# smart state

new generation of
smart contract audit

booster pool

# Booster Pool

Aug 23

2022

# TABLE OF CONTENTS

# METHODOLOGY

## MAIN TESTS LIST:

- 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

BOOSTERPOOL.SOL

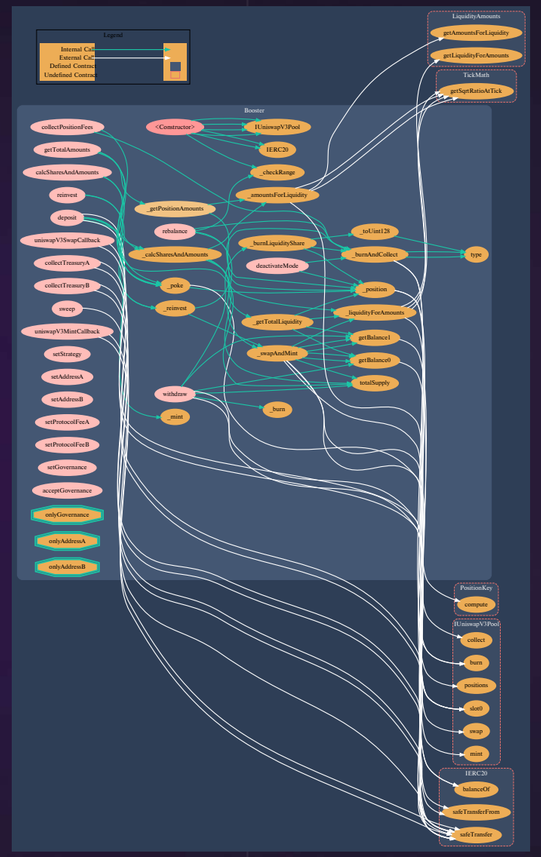
## CONTRACT METHODS ANALYSIS:

**TOKEN FLOW**

- deposit(
  uint256 amount0Desired,
  uint256 amount1Desired,
  uint256 amount0Min,
  uint256 amount1Min,
  address to)
  Vulnerabilities not detected

Tokens in, public

- _poke(int24 tickLower, int24 tickUpper)
  Vulnerabilities not detected




Pic. 1.1
BoosterPool.sol

- _calcSharesAndAmounts(uint256 amount0Desired, uint256 amount1Desired)
  Vulnerabilities not detected

**TOKEN FLOW**

- withdraw(
  uint256 shares,
  uint256 amount0Min,
  uint256 amount1Min,
  address to
  )
  Vulnerabilities not detected

Tokens out, public

- _burnLiquidityShare(
  int24 tickLower,
  int24 tickUpper,
  uint256 shares,
  uint256 BPtotalSupply
  )
  Vulnerabilities not detected

- reinvest(
  int256 swapAmount,
  uint160 sqrtPriceLimitX96
  )
  Vulnerabilities not detected

- _reinvest(
  int256 swapAmount,
  uint160 sqrtPriceLimitX96
  )
  Vulnerabilities not detected

- rebalance(
  int256 swapAmount,
  uint160 sqrtPriceLimitX96,
  int24 tickLower,
  int24 tickUpper
  )
  Vulnerabilities not detected

- _swapAndMint(
  int256 swapAmount,
  uint160 sqrtPriceLimitX96,
  int24 _baseLower,
  int24 _baseUpper
  )
  Vulnerabilities not detected

- _checkRange(int24 tickLower, int24 tickUpper, int24 _tickSpacing)
  Vulnerabilities not detected

- _burnAndCollect(
  int24 tickLower,
  int24 tickUpper,
  uint128 liquidity
  )
  Vulnerabilities not detected

- _getTotalLiquidity() internal view returns (uint128 liquidity)
  Vulnerabilities not detected

