smart state

> Smart
Contract
Audit #

TABLE OF CONTENTS

METHODOLOGY

MAIN TESTS LIST:

- Best code practices
- ERC20/BEP20 compliance (if applicable)
- Logical bugs
- General Denial Of Service(DOS)
- Locked ether
- Private data leaks
- Using components with known vulns
- Weak PRNG
- Unsed vars
- Uncheked 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

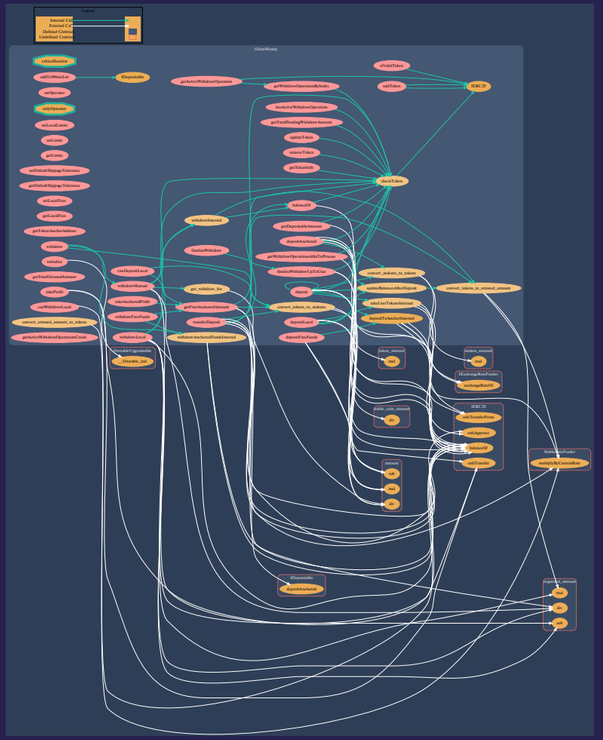
ORIONMONEY.SOL

CONTRACT METHODS ANALYSIS:

- initialize(uint256 deposit_limit, uint256 withdraw_limit)
  Function can be declared external

- setOperator(address operator, uint8 access)
  Function should be declared external.
  Function should emit an event

- checkToken(IERC20 token)
  Vulnerabilities not detected



Pic. 1.1.
OrionMoney.sol

- addToken(IERC20 token,
  IERC20 proxy_token,
  IERC20 anchored_token,
  IConversionPool conversion_pool,
  IExchangeRateFeeder exchange_rate_feeder,
  StableRateFeeder orion_rate_feeder,
  uint32 decimals)

  Function should be declared external. proxy_token agument is redundant and shoud be removed.

- updateToken(IERC20 token,
  IERC20 proxy_token,
  IERC20 anchored_token,
  IConversionPool conversion_pool,
  IExchangeRateFeeder exchange_rate_feeder,
  StableRateFeeder orion_rate_feeder,
  uint32 decimals)

  Function should be declared external. proxy_token agument is redundant and shoud be removed.

- buyItem(uint256 saleID)
  Function can be declared external.
  Recommended to put sl.sold = true before transfer eth in order to avoid potential reentrancy. Recommended to add NonReentrant modifier.

- removeToken(IERC20 token)
  Function should be declared external.

- setLocalLimits(uint256 local_deposit_limit, uint256 local_withdraw_limit)
  Function should be declared external.
  Function should emit an even.

- setLimits(uint256 deposit_limit, uint256 withdraw_limit)
  Function should be declared external.
  Function should emit an event.

- getLimits()
  Function should be declared external

- setDefaultSlippageTolerance(uint256 new_tolerance)
  Function should be declared external.
  Function should emit an event.

- getDefaultSlippageTolerance()
  Function should be declared external.

