5e53d060a9f9debd5f1735ac34f0bd91d48baa65cbe70d78

Contract name	Bytecode hash(SHA 256)
SynthFactoryV2.sol	3be7b66ff217a1de082f33232ced87d84956418a73fca998711751b03c7da9fe
ThirdPartySynthAdapter.sol	dd77a214bea69f01b8e8b66ad2e870d0d5a01394cb16275c6ea65bef407192df
Treasury.sol	f5d773331fe933f04e66a0a97fe54009c52bc7aeb896b678a1f761c46fc00a06
Whitelist.sol	39cd1ec1209db25323ec168e52a40a1d0ce08006d95c9831d33f97482e87dd5b
Link to source code:	https://gitlab.digiu.ai/blockchainlaboratory/eywa-cdp/-/tree/refactoring

Project evaluation



10/10

Get in touch 👋

@smartstatetech

@smartstate

@SmartStateAudit

@smartstatetech

@smartstate.tech

View this report on Smartstate.tech

info@smartstate.tech

smartstate.tech