- _getPositionAmounts()
  Vulnerabilities not detected

- getTotalAmounts(uint256 amountBP)
  Vulnerabilities not detected

- collectPositionFees()
  Vulnerabilities not detected

- getBalance0() public view returns (uint256)
  Vulnerabilities not detected

- getBalance1() public view returns (uint256)
  Vulnerabilities not detected

- `_position(int24 tickLower, int24 tickUpper)`
  Vulnerabilities not detected

- `_amountsForLiquidity(`
  `int24 tickLower,`
  `int24 tickUpper,`
  `uint128 liquidity`
  `)`
  Vulnerabilities not detected

- `_liquidityForAmounts(`
  `int24 tickLower,`
  `int24 tickUpper,`
  `uint256 amount0,`
  `uint256 amount1`
  `)`
  Vulnerabilities not detected

- `_toUint128(uint256 x) internal pure returns (uint128)`
  Vulnerabilities not detected

- `uniswapV3MintCallback(`
  `uint256 amount0,`
  `uint256 amount1,`
  `bytes calldata data`
  `)`
  Vulnerabilities not detected

- `uniswapV3SwapCallback(`
  `int256 amount0Delta,`
  `int256 amount1Delta,`
  `bytes calldata data`
  `)`
  Vulnerabilities not detected

**TOKEN FLOW**

- collectTreasuryA(
  uint256 amount0,
  uint256 amount1,
  address to
  )
  Vulnerabilities not detected

Tokens out, onlyAdressA

**TOKEN FLOW**

- collectTreasuryB(
  uint256 amount0,
  uint256 amount1,
  address to
  )
  Vulnerabilities not detected

Tokens out, onlyAdressB

**TOKEN FLOW**

- sweep(
  IERC20 token,
  uint256 amount,
  address to
  )
  Vulnerabilities not detected

Tokens out, onlyGovernance

- setStrategy(address newStrategy)
  Vulnerabilities not detected
- setAddressA(address newAddressA)
  Vulnerabilities not detected
- setAddressB(address newAddressB)
  Vulnerabilities not detected
- setProtocolFeeA(uint256 newProtocolFeeA)
  Vulnerabilities not detected

- `setProtocolFeeB(uint256 newProtocolFeeB)`
  Vulnerabilities not detected

- `deactivateMode()`
  Vulnerabilities not detected

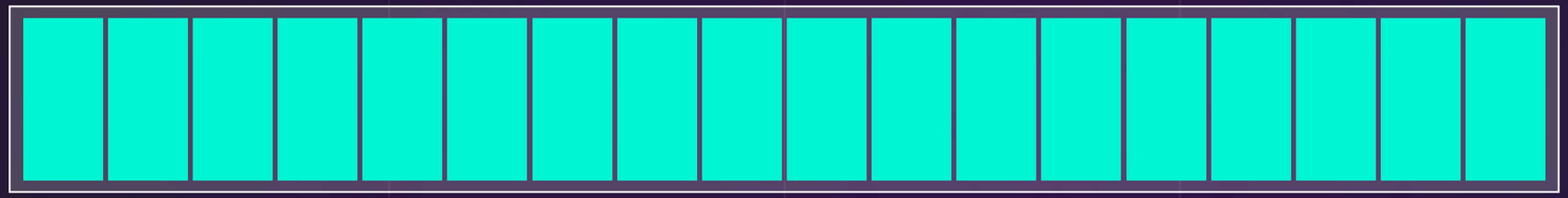
- `setGovernance(address newGovernance)`
  Vulnerabilities not detected

- `acceptGovernance()`
  Vulnerabilities not detected

# VERIFICATION CHECK SUMS

| Contract Name | Solc version | Optimization | Bytecode hash (SHA 256) |
|---|---|---|---|
| BoosterPool.sol | 0.8.7 | 200 | fdab9fd94b64d83a83ecc45a79f2db625e7a193b37af96102d1a172bf3345e39 |

# PROJECT EVALUATION



## 10/10

smart state

# GET IN TOUCH

info@smartstate.tech
smartstate.tech

View this report on smartstate.tech