- setLocalFees(uint256 new_local_deposit_fee, uint256 new_local_withdraw_fee)
  Function should be declared external.
  Function should emit an event

- getLocalFees()
  Function should be declared external

- isValidToken(IERC20 token)
  Function should be declared external

- getTokenAnchorAddress(IERC20 token)
  Function should be declared external

- getTokenInfo(IERC20 token)
  Function should be declared external

- addToWhiteList(IDepositable receiver, uint8 action)
  Function should be declared external.
  Function should emit an event

- balanceOf(IERC20 token, address user)
  Vulnerabilities not detected

- convert_atokens_to_tokens(IERC20 token, uint256 atoken_amount)
  In order to avoid precision loss it,s recommended to perform all multiplication operations before division operations.

- convert_tokens_to_atokens(IERC20 token, uint256 token_amount)
  In order to avoid precision loss it,s recommended to perform all multiplication operations before division operations.

- convert_tokens_to_orioned_amount(IERC20 token, uint256 requested_amount)
  Vulnerabilities not detected

- convert_orioned_amount_to_tokens(IERC20 token, uint256 orioned_amount)
  Vulnerabilities not detected

- getFreeAnchoredAmount(IERC20 token)
  Vulnerabilities not detected

- getTotalOrionedAmount(IERC20 token)
  Function should be declared external

- takeAnchoredProfit(IERC20 token, address receiver, uint256 amount)
  Function should be declared external.
  Function should emit an event

- getDepositableAmount(IERC20 token)
  Function should be declared external

- takeProfit(IERC20 token, address receiver, uint256 amount)
  Function should be declared external.
  Function should emit an event

- depositFreeFunds(IERC20 token, uint256 amount)
  Function should be declared external
- depositToAnchorInternal(IERC20 token, uint256 amount, uint256 min_amount)
  Vulnerabilities not detected
- withdrawAnchoredFundsInternal(IERC20 token, uint256 anchored_amount)
  Vulnerabilities not detected
- withdrawFreeFunds(IERC20 token, uint256 anchored_amount)
  Function should be declared external
- canDepositLocal(IERC20 token, uint256 amount)
  Function should be declared external
- updateBalancesAfterDeposit(IERC20 token, uint256 real_token_amount)
  Vulnerabilities not detected
- takeUserTokensInternal(IERC20 token, uint256 amount)
  Vulnerabilities not detected
- depositLocal(IERC20 token, uint256 amount)
  Function should be declared external

- deposit(IERC20 token, uint256 amount, uint256 min_amount)
  Vulnerabilities not detected
- depositAnchored(IERC20 token, address depositor, uint256 anchored_amount)
  Vulnerabilities not detected
- deposit(IERC20 token, uint256 amount)
  Function should be declared external
- canWithdrawLocal(IERC20 token, uint256 amount)
  Function should be declared external.
  Recommended to use SafeMath for math operations.
- withdrawInternal(IERC20 token, uint256 requested_amount)
  Vulnerabilities not detected
- withdrawLocal(IERC20 token, uint256 requested_amount)
  Function should be declared external
- get_withdraw_fee(IERC20 token, uint256 amount)
  Vulnerabilities not detected
- transferDeposit(IERC20 token, IDepositable receiver, uint256 requested_amount)
  Function should be declared external

- finalizeWithdrawUpToUser(IERC20 token, address stop_address)
  In order to optimize gas usage consider saving _withdraw_operations[token].length into variable instead of referencing to it in every loop iteration. Function should be declared external

- finalizeWithdraw(IERC20 token)
  Function should be declared external

- getActiveWithdrawOperationsCount(IERC20 token)
  Function should be declared external

- getWithdrawOperationsAbleToProcess(IERC20 token)
  Function should be declared external

- getWithdrawOperationByIndex(IERC20 token, uint idx)
  Vulnerabilities not detected

- getActiveWithdrawOperation(IERC20 for_token, address for_user)
  Function should be declared external

- getActiveWithdrawOperation(IERC20 for_token, address for_user)
  Function should be declared external

- hasActiveWithdrawOperation(IERC20 token, address user)
  Function should be declared external

- getTotalPendingWithdrawAmount(IERC20 token)
  Function should be declared external

Get In Touch

info@smartstate.tech

smartstate.tech